Laboratório VISGRAF

Instituto de Matemática Pura e Aplicada

Stochastic Sampling for Coordinate Neural Networks

Diana Aldana, Daniel Perazzo
Tiago Novello, Luiz Velho (supervisors)

Technical Report TR-23-03 Relatório Técnico

December - 2023 - Dezembro

STOCHASTIC SAMPLING FOR COORDINATE NEURAL NETWORKS

DIANA ALDANA, DANIEL PERAZZO
IMPA

Abstract. A fundamental process in signal representation is the sampling choice, as it may determine the reconstruction's properties and the representation's precision. Classic image sampling is done regularly on the domain, matching the capture devices' architecture, and theory is already well established in the machine learning area. However, regularly sample on the domain is only sometimes possible or convenient. This work aims to shed some light on stochastic sampling, analyze the different choices to sample, construct analog structures from the regular case, and use our implementations on the specific application of implicit curves.

1. Introduction

An important task in computer vision is media representation, as a good proxy may showcase the signal's features (such as critical points, frequency content, details, for example) or be used as input for further processing. In order to solve the task, a hypothesis is made about the space the signal belongs to and the media is then sampled in a base of said space, creating a discrete representation of the signal. Then, a dual basis is used to transform those samples into an approximation of the original media.

A common choice for the representation basis is known as unit impulse basis, which takes evenly spaced points in the media object's domain and obtains the media's values on those coordinates. This kind of sampling is known as regular sampling, and it has been widely studied for many sensory systems are constructed based on the same principle. For example, in Figure 1, we can see an example with regular sampling.

This sampling, however, has certain disadvantages:

- As enunciated on Shannon Theorem [7], the signal reconstructed from a regular sampling with N samples will have frequency band of $\frac{N}{2}$ (this is also called the Nyquist Limit). This, however, indicates that higher frequencies can only be obtained if denser sampling is done. This results from the

Date: December 8, 2023
E-mail address: daniel.perazzo@impa.br, diana.aldana@impa.br.
Key words and phrases. Sinusoidal Neural Networks, Sampling, Implicit Representations.

uniformity of samples over the domain, so stochastic sampling may be free of such a problem.
- When the space the signal belongs to has a high dimension, regular sampling becomes prohibitively expensive to compute. Then, aided by techniques such as ray tracing or path tracing, it is possible to use fewer stochastic samples to reconstruct the media object.

In the context of implicit neural representations, when training coordinate-based models such as the ones described by [10], [11] or [9], the coordinates used during training strongly impact the subsequent reconstruction of the signal. One such example is present in [6], where coordinates are carefully selected to allow the model to represent surfaces in real-time, a feat that may be impossible by training the model with coordinates uniformly spaced in the domain.

Based on the previous considerations, the objective of this report shall be the analysis of the representations given by models trained with different kinds of stochastic samples. Section 2 will give an overview of different kinds of stochastic samplings and comparisons with the regular sampling. Section 2 will explore the usual structures defined for the model in [11] and the creation of similar structures for the stochastic case. Then, an alternative to such structure is raised in section 4, considering two filters in training: Gaussian and bilateral.

Section 5 will consider a more careful choice of the training coordinates by using SIFT and Harris feature extractors and analyze an adaptive scheme to sample coordinates using the Warnock algorithm. Finally, Section 6 will present an application of stochastic sampling to reconstruct implicit curves similarly to [6] and [10].

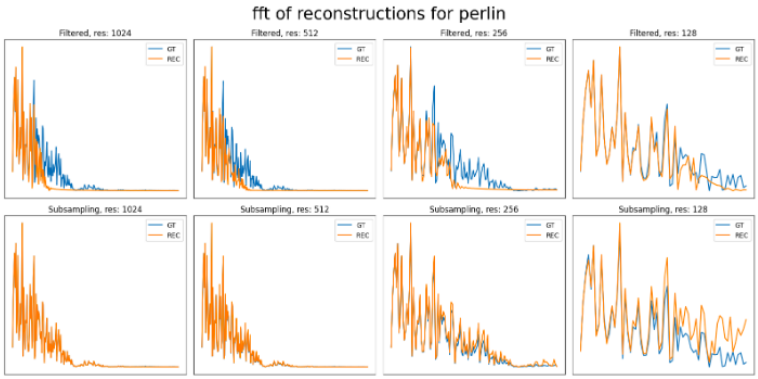

FIGURE 1. Regular Sampling Results

2. Stochastic Sampling

As shown in many previous works [3], stochastic sampling can allow for a sampling that trades aliasing error for random noise. This section will survey some common types of stochastic sampling techniques: Uniform, jittered, and Poisson disc sampling.

2.1. **Uniform**

Uniform sampling means to sample points on the space from the uniform distribution. One of the advantages is that it is fast and simple to implement. As a disadvantage, it also produces the worst results we have seen. However, there have been recent works using the properties of uniform sampling to efficiently train the network in more points in the domain (see [6]), so further study may be needed to explore this sampling potential. In our case we only sample only once in the training procedure. If we perform many samples for each epoch, it would be better.

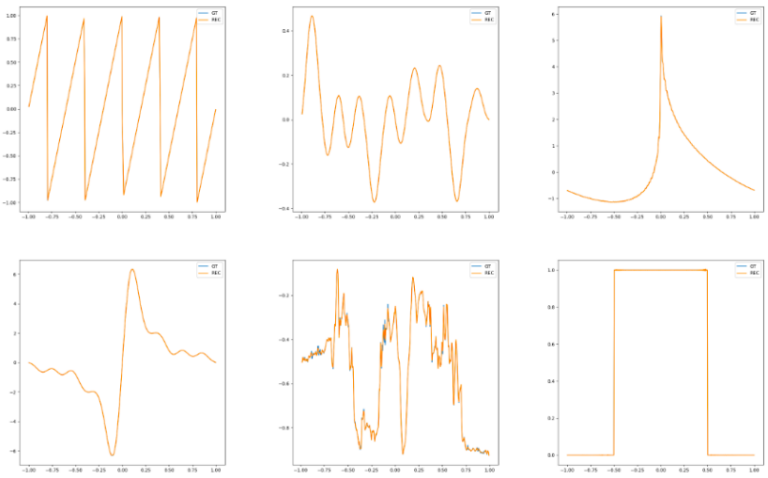

FIGURE 2. Uniform Sampling Results

2.2. **Jittered**

Jittered sampling is used to get the points from the regular sampling distribution and, with these points, apply some random noise to these points. This can induce the points to have a bit of "noise" and some stochasticity to the proceedings.

2.3. **Poisson Disk Sampling**

Poisson disk sampling is a type of random sampling in which the algorithm places the points randomly but ensures that every point is apart from the other by a radius of r. In our case, we use a modified version of [1].

Also, in the beginning, we had some problems ensuring that the points for the Poisson disk were evenly distributed around the image sample. As the image shows, we also had to perform some tuning from the hyper-parameters,

An essential characteristic of Poisson disk sampling entails its categorization as a form of blue noise. In other words, it constitutes a type of noise with absence of prominent concentrated spikes of energy (see [1]) that is often related with the retinal system. Nonetheless, a significant drawback associated with this sampling

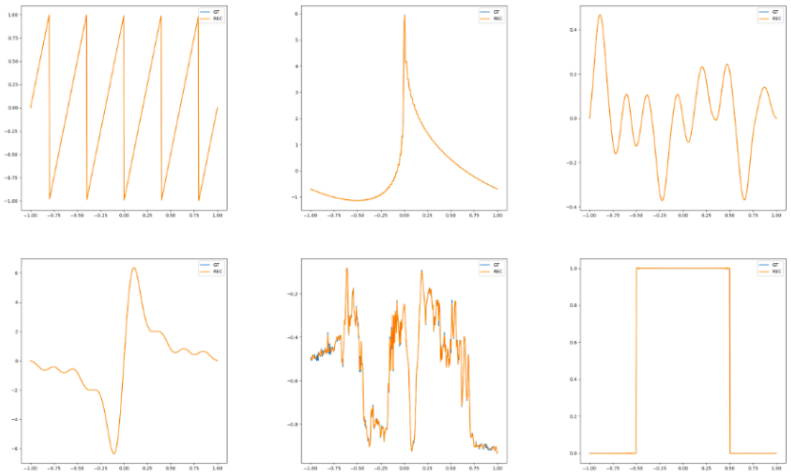

FIGURE 3. Jittered Sampling Results

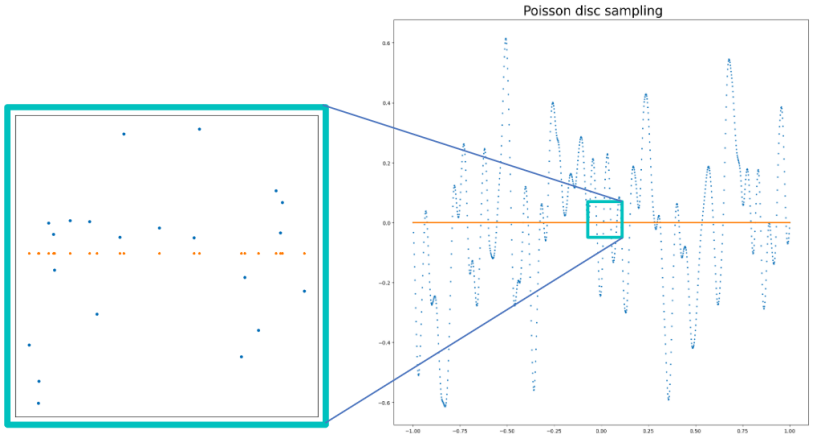

FIGURE 4. Poisson Sampling Example

methodology relates to its relatively slow execution time. Specifically, [1] reports a time complexity of approximately O(N) for the execution of Poisson disk sampling. Furthermore, the sampling creation process is highly non-parallelizable, making it necessary to pre-compute and store the training points.

3. Stochastic Pyramid

In image processing and computer vision, the denominated Gaussian and Laplacian pyramids are commonly used structures to analyze an image's features. In multiresolution theory, they appear as a natural definition as a decomposition of an image on scale and detail. They aid in applications such as image blending and morphing, texture analysis and synthesis, edge detection, etc.

Particularly, the Gaussian pyramid of an image is required to reconstruct it correctly based on regular samples; otherwise, an aliasing artifact could appear in regions with high frequencies. However, this only sometimes happens when reconstructing based on stochastic samples.

We are thus interested in creating similar structures for non-regular samples and analyzing the reconstruction properties of an image trained on such structures. Also, unlike its regular counterpart, we show that such a structure is not restricted to integer scaling for the pyramid, giving more freedom in the scale-detail domain.

3.1. **Regular MRNet Structures**

As introduced in [11], the MR-Net has devised a model architecture that generates a multiresolution representation for images. For that sake, the input image is processed into one of the following four structures:

- Gaussian pyramid: Creates a pyramid of images by filtering and downsampling (with a scale of 2) the previous level's image.
- Gaussian stack: Having constructed the Gaussian pyramid, resize the images to the original image's size.
- Laplacian pyramid: Given the image's Gaussian pyramid, the pyramid levels are subtracted to get the details in different scales. It can also be considered the discrete Laplacian of the image at different scales.
- Laplacian stack: The images that compose the Laplacian pyramid are reshaped to the original image's size to get the stack.

It's important to observe that the original image undergoes filtering to mitigate artifacts such as aliasing in the creation of pyramids. While this process is essential when the image coordinates are regular, it may be less crucial for stochastic coordinates.

3.2. **Stochastic Pyramid**

A stochastic pyramid was implemented using Poisson disc sampling in the following manner: given an image sampled on a Poisson disc grid $\mathcal{G}$ with radius r_1, to up(down)-sampling it, a new, smaller(bigger) ratio r_2 is defined, and points adhering to the Poisson disc sampling criteria (as outlined in [1]) are added(removed).

Observe that the previous algorithm presents several advantages:

- Compared to the classic Gaussian pyramid, our pyramid is agnostic to the scale s, which is given by
$$s = \max\left(\frac{r_1}{r_2}, \frac{r_2}{r_1}\right)$$
with $r_1, r_2 \in \mathbf{R}$.
- Due to the stochastic sampling nature, the points chosen during the pyramid creation process could be designated as key samples of the image, such

as critical points or those with the highest variance in the region. Leveraging higher-order information from these points can enhance the reconstruction, as demonstrated in [6].

The potential of this pyramid may lie in its ability to generate a multiresolution representation of the image within selected scale spaces, focusing on the image's feature points and eliminating the necessity for filtering at each level. Future endeavors could delve into exploring this aspect. Nevertheless, there are certain drawbacks inherently associated with this technique.

The pyramid constructs a new image at each level, diminishing the resolution proportionally to the scale. This leads to increased disk memory needed for training data, potentially rendering it impractical for higher image resolutions. Due to this, the upcoming section will suggest an alternative method to alleviate noise during training, thus acting as a natural filter.

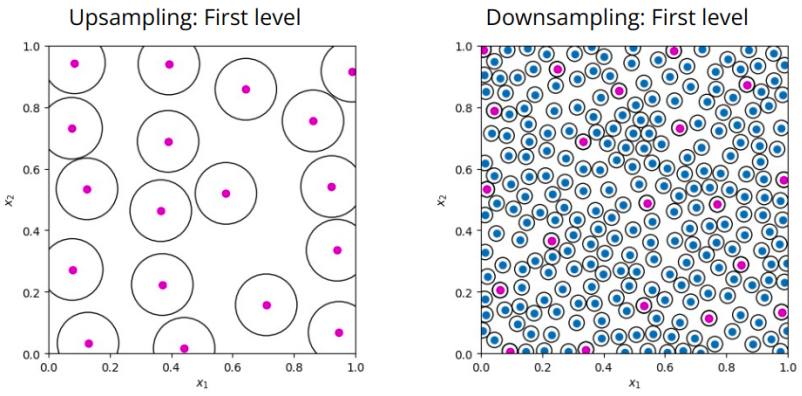

FIGURE 5. Pyramid Poisson Results 1rst level

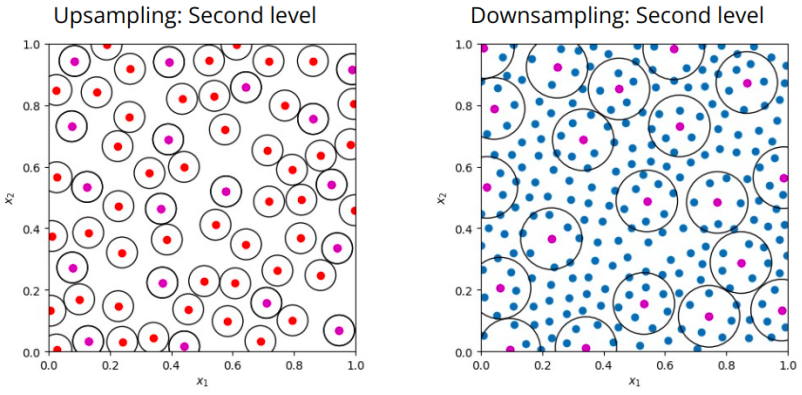

FIGURE 6. Pyramid Poisson Results 2nd level

4. Filters

Filtering plays a pivotal role in computer vision and image processing as it is a fundamental technique for enhancing, modifying, or extracting specific features from images. It is crucial for noise reduction, as it helps mitigate the impact of unwanted artifacts or disturbances in images, contributing to the improvement of image quality and the efficacy of subsequent processing steps.

More specifically, when regular samples represent the image, filtering avoids structural noise in the down-sampling process. This is very important when representing multiresolution images, as it is not possible to eliminate the structural noise product of defective samples using post-filtering.

However, as stated before, the cost of creating a Gaussian/Laplacian pyramid is undesirable. In that case, we present an alternative that considers a type of filtering during the training of the neural network.

4.1. **Structure of the Filters**

The objective of filters is to reduce the variation around a point by approximating a point value to a weighted mean over neighbors' evaluations of said point. The objective of such a process is to reduce high variation noise and avoid aliasing artifacts.

The intuition behind this process can be used to introduce a softening of the function during the training: For all points in a grid $\mathcal{G}$, it can be defined as a context grid $\hat{\mathcal{G}}$ such that f_θ will fit a mean of neighbors in $\hat{\mathcal{G}}$.

More formally, for each $x_i \in \mathcal{G}$, we will select the N_i nearest points $\{a_j^{(i)} \in \hat{\mathcal{G}}\}$ and perform the optimization algorithm minimizing the function

$$\sum_{i=1}^{N} \left(f_\theta(x_i) - \sum_{j=1}^{N_i} w_j f(a_{ij}) \right)^2 \tag{4.1}$$

where the choice of w_j defines the filter used, for example, by choosing $w_j = 1/N$ and $N_i = N$ for all i, j, we are applying the box filter.

As we can see, we are training the model to fit the points at a softened version of the ground truth image. However, the previous equation may be expensive to compute as it is necessary to interpolate the image on the stochastic points at each training step.

Then, the following modified version was implemented instead:

$$\sum_{i=1}^{N}\left(f_\theta(x_i)-\sum_{j=1}^{N_i} w_j f_\theta(x_i+v_j)\right)^2 \tag{4.2}$$

for some random (but fixed) vectors $\{v_j \in \mathcal{U}([-r_{neigh}, -r_{neigh}]^2)\}$. In this case, $\hat{\mathcal{G}} = \{x_i + v_j : x_i \in \mathcal{G}, j \in \{1, ..., N\}\}$. As the random points are already fixed, we can compute those points' evaluations before training, which speeds up the implementations.

Notice that equation 4.2 bears a strong resemblance with the model's Laplacian, in the following way: Given that f_θ is a continuous representation for the image, subtracting a mean of $f_\theta(x_l) = I_{ij}^{(filtered)}$ to $f_\theta(x_l) = I_{ij}$ would be equivalent to extract the details of the image, or in a more general view, to compute

$$I - I^{(filtered)}$$

which is a common way to compute the Laplacian for the image I.

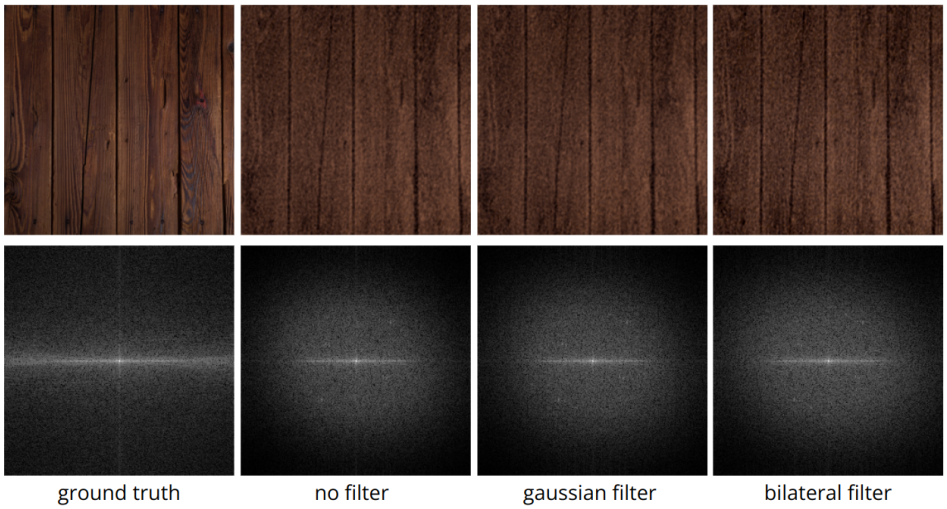

FIGURE 7. First row: image reconstruction with stochastic samples. The second row shows the magnitude of the Fourier series coefficient.

4.2. **Gaussian**

Given the equation 4.2, applying a Gaussian filter means the weights w_j are chosen based on the distance to the point. More specifically,

$$w_j = \Phi(\sigma, x_i - (x_i - v_j)) \quad with \quad \Phi(\sigma, x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(x^2/2\sigma\right)$$

then $w_j = \Phi(\sigma, v_j)$ where σ is an hyperparameter. Notice that the choice of σ is tied to the choice for r_{neigh} as if $\sigma >> r_{neigh}$, then the filter would be equivalent to the box filter, and if $\sigma << r_{neigh}$ then it would not filter the image.

The Gaussian filter is particularly popular as it has the best space-frequency localization, and it is possible to implement it efficiently using various techniques such as fast Fourier transform. However, the Gaussian filter is data-independent, making it blur noise and fine detail equally. The following section will show a special choice for the weights that seek to maintain fine-grained details such as corners and edges while blurring noise.

4.3. **Bilateral**

Unlike the Gaussian filter, the bilateral filter is defined as a weighted average of samples that takes into account the variation of intensities to preserve edges (data dependent). The rationale of bilateral filtering is that two points are close to each other if they occupy nearby spatial locations and have some similarity in the photometric range.

More precisely, if $\mathbf{p} \in \mathcal{G}$, $\mathcal{N} \subseteq \hat{\mathcal{G}}$ is the set of N neighbour points to $\mathbf{p}$ in $\hat{\mathcal{G}}$ and $I_{\mathbf{u}}$ is the value of the image on $\mathbf{u}$, then

$$\frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in \mathcal{N}} \Phi(\sigma_s, ||\mathbf{p} - \mathbf{q}||) \Phi(\sigma_r, I_{\mathbf{p}} - I_{\mathbf{q}}) I_{\mathbf{q}}$$

where $W_{\mathbf{p}}$ is a normalization factor

$$W_{\mathbf{p}} = \sum_{\mathbf{q} \in \mathcal{N}} \Phi(\sigma_s, ||\mathbf{p} - \mathbf{q}||) \Phi(\sigma_r, I_{\mathbf{p}} - I_{\mathbf{q}})$$

The bilateral filter is controlled by two parameters: σ_s and σ_r. The range parameter σ_r controls the threshold for the difference between two adjacent intensities to be considered an edge. Thus, the higher the value of σ_r, the closer to Gaussian blur. On the other hand, the spatial parameter σ_s controls the size of the neighborhood the weighted averages will be taken from.

The implementation used in this report was a direct approach for the bilateral filter, thus having a high computational cost. Future work may consider using a faster implementation, such as the bilateral grid or similar methods (see [8]).

5. Adaptive Sampling

In this section, we review the types of algorithms that we used to perform adaptive sampling. That is, a type of sampling that is based on the data being processed and can "adapt´´ to the behavior of the data.

5.1. **Feature Sampling**

We performed adaptive sampling based on SIFT [5] and the Harris corner detector [4]. For our approach, we first took the SIFT features and Harris corner detection samples and took points inside a circle with a specified radius. Then, we combined these points with those already sampled from the image.

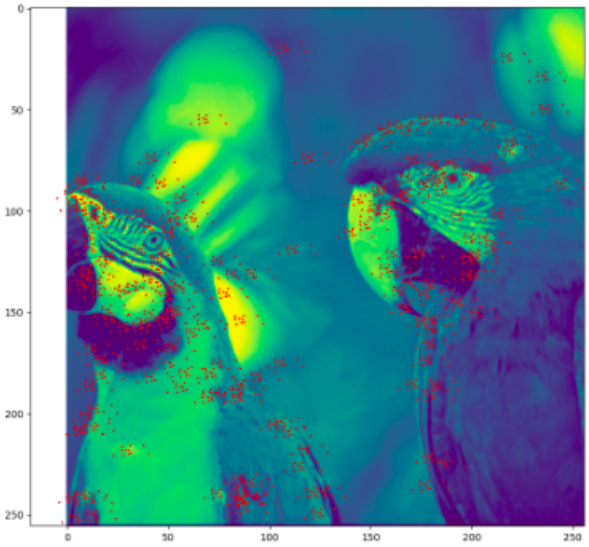

FIGURE 8. SIFT Feature Sampling, example

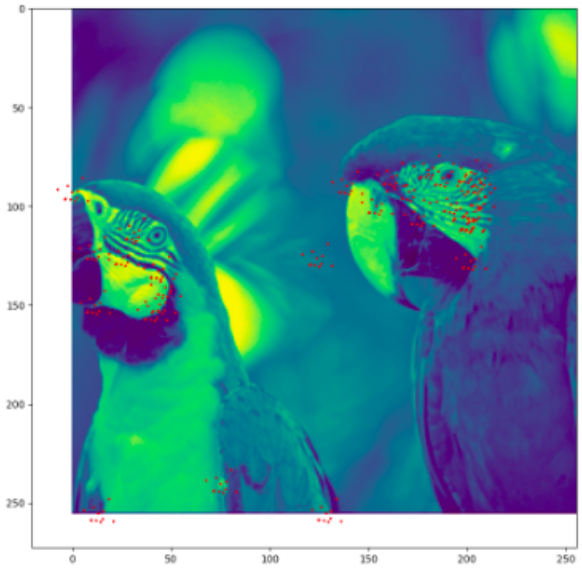

FIGURE 9. Harris Feature Sampling, example

5.2. **Adaptive Multiresolution Warnock Sampling**

For the adaptive multiresolution Warnock sampling, we were inspired by the classical Warnock's algorithm [12] to solve the visible surface problem. To do this, we used quad-trees and recursively subdivide the image till it stops at a variance threshold for the gradient. After doing this, our algorithm stops. We used the previously mentioned stochastic pyramid to perform this analysis. After doing this, we "squash the samples and use them during training.

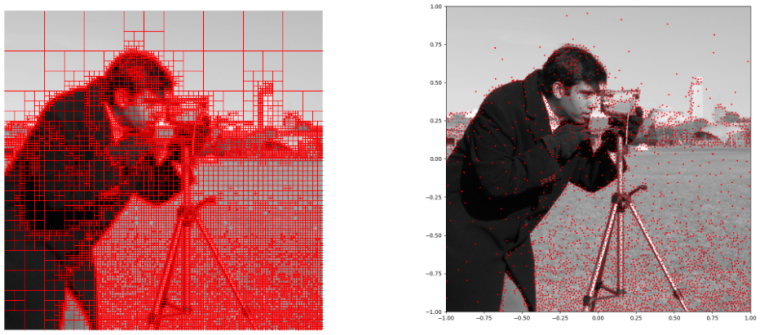

FIGURE 10. MR-Warnock Sampling. On the left, we can see the quad-tree that we used to generate the samples, and on the right, we can see the samples.

6. Implicit Curves

In this section, we briefly describe our experiments with implicit curves, both in 3D and in 2D. for the 3D curves, we adapted [6] to work with the [9] framework.

6.1. **MR-i3D**

i3D [6] is a state-of-the-art technique for reconstructing a deep neural implicit level set given a mesh. By adapting it for the MR-Net framework, we were able to make some tests for this type of framework. We used all the losses of the [6] but adapted their code for our setting.

6.2. **2D Curves**

In this section, we describe some of the tests we have done with 2D implicit curves and how we adapted i3d for 2D. We first draw the points; however, since only those points are not enough, we instead use a Chaikin subdivision [2] to get even more points. After this, we gain access to the normals and the points of these points for each vertex. Next, since we have the polygon, we can compute the exact SDF for the polygon. So, we can proceed with the standard i3d pipeline. In Figure 13, we see some of the results of using this technique.

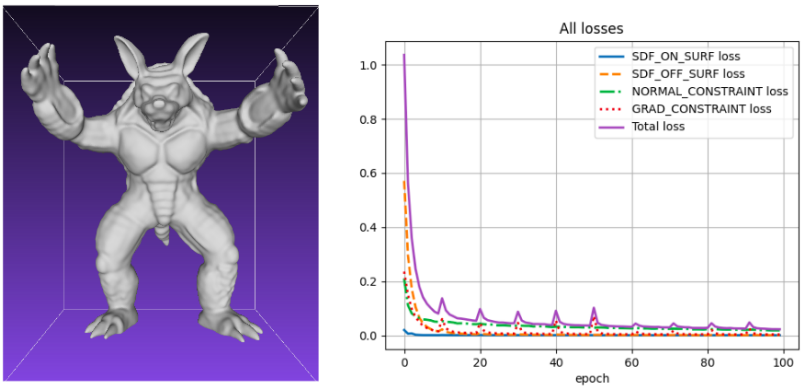

FIGURE 11. MR-i3d Training

FIGURE 12. MR-i3d Training multi-stage

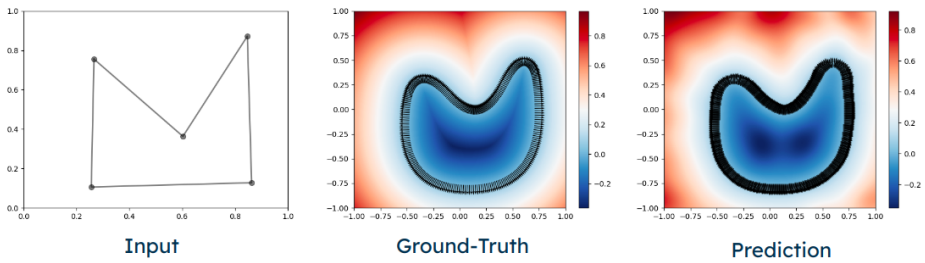

FIGURE 13. In here we can see the input, the result of the subdivision SDF with the normals, and, finally, the computed SDF and its normals

7. CONCLUSION

Although there are still many future avenues for research and open problems that this report does not address, we hope that our report assists as a compilation of different techniques. Finally, the port of i2d can be an important avenue for research. For example, future work can be done on exploring the geometry of the 2D curves.

References

[1] Robert Bridson. Fast poisson disk sampling in arbitrary dimensions. *SIGGRAPH sketches*, 10(1):1, 2007.

[2] George Merrill Chaikin. An algorithm for high-speed curve generation. *Computer graphics and image processing*, 3(4):346–349, 1974.

[3] Robert L Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics (TOG)*, 5(1):51–72, 1986.

[4] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[5] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[6] Tiago Novello, Guilherme Schardong, Luiz Schirmer, Vinicius da Silva, Helio Lopes, and Luiz Velho. Exploring differential geometry in neural implicits. *Computers & Graphics*, 108:49–60, 2022.

[7] Harry Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.

[8] Sylvain Paris, Pierre Kornprobst, Jack Tumblin, Frédo Durand, et al. Bilateral filtering: Theory and applications. *Foundations and Trends® in Computer Graphics and Vision*, 4(1):1–73, 2009.

[9] Hallison Paz, Tiago Novello, Vinicius Silva, Guilherme Schardong, Luiz Schirmer, Fabio Chagas, Helio Lopes, and Luiz Velho. Multiresolution neural networks for imaging. In *2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, volume 1, pages 174–179. IEEE, 2022.

[10] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020.

[11] Luiz Velho, Hallison Paz, Tiago Novello, and Daniel Yukimura. Multiresolution neural networks for multiscale signal representation. Technical report, Technical report, VISGRAF Lab, 2022.

[12] John Edward Warnock. *A hidden surface algorithm for computer generated halftone pictures*. The University of Utah, 1969.

IMPA - VISGRAF, Rio de Janeiro, Brazil

E-mail address: daniel.perazzo@impa.br, diana.aldana@impa.br