The approaches are trained on dataset "synthetic-50-5" as mentioned in Chapter **??** with 3000 scenes. The input matrix has 128×128 height and width. Each vertex map has dimension $128 \times 128 \times 3$, light map has $128 \times 128 \times 3$, image has dimension $128 \times 128 \times 1$. The training pipeline use batch size 8, Adam optimizer (**adam**), learning rate of 1×10^{-3}, learning schedule [8,1000], learning decay factor 0.5. The model is trained with PyTorch 1.10.0a0, CUDA 11.4.1, GPU with single NVIDIA GEFORCE RTX 3090.

1 GCNN model evaluation

The GCNN model is the base model of the whole thesis. In this section, we first evaluate the inpainting performance of the model based on light map input. Then we use the same model for normal inference task.

1.1 Light Inpainting

The light inpainting task in this section in order to evaluate the noise inpainting performance of GCNN model. It takes semi-dense light map as input to predict the fully dense light map, as shown in Figure 1. The network is trained on 3000 light maps in the "synthetic-50-5" dataset with initial learning rate 0.001, learning schedule 8,1000 with decay factor 0.5, the batch size is 8, the feature map channel is 128 in all of the gated convolution layers. All the models are tested on a laptop with a single NVIDIA GeForce 940MX, 2GB memory.

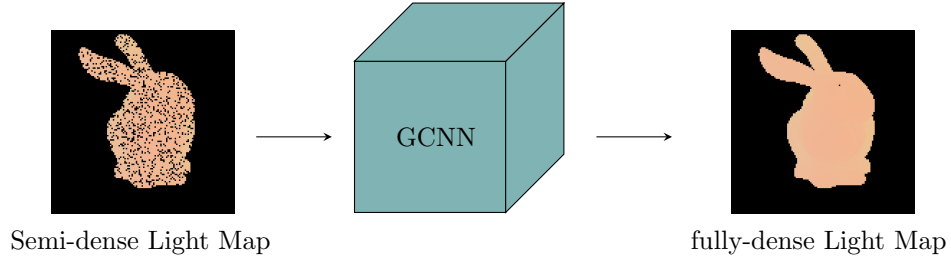Semi-dense Light Map   GCNN   fully-dense Light Map

Figure 1: Light map inpainting based on GCNN architecture.

The result is shown in Table 1. Besides the GCNN model, we use two extra model as the comparisons. The first one is GCNN-NOC, where NOC means "no skip connection", the second one is CNN, which has the same architecture as the GCNN but only use standard convolution layers in the network. The GCNN architecture achieves the average angle error lower than 1 degree. As a comparison, U-CNN model has 4 degrees error, which shows the relative weakness of noise filter competence. Besides, the skip connection also booster the performance, since the no skip connection version still has error greater than 1 degree. The both two models for the inpainting task reveals the superiority of GCNN architecture. Figure 3 and 2 gives a quantitative and qualitative evaluation.

Model	Angle	Time /ms	bz	lr-schedule	lr-df
U-GCNN	0.59	14.45	8	8,1000	0.5
U-GCNN-NOC	1.31	16.14	8	8,1000	0.5
U-CNN	4.10	10.52	8	8,1000	0.5

Table 1: The performance of the GCNN model for light map inpainting. The angle error is the average angle error of all valid pixels in the test case. bz stands for batch size, lr-schedule stands for learning rate schedule, lr-df stands for learning rate decay factor.

(a) gray-scale image   (b) Vertex(Input)   (c) ground-truth

(d) GCNN   (e) U-CNN   (f) U-GCNN-NOC

err: 0.53   err: 7.75   err: 1.17

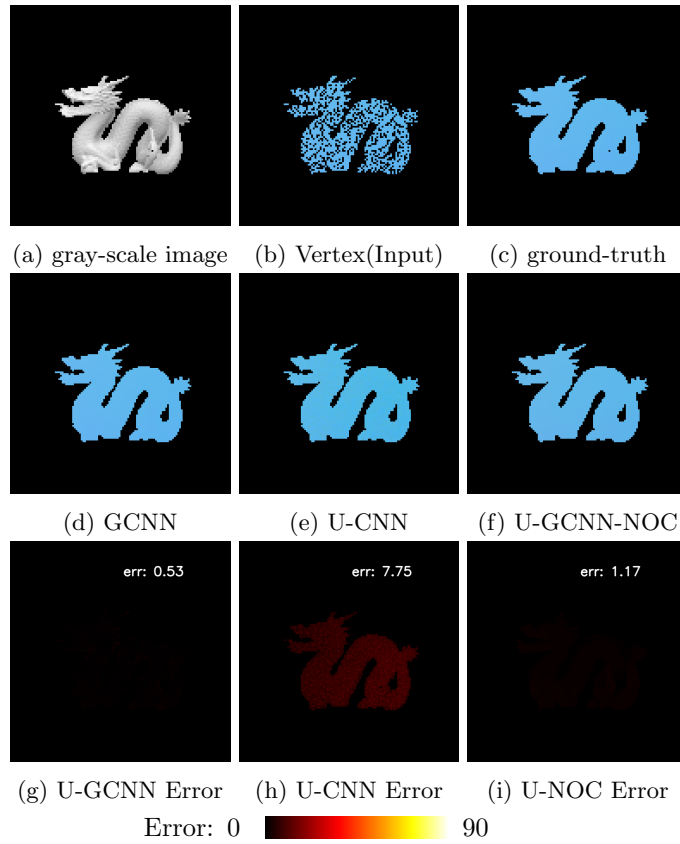(g) U-GCNN Error   (h) U-CNN Error   (i) U-NOC Error

Error: 0   90

Figure 2: GCNN Normal Inference for light map inpainting based on dragon object. The degree error map is shown in last row. The average error is shown in the upper right corner in the error map.
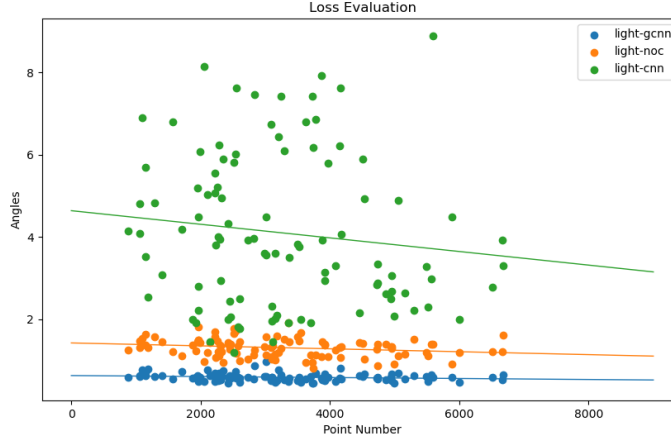
Figure 3: GCNN for light map inpainting on 100 scenes. The evaluated models are corresponding to the table 1, the line in the figure is the corresponding regression line of each model.

1.2 Surface Normal Inference

We evaluate geometry based approach for surface normal estimation using a pure GCNN architecture. The input of the model is 3D vertex map whereas the output is directly the corresponding tangent surface normal map in range $[-1, 1]$. The training parameters are as same as light inpainting task mentioned above.

A qualitative evaluation on object "dragon" is shown in Figure **??** and Figure 5, theb SVD approach is considered as a base line. The GCNN based approach has apparently better performance than SVD based approach, which completely get rid of the trace of the noise in the input vertex map. Figure 5 zooms in the right leg area of the Washington statue object to give a closer comparison with the ground-truth. As shown in the figure, the GCNN method successfully predict the surface connection area between the lower hem of the gown and the stone chair, whereas the no skip version is blurry in the same area. The two surface is still distinguishable but the sharp details has been lost. The CNN version has the skip connection thus gives a better detail than NOC version. However, if we compare the error map of U-GCNN and U-CNN in figure **??**, the basement of the stone chair that Washington sit is actually a smooth flat stage, the U-GCNN shows an accurate predict, but the same area in U-CNN has an average higher error. The whole basement surface has a overall higher error. It holds also for the other smooth areas like the side of the stone chair, the gown and the upper body of the statue. It is because the noise of the input still disturb the CNN model and take the input noise into account for normal estimation which deviate to the correct surface normal. When we look back to

the U-GCNN based method, we can found that the surface normal has better performance in the smooth area compare to the CNN approach and a sharp detail compare to the no skip connection version.
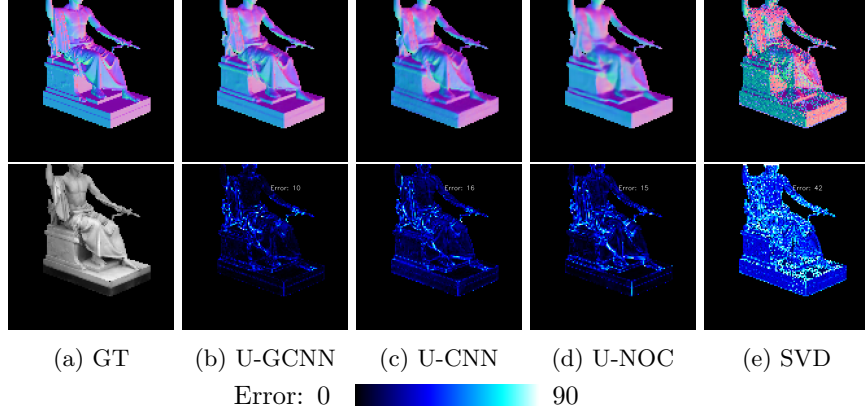

| (a) GT | (b) U-GCNN | (c) U-CNN | (d) U-NOC | (e) SVD |

Error: 0 ▬▬▬▬▬ 90

Figure 4: Surface Normal Inference based GCNN model on "Washington´´ object. The first row shows the estimated surface normal. The second row is the degree error map.
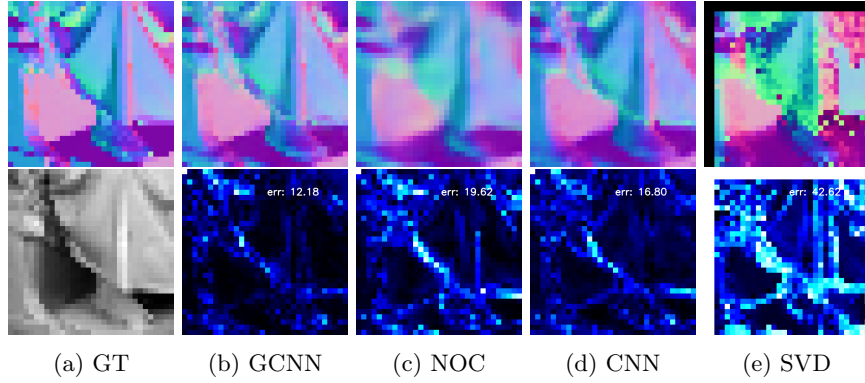

| (a) GT | (b) GCNN | (c) NOC | (d) CNN | (e) SVD |

Figure 5: Zoom in of the center region of Washington object. The first row is surface normal, the second row is the corresponding errors.

The evaluation visualization on real dataset is shown in Figure 6

We evaluate the U-GCNN model on 100 test images as a quantitative evaluation. The result is shown in Figure 7. The GCNN based method has average angle error around 10 degrees, whereas the no skip connection version has error above 12 degrees and the CNN version has error above 15 degrees.

4

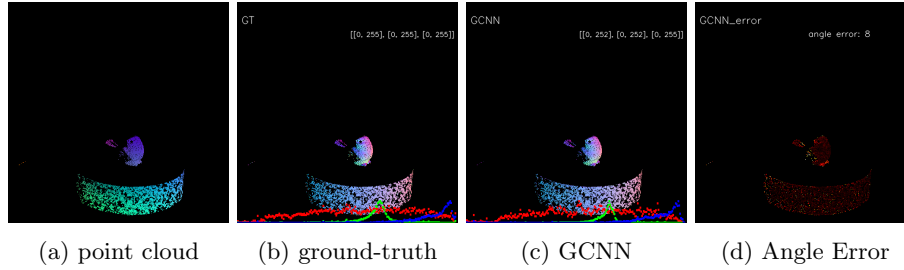(a) point cloud   (b) ground-truth   (c) GCNN   (d) Angle Error

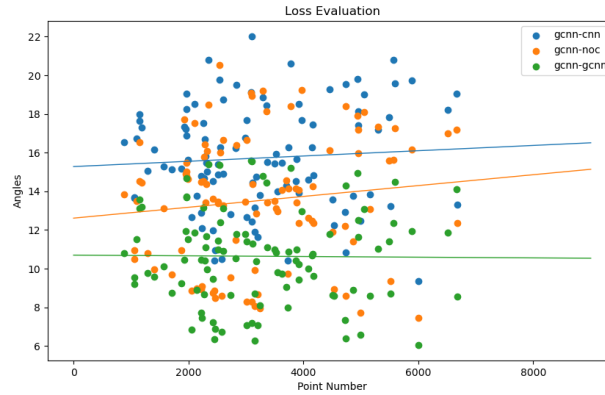Figure 6: Evaluation on Real Dataset

Figure 7: Evaluation of average angular loss on the whole test dataset with 100 scenes. The x-axis indicates the point number, the y-axis indicates the angles. The lines show in the picture are the regression line of each model.

2 Surface Normal Inference based on Calibrated Illuminated RGBD images

We evaluate the GCNN architecture based light net on the test dataset. The light net parameters are further used as the initial parameter of light branch of the Trignet.

A qualitative evaluation of light net is shown in Figure **??**. The average angular error are lower than 0.3 degree on all of test cases. In this case, to distinguish the output and the ground truth is already not easily only use naked eyes.

Figure 8 uses a scatter plot to show the average angular error on 100 different test cases. The average pixel-wise angular error is 0.17 degree as shown in Table 2. An regression line has been added in the plot to analysis the tendency of the errors. The angular error slightly goes up when valid point number in the scene increase. It is make sense since the valid pixels are usually connected

and concentrate in a single patch, the less of the area of the patch, which corresponding the number of the valid points, the less variation of the light direction among the pixels, thus better the evaluation.
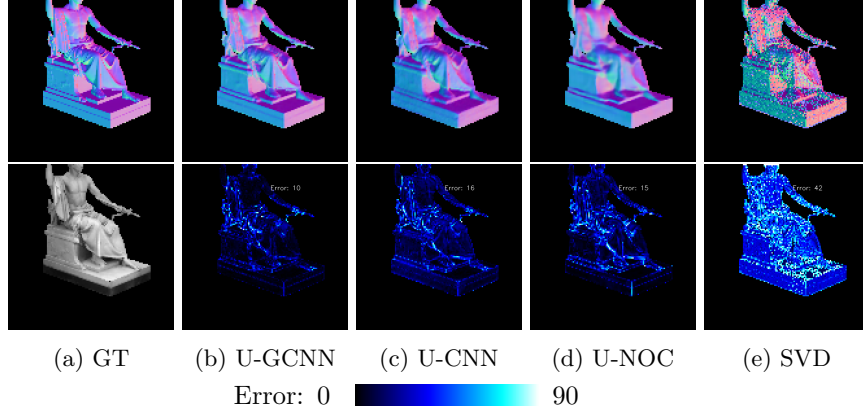

| (a) GT | (b) U-GCNN | (c) U-CNN | (d) U-NOC | (e) SVD |

Error: 0 ▮▮▮▮ 90

Figure 8: Surface Normal Inference based GCNN model on "Washington″ object. The first row shows the estimated surface normal. The second row is the degree error map.
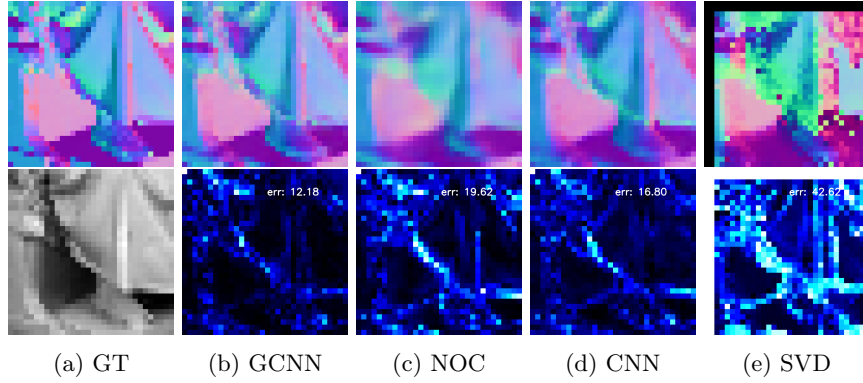

| (a) GT | (b) GCNN | (c) NOC | (d) CNN | (e) SVD |

Figure 9: Zoom in of the center region of Washington object. The first row is surface normal, the second row is the corresponding errors.

2.1 comparison

From the Figure 11 we can observe the normal difference between ground-truth and GCNN predicted normals in another dimension. It separates the interval $[-1, 1]$, which is exactly the range of normal vector, to 256 sections. Then it counts the number of points locates in each section for 3 axes. The 3 axes are fitted quite well in most of interval but other than $[-0.25, 0.25]$ for x and y

axes and interval close to -1 for z axis. Therefore a further constraint can be considered to the loss function related to the normal difference shown in this figure.

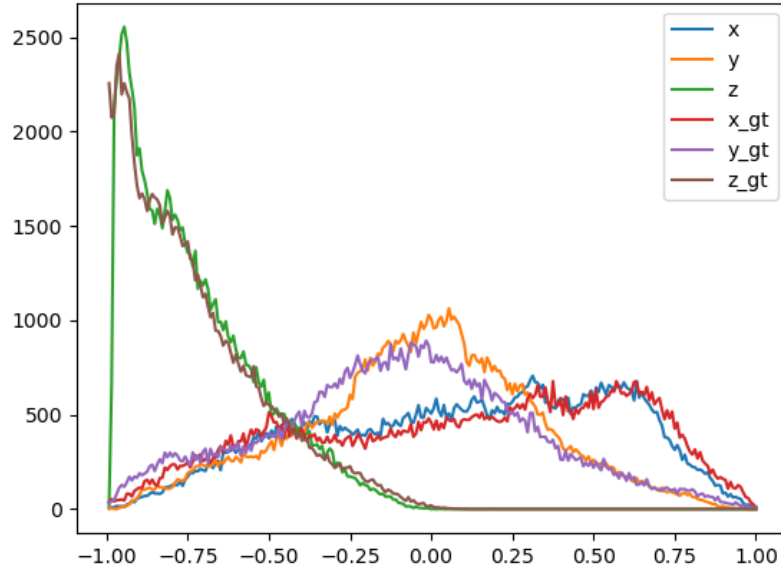It is faulty that almost no normal has -1 z-component in GCNN predicted normal map. The reason?

Figure 10: The normal difference of between GCNN and ground-truth in x, y, z-axis respectively. The y axis indicates the number of points, x axis indicates the value of normal in x/y/z axis. (The chart is based on the "dragon" scene showing above)

Model	Angle	Time /ms	bz	lr-schedule	lr-df	l/i. Nr.	over-perform
LightNet	0.17	4.72					
GCNN	10.57	5.25	8	200,1600	0.5	0	-
an3	10.46	64.86	8	3,12,1000	0.5	1	yes
an3	10.81	65.34	8	10,1000	0.5	1	no(yes)
VIL-1	10.50	31.57	8	10,1000	0.5	1	yes
VIL-1	10.82	32.61	8	3,12,1000	0.5	1	no(yes)
VIL-3	10.79	54.76	8	100,1000	0.1	3	no(yes)
VIL-10	11.10	132.32	8	100,1000	0.1	10	no(yes)

Table 2: The error of models. The angle error is the average angle error of all valid pixels in the test case. The time unit is millisecond. bz is the batch size, lr-schedule is learning rate schedule. lr-df is learning rate decay factor, l/i. Nr is the number of light-image maps used for each scene
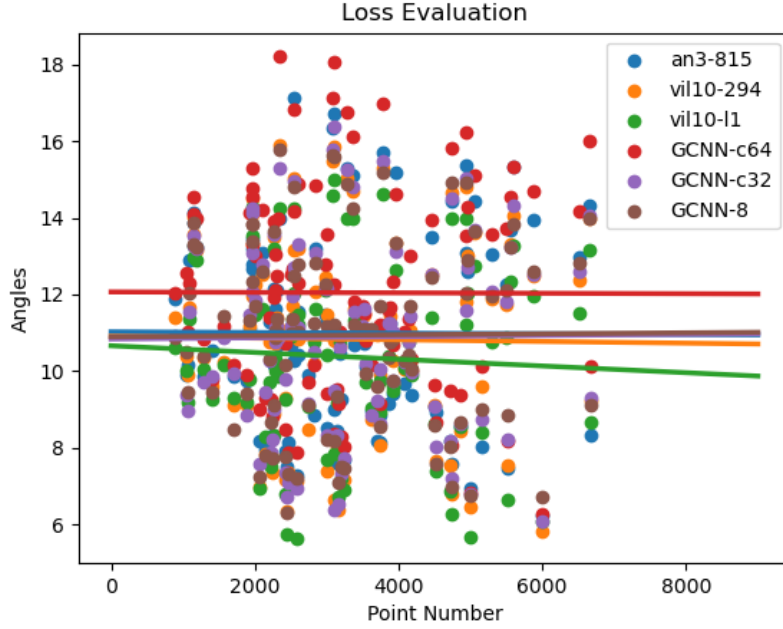

Figure 11: The normal difference of between GCNN and ground-truth in x, y, z-axis respectively. The y axis indicates the number of points, x axis indicates the value of normal in x/y/z axis. (The chart is based on the "dragon" scene showing above)