NeRF: Neural Radiance Fields for view synthesis

Кириллов Дмитрий

Подготовлено на основе статьи NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis
https://arxiv.org/abs/2003.08934

Постановка задачи

Реконструкция 3D модели сцены по ее фотографиям

Входные данные: Изображения сцены и направление взгляда на сцену

Постановка задачи

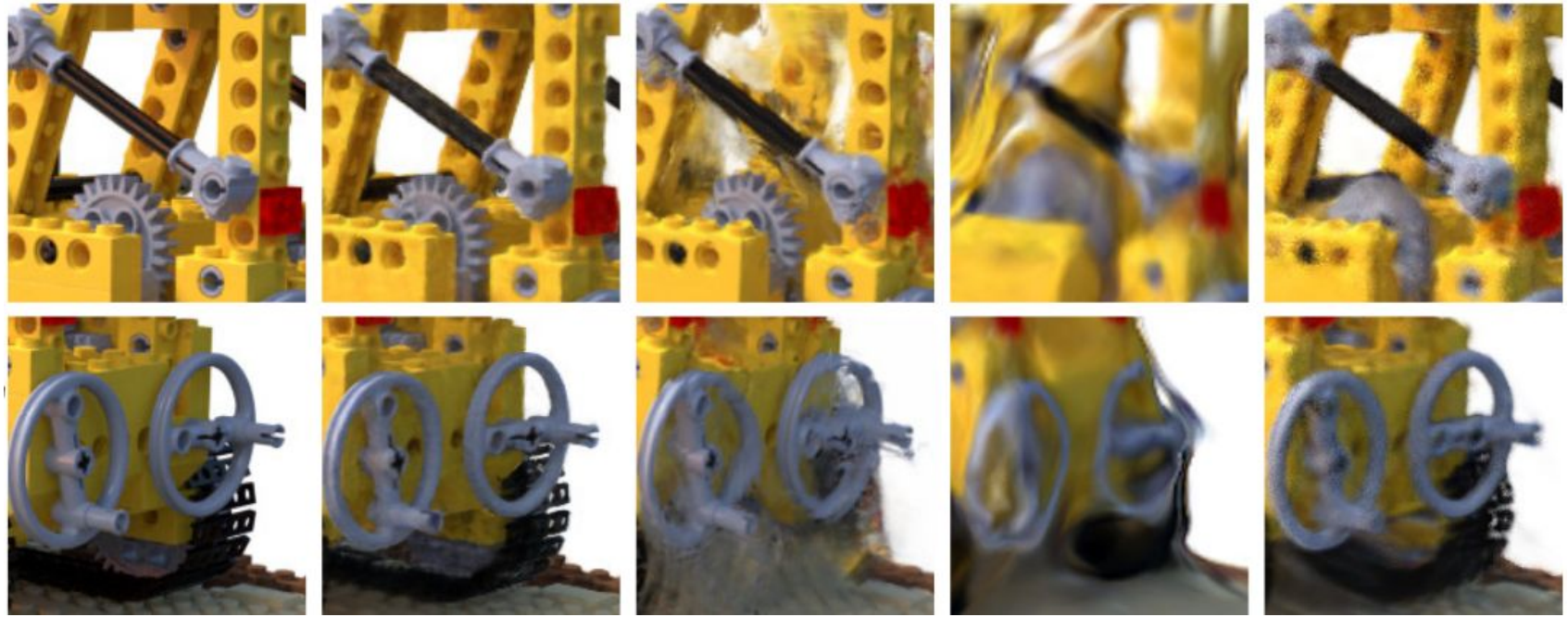
Реконструкция 3D модели сцены по ее фотографиям

Входные данные: Изображения сцены и направление взгляда на сцену

Выходные данные:

- Неформально некоторое представление сцены
- Формально изображение сцены с любого угла

Насколько хорош новый метод



Ground Truth NeRF LLFF SRN NV

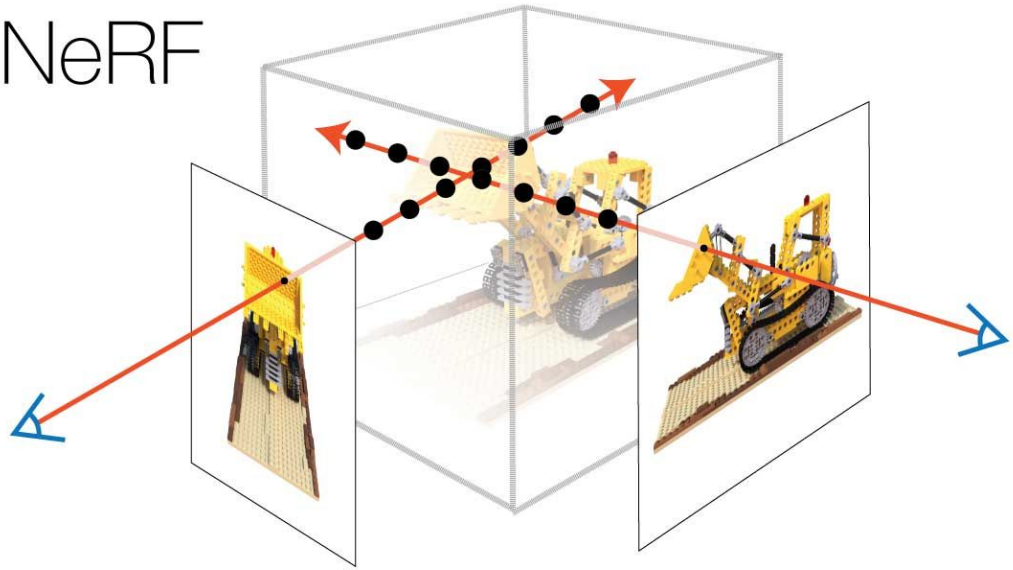
Насколько хорош новый метод

Ground Truth NeRF LLFF SRN NV

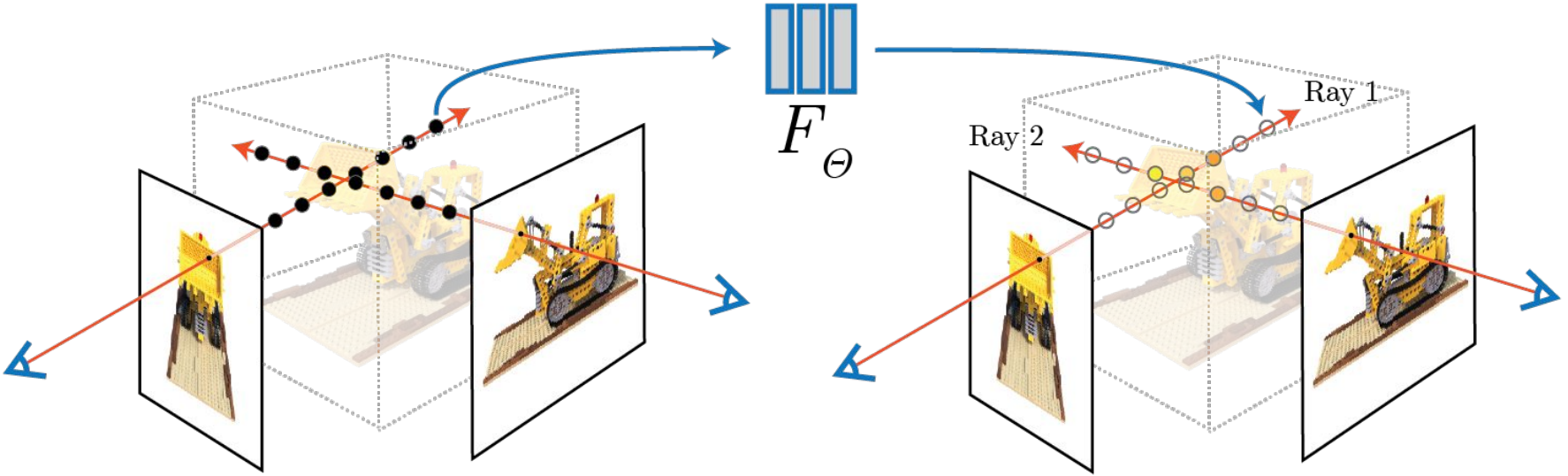
Основа метода

Сэмплирование точек вдоль луча

$$(x, y, z, \theta, \phi)$$

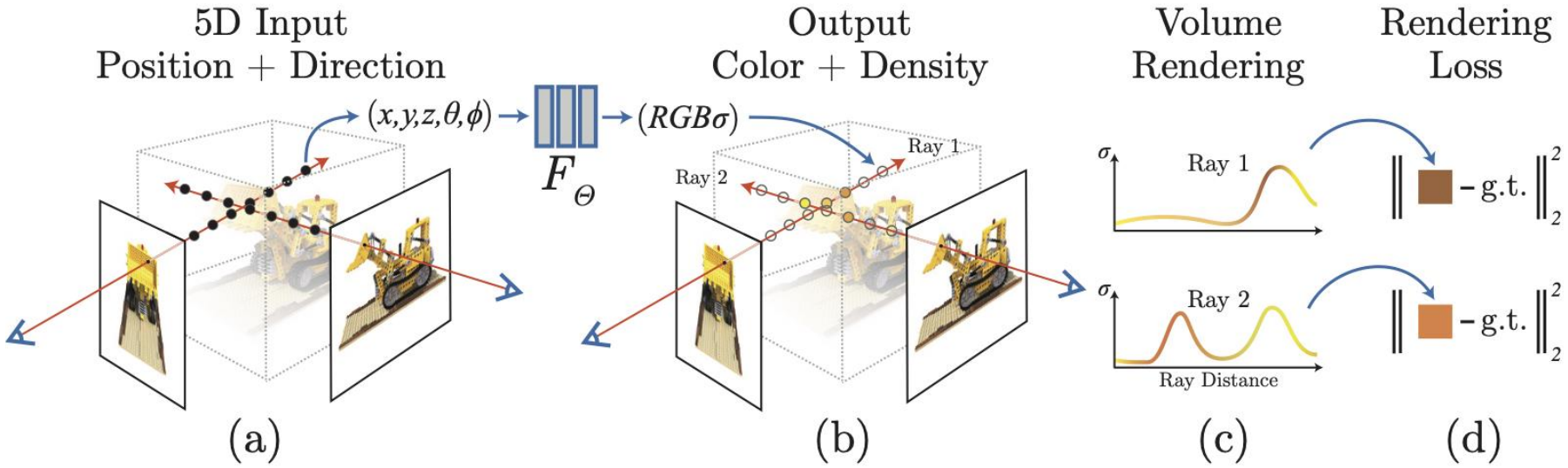


Основа метода



$$F : (x, y, z, \theta, \phi) \mapsto (RGB(x, y, z, \theta, \phi), \sigma(x, y, z))$$

Основа метода



Volumetric ray tracing

Непрерывный случай

$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}, t \in [t_{\text{бл}}, t_{\text{дал}}]$ — луч, выходящий из изображения

Volumetric ray tracing

Непрерывный случай

$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}, t \in [t_{\text{бл}}, t_{\text{дал}}]$ — луч, выходящий из изображения

$T(t) = \exp(-\int_{t_{\text{бл}}}^{t} \sigma(\mathbf{r}(s))ds)$ — “прозрачность” луча перед точкой

Volumetric ray tracing

Непрерывный случай

$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}, t \in [t_{\text{бл}}, t_{\text{дал}}]$ — луч, выходящий из изображения

$T(t) = \exp(-\int_{t_{\text{бл}}}^{t} \sigma(\mathbf{r}(s))ds)$ — “прозрачность” луча перед точкой

$$\mathrm{C}(\mathbf{r}) = \int_{t_{\text{бл}}}^{t_{\text{дал}}} T(t)\sigma(\mathbf{r}(t))RGB(\mathbf{r}(t), \mathbf{d})dt$$

плотность

Volumetric ray tracing

Дискретный случай

Сэмплируем t вдоль луча:

1. Делим луч на N равных частей
2. Из каждой части равномерно выбираем точку

$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}, t \in \{t_1, \ldots, t_N\}$ — луч, выходящий из изображения

$$t_i \sim \mathcal{U}[t_{\text{бл}} + \frac{i-1}{N}(t_{\text{дал}} - t_{\text{бл}}), t_{\text{бл}} + \frac{i}{N}(t_{\text{дал}} - t_{\text{бл}})]$$

Volumetric ray tracing

Дискретный случай

$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}, t \in \{t_1, \ldots, t_N\}$ — луч, выходящий из изображения

$T_i = \exp\left(- \sum_{j=1}^{i-1} \sigma_i \delta_i \right)$ — “прозрачность” луча перед точкой точки

Volumetric ray tracing

Дискретный случай

$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}, t \in \{t_1, \ldots, t_N\}$ — луч, выходящий из изображения

$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_i \delta_i\right)$ — “прозрачность” луча перед точкой точки

$$\mathrm{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (1 - e^{-\sigma_i \delta_i}) RGB_i$$

прозрачность луча между соседними *t*

Volumetric ray tracing

Дискретный случай

$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}, t \in \{t_1, \ldots, t_N\}$ — луч, выходящий из изображения

$T_i = \exp\left(- \sum_{j=1}^{i-1} \sigma_i \delta_i \right)$ — “прозрачность” луча перед точкой точки

$$\mathrm{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (1 - e^{-\sigma_i \delta_i}) RGB_i$$

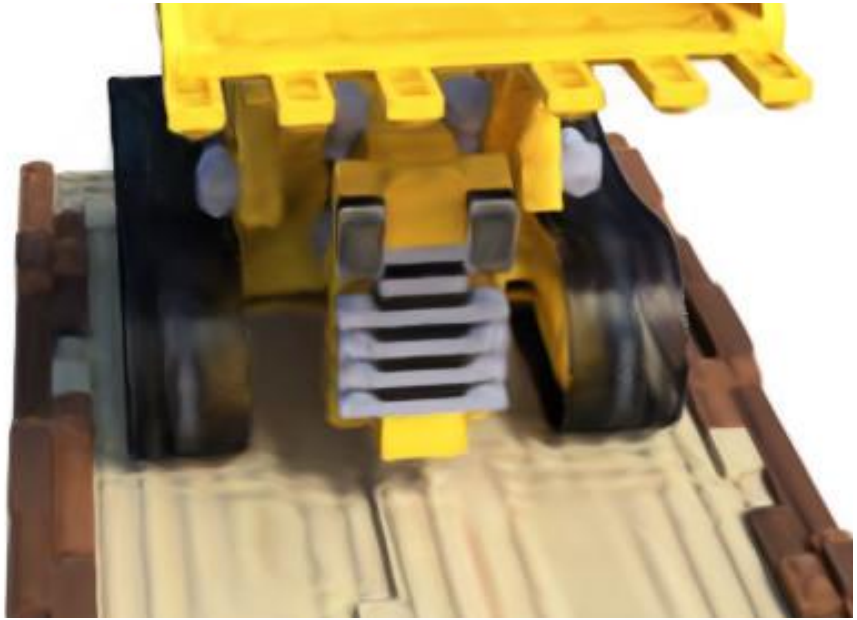
“плотность” точки

https://drops.dagstuhl.de/opus/volltexte/2010/2709/

Основа метода

Применение метода в исходном виде дает недостаточно хороший результат

Результат по метрикам хуже всех сравниваемых аналогов



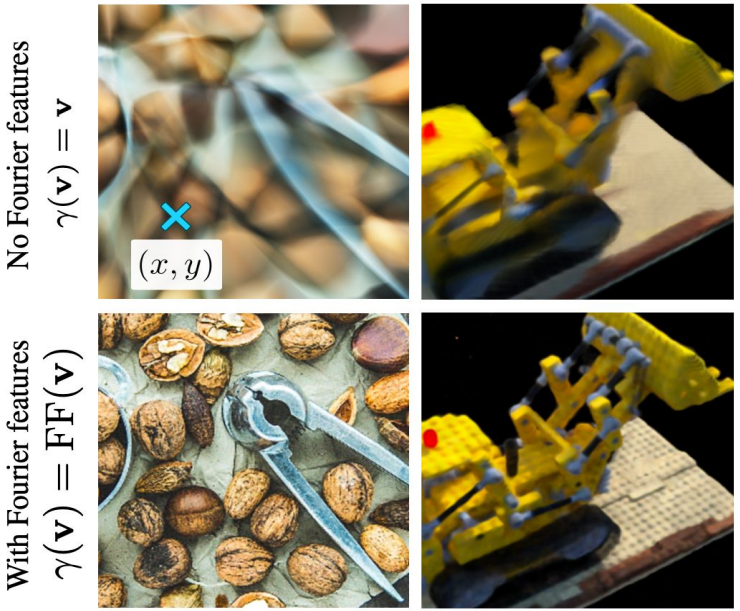
Улучшения

Positional encoding

- Нейросети с ReLU склонны выучивать медленно меняющиеся (низкочастотные) функции
- Получаемая модель будет “размытой”
- Применим к данным высокочастотное преобразование
- Нейросеть с той же архитектурой даст более резкую (высокочастотную) модель

Positional encoding

$$\gamma(x) = \left(\sin(2^0 \pi x), \cos(2^0 \pi x), \ldots, \sin(2^{L-1} \pi x), \cos(2^{L-1} \pi x) \right)$$



https://people.eecs.berkeley.edu/~bmild/fourfeat/index.html

Hierarchical volume sampling

- Две нейросети: **грубая** и **точная**
- **Грубая** работает как исходный метод
- **Грубая** дает оценку “важности” каждой точки сцены
- **Тонкая** работает с более “важными” точками

Hierarchical volume sampling

1. Выбираем $N_{\text{гр}}$ точек
2. Запускаем на них грубую нейросеть
3. Получаем изображение

   $$\hat{C}_{\text{гр}}(\mathbf{r}) = \sum_{i=1}^{N_{\text{гр}}} w_i RGB_i$$
4. Используем $\hat{w}_i = w_i / \sum w_i$ как кусочно-постоянную плотность вероятностного распределения
5. Из этого распределения выбираем $N_{\text{точн}}$ новых точек
6. Получаем выход точной сети $\hat{C}_{\text{точн}}(\mathbf{r})$ для всех $N_{\text{гр}} + N_{\text{точн}}$ точек

Результаты

Метрики

- **PSNR**(Peak Signal-to-Noise Ratio) $PSNR \sim \log\left(\frac{1}{MSE}\right)$
- **SSIM**(Structural Similarity) метрика схожести, учитывающая “восприятие ошибки”. $\mathbf{SSIM} \in [0, 1]$ Больше схожесть ⇒ больше **SSIM**
- **LPIPS**(Learned Perceptual Image Patch Similarity) Нейросети (обычно VGGNet), обученные измерять “воспринимаемое различие” Больше схожесть ⇒ меньше **LPIPS**

https://richzhang.github.io/PerceptualSimilarity/

Сравнение с аналогами

Метод	Рассеянные модели 360°			Реалистичные модели 360°			Реальные сцены (данные из LLFF)		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
SRN	33.20	0.963	0.073	22.26	0.846	0.170	22.84	0.668	0.378
NV	29.62	0.929	0.099	26.05	0.893	0.160	—	—	—
LLFF	34.38	0.985	0.048	24.88	0.911	0.114	24.13	0.798	**0.212**
NeRF	**40.15**	**0.991**	**0.023**	**0.947**	**0.081**	**0.081**	**26.50**	**0.811**	0.250

Ablation study

	Вход	#Im.	L	$(N_{\text{гр}}, N_{\text{точн}})$	PSNR↑	SSIM↑	LPIPS↓
Базовый метод	xyz	100	-	(256, -)	26.67	0.906	0.136
Без Pos. Encoding	xyzθφ	100	-	(64, 128)	28.77	0.924	0.108
Без угла взгляда	xyz	100	10	(64, 128)	27.66	0.925	0.117
Только гр. сеть	xyzθφ	100	10	(256, -)	30.06	0.938	0.109
Очень мало изобр.	xyzθφ	25	10	(64, 128)	27.78	0.925	0.107
Меньше изобр.	xyzθφ	50	10	(64, 128)	29.79	0.94	0.096
Меньше частотн.	xyzθφ	100	5	(64, 128)	30.59	0.944	0.088
Больше частотн.	xyzθφ	100	15	(64, 128)	30.81	0.946	0.096
Полная модель	xyzθφ	100	10	(64, 128)	**31.01**	**0.947**	**0.081**

Ablation study

	Вход	#Im.	L	$(N_{\text{гр}}, N_{\text{точн}})$	PSNR↑	SSIM↑	LPIPS↓
Базовый метод	xyz	100	-	(256, -)	26.67	0.906	0.136
Без Pos. Encoding	xyzθφ	100	-	(64, 128)	28.77	0.924	0.108
Без угла взгляда	xyz	100	10	(64, 128)	27.66	0.925	0.117
Только гр. сеть	xyzθφ	100	10	(256, -)	30.06	0.938	0.109
Очень мало изобр.	xyzθφ	25	10	(64, 128)	27.78	0.925	0.107
Меньше изобр.	xyzθφ	50	10	(64, 128)	29.79	0.94	0.096
Меньше частотн.	xyzθφ	100	5	(64, 128)	30.59	0.944	0.088
Больше частотн.	xyzθφ	100	15	(64, 128)	30.81	0.946	0.096
Полная модель	xyzθφ	100	10	(64, 128)	**31.01**	**0.947**	**0.081**

Ablation study

	Вход	#Im.	L	$(N_{\text{гр}}, N_{\text{точн}})$	PSNR↑	SSIM↑	LPIPS↓
Базовый метод	xyz	100	-	(256, -)	26.67	0.906	0.136
Без Pos. Encoding	xyzθφ	100	-	(64, 128)	28.77	0.924	0.108
Без угла взгляда	xyz	100	10	(64, 128)	27.66	0.925	0.117
Только гр. сеть	xyzθφ	100	10	(256, -)	30.06	0.938	0.109
Очень мало изобр.	xyzθφ	25	10	(64, 128)	27.78	0.925	0.107
Меньше изобр.	xyzθφ	50	10	(64, 128)	29.79	0.94	0.096
Меньше частотн.	xyzθφ	100	5	(64, 128)	30.59	0.944	0.088
Больше частотн.	xyzθφ	100	15	(64, 128)	30.81	0.946	0.096
Полная модель	xyzθφ	100	10	(64, 128)	**31.01**	**0.947**	**0.081**

Ablation study

	Вход	#Im.	L	$(N_{\text{гр}}, N_{\text{точн}})$	PSNR↑	SSIM↑	LPIPS↓
Базовый метод	xyz	100	-	(256, -)	26.67	0.906	0.136
Без Pos. Encoding	xyzθφ	100	-	(64, 128)	28.77	0.924	0.108
Без угла взгляда	xyz	100	10	(64, 128)	27.66	0.925	0.117
Только гр. сеть	xyzθφ	100	10	(256, -)	30.06	0.938	0.109
Очень мало изобр.	xyzθφ	25	10	(64, 128)	27.78	0.925	0.107
Меньше изобр.	xyzθφ	50	10	(64, 128)	29.79	0.94	0.096
Меньше частотн.	xyzθφ	100	5	(64, 128)	30.59	0.944	0.088
Больше частотн.	xyzθφ	100	15	(64, 128)	30.81	0.946	0.096
Полная модель	xyzθφ	100	10	(64, 128)	**31.01**	**0.947**	**0.081**

Выводы

- Результаты применения NeRF визуально качественные и превосходят аналоги по значениям рассмотренных метрик
- Простая модель
  - Базовые алгоритмы 3D рендеринга
  - MLP
- Для каждой сцены отдельная сеть
- Обучение для одной сцены занимают до 2 дней на одной NVIDIA V100

Источники

- Сатья Nerf: Representing scenes as neural radiance fields for view synthesis. (2020) https://arxiv.org/abs/2003.08934
- Видео-демонстрации, исходный код, данные https://www.matthewtancik.com/nerf