

# УНИВЕРЗИТЕТ У БЕОГРАДУ МАТЕМАТИЧКИ ФАКУЛТЕТ



МАСТЕР РАД  
на катедри за  
Рачунарство и информатику

на тему

---

## Примена неуронских поља зрачења у рендеровању

---

Коста Грујчић

Београд, 6. јун 2023.

**Ментор:**

проф. др Младен Николић

Универзитет у Београду, Математички факултет

**Чланови комисије:**

доц. др Јована Ковачевић

Универзитет у Београду, Математички факултет

доц. др Мирјана Маљковић

Универзитет у Београду, Математички факултет

**Наслов мастер рада:** Примена неуронских поља зрачења у рендеровању

**Кључне речи:** машинско учење, неуронска поља, зрачење, рендеровање

**Датум одбране:** 6. јун 2023.

*Својој мајци Јулиани*

# Списак слика

2.1	Интерпретација тачкастог модела камере . . . . .	9
4.1	Илустрација NeRF модела према [1] . . . . .	18
4.2	Поређење NeRF и mip-NeRF модела према [2] . . . . .	21
4.3	Приказ интегралне фуријеизације према [2] . . . . .	25
4.4	Илустративни пример значаја рефлексије вектора погледа око нормали. Необојена (шаховска) поља представљају вектора погледа који не- мају пресек са објектом. Величине са ознаком $r$ у индексу представљају одговарајуће рефлектоване еквиваленте. . . . .	27
4.5	Приказ сферних хармоника и сферне фуријеизације према [3] . . . . .	28
4.6	Поређење mip-NeRF и Ref-NeRF архитектура . . . . .	30
5.1	Одозго на доле, редом LEGO, CHAIR, MATERIALS, DRUMS и SHIP . . . . .	34
6.1	Приказ резултата LEGO скупа података . . . . .	39
6.2	Приказ резултата MATERIALS скупа података . . . . .	40
6.3	Приказ резултата DRUMS скупа података . . . . .	41
6.4	Приказ резултата CHAIR скупа података . . . . .	42
6.5	Приказ резултата SHIP скупа података . . . . .	43
6.6	Поређење NeRF и mip-NeRF модела на DRUMS скупу података . . . . .	44
6.7	Поређење mip-NeRF и Ref-NeRF модела на DRUMS скупу података . . . . .	45
6.8	Поређење mip-NeRF и Ref-NeRF модела на MATERIALS скупу података	45



# Списак табела

6.1	Време обучавања за сваки модел . . . . .	36
6.2	Резултати за LEGO скуп података . . . . .	37
6.3	Резултати за MATERIALS скуп података . . . . .	37
6.4	Резултати за DRUMS скуп података . . . . .	37
6.5	Резултати за CCHAIR скуп података . . . . .	38
6.6	Резултати за SHIP скуп података . . . . .	38



# Садржај

<b>1 Увод</b>	<b>3</b>
<b>2 Основни појмови рачунарске графике</b>	<b>5</b>
2.1 Светлост и боја . . . . .	5
2.2 Камера . . . . .	7
2.2.1 Тачкасти модел камере . . . . .	7
2.3 Рендеровање . . . . .	9
2.3.1 Запреминско рендеровање . . . . .	10
<b>3 Основни појмови машинског учења</b>	<b>13</b>
3.1 Неуронске мреже . . . . .	14
<b>4 Неуронска поља зрачења</b>	<b>17</b>
4.1 NeRF . . . . .	18
4.1.1 Параметризација . . . . .	18
4.1.2 Рендеровање . . . . .	19
4.1.3 Обучавање . . . . .	20
4.2 Mip-NeRF . . . . .	21
4.2.1 Интегрална фуријеизација . . . . .	21
4.2.2 Параметризација и обучавање . . . . .	25
4.3 Ref-NeRF . . . . .	26
4.3.1 Сферна фуријеизација . . . . .	28
4.3.2 Параметризација и обучавање . . . . .	29
4.3.3 Израчунавање вектора нормале . . . . .	30
<b>5 Скупови података</b>	<b>33</b>
<b>6 Експерименти</b>	<b>35</b>
6.1 Метрике . . . . .	35
6.2 Време обучавања . . . . .	36
6.3 Резултати . . . . .	37
6.4 Приказ рендера . . . . .	38
6.5 Детаљно поређење . . . . .	44
<b>7 Закључак</b>	<b>47</b>



# Глава 1

## Увод

Рендеровање је проблем стар колико и рачунарска графика. Постоји небројано пуно различитих алгоритама, приступа, техника и хеуристика за решавање овог важног проблема, а управо је рендеровање покретач низа хардверских унапређења у деценијама иза нас. С друге стране, машинско учење и вештачка интелигенција уопште, су изузетно актуелна област истраживања и примене. Употреба машинског учења у домену рачунарског вида је довела до неслуђених успеха и унапређења чак и у свакодневном животу. Проблеми попут праћења и препознавања очију, разумевање осећања и праћење лица, детекције и распознавања објекта, фотореалистични аватари, па чак и креирање потпуно *нових* слика су тако успешно решени управо употребом машинског учења. Тако се природно дошло до *неуронског рендеровања*, односно, рендеровања сцене употребом неуронских мрежа које су засигурно најпознатија фамилија модела машинског учења.

Неуронска поља зрачења су један од водећих новитета машинског учења, нарочито у оквиру подобласти рачунарског вида. У питању су врсте неуронских мрежа посебно намење за учење векторских поља. Будући да је рендеровање центар интересовања, реч је о специфичној врсти поља којима се описује боја сваке тачке. Неуронском мрежом се моделује пресликавање произвољне тачке простора из неког положаја у боју пиксела рендероване сцене. Дакле, неуронска мрежа се учи над паровима торки које представљају тачке простора и њима придружене боје пиксела при одређеном погледу.

За добијање потпуног рендера из неког погледа, потребно је применити неуронску мрежу за сваку тачку простора. Како је за очекивати да добро обучена неуронска показује известан степен уопштења, овим приступом се могу добити сцене из погледа које уопште нису приступни у подацима. Тиме се могу добити готово глатки прелази из једног погледа у ма који други. У питању су, дакле, врло моћни генеративни модели.

Отуда се и јавила идеја за упоређивањем неких од познатих архитектура ове врсте модела на референтним скуповима података уз пратећа теоријска разматрања. Сваки експеримент се може видети као поступак обучавања неуронског поља зрачења на одређеној сцени, а крајњи циљ је добијање веродостојних рендера из свих могућих углова гледања. Квалитет рендера је описан скупом метрика тако да се различити модели могу поредити на основу објективних бројчаних вредности.

Како је у рачунарству од велике важности и пратећа имплементација, значај овог

рада можемо видети и у чињеници да су сви модели у потпуности имплементирани и то на истоветан начин у смислу да се могу видети као *самостална* библиотека. То за последицу има једноставан поступак обучавања као и употребе.

Један од споредних циљева овог рада је превођење израза који су се безразложно одомаћили у српској научној заједници као англицизми. С друге стране, скраћенице остају у изворном облику и то на латиници, што читаоцу пружа могућност за непосредно претраживање навода.

Битно је напоменути да слике овог рада нису ауторске, већ су преузете из одговарјајућих научних радова који су јавно доступни. Њихов изворни облик је задржан, уз превођење појединих делова уколико је за тим било потребе.

## Глава 2

# Основни појмови рачунарске графике

У овој глави ћемо увести основне појмове рачунарске графике. Почев од светlostи и боје у свом основном, физичком облику, преко синтетичког (идеалног) модела камере па до рендеровања које представља централни појам овог рада.

### 2.1 Светлост и боја

Светлост представља електромагнетно зрачење чија је таласна дужина у интервалу од око 350 до 700 nm, које побуђује визуелни систем човека. То значи да људи не виде светлост осталих таласних дужина, тако да ћемо убудуће под видљивим спектром светлости мислити управо на овај, видљив човеку.

Потребно је увести физичке величине којима се могу квантификовати основна физичка својства светлости. Прво ћемо дефинисати радиометријске величине, а онда одговарајуће фотометријске.

**Дефиниција 2.1.1.** *Фотон је квант електромагнетног зрачења. Енергија фотона је:*

$$E = \frac{hc}{\lambda} [\text{J}],$$

где је  $h$  Планкова константа,  $c$  брзина светлоти, а  $\lambda$  таласна дужина фотона.

**Дефиниција 2.1.2.** *Укупна енергија зрачења извора зрачења је:*

$$Q_e = \int_{S^2} E d\Omega [\text{J}].$$

**Дефиниција 2.1.3.** *Флукс зрачења је укупна енергија зрачења која доспе на неку површину по јединици времена:*

$$\Phi_e = \frac{\partial Q_e}{\partial t} [\text{W}].$$

**Дефиниција 2.1.4.** *Озраченост је укупан флукс зрачења по јединици површине:*

$$E_e = \frac{\partial \Phi_e}{\partial A} \left[ \frac{\text{W}}{\text{m}^2} \right].$$

**Дефиниција 2.1.5.** Јачина зрачења је укупан флукс зрачења у неком смеру по јединичном просторном углу:

$$I_{e,\Omega} = \frac{\partial\Phi_e}{\partial\Omega} \left[ \frac{\text{W}}{\text{sr}} \right].$$

**Дефиниција 2.1.6.** Зрачење је укупан флукс зрачења у неком смеру по јединици површине и јединичном просторном углу:

$$L_{e,\Omega} = \frac{\partial^2\Phi_e}{\partial\Omega\partial A} \left[ \frac{\text{W}}{\text{sr} \cdot \text{m}^2} \right].$$

Спектралне величине се дефинишу у односу на таласну дужину. Дакле, уместо да поменуте величине посматрамо неовисно од таласне дужине, сада сваку од њих посматрамо у односу на тачно одређену таласну дужину. На пример, спектрално зрачење је  $L_{e,\Omega,\lambda} = \partial L_{e,\Omega}/\partial\lambda$ . Аналогно и за остале – тако добијамо *флукс спектралног зрачења, спектралну озраченост, јачину спектралног зрачења и спектрално зрачење*.

Након дефинисања мноштва физичких величина, коначно долазимо и до фотометријских, или како се још називају и *визуелне*. Кључна разлика у односу на радиометријске, или како се још називају и *енергетске*, је што се у овом случају у обзир узима и спектрална осетљивост посматрача. То одговара интуитивном поимању светлости – човек светлост разликује на основу боје, што је у директној коресподенцији са таласном дужином.

У основи спектралне зависности посматрача је *функција релативне светлосне осетљивости*  $V$ . Помоћу ове функције изражавамо просечну осетљивост човека на светлост одређене таласне дужине. Просечна је у смислу да може варирати у популацији, али представља врло добру апроксимацију у општем случају, поготово имајући у виду да је стандардизована од стране Међународног комитета за осветљење.

**Дефиниција 2.1.7.** Светлосни флукс је укупна енергија која протекне кроз неку површину у јединици времена:

$$\Phi_v = K \int_0^\infty \Phi_{e,\lambda} V(\lambda) d\lambda [\text{lm}].$$

За вредност константе  $K$  из 2.1.7 се узима  $683.002 \frac{\text{lm}}{\text{W}}$ . Реч је о још једном примеру стандардизације у области фотометрије.

**Дефиниција 2.1.8.** Јачина светлости је укупна снага коју емитује извор светлости у одређеном смеру по јединичном просторном углу:

$$I_v = K \int_0^\infty I_{e,\lambda} V(\lambda) d\lambda [\text{cd}].$$

**Дефиниција 2.1.9.** Осветљеност је укупан светлосни флукс на некој површини:

$$E_v = K \int_0^\infty I_{e,\lambda} E_{e,\lambda} V(\lambda) d\lambda [\text{lx}].$$

**Дефиниција 2.1.10.** Сјајност је укупан светлосни флукс који напушта, пролази или пада на површину по јединичном просторном углу и по ортогоналној пројекцији јединичне површине:

$$L_v = \frac{\partial^2 \Phi_v}{\partial \Omega \partial A \cos \theta} \left[ \frac{\text{cd}}{\text{m}^2} \right].$$

Сјајност је једина фотометријска величина коју човек непосредно опажа. Она представља мерило за субјективни утисак о мањој или већој сјајности светлеће или осветљене површине.

Већ смо поменули да људи разликују неке таласне дужине светlostи, односно виде светlost одређене боје. Ту способност нам дају три врсте чепића који се налазе у жутој мрљи мрежњаче ока. Према томе, кроз ове рецепторе, наш мозак прима свега три врсте сигнала за сваки очни надражај. Међутим, људско око није идеалан спектрометар, па поједине различите таласе просто види као светlost исте боје.

Када је реч о представљању светlostи на рачунару, јасно је да ма како то урадили ипак дискретизујемо простор видљиве светlostи. Имајући у виду да и наше око чини нешто слично, дискретизација је сасвим оправдана. Како смо већ говорили о осетљивости на три карактеристичне таласне дужине, определићемо се за представљање боја тројкама ( $R, G, B$ ) где су координате удео црвене, зелене и плаве боје редом, које се узимају за основне. Постоје и други системи боја попут HSV или CMYK, али о њима неће бити речи јер ћемо надаље користити искључиво RGB систем.

На овај начин је могуће чувати тачно  $255^3$  боја, што је и више него довољно будући да људско око разликује до 10 милиона боја. Даља практична ограничења се тичу квалитета монитора као и графичког процесора, али се тиме нећемо бавити.

## 2.2 Камера

Можемо сматрати да нам је појам камере познат из стварног света. У овој глави ћемо строго дефинисати камеру и дати један од многобројних начина њеног моделовања у домену рачунарске графике.

**Дефиниција 2.2.1.** Камера је пресликавање из  $\mathbb{R}^3$  у  $\mathbb{R}^2$ .

У питању је врло општа, готово бескорисна дефиниција. Међутим, ако размислимо о томе да се стваран свет врло добро може представити као простор димензије 3, а да се фотографија може схватити као раван, можемо увидети да је у питању заиста исправна формулатија. Оно што овом дефиницијом није обухваћено јесте како се тачно од света око нас долази до слике. Зато има смисла говорити о моделима камере као различитим парадигмама изведбе поменутог пресликавања. У овом раду ће највише бити коришћен тачкасти модел камере тако да ћемо у наставку поглавља дати његову прецизну дефиницију.

### 2.2.1 Тачкасти модел камере

Посматрајмо канонски Еуклидски простор димензије 3 и раван  $z = f$  коју ћемо звати **раван слике**. Вредност  $f$  мора бити позитивна и зове се **жисижна даљина**.

На слици 2.1 је дат пратећи приказ. У овом моделу камере се произвољна тачка  $\mathbf{x}_w = (x, y, z)$  из простора пресликају у тачку  $\mathbf{x}_p = (u, v)$  која је тачка пресека равни слике и праве која спаја  $\mathbf{x}_w$  и центар камере  $\mathbf{c}$ , који ћемо за сад поставити у координатни почетак. Другим речима, у питању је централна пројекција са центром у координатном почетку. Тривијалном применом сличности троуглова долазимо до

$$(x, y, z) \mapsto (fx/z, fy/z). \quad (2.1)$$

Праву која пролази кроз центар камере и нормална је на раван слике називамо **главном осом**, а тачку у којој се главна оса и раван слике секу називамо **главном тачком**.

Приметимо још једну особину централног пројектовања – све тачке праве која пролази кроз центар камере се пројектују у исту тачку равни слике. Зато ћемо увести хомогене координате. Отуда можемо писати  $\mathbf{x}_p = P\mathbf{x}_w$ . Овакво пресликање се може преписати и у матричном облику

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_P \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (2.2)$$

Ради сагласности са наставком, раставићемо  $P$  на прозивод

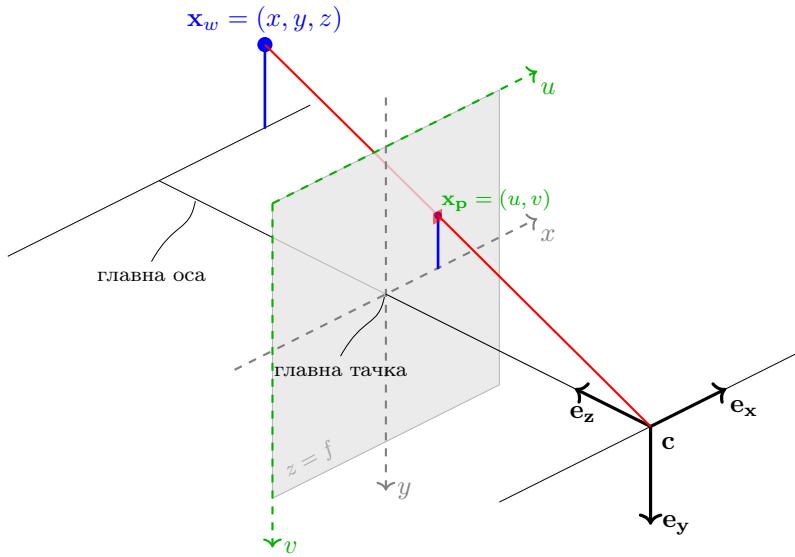
$$K_{3 \times 4} \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix}_{4 \times 4} \quad (2.3)$$

при чему су све наведене матрице реалне. Матрица  $K$  се назива **матрица калибраџије**, а њене вредности **унутрашњим параметрима** камере. С друге стране, вредности матрица  $R$  и  $t$  се називају **спољашњим параметрима** камере. Како је у једнакости 2.2 матрица  $R$  јединична, а  $t$  тривијална, матрица  $K$  се добија уклањањем последње колоне, чиме се практично поклапа са  $P$ .

Уколико уопштимо положај главне тачке и њене координате означимо са  $(p_x, p_y)$ , матрица  $K$  поприма облик

$$\begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (2.4)$$

Конечно, можемо уопштити и положај центра камере. Тада уочавамо два координатна система - онај с почетка, канонски, у ком нам је лако да баратамо и други, са центром камере као координатним почетком. Другим речима, померамо се из  $(0, 0, 0)$  у  $\mathbf{c}$ . Како је у питању кретање (чувамо оријентацију и дужину) репер камере се ротацијом и трансляцијом доводи до канонског. На тај начин употребљавамо пресликање  $P$ .



Слика 2.1: Интерпретација тачкастог модела камере

Постоји укупно 9 слободних параметара – по 3 за сваку од матрица  $K$ ,  $R$  и  $t$ . Треба истаћи да ошти облик тачкастог модела камере броји укупно 10 параметара – преостали параметар носи име **коефицијентискривљености** и уводи се уколико осе равни слике нису нормалне. Како то у овом раду неће бити случај, поменути параметар ће подразумевано бити 0. Иначе, у питању је број у пресеку прве врсте и друге колоне матрице калибрације.

Поменимо још једну терминолошку конвенцију. Координатни систем из којег вршимо пројекцију се назива **светским координатним системом**, а координатни систем индукован положајем камере **координатни систем камере**. **Координатни систем слике** је онај везан за раван слике.

Корисно је дати додатни коментар у вези са нулом у првом реду матрице калибрације. У питању је коефицијент смицања којим је могуће уопштити раван слике – правоугаоник постаје паралелограм.

## 2.3 Рендеровање

Када смо говорили о камери, видели смо да слика није ништа друго до дводимензиона пројекција стварног света. Слично је и када комплексну сцену, овог пута виртуелног, света желимо да прикажемо на монитору. Мора доћи до пројектовања света, а слику коју видимо је уствари слика добијена посматрањем света кроз виртуелну камеру. Тада процес се назива **рендеровање**. Дакле, рендеровањем се од сцена долази до конкретне репрезентације слике у пикселима.

Пре него што се упушимо у рендеровање, вратимо се корак уназад. Светлост и камеру смо, на први поглед, увели врло неповезано. Светлост смо дефинисали више експериментално ослањајући се на физичке законе, док смо камеру увели строго геометријски. Сада ћемо оба појма ставити у контекст и објаснити њихову везу.

Праве које смо уочавали у моделу камере су светлосни зраци. Раван слике у прак-

си није бесконачна, већ има своју ширину и дужину којима се одређују димензије слике. Такође, услед физичких ограничења слика се мора дискретизовати. Посматрамо ли слику на тај начин, видећемо да је она матрица чије елементе називамо **пикселима**. Како ће који пиксел да изгледа, односно које ће боје бити, одређује количина зрачења која допре до камере. Зрачење смо увели у зависности од правца, што у комбинацији са пројективним особинама модела камере одређује који ће пиксел бити *погођен*. Такође, природно је увести ограничење по питању дела простора који камера може да опажа. Величину слике смо већ поменули, али не и параметар по  $z$  оси. Ради једноставности геометрије, уводимо **предњу и задњу раван одсецања**. Све испред предње и иза задње равни одсецања не утиче на резултатујућу слику.

### 2.3.1 Запреминско рендеровање

Посматрајмо шта се дешава са светlostи приликом проласка кроз неку средину. Може доћи до:

- упијања – средина упија фотоне светлосног зрака приликом чега се ослобађа топлота или неки други вид енергије.
- емисије – како је светлост зрачење, пролазак светла кроз средину је загрева. Када средина достигне одређену температуру, може доћи до емитовања светлости.
- расипања – део фотона напушта правац зрака што доводи до мешања фотона са различитих праваца.

Према томе, више чинилаца утиче на коначно зрачење које ће путем неког светлосног зрака доћи до камере. Тачке неког светлосног зрака с почетком у  $\mathbf{o}$  и правцем  $\mathbf{d}$  једнозначно одређујемо као  $\mathbf{r}_{\mathbf{o},\mathbf{d}}(t) = \mathbf{o} + t\mathbf{d}$ .

Нека су предња и задња раван одсецања редом  $z = t_n$  и  $z = t_f$ , а светлосне зраке посматрамо из положаја камере  $\mathbf{c}$ . Према томе, до камере дуж зрака  $\mathbf{r}_{\mathbf{c},\mathbf{d}}$  допире следећа количина зрачења

$$\int_{t_n}^{t_f} T(t) \sigma(\mathbf{r}_{\mathbf{c},\mathbf{d}}(t)) L_{e,\Omega} \Big|_{\mathbf{r}_{\mathbf{c},\mathbf{d}}(t)} dt, \quad (2.5)$$

где је  $T(t) = \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}_{\mathbf{c},\mathbf{d}}(s)) ds\right)$  акумулирана пропусност зрака од  $t_n$  до  $t$ . Ова величина представља вероватноћу да светлосни зрак на путу од  $t_n$  до  $t$  не удари нити у једну препреку. Напоменимо да се интеграција врши искључиво по фотонима који се налазе на светлосном зраку од интереса, или другим речима, интеграли се само по расутој светлости.

Како нам је боја пиксела крајњи циљ, једначину 2.5 ћемо преписати у колориметријском облику

$$C(\Pi_{\mathbf{r}_{\mathbf{c},\mathbf{d}}}) = \int_{t_n}^{t_f} T(t) \sigma(\mathbf{r}_{\mathbf{c},\mathbf{d}}(t)) C(\mathbf{r}_{\mathbf{c},\mathbf{d}}(t)) dt, \quad (2.6)$$

где је  $\Pi_{\mathbf{r}_{\mathbf{c},\mathbf{d}}}$  тачка пресека светлосног зрака и равни слике, а  $C$  поље које сваку тачку пресликава у њену RGB боју.

Приказани поступак одређивања боје пиксела се назива *запреминско рендеровање*. У питању је само један од многобројних алгоритама за рендеровање који ће бити коришћен у наставку.



## Глава 3

# Основни појмови машинског учења

У овој глави ћемо формулисати теоријски оквир неопходан за разматрање и примену машинског учења.

Машинско учење је област *вештачке интелигенције*. Неформално говорећи, машинско учење обухвата алгоритме изведене из података или како се то често говори – научене из података. То значи да нема експлицитног програмирања резултујућег алгоритма, а врло често ни контроле процеса учења, већ се алгоритми дефинишу низом операција параметризованих на основу података који су изложени процесу обучавања. Процес обучавања, дакле, представља одређивање поменутих параметара којима се операције од значаја израчунавају на што исправнији начин. Приметимо да је та исправност прилично неодређен појам и зависи од примене, података и циља обучавања. Такође, врло се често не може ни квантifikовати што проблем обучавања чини утолико тежим.

Имајући у виду досег овог рада, потребно је увести појам надгледаног учења. Реч је о врсти учења у којој су уз податке присутни и додатни подаци којима је директно могуће утврдити да ли је излаз алгоритма исправан или није. На пример, ако је потребно утврдити да ли је на датој слици мачка, уз слику би постојао једнобитни податак који то недвосмислено потврђује. Према томе, скуп података можемо посматрати као скуп одбира  $\mathcal{D} = \{(x_i, y_i)\}$ , где  $x_i$  представља улаз, а  $y_i$  одговарајући излаз. То нам омогућава да посматрамо расподелу таквих одбира, односно густину расподеле  $p(x, y)$ . Природно се намеће потреба за што приближнијем одређивању поменуте расподеле, што подразумева одређивање функције  $f$  којом успостављамо везу између одговарајућих парова  $x$  и  $y$ . Кандидата за  $f$  има несагледиво много, а нама је потребна она *најбоља*, при чему се овог пута то мора формално дефинисати.

Претпоставимо да су  $y_i$  узорковани из метричког простора. Зато можемо дефинисати *функцију грешке*  $\mathcal{L}$  којом меримо квалитет апроксимације  $y_i$  вредношћу  $f(x_i)$ . Вреднујући одбирке у складу са својом густином дефинишемо *ризик*

$$\mathcal{R}(f) = \mathbb{E}(\mathcal{L}(y, f(x))) = \int \mathcal{L}(y, f(x)) p(x, y) dx dy. \quad (3.1)$$

Проблем надгледаног учења покушава да дође до  $f$  за које се ризик минимизује. Начелно, обучавање се може извести по свим могућим функцијама  $f$ . Како је то практично немогуће, претрага функција се усмерава увођењем додатних претпоставки, односно рестрикција, скупа претраге. Ми ћемо посматрати функције параметризо-

ване скупом параметара  $\Theta$ , тако да минимизовање ризика посматрамо као

$$\min_{\Theta} \mathcal{R}(f_{\Theta}). \quad (3.2)$$

У општем случају не мора постојати само један исправан излаз за конкретан улаз, што оправдава дефинисање ризика коришћењем заједничке расподеле  $p(x, y)$ . То за наше потребе неће бити неопходно. Наиме, посматраћемо условну расподелу  $p(y|x)$ .

Расподела  $p(y|x)$  је ретко кад позната. Зато се спроводи стандардни статистички третман – ризик се мења емпириским ризиком:

$$\mathcal{R}_e(f_{\Theta}, \mathcal{D}) = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \mathcal{L}(y_i, f_{\Theta}(x_i)), \quad (3.3)$$

где је  $\mathcal{S}$  узорак скупа  $\mathcal{D}$ .

Претпоставимо да се скуп параметара састоји од само једног вектора димензије свега 2, тј.  $\Theta = \{\mathbf{w}\} = \{(w_1, w_2)\}$ , а да је  $x_i \in \mathbb{R}^1$ . Имајући у виду израз 3.2, за кандидате функције  $f$ , између остalog, имамо  $\sin(x_i w_1 + w_2)$ ,  $\log(x_i w_1 w_2)$ ,  $\exp(w_2 x_i^{w_1})$ . Иако је проблем постављен као веома једноставан, а параметарском рестрикцијом начињен још једноставнијим, и даље се чини као веома тежак будући да немамо начин да за разумно коначно времена претражимо све такве кандидате. Зато је потребно посматрати тачно одређене класе функција  $f$  за које је то могуће, а међу најпознатијим су свакако **неуронске мреже**. Заправо, потребно је фиксирати конкретну архитектуру модела, а потом вршити оптимизацију параметара у складу са одабраном метриком.

## 3.1 Неуронске мреже

**Дефиниција 3.1.1.** *Неуронска мрежа је функција облика*

$$f_{\Theta}(\mathbf{x}) = \left( \prod_{i=1}^L a_i (\beta_i + W_i) \right) (\mathbf{x}),$$

где је  $\Theta = \{W_i\}_{i=1}^L$  скуп параметара који се обучава,  $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$  при чему је  $d_0 = \dim(\mathbf{x})$  и  $\{a_i\}_{i=1}^L$  функције које се примењују члан по члан. Функције  $\{a_i\}_{i=1}^L$  називамо **активационим функцијама**. Параметар  $L$  називамо **бројем слојева неуронске мреже**.

Активационе функције могу бити произвољне, докле год поштују услове димензионалности. Међутим, уколико су оне линеарне, тада неуронска мрежа постаје линеарна, односно постаје линеарна регресија. У циљу добијања што разноврснијих модела, за функције активације се узимају нелинеарне функције. Један од популарних избора је  $\text{ReLU}(x) = \max\{0, x\}$  [4].

Неуронске мреже су познате и под именом **вишеслојни перцептрони**, па ће убудуће бити коришћена и скраћеница MLP (енг. *multilayer perceptron*). Управо због овакве слојевите структуре активације једног слоја сматрамо његовим излазом, а уједно и улазом наредног слоја.

Значајан теоријски резултат даје следећа теорема [5]

**Теорема 3.1.1.** За сваку функцију  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  интеграбилну у Бохнеровом смислу и свако  $\epsilon > 0$ , постоји неуронска мрежа  $f_\Theta$  са ReLU активационим функцијама, тако да је  $\{d_i = \max\{n+1, m\}\}_{i=1}^L$  и важи

$$\int_{\mathbb{R}^n} \|f_\Theta(x) - F(x)\| dx < \epsilon.$$

Ово за последицу има да се готово свака функција може произвољно добро апроксимирати неуронском мрежом, али како доказ није конструкцијан није очигледно како таква неуронска мрежа изгледа.

Уколико су активационе функције диференцијабилне функције, то ће бити и неуронска мрежа. Зато се поступак минимизације емпиријског ризика спроводи градијентним спустом. Поступак се спроводи у две етапе – прво се неуронска мрежа примени над подацима пропагацијом унапред, а потом се пропагацијом уназад, користећи правилом извода сложене функције, изврши ажурирање параметара.

Како је градијентни спуст врста локалне претраге, није гарантовано да ће пронађени локални минимум бити и глобални. У пракси се показује да градијентни спуст даје веома добра решења.

Када је реч о имплементацији неуронских мрежа и конкретим детаљима њиховог обучавања, важно је разматрати хардверска ограничења. Наиме, ови модели могу бити веома комплексни где се ред величине броја параметара креће и до неколико милиона, па чак и до неколико милијарди. Зато се приликом обучавања прибегава разнородним техникама, често нумериčке или хеуристичке природе, којима се смањује укупно време обучавања или смањује утрошак меморије. Веома познат пример је *стохастички градијентни спуст* [6] – параметри се ажурирају на основу само једног одбира уместо целог скупа података. Како се у пракси обично, готово увек, обучавање врши на графичком процесору, ради што бољег искоришћења се врши компромис између конвенционалног и стохастичког градијентног спуста у смислу употребе већег броја одбира.



## Глава 4

# Неуронска поља зрачења

Ова глава представља средишњи део рада. Почекемо са важним дефиницијама, а потом размотрити различите архитектуре наводећи њихова важна својства и објашњавајући начин обучавања.

**Дефиниција 4.1.** *Поље је пресликавање  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ . Специјално, поље је скаларно за  $m = 1$ .*

**Дефиниција 4.2.** *Неуронско поље је поље макар делимично параметризовано неуронском мрежом.*

Неуронско поље зрачења је посебан случај неуронског поља за  $n = 5$  и  $m = 4$ . У питању је параметризација пресликавања које свакој тачки  $(x, y, z)$  придружује зрачење и пропусност и то за сваки правац одређен угловима  $\theta$  и  $\phi$ . Имајући у виду да на сликама видимо боје, зрачење је управо из видиљивог дела спектра, па ћемо га и представити као светлост на већ описан начин – тројкама  $(R, G, B)$ . Пропусност се може разумети и као вероватноћа да се зрак *зауставља* у тој тачки и убудуће ћемо га означавати са  $\sigma$ . Дакле, неуронско поље зрачења параметризује пресликавање  $(x, y, z, \theta, \phi) \rightarrow (R, G, B, \sigma)$ .

Размотримо шта добијамо оваквом поставком. Претпоставимо да је неуронско поље савршено обучено. То значи да можемо утврдити боју пиксела из сваког могућег угла гледања. Тако се сцена рендерује и из погледа који се нису нашли у скупу за обучавање.

Како је таква неуронска параметризација неодређена *a priori*, имамо слободу у виду дизајнирања архитектуре параметризације. Подробно ћемо обрадити оригинални NeRF модел и његово проширење mip-NeRF, а потом се осврнути на специфичан Ref-NeRF који је прилагођен тачно одређеним условима.

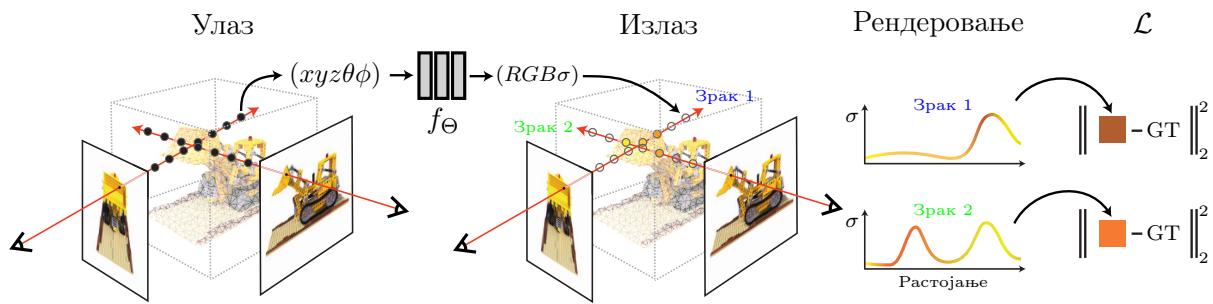
Оно што на први поглед издаваја неуронска поља зрачења од осталих параметризација неуронским мрежама јесте обучавање. Узмимо за потребе илустрације ImageNet [7] скуп података. Поменути скуп је сачињен од великог броја троканалних слика, тако да је сваки узорак изворног скupa једноставан за представити – слике су међусобно независне и по учитавању се могу лако груписати у тензоре у складу са количином меморије која нам је на располагању. Међутим, скуп података којим ми баратамо је сачињен од различитих погледа, а уз сваки поглед је придржана одговарајућа изрендерована слика. Имајући у виду улаз неуронског поља зрачења, скупу података је неопходно увести још један слој гранулације – зрак за сваки угао гледања

у одређеном погледу. Дакле, један одбирац из скупа података над којим се обучава неуронско поље зрачења се може видети као један светлосни зрак и то за један поглед уз пратећи рендер.

## 4.1 NeRF

NeRF (енг. *Neural Radiance Field*) је најстарији модел у фамилији модела који ћемо у овом раду обрадити. Заснива се на запреминском рендеровању и прва је изведба неуронских поља зрачења овог типа.

На слици 4.1 се може видети илустрација овог модела.



Слика 4.1: Илустрација NeRF модела према [1]

### 4.1.1 Параметризација

Неуронско поље зрачења се параметризује вишеслојним перцептроном, али на благо неубичајен начин. Јасно је да  $\sigma$  не зависи од угла гледања већ искључиво од положаја  $\mathbf{x}$ . С друге стране, боја пиксела зависи од положаја као и од пропусности. Зато ReLU MLP од 8 слојева са по 256 неурона на улазу добија само  $\mathbf{x}$ , док је излаз предвиђена вредност  $\sigma$  и вектор димензије 256. Тада вектор се спаја са параметрима угла гледања и пропушта кроз један слој са 128 неурона и ReLU активационом функцијом. На овај начин се добија боја пиксела која је условљена погледом. Поменуте неуронске мреже називамо редом *просторни* и *усмерени* MLP.

Чак и уз овако дефинисан поступак добијања коначног излаза, улаз и даље има малу димензију. Показано је да неуронске мреже нису у стању да науче високо-фреквентна својства из улаза ниске димензије [8]. Један од начина да се наведени проблем, познат као *спектрална пристрасност*, ублажи, јесте да се улази прво пресликају у простор веће димензије. Такво подизање не може бити произвољно, већ се мора водити рачуна о одређеним својствима. У овом тренутку се нећемо задржавати на конкретним детаљима анализе, већ смо се позвати на резултате из [10] којима се предлаже пресликање облика

$$\gamma(x) = (a_1 \cos(2\pi b_1 x), a_1 \sin(2\pi b_1 x), \dots, a_m \cos(2\pi b_m x), a_m \sin(2\pi b_m x)). \quad (4.1)$$

На овај начин се реалан број  $x \in [-1, 1]$  пресликава на површину хиперелипсоида у простору димензије  $2m$ . Конкретне вредности параметара  $\{a_1, b_1, \dots, a_m, b_m\}$  зависе од домена примене.

Према томе, неуронску мрежу  $F_\Theta$  можемо видети као композицију  $F'_\Theta \circ \gamma$  где се  $\gamma$  не обучава. У овом случају је  $\gamma : \mathbb{R} \rightarrow \mathbb{R}^{2D}$  и то конкретно

$$\gamma(x) = (\sin(2^0 \pi x), \cos(2^0 \pi x), \dots, \sin(2^{D-1} \pi x), \cos(2^{D-1} \pi x)). \quad (4.2)$$

Исто пресликање се користи за све улазе које MLP добија и примењује покоординатно, с тим што се за положај узима  $D = 10$ , а за поглед  $D = 4$ . Важно је поменути да су улазни вектори  $\mathbf{x}$  и  $\mathbf{d}$  нормализовани на  $[-1, 1]$ . Описани поступак се назива *фуријеизација*, а инспирисан је резултатима из [9].

### 4.1.2 Рендеровање

За рендеровање сцене користимо једначином 2.6. Интеграл ћемо апроксимирати Римановом сумом. У питању је појам тесно повезан са дефиницијом (Римановог) интеграла, али ћемо дефиницију дати без улажења у такве детаље. Упростено, површину испод графика функције ћемо апроксимирати сабирањем површина правоугаоника.

**Дефиниција 4.1.1.** Риманова сума ограничено реалне функције  $f$  дефинисана на сегменту  $[a, b]$  је

$$\sum_{i=1}^n f(\xi_i)(x_i - x_{i-1}),$$

зде је  $x_0 = a < x_1 < \dots < x_n = b$  подела  $[a, b]$ , а  $\xi_i$  изабрана тачка сегмента  $[x_{i-1}, x_i]$ .

Уколико користимо детерминистички приступ, одбирци који учествују у апроксимацији ће увек бити исти. То је неповољан приступ будући да би у том случају неуронско поље зрачења било увек узорковано за исти скуп параметара. Уместо тога, увешћемо еквидистантну поделу сегмента  $[t_n, t_f]$ , а потом насумице одабрати по један број из сваког елемента поделе. То формално исказујемо наредним изразом

$$t_i \sim \mathcal{U} \left[ t_n + \frac{i-1}{N}(t_f - t_n), t_n + \frac{i}{N}(t_f - t_n) \right]. \quad (4.3)$$

Препишемо једначину 2.6 у дискретном облику, у складу са 4.3

$$\hat{C}(\Pi_{\mathbf{r}_{\mathbf{c}, \mathbf{d}}}) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \Delta_i)) \mathbf{c}_i, \quad (4.4)$$

зде је  $T_i = \exp \left( -\sum_{j=1}^{i-1} \sigma_j \Delta_j \right)$ , парови  $(\mathbf{c}_i, \sigma_i)$  одговарајући у складу са избором  $t_i$ , а  $\Delta_i$  ширина елемента поделе.

Једначине 2.6 и 4.4 се видно разликују по начину на који учествује пропусност. Наиме, у складу са дефиницијом Риманове суме 4.1.1 би се појавио линеаран члан  $\sigma_i \Delta_i$ . Тада је је у апроксимацији замењен експоненцијалним  $1 - \exp(-\sigma_i \Delta_i)$ . У наставку ћемо објаснити и зашто [11].

Како узимамо да су на сваком елементу поделе боја и пропусност константни, једначину 2.6 расписујемо у складу са тим. Погледајмо како произвољни елемент поделе учествује у апроксимацији.

$$\begin{aligned}
 \hat{C}_i(\Pi_{\mathbf{r}_{\mathbf{c}, \mathbf{d}}}) &= \int_{t_i}^{t_{i+1}} T(t) \sigma_i c_i dt \\
 &= \int_{t_i}^{t_{i+1}} \sigma_i c_i \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}_{\mathbf{c}, \mathbf{d}}(s)) ds\right) dt \\
 &= \sigma_i c_i \int_{t_i}^{t_{i+1}} \exp\left(-\int_{t_n}^{t_i} \sigma(\mathbf{r}_{\mathbf{c}, \mathbf{d}}(s)) ds\right) \exp\left(-\int_{t_i}^{t_{i+1}} \sigma_i ds\right) dt \\
 &= T_i \sigma_i c_i \int_{t_i}^{t_{i+1}} \exp(-\sigma_i(t - t_i)) dt \\
 &= T_i \sigma_i c_i \frac{\exp(-\sigma_i(t - t_i))}{-\sigma_i} \Big|_{t_i}^{t_{i+1}} \\
 &= T_i(1 - \exp(-\sigma_i \Delta_i)) c_i.
 \end{aligned} \tag{4.5}$$

### 4.1.3 Обучавање

Имплементација инспирисана описаним поступком рендеровања је врло неефикасна – празан простор и прикривени региони не доприносе рендерованој слици, а непрестано се узоркују. Из тог разлога се користе два неуронска поља, за *фини* (у ознаки  $f$ ) и *груби* (у ознаки  $c$ ) предвиђања. Прво се узоркује  $N_c$  одбирача на начин објашњен у 4.3 који се потом дају грубом пољу. На основу грубих предвиђања се врши још једно узорковање, али овог пута боље навођено. Фино узорковање је пристрасније релевантним деловима простора.

Да би то спровели у дело, излаз грубог неуронског поља записујемо као збир боја дуж зрака пондерисаног непрозирношћу

$$\hat{C}_c(\Pi_{\mathbf{r}_{\mathbf{c}, \mathbf{d}}}) = \sum_{i=1}^{N_c} w_i c_i, \quad w_i = T_i(1 - \exp(-\sigma_i \Delta_i)). \tag{4.6}$$

Скалирањем тежина  $\hat{w}_i = w_i / \sum_{j=1}^{N_c} w_j$ , добија се део-по-део константна густина расподеле дуж зрака која се може представити у следећем облику

$$f_c(t) = \frac{\hat{w}_i}{\Delta_i} \text{ за } t_i \leq t < t_{i+1}. \tag{4.7}$$

Већ смо рекли да фино узорковање добијамо бољим навођењем грубог, односно, одбираци неће бити равномерно распоређени дуж зрака, већ у складу са тежином која је придружена одговарајућим интервалима. На овај начин интервали са већим тежинама више доприносе у фином узорку. Имајући у виду како су се тежине израчунавале, можемо закључити да делови зрака веће пропусности имају и већу тежину.

Да би дошли до такве фине поделе, потребно је извршити инверзно узорковање из поменуте део-по-део константне расподеле. То подразумева израчунавање  $F_c^{-1}(p_i)$ , где су  $p_i$  равномерно распоређени бројеви на јединичном интервалу, а  $F_c^{-1}$  инверзна функција расподеле чија је густина расподеле  $f_c$ . Потом се фино неуронско поље

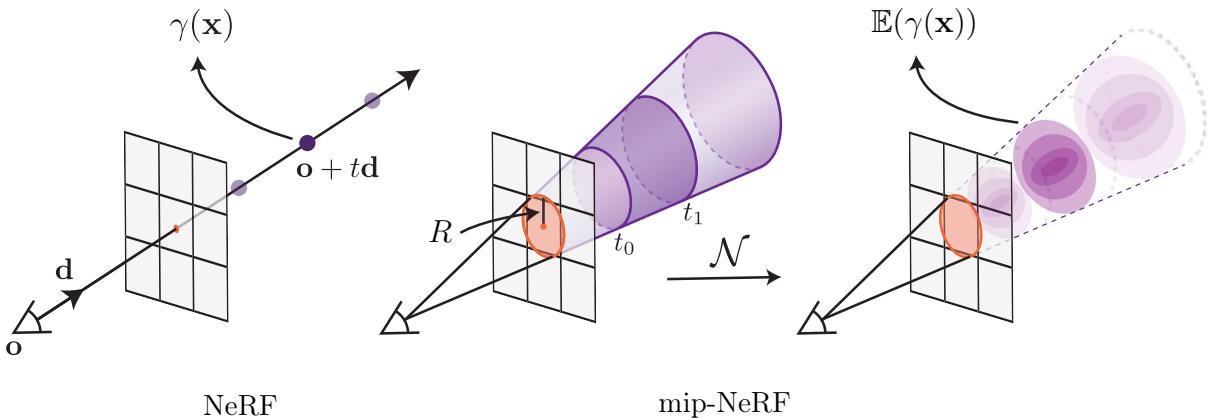
израчунава у свих  $N_c + N_f$  одбирача на начин описан у једначини 4.4. У свим експериментима се за  $N_c$  узима 64, а за  $N_f$  128.

Функција губитка која се користи за обучавање NeRF-а је

$$\mathcal{L} = \sum_{\mathbf{r}_{c,d} \in \mathcal{B}} \left( \left\| \hat{C}_c \left( \Pi_{\mathbf{r}_{c,d}^c} \right) - C \left( \Pi_{\mathbf{r}_{c,d}^c} \right) \right\|_2^2 + \left\| \hat{C}_f \left( \Pi_{\mathbf{r}_{c,d}^f} \right) - C \left( \Pi_{\mathbf{r}_{c,d}^f} \right) \right\|_2^2 \right). \quad (4.8)$$

## 4.2 Mip-NeRF

Као што смо могли да видимо у претходној глави, NeRF је далеко од идеалног теоријског модела и показује извесне недостатке. То су постојање два модела који се упоредо обучавају, узорковање дуж само једног зрака за сваки пиксел, а као што ћемо видети касније у табели 6.1, обучавање траје данима. Mip-NeRF архитектуром покушаћемо да превазиђемо или макар ублажимо наведено.



Слика 4.2: Поређење NeRF и mip-NeRF модела према [2]

Кренућемо од кључног недостатка – постојање само једног зрака по пикселу. Природно проширење је конусно емитовање зрака. Конус ћемо ограничити двема равнима чиме добијамо зарубљену купу која је једнозначно одређена центром  $\mathbf{o}$ , осом симетрије  $\mathbf{d}$  и полупречником круга  $R$  који је пресек поменуте купе и равни слике. Скуп тачака које припадају зарубљеној купи на сегменту параметра  $[t_0, t_1]$  је одређен карактеристичном функцијом

$$F(\mathbf{x}, \mathbf{o}, \mathbf{d}, R, t_0, t_1) = I \left( t_0 < \frac{\mathbf{d}^T (\mathbf{x} - \mathbf{o})}{\|\mathbf{d}\|_2^2} < t_1 \right) \wedge \left( \frac{\mathbf{d}^T (\mathbf{x} - \mathbf{o})}{\|\mathbf{d}\|_2 \|\mathbf{x} - \mathbf{o}\|_2} > \frac{1}{\sqrt{1 + (R/\|\mathbf{d}\|_2)^2}} \right). \quad (4.9)$$

### 4.2.1 Интегрална фуријеизација

Једна од, наизглед, не нарочито битних одлика NeRF-а је допринос фуријеизације улаза на начин објашњен у 4.2. Зато ћемо посебну пажњу посветити управо томе.

Главни циљ је задржати изворну фуријеизацију. Најједноставнији приступ је одређивање очекиване изворне фуријеизације по свим тачкама које припадају купи зрака. Тако добијамо

$$\gamma^*(\mathbf{o}, \mathbf{d}, R, t_0, t_1) = \frac{\int \gamma(\mathbf{x}) F(\mathbf{x}, \mathbf{o}, \mathbf{d}, R, t_0, t_1) d\mathbf{x}}{\int F(\mathbf{x}, \mathbf{o}, \mathbf{d}, R, t_0, t_1) d\mathbf{x}}. \quad (4.10)$$

Међутим, интеграл који се јавља у бројоцу не знамо да израчунамо аналитички. Зато ћемо зарубљену купу апроксимирати нормалном расподелом која нам то омогућава. Надаље претпостављамо да су тачке купе нормално распоређене, а купу мењамо елипсоидом. Претпоставићемо још да је таква нормална расподела са дијагоналном матрицом коваријансе, пре свега због једноставности даљег рачуна. Имајући у виду да је вектор погледа нормалан на базу сваке купе, као и да су базе кругови, то је зарубљена купа симетрична око вектора погледа, а самим тим је то и елипсоид. То за последицу има једнакост двеју варијанси у равни нормалној на вектор погледа. Таква нормална расподела је одређена са вредности –  $\mathbf{o}$ ,  $\mathbf{d}$ ,  $\mu_t$  (очекивано разстојање дуж зрака),  $\sigma_t^2$  (варијанса дуж зрака) и  $\sigma_r^2$  (варијанса дуж нормале зрака). Нама преостаје да одредимо очекивање и варијансу индикатора  $F(\mathbf{x}, \cdot)$ .

У наставку ћемо извести засад непознате вредности параметара  $\mu_t$ ,  $\sigma_t$  и  $\sigma_r$ . Посматрајмо зарубљену купу чија је оса симетрије нормална, односно, паралелна са свим осама. Употребом сферних координата добијамо параметризацију

$$(x, y, z) = \phi(r, t, \theta) = (rt \cos \theta, rt \sin \theta, t), \quad (4.11)$$

за  $\theta \in [0, 2\pi)$ ,  $t \geq 0$ ,  $|r| \leq R$ . Јакобијан је  $rt^2 dr dt d\theta$ .

Запремина зарубљене купе је

$$V = \int_0^{2\pi} \int_{t_0}^{t_1} \int_0^R rt^2 dr dt d\theta = \pi R \frac{t_1^3 - t_0^3}{3}. \quad (4.12)$$

Расподела у којој се тачке узоркују равномерно из зарубљене купе има густину  $rt^2/V$ , у шта се можемо уверити интеграцијом Јакобијана.

Први момент по  $t$  је

$$\begin{aligned} \mathbb{E}(t) &= \frac{1}{V} \int_0^{2\pi} \int_{t_0}^{t_1} \int_0^R t \cdot rt^2 dr dt d\theta \\ &= \frac{1}{V} \pi R^2 \frac{t_1^4 - t_0^4}{4} \\ &= \frac{3(t_1^4 - t_0^4)}{4(t_1^3 - t_0^3)}. \end{aligned} \quad (4.13)$$

Први момент по  $x$  (као и по  $y$  због симетрије је)

$$\mathbb{E}(x) = \frac{1}{V} \int_0^{2\pi} \int_{t_0}^{t_1} \int_0^R rt \cos \theta \cdot rt^2 dr dt d\theta = 0. \quad (4.14)$$

Други момент по  $t$  је

$$\begin{aligned}\mathbb{E}(t^2) &= \frac{1}{V} \int_0^{2\pi} \int_{t_0}^{t_1} \int_0^R t^2 \cdot rt^2 dr dt d\theta \\ &= \frac{1}{V} \pi R^3 \frac{t_1^5 - t_0^5}{5} \\ &= \frac{3(t_1^5 - t_0^5)}{5(t_1^3 - t_0^3)}.\end{aligned}\tag{4.15}$$

Други момент по  $x$  (као и по  $y$  због симетрије) је

$$\begin{aligned}\mathbb{E}(x^2) &= \frac{1}{V} \int_0^{2\pi} \int_{t_0}^{t_1} \int_0^R (rt \cos \theta)^2 \cdot rt^2 dr dt d\theta \\ &= \frac{1}{V} \frac{R^4}{4} \frac{t_1^5 - t_0^5}{5} \pi \\ &= \frac{3R^2(t_1^5 - t_0^5)}{20(t_1^3 - t_0^3)}.\end{aligned}\tag{4.16}$$

Када смо израчунали све ове моменте, коначно можемо одредити параметре конусне расподеле који су нам непознати. Очекивање дуж зрака  $\mu_t$  је први момент по  $t$ . Варијансу дуж зрака  $\sigma_t^2$  добијамо као  $\mathbb{D}(t) = \mathbb{E}(t^2) - \mathbb{E}(t)^2$ . И коначно,  $\sigma_r^2 = \mathbb{E}(x^2)$ .

У претходним изразима се појављују разлике виших степена, што врло често проузрокује нумеричке потешкоће када су  $t_0$  и  $t_1$  близки. Да то заиста јесте од значаја потврђује чињеница да је апроксимација коју спроводимо прецизна једино када је зарубљена купа ниска, односно, када је елипсоид нормалне расподеле пљоснат. Зато се уводе смене  $t_\mu = (t_0 + t_1)/2$  и  $t_\delta = (t_1 - t_0)/2$ . Коначно, добијамо

$$\begin{aligned}\mu_t &= t_\mu + \frac{2t_\mu t_\delta^2}{3t_\mu^2 + t_\delta^2} \\ \sigma_t^2 &= \frac{t_\delta^2}{3} - \frac{4t_\delta^4(12t_\mu^2 - t_\delta^2)}{15(3t_\mu^2 + t_\delta^2)^2} \\ \sigma_r^2 &= R^2 \left( \frac{t_\mu^2}{4} + \frac{5t_\delta^2}{12} - \frac{4t_\delta^4}{15(3t_\mu^2 + t_\delta^2)} \right).\end{aligned}\tag{4.17}$$

На овај начин смо потпуно одредили нормалну расподелу, али у координатном систему равни слике. Међутим, није тешко доћи до трансформације у светски координатни систем

$$\begin{aligned}\mu &= \mathbf{o} + \mu_t \mathbf{d} \\ \Sigma &= \sigma_t^2 \left( \frac{\mathbf{d} \mathbf{d}^T}{\|\mathbf{d}\|_2^2} \right) + \sigma_r^2 \left( \mathbf{I} - \frac{\mathbf{d} \mathbf{d}^T}{\|\mathbf{d}\|_2^2} \right).\end{aligned}\tag{4.18}$$

У досадашем разматрању смо припремили терен за дефинисање *интегралне фуријеизације* која се прилагођава новоуведеном конусном емитовању зрака. Подсећамо се идеје још једном – израчунати очекивану фуријеизацију на елипсоиду

нормалне расподеле. Дакле, потребно је пресликати улаз у простор више димензије, а потом у том простору израчунати очекивање. Да би се та два корака усклали, другачије ћемо записати 4.2

$$P = \begin{bmatrix} 1 & 0 & 0 & 2 & 0 & 0 & \dots & 2^{L-1} & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 & \dots & 0 & 2^{L-1} & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 & \dots & 0 & 0 & 2^{L-1} \end{bmatrix}^T \text{ и } \gamma(\mathbf{x}) = \begin{bmatrix} \sin(P\mathbf{x}) \\ \cos(P\mathbf{x}) \end{bmatrix}. \quad (4.19)$$

Параметри расподеле се мењају у складу са новим простором, тако да добијамо  $\boldsymbol{\mu}_\gamma = P\boldsymbol{\mu}$ . Облик матрице коваријансе следи из њене линеарности. Наиме, како је  $\text{Cov}(A\mathbf{x}, B\mathbf{y}) = A \text{Cov}(\mathbf{x}, \mathbf{y}) B^T$ , то је  $\Sigma_\gamma = P\Sigma P^T$ .

Преостаје да се израчуна очекивање у складу са вишедимензионом расподелом. Потребно је и довољно је показати понашање очекивања  $\gamma(x)$ . Прво ћемо показати очекивање у случају једнодимензионе нормалне расподеле, а потом и у случају који је нама потребан.

$$\begin{aligned} \mathbb{E}_{x \sim \mathcal{N}(\mu, \sigma^2)}(\sin x) &= \mathbb{E}(\text{Im} \exp(ix)) \\ &= \text{Im} \mathbb{E}(\exp(ix)) \\ &= \text{Im}(\phi_{x \sim \mathcal{N}(\mu, \sigma^2)}(1)) \\ &= \text{Im} \left( \exp \left( i\mu - \frac{\sigma^2}{2} \right) \right) \\ &= \sin(x) \exp \left( -\frac{\sigma^2}{2} \right), \end{aligned} \quad (4.20)$$

где је функција  $\phi_{x \sim \mathcal{N}(\mu, \sigma^2)}(t)$  је карактеристична функција нормалне расподеле.

Аналогно, добијамо  $\mathbb{E}_{x \sim \mathcal{N}(\mu, \sigma^2)}(\cos x) = \cos(x) \exp(-\sigma^2/2)$ . Уколико уопштимо претходно израчунавање, добијамо

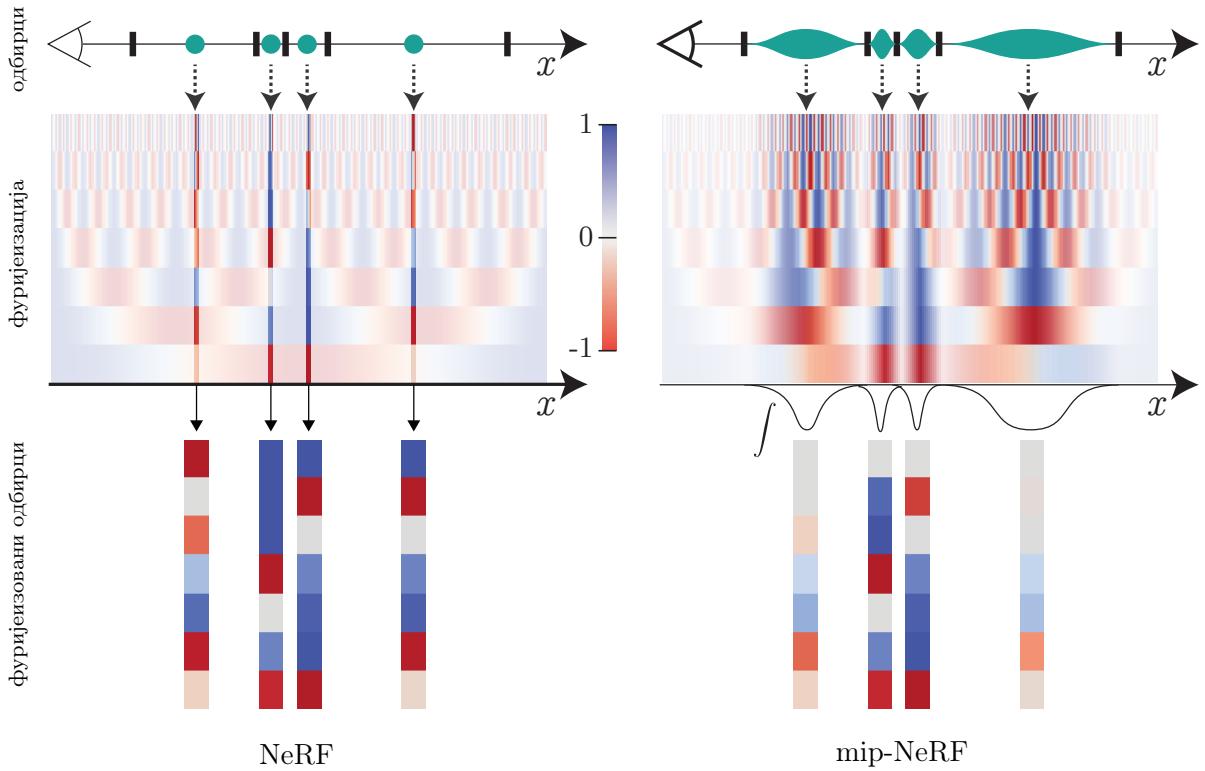
$$\begin{aligned} \gamma(\boldsymbol{\mu}, \Sigma) &= \mathbb{E}_{x \sim \mathcal{N}(\mu, \sigma^2)}(\gamma(x)) \\ &= \begin{bmatrix} \sin(\mu_\gamma) \odot \exp(-(1/2) \text{diag}(\Sigma_\gamma)) \\ \cos(\mu_\gamma) \odot \exp(-(1/2) \text{diag}(\Sigma_\gamma)) \end{bmatrix}, \end{aligned} \quad (4.21)$$

где је са  $\odot$  означено Адамарово, множење члан-по-члан.

Начин на који се изводи множење у запису 4.21 подразумева употребу само дијагоналних елемената матрице  $\Sigma_\gamma$ . Израчунавање пуне матрице изискује пуно рачунских операција и места, за чим уопште нема потребе. С тим у вези, дијагоналне чланове ћемо израчунати директно  $\text{diag}(\Sigma_\gamma) = [\text{diag}(\Sigma), 4\text{diag}(\Sigma), \dots, 4^{L-1}\text{diag}(\Sigma)]^T$  при чему је

$$\text{diag}(\Sigma) = \sigma_t^2 \left( \frac{\mathbf{d} \odot \mathbf{d}}{\|\mathbf{d}\|_2^2} \right) + \sigma_r^2 \left( I - \frac{\mathbf{d} \odot \mathbf{d}}{\|\mathbf{d}\|_2^2} \right). \quad (4.22)$$

На слици 4.3 је приказана разлика у фуријеизацији у случају NeRF и mip-NeRF модела. Како се из сваке зарубљене купе узоркује више од једног одбирка можемо посматрати расподелу одбира. Како смо увели претпоставку о нормалној расподели, односно апроксимацији зарубљене купе елипсоидом, добијамо *вретенаст* облик густине расподеле дуж једног зрака. Приметимо да тај облик није случајан и да се може видети као пресек елипсоида и равни која садржи осу симетрије.



Слика 4.3: Приказ интегралне фуријеизације према [2]

#### 4.2.2 Параметризација и обучавање

Иако mip-NeRF зраке емитује по конусу, обучавање се практично изводи на исти начин као и у случају NeRF-а. Уместо узорковања  $n$  одбирача дуж зрака, узоркује их се  $n + 1$  јер ће се узастопно у паровима користити за границе зарубљене купе. Након примене фуријеизације, прослеђују се MLP-у и добијају парови  $(\sigma, c)$ . MLP је по структури идентичан оном који се користи за параметризацију NeRF-а.

Међутим, mip-NeRF не захтева употребу финог и грубог модела, већ се цео поступак спроводи употребом само једног. Одговор лежи у употреби конусног емитовања зрака и интегралне фуријеизације.

Са  $\mathbf{k}_{\mathbf{o}, \mathbf{d}, R}$  ћемо означити зарубљену купу са врхом у  $\mathbf{o}$ , осе симетрије  $\mathbf{d}$  и полу-пречника пресека купе и равни слике са  $R$ . Поменути пресек означавамо са  $\Pi_{\mathbf{k}_{\mathbf{o}, \mathbf{d}, R}}$ . Ознаке  $f$  и  $c$  имају исто значење као и у 4.8. Функција губитка која се користи за обучавање mip-NeRF-а је

$$\mathcal{L} = \sum_{\mathbf{f}_{\mathbf{c}, \mathbf{d}, R} \in \mathcal{B}} \left( \left\| \hat{C} \left( \Pi_{\mathbf{k}_{\mathbf{c}, \mathbf{d}, R}^f} \right) - C \left( \Pi_{\mathbf{k}_{\mathbf{c}, \mathbf{d}, R}^f} \right) \right\|_2^2 + \lambda \left\| \hat{C} \left( \Pi_{\mathbf{k}_{\mathbf{c}, \mathbf{d}, R}^c} \right) - C \left( \Pi_{\mathbf{k}_{\mathbf{c}, \mathbf{d}, R}^c} \right) \right\|_2^2 \right). \quad (4.23)$$

Параметар  $\lambda$  контролише утицај грубе поделе на функцију губитка јер се само један модел користи за обе врсте поделе.

Врши се укупно по 128 узорковања за грубу и фину поделу.

### 4.3 Ref-NeRF

Иако претходне имплементације дају веома добре резултате, за одређене сцене можемо видети њихова ограничења. Посматрањем резултата на сликама и 6.2 и 6.3 можемо уочити проблем са учењем *сјајности*. Где год постоји изразита спекуларна светлост, NeRF и mip-NeRF нису у стању да је доносе верно, већ се стиче утисак дифузне рефлексије – другим речима, површина делује *обрушено*.

Проблем осветљености сјајних површина је основни мотив за увођење Ref-NeRF архитектуре. Начин на који се Ref-NeRF бори са тим проблемом је илустрован на слици 4.4, а у наставку ће та илустрација бити објашњена.

Слику можемо поделити на две колоне – на левој се налази објекат глатке површине, док се на десној налази објекат од различитих материјала. Оба објекта се налазе на сцени са четири различита извора светлости, а простор је димензије 2. С тим у вези, вектор погледа једнозначно одређујемо само једним углом  $\phi$ , док угао  $\theta$  у овом случају не постоји.

Објекат можемо видети као просту затворену криву. Параметризација ове криве нам омогућава једноставно кретање по њој, тачку по тачку. Уочавамо произвольну тачку  $\mathbf{x}$ , а у њој нормалу  $\mathbf{n}$ . Анализираћемо како се понаша осветљеност тачке на крivoј у зависности од промене тачке  $\mathbf{x}$  и вектора погледа  $\mathbf{d}$ . Претпоставимо, за почетак, да се уочена тачка помера по крivoј у смеру приказаном на слици. Уколико задржимо исти вектор погледа, на тачку  $\mathbf{x}$  ће све већи утицај да има горњи-леви извор светлости, а све мање горњи-десни. Све док у једном тренутку тачка више не буде видљива под фиксираним углом гледања, што можемо доживети као црну боју. Аналогно се дешава при фиксирању тачке  $\mathbf{x}$ , а променом вектора погледа  $\mathbf{d}$ . Дакле, у оба случаја долази до нетривијалне промене боје тачке.

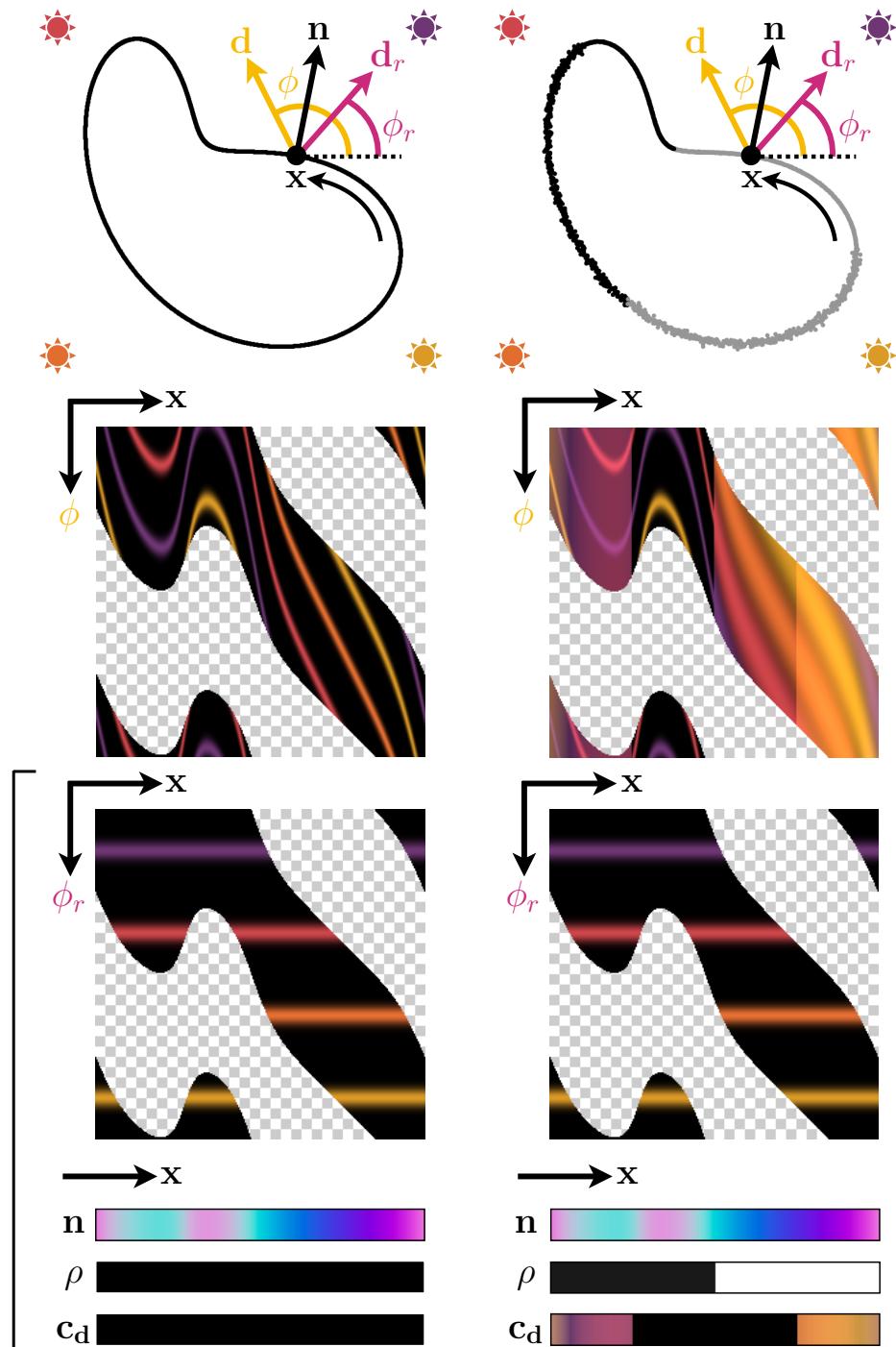
Модел који за циљ има рендеровање овакве сцене као скуп обучавања добија неки подскуп параметризације приказане у другом реду. У питању су неки од парова  $(\mathbf{x}, \phi)$  са придруженим бојама. Од модела се очекује да на основу тога научи интерполацију којом би попунио остатак параметризације. Како су промене боја у општем случају компликоване, што се посебно може видети на примеру другог објекта, таква интерполација не може бити нарочито прецизна. Зато је потребно уочити неку другу величину за коју се боја тачке мења једноставније.

Уколико се тачка коју посматрамо промени, а желимо да боја коју видимо остане иста, морамо променити вектор погледа. Променом тачке, мењамо и нормалу криве. Према томе, вектор погледа не можемо променити било како, већ за тачно онај угао за који се променила нормала. Управо то релфексију вектора погледа око нормале чини константном! Зато ћемо уместо вектора погледа користити његов вектор рефлексије. Тиме смо репараметризовали боју криве тако да се промена дешава *једноставније* што олакшава проблем обучавања модела.

Водећи рачуна о томе како је вектор  $\mathbf{d}$  усмерен, његову рефлексију око  $\mathbf{n}$  рачунамо

$$\mathbf{d}_r = \frac{\mathbf{d}}{\|\mathbf{d}\|} - 2 \left( \frac{\mathbf{d}}{\|\mathbf{d}\|} \cdot \mathbf{n} \right) \mathbf{n}. \quad (4.24)$$

О начину добијања поменуте нормале биће више речи касније.



Слика 4.4: Илустративни пример значаја рефлексије вектора погледа око нормале. Необојена (шаховска) поља представљају вектора погледа који немају пресек са објектом. Величине са ознаком  $r$  у индексу представљају одговарајуће рефлектоване еквиваленте.

### 4.3.1 Сферна фуријеизација

Поново се, као и у случају mip-NeRF архитектуре, јавља другачија врста фуријеизације, с тим што је овог пута она статистички заснована. У наставку ћемо објаснити и зашто.

Одаберимо неку тачку на сфери произвољно. Уколико је осветљење сфере у околини те тачке дифузно, тада је густина одбијених зрака равномернија јер не долази до нагле разлике у осветљењу. С друге стране, ако је у питању спекуларна рефлексија, тада је расподела одбијених зрака готово Диракова – приметно је нагомилавање густине у једној тачки.

Сличан мисаони експеримент смо могли спровести и за неки други облик који није сфера. Међутим, такви облици у општем случају нису глатки и могу имати оштре ивице, пресеке или рупе. Такве особине доприносе сложенијем простирању светlostи, па на осветљење у одређеној тачки утицај могу имати и друге тачке. Другим речима, веома је тешко установити због чега и како је нека тачка осветљена. Зато је сфера узета за идеалан пример.

Отуда идеја да посматрамо расподелу која добро моделује другачију концентрацију густине у различитим тачкама на сferи. Зато бирамо *фон Мизес-Фишерову* расподелу. Даћемо формални опис ове расподеле у случају димензије 3.

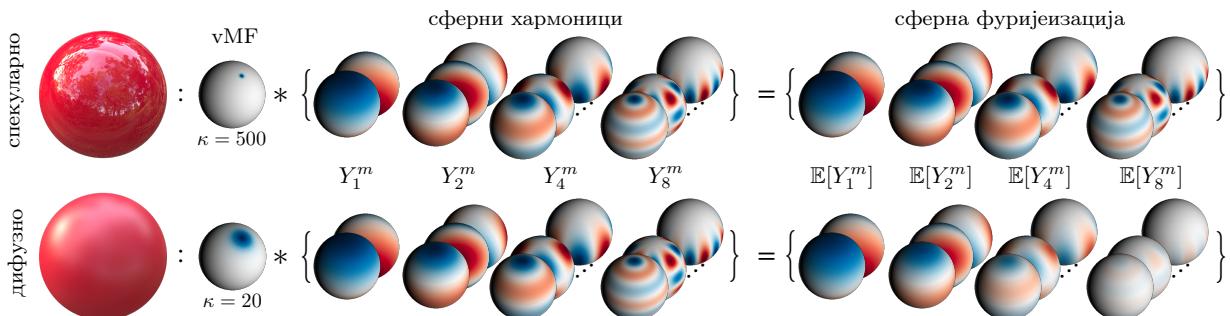
**Дефиниција 4.3.1.** Густина фон Мизес-Фишерове расподеле је дата са

$$f(\mathbf{x}) = \frac{\kappa}{4\pi \sinh \kappa} \exp(\kappa \boldsymbol{\mu}^T \mathbf{x}).$$

Параметри  $\boldsymbol{\mu}$  и  $\kappa$  се редом зову *очекивање* и *концентрација*.

У претходним изведбама смо за потребе пресликавања у простор више димензије користили Фуријеове коефицијенте, односно, *хармонике*, иако се, строго говорећи, јављају у мало другачијем облику него што је то случај у Фуријеовом развоју.

Како смо за потребе боље анализе осветљења објекта природно прешли на сферу, а у складу са тим и на одговарајућу расподелу, хармонике је потребно заменити сферним еквивалентом. Зато уводимо *сферне хармонике*, чији је приказ дат на слици 4.5.



Слика 4.5: Приказ сферних хармоника и сферне фуријеизације према [3]

**Дефиниција 4.3.2.** Сферни хармоници су решења Лапласове диференцијалне једначине  $\nabla^2 f = 0$  у сферним координатама, где је  $f : \mathbb{R}^3 \rightarrow \mathbb{C}$  два пута диференцијабилна функција. Сферни хармоник степена  $l$  и реда  $m$  је комплексна функција на јединичној сфери у ознаки  $Y_l^m$ .

Следећу теорему наводимо без доказа.

**Теорема 4.3.1.** Општи облик сферног хармоника  $Y_l^m(\theta, \phi)$  је  $N e^{im\phi} P_l^m(\cos \theta)$ , где је  $P_l^m$  одговарајући Лежандров полином, а  $N$  коефицијент нормализације.

Коефицијент нормализације можемо одредити из ортогоналности. Може се показати да из  $\int_{S^2} Y_l^m Y_{l'}^{m'} dS^2 = 0$ , следи

$$N = \sqrt{\frac{2l+1}{4\pi} \cdot \frac{(l-m)!}{(l+m)!}}.$$

Претходно уведене појмове ћемо обједнити *сферном фуријеизацијом* и то

$$\gamma(\mathbf{d}_r, \kappa) = \{\mathbb{E}_{\omega \sim vMF(\mathbf{d}_r, \kappa)}(Y_l^m(\boldsymbol{\omega})) : (l, m) \in \mathcal{M}_L\}, \quad (4.25)$$

где је  $\mathcal{M}_L = \{(l, m) : l = 1, \dots, 2^L \wedge m = 0, \dots, l\}$ .

Остаје питање израчунавања очекивања  $\mathbb{E}_{\omega \sim vMF(\mathbf{d}_r, \kappa)}(Y_l^m(\boldsymbol{\omega}))$ . Показује се [3] да је поменути израз приближно једнак

$$\exp\left(-\frac{l(l+1)}{2\kappa}\right) Y_l^m(\mathbf{d}_r). \quad (4.26)$$

Иако је изложени теоријски модел расподеле у потпуности оправдан, није јасно како долазимо до параметра концентрације. Приметимо да је у питању величина обрнуто пропорционална грубости материјала  $\rho$ . Дифузно осветљење је карактеристично за делове предмета који нису доволно сјајни, и обратно. Зато користимо идентитет  $\kappa = 1/\rho$ . За параметар  $\rho$  је задужен просторни MLP, о чему ће бити више речи у наставку.

### 4.3.2 Параметризација и обучавање

Неуронске мреже које чине параметризацију Ref-NeRF-а су у нешто другачијем облику него што смо до сада имали прилике да видимо. Непосредни излаз просторног MLP-а је сада обогаћен новим величинама. У питању су сви они параметри на које смо се до сада позивали, али и неки које до сад нисмо имали прилике да поменемо. Према томе, величине које просторни MLP треба да научи су:

- $\mathbf{c}_s, \mathbf{c}_d$  – спекуларна и дифузна боја светlostи,
- $\mathbf{s}$  – нијанса спекуларне светlostи,
- $\hat{\mathbf{n}}$  – нормала у датој тачки  $\mathbf{x}$ ,
- $\mathbf{b}$  – латентна репрезентација положаја  $\mathbf{x}$ .

Напоменимо још једном да баратамо са посебним величинама које се тичу дифузног и спекуларног осветљења, тј. боја има своју дифузну и спекуларну компоненту. Остаје да их објединимо у циљу добијања коначне боје. Како дифузно осветљење зависи само од положаја, користићемо

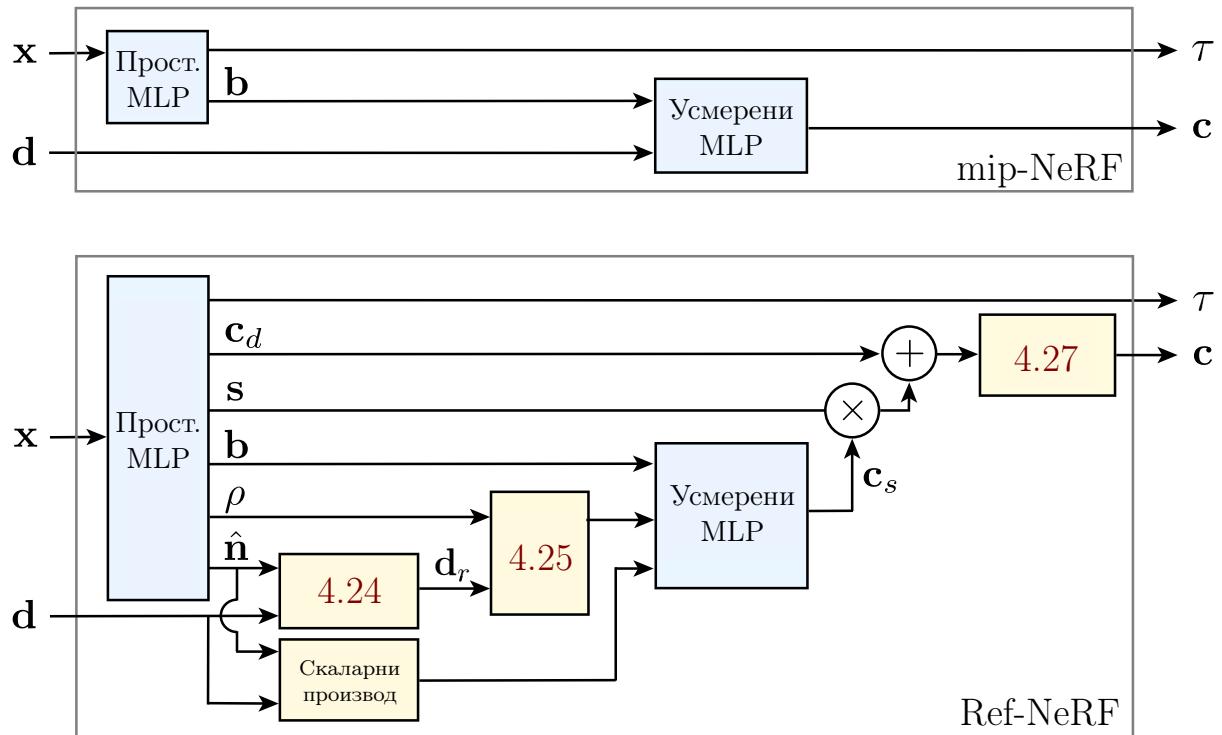
$$\mathbf{c} = \Gamma(\mathbf{c}_d + \mathbf{s} \odot \mathbf{c}_s), \quad (4.27)$$

где је  $\Gamma$  функција преноса у стандардизовани RGB систем [12].

У основи овог модела се налази mip-NeRF. За просторни MLP се користи идентична архитектура, али се за потребе усмереног MLP-а користи 8 слојева са по 256 неурона и ReLU функцијом активације. Приказ архитектуре и редоследа операција је дат на слици 4.6.

Обучавање се спроводи на исти начин као и у случају mip-NeRF-а, тако да то нећемо описивати поново.

Као основна разлика у параметризацији mip-NeRF и Ref-NeRF архитектура се јавља број параметара – Ref-NeRF их има скоро дупло више. У табели 6.1 је приказан њихов однос.



Слика 4.6: Поређење mip-NeRF и Ref-NeRF архитектура

### 4.3.3 Израчунавање вектора нормале

Као што смо рекли на самом почетку ове главе, израчунавање нормале је неопходно за обучавање. Поменули смо само начин на који се израчунава вектор рефлексије у односу на нормалу, али није било речи о томе како се долази до нормале. Управо то објашњавамо у наставку, а тиме и заокружујемо причу о новоуведеним величинама.

Вектор нормале у некој тачки се тривијално рачуна из усмереног MLP-а. Може се користити идентитет

$$\hat{\mathbf{n}}'(\mathbf{x}) = -\frac{\nabla \tau(\mathbf{x})}{\|\nabla \tau(\mathbf{x})\|}. \quad (4.28)$$

Међутим, овај метод је изузетно непрецизан јер се од самог почетка ослања на недовољно обучен усмерени модел. Отуда идеја да вектор нормале учи просторни MLP, али да поменути идентитет послужи за навођење резултата.

Спровешћемо поступак као у једначини 4.6, али ћемо, наравно, боју заменити нормалом. Дакле, за сваку тачку  $\mathbf{x}_i$  дуж зрака израчунавамо нормалу описаним поступком и тиме добијамо  $\hat{\mathbf{n}}'_i$ . Исти скуп тачака дајемо и просторном MLP-у и добијамо нормале  $\hat{\mathbf{n}}_i$ . Функцији губитка додајемо нови члан

$$\mathcal{R}_n = \sum_i w_i \|\hat{\mathbf{n}}'_i - \hat{\mathbf{n}}_i\|^2, \quad (4.29)$$

где су тежине  $w_i$  исте оне тежине као и у једначини 4.6.

Да би приморали нормале да буду окренуте ка камери, додајемо још један члан

$$\mathcal{R}_o = \sum_i w_i \max(0, \hat{\mathbf{n}}_i \cdot \mathbf{d})^2. \quad (4.30)$$

Приметимо да ће се кањжавати одбирачи који су видљиви ( $w_i \gg 0$ ), али и они код којих је нормала испод површи ( $\hat{\mathbf{n}}_i \cdot \mathbf{d} \gg 0$ ).



# Глава 5

## Скупови података

Скупови података у области неуронских поља зрачења се уграбо могу поделити на две групе – да ли је позната калибрација камере или не. У овом раду смо се определили за прву групу. Разлози су пре свега техничке природе. Проблем калибрације камере је један од централних проблема рачунарског вида и као такав је добро испитан. Може се рећи да је овај проблем *лоше условљен*. Готово незнатне разлике могу довести до значајних грешака у пројекцијама, а врло прецизна калибрација изискује и скупу опрему. Олакшавајућа околност је да су познати поступци одређивања унутрашњих и спољашњих параметара камере, али јасно је да они не могу бити ни близу прецизни као што је, на пример, роботска калибрација. С тим у вези, мишљења смо да је оправдано користити синтетички генерисане скупове података у којима ће камера бити постављена на унапред одређено место, а сви њени параметри ће бити познати по дефиницији.

За потребе овог рада, коришћени су LEGO, MATERIALS, DRUMS, CHAIR и SHIP. У питању су познати, референтни скупови података. Модели ће бити упоређени на сваком од њих у истом окружењу. Конкретно, у питању су редом багер од Лего коцки, 16 лоптица различитих материјала, бубњеви, столица и брод. На слици 5.1 су приказана по три погледа за сваки од скупова.

Сваки скуп података је подељен на три подскупа – за обучавање, проверу и тестирање. У подскуповима за обучавање и проверу се налази тачно 100 газличитих погледа, док се за тестирање користи 200. Рендери су синтетички, резолуције  $800 \times 800$  пиксела без радијалне и тангенцијалне дисторзије.



Слика 5.1: Одозго на доле, редом LEGO, CHAIR, MATERIALS, DRUMS и SHIP

# Глава 6

## Експерименти

Модели NeRF, mip-NeRF и Ref-NeRF ће бити упоређени по свакој од метрика које ћемо навести и објаснити у наставку. Такође, како је дужина трајања обучавања од великог значаја, посебну пажњу ћемо посветити и том аспекту.

Обучавање је вршено употребом сервиса AzureML са избором хиперпараметара у складу са изворним радовима, са само једним изузетком – параметар `batch_size` је мањи услед просторних ограничења. Графички процесор који је коришћен носи ознаку NVIDIA Tesla K80 са 24GB сопствене меморије. Имплементација је изведена у програмском језику Python користећи PyTorch [13].

### 6.1 Метрике

За анализирање квалитета рендерованих сцена биће коришћено више метрика како би се понашање модела сагледало из различитих углова.

**PSNR** (енг. *Peak Signal-to-Noise Ratio*). Нека је дата RGB слика  $I$  и њена апроксимација  $K$ . Средњеквадратно одступање ове две слике је

$$\text{MSE} = \frac{1}{3HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \|I_{i,j} - K_{i,j}\|_2^2. \quad (6.1)$$

Однос сигнала и шума је

$$\text{PSNR} = 10 \log_{10} \left( \frac{I_{\max}^2}{\text{MSE}} \right) [\text{dB}], \quad (6.2)$$

где је  $I_{\max}$  највећа могућа вредност коју пиксел на слици може имати. Како се канал слике обично представља једним бајтом, ова вредност у већини случајева износи 255.

Више вредности ове метрике указују на бољу апроксимацију.

**LPIPS** (енг. *Learned Perceptual Image Patch Similarity*) [14]. Идеја је имати метрику која опонаша људску процену сличности две слике. У ту сврху се користи модел  $\mathcal{F}$  обучен на сликовном скупу података. Не постоје никаква ограничења у погледу архитектуре, па чак ни скупа података, али се показује да архитектуре VGG [15] и AlexNet [16] у комбинацији са скупом слика ImageNet врло добро раде у пракси.

Означимо са  $I$  троканалну слику, а са  $K$  њену апроксимацију. Улазна слика пролази кроз модел  $\mathcal{F}$  слој по слој мењајући на том путу како ширину и висину, тако и број канала, па можемо рећи да слој  $l$  за излаз даје тензор из  $\mathbb{R}^{H_l} \times \mathbb{R}^{W_l} \times \mathbb{R}^{C_l}$ . Тада излаз је резултат примене функције активације и означићемо га са  $\alpha_l^I$  и  $\alpha_l^K$ , за сваку од слика редом. Додатно, у димензији канала спроводимо јединичну нормализацију. Метрика се дефинише као

$$\text{LPIPS} = \sum_{l=1}^L \frac{1}{H_l W_l} \sum_{i=0}^{H_l-1} \sum_{j=0}^{W_l-1} \|w_l \odot (\alpha_{l;i,j}^I - \alpha_{l;i,j}^K)\|^2, \quad (6.3)$$

где је  $w_l \in \mathbb{R}^{C_l}$  вектор тежина. Ако су сви  $w_l$  једнаки 1, метрика се своди на косинусну сличност, што за потребе овог рада и јесте урађено.

Ниже вредности ове метрике указују на бољу апроксимацију.

**SSIM** (енг. *Structual Similarity Index Measure*) [17]. Нека је дата монохроматска слика  $I$  и њена апроксимација  $K$ . Метрика се дефинише као

$$\text{SSIM} = \frac{(2\mu_I\mu_K + c_1)(2\sigma_{I,K} + c_2)}{(\mu_I^2 + \mu_K^2 + c_1)(\sigma_I^2 + \sigma_K^2 + c_2)}, \quad (6.4)$$

где су  $\mu_I$  и  $\mu_K$  узорачке средине,  $\sigma_I^2$  и  $\sigma_K^2$  узорачке дисперзије, а  $\sigma_{I,K}$  узорачка коваријанса између  $I$  и  $K$ . Вредности  $c_i$  су дефинисане као  $(k_i I_{\max})^2$ , где је су  $k_i$  константе и то обично редом 0.01 и 0.03.

Уколико је слика полихроматска, као што је случај са RGB сликама, коначна метрика се израчунава као просек метрике примене на сваки од канала слике.

Ова метрика се историјски користила за оцену квалитета дигиталног TV преноса. Дефинисана је са циљем да процени квалитет слике који корисници, односно гледаоци, опажају. Показало се да је у томе веома успешна упркос својој једноставности. Основна идеја је су пиксели корелисани када су блиски.

Више вредност ове метрике указују на бољу апроксимацију.

## 6.2 Време обучавања

Дужина трајања обучавања не зависи од скупа података, а сви модели су обучавани тачно одређен број итерација. Број итерација је узет као фиксан параметар за потребе поређења модела у истим условима. Наиме, модели су различити услед чега време потребно за извршење једне итерације обучавања није једнако. Ако би фиксирали циљну вредност метрике или време трајања обучавања, онај модел чија итерација обучавања траје најкраће би тривијално био у предности.

модел	време обучавања (у сатима)	број итерација	број параметара
NeRF	120	1000000	1200000
mip-NeRF	127	1000000	612000
Ref-NeRF	135	1000000	1100000

Табела 6.1: Време обучавања за сваки модел

Примећујемо да сви модели изискују прилично дugo време обучавања и то око пет дана. Када је реч о моделима појединачно, видимо да се mip-NeRF спорије обучава од NeRF-а иако има двоструко мање параметара. Разлог је у значајно компликованијем узорковању одбирача. Такође, Ref-NeRF захтева више времена од NeRF-а за обучавање због већег броја операција – израчунавање нормале, израчунавање рефлексије итд.

Управо зато што је потребан већи број сати за обучавање свих модела, не можемо ни један модел прогласити најбољим. На овако дugo обучавање, разлика од десет или петнаест сати не игра кључну улогу. Мада је, објективно говорећи, модел NeRF ипак по овом питању бољи од осталих.

## 6.3 Резултати

У наставку су дате табеле са метрикама за сваки од модела на сваком од скупова података.

модел	PSNR $\uparrow$	LPIPS $\downarrow$	SSIM $\uparrow$
NeRF	34.38	0.032	0.973
mip-NeRF	<b>35.80</b>	0.026	0.978
Ref-NeRF	35.79	<b>0.024</b>	<b>0.979</b>

Табела 6.2: Резултати за LEGO скуп података

модел	PSNR $\uparrow$	LPIPS $\downarrow$	SSIM $\uparrow$
NeRF	30.45	0.055	0.956
mip-NeRF	30.58	0.052	0.958
Ref-NeRF	<b>35.71</b>	<b>0.028</b>	<b>0.985</b>

Табела 6.3: Резултати за MATERIALS скуп података

модел	PSNR $\uparrow$	LPIPS $\downarrow$	SSIM $\uparrow$
NeRF	25.28	0.080	0.929
mip-NeRF	<b>25.52</b>	0.079	0.932
Ref-NeRF	<b>25.52</b>	<b>0.073</b>	<b>0.934</b>

Табела 6.4: Резултати за DRUMS скуп података

модел	PSNR $\uparrow$	LPIPS $\downarrow$	SSIM $\uparrow$
NeRF	34.93	0.029	0.979
mip-NeRF	35.20	0.028	0.981
Ref-NeRF	<b>35.84</b>	<b>0.022</b>	<b>0.984</b>

Табела 6.5: Резултати за CHAIR скуп података

модел	PSNR $\uparrow$	LPIPS $\downarrow$	SSIM $\uparrow$
NeRF	29.94	0.161	0.878
mip-NeRF	<b>30.53</b>	<b>0.154</b>	<b>0.884</b>
Ref-NeRF	29.50	0.164	0.871

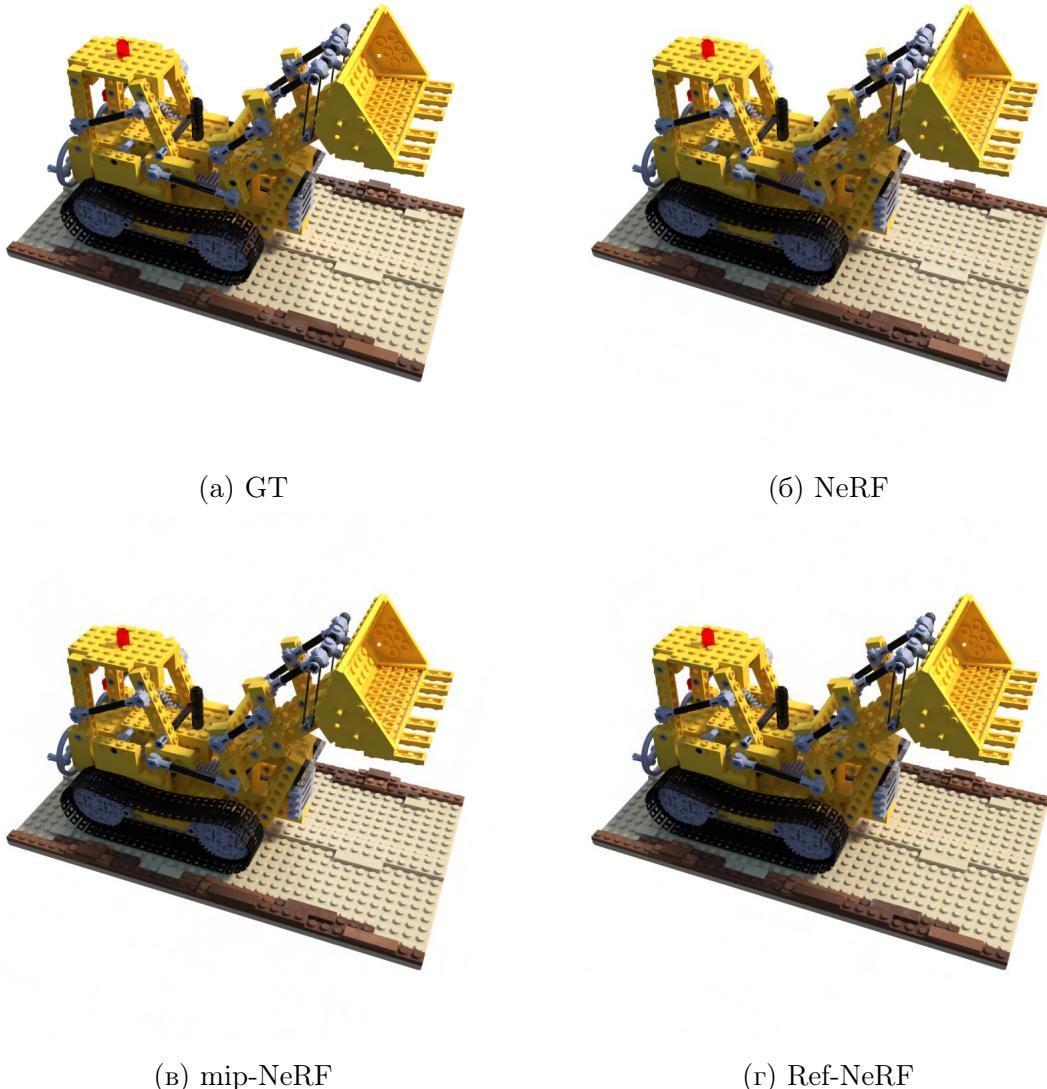
Табела 6.6: Резултати за SHIP скуп података

Можемо рећи да је у поређењу Ref-NeRF однео победу и да се намеће као први избор. Иако је за обучавање овог модела потребно највише времена, у питању је разлика од највише 10%. Побољшања су више него приметна без готово икаквих погоршања. Преостала два модела нису *значајно* лошија, тако да и њихово коришћење може дати задовољавајуће резултате.

Природно је питати се зашто је резултат на SHIP скупу података такав какав јесте. Наиме, модели су по метрикама готово па једнаки, а ово је једини скуп података на ком Ref-NeRF није бољи од других модела. Као главни разлог истичемо водену површину, која заузима највећи део слике. Касније ћемо рећи нешто више о томе, када будемо дискутовали резултате на сваком од скупова података.

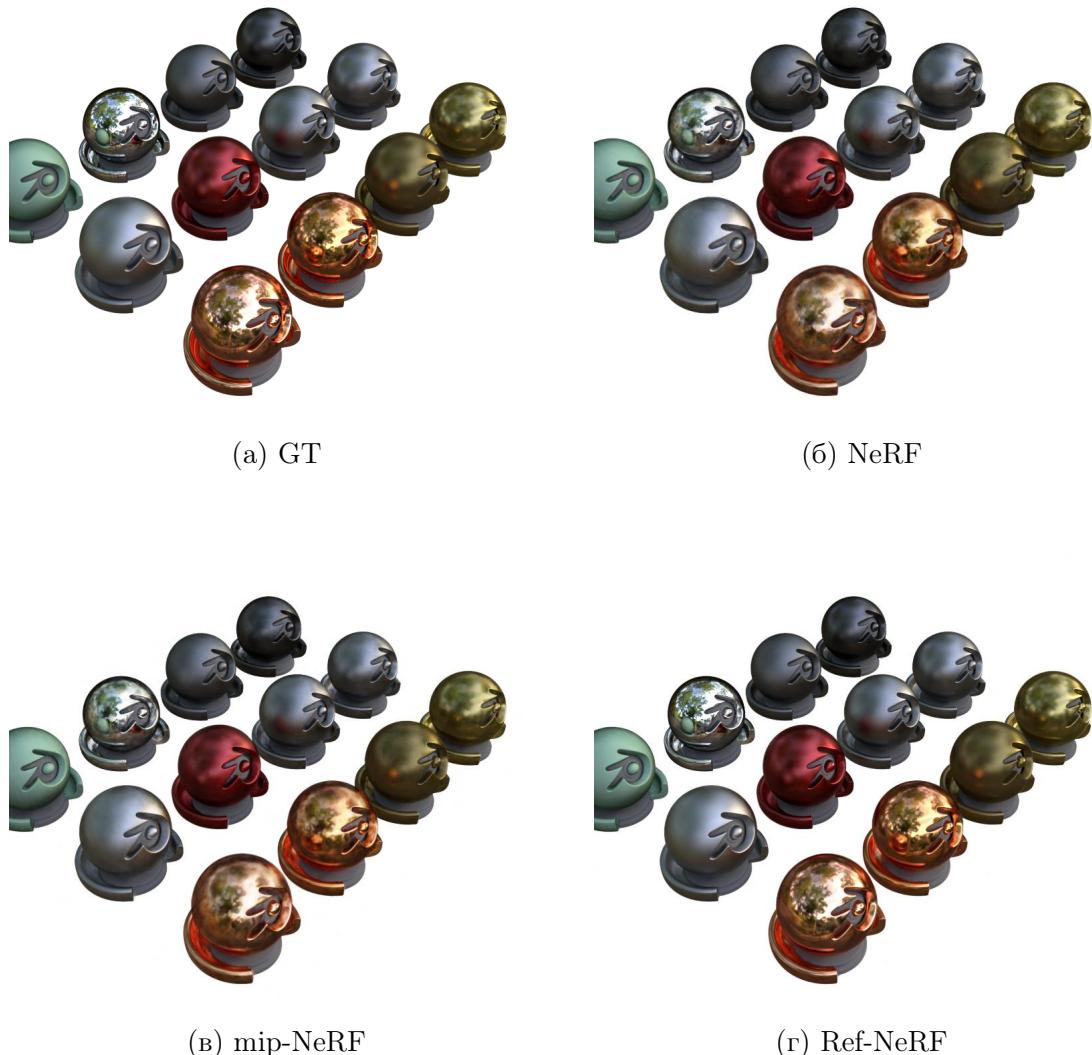
## 6.4 Приказ рендера

Ова глава служи за приказ рендера обучених модела и то у односу на рендер који представља *тачно решење*. Приказан је по један насумично одабран поглед за сваки од скупова података. Резултате ћемо упоредити искључиво визуелно, а касније ћемо дати објективно метричко поређење као и поређење свака два модела.



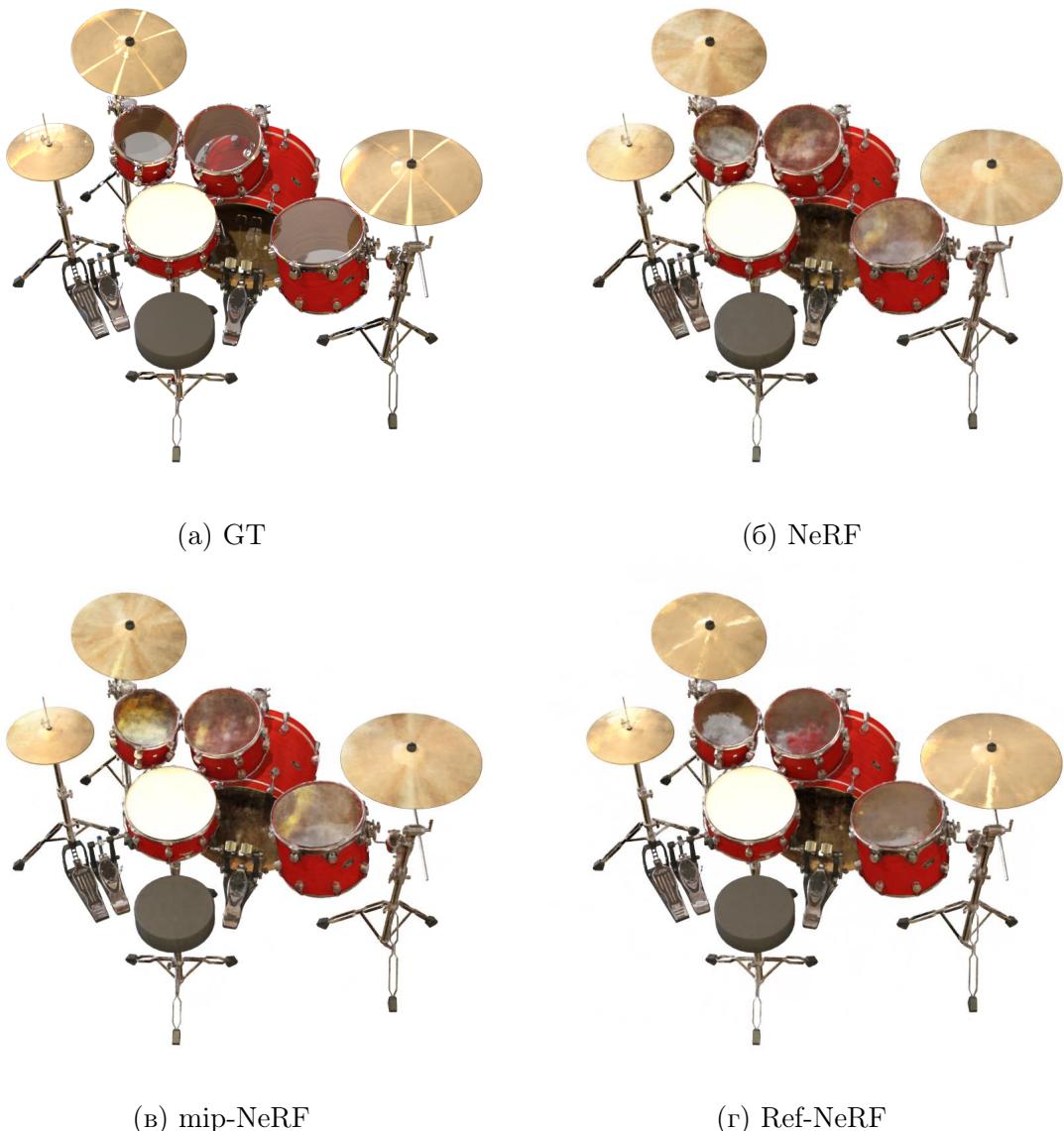
Слика 6.1: Приказ резултата LEGO скупа података

На слици 6.1 можемо видети да су сви модели у стању да дају врло квалитетан рендер на LEGO скупу података. Не постоје очиглени проблеми ни код једног од њих. Примећујемо детаље ниског нивоа – веродостојне сенке, ваљкаста испупчења на свакој коцкици, тачне боје и нијансе. Лего коцкице су иначе матирани, односно, веома је тешко видети спекуларно осветљење, што овај скуп података чини лакшим. Можемо рећи да се на овом скупу података ни један модел не истиче као значајно бољи од других.



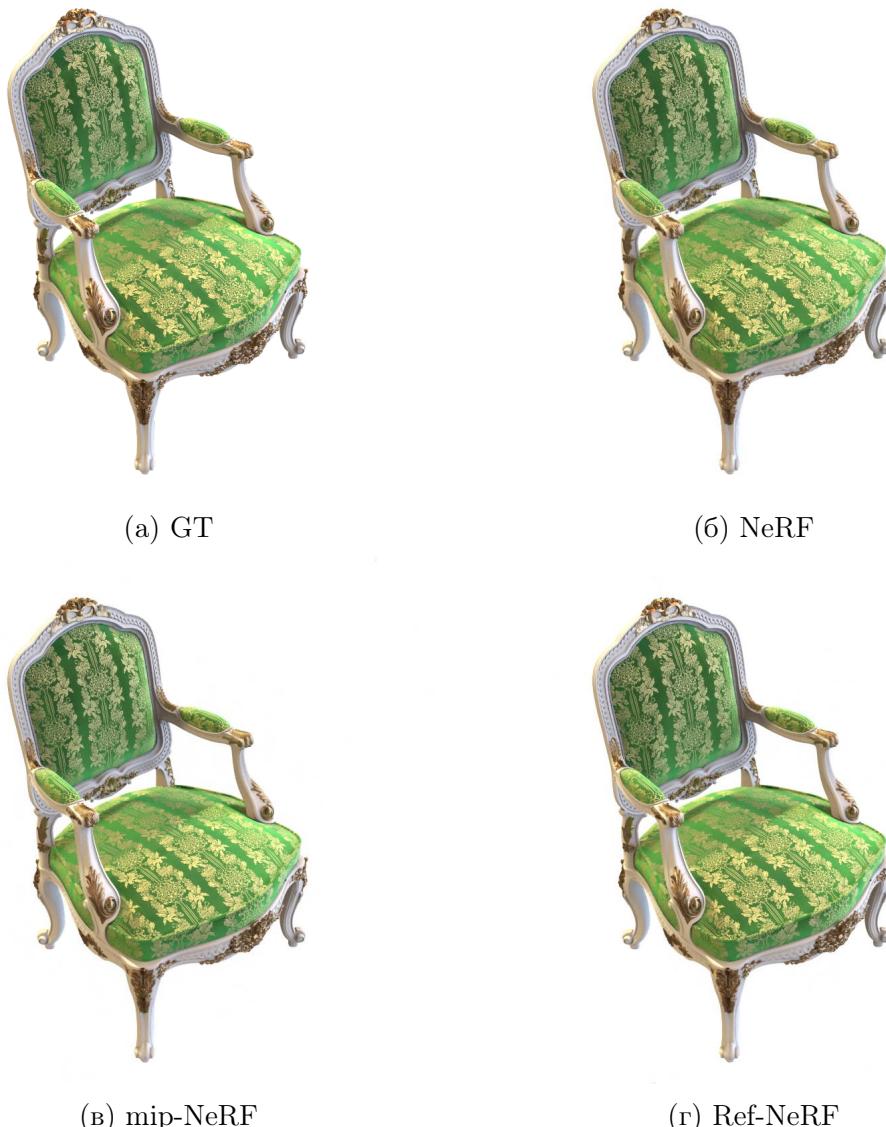
Слика 6.2: Приказ резултата MATERIALS скупа података

На слици 6.2 можемо видети квалитет рендера на MATERIALS скупу података. Овај скуп података је занимљив због присуства како матираних, тако и сјајних површина. Можемо приметити да су NeRF и mip-NeRF рендери недовољно квалитетни по питању сјајних површина и да је приметно матирање и онда када то заиста није случај. Разлике између два поменута модела нису очигледне. Међутим, Ref-NeRF је бољи управо у свим оним детаљима који се могу издвојити као мане остала два модела. То се најпре види на деловима сцене који су сјајни, попут фигурице која на слици делује најкрупније.



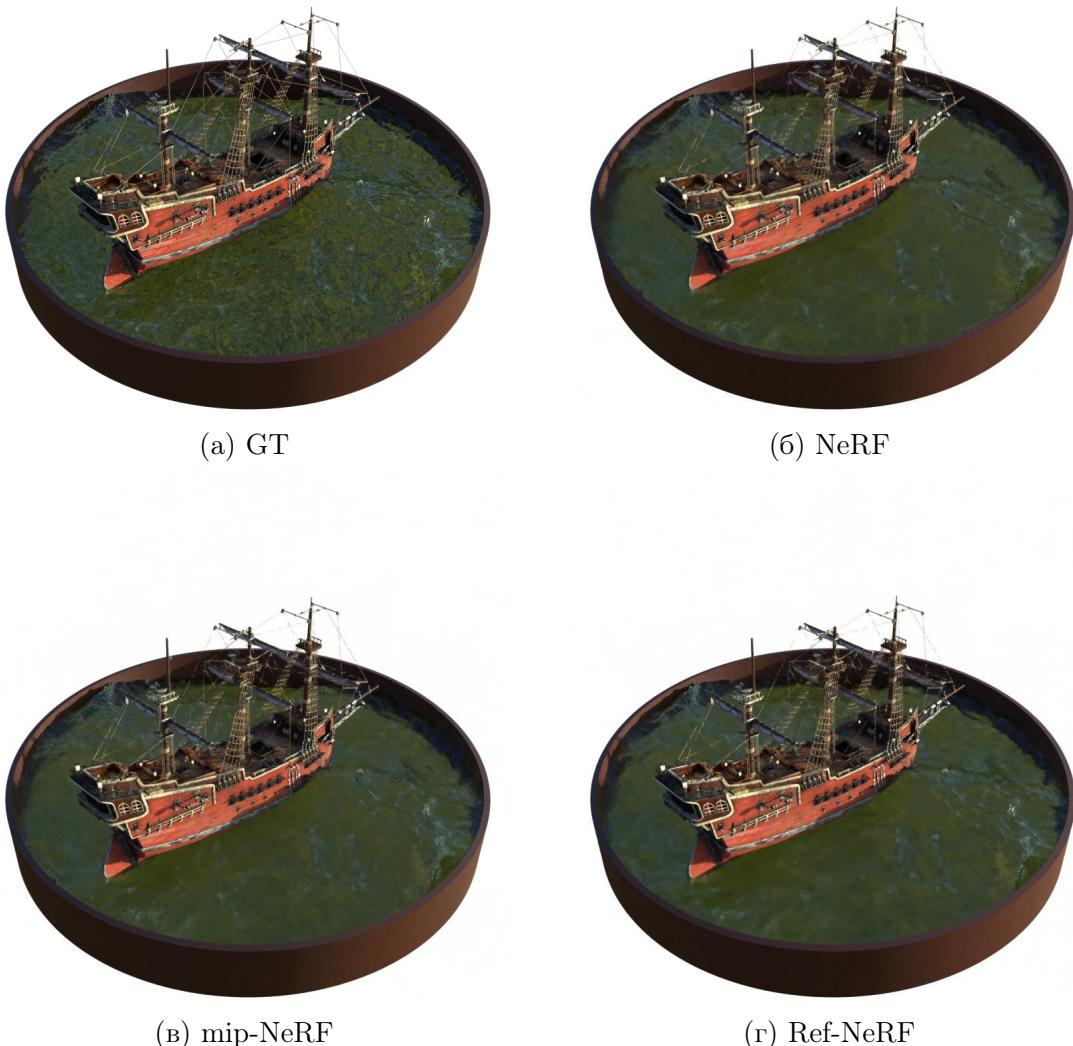
Слика 6.3: Приказ резултата DRUMS скупа података

На слици 6.3 можемо видети квалитет рендера на DRUMS скупу података. Овај скуп података има сјајне и прозирне елементе што га чини тежим за учење. То се поготово примећује код NeRF и mip-NeRF модела који нису у стању да рендерују ове елементе како треба. Чинеле готово да немају рефлексију на себи, док добоши немају прозирну површину. С друге стране, Ref-NeRF је по овом питању далеко бољи од осталих модела и приметан је квалитет рефлексије. За остале делове сцене можемо рећи да су квалитетно рендеровани од стране свих модела.



Слика 6.4: Приказ резултата CHAIR скупа података

На слици 6.4 можемо видети квалитет рендера на скупу CHAIR. У питању је лакши скуп података, будући да нема компликованих сенчења, различитих врста боја као ни сјајних површина. Само на основу приказаних рендера тешко је донети одлуку о најбољем моделу будући да су сви рендери веома слични једни другима. Као заиста крајње ситан детаљ можемо приметити боју врха столице – метални део има наранџасте делове који се искључиво примећују код Ref-NeRF модела. Слично можемо рећи и за спорадичне одсјаје металних делова на ручкама. У питању су заиста суптилне разлике.



Слика 6.5: Приказ резултата SHIP скупа података

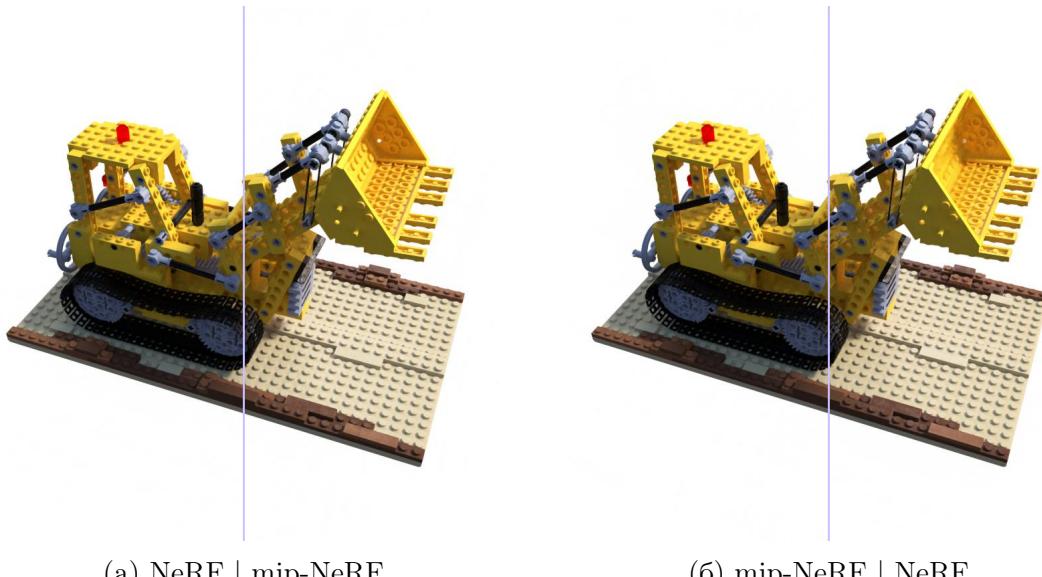
На слици 6.5 можемо видети квалитет рендера на скупу података SHIP. У питању је тежак пример скупа података будући да се на овој сцени налази водена површина. Веродостојно рендеровање воде је веома тежак проблем. Вода је провидна и у контакту са светlostи до упијања, расипања и рефлексије. Такође, вода скоро да никад није статична, а честе се појаве таласи и вирови. На основу добијених резултата можемо видети да су поменута ограничења и допринела недовољно добром рендеровању водене површине. Сви модели су водену површину начинили више стакленом него воденом са видно мањом количином детаља. Приметно је и замућење дрвених делова брода. Није могуће са сигурношћу рећи да се труп брод састоји од појединачних дасака.

## 6.5 Детаљно поређење

Пре него што се упуштимо у поређење модела, треба прокоментарисати однос метрика. Можемо видети да су метрике у међусобном складу, односно, сагласне су једна са другом. Једини изузетак су резултати на LEGO скупу података, али је у питању готово занемарљива разлика.

Само гледањем резултата у претходној глави се поставља значајно питање – постоје ли очите разлике у квалитету? На основу метрика, можемо видети да разлике у резултатима нису крупне, али да постоје оне значајније попут оних на скупу података MATERIALS. Зато ћемо се детаљнији позабавити анализом резултата.

**NeRF и mip-NeRF.** Метрике указују да разлика између ова два модела готово да не постоји. Упоређујући рендероване сцене можемо приметити извесна побољшања у случају mip-NeRF модела, али их је у општем случају врло тешко разликовати. На слици 6.6 можемо видети колико су ова два модела слична по квалитету.



Слика 6.6: Поређење NeRF и mip-NeRF модела на DRUMS скупу података

Време обучавања ових модела је, слободно можемо рећи, идентично. Међутим, како је број параметара mip-NeRF-а упона мањи, одлучујемо се управо за тај модел.

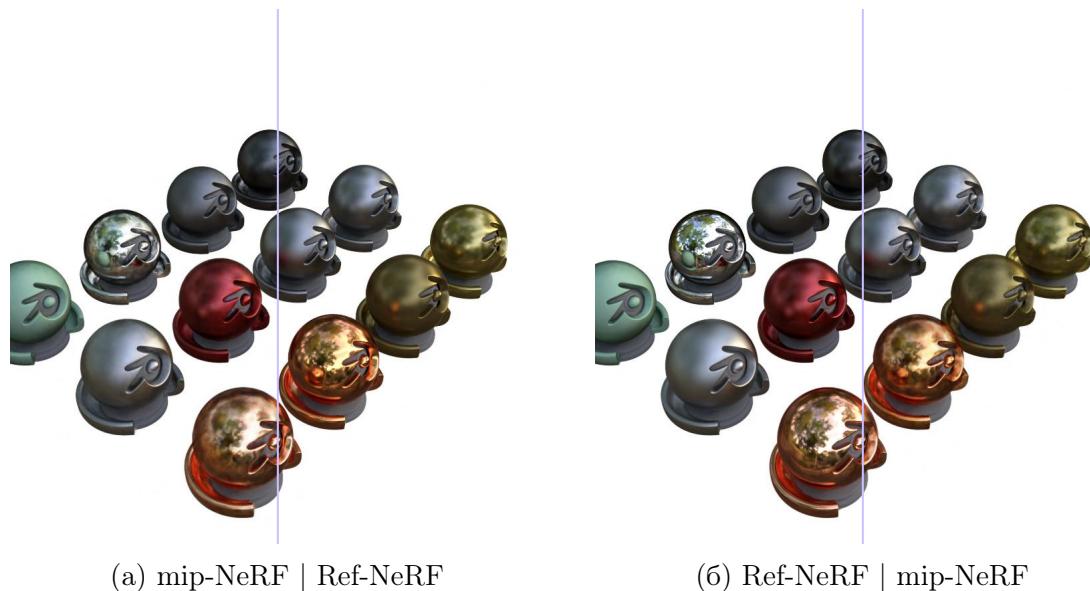
Како је mip-NeRF *бољи*, наредна поређења неће узимати у обзир NeRF када је реч о квалитету резултата.

**Ref-NeRF и mip-NeRF.** Ref-NeRF је углавном, у најгорем случају, на нивоу супарника. Видимо да се посебно истиче у MATERIALS скупу података, што и не чуди будући да је то пример скупа података са великим укупном сјајном површином. Управо то говори у прилог побољшањима које Ref-NeRF уводи. Слично закључујемо и за DRUMS скуп података.



Слика 6.7: Поређење mip-NeRF и Ref-NeRF модела на DRUMS скупу података

Управо слике 6.7 и 6.8 приказују разлике у квалитету сјајних површина, док се у осталим случајевима ова два модела понашају врло слично.



Слика 6.8: Поређење mip-NeRF и Ref-NeRF модела на MATERIALS скупу података

Ref-NeRF се најдуже обучава, а број параметара је у равни са обичним NeRF-ом. Како је за сваки модел у питању обучавање у трајању од око 5 дана, мишљења смо да разлика у том погледу није пресудна ни у ком случају. С тим у вези, доносимо закључак да је ReF-NeRF оптималан избор, али да се mip-NeRF намеће као алтернатива због двоструко мањег бројег параметара.



# Глава 7

## Закључак

У овом раду је дата основа могућег теоријског заснивања неуронских поља зрачења. Објашњени су основни физички и математички концепти, уз одговарајућа обrazложение и разлоге увођења. Сви модели су успешно имплементирани, а искоришћен је и пун потенцијал решења у облаку када је реч о њиховом обучавању. Резултати су приказани, а модели упоређени. Можемо рећи да су постигнути резултати у складу са ауторским радовима.

Коришћењем обучених модела можемо добити потпуно нове погледе сцене који нису били присутни у скупу обучавања. Такође, можемо добити глатке прелазе између погледа применом интерполације.

Даљи рад се може сагледати на основу недостатака које су обрађени модели не-двоносмислено показали.

- Неопходна је огромна рачунарска моћ. Обучавање траје дugo, а резултати још увек нису боли, па чак ни једнаки, у поређењу са традиционалним приступима.
- Нити један од обрађених модела не може да ради у реалном времену.
- Неопходно је имати веома прецизну поставку сцене, а добијени модел је обучен за само једну сцену.

На основу свега досад изреченог, можемо закључити да неуронска поља зрачења имају светлу будућност како у погледу истраживања, тако и у погледу примене. У питању је млада област која има велики потенцијал.



# Библиографија

- [1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” in *ECCV*, 2020.
- [2] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” *ICCV*, 2021.
- [3] 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,” *CVPR*, 2022.
- [4] K. Fukushima, “Cognitron: A self-organizing multilayered neural network,” *Biological Cybernetics*, vol. 20, pp. 121–136, 1975.
- [5] J. L. Sejun Park, Chulhee Yun and J. Shin, “Minimum width for universal approximation,” in *Proceedings of American Federation of Information Processing Societies: 1977 National Computer Conference*, ICLR, 2021.
- [6] J. Kiefer and J. Wolfowitz, “Stochastic estimation of the maximum of a regression function,” *The Annals of Mathematical Statistics*, vol. 23, pp. 462–466, 1952.
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.
- [8] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, and A. Courville, “On the spectral bias of neural networks,” 2019.
- [9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2017.
- [10] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, “Fourier features let networks learn high frequency functions in low dimensional domains,” 2020.
- [11] N. L. Max and M. Chen, “Local and global illumination in the volume rendering integral,” in *Scientific Visualization: Advanced Concepts*, 2010.
- [12] M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes, “Proposal for a standard default color space for the internet—srgb,” 1996.

- [13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.
- [14] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.
- [15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems* (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.
- [17] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.