

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Počítačem generovaná holografie

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 3. května 2016

David Herman

Poděkování

Mé poděkování patří Ing. Petru Lobazovi za odborné vedení, trpělivost a ochotu, kterou mi v průběhu zpracování bakalářské práce věnoval.

Abstract

The calculation of digital hologram structure is a computationally and memory intensive task. The main goal of this paper is to explore existing methods for computing digital holograms, implement selected method and compare it with the reference method, for this is considered a naive implementation of algorithm for computing a digital hologram structure. The first part of this paper is an introduction to the essence of holography and the dismantling of some of available methods to optimize the computation. The second part is focused on the actual implementation of the algorithm and presentation of the results achieved.

Abstrakt

Výpočet struktury digitálních hologramů představuje výpočetně i paměťově náročnou úlohu. Hlavním cílem této práce je prostudování metod pro výpočet struktury digitálních hologramů, implementace vybrané metody a její srovnání s referenční metodou, za tu je považována naivní implementace algoritmu pro výpočet digitálních hologramů. V první části práce je seznámení s podstatou holografie a rozebrání některých z dostupných metod pro optimalizaci výpočtu. Druhá část práce představuje samotnou implementaci algoritmu, a prezentování dosažených výsledků.

Obsah

1	Úvod	1
2	Princip Holografie	2
2.1	Klasický hologram	2
2.2	Digitální hologram	2
3	Výpočet digitálních hologramů	3
3.1	Šíření světla prostorem	3
3.2	Naivní algoritmus	4
3.2.1	Aliasing	7
3.3	Vyhledávací tabulky	10
3.4	Wavefront recording plane	11
3.4.1	Propagace optického pole	11
3.4.2	Princip WRP metody	14
3.4.3	Modifikace WRP metody	15
3.5	Objemové body	17
4	Numerická rekonstrukce	18
4.1	Zpětná propagace	18
4.2	Simulace tenké čočky	19
5	Implementace algoritmu	20
5.1	Architektura programu	21
5.2	Vstup dat	22
5.3	Vyhledávací tabulka	22
5.3.1	Generování objemového bodu	22
5.4	Implementace WRP metody	23
5.5	Numerická rekonstrukce hologramu	23
5.6	Vizualizace struktury optického pole	23

6	Dosažené výsledky	24
6.1	Srovnání s referenční metodou	24
6.2	Testování algoritmu	25
6.2.1	Objemové body	25
6.2.2	Řešení viditelnosti	26
6.2.3	Modifikovaná WRP metoda	27
7	Závěr	28
	Literatura	28
	Seznam zkratk	30
	Seznam obrázků	31

1 Úvod

Holografie je aktuálně nejdokonalejší technikou pro záznam 3D scény. Metoda klasické fotografie má zásadní nevýhodu v tom, že zaznamenáváme pouze amplitudu dopadající světelné vlny, tím ztratíme celkový 3D vjem obrazu. V případě holografie nezaznamenáváme pouze amplitudu, ale také fázi dopadající světelné vlny. Následně jsme tuto vlnu schopni pomocí hologramu reprodukovat. Kopie světelné vlny vytvořená hologramem je pak zcela k nerozeznání od původní zaznamenané vlny, která dopadala na rovinu hologramu.

Holografie se využívá v mnoha odvětvích, hologramy se využívají například v mikroskopii, strojírenství, ale i jako ochrana produktů před paděláním. Dokonce lze hologramem nahradit libovolný optický člen. V této práci se budeme zabývat výhradně obrazovou holografií, hologramy tedy budou zprostředkovávat realistický 3D vjem.

Podobně jako můžeme v počítači vytvářet fotorealistické scény a simulovat tak klasickou fotografii, lze i v případě holografie vytvářet syntetické hologramy. Nespornou výhodou tvorby digitálních hologramů je absence potřeby jakékoliv optické aparatury a stačí nám pouze počítač. Naopak zásadní nevýhodou je potřeba velkého výpočetního výkonu a zároveň dostatečně velké operační paměti. Tyto nároky se zvyšují s požadavky na kvalitu hologramu, tím spíše, chceme-li brát v potaz viditelnost jednotlivých bodů.

Existuje mnoho metod pro optimalizaci a urychlení výpočtu digitálních hologramů, některé z nich popisuje například článek s názvem *Computer-generated holograms by multiple wavefront recording plane method with occlusion culling* [5], ve kterém je zároveň představena nová metoda, díky které lze brát v úvahu viditelnost jednotlivých zdrojů světla. Cílem této práce je prostudování metod pro výpočet digitálních hologramů a implementace vybrané metody. Ta bude následně srovnána s naivní implementací algoritmu pro generování digitálních hologramů.

2 Princip Holografie

Hologram dokáže vytvořit dokonalou kopii původní světelné vlny. Tato kopie je k nerozeznání od předlohy dokonce i při pohledu mikroskopem na hologram.

2.1 Klasický hologram

Při tvorbě klasického hologramu je naším cílem světelné vlně, procházející rovinou hologramu, dát v rovině těsně za hologramem námi předem známou strukturu. Toho lze docílit díky znalosti jevu zvaného **Difrakce světla**, který můžeme velmi dobře pozorovat zejména při průchodu světelné vlny štěrbinou, jejíž šířka je srovnatelná s vlnovou délkou procházející světelné vlny. Známe-li šířku štěrbinu a vlnovou délku procházejícího světla, můžeme přesně vypočítat strukturu této světelné vlny po průchodu štěrbinou. Hologramem je pak velice komplikovaná difrakční struktura tvořená velkým množstvím těchto štěrbin. Hologramů je několik druhů, základní typy hologramů jsou reflexní a transmisní. V dnešní době jsou počítačem generované hologramy takřka výhradně transmisní.

2.2 Digitální hologram

Jelikož známe principy Fourierovské optiky, můžeme chování světelných vln šířících se prostorem simulovat v počítači a tedy generovat syntetické hologramy libovolného objektu. Při dostatečném výpočetním výkonu tak můžeme dosáhnout realistických výsledků bez nutnosti užití dalších prostředků. Objekty, jejichž hologram vytváříme jsou nejčastěji reprezentované jako mrak bodových zdrojů světla, nicméně lze vypočítat hologramy objektů reprezentovaných například trojúhelníkovou sítí. Každá z reprezentací vyžaduje specifický přístup a pro každou z reprezentací existuje mnoho metod urychlujících výpočet, eventuálně řešících viditelnost jednotlivých částí objektu. U digitálního hologramu lze taktéž provést numerickou rekonstrukci toho, co uvidíme při pohledu na hologram.

3 Výpočet digitálních hologramů

Při generování digitálních hologramů v počítači simulujeme to, co se ve skutečnosti děje v případě tvorby klasického hologramu. Před implementací libovolného algoritmu pro výpočet struktury hologramu je třeba si uvědomit, co se se světlem ve skutečnosti děje při tvorbě klasického hologramu. Díky tomu budeme moci lépe vysvětlit implementaci naivního algoritmu a také smysl jednotlivých metod pro optimalizaci výpočtu. Získáme ale také intuitivní představu o tom, co se se světlem děje a budeme tak schopni lépe posoudit správnost jednotlivých výsledků.

3.1 Šíření světla prostorem

Od bodového zdroje světla se souřadnicemi (η, ξ, ζ) se šíří kulová vlna, jejíž strukturu v rovině $(x, y, 0)$ můžeme popsat následující rovnicí [1]

$$U(x, y, 0) = \frac{A}{r} \exp(j \frac{2\pi}{\lambda} r), \quad (3.1)$$

kde j je imaginární konstanta. Hodnotou A označíme amplitudu bodového zdroje světla, λ udává vlnovou délku kulové vlny a hodnota r je vzdálenost bodového zdroje světla od bodu (x, y) v rovině $(x, y, 0)$. Tu definujeme pomocí vzorce

$$r = \sqrt{(x - \eta)^2 + (y - \xi)^2 + \zeta^2}. \quad (3.2)$$

Hodnotu kulové vlny, šířící se od bodového zdroje světla, v bodě $(x, y, 0)$ získané pomocí rovnice (3.1) nazýváme příspěvkem tohoto bodového zdroje světla v daném bodě roviny. Součet všech příspěvků všech bodových zdrojů světla v rovině $z = 0$ pak označujeme jako **objektovou vlnu**. Pro záznam struktury objektové vlny je současně s objektovou vlnou potřeba na rovinu hologramu posvítit tzv. **referenční vlnou**. Referenční vlnou může být také kulová vlna, která se šíří od bodového zdroje světla s jinými (x, y, z) souřadnicemi. Mnohem vhodnější a jednodušší je však jako referenční vlnu použít rovinnou vlnu dopadající na rovinu hologramu pod předem definovaným úhlem. Rovinnou vlnu si můžeme představit jako kulovou vlnu ve velké vzdálenosti od bodového zdroje světla, od něhož se tato kulová vlna šíří. Pro výpočet struktury rovinné vlny v bodě $(x, y, 0)$ použijeme vztah [1]

$$U(x, y, 0) = A \cdot \exp[j \frac{2\pi}{\lambda} \vec{n} \cdot (x, y, 0)]. \quad (3.3)$$

Veličina \vec{n} určuje vektor směru šíření rovinné vlny, jehož velikost je rovna jedné. Nutno poznamenat, že vlnová délka referenční vlny musí být shodná s vlnovou délkou vln tvořících objektovou vlnu. Referenční vlnu si zvolíme sami dle konkrétních požadavků na výsledný hologram, nicméně jsme ale omezeni maximálním úhlem, pod kterým může referenční vlna dopadat na rovinu hologramu. Toto omezení se odvíjí od vzorkovací frekvence plochy hologramu [1] (viz sekce (3.2.1)).

Pro vytvoření kopie zaznamenané objektové vlny je třeba hologram nasvítit zcela stejnou referenční vlnou, která byla použita při jeho tvorbě. Světelná vlna pak v rovině těsně za hologramem bude mít strukturu nerozeznatelnou od původní zaznamenané objektové vlny.

3.2 Naivní algoritmus

Před výpočtem struktury hologramu je nejprve třeba zadefinovat si několik veličin. Jednou z nich je vlnová délka λ , jejíž hodnotu zpravidla udáváme v rozsahu viditelného spektra světla, tedy přibližně od 380 nm až po 725 nm. Další veličinou, kterou je třeba definovat, je vzorkovací frekvence plochy hologramu, tu volíme většinou v řádu jednotek μm , v závislosti na požadované jemnosti a tedy kvalitě hologramu. Vzorkovací frekvence nám zároveň společně se zvoleným rozlišením hologramu dá představu o jeho reálných rozměrech.

Samotná implementace naivního algoritmu je pak velice jednoduchá. Objektovou vlnu bude tvořit mrak bodových zdrojů světla tvořený n body. Každý bod bude mít přiřazeny souřadnice (x_i, y_i, z_i) , stejně tak amplitudu A_i . Tu můžeme pro jednoduchost zvolit konstantní a prohlásit, že $A_i = 1$. Cílem implementace je pak pro každý bod roviny hologramu $(x, y, 0)$ provést výpočet hodnoty objektové vlny dle vzorce [1] [5]

$$U(x, y, 0) = \sum_{i=1}^n \frac{A_i}{r_i} \exp(j \frac{2\pi}{\lambda} r_i) \quad (3.4)$$

a následně přičíst referenční vlnu. Její amplitudu zvolíme přibližně dvojnásobku průměru amplitudy objektové vlny na ploše hologramu. Z takto vypočtené struktury optického pole hologram vytvoříme zobrazením jeho intenzity, tedy absolutní hodnoty umocněné na druhou. Pro testovací účely lze taktéž zobrazit čistě absolutní hodnotu, případně reálnou, nebo imaginární

část optického pole. Hodnotu amplitudy A u referenční vlny je nutno volit v závislosti na celkové amplitudě objektové vlny z toho důvodu, že jinak by se mohlo stát, že referenční vlna zcela přesvítí objektovou vlnu, jejíž příspěvek pak bude v porovnání s referenční vlnou zcela zanedbatelný. Stejná situace může nastat i naopak, kdy objektová vlna zcela přesvítí vlnu referenční. Obě tyto situace by měly za následek nesprávný výsledek, který ale nemusí být na první pohled zcela zřejmý. Z tohoto důvodu je velmi vhodné každou implementaci vždy nejprve otestovat na triviálních datech, například několik málo bodových zdrojů světla, kde si o výsledku lze intuitivně udělat představu předem. V případě, kdy objektovou vlnu tvoří velké množství bodových zdrojů světla lze totiž výsledek považovat takřka za matici náhodných čísel a nelze předem určit jeho strukturu.

Následující strana obsahuje pseudokód implementace naivního algoritmu, který zjistí strukturu objektové vlny v rovině hologramu. V pseudokódu není popsáno zjištění vhodné amplitudy pro referenční vlnu, její přičtení k výsledku a ani samotné zobrazení výsledku. Veškeré vzdálenosti jsou definovány v metrech.

Algorithm 1 Výpočet struktury objektové vlny v rovině hologramu

```

lambda = 633e - 9                                ▷ Vlnová délka
wavenum = 2 * PI/lambda                          ▷ Vlnové číslo
sampling = 5e - 6                                ▷ Vzorkovací vzdálenost

res_x = 1024                                     ▷ Rozlišení hologramu
res_y = 768
targ_field = alloc[res_x][res_y]

points_num = 100
p = generate_points()                            ▷ Vytvoření mraku bodových zdrojů světla

corner_x = -res_x/2 * sampling                  ▷ X souřadnice rohu
corner_y = -res_y/2 * sampling                  ▷ Y souřadnice rohu

for i = 1 : points_num do
  for j = 1 : res_x do
    for k = 1 : res_y do
      targ_x = corner_x + j * sampling           ▷ Aktuální x souřadnice
      targ_y = corner_y + k * sampling           ▷ Aktuální y souřadnice

      r = sqrt((targ_x - p[i].x)2 + (targ_y - p[i].y)2 + (p[i].z)2)
      targ_field[j][k] += exp(I * wavenum * r)/r
    end for
  end for
end for

```

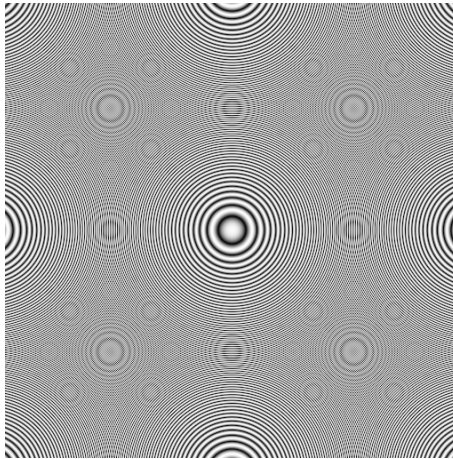
Na první pohled je zřejmá velká výpočetní náročnost, vzhledem k rozlišení hologramu a jeho vzorkování by skutečná velikost výsledku byla 0.51 cm na šířku a 0.384 cm na výšku. Při reálných výpočtech není výjimkou generovat hologramy o obsahu plochy 100 cm² a při vzorkování 1 μm. Z toho plyne i vyšší paměťová náročnost. Dalším aspektem je zde počet bodů, kterými je objekt reprezentován, v závislosti na velikosti a detailnosti objektu můžeme očekávat, že počet bodových zdrojů světla přispívajících do objektové vlny bude srovnatelný s počtem vzorků plochy hologramu.

Díky skutečnosti, že hodnota *targ_field* v bodě (*j*, *k*) je nezávislá na libovolném jiném bodě *targ_field*, je tak výpočet struktury optického pole v rovině hologramu velmi dobře paralelizovatelný na CPU (v případě GPU narážíme na problém s maximální velikostí souvislé alokované paměti, který lze ale také řešit). Paralelizace je však jen jedna z možností urychlení výpočtu.

3.2.1 Aliasing

Budeme-li testovat implementaci naivního algoritmu v situacích, kdy referenční vlna dopadá na rovinu hologramu pod velkým úhlem, případně jednotlivé bodové zdroje světla umístíme do velmi malé vzdálenosti od roviny hologramu (jednotky mm), zjistíme ve vypočtené struktuře optického pole artefakty, které s jeho skutečnou strukturou nemají nic společného. Jedná se o problém **aliasingu**, nebo také **podvzorkování**. Neřešení tohoto problému má za následek zcela chybné a fyzikálně neopodstatněné výsledky algoritmu.

Obrázek 3.1: Ukázka podvzorkovaného fázoru (viz okraje)



Abychom mohli přesně zrekonstruovat spojitý signál, využijeme **Shannonův teorém**, který nám říká: „*Přesná rekonstrukce spojitého, frekvenčně omezeného signálu z jeho vzorků je možná tehdy, pokud byl vzorkován frekvencí alespoň dvakrát vyšší, než je maximální frekvence signálu*“ [3].

Lokální frekvenci fázoru kulové vlny šířící se od bodového zdroje světla jsme schopni vyjádřit dle rovnice

$$l_x(x_0, y_0, z_0) = \frac{\partial}{\partial x} \frac{\sqrt{x^2 + y_0^2 + z_0^2}}{\lambda} \Big|_{x=x_0}. \quad (3.5)$$

Úpravou a rozepsáním rovnice (3.5) se nakonec dostaneme k ekvivalentnímu zápisu

$$l_x(x_0, y_0, z_0) = \frac{|x_0|}{\lambda \sqrt{x_0^2 + y_0^2 + z_0^2}}. \quad (3.6)$$

Abychom zjistili maximální vzorkovací vzdálenost D , kterou si můžeme dovolit nastavit, abychom se vyhnuli aliasingu, je třeba vyřešit rovnici

$$2D = \frac{1}{l_x}. \quad (3.7)$$

Limitním případem řešení rovnice (3.7) je $D = \lambda/2$, pokud tedy vzorkovací vzdálenost zvolíme maximálně na tuto hodnotu, naprosto se tím vyhneme problému aliasingu. To ale bude mít za následek obrovský nárůst výpočetního času potřebného na vygenerování hologramu, stejně tak se razantně navýší i paměťová náročnost. Pro hologramy o obsahu plochy několik jednotek, až desítek cm^2 pak totiž bude třeba generovat hologramy mnohem většího rozlišení.

Pokud k problému budeme přistupovat z druhé strany a zprvu pevně zafixujeme vzorkovací vzdálenost D , můžeme následně zjistit oblast na ploše hologramu, kde lokální frekvence fázoru kulové vlny bude rovna alespoň dvojnásobku vzorkovací vzdálenosti. Tímto způsobem sice můžeme ztratit část informace o objektové vlně, nicméně ztratíme pouze tu část, kterou bychom nemohli zcela přesně zrekonstruovat a tedy část, která by do výsledku zanesla chyby. Chybu do výsledku ale přesto zaneseme, neboť oříznutí fázoru pouze na malou část na ploše hologramu si můžeme představit jako situaci, kdy mezi bodovým zdrojem světla a rovinou hologramu je stínítko s otvorem o rozměrech menších, než je plocha hologramu. Naopak ale takto můžeme celý výpočet mírně urychlit, neboť každý bodový zdroj světla bude přispívat pouze do korektně navzorkované oblasti, která může být značně menší, než je výsledná plocha hologramu.

Zafixujme vzorkovací vzdálenost D a nyní hledějme řešení x pro rovnici (3.7). Postupnými úpravami a rozepsáním se dostaneme k řešení

$$x_{limit} = \sqrt{\frac{y^2 + z^2}{(2D/\lambda)^2 - 1}}. \quad (3.8)$$

Tuto rovnici lze následně upravit, abychom získali ekvivalentní zápis ve tvaru

$$\frac{x_{limit}^2}{z^2[(2D/\lambda)^2 - 1]^{-1}} - \frac{y^2}{z^2} = 1, \quad (3.9)$$

z čehož jasně plyne, že korektně navzorkovaná oblast, pro vzorkovací vzdálenosti D , je ohraničena hyperbolou. Analogicky bychom postupovali při zjišťování hodnoty y_{limit} , kde bychom za předpokladu, že osy x a y jsou vzorkovány stejnou vzorkovací vzdáleností, dospěli ke zcela stejnému řešení. Avšak

počítání příspěvků jednotlivých bodových zdrojů světla jen do korektně navzorkované oblasti za pomoci tohoto vzorce by znamenalo výpočet hranice pro každý řádek, respektive sloupec roviny hologramu. Pro jednoduchost tedy za korektně navzorkovanou oblast prohlásíme oblast na ploše hologramu ohraničenou obdélníkem, jehož velikost strany na ose x bude odpovídat řešení x_{limit} rovnice (3.9), do které dosadíme $y = 0$. Obdobným způsobem budeme postupovat při výpočtu velikosti strany obdélníku v ose y . Pokud bude vzorkovací vzdálenost v osách x a y stejná, bude platit $x_{limit} = y_{limit}$. Následující pseudokód popisuje výpočet struktury objektové vlny v rovině hologramu při omezení se jen na korektně navzorkovanou oblast.

Algorithm 2 Počítání objektové vlny s pomocí lokálního ořezávání

```

for  $i = 1 : points\_num$  do
   $x\_lim = (p[i].z / \sqrt{(2D / \lambda)^2 - 1}) / sampling$ 
   $y\_lim = (p[i].z / \sqrt{(2D / \lambda)^2 - 1}) / sampling$ 

   $index\_x\_start = (p[i].x - corner\_x) / sampling - x\_lim$ 
   $index\_y\_start = (p[i].y - corner\_y) / sampling - y\_lim$ 

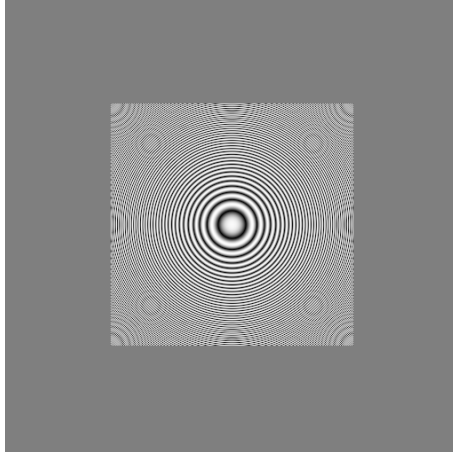
   $index\_x\_end = index\_x\_start + 2 * x\_lim$ 
   $index\_y\_end = index\_y\_start + 2 * y\_lim$ 

  for  $j = index\_x\_start : index\_x\_end$  do
    for  $k = index\_y\_start : index\_y\_end$  do
       $targ\_x = corner\_x + j * sampling$ 
       $targ\_y = corner\_y + k * sampling$ 

       $r = \sqrt{(targ\_x - p[i].x)^2 + (targ\_y - p[i].y)^2 + (p[i].z)^2}$ 
       $targ\_field[j][k] += \exp(I * wavenum * r) / r$ 
    end for
  end for
end for

```

Obrázek 3.2: Fázor (3.2.1) s omezením na korektně navzorkovanou oblast



Při fixní hodnotě vzorkovací vzdálenosti je ale také třeba brát v potaz úhel, pod kterým na rovinu hologramu dopadá referenční vlna. Interferenční rovnice (3.10) [1] nám říká, jaká bude frekvence vzniklého interferenčního vzoru, budou-li na něj dopadat dvě světelné vlny pod úhly o velikostech θ_1 a θ_2 .

$$d = \lambda / (\sin \theta_1 - \sin \theta_2). \quad (3.10)$$

Hodnota d udává vzdálenost mezi maximem, respektive minimem hodnot vzniklého interferenčního vzoru. Na základě této rovnice musíme s ohledem na zvolenou vzorkovací vzdálenost D vypočítat maximální úhly, pod kterými můžou světelné vlny dopadat na rovinu hologramu. V rámci testování budeme volit hodnotu úhlu referenční vlny rovnu 0.

3.3 Vyhledávací tabulky

Struktura optického pole kulové vlny, která se šíří od bodového zdroje světla, závisí pouze na z -ové vzdálenosti tohoto bodu od roviny hologramu. Této skutečnosti lze využít pro vytvoření vyhledávací tabulky čímž docílíme snížení celkového času potřebného k vypočtení struktury hologramu. Kvůli nutnosti uchovávat vyhledávací tabulku v paměti tím ale navýšíme paměťovou náročnost. Prostor mraku bodových zdrojů světla v z -ové ose diskretizujeme, přičemž vzorkovací vzdálenost, respektive počet vzorků zvolíme dle vlastního uvážení. Do vyhledávací tabulky následně budeme ukládat vypočtené struktury optických polí, které budou odpovídat kulovým vlnám šířících se

od bodových zdrojů světla se souřadnicemi $(0, 0, z_i)$, kde z_i je vzdálenost bodového zdroje pro i -tou položku vyhledávací tabulky. Jakmile pak zjišťujeme strukturu objektové vlny, pro každý bodový zdroj světla vybere takový prvek ve vyhledávací tabulce, který bude vypočten ve vzdálenosti nejbližší z -ové souřadnici tohoto bodového zdroje světla. Při kopírování před-vypočtené struktury optického pole z vyhledávací tabulky je ještě nutno ji posunout na základě (x, y) souřadnic právě zpracovávaného bodového zdroje světla. Už během vytváření prvků vyhledávací tabulky budeme struktury optických polí ořezávat pouze na korektně navzorkovanou oblast, zároveň je vhodné definovat si taktéž maximální rozlišení jednoho prvku ve vyhledávací tabulce, neboť pro velké vzdálenosti mohou mít prvky vyhledávací tabulky rozlišení mnohem větší, než je původní rozlišení hologramu.

Počet záznamů ve vyhledávací tabulce bude mít vliv na výslednou kvalitu hologramu. Můžeme jej nastavit na fixní hodnotu, nebo nastavovat proměnlivě na základě geometrie mraku bodových zdrojů světla, tedy na základě nejbližšího, nejvzdálenějšího bodu a celkového počtu bodů. Také by pravděpodobně bylo možné na základě histogramu z -ových souřadnic jednotlivých bodů zjistit zcela konkrétní vzdálenosti od roviny hologramu, ve kterých si následně předpočítat záznamy do vyhledávací tabulky.

3.4 Wavefront recording plane

Metoda Wavefront recording plane (též **WRP**) [5] přináší další urychlení při výpočtu struktury digitálního hologramu, avšak za cenu další ztráty části informace o objektové vlně. Abychom mohli princip metody WRP dobře vysvětlit, je nejprve třeba popsat metody propagace optického pole, tedy toho, jak se světlo šíří mezi dvěma rovnoběžnými rovinami.

3.4.1 Propagace optického pole

Již známe a umíme vypočítat strukturu optického pole světelné vlny šířící se od bodového zdroje světa, respektive od mraku bodových zdrojů světla. Propagace optického pole nám popisuje, jak se světlo šíří mezi dvěma rovnoběžnými rovinami. Algoritmy počítající propagaci optického pole mezi dvěma rovnoběžnými rovinami využívají rychlou Fourierovu transformaci, tu jsme schopni pro n vzorků vypočítat s časovou složitostí $O(n \log n)$ [1].

Tato práce využívá dva různé způsoby pro vypočtení propagace optického pole a to **Rayleigh-Sommerfeldův integrál** a metodu **rozkladu do úhlového spektra**. Oba způsoby propagace jsou popsány v následujících sekcích, včetně jejich srovnání.

Pokud bychom počítali propagaci klasickým způsobem, kdy bychom v cílové rovině pouze sečetli příspěvky všech bodů ze zdrojové roviny, celková složitost propagace by pak byla $O(n^4)$. Fourierovu transformaci pro dvourozměrný signál jsme ale schopni vypočítat se složitostí $O(n^2 \log n)$.

Rayleigh-Sommerfeldův integrál

Rayleigh-Sommerfeldův integrál [6] [1] je postup využívající tři rychlé Fourierovy transformace, z toho dvě dopředné a jednu zpětnou. Lze jej vyjádřit následující rovnicí.

$$U(x, y, z) = -\frac{1}{2\pi} \iint_{-\infty}^{\infty} U(\eta, \xi, 0) \left(j \frac{2\pi}{\lambda} - \frac{1}{r} \right) \frac{\exp(j \frac{2\pi}{\lambda} r) z}{r} d\eta d\xi, \quad (3.11)$$

kde $r = \sqrt{(x - \eta)^2 + (y - \xi)^2 + z^2}$. Díky rovnoběžnosti rovin můžeme rovnici (3.11) zapsat pomocí konvoluce [6]

$$U(x, y, z) = \iint U(\eta, \xi, 0) k(x - \eta, y - \xi, z) d\eta d\xi, \quad (3.12)$$

přičemž konvoluční jádro (impulsní odezvu) definujeme jako

$$k(x, y, z) = -\frac{1}{2\pi} \left(j \frac{2\pi}{\lambda} - \frac{1}{r} \right) \frac{\exp(j \frac{2\pi}{\lambda} r) z}{r}. \quad (3.13)$$

Aplikací konvolučního teorému pak rovnici (3.12) přepíšeme na tvar

$$U(x, y, z) = \mathcal{F}^{-1} \{ \mathcal{F}[U(x, y, 0)] \mathcal{F}[k(x, y, z)] \}, \quad (3.14)$$

operace \mathcal{F}^{-1} a \mathcal{F} označují zpětnou a dopřednou Fourierovu transformaci.

Nutno poznamenat, že struktura konvolučního jádra vypočtená ve vztahu (3.13) se může také potýkat s problémem aliasingu. Stejně jako v případě fázoru kulové vlny, která se šíří od bodového zdroje světla i zde je třeba omezovat se na korektně navzorkovanou oblast. Lokální frekvenci v ose x ve struktuře konvolučního jádra pro *Rayleigh-Sommerfeldův integrál* lze vyjádřit jako

$$l_x(x, y, z) = \frac{|x|}{\lambda r}, \quad (3.15)$$

obdobně lze zapsat vzorec pro zjištění lokální frekvence v ose y .

Rozklad do úhlového spektra

Na rozdíl od *Rayleigh-Sommerfeldova integrálu*, metoda *rozkladu do úhlového spektra* popisuje propagaci světla mezi dvěma rovnoběžnými rovinami, která ke svému výpočtu potřebuje pouze dvě rychlé Fourierovy transformace. Její předpis zjednodušeně můžeme zapsat pomocí rovnice [2] [1]

$$U(x, y, z) = \mathcal{F}^{-1}\{\mathcal{F}(U(x, y, 0)) \cdot \exp[-j2\pi z(1/\lambda^2 - f_x^2 - f_y^2)^{1/2}]\}. \quad (3.16)$$

Hodnoty f_x a f_y jsou frekvence v osách x a y . Srovnáme-li vztahy (3.14) a (3.16), dojdeme k závěru, že přenosovou funkci můžeme analyticky vyjádřit vztahem

$$\mathcal{F}(k(x, y, z)) = \exp[-j2\pi z(1/\lambda^2 - f_x^2 - f_y^2)^{1/2}]. \quad (3.17)$$

Lokální frekvence přenosové funkce jsou pak vyjádřeny rovnicemi

$$l_x(x, y, z) = \frac{|zf_x|}{(1/\lambda^2 - f_x^2 - f_y^2)^{1/2}}, \quad (3.18)$$

$$l_y(x, y, z) = \frac{|zf_y|}{(1/\lambda^2 - f_x^2 - f_y^2)^{1/2}}. \quad (3.19)$$

Srovnání obou metod

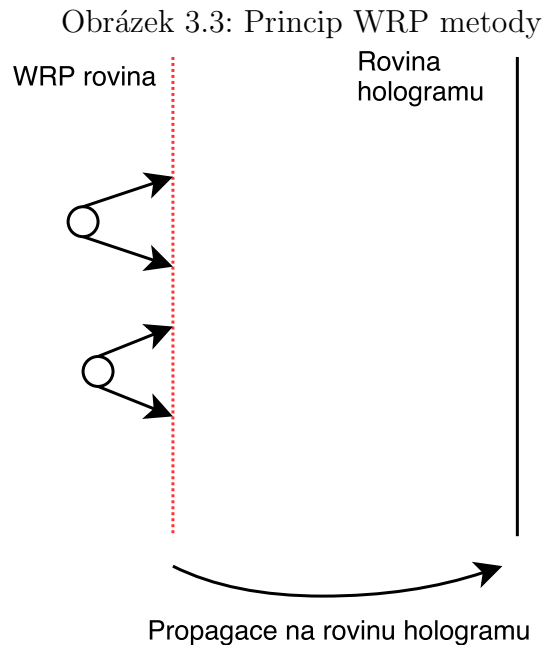
Při zobrazení struktury propagačních jader obou metod zjistíme, že lokální frekvence se u Rayleigh-Sommerfeldova integrálu zvyšuje s klesající propagační vzdáleností, respektive snižuje s rostoucí vzdáleností. U metody rozkladem do úhlového spektra je tento jev přesně opačný, mezi propagační vzdáleností a lokální frekvencí je přímá úměra. To ostatně plyne i ze vztahů pro výpočet propagačních jader obou metod.

Z předchozího odstavce je zřejmé, že propagace pomocí metody Rayleigh-Sommerfeldova integrálu je vhodná pro větší propagační vzdálenosti, kdežto pro malé vzdálenosti je vhodnější metoda pomocí rozkladu do úhlového spektra. Snadno lze také naimplementovat rozhodující mechanismus, který nejprve zjistí, zda jádro jedné metody podléhá aliasingu a pokud ano, použije se metoda druhá. Ve většině případů se ukazuje, že je-li jádro jedné z těchto metod aliasované, jádro druhé metody aliasingu nepodléhá.

3.4.2 Princip WRP metody

WRP metoda využívá faktu, že čím menší je vzdálenost zdroje světla od roviny hologramu, čím menší je korektně navzorkovaná oblast, do které tento bod přispívá. Nejprve tedy vypočteme strukturu objektové vlny v rovině těsně před mrakem bodových zdrojů světla a následně provedeme propagaci takto vypočteného optického pole na rovinu hologramu. Využijeme-li vyhledávací tabulku, značně pomocí WRP metody zredukujeme velikost vyhledávací tabulky v paměti, neboť její prvky budou předpočítány v menších vzdálenostech.

Výhodu použití WRP metody lze snadno demonstrovat na srovnání složitosti s naivní implementací. Složitost naivní implementace pro M bodů je $O(M \times N^2)$, N^2 je počet vzorků roviny hologramu. Počet vzorků na WRP rovině, do které přispívá zdroj světla nyní označme jako L^2 . Společně se složitostí propagace optického pole nám pak vyjde složitost WRP metody rovna $O(M \times L^2 + N^2 \log N)$. Tento tvar můžeme upravit na podobu $O(M \times N^2 (\frac{L^2}{N^2} + \frac{\log N}{M}))$, z čehož vyplývá, že WRP metoda v porovnání s naivní implementací přináší efektivní výpočet v případě, pokud platí $(\frac{L^2}{N^2} + \frac{\log N}{M}) < 1$. Hodnota $\frac{\log N}{M}$ je typicky zanedbatelná a hodnota L se snižuje společně s klesající vzdáleností WRP roviny od zdroje světla.



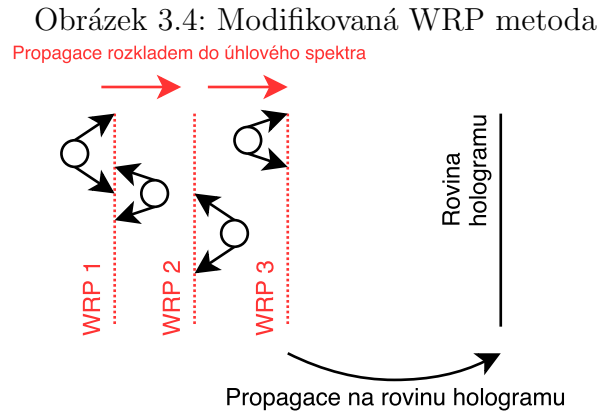
3.4.3 Modifikace WRP metody

Pro další snížení času potřebného pro výpočet struktury hologramu můžeme WRP metodu modifikovat tak, že nebudeme zjišťovat objektovou vlnu v rovině těsně před mrakem bodových zdrojů světla, ale celý mrak proložíme předem definovaným počtem WRP rovin s konstantní vzdáleností mezi sebou. Každý bodový zdroj světla pak bude přispívat pouze do nejbližší WRP roviny. Zde je třeba zmínit, že je-li relativní vzdálenost bodu od roviny, do které přispívá, záporná, je nutno jako příspěvek použít konjugovanou vlnu, tedy vlnu se záporným znaménkem u imaginární konstanty. Metoda pak funguje tak, že vypočteme strukturu optického pole z příspěvků bodových zdrojů světla u nejbližší WRP roviny a tu následně propagujeme na nejbližší bližší WRP rovinu a celý postup opakujeme. Nejbližší WRP rovinu k rovině hologramu pak propagujeme rovnou na rovinu hologramu.

Problém nastává v situacích, kdy je bodový zdroj světla velmi blízko WRP roviny, do které přispívá, nebo dokonce tato rovina přímo prochází tímto bodem. Může se stát, že bod tak bude přispívat do oblasti o velikosti menší, než je vzorkovací vzdálenost a díky tomu se tento bod do objektové vlny vůbec nepromítne. Proto je nutné definovat minimální vzdálenost, ve které bodové zdroje světla od WRP rovin mohou být a body, které jsou blíže, do této vzdálenosti posunout.

Tato metoda má hned několik výhod, jednak minimalizuje velikost vyhledávací tabulky, čímž i urychluje výpočet optického pole v rámci jedné roviny. Především je ale mezi jednotlivými rovinami konstantní vzdálenost, takže pro propagaci mezi těmito rovinami si můžeme předpočítat propagační jádro. Na následující straně je popsáno, jak tato metoda umožňuje snadným způsobem řešit viditelnost jednotlivých zdrojů světla.

Rozepsáním složitosti modifikované WRP metody můžeme ukázat její efektivnost oproti jednoduché WRP metodě. Na základě vztahu (3.9) můžeme říct, že se zvyšujícím se k se přímo úměrně snižuje hodnota L , nicméně pro výpočet struktury hologramu je pak třeba k propagací optického pole. Pro k WRP rovin tedy bude mít modifikovaná WRP metoda celkovou složitost $O(M \times N^2 (\frac{L^2/k^2}{N^2} + k \frac{\log N}{M}))$. Optimální počet WRP rovin je pak roven takové hodnotě $k > 0$, pro kterou má vztah $\frac{L^2/k^2}{N^2} + k \frac{\log N}{M}$ minimální hodnotu.



Řešení viditelnosti

Při výpočtu digitálního hologramu narážíme na problém viditelnosti jednotlivých bodů. Mějme dva body se souřadnicemi $(0, 0, 0.1)$ a $(0, 0, 0.11)$, metody dosud popsané v této práci zcela ignorovaly fakt, že při pohledu na hologram pod nulovým úhlem bychom neměli vůbec zaznamenat příspěvek vzdálenějšího bodu. V případě komplikovaného objektu tvořeného mrakem bodových zdrojů světla je pak vidět zároveň přední i zadní část objektu, což kazí celkový 3D vjem hologramu.

Možností řešení tohoto problému je několik, avšak mnoho z nich je výpočetně náročných. Článek [5], zmíněný v úvodu, představuje nový způsob, jak řešit viditelnost jednotlivých bodových zdrojů světla, který v kombinaci s modifikovanou WRP metodou přináší překvapivě dobré výsledky.

Před připočtením příspěvku každého bodového zdroje světla je malá oblast optického pole, kam tento bod přispívá, utlumena [5], eventuálně nastavena na hodnotu 0. Následně se připočte příspěvek daného zdroje světla a tento postup se opakuje pro každý přispívající zdroj světla. Nutno dostat, že pro fungování této metody je bezpodmínečně nutné příspěvky jednotlivých bodových zdrojů světla přidávat postupně od nejvzdálenějšího, po nejbližší. Oblast optického pole, jenž budeme utlumovat je ve tvaru čtverce o konstantní velikosti hrany m . Výpočet velikosti této oblasti předpokládá rovnoměrné rozložení bodových zdrojů světla v oblaku. Nejprve je třeba vypočítat hustotu bodů na pixel ρ_p při promítnutí oblaku na rovinu. To zjistíme pomocí vztahu

$$\rho_p = \frac{n}{D^2}, \quad (3.20)$$

příčemž n je počet bodových zdrojů světla v mraku, h a w jsou výška a šířka mraku a D je vzorkovací vzdálenost. Velikost oblasti m je pak dána rovnicí

$$m = \frac{1}{2} \frac{D}{\rho_p}. \quad (3.21)$$

Smyslem tohoto vzorce je, aby se body, ležící ve stejné vzdálenosti, touto metodou řešení viditelnosti nikterak neovlivňovaly.

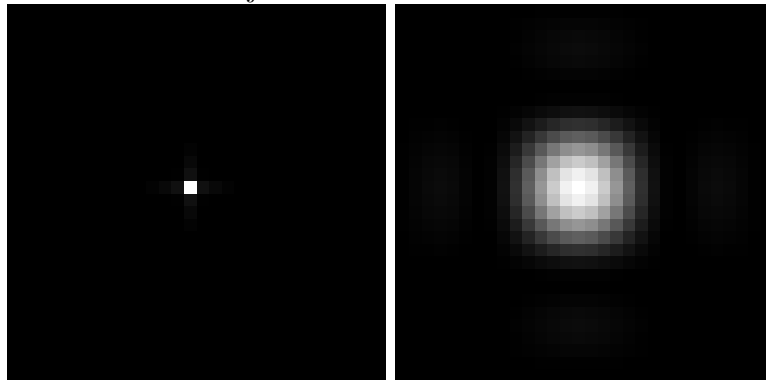
3.5 Objemové body

V reálném světě kolem nás neexistuje nic jako bezrozměrný, bodový, zdroj světla. Nadefinováním objemu jednotlivým zdrojům světla tedy dosáhneme realističtějších výsledků [5]. Bodový zdroj světla tak nahradíme množinou bodů, kde každý bod má amplitudu přiřazenou pomocí rovinné Gaussovské distribuce dané vztahem

$$g(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \quad (3.22)$$

Hodnota σ je směrodatná odchylka distribuce, pomocí které upravujeme vzhled objemového zdroje světla, stejně tak lze navolit jeho rozměry. Vhodnou volbou těchto dvou veličin můžeme dosáhnout dojmu hladkého povrchu objektu. V kombinaci s vyhledávací tabulkou lze objemové body velice snadno integrovat do takřka libovolného algoritmu, jelikož objemový bod použijeme jako referenční prvek při vytváření vyhledávací tabulky.

Obrázek 3.5: Porovnání bodového a objemového zdroje světla, obrázek vlevo ukazuje bodový zdroj světla, obrázek vpravo pak objemový zdroj světla. Reálná velikost obou obrázků je 0.18×0.18 mm.



4 Numerická rekonstrukce

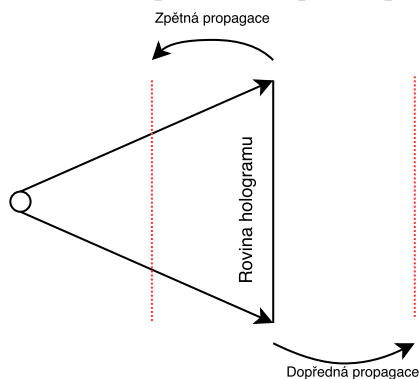
Dokážeme-li vytvořit digitální hologram, jsme schopni provést také rekonstrukci toho, co přibližně uvidíme při pohledu na hologram. K tomuto účelu můžeme použít například zpětnou propagaci optického pole, případně můžeme provést simulaci tenké čočky.

4.1 Zpětná propagace

Rovnice (3.14) a (3.16) nám říkají, jak bude vypadat struktura optického pole po tom, co urazí nezápornou vzdálenost, mluvíme zde o dopředné propagaci. Propagace zpětná nám popisuje strukturu optického pole v záporné vzdálenosti, tedy popisuje jeho strukturu před tím, než se našířilo do aktuálního stavu. Výpočet zpětné propagace je takřka totožný s výpočtem propagace dopředné, jediným rozdílem je změna znaménka imaginární konstanty při výpočtu propagačního jádra.

Máme-li vypočtenou strukturu optického pole kulové vlny, která se šíří od zdroje světla se souřadnicemi $(0, 0, 0.1)$, tak zpětnou propagací o vzdálenost 10 cm a následnou vizualizací výsledku zjistíme, že uprostřed obrázku je rozsvícen jeden světlý pixel, přičemž ostatní jsou tmavé. To odpovídá popsané geometrii scény a tento jeden pixel představuje zdroj světla, který uvidíme při pohledu na hologram. Nevýhodou použití zpětné propagace je však nemožnost simulace pohledu na hologram pod různými úhly. K tomu nám slouží simulace tenké čočky.

Obrázek 4.1: Dopředná a zpětná propagace



4.2 Simulace tenké čočky

Chceme-li získat věrohodnou představu o tom, co se ve skutečnosti promítne na sítnici oka při pohledu na hologram, je nutné provést simulaci tenké čočky, neboť světlo dopadající na sítnici oka zprvu prochází oční čočkou.

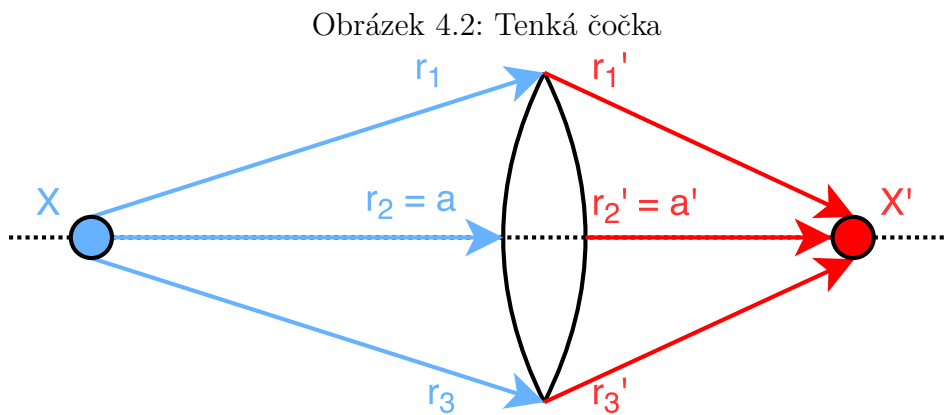
Tenká čočka je optický člen, u něhož zanedbáváme jeho tloušťku. Můžeme prohlásit, že čočka upravuje fázi procházejícího světla. Pokud bychom chtěli simulovat lidské oko, pak mezi hologram a sítnici oka tedy postavíme takovou čočku, která nám pozmění fázi světla tak, že se světelné vlny sečtou ve vzdálenosti sítnice od oční čočky (tedy přibližně 20 mm).

Změnu fáze komplexního čísla lze provést tak, že jej pouze vynásobíme jiným komplexním číslem. Díky znalosti geometrie toho, v jaké vzdálenosti je čočka umístěná od roviny hologramu a zároveň v jaké vzdálenosti je sítnice od čočky, víme dopředu, jakým způsobem nám musí čočka pozměnit fázi procházejícího světla.

Propagací optického pole do vzdálenosti, ve které je umístěna čočka, se fáze optického pole změní o $2\pi r/\lambda$, následnou propagací na sítnici se fáze změní o další $2\pi r'/\lambda$. Tenká čočka tedy musí provést takový fázový posun, který od optického pole odečte fázový posun vzniklý jeho propagací do nulové vzdálenosti. Změnu fáze čočky tedy můžeme vyjádřit pomocí

$$\varphi = -\frac{2\pi}{\lambda}(r + r') = -\frac{2\pi}{\lambda}[(x^2 + y^2 + a^2)^{1/2} + (x^2 + y^2 + a'^2)^{1/2}], \quad (4.1)$$

kde a je z -ová vzdálenost zdroje světla od roviny čočky a a' je vzdálenost sítnice, respektive roviny, kam čočka zaostřuje, od čočky.



5 Implementace algoritmu

V kapitole 3 byly zmíněny metody pro urychlení výpočtu struktury digitálního hologramu. Většina z těchto metod je v rámci bakalářské práce implementována, přičemž celý algoritmus vychází z článku [5], který byl zmíněn v úvodu. Tato kapitola se věnuje popisu samotné implementace. Větší důraz je zde kladen na celkovou architekturu programu a jeho krokům, než na podrobný popis jednotlivých funkcí.

Celý program je vytvořen v jazyce C, jako hlavní důvod pro volbu tohoto jazyka je jeho rychlost a kontrola nad alokovanou pamětí, neboť výpočet digitálních hologramů je výpočetně i pamět'ově náročná úloha. Současně je implementace algoritmu velice jednoduchá a nevyžaduje použití složitých datových struktur, ani složitou objektovou dekompozici. Jednotlivé kroky programu po spuštění vypadají následovně

1. Načti vstupní data z konfiguračního souboru
2. Načti zdrojový objekt tvořený mrakem bodových zdrojů světla
3. Vytvoř vyhledávací tabulku
4. Proveď modifikovanou WRP metodu
 - (a) Vypočti maximální vzdálenost, ve které se zdroje světla mohou nacházet od jednotlivých WRP rovin
 - (b) K nejbližší WRP rovině přiřti příspěvky zdrojů světla a tuto rovinu propaguj na nejbližší bližší WRP rovinu. Takto postup opakuj, dokud nebude provedena propagace na nejbližší WRP rovinu.
 - (c) Přiřti příspěvky zdrojů světla k nejbližší WRP rovině a tuto rovinu propaguj na rovinu hologramu.

Posledním krokem programu je pak výpočet numerické rekonstrukce pomocí simulace tenké čočky. Tento krok však přímo nesouvisí s implementací algoritmu a v programu je pouze za účelem otestování výsledku.

5.1 Architektura programu

Program je rozdělen do několika zdrojových souborů pro zvýšení čitelnosti kódu. V této sekci jsou zmíněny pouze zdrojové soubory s funkčním kódem (příponou `.c`), hlavičkové soubory zde nejsou uvedeny.

- `common.c` - Pomocné funkce, využitelné kdekoliv v programu.
- `config.c` - Funkce obsluhující načtení a parsování konfiguračního souboru. Jsou zde také funkce pro získání hodnot jednotlivých konfiguračních voleb, které jsou tak dostupné odkudkoliv z programu.
- `hologram.c` - Implementace zobrazení struktury optického pole do obrázku a využitím knihovny `LodePNG`.
- `light_source.c` - Funkce pro načtení mraku zdrojů světla do paměti.
- `log.c` - Funkce zajišťující výpis zpráv na konzoli.
- `lookup_table.c` - Implementace vyhledávací tabulky, zde je také definice funkce pro vynošení objemového bodu.
- `main.c` - Hlavní funkce (vstupní bod) programu.
- `memory.c` - Wrapper pro funkce `malloc` a `fftw_malloc`, které kontrolují, zda se paměť podařilo alokovat.
- `propagation.c` - Modul implementující propagaci optického pole, je zde implementace jak *Rayleigh-Sommerfeldova integrálu*, tak metoda *rozkladu do úhlového spektra*.
- `reconstruction.c` - Funkce pro numerickou rekonstrukci hologramu.
- `wrp.c` - Implementace modifikované WRP metody.

Pro vizualizaci výsledku byla použita knihovna **LodePNG**, jejíž zdrojový kód je uložen v souboru `lodepng.c`. Autorem knihovny je *Lode Vandevenne*. Zároveň je v programu využívána knihovna pro výpočet rychlé Fourierovy transformace **FFTW3**.

5.2 Vstup dat

Funkce zajišťující vstup dat jsou definovány v souboru `config.c`. Veškerá vstupní data programu jsou zadávána pomocí konfiguračního souboru, jeho název je pak zadán jako parametr spustitelného souboru. Při neuvedení žádného parametru je jako výchozí název konfiguračního souboru použit `holo-project.cfg`. Program je také schopen detekovat chybějící povinné volby v konfiguraci a zřetelně na ně uživatele upozornit. Podrobný popis struktury konfiguračního souboru, včetně všech jeho voleb, je uveden v ukázkovém konfiguračním souboru na příloženém CD.

5.3 Vyhledávací tabulka

Implementace vyhledávací tabulky je definována v souboru `lookup_table.c` a je řešena uložením jednotlivých prvků do pole, přičemž prvky v tomto poli jsou seřazeny vzestupně dle z -ové vzdálenosti, ve kterých jsou fázory předpočítány. Položkou vyhledávací tabulky je struktura obsahující informace o skutečné velikosti, o velikosti v pixelech, vzdálenosti na ose z a nakonec samotnou strukturu fázoru objektové vlny, která se šíří od referenčního prvku. Ten může být jak objemový, tak bodový zdroj světla, v závislosti na nastavení. Počet prvků vyhledávací tabulky je definován v konfiguračním souboru.

Mezní hodnoty z -ových vzdáleností, ve kterých jsou předpočítány struktury optických polí do vyhledávací tabulky jsou zadány uživatelem. Počáteční hodnota je určena explicitně a lze ji interpretovat jako minimální vzdálenost, ve které se bod od roviny musí nacházet. Horní hodnota je pak odvozena od počtu rovin v rámci modifikované WRP metody.

5.3.1 Generování objemového bodu

Objemový bod je reprezentován množinou bodových zdrojů světla, jejichž počet je explicitně určen v konfiguračním souboru společně s jeho velikostí a směrodatnou odchylkou, která slouží pro výpočet amplitudy dle vztahu (3.22). Pro zjednodušení jsou jednotlivé bodové zdroje světla rozloženy rovnoměrně v rámci čtvercové plochy. Je-li uživatelem zadána nulová velikost objemového bodu, pak je objemový bod degradován na bodový zdroj světla.

5.4 Implementace WRP metody

Implementaci modifikované WRP metody lze nalézt v souboru `wrp.c`. Počet WRP rovin v rámci metody je specifikován v konfiguračním souboru. Tímto počtem rovin je pak mrak zdrojů světla proložen tak, že všechny roviny jsou uvnitř mraku a mají od sebe konstantní vzdálenost.

Velikost oblasti utlumené před přičtením příspěvku zdroje světla, pro korekci viditelnosti, může uživatel explicitně určit. Nestane se-li tak, je velikost této oblasti vypočtena dle vztahu (3.21). Možnost volby této velikosti uživatelem je zde z důvodu, že vztah (3.21) předpokládá rovnoměrné rozložení zdrojů světla v rámci mraku, to však nemusí být vždy splněno.

Algoritmus předpokládá malou vzdálenost mezi WRP rovinami, proto je zde použita propagace rozkladem do úhlového spektra. Při propagaci na rovinu hologramu je na základě rovnice (3.15) nejprve zjištěno, zda je propagační jádro Rayleigh-Sommerfeldova integrálu aliasováno a pokud ne, použije se tato metoda propagace, jinak je použita propagace pomocí rozkladu do úhlového spektra.

Pro výpočet Fourierových transformací je použita knihovna **FFTW3**, která podporuje i vícejádrové zpracování. Počet jader, které budou využity při výpočtu rychlé Fourierovy transformace je specifikovatelný v konfiguračním souboru.

5.5 Numerická rekonstrukce hologramu

Numerická rekonstrukce hologramu (soubor `reconstruction.c`) je řešena simulací tenké čočky. Vzdálenosti čočky od roviny hologramu a bodu, do kterého čočka zaostřuje jsou nastaveny v konfiguračním souboru.

5.6 Vizualizace struktury optického pole

Vizualizace dat je implementována v souboru `hologram.c`, je řešena zobrazením absolutní hodnoty optického pole do obrázku ve formátu PNG. K tomu je použita knihovna `LodePNG`, kterou jsem zvolil kvůli jednoduchosti použití.

6 Dosažené výsledky

V této kapitole jsou uvedeny výsledky srovnání algoritmu s naivní implementací. Dále je zde také demonstrace efektů jednotlivých metod, implementovaných v rámci práce, na netriviálních datech.

6.1 Srovnání s referenční metodou

Srovnání algoritmu s referenční metodou probíhalo na mraku bodových zdrojů světla vygenerovaných uvnitř koule o poloměru 0.5 mm. Počítač, na kterém testy probíhaly má procesor Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz a jako operační systém byl použit Debian Jessie GNU/Linux 64bit. Během testů byla u knihovny FFTW3 vypnuta podpora vícejádrového zpracování. Tabulka (6.1) udává konfiguraci, se kterou byly srovnávací testy spouštěny.

Tabulka 6.1: Parametry výpočtu v rámci srovnání implementací

Název	hodnota
Rozlišení hologramu	1024x768 pixelů
Vzorkovací vzdálenost	9 μm
Vlnová délka	633 nm
Vzdálenost objektu od roviny hologramu	0.5 m

V následující tabulce jsou uvedeny časy běhů obou implementací. Není zde zahrnuta vizualizace výsledku, ani numerická rekonstrukce. V rámci implementovaného algoritmu byla vždy použita jedna WRP rovina.

Tabulka 6.2: Srovnání časů běhu obou implementací

Počet bodů v mraku	Impl. algoritmus	Zpracování jednoho bodu	Naivní impl.	Zpracování jednoho bodu
10	0.978 s	0.0978 s	0.930 s	0.0930 s
100	1.002 s	0.0100 s	8.799 s	0.0879 s
1000	1.087 s	0.00108 s	87.753 s	0.0875 s
10000	1.762 s	0.000176 s	894.124 s	0.0894 s

Z tabulky (6.2) je jasně vidět, že algoritmus implementovaný v rámci této práce je v naprosté většině případů efektivnější, než naivní implementace. Na triviálních datech však naivní implementace dosahuje lepších časů běhu.

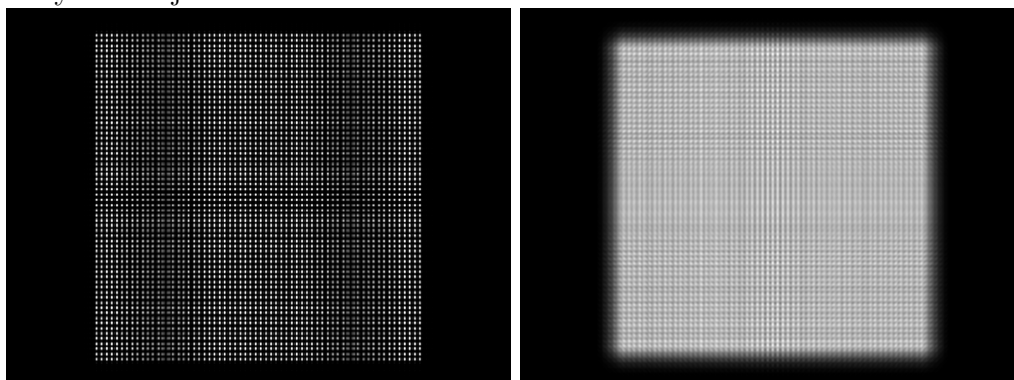
6.2 Testování algoritmu

Implementovaný algoritmus byl také testován na netriviálních datech, kde byl zdrojový objekt tvořen jednotkami, až stovkami tisíci zdrojů světla. Na těchto datech byly také zkoumány výsledky použití objemových zdrojů světla, či efekt metody pro řešení viditelnosti jednotlivých zdrojů světla. U jednotlivých pokusů jsou vždy uvedeny konfigurace, se kterými byly hologramy generovány.

6.2.1 Objemové body

Testování aplikace objemových bodů probíhalo na mraku zdrojů světla, které byly rovnoměrně vygenerované uvnitř plochy čtverce o velikosti hrany 6 mm a vzdálenosti od roviny hologramu 0.5 m. Počet zdrojů světla byl 4096, rozlišení hologramu 1024×768 a vzorkovací vzdálenost $9 \mu\text{m}$. Následující obrázek ukazuje numerické rekonstrukce vytvořené za účelem srovnání použití bodových a objemových zdrojů světla a tedy vytvoření dojmu hladkého povrchu.

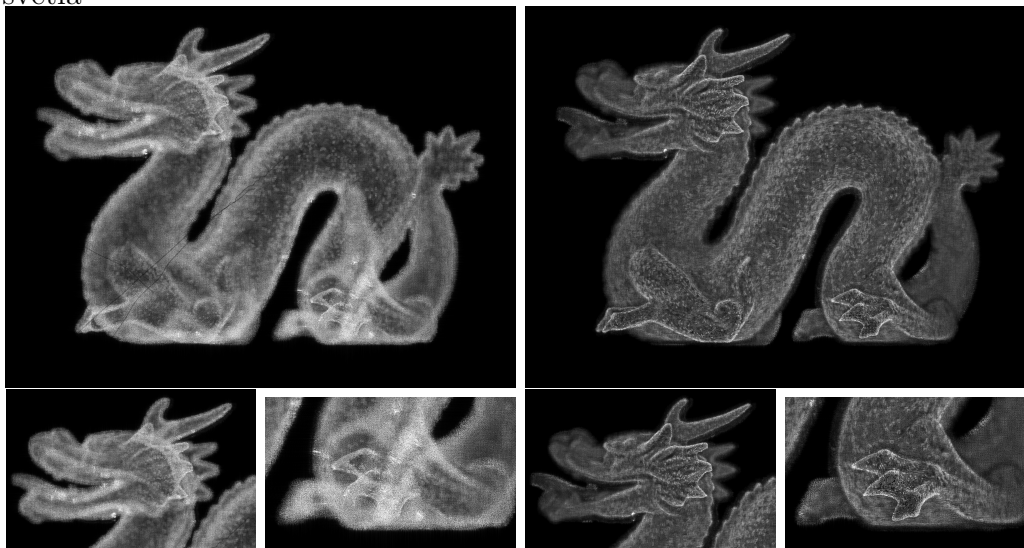
Obrázek 6.1: Srovnání bezrozměrných a objemových bodů. Rekonstrukce vlevo je za použití bodových zdrojů světla, rekonstrukce vpravo pak objemových zdrojů světla.



6.2.2 Řešení viditelnosti

Následující obrázek ukazuje numerické rekonstrukce hologramu, přičemž na sadě obrázků vlevo je rekonstrukce hologramu bez zapnuté metody řešení viditelnosti, obrázky vpravo jsou rekonstrukce hologramu, kde byl tento mechanismus zapnutý. Zdrojový objekt [4] byl tvořen 437645 zdroji světla, rozlišení hologramu bylo 6144×6144 pixelů a hodnota vzorkovací vzdálenosti byla zvolena $7 \mu\text{m}$. Vygenerování hologramu zabralo přibližně 97 sekund, přičemž byly použity dvě WRP roviny.

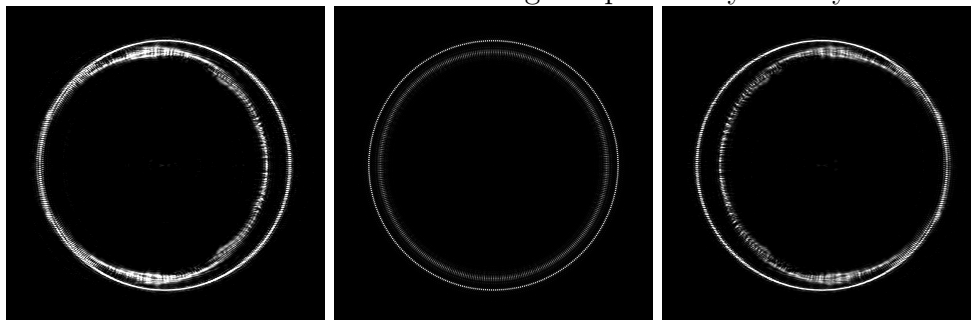
Obrázek 6.2: Demonstrace efektu řešení viditelnosti, levá sada je bez řešení viditelnosti, u obrázků vpravo je brána v potaz viditelnost jednotlivých zdrojů světla



Rekonstrukce pohledu pod úhlem

Pro ověření výsledků byla také implementována podpora numerické rekonstrukce toho, co uvidíme při pohledu na hologram pod nenulovým úhlem. Obrázek na následující straně ukazuje srovnání pohledů na stejný hologram pod úhly 0° a 1° z levé a pravé strany. Obrázek vlevo tedy odpovídá pohledu z levé strany, obrázek uprostřed je pohled přímo a obrázek vpravo je pohled z pravé strany. Objekt tvořily dvě kružnice o poloměru 6 mm ve vzdálenostech 0.5 m a 0.6 m . Rozlišení hologramu bylo 3072×3072 pixelů a vzorkovací vzdálenost byla $9 \mu\text{m}$.

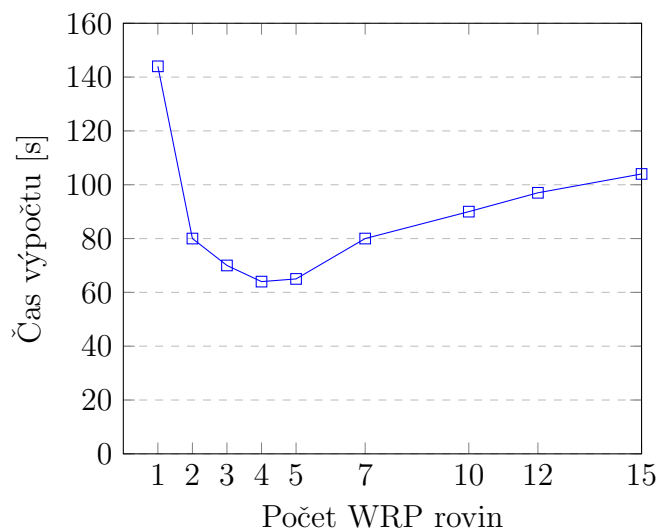
Obrázek 6.3: Pohled na hologram pod různými úhly



6.2.3 Modifikovaná WRP metoda

V rámci testování, jak moc změna počtu WRP rovin zefektivní výpočet struktury hologramu, byl vytvořen mrak 500000 bodových zdrojů světla, přičemž šířka a výška mraku byly 6 mm, hloubka pak 40 mm. Rozlišení hologramu bylo 3072×3072 pixelů a vzorkovací vzdálenost byla $7 \mu\text{m}$.

Čas výpočtu struktury hologramu v závislosti na počtu WRP rovin



Z předchozího grafu je jasně vidět, že pro počet WRP rovin existuje optimální hodnota. Průběh grafu taktéž odpovídá odvozenému vztahu pro celkovou složitost modifikované WRP metody, který je zmíněn na konci sekce (3.4.3). Počet WRP rovin je tedy nutno volit na základě konkrétních parametrů hologramu a podobě mraku zdrojů světla.

7 Závěr

Počítačem generovaná holografie je výpočetně náročná úloha, pro kterou existuje mnoho metod, které výpočet urychlují. V rámci této bakalářské práce bylo popsáno několik z nich. Zároveň byl také implementován algoritmus pro výpočet digitálních hologramů, který tyto metody implementuje. Algoritmus byl srovnán s naivní implementací, kde se ukázalo, že přináší efektivní výpočet.

Další práce na projektu by rozhodně zahrnovala paralelizaci výpočtu vyhledávací tabulky a propagačních jader. Dále bych algoritmus upravil tak, aby bylo možné výsledný hologram vypočítat po částech. Díky tomu by pak algoritmus bylo možné například předělat do distribuované podoby.

Díky této bakalářské práci jsem nabyl nových znalostí z Fourierovské optiky a seznámil se s principy holografie, se kterou jsem do té doby neměl vůbec žádné zkušenosti.

Literatura

- [1] *Computer generated holography at the University of West Bohemia* [online]. University of West Bohemia, 2016. [cit. 2016-28-04]. Computer generated holography research group. Dostupné z: <http://holo.zcu.cz/>.
- [2] MATSUSHIMA, K. – SHIMOBABA, T. Band-Limited Angular Spectrum Method for Numerical Simulation of Free-Space Propagation in Far and Near Fields. *Opt. Express*. Říjen 2009, 17, 22, s. 19662–19673. doi: 10.1364/OE.17.019662. Dostupné z: <http://www.opticsexpress.org/abstract.cfm?URI=oe-17-22-19662>.
- [3] ROUBAL, P. *Informatika a výpočetní technika pro střední školy*. Brno : Computer Press, a.s., 2012. ISBN 978-80-251-3228-9.
- [4] Stanford Scanning Repository. The Stanford 3D Scanning Repository. <http://graphics.stanford.edu/data/3Dscanrep/>.
- [5] SYMEONIDOU, A. et al. Computer-generated holograms by multiple wavefront recording plane method with occlusion culling. *Opt. Express [online]*. Srpen 2015, 23, 17, s. 22149–22161. doi: 10.1364/OE.23.022149. Dostupné z: <http://www.opticsexpress.org/abstract.cfm?URI=oe-23-17-22149>.
- [6] VOELZ, D. G. *Computational Fourier Optics: A MATLAB Tutorial*. SPIE Press. Bellingham : Society of Photo Optical, 2011. ISBN 978-0-8194-8204-4.

Seznam zkratek

- CPU** Central Processing Unit - hlavní výpočetní jednotka počítače
- GPU** Graphics Processing Unit - grafická výpočetní jednotka počítače
- FFT** Fast Fourier Transform - algoritmus pro výpočet Fourierovy transformace
- FFTW** Fastest Fourier Transform in the West - knihovna pro výpočet FFT
- WRP** Wavefront Recording Plane - metoda pro výpočet digitálních hologramů

Seznam obrázků

3.1	Ukázka podvzorkovaného fázoru(viz okraje)	7
3.2	Fázor (3.2.1) s omezením na korektně navzorkovanou oblast .	10
3.3	Princip WRP metody	14
3.4	Modifikovaná WRP metoda	16
3.5	Porovnání bodového a objemového zdroje světla, obrázek vlevo ukazuje bodový zdroj světla, obrázek vpravo pak objemový zdroj světla. Reálná velikost obou obrázků je 0.18×0.18 mm. .	17
4.1	Dopředná a zpětná propagace	18
4.2	Tenká čočka	19
6.1	Srovnání bezrozměrných a objemových bodů. Rekonstrukce vlevo je za použití bodových zdrojů světla, rekonstrukce vpravo pak objemových zdrojů světla.	25
6.2	Demonstrace efektu řešení viditelnosti, levá sada je bez řešení viditelnosti, u obrázků v pravo je brána v potaz viditelnost jednotlivých zdrojů světla	26
6.3	Pohled na hologram pod různými úhly	27