



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

## FOTOGRAFOVÁNÍ S VYUŽITÍM SVĚTELNÉHO POLE

LIGHT FIELD PHOTOGRAPHY

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Karel Svoboda

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Slanina, Ph.D.

BRNO 2016



# Diplomová práce

magisterský navazující studijní obor Elektronika a sdělovací technika

Ústav radioelektroniky

**Student:** Bc. Karel Svoboda

**ID:** 138682

**Ročník:** 2

**Akademický rok:** 2015/16

## NÁZEV TÉMATU:

### Fotografování s využitím světelného pole

#### POKYNY PRO VYPRACOVÁNÍ:

Prostudujte princip využití tzv. plenoptického snímání pro fotografování s využitím světelného pole. Vypracujte rešerši relevantní literatury a vytvořte interaktivní aplikaci, která bude názorně demonstrovat princip plenoptického snímání.

Vytvořte program, který bude umožňovat zobrazení surových snímků z plenoptického fotoaparátu Lytro a renderování obrázků s výběrem roviny zaostření. Pro realizaci programu je doporučeno prostředí Matlab.

#### DOPORUČENÁ LITERATURA:

[1] GEORGIEV, T., LUMDAINE, A. Focused Plenoptic Camera and Rendering. [Online] [cit. 13. května 2015] Dostupné na WWW: <http://www.tgeorgiev.net/rendering.pdf>

[2] DANSEREAU, D. Light Field Toolbox v 0.4 [Online] [cit. 13. května 2015]. Dostupné na www: <http://www.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4>

**Termín zadání:** 8.2.2016

**Termín odevzdání:** 19.5.2016

**Vedoucí práce:** Ing. Martin Slanina, Ph.D.

**Konzultant diplomové práce:**

**doc. Ing. Tomáš Kratochvíl, Ph.D., předseda oborové rady**

#### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do dílech autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

## **ABSTRAKT**

Tato práce si klade za cíl objasnit čtenáři pojmy jako světelné pole, plenoptický fotoaparát nebo digitální čočka. Dále vysvětlit princip vykreslování snímků s možností výběru roviny zaostření, hloubky ostrosti, změny perspektivy a částečnou změnou úhlu pohledu. Hlavními výstupy práce jsou skripty pro renderování snímků z fotoaparátu Lytro a interaktivní aplikace, která názorně demonstuje principy plenoptického snímání.

## **KLÍČOVÁ SLOVA**

Světelné pole, plenoptický fotoaparát, kamerové pole, digitální objektiv, 4D snímek, přeostrění, Lytro.

## **ABSTRACT**

The aim of this thesis is to explain terms like light field, plenoptic camera or digital lens. Also the goal is to explain the principle of rendering the resulting images with the option to select the plane of focus, depth of field, changes in perspective and partial change in the angle of the point of view. The main outputs of this thesis are scripts for rendering images from Lytro camera and the interactive application, which clearly demonstrates the principles of plenoptic sensing.

## **KEYWORDS**

Light field, plenoptic camera, camera array, digital optic, 4D image, refocusing, Lytro.

SVOBODA, K. *Fotografování s využitím světelného pole*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2015. 36 s., 5 s. příloh. Diplomová práce. Vedoucí práce: Ing. Martin Slanina, Ph.D.

# PROHLÁŠENÍ

Prohlašuji, že svoji diplomovou práci na téma Fotografování s využitím světelného pole jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

.....

(podpis autora)

# PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Martinu Slaninovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

Experimentální část této diplomové práce byla realizována na výzkumné infrastruktuře  
vybudované v rámci projektu CZ.1.05/2.1.00/03.0072

**Centrum senzorických, informačních a komunikačních systémů (SIX)**  
operačního programu Výzkum a vývoj pro inovace.

# OBSAH

<b>Seznam obrázků</b>	<b>ix</b>
<b>Úvod</b>	<b>1</b>
<b>1 Světelné pole</b>	<b>2</b>
<b>2 Plenoptické snímání</b>	<b>5</b>
2.1 Nedostatky klasického fotoaparátu .....	5
2.2 Konstrukce plenoptického fotoaparátu .....	6
2.3 Možnosti plenoptického fotoaparátu .....	8
2.3.1 Změna úhlu pohledu a 3D fotografie .....	8
2.3.2 Změna zaostření .....	11
2.3.3 Změna hloubky ostrosti .....	13
2.3.4 Změna perspektivy .....	14
<b>3 Interaktivní aplikace</b>	<b>18</b>
<b>4 Renderovací program</b>	<b>22</b>
4.1 Postup získání surových dat .....	22
4.2 Úprava RAW dat .....	24
4.3 Kalibrace .....	27
4.4 Renderování snímku z určitého úhlu pohledu .....	29
4.5 Renderování snímků s různou rovinou ostrosti .....	31
4.6 Renderování snímků s různou hloubkou ostrosti .....	33
4.7 Renderování snímků s různou perspektivou .....	33
4.8 Časová náročnost .....	34
4.9 Porovnání s komerčním programem .....	35
<b>5 Závěr</b>	<b>36</b>
<b>Literatura</b>	<b>37</b>
<b>Seznam symbolů, veličin a zkratk</b>	<b>38</b>
<b>A Zdrojové kódy</b>	<b>39</b>
A.1 Úprava RAW dat .....	39

A.2	Vykreslení snímku z určitého úhlu .....	40
A.3	Funkce vykreslující světelné paprsky (interaktivní aplikace).....	41
<b>B</b>	<b>Návod k laboratorní úloze</b>	<b>42</b>



# SEZNAM OBRÁZKŮ

Obrázek 1.1 Popis světelného pole. ....	2
Obrázek 1.2 Soustava paprsků tvořící jeden pixel. ....	3
Obrázek 1.3 Vliv nedoostření. ....	3
Obrázek 1.4 Plenoptický fotoaparát. ....	4
Obrázek 2.1 Princip klasického fotoaparátu. ....	5
Obrázek 2.2 Přivření clony fotoaparátu. ....	6
Obrázek 2.3 Pole 96 kamer Stanford [6]. ....	6
Obrázek 2.4 Pelican Imaging [7]. ....	7
Obrázek 2.5 Lytro první komerčně dostupný plenoptický fotoaparát [8]. ....	7
Obrázek 2.6 Mikročočky fotoaparátu Lytro. ....	8
Obrázek 2.7 Vykreslování snímku z levé strany mikročoček. ....	9
Obrázek 2.8 Skládání snímku. ....	9
Obrázek 2.9 Vliv překročení ohniskové vzdálenosti (vlevo před, vpravo za ohniskovou vzdáleností). ....	9
Obrázek 2.10 Vliv zrcadlení na výslednou fotografii (vpravo správně odzrcadleno). ...	10
Obrázek 2.11 Stereoskopický snímek. ....	11
Obrázek 2.12 Vytvoření matice snímků. ....	12
Obrázek 2.13 Přeastření pomocí posunutí jednotlivých snímků. ....	12
Obrázek 2.14 Různé roviny ostrosti. ....	13
Obrázek 2.15 Detail výsledného obrázku. ....	13
Obrázek 2.16 Přivření digitální clony. ....	14
Obrázek 2.17 Změna hloubky ostrosti. ....	14
Obrázek 2.18 Výběr pixelu dle místa průchodu paprsku hlavní čočkou. ....	15
Obrázek 2.19 Ortografická fotografie. ....	15
Obrázek 2.20 Fotografie s perspektivou. ....	16
Obrázek 2.21 fotografie s širokým úhlem pohledu. ....	16
Obrázek 2.22 Rozšíření optiky. ....	17
Obrázek 2.23 Výsledné snímky s různou perspektivou. ....	17
Obrázek 3.1 Vzhled interaktivní aplikace. ....	18
Obrázek 3.2 Změna polohy objektů v prostoru. ....	19

Obrázek 3.3 Změna polohy hlavní čočky. ....	19
Obrázek 3.4 Změna polohy pole mikročoček. ....	20
Obrázek 3.5 Změna ohniskové vzdálenosti hlavní čočky. ....	20
Obrázek 3.6 Změna ohniskové vzdálenosti mikročoček. ....	21
Obrázek 4.1 Importované soubory do PC. ....	22
Obrázek 4.2 Stupně šedi. ....	24
Obrázek 4.3 Barevný vzor. ....	25
Obrázek 4.4 Obarvení podle vzoru 'bggr'. ....	25
Obrázek 4.5 Vyvážení bílé. ....	26
Obrázek 4.6 Převod na <i>sRGB</i> . ....	26
Obrázek 4.7 Korekce gradačního zkreslení. ....	27
Obrázek 4.8 Rozteč <i>X Y</i> . ....	27
Obrázek 4.9 Určení středů mikročoček bez rozšíření a s rozšířením matice. ....	28
Obrázek 4.10 Chyby kalibrace. ....	29
Obrázek 4.11 Velikost vybíraných bloků 4x4pix. ....	30
Obrázek 4.12 Bloky s poměrem 13/15. ....	30
Obrázek 4.13 Různé pohledy na scénu. ....	31
Obrázek 4.14 Vzor vybíraných pixelů. ....	31
Obrázek 4.15 Vliv počtu použitých snímků na kvalitu výsledného snímku. ....	32
Obrázek 4.16 Přeostrění – dosažená kvalita. ....	32
Obrázek 4.17 Změna hloubky ostrosti – dosažená kvalita. ....	33
Obrázek 4.18 Změna perspektivy – přehnaný posuv. ....	34
Obrázek 4.19 Procentuální vyjádření časové náročnosti jednotlivých kroků. ....	34
Obrázek 4.20 Porovnání kvality výsledných snímků. ....	35
Obrázek 4.21 Porovnání kvality – detail. ....	35

# ÚVOD

Již pravěcí lidé se ve svých jeskyních snažili zachytit scény z jejich životů, postačovala jim k tomu pouze jejich paměť, představivost a omezené nástroje (barva z drcených nerostů smíchaných s tuky nebo vodou a štětcí ze srsti zvířat [1]). Tato technika se postupem času vyvíjela a vyvíjí, a tak vznikla například výzdoba v Sixtinské kapli, nebo obrazy Leonarda Da Vinci. Během tohoto vývoje proběhlo mnoho důležitých technických vylepšení, například v použitých barvách, geometrii, perspektivě, avšak z hlediska zachycování reálné scény se jednalo pouze o evoluci nikoliv revoluci.

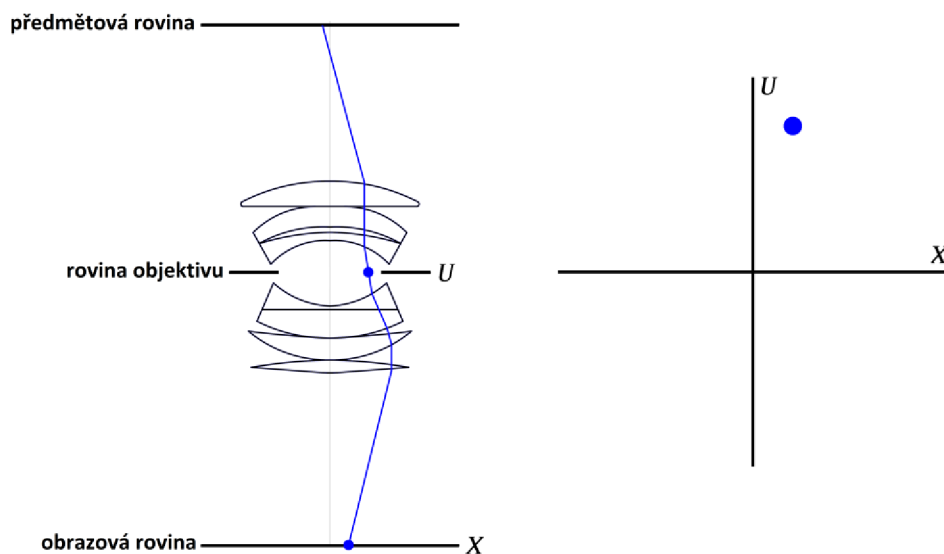
Revoluce v zachycování okolí přišla až s první fotografií roku 1826, jako záznamové médium posloužila vyleštěná cínová deska pokrytá petrolejovým roztokem a čas expozice byl 8 hodin [2]. Kvalita této fotografie se s tehdejšími obrazy nedala porovnat, avšak scéna již byla zachycena podle fyzikálních zákonů, nikoliv pomocí paměti a představivosti. I technika zachycování fotografií se obrovským způsobem vyvinula a dnešní fotoaparáty jsou neporovnatelné s těmi historickými. Po technické stránce došlo k obrovskému vývoji jak optiky, tak i záznamového média a tak dnes neudáváme délku expozice v hodinách, ale ve zlomcích sekundy. Je až neuvěřitelné, že po celou tuto dobu zůstává princip pořizování fotografie stejný, přicházející světlo se pomocí soustavy optických čoček nasměruje s určitou rovinou ostrosti na světlocitné médium, kam se nechá určitou dobu dopadat. Výsledná fotografie je potom jakýsi plošný výřez reálné scény, který vznikl součtem paprsků, které dopadly do jednotlivých bodů záznamového média. Tímto způsobem záznamu se ovšem spousta informací o scéně, které projdou objektivem, ztratí.

Plenoptické fotoaparáty jsou z hlediska informací, které o scéně zaznamenávají zcela revoluční [3]. Kromě intenzity zachycují i směr, kterým jednotlivý paprsek do fotoaparátu přišel. Pomocí těchto informací a příslušného softwaru je možné pořízenou fotografii libovolně přeostřit, změnit hloubku ostrosti a v určité míře měnit i úhel pohledu na scénu. Vývoj plenoptických fotoaparátů je nyní na samém počátku, ale již dnes existují komerčně dostupné modely, které pořizují fotografie srovnatelné kvality jako klasické fotoaparáty.

# 1 SVĚTELNÉ POLE

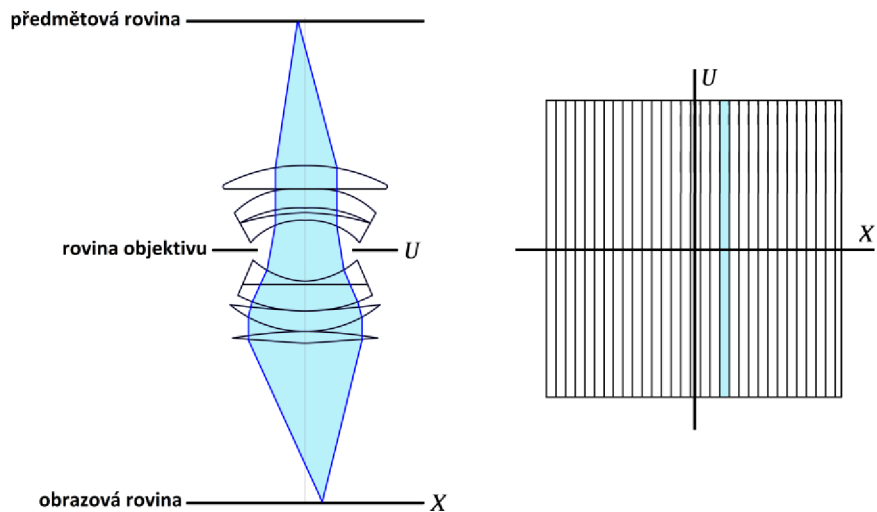
Koncept využití světelného pole při fotografování je takový, že při zmáčknutí spouště je zaznamenána prakticky kompletní scéna, kterou má objektiv před sebou. Z pořízených dat je poté možné vytvořit jakýkoliv snímek, jakoby byla před vámi reálná scéna. Toto je umožněno díky zaznamenání veškerých informací, které projdou objektivem. Včetně informace, odkud jednotlivé paprsky přišly a kam na hlavní čočku fotoaparátu dopadly.

K ilustraci tohoto principu slouží přehledný 2D diagram na obrázku 1.1 vlevo, jednotlivé paprsky jsou zde popsány směrovou souřadnicí  $u$ , kde procházejí rovinu objektivu, a prostorovou souřadnicí  $x$ . Toto zobrazení lze převést do kartézského souřadného systému, viz obrázek 1.1 vpravo, kde je možné jakýkoliv paprsek procházející fotoaparátem zobrazit jako bod se souřadnicemi  $(x, u)$ . Při uvážení reálné 3D scény můžeme libovolný paprsek popsat dvěma směrovými souřadnicemi  $(u, v)$  a dvěma prostorovými souřadnicemi  $(x, y)$ . Celkově nám vznikne 4D plenoptická funkce  $(x, y, u, v)$  [4], kterou lze při uvážení poloprůhledných předmětů ještě rozšířit o rozměr vzdálenosti, ze které paprsek přichází na 5D plenoptickou funkci [5].



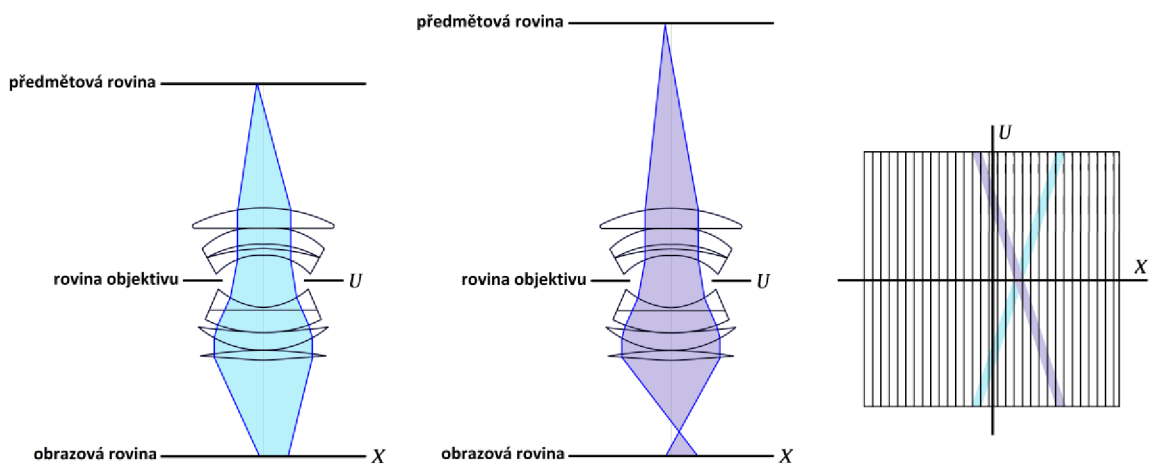
Obrázek 1.1 Popis světelného pole.

Při fotografování klasickým fotoaparátem se zaznamenají pouze 2D data. Obrázek 1.2 ukazuje bod v prostoru, který je přesně zaostřen. Takovýto bod se v kartézském souřadném systému zobrazí jako jeden svislý proužek. Pro všechny směry  $u$  připadá jedno  $x$  v digitální fotografii například jeden pixel [4].



Obrázek 1.2 Soustava paprsků tvořící jeden pixel.

Jednotlivé pruhy představující pixely zabírají celý rozsah  $u$  a jejich počet udává rozlišení fotoaparátu. Obrázek 1.3 ukazuje případ, kdy jsou body v prostoru mimo rovinu zaostření. Pokud se bod v prostoru pohybuje od roviny zaostření směrem k fotoaparátu, bude se pruh reprezentující paprsky přicházející z tohoto bodu stáčet po směru hodinových ručiček. Naopak pokud je bod ve větší vzdálenosti, než je rovina ostrosti, bude pruh vlivem zrcadlení natočen proti směru hodinových ručiček. Je zřejmé, že jednomu bodu v prostoru odpovídá několik souřadnic  $x$ , to v klasické fotografii znamená, že je bod rozostřen.



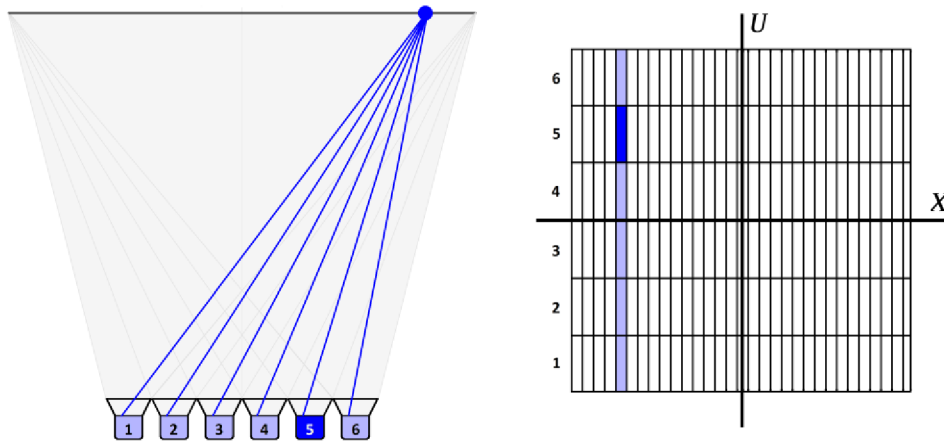
Obrázek 1.3 Vliv nedoostření.

Při fotografování 3D scény klasickým fotoaparátem se výsledný snímek vytvoří podle vztahu [4]

$$E_F(x, y) = \frac{1}{F^2} \iint L_F(x, y, u, v) du dv, \quad (1.1)$$

kde  $E_F$  představuje výsledný snímek,  $F$  vzdálenost roviny objektivu od obrazové roviny a  $L_F$  4D plenoptickou funkci.

Úlohou plenoptických fotoaparátů je zajistit co možná nejjemnější rozlišení v  $u$  a  $v$ , při zachování co nejvyššího rozlišení  $x$  a  $y$ . Obrázek 1.4 vlevo zobrazuje 2D pole 6 kamer. Všechny kamery mají shodné zorné pole, ale liší se úhlem pohledu na scénu. Díky tomu je možné vybírat pro konkrétní  $x$  ze šesti různých pixelů v ose  $u$ , například zvýrazněný pixel pocházející z kamery 5. Takovéto rozlišení může být dostačující, ale je nutné si uvědomit, že pro reálnou 3D scénu a stejné rozlišení je nutné použít  $6^2$  kamer, což už je značné množství.



Obrázek 1.4 Plenoptický fotoaparát.

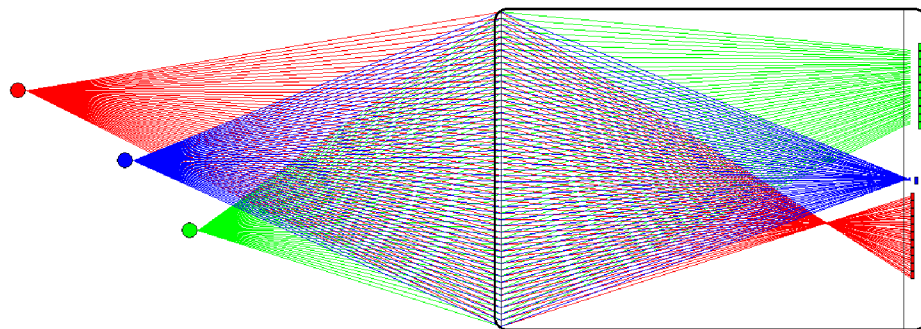
Alternativou ke kamerovému poli je použití fotoaparátu vybaveného polem mikročoček, položených těsně nad fotosenzor. Tyto mikročočky nasměrují světelné paprsky na senzor takovým způsobem, že je dodatečně možné rozlišit, ze kterého směru přišly. Nevýhodou tohoto přístupu je, že se pixely na jednom fotosenzoru dělí mezi rozměry  $x$  a  $y$  [4]. Z tohoto důvodu je rozlišení výsledných snímků pořízených tímto druhem kamer výrazně nižší, než u snímků pořízených klasickým fotoaparátem. Plenoptickým fotoaparátům se blíže věnuje kapitola 2.

## 2 PLENOPTICKÉ SNÍMÁNÍ

Název plenoptický fotoaparát vychází z anglického „plenty of optic“ doslova přeloženo jako mnoho optický. Tyto fotoaparáty se vyznačují tím, že zaznamenávají nejen intenzitu světla, ale i směr, odkud světelné paprsky přišly. Pro renderování výsledných snímků je zapotřebí výkonný počítač a příslušný software.

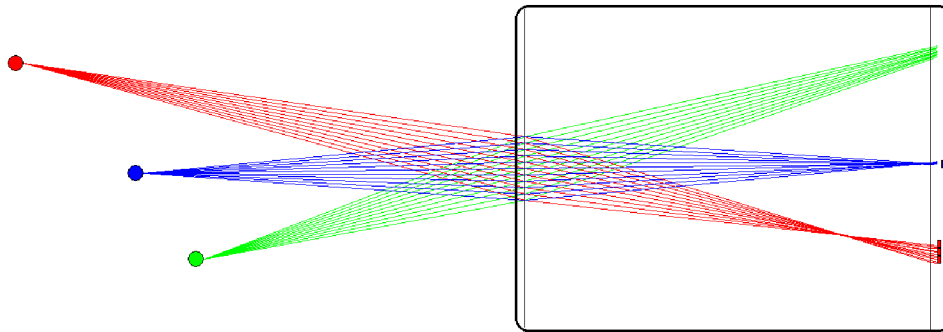
### 2.1 Nedostatky klasického fotoaparátu

Klasický fotoaparát prošel dlouhým vývojem a dnešní modely dosahují výborné obrazové kvality. Trpí však některými nedokonalostmi, které vycházejí klasického pojetí konstrukce. Například špatně zaostřenou fotografii nelze dodatečně přeastřit, nevhodně zvolenou hloubku ostrosti není možné změnit. Tyto vlastnosti jsou zcela přirozené, neboť při pořízení snímku se prostorové informace o scéně nezachytí. Obrázek 2.1 v jednoduchosti ukazuje princip klasického fotoaparátu. V levé části se nacházejí 3 body v prostoru před fotoaparátem. Optická soustava fotoaparátu je pro názornost reprezentována pouze jedinou čočkou a záznamovým senzorem v pravé části. Z obrázku je patrné, že fotoaparát je zaostřen na vzdálenost modrého bodu, ten se na senzoru promítl jako jediný bod. Dále je vidět, že zbylé dva body se na senzoru promítly do určité oblasti, velikost této oblasti je přímo úměrná vzdálenosti od roviny zaostření. Tato skutečnost způsobuje, že jsou objekty mimo rovinu zaostření na fotografii rozmazané.



Obrázek 2.1 Princip klasického fotoaparátu.

Toto rozmazání lze z velké části omezit, pokud zmenšíme otvor, kterým do fotoaparátu přichází světlo, tzv. clonu. Tím se nám rozšíří hloubka ostrosti, která udává rozsah vzdáleností od fotoaparátu, v němž se jeví snímané objekty jako ostré. Výsledek je znázorněn na obrázku 2.2. Je patrné, že se body na senzoru při stejném rozestavení promítly do menších oblastí.



Obrázek 2.2 Přivření clony fotoaparátu.

Tímto způsobem lze zajistit, že bude celá scéna téměř ostrá. Bohužel má tato technika praktické nevýhody, při přivření clony dopadá na senzor méně světla a fotografie se stává tmavší. Pro stejnou světlost fotografie je nutné prodloužit dobu expozice, nebo zvýšit citlivost senzoru. Oba způsoby však mají negativní dopad na obrazovou kvalitu pořízené fotografie. Při prodloužení doby expozice bude fotografie náchylnější na rozmazání vlivem pohybu fotoaparátu a pohybu na scéně. Při zvýšení citlivosti senzoru se na fotografii více objeví šum. V praxi se pro dosažení dostatečně exponované fotografie při daném nastavení clony a množství dopadajícího světla využívá kombinace obou přístupů - tedy práce s dobou expozice a nastavení citlivosti snímače.

## 2.2 Konstrukce plenoptického fotoaparátu

Jak se vyvíjely klasické fotoaparáty, tak i plenoptické fotoaparáty již prošly jistou evolucí. Za nejjednodušší možný plenoptický fotoaparát by se dala považovat obyčejná stereoskopická kamera, zaznamenává totiž celou scénu ze dvou směrů, a tak má informaci o směru odkud a kam paprsky přišly. Dalším krokem vpřed bylo zkonstruování pole kamer, které scénu zachycovaly z mnoha úhlů, příkladem může být pole 96 kamer zkonstruované na univerzitě ve Stanfordu, které je zachyceno na obrázku 2.3.



Obrázek 2.3 Pole 96 kamer Stanford [6].



Takovéto pole kamer se dá již považovat za plnohodnotný plenoptický fotoaparát, který scénu zachycuje pod mnoha úhly a má tak dobrou informaci o prostoru ve scéně. Toto řešení je však kvůli pořizovací ceně, fyzickým rozměrům a složité kalibraci vhodné pouze pro laboratorní účely.

Pořizování snímků pomocí pole kamer má jistě svoji budoucnost, jak také dokazuje společnost Pelican Imaging, které se v roce 2013 velice úspěšně podařilo integrovat takovéto pole do mobilního zařízení (tabletu i chytrého telefonu). Na obrázku 2.4 vidíme, že je toto pole složeno pouze z 16 fotoaparátů a je opravdu velmi malé. Výrobce udává, že má každá z kamer rozlišení 1000x750 pixelů a výsledné snímky dosahují rozlišení 8 MP [7].



Obrázek 2.4 Pelican Imaging [7].

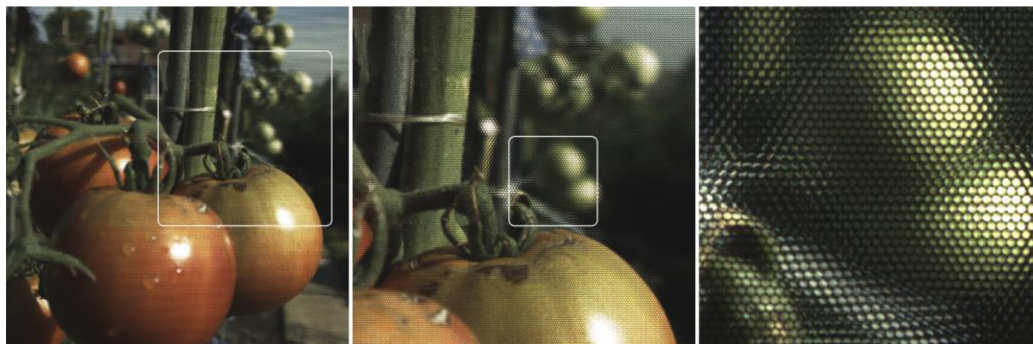
Obrovským krokem v před bylo v roce 2011 uvedení prvního komerčně dostupného plenoptického fotoaparátu Lytro. První model je zachycen na obrázku 2.5.



Obrázek 2.5 Lytro první komerčně dostupný plenoptický fotoaparát [8].

Toto zařízení, které má na délku cca 12cm, nahradilo objemné a drahé pole kamer. Lytro je vybaveno senzorem s rozlišením 3280x3280 pixelů, rozlišení výsledných snímků je 1080x1080 obrazových bodů. Konstrukcí tato kamera vychází z klasického digitálního fotoaparátu. Největším rozdílem je, že těsně před senzorem je vytvořeno pole cca 100 000 mikroskopických čoček, které překrývá celý senzor. Jednotlivé

mikročočky mají velikost cca 10x10 pixelů [8]. Na Obrázku 2.6 se nachází surová fotografie pořízená fotoaparátem Lytro, při dostatečném přiblížení jsou patrné jednotlivé čočky.



Obrázek 2.6 Mikročočky fotoaparátu Lytro.

Z obrázku je také patrné, že funkce mikročoček není totožná jako v případě pole kamer. V kamerovém poli zaznamenávají jednotlivé kamery celou scénu, namísto toho mikročočky zaznamenávají jen velmi malou část scény. Nedosáhne se tedy stejného výsledku při použití stejného počtu kamer a mikročoček.

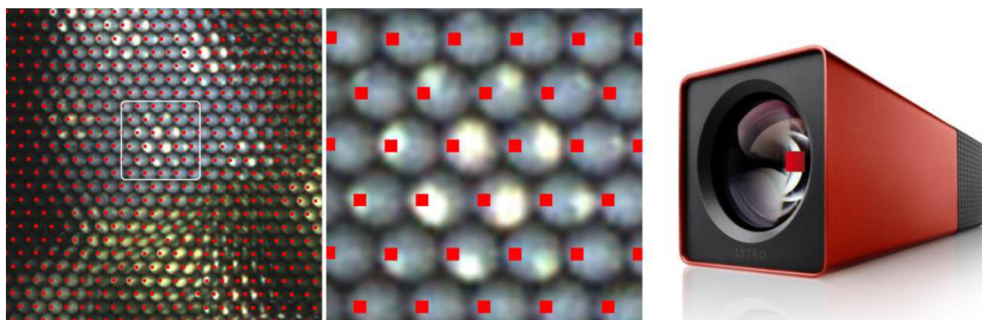
## 2.3 Možnosti plenoptického fotoaparátu

Jak již bylo zmíněno, plenoptický fotoaparát disponuje mnoha možnostmi, které klasický fotoaparát neumožňuje. V této kapitole bude, co možná nejnázorněji, vysvětlen princip vykreslování jednotlivých efektů ze surových snímků. Jako ilustrační obrázky jsou v kapitole přiřazeny snímky z fotoaparátu Lytro vykreslené pomocí skriptů v prostředí Matlab, použité principy renderování jsou však u všech plenoptických fotoaparátů obdobné.

### 2.3.1 Změna úhlu pohledu a 3D fotografie

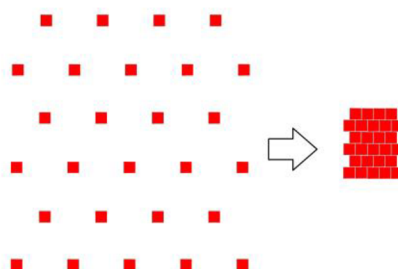
Při zpracovávání surového snímku je základním úkolem vykreslení jednotlivých snímků z různých úhlů, tím dostaneme stejně uspořádaná data, jakoby byla focena pomocí pole kamer. Z takto uspořádaných dat potom vychází i ostatní techniky zpracování, a také je další zpracování lépe pochopitelné.

Při vykreslování snímku z určitého úhlu se využije jen určitá část surových dat. Na obrázku 2.7 jsou znázorněny zvětšené výřezy surového snímku s mikročočkami. Červené čtverečky ukazují, které pixely jsou použity do výsledného snímku. V tomto případě jsou vybírány pixely z levé strany mikročoček a to ve výsledku znamená, že vykreslený snímek bude vypadat, jakoby byl vyfocen z levé strany fotoaparátu, jak je to naznačeno na obrázku 2.7 vpravo.



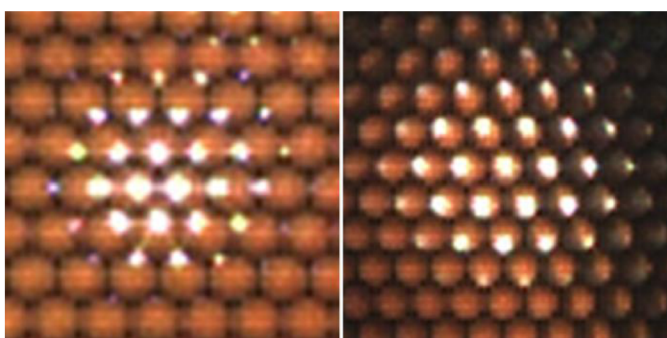
Obrázek 2.7 Vykreslování snímku z levé strany mikročoček.

Vybrané pixely se poskládají vedle sebe, jak je to znázorněno na obrázku 2.8. Je patrné, že rozlišení výsledné fotografie je mnohonásobně nižší než rozlišení surového snímku, to je také největší nevýhodou plenoptického snímání [4].



Obrázek 2.8 Skládání snímku.

Problémem při skládání snímku je fakt, že pokud se objekt na scéně nachází za ohniskovou vzdáleností, mikročočky obraz horizontálně i vertikálně převrátí. U fotoaparátu Lytro činí tato vzdálenost cca 30cm. Na obrázku 2.9 jsou zobrazeny 2 objekty, na kterých jsou odlesky. Levý objekt se nachází před ohniskovou vzdáleností a tak mikročočky obraz nepřevracují. Části odlesku jsou soustředěny směrem ke středu odlesku. Naopak objekt na pravé straně obrázku se nachází za ohniskovou vzdáleností, to způsobí, že obraz je po částech odzrcadlený. Části odlesku jsou soustředěny směrem od středu.



Obrázek 2.9 Vliv překročení ohniskové vzdálenosti (vlevo před, vpravo za ohniskovou vzdáleností).

Díky této vlastnosti a faktu, že z pod každé mikročočky vybíráme více, než jeden pixel, vzniká otázka, zda při vykreslování pixely pod mikročočkou zrcadlit či nikoli. Pro demonstraci rozdílu kvality výsledného snímku poslouží obrázek 2.10. Objekty se nachází za ohniskovou vzdáleností, a tak je nutné obraz pod jednotlivými mikročočkami odzrcadlit. V levé části obrázku nebylo zrcadlení provedeno, a tak na sebe pixely z jednotlivých čoček nenavazují. V pravé části je díky zrcadlení kvalita obrazu výrazně vyšší.



Obrázek 2.10 Vliv zrcadlení na výslednou fotografii (vpravo správně odzrcadleno).

Problémem ovšem je, že na většině reálných scén se vyskytují zároveň objekty před i za ohniskovou vzdáleností. Tento problém je možné v některých případech zcela ignorovat, jelikož na kvalitu výsledného snímku má jen zanedbatelný vliv. Druhou možností je použití takzvaných hloubkových map, které určí, zda jednotlivé mikročočky zrcadlit či nikoli.

Pomocí těchto technik je možné téměř plynule měnit úhel pohledu na scénu. Možný rozhled je dán velikostí optiky (hlavní čočky) fotoaparátu. Fotoaparát Lytro má průměr hlavní čočky cca 28 mm což znamená, že levý a pravý snímek se liší v pohledu na scénu jen o 28 mm. Tento malý rozdíl ovšem postačuje pro většinu efektů požadovaných od plenoptického fotoaparátu. Na obrázku 2.11 je pomocí červeného a modrého filtru znázorněn rozdíl mezi levým a pravým snímek. Obrázek je sestaven tak, že objekty v popředí jsou z levé a pravé fotografie srovnány přesně přes sebe. Různý pohled na scénu způsobuje, že se poloha objektů v pozadí na obou fotografiích liší.

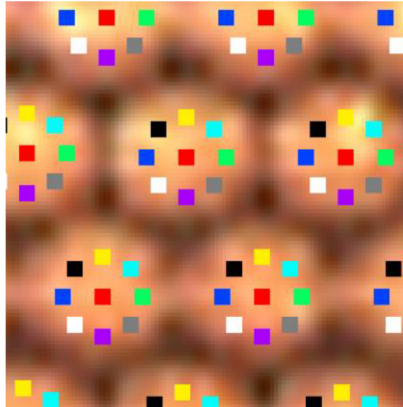


Obrázek 2.11 Stereoskopický snímek.

### 2.3.2 Změna zaostření

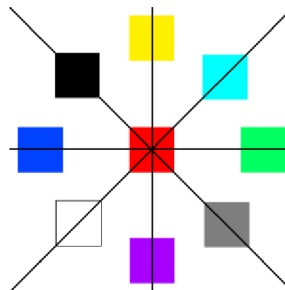
Zřejmě nejužitečnější vlastností plenoptického fotoaparátu je možnost měnit rovinu ostrosti po pořízení snímku. Existuje několik metod, jak ze surového snímku vytvořit výsledný snímek s konkrétní rovinou ostrosti. Nejnázornější používaná metoda se jmenuje „posuň a přičti“ (z anglického shift and add) [9]. Metoda vychází z dat uspořádaných, jakoby byly pořízeny polem kamer. Přičemž je nutné znát relativní polohu mezi jednotlivými kamerami. Dalším krokem je určení pomyslného středu pole kamer. Posledním krokem je sečtení fotografií z jednotlivých kamer posunutých o konstantu vynásobenou rozdílem polohy konkrétní kamery a pomyslného středu. Přičemž tato konstanta udává vzdálenost roviny ostrosti.

Při sestavování fotografie z fotoaparátu Lytro použijeme nejprve techniku vykreslení snímků z různých úhlů. Na obrázku 2.12 se nachází možný příklad vykreslení devíti snímků z různých úhlů pohledu. Pro vytvoření výsledného snímku se tedy použije devětkrát tolik pixelů, než bylo nutné pro vytvoření snímku z určitého úhlu pohledu, avšak výsledné rozlišení bude stejné, a to vlivem překrytí snímků přes sebe. Části jednotlivých snímků jsou na obrázku 2.12 barevně rozlišeny (snímek z konkrétního úhlu se složí z pixelů znázorněných stejnou barvou). Červený snímek se nachází uprostřed, a tak bude zvolen jako pomyslný střed.



Obrázek 2.12 Vytvoření matice snímků.

Přestření provedeme překrytím snímků z různých úhlů pohledu přes sebe. Černé linky na obrázku 2.13 znázorňují možný posuv jednotlivých snímků, přičemž je vždy nutné zachovat poměry vzdáleností jednotlivých snímků od středu. Jednotlivé posuvy realizují konkrétní vzdálenosti ostrosti. Konstantu určující rovinu zaostření zvolíme například  $K=0,5$ , pixely vybírané pro snímek označený černou barvou jsou od pomyslného středu posunuty v ose  $X$  o  $-7$  pixelů a v ose  $Y$  o  $7$  pixelů to znamená, že snímek označený černou barvou bude posunut vůči středovému snímku o  $\Delta X=0,5*(-7)=-3,5$  pixelu a  $\Delta Y=0,5*(7)=3,5$  pixelu



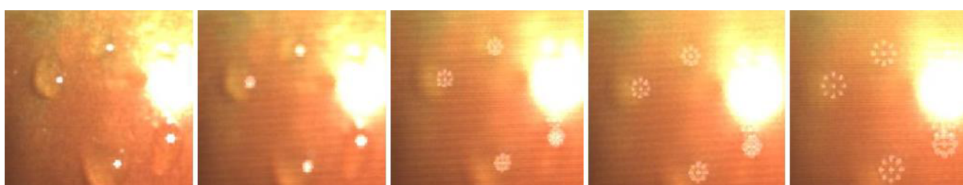
Obrázek 2.13 Přestření pomocí posunutí jednotlivých snímků.

Na obrázku 2.14 jsou zobrazeny tři výsledné snímky každý z různou rovinou ostrosti. Všechny tři snímky byly vytvořeny ze stejných pixelů surového snímku. Jediným rozdílem je vzájemné posunutí.



Obrázek 2.14 Různé roviny ostrosti.

Na Obrázku 2.15 se nachází detail odlesku pro různé roviny zaostření. Na tomto detailu je názorně vidět, jak posouvání snímku z různých úhlů funguje. Jelikož je odlesk velmi výrazný, na výsledných snímcích jsou patrné příspěvky jednotlivých snímků. Ideální situací by bylo, aby jednotlivé snímky vytvořily na výsledném snímku jednu souvislou plochu, na které nebude jednotlivé příspěvky možné nijak rozlišit.



Obrázek 2.15 Detail výsledného obrázku.

Toto je možné zaručit například omezením přestření, na obrázku 2.14 by vyhovovaly první dva snímky. Druhou možností je použít pro vykreslení výsledného snímku více zdrojových dat (více snímků z více úhlů).

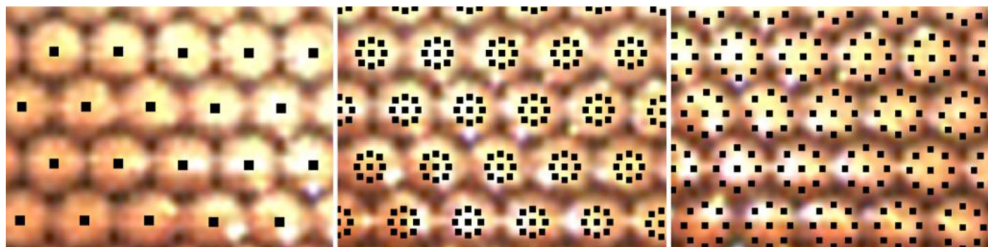
### 2.3.3 Změna hloubky ostrosti

Hloubka ostrosti u fotografie udává šířku prostoru kolem roviny zaostření, která se na fotografii jeví jako ostrá. I tuto vlastnost je možné po pořízení snímku plenoptickým fotoaparátem libovolně měnit. Při vytváření snímků s různou hloubkou ostrosti je nejprve nutné určit požadovanou rovinu ostrosti.

Na rozdíl od změny zaostření, kde všechny výsledné snímky byly vytvořeny ze stejných zdrojových dat a měnil se pouze posuv jednotlivých snímků přes sebe, tak při změně hloubky ostrosti zůstává tento posuv neměnný a mění se použitá zdrojová data.

Při vytváření snímků s různou hloubkou ostrosti je princip naprosto stejný, jako u klasického fotoaparátu, viz kapitola 2.1 obrázek 2.2. Prakticky jde o přivření „digitální clony“, na obrázku 2.16 je názorně vidět, jak přivírání digitální clony vypadá v praxi. Pokud jsou pro výsledný snímek vybírána pouze data zprostřed mikročoček, odpovídá to téměř uzavřené cloně obrázek 2.16 vlevo, naopak vpravo je clona úplně otevřená jelikož je plocha mikročočky využita až ke krajům. Při použití více zdrojových

pixelů by se zlepšila pouze kvalita fotografie, hloubka ostrosti by zůstala zachována. Mezi těmito krajními stavy je možné plynule přecházet a tak plynule měnit i hloubku ostrosti.



Obrázek 2.16 Přivření digitální clony.

Výsledné snímky zkomponované se stejnou vzdáleností ostrosti a ze zdrojových dat zvýrazněných na obrázku 2.16 se nacházejí na obrázku 2.17, pořadí snímků bylo zachováno.



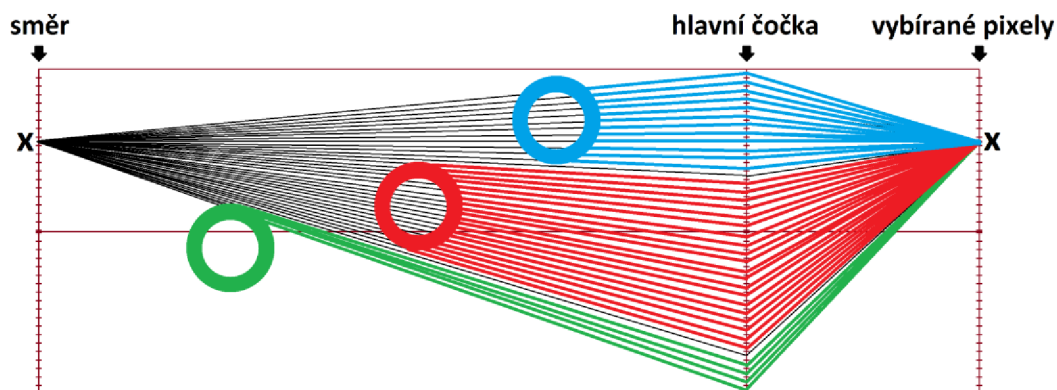
Obrázek 2.17 Změna hloubky ostrosti.

### 2.3.4 Změna perspektivy

Další schopností plenoptického fotoaparátu je změna perspektivy od téměř ortografické fotografie po fotografii s širokým úhlem pohledu. Rozsah změn perspektivy je závislý na velikosti optiky a zorném poli kamery.

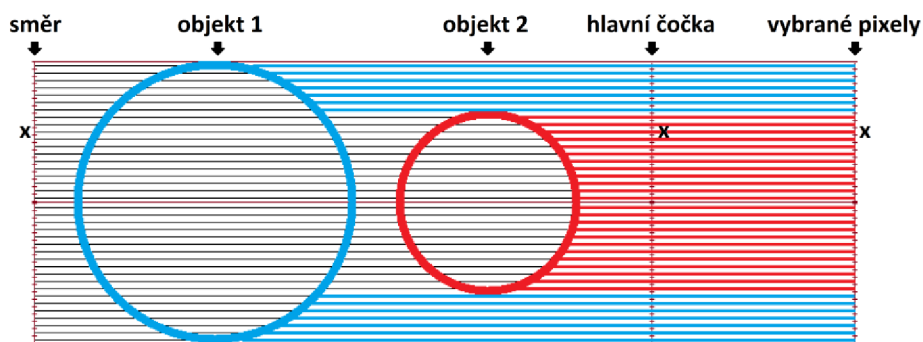
Při vytváření výsledných snímků si můžeme paprsky pro jednotlivé pixely vybírat podle toho, kterou částí hlavní čočky do fotoaparátu vstoupily. Situaci osvětluje obrázek 2.18, na pravé straně obrázku se nachází osa výsledného snímku, pro který je nutné vybrat příslušné pixely. Každý pixel výsledného snímku odpovídá určitému směru v reálné scéně (levá polovina obrázku 2.18). Při výběru pixelu  $x$  tedy vybíráme mezi paprsky, které přišly do hlavní čočky ze směru  $x$ . Na obrázku je vidět, že podle části hlavní čočky, ze které vybereme paprsky, bude na pixelu  $x$  zaznamenána část modrého, červeného nebo zeleného objektu. Takového rozhodnutí můžeme provést nezávisle pro každý jednotlivý pixel.





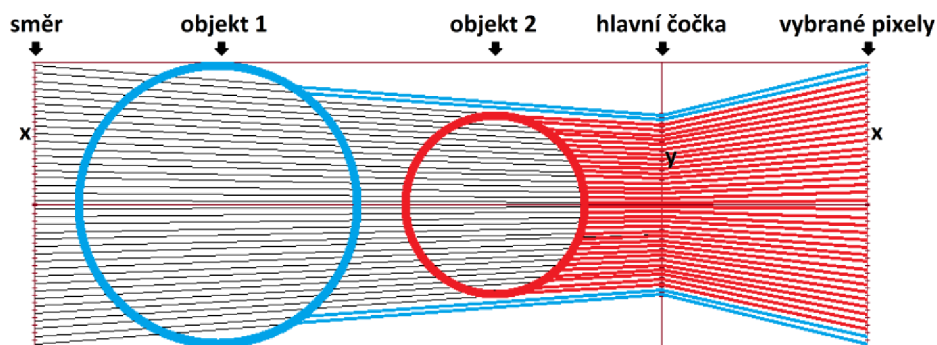
Obrázek 2.18 Výběr pixelu dle místa průchodu paprsku hlavní čočkou.

Na obrázku 2.19 je vyobrazen příklad ortografického snímání, kdy pixel  $x$  ze směru  $x$  je vybírán z bodu  $x$  na hlavní čočce. Takovéto uspořádání zaručí, že větší objekt 1 na výsledné fotografii není menším objektem 2 zakryt. Poměry velikostí objektu 1 a objektu 2 ve skutečnosti i na fotografii jsou zachovány bez ohledu na vzdálenosti jednotlivých objektů od hlavní čočky.



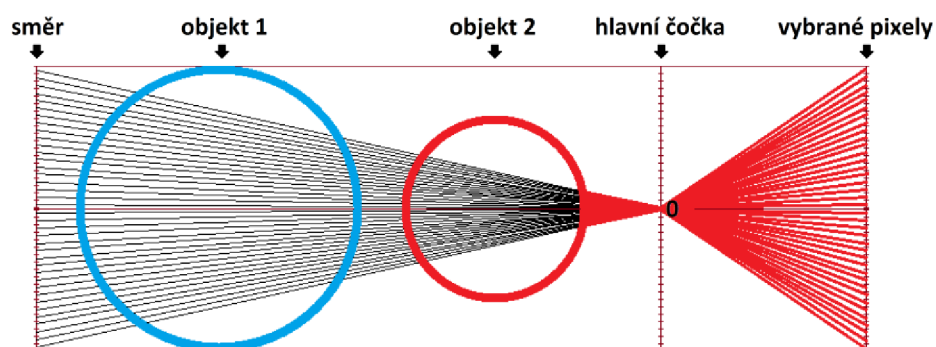
Obrázek 2.19 Ortografická fotografie.

Na obrázku 2.20 je vyobrazena stejná scéna jako na obrázku 2.19, rozdíl je v tom, že jsou pixely pro výslednou fotografii vybírány pouze z vnitřní části čočky. Například pixel  $x$ , který odpovídá směru  $x$  je vybrán z bodu  $y$  na hlavní čočce. Takovýto způsob vykreslování způsobí, že na výsledném snímku bude viditelná perspektiva. Menší objekt 2 téměř zakrývá větší objekt 1.



Obrázek 2.20 Fotografie s perspektivou.

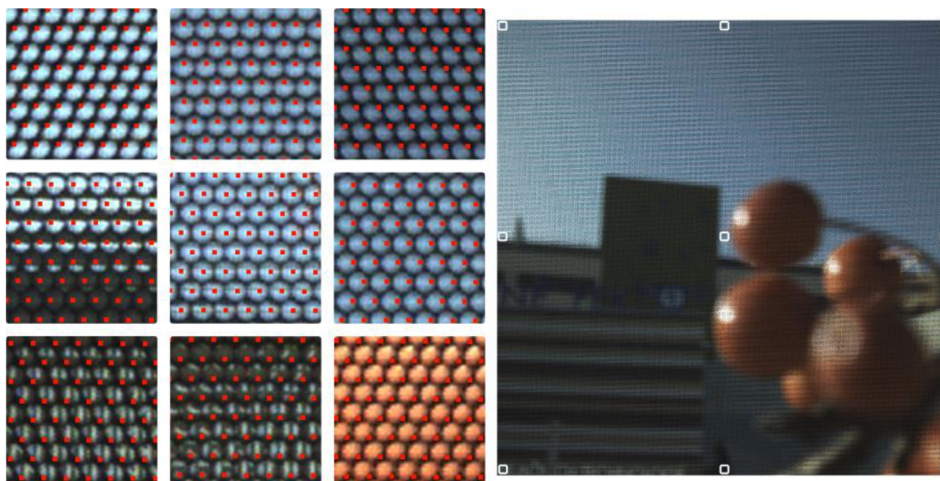
Mezní situaci zobrazuje obrázek 2.21, kde jsou veškeré pixely odpovídající svým konkrétním směrům vybírány pouze ze středu hlavní čočky (bod 0). Na výsledném snímku je vidět pouze část objektu 2, objekt 1 je zcela zakryt.



Obrázek 2.21 fotografie s širokým úhlem pohledu.

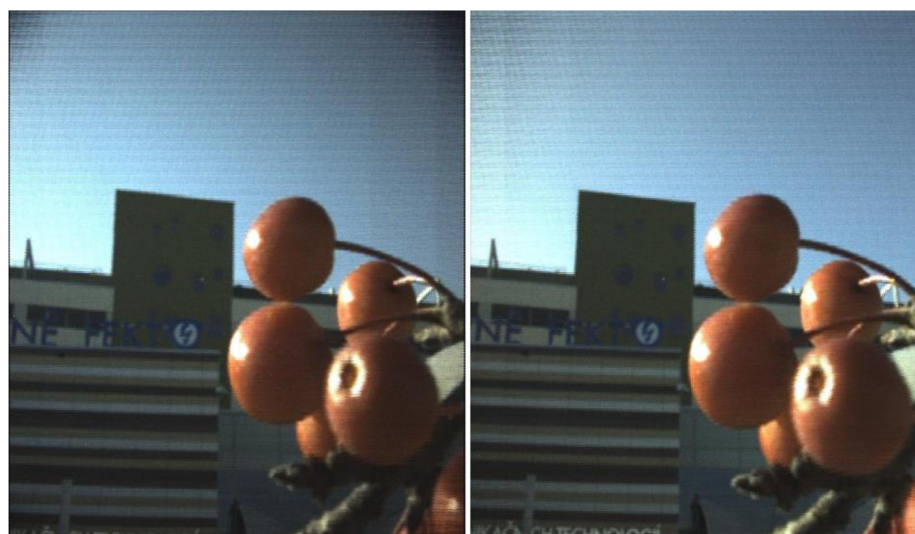
Na obrázcích 2.19-2.21 je zobrazena kamera, která má hlavní čočku srovnatelně velkou s objekty na scéně, a zabírá pouze prostor před sebou. Kamera Lytro má relativně malou optiku a má relativně velké zorné pole. Tyto parametry dohromady způsobují, že výsledný efekt z tohoto fotoaparátu není příliš patrný.

Samotné renderování snímků s tímto efektem je velmi prosté. Obrázek 2.22 znázorňuje, které pixely je nutno použít aby se efekt maximálně přiblížil efektu z obrázku 2.19. Použité pixely jsou na obrázku zvýrazněny červenými čtverečky, bílé čtverečky na pravé straně obrázku ukazují, z jaké části surového obrázku zvětšené výřezy pocházejí. Například v levém horním rohu fotografie jsou vybírány pixely z levého horního rohu mikročočky, obdobně je to i s ostatními směry. Směrem ke středu snímku jsou pixely vybírány blíže středu mikročočky a uprostřed snímku jsou pixely vybírány zprostřed mikročočky (stejně vybírání pixelů jako na obrázku 2.19).



Obrázek 2.22 Rozšíření optiky.

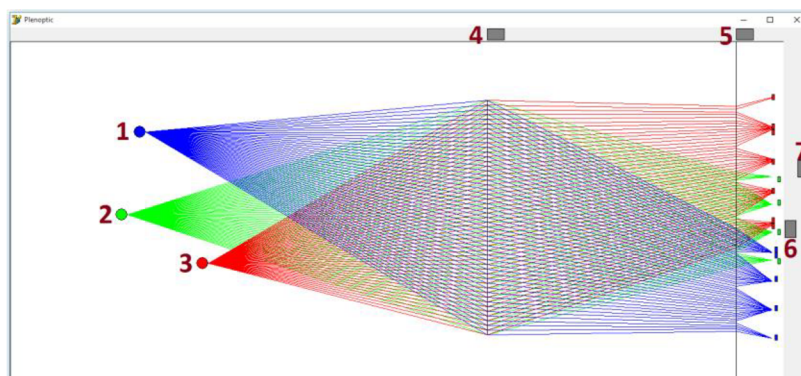
Na obrázku 2.23 vlevo se nachází výsledný snímek vykreslený tímto způsobem, porovnaný s klasicky vytvořeným snímkem vpravo. Na první pohled jsou téměř stejné, ale po bližším zkoumání je patrné, že objekty, které se nachází v blízkosti objektivu, jsou na pravém snímku větší a naopak vzdálené objekty menší. Toto má za následek rozdílná perspektiva. Z obrázku 2.23 je také zřejmé, že tento efekt s kamerou Lytro není příliš výrazný, to ovšem neznamená, že by nebyl prakticky využitelný. Stejně jako většina efektů nejlépe vynikne až animováním několika snímků za sebou.



Obrázek 2.23 Výsledné snímky s různou perspektivou.

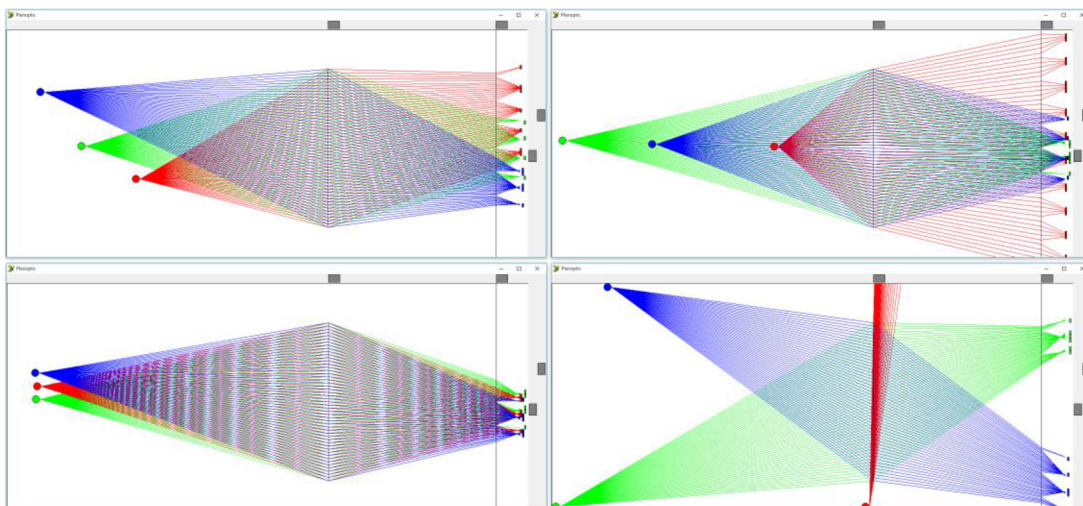
### 3 INTERAKTIVNÍ APLIKACE

V rámci semestrálního projektu byla vytvořena interaktivní aplikace, která má případným zájemcům o tuto problematiku pomoci s pochopením principu plenoptického snímání. Aplikace byla vytvořena v grafickém vývojovém prostředí Delphi 7.0 (Build 4.453) společnosti Borland v jazyce Object Pascal. Zdrojové kódy programu jsou k dispozici jako elektronická příloha diplomové práce. Výsledná podoba aplikace byla dle požadavků vedoucího práce upravena pro studijní účely. Aplikace znázorňuje, jak světlo prochází plenoptickým fotoaparátem a jak dopadá na senzor. Na obrázku 3.1 je zobrazen výsledný vzhled aplikace.



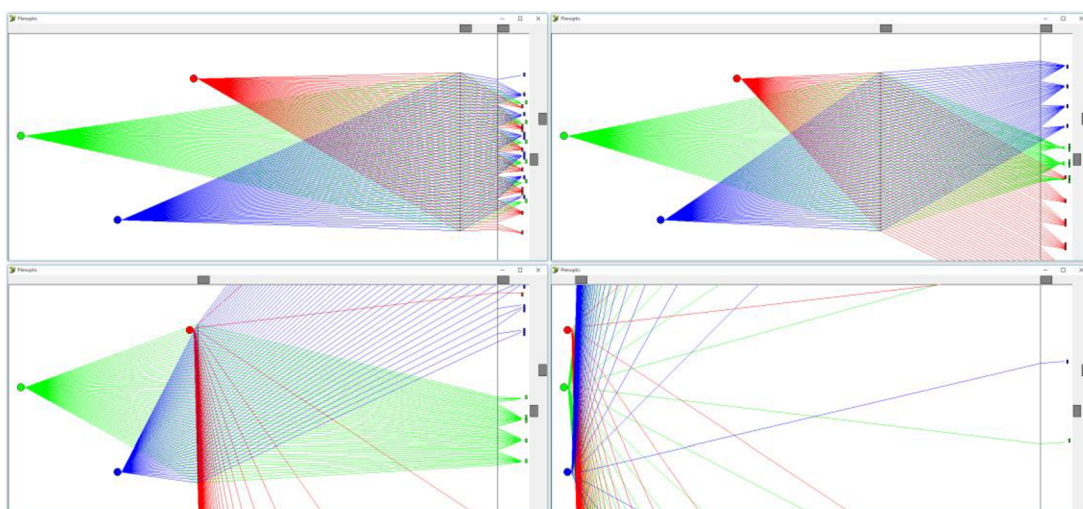
Obrázek 3.1 Vzhled interaktivní aplikace.

Aplikace obsahuje 7 interaktivních prvků, prvky 1-3 znázorňují objekty umístěné v  $2D$  prostoru před čočkou fotoaparátu. Objekty se přesouvají pomocí myši, přičemž je zajištěno, aby se nedostaly mimo prostor aplikace, nebo za rovinu čočky. Různé polohy objektů znázorňuje obrázek 3.2. Paprsky procházející do fotoaparátu jsou během přesouvání objektů automaticky překreslovány.



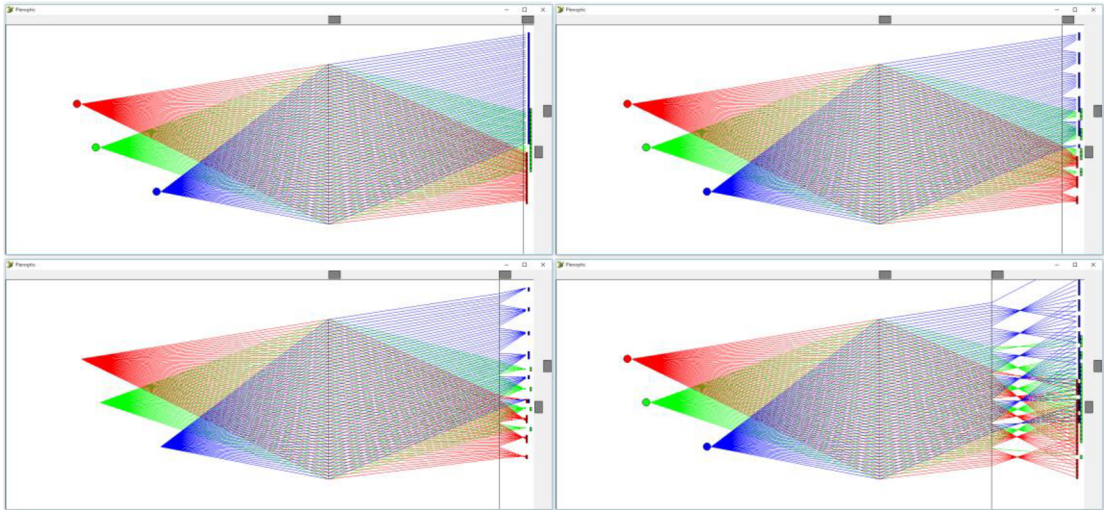
Obrázek 3.2 Změna polohy objektů v prostoru.

Prvek 4 z obrázku 3.1 funguje jako posuvník slouží ke změně polohy hlavní čočky fotoaparátu. Je zajištěno, aby se posuvník nedostal mimo povolené meze, objekty 1-3 se v případě přiblížení hlavní čočky automaticky odsouvají. Obrázek 3.3 ukazuje různé pozice tohoto posuvníku.



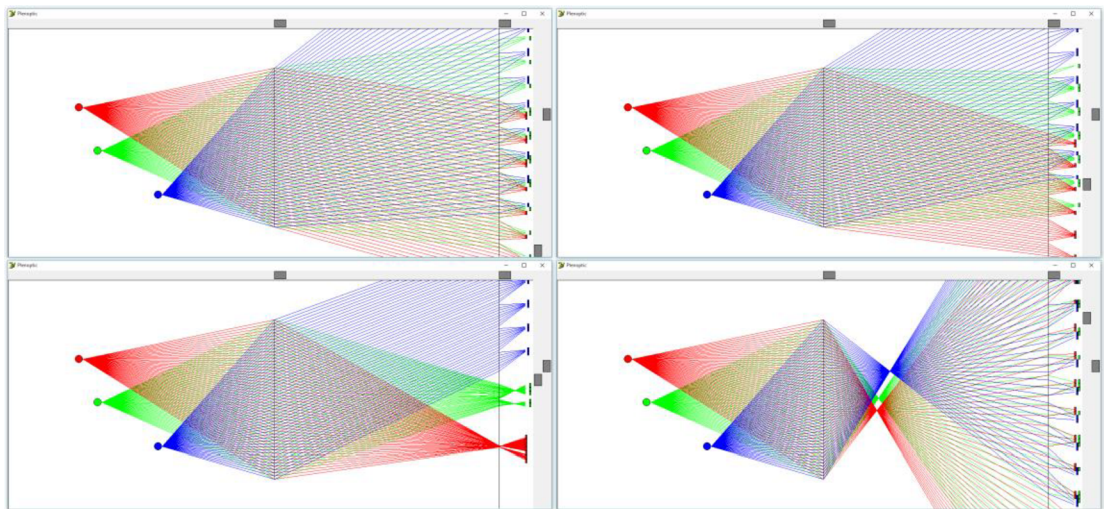
Obrázek 3.3 Změna polohy hlavní čočky.

Prvek 5 z obrázku 3.1 funguje jako posuvník slouží ke změně vzdálenosti pole mikročoček od senzoru. Je zajištěno, aby se posuvník nedostal mimo povolené meze. Obrázek 3.4 ukazuje různé pozice posuvníku.



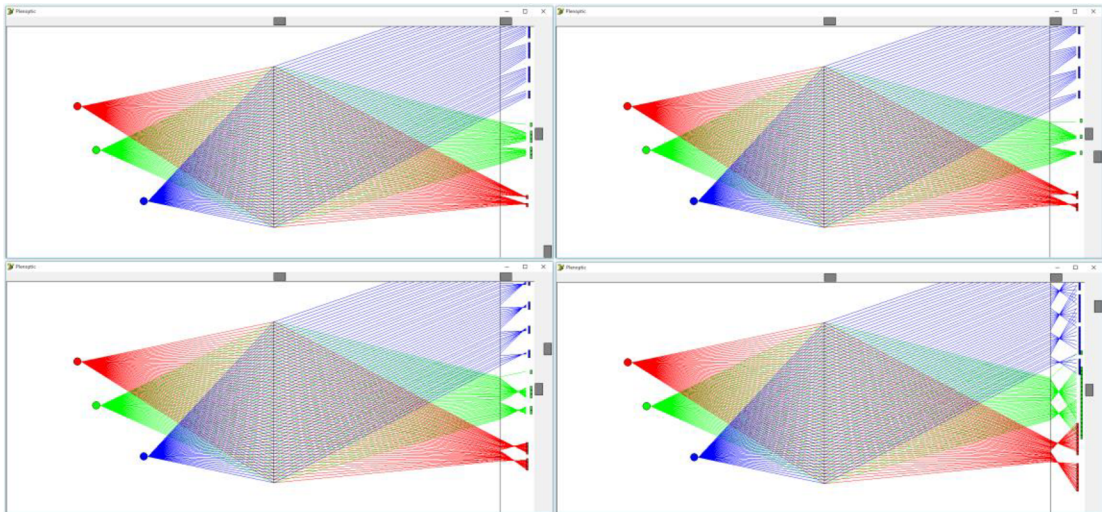
Obrázek 3.4 Změna polohy pole mikročoček.

Prvek 6 z obrázku 3.1 určuje ohniskovou vzdálenost hlavní čočky fotoaparátu. Různé ohniskové vzdálenosti zobrazuje obrázek 3.5.



Obrázek 3.5 Změna ohniskové vzdálenosti hlavní čočky.

Prvek 7 z obrázku 3.1 určuje ohniskovou vzdálenost mikročoček fotoaparátu. Různé ohniskové vzdálenosti zobrazuje obrázek 3.6.



Obrázek 3.6 Změna ohniskové vzdálenosti mikročoček.

Všechny tyto ovládací prvky dohromady umožňují uživateli navrhnout vlastní plenoptický fotoaparát a prohlédnout si, jak jím prochází světlo na senzor.

Aplikace záměrně pracuje s určitou mírou abstrakce, pro přehlednost je namísto objektivu složeného se soustavy čoček použita pouze jediná čočka. Poměr velikosti hlavní čočky a snímacího senzoru neodpovídá skutečnému fotoaparátu. Počet mikročoček a rozlišení je mnohonásobně nižší než na reálném senzoru. Tyto abstrakce vytvářejí určité nepřesnosti, avšak dělají tuto aplikaci velmi demonstrativní.

## 4 RENDEROVACÍ PROGRAM

Pro ověření teoretických poznatků byly v programovacím prostředí Matlab vytvořeny skripty pro renderování snímků pořízených plenoptickým fotoaparátém Lytro. Tyto skripty také posloužily k vygenerování obrázků pro tuto práci. Skripty umožňují zobrazit surové snímky, provést kalibraci na konkrétní hardware, vygenerovat snímky s různou rovinou ostrosti, hloubkou ostrosti, perspektivou a pohledem na scénu. Tato kapitola rozebírá celý proces od pořízení snímku, stažení do *PC*, extrahování, až po vykreslení finálních snímků. Kompletní skripty jsou k dispozici jako elektronická příloha této práce.

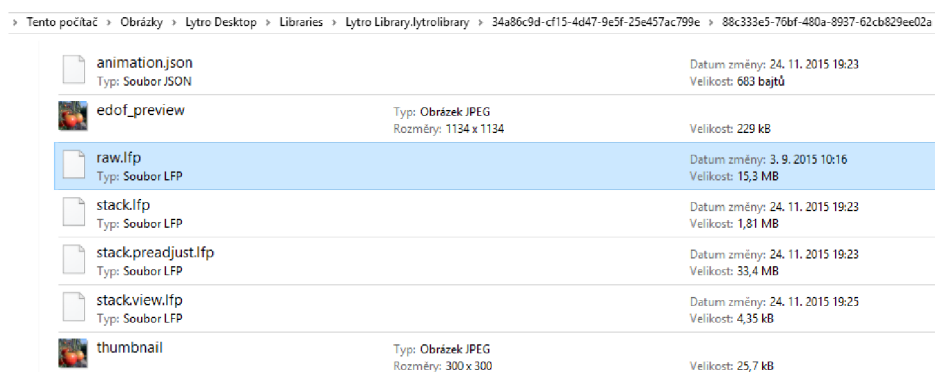
### 4.1 Postup získání surových dat

Po pořízení snímku se do paměti fotoaparátu neuloží snímek se standardním formátem, ale speciální datový soubor, který obsahuje mimo samotných obrazových *RAW* dat, také mnoho metadat. Pro import souborů do *PC* je nutné využít oficiální software Lytro. Tento software je dostupný zdarma na webových stránkách výrobce fotoaparátu. Mimo importu souborů do *PC* tento software umožňuje renderování snímků s různou rovinou ostrosti, základní funkce pro úpravu fotografií a animaci přeastření.

Po importu nalezneme jednotlivé fotografie ve složce:

**C:\Users\Uzivatel\Pictures\Lytro Desktop\Libraries\LytroLibrary.lytrolibrary\  
34a86c9d-cf15-4d47-9e5f-25e457ac799e**

nebo obdobné. Jednotlivé fotografie jsou zde rozmístěny do samostatných složek, které obsahují náhledy fotografií, *RAW* data a další pomocné soubory, tak jak je to znázorněno na obrázku 4.1.



Obrázek 4.1 Importované soubory do *PC*.



Pro další zpracování snímků nám poslouží pouze soubor „raw.lfp“. Obsah tohoto souboru je obdobný s *RAW* soubory z tradičních digitálních kamer. Obsahuje, jak pomocná metadata, tak i nekomprimovaná obrazová data ze senzoru, která je ovšem nutné dále zpracovat.

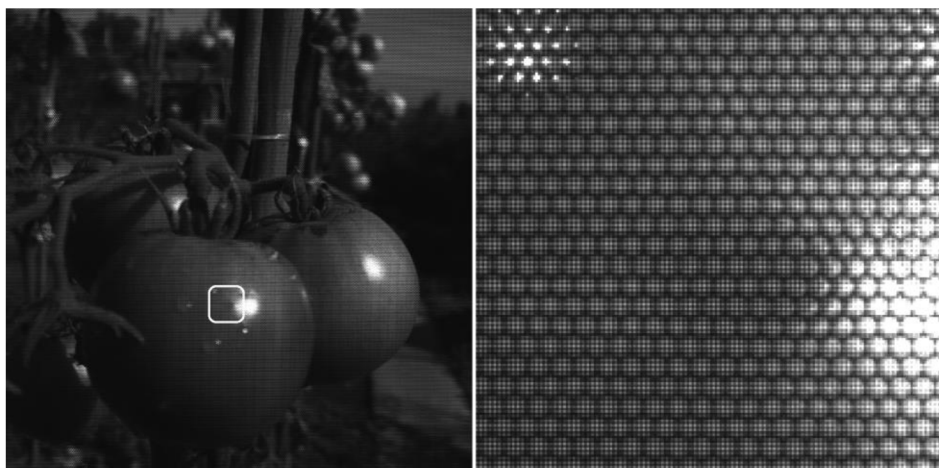
Kamera Lytro disponuje senzorem o rozlišení 3280 x 3280 pixelů a dvanáctibitovým rozlišením pro každou barvu. Za účelem úspory paměti jsou každé 2 pixely (24 bitů) zabaleny společně do 3 bajtů. Délka tohoto oddílu je vždy přesně 16 137 600 bajtů. Data jsou seřazena podle následujícího vzoru: 12 34 56 musí být dekódováno na 03 12 06 45 [10]. Řazení pixelů začíná v levém horním rohu fotografie a postupuje po řádcích. Výsledný snímek lze zobrazit ve stupních šedi. Pixely odpovídají jednotlivým barvám *RGB* a to podle barevného vzoru *BGGR*.

Metadata obsažená v souboru *LFP* obsahují informace o typu snímku, jeho velikosti, orientaci, způsobu zakódování, endianitě (použitá je typu „big“), použité barevné mozaice. Dále obsahují informace o nastavení senzoru při pořizování fotografie, jako je nastavení gammy, citlivost *ISO*, vyvážení bílé. Dále pak informace o čase pořizení fotografie, teploty základní desky a čoček, hodnoty z akcelerometru, sériového čísla senzoru a kamery, informace o módu kamery, firmwaru, modelu kamery atd. Velice důležité jsou informace o konkrétním použitém senzoru zejména úhel pootočení pole mikročoček na senzoru.

První, kdo dokázal pomocí zpětného inženýrství rozložit *LFP* Lytro formát byl Nirmal Patel. Vytvořil aplikaci pracující v příkazovém řádku: „lfpsplitter.c“, která rozloží *LFP* soubor na jednotlivé části. Na práci Nirmala Patela navázal Donald G. Dansereau, který napsal obdobné skripty pro rozložení souboru *LFP* v prostředí Matlab. Tyto skripty dále poslouží pro extrahování obrazových *RAW* dat a meta dat.

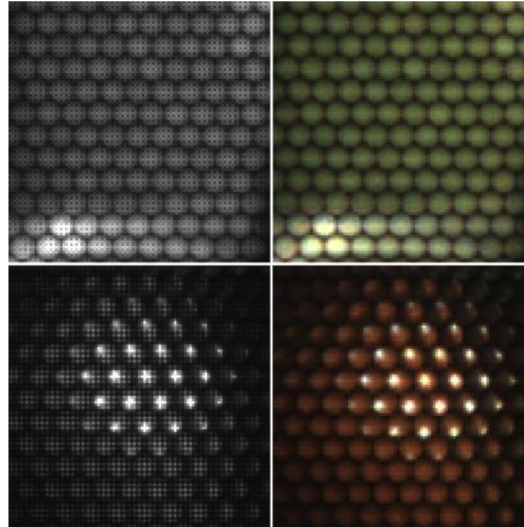
## 4.2 Úprava RAW dat

Surová obrazová data získaná extrahováním je možné přímo zobrazit ve stupních šedi. Po přímém zavolání příkazu `imshow(rawdata)`, bude výsledný snímek téměř celý černý, toto je způsobeno tím, že je dvanáctibitová informace uložena v šestnáctibitovém prostoru a následně chybně interpretována. Problém lze jednoduše vyřešit vynásobením všech pixelů hodnotou 16, která provede bitový posun hodnot o čtyři pozice směrem k bitům s nejvyšší vahou (*MSB*). `imshow(rawdata(:, :, :) * 16)`. Výsledek znázorňuje obrázek 4.2.



Obrázek 4.2 Stupně šedi.

Pro obarvení snímku poslouží funkce Matlabu - „*demosaic*“ parametrem této funkce je použitý barevný vzor. Z metadat snímku bylo zjištěno, že použitý barevný vzor je „*bggr*“. Tento fakt je možné snadno ověřit pomocí testovacích snímků. Na obrázku 4.3 je patrné, které pixely patří konkrétní barvě. Po bližším pozorování zjistíme také, že pixelů popisující zelenou barvu je 2x více než červených nebo modrých. Světlejší pixel vyjadřuje výraznější barevnou složku.



Obrázek 4.3 Barevný vzor.

Zavoláním funkce `rawdata = demosaic(rawdata, 'bggr');` dostaneme z matice `uint16 3280x3280` matici `uint16 3280x3280x3`. Výsledek je znázorněn na obrázku 4.4.



Obrázek 4.4 Obarvení podle vzoru 'bggr'.

Tyto barvy zatím nejsou korektní, v této fázi je zcela zanedbán vliv nastavení *CMOS* senzoru.

Dalším krokem bude vyvážení bílé barvy. Nejprve je nutné správně extrahovat příslušná metadata, jedná se o 3 hodnoty popisující zisk jednotlivých barevných kanálů RGB. Těmito hodnotami je nutné vynásobit každý pixel matice, například vyvážení červené barvy: `rawdata(:, :, 1) = rawdata(:, :, 1) .* Whitebalance;`

Následný efekt ukazuje obrázek 4.5 vlevo, rozdíl oproti obrázku 4.4 není příliš patrný, ale při bližším pozorování se barvy opravdu jeví jako přirozenější. Ovšem důležitost vyvážení bílé se naplno projeví až po kompletním zpracování snímku. Na obrázku 4.5 vpravo se nachází kompletně zpracovaný *RAW* s a bez vyvážení bílé.

Vidíme, že zde už je rozdíl poměrně výrazný.



Obrázek 4.5 Vyházení bílé.

Dalším krokem je převedení barev z *RGB* do *sRGB*. To bylo provedeno maticovým násobením *RGB* snímku s maticí pro korekci barev (*CCM*).

$$sRGB = RGB * CCM , \quad (4.1)$$

Příčemž *RGB* představuje matici snímku vhodně převedenou na tvar, kde každý pixel představuje jeden řádek matice a celkový rozměr matice je tedy  $10758400 \times 3$  ( $3280 \times 3280 \text{pix} = 10758400$ ). *CCM* byla získána z metadat snímku, jedná se o matici o rozměru  $3 \times 3$ . Výslednou matici *sRGB* je nutné před zobrazením nejprve přeskládat do původního stavu ( $3280 \times 3280 \times 3$ ). Je patrné, že barevné podání obrázku 4.6 je neporovnatelně lepší než na obrázku 4.5.



Obrázek 4.6 Převod na *sRGB*.

Posledním krokem bude v tuto chvíli korekce gradačního zkreslení. Hodnotu gradačního zkreslení získáme z metadat snímku, poté každý prvek matice

touto hodnotou umocníme. Výsledný efekt vidíme na obrázku 4.7. Barvy nejsou přehnaně křiklavé a obrázek působí přirozeněji.

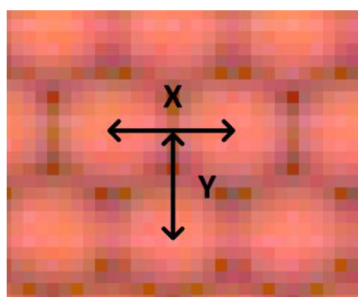


Obrázek 4.7 Korekce gradačního zkrvení.

### 4.3 Kalibrace

První nutný krok pro práci se světelným polem je kalibrace, jejímž úkolem je nalézt středy jednotlivých mikročoček. Zde se velmi nepříznivě projevuje technologická nepřesnost nanesení pole mikročoček na senzor. Pole jsou na jednotlivých vyrobených kamerách náhodně posunuty a pootočený. Z tohoto důvodu je nutné kalibrovat algoritmus na konkrétní senzor.

Pro účel kalibrace byl napsán samostatný skript. Prvním úkolem je nalézt přesnou rozteč dvou sousedních mikročoček a to jak v ose  $X$ , tak v ose  $Y$ , viz obrázek 4.8.



Obrázek 4.8 Rozteč  $X Y$ .

Hodnoty  $X$ ,  $Y$  nejsou celá čísla, tudíž je při vybírání středových pixelů nutné zaokrouhlování, které způsobuje poměrně vysokou chybu. Z tohoto důvodu je vhodné, celé pole rozšířit. Pro rozšíření byla použita funkce `imresize()`. Parametr funkce byl zvolen  $p = 4$ , což znamená, že v obou osách čtyřnásobně zvýší rozlišení. Tato hodnota se ukázala jako dobrý kompromis mezi přesností a výpočetní náročností. Navržená kostra kódu pro hledání středů může vypadat například takto:

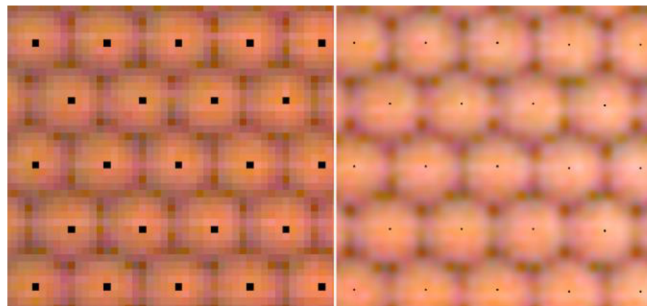
```

for w=0:pocetbuneky
  for q=0:pocetbunekx
    if rem(w,2) == 0
      stredx=pocatekx+round(q*bunkax);
    else
      stredx=pocatekx+round(q*bunkax+bunkax/2);
    end
    stredy=pocateky+round(w*bunkay);
    rawdata(stredy, stredx, :)=0;
  end
end
end

```

Ve dvou vnořených cyklech, které se vykonají tolikrát, kolik je v jednotlivých osách mikročoček, se určí souřadnice  $X$ ,  $Y$  všech středů. Ty se do obrázku zaznamenají jako černý bod. Podmínka `if rem(w,2) == 0` rozlišuje sudé a liché řádky. Souřadnice středu první použité mikročočky byly zjištěny přímo ze snímku.

Pro zjištění přesných rozměrů mikročoček je možné použít následující postup. Nejprve je nutné odhadnout přibližné hodnoty  $X$  a  $Y$ , poté vykonat předešlý algoritmus s parametry například `pocetbuneky = 100`; a `pocetbunekx = 100`; Zjištěná chyba na 100 mikročočkách bude činit například `+20pix`. Z toho vyplývá, že je nutné k odhadu přičíst hodnotu  $20/100 = 0,2$ . Tento proces je možné pro přesnější určení rozměrů opakovat i vícekrát. Výsledek algoritmu ukazuje obrázek 4.9, na levé straně bez rozšíření, na pravé s rozšířením pole. Je patrné, že bez rozšíření pole by bylo určení středu velice nepřesné.



Obrázek 4.9 Určení středů mikročoček bez rozšíření a s rozšířením matice.

V této fázi je nutné vyřešit problém s pootočením pole mikročoček na senzoru. Úhel pootočení je součástí metadat a jeho hodnota je pro konkrétní použitou kameru rovna: `rotace = 0.00478` rad. Pro uvážení vlivu rotace postačí přidat do předešlého algoritmu tyto dva řádky:

```

stredy=round(stredy+sin(rotace)*stredx);
stredx=round(stredx-sin(rotace)*stredy);

```

Správnost kalibrace je vhodné nakonec zkontrolovat. Na obrázku 4.10 vlevo vidíme chybu způsobenou špatně nastavenou šířkou mikročoček, uprostřed výškou mikročoček a vpravo kombinací obou, stejný efekt má i špatně nastavený úhel pootočení. Vidíme, že chyby se projevují velice pravidelnými tvary. Tmavší pruhy jsou způsobeny tím, že pixely jsou vybírány z tmavší oblasti mezi mikročočkami, namísto ze středu

mikročoček.



Obrázek 4.10 Chyby kalibrace.

## 4.4 Renderování snímku z určitého úhlu pohledu

Tato kapitola podrobně popisuje algoritmus renderování snímků z určitého úhlu pohledu a navazuje tak na kapitolu 2.3.1, kde byla tato problematika probrána z teoretického hlediska. Nejprve je nutné příslušně upravit *RAW* data obrázku a kalibrovat algoritmus na konkrétní Hardware, jak bylo popsáno v předešlých kapitolách. Jádrem algoritmu je obdobné jako v předešlé kapitole s tím rozdílem, že nevybíráme pouze jeden středový pixel, ale celý blok pixelů, například dvěma vnořenými cykly takto:

```
for e=1:blokY
    for r=1:blokX
        vyslednysnimek(kamy+e,kamx+r,:)= rawdata(stredy+e,stredx+r,:);
    end
end
```

Druhou možností je nahrazení dvou vnořených cyklů maticovou funkcí Matlabu, která několikanásobně urychlí provedení algoritmu. Následující příkaz obtiskne blok pixelů z matice *rawdat* do výsledného snímku.

```
vyslednysnimek(kamy:kamy+blokY,kamx:kamx+blokX,:)=
rawdata(stredy:stredy+blokY,stredx:stredx+blokX,:);
```

Možnou nevýhodou tohoto zápisu může být určitá nepřehlednost kódu.

Velikost bloku je nutné volit takovým způsobem, aby na sebe sousedící mikročočky co nejlépe navazovaly a zároveň byl zachován správný poměr stran výsledného snímku. Bylo vyzkoušeno mnoho různých poměrů například  $\text{blokX} = 4$  a  $\text{blokY} = 4$ , který velmi dobře vyhovuje podmínce vzájemného navázání dvou sousedních mikročoček, ale poměr stran je nevyhovující, což se projeví jako zúžení snímku, viz obr 4.11.



Obrázek 4.11 Velikost vybíraných bloků 4x4pix.

Správný poměr stran vybíraného bloku by měl být shodný s poměrem vzdáleností  $X$  a  $Y$  dvou středů mikročoček.  $X = 40,0216$  a  $Y = 34,6655$ , poměr se tedy rovná  $Y/X = 0,86617$  toto číslo je možné vyjádřit velmi přesně zlomkem  $13/15$ . Tento poměr lze programově realizovat například tak, že bude 2x vybrán blok 4:5 a jednou blok 5:5. Výsledný snímek i detail s rozeznatelnými bloky zobrazuje obrázek 4.12. Na detailu snímku je vidět, jak se střídají řádky s výškou 4 pix a výškou 5 pix. Dále je na detailu snímku patrné, že navázání jednotlivých bloků není zcela dokonalé. Tato nedokonalost při dalším zpracování snímku např. realizaci přestřehání téměř vymizí.



Obrázek 4.12 Bloky s poměrem 13/15.

Vytvoření snímku z různého úhlu pohledu je velice snadné postačí vybíraný blok pixelů posunout směrem od středu mikročočky požadovaným směrem. Na obrázku 4.13 vidíme snímek vykreslený z levého okraje, ze středu a pravého okraje mikročoček. Je patrné, že snímky z okrajů jsou mírně tmavší, to je způsobeno tím, že do pixelů umístěných pod okrajem mikročoček dopadá méně světla.





Obrázek 4.13 Různé pohledy na scénu.

Při přemrštěném posuvu se vybraný blok může posunout na rozhraní, nebo až pod vedlejší buňku.

## 4.5 Renderování snímků s různou rovinou ostrosti

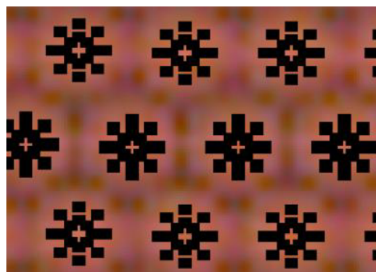
Tato kapitola navazuje na kapitolu 2.3.2, kde byla možnost přeastření rozebrána po teoretické stránce. Pro vykreslení snímků s různou rovinou ostrosti je nejprve nutné vytvořit sadu snímků s různým pohledem na scénu a poté tyto snímky vhodným způsobem prolnout.

Algoritmus pro vykreslení snímků z různých úhlů pohledu je velmi vhodné volat jako funkci, jejíž vstupní parametry představují posuvy  $X$  a  $Y$  od středu mikročoček.

```
posuvy_XY = [5 0 -5 0 10 0 -10 0 7 7 -7 -7 3 3 -3 -3;
             0 5 0 -5 0 10 0 -10 7 -7 7 -7 3 -3 3 -3];

for o=1:size(posuvy_XY,2)
    uhelpohledu(posuvy_XY(1,o),posuvy_XY(2,o));
    figure()
    imshow(vyslednamatice(:, :, :, o))
end
```

Tato konstrukce vykreslí 16 snímků z různých úhlů pohledu. V proměnné *posuvy\_XY* jsou uloženy požadované posuvy, které ve výsledku vytváří vzor z obrázku 4.14.



Obrázek 4.14 Vzor vybraných pixelů.

Počet použitých snímků má zásadní vliv na kvalitu výsledného obrázku, jak také ukazuje obrázek 4.15. Obrázek vlevo byl vytvořen se 2 snímky, obrázek uprostřed z 8 snímků a obrázek vpravo z 21 snímků. Zásadní rozdíl je hlavně v kvalitě

rozostřených míst, kde jsou v případě použití nízkého počtu snímků patrné jednotlivé příspěvky.



Obrázek 4.15 Vliv počtu použitých snímků na kvalitu výsledného snímku.

Jednotlivé snímky je vhodné ukládat do jedné matice. Pro další zpracování je nezbytně nutné znát posuvy jednotlivých snímků. Následující algoritmus vykreslí 50 snímků z postupně se měnící rovinou ostrosti. Rozsah hodnot od -2,5 do 2,5 určuje rovinu zaostření, tak jak bylo vysvětleno v kapitole 2.3.2.

```
for i=-2.5:0.1:2.5
    rovinaostrosti= i;
    for o=1:size(posuvy_XY,2)
        funkce_prolnuti(posuvy_XY(1,o),posuvy_XY(2,o),rovinaostrosti);
    end
    figure()
    imshow(vysledny_snimek) %snímek s rovinou ostrosti i
end
```

Přičemž funkce `prolnuti` pouze sečte příslušně posunuté jednotlivé snímky. Kvalitu přeostrění ukazuje obrázek 4.16.



Obrázek 4.16 Přeostrění – dosažená kvalita.

## 4.6 Renderování snímků s různou hloubkou ostrosti

Dalším realizovaným efektem je renderování snímků s různou hloubkou ostrosti, teoreticky rozebrané v kapitole 2.3.3. Změna hloubky ostrosti je pouze přídáním funkce k renderování snímků s různou rovinou ostrosti. Jedinou nutnou úpravou je vynásobení proměnné udávající posuvy dílčích snímků hodnotou od 0-1, která bude udávat hloubku ostrosti. Nevýhodou renderování různých hloubek ostrosti je, že pro každý další obrázek je nutné vykreslit nové dílčí snímky. Obrázek 5.17 ukazuje 2 snímky, se shodnou rovinou zaostření lišící se pouze hloubkou ostrosti.



Obrázek 4.17 Změna hloubky ostrosti – dosažená kvalita.

## 4.7 Renderování snímků s různou perspektivou

Nejzajímavějším realizovaným efektem je takzvaná změna perspektivy teoreticky rozebraná v kapitole 2.3.4. Při vytváření snímků s tímto efektem se bloky pixelů vybírají podle umístění na obrázku. Na středu obrázku se bloky vždy vybírají ze středu mikročoček, směrem ke krajům obrázku se plynule posouvají i vybrané bloky pixelů. Velikost a směr tohoto posuvu udává výslednou perspektivu. V zásadě jde o vytvoření dvojice lineárních rovnic, které budou vybíraný blok posouvat v jednotlivých osách. Tuto funkci zastane například rovnice 4.2,

$$posuvX = perspektiva - \left( x * \frac{perspektiva * 2}{pocet} \right), \quad (4.2)$$

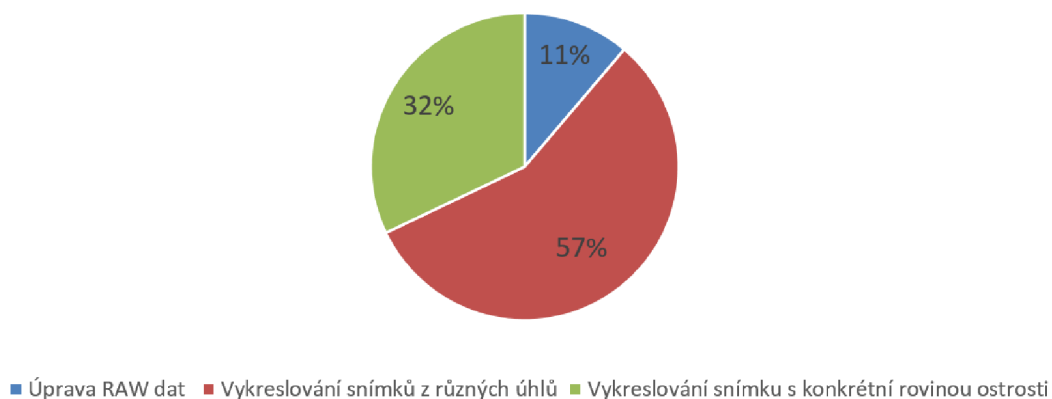
kde  $posuvX$  udává výsledný posuv v ose  $x$ ,  $perspektiva$  udává zvolenou hodnotu maximálního posuvu (na krajích obrázku), proměnná  $x$  udává konkrétní pořadové číslo mikročočky a proměnná  $pocet$  udává celkový počet mikročoček v ose  $x$ . Přehnaný posuv se projeví tmavšími rohy výsledného snímku, jak také ukazují obrázky 4.18.



Obrázek 4.18 Změna perspektivy – přehnaný posuv.

## 4.8 Časová náročnost

Vzhledem k velkým rozměrům zpracovávaných polí a velkému počtu prováděných operací je zpracování snímků světelného pole značně výpočetně náročné. Za účelem zefektivnění algoritmů bylo provedeno mnoho optimalizací. Snímky byly renderovány pomocí MATLAB R2013a pod Windows 10 na sestavě s procesorem Intel® Core™ i5 4200M CPU @ 2,50GHz, 8GB RAM. Časová náročnost zpracování jednotlivých fází snímků byla opakovaně měřena funkcí Matlabu „tic – toc“. Výsledný čas potřebný pro úpravu RAW dat 7,75 s, vykreslení jednoho snímku z určitého úhlu pohledu 2,37 s, vykreslení snímku s konkrétní rovinou ostrosti 0,4 s. Kompletní zpracování a vygenerování sady 50 snímků s různou rovinou ostrosti, z nichž je každý složený z 16 snímků z různých úhlů pohledu, potvrzuje tedy cca 67 s. Procentuální časovou náročnost jednotlivých kroků vyjadřuje graf na obrázku 4.19.



Obrázek 4.19 Procentuální vyjádření časové náročnosti jednotlivých kroků.

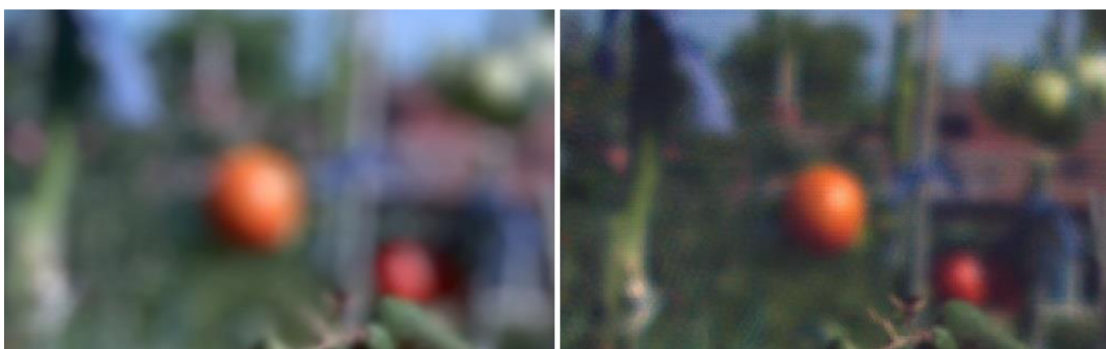
## 4.9 Porovnání s komerčním programem

Komerčně dostupný software dodávaný s kamerou Lytro umožňuje z výše uvedených efektů vykreslovat pouze snímky s proměnnou rovinou zaostření. Cílem vytvořených skriptů primárně bylo demonstrovat možnosti plenoptických fotoaparátů a zároveň se kvalitou co nejvíce přiblížit snímkům vytvořeným komerčním softwarem. Obrázek 4.20 porovnává snímek vytvořený komerčním softwarem vlevo se snímkem vytvořeným pomocí výše popsaných skriptů. Porovnávané snímky se jemně liší v nastaveném jasu a živosti barev, tyto rozdíly ovšem nesouvisejí s kvalitou snímku, ale spíše se vkusem pozorovatele. Dále si můžeme všimnout, že komerční software dosahuje menší hloubky ostrosti (pozadí je více rozostřeno).



Obrázek 4.20 Porovnání kvality výsledných snímků.

Nejvýraznější rozdíl odhaluje detail rozostřené plochy na obrázku 4.21, zde je vidět, že snímek vytvořený komerčním softwarem (vlevo) neobsahuje žádné nespojitosti a rozostření je velice plynulé. Na obrázku vpravo jsou patrné jemné nespojitosti.



Obrázek 4.21 Porovnání kvality – detail.

## 5 ZÁVĚR

V práci byla rozebrána teorie světelného pole, na jejímž principu pracují plenoptické fotoaparáty. Byl vysvětlen zásadní rozdíl mezi klasickým a plenoptickým fotoaparátem, který spočívá v rozdílné (bohatější) informaci, která je fotoaparátem o scéně pořízena. Dále byly popsány konstrukce plenoptických fotoaparátů od pole kamer, až po kompaktní kameru Lytro, která je nyní komerčně dostupná. Dále byly v práci postupně popsány jednotlivé metody renderování výsledných snímků, díky kterým jsou plenoptické fotoaparáty tak převratné. Například možnost jemné změny úhlu pohledu na scénu, změna perspektivy scény, možnost volby roviny ostrosti a hloubky ostrosti. Přičemž tyto možnosti najdou své uplatnění hlavně při animaci, během které se jednotlivé parametry fotografie plynule mění. Cenou za tyto schopnosti je relativně nízké rozlišení výsledných snímků, což je nejvýraznější nevýhodou plenoptických fotoaparátů.

V rámci práce byla vytvořena interaktivní aplikace, která názorně ukazuje, jak světlo prochází plenoptickým fotoaparátem a jak je zaznamenáváno na senzor. Tato aplikace by měla primárně sloužit pro studijní účely a pro základní pochopení plenoptického snímání.

Dále byla v prostředí Matlab vytvořena série skriptů pro renderování snímků z fotoaparátu Lytro. Vytvořené skripty umožňují ze surových dat zkomponovat veškeré efekty popsané v této práci. Tyto skripty také velmi dobře posloužily k ověření teoretických poznatků a vygenerování obrázků pro tuto práci.

Závěrem byla na tuto problematiku navržena laboratorní úloha. Stručný návod k této úloze je k práci přiřazen ve formě přílohy, příslušné skripty jsou k práci přiřazeny jako elektronická příloha.

# LITERATURA

- [1] RYCHLÍK, Martin. Antropologie umění: Etnoestetické studie s důrazem na preliterární a mimoevropské kultury. Praha, 2014. DIZERTAČNÍ PRÁCE. UNIVERZITA KARLOVA V PRAZE. Vedoucí práce PhDr. Václav SOUKUP, CSc.
- [2] HLAVÁČ, Ľudovít. Dejiny fotografie. Martin : Osveta, 1987.
- [3] LACY, Sarah. Lytro Launches to Transform Photography with \$50M in Venture Funds (TCTV) [online]. TechCrunch, [cit. 2011-08-18]. Dostupné z <http://techcrunch.com/2011/06/21/lytro-launches-to-transform-photography-with-50m-in-venture-funds-tctv/>
- [4] R. Ng, Digital Light Field Photography, Stanford University, 2006.
- [5] From M. Landy and J. A. Movshon (eds), Computational Models of Visual Processing (pp. 3-20). Cambridge, MA: MIT Press (1991).
- [6] The Stanford Multi-Camera Array. Stanford Computer Graphics Laboratory [online]. Stanford [cit. 2015-12-08]. Dostupné z: <http://graphics.stanford.edu/projects/array/>
- [7] An Ultra-Thin High Performance Monolithic Camera Array. Pelican imaging [online]. Santa Clara, CA, 2008 [cit. 2015-12-08]. Dostupné z: <http://www.pelicanimaging.com/technology/paper.html>
- [8] Lytro. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-12-08]. Dostupné z: <https://en.wikipedia.org/wiki/Lytro>
- [9] Bushberg JT, Siebert JA, Leidholdt EM, Boone JM. The essential physics of medical imaging. 2nd ed. New York, NY: Lippincott Williams & Wilkins; 2002.
- [10] *Computational photography of light-field camera and application to panoramic photography*. Praha, 2014. Diplomová práce. Univerzita Karlova v Praze. Vedoucí práce Ing. Filip Šroubek, Ph.D.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

$2D$	Dvoudimenzionální funkce.
$3D$	Trojdimenzionální funkce.
$4D$	Čtyřdimenzionální funkce.
$5D$	Pětidimenzionální funkce.
$u$	Směrová souřadnice.
$v$	Směrová souřadnice.
$x$	Prostorová souřadnice.
$y$	Prostorová souřadnice.
$E_F$	Matice výsledného snímku.
$F$	Vzdálenost roviny objektivu od obrazové roviny.
$L_F$	Plenoptická funkce.
$du$	Derivace podle $u$ .
$dv$	Derivace podle $v$ .
$MP$	Megapixel.
$RAW$	Z anglického raw, surový.
$RGB$	Z anglického red green blue, červená zelená modrá.
$PC$	Z anglického personal computer, osobní počítač.
$CMOS$	Z anglického Complementary Metal–Oxide–Semiconductor, doplňující se kov-oxid-polovodič.
$CCM$	Z anglického color correction matrix, matice pro korekci barev.



# A ZDROJOVÉ KÓDY

## A.1 Úprava RAW dat

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rawdata=ans.RawImg;
rawdata = demosaic(rawdata,'bggr');
rawdata= double(rawdata(:,:,:));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Whitebalance=[0 0 0];
Whitebalance(1)=ans.Metadata.image.color.whiteBalanceGain.r;
Whitebalance(2)=ans.Metadata.image.color.whiteBalanceGain.gr;
Whitebalance(3)=ans.Metadata.image.color.whiteBalanceGain.b;
rawdata(:,:,1)=rawdata(:,:,1).*Whitebalance(1);
rawdata(:,:,2)=rawdata(:,:,2).*Whitebalance(2);
rawdata(:,:,3)=rawdata(:,:,3).*Whitebalance(3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CCM = reshape((ans.Metadata.image.color.ccmRgbToSrgbArray), 3,3);
LFSize = size(rawdata);
NDims = numel(LFSize);
rawdata = reshape(rawdata, [prod(LFSize(1:NDims-1)), 3]);
rawdata = rawdata * CCM;
rawdata = reshape(rawdata, [LFSize(1:NDims-1), 3]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Gamma= ans.Metadata.image.color.gamma;
rawdata(:,:,:) = real(rawdata(:,:,:).^(Gamma)).*100;
rawdata = uint16(rawdata);
rawdata=imresize(rawdata,4,'lanczos3');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## A.2 Vykreslení snímku z určitého úhlu

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
bunkay=34.6655;
bunkax=40.0216;
rotace=0.00478;
kalibracex=2;
posunX=int16(posunX);
posunY=int16(posunY);

vysled(:, :, :, indexmatice)=0;
pom1=0;
pom2=0;
for w=0:374
    for q=0:321
        if rem(w,2) == 0
            stredx=120+round(q*bunkax)-kalibracex;
            licheradky=0;
        else
            stredx=160+round(q*bunkax-bunkax/2)-kalibracex;
            licheradky=2;
        end
        stredy=28+round(w*bunkay);
        stredy=round(stredy+sin(rotace)*stredx)+posunY;
        stredx=round(stredx-sin(rotace)*stredy)+posunX;
        kamy= pom1*5+pom2*4+1+40+20;
        kamx= q*5+1+licheradky+40+20;
        if rem(w,3) == 0
            if vzdalenost == 1
                vysled(kamy:kamy+4, kamx:kamx+4, :, indexmatice)=rawdata(stredy:stredy+4,
stredx:stredx+4, :);    %1x    4:4
            else
                vysled(kamy:kamy+4, kamx:kamx+4, :, indexmatice)=rawdata((stredy+stredy+1)
)-(stredy:stredy+4), (stredx+stredx+1)-(stredx:stredx+4), :);
            end
        else
            if vzdalenost == 1
                vysled(kamy:kamy+3, kamx:kamx+4, :, indexmatice)=rawdata(stredy:stredy+3,
stredx:stredx+4, :);    %2x    4:5
            else
                vysled(kamy:kamy+3, kamx:kamx+4, :, indexmatice)=rawdata((stredy+stredy+1)
)-(stredy:stredy+3), (stredx+stredx+1)-(stredx:stredx+4), :);
            end
        end
    end
    if rem(w,3) == 0
        pom1=pom1+1;
    else
        pom2=pom2+1;
    end
end
indexmatice=indexmatice+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

### A.3 Funkce vykreslující světelné paprsky (interaktivní aplikace)

```
for n:=1 to 51 do begin
  Image1.Canvas.Pen.Color:= cllime;
  Image1.Canvas.moveto(Shapel.left+20,Shapel.top+10-24);
  Image1.Canvas.LineTo(objektiv,92+n*8);

  x1:=(objektiv-(Shapel.left+20));
  y1:=(92+8*n-(Shapel.top+10-24));
  uhel:=y1/x1;

  lom:=round((26-n)*8*((pole-objektiv)/ohnisko));
  sou:=92+n*8+round(uhel*(pole-objektiv))+lom;
  Image1.Canvas.moveto(objektiv,92+8*n);
  Image1.Canvas.LineTo(pole,sou);

  y1:=sou-(92+8*n);
  x1:=pole-objektiv;
  uhel:=y1/x1;

  if sou>=0 then
  begin
    if sou<575 then
    begin
      lom:=round((30-((sou)mod 60))*((senzor-pole)/ohnisko2));
      sou2:=sou + round(uhel*(senzor-pole))+lom;
      Image1.Canvas.moveto(pole,sou);
      Image1.Canvas.LineTo(senzor,sou2);
      Image1.Canvas.Pen.Color:= cldefault;
      Image1.Canvas.Brush.Color:=cllime;
      Image1.Canvas.Rectangle(senzor+10,((sou2)div10)*10,
senzor+15,((sou2) div 10)*10+10);
    end;
  end;
end;
```

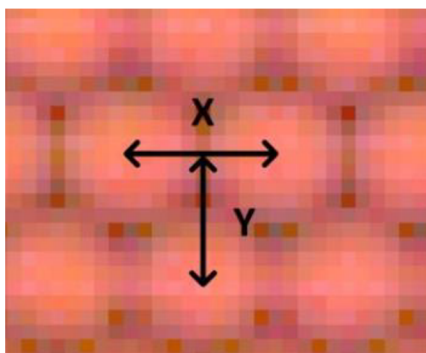
## B NÁVOD K LABORATORNÍ ÚLOZE

1. Seznamte se s plenoptickým fotoaparátem Lytro. Pořídte sérii fotografií, které budete dále zpracovávat. Při výběru scény dbejte na to, aby byly na fotografii zachyceny jak blízké objekty do vzdálenosti 25 cm od objektivu fotoaparátu, tak objekty ve vzdálenosti nad 10 m. Takto rozvržená scéna je optimální pro demonstraci schopností plenoptických fotoaparátů. Po pořízení snímků připojte fotoaparát pomocí USB kabelu k PC. Na PC spusťte **program Lytro** a importujte pořízené fotografie. Pořízené fotografie si zde můžete prohlédnout a libovolně přestřít. Surová RAW data najdete v PC na adrese:

**C:\Users\Uzivatel\Pictures\Lytro Desktop\Libraries\ LytroLibrary.lytrolibrary\34a86c9d-cf15-4d47-9e5f-25e457ac799e**

nebo obdobné. Pro každou fotografii je zde vytvořena samostatná složka. Pro další zpracování bude využit pouze soubor **raw.lfp**. Soubor si zkopírujte do vlastního adresáře a vhodně pojmenujte.

2. Otevřete skript **lab\_zpracovani\_RAW.m** doplňte název RAW souboru a spusťte. Postupně se vyrenderuje série snímků popisující jednotlivé fáze zpracování surových plenoptických RAW dat. Prohlédněte si detail snímků a popřemýšlejte, jaké informace jsou v něm obsaženy. Prostudujte, jak jednotlivé fáze zpracování ovlivňují obrazovou kvalitu výsledného snímku. Postupujte postupně od Figure1 dále.
3. Dalším úkolem bude kalibrace na konkrétní hardware. Otevřete skript **lab\_kalibrace.m**, úkolem kalibrace je najít středy jednotlivých mikročoček. Nejprve vyplňte název RAW souboru. Dále je nutné nalézt přesnou rozteč dvou sousedních mikročoček a to jak v ose X, tak v ose Y podle obrázku 1.



Obrázek 1 Rozteč X Y.

Pro zjištění přesných rozměrů mikročoček je možné použít následující postup. Odhadnout přibližné hodnoty X a Y, poté vykonat algoritmus s parametry například

$pocetbunex = 100$  a  $pocetbuney = 100$ . Z vykresleného snímku odečíst chybu. Zjištěná chyba na 100 mikročočkách bude činit například +20pix. Z toho vyplývá, že je nutné k odhadu přičíst hodnotu  $20/100 = 0,2$ . Tento proces je možné pro přesnější určení rozměrů opakovat i vícekrát. Všimněte si, jaké další parametry jsou pro provedení kalibrace nutné. Až budete s výsledkem spokojeni, přejděte na další úkol.

- 4 . Otevřete si skript **lab\_zmena\_uhlu.m**, cílem této úlohy je předvést možnost renderování snímků z různých úhlu pohledu z jediné plenoptické fotografie. Do skriptu doplňte jméno RAW souboru a spusťte. Vygeneruje se série snímků z různých úhlů pohledu. V levém horním rohu je znázorněno, ze kterých zdrojových pixelů je snímek složen. Posuvy od středů mikročoček jsou ve skriptu uloženy v proměnné `posuvy_XY`.
- 5 . V tomto bodě si vyzkoušíte schopnost plenoptického fotoaparátu Lytro přeastřovat snímky až po jejich pořízení. Otevřete si skript **lab\_preostreni.m** doplňte název RAW souboru, zvolte typ výstupu (jpeg/gif), a rozsah ostření podle komentářů přímo ve skriptu. Renderování snímků může trvat i několik minut. Po provedení skriptu se výsledné snímky uloží do zdrojového adresáře.