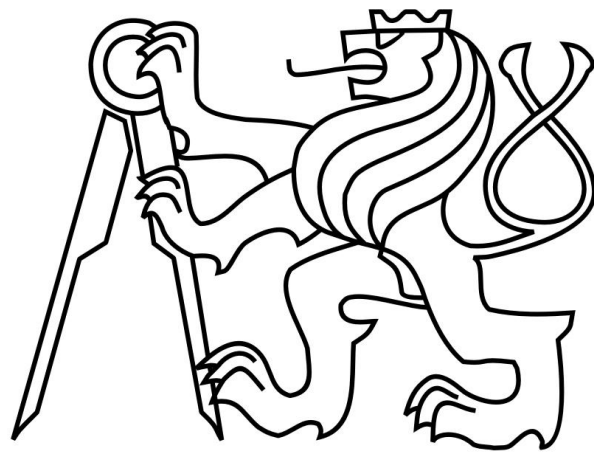


# České Vysoké Učení Technické

Fakulta strojní – Ústav mechaniky, biomechaniky a  
mechatroniky



Zpracování obrazu pro systém couvacího  
asistenta v automobilu

Vypracoval: **Bc. Patrik Vacek**  
Obor: **Mechatronika**

Akademický rok: **2017/2018**

zadaní

## **Prohlášení**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně s použitím uvedených zdrojů a literatury. Souhlasím také s případnou publikací výsledků diplomové práce nebo její podstatné části, pokud budu uveden jako její spoluautor.

V Praze, dne

Podpis

## **Anotace**

V rámci této diplomové práce je představen stručný přehled možností využití kamer pro parkovacího asistenta. Dále jsou uvedeny metody zpracování obrazu, které by mohly být vhodné pro účely asistovaného parkování. Další část byla zaměřena na metodu stereoskopie pro určení hloubky obrazu a tento přístup byl v praktické části testován na zkonstruované optické soustavě. Poté byl v práci popsán a proveden postup kalibrace a těchto kamerových modulů z Raspberry Pi, které byly programovány v jazyce Python. Experimentální část práce spočívala ve využití stereo vidění za účelem rekonstrukce scény pro využití v parkovacím asistentovi. Naměřená data byla interpretována a zpracována v prostředí Matlab. Hlavní použité metody zpracování obrazu byly segmentace, triangulace a detekce význačných bodů.

Klíčová slova: Zpracování obrazu, Kamery, Parkovací asistent, Stereo vize, Matlab

## **Annotation**

This diploma thesis contains brief overview about usage of cameras for needs of the parking assistant. The thesis also presents basic method of image processing, which can be recommended for assisted parking. Next part targets specifically method called stereo vision for it's ability to recover depth of the image. This particular method was tested in practical part of the thesis on constructed optical system. This asked for camera calibration, which is also mentioned, and applied on Raspberry Pi cameras. The cameras are programmed in Python, whereas all methods of image processing applied to acquired images are processed in Matlab. The experimental part aimed for usage of stereo vision in terms of reconstructing the scene to be later used in parking assistant. All acquired data were interpreted Matlab. Main used methods of image processing were segmentation, triangulation a detection of featured points.

Key words: Image processing, Cameras, Parking assistant, Stereo vision, Matlab

## **Poděkování**

Tímto bych chtěl poděkovat vedoucímu diplomové práce Ing. Janu Zavřelovi PhD. za konzultace a rady při tvorbě této diplomové práce, dále své rodině, která mě neustále podporovala a pomáhala mi během studia a dále bych chtěl poděkovat Bc. Vladislavu Kociánovi za motivační zprávy, které mě udržely bdělého při práci.

# 1 OBSAH

2	Seznam obrázků .....	14
3	Úvod .....	8
3.1	Cíle práce.....	9
4	Parkovací senzory .....	10
4.1	Zpětné kamery .....	10
4.2	360° kamery .....	11
4.3	Autonomní parkovací asistent .....	11
5	metody zpracování obrazu.....	12
5.1	Předzpracování obrazu .....	13
5.1.1	Změna kontrastu .....	13
5.1.2	Filtrace šumu.....	13
5.1.3	Histogram.....	14
5.2	Segmentace.....	14
5.3	Detekce význačných bodů.....	15
5.4	Klasifikace obrazu.....	15
5.5	Stereo Zobrazení .....	17
5.5.1	Chyba v měření vzdálenosti .....	20
5.5.2	Korespondence bodů .....	22
6	Kalibrace kamery.....	24
6.1	Pin hole model .....	25
6.2	Zkreslení čočkou [16] .....	28
6.3	Zkalibrování vlastní kamery .....	29
7	Hardware .....	33
7.1	Raspberry Pi [19].....	34
7.2	Konstrukce .....	35
8	Měření na kamerovém systému .....	38
9	Závěr.....	59
10	Reference .....	61

## 2 SEZNAM OBRÁZKŮ

Obr. 1: Grafické rozhraní parkovacího asistenta [3] .....	10
Obr. 2: Umístění parkovacích kamer do všech směrů pohledu [4] .....	11
Obr. 3: Ukázka aplikace histogramové ekvalizace na příkladu z matlabu [6] .....	14
Obr. 4: Ukázka segmentace na příkladu z matlabu [6] .....	14
Obr. 5: Ukázka detekce automobilů v příkladu z matlabu [6] .....	16
Obr. 6: Analogie s lidským způsobem vidění [10] .....	17
Obr. 7: Model stereoskopie za předpokladu dvou paralelních optických os [9] .....	18
Obr. 8: Princip promítání korespondujících bodů [11] .....	22
Obr. 9: Detekce bodů na epipolární přímce [11] .....	23
Obr. 10: Příklady využití kalibrace kamery [6] .....	24
Obr. 11: Znázornění pin-hole modelu [13] .....	25
Obr. 12: Projekce pin-hole modelu a znázornění parametrů [14] .....	26
Obr. 13: Schématické využití parametrů při transformaci obrazu [15] .....	27
Obr. 14: Typy zkreslení obrazu – bez zkreslení, soudkovitost, poduškové [16] .....	28
Obr. 15: Ukázka nafocených obrázků šachovnice při kalibraci .....	29
Obr. 16: Vložený souřadnicový systém v programu Matlab .....	30
Obr. 17: Znázornění extrinsických parametrů ve všech 50 snímcích kalibrace; detekovaná vzdálenost kamer od šachovnice .....	31
Obr. 18: Průměrná chyba zpětné projekce bodů po aplikaci parametrů; červený křížek – reprojekované body .....	32
Obr. 19: Vývojová deska Raspberry Pi a její kamerový modul [18] .....	34
Obr. 20: Model sestavy konstrukce navržený v program Autodesk Inventor .....	35
Obr. 21: Postavená měřicí soustava .....	35
Obr. 22: Propojení s Raspberry v program Putty .....	36
Obr. 23: Přenos souborů v program Filezilla mezi počítačem a Raspberry .....	37
Obr. 24: Program pro vyfocení synchronizovaných fotografií při zmáčknutí tlačítka .....	37
Obr. 25: Scéna s míčem, zesilovačem, klavírem, s madrací a tabulí .....	38

Obr. 26: Scéna po aplikaci koeficientů zkreslení.....	39
Obr. 27: Detekce význačných bodů nalezených v obou fotografiích.....	40
Obr. 28: Rekonstrukce scény zobrazená v matlabovském cloudu po redukcí pixelů.....	41
Obr. 29: 3D souřadnice měřených bodů.....	42
Obr. 30: Nová testovací scéna, kuchyně s linkou a barovými židlemi .....	43
Obr. 31: Rekonstrukce scény kuchyně v 3D souřadnicích .....	43
Obr. 32: Histogram zkonstruovaný ze získaných 3D souřadnic.....	45
Obr. 33: Logická mapa překážek .....	45
Obr. 34: Logická mapa se zakreslenou trajektorií a automobilem .....	46
Obr. 35: Obrázek dvou blízkých automobilů z obou kamer po odstranění soudkovitosti .....	47
Obr. 36: Detekce význačných bodů ve snímku .....	47
Obr. 37: Rekonstrukce scény dvou automobilů.....	48
Obr. 38: Zašuměná logická mapa překážek .....	49
Obr. 39: Naprogramovaná maska pro segmentaci.....	50
Obr. 40: Binární plocha segmentace obrazu.....	50
Obr. 41: Detekce význačných bodů segmentované části obrazu .....	51
Obr. 42: Rekonstrukce scény po segmentaci - bez detekované silnice .....	51
Obr. 43: 3D souřadnice měřených bodů v cloudu .....	52
Obr. 44: 2D Clusterování automobilů – modrá barva značí červené auto, červená značí černé a okolí.....	53
Obr. 45: Logická mapa překážek u sebe stojících automobilů.....	53
Obr. 46: Nová scéna umožňující zaparkování.....	54
Obr. 47: Segmentace scény s automobily daleko od sebe .....	55
Obr. 48: Nová maska pro zachycení vzdálenější vozidel.....	55
Obr. 49: Detekce bodů po aplikování nové masky .....	56
Obr. 50: Měřené souřadnice automobilů – předek karoserie a zadní světlo .....	56
Obr. 51: Logická mapa překážek parkovacího místa .....	57
Obr. 52: Ilustrace parkovacího manévru bez mechanismů křivosti, kinematiky a rozměrů vozidla .....	58



## Použité symboly

$\hat{\mathbf{x}}$	Souřadnice zpětně projektovaných bodů	[pixel]
$C_x, C_y$	Souřadnice optické osy v pin-hole modelu	[pixel]
$E_{\text{proj}}$	Reprojekční chyba	[pixel]
$f$	Ohnisková vzdálenost	[pixel]
$\mathbf{H}$	Matice homografie	
$\mathbf{K}$	Matice intrinsických parametrů	
$k_1, k_2$	Koeficienty radiálního zkreslení	-
$m_e$	Euklidovská vzdálenost měřených bodů	[m]
$\mathbf{P}$	Bod skutečného objektu	
$\mathbf{R}$	Matice rotace extrinsických parametrů	
$S$	Seperace paralelních kamer	[m]
$\underline{\mathbf{t}}$	Vektor translace extrinsických parametrů	[m]
$u$	Vzdálenost promítací roviny od čočky kamery	[m]
$v$	Vzdálenost objektu od čočky kamery	[m]
$u_j$	Souřadnice u j-tého bodu v promítací rovině	[pixel]
$v_j$	Souřadnice v j-tého bodu v promítací rovině	[pixel]
$x$	Souřadnice obrazu bodu od středu obrazu	[pixel]
$X$	Skutečná vzdálenost od středu obrazu	[m]
$\delta x$	Chyba při měření vzdálenosti obrazu bodu	[pixel]
$\delta v$	Chyba při měření vzdálenosti objektu	[m]
$\Delta x$	Rozdíl vzdáleností obrazu ve dvou snímcích	[pixel]
$\underline{\mathbf{k}}$	Vektor koeficientů radiálního zkreslení	-

### 3 ÚVOD

S rostoucím požadavkem po komfortní a bezpečné jízdě automobilem vznikl koncept asistenčních parkovacích systémů, které mají řidiči poskytnout informace o žádaném parkovacím manévru a do jisté míry mu parkovací manévr usnadnit. Takovýto parkovací asistent může řidiči ušetřit čas, neboť má zajistit ideální nájezd do parkovací pozice na základě dat ze senzorů a parametrech vozidla, a tím pádem nemusí řidič při chybě opakovaně znovu zahajovat parkování a provádět úkon na více pokusů. Výhoda asistovaného parkování tkví i v tom, že řidič při jízdě např. s přívěsem nebo s problematickým výhledem získává větší jistotu o momentální situaci při parkování. Asistent kdykoli upozorní na blížící kolizi a dá řidiči najevo, co už si nemůže dovolit.

Trendem moderních automobilů je snaha nabídnout maximální bezpečnost i designově stylističtější vzhled. Tyto dva požadavky často vyžadují větší a silnější karoserie, které omezují viditelnost řidiče a tudíž zhoršují schopnost řidiče bezpečně zvládnout parkovací manévr.

V roce 2011 v průzkumu zadaném automobilovou společností Ford přiznala třetina evropských řidičů, že má s podélným parkováním problémy [1]. Mnozí ho dokonce označili za více stresující než návštěvu zubaře. Města jsou čím dál tím více pohlcena automobily a už jen najít místo na parkování obírá řidiče o spoustu času i nervů. Mnozí lidé, i když parkovací místo najdou, nedokáží odhadnout správný manévr a stresují se při opakovaném najíždění do parkovacího místa, navíc mohou mít i strach z okolních pozorovatelů, že budou vypadat v jejich očích špatně. Také je možné, že i při volném místě řidič usoudí, že se do parkovacího místa se svým vozem nevejde nebo pochybuje o svých jezdeckých schopnostech, a stráví dalších deset minut hledáním jiného parkovacího místa.

Toto všechno jsou možné důvody, proč stále roste zájem o nadstandardní parkovací vybavení, které řidiči spočítá vzdálenost od místa kolize, optimální natočení volantů a trajektorii manévru a které nabídne řidiči pocit jistoty a bezpečí.

### **3.1 CÍLE PRÁCE**

Tato práce si klade za cíl poskytnout informace o možnostech parkovacích systémů využívajících metod zpracování obrazu a tyto metody pro účely asistovaného parkování analyzovat a vyzkoušet. Budou zde popsány způsoby modifikace obrazu za účelem získání informací a použitelných dat z kamer. Součástí bude i zhotovení kamerového aparátu, na kterém bude některý z těchto přístupů vyzkoušen. Dále si práce klade za cíl nabídnout možné zpracování těchto informací pro využití v parkovacím asistentovi.

Práce se rozhodně nesnaží o zhotovení funkčního prototypu parkovacího asistenta ani se nesnaží vyvinout funkční a odladěný algoritmus pro zpracování obrazu. Nejde o robustní detektor překážek. Práce slouží spíše jako studie pro návrh parkovacího asistenta využívajícího zpracování obrazu a slouží jako návod, jak k takovému návrhu přistupovat a orientovat se v něm.

## 4 PARKOVACÍ SENZORY

Parkovací senzory už nejsou žádnou novinkou, jsou mnohdy součástí základní výbavy automobilu a patří mezi nejprodávanějších zařízení v automobilovém průmyslu [2]. Senzory se připevní nebo jsou již zabudované okolo karoserie, často u nárazníků, a detekují překážky či zjišťují jejich vzdálenosti od hran automobilu. Při přiblížení překážky signalizují řidiči hlasovým pípáním blížící se kolizi a při stálém zmenšování mezery od překážky přidávají na intenzitě. Patří mezi ně ultrazvuk, radar a kamery.

Některé pokročilejší výbavy parkovacích systémů využívají k signalizaci řidiči grafické rozhraní, kde zobrazují, která část automobilu se blíží překážce. To dodává jezdcu upřesněnou polohu, kde ke kolizi může dojít, a řidič tak může provádět složitější manévry s větší jistotou a přesností.



Obr. 1: Grafické rozhraní parkovacího asistenta [3]

### 4.1 ZPĚTNÉ KAMERY

Další možností jsou kamerové senzory. Kamera je umístěna vzadu na vozidle, mírně skloněna dolů, a přes grafické rozhraní na palubní desce zobrazuje pohled dozadu mnohem jasněji a praktičtěji než zpětná zrcátka. Nabízejí široký pohled za automobil, takže řidič ví přesně, kam zacouvá. Kromě samotného obrazu bývá často součástí i vykreslení plánované trajektorie vozidla v závislosti na aktuálním natočení kol. Tento kamerový způsob parkovacího asistenta bude dále řešen v této diplomové práci.

## 4.2 360° KAMERY

Mezi finančně nejnáročnější způsoby patří 360° kamera. Automobily mající tento systém mají po celém obvodu karoserie malé kamerky, které jsou dohromady schopny složit 2D obraz pohledu shora. Toto umožní jezdcí se rychle orientovat a odhadovat, jestli se do daného parkovacího místa vejde, či nikoliv. Např. BMW X5 Systém může i řidiči na grafickém rozhraní umožnit přepnutí na libovolnou kameru a získat pohled v jakémkoli směru [2].



Obr. 2: Umístění parkovacích kamer do všech směrů pohledu [4]

## 4.3 AUTONOMNÍ PARKOVACÍ ASISTENT

Syntéza parkovacích senzorů, kamer a inteligentního řízení představuje autonomního parkovacího asistenta. Systém dokáže najít místo a vyhodnotit velikost prázdného místa k zaparkování a s povolením řidiče, který řídí pouze rychlost manévru, samotný zaparkovat. Většina těchto parkovacích systémů zvládá oba manévry parkování (podélné i příčné). Existují i typy, kdy řidič nemusí v automobilu vůbec sedět a parkování umožňuje vzdáleně pomocí chytrého telefonu. Tyto systémy se mohou uplatnit v nadcházející éře autonomních vozidel, avšak každý řidič musí mít stále manévr pod kontrolou a vždy přebírá za parkovací manévr zodpovědnost.

## 5 METODY ZPRACOVÁNÍ OBRAZU

Digitální zpracování obrazu je dnes velice populárním odvětvím, jelikož poskytuje sofistikovanější obrazová data pro strojové vnímání, která jsou zároveň blíže slučitelná s lidskou interpretací okolního světa. Zpracování obrazu je výpočetně velmi náročné, ovšem oproti dřívějším dobám je dnes tento výpočetní výkon snadno dostupný a zároveň jsou dostupné open source moduly a knihovny [7], kde se digitální zpracování obrazu řeší. Toto odvětví se těší velké popularitě a vzniká velké množství startupů zaměřených právě na Image processing. [5]

Zpracování obrazu je technika obohacení, změny, transformace a dalších možností surového obrazu získaného z kamer a senzorů, která se používá denodenně v množství aplikacích. V klasické kategorizaci se jedná o obrazovou segmentaci, detekci význačných bodů, klasifikaci objektů i celých obrazů a vylepšení vlastností obrazu. Například jeden ze základních případů, se kterým se image processing potýká, je zpracování digitálního snímku kvůli redukci nebo odstranění šumu či dalších jevů nepatřících do obrazu, které vzniknou během formování obrazu v kameře nebo při transformacích.

Matematicky je obraz definovaný jako dvoudimenzionální funkce  $f(x,y)$ , kde jsou  $x$  a  $y$  na sebe kolmé souřadnice v rovině a jednotlivé body páru souřadnic mají amplitudu  $f$ , což představuje úroveň intenzity barvy šedi. Za předpokladu, že souřadnice  $x$ ,  $y$  a jejich funkcionální hodnoty  $f$  jsou konečné a diskrétní, hovoříme o tzv. digitálním obrazu. Digitální obraz je tedy zhotoven z konečného počtu elementů, které mají svoji danou polohu a hodnotu. Těmto elementům říkáme pixely. V šedém snímku mají pixely danou intenzitu pouze jednou hodnotou, barevný snímek je tvořen (dle modelu RGB používaného v počítačích) kombinací intenzit červené, zelené a modré barvy, tudíž jsou pixely popsány třemi hodnotami intenzit těchto barev. [5]

## **5.1 PŘEDZPRACOVÁNÍ OBRAZU**

Image preprocessing je předzpracování obrazu pro další použití. Jedná se většinou o data z obrazu získaná kamerou, kde se snažíme upravit geometrii a hodnoty světelné intenzity pixelů. Tyto chyby jsou napravovány pomocí vhodných matematických modelů, ať už deterministických či statistických. Tyto modely se aplikují za účelem zlepšení vizuálního dojmu obrazu nebo za účelem transformace obrazu do formy, která je vhodná pro lidské vnímání či vhodná pro strojovou interpretaci. Často obraz obdrženy z digitálních kamer, nebo třeba i ze satelitů, ztrácí kontrast a svoji světlost z důvodů limitovaných možností technických subsystémů v senzoru nebo z podmínek osvětlení. Obraz mívá i několik typů zkreslení šumem. Do metod v preprocessingu patří kontrastní vylepšení, zdůraznění hran v obraze, přebarvení pixelů, filtrování šumu, zostření a zesílení intenzity. Takto upravený obraz je pak vhodnější do obrazových analýz, zobrazení a v neposlední řadě do snazšího procesu detekce význačných bodů. Tyto metody preprocessing nezískávají dodatečné informace z obrazu, nýbrž jen umocňují charakteristiky obrazu.

### **5.1.1 Změna kontrastu**

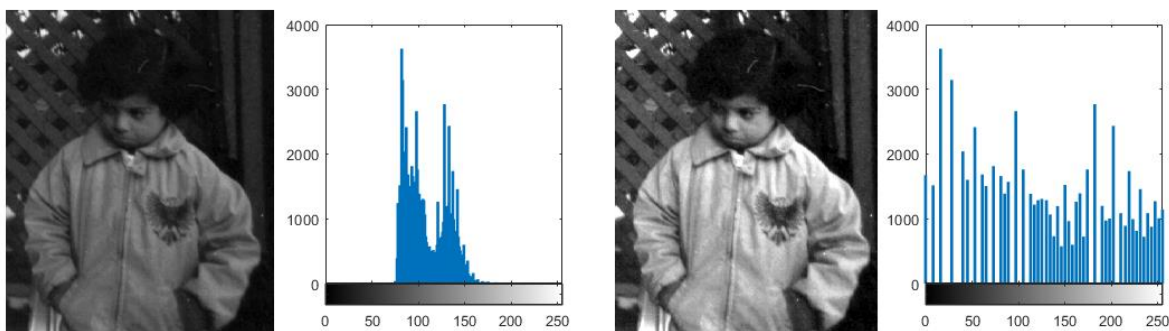
Některé typy snímků (např. voda, pouštní a sněžné oblasti, mraky, silnice apod.) jsou z hlediska barev homogenní, čili nemají moc rozdílů v intenzitách barev, jeví se všude stejně. Kdyby se takový snímek vložil do histogramu, byl by definován pouze úzkým pruhem hodnot zastoupení. Tato homogenita může vzniknout i chabým osvětlením scény. Z toho vyplývá, že výsledný obraz není lehce interpretovatelný, zdá se jednolitý. Kontrastní metody byly navrženy pro tyto časté případy homogenit, aby dynamicky roztáhly úzký interval intenzit do plného rozmezí spektra.

### **5.1.2 Filtrace šumu**

Filtrace šumu je používána pro odstranění nepodstatných, nedůležitých nebo i fiktivních informací z obrazu. Používají se různé specifických druhy filtrů, např. low pass, high pass, mean filtr a další.

### 5.1.3 Histogram

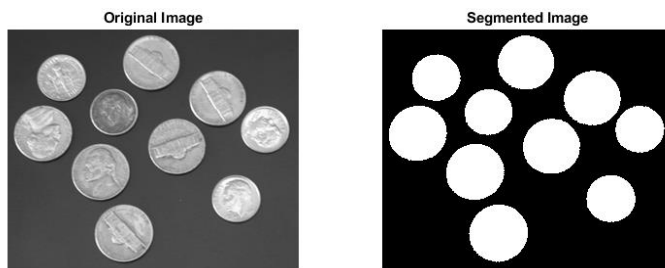
Histogram je často používaná prvotní operace v předzpracování obrazu. Reflektuje vlastnosti obrazu. V histogramu se dají lépe vypozorovat vlastnosti snímku, které pak můžou být na základě těchto dat upraveny. Jeden z častých případů použití histogramu je histogramová ekvalizace. Histogramová ekvalizace je nelineární operace, při které dojde k redistribuci hodnot pixelů tak, že v průměru vznikne obraz se stejným počtem pixelů v každé hodnotě hladiny intenzity. Ve výsledku tedy vznikne plochý histogram.



Obr. 3: Ukázka aplikace histogramové ekvalizace na příkladu z matlabu [6]

## 5.2 SEGMENTACE

Segmentace je jeden z klíčových postupů ve zpracování obrazu. Role segmentace je rozdělení obrazu na jednotlivé objekty a kompaktní části. Proces segmentace záleží na konkrétním problému a konkrétním žádaném objektu, měl by být ukončen, když je žádaný objekt vyizolován z obrazu. Např. při aplikaci v parkovacím asistentu, kde se zaměřujeme na automobily, je třeba nejprve segmentovat silnici z obrazu pryč a poté zredukovat obsah snímku na samotné zaparkované automobily.



Obr. 4: Ukázka segmentace na příkladu z matlabu [6]



Samotná segmentace se řídí podle hodnoty thresholdu, hranice k překonání. Po tresholdingu vznikne binární obraz, kde mají všechny pixely segmentovaných objektů hladinu intenzity jedna a všechny pixely pozadí mají nulu, čili objekty oddělí na promítací rovině bílou barvou a pozadí zůstane černé. Thresholding může být definován jako mapování obrazu do binární logiky:

$$S(x,y) = \begin{cases} 0, & \text{jesliže } g(x,y) < T(x,y) \\ 1, & \text{jesliže } g(x,y) \geq T(x,y) \end{cases} \quad (1)$$

kde  $S(x,y)$  jsou hodnoty segmentovaného obrazu,  $g(x,y)$  je hodnota intenzity barvy pixelů  $(x,y)$  a  $T(x,y)$  je hraniční hodnota thresholdu v místě  $(x,y)$ . V nejjednodušším případě je hodnota  $T(x,y)$  nezávislá na souřadnicích a je pevně daná pro celý obraz. Hodnota může být například zvolena na základě histogramu snímku odstínu šedi.

### 5.3 DETEKCE VÝZNAČNÝCH BODŮ

Metody pro detekci význačných bodů jsou používány pro extrahování vysokoúrovňových rysů za účelem klasifikace objektu. Význačné rysy jsou ty podobnosti, které unikátně popisují dané objekty, např. velikost, tvar, kompozice v obraze, umístění atd. Před použitím této metody může být aplikována segmentace pro izolování žádaného objektu ze scény pro snadnější měření rysů a význačných bodů. Kvantitativní popis těchto rysů pak může být použit pro klasifikaci a popis objektů či celého obrazu.

Tato část procesu zpracování obrazu je kritická, neboť má pozorovatelně největší vliv na úspěšnost a efektivitu rozpoznávacího systému, popřípadě systému na měření vzdáleností. Je to nejdůležitější faktor v dosažení vysoké kvality rozpoznávání v obraze. [5,7]

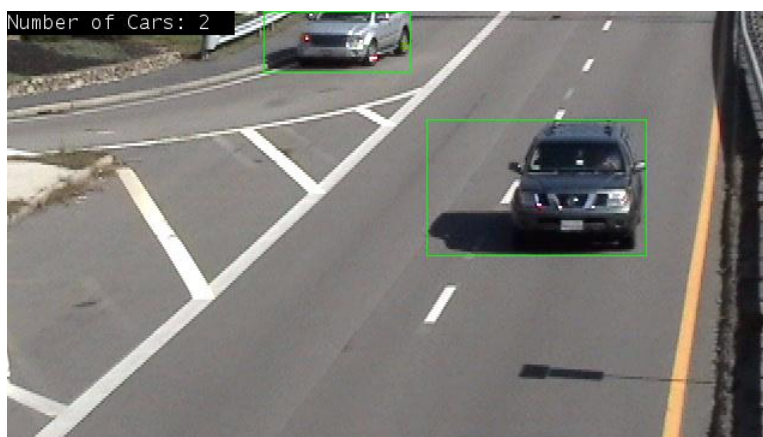
### 5.4 KLASIFIKACE OBRAZU

Klasifikace obrazu spočívá v přiřazování jednotlivých pixelů nebo skupin pixelů v závislosti na jejich hladině intenzity šedé barvy. Je to jedna z nejčastěji používaných metod pro získávání

informací z obrazu. Často je zapotřebí více obrazů objektu a detekování více typů rysů. Ke klasifikování objektů v obraze se v dnešní době mnohdy používají neuronové sítě a jejich složitější typ známý pod názvem “Deep learning”, kdy programátor musí naučit model správné detekci na trénovacích obrázcích.

V závislosti na interakci programátora a počítače během klasifikace objektů dělíme klasifikaci na dva typy: řízenou a neřízenou [8].

Při řízené klasifikaci rozhoduje člověk, kterou kategorii objektu přiřadí ke konkrétním pixelům a segmentům. Po konci procesu klasifikace musí supervizor projít výsledné datasety a ručně přiřadit špatně zvolené klasifikace pro daný objekt do správných korelací. U neřízené klasifikaci rozhoduje počítač, které třídy jsou obsaženy v obraze na základě statistických rozdílů ve spektrálních charakteristikách pixelů.

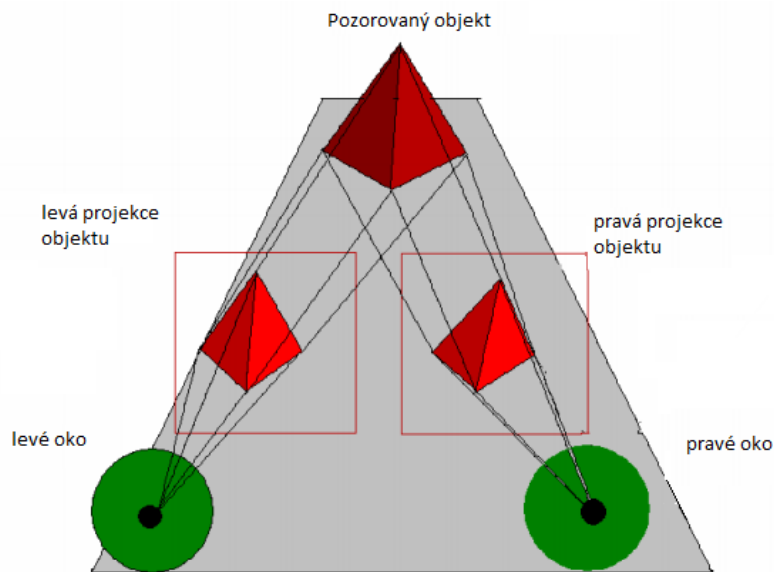


*Obr. 5: Ukázka detekce automobilů v příkladu z Matlabu [6]*

Všechny tyto metody zpracování obrazu slouží pro svoje účely a pomáhají extrakci informací o vyfocené scéně. Žádná z nich ovšem neposkytuje informaci o geometrické vzdálenosti objektů a jak daleko se objekt ve scéně nachází vůči kameře. Tato znalost je kritická pro detekování překážek a pro praktické využití je nutná i znalost vzdálenosti překážek, tedy znalost třetího rozměru v 3D souřadnicovém systému scény. Je tedy nutné najít metodu, která tyto informace dokáže zjistit a z vyfocených snímků je schopna rekonstruovat scénu a především určit reálnou vzdálenost objektů ve skutečném rozměru.

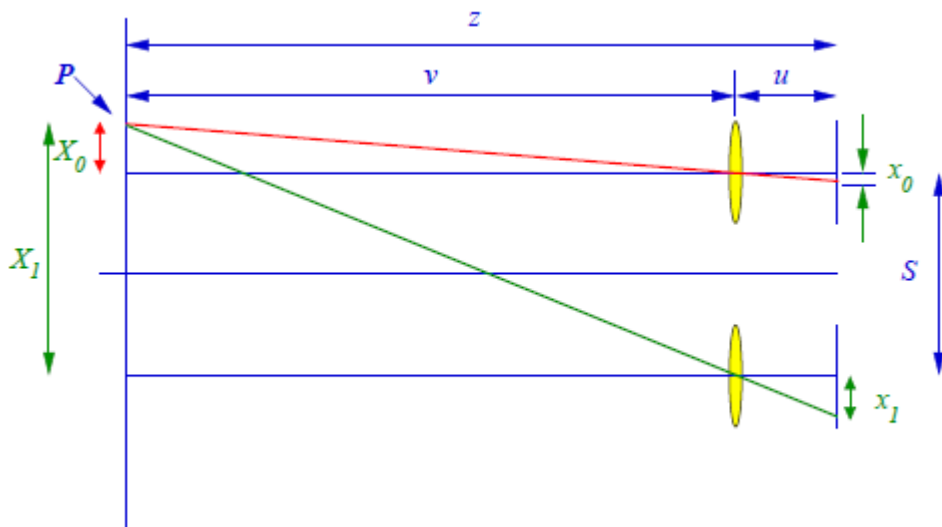
## 5.5 STEREO ZOBRAZENÍ

Metodou pro zjištění informace o hloubce obrazu, tedy vzdálenosti jednotlivých objektů je použití stereo vize. Tento přístup se používá u systémů s nutností znát vzdálenosti v ose obrazu, například v robotickém vidění, vyhledávání objektů a právě i v automatickém mapování. Ve stereovizi používáme dvě separované kamery, umístěné od sebe v konstantní vzdálenosti. Nejjednodušší případ je, že jsou optické osy obou kamer paralelně a jsou od sebe separovány vzdáleností, v tomto případě vzdáleností  $S$ . Při přesně paralelních osách, se ve dvou obrazech posouvají body pouze v horizontálním směru a z tohoto rozdílu můžeme získat informaci o hloubce. Tato metoda je ekvivalentní lidskému očnímu systému, kde má člověk dvě oči od sebe v konstantní vzdálenosti a je schopen přes dva pohledy složit troj-dimenzionální objekt a odhadovat vzdálenosti objektů (hloubku).



Obr. 6: Analogie s lidským způsobem vidění [10]

Vezměme stereosystém (viz obr. 7), kde předpokládáme paralelní optické osy, které jsou od sebe ve vzdálenosti  $S$ . Bod  $P$  se nachází ve vzdálenostech  $X_0$  a  $X_1$  od optických os kamer. Na promítací rovině, čili na vyobrazených fotografiích, je obraz bodu  $P$  ve vzdálenosti  $x_0$  a  $x_1$  od středu obrazu. Při těchto předpokladech z podobnosti trojúhelníků ze schématu vyplývá:



Obr. 7: Model stereoskopie za předpokladu dvou paralelních optických os [9]

$$\frac{x_0}{u} = \frac{X_0}{v} \quad (2)$$

$$\frac{x_1}{u} = \frac{X_1}{v} \quad (3)$$

kde  $u$  je vzdálenost každé čočky od detekovací roviny, senzoru, kterou předpokládáme stejnou pro obě kamery, neboť v tomto případě používáme stejné. Máme tedy souřadnice obrazu bodu

**P:  $x_0$  a  $x_1$ .**

$$x_1 = \frac{uX_1}{v} \quad (4)$$

$$x_0 = \frac{uX_0}{v} \quad (5)$$

Vzdálenost optických os známe, takže při předpokladu paralelních kamer vyplývá z geometrie:

$$X_1 = X_0 + S \quad (6)$$

A po substituci z (4) a (5):

$$x_1 = x_0 + \frac{uS}{v} \quad (7)$$

kde  $v$  je hloubková vzdálenost bodu  $P$  od čoček kamer, můžeme nyní vyjádřit  $v$  a získat informaci o hloubce obrazu:

$$v = \frac{uS}{(x_1 - x_0)} = \frac{uS}{\Delta x} \quad (8)$$

kde  $\Delta x$  je rozměrový rozdíl obrazu bodu  $P$  v dvou pořízených snímcích (možno počítat v pixelech).

Z fyzikálních rovnic pro čočku získáme:

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \quad (9)$$

Skoro v každém praktickém případě máme velikost vzdálenosti měřeného objektu v porovnání s ohniskovou vzdáleností mnohonásobně větší, tudíž vyplývá  $v \gg u$ . Poté lze učinit aproximace pro  $u$ , a to taková, že  $u \approx f$ . Lze tedy brát hodnotu vzdálenosti roviny detektoru jako ohniskovou vzdálenost čočky. Po aplikování této jednoduché aproximace vyplývá:

$$v \approx \frac{fS}{\Delta x} \quad (10)$$

Můžeme tedy z hodnot ohniskové vzdálenosti čočky a vzdáleností kamer od sebe, které jsou vždycky konstantní, měřit z  $\Delta x$ , což je rozdíl obrazů bodů mezi dvěma snímky, hloubkovou vzdálenost bodu **P**. Je tedy nutné identifikovat korespondující body na obou snímcích a z nich můžeme přes tyto geometrické vztahy vypočítat vzdálenost.

### 5.5.1 Chyba v měření vzdálenosti

Každé měření s sebou přináší možné odchylky a chyby. V obou měřeních v obraze jsme schopni odhadnout pozici konkrétního bodu **P** (jednoho pixelu) s určitou přesností. Když tedy označíme chybu  $\delta x$  v měření  $\Delta x$ , můžeme psát:

$$\Delta x = \Delta x_0 \pm \delta x \quad (11)$$

Definujeme ideální vzdálenost bodu **P** od čočky jako:

$$v_0 = \frac{uS}{\Delta x_0} \quad (12)$$

Potom kvůli chybě měření  $\Delta x_0$  vznikne chyba v měření vzdálenost  $\delta v$  a tedy:

$$v = v_0 \pm \delta v \quad (13)$$

$\delta v$  najdeme pomocí Taylorova rozkladu  $\delta x$  [9]:

$$\delta v = \frac{uS}{\Delta x_0^2} \delta x \quad (14)$$

Substitucí za  $\delta x_0$  z (12) získáme:

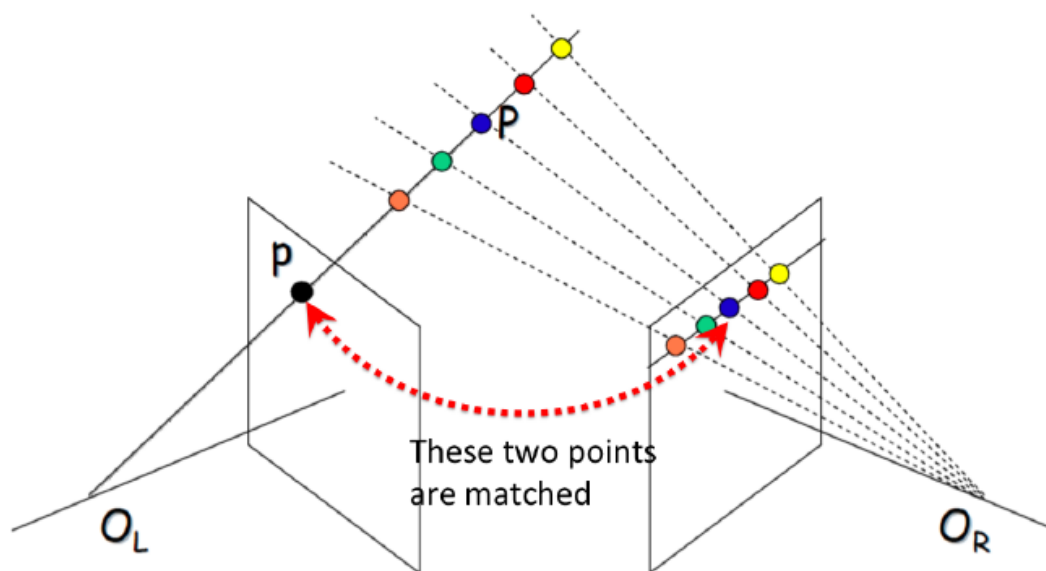
$$\delta v = \frac{v_0^2}{uS} \delta x \quad (15)$$

Což ukazuje, že pro zafixované kamery separované vzdáleností  $S$  a průměrnou chybu měření  $\delta x$  roste chyba měření hloubky s druhou mocninou vzdálenosti měřeného objektu. Takže k získání dobré přesnosti měření je teoreticky výhodné mít kamery od sebe ve větší vzdálenosti. (ovšem tento předpoklad se při praktické zkoušce neosvědčil, více v další kapitole). Pro vzdálené objekty bohužel klesá přesnost nenávratně.

V praktickém provedení pořídíme dva obrázky ze dvou kamer od sebe vzdálených délkou  $S$ , identifikujeme společné korespondující body mezi těmito dvěma obrázky a aplikujeme odvozenou triangulaci a získáme 3D souřadnice skutečných bodů. Celý tento postup ovšem kriticky záleží na zmíněné identifikaci bodů, problému korespondence.

### 5.5.2 Korespondence bodů

Nalezení stejných párů bodů v dvou obrazech, které tvoří 2D projekci stejného skutečného 3D bodu vypočítaného triangulací, se nazývá problém korespondence bodů. Při chybné korespondenci bodů může dojít k selhání algoritmu a špatné interpretaci scény. I při dostatečně optimalizovaném a robustním algoritmu se objevuje základní problém této metody a to ten, že ve stereo vizi může pozorovaný objekt vypadat ze dvou pohledů úplně jinak. Je tedy možné, že pro dva odlišné pohledy se body a korespondenční oblasti nebudou správně shodovat vůbec, nebo chybně. Existuje i další problém, kdy ve scéně s překážkou jedna kamera objekt zachytí a druhé překáží v zorném poli jiný objekt a tedy k žádné korelaci bodů nedojde.



Obr. 8: Princip promítání korespondujících bodů [11]

Tyto problémy jsou do jisté míry zmírněny snížením vzdálenosti obou kamer od sebe, jejich přiblížením. Ze vztahu (14) ovšem vyplývá, že tato kompenzace přináší sníženou přesnost určení hloubky v obraze. Z pořízených snímků a testování ale vzešel poznatek, že se nevyplatí zvyšovat separaci, neboť výrazně klesají přesnosti při práci s obrazem i přesnost kalibrace.





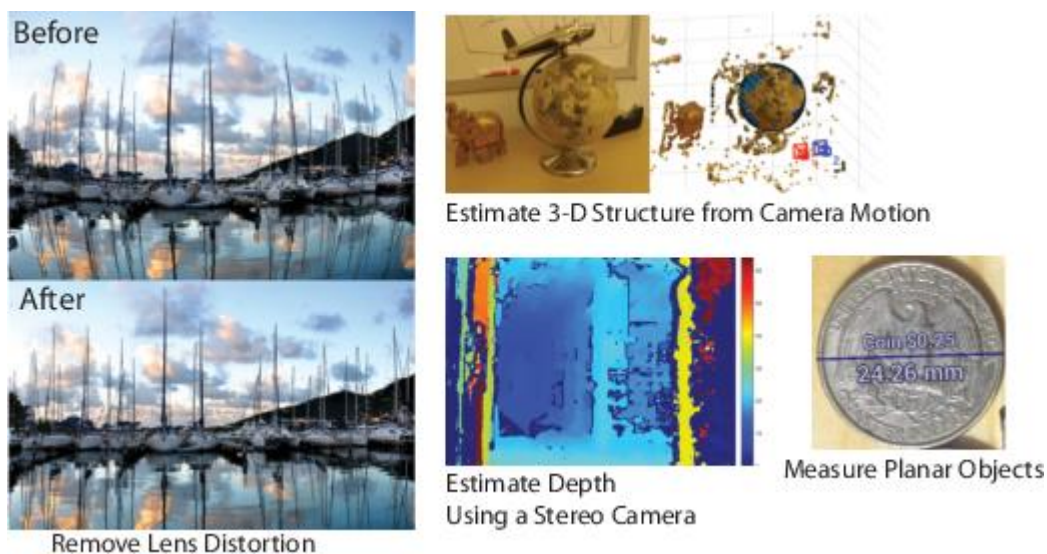
Obr. 9: Detekce bodů na epipolární přímce [11]

Na obr. 8 je schematicky znázorněn proces hledání možností stejných bodů ve dvou promítacích rovinách. Na obr. 9 lze ilustrovat hledání společných bodů na reálné fotografii. Při paralelním umístění kamer víme, že se bod promítnutý na jedné fotografii bude vyskytovat ve stejné výšce (stejná Y-ová souřadnice) i na druhé. Toto značí tyrkysová čára na pravém snímku. Pro obecnou polohu kamer se tato čára výskytu stejných pixelů nazývá epipolární přímka. Na levém snímku jsme tedy detekovali význačný bod, označený červeným kolečkem, snažíme se najít jeho protějšek (stejnou, popřípadě podobnou hodnotu barvy pixelu) na epipolární přímce v druhém snímku. Zkoušíme postupně všechna místa (záleží na typu použitého algoritmu) a porovnáváme hodnoty pixelů a hledáme co největší shodu. Zkoušená místa označují žluté čtverečky a modré kolečko označuje pixel s největší shodou, čili korespondující bod. Pro porovnání je znázorněno červeným kolečkem místo bodu z první fotografie. Vzdálenost mezi těmito souřadnicemi bude se vzdáleností objektů, tedy hloubkou obrazu, klesat.

## 6 KALIBRACE KAMERY

Znalost projekčních parametrů kamery je esenciální pro měření geometrických vzdáleností v počítačovém vidění. K zjištění použitelných informací z kamerových snímků pro rekonstrukci scény je tedy zásadní provést kalibraci kamery, nalezení interních a externích parametrů, které ovlivňují projekci obrazu. Použitá kamera funguje na principu *“pinhole”* modelu (temná komora) [12]. Obraz těchto kamer s sebou přináší výrazné zkreslení, které je třeba odstranit právě pomocí kamerové kalibrace. Toto zkreslení výsledného obrazu je konstantní a může být matematicky upraveno.

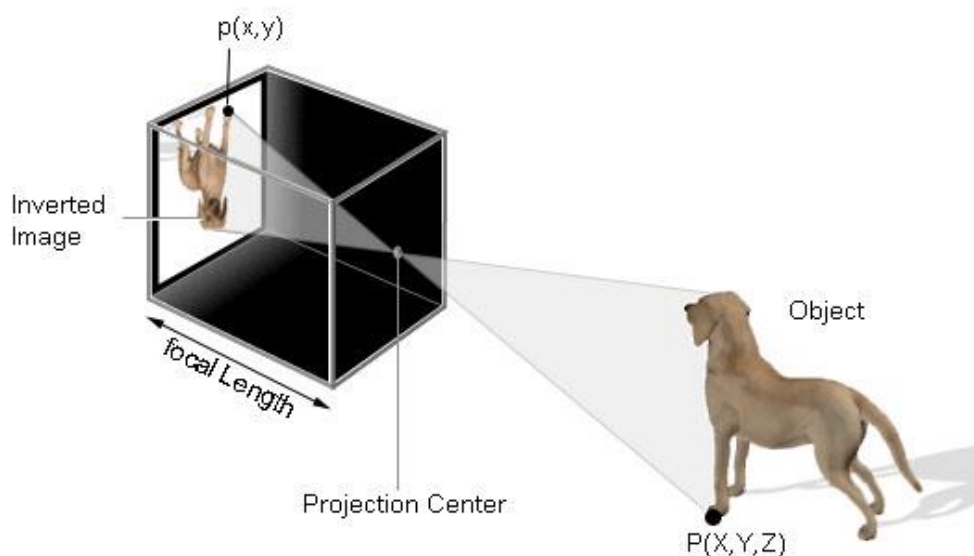
Důležitější ovšem je, že lze kalibrací určit vztah mezi měřítkem obrazu (pixelech) a skutečnou vzdáleností scény (milimetry, metry). Po kalibraci kamery je tedy možné ve velké přesnosti měřit z nafocených obrázků skutečné vzdálenosti vyfocených bodů a objektů. Tyto vzdálenosti pak využije parkovací asistent pro navigaci a plánování optimálních trajektorií a vytvoření mapy okolí automobilu a vyznačí překážky a volná místa.



Obr. 10: Příklady využití kalibrace kamery [6]

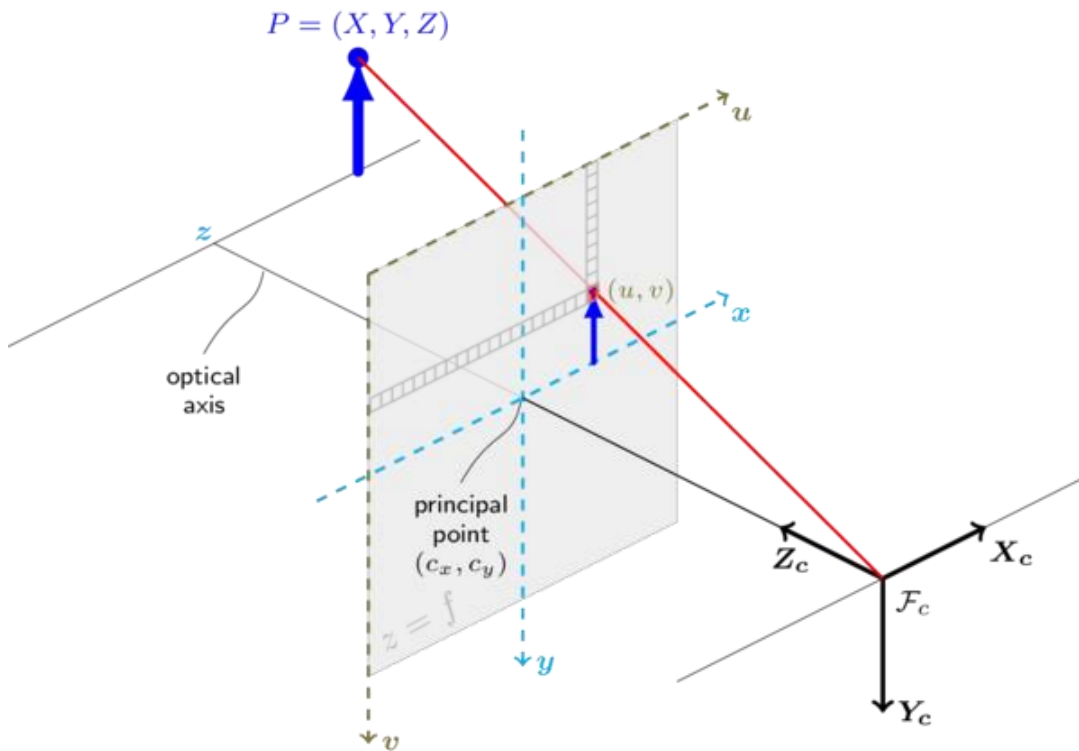
## 6.1 PIN HOLE MODEL

Při zachycení snímku na kameru v zásadě promítáme třídimenzionální scénu do dvourozměrného obrazu. Každý bod v reálném třírozměrném světě, každý proud fotonů, se při fotografování promítne v 2D průmětně obrazu. Tento pin hole model popisuje vztah mezi prostorem bodů ve skutečném světě a jejich projekcí do obrazu. Výrobce kamer se snaží co nejlépe simulovat ideální průmět komorou, ale vlivem technických nedokonalostí se objevují chyby typu radiální a tangenciální zkreslení, zašumění a další nepřesnosti.



Obr. 11: Znárodnění pin-hole modelu [13]

Parametry reálného modelu jsou reprezentovány tzv. camera matrix. Tato matice určuje relaci mezi mapováním 3D skutečných bodů ze scény do promítací roviny obrazu. Kalibrační algoritmus optimalizačně počítá tyto parametry a pomocí tzv. intrinsic a extrinsic parametrů, které fyzikálně popisují reálné vlastnosti kamery. Intrinsic parametry obsahují ohniskovou vzdálenost a pozici optické osy, která může být vychýlena a není vždy ve středu obrazu a extrinsic parametry popisují pozici kamery ve scéně. Obsahují matici rotace a translace, jejíž parametry popisují transformaci souřadnicového systému mezi promítací 2D rovinou a 3D scénou.



Obr. 12: Projekce pin-hole modelu a znázornění parametrů [14]

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (16), [14]$$

- $[X, Y, Z]$  souřadnice skutečného bodu ve scéně ve skutečném souřadnicovém systému v rozměrech délky [metry, milimetry]
- $[u, v]$  souřadnice promítnutého bodu v promítací rovině, v obraze v pixelech
- $[R \mid t]$  matice rotace a translace, parametry popisující transformaci v 3D systému
- $c_x, c_y$  souřadnice optické osy, většinou umístěny ve středu obrazu
- $f_x, f_y$  ohniskové vzdálenosti v pixelech

Intrinsické parametry, popisující optické parametry, vždy stejné pro danou kameru, pokud se nepřenastaví ostření.

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (17)$$

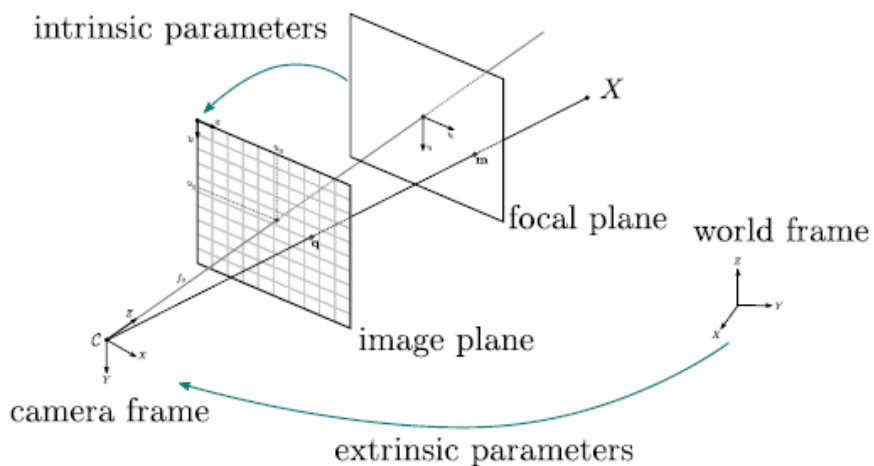
Lze z nich vypočítat i ohniskovou vzdálenost v milimetrech:

$$f_x = \frac{F}{p_x} \quad (18)$$

- $F$  Ohnisková vzdálenost v milimetrech
- $p_x$  počet pixelů na milimetr délky v obraze.

Extrinsické parametry, popisující polohu kamery ve scéně, mění se při změně polohy kamery.

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \quad (19)$$



Obr. 13: Schématické využití parametrů při transformaci obrazu [15]

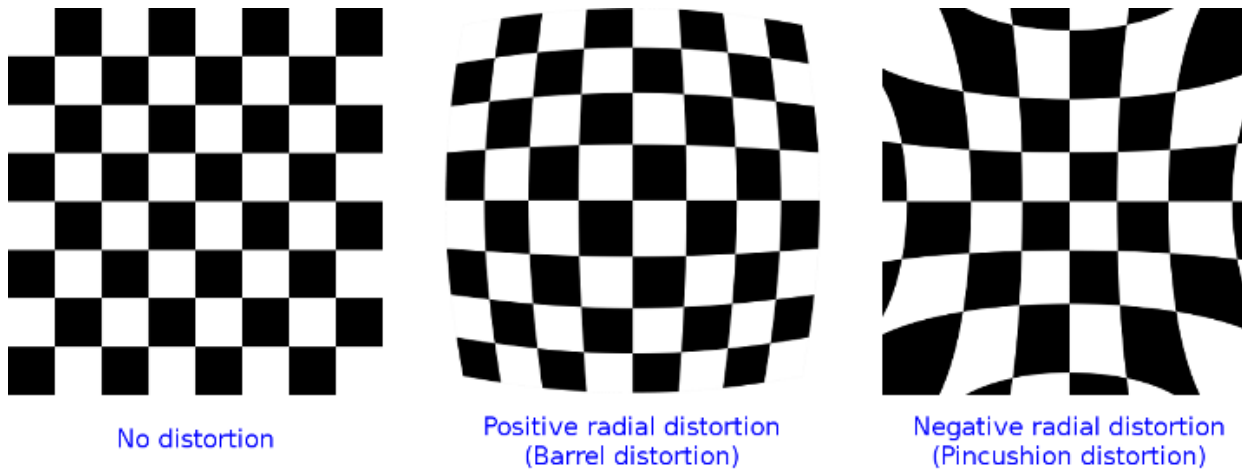
## 6.2 ZKRESLENÍ ČOČKOU [16]

Protože v reálném fotografování používáme v pin hole kameře čočku, musíme pracovat i se zkreslením, které je dáno technickou nedokonalostí čočky. Nejvíce se projeví radiální zkreslení, které závisí nelineárně pouze na vzdálenosti od středu obrazu. Ke korekci použijeme 2 parametry, které budou optimalizovány v kalibraci kamery.

$$x'' = x'(1 + k_1 r^2 + k_2 r^4) \quad (20)$$

$$y'' = y'(1 + k_1 r^2 + k_2 r^4) \quad (21)$$

- $r^2 = x'^2 + y'^2$
- $k_1, k_2$  koeficienty radiálního zkreslení



Obr. 14: Typy zkreslení obrazu – bez zkreslení, soudkovitost, poduškové [16]

*Poznámka: existuje i model tangenciálního zkreslení, ale my uvažujeme pouze dvouparametrové radiální, které je pro přesnost dostačující. Tangenciální je zanedbatelné.*

### 6.3 ZKALIBROVÁNÍ VLASTNÍ KAMERY

Skutečné projekční parametry kamery závisí na několika technických aspektech a nemusejí být výrobcem uváděny, navíc v případě ostření pomocí čočky, jsou parametry proměnlivé a závisí na aktuální konfiguraci. Způsobů kalibrace je více, zde bude rozebrána metoda „self-calibration“, vlastní zkalibrování. Z této metody budou získány následující parametry [17]:

- intristické parametry, vnitřní transformační parametry zahrnující ohniskovou vzdálenost a pozici optické osy
- parametry nelineárního zkreslení obrazu nedokonalou čočkou (lens distortion)
- extristické parametry, 3D rotace a translace pro každou polohu referenčního rastru

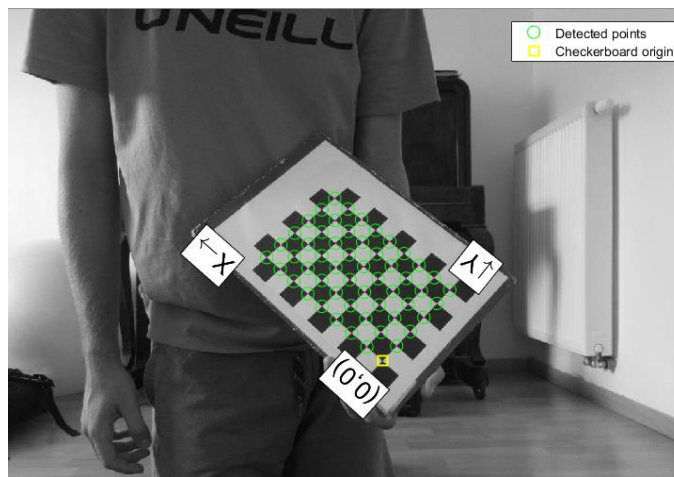
K odhadu parametrů kamery je třeba mít sadu identifikovaných skutečných 3D bodů a k tomu odpovídající 2D body zobrazené v obraze. Tyto korespondující body se dají získat několika snímky (doporučeno 30-50) kalibračních rastrů, nejčastěji černobílá šachovnice, u které se dá změřit délka čtverečků. Použitím těchto odpovídajících bodů lze optimalizačně získat parametry kamery a jejich přesnost se dá porovnat kritériem zpětné projekce „reprojection error“. Jedná se o zpětné dosazení bodů podle aktuálně vypočítaných parametrů a porovnají se vzdálenostní rozdíly oproti původně detekovaným v pixelech. Toto kritérium se snažíme minimalizovat a docílit přesné zpětné projekce. Průměrná přípustná chyba se pohybuje do jednoho pixelu, vyšší hodnoty naznačují nepřesnou kalibraci.

Nejprve pořídíme dostatečný počet (ideálně 30-50) snímků kalibračního rastru (šachovnice) pod různými úhly natočení a nejlépe ve vzdálenosti, kde předpokládáme oblast měření.



Obr. 15: Ukázka nafocených obrázků šachovnice při kalibraci

Z nafocených obrázků extrahujeme předpokládané skutečné body, u kterých známe souřadnice ve vhodně zvoleném souřadnicovém 3D systému a najdeme korespondující body v pixelech v pořízených snímcích. Souřadnicový systém zvolíme v rohu šachovnice a tedy všechny body šachovnice budou ležet v jedné rovině (eliminujeme Z-ovou souřadnici)



Obr. 16: Vložený souřadnicový systém v programu Matlab

Z naměřených bodů sestavíme homografické zobrazení, lineární vztah mezi skutečnými body a naměřenými na obraze z každého snímku. Pro body na šachovnici (kde budou všechny souřadnice ve směru Z nulové) bude model vypadat takto:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (22)$$

Kde součin matic intrinsických a extrinsických parametrů bude roven lineárnímu homografickému zobrazení **H**:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (23)$$

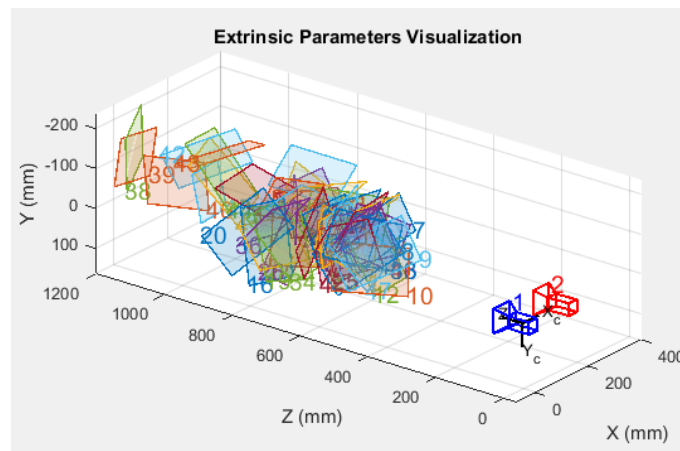


$\mathbf{H}$  má  $9-1 = 8$  stupňů volnosti (je invariantní z hlediska měřítka, jeden parametr zvolíme). Jeden pár korespondujících bodů dává 2 rovnice (souřadnice ve dvou směrech  $x, y$ ), tudíž potřebujeme alespoň 4 páry bodů k výpočtu homografie. Vypočtené body poslouží jako počáteční hodnoty optimalizace, kde bude kritériem rozdíl vzdáleností pixelů měřených a zpětně promítnutých korespondujících bodů definovaný vztahem (16) s aktuální hodnotou  $\mathbf{H}$ .

$$E_{proj} = \sum_{j=1}^N \left\| \underline{\mathbf{u}}_j - \mathbf{H} \cdot \underline{\mathbf{X}}_j \right\|^2 \quad (24), [16]$$

- $\underline{\mathbf{X}}_j$  vektor bodů skutečného souřadnicového systému šachovnice v milimetrech
- $\underline{\mathbf{u}}_j$  vektor souřadnic promítnutých bodů v promítací rovině v pixelech
- $\mathbf{H}$  matice homografického promítání
- $N$  Počet pořízených snímků

Z homografických zobrazení lze odhadnout intrinsické parametry s využitím ortonormality systému, Choleskyho faktorizace a optimalizace metodou nejmenších čtverců, detailní postup je k nahlédnutí v [16, 17]. Parametry jsou nalezeny nejprve bez aplikace koeficientů zkreslení chybou čočky, které se objeví až ve finální optimalizaci. Jakmile máme intrinsické parametry, získáme extrinsické 3D parametry rotace a translace pro každý snímek kalibrace zvlášť [17].

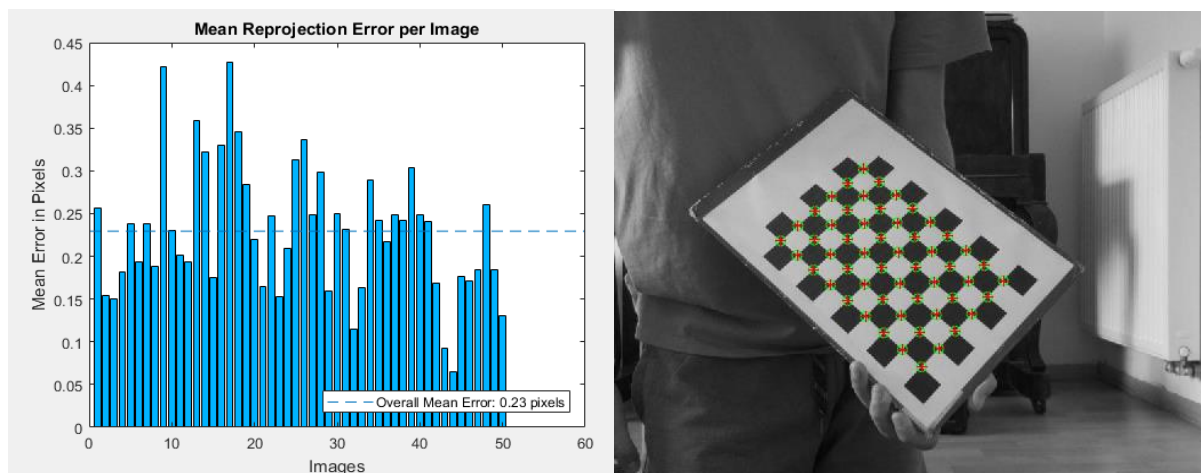


Obr. 17: Znárodnění extrinsických parametrů ve všech 50 snímcích kalibrace; detekovaná vzdálenost kamer od šachovnice

V posledním kroku použijeme všechny předešlé parametry, které byly lineárně spočteny a použijeme je jako výchozí hodnoty do nelineární optimalizace metodou nejmenších čtverců - Levenberg-Marquardt [6] přes všechny polohy kalibračního rastru a všechny pořízené fotografie. Jako u předešlé optimalizace i zde tvoří kritérium minimalizace zpětné projekce bodů.

$$\min_{(\mathbf{K}, \boldsymbol{\kappa}, \mathbf{R}_i, \mathbf{t}_i)} \sum_i \sum_j \|\mathbf{u}_{ij} - \hat{\mathbf{x}}(\mathbf{K}, \boldsymbol{\kappa}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{X}_{ij})\|^2 \quad (25), [16]$$

- $\mathbf{K}$  matice intrinsických parametrů
- $\boldsymbol{\kappa}$  vektor radiálních koeficientů zkreslení
- $\hat{\mathbf{x}}$  vektor zpětně promítnutých bodů po aplikaci kamerových parametrů
- $i$  Počet pořízených snímků
- $j$  označení páru



Obr. 18: Průměrná chyba zpětné projekce bodů po aplikaci parametrů; červený křížek – reprojekované body

Všechny výpočty byly prováděny v programu Matlab pomocí předdefinovaných funkcí zaměřených na kalibraci kamer a jsou k nalezení v příložených skriptech Matlabu včetně vypočtených kamerových parametrů. K zjištění parametrů lze využít i příslušný toolbox popsáný v [6].

## 7 HARDWARE

K vyzkoušení a testování algoritmů pro zpracování obrazu je nutné mít funkční a dostatečně přesný kamerový systém. Pro potřebu stereoskopie je tedy nutné mít dvě kamery fixně umístěné vedle sebe v konstantní vzdálenosti a mít paralelně seřizené optické osy. Pro potřeby této diplomové práce a popř. dalšího vývoje algoritmů vhodných pro použití v parkovacím asistentu byla postavena testovací soustava, která umožňuje pořídit poměrně kvalitní fotografie.

Jako kamery byly vybrány kamerové moduly pro vývojovou desku Raspberry Pi. Každá tato kamera pro přenos obrazu potřebuje být propojena kabelem CSI s deskou Raspberry Pi. Tato varianta byla zvolena jako vhodná pro svoji relativně nízkou cenovou kategorii a přitom stále poskytuje kvalitní obraz ke zpracování s možnostmi nastavení rozlišení a dalšími základními úpravami obrazu. Dále je možné implementovat algoritmy přímo do desky Raspberry a tudíž může systém běžet online. Raspberry Pi verze 3, která byla zakoupena, rovněž přináší výhodu připojení Wi-fi. To značně usnadní manipulaci, přenos dat a testování. Deska přes velkou náročnost zpracování dat v obraze zároveň poskytuje dostačující výpočetní výkon, který by se pro budoucí vývoj mohl použít pro online zpracování obrazu a tedy i do skutečného parkovacího asistenta. Raspberry Pi rovněž podporuje open source knihovny OpenCV v programovacím jazyce Python, které jsou vhodné pro autonomní implementaci parkovacího asistenta do vestavěného (embedded) systému. Desku je také možné programovat i v prostředí Matlab, za předpokladu licenčního přístupu do Raspberry toolboxu. Další důvod je i to, že Raspberry Pi původně vznikl jako studijní pomůcka pro vývoj a programování a existují tedy četné tutoriály a i přes dlouhé hodiny zkoušení a zjišťování problémů, je Raspberry Pi dostatečně přístupné. Posledním důvodem je i to, že součástí projektu, kterého se tato diplomová práce týká, je vývoj na testovacím modelu auta, u kterého se Raspberry Pi využívá, čili je důvodem i kompatibilita.

## 7.1 RASPBERRY PI [19]



Obr. 19: Vývojová deska Raspberry Pi a její kamerový modul [18]

### Raspberry Pi

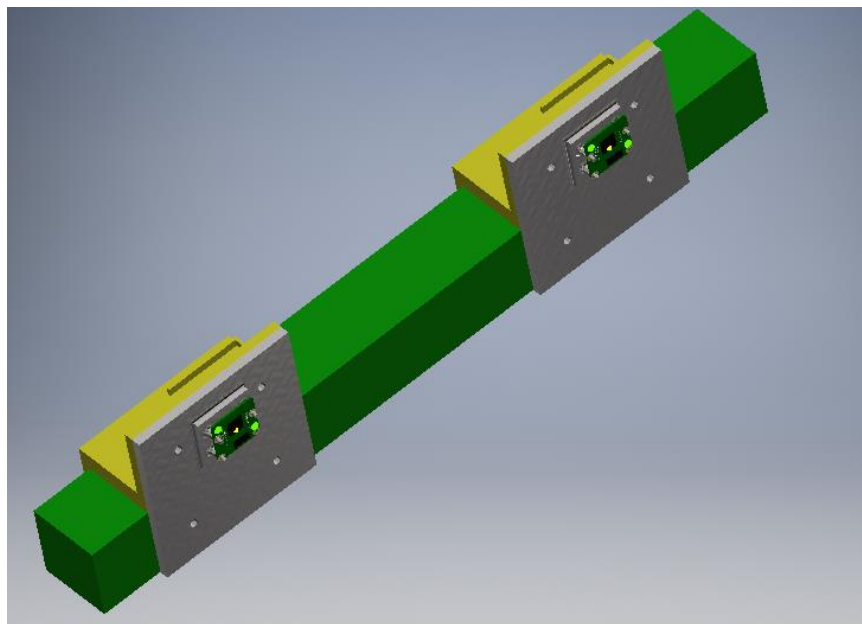
- Procesor (CPU): 1.2GHz 64-bitový čtyřjádrový ARM Cortex-A53
- Paměť (SDRAM): 1 GB (sdílená s GPU)
- USB 2.0 porty: 4
- Video výstup: HDMI (rev 1.3 & 1.4), 14 HDMI rozlišení od 640×350 do 1920×1200
- Video vstup: 15-pinový MIPI konektor kamerového rozhraní (CSI)
- Integrovaná síť: 10/100 Mbit/s Ethernet + WiFi a Bluetooth
- Nízkoúrovňové periferie: 17 x GPIO
- Interní paměť: MicroSDHC slot
- Rozměry: 85,60 mm × 56,5 mm
- hmotnost: 45 g

### Kamera

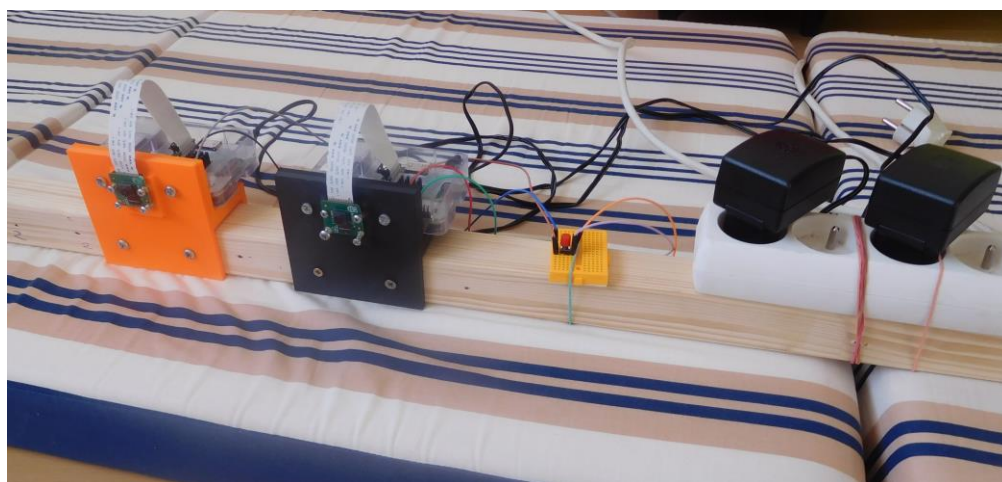
- Verze: 2
- IR filtr: ANO
- Čip: 8Mpx Sony IMX219 CCD
- Foto: až 3280x2464 px
- Video: 1080p @ 30fps; 720p @ 60 fps; 640x480p @ 90 fps
- Konektor: CSI
- Rozměry: 25 × 20 × 9 mm
- Montážních otvory: 4
- Velikost pixelu: 1.12  $\mu\text{m}$  x 1.12  $\mu\text{m}$

## 7.2 KONSTRUKCE

Úchyt pro kamery byl navržen v programu Autodesk Inventor (viz obr. 20) a vytištěn na 3D tiskárně. Bylo využito děr kamerových modulů, které byly nejprve nasazeny do trnů a pojištěny šroubem, ale poté kvůli přesnosti nastavení kamer byly přímo přivrtány k plastovému vytištěnému dílu a jemná kalibrace polohy byla doladována šroubem.



Obr. 20: Model sestavy konstrukce navržený v program Autodesk Inventor

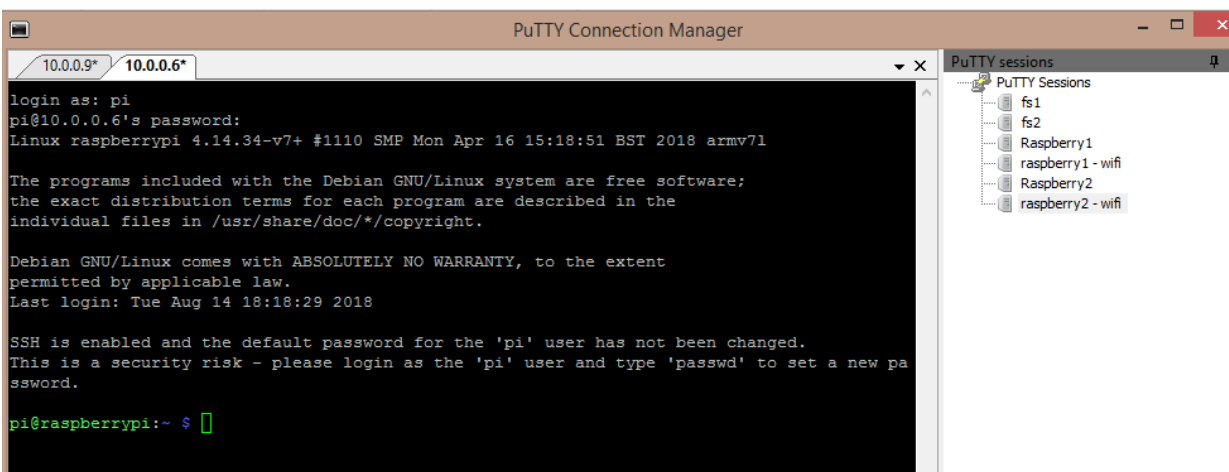


Obr. 21: Postavená měřicí soustava

Kamery a Raspberry jsou přivrtány na plastový úchyt, který je přišroubovaný na dřevěný hranol. Ke kompaktnímu provedení a možnosti libovolného přenášení je přiděláno na tyč i tlačítko pro okamžité pořízení snímku a 5V napájecí adaptéry určené konkrétně pro Raspberry.

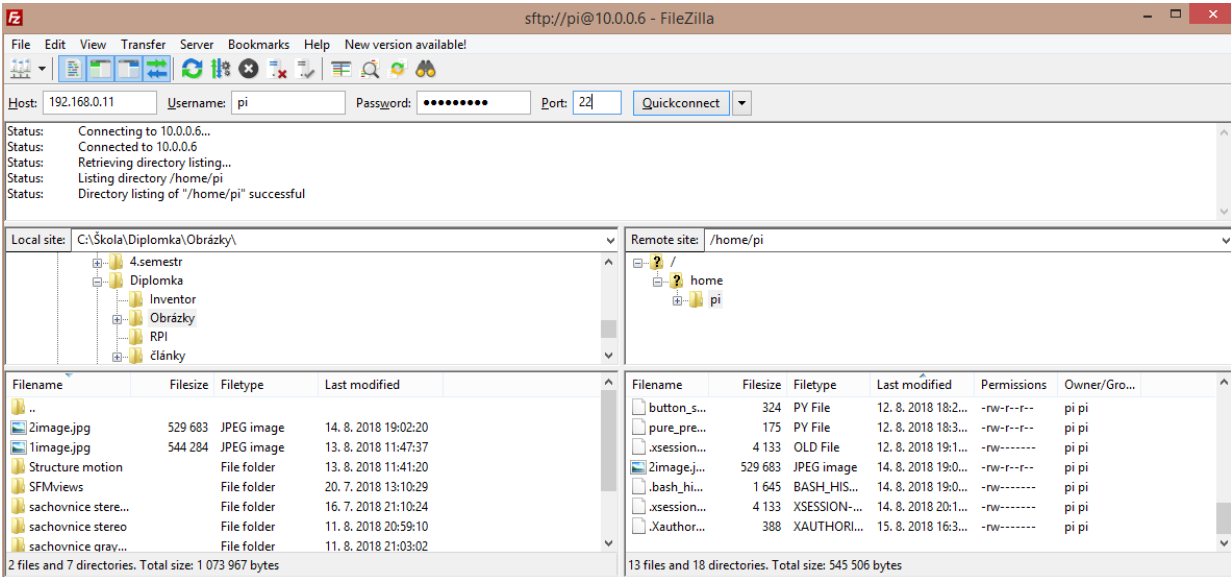
Pro přístup do mikropočítače Raspberry bylo nutné nejprve desku připojit k routeru a nastavit interní IP adresu v Raspberry a poté přes ethernet nebo Wi-fi zapisovat data a programy. Pro automatizované a pohodlné focení bylo nastaveno Wi-fi SSH připojení pomocí programů Putty a Filezilla. Tyto programy jsou k dispozici zdarma.

Pomocí programu Putty bylo Raspberry Pi programováno. Operační systém Raspberry Pi, tkz. Raspbian, je založený na open source operačním systému Linux, konkrétně na Debianu. Všechny příkazy měly tedy syntaxi jako v terminálovém prostředí Linuxu.



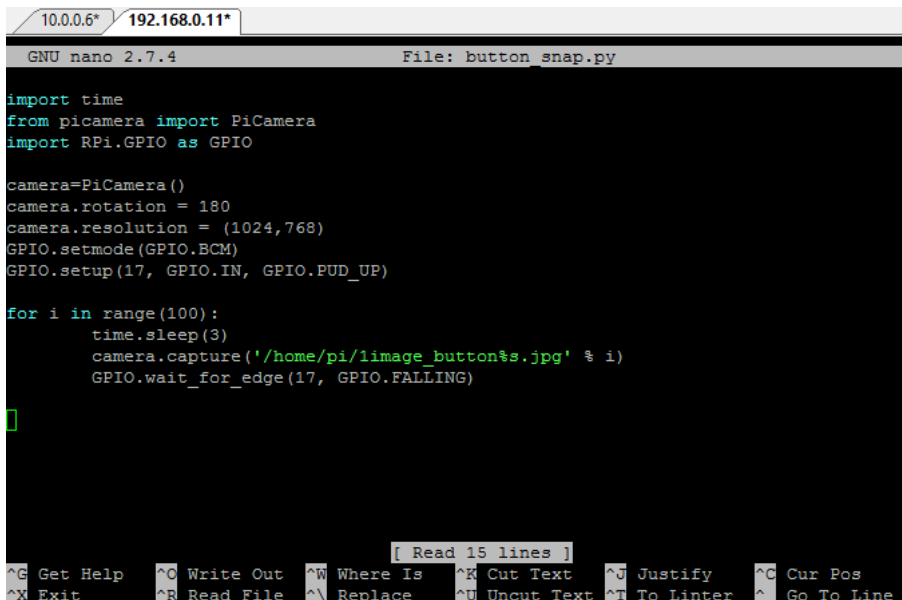
Obr. 22: Propojení s Raspberry v program Putty

Přenos souborů s fotografiemi probíhal přes již zmíněný program Filezilla, který vytvoří SSH tunel do Raspberry a po Wi-fi přenáší do počítače soubory s obrázky ke zpracování.



Obr. 23: Přenos souborů v program Filezilla mezi počítačem a Raspberry

V Raspberry byly napsány jednoduché programy pro pořízení fotek v programovacím jazyce Python a pro prohlížení aktuálně snímaného obrazu pro zaostření čočky. Jelikož jsou použity dvě kamery se dvěma boardy, bylo třeba sesynchronizovat spuštění kamer. To bylo provedeno přes jednoduchou logiku pomocí tlačítka, aby kamery vyfotily snímek ve stejný okamžik.



Obr. 24: Program pro vyfocení synchronizovaných fotografií při zmáčknutí tlačítka

## 8 MĚŘENÍ NA KAMEROVÉM SYSTÉMU

Jak již bylo řečeno, na základě poznatků bylo rozhodnuto, že nevhodnější metodou pro použití v parkovacím asistentu je stereo vidění, které přináší informaci o vzdálenosti objektů v obraze. Po zhotovení kamerového systému a následné kalibrace jsou již k dispozici všechny prvky pro zhotovení 3D mapy bodů scény. Nejprve byl algoritmus stereo vidění testován na zkušební scéně v interiéru (bez vozidel).

V této fázi byla často měněna konfigurace kamer, prováděna opakovaná kalibrace a zkoušena vzdálenost separace optických os. Při rozestupu kamer daleko od sebe ve vzdálenosti 430 mm byla odhadovaná hodnota kalibrace 460 mm a zjištěna špatná schopnost detekce význačných bodů. Při kamerách blízko u sebe (90 mm) byl odhad kalibrace správný, ale s obavou o chybu měření vzdálenosti vycházející z (15) byla vyzkoušena vzdálenost 144mm, která byla správně odhadnuta i kalibrací a zajišťovala uspokojivý odhad význačných bodů.

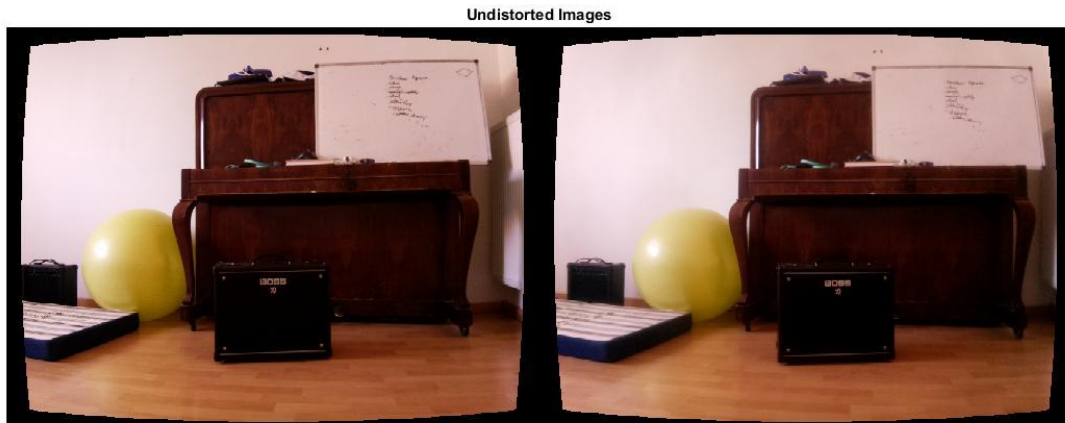
Navržený algoritmus byl vyzkoušen na následující scéně:



*Obr. 25: Scéna s míčem, zesilovačem, klavírem, s madrací a tabulí*

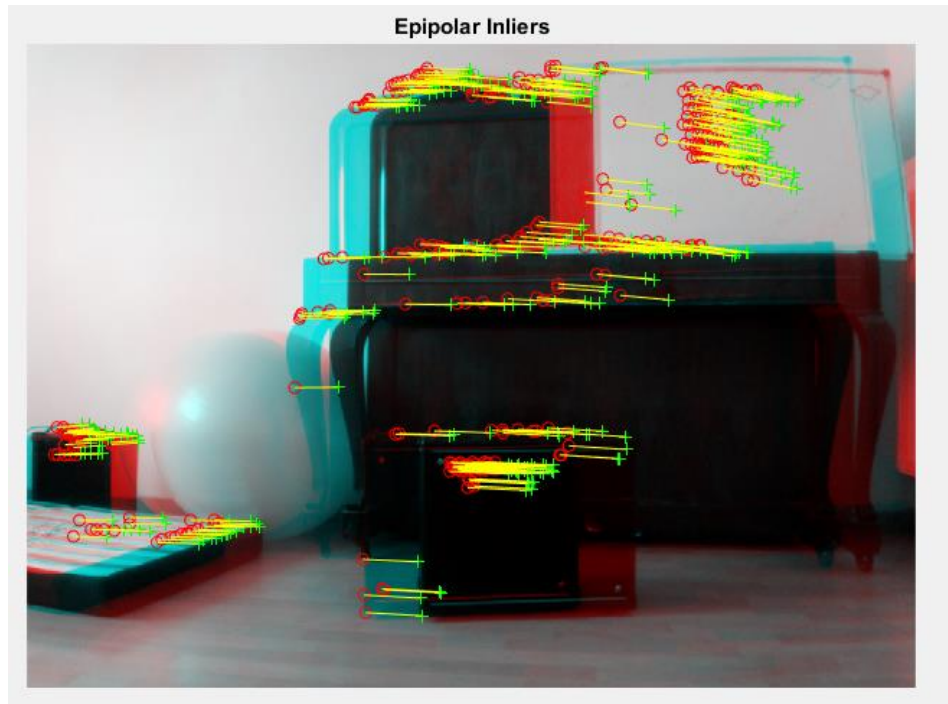


Ze všeho nejdříve je důležité odstranění zkreslení, zkreslení obrazu vlivem nedokonalosti optické čočky, pomocí parametrů kalibrace (viz kapitola 5.3). V tomto případě není změna tolik znát, ale při použití jiných kamer nebo jiném rozložení scény je nutné parametry zkreslení aplikovat.



Obr. 26: Scéna po aplikaci koeficientů zkreslení

Poté přichází nejdůležitější část celého procesu, detekování význačných bodů. Z těchto bodů určíme 3D souřadnice po epipolárních přímkách, které budou tvořit rekonstruovanou scénu. K detekci byly postupně testovány metody SURF - Speeded Up Robust Features [20] a metoda detekce rohů Shi-Tomasi, která využívá vlastní čísla [21]. Tyto metody se zaměřují na detekci hran, rohů a kontur. To je příhodné vzhledem k vizuální stránce automobilů, kde nemůžeme počítat s konkrétní barvou nebo s definovatelnými barevnými přechody a naopak vynikne členitost karoserie. Po testování na více fotografiích bylo rozhodnuto používat metodu minima vlastních čísel, Shi-Tomasi. Důvodem bylo subjektivní zhodnocení lepší přesnosti této metody, ovšem metoda SURF se zdála být taktéž použitelná. Po aplikaci detekce bodů vypadal obrázek následovně.



Obr. 27: Detekce význačných bodů nalezených v obou fotografiích

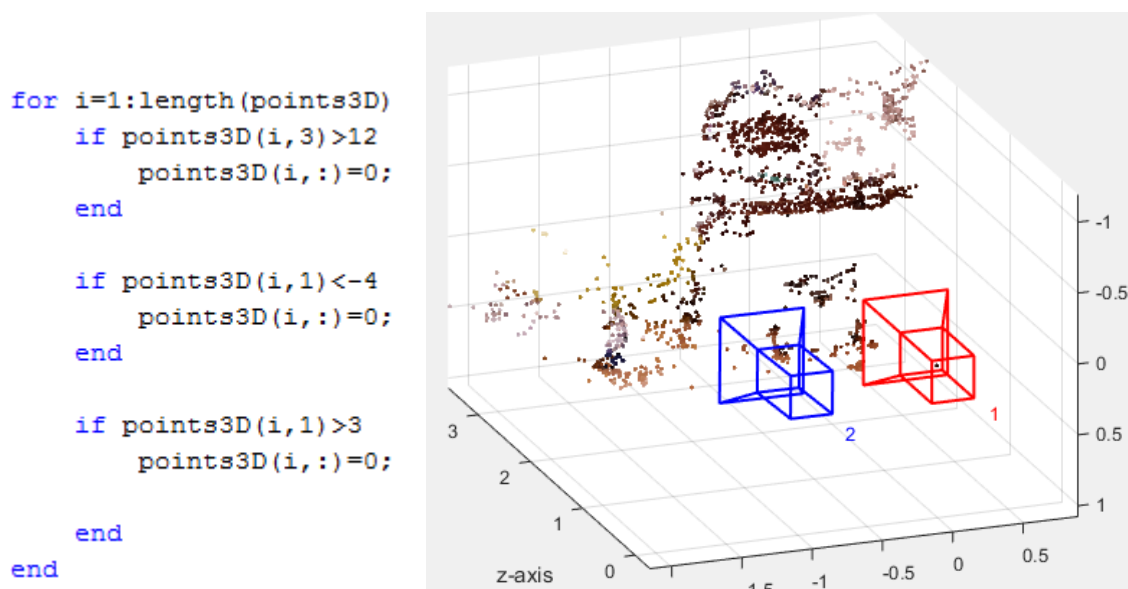
Tento obrázek se skládá z obou snímků dvou kamer, je možné všimnou si tyrkysového a červeného odstínu barev u jednotlivých objektů (podobně jako ve 3D promítání filmů). Tento vytvořený stereogram v sobě obsahuje označené body, které byly nalezeny v obou snímcích s velkou mírou jistoty a byly spárovány. Červené „kolečko“ označuje bod na obrázku z první kamery a je spojen s korespondujícím bodem z druhé kamery, který je označen zeleným „pluskem“. Tyto korespondující pixely použijeme k triangulaci k výpočtu vzdáleností dle (12).

```

%Triangulace
f=(cameraParams.FocalLength(1)+cameraParams.FocalLength(2))/2; % focal length [pixel]
S=0.144; % separace kamer [m]
delta_x=sqrt((matchedPoints2(:,1)-matchedPoints1(:,1)).^2 + ...
    (matchedPoints2(:,2)-matchedPoints1(:,2)).^2); % vzdálenost korespondujících bodů [pixel]
v=f*S./delta_x; % vzdálenost jednotlivých význačných bodů od kamer [metr]
f=cameraParams.FocalLength(1);
X0=(matchedPoints1(:,1)-cameraParams.ImageSize(2)/2).*v/f;
f=cameraParams.FocalLength(2);
Y0=(matchedPoints1(:,2)-cameraParams.ImageSize(1)/2).*v/f;
v=transpose(v);
points3D(:,3)=v;
points3D(:,1)=X0;
points3D(:,2)=Y0;

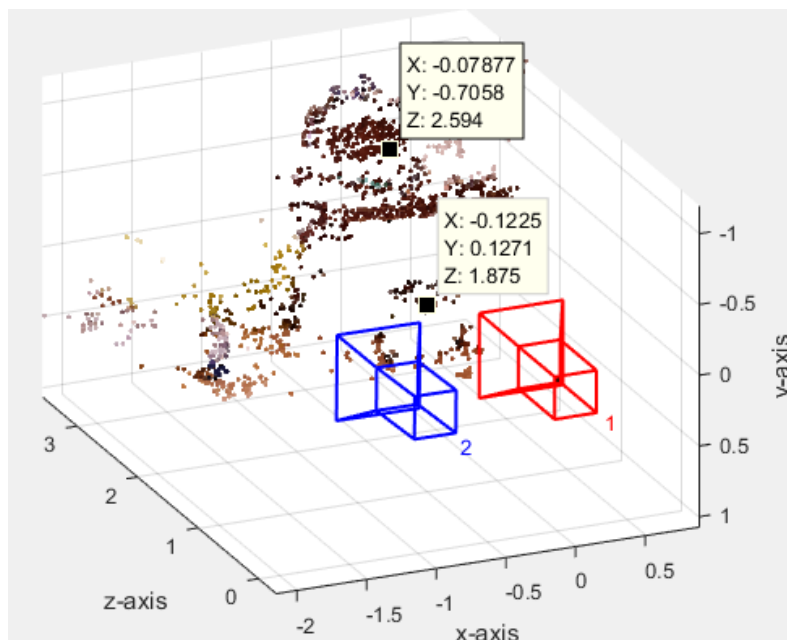
```

Výsledek byl zobrazen v 3D bodovém „Cloudu“ v Matlabu, kde bylo nutno „osekat“ osamocené pixely, které se chybou detekce zobrazily směrem k nekonečnu nebo mimo skutečnou scénu. K tomu byl použit následující algoritmus, který tyto body zobrazil do počátku souřadnicového systému a poté v dalším postupu body úplně eliminoval. Výsledná rekonstrukce scény má počátek souřadnicového systému v promítací rovině (prakticky měřeno od čočky kamery) první kamery, viz obr. 28. Vzdálenost objektů je určena souřadnicí Z a její velikost je uvedena v metrech. Zbylé souřadnice (X, Y) zatím nevyžadují míru přesnosti.



Obr. 28: Rekonstrukce scény zobrazená v matlabovském cloudu po redukci pixelů

V 3D souřadnicích byly odměřeny dva body a svinovacím metrem změřena jejich reálná vzdálenost od první kamery. Jedná se o bod vrchní hrany zesilovače (černá skříňka) a jeden konkrétní bod strany klavíru, viz obr. 25).



Obr. 29: 3D souřadnice měřených bodů

První bod na zesilovači má tedy vzdálenost 1.875 m a bod na klavíru 2.594 m. Naměřené hodnoty na zesilovači činí 1.70 metru a klavír 2.35 metru. Z toho vyplývá, že chyba měření vzdálenosti se pohybuje okolo 10% v tomto délkovém rozmezí. Z (15) vyplývá, že chyba s přibývajícím vzdáleností roste, ovšem odchylka 10% dává lepší výsledek, než bylo očekáváno.

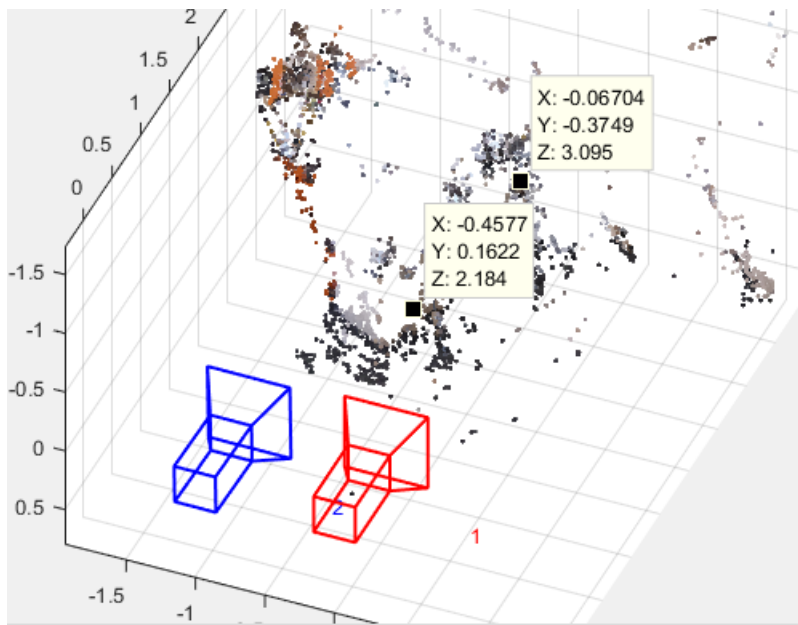
Vzhledem k parkování se s přibližující vzdáleností vozu k objektu bude chyba podle předpokladů zmenšovat a její význam zanikne. Důležitý je v tomto případě odhad vzdálenosti, zda se auto do místa vejde či nikoliv.

Před použitím na další testovací scénu byl kamerový aparát pomocí šroubů lépe vycentrován, aby se optické osy více vyrovnaly a snímky měly vyšší pravděpodobnost, že se korespondující body nacházejí na stejných úrovních souřadnic, a tím pádem byla větší šance na správnou detekci.



Obr. 30: Nová testovací scéna, kuchyně s linkou a barovými židlemi

Po použití Triangulace:



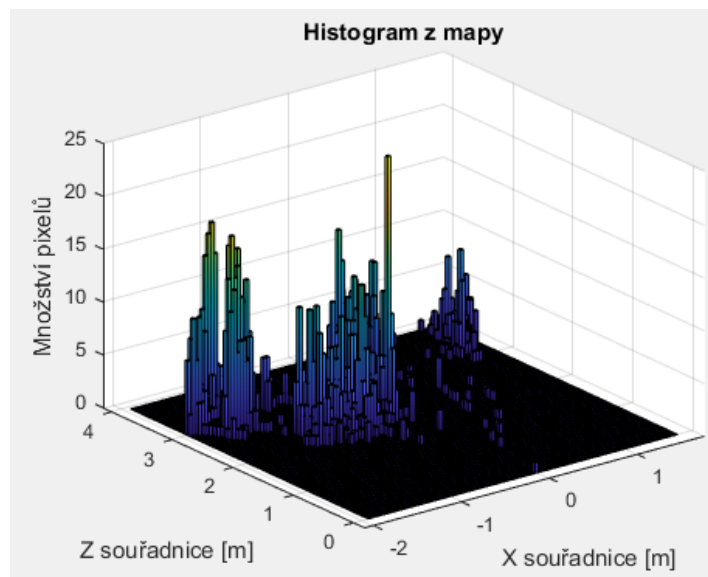
Obr. 31: Rekonstrukce scény kuchyně v 3D souřadnicích

Zde jsou objekty už pro lidské vnímání hůře detekovatelné. Barvy pixelů splývají a občas chybí význačné kontury, podle kterých by se dalo orientovat. V tomto případě by chtělo modifikovat obraz nebo použít vhodnější metodu pro extrakci význačných bodů. Jelikož nám ale nejde o tuto

konkrétní scénu, tak obraz upravovat nebudeme a najdeme konkrétní body k naměření, které dokážeme určit. Jedná se o dvě barové židle, které se nacházejí odděleně od barového stolu a můžeme je tedy identifikovat. Po provedení měření reálné vzdálenosti se vypočtená vzdálenost s drobnou odchylkou shodovala. To může být zapříčiněno zkorigováním kamer blíže k ideální konfiguraci nebo odchylkou měření v prospěch experimentu. Tento poznatek přesného měření je potěšující. Důležité je, že systém je v tomto rozmezí vzdáleností schopen určit odpovídající vzdálenost objektů.

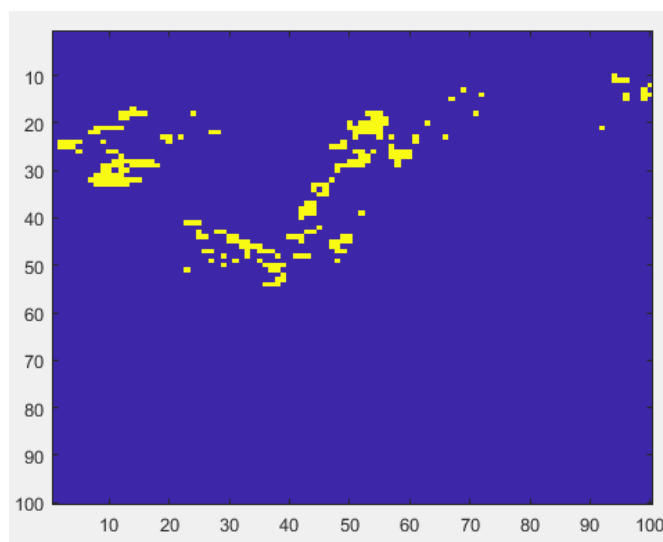
K využití pro parkovacího asistenta je potřeba zrekonstruované body vhodně interpretovat. Po vyzkoušení metod „Clusterování“, ohraničení bodů pro vytvoření pseudo-kompaktního objektu a histogramu, se ukázal nejvhodnější pro tuto chvíli onen 2D histogram. Idea je taková, že se na celou scénu podíváme „shora“, čili do průmětu roviny XZ. Tímto způsobem zanedbáme Y-ovou souřadnici, ale to lze s čistým vědomím připustit, neboť tato informace pro nás při reálném parkování není důležitá.

Souřadnice 2D histogramu rozdělíme pravidelně na určitý počet binů, které budou obsahovat promítnuté body ze svých přidělených souřadnic. Binem nazýváme statistický interval, do něhož ukládáme body histogramu. Počet binů v obou směrech (Z, X) bude roven 100 a ještě bude nastavena podmínka, která vylučuje body v binech, kde jich je velmi málo, méně než tři. To je z důvodu eliminace náhodných a chybných pixelů rozmístěných ve scéně, které by mohly tvořit fiktivní překážky.



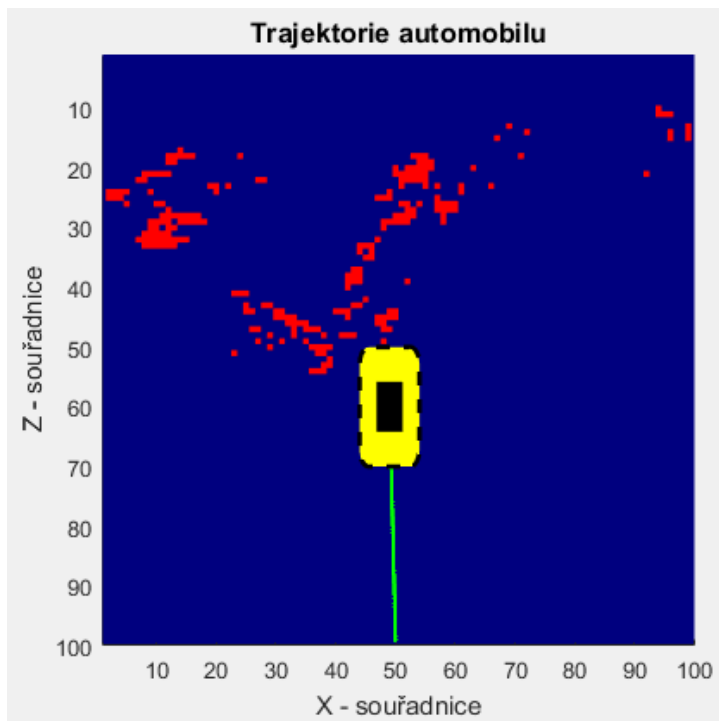
Obr. 32: Histogram zkonstruovaný ze získaných 3D souřadnic

Tento výsledný histogram evokuje překážky v podobě statistických sloupců. Průmětem do roviny ZX z něj lze vytvořit logickou mapu překážek, kde místo s alespoň nějakým počtem pixelů je rovno logické 1 a místo s prázdným binem je rovno logické 0. Tento postup mapy překážek se často používá v robotice a autonomním mapování. Výsledkem tedy vznikne čtvercová matice složená z jedniček a nul, která se graficky zobrazí takto:



Obr. 33: Logická mapa překážek

Na obr. 33 označuje žlutá barva elementy matice rovny jedné, překážky, a modré elementy matice rovny nule, čili prázdné místa. Matice má rozměr 100x100, což může odpovídat procentům z celkové velikosti scény. Přepočítání na metry se dá snadno zrealizovat. Tento způsob grafického zobrazení mapy už může být v parkovacím asistentovi použit pro snadné navádění. Lze do něj zakreslit trajektorii i graficky znázornit automobil, kde se právě vůči výchozí zmapované scéně nachází, viz obr 34.



Obr. 34: Logická mapa se zakreslenou trajektorií a automobilem

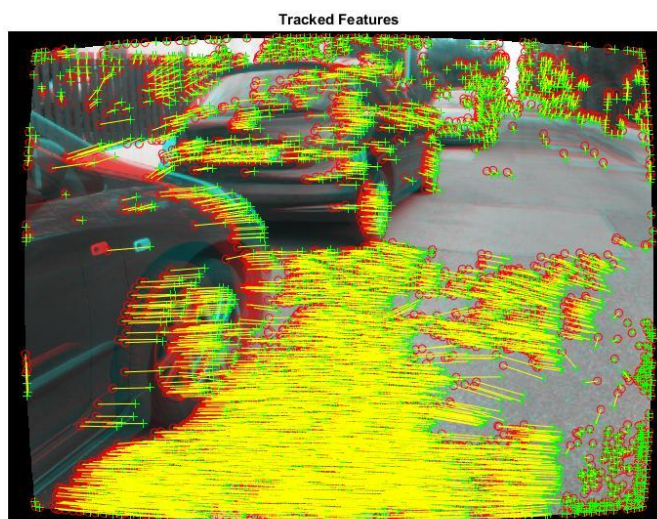
Na ilustračním obr. 34 jsou červeně znázorněny překážky a žlutý obdelník s černým středem značí auto, které jede po zelené trajektorii a právě detekuje před sebou překážku a automaticky zastavuje. Na tomto principu může být v parkovacím asistentu založeno grafické rozhraní pro řidiče a zároveň přehledné plánování trajektorie. Nynější trajektorie je vytvořena pomocí skriptu uvedeného v [22], který napsal Michael Kleder. Algoritmus je použit pouze z důvodu testování mapy a pro názornější ukázkou vykreslení trajektorie. Nenesou v sobě informace o křivosti nebo kinematickém modelu automobilu, je použit čistě ilustrativně. Skript neobsahuje nakreslený automobil.



Po těchto zkušebních případech a přípravě zobrazovacích algoritmů bylo zahájeno testování na skutečných automobilech v exteriéru. První situace simulovala nedostatečnou velikost parkovacího místa k podélnému zaparkování. Snímky byly vyfoceny ve směru projíždějícího auta, které má v zadní části umístěn parkovací asistent, jehož kamery snímají okolí a hledají volné místo k zaparkování.



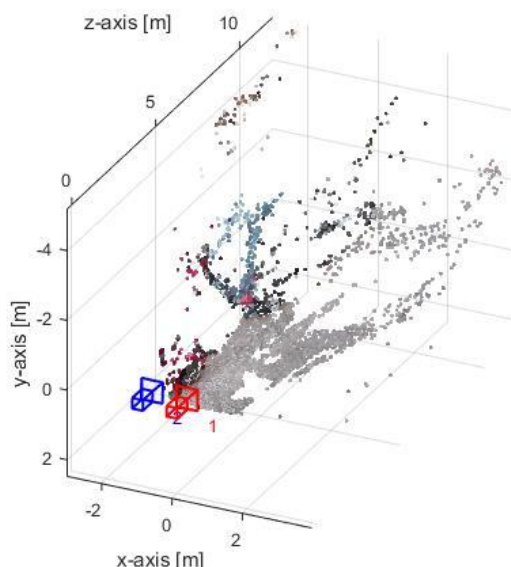
Obr. 35: Obraz dvou blízkých automobilů z obou kamer po odstranění soudkovitosti



Obr. 36: Detekce význačných bodů ve snímku

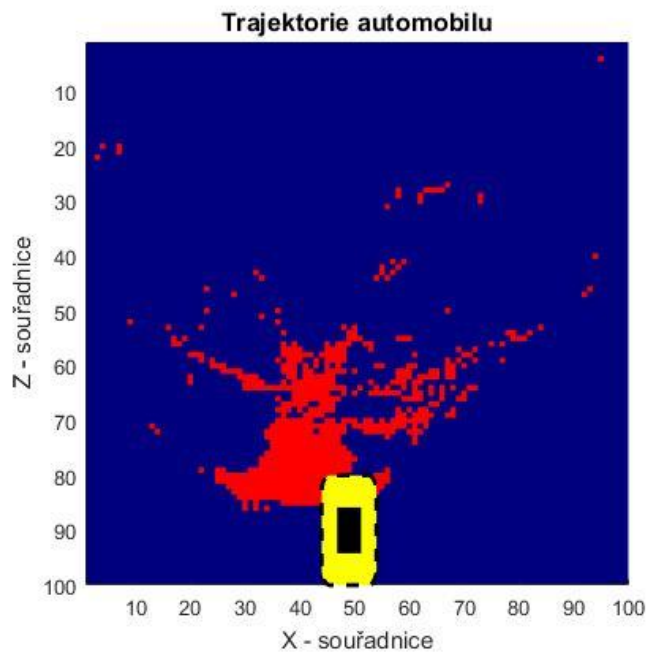
Jak si lze z obr. 36 povšimnout, v exteriéru nastávají další komplikace s pozadím scény. Největším problémem zde tvoří silnice, která obsahuje obrovský počet korespondujících bodů. I když by měly být body teoreticky na sobě nezávislé, tato přemíra bodů může být matoucí pro výpočty a další zpracování obrazu.

Tento snímek pochází z druhé sady fotografování, kde byla kamera namířena mírně směrem k zemi. V prvním sadě při kamerě rovnoběžně byl obdobný problém, kdy byla takto zaměřena ve velké míře obloha a stromy, které měly taktéž negativní efekt a v rekonstrukci splývaly se zadním automobilem, tedy kontaminovaly měřící data. Tento poznatek směřování kamer mírně směrem dolů se může ukázat jako vhodný pro realizaci parkovacího asistenta.



Obr. 37: Rekonstrukce scény dvou automobilů

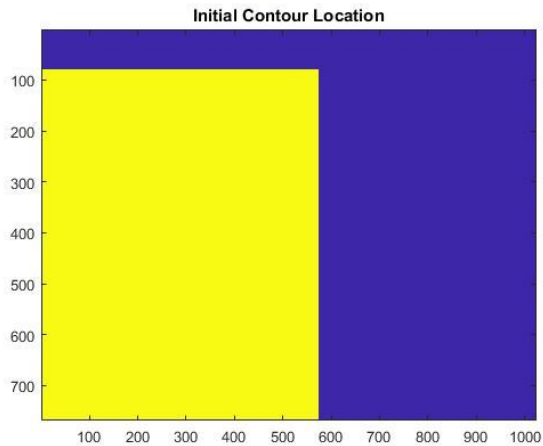
Jak je vidět z rekonstruované scény, kamery byly schopny zachytit kolo a přední část karoserie prvního vozu a zadní část druhého černého vozu. Nejmarkantněji jsou vidět zadní červená světla. Ovšem scéna obsahuje i dost šumu v podobně detekované silnice, která pohybu vozu samozřejmě nevádí, přesto je však vnímána jako překážka.



Obr. 38: Zašuměná logická mapa překážek

Při vykreslení mapy překážek z histogramu nelze určit, co je silnice a co reálná překážka. Nepomůže ani odstranění binů s minimálním počtem bodů, neboť hustota naměřené silnice je veliká a ztratili bychom i skutečné překážky. Je tedy třeba najít způsob, jak se nasnímané cesty zbavit.

Jako řešení byla zvolena modifikace obrazu umožňující zaměřit relevantní obsah obrazu a pracovat pouze s ním. Tento požadavek vede na segmentaci obrazu (viz Kapitola 4.2). Bylo nutné přijít s charakterizací, která nepostihne žádané objekty – automobily a která do maximální možné míry vynechá okolí. Toho bylo docíleno vytvořením masky pro aplikaci iterativního algoritmu aktivních kontur [6], který se snaží oddělit objekty v popředí snímku. Využilo se předpokladu, že zaparkovaná auta budou s největší pravděpodobností detekována po stranách silnice, tedy po stranách nasnímaného obrazu.



```

I1 = readimage(images, 1);
I2 = readimage(images, 2);

mask = zeros(size(I1));
mask(80:end,1:end-450) = 1;

bw1 = activecontour(I1,mask,200);
bw2 = activecontour(I2,mask,200);

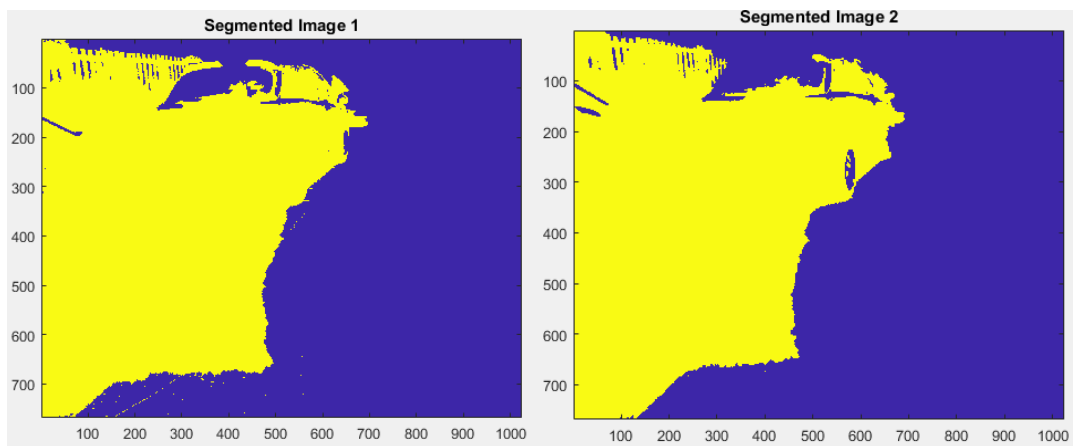
bw1=uint8(bw1);
bw2=uint8(bw2);

A1=I1.*bw1;
A2=I2.*bw2;

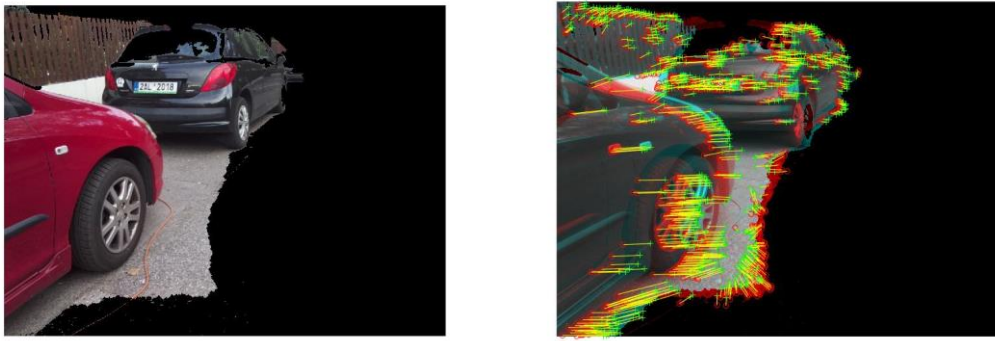
```

Obr. 39: Naprogramovaná maska pro segmentaci

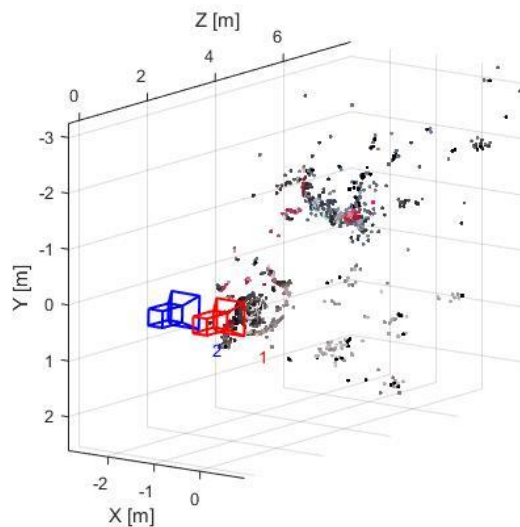
Vytvořila se maska dle obr. 39, kde žlutá barva představuje oblast plánované segmentace a fialová vynechanou oblast. Algoritmus přihlíží i k vynechané oblasti, ale neklade na ni důraz. Po aplikaci algoritmu vznikne segmentovaná část obrazu v podobě logické mapy, podle které nasnímáme původní snímek a znovu provedeme operace.



Obr. 40: Binární plocha segmentace obrazu

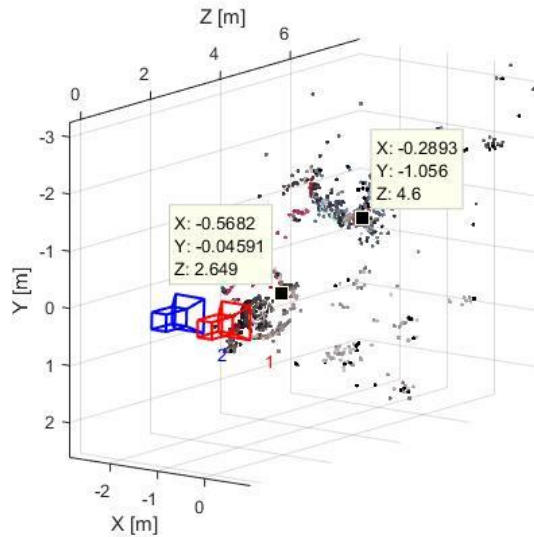


Obr. 41: Detekce význačných bodů segmentované části obrazu



Obr. 42: Rekonstrukce scény po segmentaci - bez detekované silnice

Jak si lze povšimnout z rekonstruované scény a z detekce význačných bodů, velká většina bodů tvořící silnici už neexistuje. Můžeme tedy přistoupit k ověření věrohodnosti získaných informací. Je třeba si také uvědomit, že druhé auto je značně posunuté ve směru Y. To je, jak již bylo zmíněno, zapříčiněno směrem optické osy kamer, protože obraz vztahujeme k počátku souřadnicového systému první kamery.



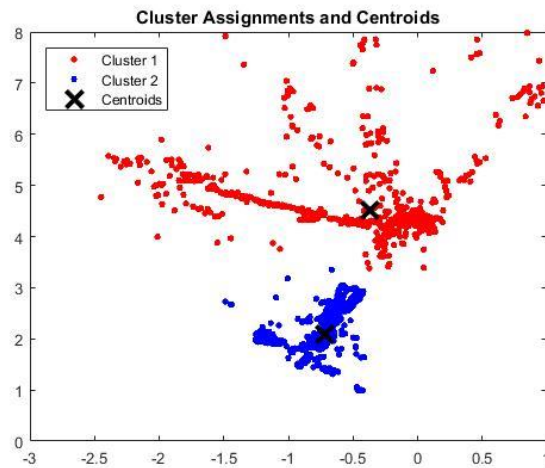
Obr. 43: 3D souřadnice měřených bodů v cloudu

Byly tedy nalezeny dva body, předek karoserie prvního automobilu a zadní světlo druhého, mezi nimiž byla změřena vzdálenost pásovým metrem. Tyto dva body představují rozestup dvou automobilů, čili určují velikost volného parkovacího místa. Při výpočtu této velikosti nestačí pouze porovnat hloubkové souřadnice bodů, nýbrž provést výpočet i z X-ové souřadnice. Postačí obyčejná Euklidovská vzdálenost dvou bodů v promítnuté rovině XZ.

$$m_e = \sqrt{(x_2 - x_1)^2 - (z_2 - z_1)^2} \quad (26)$$

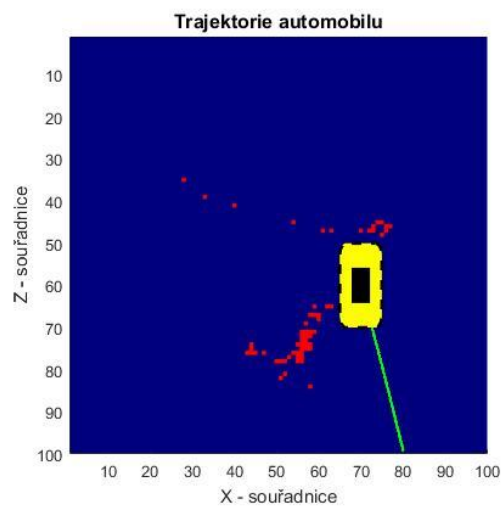
Reálná vzdálenost mezi těmito dvěma body byla naměřena přibližně 1.9 metru. Vypočítaná hodnota činí 1.926 metru. Vypočtená vzdálenost odpovídá poměrně velmi přesně a tedy odhad velikosti parkovacího místa se zdá věrohodný.

Z této informace získá automobil vzdálenost, kterou může porovnat se svými rozměry a při připočtené minimální bezpečné délce potřebné na provedení manévru vyhodnotí, že se do tohoto parkovacího místa nevejde. Pro automatické určení těchto bodů definujících velikost parkovacího místa může posloužit vytvořená logická mapa překážek, či metoda Clusterování bodů [6]. Při metodě Clusterování se označí jednotlivé body a v závislosti na počtu zadaných skupin se uskupí a vytvoří se pravděpodobný střed, centroid, objektu.



Obr. 44: 2D Clusterování automobilů – modrá barva značí červené auto, červená značí černé a okolí

Tato metoda by mohla sloužit pro kategorizaci a klasifikaci automobilů, ale vzhledem k nejistotě přiřazených bodů by vyžadovala speciální úpravy, aby byla vhodná pro parkovacího asistenta. Automatické označení středů může definovat objekt, nicméně nedostatečně ohraničený rozsah objektu způsobí nepřesný odhad velikosti překážky. Z tohoto důvodu bylo rozhodnuto, že je metoda nepoužitelná a použila se místo ní opět logická mapa překážek. Metoda je uváděna pouze informativně.



Obr. 45: Logická mapa překážek u sebe stojících automobilů

V tomto případě máme k dispozici na základě údajů z histogramu dva shluky bodů, první obsáhlejší značí obrys červeného auta a v shluk v horní polovině mapy označuje zadní světlo černého automobilu. Z této mapy se dá usoudit příliš malý prostor pro manévrování a zaparkování. Parkovací asistent by tedy nenavrhl toto místo k zaparkování.

Poznámka: Předmětem této diplomové práce není optimální výpočet trajektorie nebo skutečný kinematický model vozidla pro odpovídající měřítko a manévr. Obr 45 slouží k vytvoření překážek a zmapování oblasti, trajektorie a automobil jsou pouze ilustrativní. Představují způsob, jakým by tyto informace mohly být použity.

Druhou situací pro podélné zaparkování byl případ, kdy bylo místo dostatečně veliké a automobily daleko od sebe. Tento případ simuloval možnost zaparkování.



*Obr. 46: Nová scéna umožňující zaparkování*

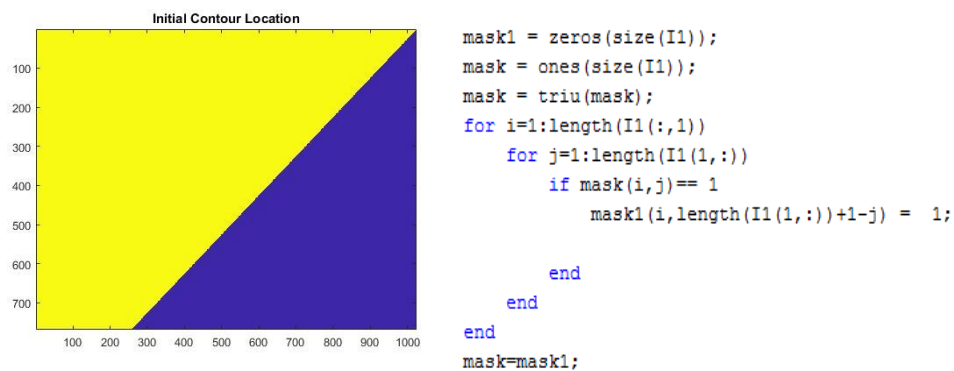


Aplikujme stejný postup jako v předchozím případě. V tomto případě ovšem ale zjistíme, že vytvořená maska nedostačuje k detekci bodů, neboť věrohodně nezasahuje do místa druhého automobilu, který je teď ve větší dálce.



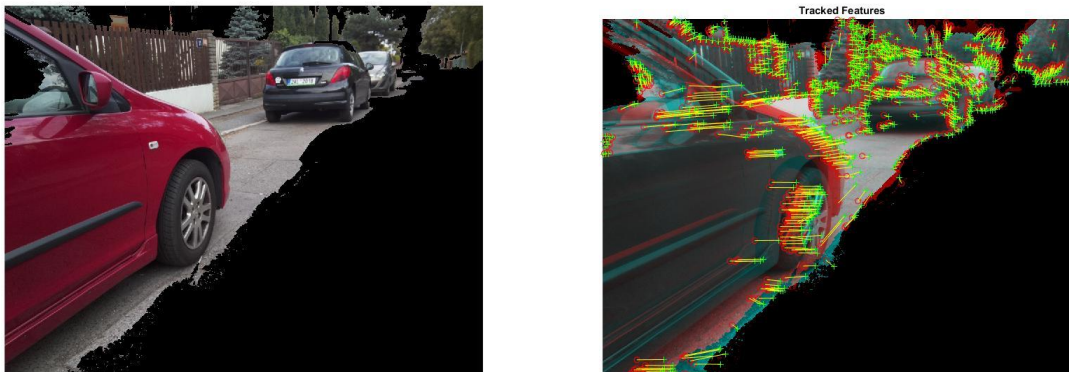
Obr. 47: Segmentace scény s automobily daleko od sebe

Aby se vyřešil tento problém, byla vytvořena nová maska s předpokladem, že maska dokáže zachytit celou stranu ulice. Byla tedy vytvořena maska zabírající horní část obrazu nad diagonálou, která je schopna detekovat i blízké části automobilu (např. kolo červeného automobilu) stojícího vedle kamer, bere větší část ve spodním levém rohu obrazu.



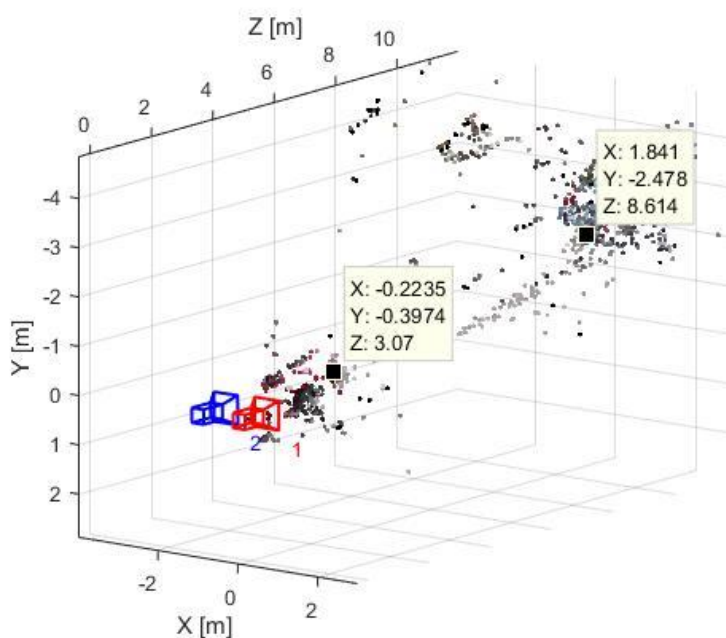
Obr. 48: Nová maska pro zachycení vzdálenější vozidel

Poté vypadala detekce význačných bodů takto:



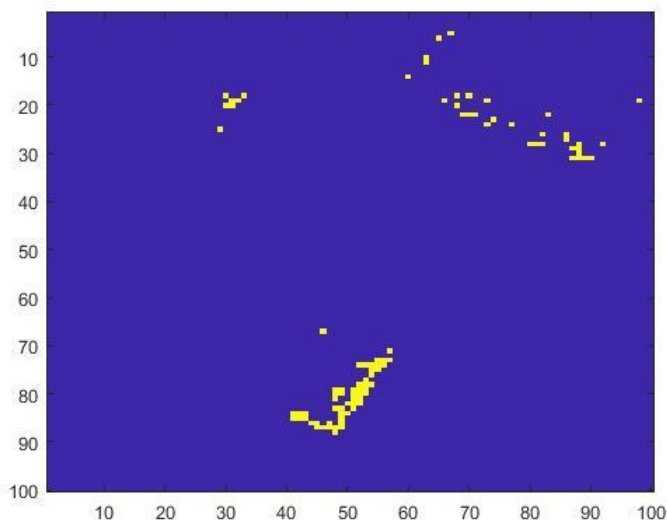
Obr. 49: Detekce bodů po aplikování nové masky

Následovala rekonstrukce scény a určení bodů, které budou změřeny. Opět bude vzdálenost měřena od přední části karoserie červeného automobilu a zadního světla černého automobilu. Tyto body zajišťují přesnost identifikace, dobré měření a zároveň definují parkovací místo.



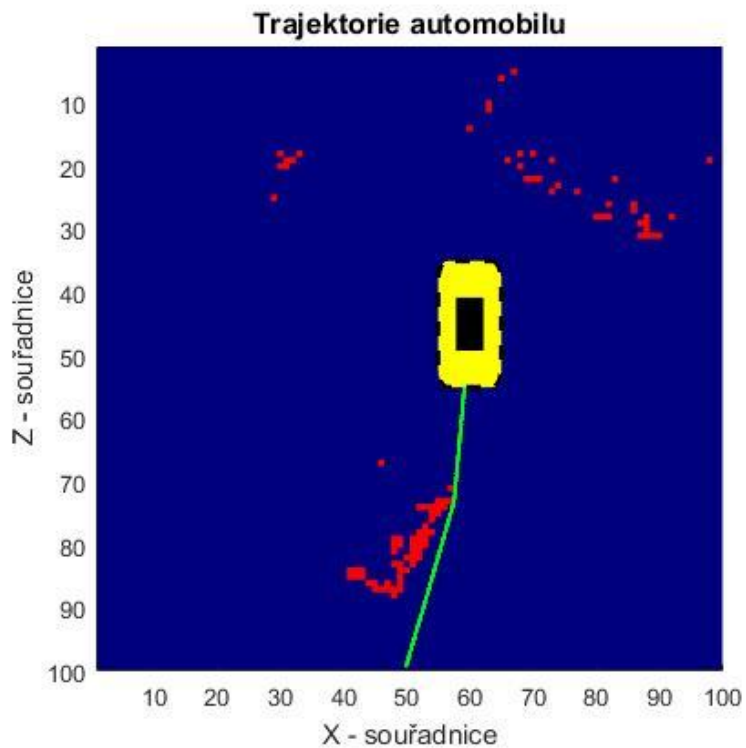
Obr. 50: Měřené souřadnice automobilů – předek karoserie a zadní světlo

Opět provedeme výpočet Euklidovské vzdálenosti (26) v rovině XZ, vypočtená vzdálenost je rovna 5.916 m. Oproti tomu je naměřená vzdálenost 6.8 m. To znamená, že chyba měření vzdálenosti vzrostla na 15 % a dle teorie se bude s rostoucí vzdáleností stále navyšovat. Opakovaná měření s jinými záběry toto znovu potvrdila. Lze tedy říct, že tato metoda ztrácí účinnost s rostoucí vzdáleností objektů. Otázka pro návrh parkovacího asistenta zní, jestli je tato přesnost dostačující nebo jestli není vhodné použít pro snímání větších vzdáleností další kamery zkalibrované do větších dálek popřípadě na delší vzdálenosti změnit algoritmus výpočtu, který by se jevil optimální.



Obr. 51: Logická mapa překážek parkovacího místa

Na obr. 51 je vidět logická mapa překážek. Spodní shluk bodů tvoří červené auto a v pravém horním rohu si lze povšimnout obrysu karoserie černého automobilu. Část bodů v levém rohu je detekovaný plot pozemku, který sice není přítomen přímo na silnici nebo na parkovišti, ale pořád tvoří překážku a hranici, kam by auto nemělo zajet. Jedná se tedy o relevantní obstrukci a nelze jej považovat za šum.



*Obr. 52: Ilustrace parkovacího manévru bez mechanismů křivosti, kinematiky a rozměrů vozidla*

Finální zobrazení mapy překážek a parkovacího manévru je vidět na obr. 52. Parkovací asistent vyhodnotil na základě měření vzdálenosti volné místo mezi dvěma skupinami bodů a vykreslil trajektorii k parkování. Znovu je nutné připomenout, že trajektorie a automobil neodpovídají podmínce křivosti, geometrii automobilu a kinematické schopnosti zatáčet, představují pouze grafické rozhraní.

## 9 ZÁVĚR

V rámci této diplomové práce byl poskytnut stručný přehled o typech parkovacích asistentů, používajících metod zpracování obrazu z kamer. Část těchto metod byla rozebrána a vyzkoušena v prostředí Matlab a pár z nich bylo použito při návrhu systému použitelného pro parkovacího asistenta.

Tento vytvořený kamerový systém sloužil k prozkoumání možností zpracování obrazu v detekci překážek a určování vzdáleností a vizuálního měření scény. Bylo rozhodnuto pro použití metody stereo vidění, neboť tento přístup na rozdíl od dalších metod nabízí možnost měření vzdáleností objektů a dokáže ze dvou obrazů dvou paralelních kamer rekonstruovat scénu do 3D souřadnic. Kamerový systém tvořily dvě kamery Raspberry Pi, které byly programovány přes Raspberry Pi vývojový mikropočítač v jazyce Python, který je zdarma a tedy nabízí možnost dalšího vývoje algoritmů na této platformě.

Metoda stereo vize byla schopna měřit vzdálenosti s uspokojivou přesností, avšak tato přesnost klesá s druhou mocninou vzdálenosti měřeného objektu. Pokles přesnosti se potvrdil v měření automobilu, který se nacházel 10 metrů od kamerového aparátu a odchylka přesnosti vzrostla na 15 %. Při přesnější justáži kamer, především srovnání optických os, byl vidět nárůst přesnosti, ovšem z principu metody nelze očekávat dobré výsledky ve vzdálenějších hloubkách scény. Teoreticky se dá přesnost ještě zlepšit zvýšením vzdálenosti obou kamer od sebe, bohužel ale tato kompenzace přináší ještě větší zhoršení kvality párování korespondujících detekovaných bodů, které se ukázaly pro přesnost důležitější. Reálným zlepšením do budoucna by se dalo docílit pořízením kamer s kvalitnější a přesnější optikou, které by byly vůči sobě přesně fixovány. Další možností je použití dalších kamer zkalibrovaných ve větších hloubkách, které by mohly používat optimálnější algoritmy pro delší vzdálenosti.

Co se týče obrazu, podařilo se vysegmentovat nežádoucí vlivy z pozadí. Vzhledem k zmenšení oblasti obrazu po segmentaci bylo docíleno i snížení výpočetní náročnosti následujících algoritmů, neboť je stačilo aplikovat pouze na pozůstalé pixely. Toto i snížení rozlišení může dramaticky usnadnit online provoz parkovacího asistenta z hlediska výpočetního výkonu. Vytvořená maska pro segmentaci byla založena na praktickém předpokladu, že automobily stojí zaparkované po stranách ulice. Může ovšem vzniknout situace, kdy si řidič nevšimne automobilu stojícího přímo za ním a hrozilo by, že do něj při couvání nabourá. Vytvořená maska obsahuje i část středu snímaného obrazu a při bližší vzdálenosti by měla postihnout i stojící automobil na silnici. Tato teorie by ovšem vyžadovala další testování a míra takovéto robustnosti nebyla cílem této diplomové práce.

Výsledná data byla převedena do formy 2D mapy pohledu shora, která simuluje grafické rozhraní, které by mohl řidič používat k orientaci a k čtení trajektorie a požadovaného natočení volantu. Tato mapa překážek by sloužila i k samotnému výpočtu optimální trajektorie a optimálního parkovacího manévru. Do této mapy by bylo dále vhodné naitovat geometrické obrazce, odpovídající shlukům pixelů. Tyto obrazce by pak tvořily kompaktní překážky místo jednotlivých bodů, to by potom vyžadovalo složitější algoritmy při zachování věrohodnosti.

Vytyčené cíle byly splněny. K návrhu prototypu parkovacího asistenta považuji využití strojového stereo vidění za vhodné. Pro samostatně funkční jednotku by bylo důležité naprogramovat zpracování obrazu takovým způsobem, že by asistent z nasnímaných obrazů postupně skládal celou scénu dohromady a nefungoval by pouze na aktuálních záběrech. Při propojení s řídicí jednotkou vozidla pro získání informací o rychlosti a natočení volantu a při rekonstrukci celkové scény, by se dalo lépe určit, kdy a jakou udělal uživatel chybu a dalo by se ho navést zpátky na správnou trajektorii a tím by mohl systém bezpečně splňovat funkci parkovacího asistenta pro osobní automobily.

## 10 REFERENCE

- [1] Parkovací asistent – jak to funguje? – AutoRevue.cz. AutoRevue.cz – Auta, testy, novinky, fotografie [online]. [cit. 16.8.2018]. Dostupné z: <https://www.autorevue.cz/parkovaci-asistent--jak-to-funguje>
- [2] [online]. Dostupné z: <https://www.carwow.co.uk/guides/glossary/parking-systems-explained> [cit. 18.8.2018].
- [3] Jak fungují parkovací senzory? A co všechno umí parkovací asistent?. Automix.cz [online]. Copyright © [cit. 16.08.2018]. Dostupné z: <https://automix.cars.cz/zivot-ridice/jak-funguji-parkovaci-senzory-a-co-parkovaci-asistent-20170607.html>
- [4] China Small Car Mobile DVR Camera System - China Mobile Dvr, Car Dvr. China Car Camera manufacturer, Car Security, Mobile Security supplier - Shenzhen Vcomsky Technology Co., Ltd. [online]. Copyright © 1998 [cit. 20.08.2018]. Dostupné z: <https://vcomsky.en.made-in-china.com/product/RSIEXFoBELWY/China-Small-Car-Mobile-DVR-Camera-System.html>
- [5] An Overview on Image Processing Techniques | Open Access Journals. Research and Reviews - International Journals [online]. Copyright © 2018 [cit. 20.08.2018]. Dostupné z: <http://www.rroij.com/open-access/an-overview-on-image-processing-techniques.php?aid=47175>
- [6] MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink. MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink [online]. Copyright © 1994 [cit. 16.08.2018]. Dostupné z: <https://www.mathworks.com/>
- [7] Howse J.: OpenCV Computer Vision with Python. Packt Publishing Ltd., 2013 ISBN 978-1-78216-392-3
- [8] Overview of image classification—ArcGIS Pro | ArcGIS Desktop. [online]. Copyright © 2018 Esri. [cit. 20.08.2018]. Dostupné z: <https://pro.arcgis.com/en/pro-app/help/analysis/image-analyst/overview-of-image-classification.htm>

[9] The University of Edinburgh, School of Physics Theory of Image Processing (P00809), Lecture 10 Revised: 10 November 2006

[10] Facultatea de Matematică și Informatică | Universitatea Babeș-Bolyai [online výukový materiál]. Copyright © [cit. 14.08.2018]. Dostupné

z: [http://www.cs.ubbcluj.ro/~csatol/mach\\_learn/bemutato/Matelstvan\\_StereoVision.pdf](http://www.cs.ubbcluj.ro/~csatol/mach_learn/bemutato/Matelstvan_StereoVision.pdf)

[11] Computer science | University of Toronto [online výukový materiál]. [cit. 11.8.2018]

Dostupné z: [http://www.cs.toronto.edu/~fidler/slides/2015/CSC420/lecture12\\_hres.pdf](http://www.cs.toronto.edu/~fidler/slides/2015/CSC420/lecture12_hres.pdf)

[12] Understanding Camera Calibration | PERPETUAL ENIGMA. PERPETUAL ENIGMA | PERENNIAL FASCINATION WITH ALL THINGS TECH [online]. Dostupné

z: <https://prateekvjoshi.com/2014/05/31/understanding-camera-calibration/>

[13] The Pinhole Camera - Anagram Engineering. Softwareentwicklung für Mobile-, Web- und integrierte Plattformen [online]. Dostupné z: <https://www.anagram.at/diplomarbeit/the-pinhole-camera/>

[14] Camera Calibration and 3D Reconstruction — OpenCV 2.4.13.7 documentation. OpenCV documentation index [online]. Copyright © Copyright 2011 [cit. 20.08.2018]. Dostupné

z: [https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html)

[15] [online] <http://openmvg.readthedocs.io/en/latest/openMVG/cameras/cameras/>

[16] Albert-Ludwigs-Universität Freiburg Arbeitsgruppe: Autonome Intelligente Systeme [online výukový materiál]. Dostupné z: [http://ais.informatik.uni-](http://ais.informatik.uni-freiburg.de/teaching/ws10/robotics2/pdfs/rob2-10-camera-calibration.pdf)

[freiburg.de/teaching/ws10/robotics2/pdfs/rob2-10-camera-calibration.pdf](http://ais.informatik.uni-freiburg.de/teaching/ws10/robotics2/pdfs/rob2-10-camera-calibration.pdf)

[17] Zhang's Camera Calibration Algorithm: In-Depth Tutorial and Implementation, Department of Digital Media, University of Applied Sciences Upper Austria, School of Informatics, Wilhelm Burger, Revised. 16.5.2016, [online]



[18] [online]. Dostupné z: <http://rpishop.cz/kamery/329-raspberry-pi-kamera-modul-v2.html>

[19] Camera Module - Raspberry Pi Documentation. Raspberry Pi — Teach, Learn, and Make with Raspberry Pi [online].

Dostupné z: <https://www.raspberrypi.org/documentation/hardware/camera/>

[20] Speeded up robust features - Wikipedia. [online]. Dostupné

z: [https://en.wikipedia.org/wiki/Speeded\\_up\\_robust\\_features](https://en.wikipedia.org/wiki/Speeded_up_robust_features)

[21] Shi-Tomasi Corner Detector & Good Features to Track — OpenCV 3.0.0-dev documentation. OpenCV documentation index [online]. Copyright © Copyright 2011 [cit.

20.08.2018]. Dostupné z: [https://docs.opencv.org/3.0-](https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_shi_tomasi/py_shi_tomasi.html)

[beta/doc/py\\_tutorials/py\\_feature2d/py\\_shi\\_tomasi/py\\_shi\\_tomasi.html](https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_shi_tomasi/py_shi_tomasi.html)

[22] Shortest Path with Obstacle Avoidance (ver 1.3), Michael Kleder, October 2005, Matlab script