

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

Bakalářská práce

Displej s pohyblivým pozorovatelem

Prohlášení

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

V Plzni dne 16.5.2008 Michal Mikšovic,

Display for moving observer

Main goal of this work is to design and describe a solution of finding observers head in space using two web cameras and implement an application that demonstrates the system. There are introduced some commercial hardware and software solutions for head tracking, described the functionality, advantages and disadvantages.

In the following chapters there is described the own solution. First there is a camera calibration model that is based on geometric deduction. Then there is a part that deals with lens distortion problem and the method for measurement of lens distortion is described. Method for removing lens distortion for better reconstruction precision is also demonstrated.

Next chapters deal with reconstruction and camera frames in time synchronization. The implementation is described in the end. There are two versions. Offline version that analyzes and makes reconstructions from the pre-recorded data and real time version where the results of reconstruction are shown in real time and user can see the dynamic behavior of the system.

Obsah

1. ÚVOD – TRACKINGOVÉ SYSTÉMY	- 7 -
1.1. CÍLE PRÁCE.....	- 7 -
1.2. TRACKINGOVÉ SYSTÉMY	- 7 -
1.3. HEAD TRACKERS.....	- 8 -
1.4. EXISTUJÍCÍ TRACKOVACÍ SYSTÉMY	- 10 -
1.4.1. <i>Optitrack</i>	- 10 -
1.4.2. <i>SmartNav</i>	- 11 -
1.4.3. <i>TrackIR</i>	- 12 -
1.4.4. <i>InterTrax2</i>	- 12 -
1.4.5. <i>ADL-1</i>	- 12 -
1.4.6. <i>Wrighttrack</i>	- 13 -
1.4.7. <i>Fastrak</i>	- 14 -
1.4.8. <i>Isotrak II</i>	- 14 -
1.4.9. <i>Johnny Chung Lee project</i>	- 15 -
2. NÁVRH HEADTRACKERU	- 16 -
2.1. KALIBRACE	- 16 -
2.1.1. <i>Direct linear transformation</i>	- 16 -
2.1.2. <i>Vlastní metoda kalibrace</i>	- 21 -
2.2. RADIÁLNÍ DISTRORZE ČOČEK.....	- 25 -
2.2.1. <i>Aproximace zkreslení polynomem</i>	- 27 -
2.2.2. <i>Metoda měření</i>	- 27 -
2.2.3. <i>Návrh řešení</i>	- 29 -
2.2.4. <i>Odstranění distorze</i>	- 30 -
2.3. REKONSTRUKCE	- 31 -
2.3.1. <i>Nalezení bodu ve snímku</i>	- 31 -
2.3.2. <i>Synchronizace snímků</i>	- 33 -
2.3.3. <i>Proces rekonstrukce</i>	- 35 -
2.4. VLASTNOSTI MODELU.....	- 37 -
2.4.1. <i>Přesnost kalibrace</i>	- 37 -
2.4.2. <i>Přesnost nalezení objektu</i>	- 40 -
2.4.3. <i>Viditelnost bodu z kamer</i>	- 41 -
3. NÁVRH IMPLEMENTACE.....	- 43 -
3.1. POUŽITÉ TECHNOLOGIE	- 43 -
3.2. OFFLINE VERZE.....	- 44 -
3.3. REALTIME VERZE.....	- 44 -
3.3.1. <i>Popis algoritmu</i>	- 44 -
4. TESTOVÁNÍ	- 46 -
4.1. UMĚLÁ DATA.....	- 46 -
4.1.1. <i>Kalibrace</i>	- 46 -
4.1.2. <i>Rekonstrukce</i>	- 47 -
4.2. REÁLNÁ DATA – OFFLINE VERZE APLIKACE	- 50 -
4.2.1. <i>Kalibrace</i>	- 51 -
4.2.2. <i>Test – odměření bodů</i>	- 53 -
4.3. REÁLNÁ DATA – REALTIME VERZE APLIKACE	- 54 -

4.1.	ODEZVA SYSTÉMU	- 57 -
4.2.	DISKUZE VÝSLEDKŮ	- 58 -
5.	ZÁVĚR.....	- 58 -
6.	PŘÍLOHY	- 59 -
A.	UŽIVATELSKÁ DOKUMENTACE.....	- 59 -
	OFFLINE VERZE	- 59 -
	REALTIME VERZE	- 62 -
B.	PŘEKLAD	- 64 -
C.	DALŠÍ MĚŘENÍ	- 65 -
	TEST – POMALÝ POHYB	- 65 -
	TEST – RYCHLÝ POHYB V OSE X.....	- 68 -
	<i>Reálná data – rychlý pohyb v ose z.....</i>	<i>- 71 -</i>
	<i>Umělá data – Blender motion blur</i>	<i>- 73 -</i>
D.	POPIS NAMESPACEŮ.....	- 75 -
E.	POPIS NEJDŮLEŽITĚJŠÍCH TŘÍD	- 77 -
F.	SEZNAM TRACKERŮ	- 80 -
	REFERENCE	- 82 -

1. Úvod – Trackingové systémy

1.1. Cíle práce

Cílem této práce je navrhnout a implementovat systém pro nalezení hlavy pozorovatele v prostoru. Systém bude používat dvě webové kamery. Kalibrace kamer by měla být co nejjednodušší, nemusí být však úplně automatická. K zvýraznění hlavy pozorovatele je možné použít aktivního prvku (diody).

1.2. Trackingové systémy

Tracking je používán ve virtuálních prostředích (systémech), kde je potřeba zjistit pozici reálného fyzického objektu v prostoru. Tyto systémy jsou pak používány v simulacích virtuální reality. V některých aplikacích virtuální reality je potřeba získat i orientaci objektu. Pak je objekt popsán polohou v kartézských souřadnicích (x,y,z) a třemi úhly určujícími otočení okolo jednotlivých os. Popisovaný objekt má pak 6 stupňů volnosti.

Trackovací systémy jsou používány k měření pohybu (polohy v čase) hlavy nebo rukou, někdy i očí. Tyto vstupy jsou pak použity k zpětné (nejčastěji vizuální) vazbě ve vztahu k pozici uživatele. Fungují na nejrůznější bázi. Například při použití magnetických senzorů je přijímač umístěn na hlavu uživatele, při změně polohy hlavy se změní i poloha přijímače. Přijímač zachycuje signály z vysílače, který generuje nízkofrekvenční magnetické pole. Poloha hlavy uživatele je vzorkována elektronickou jednotkou, která používá algoritmus pro zjištění polohy a orientace přijímače vůči vysílači. Kromě magnetických trackerů jsou tu pak mechanické, optické a akustické (ultrazvukové). Kromě hlavy je možné umístit všechny tyto typy trackerů na ruce nebo jiné části těla a měřit jejich polohu. Některé trackery jsou vybaveny dalšími funkcemi a je s nimi možné provádět operace jako s myší v prostoru. Trackery pro sledování pozice očí fungují odlišným způsobem. Nehledají pozici nebo orientaci hlavy, ale směr, kterým jsou nasměrovány oči vzhledem k hlavě. Tyto informace jsou pak použity k zjištění směru pohledu uživatele. Trackery pro sledování očí (Eye trackers) jsou elektromagnetické, optické, nebo elektrooptické.

Trackery jsou také používány v aplikacích AR (augmented reality). V těchto systémech uživatel vidí reálný svět okolo sebe v kombinaci s virtuálními objekty počítačové grafiky umístěnými do reálného světa. Jedním z největších problémů AR je správně umístit reálné virtuální objekty do obrazu reálného světa. Časový interval mezi měřením pozice hlavy a zkomponováním reálného a virtuálního světa není zanedbatelný (trackovací systém pracuje s určitou latencí). Při pohybu pak virtuální objekty nesedí přesně mezi reálné. Lidské oko je velmi citlivé na jakékoliv chyby, které jsou tolerovatelné ve virtuální realitě, kde se jedná pouze o zpoždění, nikoliv však v AR, kde virtuální objekty „plavou“ mezi reálnými. V dalším textu se zaměříme pouze na hledání pozice hlavy – head trackery.

1.3. *Head Trackers*

Jak už bylo zmíněno v kapitole 1.1, tato práce se zaměřuje na tracking hlavy v prostoru. Toto odvětví trackovacích systémů používá nejčastěji magnetickou, akustickou a optickou technologii. V tabulce 1.1 je přehled dostupných typů systémů, jejich výhody a nevýhody.

Kvalitu těchto řešení lze popsat několika charakteristikami, které lze změřit a porovnat.

- Rozlišení. Měří přesnost, s kterou je systém schopen najít pozici. Je měřena v centimetrech kde vysílač a přijímač naleznou odlišnou polohu. Orientace je měřena ve stupních.
- Přesnost. Je rozmezí, v kterém existuje správná nalezená pozice. Je to chybová funkce způsobená měřeními a je statisticky vyjádřena ve stupních jako RMS (root mean square) – pro orientaci a v cm pro pozici.
- Schopnost systému reagovat. Obsahuje:
 - o Frekvence vzorkování. Frekvence, s kterou senzory poskytují data.
 - o Frekvence dat. Počet spočítaných pozicí objektu za vteřinu.
 - o Frekvence aktualizace. Frekvence, s kterou systém vrací souřadnice nových bodů k připojenému počítači.
 - o Latence. Zpoždění (v ms) mezi pohybem objektu a výstupem nových souřadnic ze systému.

Další charakteristikou systému je opakovatelnost. Přesnost měření pozice a orientace mezi více objekty je výsledkem přesnosti měření jednotlivých objektů (odchyly) a trvalé odchylky systému. Proto opakovatelnost je méně rozptýlená několika měřeními objektu ve stejné stacionární pozici. Poskytuje představu o přesnosti měření a je vyjádřena v centimetrech a úhlových stupních. Dalším měřítkem specifickým pro inerciální systémy je postupné zanášení chyby. Tyto systémy provádí měření akcelerometry a gyroskopy. Výsledek je vždy relativní k startovací pozici a postupně do systému zanáší chyby – po čase je potřeba systému dodat absolutní pozici pro recalibraci.

technologie	Popis	Výhody	Nevýhody
Mechanické	Měří změnu pozice fyzickým propojením reálného objektu a kloubového mechanického ramene	Přesnost Malá latence Nejsou problémy s přímou viditelností a interferencí magnetických signálů Dobry pro tracking v malém prostoru	Rušivé pro uživatele (mechanická část obleku)
Magnetické	Používá několik cívek ve vysílači, které generují magnetické pole. Magnetické senzory v přijímači měří sílu a úhly magnetického pole.	Levné Přesné Není problém s přímou viditelností Nízký šum Umožňuje snímat pohyby celého těla Velké rozpětí ve velikosti prostředí (velké i malé prostory)	Feromagnetické a/nebo vodivé povrchy zapříčiňují zkreslení elektromagnetického pole Elektromagnetická interference Ztrácí přesnost s rostoucí vzdáleností Vysoká latence kvůli filtrování
Neinerciální (bez zdroje)	Používá pasivní magnetické senzory, které se vztahují k magnetickému poli země. Je měřeno otočení okolo os x,y,z a derivací úhlové zrychlení a rychlost.	Levné Není potřeba vysílač Přenosné	Jenom 3 stupně volnosti Obtížné zaznamenat pohyb mezi magnetickými hemisférami
Optické	Používá nejrůznější detektory, nejčastěji kamery, k detekování světla, které vyzařuje snímaný objekt. Často je použito infračervené světlo, aby nekolidovalo se světlem, které vyzařují ostatní objekty.	Dobrá dostupnost Lze použít ve velkých prostorech Rychlý Nejsou problémy s magnetickou interferencí Vysoká přesnost	Nutná přímá viditelnost Limitováno koherencí a intenzitou světelných zdrojů Vyšší váha Vysoká cena
Akustické (ultrazvuk)	Používá tři mikrofóny a tři zvukové generátory k počítání vzdálenosti mezi zdrojem a přijímačem triangulací. Frekvence je nad 20 kHz, takže lidským uchem není slyšet.	Nízká cena Není problém s magnetickou interferencí Nízká váha	Interference ultrazvuku Nízká přesnost (zvuk se šíří různě podle podmínek prostředí) Ozvěny vytváří šum Nutná přímá viditelnost
Inerciální	Používají akcelerometry a gyroskopy. Orientace objektu je počítána integrací výstupů gyroskopů proporcionálně ke každé z os. Změna pozice lze počítat dvojitou integrací výstupu akcelerometrů.	Neomezený prostor Rychlý Není potřeba přímá viditelnost Nejsou problémy s magnetickou interferencí Přímo měří orientaci Malá velikost Nízká cena	Pouze 3 stupně volnosti Postupně zanáší chybu Nepřesné pro pomalé změny pozice

tab. 1.1

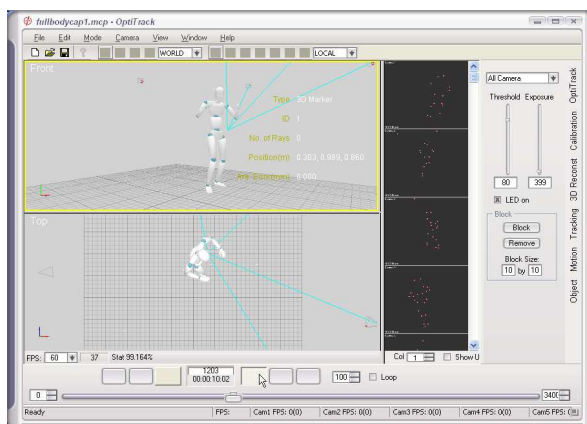
1.4. Existující trackovací systémy

V této části budou zmíněny a popsány některé existující systémy pro trackování bodu (bodů) v prostoru i v rovině. Všechny tyto systémy používají speciální hardwarové prostředky. Nezisková společnost Institute for Defence Analyses provedla podrobnou studii všech dostupných rozhraní pro virtuální prostředí. Výsledkem je mimo jiné podrobný popis všech dostupných head trackerů, jejich parametrů a cen (viz [1]). Tato studie je bohužel neaktuální (1996), proto budou uvedeny pouze ty systémy, které jsou dostupné i dnes. K systémům, které nejsou zahrnuty v této studii (vznikly později) budou uvedeny pouze informace dostupné na internetu na stránkách prodejce. Protože těchto systémů existuje několik desítek a podrobný průzkum trhu není cílem této práce, budou uvedeny pouze některé pro představu. V příloze je uvedena tabulka s cenami a některými parametry.

1.4.1. Optitrack

Společnost NaturalPoint nabízí kompletní řešení trackování bodu (bodů) v prostoru – Optitrack (Optical motion capture and tracking). Zabývá se jak odvětvím softwaru, tak hardwaru. Optitrack používá tzv. reflective marker technologii. Kamery mají vestavěný zdroj infračerveného světla, které se odráží od reflexivních značek (reflective marker) rozmístěných na hledaných pozicích. Ve snímaných obrazech jsou pak tyto značky výrazně odlišeny od pozadí svojí intenzitou. Pro vzdálenější objekty jsou použity aktivní značky, které samy vyzářují infračervené světlo. K snímání scény jsou použity obvykle tři až čtyři kamery. Kamery obsahují hardware pro nalezení těchto bodů ve snímaných obrazech a posílají informace o souřadnicích po USB rozhraní do počítače. Kamery pořizují obrázky s frekvencí 120 snímků za vteřinu. Software obsahuje informace o kalibraci kamer a provádí rekonstrukci bodu(ů) v prostoru. Nabízená řešení jsou

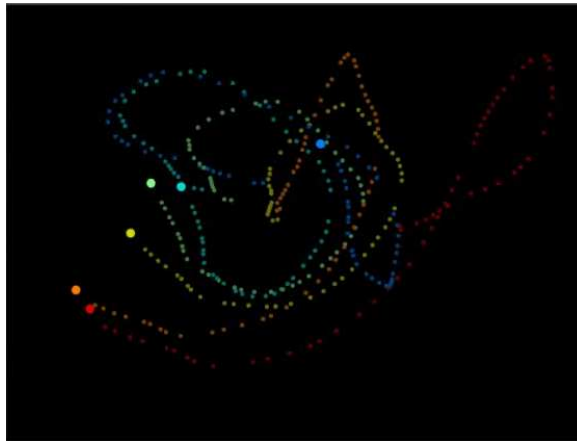
- Full body motion capture – pro zaznamenání pohybu všech částí těla (viz obr. 1.1). Systém obsahuje čtyři kamery.



obr. 1.1

- Rigid body capture – pro zaznamenání polohy hlavy a rukou. Systém se třemi kamerami.
- Point cloud tracking – pro zaznamenání libovolného počtu bodů. Systém obsahuje libovolný počet kamer.

Kalibrace systému je pro uživatele velmi jednoduchá. Stačí zaznamenat sekvenci, kde libovolně pohybujeme jedním bodem (značkou) všemi směry (viz obr. 1.2). Bod musí být viditelný ze všech kamer, které chceme kalibrovat. Z těchto sekvencí se pak spočítá jejich poloha a orientace.



obr. 1.2

1.4.2. SmartNav

Tento systém umožňuje ovládat počítač pomocí hlavy, nebo jiné libovolné části těla. Samolepící reflexivní papírový bod je nalepen nejčastěji na čelo. Speciální kamera vysílá infračervené světlo, které je tímto bodem odraženo a snímáno. Tato kamera obsahuje filtr, který propouští pouze infračervené světlo. Jde o podobný princip jako u produktu Optitrack (viz obr. 1.3) Frekvence je 120 snímků za vteřinu. Jedná se pouze o rekonstrukci bodu v rovině – informace o vzdálenosti pozorovatele není počítána.



obr. 1.3

1.4.3. TrackIR

Tento systém sleduje pohyb hlavy i její orientaci ve všech osách (6 stupňů volnosti). Je nejčastěji používán v herním průmyslu, zejména v simulátorech letadel. Uživatel si připevní trojici reflexivních plošek na hlavu (viz obr. 1.4). Tyto plochy odrážejí infračervené světlo a jsou trackovány v prostoru obdobně jako u výše zmíněných produktů.



obr. 1.4

1.4.4. InterTrax2

Systém od společnosti Intersense. Má tři stupně volnosti (rotace v 3 osách). Tento systém je uváděn jako fyzicky nejmenším dosud dostupným řešením. Cena tohoto systému je \$995 (2008).

Počet stupňů volnosti	3
Frekvence aktualizace	256 Hz
Latence	4 msec
Úhlové rozlišení	0.02°



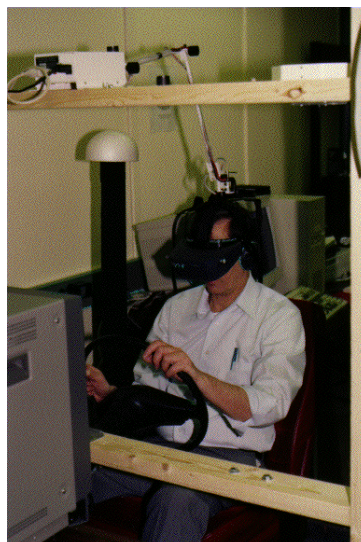
obr. 1.5

1.4.5. ADL-1

ADL-1 od společnosti Shooting Star Technology je mechanický systém s 6 stupni volnosti. Uživatel má na hlavě čelenku, která je připevněna k mechanické ruce. Může sedět před

monitorem, nebo použít virtuální helmu. Senzory připevněné k mechanické ruce měří úhly mezi klouby ruky. Procesor z úhlů vypočte pozici a orientaci hlavy a odesílá data do počítače. Cena systému byla \$1,299 (1996 - dnes není uvedena).

Frekvence aktualizace	240 Hz
Frekvence vzorkování	240 Hz
Latence	0.35-1.8 msec
Přesnost	0.2 in
Lineární rozlišení	~0.025 in
Úhlové rozlišení	0.15 -0.3°
Pracovní objem	Poloviční válec, ~36 in v průměru, 18 in výška



obr. 1.6

1.4.6. Wriptrack

Wriptrack od společnosti Vidtronics, Inc. je mechanický systém s 6 stupni volnosti. Je navržen pro použití s PC a desktopovými VR systémy. Je řízen mikroprocesorem a určuje úhly pomocí potenciometrů. Je navržen pro systém MS-DOS. Cena systému je \$795.

Frekvence aktualizace	300 Hz
Úhlové rozlišení	0.1°
Pracovní objem	¼ koule 40 in průměr
Opakovatelnost	0.1 in

1.4.7. Fastrak

Polhemus Fastrak byl vyvinut ze systému Isotrak, který patří mezi první systémy. Používá analogovou technologii a má velkou latenci, vysoký šum a malou citlivost senzorů. Tyto problémy byly vyřešeny použitím digitální technologie zpracování signálu. Systém může obsahovat až 4 přijímače a až 8 systémů může být zapojeno dohromady. Cena systému byla \$6,050 (1996).

Frekvence aktualizace	120 Hz / počtem přijímačů
Latence	4 msec
Lineární Přesnost	0.2 in
Úhlová přesnost	0.15°
Lineární Rozlišení	0.0002 in/in
Úhlové rozlišení	0.025°
Pracovní vzdálenost	Až 10 ft



3D-BIRD mounted on Sony LDI-100 HMD. Audi TT Coupe virtual interior and exterior environment courtesy Eon Reality, Sweden.

obr. 1.7

1.4.8. Isotrak II

Isotrak II od firmy Polhemus je levnější verzí Fastracku s některými horšími vlastnostmi. Skládá se z elektronické jednotky, jednoho vysílače a 1 – 2 přijímačů. Jedním z přijímačů může být 3Ball, 3D polohovací vstupní zařízení, které spojuje pohyby ruky a těla. Toto přídatné zařízení stojí \$800, základní cena je \$2,875.

Frekvence aktualizace	60 Hz / počtem přijímačů
Latence	20 msec
Lineární Přesnost	0.1 in
Úhlová přesnost	0.75°
Lineární Rozlišení	0.0015 in/in
Úhlové rozlišení	0.1°
Pracovní vzdálenost	Až 15 ft

1.4.9. Johnny Chung Lee project

Johnny Chung Lee navrhl zajímavé řešení headtrackingu použitím ovladače k herní konzoli Nintendo. Systém používá kameru, která snímá infračervené světlo (Nintendo Wii) a dvě diody umístěné na hlavě. Pro podrobnosti o projektu viz [1].



obr. 1.8

2. Návrh Headtrackeru

Při návrhu optického headtrackeru s dvěma kamerami je potřeba vyřešit celou řadu aspektů, které budou probrány v následujících podkapitolách. Základními aspekty je však kalibrace (kap. 2.1) a rekonstrukce (kap. 2.3). Při kalibraci je kromě orientací a pozic kamer nutné zjistit radiální distorzi čoček (kap. 2.2), při rekonstrukci je to pak nalezení světlého bodu v obrázku (kap. 2.3.1) a synchronizace snímků z kamer (kap. 2.3.2).

2.1. Kalibrace

Kalibrace je proces, při kterém je nalezena poloha kamer, jejich orientace, popřípadě jsou zjištěny některé vnitřní parametry kamery. Metod pro kalibraci je celá řada. Od nejjednodušších, kde jsou všechny parametry zadávány ručně, až po složité matematické modely hledání korespondencí obrazu mezi kamerami a automatickým určením jejich poloh. Parametry kalibrované kamery se často dělí na vnější a vnitřní. Vnější parametry popisují polohu a orientaci kamery nezávisle na optické soustavě a čipu. Naopak vnitřní parametry popisují vlastnosti kamery nezávisle na okolním světě. Pro kompletní kalibraci kamery je potřeba získat tyto parametry.

Vnější parametry

- Poloha ohniska – nejčastěji vektor v kartézských souřadnicích
- Orientace kamery – vektor v kartézských, popřípadě ve sférických souřadnicích
- Rotace okolo optické osy – úhel

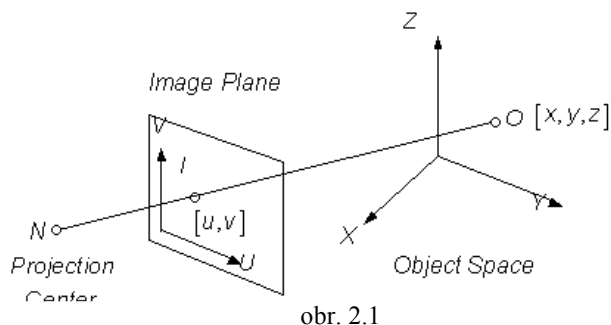
Vnitřní parametry

- Ohnisková vzdálenost
- Velikost čipu – je potřeba znát šířku i výšku
- Rozlišení čipu
- Funkce popisující radiální distorzi čočky – nejčastěji polynom

2.1.1. Direct linear transformation

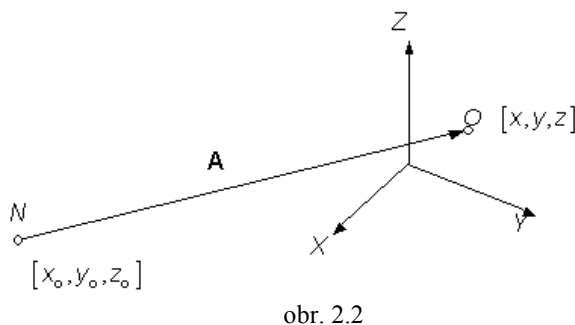
Direct linear transformation je metoda, která dává do vztahu objekty zobrazené před kamerou s orientací kamery. Je zde uvedena pro představu a návrh vlastní metody v kapitole 2.1.2 vychází pouze z geometrické inspirace touto metodou.

Snímání obrazu použitím kamery je zjednodušeně promítání paprsků světla odražených od objektů reálného světa přes optickou soustavu na čip. Pokud zjednodušíme snímanou scénu na pouhý bod v prostoru a zanedbáme optickou soustavu kamery, pak promítáme bod z prostoru objektu do roviny čipu.

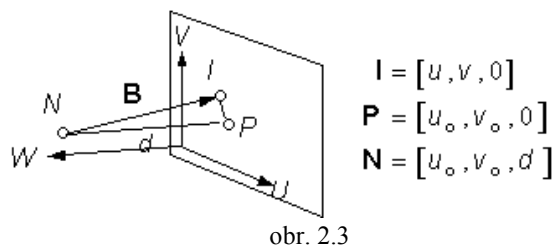


Na obrázku 2.1 jsou zobrazeny dva souřadné systémy. Systém XYZ prostoru objektu a systém UV roviny čipu. Bod O z prostoru objektu je promítán do bodu I v rovině čipu. $[x, y, z]$ jsou souřadnice bodu O v prostoru, $[u, v]$ jsou souřadnicemi bodu I v rovině čipu. Body O, I a N jsou kolineární¹. Podmínka kolinearit je základem DLT metody.

Předpokládejme, že pozice středu projekce (bod N) je v bodě $[x_0, y_0, z_0]$. Vektor \mathbf{A} mířící od bodu N do bodu O je pak $[x - x_0, y - y_0, z - z_0]$.



Přidáním další osy W do souřadného systému čipu se stane systém trojrozměrným. Všechny body, které leží v rovině čipu, mají tuto souřadnici nulovou.



¹ Množina bodů je kolineární, pokud všechny body z množiny leží na přímce.

Na obrázku 2.3 je dále zobrazen nový bod P (principal point). Úsečka, ze středu projekce (N) do roviny čipu (P), rovnoběžná s osou W a kolmá na rovinu čipu, se nazývá *principal axis*. Bod P je průsečíkem roviny čipu a *principal axis*. Vzdálenost d je vzdáleností mezi body N a P . Souřadnice bodu P jsou $[u_0, v_0, 0]$. Pozice bodu N v souřadném systému čipu je $[u_0, v_0, d]$. Vektor B mířící z bodu N do bodu I je pak $[u - u_0, v - v_0, -d]$.

Zatímco body O, I a N jsou kolineární, vektory A a B jsou na stejné přímce. Podmínka kolinearit se dá pak zapsat

$$B = c \cdot A \quad (2.1)$$

Protože vektory A a B byly původně popsány v souřadném systému čipu a souřadném systému objektu, abychom mohli s nimi relativně pracovat, musí být převedeny do společného souřadného systému. Jednou z možností je transformovat vektor A do souřadného systému čipu.

$$T_{I/O} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$A^{(I)} = T_{I/O} \cdot A^{(O)} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot A^{(O)} \quad (2.2)$$

$A^{(I)}$ je vektor A vyjádřený v souřadném systému čipu. $A^{(O)}$ je vektor A vyjádřený v souřadném systému objektu a $T_{I/O}$ je transformační matice mezi těmito souřadnými systémy. Dosazením rovnice (2.1) do (2.2) dostaneme

$$\begin{bmatrix} u - u_0 \\ v - v_0 \\ -d \end{bmatrix} = c \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix}$$

$$u - u_0 = c[r_{11}(x - x_0) + r_{12}(y - y_0) + r_{13}(z - z_0)]$$

$$v - v_0 = c[r_{21}(x - x_0) + r_{22}(y - y_0) + r_{23}(z - z_0)] \quad (2.3)$$

$$-d = c[r_{31}(x - x_0) + r_{32}(y - y_0) + r_{33}(z - z_0)]$$

Z rovnice (2.3) dostaneme

$$c = \frac{-d}{r_{31}(x - x_0) + r_{32}(y - y_0) + r_{33}(z - z_0)} \quad (2.4)$$

substitucí (2.4) za (2.3) dostaneme

$$u - u_0 = -d \frac{r_{11}(x - x_0) + r_{12}(y - y_0) + r_{13}(z - z_0)}{r_{31}(x - x_0) + r_{32}(y - y_0) + r_{33}(z - z_0)}$$

$$v - v_0 = -d \frac{r_{21}(x - x_0) + r_{22}(y - y_0) + r_{23}(z - z_0)}{r_{31}(x - x_0) + r_{32}(y - y_0) + r_{33}(z - z_0)}$$

Vektory u , v , u_0 , v_0 jsou ze souřadného systému čipu, který je v jednotkách reálného světa (jako jsou např. centimetry). Reálně čip kamery používá jiný souřadný systém, nejčastěji pixely. Proto přepíšeme

$$u - u_0 \Rightarrow \lambda_u (u - u_0)$$

$$v - v_0 \Rightarrow \lambda_v (v - v_0)$$

$$u - u_0 = \frac{-d}{\lambda_u} \frac{r_{11}(x - x_0) + r_{12}(y - y_0) + r_{13}(z - z_0)}{r_{31}(x - x_0) + r_{32}(y - y_0) + r_{33}(z - z_0)}$$

$$v - v_0 = \frac{-d}{\lambda_v} \frac{r_{21}(x - x_0) + r_{22}(y - y_0) + r_{23}(z - z_0)}{r_{31}(x - x_0) + r_{32}(y - y_0) + r_{33}(z - z_0)}$$

Kde $[\lambda_u, \lambda_v]$ jsou koeficienty pro převod jednotek os U a V. Původní jednotky os U a V mohou být libovolné, koeficienty $[\lambda_u, \lambda_v]$ mohou být různé pro každou s os¹. Pokud přepíšeme výraz pro osy x, y a z, dostáváme:

$$u = \frac{L_1 x + L_2 y + L_3 z + L_4}{L_9 x + L_{10} y + L_{11} z + 1}$$

$$v = \frac{L_5 x + L_6 y + L_7 z + L_8}{L_9 x + L_{10} y + L_{11} z + 1}, \text{ kde}$$

$$[d_u, d_v] = \left[\frac{d}{\lambda_u}, \frac{d}{\lambda_v} \right]$$

$$D = -(x_0 r_{31} + y_0 r_{32} + z_0 r_{33})$$

$$L_1 = \frac{u_0 r_{31} - d_u r_{11}}{D}$$

$$L_2 = \frac{u_0 r_{32} - d_u r_{12}}{D}$$

$$L_3 = \frac{u_0 r_{33} - d_u r_{13}}{D}$$

¹ Například pokud jednotlivé pixely na čipu nebudou čtvercové

$$L_4 = \frac{(d_u r_{11} - u_0 r_{31})x_0 + (d_u r_{12} - u_0 r_{32})y_0 + (d_u r_{13} - u_0 r_{33})z_0}{D}$$

$$L_5 = \frac{v_0 r_{31} - d_v r_{21}}{D}$$

$$L_6 = \frac{v_0 r_{32} - d_v r_{22}}{D}$$

$$L_7 = \frac{v_0 r_{33} - d_v r_{23}}{D}$$

$$L_8 = \frac{(d_v r_{21} - v_0 r_{31})x_0 + (d_v r_{22} - v_0 r_{32})y_0 + (d_v r_{23} - v_0 r_{33})z_0}{D}$$

$$L_9 = \frac{r_{31}}{D}$$

$$L_{10} = \frac{r_{32}}{D}$$

$$L_{11} = \frac{r_{33}}{D}$$

Koeficienty L_1 až L_{11} jsou DLT parametry, které vyjadřují vztah mezi souřadným systémem objektu a souřadným systémem čipu.

Po rozšíření této metody pro n kontrolních bodů (n bodů O) lze kalibrovat kamery a získat koeficienty popisující tento vztah. Minimální počet kontrolních bodů pro získání 11 parametrů je 6. Existují i další modifikace této metody (podrobně viz [3]), které zahrnují i optickou soustavu kamery.

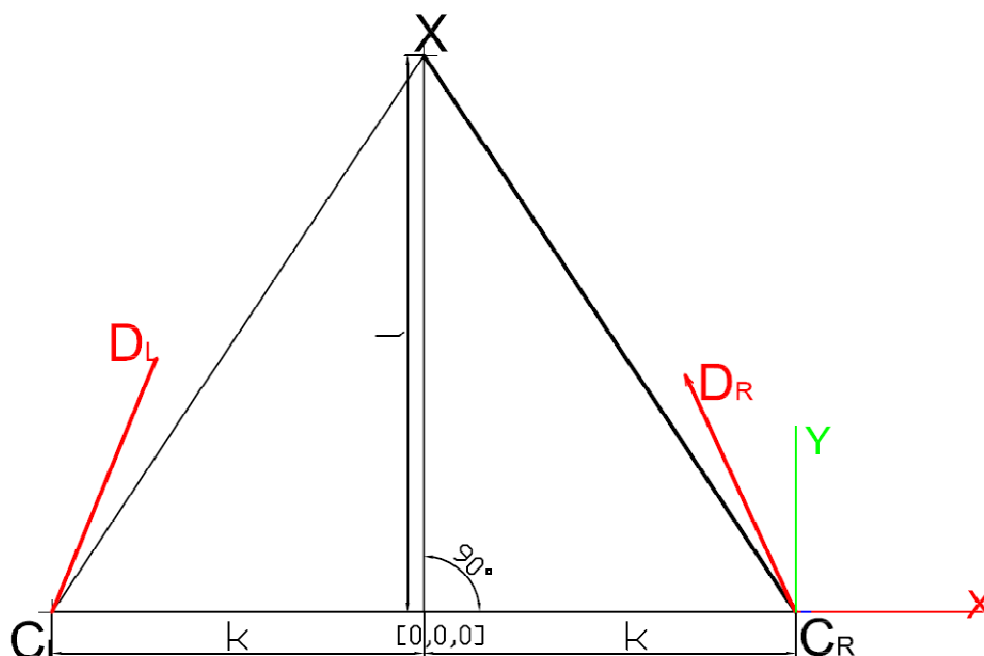
2.1.2. Vlastní metoda kalibrace

Pro odvození matematického modelu kalibrace je nejprve nutné stanovit omezující podmínky poloh a orientací kamer. Pro vytvoření co nejjednoduššího modelu byly stanoveny tyto podmínky:

- Kamery musí ležet ve stejné vertikální rovině
- Rotace kamer okolo optické osy není umožněna
- Kalibrační bod musí ležet ve vertikální rovině kamer a jeho souřadnice jsou přesně stanoveny (viz kap. 2.1.2)

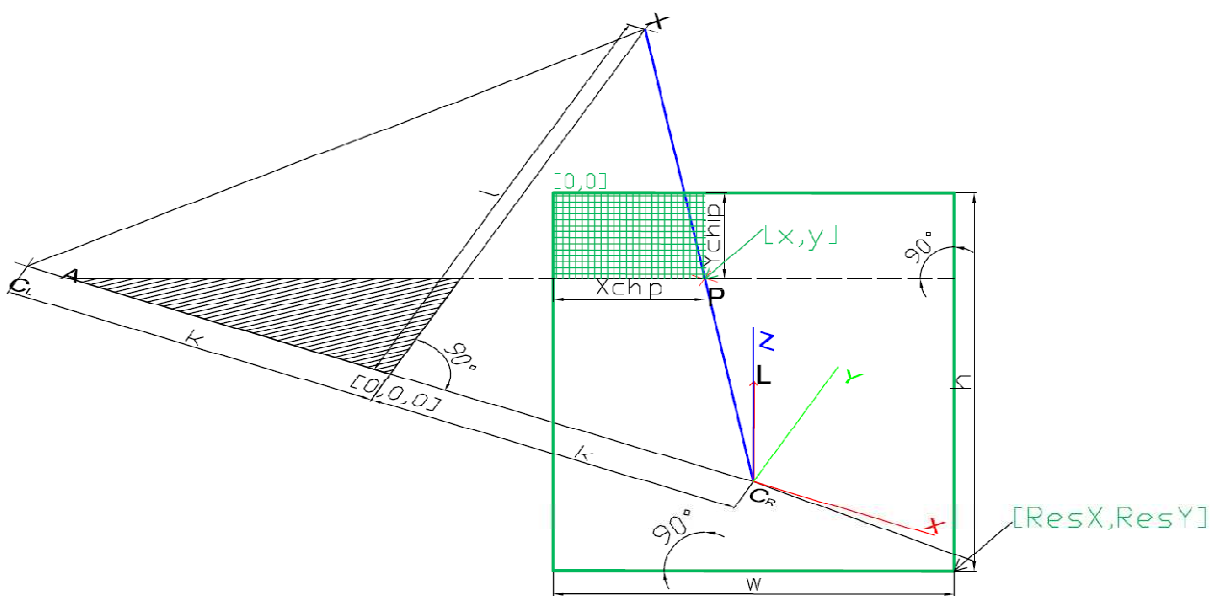
V této části nebudeme uvažovat optickou soustavu kamery, ale pouze její zjednodušený model (tzv. pinhole camera).

Na obrázku 2.4 je zjednodušené zobrazení modelu, z kterého budeme dále vycházet. Popišme si nyní některé symboly, které budou použity v matematickém odvození. Ohniska kamer jsou na souřadnicích C_L a C_R (dále pozice kamer). Počátek souřadného systému světa byl zvolen v polovině vzdálenosti mezi kamerami. Kamery jsou ve stejné rovině a obě jsou vzdáleny o vzdálenost k od počátku souřadnic. Souřadnice kamer jsou tedy $C_L = (-k, 0, 0)$ a $C_R = (k, 0, 0)$.



obr. 2.4

Bod X symbolizuje bod pro kalibraci a jeho vzdálenost od počátku souřadnice je l . Jeho souřadnice jsou tedy vždy $X = (0, l, 0)$. Vektory D_R a D_L symbolizují orientaci kamer. Tyto vektory jsou neznámé a měli by být výsledkem kalibrace. Jejich souřadnice jsou značeny $D_R = (D_{R,x}, D_{R,y}, D_{R,z})$ a $D_L = (D_{L,x}, D_{L,y}, D_{L,z})$. Cílem následujících odvození je tyto souřadnice explicitně vyjádřit. Ohniskové vzdálenosti kamer jsou $\|D_R\|$ a $\|D_L\|$.



obr. 2.5

Na obrázku 2.5 je zobrazena pravá kamera s čipem (zelenou barvou). Čip je ve skutečnosti až za ohniskem kamery a obraz je na něm horizontálně a vertikálně otočen, pro účely tohoto modelu to ale není relevantní. Pro zjednodušení je umístěn před ohnisko a vzdálen od polohy pravé kamery o ohniskovou vzdálenost $\|D_R\|$. Bod L je ve středu čipu a vektor D_R je kolmý na rovinu čipu.

Paprsek (modrou barvou) vycházející z bodu X , který míří do ohniska kamery protíná rovinu čipu v bodě P . Výsledkem jsou diskrétní souřadnice (x,y) získané z x -tého, y -tého pixelu na čipu¹. Vzdálenosti X_{chip} a Y_{chip} jsou v souřadnicích světa. Protože jednou z omezujících podmínek kalibrace kamer je znemožnění rotace okolo optické osy kamery, je polopřímka \overline{AP} v rovině čipu. w (h) je šířka (výška) čipu.

Na obrázku 2.6 je zevrubně rozkreslená pravá kamera. Pohled je trochu z jiného úhlu než na obr. 2.5, aby byly vidět podstatné úhly.

¹ Souřadnice zobrazené zelenou barvou nejsou v souřadném systému světa, ale čipu.

$$\text{vzdálenost } |P_{L_x} L| = \frac{h}{2} - Y_{Chip},$$

$$\text{úhel } \alpha = \arctan\left(\frac{|P_{L_y} L|}{\|D\|}\right),$$

$$\text{úhel } \beta = \arctan\left(\frac{|P_{L_x} L|}{\|D\|}\right),$$

$$\text{úhel } \gamma = \arctan\left(\frac{l}{k}\right).$$

K popisu vektoru D jsou použity sférické souřadnice. Pro přechod do sférických souřadnic platí následující vztahy.

$$\theta = \pi - (\gamma + \alpha), \quad (2.1)$$

$$\phi = \beta + \frac{\pi}{2}, \quad (2.2)$$

$$\rho = \|D\|.$$

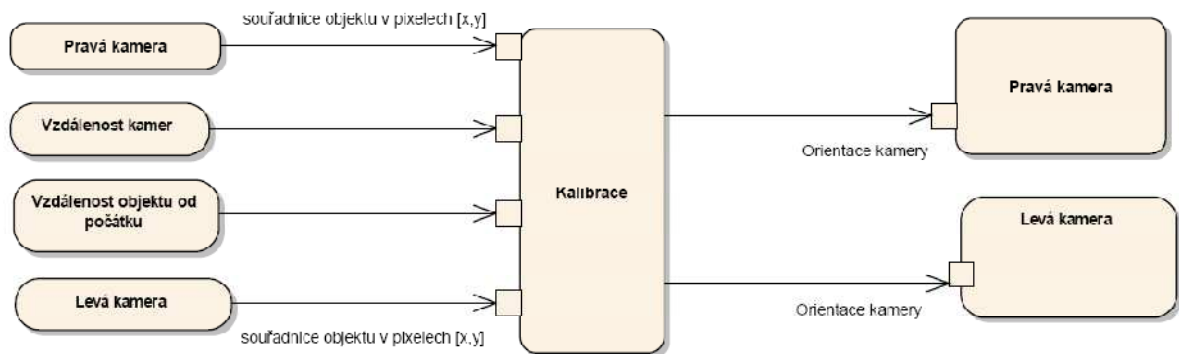
Přechodem do kartézských souřadnic získáme orientaci kamery (vektor D):

$$D_x = \|D\| \cdot \sin(\phi) \cos(\theta),$$

$$D_y = \|D\| \cdot \sin(\phi) \sin(\theta),$$

$$D_z = \|D\| \cdot \cos(\phi).$$

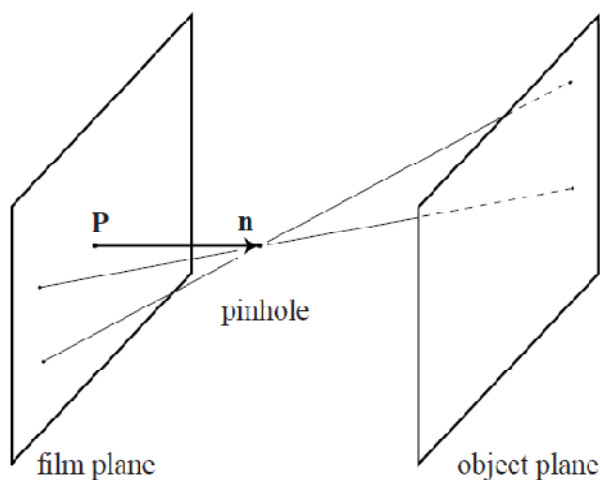
Celý proces kalibrace je schematicky popsán na obrázku 2.7. Vstupem do kalibrace jsou nalezené souřadnice objektu v obrazech levé a pravé kamery. Vzdálenost kamer a vzdálenost objektu od počátku. Výstupem z kalibrace jsou pak vypočtené orientace kamer.



obr. 2.7

2.2. Radiální distorze čoček

Před zaznamenáním obrazu na fotocitlivý čip prochází paprsky skrz optickou soustavu kamery. Tato soustava – nejčastěji čočka – trpí často nějakou distorzí. Zde přestává fungovat zjednodušený model kamery (pinhole camera) a je potřeba tuto distorzi odstranit.



obr. 2.8

Distorze čočky je radiální, tzn. stejná pro určité vzdálenosti od středu obrazu. V konstantní vzdálenosti před kamerou definujeme rovinu (viz obr. 2.8), která je rovnoběžná s rovinou čipu.

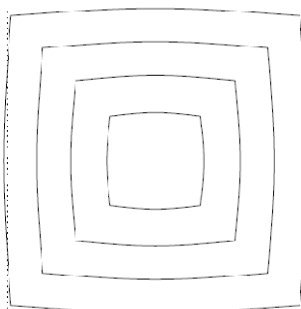
Když pomineme vliv optické soustavy, potom vztah vzdálenosti od optické osy v rovině objektu a vzdálenosti v rovině čipu lze popsat lineární funkcí.

$$L(r) = k \cdot r, \text{ kde}$$

$L(r)$ je vzdálenost v rovině čipu, $k \in \mathfrak{R}$ je koeficient a r je vzdálenost od optické osy v rovině objektu. Pokud přidáme optickou soustavu kamery, funkce $L(r)$ je lineární pouze pro kamery s kvalitní optikou. Definujme zvětšení (magnification) vztahem

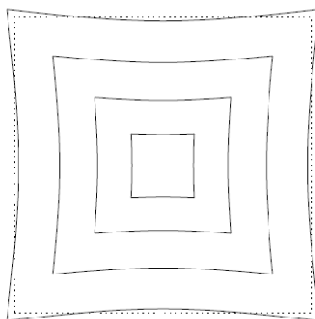
$$M = \frac{dL}{dr}, \text{ pak pokud}$$

$\frac{dM}{dr} < 0$, zkreslení je soudkovité (z anglického barrel distortion) viz obr. 2.9.



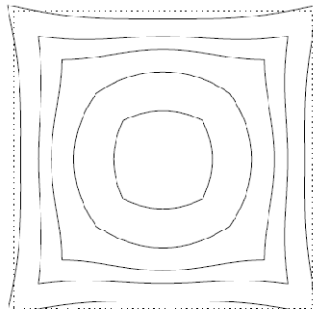
obr. 2.9

Pokud $\frac{dM}{dr} > 0$, zkreslení obrazu je poduškovité (anglicky pincussion distortion) viz obr. 2.10.



obr. 2.10

Obecně může derivace $\frac{dM}{dr}$ nabývat kladných i záporných hodnot a být kombinací obou zkreslení – viz obr. 2.11.



obr. 2.11

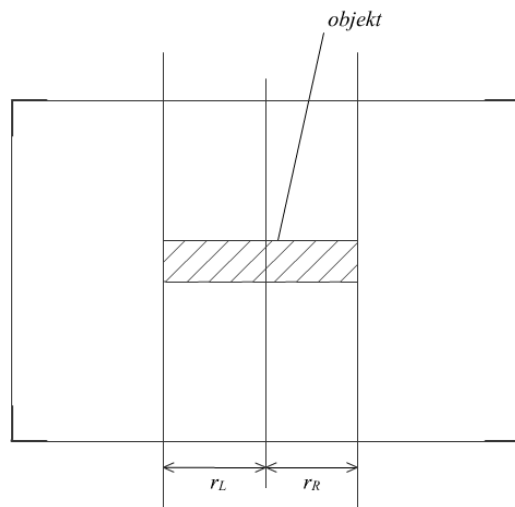
2.2.1. Aproximace zkreslení polynomem

Jak už bylo řečeno v kapitole 2.2.1 **Error! Reference source not found.**, nemůžeme očekávat, že funkce $L(r)$ bude lineární. Proto je potřeba nejprve provést určitá měření, potom nalézt funkci popisující chybu zkreslením čočky. Po vyhledání bodu v nasnímaném obraze (podrobně viz kap. 2.3.1 **Error! Reference source not found.**) budou tyto souřadnice upraveny odečtením této chyby.

2.2.2. Metoda měření

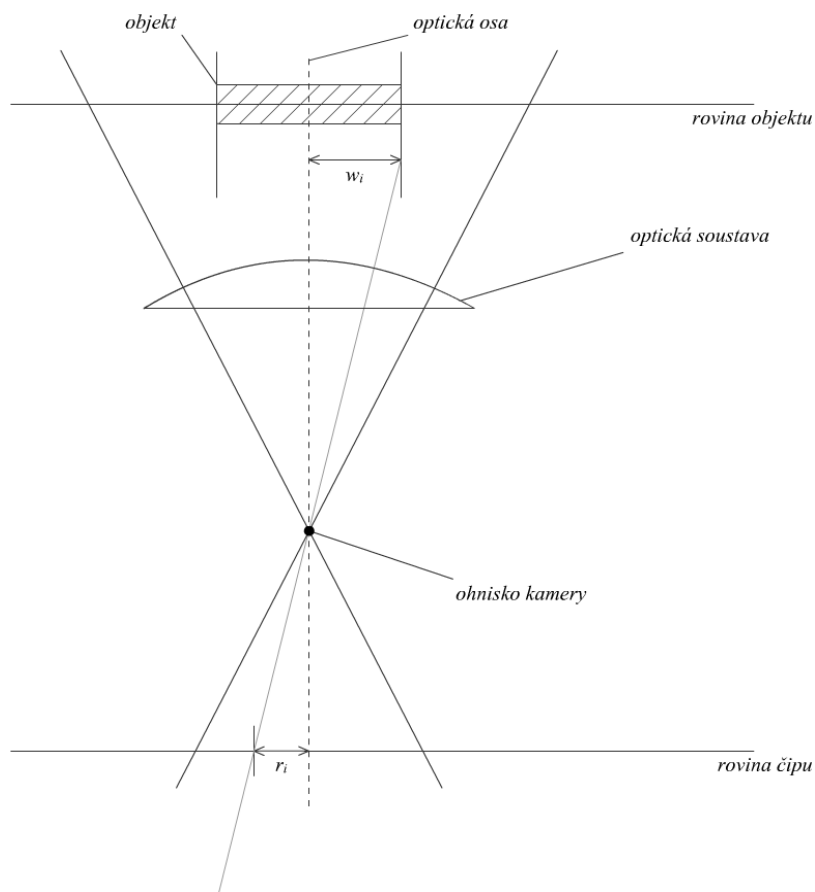
K pochopení následujících úvah je potřeba se zmínit o způsobu měření dat, s kterými bude dále proveden výpočet. Do roviny rovnoběžné s rovinou čipu kamery jsou postupně umísťovány objekty se známou velikostí v souřadnicích světa. Tato rovina má konstantní vzdálenost od kamery pro všechny objekty. Tyto objekty jsou snímány a je měřena velikost těchto objektů na čipu (viz obr. 2.12). Objekty musí být co nejbližší středu optické osy. Na snímaných obrázcích by střed objektu měl být na souřadnicích $\left[\frac{w}{2}, \frac{h}{2}\right]$, kde w (h) je horizontální (vertikální) rozlišení obrázku v pixelech. Pro zjednodušení předpokládejme, že objekty jsou podélného tvaru a před kameru jsou umístěny na šířku.

Označme w_i polovinu šířky objektu při i -tém měření a r_i horizontální vzdálenost od středu čipu (viz obr. 2.13). Měření jsou prováděna s postupně se zvětšujícími objekty ($w_i < w_{i+1}$).



obr. 2.12

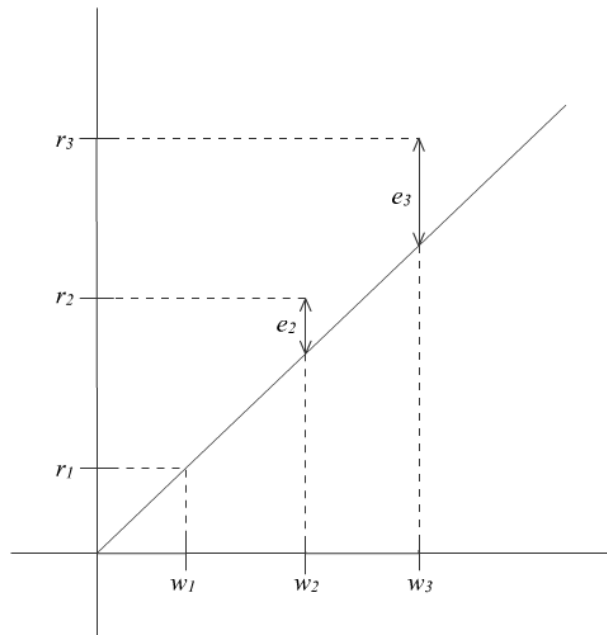
Protože lze předpokládat, že uživatel nenastaví střed objektu přesně do středu optické osy, je hodnota r_i průměrem vzdálenosti zleva a zprava od středu (viz obr. 2.12).



obr. 2.13

2.2.3. Návrh řešení

U čočky s distorzí mají naměřené hodnoty w_i a r_i mezi sebou nelineární závislost (viz obr. 2.14). Jinak řečeno lineární funkce $L(w) = w \cdot \frac{r_1}{w_1}$ neprochází ostatními naměřenými daty $L(w_i) \neq r_i$. Pokud velikost objektu w_1 má po průchodu optickou soustavou na čipu velikost r_1 , chtěli bychom, aby w_2 měl velikost $L(w_2)$. U čočky s distorzí tomu tak však není. Zde je uvedeno jedno z možných řešení.



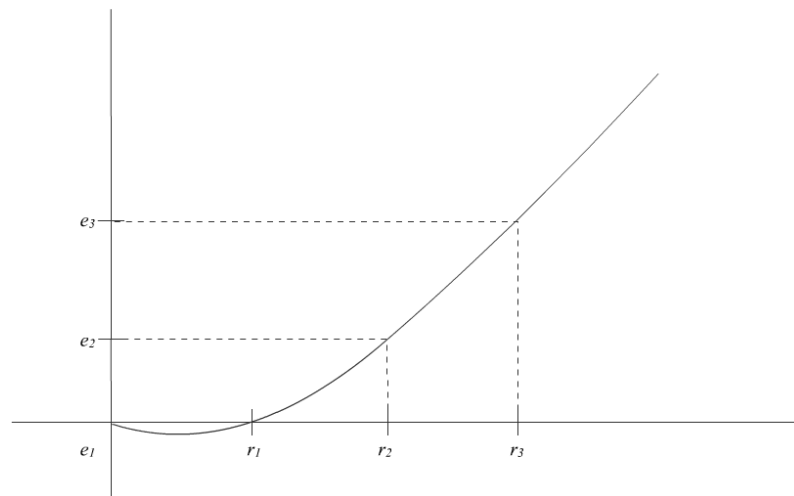
obr. 2.14

Považujme funkci $L(w)$ za správný vztah hodnot w a r . Členy e_i posloupnosti

$$e_i = L(w_i) - r_i = w_i \cdot \frac{r_1}{w_1} - r_i$$

nám určují chyby vzniklé distorzí čočky (viz obr. 2.14). Tyto hodnoty proložíme polynomem¹ $P(r)$ (viz obr. 2.15).

¹ Stupeň polynomu závisí na počtu měření.



obr. 2.15

2.2.4. Odstranění distorze

Při rekonstrukci bodu v prostoru (podrobně viz kap. 2.2) korigujeme nalezený bod v rovině čipu polynomem $P(r)$ následujícím způsobem.

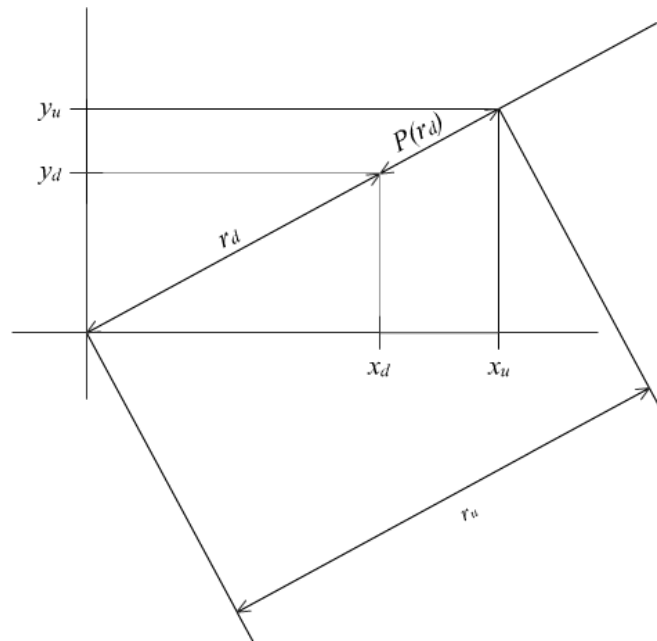
Nechť má nalezený bod v obrázku souřadnice (x, y) . Z těchto souřadnic můžeme spočítat souřadnice bodu na čipu (x_{Chip}, y_{Chip}) (podrobně viz kap. 2.1.2). Vzdálenost těchto souřadnic od středu čipu označme r_d . Odhad chyby ve vzdálenosti r_d od středu čipu zjistíme dosazením do polynomu $P(r_d)$. Odhadovaná vzdálenost od středu r_u bez zkreslení čočky je pak dána vztahem

$$r_u = r_d - P(r_d).$$

Z podobnosti trojúhelníků (viz obr. 2.16) lze spočítat bod bez distorze v souřadném systému čipu následujícími vztahy.

$$x_u = \frac{x_{chip} \cdot r_u}{r_d},$$

$$y_u = \frac{y_{chip} \cdot r_u}{r_d}.$$



obr. 2.16

2.3. Rekonstrukce

2.3.1. Nalezení bodu ve snímku

Nasnímané obrázky z kamery mají velmi tmavé pozadí a jasně rozlišitelný světlý bod (viz obr. 2.17) (podrobně je pokus popsán v kapitole 4.2.). K segmentaci obrázku je tedy dostačující použít metodu konstantního prahování. Konstantní prahování nám rozdělí obrázek do dvou množin pixelů.

$$M_B = \{Jas(x, y) < T; x = \{1 \dots X_{Res}\}, y = \{1 \dots Y_{Res}\}\}$$

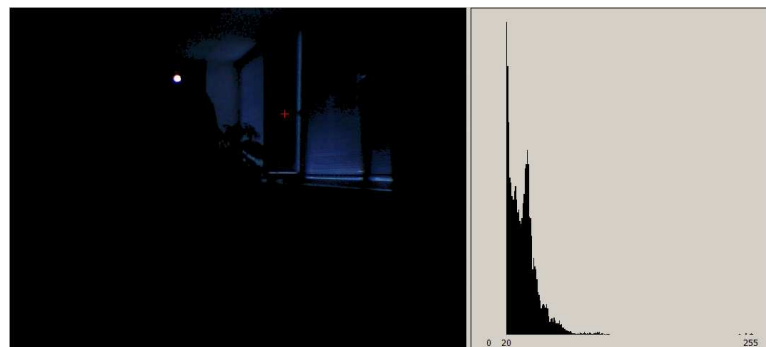
$$M_F = \{Jas(x, y) \geq T; x = \{1 \dots X_{Res}\}, y = \{1 \dots Y_{Res}\}\}, \text{ kde}$$

T je prahová hodnota jasu, x a y jsou souřadnice pixelů v obrázku. Množina M_B obsahuje pixely pozadí a množina M_F pixely oblasti s výrazným jasnem (větším než T).



obr. 2.17

Nalezení bodu bylo provedeno zprůměrováním x-ových a y-ových souřadnic pixelů v množině M_F . Na obrázcích 2.17– 2.20 je zobrazeno nalezení souřadnic (červený kříž) při použití prahových hodnot¹ 20, 50, a 150 a histogramy obrázků, kde hodnoty pixelů z množiny M_B jsou nahrazeny černou barvou.

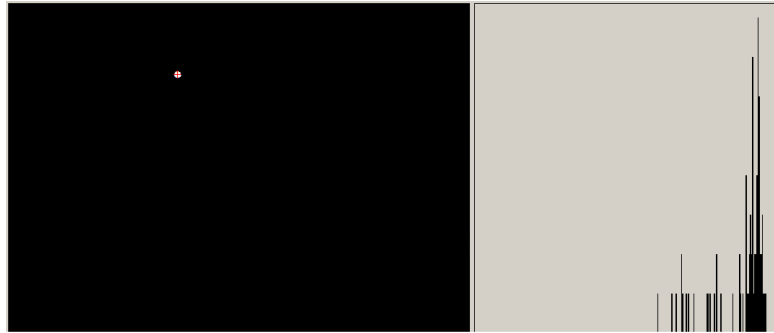


obr. 2.18



obr. 2.19

¹ Rozmezí jasu pixelů u použitého obrázku je 0 – 255.



obr. 2.20

2.3.2. Synchronizace snímků

Dalším z problémů rekonstrukce je synchronizace sekvencí. Každý snímek sekvence má určitou expoziční dobu. Tato doba se mění v závislosti na světelných podmínkách. Protože kamery mají odlišnou polohu a orientaci (dokonce to mohou být i různé typy kamer), světelné podmínky pro každou z nich jsou různé. Proto jsou různé i expoziční časy jednotlivých snímků a počet zaznamenaných obrázků sekvence za vteřinu. Z těchto důvodů nemůžeme očekávat, že obrázky z jednotlivých kamer budeme dostávat synchronizovaně.

Řekněme, že dostaneme z jedné kamery snímek exponovaný v rozmezí časů Δt_L a druhý snímek zachycuje scénu v rozmezí časů Δt_R . Pokud se snímáný objekt pohybuje, je zachycen na snímcích rozmazaně. Předpokládejme, že objekt se během expozice pohyboval po přímce rovnoměrnou rychlostí. Pokud najdeme střed objektu na snímku (viz kap. 2.3.1) nalezneme jeho polohu v čase $t_L = \frac{\Delta t_L}{2} + t_{L1}$, popřípadě $t_R = \frac{\Delta t_R}{2} + t_{R1}$ (v polovině času expozice). Často ale $t_L \neq t_R$ vlivem výše uvedených vlivů. Během času $t_\xi = |t_L - t_R|$ může objekt urazit nezanedbatelnou vzdálenost $s = t_\xi \cdot v$ a je vhodné tuto skutečnost při rekonstrukci zohlednit. Nabízí se následující řešení. Vezmeme souřadnice nalezeného bodu v obraze z několika předchozích snímků a proložíme je polynomem následujícím způsobem.

Označme t_i^L čas v polovině expozice (dále jen čas) posledního zaznamenaného snímku levé kamery (i -tého), t_j^R čas posledního zaznamenaného snímku (j -tého) pravé kamery. Souřadnice $x_{t_i^L}$ je x-ová souřadnice nalezeného bodu v obraze levé kamery v čase t_i^L , obdobně označme $y_{t_i^L}$, pro y-ovou souřadnici v čase t_i^L a $x_{t_j^R}$, $y_{t_j^R}$ souřadnice pro pravou kameru v čase t_j^R . Řekněme, že nastane situace, kdy dostaneme snímek z pravé kamery (aktuální čas $t = t_j^R$). Z levé kamery máme ale jen snímek z času t_i^L (víme, že často $t_i^L \neq t_j^R$). Otázka zní, kde by se nacházel objekt levé kamery v čase t . K tomu nám může posloužit interpolace polynomem.

Pro nalezení hodnoty x pro levou kameru v čase t sestavíme soustavu rovnic pro výpočet koeficientů polynomu z několika předchozích hodnot¹.

	i-2	i-1	i
$x_{t_j^R}$	$x_{t_{j-2}^R}$	$x_{t_{j-1}^R}$	$x_{t_j^R}$
t_j^R	t_{j-2}^R	t_{j-1}^R	t_j^R

tab. 2.1

Musí platit

$$x_{t_{j-2}^R} = a_0 + a_1 t_{j-2}^R + a_2 (t_{j-2}^R)^2$$

$$x_{t_{j-1}^R} = a_0 + a_1 t_{j-1}^R + a_2 (t_{j-1}^R)^2$$

$$x_{t_j^R} = a_0 + a_1 t_j^R + a_2 (t_j^R)^2.$$

Maticový zápis

$$A = \begin{bmatrix} 1 & t_{j-2}^R & (t_{j-2}^R)^2 \\ 1 & t_{j-1}^R & (t_{j-1}^R)^2 \\ 1 & t_j^R & (t_j^R)^2 \end{bmatrix},$$

$$x = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix},$$

$$b = \begin{bmatrix} x_{t_{j-2}^R} \\ x_{t_{j-1}^R} \\ x_{t_j^R} \end{bmatrix}.$$

Řešením soustavy rovnic $Ax = b$ dostáváme hodnoty koeficientů polynomu P . Požadovanou hodnotu x -ové složky předpokládané pozice objektu snímaného levou kamerou dostaneme dosazením času t do polynomu P . Obdobně lze postupovat při výpočtu y -ové složky. Pokud dostaneme nejprve snímek z levé kamery, celá situace je inverzní. Po nalezení souřadnic objektu z levé a pravé kamery můžeme přistoupit k rekonstrukci polohy bodu v prostoru.

¹ Ukázka výpočtu je při použití třech hodnot a polynomu druhého stupně

2.3.3. Proces rekonstrukce

Pro provedení rekonstrukce bodu v prostoru je potřeba znát následující vstupní hodnoty

- polohy kamer
- orientace kamer
- ohniskové vzdálenosti
- velikosti čipů
- rozlišení kamer
- souřadnice objektu na snímcích (levé a pravé kamery)

Pokud známe všechny tyto informace, můžeme popsat paprsky, které přilétly na čip levé a pravé kamery z hledaného objektu.

Pro pravou kameru vypočteme hodnoty úhlů α a β ze vztahů z kap. 2.1.2

$\alpha = \arctan\left(\frac{|P_{L_y} L|}{\|D\|}\right)$ a $\beta = \arctan\left(\frac{|P_{L_x} L|}{\|D\|}\right)$. Paprsek ve sférických souřadnicích má pak tvar

$$p_{L\theta} = \theta_R + \alpha$$

$$p_{L\phi} = \phi_R - \beta$$

$$p_{L\rho} = 1.$$

Třetí složka sférického vektoru paprsku (jeho délka) nemá pro nás zatím význam, proto je nastavena na 1.

Přejdeme do reprezentace vektoru v kartézských souřadnicích (viz kap. 2.1.2). Nyní máme směr paprsku i jeho polohu (poloha ohniska pravé kamery). Obdobně budeme postupovat pro levou kameru.

Spočítáme parametry t a v pro minimální vzdálenost mezi dvěma přímkami následujícím způsobem (podrobně viz [4]).

$$p_1 = u_0 + ut$$

$$p_2 = v_0 + vs$$

kde p_1, u_0, u, p_2, v_0, v jsou vektory v prostoru R^3 a s a t jsou reálná čísla. Pak druhá mocnina vzdálenosti mezi dvěma přímkami je:

$$d^2 = \|p_1 - p_2\|^2 = (u_0 + ut - v_0 - vs) \cdot (u_0 + ut - v_0 - vs)$$

Tuto rovnici lze rozepsat a přeskupit do následujícího tvaru:

$$d^2 = \|u\|^2 t^2 + 2(uu_0 - uv_0)t - 2(uv)st + 2(vv_0 - vu_0)s + \|v\|^2 s^2 + \|u_0\|^2 + \|v_0\|^2 \quad (2.3)$$

V této rovnici je násobení mezi vektory skalární. Pro zjednodušení zápisu zavedeme následující substituce:

$$A = \|u\|^2,$$

$$B = 2(uu_0 - uv_0),$$

$$C = 2(uv),$$

$$D = 2(vv_0 - vu_0),$$

$$E = \|v\|^2,$$

$$F = \|u_0\|^2 + \|v_0\|^2.$$

Ted' lze rovnici (2.3) přepsat do tvaru:

$$d^2 = At^2 + Bt - Cst + Ds + Es^2 + F.$$

Pokud výrazy A a E jsou nenulové, tato rovnice představuje paraboloid v třírozměrném prostoru. Pokud je splněna následující podmínka má daný problém pouze jedno řešení.

$$\det \begin{pmatrix} 2A & -C \\ -C & 2E \end{pmatrix} \neq 0, \text{ pak}$$

$$s = \frac{2AD + BC}{C^2 - 4AE} \text{ a}$$

$$t = \frac{Cs - B}{2A}.$$

Nyní známe parametry t a v . Dosazením do výrazů p_1 a p_2 získáme dva vektory. Jsou to vektory pozic na přímkách, kde vzdálenost přímek je minimální. Hledaný bod nalezneme zprůměrováním těchto dvou vektorů.

2.4. Vlastnosti modelu

2.4.1. Přesnost kalibrace

Při kalibraci vnějších parametrů systému může dojít k nepřesnostem v měření vzdálenosti mezi kamerami a při umístění kalibračního bodu před kamery. Poslední nepřesnost může vzniknout rotací kamery okolo optické osy. V omezujících podmínkách jsme ale tuto rotaci zakázali, proto není v modelu z kap 2.1.2 zahrnuta a nebudeme s ní dále pracovat. Situace, kdy špatně změříme vzdálenost mezi kamerami, je také nezajímavá. Největším problémem kalibrace je přesné umístění kalibračního bodu X před kamery.

Uvažme situaci, kdy kalibrační bod při kalibraci neleží na pozici $X = (0, l, 0)$, ale na pozici $X_\xi = (x_\xi, l + y_\xi, 0)$, kde x_ξ a y_ξ jsou chyby umístění bodu v x-ové, y-ové souřadnici (chybu v ose z zatím neuvažujeme). Po rozepsání rovnic 2.1 a 2.2 pro výpočet orientace pravé kamery ve sférických souřadnicích z kap 2.1.2 dostáváme výrazy

$$\theta = \lambda - \left(\arctan\left(\frac{l}{k}\right) + \arctan\left(\frac{\frac{w}{2} - x \cdot \frac{w}{\text{Res}_x}}{\|D\|}\right) \right) \quad (2.4)$$

$$\phi = \arctan\left(\frac{\frac{h}{2} - y \cdot \frac{h}{\text{Res}_y}}{\|D\|}\right) + \frac{\pi}{2} \quad (2.5)$$

$$\rho = \|D\|$$

Při kalibraci s chybným bodem X_ξ je potřeba změnit hodnoty l a k v modelu na hodnoty s chybou.

$$l_\xi = l + y_\xi$$

$$k_\xi = k - x_\xi$$

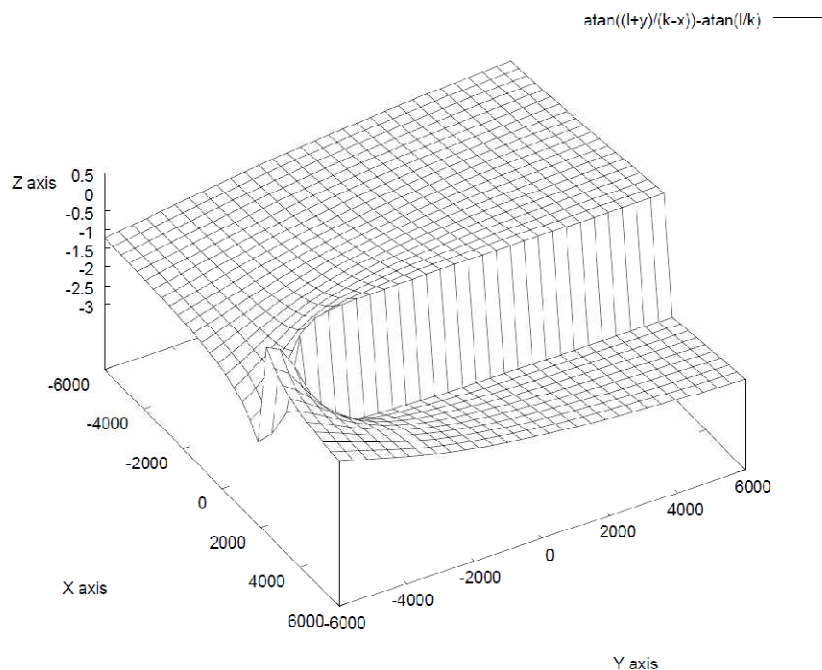
Z výrazů 2.4 a 2.5 pro výpočet orientace kamery je patrné, že hodnoty l a k se vyskytují pouze ve výrazu pro výpočet úhlu θ . Dosadíme chybné hodnoty l_ξ a k_ξ .

$$\theta_\xi = \lambda - \left(\arctan\left(\frac{l + y_\xi}{k - x_\xi}\right) + \arctan\left(\frac{\frac{w}{2} - x \cdot \frac{w}{\text{Res}_X}}{\|D\|}\right) \right)$$

Porovnáním výrazu θ_ξ a θ a úpravou dostaneme výraz

$$\xi_\theta(x_\xi, y_\xi, l, k) = \|\theta_\xi - \theta\| = \left\| \arctan\left(\frac{l + y_\xi}{k - x_\xi}\right) - \arctan\left(\frac{l}{k}\right) \right\|$$

Tato funkce nám určuje chybu v úhlu θ při posunutí kalibračního bodu o x_ξ , y_ξ , při daných vzdálenostech k a l . Z uvedeného výrazu si lze těžko představit jaký bude mít vliv posunutí kalibračního bodu na úhel θ . Proto je pro představu na obrázku 2.21 zobrazena funkce $\xi(x_\xi, y_\xi, 5000, 2000)$.

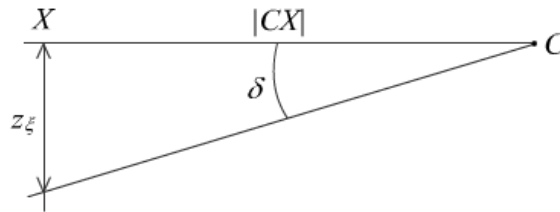


obr. 2.21

Pokud pohneme kalibračním bodem o vzdálenost z_ξ v ose z , jeho souřadnice budou $X_\xi = (0, l, z_\xi)$ (viz obr. 2.22). Platí následující vztahy:

$$\text{Vzdálenost } |CX| = \sqrt{k^2 + l^2}$$

$$\text{úhel } \delta = \arctan\left(\frac{z_{\xi}}{|CX|}\right).$$

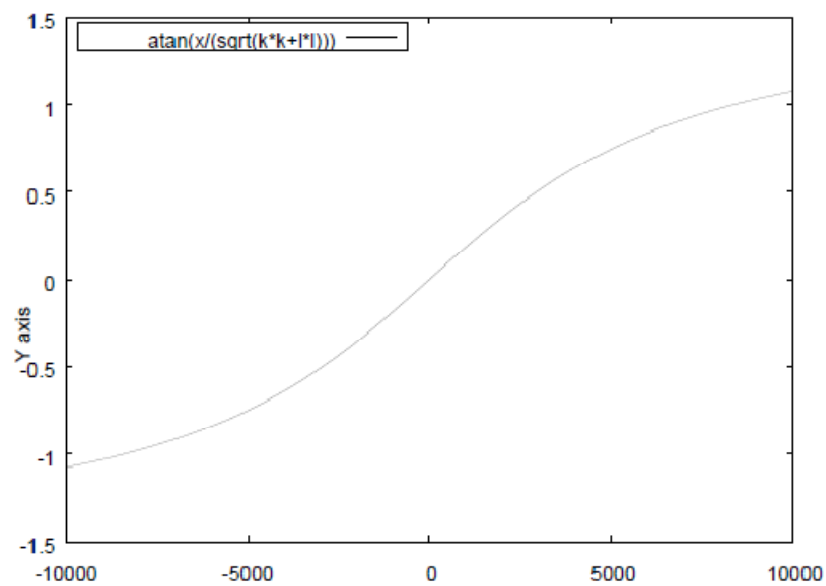


obr. 2.22

Velikost úhlu δ určuje přímo chybu úhlu ϕ_{ξ} při kalibraci orientace kamery. Chybová funkce pro úhel ϕ je tedy.

$$\xi_{\phi}(z_{\xi}, k, l) = \arctan\left(\frac{z_{\xi}}{\sqrt{k^2 + l^2}}\right)$$

Ukázka chybové funkce pro úhel $\xi_{\phi}(z_{\xi}, 5000, 2000)$.



X axis

obr. 2.23

2.4.2. Přesnost nalezení objektu

Protože kamery mají vždy omezenou rozlišovací schopnost, ať už jde o rozlišení, nebo počet úrovní jasu jednotlivých pixelů, souřadnice rekonstruovaného bodu v prostoru budou také diskrétní. Pro začátek zanedbáme jas jednotlivých pixelů. Označme šířku pixelu na čipu w_{Pixel} (pro výpočet viz kap. 2.1.2), vzdálenost roviny čipu od roviny pozorovatele l a ohniskovou vzdálenost kamery f (viz obr. 2.24), pak

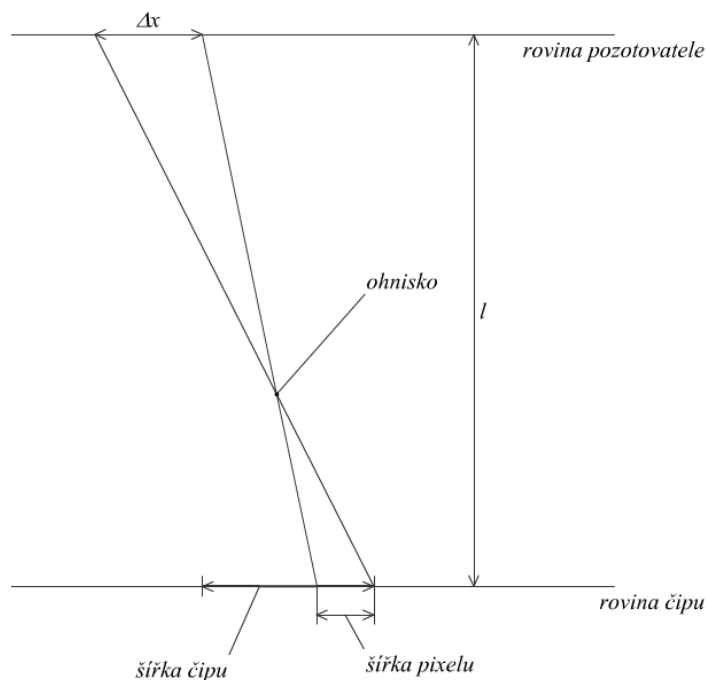
$$\Delta x = (l - f) \cdot \frac{w_{Pixel}}{f}, \text{ obdobně}$$

$$\Delta y = (l - f) \cdot \frac{h_{Pixel}}{f}.$$

Pohybem objektu odrážejícího světlo v rovině pozorovatele v rozmezí vzdálenosti Δx (Δy) bude paprsek spadat vždy do stejného senzoru na čipu.

Ve skutečnosti ale objekt často odráží světlo do více senzorů najednou a hranice objektu není dokonale ostrá. Proto když uvážíme rozmezí jasu jednotlivých pixelů, hranice objektu při pohybu nejprve postupně snižuje hodnoty jasu jednoho pixelu, než je jas nulový. Jinak řečeno – otázka zní, o kolik bychom museli posunout kameru, aby sejmutý obrázek byl jiný než před posunutím. Využitím této skutečnosti je zde teoretická možnost rozlišení

$$\Delta x = (l - f) \cdot \frac{w_{Pixel}}{j \cdot f}, \text{ kde } j \text{ je počet možných jasů.}$$



obr. 2.24

V našem modelu (při použití konstantního prahování) s takovou přesností samozřejmě nemůžeme počítat. Vlivem šumu a zkreslení optickou soustavou nelze obecně této přesnosti nikdy dosáhnout¹.

2.4.3. Viditelnost bodu z kamer

Na čip kamery dopadnou jen paprsky odražené od objektů ležících v pravidelném čtyřbokém jehlanu ubíhajícího do nekonečna. Vrchol tohoto jehlanu leží v ohnisku kamery. Hrany stěn jehlanu jsou definovány čtyřmi parametrickými přímkami ve sférických souřadnicích.

Označme $\varphi = \arctan\left(\frac{w}{2\|D\|}\right)$ polovinu velikosti zorného úhlu kamery v horizontální rovině,

$\omega = \arctan\left(\frac{h}{2\|D\|}\right)$ polovinu velikosti zorného úhlu kamery ve vertikální rovině, pak tyto čtyři

parametrické přímky² ve sférických souřadnicích definují hrany jehlanu.

$$\theta_{LT} = \theta + \varphi, \phi_{LT} = \phi - \omega, \rho_{LT} = t$$

¹ Podrobná analýza těchto skutečností je nad rámec této práce

² indexy L, R, B, T jsou z anglického left (levá), right (pravá), bottom (spodní), top (horní)

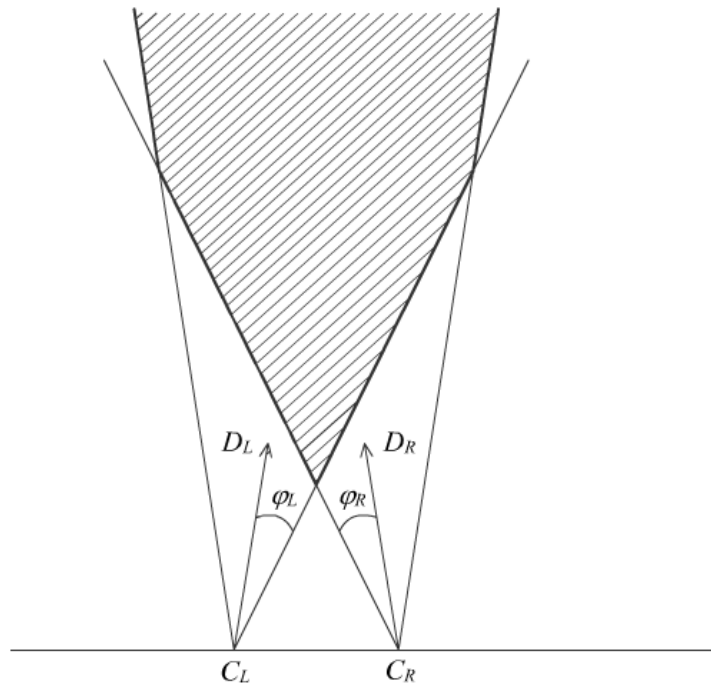
$$\theta_{RT} = \theta - \varphi, \phi_{RT} = \phi - \omega, \rho_{RT} = t$$

$$\theta_{LB} = \theta + \varphi, \phi_{LT} = \phi + \omega, \rho_{LB} = t$$

$$\theta_{RB} = \theta - \varphi, \phi_{LT} = \phi + \omega, \rho_{RB} = t, \text{ kde}$$

$t \in \{1..∞\}$ a θ, ϕ jsou úhly definující vektor D ve sférických souřadnicích (viz kap. 2.1.2).

Pokud chceme, aby byl objekt snímán oběma kamerami, musí ležet v průniku těchto dvou jehlanů (viz obr. 2.25). Objekt zmizí ze záběru jedné kamery doleva nebo doprava, pokud bude úhel mezi objektem, ohniskem kamery a optickou osou větší než φ . Obdobně je tomu tak ve vertikální ose a úlem ω .



obr. 2.25

3. Návrh implementace

3.1. Použité technologie

Protože jsem zastáncem objektového přístupu v každém případě, v kterém je možné ho použít, dal jsem si za cíl použít k popisu modelu plně objektovou technologii. Pro implementaci jsem použil jazyk C# a programový balík .NET od firmy Microsoft, s kterým mám nejvíce zkušeností. Tato technologie používá řízené prostředí, ve kterém není jazyk kompilován přímo do strojového kódu, ale do tzv. intermediate language (další IL). Kompilovaný kód v jazyce IL je poté možné spustit na virtuálním stroji (Common language runtime – CLR). Konkrétní CLR je závislý na platformě a poskytuje správu paměti (garbage collector), obsluhu výjimek a odstiňuje programátora od konkrétní platformy. Jazyk IL není závislý na platformě a dokonce ani na programovacím jazyce. Je tedy možné kombinovat různé programovací jazyky s dostupným IL překladačem. Tato technologie je pro programátora velmi pohodlná a poskytuje širokou sadu funkcionalit. Má ale jednu velkou nevýhodu, a tou je nízká rychlost. Proto byla použita pouze na nejvyšší vrstvu modelu a grafického uživatelského rozhraní.

Pro vrstvu, která pracuje s kamerami, byla použita technologie Microsoft DirectShow a jazyk C++. Tato technologie je v neřízeném prostředí. Knihovna obsahuje implementace rozhraní pro nejrůznější datové toky. Jedním z nich je i vstup z webových kamer. Výsledná kompilace je specifická pro daný typ procesoru.

Pro přechod od neřízeného k řízenému prostředí byl použit jazyk C++ v prostředí .Net. Tento jazyk jako jediný umožňuje snadno kombinovat tyto dva světy. Dokonce je možné sloučit řízený a neřízený kód v jednom projektu. V této technologii byl implementován wrapper, které obaluje neřízený kód a poskytuje informace prostřednictvím událostí do řízeného prostředí.

Poslední vrstvou systému je vrstva 3D vizualizace. K implementaci byla použita technologie Microsoft DirectX.

3.2. Offline verze

První verze implementace ověřuje model na již zaznamenaných sekvencích z kamer. K záznamu byly použity dva počítače, pak byly sekvence zpětně analyzovány implementovaným softwarem. Část z kapitoly o synchronizaci snímků nebyla implementována, protože vstupní sekvence obsahují vždy konstantní počet snímků. Pokud nestačí expoziční doba daná frekvencí snímání (fps) na expozici, snímek se opakuje v sekvenci znovu, dokud není pořízen nový. Teorie z kapitoly 2.3.2 byla ověřena až druhou verzí implementace. Detailnější popis tříd a uživatelská příručka jsou uvedeny v příloze.

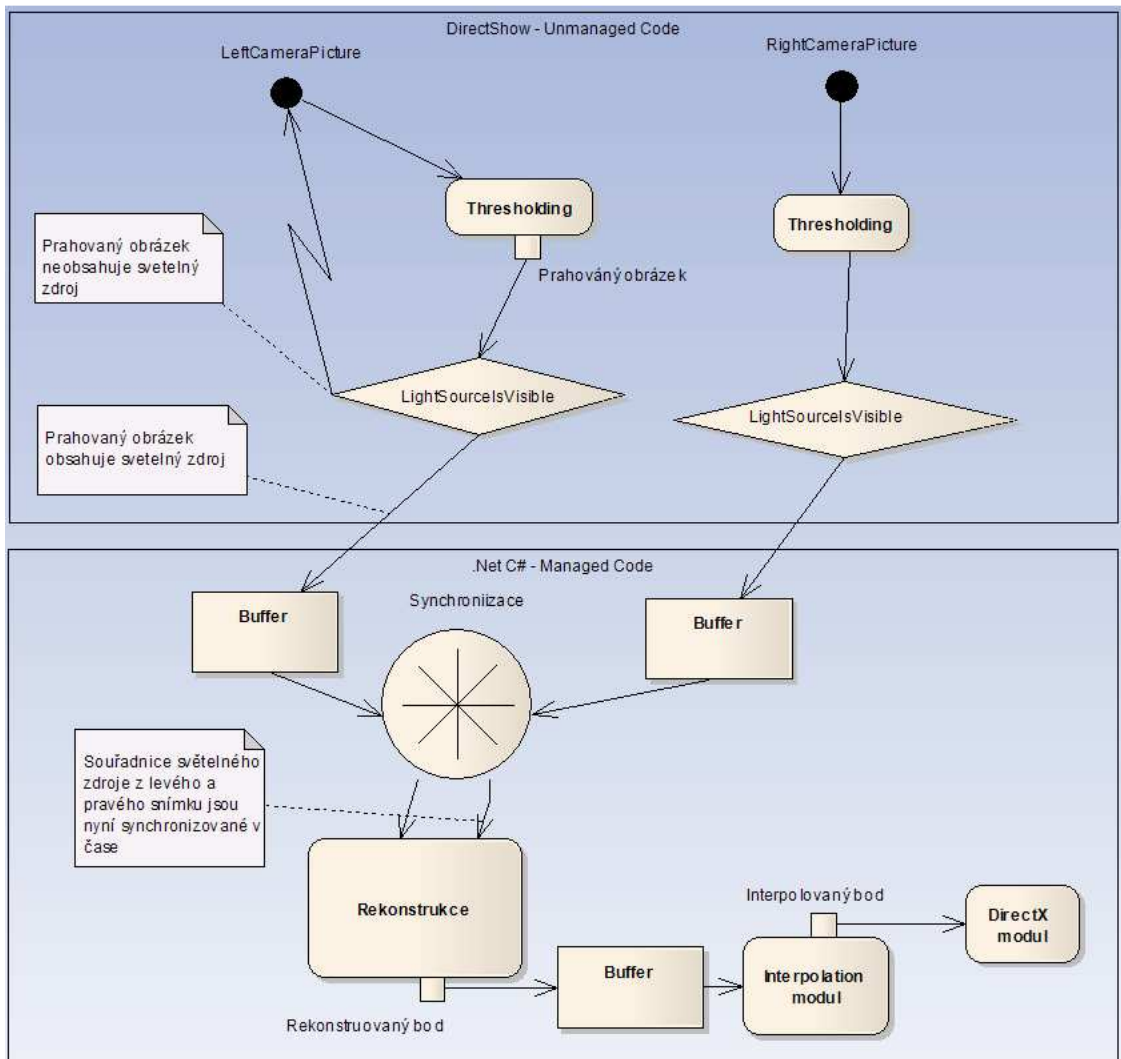
3.3. Realtime verze

Tato verze funguje v reálném čase a používá všechny metody z kapitoly 2.3. Obsahuje 3D ukázkovou aplikaci, kde je zobrazena rekonstrukce v reálném čase. Uživatelská příručka je v příloze.

3.3.1. Popis algoritmu

Celý proces rekonstrukce je schematicky zobrazen na obrázku 3.1. Kamery pořizují snímky v reálném čase. Ty jsou pak prahovány. Pokud je v prahovaném obrázku větší počet pixelů, než je nastavená minimální hodnota¹, pak jsou souřadnice těchto pixelů zprůměrovány a odeslány spolu s identifikací kamery přes událost. Dále se pokračuje v řízeném prostředí. Zde jsou obrázky zachyceny přes událost a přidány do bufferu. Zobrazovací directX modul při požadavku na překreslení vyvolá událost. Tato událost je odchycena v modulu interpolation. Zde jsou z bufferu posledních několika rekonstrukcí spočteny souřadnice pozice výsledného bodu a jsou přiřazeny do argumentů události. V modulu directX jsou tyto souřadnice použity k vykreslení rekonstruované pozice.

¹ Záleží na velikosti světelného zdroje a jeho vzdálenosti od kamer - při testování ve vzdálenosti 1-3 metry se osvědčila hodnota 5 pixelů



obr. 3.1

Interpolace je tedy použita v aplikaci na dvou místech. Nejprve pro synchronizaci snímků – tak jak je popsáno v kapitole 2.3.2. Rekonstrukce je tedy prováděna na vstupních souřadnicích nalezených světelných bodů z kamer, které jsou synchronizovány v čase. Tyto rekonstrukce jsou pak uloženy do bufferu (zásobník 3D rekonstruovaných bodů). Ve chvíli, kdy vykreslovací modul generuje událost pro vykreslení, je spočtena další interpolací aktuální pozice z několika předchozích výsledků rekonstrukce pro zvýšení plynulosti pohybu.

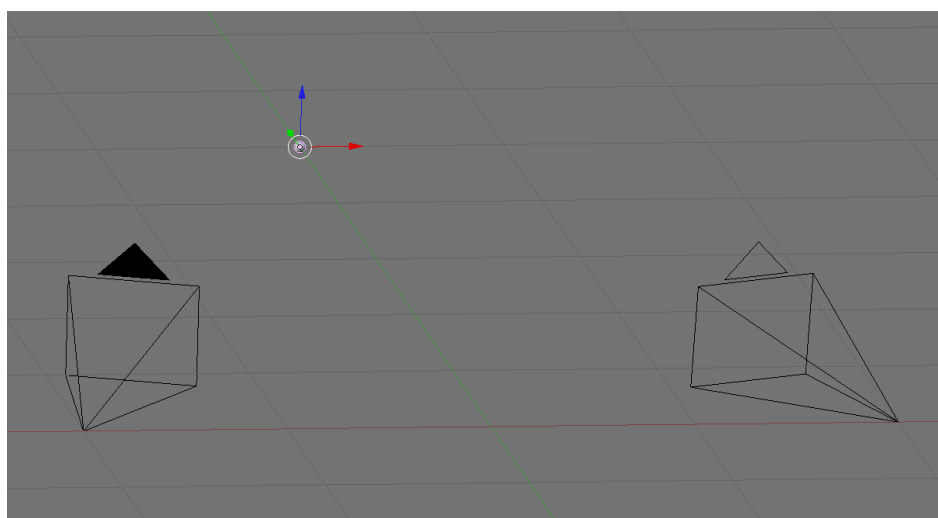
Pro synchronizaci kamer se nakonec nejlépe osvědčila lineární interpolace. Respektive při interpolaci polynomem vyššího stupně nebyly zaznamenány subjektivně lepší výsledky. Experimenty s algoritmem pro interpolaci rekonstruovaných bodů jsou popsány v kapitole 4.3.

4. Testování

Testování systému bylo provedeno jak na umělých, tak reálných datech. Na první verzi implementace byly provedeny testy umělých dat a ověření algoritmu rekonstrukce. Dynamické vlastnosti systému byly ověřeny druhou verzí implementace.

4.1. Umělá data

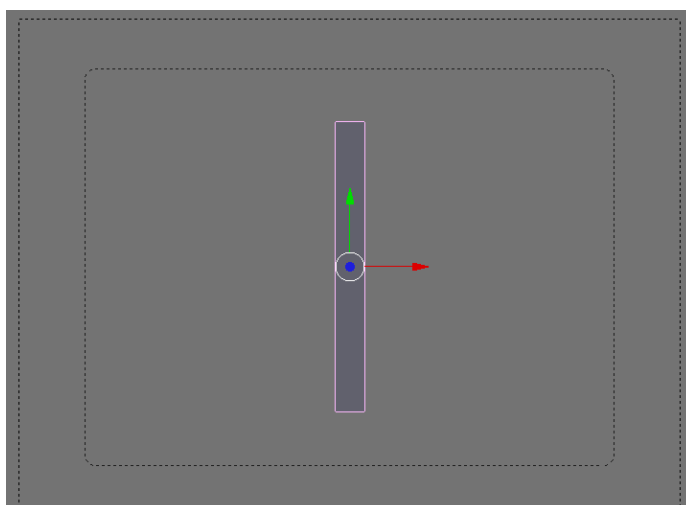
Data byla generována modelovacím a animačním nástrojem Blender 2.42a. Byla vytvořena jednoduchá scéna (viz obr. 4.1).



obr. 4.1

4.1.1. Kalibrace

Ve scéně byly rozmístěny dvě kamery na známých pozicích se známými orientacemi. Objekt byl umístěn před kamery, tak jak je popsáno v kapitole 2.1.2. Nejprve bylo provedeno měření velikosti čipu (viz obr. 4.2). Ohnisková vzdálenost byla nastavena na fixní hodnotu a byl renderován objekt známé šířky a délky. Tím bylo ověřeno, že blender pracuje se čtvercovými pixely. Při ohniskové vzdálenosti 35 byla spočítána šířka (výška) čipu 32 (24). Bylo provedeno i měření distorze čoček. Podle očekávání byl polynom konstantní nulová funkce (podrobně viz kap. 2.2.3), takže žádná distorze čočky v renderovaných obrázcích není.



obr. 4.2

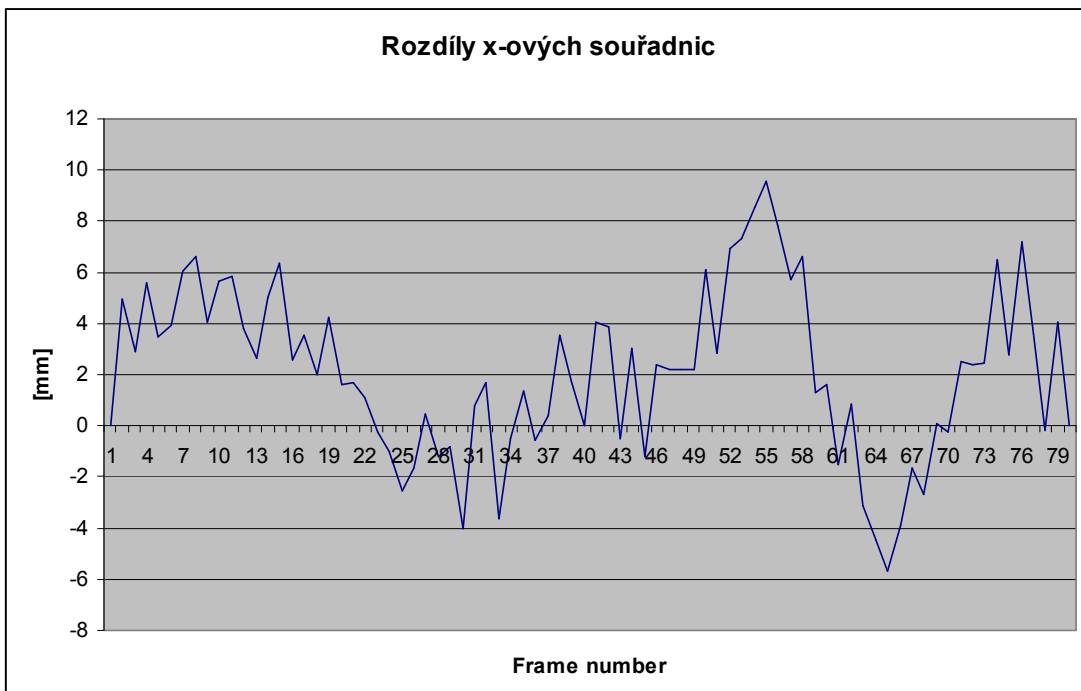
Souřadnice kamer byly $(x,y,z) = (-2000,0,0)$ pro levou a $(2000,0,0)$ pro pravou kameru. Kalibrační bod byl umístěn na pozici $(0, 5000, 0)$. Úhly kamer při kalibraci jsou zobrazeny v tabulce 4.1. Všechny uvedené hodnoty úhlů jsou ve stupních, souřadnice v milimetrech.

	Vypočtené hodnoty	Hodnoty z blenderu	Chyba
Levá kamera úhel ϕ	85,1009	85	0,1009
Levá kamera úhel θ	69,9987	70	0,0013
Pravá kamera úhel ϕ	95,0615	95	0,0615
Pravá kamera úhel θ	120,0117	120	0,0117

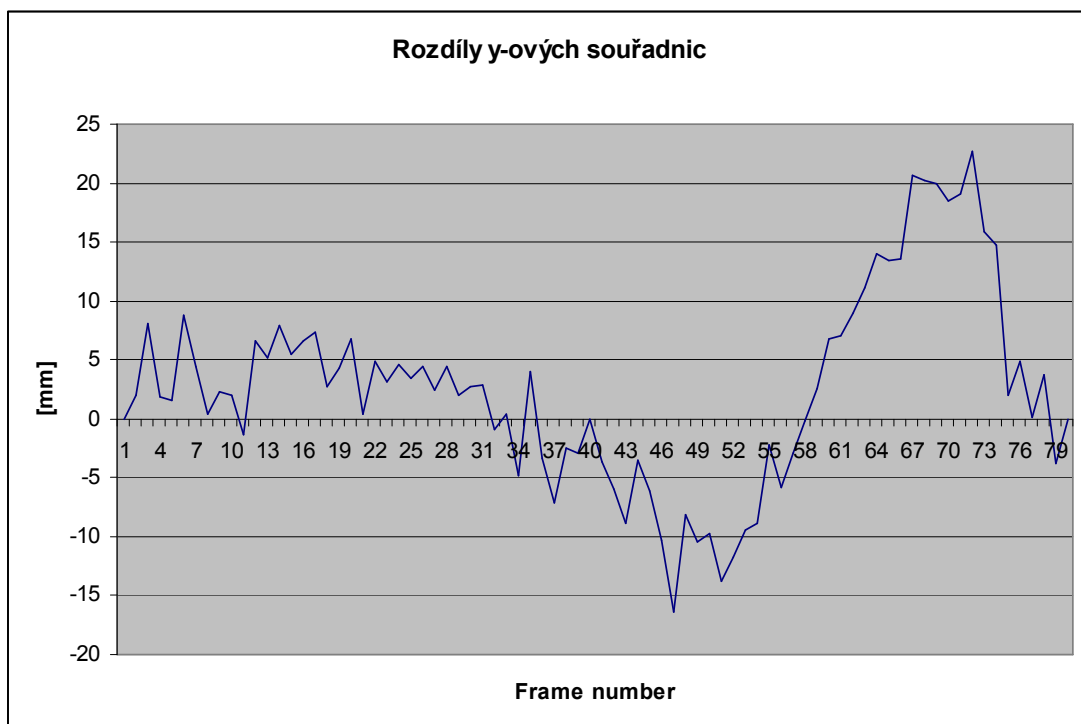
tab. 4.1

4.1.2. Rekonstrukce

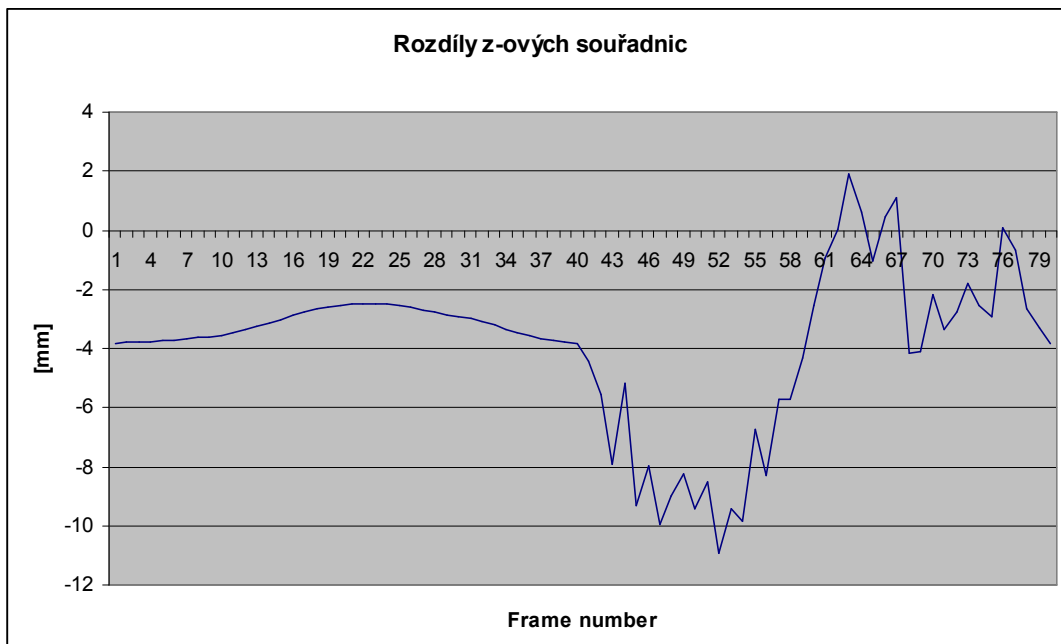
Po kalibraci byla provedena rekonstrukce souřadnic bodu v prostoru. V animaci se bod pohyboval nejprve v rovině os x a z, pak v rovině os x a y. Na grafech 4.1 – 4.3 jsou postupně znázorněny odchylky původních souřadnic bodů v blenderu a rekonstruovaných souřadnic, v grafu 4.4 vzdálenost původních a rekonstruovaných bodů.



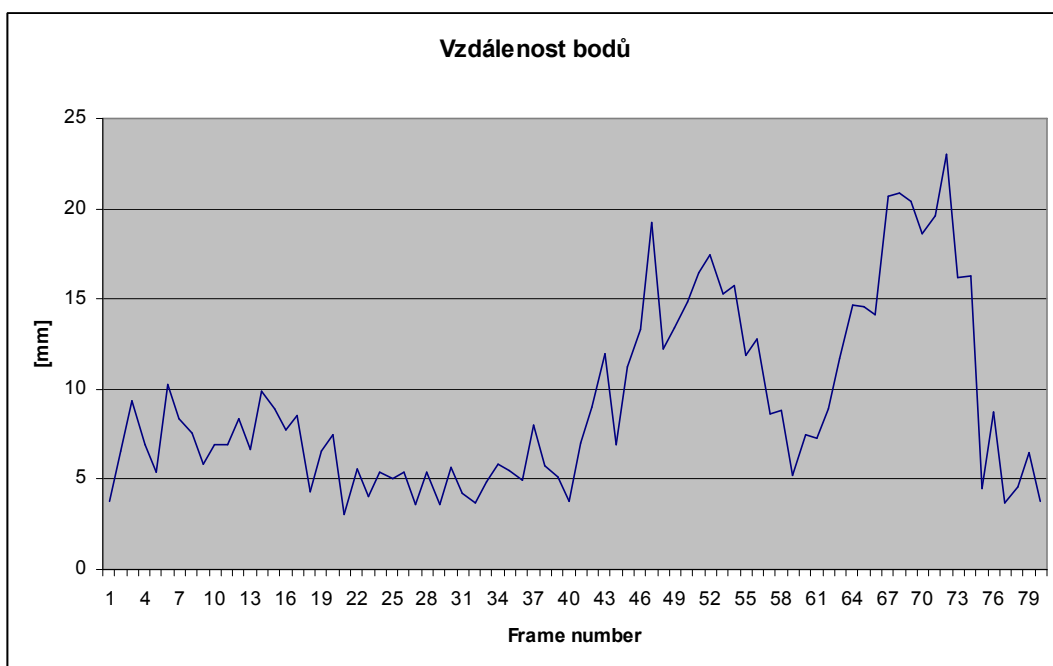
graf 4.1



graf 4.2



graf 4.3



graf 4.4

Z výsledků je patrné, že rekonstruovaný bod se nevzdaloval od původního bodu o více než 20 mm. Nejvíce je systém citlivý na osu y. Intuitivně informací o x-ové a z-ové pozici bodu je v nasnímaných obrázcích více. Při výpočtu souřadnice v ose y (vzdálenost od kamer) je chyba největší. Grafy z ostatních měření jsou obsaženy v příloze.

4.2. Reálná data – offline verze aplikace

Testování na reálných datech je problematictější. Nalezené rekonstruované body nemůžeme porovnat s původními pozicemi bodu v prostoru, protože tyto pozice neznáme. Samozřejmě je možné odměřit pozici bodu u jednotlivých snímků, těžko ale u celé sekvence. Navíc tato měření jsou také zatížena chybou, která je často větší než chyba rekonstrukce. Přesto několik bodů bylo takto naměřeno, viz kap. 4.2.2.

K nahrání sekvencí byly použity dvě kamery Creative WebCam Live (viz obr. 4.3) a software Creative WebCam Center. Kamery mají rozlišení 640x480 pixelů. Frekvence snímků je maximálně 30 fps (podle světelných podmínek).

Sekvence byly synchronizovány zapnutím čelové lampy. Po nahrání sekvencí, byl nalezen snímek, kde se objevil světlý bod. Od těchto snímků (pro levou a pravou kameru) byly sekvence exportovány programem Adobe Premiere Pro 1.5.



obr. 4.3

Pro generování světelného bodu byla použita čelní lampa s osmi diodami (viz obr. 4.4). Byla přelepena neprůhlednou černou lepenkou, do které byl vyříznut otvor pro průchod světla. Mezi lepenku a sklo lampy byl vložen bílý papír jako stínítko. Pak lampa svítila na papír a nedocházelo k nežádaným efektům při přímém osvětlení kamery. Kamery byly nastaveny na minimální dobu expozice, aby světelný bod byl v obrázku snadno rozpoznatelný.



obr. 4.4

4.2.1. Kalibrace

Obě kamery byly v jedné rovině na dvou monitorech. Souřadnice (0,0,0) byla přesně v rozhraní mezi monitory (viz obr. 4.5). Pozorovatel byl přesně ve středu mezi kamerami. Toho bylo docíleno provázkem, v jehož polovině byla značka. Dále bylo nutné nastavit světelný bod do stejné výšky od země, jako jsou kamery. Změřené vzdálenosti jsou v tabulce.

Vzdálenost kamer	714
Vzdálenost pozorovatele	1123



obr. 4.5

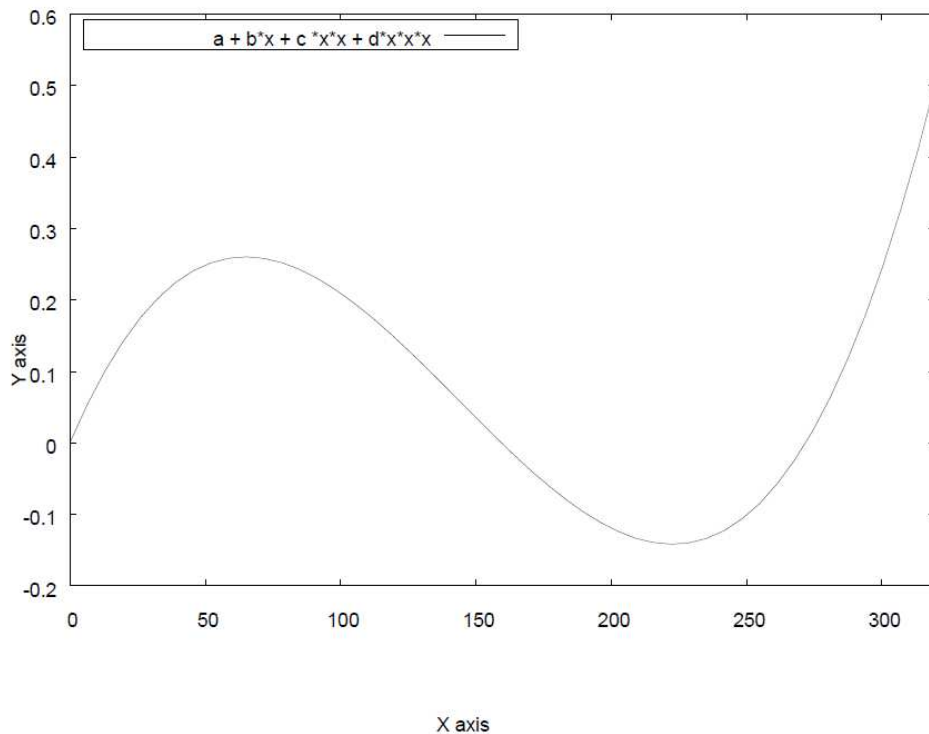
Pro nalezení koeficientů polynomu pro interpolaci chyby vzniklé distorzí čočky byl pořízen snímek 40 cm měřicího pásma (viz obr. 4.6). Polynom interpolující chybu měl překvapivě všechny koeficienty nulové. V tabulce 4.2 jsou naměřené hodnoty

80px	50mm
160px	100mm
220px	150mm

tab. 4.2

Z naměřených hodnot je patrné, že s rostoucími objekty v reálném světě rostou lineárně i jejich zobrazená velikost na čipu. Měření bylo opakováno a podařilo se najít polynom s velmi nízkými koeficienty. Graf 4.5 zobrazuje závislost vzdálenosti od středu snímku (x) a zkreslení čočkou (y) v pixelech.

Koeficienty polynomu – $a=0$, $b=0.0088$, $c=-8.8397 \cdot 10^{-5}$, $d=2.0496 \cdot 10^{-7}$.



graf 4.5

Z grafu je patrné, že při vzdálenosti 300 pixelů od středu obrázku máme zkreslení čočkou pouhých 0.5 pixelu.



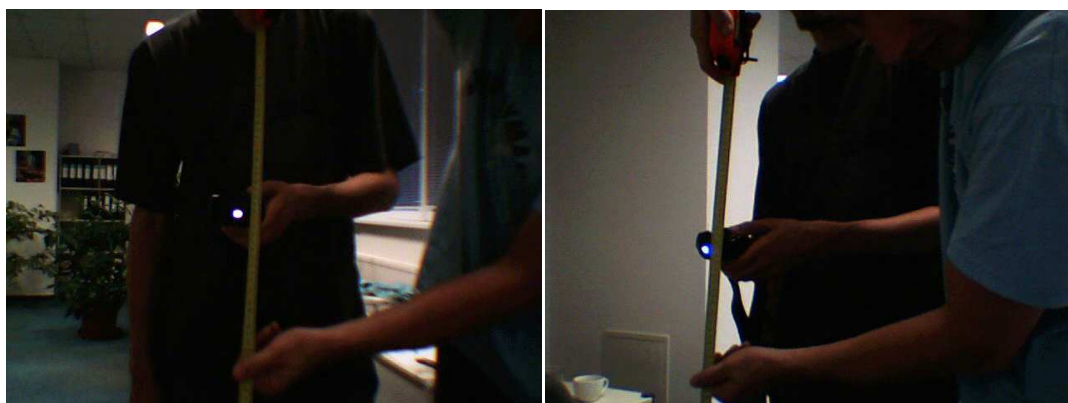
obr. 4.6

4.2.2. Test – odměření bodů

V tomto testu byly odměřeny souřadnice světelného bodu (viz obr. 4.7). Původní souřadnice a zrekonstruované souřadnice jsou v tabulce. Hodnoty jsou v milimetrech.

	x	y	z
původní	55	725	15
zrekonstruované	65.2	741.4	21.3
chyba	10.2	16,4	6,3

tab. 4.3

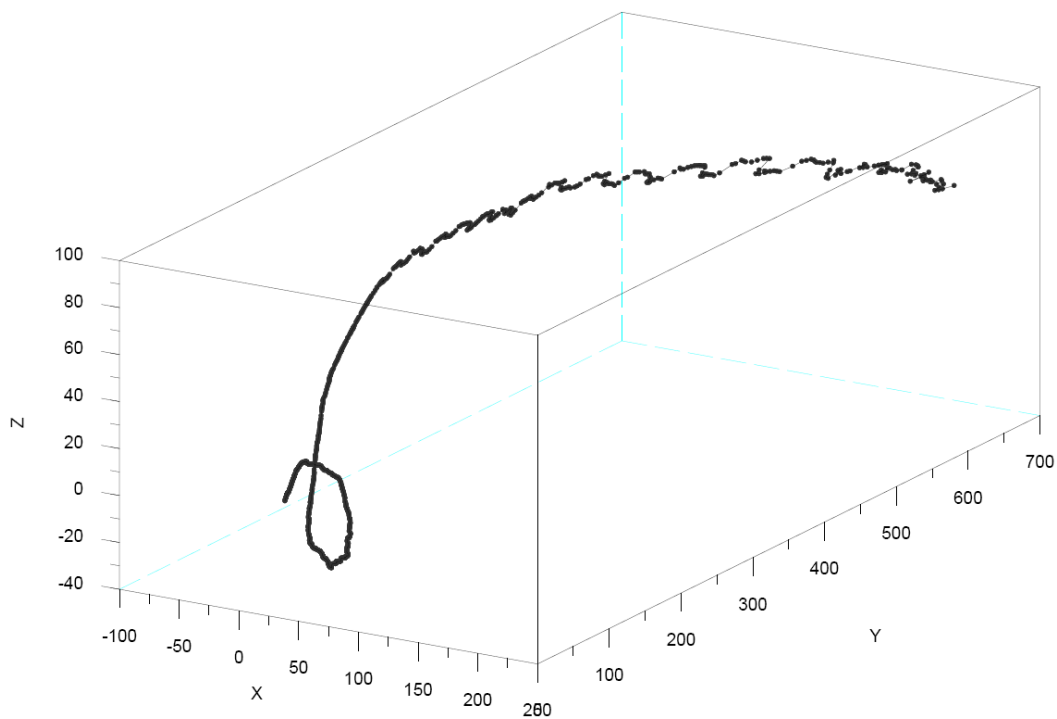


obr. 4.7

Odstranění radiální distorze, nemělo na výsledek opět skoro žádný vliv.

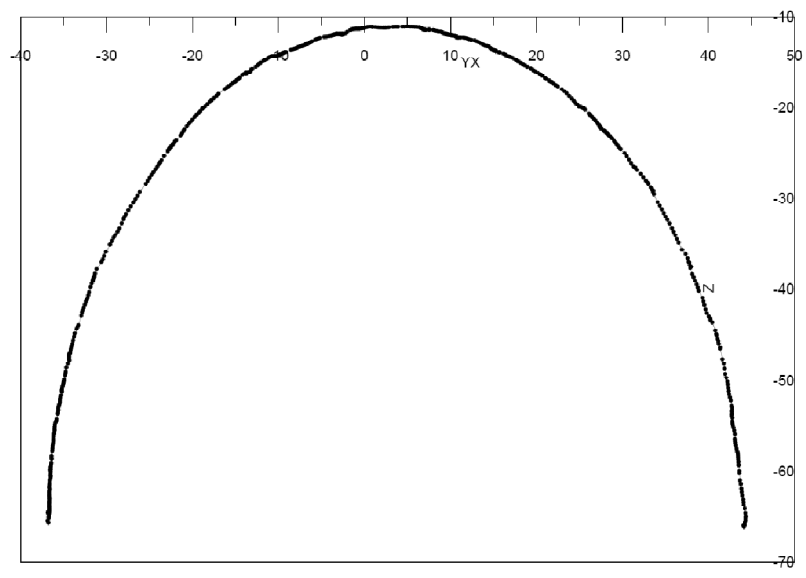
4.3. Reálná data – realtime verze aplikace

Na verzi aplikace, která zobrazuje data v reálném čase, byla testována dynamika systému. V grafu 4.6 je zobrazena rekonstrukce spirálovitého pohybu. Je patrné, že s rostoucí vzdáleností v ose y, jsou výsledky zkreslenější. Obrázky s kamer jsou synchronizovány lineární interpolací. Zrekonstruované body zatím nejsou interpolovány.



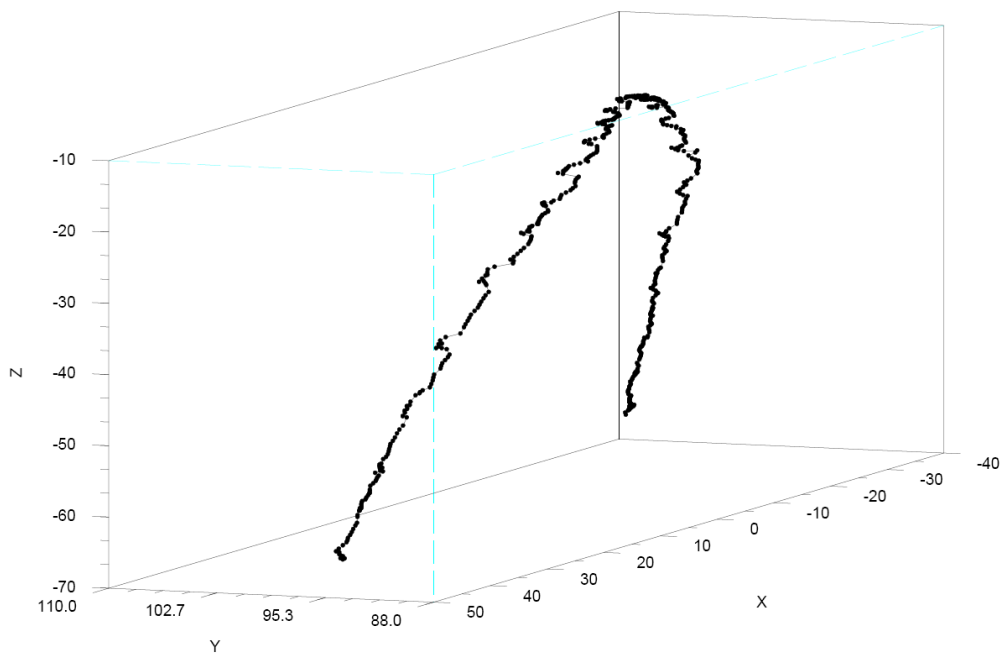
graf 4.6

V dalším pokusu byla dioda přivázána na provázek a pohybem s napnutým provázkem okolo středu uchycení byl vytvořen půlkruh. Z pohledu os x a z se zaznamenaná geometrie jeví celkem příznivě viz graf 4.7.



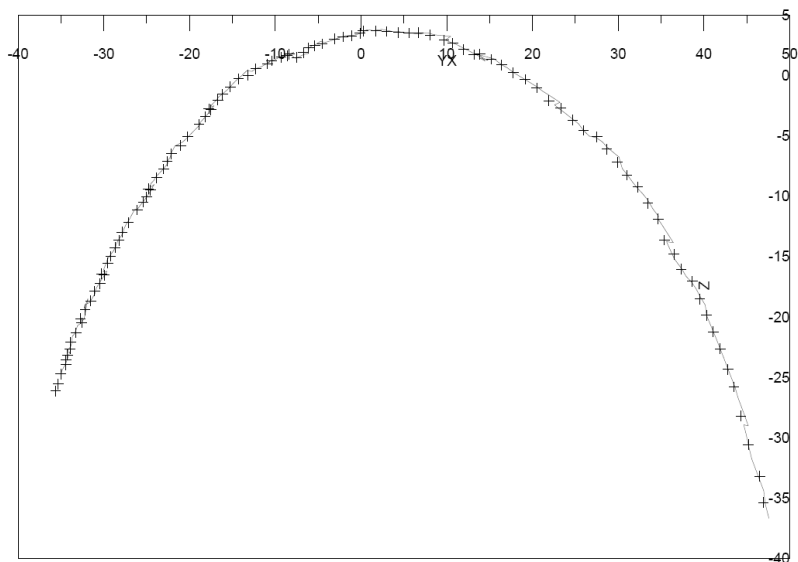
graf 4.7

Po zobrazení stejné množiny bodů z jiného úhlu je ale patrné zkreslení v ose y (viz graf 4.8).



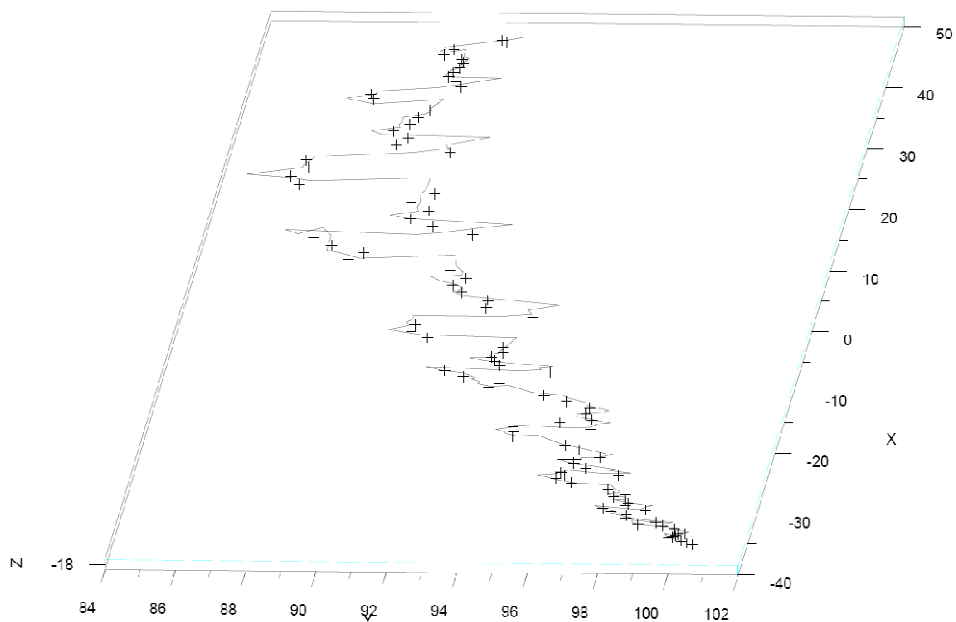
graf 4.8

Obdobný pohyb s interpolací rekonstruovaných bodů lineární interpolací a zprůměrováním třech posledních hodnot je zobrazen v grafu 4.9.



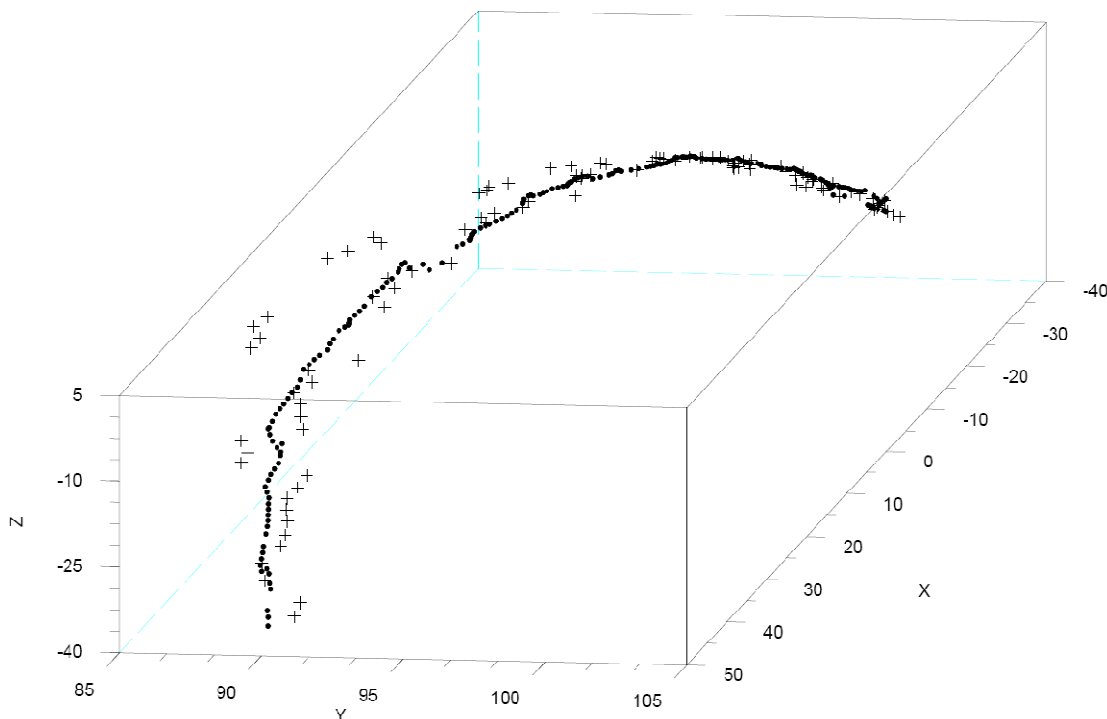
graf 4.9

Opět je patrné zkreslení v ose y (graf 4.10).



graf 4.10

Konečné řešení, které dávalo nejlepší subjektivní výsledky interakcí se systémem, byla lineární interpolace a následné zprůměrování větší množiny bodů pro výpočet y-ové souřadnice. Ideálním počtem okolních hodnot se jevila hodnota 5 pro výpočet y-ové souřadnice 3 pro ostatní osy. Systém má potom větší odezvu v ose y, rekonstruovaný bod je ale méně zkreslený (viz graf 4.11).



graf 4.11

4.1. Odezva systému

Dalším z testů byl test odezvy systému. Před monitor byla postavena DV kamera, která snímala scénu tak, aby byl vidět pozorovatel se světelným bodem v ruce a zároveň výsledek rekonstrukce zobrazený na monitoru. Na kameru byla zaznamenána sekvence prudkého a jasně identifikovatelného pohybu uživatele. Pak v programu Adobe Premiere byla tato sekvence analyzována. Při rozdělení PALového obrazu na pulsímky byla získána frekvence snímání 50 snímků za vteřinu. Manuálním odměřením počtu snímků mezi pohybem uživatele a výsledným zobrazením stejné reakce na monitoru bylo po zprůměrování několika pokusů naměřeno 20 pulsímků pro pohyb v ose y a 15 pulsímků v ostatních osách. Odezva systému je tedy zhruba 0.4 vteřiny v ose y a 0.3 vteřiny v ostatních osách. V této hodnotě jsou započteny všechny fáze od snímání kamer až po výsledný rendering. Měření bylo provedeno na počítači s procesorem Intel Core Duo 2,33GHz.

4.2. Diskuze výsledků

Na výsledcích s umělými daty bylo ověřeno, že výše uvedené algoritmy fungují. Odchylka do 20 milimetrů, kde hledaný objekt se pohyboval v rozmezí 2 metrů ve všech osách, je uspokojivá. V příloze jsou měření s umělými daty renderovanými s použitím rozmazání pohybujících se objektů (motion blur). U těchto dat jsou výsledné odchylky do 70 milimetrů. Obdobný výsledek se dá očekávat i u reálných dat. Maximální velikost příčky mezi paprsky, nepřesahující hodnotu 30 milimetrů, není přesný ukazatel přesnosti. Byla by ale náhoda, když vzdálenosti mezi paprsky se pohybují v takto nízkých hodnotách, aby rekonstruovaná data měly řádově větší chybu při tolika testech. Odměření bodu dopadlo také v celku uspokojivě. Vyšší zkreslení v ose y roste se vzdáleností od kamer. Tato vlastnost ve srovnání s lidským stereoskopickým vnímáním je očekávatelná. Odhad, zda je objekt více či méně vlevo (vpravo), nebo nahoře (dole) je daleko přesnější než odhad jeho vzdálenosti. Pokud se tato vzdálenost zvětšuje a vzdálenost očí je vůči této vzdálenosti zanedbatelná, obrazy z obou očí jsou téměř stejné a přesnost odhadu vzdálenosti špatná (např. při sledování hvězd). Analogicky pokud bude vzdálenost kamer zanedbatelná oproti vzdálenosti objektu, bude zkreslení velké. Proto je vhodné rozmístit kamery s ohledem na vzdálenost, v které se uživatel hodlá pohybovat.

5. Závěr

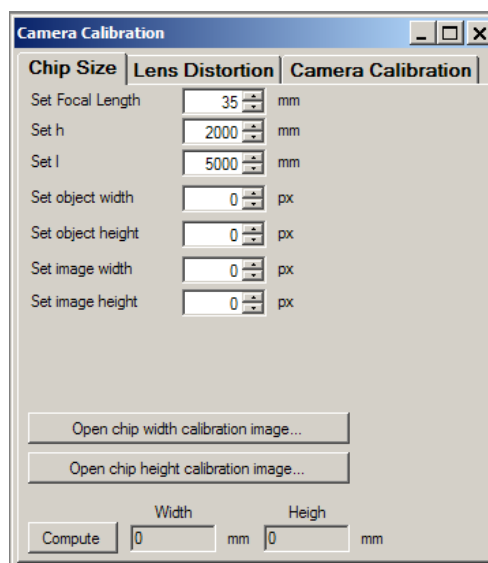
Řešení tohoto problému zasahuje do mnoha odvětví vědy a techniky. Od problémů s optickou soustavou kamery a hardwaru, přes geometrii až po implementaci. Přes všechny problémy se nakonec podařilo vytvořit funkční aplikaci pro demonstraci rekonstrukce. Latence systému je poměrně vysoká ve srovnání s komerčně dostupnými systémy. Cena těchto systémů při koupi hardwaru a softwaru se ale pohybuje v řádech stovek až tisíců dolarů. Hardware pro toto řešení se pohybuje v řádu stovek až tisíců korun (dvě webové kamery). Tento systém by šel určitě dále zlepšovat zevrubnějším prozkoumáním jednotlivých aspektů. Jedním z nich by mohla být automatická kalibrace systému.

6. Přílohy

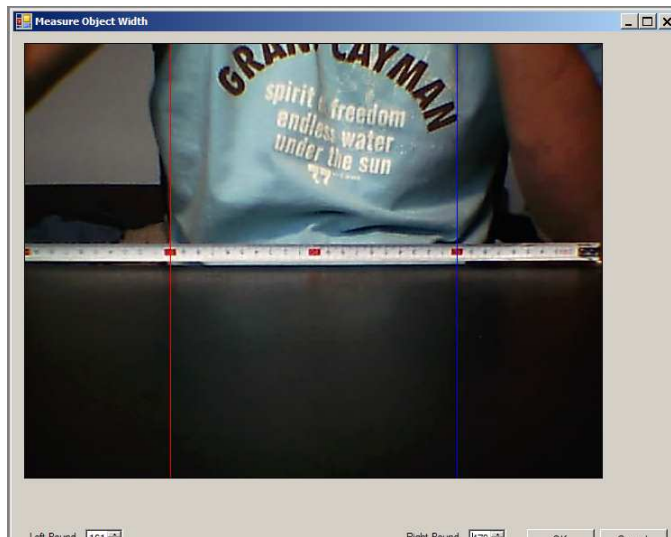
A. Uživatelská dokumentace

Offline verze

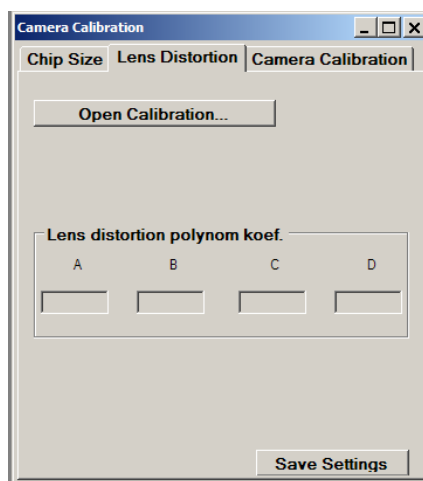
Po spuštění programu jsou v menu k dispozici položky Calibration, Point reconstruction, Threshold test a Text Out. Položka Calibration umožňuje uživateli nastavit kalibraci kamery. Záložka chip size je pro změření velikosti čipu kamery.



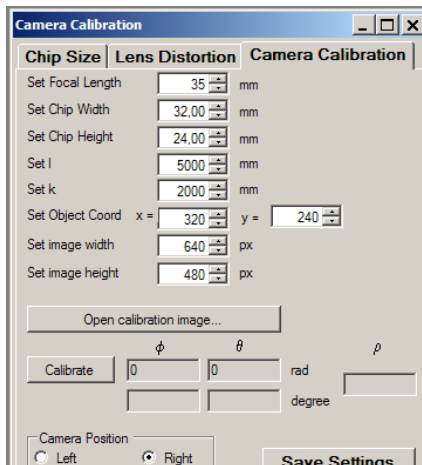
Před spočtením velikosti čipu (button compute) musí být zadána velikost objektu v světových souřadnicích (parametr h) a jeho vzdálenost od kamery (parametr l). Dále musí být zadána ohnisková vzdálenost kamery. Šířka a výška objektu může být zadána ručně, nebo použitím formuláře pro měření velikosti objektu. Buttony „open chip width (height)“ umožní nahrát obrázek, kde lze označit šířku (výšku) objektu v pixelech levou (horní) a pravou (dolní) hranicí.



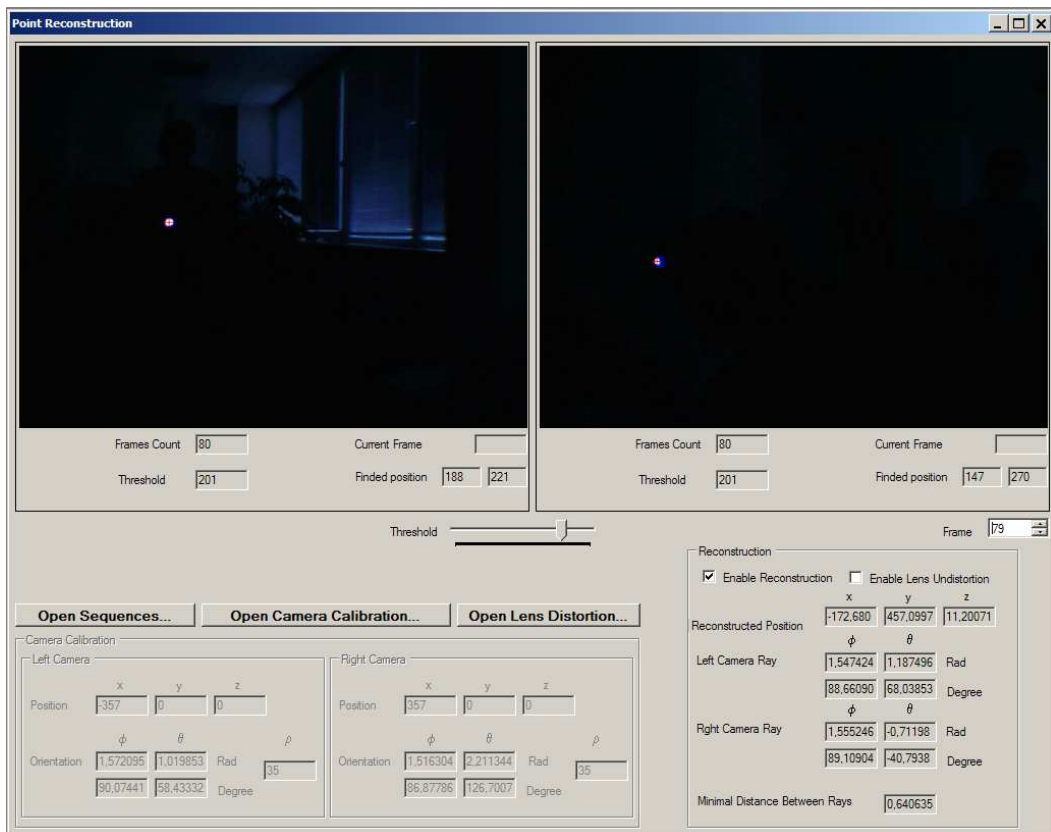
Záložka lens distortion slouží k nalezení polynomu popisujícího distorzi čočky kamery. Stisknutím buttonu “Open Calibration” je otevřen formulář, kde lze provést tři měření šířky objektů. Po uskutečnění měření jsou zobrazeny koeficienty polynomu. Ten lze buttonem “Save Settings” uložit do souboru formátu XML.



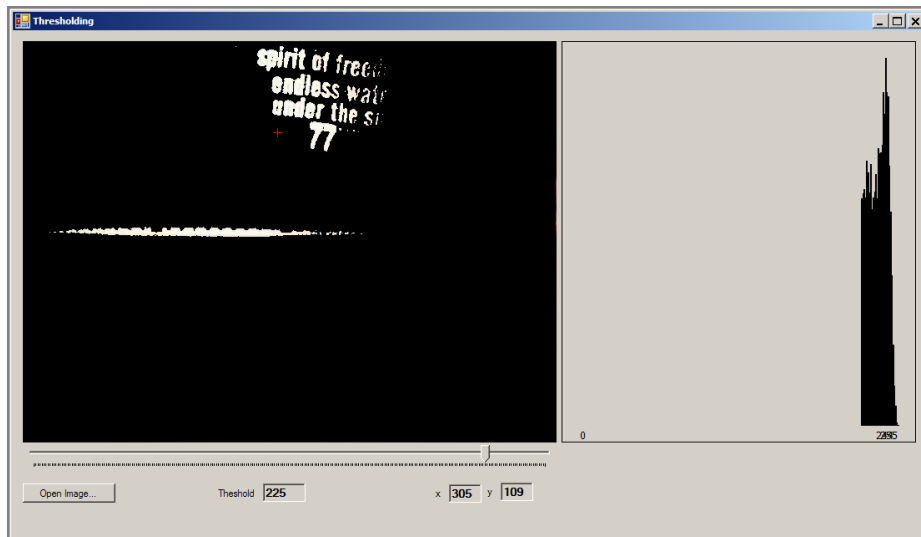
Poslední záložka “Camera Calibration” slouží ke kalibraci orientace kamery. Musí být zadána vzdálenost kamery od počátku (parametr k) a vzdálenost kalibračního objektu (parametr l). Souřadnice kalibračního objektu mohou být zadány ručně, nebo pomocí formuláře po stisknutí buttonu “Open calibration image”. Před spočtením úhlů kamery je důležité zvolit, jestli je kamera pravá, nebo levá (levou kameru vidí pozorovatel vpravo!). Buttonem „Save settings“ je opět kalibrace uložena do XML souboru.



Položka „Point Reconstrucion“ spouští formulář pro kalibraci bodu v prostoru. Je nutné otevřít sekvence obrázků pro pravou a levou kameru (button Open Sequences), nahrát kalibrační informace o levé a pravé kameře (button Open Camera Calibration), popřípadě soubor pro odstranění distorze. Po zaškrtnutí checkboxu „Enable Reconstruction“ a iterováním přes jednotlivé snímky (Frame) je prováděna rekonstrukce. Lze nastavit parametr Threshold, který určuje práh při hledání objektu v obrázcích.



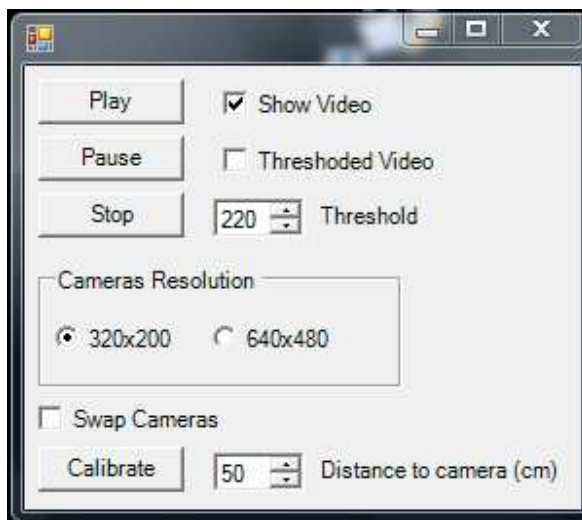
Položka Threshold Test umožňuje uživateli vyzkoušet, jak bude nastavení prahu reagovat na vyhledávání objektu.



Poslední položka Text out, je pro výpisy naměřených dat.

Realtime verze

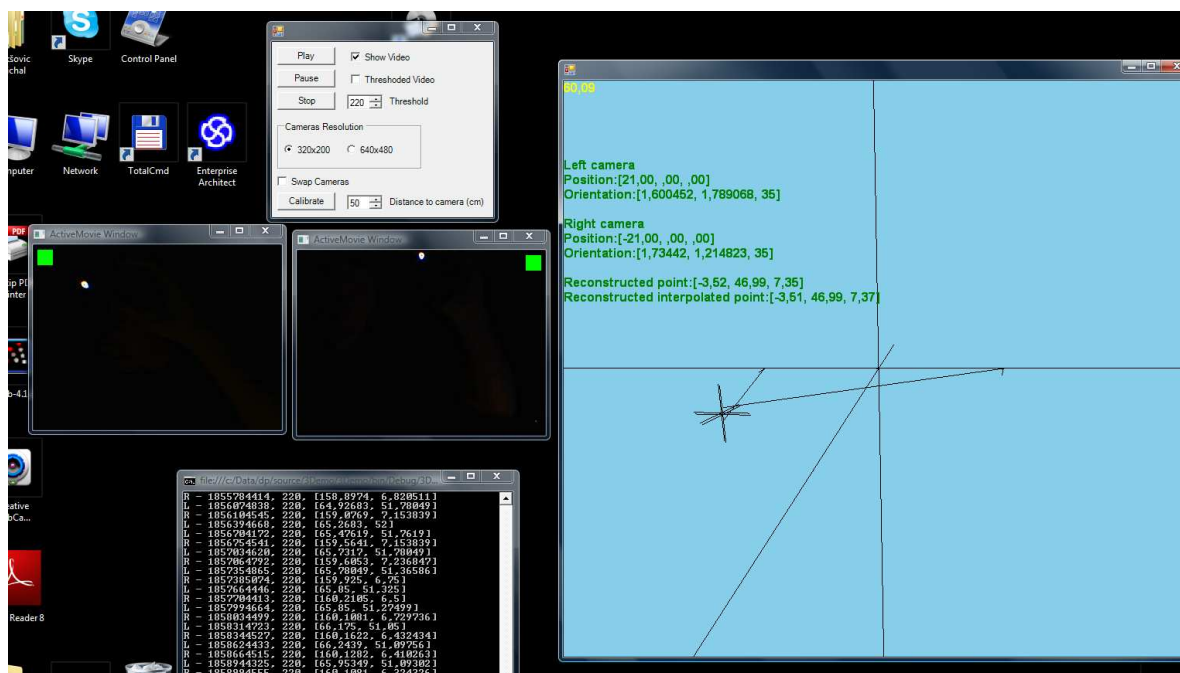
Po spuštění aplikace se zobrazí dialog (viz níže). Po stisknutí tlačítka play jsou zobrazeny obrazy kamer. Levá (pravá) kamera je identifikována zeleným čtvercem v levém (pravém) horním rohu. Uživatel může tyto kamery prohodit zaškrtnutím checkboxu Swap Cameras. Dále je možné nastavit rozlišení kamer, threshold pro získání světlého bodu z obrazů a vzdálenost kalibračního bodu od počátku. Další informace o kamerách jsou uloženy v adresáři aplikace v XML souborech.



Nastavení vzdálenosti kamer je možné provést v offline verzi aplikace, nebo jednoduše přímou editací XML souboru. Například při vzdálenosti kamer 42 cm je nutné nastavit v souboru LeftCamera.Xml pozici na 21 a RightCamera.Xml -21 (souřadný systém je zobrazen v kapitole 2.1.2)

```
<?xml version="1.0" encoding="utf-8"?>
<camera>
  <position>
    <vector3D x="21" y="0" z="0" />
  </position>
  <orientation>
    <vectorSpherical fi="1.7955074952095393" theta="2.245537269018449" rho="35" />
  </orientation>
  <chipSize width="32" height="24" />
</camera>
```

Pak umístěním diody na nastavenou vzdálenost od počátku a stisknutím tlačítka Calibrate je provedena kalibrace. Vypočtené orientace kamer jsou uloženy do Xml souboru, takže při znovuspouštění aplikace není nutné kalibraci opakovat. V okně s 3D vizualizací je možné levým tlačítkem myši a posunem měnit úhly pohledu. Kolečkem myši se vzdalovat a přibližovat ve scéně. Stisknutím tlačítka D se kamera posune na místo rekonstruovaného bodu a v počátku souřadnic je zobrazen 3D objekt.



V okně konzole jsou vypisovány souřadnice nalezené diody v obrázcích a časy pořízení snímků. Písmenem L a R je identifikována levá a pravá kamera.

B. Překlad

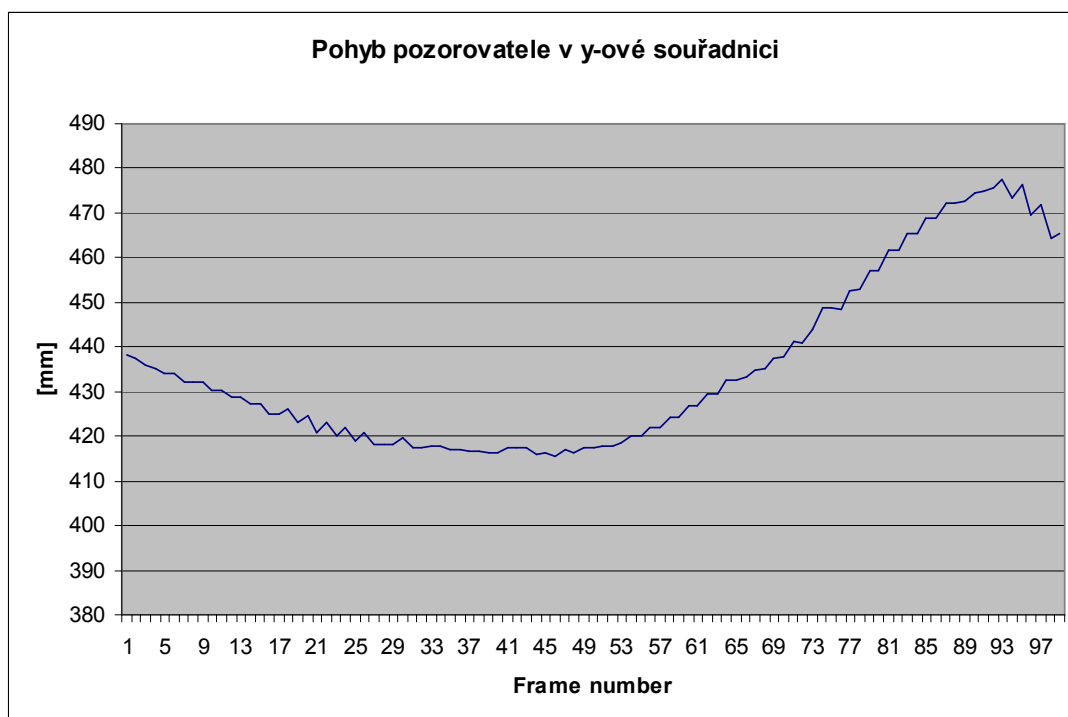
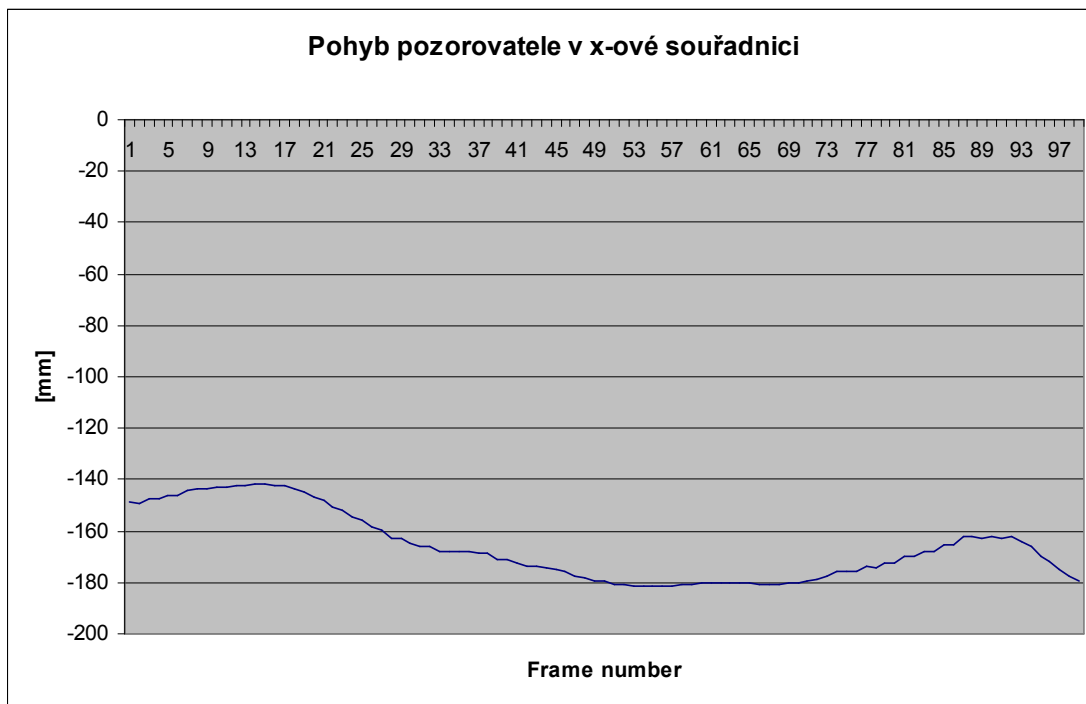
Pro překlad zdrojových souborů je nutné provést následující kroky

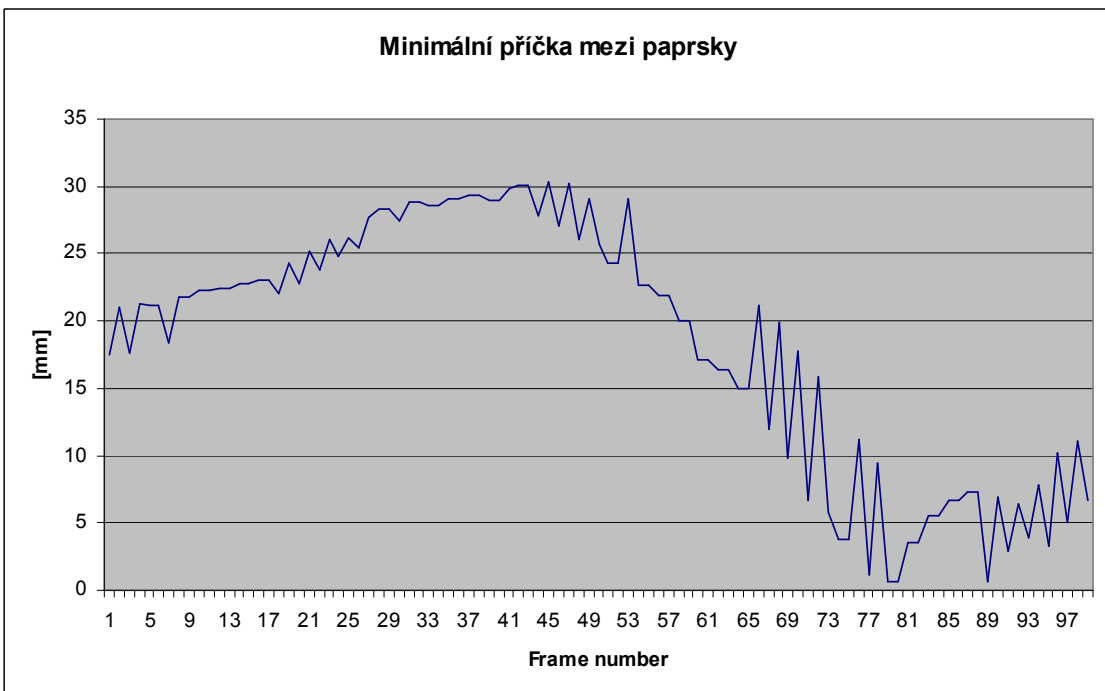
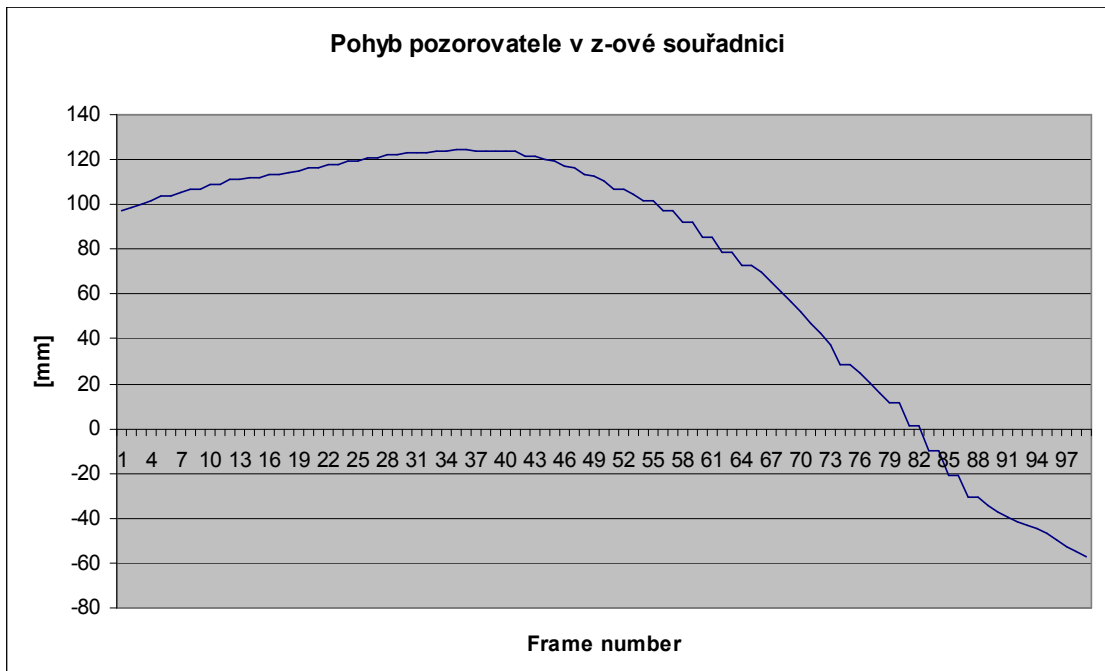
- Je potřeba stáhnout a nainstalovat DirectShow – dříve byla tato knihovna součástí knihovny directX, se nachází v balíku knihoven platform sdk. Poslední verze: Windows® Server 2003 SP1 Platform SDK
- Ve Visual Studiu je potřeba přidat cesty k hlavičkovým souborům. Tato funkce se nachází v menu tools/options/Project and Solutions VC++ Directories. Je nutné přidat tyto cesty¹:
 - záložka include Files – přidat DshowInstalDir\include
 - záložka library Files – přidat DshowInstalDir\lib
- Zkontrolovat zda jsou nastaveny cesty k library a include souborům pro knihovnu directX
- Přeložení base classes pro vytváření vlastních filtrů – z příkazové řádky nastavit proměnné prostředí – spustit vcvars32.bat v adresáři vc++/bin přepnout do adresáře „Microsoft Platform SDK for Windows Server 2003 R2\Samples\Multimedia\DirectShow\BaseClasses\“ – zadat příkaz nmake pro přeložení release verze base classes nebo nmake NODEBUG=1 pro debug verzi
- nastavení cest k include a lib souboru z baseclasses do Visual Studia viz výše

¹ Místo nastavení v záložce include Files lze DshowInstalDir\include nastavit v Project properties/configuration/properties/c++ /general/add additional include directories – v tomto případě nastavení platí pouze pro aktuální projekt a je pro ostatní je nutné opakovat.

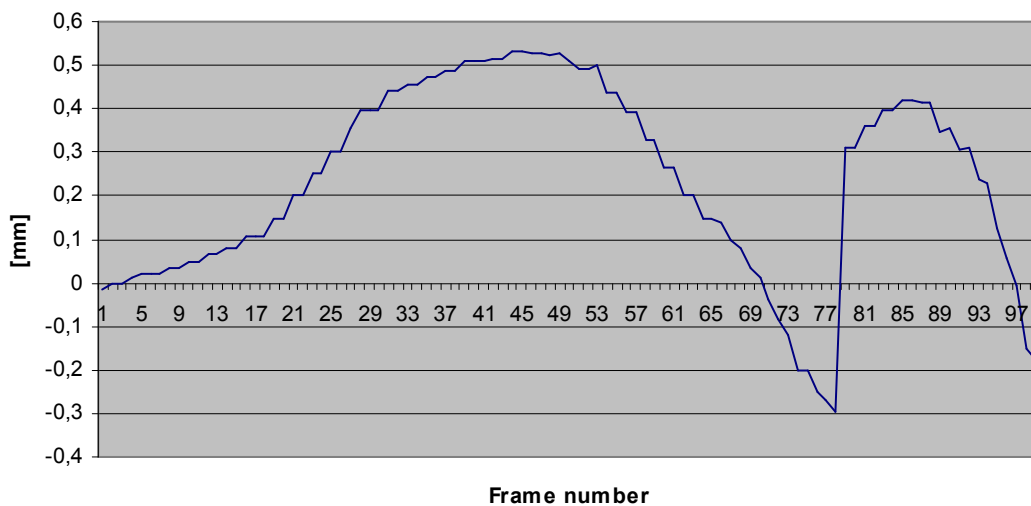
C. Další měření

Test – pomalý pohyb

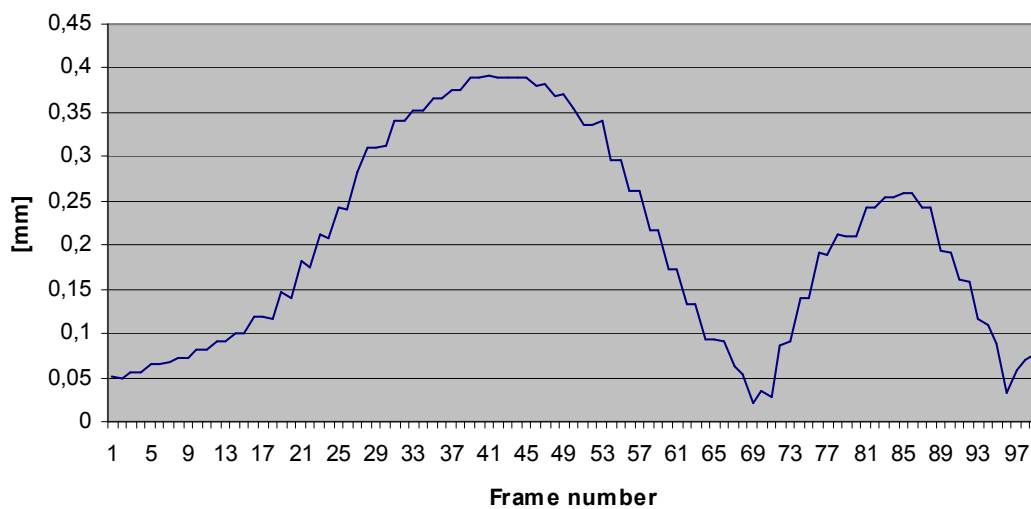




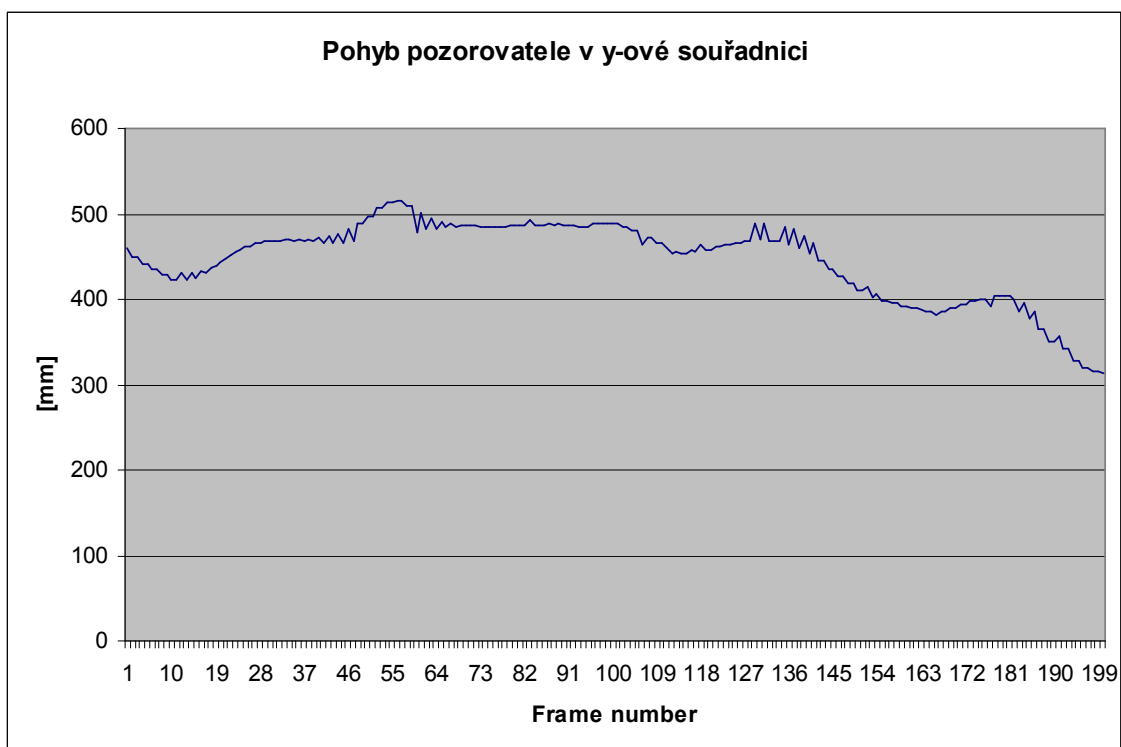
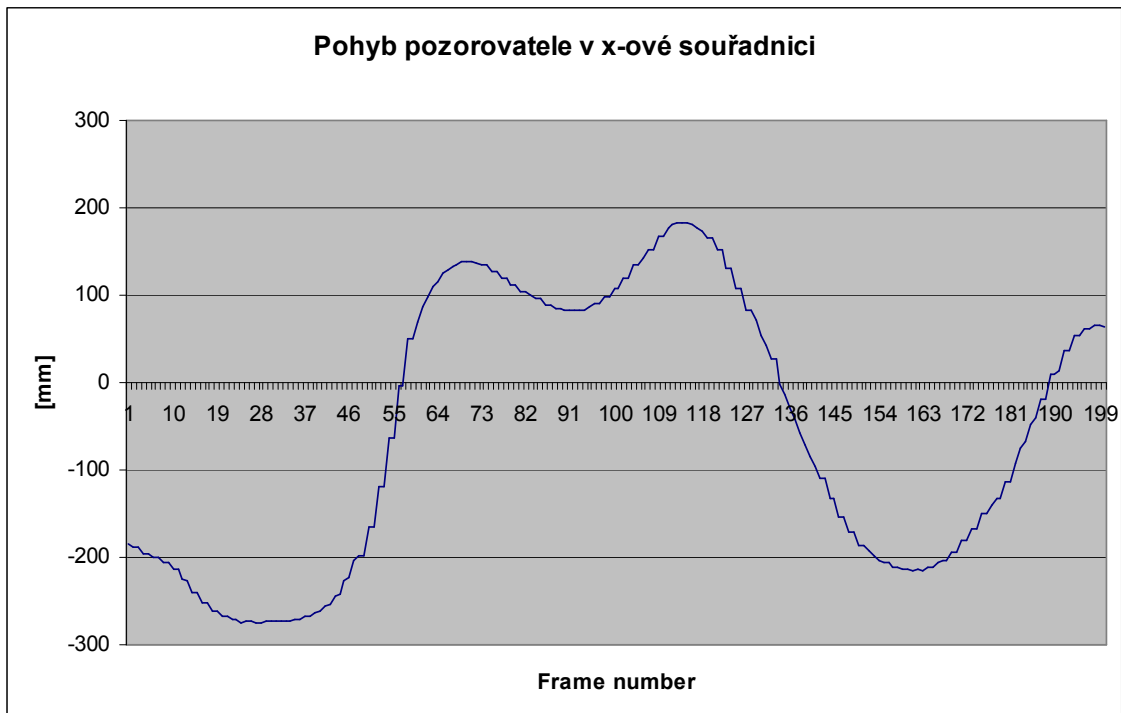
Rozdíl minimálních příček s/bez odstranění distorze čoček

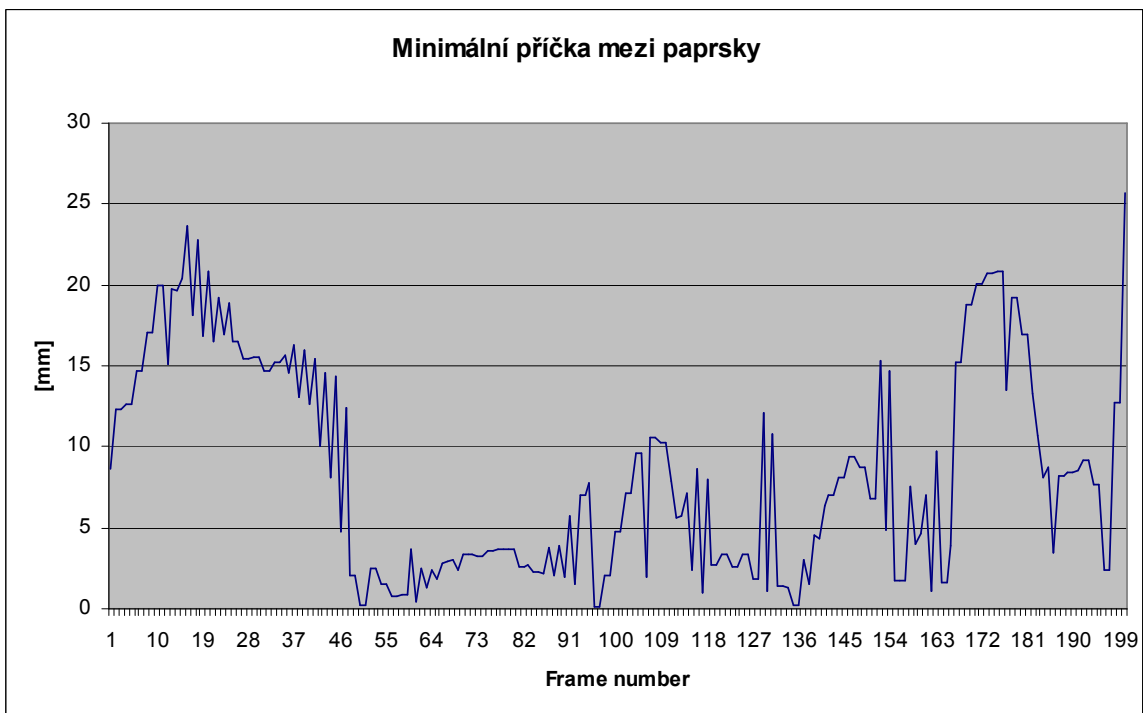
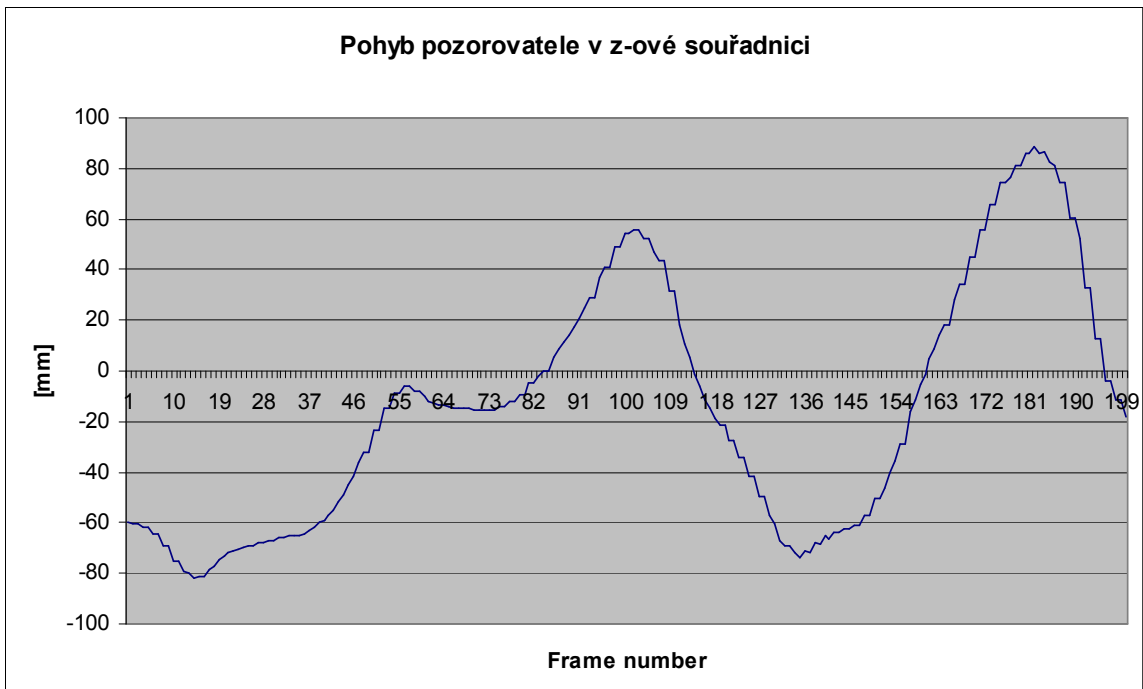


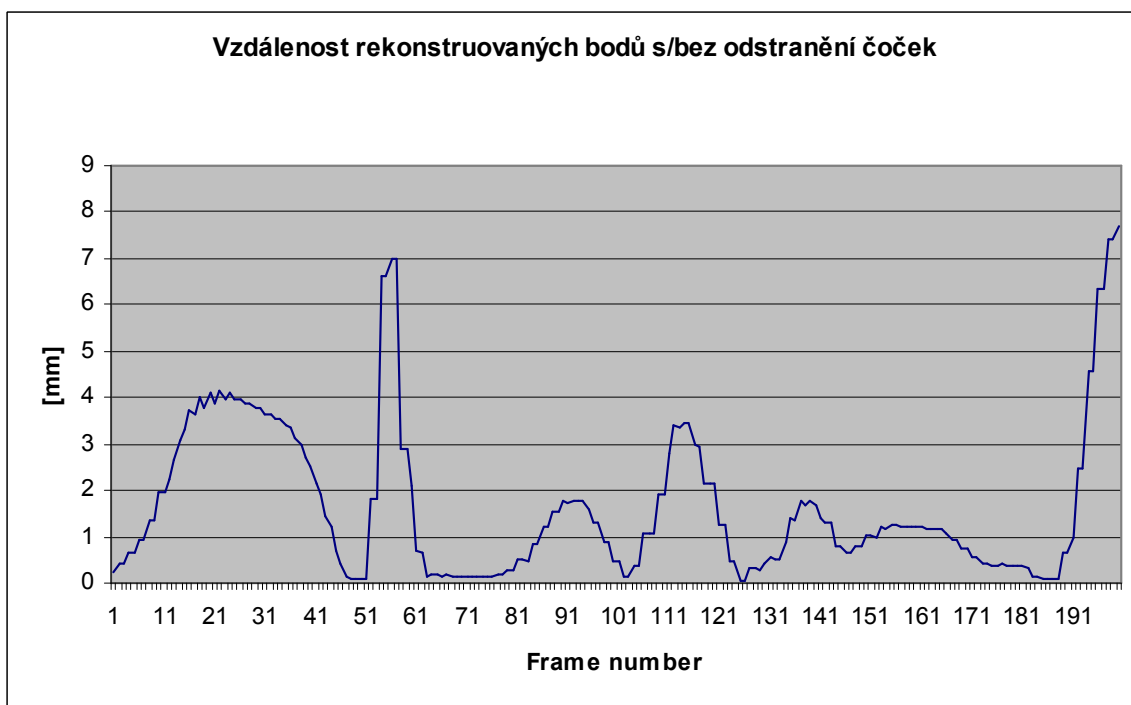
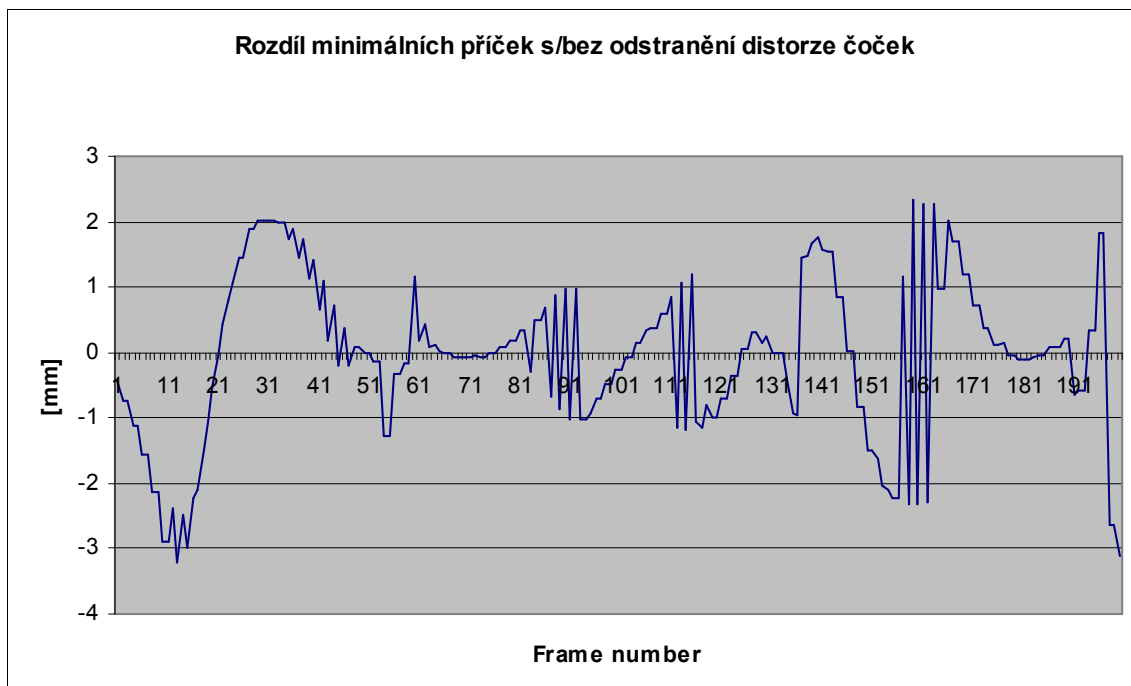
Vzdálenost rekonstruovaných bodů s/bez odstranění čoček



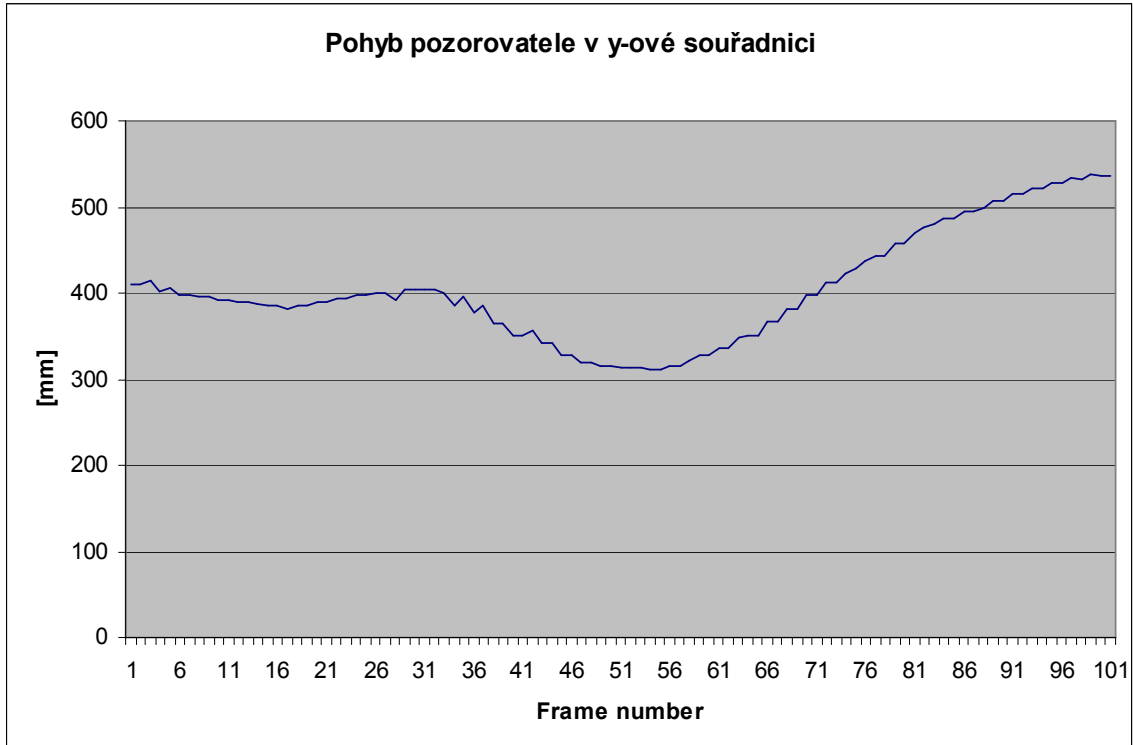
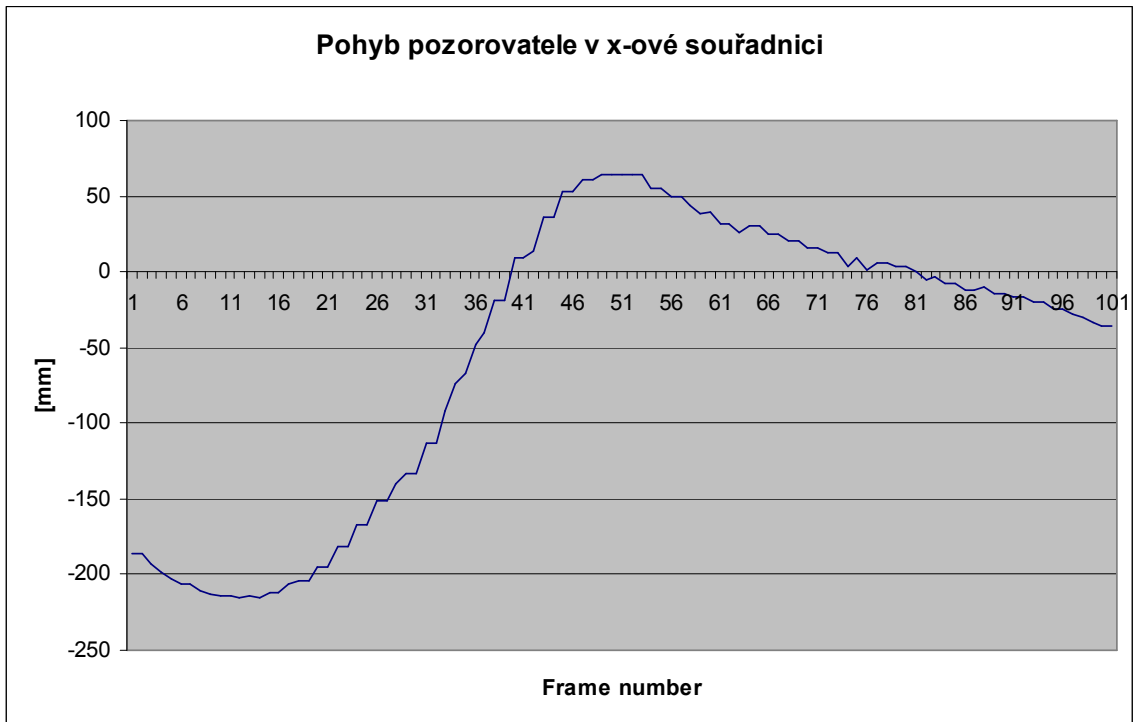
Test – rychlý pohyb v ose x

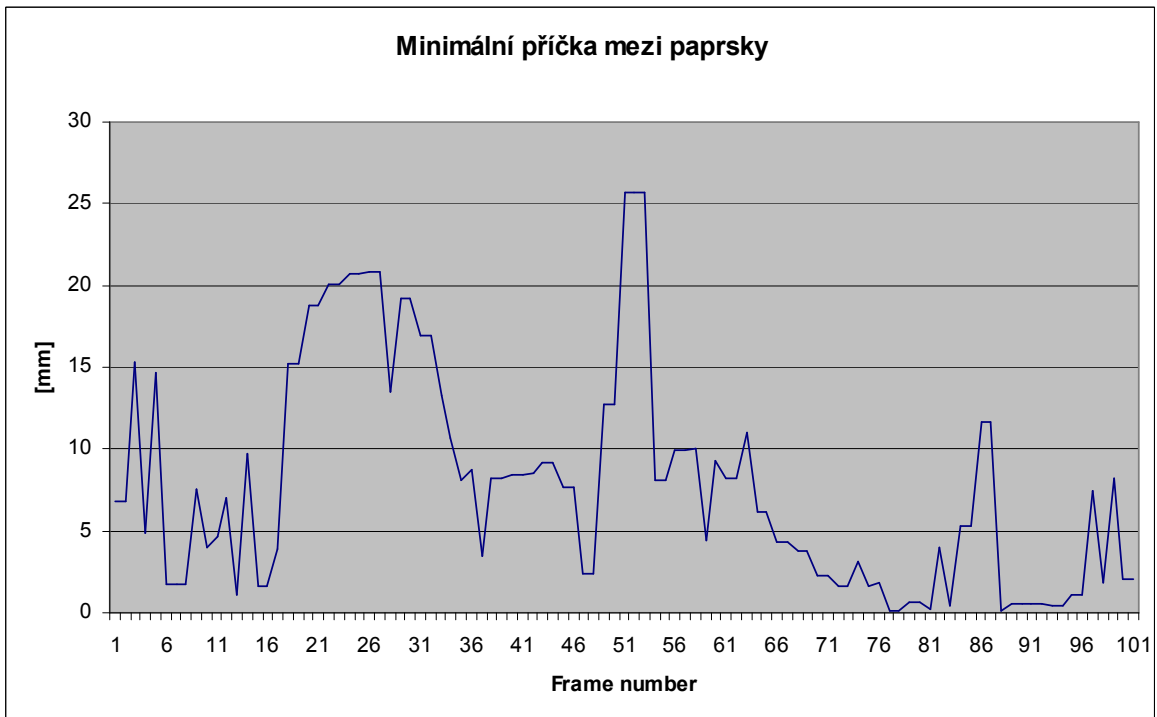
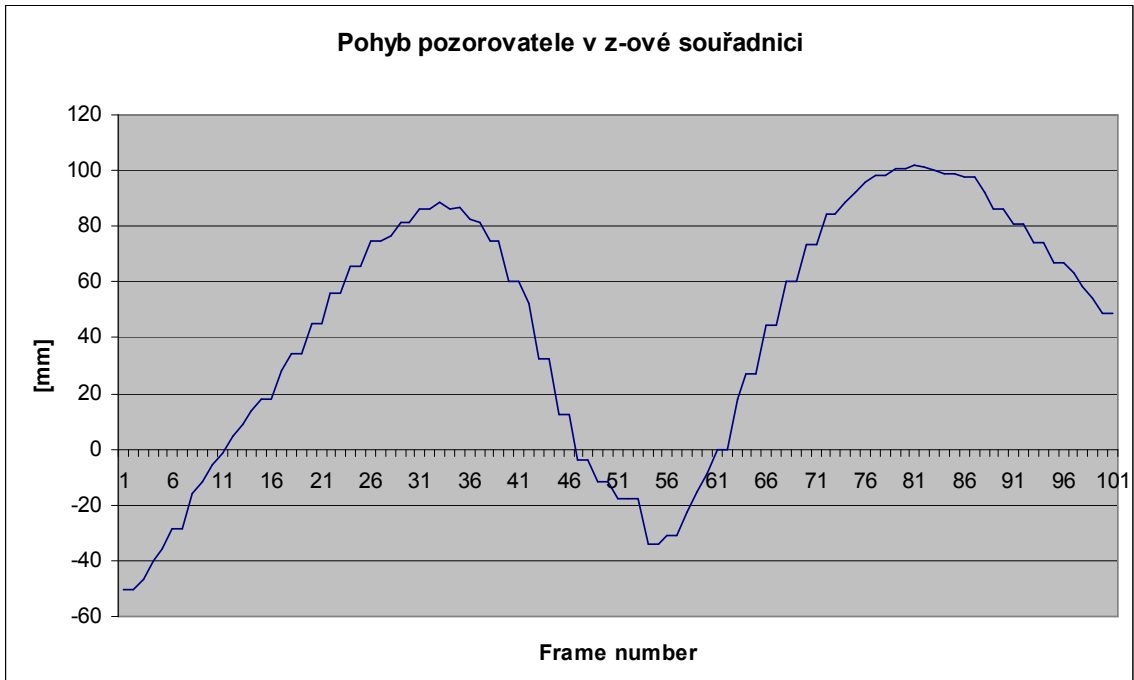




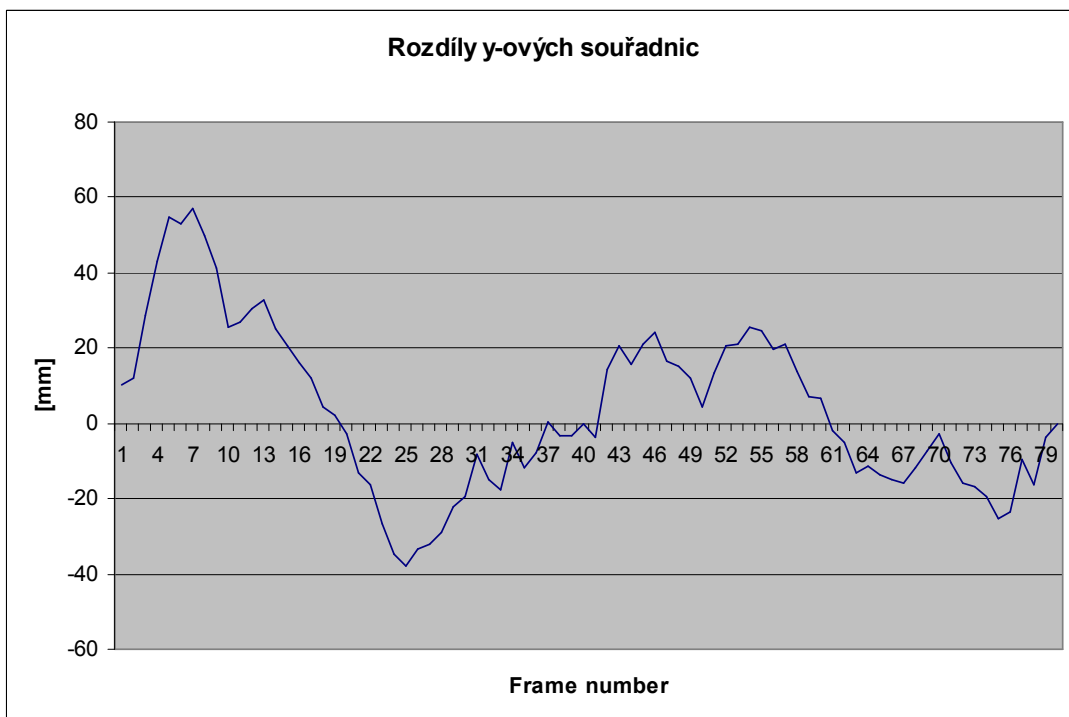
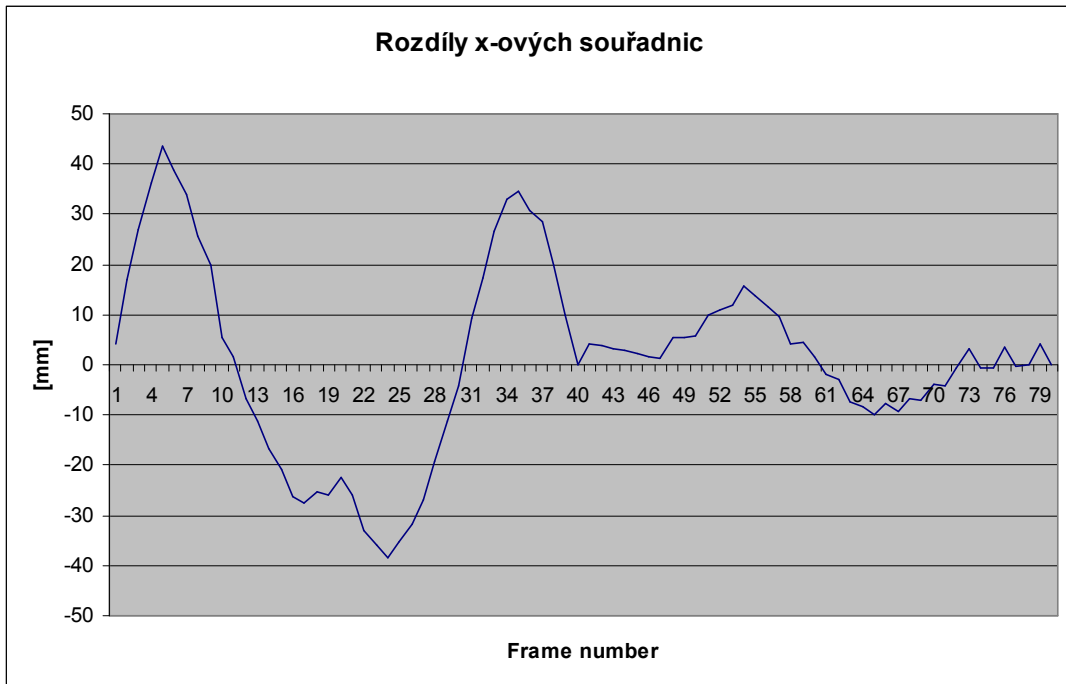


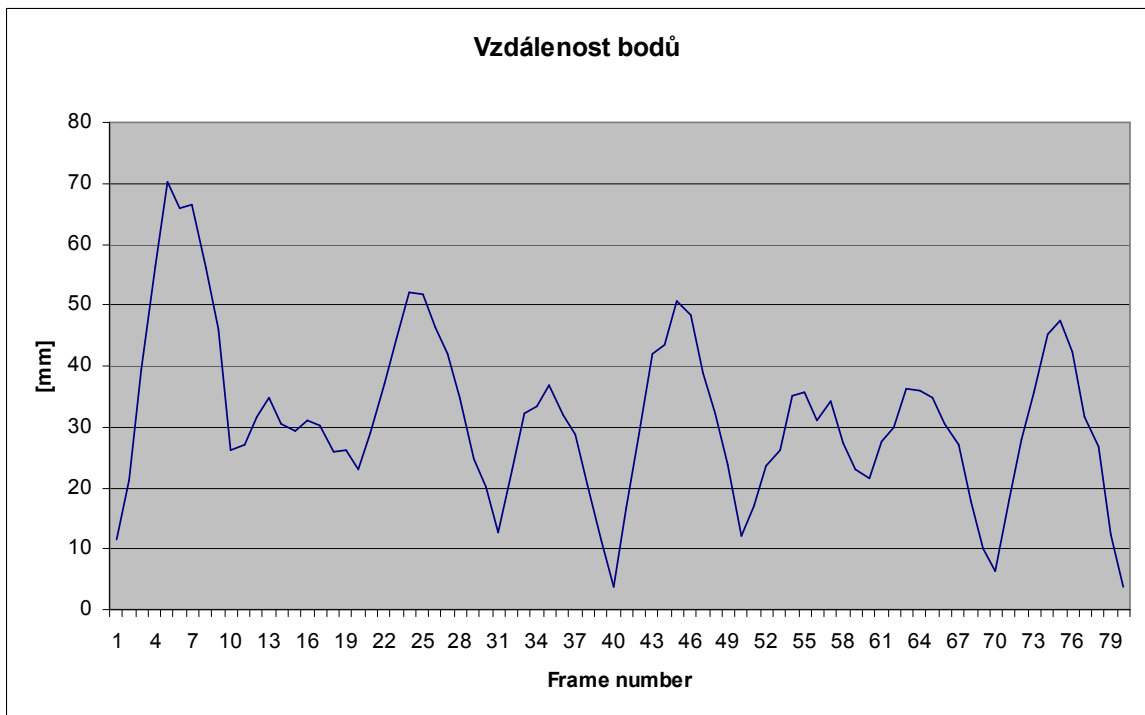
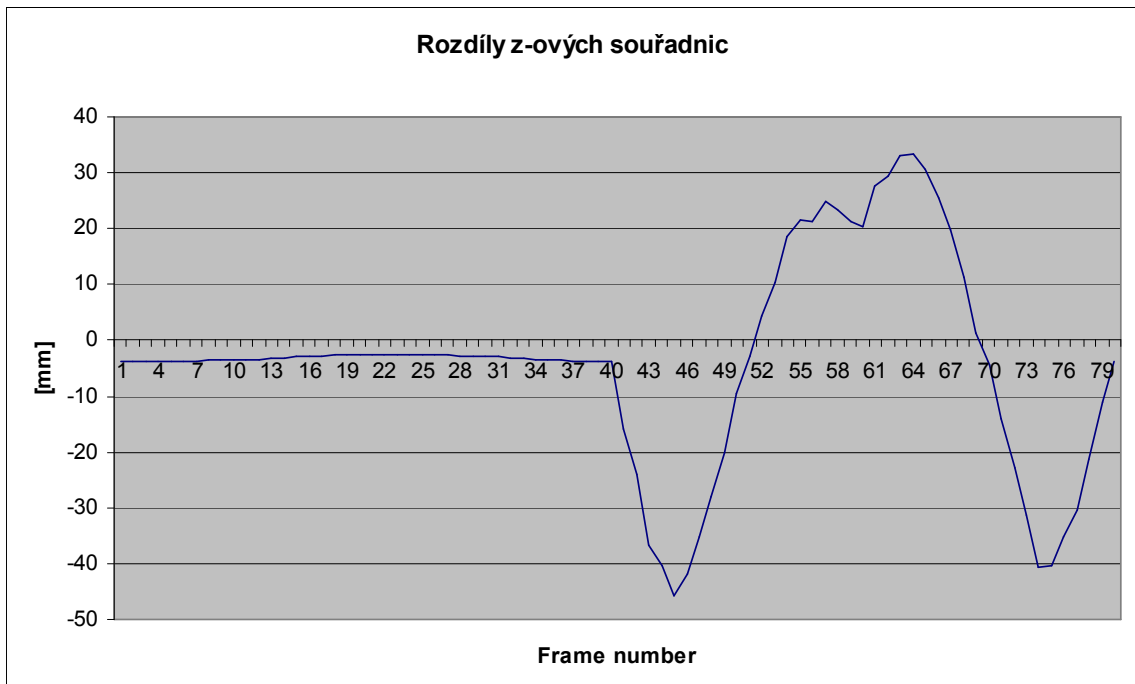
Reálná data – rychlý pohyb v ose z





Umělá data – Blender motion blur





D. Popis Namespaců

Namespace, neboli jmenný prostor, je množina názvů, kde každý název je unikátní. Jde především o názvy tříd. Ostatní identifikátory jsou unikátní v rámci jedné třídy¹. Namespace může mít více úrovní dělení (stromovitá struktura). V knihovně .Net jsou všechny knihovní třídy rozděleny podle logických souvislostí a funkcionalit. Pokusíme se o podobné rozdělení.

Veškeré implementované třídy jsou v namespace `PointTracking`, dále jsou děleny následovně.

- **PointTracking.CameraCalibration**

Tento namespace obsahuje třídy pro veškeré činnosti týkající se kalibrace kamer. Jsou zde všechny formuláře s nastavením pro kalibraci, metody pro výpočet kalibrace a znovupoužitelné vizuální komponenty (pro přehled tříd viz obr. obr. 6.1).

- **PointTracking.CameraCalibration.Data**

Zde jsou obsaženy třídy s datovou podporou pro předchozí namespace. Jsou zde struktury pro popis kamer, vektorů, polynomů atd...



obr. 6.1

¹ Kromě tzv. delegátů.

- **PointTracking.PointPositionReconstruction**

V tomto namespace jsou třídy pro rekonstrukci bodu v prostoru, formuláře s nastaveními týkajícími se rekonstrukce a znovupoužitelné vizuální komponenty. Jsou zde také třídy pro vizualizaci sekvencí z kamer (pro přehled tříd viz obr. 6.2).

- **PointTracking.PointPositionReconstruction.Data**

Zde je opět datová podpora pro rekonstrukci bodu v prostoru. Jsou to datové třídy pro popis snímků sekvence a celé sekvence. Při rekonstrukci je nutné pracovat i s vektory a dalšími daty definovanými v namespace `PointTracking.CameraCalibration.Data`, proto tento namespace je zde také použit. Bez toho není tento namespace schopen samostatné funkcionality.

- **PointTracking.Accessories**

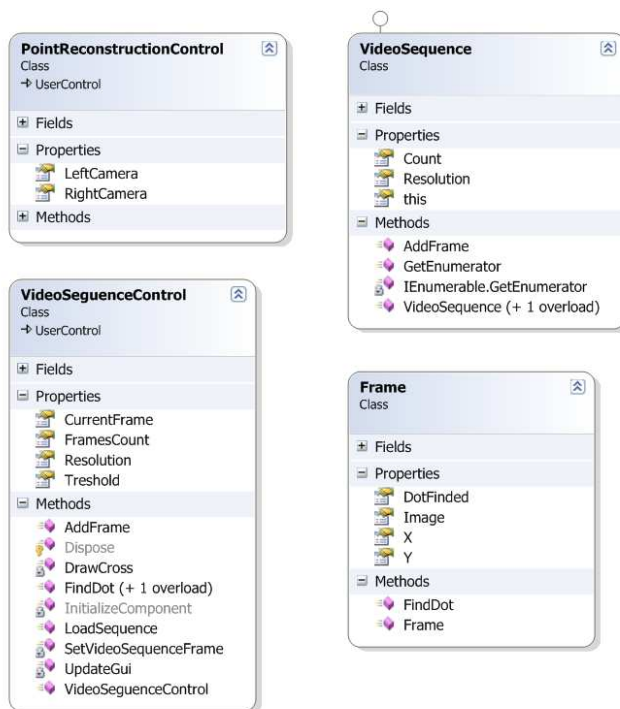
Zde jsou třídy poskytující některé další funkcionality aplikace jako jsou například výpisy výsledků nebo výpočet histogramu.

- **PointTracking.Accessories.Data**

Data pro třídu `PointTracking.Accessories`.

- **PointTracking.MainProject.MDIForms**

Tyto namespace obsahují třídy pro vytvořenou demo aplikaci. Používá všechny uvedené namespace. Zaobaluje vizuální komponenty do formulářů.



obr. 6.2

E. Popis nejdůležitějších tříd

V této kapitole budou uvedeny některé důležité třídy, jejich metody a atributy.

- **PointTracking.CameraCalibration.Data.Vector3D**

Tato třída symbolizuje vektor v kartézských souřadnicích. Kromě souřadnic obsahuje vlastnost `VectorSpherical`, která vrátí instanci vektoru ve sférických souřadnicích. Metoda `Vector3D Mean(Vector3D vector)` vrací pozici v polovině vzdálenosti mezi dvěma vektory, metoda `double Distance(Vector3D vector)` vzdálenost mezi dvěma vektory. Vektor je možné uložit metodou `void SaveToXML(XmlWriter xmlWriter)`, nebo nahrát statickou metodou `Vector3D LoadFromXML(XmlNode xmlNode)`. Třidu je možné inicializovat jednotlivými souřadnicemi, nebo vektorem v kartézských či sférických souřadnicích.

- **PointTracking.CameraCalibration.Data.VectorSpherical**

Vektor ve sférických souřadnicích. Třída umí vracet jednotlivé úhly v radiánech, nebo ve stupních. Obdobně jako třída `Vector3D` může být uložena, nebo nahrána z XML formátu. Vlastnost `Vector3D` vrací instanci v reprezentaci kartézských souřadnic. Třída může být inicializována kartézským vektorem, nebo jednotlivými úhly ve sférické soustavě souřadnic.

- **PointTracking.CameraCalibration.Data.Vector3DwithPosition**

Třída obsahuje dvojici vektorů pro určení polohy a orientace parametrické přímky. Metoda `Vector3D ComputeParametricPoint(double t)` vrací polohu na přímce s parametrem `t`. Metoda `double ComputeMinimalDistanceToRay(Vector3DwithPosition ray)` vrací minimální vzdálenost dvou přímek, metoda `Vector3D ComputeCenterPointToRay(Vector3DwithPosition ray)` vrací bod polovině minimální příčky mezi dvěma přímkami (viz kap. 2.3.2),

- **PointTracking.CameraCalibration.Data.Polynomial**

Struktura reprezentuje polynom libovolného stupně. Je inicializována polem koeficientů. Opět je zde možnost strukturu uložit (nahrát) do (z) formátu XML. Metoda `double FunctionValue(double r)` vrací funkční hodnotu polynomu s argumentem `r`.

- **PointTracking.CameraCalibration.Data.LensUndistortFunction**

Tato struktura obsahuje polynom pro odstranění radiální distorze čoček. Metoda `double UndistortR(double r)` vrací hodnotu chyby (radiální distorze) ve vzdálenosti `r` od středu obrazu (podrobně viz kap. 2.2.3). Metoda `PointF UndistortPixel(int imageWidth, int imageHeight, PointF distortedPoint)` vrací korigované souřadnice pixelu (podrobně viz kap. 2.2.4).

- **PointTracking.CameraCalibration.Data.Camera**

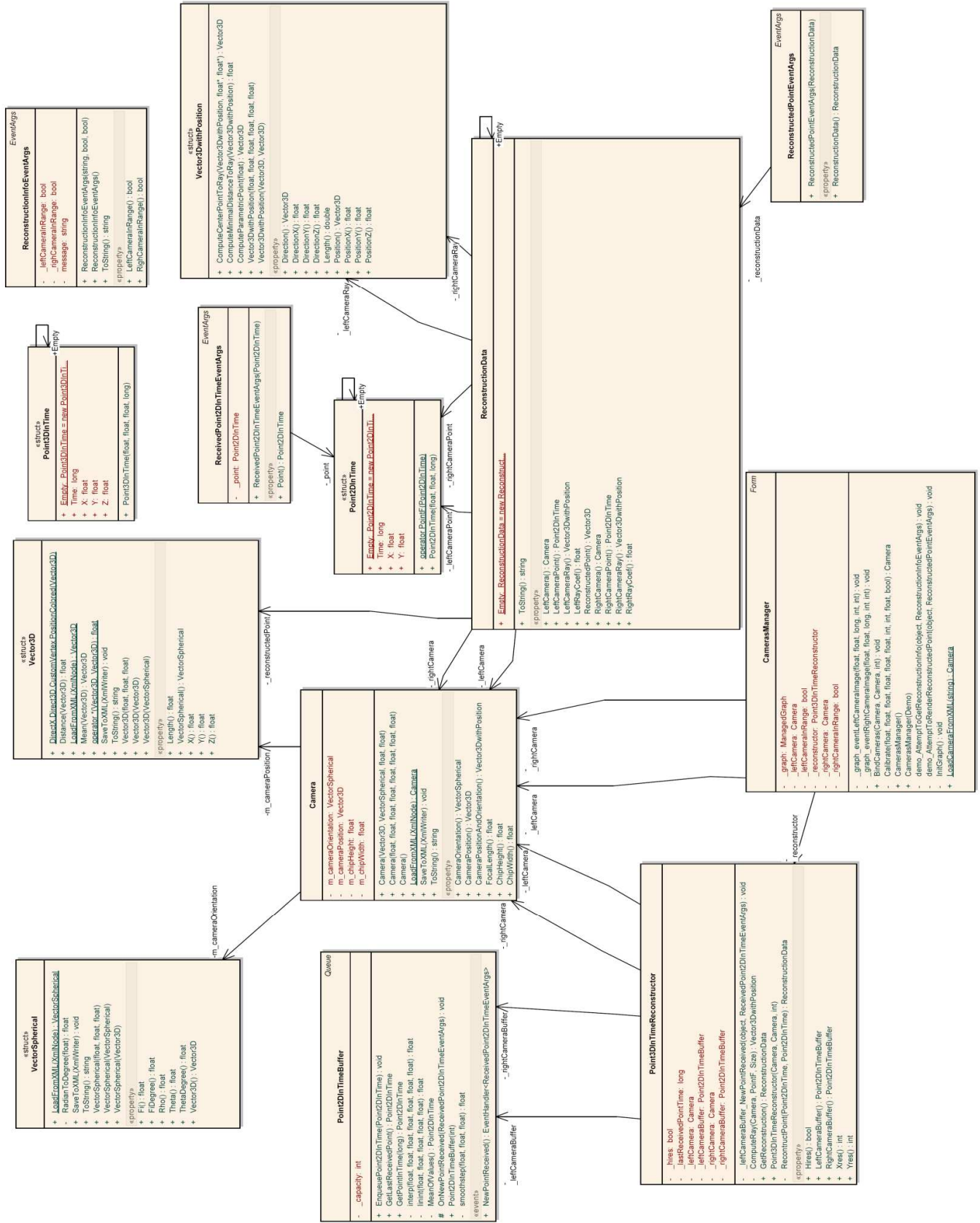
Tato třída symbolizuje kameru. Obsahuje pozici a orientaci kamery, ohniskovou vzdálenost, šířku a výšku čipu. Dále obsahuje metody pro uložení do (z) XML formátu.

- **PointTracking.CameraCalibration.CameraCalibrationControl**

Tato vizuální komponenta umožňuje uživateli nastavit vstupní data a výpočet velikosti čipu, radiální distorze čočky a kalibraci kamery. Třída využívá výše zmíněné datové struktury. Vypočtené hodnoty umí uložit do XML formátu (viz obr. 6.3).

```
<?xml version="1.0" encoding="utf-8" ?>
- <camera>
- <position>
  <vector3D x="-357" y="0" z="0" />
</position>
- <orientation>
  <vectorSpherical fi="1.572095027363458" theta="1.0198539459906186" rho="35" />
</orientation>
  <chipSize width="28.7179487179487" height="21.8181818181818" />
</camera>
```

obr. 6.3



F. Seznam trackerů

Seznam trackerů a jejich parametrů ze studie Institute for Defence Analyses 1996 [1].

Technology	Product	Vendor	DOF	Frequency	Latency	Resolution	Working Volume	Price
Mechanical	BOOM3C	FakeSpace, Inc.	6	>70 Hz	200 msec	0.1°	6 ft diameter, 2.5 ft high	\$95,000 ^a
	PUSH	FakeSpace, Inc.	6	>70 Hz	200 msec	0.1°	2 ft diameter	\$45,000 ^a
	ADL-1	Shooting Star Technology	6	240 Hz	0.35 - 1.8 msec	0.25 in, 0.15 - 0.3°	35 in diameter, 18 in high	\$1,299
	WrightTrac	Vidtronics, Inc.	6	300 Hz	3.3 msec	0.1° per axis	1/4 sphere, 40 in diameter	\$795
Magnetic	Fastrak	Polhemus	6	120 Hz/number of receivers	4 msec ^b , 8.5 msec ^{bc}	0.0002 in/in range 0.025°	10 - 30 ft	\$6050
	Isotrak II	Polhemus	6	60 Hz/number of receivers	70 msec ^b , 40 msec, 23-45 msec ^b	0.0015 in/in, range 0.1°	5 ft	\$2,875
	Insidetrak	Polhemus	6	60 Hz/number of receivers	12 msec ^b	0.0003 in/in, range 0.03°	5 ft	\$999
	Ultratrak	Polhemus	6	30 - 60 Hz	12 msec ^b	0.05 in at 5 ft range, 0.25 ft at 15 ft range 0.1° RMS	2 - 15 ft	\$23,250 - \$32,250
	Ultratrak 120	Polhemus	6	60 - 120 Hz	12 msec ^b	0.15 in at 5 ft range, 0.25 ft at 15 ft range 0.1° RMS	2 - 15 ft	\$39,500 - \$71,500
	Flock of Birds	Ascension Technology Corp.	6	Up to 144 Hz	7.5 msec ^b , 39-47 msec	0.1° RMS at 12 in	3 - 8 ft	\$2,695 basic system, \$8,090 ERT system
	PC/BIRD	Ascension Technology Corp.	6	Up to 144 Hz	10 msec	0.08 in translation, 0.15° rotation	4 ft	\$2,475
	SpacePad	Ascension Technology Corp.	6	120 Hz/number of receivers	<8 msec ^c	Not available	16 x 16 ft	\$985
Sourceless, non-inertial	CyberTrack 3.2	General Reality Company	3	30 Hz	<50 msec	0.15° heading, 0.12° tilt	360° horiz, ±55° tilt	\$850
	Wayfinder - VR	Precision Navigation, Inc.	3	30 Hz	33.3 msec ^d	±0.1° heading, ±0.1° tilt	360° horiz, ±20°, ±50° tilt	\$599 - \$699

Technology	Product	Vendor	DOF	Frequency	Latency	Resolution	Working Volume	Price
Sourceless, non-inertial	CyberMaxx	Victor Maxx Technologies	3	75 Hz	29.6 msec	0.1° vertical, 0.1° horiz	360° horiz, ±45° tilt	\$799
	Mouse-Sense 3D	RPI Advanced Technology Group	3	8 Hz	125 msec ^c	±1° headings, ±1° tilt	360° horiz, ±25 - 55° tilt	\$750
Optical	SELSPOT II	Selcom AB	6	10 KHz	Not given	0.025% of millirad	Up to 200 m	\$29,980
	OPTOTRAK 3020	Northern Digital, Inc.	6	600 Hz	Not given	0.01 mm at 2.25 m	Not given	\$57,400
	MacReflex	Qualisys, Inc.	6	50-200 Hz	Not given	Not given	0.5 - 30 m (indoors), 0.5 - 9 m (outdoors)	\$38,500 - \$48,500
	DynaSight	Origin Instruments Corporation	3	64 Hz max	16-31 msec	0.1 mm cross range 0.4 mm down range	0.1 - 1.5 m for 7 mm target, up to 1 - 6 m for 7 mm target	\$2,195
	RK-447 Multiple Target Tracking System	ISCAN, Inc.	6	60 Hz	16 msec	Not given	Not given	\$36,800
	Head/Hand XYZ Tracker	Fifth Dimension Technologies	3	20 Hz	Not given	Not given	Up to 3 m	\$345
Acoustic (Ultrasonic)	GPI2-3D (Free-point 3D)	Science Accessories Corp.	3	150 Hz/number of emitters	Not given	0.002 in	3.25 x 3.25 x 3.25 ft up to 16 x 8 x 8 ft	\$4,995 - \$6,995
	Logitech 3D Mouse and Head Tracker	Logitech	6	50 Hz	72 msec ^e	1/250 in (linear), 1/10° angular	5 ft long, 100° cone	Not given
	MotionPak	Systron-Donner	6	60 Hz	Not given	0.004°/sec.	Not given	\$10,000
Inertial	GyroPoint	Gyration, Inc.	3	Not given	Not given	0.2°	75 ft	\$299

a. Includes visual system.

b. Unfiltered

c. (Adelstein, Johnston, and Ellis, 1995)

d. Calculated as 1/Frequency and converted to milliseconds.

e. (Ware and Balakrishnan 1994).

Reference

- [1]. Review of Virtual Environment Interface Technology. [Online] [Citace: 15. 3 2008.] <http://www.hitl.washington.edu/scivw/IDA/>.
- [2]. **Lee, Johnny Chung.** Johnny Chung Lee > Projects > Wii. [Online] [Citace: 20. 3 2008.] <http://www.cs.cmu.edu/~johnny/projects/wii/>.
- [3]. Direct Linear Transformation (DLT). [Online] [Citace: 15. 10 2007.] <http://www.kwon3d.com/theory/dlt/dlt.html>.
- [4]. **Bard, Michel.** *Denny Himel The Minimum Distance Between Two Lines in n-Space.* 2001.
- [5]. **HARTLEY, R., AND A.ZISSERMAN.** *Multiple View Geometry.* místo neznámé : Cambridge University Press, 2000.
- [6]. **Ojanen, Harri.** *Automatic Correction of Lens Distortion by Using.* 1999.
- [7]. **Clarke, T.A. & Fryer.** *The development of camera calibration methods and models.* místo neznámé : Photogrammetric Record, 1998.
- [8]. **BOUGUET, J.** *Camera Calibration Toolbox for Matlab.*
- [9]. An Investigation of Current Virtual Reality Interfaces. [Online] [Citace: 10. 3 2008.] <http://www.acm.org/crossroads/xrds3-3/vrhci.html>.