

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## POZICE OBJEKTU ZE SOUSTAVY KAMER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. RADEK DOSTÁL

BRNO 2011



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## **POZICE OBJEKTU ZE SOUSTAVY KAMER**

OBJECT POSITION FROM MULTIPLE CAMERAS

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. RADEK DOSTÁL**

**VEDOUČÍ PRÁCE**

SUPERVISOR

**Ing. MICHAL HRADIŠ**

BRNO 2011

## Abstrakt

Tato práce se zabývá rekonstrukcí pozice golfového míčku ze soustavy kamer, která bude použita v rámci vývoje golfového simulátoru. Systém využívá fotogrametrické kalibrace a triangulačního algoritmu pro získání souřadnic bodu. V práci jsou také diskutovány možnosti pro volbu kamer. Konečným výsledkem je sestavení prototypu zařízení.

## Abstract

This thesis deals with reconstruction of golf ball position using multiple cameras. Reconstruction will be used for golf simulator project. System is using fotogrametric calibration and triangulation algorithm for obtaining point coordinates. Work also discuss options for camera selection. The result is making of prototype of the simulator.

## Klíčová slova

Počítačové vidění, dírkový model kamery, epipolární geometrie, kalibrace kamery, rekonstrukce bodu v prostoru, golf, simulátor, golfový míček, triangulace, fundamentální matice, radiální distorze, tangenciální distorze, vysokorychlostní kamery

## Keywords

Computer vision, pinhole camera model, epipolar geometry, camera calibration, position reconstruction, golf, simulator, golf ball, triangulation, fundamental matrix, radial distortion, tangential distortion, highspeed cameras.

## Citace

Radek Dostál: Pozice objektu ze soustavy kamer, diplomová práce, Brno, FIT VUT v Brně, 2011

# Pozice objektu ze soustavy kamer

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Michala Hradiše. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Radek Dostál  
24. května 2011

## Poděkování

Děkuji vedoucímu mé práce Ing. Michalu Hradišovi za odborné vedení a čas, který mi věnoval při konzultacích. Dále bych chtěl poděkovat panu Ing. Petru Chmelařovi za cenné rady, společnosti Inven Solution s.r.o. v zastoupení pana Marka Váni za materiální vybavení a v neposlední řadě Fakultě Informačních technologií při VUT stejně tak jako společnosti Golf Resort Brno CZ za poskytnutí prostor pro testování a vývoj.

© Radek Dostál, 2011.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1 Úvod</b>	<b>3</b>
1.1 Struktura práce . . . . .	4
<b>2 Modely kamery</b>	<b>5</b>
2.1 Model dírkové kamery . . . . .	5
2.2 Obecná perspektivní kamera . . . . .	7
2.3 Radiální distorze . . . . .	8
2.4 Tangenciální distorze . . . . .	8
2.5 Kalibrace kamery . . . . .	9
2.5.1 Fotogrametrická kalibrace . . . . .	10
2.5.2 Automatická kalibrace . . . . .	10
2.5.3 Další metody . . . . .	10
<b>3 Systém více kamer</b>	<b>11</b>
3.1 Epipolární geometrie . . . . .	11
3.2 Fundamentální matice . . . . .	11
3.3 Kalibrace systému kamer . . . . .	12
3.4 Rekonstrukce bodu v prostoru . . . . .	13
3.4.1 Triangulace . . . . .	14
<b>4 Analýza a návrh systému</b>	<b>15</b>
4.1 Existující řešení . . . . .	15
4.1.1 Trenažér FullSwing Golf . . . . .	16
4.1.2 Trenažér AboutGolf . . . . .	16
4.2 Analýza požadavků na systém . . . . .	16
4.2.1 Přesnost . . . . .	17
4.2.2 Rychlost snímání . . . . .	19
4.2.3 Osvětlení . . . . .	21
4.2.4 Rychlost zpracování . . . . .	21
4.2.5 Prostor a bezpečnost . . . . .	21
4.3 Volba kamer . . . . .	22
4.4 Rozmístění kamer . . . . .	24
4.5 Volba algoritmů . . . . .	24
<b>5 Implementace a sestavení systému</b>	<b>26</b>
5.1 Použité technologie . . . . .	26
5.2 Problémy přinášející konkrétní platformy . . . . .	26
5.2.1 PlayStation Eye a Linux . . . . .	26

5.2.2	OpenCV a Linux . . . . .	26
5.2.3	PlayStation Eye a Microsoft Windows . . . . .	27
5.3	Implementace aplikace . . . . .	27
5.3.1	OneCamCalib . . . . .	27
5.3.2	TwoCamsImageSave . . . . .	27
5.3.3	StereoCalib . . . . .	27
5.3.4	GetPosition . . . . .	27
5.3.5	TwoCamsVideoSave . . . . .	28
5.3.6	GolfSim . . . . .	28
5.4	Sestavení prototypu systému . . . . .	28
<b>6</b>	<b>Zhodnocení výsledného řešení</b>	<b>32</b>
6.1	Přesnost kalibrace a rekonstrukce pozice statických objektů . . . . .	32
6.2	Přesnost rekonstrukce odpalu golfového míčku . . . . .	34
6.3	Zhodnocení systému vzhledem k požadovaným vlastnostem . . . . .	37
6.4	Možná rozšíření . . . . .	38
<b>7</b>	<b>Závěr</b>	<b>39</b>
<b>A</b>	<b>Obsah CD</b>	<b>42</b>
<b>B</b>	<b>Plakát</b>	<b>43</b>

# Kapitola 1

## Úvod

Popularita golfu v současné době velmi dynamicky roste. Jeho hraní a trénování však v zimních měsících limituje počasí. Většina golfistů v našich klimatických podmínkách se tedy „stěhuje“ na zimu do tzv. „indoor center“, kde trénují odpaly do sítě, případně hrají na simulátorech.

Golfový simulátor pracuje na principu, kdy golfista odehraje míček do promítacího plátna, na kterém je zobrazena projekce hřiště. Trajektorie letu míčku je analyzována systémem a následně je dopočítána aproximace její zbývající části, která je posléze na plátno promítnuta.

Systém pro snímání a analýzu odpalu (anglicky označovaný jako „launch engine“) má uplatnění i mimo prostředí simulátorů. V první řadě se jedná o tzv. „club fitting“, což je jak anglický název napovídá individuální úprava holí podle vlastností švihů hráče. Systém pro „club fitting“ je však od toho určeného pro simulátor mírně odlišný především v parametrech, které je třeba sledovat. U simulátoru se zabýváme horizontálním a vertikálním úhlem, rychlostí a případně rotací míče, kdežto pro „club fitting“ je třeba sledovat také parametry hole - její rychlost, úhel náběhu nebo místo impaktu hole s míčem. „Launch engine“ může být také využit jako zdroj dat pro trenéry, či hráče samotné, kteří tak mohou svou hru lépe analyzovat a zdokonalovat se v ní.

Velké uplatnění má ve světě golfu také video analýza golfového švihů, která je na rozdíl od snímání míčku jednodušší, neboť pro ni postačují méně rychlé kamery, které jsou běžně dostupné. Zajímavou prací je také [22], kde se autoři snaží rekonstruovat trajektorii letu míčku podle záběrů jeho dopadu.

Cílem této práce je podílet se na vývoji golfového simulátoru dostupného širší komunitě, neboť jeho pořizovací cena je nemalá<sup>1</sup>. Nižší cena by zpřístupnila hru i méně majetným hráčům, neboť by klesly náklady na nákup a provoz simulátoru. Tak by bylo umožněno „indoor“ centrům snížit cenu jejich pronájmu, která se v současnosti pohybuje v rozmezí 400 – 500 CZK/hod<sup>2</sup>. Navrhnout a sestavit celý simulátor je složitý úkol představující několik let vývoje, a proto se v této práci zaměřím pouze na určování pozice míčku při odpalu. Pro tento cíl využijeme soustavu dvou případně více kamer, z jejichž obrazu lze rekonstruovat kompletní prostorovou informaci o pozici míčku. Díky tomu můžeme aproximovat křivku, po které se míček pohybuje a díky vhodně zvolenému fyzikálnímu vztahu odhadnout pozici dopadu míčku v předem zvolených podmínkách. Modulem pro rozpoznávání míčku ve videu a simulací jeho reálné trajektorie se zabývá [7].

<sup>1</sup>jak uvádím v kapitole 4.1.2 cena nejlevnějšího simulátoru od společnosti aboutGolf se pohybuje kolem 47 000 USD

<sup>2</sup>cena platí pro Golf Resort Brno v zimě 2010/2011

## 1.1 Struktura práce

Protože se tato práce bude zabývat určováním pozice objektu v prostoru při využití systému více kamer, zprostředkovává kapitola 2 úvod do teorie projektivní geometrie a seznámí čtenáře s jednoduchým matematickým modelem kamery. Následovat bude kapitola 3, jež přenesení zavedenou teorii do systému více kamer a ukáže, jak lze určit absolutní pozici objektu v prostoru. Kapitola 4 se zabývá požadavky, které jsou kladeny na činnost kamerového systému i celého simulátoru, a navrhuje způsob řešení jednotlivých problémů. Následuje kapitola 5 popisující způsob implementace navrženého systému a jeho sestavení. Poslední kapitola prezentuje dosažené výsledky a porovnává je s kritérii, jež byly na systém kladeny a s existujícími řešeními.

Tato práce navazuje na semestrální projekt, v rámci něhož byla napsána druhá a třetí kapitola.

## Kapitola 2

# Modely kamery

Kamera je zařízení, které převádí 3-D prostor do jeho 2-D projekce. Tato kapitola přináší úvod do projektivní geometrie a seznamuje s několika matematickými modely dírkových kamer. Základní model budu generalizovat a postupně na něm demonstrovat problémy, které reálné kamery přinášejí a se kterými je třeba počítat.

### 2.1 Model dírkové kamery

Dírková kamera (viz [20]), případně camera obscura, v anglické literatuře označována jako „pin-hole camera“, je velmi jednoduché optické zařízení v podobě krabičky, v jejíž stěně je malý otvor, kterým do ní prochází světlo a na protilehlé straně vytváří obraz vnějšího prostoru.

Pro matematický model uvažujme ideální dírkovou kameru, jejíž otvor (*střed projekce*) je nekonečně malý. Střed projekce zároveň položíme jako počátek Euklidovského souřadného systému. Rovina, na kterou se zobrazuje vnější obraz, se nazývá *projekční* případně *obrazová*. Příмка ze středu projekce kolmá k projekční rovině se nazývá *hlavní osa* a její průsečík s rovinou se označuje jako *hlavní bod zobrazení*. Atribut, jenž ovlivňuje velikost promítaného obrazu, je *ohnisková vzdálenost*, což je vzdálenost od středu projekce k hlavnímu bodu.

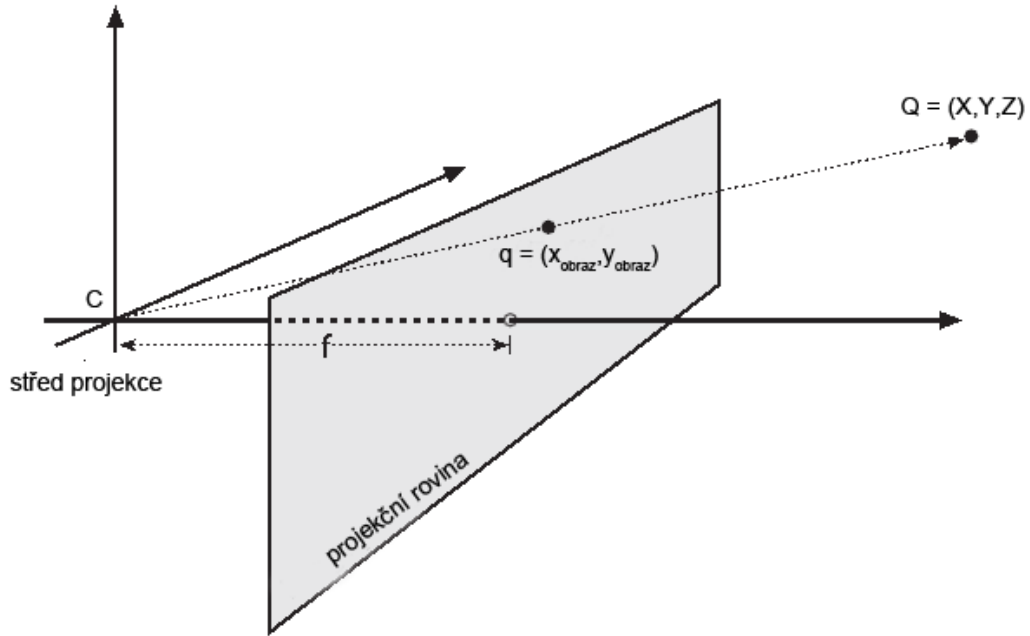
Díky podobnosti trojúhelníků můžeme snadno odvodit vztah 2.1, pomocí něhož převedeme bod z 3-rozměrného Euklidovského prostoru  $\mathbb{R}^3$  do 2-rozměrného prostoru  $\mathbb{R}^2$ .

$$x_{obraz} = -\frac{fX}{Z}, y_{obraz} = -\frac{fY}{Z}, \quad (2.1)$$

kde  $X, Y$ , a  $Z$  reprezentují souřadnice zobrazovaného bodu,  $X_{obraz}, Y_{obraz}$  souřadnice transformovaného bodu do projekční roviny a  $f$  ohniskovou vzdálenost. Záporné znaménko u levé strany obou vzorců ukazuje na jednu z vlastností dírkové kamery, a to, že obraz jí pořízený je převrácený. Abychom se vyvarovali tohoto efektu, můžeme obraz převrátit tím způsobem, že umístíme zobrazovací rovinu před střed projekce. Vzniklou situaci demonstruje obrázek 2.1. Výsledný vztah pak bude mít podobu:

$$x_{obraz} = \frac{fX}{Z}, y_{obraz} = \frac{fY}{Z} \quad (2.2)$$

Po převodu do homogenních souřadnic můžeme vztah 2.2 vyjádřit pomocí matic:



Obrázek 2.1: Obrázek demonstrující model dírkové kamery. Bod  $Q$  je promítaný bod z  $\mathbb{R}^3$  na bod  $q$  v  $\mathbb{R}^2$ .  $f$  je ohnisková vzdálenost a  $C$  střed projekce. Převzato z [2].

$$Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (2.3)$$

Proměnná  $Z$  odpovídá hloubce původního bodu a můžeme ji definovat jako libovolný kladný skalár  $\lambda \in \mathbb{R}^+$ . Nyní můžeme rozepsat matici:

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

Matice ze vztahu 2.4 definovaná jako:

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

se nazývá *kalibrační matice* a popisuje vnitřní parametry kamery.

Dále se pokusme rozšířit model o posunutí a otočení kamery. Naší snahou je převést bod ve světovém souřadném systému do souřadného systému kamery. Využijeme vztahu 2.6, který vyjadřuje transformaci jedné projekce na jinou pomocí translace a rotace:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}, \quad (2.6)$$

kde  $R \in \mathbb{R}^{3 \times 3}$  je matice rotace. Její detailní podobu a způsob jak ji lze získat je možno najít například v druhé kapitole [8] nebo v [2].  $T$  je vektor translace daný rozdílem počátku světového souřadného systému a počátku souřadného systému kamery. Tato matice reprezentuje vnější parametry kamery. Konečný model obecné dírkové kamery tedy můžeme popsat následujícím vztahem:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}. \quad (2.7)$$

## 2.2 Obecná perspektivní kamera

Model dírkové kamery je ideální a nepůsobí na něj žádné odchylky a další negativní faktory. V reálné situaci ovlivňuje kameru mnoho jevů, které je třeba brát při výpočtech v potaz. Nyní tedy můžeme přistoupit k prvnímu zobecnění. Výše uvedený model předpokládá, že počátek souřadného systému výsledné projekce je v průsečíku hlavní osy zobrazení se zobrazovací rovinou (tedy v hlavním bodě zobrazení). Tato situace však není příliš obvyklá a zpravidla indexujeme body v obraze od horního levého rohu. Tohoto posunutí můžeme dosáhnout pomocí úpravy vztahu 2.2 následovně:

$$x_{obraz} = \frac{fX}{Z} + c_x, y_{obraz} = \frac{fY}{Z} + c_y, \quad (2.8)$$

kde  $c_x$  značí posunutí na horizontální ose a  $c_y$  posunutí na ose vertikální. Toto posunutí lze snadno vyjádřit upravením kalibrační matice do následující podoby:

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.9)$$

Dírkový model kamery předpokládá čtvercovou podobu pixelu. Obecně však může být poměr stran obrazového bodu (např. u CCD snímače) různý, což je třeba zohlednit i v prezentovaném modelu. Nahradíme tedy ohniskovou vzdálenost  $f$  vztahem  $f_x = fm_x$  resp.  $f_y = fm_y$ , kde  $m_{x,y}$  jsou rozměry jednotlivých prvků snímače. Posledním parametrem, který je součástí kalibrační matice je zkosení  $s_0$ , ten má však pro většinu dostupných kamer hodnotu 0. Tím jsme se dopracovali až k výslednému modelu, který lze popsat následovně:

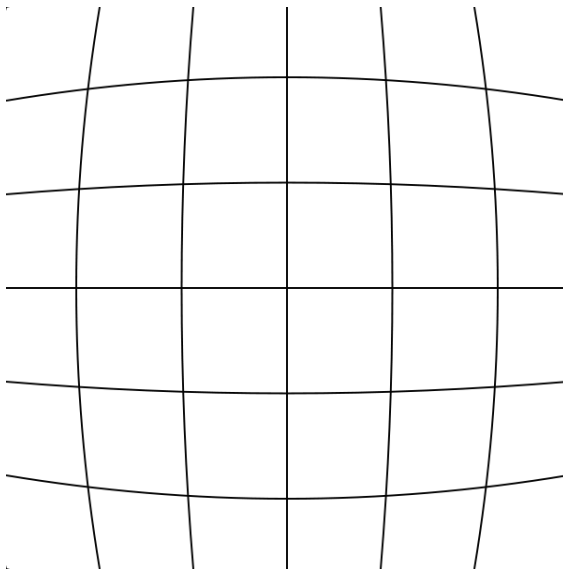
$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s_0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}. \quad (2.10)$$

Nyní můžeme definovat pojem *projekční matice*  $\Pi$  dle vztahu:

$$\Pi = \begin{bmatrix} f_x & s_0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \quad (2.11)$$

a vztah 2.10 vyjádřit jako:

$$\lambda x' = \Pi X \quad (2.12)$$



Obrázek 2.2: Obrázek ukazuje deformaci způsobenou radiální distorzí tzv. „soudkovitost“.

a ten vydělit proměnnou  $\lambda$ , čímž se dostaneme ke vztahům, které vyjadřují výpočet souřadnic bodů

$$x' = \frac{\pi_1^T X_0}{\pi_3^T X_0}, \quad y' = \frac{\pi_2^T X_0}{\pi_3^T X_0}, \quad z' = 1, \quad (2.13)$$

kde  $\pi_1^T, \pi_2^T, \pi_3^T \in \mathbb{R}^4$  jsou jednotlivé řádky projekční matice  $\Pi$ .

## 2.3 Radiální distorze

Tento druh zkreslení obrazu (viz obrázek 2.2) je způsoben nepřesnou výrobou čoček, které se vyskytuje nejvíce u levnějších fotoaparátů a kamer. Ideální čočka láme všechny paprsky světla, které na ni dopadají, do ohniska. V praxi se však kvůli nedokonalostem, které vznikají při její výrobě, lámou s určitou odchylkou. Jejím vlivem roste od středu, kde se většinou nevyskytuje, směrem ke krajům obrazu.

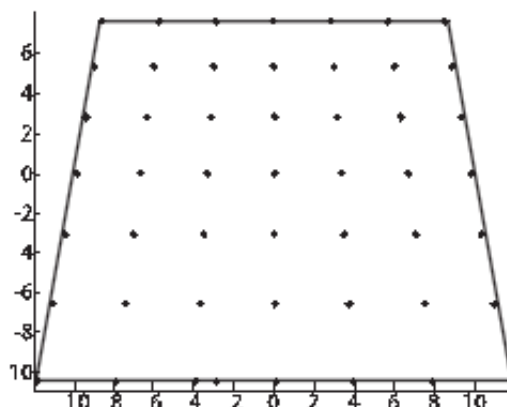
Radiální zkreslení můžeme matematicky aproximovat pomocí Taylorova rozvoje:

$$\begin{aligned} x &= x_d(1 + a_1 r^2 + a_2 r^4), \\ y &= y_d(1 + a_1 r^2 + a_2 r^4), \end{aligned} \quad (2.14)$$

kde  $x_d$  a  $y_d$  jsou souřadnice bodu původního snímku,  $x$  a  $y$  jsou souřadnice opraveného bodu,  $a_1$  a  $a_2$  jsou body určující zkreslení obrazu a konečně  $r = \sqrt{x_d^2 + y_d^2}$ . V případě, že je obraz radiální distorzí hodně ovlivněn (např. snímek pořízený pomocí tzv. rybího oka), je možné přidat ještě další člen tohoto rozvoje v podobě  $a_3 r^6$ .

## 2.4 Tangenciální distorze

Druhým nejčastějším zkreslením, které se vyskytuje, je zkreslení tangenciální (viz obrázek 2.3). Projevuje se opět především u levných kamer a fotoaparátů. Vzniká pokud sní-



Obrázek 2.3: Obrázek ukazuje deformaci způsobenou tangenciální distorzí. Převzato z [2].

mací senzor není paralelní se zobrazovací rovinou, k čemuž může dojít například pokud není rovnoměrně nanesené lepidlo, které senzor připevňuje k desce plošných spojů.

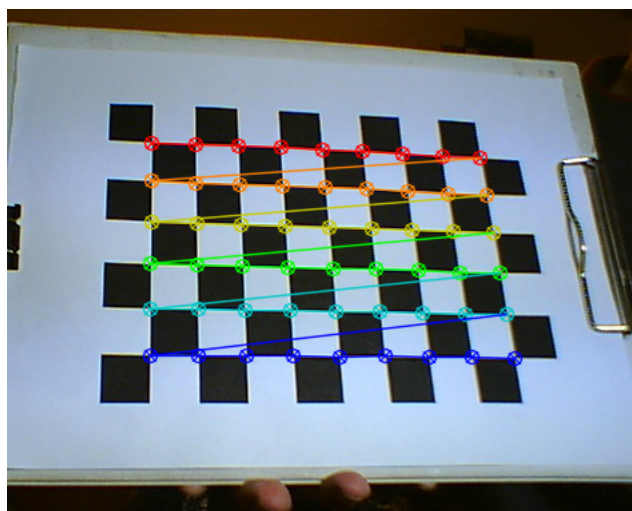
Tangenciální distorzi můžeme opět matematicky korigovat pomocí aplikace Taylorova rozvoje:

$$\begin{aligned} x &= x_d + [2b_1y_d + b_2(r^2 + 2x_d^2)], \\ y &= y_d + [b_1(r^2 + 2y_d^2) + 2b_2x_d], \end{aligned} \quad (2.15)$$

přičemž je zachováno pojmenování ze vztahu 2.14, proměnné  $b_1$  a  $b_2$  určují míru tangenciálního zkreslení.

## 2.5 Kalibrace kamery

Pod pojmem kalibrace kamery se skrývá získávání vnitřních parametrů kamery a koeficientů pro eliminaci distorzí.



Obrázek 2.4: Obrázek z demonstrační aplikace zobrazuje kalibrační šachovnici, na které jsou označeny rozpoznané rohy.

### 2.5.1 Fotogrametrická kalibrace

Jde o nejběžnější metodu kalibrace, která je velmi přesná. Využívá se zde objektů, které mají známý tvar a rozměry viz obrázek 2.4. Po umístění takového tělesa do scény jsou ve snímku nalezeny známé body (rohy, přímky apod.). Pokud máme snímků k dispozici více můžeme pomocí řešení soustav rovnic získat vnitřní i vnější parametry kamery. Vhodným kalibračním předmětem je například šachovnice. Jeden z možných postupů při kalibraci, jenž je například implementován v knihovně OpenCV, je [21].

Pro získání distorzních koeficientů můžeme využít např. algoritmus [3], který opět využívá soustavy rovnic, kdy jednotlivé rovnice jsou získány aplikací vztahů 2.14 a 2.15. Tyto rovnice popisují vztah mezi bodem, který je ovlivněn distorzí a bodem, který je „opraven“. Tento algoritmus je opět využíván například knihovnou OpenCV.

### 2.5.2 Automatická kalibrace

Tento způsob kalibrace pracuje na principu, kdy není třeba mít kalibrační předmět, ale body získáváme změnou parametrů pohledu kamery (posunutí, rotace, změna ohniskové vzdálenosti nebo jejich kombinace) ve statické scéně. Velkou výhodou tohoto postupu je dostupnost kalibrace i mimo laboratorní prostředí, což je vhodné například pro roboty. Bohužel lze takto získat pouze vnitřní parametry kamery a získané hodnoty nebývají tak přesné jako při využití tradičních způsobů kalibrace.

### 2.5.3 Další metody

Kromě výše zmíněných způsobů lze využít například algoritmus pro hledání úběžníků popsany v [6]. Další alternativou je metoda popsaná v [9], která využívá rotaci staticky umístěné kamery.

## Kapitola 3

# System více kamer

V této kapitole vytvoříme model dvou zkalibrovaných kamer a seznámíme se s algoritmy, díky kterým je možno získat z pořízených snímků pozici bodu v prostoru.

### 3.1 Epipolární geometrie

Nejprve si definujme projektivní geometrii, která nám bude reprezentovat vztah mezi dvěma různými pohledy. Tato geometrie se nazývá *epipolární* viz obrázek 3.1 a závisí na vnitřních parametrech obou kamer a na jejich relativní pozici. Přímka, která spojí obě pozice kamer se nazývá výchozí (v anglické terminologii „baseline“), a průsečíky této přímky s promítací rovinou nazýváme *epipóly* a označujeme je  $e$  resp.  $e'$ . *Epipolární rovina* je taková rovina, jež obsahuje výchozí přímku, a konečně *epipolární přímka* je průsečnicí epipolární roviny a roviny projekční. Epipolární rovina tedy protíná obě promítací roviny a takto vzniklé epipolární přímky definují odpovídající si body v obou obrazech.

### 3.2 Fundamentální matice

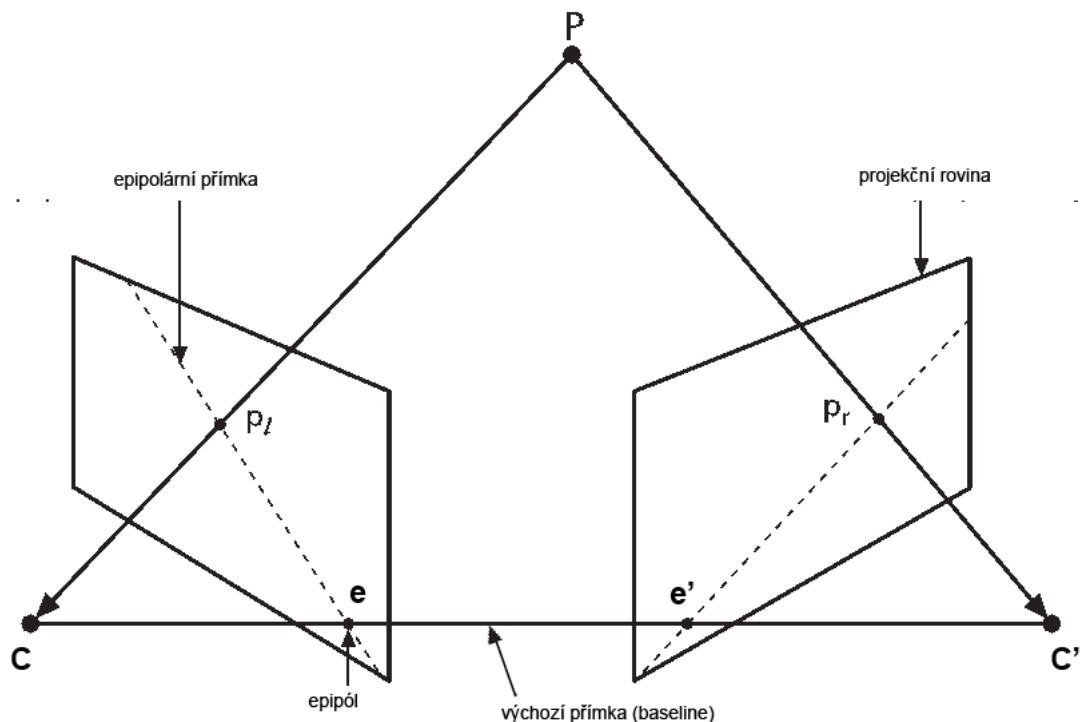
Epipolární geometrii můžeme reprezentovat pomocí *fundamentální matice*  $F$ . Tato matice reprezentuje projektivní mapování bodu jednoho obrazu na epipolární přímku druhého obrazu. Fundamentální matici můžeme snadno geometricky odvodit pomocí dvou kroků:

1. Spojíme bod  $x$  s bodem  $C$ , který udává pozici kamery jednoho pohledu a najdeme jeho průsečík s libovolně zvolenou imaginární rovinou  $\pi$ , která však nesmí procházet žádným z bodů  $C$  a  $C'$ . Poté spojíme tento průsečík s pozicí druhé kamery  $C'$  a najdeme průsečík této přímky se zobrazovací rovinou druhé kamery. Tím získáme obraz bodu  $x'$  odpovídající bodu  $x$ .
2. Spojením bodu  $x'$  a epipólu  $e'$  získáme epipolární přímku  $l'$ .

Pokud fundamentální matice  $F$  vyjadřuje vztah mezi kamerami  $(P, P')$ , pak transponovaná matice  $F^T$  bude vyjadřovat vztah  $(P', P)$ . Pomocí takto získané matice můžeme snadno vyjádřit každému bodu  $x$  jeho odpovídající epipolární přímku  $l'$  vztahem

$$l' = Fx \tag{3.1}$$

Tuto matici lze získat několika algoritmy, pro každý je však třeba nejméně osm bodů (existuje i algoritmus pro sedm bodů, který však není příliš přesný a navíc může vracet více



Obrázek 3.1: Obrázek ilustruje pojmy týkající se epipolární geometrie.

řešení). Základní metoda pro získání této matice řeší soustavu rovnic a pomocí nadbytečných bodů dále zpřesňuje výsledek metodou nejmenších čtverců. Dalšími metodami jsou RANSAC a LMedS, které jsou velmi robustní, ale potřebují mnohem více bodů pro přesný výsledek.

### 3.3 Kalibrace systému kamer

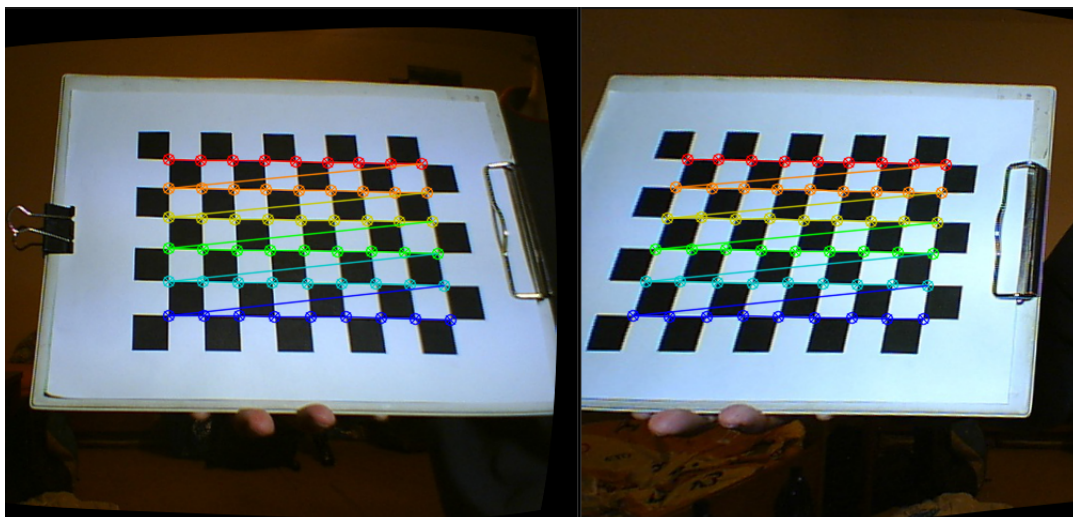
Tato kalibrace je rozšířením postupů zmíněných v kapitole 2.5 o metody získání fundamentální matice (viz kapitola 3.2) a také o metody výpočtu transformací relativní pozice jedné kamery na druhou. Toho lze docílit tak, že pomocí stejné scény zkalibrujeme obě kamery zvlášť a využijeme získaných vnějších parametrů kamer následovně:

$$R = R_p(R_l)^T \quad (3.2)$$

$$T = T_p - RT_l \quad (3.3)$$

kde  $R_p$ ,  $T_p$  jsou rotační matice a matice posunutí pravé kamery.  $R_l$  a  $T_p$  jsou matice levé kamery.

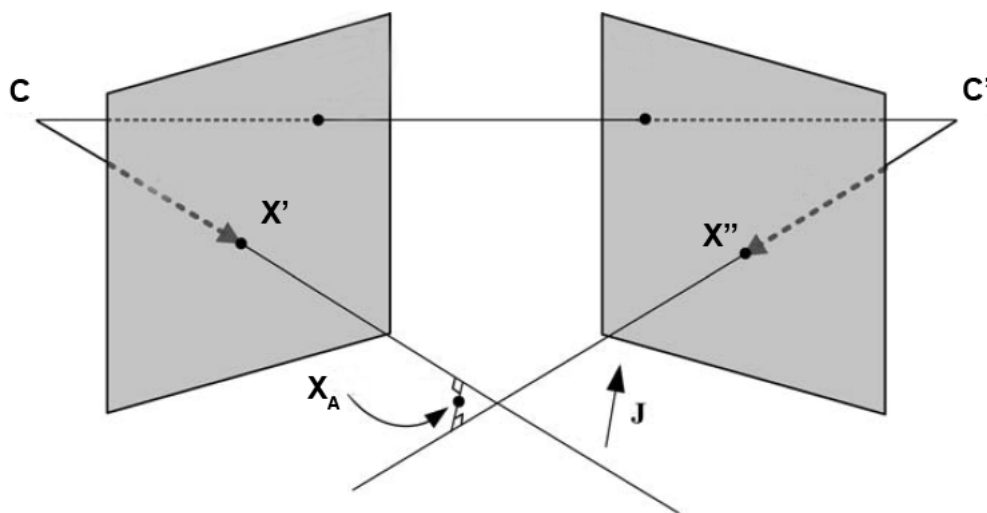
Problémem tohoto algoritmu je nutnost mít kalibrační těleso v záběru všech kamer, což může být problém, pokud potřebujeme kamery umístit například proti sobě. V těchto případech lze použít algoritmus, jehož autoři jsou Peter Sturm a Bill Triggs, a který je popsán například v [10] nebo v [23]. Ten umožňuje zkalibrovat kamery pomocí alespoň osmi bodů, které jsou rozpoznány na všech kamerách. Algoritmus je podrobně .



Obrázek 3.2: Obrázek z testovací aplikace, který prezentuje obrázky ze zkalibrovaného systému, které jsou následně zdeformovány tak, aby si na nich nalezené body odpovídaly.

### 3.4 Rekonstrukce bodu v prostoru

Pokud máme k dispozici zkalibrovaný systém a známe tedy všechny parametry kamer, lze pro rekonstrukci bodu použít tzv. triangulaci. V případě, že neznáme vnitřní případně vnější parametry kamer, stále jsme schopni rekonstrukci provést, ačkoliv jsme limitováni některými parametry. Využít k tomu můžeme například algoritmy popsané v sedmé kapitole knihy [4].



Obrázek 3.3: Obrázek znázorňuje výpočet aproximace bodu pomocí triangulace.

### 3.4.1 Triangulace

Veźměme si body  $x'$  a  $x''$ , což jsou promítнутé body do zobrazovací roviny obou kamer. Naším úkolem je určit pozici původního bodu  $X$ . Pokud bychom počítali s ideálním modelem kamery, stala by se tato úloha triviální:

1. Převědeme body  $x'$  a  $x''$  do světových souřadnic, čímž získáme body  $X'$  a  $X''$ .
2. Spojíme pozici kamery  $C$  s promítaným bodem  $X'$  resp.  $C'$  s bodem  $X''$ , čímž získáme přímky  $p$  a  $p'$ .
3.  $X$  by ležel na průsečíku přímek  $p$  a  $p'$ .

V reálné situaci se však přímky nikdy neprotnou, protože body jsou v diskrétním prostoru a navíc jsou data získaná z kamery ovlivněná šumem a dalšími nepřesnostmi. Situaci však můžeme vyřešit aproximací průsečíku  $X_A$  takového, že jeho vzdálenost od obou přímek  $p$  a  $p'$  bude minimální. To lze vyjádřit následujícím vztahem (převzato z [4] a [11])

$$ax' + cJ = T + bR^T x'', \quad (3.4)$$

kde  $a, b, c \in \mathbb{R}$  jsou parametry rovnic,  $R^T$  a  $T$  udávají relativní rotaci a posunutí mezi kamerami. Rovnice  $ax'$  je parametrická rovnice přímky  $Cx'$ , která má se má hodnotu  $C$  pro  $a = 0$  a  $x'$  pro  $a = 1$ . Rovnice  $T + R^T x''$  reprezentuje druhou přímku  $C'x''$ , kde opět platí pro  $b = 0$  má hodnotu  $C'$  a pro  $b = 1$  hodnotu  $x''$ . Konečně  $\vec{J}$  je vektor kolmý k oběma přímkám a tedy platí:

$$J = x' \times Rx''. \quad (3.5)$$

Nyní můžeme oba předchozí vztahy spojit a získáme tak:

$$ax' + c(x' \times Rx'') = T + bR^T x''. \quad (3.6)$$

Do vztahu 3.6 nyní dosadíme matice  $T$ ,  $R$  a pomocí přímek  $ax'$  resp.  $T + R^T x''$  vyjádříme koncové body spojnice obou přímek, která bude rovnoběžná s vektorem  $\vec{J}$ . Námí hledaným bodem bude střed  $X_A$ , který bude mít k oběma nalezeným koncovým bodům stejnou vzdálenost viz obrázek 3.3.

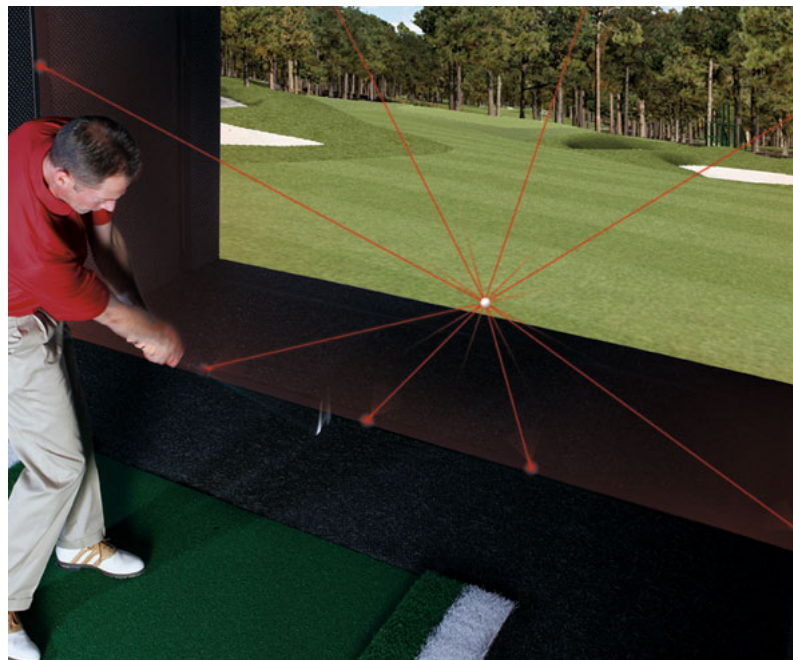
## Kapitola 4

# Analýza a návrh systému

Tato kapitola zmíní existující řešení a seznámí čtenáře s požadavky, které jsou na systém kladeny. Následuje diskuse o volbě kamer, jejich rozmístění a použitých algoritmech.

### 4.1 Existující řešení

Výrobou golfových simulátorů se zabývá několik společností. Vybral jsem dvě, na jejichž produkty lze běžně narazit v českých „indoor“ centrech. Obě společnosti se věnují tomuto odvětví přes 20 let.



Obrázek 4.1: Obrázek demonstrující princip funkce golfového simulátoru společnost Full Swing Golf. Převzato z [16].

### 4.1.1 Trenažér FullSwing Golf

Výrobce golfových simulátorů, které patří v České republice mezi nejpopulárnější. Pro rekonstrukci trajektorie jsou využity, jak je znázorněno na obrázku 4.1, dvě clonové soustavy infračervených paprsků, které jsou od sebe vzdáleny přesně 3,8 cm, což je o něco méně než velikost golfového míčku (4,2 cm). Tím je zaručeno, že každý průlet míčku soustavou paprsků je zachycen, a jelikož jsou soustavy dvě, získá tak systém dva body, díky kterým lze vytvořit aproximaci dráhy letu.

Společnost FullSwing nabízí své simulátory v mnoha provedeních určených jak pro domácnosti, tak pro tzv. „indoor centra“. Na modelech simulátoru, které jsem měl možnost vidět byla vizualizace realizována pomocí hry Microsoft Links z roku 2004. Dle [16] jsou však nové modely odproštěny od tohoto zastaralého řešení a využívají nový systém označovaný jako *E6*.

### 4.1.2 Trenažér AboutGolf

Společnost aboutGolf se zaměřuje především na přesnost svých simulátorů. Jejich řešení je postaveno na technologii 3Trak, která využívá vysokorychlostních kamer (viz Obrázek 4.2). Kromě simulátorů lze tuto technologii použít také pro tzv. „club fitting“, což je úprava holí hráče podle jeho švihů (rychlost, trajektorie apod.). Dle [17] začíná cena nejlevnějších simulátorů tohoto výrobce na 47 000 USD. Pro vizualizaci je využito vlastního řešení výrobce s názvem *AboutGolf Indoor Golf Environment*.

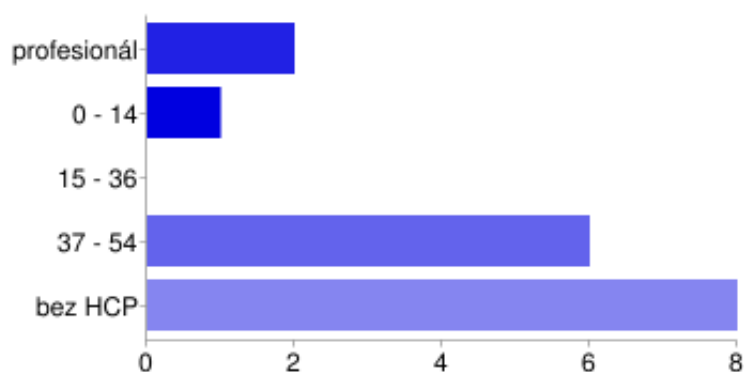


Obrázek 4.2: Obrázek systému 3Trak pro zpracování dat společnosti aboutGolf. Převzato z [12].

## 4.2 Analýza požadavků na systém

Využití kamerového systému pro získání prostorové informace klade určité požadavky na prostředí. Stejně tak princip fungování golfového simulátoru a golfové hry samotné má svá specifika, která je nutno analyzovat než se pustíme do samotného návrhu a implementace systému. V první řadě je třeba analyzovat požadovanou přesnost řešení, rychlost snímání, množství světla potřebného pro kamery a požadavky na prostor. Pro stanovení některých požadavků jsem sestavil krátkou anketu, kterou jsem distribuoval mezi několik

golfistů, včetně dvou profesionálních trenérů. Obrázek 4.3 ukazuje rozložení respondentů podle úrovně jejich hry, definované golfovým hendikepem (viz [13]).



Obrázek 4.3: Rozložení respondentů ankety podle jejich hendikepu.

#### 4.2.1 Přesnost

Je nutno odlišit přesnost celého systému a jeho modulu pro rekonstrukci bodu. Oba parametry spolu úzce souvisejí avšak na přesnosti celého systému má velký podíl také fyzikální model, který počítá simulaci dráhy letu míčku.

#### Rekonstrukce bodu

Navržený systém samozřejmě nebude nikdy dokonale přesný. Na odchylku určení pozice má vliv mnoho faktorů. Uvedme si tedy ty nejzávažnější z nich:

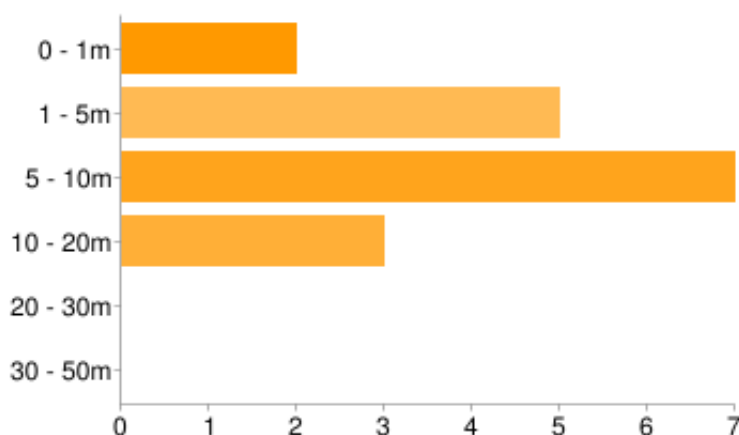
- **kalibrace kamery** – špatně určené vnitřní parametry kamery, případně koeficienty radiální a tangenciální distorze
- **kalibrace systému kamer** – nepřesně určená vzájemná poloha kamer
- **aproximace radiální a tangenciální distorze** – distorze jsou eliminovány pomocí modelu, který má omezenou přesnost viz kapitola 2.3 a 2.4
- **zanedbání ostatních distorzí** – kromě radiální a tangenciální distorze se v obrazech objevují další odchylky, které však nemají tak velký vliv, a proto je zanedbáváme (např. chromatická odchylka)
- **nízké rozlišení kamery** – v obrazech s nízkým rozlišením lze těžko nalézt vzory pro rozpoznání objektu v obraze
- **šum kamery** – při špatném osvětlení je v obraze více šumu, což opět vede ke komplikacím při rozpoznávání
- **rozpoznání objektu v obraze** – špatně určená pozice vlivem nevhodně zvoleného algoritmu, případně filtru
- **synchronizace kamer** – snímky získané z obou kamer mohou být pořízené v různý okamžik

Jak je vidět většina bodů je ve velké míře ovlivněna kvalitou kamerového systému, některé alternativy jsou diskutovány v kapitole 4.3.

### Aproximace trajektorie letu míče

Přesnost celého řešení je závislá na správném určení pozice míče ve scéně při jeho odpalu a krátkém letu a následné simulaci trajektorie jeho letu.

Požadavky na přesnost systému jsem stanovil pomocí zmíněné ankety. Otázka v ní měla znění „Jaká je dle Vás maximální akceptovatelná odchylka při plné ráně do vzdálenosti 150 metrů (brán v potaz pouze let míčku vzduchem)?“ Výsledek je vidět na obrázku 4.4. Nejvíce respondentů (44 %) (včetně profesionálního trenéra) odpovědělo, že by se spokojili s přesností 5 – 10 m, což bych zvolil jako vhodný požadavek na řešení.



Obrázek 4.4: Výsledek anketní otázky „Maximální akceptovatelná odchylka při ráně na 150 m?“ položené mezi hráči golfu.

### Vliv chyb určení pozice na přesnost systému

Nyní se blíže podívejme jaký vliv mohou mít chyby určení pozice míčku v prostoru na výsledek celé simulace. Vztahy použité v následující části lze dohledat například v [1] nebo jakékoliv učebnici analytické geometrie. Předpokládejme, že v Euklidovském prostoru použitým v následujících výpočtech platí, že jeho jednotkou je 1 cm. Pro zjednodušení budu vše demonstrovat pomocí modelu šikmého vrhu ve dvojrozměrném prostoru, převzatého z [5]:

$$d = \frac{v_0^2}{g} \sin(2\alpha), \quad (4.1)$$

kde  $d$  je vzdálenost dopadu,  $v_0$  počáteční rychlost,  $g$  gravitační zrychlení a konečně  $\alpha$  je úhel, pod kterým je těleso vrženo. Mějme model, kde máme body  $A, B \in \mathbb{R}^2$ . Tato dvojice bodů je výsledkem systému pro rekonstrukci pozice. Nyní z nich vypočítáme veličiny, které potřebujeme dosadit do vztahu 4.1. Bod  $A$ , nechť je vždy v počátku soustavy a má tedy souřadnice  $(0, 0)$ . Definujme si vektor  $\vec{u} = B - A$ , jeho velikost  $d_u = \sqrt{u_1^2 + u_2^2}$  a úhel, jenž svírá s rovinou  $\alpha = \arcsin \frac{B_y - A_y}{d_u}$ . Přidejme ještě empiricky získanou konstantu  $t = 0,008s$ , která představuje dobu mezi dvěma snímky při použití kamery umožňující

délka odpalu [m]	délka s odchylkou [m]	abs. odchylka [m]	rel. odchylka [%]
100,185	105,6	5,415	5,405
123,758	129,65	5,892	4,761
151,312	157,365	6,053	4,00
200,688	206,422	5,734	2,857
250,86	256,435	5,575	2,222

Tabulka 4.1: Tabulka znázorňující odchylky při špatně určené horizontální souřadnici bodu  $B$ .

délka odpalu [m]	délka s odchylkou [m]	abs. odchylka [m]	rel. odchylka [%]
100,185	106,078	5,893	5,882
123,758	130,447	6,689	5,405
151,312	159,276	7,964	5,263
200,688	211,837	11,149	5,555
250,86	265,195	14,335	5,714

Tabulka 4.2: Tabulka znázorňující odchylky při špatně určené vertikální souřadnici bodu  $B$ .

získat 125 snímků za vteřinu, a můžeme dle vztahu pro výpočet okamžité rychlosti určit  $v_0 = d_{ut}^{-1}$ . Souřadnice bodu  $B$  budeme volit tak, aby výsledná vzdálenost odpovídala předem zvoleným konstantám a úhel  $\alpha$  přitom zůstal v intervalu  $15^\circ$  až  $45^\circ$ , které jsou charakteristické pro golfový odpal. Pro každý takto získaný bod určíme vzdálenost dopadu  $d_{presne}$  podle 4.1, dále k jeho souřadnicím připočteme vzdálenost jedna pro horizontální osu  $x$  a následně pro vertikální osu  $y$ , poté opět určíme vzdálenost dopadu  $d_x$  resp.  $d_y$ . Odečtením obou délek ( $err_{abs} = |d_x - d_{presne}|$  resp.  $err_{abs} = |d_y - d_{presne}|$ ) získáme absolutní odchylku a převedením na procenta odchylku relativní.

Pojďme se blíže podívat na tabulku 4.1. Zvýšením horizontální souřadnice bodu  $B$ , se také zvýšila rychlost míčku, neboť se prodloužila délka vektoru  $\vec{u}$ . Tím se však i nepatrně snížil úhel  $\alpha$ . V závislosti na délce odpalu je však tato absolutní chyba konstantní a tedy u delších ran již má jen nepatrný vliv na přesnost. Neboť je absolutní chyba konstantní, relativní chyba vzhledem k délce odpalu lineárně klesá, což ukazuje graf 4.6.

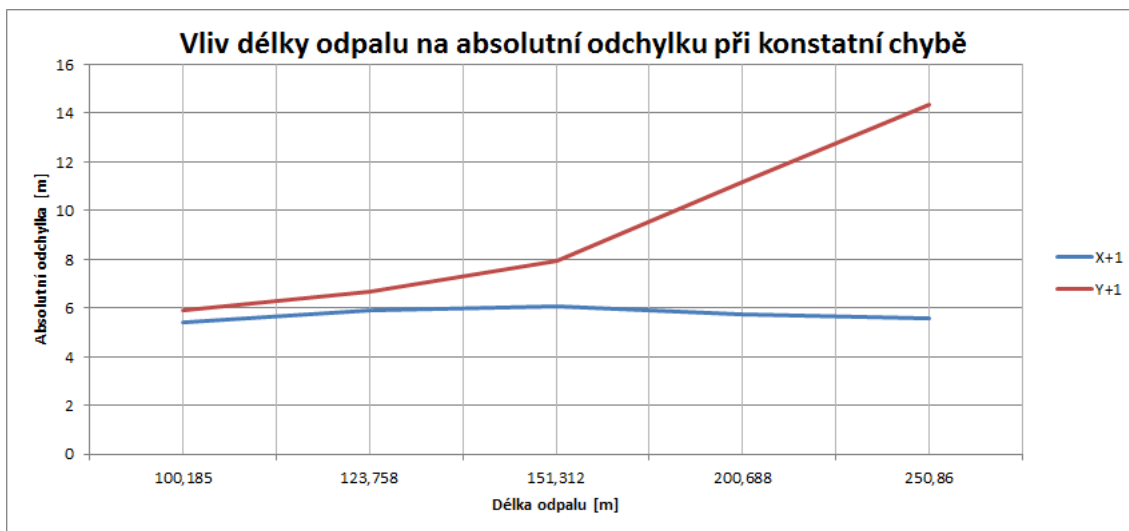
Tabulka 4.2 nám tentokrát prezentuje změnu na vertikální ose. Ta má naopak daleko větší vliv na úhel  $\alpha$ , než na velikost vektoru  $\vec{u}$ , což má za následek lineární zvětšování absolutní odchylky vzhledem k délce odpalu míčku jak demonstruje graf 4.5.

Výpočty jsem prováděl i s odečítáním hodnot, ale rozdíl absolutních hodnot odchylek byl maximálně v řádu tisícín.

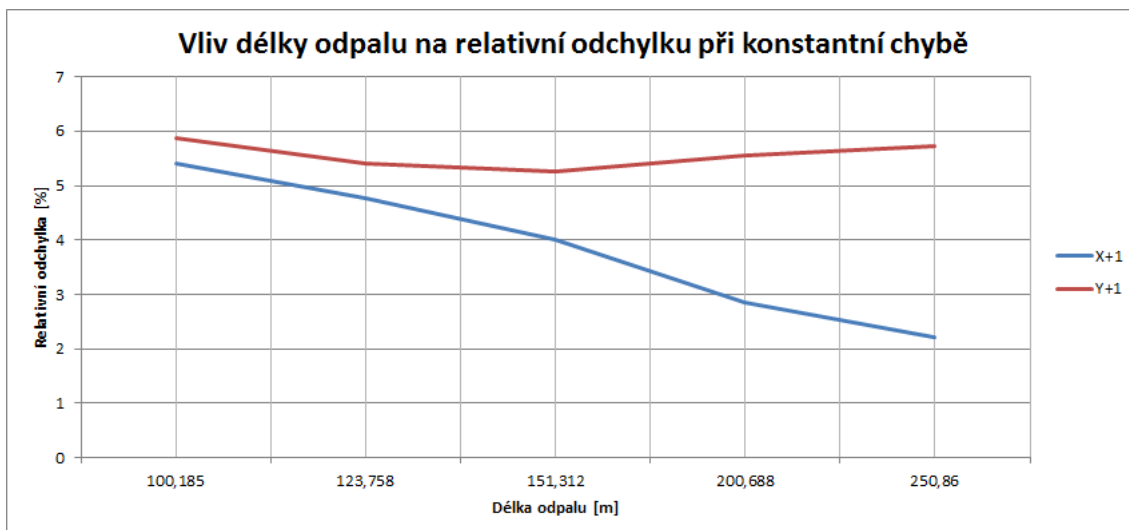
Z výše uvedeného jednoznačně plyne, že pro dlouhé rány bude třeba správně určit především vertikální pozici míčku, neboť chyba o velikosti 1 cm nám může přinést odchylku až 14 m při ráně do vzdálenosti 250 m.

## 4.2.2 Rychlost snímání

Nejzásadnějším problémem, který nastává při snímání letu golfového míčku je jeho rychlost. Na rozdíl od švihů hole, jehož rychlost se podle [19] u nejdále odpalujících hráčů americké PGA Tour pohybuje přes  $200 \text{ kmh}^{-1}$ , je rychlost letu míčku podstatně vyšší (světový rekord je podle [18]  $328 \text{ kmh}^{-1}$ , což je  $91,1 \text{ ms}^{-1}$ ). Udávané rychlosti švihů hole hráčů PGA Tour odpovídá rychlost letu míčku podle stejných statistik  $300 \text{ kmh}^{-1}$ .



Obrázek 4.5: Graf znázorňující závislost absolutní chyby na délce odpalu.



Obrázek 4.6: Graf znázorňující závislost relativní chyby na délce odpalu.

Uváděné rychlosti kladou poměrně vysoké nároky na kamerový systém. Budeme počítat, že maximální možná rychlost letu míčku je  $90 \text{ ms}^{-1}$  a potřebujeme minimálně dva snímky, na kterých určíme pozici míčku, abychom z ní mohli určit trajektorii odpalu a rychlost letu. Dále předpokládejme, že máme k dispozici  $2 \times 2 \text{ m}$  prostoru. Pro tento hypotetický příklad potřebujeme pro zachycení dvou bodů snímání v intervalu  $\frac{1}{45} \text{ s}$ . Toto by však platilo v ideálním případě, kdyby kamera zachytila snímky přesně v okrajích snímaného prostředí. V nejhorším případě by bylo třeba rychlosti dvojnásobné, tedy  $\frac{1}{90} \text{ s}$ . Samozřejmě, že naší snahou bude mít rychlost co nejvyšší, neboť ta nám přinese více bodů a tím se zpřesní výpočet. V ideálním případě, pokud by rychlost kamery byla opravdu vysoká (řádově tisíce snímků za sekundu), by se dala odečíst i rotace udělená míčku, což konvenční technologie neumožňují.

Naprosto odlišný přístup nabízí využití délky expozice. Pokud je expozice relativně dlouhá, vznikne na snímku stopa za míčkem. V extrémním případě by se dal exponovat pouze jeden snímek, z jehož stopy bychom zjistili trajektorii letu míčku. Rychlost by se však musela změřit jiným zařízením (například radarem).

Ideálně se tak jeví využití hybridního systému, kdy získáme z kamery několik snímků s delší expozicí, aby se na nich objevila stopa. Údaje pak můžeme získávat jak z jednotlivých stop za míčkem, tak i z různých snímků.

### 4.2.3 Osvětlení

Vzhledem ke konstrukci golfového simulátoru, jehož dominantním prvkem je projektor vizualizující golfové hřiště a trajektorii odehrané rány, je třeba zachovat v místnosti malé množství světla, aby byla projekce viditelná. Při využití kamerového systému pro rekonstrukci to bude klást vysoké nároky na kvalitu kamer a objektivů.

### 4.2.4 Rychlost zpracování

Požadavky na rychlost zpracování nebudou nikterak kritické. Je pochopitelné, že hráč nemůže čekat desítky sekund na výpočet, ale na druhou stranu ani zpracování v reálném čase není potřeba. Nutné je, aby byla trajektorie letu míče vykreslena dostatečně rychle, aby hráči nepřišlo čekání zdlouhavé. Protože výsledný systém by měl dráhu letu animovat, bylo by vhodné získat pozici co nejdříve. Pokud budeme počítat rychlost letu míče  $90 \text{ ms}^{-1}$  a délku odpalu přes  $300 \text{ m}$ , pak míč letí při reálné hře déle než  $3 \text{ s}$ , což by měla být horní mez délky výpočtu. Zvolenému limitu odpovídá i graf 4.7 reprezentující výsledek anketní otázky „Maximální doba výpočtu dopadu míčku?“.

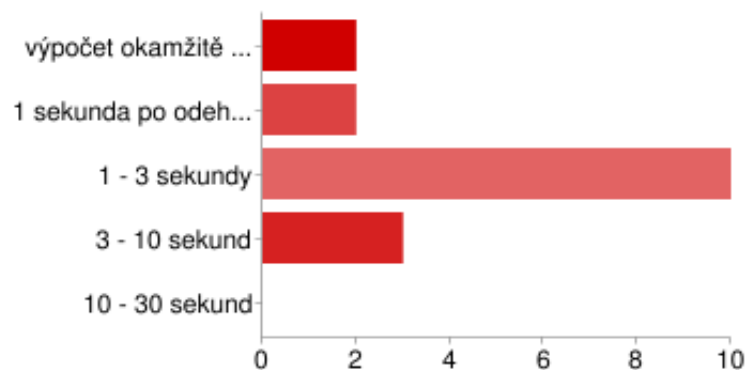
### 4.2.5 Prostor a bezpečnost

Nezanedbatelným požadavkem celého simulátoru bude prostor. Předpokládejme, že budeme disponovat alespoň kamerami s rychlostí  $90 \text{ fps}$ <sup>1</sup>. Poté by nám při rozumně<sup>2</sup> trefeném odpalu měla stačit krychle s podstavou  $2 \times 2 \text{ m}$  zmíněná v 4.2.2. Pokud jde o výšku musí prostor umožňovat hráči švih s libovolnou holí, rozumné minimum jsou dle mého názoru a zkoumání konkurenčních zařízení  $3,5 \text{ m}$ . Vertikální pozice míče by neměla<sup>3</sup> na tak malém prostoru dosáhnout zvolené minimální hodnoty. Prostor musí být samozřejmě z důvodu bezpečnosti hráčů ohraničen alespoň sítěmi, které míč zachytí. V ideálním případě by bylo

<sup>1</sup>snímků za vteřinu (z anglického „frames per second“)

<sup>2</sup>horizontální odchylka do  $45^\circ$  od paralelního směru s boční stěnou

<sup>3</sup>většina míčů je odpalována pod vertikálním úhlem  $8^\circ - 60^\circ$



Obrázek 4.7: Výsledek anketní otázky „Maximální akceptovatelná doba pro zpracování odpalu?“ položené mezi hráči golfu.

vhodné vybudovat pevnou konstrukci potaženou tmavým kobercem, který ztlumí náraz míčku a zároveň poskytne vhodné pozadí pro rozpoznávání míčku na snímcích. Při použití sítě je třeba za ni umístit jednobarevnou tmavou plachtu.

### 4.3 Volba kamer

Jak jsem uvedl v odstavci 4.2.2 věnovaném požadované rychlosti snímání, pokud budeme chtít jistotu zachycení míčku alespoň na dvou snímcích musíme systém vybavit kamerami s rychlostí nejméně 90 *fps*. S použitím rychlejších kamer pak přirozeně poroste přesnost systému, neboť budeme mít k dispozici více bodů. Můžeme také využít větší rozlišení, a tedy přesněji identifikovat míček v obraze. V ideálním případě by bylo vhodné použít velmi rychlé kamery, které by dokázaly sledovat i rotaci míčku. Během hledání vhodného typu pro vytvářený systém jsem kontaktoval zástupce některých společností zastupujících výrobce vysokorychlostních kamer a získal tak několik vhodných kandidátů.

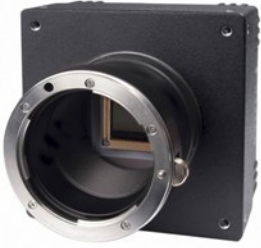
#### Basler A503k

Jedna z nejrychlejších kamer známého německého výrobce Basler. Vyšší model A504k umožňuje ještě vyšší rychlost 500 *fps* při rozlišení  $1280 \times 1024$ , ale je také o cca 30 tisíc korun dražší. Obě kamery podporují synchronizované ukládání a mají redukci na stativ. K provozu kamer je nutné zakoupit objektiv a kartu do PC pro připojení pomocí rozhraní CameraLink. Celý systém dvou kamer (2× kamera A503k, 2× CameraLink kabel, 2× objektiv Nikkor 35mm 1:1.8D a 2× BitFlow Karbon CL (PCI Express karta pro připojení rozhraní CameraLink)) by tak vyšel, dle cenového odhadu zástupce společnosti ELCOM, na necelých 504 000,- CZK bez DPH<sup>4</sup>, což je nepřijatelně mnoho.

#### AOS S-PRI

Vstupní model v segmentu vysokorychlostních kamer od švýcarského výrobce AOS Technologies. Zařízení je připojována pomocí rozhraní „Gigabit Ethernet“ a není tedy třeba

<sup>4</sup>cena byla stanovena 6.10.2010

Basler A503k F	
	Výrobce: Basler AG
	Zastoupení v ČR: ELCOM, a.s.
	Režimy snímání: 1280 × 1024 @ 402 <i>fps</i> , 1280 × 512 @ 804 <i>fps</i>
	Závit: F-mount (Nikon)
	Rozhraní: CameraLink
	Barevná: NE
	Synchronizace: ANO
	Cena (6.10.2010): 199 030,- CZK bez DPH

Tabulka 4.3: Přehled parametrů kamery Basler A503k

žádného speciálního vybavení pro její připojení k počítači. Kamera podporuje synchronizované nahrávání avšak modul není v základní ceně a je nutné jej dokoupit. Cena celého systému by jen o málo převyšovala cenu soustavy s Basler A503k, při vyšší rychlosti získaných kamer.

AOS S-PRI	
	Výrobce: AOS Technologies AG
	Zastoupení v ČR: Proxis, spol. s r.o.
	Režimy snímání: 800 × 600 @ 1250 <i>fps</i>
	Závit: C-mount
	Rozhraní: Gigabit Ethernet
	Barevná: NE
	Synchronizace: ANO
	Cena (7.10.2010): 250 000,- CZK bez DPH

Tabulka 4.4: Přehled parametrů kamery AOS S-PRI

## PlayStation Eye

Velmi levná kamera od společnosti Sony, vyvinutá pro platformu PlayStation Move, která je určená pro ovládání herní konzole PlayStation pohybem. Díky masovému rozšíření je cena této kamery velmi příznivá, avšak díky svému určení má samozřejmě řadu nedostatků. Jde především o nemožnost vyměnit objektiv či nemožnost synchronizovat nahrávání z více kamer. Dalším úzkým hrdlem je rozhraní USB 2.0, které se u tohoto řešení dostává na okraj svých možností (zvláště při připojení dvou kamer), na druhou stranu však opět není třeba žádného dodatečného vybavení. Rychlost kamery splňuje dříve definované minimální požadavky, ale pouze při rozlišení 320 × 240. Tuto kameru lze používat jak na platformě Windows<sup>5</sup>, tak i na platformě Linux, kde jsou ovladače přímo v jádře operačního systému. Cena snímacího systému je pak složena pouze z ceny dvou kamer a vychází na necelých 1 100,- CZK bez DPH.

<sup>5</sup> ovladače a vývojový „toolkit“ jsou k dispozici na adrese <http://codelaboratories.com/downloads/>

PlayStation Eye	
	Výrobce: Sony Computer Entertainment Inc.
	Zastoupení v ČR: Sony Europe Limited
	Režimy snímání: 640 × 480 @ 60 <i>fps</i> , 320 × 240 @ 125 <i>fps</i>
	Závit: bez vyměnitelného objektivu
	Rozhraní: USB 2.0
	Barevná: ANO
	Synchronizace: NE
Cena (19.5.2011): 585,- CZK bez DPH (T.S.Bohemia a.s.)	

Tabulka 4.5: Přehled parametrů kamery PlayStation Eye

### Zvolené řešení

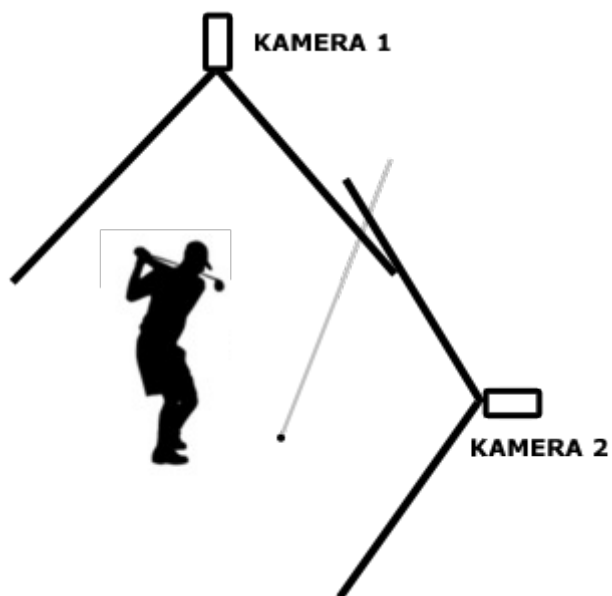
Z výše uvedených variant přicházejí v úvahu dvě alternativy. Ačkoliv by byl systém s kamerami AOS velmi drahý přinesl by velký náskok před konkurenčními řešeními, neboť by umožnil sledovat rotaci udělenou míčku, a tím by pokročilým golfistům přinesl možnost tvarovat rány. Druhou možnou cestou je snaha o co nejlevnější řešení, které přináší kamera PlayStation Eye za cenu větší nepřesnosti. Pro vývoj jsem se rozhodl zvolit levné řešení, které při jeho potenciálním komerčním úspěchu lze volbou lepších kamer zdokonalovat.

## 4.4 Rozmístění kamer

Pro implementaci systému bych navrhoval použití dvou kamer, které jsou nutné pro přesnou rekonstrukci bodu. Instalace více kamer může být vhodná pro případ, že by se jedna z kamer zastínila třeba samotným hráčem. Další kamera by také mohla mít příznivý vliv na zpřesnění rekonstrukce bodu. Na druhou stranu by třetí kamera (v případě dražších modelů) značně zvýšila cenu systému a zvýšila by nároky na hardware zpracovávající data. Po několika experimentech s umístěním kamer, jsem jako nejlepší variantu vyhodnotil rozmístění, kdy je jedna kamera situována kolmo k předpokládané dráze odpalu a druhá připevněná na stropě nad hráčem, který ji tedy nemůže, na rozdíl od umístění kamery za hráčem, nevhodným pohybem zakrýt. Situaci znázorňuje obrázek 4.8.

## 4.5 Volba algoritmů

Neboť jsem v kapitole 4.3 zvolil levnou variantu zařízení s rozlišením pouhých 320 × 240 obrazových bodů a kamery jsou rozmístěny téměř kolmo k sobě ve vzdálenosti cca 2 m od pozice hráče, je třeba se zamyslet nad volbou algoritmu pro kalibraci i jeho provedením. Protože má kalibrace velký vliv na chybovost systému, je vhodné použít co nejpřesnější algoritmus ze zmíněných v kapitole 2.5. Pozice kamer je konstantní a je tedy možné provést kalibraci pouze jednou po jeho sestavení. Pro tento případ je vhodná kalibrace pomocí tělesa známého tvaru (např. často používané šachovnice) popsaná v 2.5.1. Jelikož je však rozlišení kamer příliš malé, je pro zvýšení přesnosti vhodné zkalibrovat kamery nejprve samostatně pro vyšší rozlišení (např. 640 × 480), které však využijeme pouze pro kalibraci soustavy kamer a následně i pro rozlišení nižší (v tomto případě 320 × 240). Nyní můžeme provést kalibraci celé soustavy podle 3.3 a získat tak transformační matice pro převod bodu z jedné



Obrázek 4.8: Náčrt demonstrující použité rozmístění kamer.

kamery do druhé. Pro reálné použití si tedy můžeme zvolit, které rozlišení budeme používat a podle toho vybrat příslušné vnitřní matice kamer. Transformační matice soustavy zůstanou stejné pro libovolné rozlišení. Vždy je však nutné kalibrovat pomocí stejné šachovnice, neboť velikost jednoho čtverce se stanoví jako jednotka souřadného systému.

Pro rekonstrukci bodu v prostoru využijí triangulační algoritmus popsáný v kapitole [3.4.1](#). Kontrolu přesnosti získaného bodu lze provést pomocí zpětné projekce 3D pozice do obrazů obou kamer a jeho následným porovnáním s bodem předaným na vstup triangulační funkce.

## Kapitola 5

# Implementace a sestavení systému

### 5.1 Použité technologie

Snahou při vývoji bylo sestavit řešení, které nebude platformově závislé. S tím jsou však spojeny některé problémy způsobené především volbou konkrétních kamer a rozdílností implementace použitých knihoven. Jako implementační jazyk jsem zvolil C++ podle standardu z roku 1998. Celý systém je pak postaven na knihovně OpenCV, která zprostředkovává většinu potřebných algoritmů. Během vývoje vyšla verze 2.2.0, která přináší některá vylepšení a především pod operačním systémem Linux doplňuje část chybějících funkcí pro ovládání zvolených kamer (např. možnost nastavit rozlišení). Pro vývoj na platformě společnosti Microsoft jsem využíval operační systém Windows 7 64-bit a vývojové prostředí *Visual Studio 2010*. V OS Linux (jádro verze 2.6.38) jsem použil prostředí *Anjuta* a překladač *g++* z rodiny překladačů GNU Compiler Collection.

### 5.2 Problémy přinášející konkrétní platformy

#### 5.2.1 PlayStation Eye a Linux

Ovladač pro kamery PlayStationEye je přímo součástí jádra operačního systému Linux. Zde bych však doporučil verzi 2.6.37 a novější, která opravuje chybu předchozích jader. Bohužel knihovna OpenCV, ani v nejnovější verzi 2.2.0, neumožňuje změnu rychlosti snímání kamery pomocí funkce `cvSetCaptureProperty()` a poskytuje pouze přednastavenou hodnotu 30 snímků za vteřinu. Pro změnu rychlosti lze však změnit přednastavenou hodnotu v souboru `modules/highgui/src/cap_libv4l.cpp` zdrojových kódů knihovny OpenCV. Konkrétně se jedná o proměnnou `setfps.parm.capture.timeperframe.denominator` na řádce číslo 1446. Poté je samozřejmě nutné knihovnu znovu přeložit a nainstalovat obvyklým způsobem.

#### 5.2.2 OpenCV a Linux

Implementace knihovny OpenCV pro operační systém Linux však přináší i další problémy, zejména při snaze nahrát synchronizované video ze dvou kamer. Ačkoliv se jednalo o stejné kamery, a stejný program přeložený na Windows platformě pracoval správně, v Linuxu se snímky obou kamer diametrálně rozcházel, což bohužel vzhledem k požadavkům systému odsuzuje tuto platformu k nepoužitelnosti.

### 5.2.3 PlayStation Eye a Microsoft Windows

Pro operační systém společnosti Microsoft lze pro zvolené kamery získat vývojářský balíček, který kromě ovladačů umožňujících používání kamery PlayStationEye jako standardní webkamery, přináší knihovny pro vývoj a získávání snímků z těchto kamer. Společnost Code Laboratories, která za tímto balíčkem stojí dokonce na svých stránkách [14] uvádí, že je možné s pomocí jejich řešení získat z kamery až 187 snímků za vteřinu. Během vývoje se mi však nepodařilo pokořit ani hranici 135 fps. V balíčku jsou přiložené také ukázkové zdrojové soubory prezentující jak provázat rozhraní kamer s knihovnou OpenCV. Toto komfortní rozhraní, které poskytuje komplexní nastavení kamer od rozlišení, rychlosti přes jas, kontrast až po tzv. „gain“ ovlivňující množství šumu v obraze, je však bohužel platformově závislé. Pokud bychom lpěli na multiplatformnosti bylo by třeba rozhraní pro získání obrazu abstrahovat s pomocí některého návrhového vzoru (např. „Abstract factory“) a direktiv překladače, neboť implementace OpenCV pro Windows bohužel neumožňuje získání snímků z této kamery standardními knihovními funkcemi.

## 5.3 Implementace aplikace

Celý systém jsem rozložil do více podpůrných programů. Ty by bylo vhodné v budoucnosti spojit do jedné ovládací aplikace, která by jej spravovala.

### 5.3.1 OneCamCalib

Jde o jednoduchý program využívající funkci `cvCalibrateCamera2()` z knihovny OpenCV. Po získání předem zvoleného počtu snímků šachovnice se kamera zkalibruje a výsledek je uložen ve formě XML souborů.

### 5.3.2 TwoCamsImageSave

Velmi jednoduchá aplikace, která ukládá snímky z dvou kamer současně. Během zobrazování snímků v nich hledá kalibrační šachovnici o zvolených rozměrech. Po stisku klávesy 's' uloží do aktuálního adresáře obrázku z obou kamer odlišené názvem.

### 5.3.3 StereoCalib

Tento program, jak jeho název napovídá, obstarává kalibraci dvou kamer. Vstupem je série souhlasných snímků z obou zařízení obsahující kalibrační šachovnici (získanou například programem *TwoCamsImageSave*) a XML soubory s vnitřními parametry kamer a jejich distorzními koeficienty. Jádrem programu je funkce `cvStereoCalibrate()`, která implementuje samotnou stereo kalibraci. Tato funkce umí spočítat i vnitřní parametry a distorzní koeficienty, avšak samostatné zkalibrování dává mnohem lepší výsledky především distorzních koeficientů. Proto je funkci nastaven příznak `CV_CALIB_FIX_INTRINSIC`, a na její vstup jsou předány načtené matice získané programem *OneCamCalib*.

### 5.3.4 GetPosition

Po získání údajů o vzájemné pozici kamer, již můžeme využívat triangulaci, což zprostředkovává tento program. Jeho vstupem jsou synchronizovaná videa a XML soubory s maticemi reprezentujícími transformace mezi kamerami a vnitřními parametry obou kamer. Z matic

reprezentujících vlastnosti kamer jsou následně seskládány jejich projekční matice podle vztahu 2.10. Ty jsou dále předány společně se souhlasnými body, označenými v obraze výběrem myši, funkci `cvTriangulatePoints()`. Výstupem jsou rekonstruované body převedené do světových souřadnic (počátek souřadného systému je v místě kamery označené jako první).

### 5.3.5 TwoCamsVideoSave

Pro ukládání videosekvencí z kamer PlayStation Eye je z důvodu zmiňovaných v kapitole 5.2 použito rozhraní *CL Eye Platform* společnosti *Code Laboratories* a aplikace je tedy určena pouze pro Microsoft Windows. Jelikož požadavky na rychlost ukládání jednotlivých snímků jsou vysoké, pracuje program ve dvou průchodech, nejdříve jsou obě videa nahrána do paměti, a až následně jsou uložena na disk v nekomprimované podobě pomocí funkcí z knihovny OpenCV.

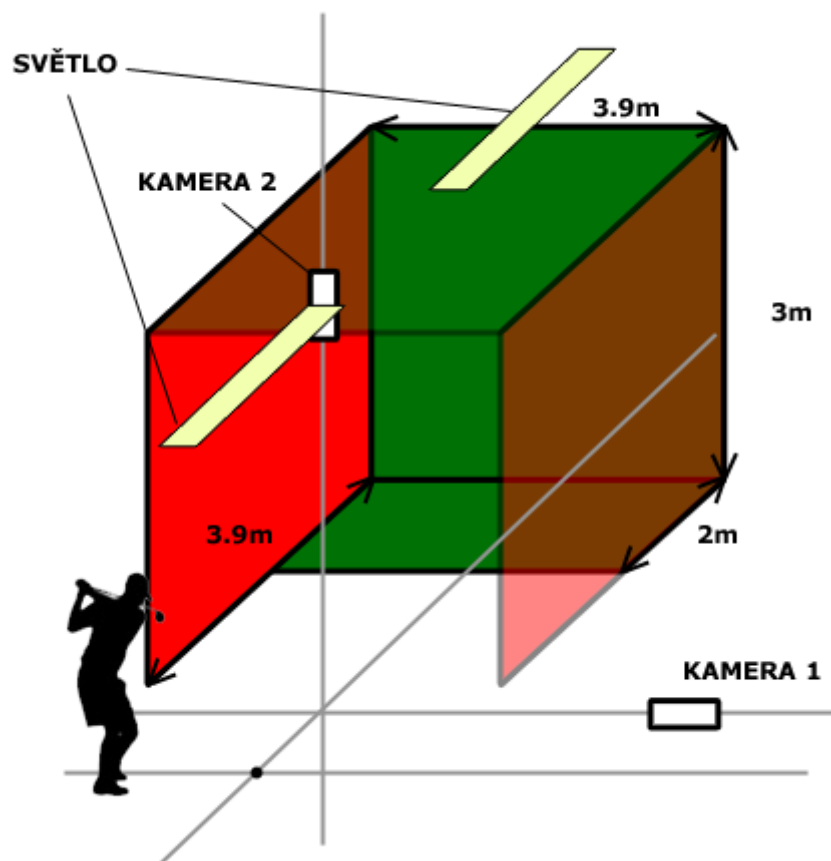
### 5.3.6 GolfSim

Aplikace agregující všechny dosud implementované moduly systému (vyjma výše uvedených nástrojů pro kalibraci). Zejména spojuje modul pro rozpoznávání míčku v obraze, získání jeho pozice v trojrozměrném prostoru a následně fyzikální rekonstrukci předpokládané dráhy letu míčku.

Jedná se o konzolovou aplikaci, které jsou při spuštění předány cesty ke dvěma videosekvencím. Po spuštění se vytvoří instance třídy `Application` a zavolá se její metoda `Run()`, které jsou předány soubory obsahující video. Soubory jsou pomocí instance třídy `Video` načteny a uloženy po jednotlivých snímcích do kontejneru `vector<>` ze standardní šablonové knihovny jazyka C++. Načtení celého videa do paměti probíhá z důvodů budoucí snadné integrace aplikace *TwoCamsVideoSave*. Nahrané video se předá instanci třídy `Analyzer`, která zpracovává připravené snímky. Nejdříve inicializuje objekt třídy `Stereo`, který si načte potřebné matice (vnitřní parametry kamery, distorzní koeficienty a transformace mezi kamerami) a následně začne zpracovávat video po jednotlivých dvojicích snímků. Ty jsou předány objektu třídy `Detector`, která v nich hledá golfový míček a vrací strukturu obsahující polohu nalezené stopy. Výstupní data jsou předány metodě `GetPosition()` třídy `Stereo`, která vrátí rekonstruovaný bod v 3D prostoru. Jelikož máme k dispozici projekční matice obou kamer, můžeme se pokusit provést zpětnou projekci bodu do obrazu, což provádí metoda `Stereo::ProjectPoint()`. Pokud je vzdálenost zpětně získaného bodu od toho původního větší než 25 obrazových bodů, není bod dále zpracováván. Díky tomu vyčistím výstupní data o body, při nichž selhal vyhledávací algoritmus (rozpoznán v jednom obraze špatný bod nebo se špatně ořezala stopa míčku). Výsledné body jsou předány objektu třídy `Physics`, který vypočítá potenciální trajektorii míčku. Třídy `Physics` a `Detector` jsou implementovány v rámci práce [7].

## 5.4 Sestavení prototypu systému

Pro potřeby vývoje a testování implementovaných částí jsem sestavil prototyp simulátoru. Jelikož modul pro grafický výstup zatím není vyvinutý, sestavil jsem pouze běžné golfové odpaliště, které je znázorněno na obrázku 5.1 a míčky odpaloval do golfové sítě. Pro osvětlení místnosti byly použity dvě standardní zářivky umístěné uprostřed stropu paralelně se



Obrázek 5.1: Náčrt prototypu simulátoru použitého pro vývoj.

směrem odpalu<sup>1</sup>. Síť a zem bylo třeba ze dvou stran zakrýt černou plachtou jak ukazuje obrázek 5.3, což velmi zvýšilo úspěšnost vyhledávání míčku v obraze.

Zvolil jsem kamery PlayStation Eye (viz 4.3) a rozmístil je jak bylo popsáno v kapitole 4.4 (viz obrázek 5.2). Kamery jsem zkalibroval nejdříve jednotlivě s pomocí aplikace *One-CamCalib* pro režimy  $320 \times 240@125$  a  $640 \times 480@30$  a následně celou soustavu pomocí *StereoCalib* pro režim  $640 \times 480@30$  z důvodu popsaného v kapitole 4.5. Pro kalibraci jsem použil šachovnice o rozměrech  $7 \times 6$  s velikostí pole  $15 \times 15$  cm.

Nyní můžeme z kamer získat záznam odpalu například pomocí programu *TwoCamsVideoSave*, případně libovolného jiného a videosekvence předat aplikaci *GolfSim* na vstup. Ten zpracuje snímky a provede rekonstrukci odpalu do prostoru jak ukazují obrázky 5.4 a 5.5.

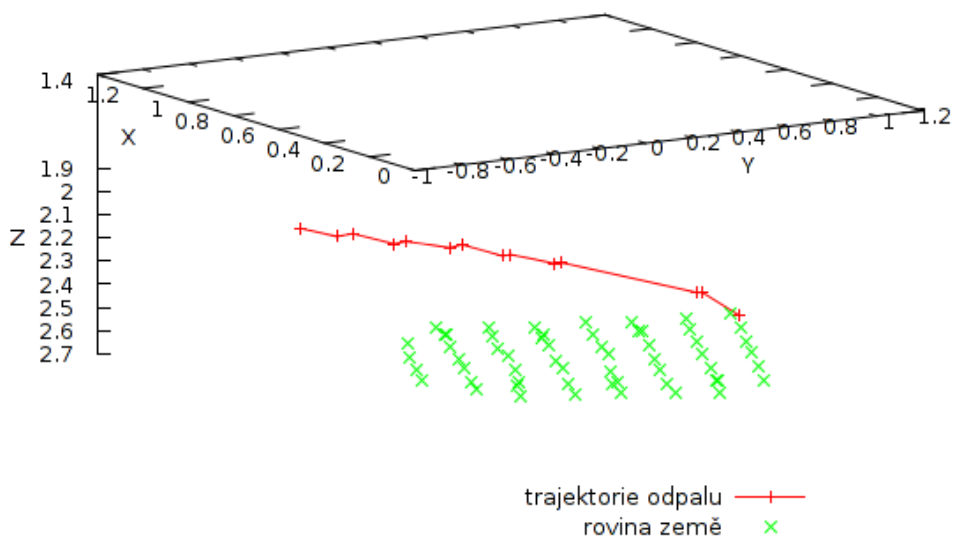
<sup>1</sup>s předpokladem, že hráč se snaží zahrát rovnou ránu



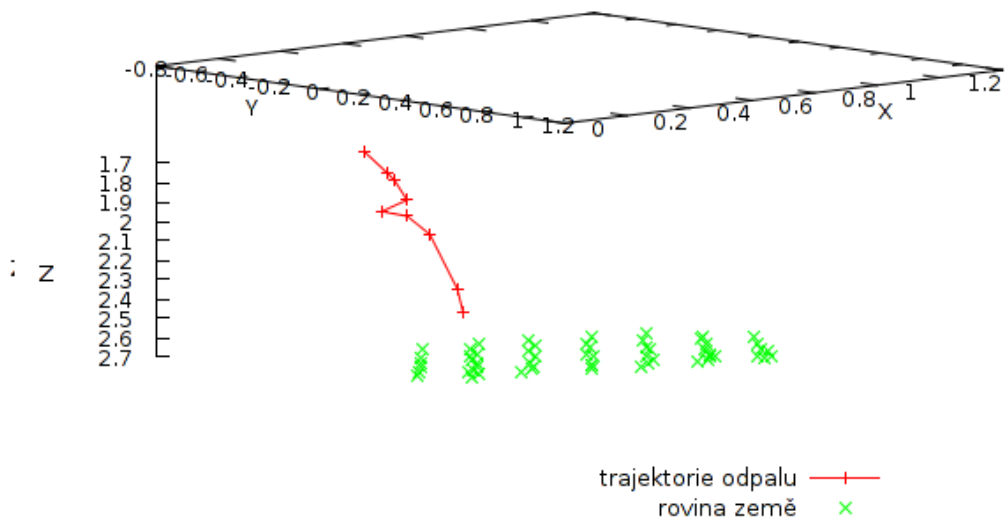
Obrázek 5.2: Fotografie demonstrující rozmístění kamer v prototypu simulátoru. Kamery jsou zvýrazněny modrým kolečkem.



Obrázek 5.3: Fotografie zachycující sestavený prototyp simulátoru.



Obrázek 5.4: Ukázka výpočtu pozice míčku během odpalu.



Obrázek 5.5: Ukázka výpočtu pozice míčku během odpalu.

## Kapitola 6

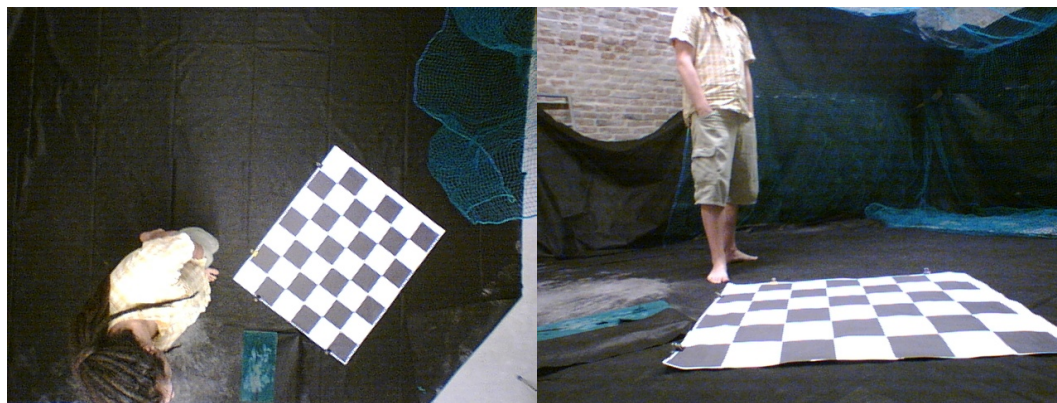
# Zhodnocení výsledného řešení

### 6.1 Přesnost kalibrace a rekonstrukce pozice statických objektů

Přesnost získaných matic lze ověřit jednoduchým testem. Do scény obou kamer vložíme předmět o známých rozměrech. Označíme v obou obrazech body, jenž předmět vymezují a vypočítáme jejich vzdálenost. Porovnáním velikosti získáme odhad přesnosti systému.

#### Kalibrační šachovnice

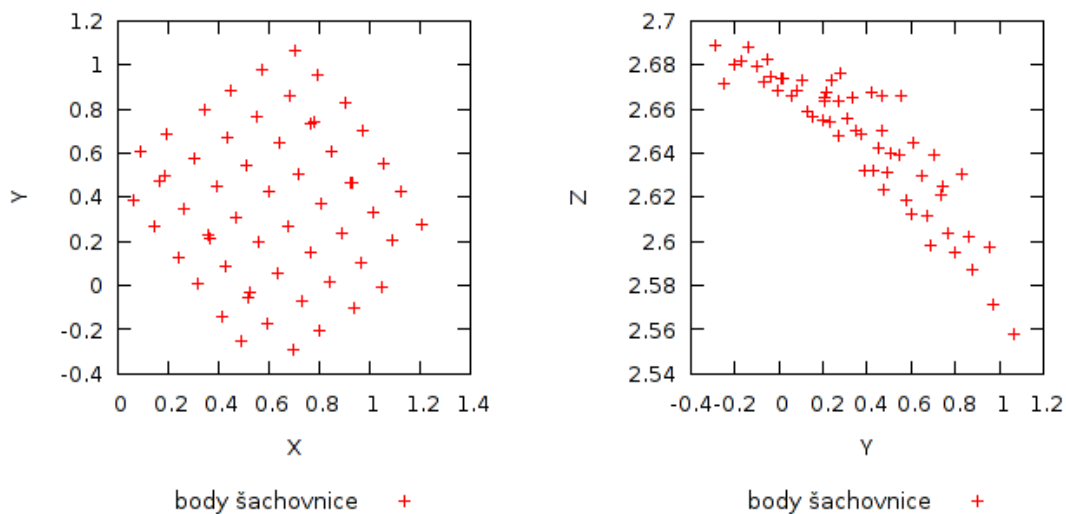
Pro první test jsem zvolil jako předmět kalibrační šachovnici s velikostí čtverce  $15 \times 15 \text{ cm}$ , kterou jsem položil na zem tak, aby byla v záběru obou kamer, jak ukazuje obrázek 6.1. Nyní jsem označil hranice všech polí a provedl zpětný výpočet pozice bodů v 3D prostoru. Celý test probíhal s kamerami rozmístěnými podle uvedeného (viz kapitola 4.4) schématu v režimu  $640 \times 480@30$ .



Obrázek 6.1: Snímky kalibrační šachovnice z obou kamer.

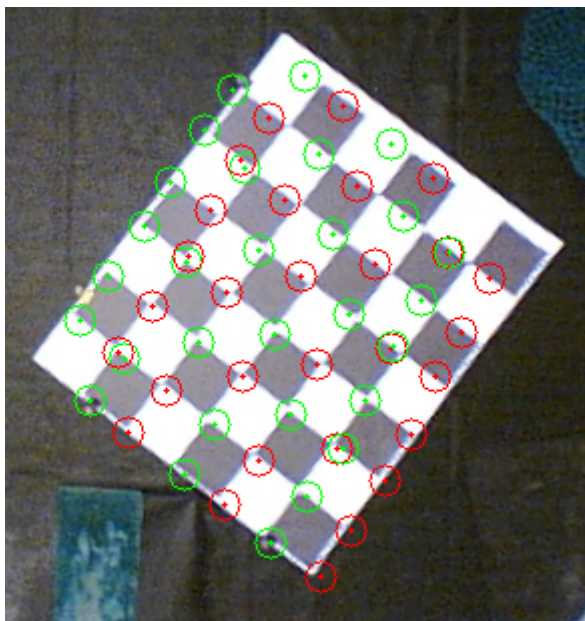
Test jsem provedl dvakrát. Poprvé jsem označoval body podél *osy y* (vertikálně) a následně podél *osy x*. Průměrná velikost pole získaná všemi měřeními byla  $14,50 \times 14,39 \text{ cm}$ , což znamená průměrnou chybu  $0,5 \times 0,61 \text{ cm}$ . Rekonstruované body teoreticky tvoří rovinu, která však není rovnoběžná se zemí neboť počátek soustavy je umístěn v pozici horní

kamery. Ta však není nasměrována přesně kolmo k zemi. Výsledek testu je znázorněn obrázkem 6.2.



Obrázek 6.2: Body šachovnice získané rekonstrukci do 3D.

Přesnost rekonstrukce klesala v okrajích snímku jak dokládá obrázek 6.3, který ukazuje výsledek zpětné projekce rekonstruovaných bodů do obrazu.



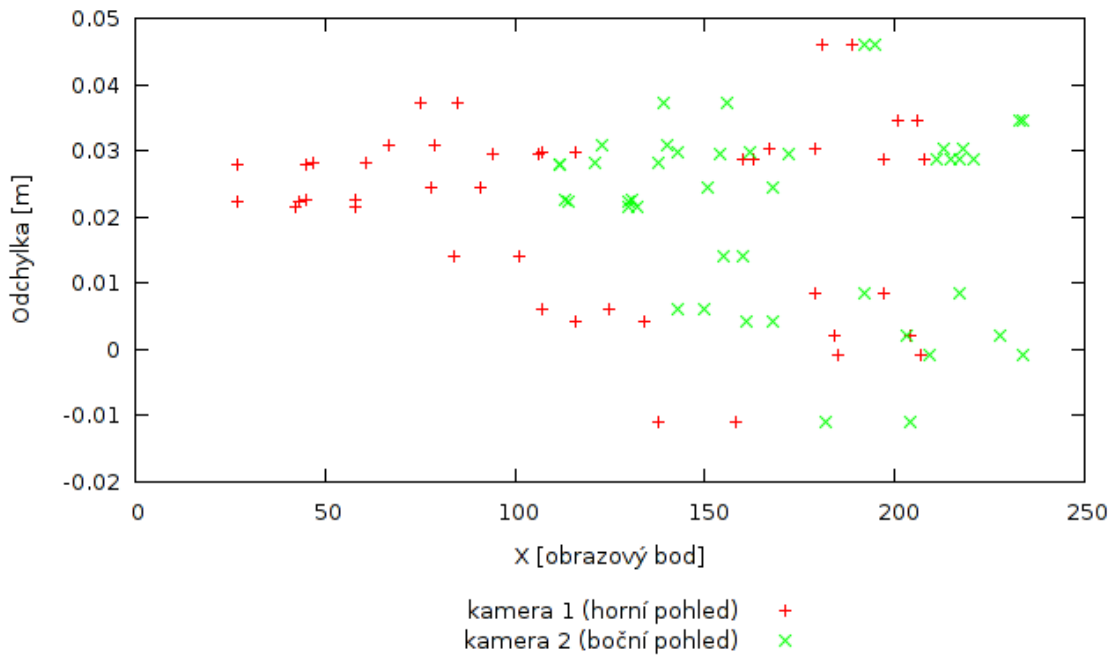
Obrázek 6.3: Výřez snímku kalibrační šachovnice se znázorněnými body, které byly získány zpětnou projekcí.

Vzniklou odchylku může ovlivnit několik faktorů. Označování bodů bylo prováděno manuálně a mohlo tedy dojít k drobným nepřesnostem v rozsahu několika pixelů. Kalibrační

matice byla vytisknuta na papíře, který byl přiložen na karton. I tak však docházelo v některých místech k jeho zvlnění. Největší odchylka je však samozřejmě způsobena přesností kalibrace. Vzhledem k vzdálenosti objektu od kamer a jejich technickým parametrům, je však dle mého názoru uvedená odchylka v přijatelných mezích.

### Čtverec o známých rozměrech

Druhý test jsem provedl s pomocí černého čtverce, o velikosti  $20 \times 20 \text{ cm}$ , vytištěného na papír a připevněného na tvrdé desky. Ty byly posouvány ve scéně a v různých pozicích byla opět prováděna rekonstrukce okrajových bodů do 3D a měření jejich vzdálenosti. Test jsem provedl pro obě rozlišení, které kamery nabízejí.



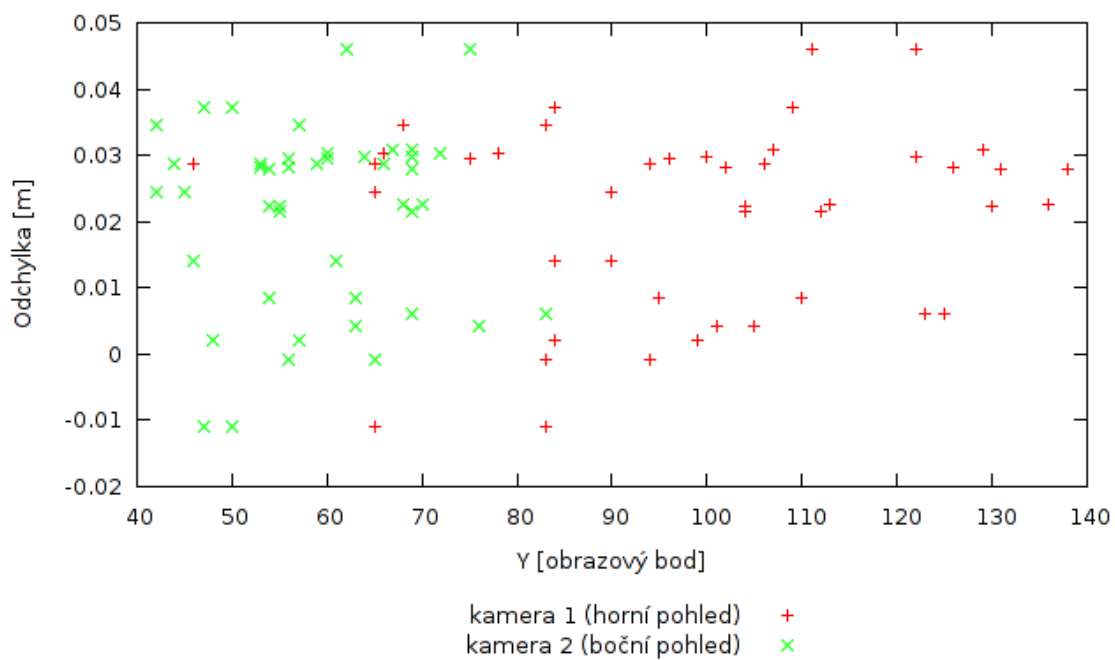
Obrázek 6.4: Vliv horizontální pozice bodu ve snímku na přesnost rekonstrukce pro rozlišení  $320 \times 240$ .

Z grafů vyplývá, že umístění bodu v obraze nemá na přesnost rekonstrukce vliv. Co je však z grafů patrné je přesnost měření pro různá rozlišení. Vzdálenost bodů pro rozlišení  $320 \times 240$  (viz obrázky 6.4 a 6.5, při 22 vzorcích, má průměrnou odchylku  $2,12 \text{ cm}$ , což je poměrně hodně. Průměrná odchylka pro rozlišení  $640 \times 480$  (viz obrázky 6.6 a 6.7), vypočítaná z 19 bodů, je však pouze  $0,91 \text{ cm}$ .

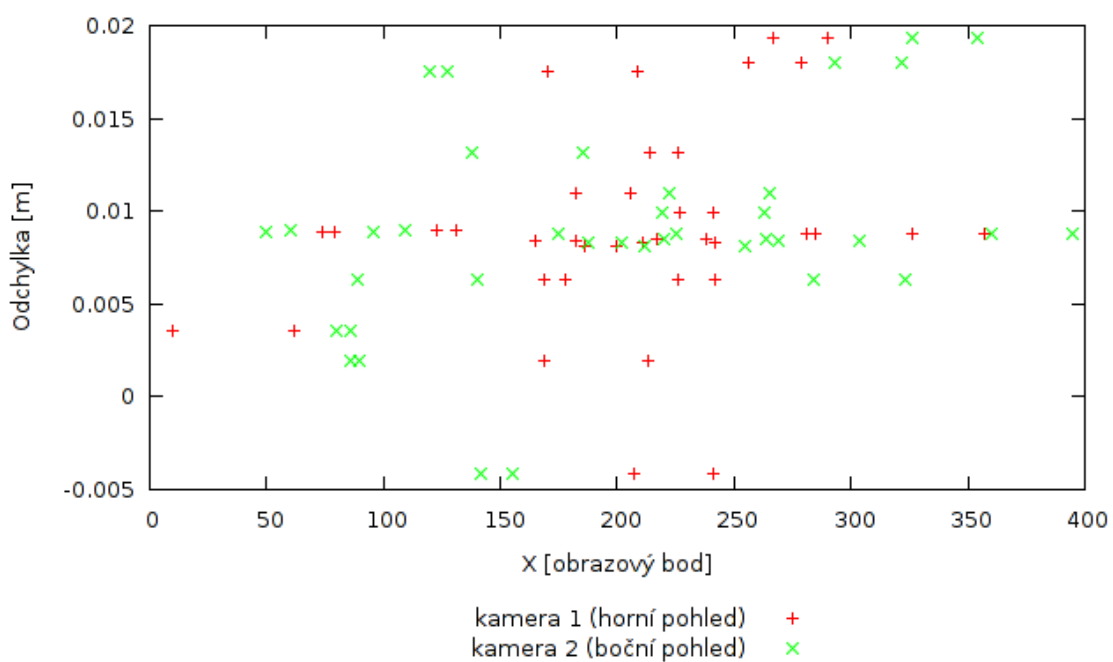
## 6.2 Přesnost rekonstrukce odpalu golfového míčku

Nejdůležitějšími parametry odpalu jsou vertikální a horizontální úhly. Test přesnosti měření provedu pomocí projekce získaných 3D bodů zpět do obrazu obou kamer a porovnáním jejich úhlu s úhlem získaným přímo z videa (viz obrázek 6.8).

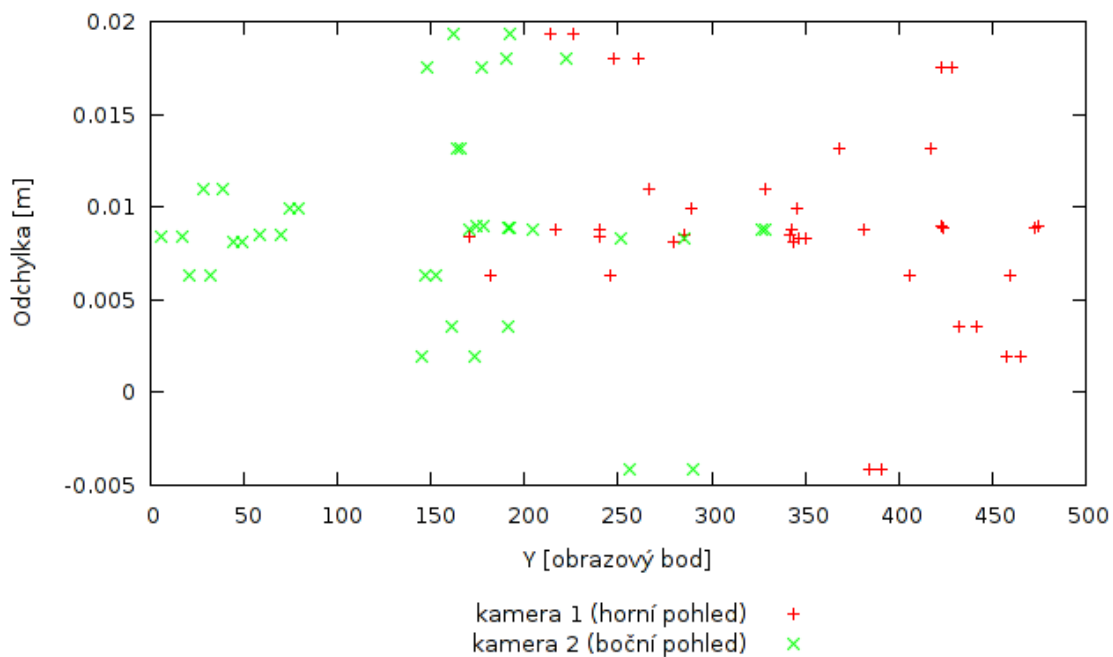
Hodnota úhlů získaných z videa není konstantní, neboť stopa je v každém snímku vlivem šumu a nízkého rozlišení jinak rozpoznána. To však nemá vliv na rekonstrukci, neboť jí



Obrázek 6.5: Vliv vertikální pozice bodu ve snímku na přesnost rekonstrukce pro rozlišení  $320 \times 240$ .



Obrázek 6.6: Vliv horizontální pozice bodu ve snímku na přesnost rekonstrukce pro rozlišení  $640 \times 480$ .

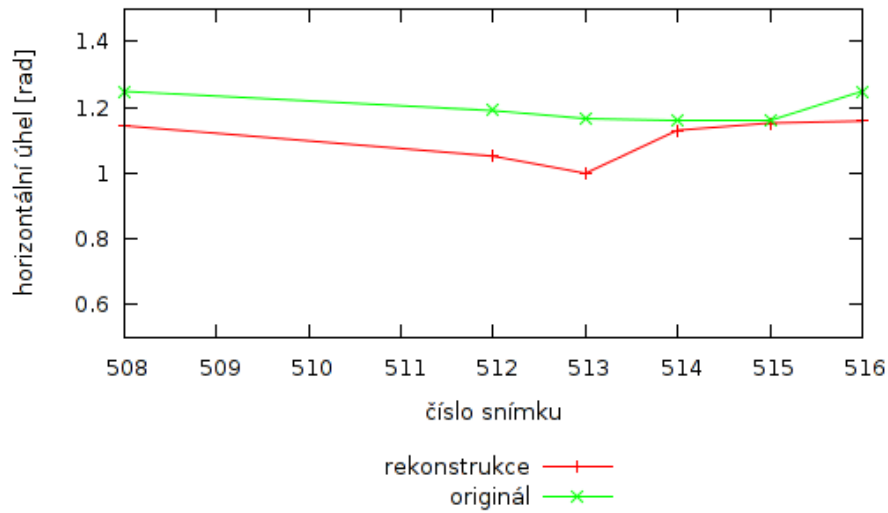


Obrázek 6.7: Vliv vertikální pozice bodu ve snímku na přesnost rekonstrukce pro rozlišení  $640 \times 480$ .

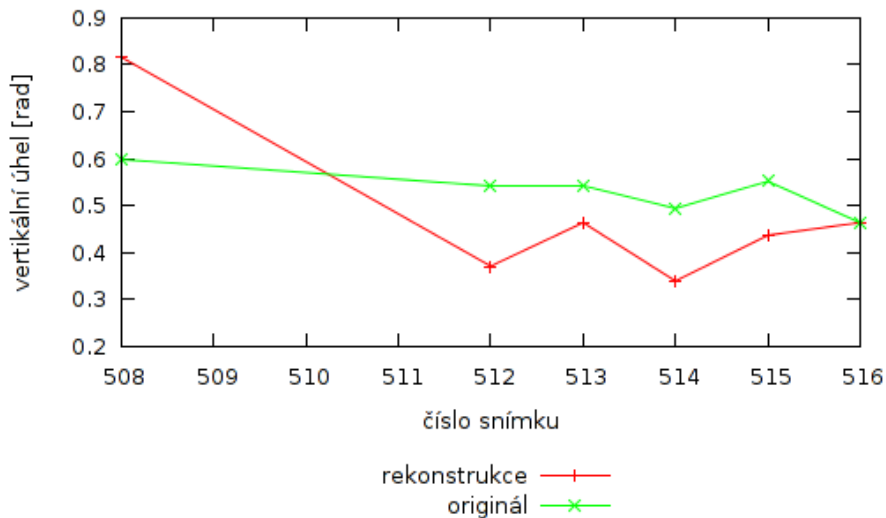


Obrázek 6.8: Ukázka srovnání úhlů ve videu.

jsou předány krajní body stopy míčku, a výpočet pro každý snímek je tedy autonomní. Průměrná horizontální odchylka je v případě, který reprezentuje obrázek 6.9,  $-0,0890 \text{ rad}$ , což je  $-5,01^\circ$ . Chyba rozdílu vertikálního úhlu je v průměru  $-0,0498 \text{ rad}$  ( $-2,58^\circ$ ), jak ukazuje obrázek 6.10. Odchylka systému je však ve skutečnosti menší, neboť projekční matice zatížené chybou (viz předchozí kapitola 6.1) byly pro test použity jak pro získání pozice bodu v prostoru, tak pro jeho zpětnou projekci do obrazu kamer.



Obrázek 6.9: Srovnání horizontálního úhlu získaného rekonstrukcí proti úhlu získanému z videa



Obrázek 6.10: Srovnání vertikálního úhlu získaného rekonstrukcí proti úhlu získanému z videa

### 6.3 Zhodnocení systému vzhledem k požadovaným vlastnostem

Porovnejme vlastnosti sestaveného systému s požadavky uvedenými v kapitole 4.2. Body jsou do prostoru rekonstruovány v průměru s centimetrovou odchylkou. Podle výše odkazované kapitoly, taková chyba znamená pro ránu na 150 m, chybu asi 6 – 8 m. Bohužel další odchylku vytváří použitý fyzikální modul, jehož přesnost je podle [7] 5 – 15 m.

Dále byla požadována doba zpracování nejlépe do 3 sekund po odpalu. Tento požadavek

je bez problémů splněn. Aplikace je schopna zpracovat 1300 dvojic snímků za 41 vteřin. Při snímání kamerou s rychlostí 125 snímků za vteřinu je pro odpal potřeba zpracovat pouze 10 – 20 snímků.

## 6.4 Možná rozšíření

Vývoj golfového simulátoru nabízí velmi mnoho možností pro zlepšení, neboť sestavit dokonale přesné zařízení je téměř nemožné. Vyberu proto ty, která mi připadají zásadní, a které v budoucnosti plánuji do projektu integrovat.

### Synchronizace kamer

Během vývoje jsme společně s autorem [7] diskutovali možnost použít vyšší rozlišení, které kamery nabízejí za cenu jejich nižší rychlosti. Proto jsme některé testy prováděli v režimu  $640 \times 480@60$ . U takto pořízených videosekvencí se však velmi projevila nesynchronizovanost kamer, a proto byly výsledky nepoužitelné. Navíc v tomto režimu byly stopy méně zřetelné a mnohem delší, což je přirozeným důsledkem delší expozice. Ačkoliv při vyšší rychlosti je chyba způsobená špatným synchronizováním kamer mnohem menší stále zvyšuje nepřesnost, a proto by bylo vhodné ji odstranit. Způsobů jak toho docílit je několik. Nejelegantnějším by bylo kamery synchronizovat pomocí hardware, což většina dražších modelů podporuje. Další alternativou je využít časových razítek, která by se dala přidat do datové struktury případně zakódovat přímo do videa.

### Změna použitých kamer a rozhraní

Navržený systém je velmi limitovaný zvolenými kamerami. Výhodou celého řešení je však nezávislost na použitých kamerách a jejich relativně snadná výměna za výkonnější. Ideální by byly kamery s rychlostí kolem 2000 snímků za vteřinu při rozlišení  $800 \times 600$ . Otázkou zůstává do jaké míry by bylo vhodné použít například kamery v cenové hladině desítek tisíc korun, kdy nezískáme vyšší rychlost, ale budeme moci použít kvalitnější objektiv případně funkce pro synchronizaci apod. Dražší kamery navíc využívají vhodnějších rozhraní pro přenos dat jako Gigabit Ethernet, IEEE 1394 nebo CameraLink.

### Osvětlení

Při použití vysokorychlostních kamer, které mají krátkou expoziční dobu, je množství světla limitujícím faktorem systému. Přílišné osvětlení je nemožné z důvodu přítomnosti projektoru. Situaci by bylo možno řešit osvětlením mimo viditelné spektrum. PlayStation Eye však naneštěstí mají filtr, který odstiňuje infračervené světlo. Kamery lze však rozebrat a filtr odstranit, jak je popsáno v [15].

### Implementace v hardware

Pokud by se použilo kamer s vyšší rychlostí nebo vyšším rozlišením bylo by potřeba urychlit prováděné algoritmy. Jednou z variant by bylo implementovat některé algoritmy do hardware, který by předzpracovával data pro finální výpočet prováděný v počítači. Další alternativou by mohlo být akcelerování některých algoritmů pomocí GPGPU<sup>1</sup>.

<sup>1</sup>z anglického „General-purpose computing on graphics processing units“ - využití grafických karet pro obecné výpočty

# Kapitola 7

## Závěr

Cílem práce bylo rekonstruovat pozici golfového míčku ze soustavy dvou případně více kamer. Úvodní kapitola seznamuje s modely kamer a problémy, které při jejich používání vznikají. Následuje kapitola o teorii geometrie v systému více kamer, která taktéž prezentuje vybrané algoritmy pro kalibraci takového systému a způsob rekonstrukce bodu v něm. Kapitola 4 se zabývá analýzou požadavků a návrhem systému. V rámci této části jsou zvoleny kamery, jejich rozmístění a algoritmy pro kalibraci a určení pozice objektu v prostoru. Další část práce je věnována sestavení systému a implementaci demonstrační aplikace. Tato kapitola také diskutuje problémy, se zvolenými kamerami, které se během vývoje vyskytly a nabízí jejich řešení. Poslední kapitola zhodnocuje přesnost systému na reálných datech a nabízí možná rozšíření.

V rámci této práce byl sestaven prototyp golfového simulátoru, který umožňuje rekonstruovat odehranou ránu a předat potřebná data souběžně implementovanému modulu zabývající se fyzikálním modelem. V porovnání s konkurenčními řešeními je navržený systém velmi modulární a jeho výkonnost a především cenu určuje model zvolených kamer. Při využití nejlevnějších kamer se cena navrženého systému pohybuje v řádu desítek tisíc korun. S těmito kamerami modul pro rekonstrukci bodu může pracovat s odchylkou přibližně  $2,5 \text{ cm}$ . Pro výrazné zlepšení vlastností systému by bylo vhodné zvolit kamery nabízející rozlišení alespoň  $640 \times 480$ . Při použití velmi rychlých kamer by bylo možné sledovat i rotaci míčku, avšak takové kamery by přinesly další problémy, s nimiž by bylo třeba se vypořádat.

# Literatura

- [1] Bartsch, H.-J.: *Matematické vzorce*. Mladá fronta, 2002.
- [2] Bradski, G.; Kaehler, A.: *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 2008, iISBN 978-0-596-51613-0.
- [3] Brown, D. C.: Close-Range Camera Calibration. *Photogrammetric Engineering*, , č. 37, 1971: s. 856 – 866.
- [4] Cyganek, B.; Siebert, J. P.: *An Introduction to 3D Computer Vision Techniques and Algorithms*. John Wiley & Sons, 2009, iISBN 978-0-470-01704-3.
- [5] Halliday, D.; Resnick, R.; Walker, J.: *Fyzika - Mechanika*. VUTIUM, 2000.
- [6] Hartley, R.; Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2006, iISBN 0-521-54051-8.
- [7] Hlobil, M.: *Simulace reálné trajektorie golfového míčku z videa*. Diplomová práce, FIT VUT v Brně, 2011.
- [8] Ma, Y.; Soatto, S.; Košecká, J.; aj.: *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003, iISBN 978-0-387-00893-6.
- [9] Stein, G. P.: Accurate internal camera calibration using rotation, with analysis of sources of error. In *Proceedings of the Fifth International Conference on Computer Vision, ICCV '95*, IEEE Computer Society, 1995, ISBN 0-8186-7042-8, s. 230–.
- [10] Sturm, P.; Trigs, B.: A Factorization Based Algorithm for Multi-Image Projective Structure and Motion. *4th European Conference on Computer Vision*, 1996: s. 709–720.
- [11] Trucco, E.; Verri, A.: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.
- [12] WWW stránky: aboutGolf and the 3Trak Launch Monitor.  
[http://thesandtrap.com/b/throwing\\_darts/aboutgolf\\_and\\_the\\_3trak\\_launch\\_monitor](http://thesandtrap.com/b/throwing_darts/aboutgolf_and_the_3trak_launch_monitor).
- [13] WWW stránky: Co je to hendikep a jak se počítá.  
<http://www.golf-score.cz/golf/hendikep/>.
- [14] WWW stránky: Code Laboratories: About Eye Platform.  
<http://codelaboratories.com/products/eye/sdk/visual/>.

- [15] WWW stránky: Code Laboratories: PS3Eye Disassembly & IR Filter.  
<http://codelaboratories.com/research/view/ps3-eye-disassembly>.
- [16] WWW stránky: Full Swing Golf. <http://www.fullswinggolf.com/>.
- [17] WWW stránky: GEAR PATROL: aboutGolf PGA Tour Simulators.  
<http://gearpatrol.com/blog/2009/07/06/aboutgolf-simulators/>.
- [18] WWW stránky: Golf Ball - Wikipedia the free encyclopedia.  
[http://en.wikipedia.org/wiki/Golf\\_ball](http://en.wikipedia.org/wiki/Golf_ball).
- [19] WWW stránky: PGA Tour Statistics.  
<http://www.pgatour.com/r/stats/filter/?4>.
- [20] WWW stránky: Pinhole camera - Photographs, Information, PinholeDesigner.  
<http://www.pinhole.cz/>.
- [21] Zhang, Z.: A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 22, č. 11, 2000: s. 1330 – 1334.
- [22] Zupančič, T.; Jaklič, A.: *Automatic Golf Ball Trajectory Reconstruction and Visualization*. University of Ljubljana.
- [23] Šolony, M.: *Lokalizace objektů v prostoru*. Diplomová práce, FIT VUT v Brně, 2009.

# Příloha A

## Obsah CD

- Návrh plakátu
- Technická zpráva v elektronické podobě
- Videoprezentace vytvořeného systému
- Vzorová data pro test aplikace
- Zdrojové kódy implementovaných aplikací

**Příloha B**

**Plakát**