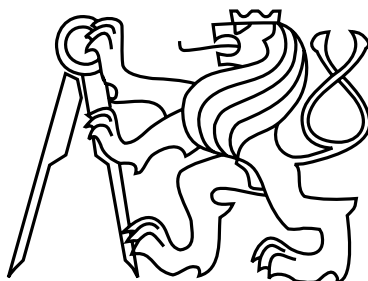


# Na tomto místě bude oficiální zadání vaší práce

- Toto zadání je podepsané děkanem a vedoucím katedry,
- musíte si ho vyzvednout na studijním oddělení Katedry počítačů na Karlově náměstí,
- v jedné odevzdané práci bude originál tohoto zadání (originál zůstává po obhajobě na katedře),
- ve druhé bude na stejném místě neověřená kopie tohoto dokumentu (tato se vám vrátí po obhajobě).



České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačové grafiky



Diplomová práce

## **Snímání pohybu pomocí systému Kinect**

*Bc. Petr Stolař*

Vedoucí práce: Ing. Jan Buriánek

Studijní program: Otevřená informatika, strukturovaný, Navazující magisterský

Obor: Počítačová grafika a interakce

26. prosince 2012



## Poděkování

Chtěl bych především poděkovat svému vedoucímu práce Ing. Janu Buriánkovi za nabídnutí možnosti zpracování tohoto tématu, za jeho konstruktivní kritiku, nápady, rady a připomínky. Dále bych chtěl poděkovat své přítelkyni a rodině za jejich podporu. Také bych chtěl poděkovat rodině své přítelkyně za zapůjčení zrcadla pro účely experimentu a tímto se jim ještě jednou omlouvám za jeho rozbití. Dále bych chtěl poděkovat všem osobám, které se účastnily experimentu a pomohly mi tak se stanovením zdravé báze cyklu chůze.



## Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 26. prosince 2012

.....





# Abstract

This thesis introduces the technology of motion capture, which is used for example in entertainment industry (movies, games, etc.), but also in medical practice, army and robotics. Motion capture is used in medicine in gait laboratories for gait analysis. Gait analysis is the only one tool that allows to understand complex kinetic and kinematic processes occurring during the gait cycle. The thesis describes the problem of motion capture for purposes of the study of human movement with application in medicine. Result of this work is the design and implementation of the experimental system which imitates gait laboratory system. Implemented system records motions through one or more Microsoft Kinect devices and visualizes recorded gait cycle.

# Abstrakt

Práce seznamuje s technologiemi snímání pohybu, které se využívají například v zábavním průmyslu (filmy, hry, atd.), ale také v medicínské praxi, v armádě či robotice. V medicíně se technologie snímání pohybu využívá v laboratořích chůze k analýze chůze. Analýza chůze je jediný nástroj, který umožňuje pochopit složité kinetické a kinematické procesy vznikající při cyklu chůze. Práce popisuje problematiku snímání pohybu pro účely studie pohybu člověka s aplikací v lékařství. Výsledkem práce je návrh a realizace experimentálního systému napodobujícího systém laboratoře chůze. Realizovaný systém snímá pohyb prostřednictvím jednoho nebo více zařízení Microsoft Kinect a zaznamenaný cyklus chůze vizualizuje.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Zadání . . . . .	1
1.2	Motivace . . . . .	1
1.3	Struktura práce . . . . .	2
<b>2</b>	<b>Analýza</b>	<b>3</b>
2.1	Snímání pohybu . . . . .	3
2.1.1	Uplatnění technologie snímání pohybu . . . . .	3
2.1.2	Metody snímání pohybu . . . . .	4
2.1.3	Produkční řetězec . . . . .	7
2.2	Analýza chůze . . . . .	10
2.2.1	Kinematické proměnné . . . . .	11
2.3	Pohybový snímač Microsoft Kinect . . . . .	14
2.3.1	Jak zařízení funguje . . . . .	15
2.3.2	Kinect jako 3D měřicí zařízení . . . . .	15
2.3.3	Markerless snímání pohybu . . . . .	17
2.3.4	Dostupné technologie a knihovny . . . . .	21
<b>3</b>	<b>Návrh systému</b>	<b>25</b>
3.1	Komponenty systému . . . . .	26
3.1.1	VRPN Server . . . . .	27
3.1.2	Kinect Recorder . . . . .	29
3.1.3	Gait Analyzer . . . . .	33
3.2	Snímání pomocí Kinectu . . . . .	38
3.2.1	Kalibrace . . . . .	38
3.2.2	Dynamický záznam . . . . .	40
3.2.3	Analýza výsledku . . . . .	43
3.3	Struktura dat . . . . .	44
3.3.1	Data posílaná VRPN serverem do sítě . . . . .	44
3.3.2	C3D data . . . . .	45
3.3.3	Graf cyklu chůze . . . . .	48
<b>4</b>	<b>Experiment</b>	<b>51</b>
4.1	Návrh experimentu . . . . .	51
4.2	Naměřená data . . . . .	52

4.3	Vyhodnocení . . . . .	54
<b>5</b>	<b>Závěr</b>	<b>59</b>
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>65</b>
<b>B</b>	<b>Obrazová příloha</b>	<b>67</b>
<b>C</b>	<b>Obsah přiloženého CD</b>	<b>79</b>

# Seznam obrázků

2.1	Eadweard Muybridge: Kůň v pohybu. Převzato z [1]. . . . .	4
2.2	Zleva: Elektromechanický, elektromagnetický a inerciální systém snímání pohybu. (Obrázky převzaty z [6],[19],[16]) . . . . .	5
2.3	Vlevo: Laboratoř chůze s pasivním optickým systémem snímání pohybu. Vpravo: Oblek s markery pro aktivní optický systém. (Obrázky převzaty z [3],[11]) . .	5
2.4	Rotace a translace relativní vůči referenční kameře 1. (Inspirováno z [30]). . .	9
2.5	Výpočet středu markeru metodou nejmenší čtverců. (Inspirováno z [32]). . . .	9
2.6	Oxford Foot Model - přední a zadní pohled rozmístění markerů. (Obrázek převzat z [17]). . . . .	9
2.7	Cyklus chůze. (Inspirováno z [33]). . . . .	12
2.8	Kinematika a kinetika kloubů kyčle, kolene a kotníku. První řádek grafů ilustruje kinematiku kloubů (rotace ve stupních). Prostřední řádek znázorňuje momenty sil působící kolem kloubů a třetí řádek znázorňuje výkon kloubu. Přerušovaná čára v grafu značí standardní odchylku, plná čára průměr. (Obrázek převzat z [25]). . . . .	12
2.9	Anatomické roviny lidského těla. Autor: Edoarado [CC-BY-SA-3.0], via Wikimedia Commons. . . . .	13
2.10	Kinematické proměnné. (Inspirováno z [36]). . . . .	13
2.11	Microsoft Kinect. (Obrázek převzat z [31]). . . . .	14
2.12	Microsoft Kinect. (Obrázek převzat z [31]). . . . .	15
2.13	Vlevo: Místnost ozářená Kinectem. Vpravo: Pohled do IR projektoru emitujícího IR paprsky. . . . .	16
2.14	Inverzní hloubka vrácená Kinectem jako funkce reálné hloubky. (Obrázek převzat z [35]). . . . .	17
2.15	Dvě různé pózy pro jednu siluetu. (Obrázek převzat z [38]). . . . .	18
2.16	<b>Funkce hloubkového obrazu.</b> Žlutý kříž představuje právě zpracovávaný obrazový bod. Červená kolečka odpovídají offsetům obrazových bodů z rovnice 2.3. V (a) jsou dva příklady bodů kdy hloubková funkce vrátí velké rozdíly hloubky. Naopak v (b) odpoví malým rozdílem. (Obrázek převzat z [24]). . . .	20
2.17	<b>Randomizovaný rozhodovací les.</b> Les je tvořen stromy. Každý strom obsahuje vnitřní (dělicí) uzly (modře) a listy (zeleně). Červená šipka indikuje různé cesty skrz stromy pro stejný vstup. (Obrázek převzat z [24]). . . . .	20
2.18	Hardwarová a softwarová komunikace s aplikací. (Inspirováno z [31]). . . . .	22
2.19	Architektura Microsoft Kinect SDK. (Obrázek převzat z [31]). . . . .	23

2.20	Koncept OpenNI se třemi vrstvami - aplikační vrstva, rozhraní OpenNI a hardwarová vrstva. (Obrázek převzat z [10]). . . . .	24
3.1	Schéma navrženého systému složeného z VRPN serverů, Kinect Recorderu a Gait Analyzatoru. . . . .	27
3.2	VRPN server komunikuje s Kinectem pomocí VRPN trackeru s knihovnou OpenNI. Nasnímaná data VRPN server zasílá Kinect Recorderu ke zpracování. . . . .	28
3.3	Kinect Recorder a jeho součásti. . . . .	30
3.4	Gait Analyzer a jeho komponenty. . . . .	33
3.5	Virtuální model kostry. . . . .	35
3.6	Gait Analyzer. V levé části se zobrazuje animace chůze. Vpravo je graf vizualizující kinematické proměnné. . . . .	36
3.7	Kalibrační šachovnice. Vlevo šachovnice vytištěná na papír. Vpravo šachovnice na zrcadle (pouze bílé čtverce). Shora dolů: RGB obraz, IR obraz, hloubkový obraz. . . . .	38
3.8	Reprojekční chyby v pixelech. Každá barva je přiřazena jednomu snímku v sérii. Levý obrázek znázorňuje reprojekční chyby v hloubkovém obraze. Vpravo jsou reprojekční chyby v infračerveném obraze. Kalibrace byla prováděna pomocí šachovnice umístěné ve stejných pozicích pro IR i hloubkový obraz. V tomto případě vykazuje kalibrace z hloubkového obrazu cca 3x větší chybovost reprojekcí. . . . .	40
3.9	Vzájemná poloha dvou Kinectů spočtená po stereo kalibraci. Součástí obrázku jsou i pozice jednotlivých šachovnic. V tomto případě se kalibrace prováděla z 16-ti snímků. . . . .	41
3.10	Pozice levého Kinectu (kamera číslo 16) vzhledem k počátku s. s. . . . .	41
3.11	Kalibrační $\psi$ póza. . . . .	42
3.12	Graf vizualizující kloubní kinematiku pravého kolena v sagitální rovině. Zde jsou navíc srovnávána data ze dvou záznamů. . . . .	43
4.1	Pozice Kinectů vůči počátku získané po kalibraci systému. . . . .	52
4.2	Cyklus chůze levé kyčle. (a) sagitální rovina, (b) transverzální rovina, (c) frontální rovina. Zelená oblast odpovídá zdravé bázi. Modrá křivka odpovídá cyklu chůze participanta s omezenou hybností. . . . .	53
4.3	Cyklus chůze levého kolena. (a) sagitální rovina, (b) transverzální rovina, (c) frontální rovina. Zelená oblast odpovídá zdravé bázi. Modrá křivka odpovídá cyklu chůze participanta s omezenou hybností. . . . .	53
4.4	Snímek z animace cyklu chůze participanta simulujícího poruchu hybnosti. . . . .	54
4.5	Participant simulující poruchu hybnosti s taškou obsahující 10 kg závaží. . . . .	54
4.6	<b>Kyčelní flexe - extenze.</b> Sagitální rovina. Vlevo zdravá báze naměřená navrženým systémem, vpravo referenční zdravá báze (převzato z [27]). Tučná čára značí průměr, tenčí čára (respektive zelená plocha) značí standardní odchylku. . . . .	56
4.7	<b>Kyčelní rotace.</b> Transverzální rovina. Vlevo zdravá báze naměřená navrženým systémem, vpravo referenční zdravá báze (převzato z [27]). Tučná čára značí průměr, tenčí čára (respektive zelená plocha) značí standardní odchylku. . . . .	56

4.8	<b>Kyčelní abdukce - addukce.</b> Frontální rovina. Vlevo zdravá báze naměřená navrženým systémem, vpravo referenční zdravá báze (převzato z [27]). Tučná čára značí průměr, tenčí čára (respektive zelená plocha) značí standardní odchylku. . . . .	56
4.9	<b>Kolenní flexe - extenze.</b> Sagitální rovina. Vlevo zdravá báze naměřená navrženým systémem, vpravo referenční zdravá báze (převzato z [27]). Tučná čára značí průměr, tenčí čára (respektive zelená plocha) značí standardní odchylku. . . . .	57
4.10	<b>Kolenní rotace.</b> Transverzální rovina. Vlevo zdravá báze naměřená navrženým systémem, vpravo referenční zdravá báze (převzato z [27]). Tučná čára značí průměr, tenčí čára (respektive zelená plocha) značí standardní odchylku. . . .	57
4.11	<b>Kolenní abdukce - addukce.</b> Frontální rovina. Vlevo zdravá báze naměřená navrženým systémem, vpravo referenční zdravá báze (převzato z [27]). Tučná čára značí průměr, tenčí čára (respektive zelená plocha) značí standardní odchylku. . . . .	57
B.1	Aplikace Kinect Recorder . . . . .	67
B.2	Aplikace Kinect Recorder - přidání nového VRPN trackeru. Tato operace způsobí vytvoření nového VRPN remote trackeru, který se připojí k uvedenému VRPN serveru. . . . .	68
B.3	Aplikace Kinect Recorder - definice pořadí Kinectů a načtení kalibračního souboru. . . . .	69
B.4	Aplikace Kinect Recorder - sloučení nasnímaných koster do jediné výsledné. .	70
B.5	Aplikace Gait Analyzer - porovnání dvou záznamů. . . . .	71
B.6	Experiment - simulovaná laboratoř chůze. . . . .	72
B.7	Experiment - simulovaná laboratoř chůze. . . . .	73
B.8	Experiment - kalibrace systému. . . . .	74
B.9	Experiment - dynamický záznam. . . . .	75
B.10	Experiment - dynamický záznam. . . . .	76
B.11	Experiment - participant simulující poruchu hybnosti. Na levém rameni nese tašku obsahující závaží o hmotnosti 10 kg, která způsobuje vychýlení těžiště těla, což má vliv i na samotnou chůzi. . . . .	77
C.1	Obsah příloženého CD . . . . .	79





# Seznam tabulek

3.1	Základní struktura C3D souboru. . . . .	45
3.2	Struktura sekce hlavičky. . . . .	46
3.3	Struktura skupiny parametrů. . . . .	47
3.4	Struktura parametru skupiny. . . . .	47
3.5	Struktura 3D / analogových dat. . . . .	48
3.6	Organizace analogových dat pro jeden snímek. . . . .	48



# Kapitola 1

## Úvod

### 1.1 Zadání

Úkolem této práce je prostudování a popsání problematiky snímání pohybu pro účely studie pohybu člověka s aplikací v lékařství. Součástí práce musí být popis kompletního produkčního řetězce této technologie. Dále je potřeba navrhnout experimentální systém napodobující systém tzv. laboratoře chůze (gait laboratory), který pro snímání pohybu využije jednoho nebo více systémů Microsoft Kinect. Nakonec je třeba získané poznatky ověřit experimentálně a pokud možno použít srovnávací data z reálné laboratoře chůze.

### 1.2 Motivace

Problematika snímání pohybu vychází z oboru počítačového vidění. Základem je zařízení, které zaznamenává pohyby člověka, zvířete či nějaké jiné pohyby a ty pak digitalizuje pro další zpracování. Technologie snímání pohybu, anglicky Motion capture (často používaná zkratka MoCap), se v dnešní době používá zejména v zábavním průmyslu (filmy, hry, atd.), ale také v medicínské praxi, v armádě či robotice.

V závislosti na účelu použití těchto zařízení je velmi důležitá rychlost a přesnost s jakou pohyb snímají. Například pohyb odpalování golfového míčku golfistou vyžaduje extrémně rychlé snímání, abychom zachytili celý průběh odpalu. Na druhé straně jiné jevy mohou být pomalejšího charakteru a i přesnost zaznamenání není tolik klíčová (např. hraní PC her). V důsledku toho je na trhu velké množství zařízení, které se liší nejen principem snímání pohybu, ale především cenou. A cena je rozhodující veličinou při pořizování takového systému.

V nedávné době se na trhu objevil systém Kinect od firmy Microsoft [7]. Jedná se o cenově dostupné zařízení, které umožňuje snímání pohybu bez využití speciálních obleků či pomocných značek (tzv. markerů). Díky svým vlastnostem a především ceně se stal nejlépe prodávaným herním zařízením v historii a zapsal se tak do Guinnessovy knihy rekordů. Tato práce si dává za cíl prozkoumat dostupné systémy pro snímání pohybu a zjistit možnosti využití cenově dostupné technologie Microsoft Kinect, především pro analýzu lidské chůze. Microsoft Kinect je zařízení původně určené jako periférie k herní konzoli Xbox 360 [15]. Její primární účel je tedy hraní her. Cílem práce je zodpovězení následující otázky:

Je snímková rychlost 30 FPS a dosahovaná přesnost zařízení Kinect dostatečná pro seriózní analýzu chůze v lékařské praxi? Motivace je zřejmá. Lze nahradit finančně nákladná zařízení používaná v tzv. laboratoři chůze systémem podstatně levnějším? K zodpovězení otázky je nutné nastudovat dostupnou literaturu společně s alternativními možnostmi programových rozhraní (API). Výsledkem celého snažení bude návrh a realizace systému pro analýzu pohybu nasnímaného zařízením Kinect. Přitom je třeba počítat s možností zapojení většího počtu Kinectů.

### 1.3 Struktura práce

Struktura diplomové práce je tato. Následující druhá kapitola **Analýza** rozebírá celou problematiku snímání pohybu společně s analýzou chůze. Závěr této kapitoly je věnován pohybovému snímání Microsoft Kinect. Třetí kapitola **Návrh systému** popisuje experimentální systém, který umožní zaznamenat a analyzovat nasnímanou chůzi. Součástí kapitoly je také popis celého produkčního řetězce navrženého systému a dále struktura dat se kterými systém pracuje. Čtvrtá kapitola **Experiment** je věnována experimentálnímu ověření navrženého a implementovaného systému. Následuje poslední pátá kapitola **Závěr** shrnující dosažené výsledky práce.

# Kapitola 2

## Analýza

Celá diplomová práce se zabývá snímáním pohybu. Tato kapitola nejprve seznamuje čtenáře s uplatněním, metodami a způsobem použití snímání pohybu. Přitom se snaží podávat informace v kontextu s cílem práce. Připomeňme, že cílem je návrh a implementace experimentálního prostředí pro analýzu chůze. Dále je popsána analýza chůze, která umožňuje chůzi studovat a vzešlé poznatky aplikovat v klinické praxi. Závěr kapitoly je věnován pohybovému snímači Microsoft Kinect.

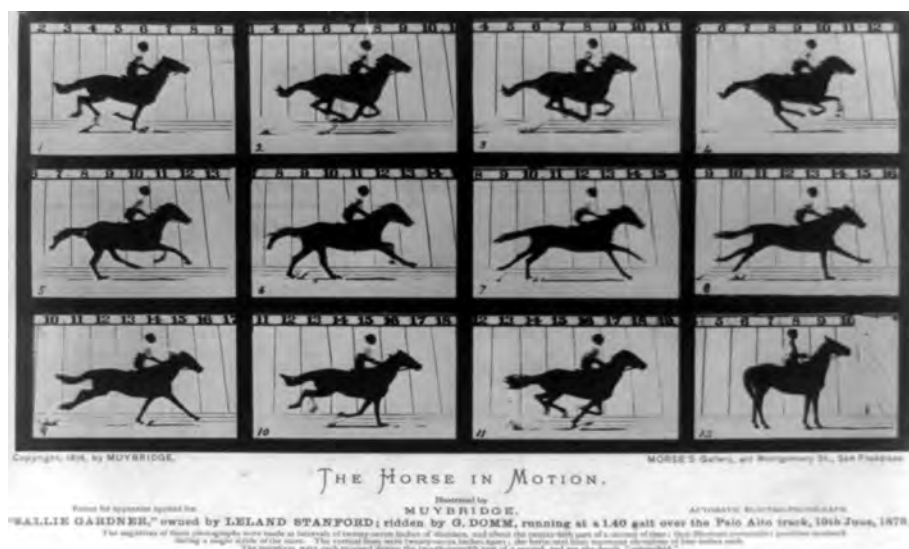
### 2.1 Snímání pohybu

#### 2.1.1 Uplatnění technologie snímání pohybu

Snímání pohybu se poprvé v dějinách lidstva objevuje už v 19. století. Jedním z prvních představitelů je Eadweard Muybridge, který se proslavil sérií snímků „Kůň v pohybu“ pořízených několika fotoaparáty najednou. Každý fotoaparát exponoval snímek v odlišný okamžik. Po složení a přehrání těchto snímků vidí pozorovatel skutečný pohyb koně. Ve 20. století došlo k vylepšení této metody Haroldem E. Edgertonem, který vynalezl vysokorychlostní fotografii s využitím stroboskopu [5].

Poměrně mladým oborem je snímání pohybu pro použití v počítačové animaci. Počátky sahají do sedmdesátých let dvacátého století a teprve v posledních pár letech se více rozšiřuje. Snímání pohybu umožňuje zaznamenávat pohyby lidského těla (nebo nějaké jiné pohyby) a přitom je okamžitě, či později ze záznamu aplikovat na virtuální postavy v počítači. Nasnímané pohyby mohou být namapovány na kostru virtuální postavy nebo na deformaci kůže pro animaci emocí. Stejně dobře jako v animaci se snímání pohybů využívá při analýze pohybu v robotice, kdy robot musí být schopný vnímat pohyby člověka, aby s ním mohl interagovat či kooperovat. Snímání pohybu dnes nalézá uplatnění i v medicíně a sportu.

V medicíně se využívá ke generování biomechanických dat pro analýzu chůze (tzv. gait analysis). Dále pro analýzu mechaniky kloubů, pro diagnostiku páteře a sportovní medicínu. Analýza chůze se v medicíně používá kvůli přesnému zaznamenání všech mechanismů, které se vyskytují v cyklu chůze. Díky tomu je snadné odhalit různé abnormality. Analýza chůze například pomáhá lékařům k upřesnění indikací chirurgické léčby pohybového aparátu dětí postižených dětskou mozkovou obrnou a k monitorování stavu pacienta před a po operaci pohybového aparátu.



Obrázek 2.1: Eadweard Muybridge: Kůň v pohybu. Převzato z [1].

### 2.1.2 Metody snímání pohybu

Existuje několik principů zaznamenávání pohybu. Systémy se dělí do následujících skupin - elektromagnetické, elektromechanické, inerciální a optické. Následující text vychází z článků [19],[21],[22] a práce [36].

#### Elektromagnetické systémy

Snímání pohybu je založeno na sledování relativních změn toku elektrického proudu v magnetickém poli. Systém je tvořen soustavou přijímačů (senzorů) a vysílačem (viz. obrázek 2.2). Vysílač generuje nízkofrekvenční magnetické pole, které je detekováno přijímači. Přijímače jsou připojeny k elektronické řídicí jednotce, která zaznamenaná data filtruje, zesílí a pošle do centrálního počítače. Výhodou tohoto systému je možnost přímo získat orientaci senzorů, bez dodatečných výpočtů. Další výhodou je, že nedochází k sebe-zakrývání senzorů, tak jako u optických systémů. Naopak nevýhodou je limitovaný pracovní prostor snímané osoby z důvodu přítomné nutné kabeláže. Nutno podotknout, že existuje i bezdrátová varianta. Ovšem největším problémem tohoto systému je vada vlastního magnetického pole, vlivem interferencí s jinými magnetickými poli. Díky tomu mohou různé přístroje (např. mobilní telefony) rušit generovaný signál.

#### Elektromechanické systémy

Často je tento systém nazýván jako exo-skeletální díky způsobu připevnění k tělu (viz. obrázek 2.2). Systém je tvořen soustavou pevných segmentů, které se přichytí na klíčovém místě k lidskému tělu. Jednotlivé části exo-skeletonu jsou propojeny klouby s potenciometry měřícími orientaci. Pozice kloubu je určena ze známé velikosti segmentů systému. Jakýkoliv pohyb člověka je tak přenášen do exo-skeletonu a zaznamenán. Výhodami jsou velká přes-



Obrázek 2.2: Zleva: Elektromechanický, elektromagnetický a inerciální systém snímání pohybu. (Obrázky převzaty z [6],[19],[16])



Obrázek 2.3: Vlevo: Laboratoř chůze s pasivním optickým systémem snímání pohybu. Vpravo: Oblek s markery pro aktivní optický systém. (Obrázky převzaty z [3],[11])

nost a dostupnost. Stejně jako u elektromagnetických systémů nedochází k sebe zakrývání. Základní nevýhodou tohoto systému je velikost a omezení pohybu subjektu.

### **Inerciální systémy**

Jedná se o systém implementovaný do speciálního obleku (viz. obrázek 2.2). V tomto obleku jsou na klíčových místech umístěné gyroskopy, které zaznamenávají orientace segmentů těla. Pozice kloubu je získána ze známé vzdálenosti gyroskopů v obleku. Inerciální systémy vynikají vysokou přesností a dostupností. Nevýhodou tohoto systému je neschopnost určit absolutní polohu snímaného subjektu. Problémem tedy nastává pokud chceme nasnímat např. skok, či chůzi do schodů. Tato nevýhoda je na druhé straně vykoupena skladností a malými nároky na velikost prostředí, ve kterém se systém používá.

### **Optické systémy**

Optické systémy jsou v dnešní době nejvíce rozšířené. Lze rozlišit dvě základní skupiny systémů - aktivní a pasivní. Další samostatnou skupinou jsou pak tzv. markerless systémy.

#### **Pasivní optické systémy**

Základem je soustava kamer rozmístěných v prostoru a reflexní kuličky, tzv. markery, připevněné ke sledovanému objektu (viz. obrázek 2.3). Kamery jsou opatřeny skupinou LED, které vyzařují světlo ve směru pohledu kamery. K tomuto účelu se využívají LED emitující červené světlo, světlo blízké infračervenému pásmu a světlo infračervené. Od retroreflexních markerů se generované záření odrazí zpět do kamery. Zde prochází optickým filtrem, který propouští jenom světlo se stejnou charakteristikou, jakou má světlo vyzařované. Kamera tak opravdu vidí pouze snímaný marker.

#### **Aktivní optické systémy**

Jedná se o obdobný systém jako jsou pasivní optické systémy. Rozdíl je v tom, že kamery nejsou vybaveny LED emitující světlo o určité vlnové délce, protože touto LED disponuje snímaný marker (viz. obrázek 2.3). Světlo z markeru je tak přímo vyzářeno do objektivu snímající kamery.

Pro pasivní i aktivní optický systém pak platí následující. Aby bylo možné pozici markeru ve scéně zrekonstruovat, musí marker vidět alespoň dvě kamery systému. Proč tomu tak je se čtenář dočte v sekci 2.1.3. Tento typ systému je schopný poskytnout pouze informaci o pozici markeru ve scéně. Pro získání orientace jsou nezbytné dodatečné výpočty. Tyto systémy jsou velice přesné. Nevýhodou je výrazné okolní světlo, možné zákryty kamer nebo zrcadlové objekty. U aktivních systému je nevýhodou napájení aktivních markerů. Musí být přítomen nějaký zdroj energie, často baterie, kterou je nutné dobíjet.



## Markerless optické systémy

Markerless optický systém oproti ostatním optickým systémům nevyžaduje připevnění speciálních markerů ke snímanému objektu ani žádný speciální oblek či něco podobného. Stačí pouze kamera která snímá scénu. Sofistikovaný algoritmus pak data z kamery zanalyzuje a z nich získá pohyby snímaného objektu. Příkladem je algoritmus [29], který v každém snímku obrazu RGB kamery detekuje siluetu postavy a z ní určí pózu. Aby se předešlo nejednoznačností, používá se dvou pohledově na sebe kolmých kamer. Technika využívající hloubkový senzor funguje obdobně. V každém bodě obrazu se změří vzdálenost objektu před kamerou. Tím vznikne kompletní hloubková mapa pozorované scény. Algoritmus hloubkovou mapu prostuduje a určí finální pózu snímaného člověka. Díky hloubkovému senzoru je míra nejednoznačnosti mnohem menší než u RGB kamer. Na tomto principu funguje systém Microsoft Kinect [7], [24], blíže popsáný v sekci 2.3.

### 2.1.3 Produkční řetězec

Pojem produkční řetězec si můžeme představit jako množinu úkonů potřebnou k nasnímání dat, jejich archivaci a následnému použití. Níže v textu jsou popsány klíčové body a náročnost jednotlivých kroků zejména pro účely studie pohybu člověka v lékařství. K tomuto účelu se využívají především pasivní optické systémy. Vše co je dále v textu zmíněno se vztahuje k pasivním optickým systémům.

## Kalibrace systému

První krok nutný pro správné fungování systému je kalibrace kamer. Jedná se o proces získání vnitřních a vnějších parametrů kamery. Vnitřními parametry se rozumí zejména ohnisková vzdálenost a střed promítání kamery. Vnějšími pak vzájemná pozice a orientace kamer. Uvažujme  $m$  kamer. Označme je od 1 do  $m$ . Kamera 1 je referenční kamerou. Pak můžeme prohlásit, že pozice a orientace kamery  $j$  je určena rotací  $R_j$  a translací  $t_j$  relativní ke kameře 1, jak ukazuje obrázek 2.4. Součástí kalibrace systému je tedy získání co nejpřesnějších  $m - 1$  rotací a  $m - 1$  translací. Přesnost je velice důležitá. Nepřesná kalibrace má za následek nepřesné nasnímání, díky čemuž může být ovlivněna diagnóza vyhodnocujícího lékaře. V tomto textu se zaměříme především na vnější parametry. V našem případě můžeme říci, že pracujeme s vnitřně zkalibrovanými kamerami.

V literatuře se objevuje několik numerických metod a matematických modelů jak problému kalibrace řešit. Např. Devy [23] uvádí globální nelineární minimalizační techniku. Joan Lasenby [30] zase využívá ke kalibraci výpočty geometrické algebry. Zde si nastíníme pouze základní teorii. Pro kalibraci slouží tzv. kalibrační hůlka. Jedná se o hůlku s několika markery. Aby mohly být kamery vzájemně zkalibrované, musí se části jejich pohledových jehlanů překrývat. Obecně musí kamera  $i$  vidět stejný marker, který vidí kamera  $j$ . Rotace  $R$  a translace  $t$  jsou definovány následovně. Zvolme bod  $\mathbf{P}$  jako bod v 3D prostoru.  $\mathbf{X}_i$  a  $\mathbf{X}_j$  jsou souřadnice bodu  $\mathbf{P}$  v souřadném systému kamery  $i$ , respektive  $j$ . Pak pro  $\mathbf{X}_i$  a  $\mathbf{X}_j$  platí

$$\mathbf{X}_j = R * \mathbf{X}_i + t, \quad (2.1)$$

kde  $R$  je rotační matice  $3 \times 3$  a  $t$  je translační vektor. Máme tedy šest neznámých ( $t = [x, y, z]^T$  a v matici  $R$  tři úhly). Kalibrační hůlka se vloží do snímané oblasti a zaznamená se v několika rozdílných pozicích. Pozice markerů v zaznamenaných obrazech poskytují informace o parametrech kamery. Výpočet těchto parametrů je odvozen od zvolené numerické metody, která kalibraci počítá.

### Kalibrace subjektu

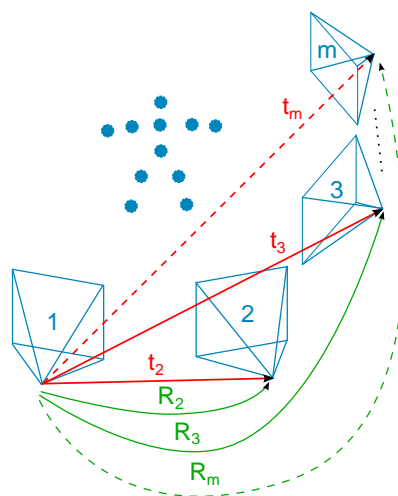
Dalším krokem před zaznamenáním pohybu je připevnění markerů na klíčová místa na lidském těle. Bavíme-li se o snímání pohybu pro účely analýzy chůze, můžeme využít tzv. Oxford Foot Model (OFM) [39]. Jedná se o klinicky ověřený model, vyvinutý Nuffieldským ortopedickým centrem v Oxfordu, předním světovým centrem pro klinickou a výzkumnou analýzu chůze, ve spolupráci s Oxfordskou universitou. Model detailně popisuje anatomické rozmištění markerů na dolní končetině lidského těla - obrázek 2.6.

Po připevnění všech markerů na definovaná místa, se může přikročit ke kalibraci subjektu. K tomu slouží tzv. statický záznam. Subjekt se postaví do snímané oblasti, nejlépe do pozice písmene T (tzn. s rozpaženými rukama) a systém zaznamená několik snímků. Vybere se jeden snímek na kterém jsou vidět všechny markery a provede se kalibrace. Kalibraci se rozumí zahrnutí dodatečné informace důležité pro nasnímání dat. Systém často vyžaduje znalost antropometrických parametrů subjektu. Jde například o celkovou výšku, váhu, délku jednotlivých částí těla, šířku kloubu, atd. Dále je nutné v systému správně označit všechny markery, které kamery snímají. Správné označení je klíčové. Nelze například zaměnit levou nohu s pravou. Nejenže by nasnímaný výsledek neodpovídal realitě, ale navíc by vedl ke špatnému vyhodnocení.

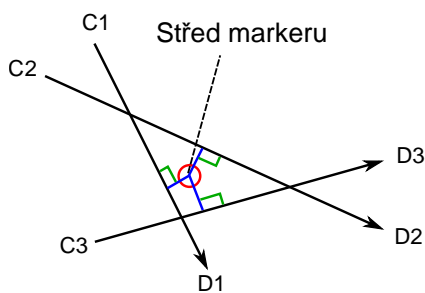
### Snímání

Po úspěšné kalibraci kamer a subjektu můžeme začít se samotným zaznamenáváním pohybu - často se nazývá dynamický záznam. Přihlédneme-li k tomu, že účel našeho snímání je analýza pohybového aparátu, pak stačí aby se subjekt prošel snímanou oblastí. Chůze jako taková je periodicky opakující se jev. Nemusí se tedy zaznamenávat příliš dlouhá sekvence. Stačí zaznamenat pouze jednu opakující se část. Aby se předešlo zkreslení při vyhodnocování, je vhodné uskutečnit více záznamů, z nichž se použije určité procento nejlepších záznamů (např. výběr 6-ti nejlepších záznamů ze 40-ti). Zprůměrování většího počtu záznamů navíc umožní více generalizovat cyklus chůze subjektu, což má vliv i na přesnější vyhodnocení.

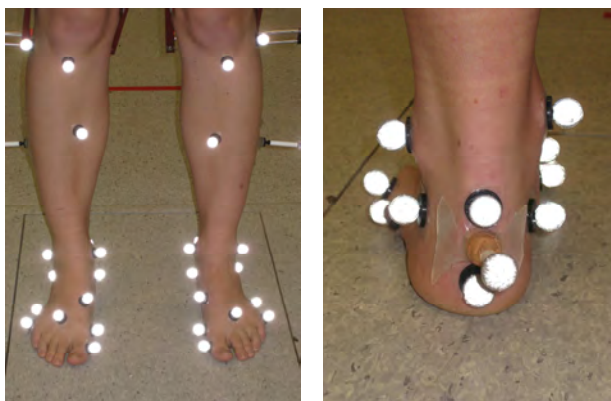
Jak vlastně snímání, respektive 3D rekonstrukce funguje? V každém snímku se identifikují všechny kamery, které vidí konkrétní marker. Ilustrace 2.5 demonsturuje situaci, kdy tři pozorovatelé (C1, C2, C3) vidí jeden bod v 3D prostoru. Pozorovatel C1 vidí bod ve směru z C1 do D1, pozorovatel C2 vidí bod ve směru z C2 do D2 a konečně pozorovatel C3 vidí bod ve směru z C3 do D3. Vlivem chyb v měření není průsečíkem tří paprsků pouze jeden bod, ale rovnou body tři. Pro nalezení co nejméně odlišné pozice markeru ve 3D prostoru se používá optimalizační technika nejmenších čtverců. Střed markeru je metodou nejmenších čtverců určen tak, že součet čtverců nejkratších vzdáleností ze středu markeru ke každému paprsku je minimální. Takto určený bod je vyhodnocen jako nejlepší. Díky synchronizaci kamer mohou být v každém snímku přesně a rychle určeny 3D pozice všech markerů. Pro



Obrázek 2.4: Rotace a translace relativní vůči referenční kameře 1. (Inspirováno z [30]).



Obrázek 2.5: Výpočet středu markeru metodou nejmenší čtverců. (Inspirováno z [32]).



Obrázek 2.6: Oxford Foot Model - přední a zadní pohled rozmístění markerů. (Obrázek převzat z [17]).

3D rekonstrukci je důležité, aby byl marker v daný okamžik viděn alespoň dvěma kamerami. V opačném případě nemáme dostatek informace pro jeho 3D rekonstrukci.

Bavíme-li se o snímání pohybu ve spojitosti s lékařstvím, je třeba mít na paměti, že snímáný subjekt (pacient) trpí určitým onemocněním pohybového aparátu. V důsledku toho můžeme narazit na různé problémy při snímání. Často se stává, že pacient není schopen chůze bez pomoci asistenční osoby nebo pomůcek jako jsou berle, či speciální chodítka. Asistent, ale i zmíněné pomůcky mohou zakrýt markery připevněné k pacientovi. Nevidí-li kamery daný marker, nemohou pak zaznamenat ani jeho trajektorii. Navíc jakýkoli kontakt s markerem může způsobit odpadnutí markeru z těla pacienta a musí být opět připevněn. Pravděpodobnost že se tak stane s chodícími pomůckami samozřejmě roste.

## Zpracování dat

Před vyhodnocením nasnímaných dat je vhodné na data aplikovat různé filtry zvyšující jejich kvalitu. Jedním z filtrů může být filtr na odstranění chvění způsobené šumem v datech. Síla filtru určuje jak moc bude trajektorie vyhlazena a tím chvění potlačeno. Během vytváření dynamického záznamu se může stát, že některé markery jsou na krátký okamžik zastíněny a žádná z kamer jejich pozici nezachytí. Tento okamžik je ve výsledné trajektorii označován jako trhlina, která má trvání několika snímků. Pokud není trhlina příliš velká, lze ji dopočítat například pomocí kubické interpolace křivky. Další možností je vyplnění jednotlivých trhlin pomocí nějakého vzoru, tzn. stejnou částí podobné trajektorie bez trhlin (často se využívá u markerů pánve, které mají podobnou trajektorii).

Posledním krokem celého produkčního řetězce je archivace nasnímaných dat. Jelikož má snímání pohybu velký rozsah uplatnění, existuje velké množství formátů pro archivaci. V medicíně se velmi často používá pro zaznamenání biomechanických dat formát souboru C3D [32]. Jeho výhodou je možnost uchování jak trajektorií markerů, tak i další analogové informace (např. z tlakové plošiny) a to vše v jednom souboru. Jsou-li zaznamenaná data uložena ve zvoleném formátu, už nic nebrání jejich použití v dalších aplikacích, umožňujících jejich analýzu či jiné zpracování.

## 2.2 Analýza chůze

Chůze je základním stereotypem člověka a je pro něj natolik typická, že se stala jednou z charakteristik rodu *homo sapiens* [42]. Schopnost vykonávat tento pohyb je dána souhrou centrálního nervového systému společně s pohybovým aparátem. Analýza chůze (angl. gait analysis) umožňuje tento pohyb podrobně studovat a vzešlé poznatky aplikovat v klinické praxi.

Moderní systémy dramaticky změnil přístup k léčbě poruch pohybového aparátu. V současných českých podmínkách byla dosud indikace k operaci u pohybových poruch postavena na klinickém vyšetření v leže, resp. ve stoje a chůzi [34]. Netrénovaným okem je v podstatě nemožné postihnout veškeré abnormality vyskytující se od pánve až po hlezno. Navíc výsledná diagnóza je silně zatížena subjektivním posouzením vyšetřujícího lékaře. Laboratoř chůze vybavená optickými systémy, tlakovými plošinami a EMG systémy umožňuje přesně zaznamenat celý cyklus chůze a vyšetřující lékař může z exaktních numerických data

navrhnout způsob léčby. Moderní analýza chůze umožňuje předoperativní posouzení konkrétních patologií pacienta. Po operaci navíc poskytuje přesné posouzení výsledku. Díky tomu lze kriticky zhodnotit, zda byl operační zákrok úspěšný či nikoliv. Přehodnocováním výsledků mohou být neúspěšné operační výkony zavrženy a samotná léčba déle rozvíjena a zefektivňována [25].

Analýza chůze je založena na zkoumání kinematických a kinetických parametrů chůze. Kinematika se vztahuje především k orientacím kloubů, trupu a pánve, bez ohledu na zúčastněné působící síly. Měření je vizualizováno grafem v kartézském souřadném systému, kde osa  $x$  znázorňuje časový interval vyjádřený v procentech cyklu chůze a osa  $y$  znázorňuje výchylky kloubu vyjádřené ve stupních (obrázek 2.8). Kinetika zahrnuje měření sil, vnitřních (produkovaných muskulární aktivitou) a vnějších (produkovaných gravitací a vnějším zatížením) a měření momentů sil a práce vykonané mezi klouby způsobující pohyb. Měření je vyjádřeno v N (Newton), Nm (Newtonmetr) a W (Watt) na ose  $y$ , kdežto osa  $x$  opět odpovídá procentům cyklu chůze [33].

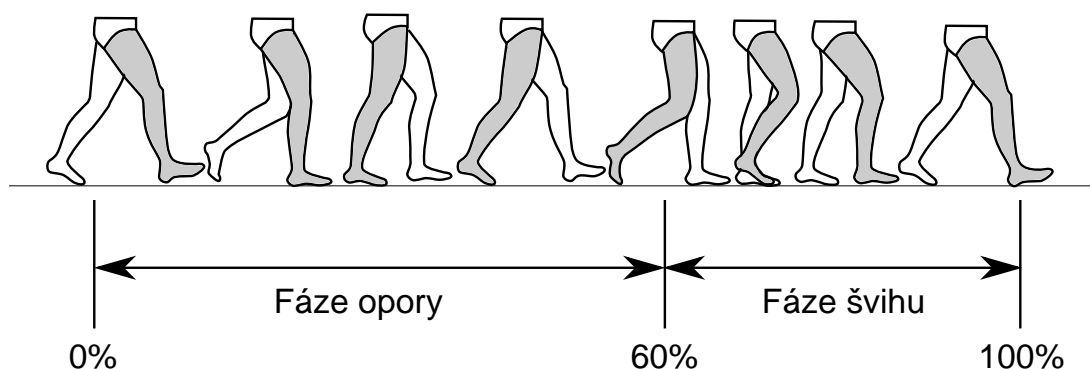
Fyziologicky plní svaly úlohu akceleratorů, decelerátorů a stabilizátorů. Když sval působí jako akcelerator, vždy vyvolává zkracující kontrakci. Decelerátor vyvolává kontrakci prodloužení a konečně stabilizátory působí více nebo méně isometricky (tzn. produkují sílu bez signifikantní změny své délky) [25].

U spastiků (jedinců kteří mají zvýšené svalové napětí) je nejčastější deformitou pes equinus. Jedná se o vadu nohy, při které pata nedošlapuje na podložku. U zdravých subjektů dochází ke kontaktu nejprve paty, potom celé nohy a pak odrazu od špičky nohy [34]. Celý tento děj se nazývá cyklus chůze (gait cycle). Jedná se o sled po sobě jdoucích uniformě opakovaných pohybů v prostoru a čase. Cyklus chůze je rozdělen na dvě části, jak je naznačeno na obrázku 2.7. První je fáze opory a trvá cca 60% cyklu. Druhá je fáze švihy, která má délku trvání cca 40% cyklu. Fáze opory začíná když se pata dotkne podložky (0% cyklu), trvá tak dlouho než dojde ke kontaktu celého chodidla s podložkou a končí v momentě, kdy palec nohy opouští podložku (60% cyklu). Švihová fáze začíná, když palec nohy opustí podložku, trvá tak dlouho jako švih nohy a končí v momentě, kdy se pata opět dotkne podložky (100% cyklu) [33].

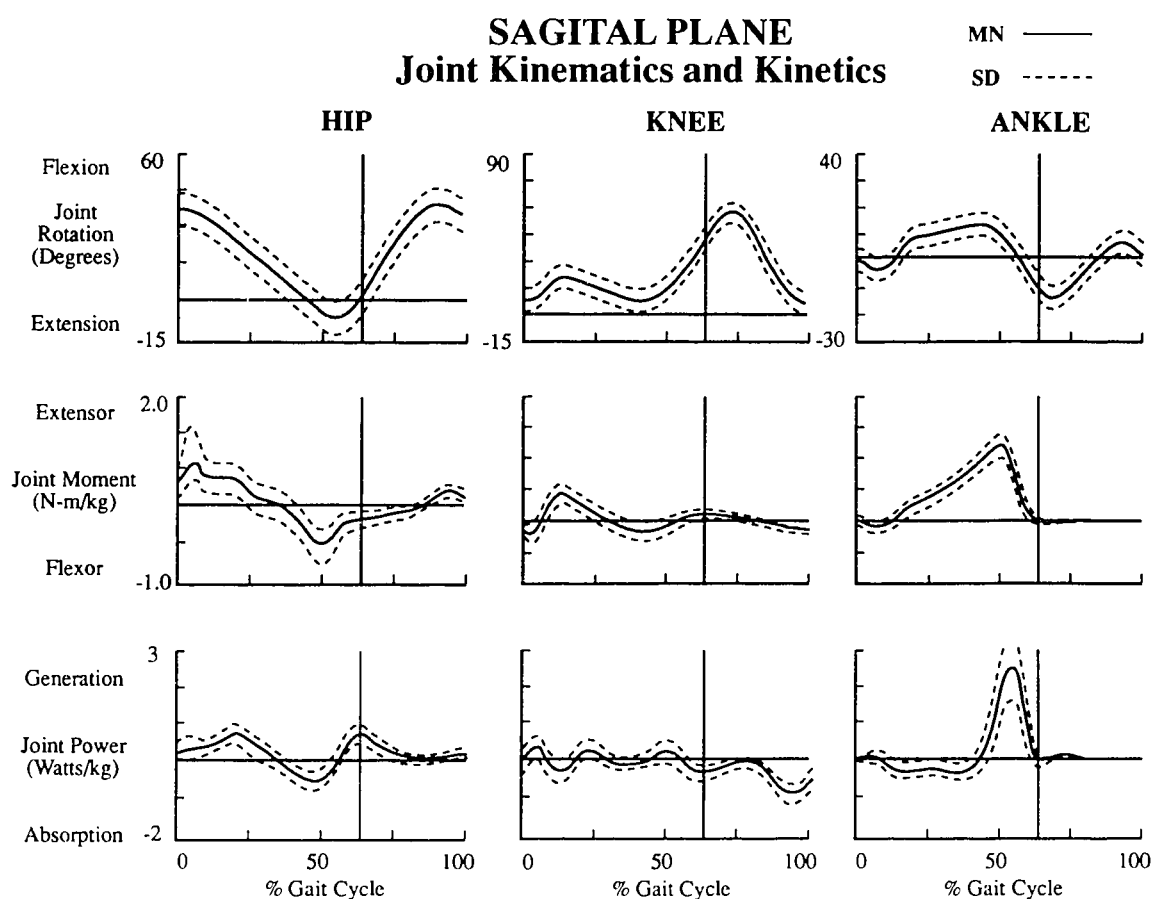
Analýza chůze je jediný nástroj, který nám umožňuje pochopit výše uvedené procesy. Jde o neocenitelnou pomůcku pro řadu neurologů, fyzioterapeutů, ortopedů a protetiků. V České republice existuje pouze pár laboratoří umožňující analýzu chůze. Přitom pouze v Brně, ve Fakultní dětské nemocnici, ji využívají v klinické praxi. Analýza chůze zde nachází uplatnění pro indikace léčby pacientů trpících dětskou mozkovou obrnou.

### 2.2.1 Kinematické proměnné

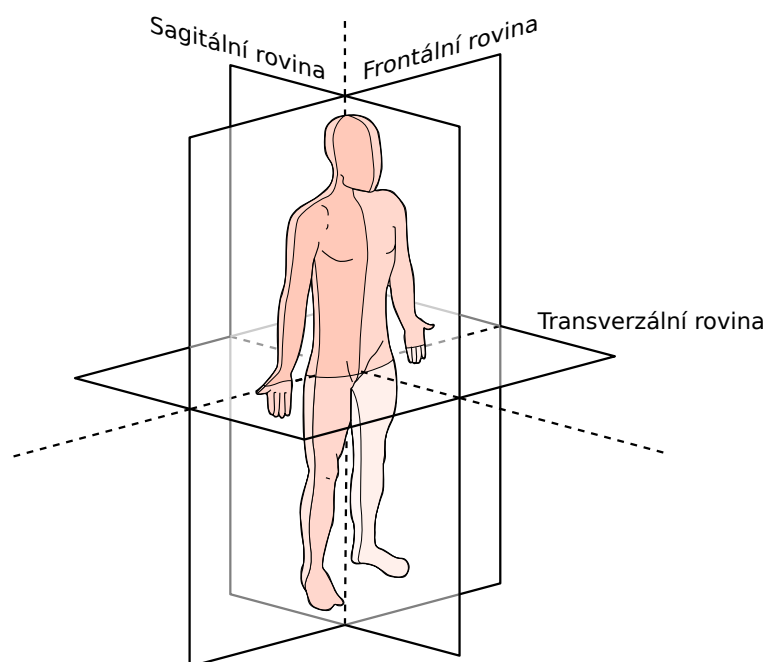
Abychom porozuměli parametrům chůze je nutné znát základní anatomické pojmy. Stejně jako počítačová grafika má i medicína různé souřadné systémy. Dalo by se říci, že lékaři pouze doplnili různá pojmenování, která pro počítačovou grafiku nejsou až tak důležitá. Když se v počítačové grafice mluví o 3D prostoru, tak každý grafik dobře ví, že tento prostor je tvořen třemi na sebe kolmými rovinami  $X$ ,  $Y$ ,  $Z$ . Lékaři tyto tři roviny pojmenovávají *sagitální*, *transverzální* a *frontální*. Laicky řečeno, sagitální rovina rozděluje lidské tělo na pravou a levou polovinu, frontální rovina na přední a zadní polovinu a transverzální rovina na horní a dolní polovinu těla, viz. obrázek 2.9.



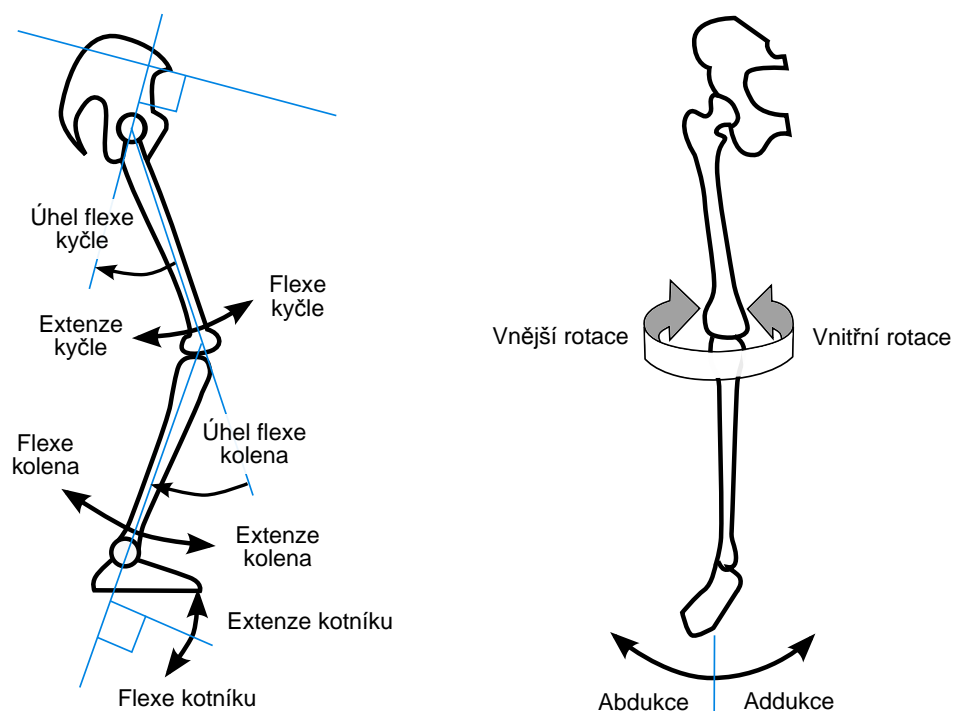
Obrázek 2.7: Cyklus chůze. (Inspirováno z [33]).



Obrázek 2.8: Kinematika a kinetika kloubů kyčle, kolene a kotníku. První řádek grafů ilustruje kinematiku kloubů (rotace ve stupních). Prostřední řádek znázorňuje momenty sil působící kolem kloubů a třetí řádek znázorňuje výkon kloubu. Přerušovaná čára v grafu značí standardní odchylku, plná čára průměr. (Obrázek převzat z [25]).



Obrázek 2.9: Anatomické roviny lidského těla. Autor: Edoarado [CC-BY-SA-3.0], via Wikimedia Commons.



Obrázek 2.10: Kinematické proměnné. (Inspirováno z [36]).

Každý kloub lidského těla má svůj lokální souřadný systém, který je popsán těmito třemi rovinami. Kloub změnou své orientace vyvolává různá natočení končetiny. Podle toho od které osy a na kterou stranu se končetina otáčí, rozlišujeme flexi - extenzi, abdukci - addukci a vnitřní - vnější rotaci. Vždy platí, že jedna rotace je opakem druhé (např. flexe je opak extenze). Celou situaci nejlépe znázorňuje ilustrace 2.10.

Pro analýzu chůze je klíčové zaznamenat všechny tyto rotace a odpovídající úhly jako kinematické proměnné a to pro celou dolní končetinu. Tzn. pro kyčelní a kolenní kloub a kotník.

## 2.3 Pohybový snímač Microsoft Kinect

Microsoft Kinect (Obr. 2.11) je pohybový senzor původně určený jako periferie k herní konzoli XBox 360 [15]. Pomocí Kinectu se stává hráčovo tělo, respektive jeho pohyby, ovladačem. Pohybový senzor snímá a sleduje veškeré jeho pohyby. Díky přirozenému intuitivnímu ovládání se stává hraní her více imerzivní.



Obrázek 2.11: Microsoft Kinect. (Obrázek převzat z [31]).

Kinect se okamžitě po uvedení na trh stal prodejním rekordmanem. Jedná se o nejrychleji prodávanou spotřební elektronikou historie. Za prvních 60 dní od zahájení prodeje dodal Microsoft prodejcům po celém světě 8 milionů kusů Kinectu.

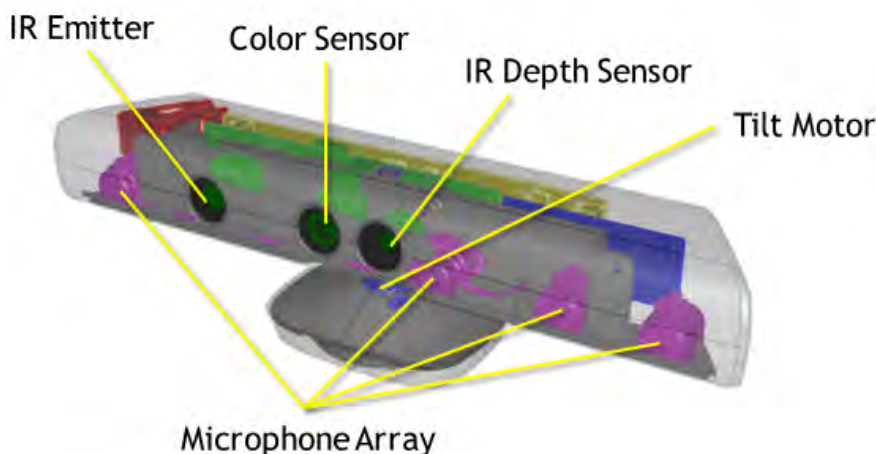
V podstatě ihned po uvedení na trh lidé zjistili, že se jedná o zajímavé zařízení s obrovským potenciálem. Jelikož byl Kinect určen primárně pro herní konzoli, neexistovaly proto žádné ovladače pro PC. Společnost Adafruit Industries (zabývá se open source hardwarem) se proto rozhodla vyhlásit soutěž s peněžitou výhrou pro toho, kdo naprogramuje open source ovladače pro Kinect. Během několika dní se podařilo hackerovi s přezdívkou AlexP vyvinout první ovladače pro PC. Vyhrál tak odměnu tři tisíce dolarů.

Microsoft na hacknutí svého zařízení nereagoval žalobami. Jedna z klíčových osob vývojářského týmu prohlásila, že Kinect vlastně hacknut nebyl. Došlo pouze k tomu, že někdo napsal open source ovladače pro PC, aby mohl s Kinectem komunikovat přes USB, které není nijak chráněno proti čtení. Tudíž mohl lehce číst data ze senzorů.



### 2.3.1 Jak zařízení funguje

Microsoft Kinect se skládá z následujících částí (viz. obrázek 2.12). Stabilitu zařízení zajišťuje základna disponující motorkem, který dovoluje tělo přístroje naklápět nahoru a dolů o 30 stupňů. V těle jsou ukryty čtyři mikrofony, sloužící k hlasovému ovládání. Jeden mikrofón je nalevo a tři napravo. Šířka Kinectu je způsobena právě rozmístěním mikrofónů. Zjistilo se totiž, že takovéto nastavení je optimální pro sledování hlasů a určení místa ze kterého zvuk přišel. Součástí těla je dále RGB kamera umožňující pořizování video záznamů nebo rozpoznání obličeje. Nejdůležitější částí Kinectu je bezesporu 3D hloubkový senzor. Je tvořen dvěma částmi. Projektorem, který emituje záření blízké infračervenému a IR VGA kamera, která sbírá světlo, vyzařené projektorem, odražené od objektů ve scéně. Je třeba zmínit rychlost snímání a rozlišení optické soustavy Kinectu. Rychlost odpovídá frekvenci 30ti snímků za sekundu a rozlišení 640x480 pixelů.

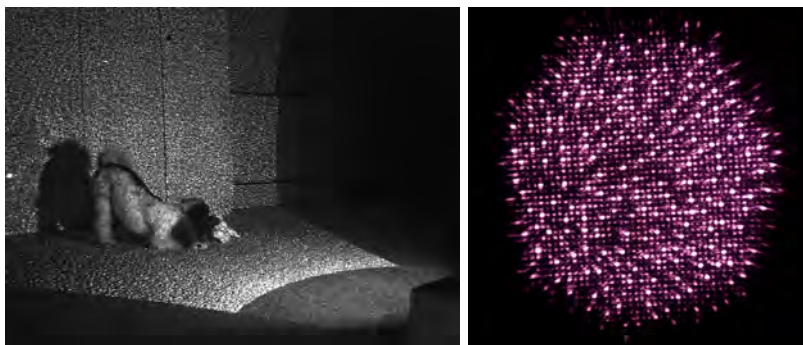


Obrázek 2.12: Microsoft Kinect. (Obrázek převzat z [31]).

Technologie skrývající se uvnitř je neuvěřitelně fantastická. Pro mapování celého prostoru scény se využívá hloubkový senzor. Ten funguje na podobném principu jako sonar. IR projektor vysílá paprsky tvořící určitý vzor (Obr. 2.13 vpravo) a systém měří jak dlouho trvá, než se paprsek odrazí od objektu ve scéně zpět do IR kamery. Pokud víme jak dlouho trvá cesta zpět do kamery, víme jak daleko je objekt ve scéně. Vyzářený paprsek urazí přibližně jeden metr za 3 nanosekundy. Tisíce paprsků (Obr. 2.13 vlevo) jsou pak schopny vytvořit přesnou hloubkovou mapu prostředí.

### 2.3.2 Kinect jako 3D měřicí zařízení

Jak bylo zmíněno v předchozí sekci, zařízení Kinect je vybaveno IR projektorem a IR kamerou. Tyto dvě součásti fungují jako hloubková kamera, která se používá k triangulaci bodů prostoru. Další součástí Kinectu je RGB kamera, sloužící k rozpoznávání obličejů a dalšího obsahu. Kinect jako měřicí zařízení poskytuje tři typy výstupu. Barevný (RGB) obraz, IR obraz a (inverzní) hloubkový obraz. Práce Jana Smíška a kol. [35] analyzuje a ex-



Obrázek 2.13: Vlevo: Místnost ozářená Kinectem. Vpravo: Pohled do IR projektoru emitujícího IR paprsky.

perimentálně vyšetřuje Kinect a jeho měřicí schopnosti. Následující zjištění pocházejí právě z této práce.

### RGB obraz

Barevná kamera umožňuje snímat scénu v 8-bitovém VGA rozlišení  $640 \times 480$  pixelů s frekvencí 30 Hz. Snímač kamery využívá konvenční řešení s Bayerovou maskou. Kamera disponuje zorným polem  $63 \times 50$  stupňů, ohniskovou vzdáleností 2.9 mm a velikostí pixelu  $2.8 \mu m$ . Kamera tak díky svým vlastnostem poskytuje barevnou obrazovou informaci střední kvality.

### IR obraz

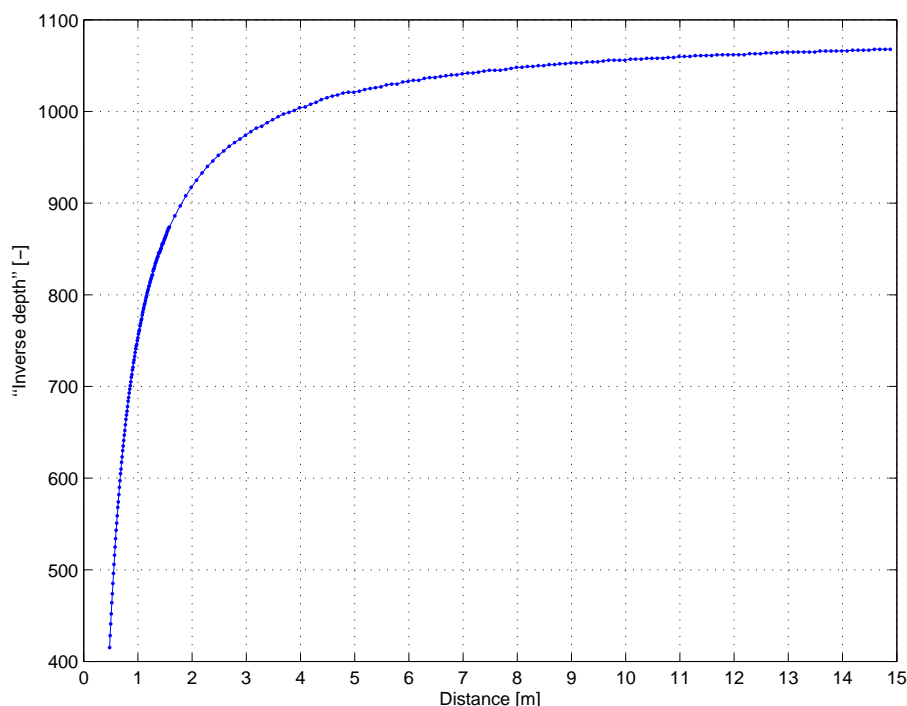
IR kamera slouží ke sběru paprsků emitovaných IR projektorem. Rozlišení obrazu je VGA  $640 \times 480$  pixelů. Monochromatický obraz je snímán rychlostí 30 Hz s 11-bitovou hloubkou. Kamera disponuje zorným polem  $57 \times 45$  stupňů, ohniskovou vzdáleností 6.1 mm a velikostí pixelu  $5.2 \mu m$ .

### Hloubkový obraz

Hloubkový obraz je zrekonstruován triangulací bodů z IR obrazu. Jak je zmíněno v [35], Kinect vrací jakousi "inverzní hloubku"  $d$  (obrázek 2.14) podél osy  $z$ , místo aktuální hloubky  $z$  odpovídající hloubce ve scéně. K určení souřadnic 3D bodu z odpovídajícího pixelu  $[x, y, d]$  hloubkového obrazu je třeba znát kalibrační matici kamery  $K$ . Pokud bychom neuvažovali zkreslení kamery, získáme souřadnice 3D bodu  $X$  v prostoru jako

$$X = \frac{1}{d} \left( K^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \right) \quad (2.2)$$

kde  $x$  a  $y$  jsou souřadnice pixelu v hloubkovém obraze a  $d$  je hodnota tohoto pixelu.



Obrázek 2.14: Inverzní hloubka vrácená Kinectem jako funkce reálné hloubky. (Obrázek převzat z [35]).

### 2.3.3 Markerless snímání pohybu

Problém snímání pohybů člověka bez použití markerů se vědci snaží vyřešit přibližně 20 let. Snímání a analýza lidských pohybů má uplatnění v mnoha odlišných aplikacích, zahrnujících interakci člověka s počítačem, hraní her, bezpečnost a dokonce i zdravotnictví. Hlavní výzvou v této oblasti je interpretace pózy člověka, která se promítá do dvoudimenzionálního prostoru. Přitom se musí řešit odlišné tvary a velikosti lidských postav, jejich oblečení, délka vlasů případně osvětlení.

Během let výzkumu se objevily různé metody umožňující snímání pohybů člověka. Tyto metody jsou založené na zpracování dat z jedné kamery, systému více kamer a senzorů umožňujících vidět světlo o vlnových délkách, které lidské oko není schopné vidět. Robustní určení pózy v třidimenzionálním prostoru není jednoduchá úloha. Navíc pokud se využívá pouze jedna optická kamera snímající jas ve scéně. Při využití metod založených na informaci o siluetě postavy dochází často k chybám při sebe zakrývání končetin. Naopak metody které nejsou založeny na siluetě postavy, mívají více problémy s osvětlením a texturami v obraze. Úloha se v poslední době značně ulehčila uvedením hloubkových senzorů, které se navíc stávají více cenově dostupné. Hloubková informace zjednodušuje problém dvojznačnosti, který může nastat při použití barevného obrazu. Může se například stát, že silueta pózy odpovídá dvěma pózám postavy, jak naznačuje obrázek 2.15. Oproti intenzitním senzorům nejsou hloubkové kamery náchylné na osvětlení scény. Navíc výstup nijak neovlivňují ani barvy a textury ve scéně.



Obrázek 2.15: Dvě různé pózy pro jednu siluetu. (Obrázek převzat z [38]).

Rozšířením hloubkových kamer se objevilo velké množství prací a článků, které se zabývaly určením pózy člověka z hloubkového obrazu. Grest a kolektiv [26] využívají k odhadnutí pózy člověka metodu ICP (Iterative Closest Point). Parametry pózy jsou přitom aktualizovány použitím inverzní kinematiky založené na hustotě korespondencí mezi pozorováním (body hloubkového obrazu) a body modelu člověka, které jsou nalezeny metodou nejbližší soused. Zhu a Fujimura [40] staví heuristické detektory pro určení pózy horní poloviny těla (hlava, trup, ruce) použitím relaxace v lineárním programování. Nevýhodou jejich řešení je nutnost T-pózy k určení velikosti modelu. Jejich další společná práce [41] pojednává o tzv. Bayesian frameworku, který kombinuje výhody lokální optimalizace ICP a klíčových bodů (anatomických bodů lidského těla). Kar [28] ukazuje jak využít hloubkový obraz společně s barevným obrazem k určení pózy člověka.

### Metoda využívaná Microsoftem

Fitzgibbon a Tsunoda prezentovali [38] a dále Shotton a kol. [24] popsali možnosti a schopnosti Kinectu, včetně algoritmu používaného ke skeletálnímu trackingu. Při jeho vývoji byl kladen důraz na dvě věci - výpočetní efektivnost a robustnost. Pro řešení problému skeletálního trackování byl zvolen přístup strojového učení. Myšlenka byla taková, že se počítač naučí rozpoznávat velké množství lidí různých tvarů a velikostí, v rozdílných pozicích a také se naučí rozpoznat části lidského těla. Počítač pak jednoduše porovná obraz pořízený kamerou s jedním obrazem který zná a stanoví o jakou pózu se jedná. Aby se redukovalo množství dat, systém se dívá na každý obrazový bod separátně. Nepotřebuje tak znát všechny možné kombinace póz všech končetin.

Jak metoda ve stručnosti funguje? Ze vstupního hloubkového obrazu se segmentuje silueta postavy (popředí) na několik částí těla, které jsou definovány jako blízké kloubům kostry. Poté se lokalizují prostorové módy každé části a tak se generují eventuelní 3D pozice pro klouby těla a to dokonce pro více uživatelů najednou.

V následující textu je trochu více popsán samotný algoritmus pro nalezení virtuální kostry člověka, snímaného hloubkovým senzorem Kinect. Nejprve je popsáno jakým způsobem se tělo značkuje. Následuje zmínka o funkci hloubkového obrazu. Poté je popsán randomizovaný rozhodovací les jako efektivní klasifikátor a konec sekce je věnován určení možných pozic kloubů.

### Značkování částí těla

Klíčovou roli v celé metodě hraje označení částí lidského těla. Díky tomu je systém schopen rozpoznat, kde se nachází kterákoli končetina, hlava či trup. Shotton a kol. [24] definují několik značek, které pokrývají části lidského těla. Některé značky přímo korespondují s klouby kostry, které nás zajímají, jiné vyplňují mezery nebo mohou být použity k predikci dalších kloubů. V již zmíněném textu [24] se využívá 31 značek odpovídajících částem lidského těla: LU/RU/LW/RW hlava, krk, L/R rameno, LU/RU/LW/RW paže, L/R loket, L/R zápěstí, L/R ruka, LU/RU/LW/RW trup, LU/RU/LW/RW noha, L/R koleno, L/R kotník, L/R chodidlo (L - left, R - right, U - upper, W - lower). Odlišení levých částí od pravých umožňuje klasifikátoru jednoznačně rozlišit levou a pravou stranu těla.

### Funkce hloubkového obrazu

V této podsekcí je zmíněna porovnávací funkce hloubky obrazu. Pro daný pixel  $x$  funkce spočte

$$f_{\theta}(I, x) = d_I \left( x + \frac{u}{d_I(x)} \right) - d_I \left( x + \frac{v}{d_I(x)} \right), \quad (2.3)$$

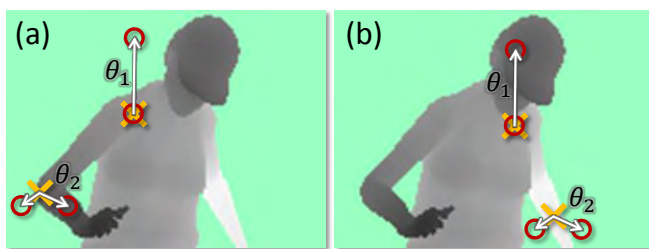
kde  $d_I(x)$  je hloubka pixelu  $x$  v obraze  $I$  a parametry  $\theta = (u, v)$  popisují offsety  $u$  a  $v$ . Normalizace offsetů  $\frac{1}{d_I(x)}$  zajišťuje funkci invariantnost vůči hloubce. Když offset pixelu bude ležet na pozadí nebo mimo hranice obrazu, vrátí  $d_I(x)$  velkou kladnou hloubku. Naopak, když bude offset pixelu ležet na siluete postavy, vrátí  $d_I(x)$  menší hodnotu hloubky než jakou má pozadí.

Obr. 2.16 ukazuje dvě funkce pro různé pixely  $x$ . Funkce  $f_{\theta_1}$  nalézá v obraze horní vodorovnou hranici obrysu objektů. Rovnice (2.3) pro funkci  $f_{\theta_1}$  směřující nahoru odpoví velkou pozitivní hodnotou pro pixely  $x$ , které jsou blíže vodorovné hranici siluety postavy v obraze. Pro pixely umístěné níže odpoví hodnotou blízkou nule. Funkce  $f_{\theta_2}$  funguje obdobně, ale oproti funkci  $f_{\theta_1}$  pomáhá v obraze najít úzké svislé struktury jako jsou ruce a nohy.

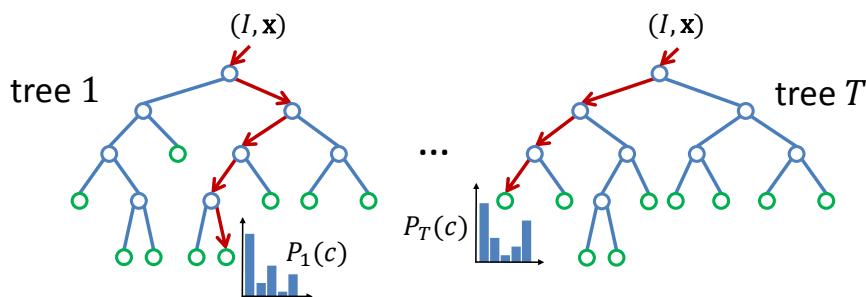
Samostatně tyto funkce poskytují slabou informaci o tom, které části těla pixel patří. Proto se jich využívá v kombinaci s rozhodovacím lesem kde jsou dostatečně přesné. Vytvoření těchto funkcí bylo motivováno jejich výpočetní efektivností. Není třeba žádné předzpracování obrazu. Díky svým vlastnostem se přímo nabízí implementace a paralelizace výpočtů na GPU.

### Randomizovaný rozhodovací les

Randomizované rozhodovací stromy a lesy jsou obecně vybaveny rychlými a efektivními klasifikátory, čímž se uplatňují v různých úlohách jako např. nalezení klíčových bodů v obraze. Jejich výhodou je možnost efektivní implementace na grafických kartách, jejichž architektura umožňuje paralelní zpracování úloh. Ilustrace 2.17 znázorňuje les tvořený  $T$  rozhodovacími stromy. Každý strom je složen z vnitřních uzlů a listů. Vnitřní uzel, tzv. dělicí, obsahuje funkci hloubkového obrazu  $f_{\theta}$  z předchozí sekce a práh  $\tau$ . Klasifikace pixelu  $x$  začíná v kořenovém uzlu a opakovaně se vyhodnocuje rovnice (2.3), čímž se traverzuje strom vlevo nebo vpravo



Obrázek 2.16: **Funkce hloubkového obrazu.** Žlutý kříž představuje právě zpracovávaný obrazový bod. Červená kolečka odpovídají offsetům obrazových bodů z rovnice 2.3. V (a) jsou dva příklady bodů kdy hloubková funkce vrátí velké rozdíly hloubky. Naopak v (b) odpoví malým rozdílem. (Obrázek převzat z [24]).



Obrázek 2.17: **Randomizovaný rozhodovací les.** Les je tvořen stromy. Každý strom obsahuje vnitřní (dělicí) uzly (modře) a listy (zeleně). Červená šipka indikuje různé cesty skrz stromy pro stejný vstup. (Obrázek převzat z [24]).

podle porovnání s prahem  $\tau$ . Po dosažení listu stromu  $t$  se zaznamená naučená pravděpodobnost  $P_t(c|I, x)$  podle značení částí těla  $c$ . Pravděpodobnosti jsou pak společně zprůměrovány ze všech stromů lesa, aby daly výsledné ohodnocení

$$P(c|I, x) = \frac{1}{T} \sum_{t=1}^T P_t(c|I, x). \quad (2.4)$$

Každý strom je naučen na rozdílnou náhodně vygenerovanou množinu obrázků. Náhodně je vybrána podmnožina 2000 pixelů z každého obrázku, aby se zajistilo hrubé rozložení mezi částmi těla.

### Možné pozice kloubů

Rozpoznání částí těla popsané výše je vyvozeno z informace získané po jednotlivých pixelech. Tato informace musí být nyní sdružena přes všechny pixely pro vygenerování možných 3D pozic kloubů kostry. Pozice kloubů jsou pak výstupem celého algoritmu zjišťování pózy

člověka a mohou tak být použity pro skeletální tracking, umožňující automatickou inicializaci a zotavování se z chyb.

Jednoduchou možností jak 3D pozice kloubů získat, je odhadnutí globálních 3D středů pravděpodobných těžišť každé části těla. Nicméně v některých situacích není kvalita dostatečná. Místo toho se používá přístup nalézání lokálních módů (local mode-finding) založených na metodě mean shift s váženým Gaussiánským jádrem. V [24] definují odhad hustoty pro jednotlivé části těla jako

$$f_c(X) \propto \sum_{i=1}^N w_{ic} \exp \left( - \left\| \frac{X - X_i}{b_c} \right\|^2 \right) \quad (2.5)$$

kde  $X$  je souřadnice v 3D prostoru světa,  $N$  je počet pixelů obrazu,  $w_{ic}$  je váha pixelu  $i$ ,  $X_i$  je reprojekční obraz bodu  $x_i$  do prostoru světa daného hloubkou  $d_I(x_i)$  a  $b_c$  je po částech naučená šířka pásma. Váha  $w_{ic}$  bere v úvahu pravděpodobnost výskytu části těla v pixelu  $i$  a povrch oblasti pixelu v prostoru světa

$$w_{ic} = P(c|I, x) d_I(x_i)^2. \quad (2.6)$$

Tím je zaručeno, že odhad hustoty je invariantní vůči hloubce a dává malé, ale významné vylepšení v přesnosti predikce kloubu.

Segmentační metoda mean shift se používá k nalezení lokálních módů. Metoda shlukuje body obrazu na základě podobnosti jejich vzhledu a blízkosti jejich pozice. Všechny obrazové body, nad naučeným pravděpodobnostním prahem  $\lambda_c$ , jsou použity jako počáteční body pro část  $c$ . Výsledný odhad je dán jako součet vah pixelů dosažených v každém módu. Detekované módy leží na povrchu těla. Každý mód je proto vložen zpět do scény o naučený z offset  $\zeta_c$ , který umožní získat konečnou pozici kloubu. Tento jednoduchý a efektivní přístup byl Shottonem a kol. [24] otestován na velkém množství dat a shledán v praxi za dobře fungující.

### 2.3.4 Dostupné technologie a knihovny

V této sekci jsou uvedeny dostupné technologie a knihovny umožňující snímání pohybu zařízením Microsoft Kinect. V současné době existuje více možností jak problém zpracování dat z Kinectu řešit.

Přestože byl Kinect původně plánován pro herní konzoli XBox 360, byl v podstatě od svého vypuštění na trh zkoumán vývojáři po celém světě a to pro využití nejen v oblasti herního průmyslu. Vývojáři toužili po možnosti připojení Kinectu k PC a tak vznikly různé výzkumné projekty, ze kterých vzešly ovladače a aplikační rozhraní pro obsluhu Kinectu.

#### Microsoft Kinect SDK

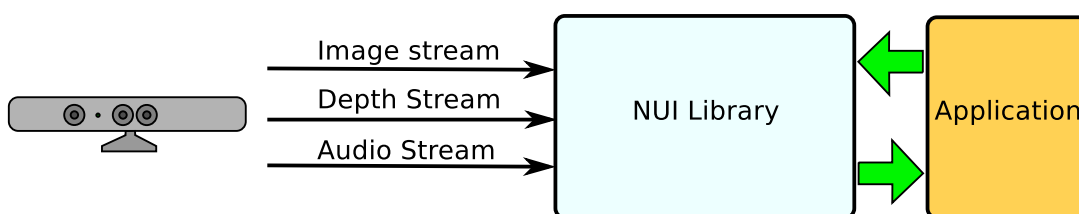
Přestože výrobce zařízení koncipoval pro hraní her na konzoli XBox 360, neuniklo mu, že vzniká stále více projektů využívající Kinect s PC. Proto na začátku roku 2012 vydal vlastní vývojovou sadu nástrojů, se kterou vydal i novou verzi zařízení Kinect s označením Kinect for Windows. V podstatě se jedná o identické zařízení s tím rozdílem, že Kinect for Windows je jediné plně testované a podporované zařízení pro použití s operačním systémem Windows. Přesto i s oficiálním SDK lze používat Kinect pro XBox 360. Jediné s čím musí uživatel

počítat je, že některé funkce nebude mít zpřístupněné (např. blízký mód). Navíc pokud se vývojář rozhodne komerčně šířit svojí Windows aplikaci, musí tak činit pouze se zařízením Kinect for Windows.

Kinect SDK poskytuje sofistikovanou knihovnu s nástroji pomáhajícími vývojářům vytvářet aplikace s přirozeným uživatelským rozhraním (Natural User Interface - NUI). Obrázek 2.18 ilustruje hardwarovou a softwarovou komunikaci s využitím Kinect SDK. Samotné SDK je tvořeno následujícími pěti komponentami (viz. obrázek 2.19):

- Kinect hardware - hardwarové komponenty zahrnující Kinect senzor a také USB hub, skrz který je Kinect připojen k počítači.
- Kinect ovladače - ovladače pro systém Windows, pomocí kterých lze komunikovat se zařízením.
- Audio a video komponenty - jedná se o rozhraní pro skeletální tracking, audio a zobrazování barevného a hloubkového obrazu.
- DirectX Media Object (DMO) - technologie pro zpracování audio signálu a lokalizaci zdroje zvuků.
- Standardní Windows API - vývojová sada nástrojů dostupná v systému Windows.

Podle použití může aplikace využívat buď základní nebo blízký mód hloubkového obrazu. Základní mód umožňuje zaznamenávat data hloubkového obrazu v rozsahu 0.8m až 4m od zařízení, zatímco blízký mód umožňuje záznam v rozsahu od 0.4m do 3m. Blízký mód je výhodný například v situaci, kdy člověk sedí u stolu s PC a Kinect má posazen na monitoru. Schopností Kinectu je snímání až dvou virtuálních koster najednou a detekce až šesti lidí v obraze. Kinect snímá v základním režimu kostru uživatele ve stoje, avšak je schopen snímat uživatele i v sedě. V prvním případě je kostra tvořená 20 klouby, zatímco v druhém snímá Kinect pouze 10 kloubů horní části těla. Při snímání kostry uživatele se používá systém auto-kalibrace. Není tedy nutná žádná kalibrační póza. Stačí aby Kinect uživatele viděl a během okamžiku je kostra trackována.

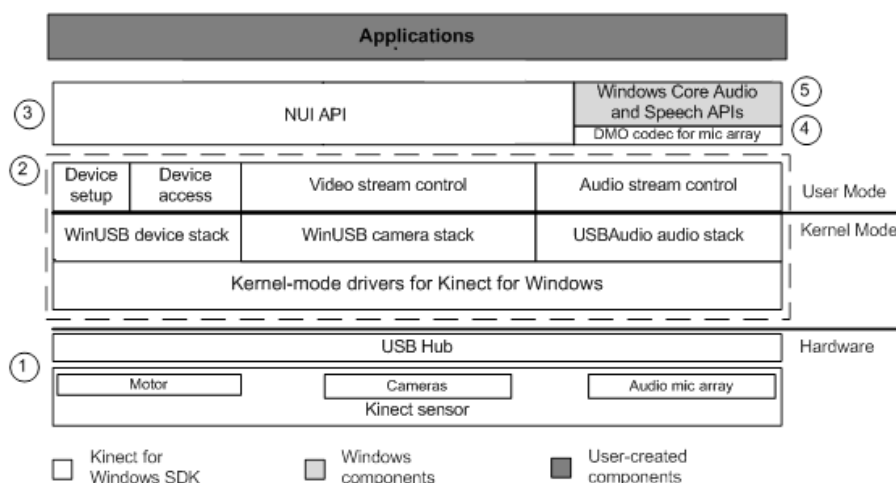


Obrázek 2.18: Hardwarová a softwarová komunikace s aplikací. (Inspirováno z [31]).

## OpenNI

Firma PrimeSense [12] vytvořila pro své PrimeSensor kamery (sdílející technologii s Kinectem) ovladače a API, jejichž zdrojové kódy zveřejnila. Díky tomu společnost PrimeSense předložila neziskové organizaci OpenNI [10] nástroj pro vytvoření vlastního multiplatformního





Obrázek 2.19: Architektura Microsoft Kinect SDK. (Obrázek převzat z [31]).

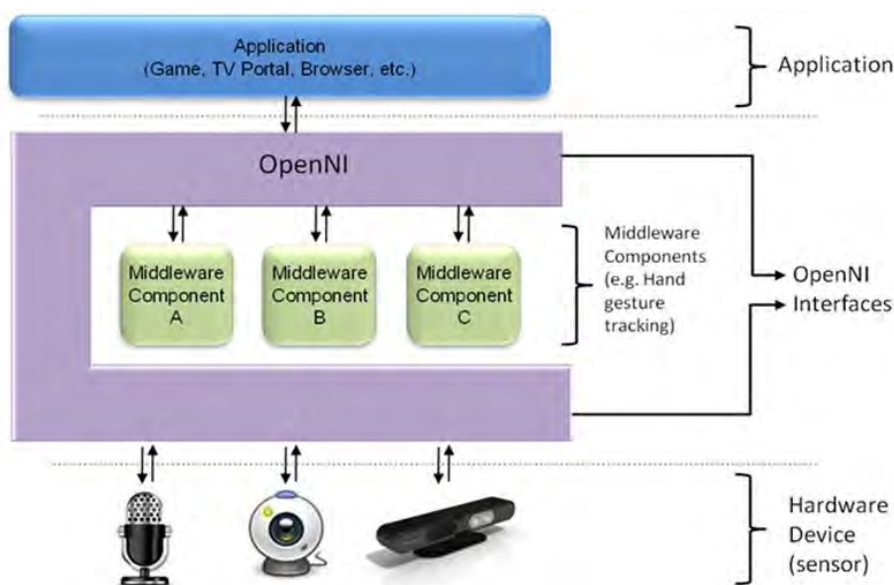
open source frameworku. OpenNI framework je abstraktní vrstva, která poskytuje rozhraní pro fyzická zařízení a middleware komponenty. Framework lze použít pro práci s různými 3D senzory, webovými a IR kamerami. Není tedy určen pouze pro Microsoft Kinect.

Koncept OpenNI je tvořen třemi vrstvami (obrázek 2.20). Horní vrstva reprezentuje aplikaci implementující „přirozenou interakci“. Prostřední vrstva představuje OpenNI, poskytující rozhraní pro komunikaci se zařízeními a middleware komponentami, které zpracovávají syrová data ze zařízení. Spodní vrstva reprezentuje zařízení, které snímá scénu.

OpenNI framework umožňuje přistupovat ke všem součástem zařízení Kinect. Máme možnost získávat informace z hloubkového senzoru, IR a RGB kamery, ale také z integrovaných mikrofónů. Systém middleware komponent je velice sofistikovaný. Buď máme na výběr komponenty připravené OpenNI, nebo si můžeme naprogramovat vlastní. V současné době OpenNI nabízí komponenty pro analýzu celého těla, umožňující získání datových struktur obsahujících informace o pozici a orientaci kloubů, těžišti těla, atd. Dále existují komponenty na detekci předdefinovaných gest, snímání pozice rukou (pohyb ruky může být např. namapován na pohyb myši) a analyzátor scény, který umožňuje rozpoznat popředí od pozadí, určit pozici podlahy atd.

Middleware komponenta NITE, navržená pro analýzu celého těla, nabízená OpenNI, umožňuje snímání virtuální kostry uživatele čítající 15 kloubů. Stejně jako Microsoft Kinect SDK, i NITE je schopen auto-kalibrace uživatele. Přitom můžeme snímat pozice a orientace kloubů celé kostry nebo pouze horní poloviny či dolní poloviny těla.

Je nutno poznamenat, že organizace OpenNI vydáním svého frameworku předběhla samotný Microsoft. Microsoft dlouhou dobu vyvíjel své řešení Kinect SDK a aby vývojáře alespoň částečně uspokojil, vydal betaverzi, které trvalo poměrně dlouho, než se stala hotovým řešením. Navíc nebylo možné betaverzi používat pro komerční účely. Z tohoto důvodu vznikalo mnohem více aplikací využívající řešení od OpenNI.



Obrázek 2.20: Koncept OpenNI se třemi vrstvami - aplikační vrstva, rozhraní OpenNI a hardwarová vrstva. (Obrázek převzat z [10]).

## OpenKinect

Dalším open source projektem je OpenKinect [9]. Jedná se o vůbec první knihovnu, která umožnila komunikaci s Kinectem. Dnes tato komunita, čítající asi 2000 vývojářů, postupně vylepšuje vlastní knihovnu *libfreenect*, poskytující jednoduché a přímočaré rozhraní ke Kinectu. Knihovna *libfreenect*, která je společně s ovladačem součástí OpenKinect, umožňuje pouze základní nízkoúrovňovou komunikaci se zařízením. K dispozici tak není ani žádný algoritmus pro skeletální tracking. OpenKinect tak v současné době nemůže konkurovat MS Kinect SDK, ani OpenNI.

## Kapitola 3

# Návrh systému

Tato kapitola pojednává o návrhu a realizaci experimentálního systému pro analýzu chůze, který využívá jedno nebo více zařízení Microsoft Kinect. Úvodem je potřeba zmínit motivaci použití více zařízení Kinect. V sekci 2.3 byl popsán hloubkový senzor Kinectu společně s algoritmem pro skeletální tracking. Stejně jako u optických systémů snímání pohybu, tak i u hloubkového senzoru nastává problém v okamžiku, kdy senzor „nevidí“ některou část těla. K této situaci dochází například když je snímána osoba bokem ke Kinectu nebo když jedna končetina zastíní druhou. Potenciálním místem zastínění je fáze cyklu chůze, kdy jedna noha mívá druhou. Cílem je tedy pokrýt tyto stavy a pokusit se je korektně zaznamenat. K tomu se využije větší množství Kinectů. Každý Kinect snímá subjekt z odlišného úhlu. Správné rozestavení zajišťuje, že v okamžiku, kdy jeden Kinect některou z končetin nevidí, ji uvidí jiný Kinect. Systém tvořený více Kinecty ale přináší i problémy. V prvé řadě se musí data ze všech Kinectů transformovat do jednoho souřadného systému a teprve poté s nimi lze společně pracovat. Daleko větší problém způsobují interference způsobené vzorem, který každý infračervený projektor Kinectu generuje (viz. obrázek 2.13). Díky těmto interferencím dochází k selhávání vyhodnocení pózy snímání osoby. S větším počtem Kinectů stoupá i pravděpodobnost výskytu interferencí. Nelze tedy říci, že větší počet Kinectů spásí problém sebe zakrývání.

Než bude diskutován samotný návrh systému, musí být rozhodnuto o technologiích, které by měl navrhovaný systém využívat. Na konci předchozí kapitoly byly zmíněny dostupné knihovny pro práci s daty ze systému Kinect. V úvahu přichází buď Microsoft Kinect SDK nebo produkt od organizace OpenNI. Důležitým kritériem je schopnost skeletálního trackingu. Obě knihovny skeletální tracking podporují, avšak v přímém porovnání vychází řešení od Microsoftu lépe. Microsoft umožňuje snímat až 20 kloubů (z toho 10 v dolní polovině těla), kdežto řešení NITE od OpenNI nabízí pouze 15 kloubů (z toho jen 6 v dolní polovině těla). Další výhodou je, že Microsoft nabízí u všech kloubů jak pozici, tak i jejich orientaci (v lokálním a globálním souřadném systému). Komponenta NITE poskytuje také pozice a orientace všech kloubů, ale pouze v globálním souřadném systému (tzn. relativně vůči kameře). Navíc orientace zápěstí a kotníku nejsou nezávisle trackovány, ale jsou nastaveny podle orientace loktu a kolena. Jinými slovy nasnímané orientace zápěstí a kotníku jsou jen odhadované a nemusí odpovídat skutečné póze snímání osoby.

Jelikož systém bude využívat více zařízení Kinect, musí se zaznamenaná data transformovat do stejného souřadného systému. Je tedy zapotřebí provádět kalibraci zařízení. Dobrá

kalibrace je klíčovým faktorem pro správnou transformaci dat nasnímaných hloubkovým senzorem. V literatuře [18] se uvádí, že kalibraci lze provádět buď pomocí infračerveného nebo hloubkového obrazu. Knihovna OpenNI umožňuje získání obou obrazů, kdežto Kinect SDK od Microsoftu nabízí pouze hloubkový obraz. V sekci 3.2.1 jsou popsány a zhodnoceny obě metody kalibrace. Ze závěrů vyplývá, že kalibrace z infračerveného obrazu je lepší variantou. Z uvedených poznatků plyne, že pro experimentální systém bude vhodnější řešení poskytované frameworkem OpenNI, přestože Microsoft nabízí o něco lepší skeletální tracking.

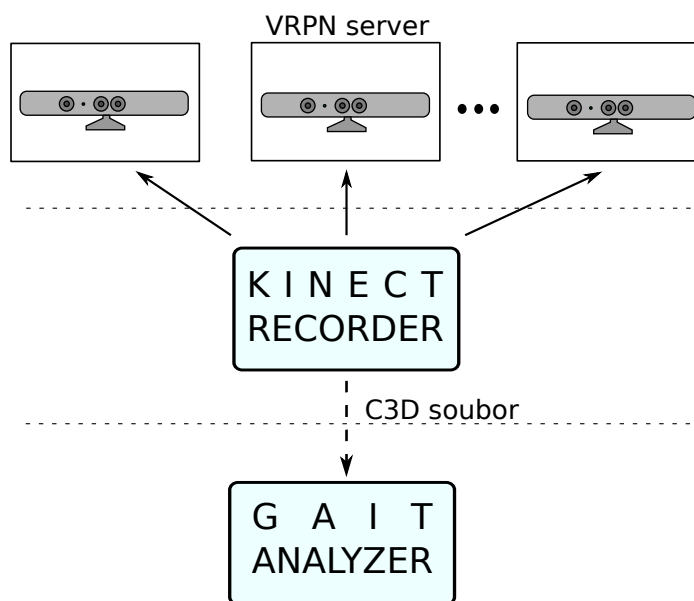
Se zapojením více zařízení Kinect k jednomu PC nastává jeden problém. Kinect je připojen k PC přes USB rozhraní, které má své limity co se rychlosti a přenosu dat týče. Je proto nutné, aby byl každý Kinect připojen k samostatnému USB řadiči. Pokud se připojí více Kinectů k jednomu USB řadiči, pak funguje pouze jeden z nich. Dnešní počítače bývají standardně vybaveny pouze dvěma řadiči. Toto omezení je třeba nějakým způsobem obejít. Možným řešením je osazení počítače hardwarem, který počet řadičů rozšíří. Lepší řešení nabízí knihovna VRPN (Virtual Reality Peripheral Network) [37], která implementuje síťové rozhraní mezi aplikací a množinou fyzických zařízení, jako jsou různé trackery atd. Tato technologie umožní zapojení většího počtu Kinectů pomocí více počítačů zapojených do počítačové sítě.

Nasnímaná data je třeba nějakým způsobem vizualizovat. Jde především o animaci 3D modelu reflektující pohyb snímané osoby. K tomuto účelu dobře poslouží multiplatformní herní engine Unity 3D [14], který umožňuje vytváření desktopových, mobilních ale dokonce i webových aplikací. Obrovskou výhodou tohoto systému je jeho portovatelnost na velké množství systémů a zařízení. Dále je nutné vizualizovat graf mapující kinematiku cyklu chůze. Pro kreslení grafů existuje velké množství již hotových řešení. Nejvhodnějším se ukázal být JavaScriptová knihovna Flot [4].

### 3.1 Komponenty systému

Na základě výše specifikovaného problému, se jevílo jako nejvhodnější rozdělit návrh systému do tří komponent, postihujících tři hlavní požadavky. Problematiku snímání pohybů pomocí více zařízení Kinect současně, řeší komponenta **VRPN server**. Záznam a sloučení dat, nasnímaných VRPN serverem, realizuje komponenta **Kinect Recorder**. Data obsahující parametry cyklu chůze zaznamenaná Kinect Recorderem je potřeba vhodně vizualizovat. K tomu slouží komponenta **Gait Analyzer**.

Pro potřeby kalibrace byla dále implementována jednoduchá aplikace **IRCapture**, jejíž jedinou schopností je záznam jednotlivých snímků infračerveného obrazu. Kalibrační nástroj pak umožňuje z těchto snímků provést kalibraci více zařízení Kinect (viz. sekce 3.2.1 *Snímání pomocí Kinectu - Kalibrace*). Kdykoliv se v následujícím textu hovoří o infračerveném obraze, je tento obraz pořízen právě aplikací IRCapture.

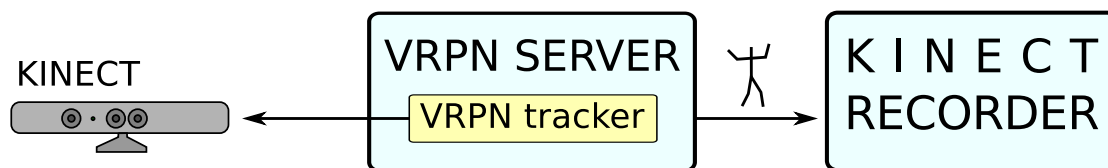


Obrázek 3.1: Schéma navrženého systému složeného z VRPN serverů, Kinect Recorderu a Gait Analyzátoru.

### 3.1.1 VRPN Server

Technologie VRPN je designována jako server-klient platforma. VRPN server zajišťuje síťovou komunikaci mezi systémem Kinect a klientskou aplikací Kinect Recorder. V terminologii VRPN se snímací zařízení nazývají *trackery*. Každý tracker má svou implementaci závislou na zvoleném hardwaru a používaných knihovnách. Tracker pak s aplikacemi komunikuje skrz VRPN server. Výhodou technologie VRPN je, že aplikace komunikuje pouze s VRPN serverem, který má obecné rozhraní. Aplikace tak nemusí vůbec znát implementaci samotného trackeru. Technologie VRPN je tedy dobrá nejen z hlediska rozšiřitelnosti množství zapojených Kinectů, ale zároveň umožňuje lehce měnit vnitřní implementaci trackeru, aniž by to výrazným způsobem mělo vliv na zbytek navrženého systému. Tato vlastnost je výhodná v okamžiku, kdy bychom chtěli zaměnit knihovnu NITE pro skeletální tracking za jiné řešení (např. za to od Microsoftu).

Účelem navrženého serveru (obrázek 3.2) je posílání nasnímané kostry do Kinect Recorderu, který data dále zpracovává. Součástí VRPN serveru je tracker implementovaný v jazyce C++, který ke snímání kostry využívá knihovnu OpenNI s komponentou NITE. Pomocí VRPN serveru se sítí přenáší informace o pozici a orientaci každého kloubu, jeho identifikačním čísle, časovém razítku kdy byla kostra spočítána a dále informace o věrohodnosti kloubu. Časové razítko slouží jako nástroj jednoduché synchronizace při slučování koster. Věrohodnost kloubu je schopnost komponenty NITE, která říká jak je výstup relevantní. V současné době se jedná o velice jednoduchou a omezenou verzi odhadu. Věrohodnost kloubu může nabývat hodnot 1, 0.5 a 0. Hodnota 1 značí, že výsledek je nejspíše dobře. Hodnota 0.5 značí, že výsledek je možná špatně a hodnota 0 značí, že určení pózy selhalo. Informace o věrohodnosti kloubu se přenáší také kvůli slučování koster z více Kinectů.



Obrázek 3.2: VRPN server komunikuje s Kinectem pomocí VRPN trackeru s knihovnou OpenNI. Nasnímaná data VRPN server zasílá Kinect Recorderu ke zpracování.

Při spouštění VRPN serveru je nutné zadat jméno trackeru a port na kterém bude vytvořeno síťové spojení. V inicializační funkci *InitTracker* (viz. kód 3.1) se pak vytvoří „naslouchající“ VRPN spojení na definovaném portu. Dále se vytvoří VRPN server s požadovaným jménem společně s trackerem a zaregistrují se callback funkce pro obsluhu událostí (nový uživatel ve scéně, zahájení kalibrace uživatele, (ne)úspěšná kalibrace uživatele, atd.) vyvolaných knihovnou OpenNI. Po inicializaci trackeru jsou v cyklu ve funkci *RunServer-Loop* volány hlavní smyčky VRPN serveru a síťového spojení implementované v knihovně VRPN. Tyto smyčky fungují na principu producent - konzument. VRPN server v každém cyklu smyčky vygeneruje novou zprávu s trajektoriemi kloubů. Síťové spojení v každém cyklu své smyčky kontroluje, zda není připravena nová zpráva a pokud je, odešle ji. Než však odešle první data, naslouchá síťové spojení na požadovaném UDP portu a čeká na připojení klientské aplikace Kinect Recorder. Po navázání spojení zasílá VRPN server nasnímaná data Kinect Recorderu. Struktura zasílaných dat je popsána na konci této kapitoly v sekci 3.3.1.

V průběhu implementace VRPN trackeru pro Kinect se vyskytla jedna významná překážka. Problém se týkal připojení více Kinectů k jednomu počítači. Zdrojem problému je chyba frameworku OpenNI, respektive implementace produkčního uzlu User Generator v komponentě NITE. Tato komponenta slouží ke generování dat popisujících snímaného uživatele. Chyba se projevovala tím způsobem, že výstup ze dvou zapojených Kinectů byl naprosto identický. Tzn. oba Kinecty vracely kostru ve stejné pozici a orientaci. Nalezení řešení tohoto problému zabralo poměrně dost času. UserGenerator v komponentě NITE je navržen podle návrhového vzoru singleton. Z tohoto důvodu nemůže být použit v jedné aplikaci pro obsluhu více Kinectů. Pro navržený VRPN server není singleton až tak velký problém, protože každý server se spouští jako samostatná instance procesu (aplikace). OpenNI definuje tzv. produkční uzly (production nodes), které generují data pro aplikaci. Vedle již zmiňovaného User Generatoru, existuje ještě několik dalších, jako např. Depth Generator, produkující hloubkovou mapu z hloubkového senzoru. Při inicializaci OpenNI kontextu je z těchto uzlů vytvořen celý strom. Z oficiální dokumentace jasně plyne, že při výčtu uzlů odpovídajících User Generátoru, bude počet uzlů odpovídat počtu připojených Kinectů. Tento předpoklad se ale nenaplnil, protože počet odpovídajících uzlů je významně vyšší. A toto je právě místo problému. Stačilo tedy vybrat uzel odpovídající „správnému“ User Generátoru. V různých zdrojích na internetu je diskutováno jaké číslo uzlu odpovídá kterému Kinectu. Závěr je ale takový, že pořadí produkčních uzlů je generováno podle počtu připojených Kinectů a téměř každý vývojář uvádí jiná pořadí.

Zdrojový kód 3.1: Funkce inicializující VRPN server

```

/**
 * @param userGenerator    UserGenerator generuje data popisující
 *                          rozpoznaneho uzivatele ve scene.
 * @return                  Vrací status inicializace.
 */
XnStatus InitTracker(UserGenerator &userGenerator)
{
    XnStatus rc = XN_STATUS_OK;
    // vytvoreni sitoveho spojeni
    g_Connection = new vrpn_Connection_IP(g_ServerPort);

    // vytvoreni VRPN serveru s trackerem
    g_Tracker = new vrpn_Tracker_Kinect(g_ServerName, g_Connection, &
        g_Context, userGenerator);

    // registrace OpenNI callback funkci
    rc = g_Tracker->RegisterCallbacks();

    return rc;
}

```

### 3.1.2 Kinect Recorder

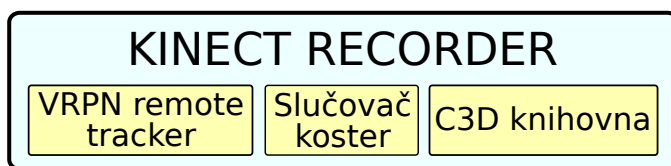
Navržená komponenta Kinect Recorder slouží ke zpracovávání všech nasnímaných koster, které přijme prostřednictvím síťového spojení od VRPN serveru. Jejím hlavním úkolem je transformace všech nasnímaných koster do jediné výsledné kostry a uložení celého záznamu do C3D souboru. Komponenta je navržena tak, aby bylo možné připojit libovolný počet zařízení Kinect. Kinect Recorder lze provozovat jak se systémem čítajícím více Kinectů, tak i se systémem s jedním Kinectem. Slučování koster je podmíněno systémem tvořeným alespoň dvěma Kinecty. Snímání jedním Kinectem je taktéž možné. V tomto případě se data neslučují, ale přímo exportují do C3D souboru.

Kinect Recorder je implementován v programovacím jazyce C++ jako desktopová aplikace. Jelikož se jedná o sofistikovanější nástroj, bylo nutné navrhnout grafické rozhraní, které umožní uživateli s aplikací rozumně interagovat. Pro tento účel byla zvolena knihovna Qt [13] a OpenGL [8].

Připojení komponenty Kinect Recorder k VRPN serveru společně se záznamem dat zajišťuje tzv. **VRPN remote tracker**. Transformaci nasnímaných virtuálních koster do jediné výsledné kostry realizuje **algoritmus na slučování koster**. Výsledná kinematika je uložena do binárního C3D souboru prostřednictvím vlastní **C3D knihovny**. Jednotlivé části komponenty Kinect Recorder (viz. obrázek 3.3) jsou detailně popsány v následujícím textu.

#### VRPN remote tracker

VRPN remote tracker implementuje klientskou část aplikace, která zpracovává veškerá přichodící data od VRPN serveru. Aplikace Kinect Recorder komunikuje s každým VRPN serverem prostřednictvím jednoho VRPN remote trackeru. Tzn. pro každý server se vytváří jeden



Obrázek 3.3: Kinect Recorder a jeho součásti.

klient. Při navazování spojení se serverem je nutné znát jméno serveru, IP adresu a port na kterém je server spuštěn. Tyto parametry jsou zapsány jako řetězec znaků ve tvaru *Jmeno-Trackeru@hostname:port* (např. *Tracker0@localhost:3838*) a dále předány implementaci VRPN spojení, která se postará o navázání komunikace se serverem. Smyslem VRPN remote trackeru není pouze komunikace s VRPN serverem, ale také záznam dat přijatých od serveru. Zaznamenávat data je možné ve dvou módech. První mód zaznamenává pouze poslední přijatou kostru, kdežto druhý mód zaznamenává veškerá data ve vymezeném časovém intervalu. Délka časového intervalu je definována interakcí s prvky uživatelského rozhraní aplikace. První mód se používá v okamžiku, kdy se zobrazuje pouze aktuální nasnímaná kostra a data se nenahrávají. Druhý mód umožňuje data nejen zobrazovat, ale také zaznamenávat k dalšímu zpracování.

Pokud se provádí snímání z více Kinectů, pak první provedená operace po ukončení nahrávání je oříznutí všech záznamů tak, aby začínaly ve stejný okamžik. VRPN server začíná generovat data až v okamžiku, kdy je virtuální kostra zkalibrována. Celý produkční řetězec společně se zmíněnou kalibrací snímané osoby je popsán v sekci 3.2. Je v podstatě pravidlem, že ke kalibraci nedojde u všech Kinectů ve stejný okamžik. Oříznutí několika prvních snímků umožní zarovnat všechny nahrané záznamy, což je dobrá počáteční podmínka k sloučení všech koster do jediné výsledné kostry.

### Algoritmus slučování koster

Jak již bylo zmíněno, ke slučování koster je nejprve nutné transformovat nasnímaná data do jednoho společného souřadného systému. K tomu se využívá kalibrační techniky popsané níže v sekci 3.2.1 *Kalibrace*. Díky znalosti transformace, můžeme pomocí vztahu (2.1), ze sekce 2.1.3, provádět převody nasnímaných koster do stejného souřadného systému. Na této myšlence je postaven algoritmus implementovaný v aplikaci Kinect Recorder, který umožňuje slučování koster.

Nasnímaný záznam je tvořen posloupností snímků. V každém snímku známe pozice a orientace jednotlivých kloubů, společně s časovým razítkem a věrohodností každého kloubu. Díky znalosti časového razítka je zaručena jednoduchá synchronizace, kdy se vybírají data s co nejbližším časovým razítkem. Algoritmus (viz. pseudokód 1) funguje následovně. V každém snímku pro každý kloub se transformují data ze všech Kinectů do stejného souřadného systému. Transformovaná data se poté vloží do pomocné prioritní fronty, která určuje pořadí dat na základě údaje o věrohodnosti kloubu. Výhodou tohoto řešení je, že nejlepší kandidáti pro výslednou kostru jsou na začátku fronty. Poté přichází na řadu slučování koster (viz. pseudokód 2), které se řídí následujícími pravidly podle věrohodnosti kloubu.



- Pokud je na začátku fronty záznam s věrohodností nula, pak žádný z Kinectů nebyl schopen rozpoznat pózu snímané osoby. Výsledná pozice a orientace kloubu je pak stejná jako v předcházejícím snímku.
- Pokud má první záznam fronty větší věrohodnost než záznam druhý v řadě, pak výsledná pozice a orientace odpovídá prvnímu záznamu.
- Pokud je na začátku prioritní fronty několik záznamů se stejným údajem o věrohodnosti, pak se mezi těmito daty interpoluje. V případě pozic kloubů se používá lineární interpolace, v případě orientace jde o sférickou lineární interpolaci.

---

**Pseudokód 1:** Slučování nasnímaných koster. Transformace dat do společného souřadného systému.

---

```

1 foreach frame f do
2   foreach jointId jID do
3     foreach calibration c do
4       joint ← getRecordedData (f, jID);
5       /* transformace kloubu do společného souřadného systému */
6       tJoint ← transformJoint(joint, c);
7       /* vložení transformovaných dat do fronty */
8       priorityQueue.push(tJoint)
9     end
10    /* sloučení všech kloubů do jediného výsledného */
11    mergeJoints(priorityQueue, jID, f); // viz. Pseudokód 2
12  end
13 end

```

---

### Knihovna pro práci s C3D soubory

Množství generovaných dat zařízením Kinect je poměrně velké. Vezmeme-li v potaz, že zařízení generuje virtuální kostru tvořenou 15-ti klouby ve frekvenci 30 Hz, pak deset sekund nahrávky tvoří 4500 záznamů. Přitom potřebujeme uchovat minimálně informaci o pozici a orientaci kloubu. Právě z důvodu velkého množství generovaných dat, byla navržené komponentě Kinect Recorder přidána schopnost ukládání a načítání zaznamenané kinematiky do binárního C3D souboru [32]. Jedná se o standardní datový formát, využívaný v biomechanice, animaci a v laboratorích chůze, k zaznamenávání třídimenzionálních a analogových dat. Binární verze formátu umožňuje, aby ukládání ale především načítání a práce s daty byla dostatečně rychlá. To, že Kinect Recorder zajišťuje jak ukládání tak načítání C3D souboru, má tu výhodu, že provedený dynamický záznam ze všech Kinectů je možné nejprve pouze uložit a teprve později se k datům vrátit a zpracovat je. Tato vlastnost je velice cenná při pořizování většího počtu dynamických záznamů. Snímaný subjekt tak nemusí čekat před dalším záznamem na zpracování pořízených dat.

Schopnost načítání a ukládání do C3D binárního souboru je realizována pomocí samostatné knihovny implementované v programovacím jazyce C++. Knihovna je naprogramována

---

**Pseudokód 2:** Slučování nasnímaných koster. Určení výsledné pozice a orientace.

---

```

Vstup : priorityQueue, jointID, frame
Výstup: mergedSkeleton // Finální pozice a orientace kloubu
1 begin
    // nejlepší věrohodnost ve frontě
2   bestConfidence = priorityQueue.top().jointConfidence;
3   priorityQueue.pop();
    /* Pokud je na začátku fronty záznam s věrohodností nula, pak
       výsledek je stejný jako v předchozím snímku. */
4   if bestConfidence == 0 then
5       | mergedSkeleton [frame] [jointID] ← mergedSkeleton [frame - 1] [jointID];
6       | return;
7   end
    /* Pokud má první záznam fronty větší věrohodnost než záznam druhý v
       řadě, pak výsledná pozice a orientace odpovídá prvnímu záznamu. */
8   if bestConfidence > priorityQueue.top().jointConfidence then
9       | mergedSkeleton [frame] [jointID] ← priorityQueue.top();
10      | return;
11  end
12  mergedSkeleton [frame] [jointID] ← priorityQueue.top();
13  priorityQueue.pop();
    /* Pokud je na začátku prioritní fronty několik záznamů se stejným
       údajem o věrohodnosti, pak se mezi těmito daty interpoluje. */
14  while priorityQueue is not empty() and priorityQueue.top().jointConfidence ==
      bestConfidence do
15      | interpolatePos(mergedSkeleton [frame] [jointID], priorityQueue.top());
16      | interpolateOri(mergedSkeleton [frame] [jointID], priorityQueue.top());
17      | priorityQueue.pop();
18  end
19 end

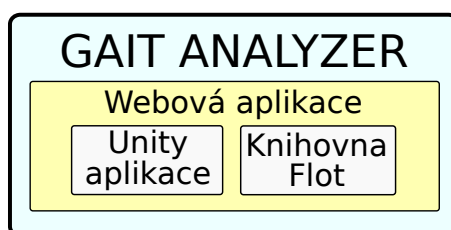
```

---

tak, aby obsah C3D souboru byl lehce rozšiřitelný a přitom odpovídal standardu C3D. Je tedy možné přidávat třídímenzionální a analogová data, ale také libovolné množství dalších parametrů (např. jméno trackeru, který záznam pořídil). Tím hlavním co se do C3D souboru ukládá jsou 3D pozice a orientace kloubů. Pozice kloubu odpovídá třídímenzionálním datům definovaným standardem C3D. Problémem se zdají být orientace kloubů, které jsou v kvaternionech. Tzn. jedná se o čtyřrozměrný vektor, který je nutné nějak transformovat na třírozměrný. Orientaci lze vyjádřit různými způsoby. Jelikož se jedná o třídímenzionální rotaci, lze ji rozložit na tři rotace v jednotlivých souřadných osách. Řešení jak uložit orientaci kloubu jako třídímenzionální data C3D souboru je poměrně jednoduché. Čtyřrozměrný kvaternion se převede na tři eulerovské úhly a tím získáme data, která už lze uložit do C3D souboru. Při načítání se postupuje opačně. Eulerovské úhly se převádí zpět na kvaterniony. Kromě třídímenzionálních dat, se do C3D souboru ukládají ještě časová razítka jednotlivých kloubů a to jako analogová data definovaná standardem C3D. Jak je vygenerovaný C3D soubor strukturován je popsáno v samostatné sekci 3.3.2 na konci kapitoly.

### 3.1.3 Gait Analyzer

Pro analýzu chůze je nezbytné poskytnout nástroj, který vhodným způsobem zobrazuje nasnímaný záznam jako animaci třídímenzionálního modelu s možností interakce (změna pohledu kamery, pozastavení animace, atd). Dále je třeba vizualizovat průběh orientace kloubů v anatomických osách jako křivku grafu v procentech cyklu chůze (tak jako na obrázku 2.8). Pro analýzu chůze byla navržena komponenta Gait Analyzer. Hlavním úkolem této komponenty je poskytnout dostatečnou vizualizaci kloubní kinematiky a všech kinematických proměnných. Vyšetřovaná data jsou uložena v binárním C3D souboru, který generuje výše popsaná aplikace Kinect Recorder.



Obrázek 3.4: Gait Analyzer a jeho komponenty.

Jak znázorňuje obrázek 3.4, Gait Analyzer je tvořen třemi částmi. První část, pojmenovaná **Unity aplikace**, řeší problém zobrazení 3D animace prostřednictvím multiplatformního herního engine Unity 3D [14], který umožňuje vytváření desktopových, mobilních ale i webových aplikací. Vizualizaci průběhu orientace kloubu v anatomických osách zajišťuje JavaScriptová **knihovna Flot** [4], která je druhou částí komponenty Gait Analyzer. Přenos dat a vzájemná komunikace mezi těmito dvěma součástmi je realizována prostřednictvím **webové aplikace**.

## Webová aplikace

Webová aplikace zajišťuje komunikaci mezi Unity aplikací a JavaScriptovou knihovnou Flot. Unity aplikace se spouští v Unity webovém přehrávači. Skrz tento přehrávač lze s Unity aplikací komunikovat pomocí JavaScriptu, prostřednictvím funkce *SendMessage*, která způsobí zavolání vnitřního obsahu aplikace. Z Unity aplikace lze zase volat JavaScriptový kód příkazem *Application.ExternalCall()*. Tímto způsobem může Unity aplikace předat data knihovně Flot, která v grafu zobrazí odpovídající křivku.

Jeden z problémů, který bylo třeba vyřešit, byl způsob jakým Unity aplikace načte C3D binární soubor. Z bezpečnostních důvodů není možné přímo přistupovat z webové aplikace Unity k lokálním souborům umístěným na disku počítače. Jediný možný způsob jak data do webové aplikace přenést, je nahrání do nějakého úložiště na webovém serveru a z tohoto místa teprve data stáhnout do Unity aplikace. Aby aplikace Gait Analyzer odpovídala standardům moderního webu, provádí se upload C3D souboru prostřednictvím technologie AJAX [2], která komunikuje s php skriptem na straně serveru. AJAX umožňuje upload souboru bez nutnosti opětovného načtení stránky. Pokud nedojde při uploadu souboru k chybě, je pomocí JavaScriptu zavolána Unity funkce *SendMessage()* (viz. kód 3.2). Tím dostane aplikace zprávu o tom, že si může stáhnout C3D soubor a zpracovat ho (viz. kód 3.3).

Zdrojový kód 3.2: JavaScriptová komunikace s Unity aplikací skrz internetový prohlížeč.

```
// získání objektu webového přehrávače Unity
var unity = GetUnity();
// zaslání zprávy o připravenosti C3D dat
// parametr object – objekt kterému se zpráva posílá
// parametr "InitC3dData" – funkce kterou objekt implementuje
// parametr url – vstupní parametr funkce InitC3dData
unity.SendMessage(object, "InitC3dData", url);
```

Zdrojový kód 3.3: Funkce pro stažení url obsahu do Unity aplikace.

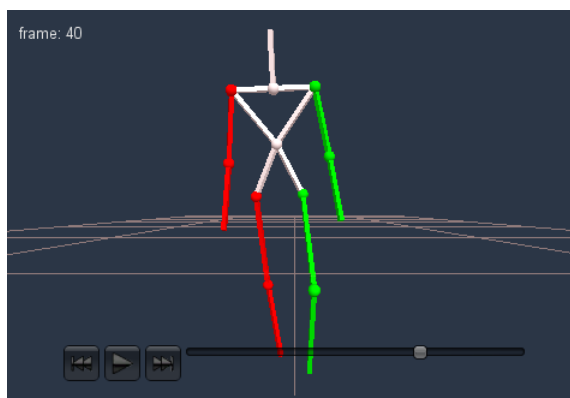
```
/// <param name="fileUrl">
///   Url ze kterého se má stáhnout C3D soubor.
/// </param>
IEnumerator InitC3dData(string fileUrl)
{
    // začne stahovat dane url
    WWW www = new WWW(fileUrl);

    // čekání na dokončení stahování
    yield return www;

    // zpracování staženého C3D souboru
    C3dStatus status = this.c3dReader.Init(www.bytes);
    if (status != C3dStatus.STATUS_OK)
        Debug.Log("Errorr code = " + status);
}
```

## Unity aplikace

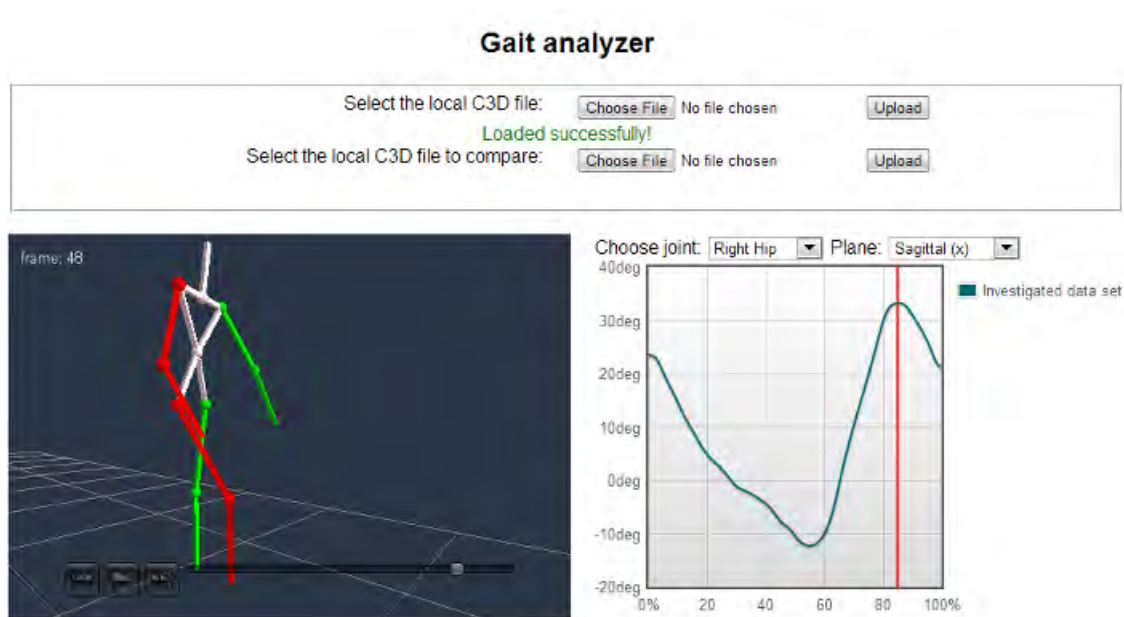
Aplikace postavené na herním engineu Unity mohou být implementovány v programovacích jazycích JavaScript, C# a Boo. Navržená aplikace je naprogramována v jazyce C# a jejím úkolem je načtení C3D souboru a zobrazení 3D animace nasnímané chůze. Bohužel Unity aplikace publikované ve webovém prohlížeči nemohou z bezpečnostních důvodů používat tzv. plugíny v podobě externích knihoven, tak jako desktopové aplikace Unity. Pokud by totiž byla aplikace Gait Analyzer desktopovou aplikací, bylo by možné použít již implementovanou knihovnu pro načítání C3D binárních souborů, kterou používá aplikace Kinect Recorder. Kvůli tomuto nedostatku bylo nutné C++ kód knihovny, pro načítání C3D souborů, přepsat do programovacího jazyka C# a integrovat ho do Unity aplikace.



Obrázek 3.5: Virtuální model kostry.

Když Unity aplikace načte C3D soubor, jsou data připravená k aplikaci na 3D virtuální model kostry (obrázek 3.5). Realizovaný model je tvořen kostmi a klouby. Každá kost je reprezentována kvádrem o délce odpovídající skutečné délce kosti snímaného subjektu. Pro lepší názornost jsou klouby virtuální kostry znázorněny kulovými objekty. Na začátku této kapitoly bylo zmíněno, že používaný framework OpenNI nabízí pouze pozice a orientace kloubů v souřadném systému světa (tzn. relativně vůči kameře). Především u rotací je to velký problém. Při analýze chůze jsou totiž důležité lokální orientace kloubů, nikoliv globální. Unity se v tomto případě stává velice cenným nástrojem a to z toho důvodu, že dokáže globální rotace transformovat na lokální. Transformaci lze provést aplikací globálních rotací na hierarchický model tvořený strukturou objektů. Tato hierarchická struktura musí samozřejmě odpovídat anatomické struktuře lidského těla. Tzn. například lidská noha má hierarchickou strukturu: kyčel > koleno > kotník.

Na hierarchický model se aplikují globální orientace, tím se model transformuje do nasnímané pózy, čímž se spočítají i lokální úhly. Potom již lze jejich průběh vizualizovat v grafu. Problém ovšem je, že velikosti segmentů hierarchického modelu nemusí odpovídat skutečným délkám segmentů těla. Přitom cílem je zobrazovat co nejvěrnější virtuální model reflektující snímaný objekt. Ve výsledné aplikaci je to řešeno tak, že model zobrazované kostry je vypočítáván pouze z nasnímaných pozic jednotlivých kloubů. Délka kosti je pak spočítána jako délka vektoru dvou sousedních kloubů a orientace kosti je dána směrem vektoru (viz. kód 3.5). Dále je v aplikaci pomocná hierarchická struktura, na kterou jsou aplikovány globální



Obrázek 3.6: Gait Analyzer. V levé části se zobrazuje animace chůze. Vpravo je graf vizualizující kinematické proměnné.

orientace (viz. kód 3.4). Tato hierarchická struktura se samozřejmě nezobrazuje. Po načtení C3D souboru jsou orientace na tuto hierarchii aplikovány a jsou předpočítány lokální úhly všech kloubů ve všech osách jako eulerovské úhly. Předpočítané eulerovské úhly jsou uloženy v paměti, aby se nemusely znovu počítat.

### Knihovna Flot pro vykreslování grafů

Vizualizace průběhu kloubní kinematiky je klíčovou částí systému. Graf popisující cyklus chůze slouží jako podklad pro analýzu. Způsob prezentace dat má proto obrovský vliv na výslednou interpretaci. K vizualizaci grafu dobře posloužila JavaScriptová knihovna Flot [4]. Knihovna Flot je poměrně populární, o čemž svědčí i velké množství rozšíření, které vytvořili samotní uživatelé. Vizuální stránku grafu lze jednoduše nastavovat pomocí parametrů při inicializaci. Použití je pak opravdu jednoduché. Stačí knihovně předložit dvoudimenzionální pole dat a knihovna se postará o zbytek. Tzn. o nastavení mřížky společně s popisky jednotlivých os, ale především o vykreslení samotného grafu. Výsledná aplikace je vyobrazená na obrázku 3.6.

Když Unity aplikace předpočítá lokální úhly kloubů, zavolá JavaScriptovou funkci *unityIsReady()*, umístěnou na webové stránce. Tím oznámí, že jsou data připravena. Webová stránka pak požádá Unity aplikaci o data pro konkrétní kloub a anatomickou osu. Unity data nejprve transformuje do požadované formy a poté zavolá JavaScriptovou funkci *setData(data)* (viz. kód 3.6) na webové stránce. Funkce *setData(data)* předá data knihovně Flot, která zobrazí korespondující graf.

Zdrojový kód 3.4: Funkce aplikující globální orientaci na pomocnou hierarchii.

```

/// <param name="jointId">Unikatní id kloubu.</param>
/// <param name="orientation">Orientace kloubu nactena z C3D</param>
private void ApplyRotation(int jointId, Quaternion orientation)
{
    // vypocet nove orientace
    Quaternion newRotation = orientation * initialRotations[jointId];
    // aplikace globalni orientace
    this.transforms[jointId].rotation = newRotation;
}

```

Zdrojový kód 3.5: Funkce zobrazující jeden segment kostry společně s kloubem.

```

/// <param name="jointStart">Kloub ze ktereho kost smeruje.</param>
/// <param name="jointEnd">Kloub do ktereho kost smeruje.</param>
/// <param name="bone">Objekt reprezentující kost.</param>
/// <param name="joint">Objekt reprezentující kloub.</param>
private void DrawLimb(int jointStart, int jointEnd, GameObject bone, ←
    GameObject joint)
{
    // transformace kosti do pozadovane pozice
    bone.transform.position = this.positions[jointStart];
    // vypocet smeru kam je kost namirena
    Vector3 dir = this.positions[jointEnd] - this.positions[jointStart];
    // zmena rotace kosti podle smeru
    bone.transform.rotation = Quaternion.LookRotation(dir);
    // uprava velikosti kvadru aby odpovídal delce kosti
    bone.transform.localScale = new Vector3(0.02, 0.02, dir.magnitude);

    // transformace pozice kloubu
    joint.transform.position = positions[jointStart];
}

```

Zdrojový kód 3.6: Nastavení dat grafu kinematiky kloubu.

```

/// <param name="data">Kinematika kloub.</param>
function setData(data)
{
    window.graphData = new Array();
    // transformace dat do 2D grafu
    for (var i = 0; i < data.length; i++)
        window.graphData.push([i, data[i]]);

    // nastaveni parametru grafu a jeho vykresleni
    window.plot.setData(
        {color: "rgb(0, 100, 123)", data: window.graphData});
    window.plot.setupGrid();
    window.plot.draw();
}

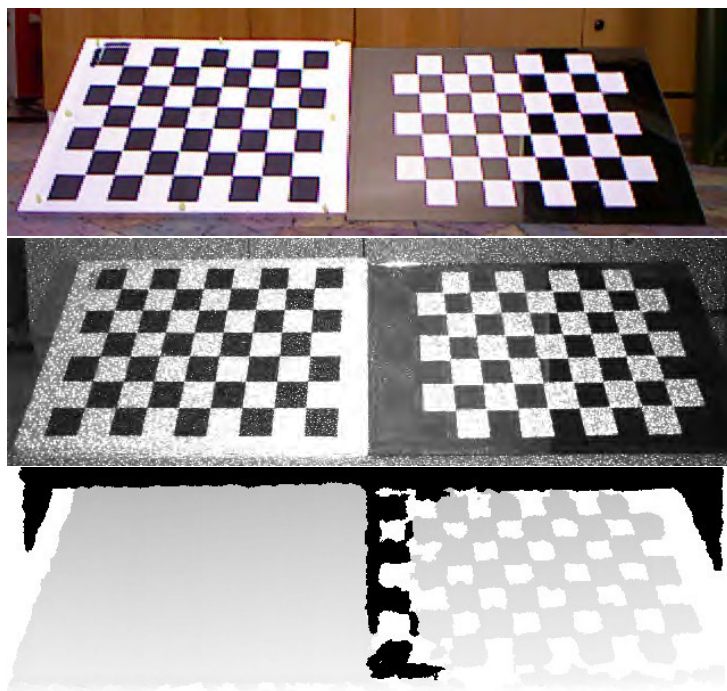
```

## 3.2 Snímání pomocí Kinectu

V této sekci je popsán produkční řetězec navrženého systému. Uvedené principy jsou obdobné jako v produkčním řetězci optických systémů, uvedeném v sekci 2.1.3. Nejprve je nutné udělat kalibraci systému. Poté je možné provést dynamický záznam a nakonec výsledky zanalyzovat.

### 3.2.1 Kalibrace

Kalibrace se provádí z toho důvodu, aby bylo možné slučovat nasnímané kostry z více zařízení Kinect. Před zahájením dynamického záznamu je tedy důležité tuto operaci pečlivě provést. Kalibrace se provádí pro hloubkový senzor, protože generuje data, ze kterých se určí póza snímaného subjektu. Oproti optickým systémům, kde se používá ke kalibraci hůlka s několika markery, používá navržený systém tzv. kalibrační šachovnici (obrázek 3.7). Šachovnicový vzor má tu výhodu, že pomocí algoritmů na detekci hran v obraze, lze poměrně jednoduše najít strukturu šachovnice. V oboru počítačového vidění je kalibrace kamery pomocí šachovnicového vzoru často používanou technikou. Existuje několik nástrojů implementujících algoritmus kalibrace. Pro účely navrženého systému byl zvolen Camera Calibration Toolbox for Matlab [20]. Jak již bylo zmíněno, framework OpenNI nabízí jak infračervený, tak i hloubkový obraz. Text [18], o markerless snímání pohybu, popisuje možnost kalibrace z hloubkového i infračerveného obrazu. Obě techniky byly otestovány.



Obrázek 3.7: Kalibrační šachovnice. Vlevo šachovnice vytištěná na papír. Vpravo šachovnice na zrcadle (pouze bílé čtverce). Shora dolů: RGB obraz, IR obraz, hloubkový obraz.



### Kalibrace pomocí hloubkového obrazu

Šachovnicový vzor vytištěný na papír se v hloubkovém obraze jeví jako jednolitá celistvá plocha. Je to dáno tím, že hloubkový senzor zaznamenává hloubku, nikoliv intenzity jasů v obraze. Berger a kol. [18] našli řešení jak i pomocí hloubkového senzoru nalézt šachovnicový vzor. Hloubkový senzor v každém pixelu měří vzdálenost objektů ve scéně. Bylo zjištěno, že odrazivé povrchy jako zrcadlo nebo aluminiová fólie, dokáží odrazit emitovaný paprsek z IR projektoru někam do prostoru, kde je změřena jiná než skutečná vzdálenost odrazivého povrchu. Pokud je tedy šachovnice složená ze dvou typů materiálů (difúzního a odrazivého), pak hloubkový senzor nasnímaná data interpretuje jako šachovnici (viz. obrázek 3.7).

Hloubkový obraz vzniká interpretací hloubkové mapy, která v každém pixelu nese 11-ti bitovou informaci o hloubce. Pokud je výsledný hloubkový obraz kódován do osmi bitového intenzitního obrázku, dochází samozřejmě ke ztrátám informace. Navíc všudypřítomný šum se na hranách s velkým rozdílem hloubek projevuje „chvěním“ pixelů v obraze. Výsledná hrana pak nemá tvar linie, ale „zubaté“ čáry. Interference IR senzoru s jinými zdroji infračerveného světla má na výsledný hloubkový obraz také vliv. Šachovnice v hloubkovém obraze proto nezachovává přesné linie tvořené šachovnicovým vzorem.

Před samotnou kalibrací se pořídí několik snímků kalibrační šachovnice v různých pozicích. Důležité přitom je, aby kalibrované Kinecty viděly celou šachovnici. V těchto snímcích se pak hledají rohy šachovnice. Jelikož hrany v obraze úplně přesně neodpovídají svislým a vodorovným liniím šachovnice, algoritmus kalibračního nástroje pro automatické nalezení rohů selhává. Ruční výběr rohů je zase náročný na čas a přesnost „naklikání“. Vstupem kalibračního algoritmu jsou nalezené rohové body. Výstupem pak vnitřní a vnější parametry kamery. Z několika pořízených snímků máme dostatečné množství korespondujících projekcí  $u_i$  každého rohu šachovnice  $X_i$ . Snahou je výpočet projekční matice kamery  $P$

$$u \simeq PX. \quad (3.1)$$

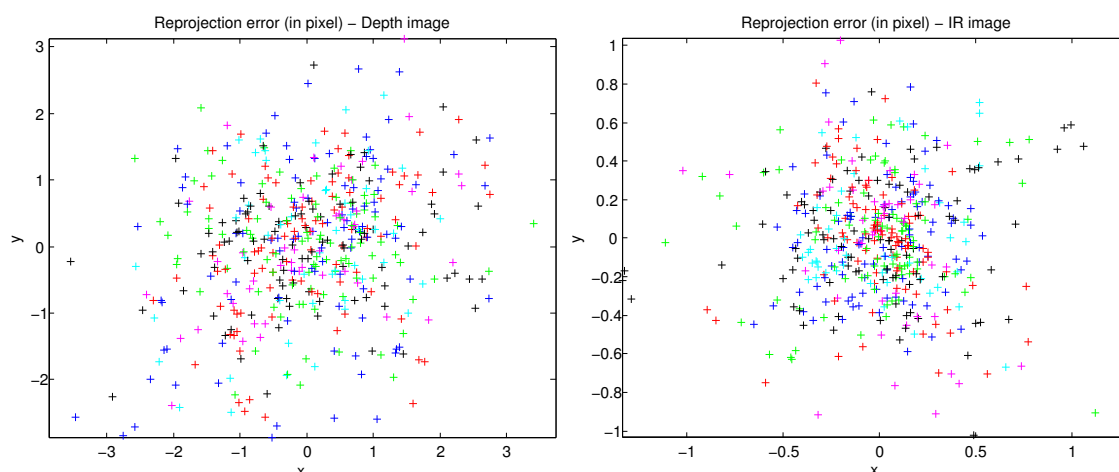
Dekompozicí projekční matice  $P$ , lze získat vnější  $(R, t)$  i vnitřní parametry kamery, obsažené v kalibrační matici  $K$

$$P = K [Rt]. \quad (3.2)$$

Díky obtížné detekci rohů v hloubkovém obraze, která má za následek nepřesnosti korespondencí, jsou výsledkem velké reprojekční chyby, jak ilustruje obrázek 3.8. Z tohoto důvodu není přesná ani výsledná kalibrace. A to je důvod proč navržený systém používá kalibraci z infračerveného obrazu.

### Kalibrace pomocí infračerveného obrazu

Při kalibraci pomocí infračerveného obrazu se stejně jako u hloubkového obrazu nejprve pořídí několik snímků kalibrační šachovnice v různých pozicích. Poté se pomocí kalibračního nástroje naleznou rohy v šachovnici každého snímku. Nejprve se kalibrace provádí pro každý Kinect samostatně. Tím získáme parametry každého Kinectu. Kromě kalibrace jedné kamery, umožňuje kalibrační nástroj kalibrovat také stereo systém (tvořený dvěma kamerami). Stereo systém se kalibruje ze dvou samostatných kalibrací. Soustavu několika Kinectů proto rozdělíme



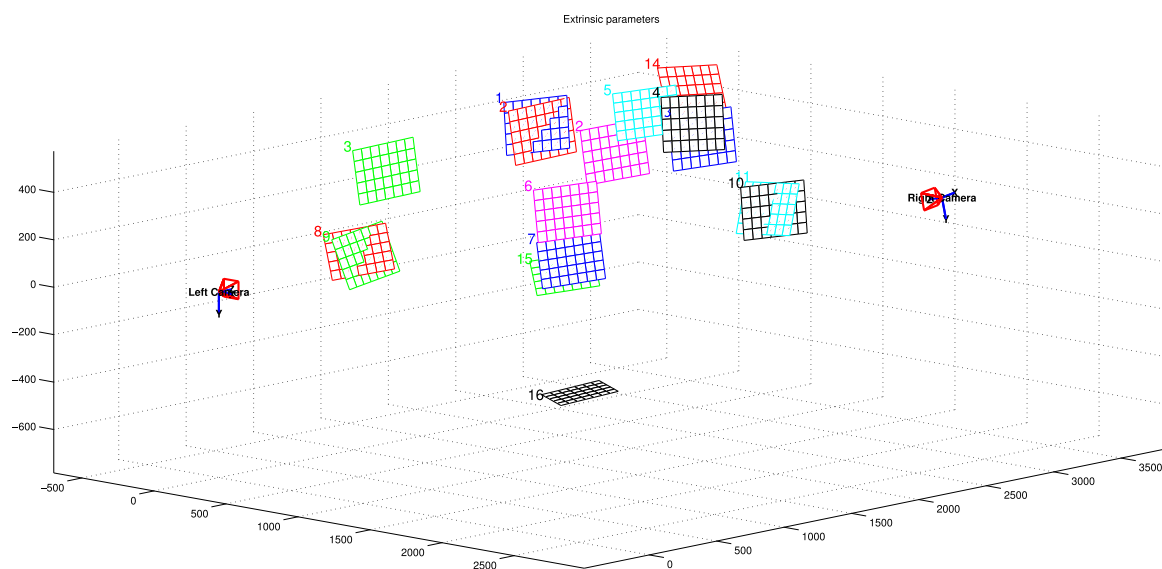
Obrázek 3.8: Reprojekční chyby v pixelech. Každá barva je přiřazena jednomu snímku v sérii. Levý obrázek znázorňuje reprojekční chyby v hloubkovém obraze. Vpravo jsou reprojekční chyby v infračerveném obraze. Kalibrace byla prováděna pomocí šachovnice umístěné ve stejných pozicích pro IR i hloubkový obraz. V tomto případě vykazuje kalibrace z hloubkového obrazu cca 3x větší chybovost reprojekcí.

po dvou a vždy dva sousední Kinecty kalibrujeme nezávisle na ostatních, jako by se jednalo o stereo kameru. Z kalibrace získáme vzájemnou polohu (viz. obrázek 3.9) těchto dvou Kinectů. Na obrázku 3.9 si čtenář může povšimnout pozice kalibrační šachovnice číslo 16. Pozice této šachovnice definuje počátek souřadného systému a orientaci souřadných os. Když se na počátku kalibruje každý Kinect zvlášť, než se přistoupí ke stereo kalibraci, je možné z této kalibrace získat transformaci mezi počátkem a pozicí kamery (viz. obrázek 3.10). To nám dovoluje definovat vlastní souřadný systém laboratoře chůze. Motivací tohoto postupu je, aby směr chůze snímání osoby byl schodný s jednou ze souřadných os. Podobný způsob se používá při kalibraci optického systému laboratoře chůze. Výběr počátku souřadného systému se provádí umístěním kalibrační hůlky na patřičné místo. Typicky to bývá jeden z rohů tlakové plošiny.

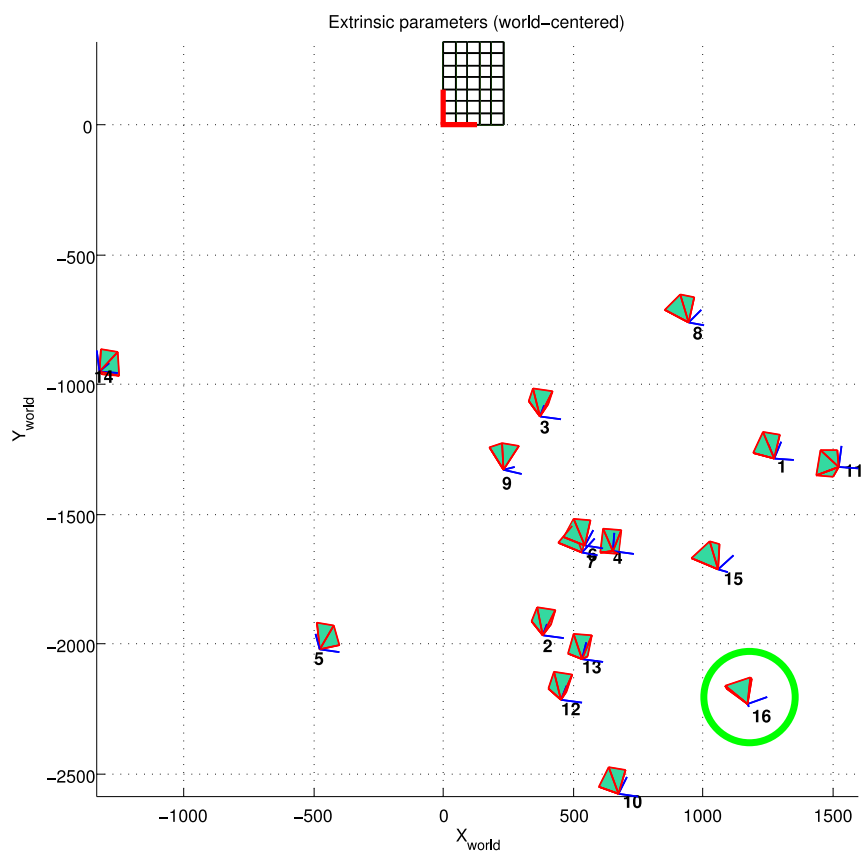
Vnější parametry kalibrace společně s transformacemi kamer do souřadného systému laboratoře, jsou uloženy do textového souboru. Textový soubor obsahuje vždy pouze kalibraci pro dva sousední Kinecty. Tento soubor je pak načten aplikací Kinect Recorder, která na základě uvedených parametrů provede transformaci dat a sloučení koster.

### 3.2.2 Dynamický záznam

Po kalibraci lze již provést dynamický záznam. Pro každý Kinect se spustí samostatný VRPN server. Každý server je v systému identifikován svým jménem, které je unikátní pro konkrétní IP adresu počítače, na kterém je server spuštěn. Například na IP adrese 192.168.1.2 mohou být spuštěny servery se jmény *Tracker0* a *Tracker1*. Nic ale nebrání, aby na jiné IP adrese běžely trackery se stejnými jmény. Poté se aplikace Kinect Recorder připojí ke všem VRPN serverům. Po úspěšném navázání spojení je vše připravené ke snímání. Dále se musí provést

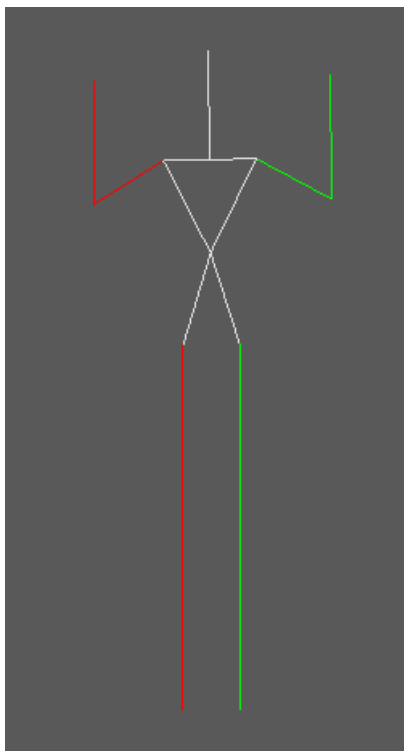


Obrázek 3.9: Vzájemná poloha dvou Kinectů spočtená po stereo kalibraci. Součástí obrázku jsou i pozice jednotlivých šachovnic. V tomto případě se kalibrace prováděla z 16-ti snímků.



Obrázek 3.10: Pozice levého Kinectu (kamera číslo 16) vzhledem k počátku s. s.

kalibrace snímané osoby. Kalibrace se provádí tak, že si subjekt stoupne do snímaného prostoru a provede kalibrační  $\psi$  pózu (viz. obrázek 3.11). Framework OpenNI se pak postará o automatické zkalibrování uživatele. Proces kalibrace zajistí správné označení jednotlivých kloubů. Tzn. identifikuje která končetina je levá, která pravá atd.



Obrázek 3.11: Kalibrační  $\psi$  póza.

Po úspěšné kalibraci se v aplikaci Kinect Recorder zobrazí náhledy virtuálních koster snímané osoby. Pak je možné pořídit záznam cyklu chůze. Kliknutím na příslušné tlačítko se aktivuje funkce nahrávání a je možné pohyby zaznamenávat. Nasnímá se chůze subjektu a opětovným stiskem tlačítka se nahrávání ukončí. Po ukončení záznamu je možné na časové ose zvolit výřez záznamu se kterým se bude pracovat. V ten okamžik je možné přikročit ke slučování nasnímaných koster.

Před sloučením se však musí načíst kalibrační soubor s parametry kalibrace. Kalibrační soubor vždy obsahuje kalibrační údaje pro dva sousední Kinecty. Při načítání kalibračního souboru je proto uživatel vyzván, aby stanovil jaké je pořadí Kinectů. Který je vlevo a který vpravo (z pohledu kamery). Tuto dodatečnou informaci potřebuje aplikace Kinect Recorder znát, aby mohla správně transformovat nasnímaná data. Spuštění procesu sloučení koster je namapováno na kliknutí na příslušné tlačítko grafického rozhraní aplikace. Sloučení do jediné výsledné kostry proběhne prakticky okamžitě. V současné implementaci nelze nastavovat žádným způsobem parametry slučování. Z tohoto důvodu je i GUI aplikace velmi jednoduché (viz. příloha B).

Vzniklou finální kostru lze nakonec vyexportovat do binárního C3D souboru. Tím jsou

data připravena k analýze v aplikaci Gait Analyzer.

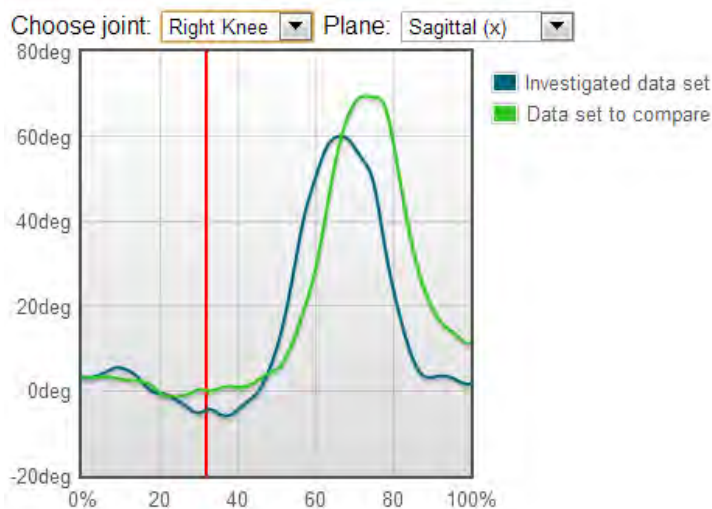
Při porovnání navrženého systému a optického systému, používaného v laboratořích chůze, nalezneme následující rozdíly. U navrženého systému se nemusí provádět statický záznam, protože kalibraci snímaného subjektu lze provést až při dynamickém záznamu. Snímaná osoba je totiž zkalibrována automaticky. K automatické kalibraci je však zapotřebí tzv. kalibrační póza. Oproti tomu optický systém zase potřebuje pomoci s označením jednotlivých markerů při statickém záznamu.

### 3.2.3 Analýza výsledku

K analýze výsledků slouží webová aplikace Gait Analyzer. Gait Analyzer nejprve načte C3D soubor s nasnímanou kinematikou a poté zobrazí animaci nasnímané chůze a graf popisující kinematické proměnné. Mezi jednotlivými proměnnými a anatomickými rovinami lze jednoduše přepínat.

Někdy je vhodné nasnímaná data porovnávat s jinými daty. Analýza chůze se často používá k zhodnocení výsledků operace pohybového aparátu. Proto se zaznamená cyklus chůze před a po operaci a oba cykly se pak porovnají. Při implementaci Gait Analyzeru se na to pamatovalo a proto je možné načíst ještě jeden C3D soubor se srovnávacími daty. Data ze srovnávacího souboru se vizualizují pouze v grafu. V animaci se druhá kostra nezobrazuje. Je to z toho důvodu, aby nedošlo ke zmatení interpretace dat. Animace chůze plní doplňující funkci. Slouží především k tomu, aby si vyšetřující osoba mohla dát data z grafu do kontextu s nasnímanou pózou.

Graf vizualizuje kinematická data jako křivku normalizovanou na procenta cyklu chůze. Hodnoty v grafu odpovídají stupňům flexe/extenze, abdukce/addukce a vnější a vnitřní rotaci kloubu.



Obrázek 3.12: Graf vizualizující kloubní kinematiku pravého kolena v sagitální rovině. Zde jsou navíc srovnávána data ze dvou záznamů.

### 3.3 Struktura dat

V této sekci je popsána struktura dat, se kterou pracují jednotlivé komponenty navrženého systému. Nejprve jsou popsána data posílaná VRPN serverem do sítě. Následuje popis struktury C3D souboru. Konec této kapitoly je věnován datům zobrazovaným v grafu cyklu chůze.

#### 3.3.1 Data posílaná VRPN serverem do sítě

Jak již bylo zmíněno, VRPN server zasílá klientské aplikaci Kinect Recorder nasnímané kostry prostřednictvím síťového spojení. Když OpenNI zaznamená novou pózu uživatele, tak se nejprve data transformují, aby odpovídala VRPN standardu. Podle standardu VRPN musí být 3D pozice kloubů v metrech. OpenNI ale vrací pozice kloubů v milimetrech. Před vygenerováním zprávy je tedy nutné provést jednoduchý převod jednotek z milimetrů na metry. OpenNI také uvádí orientaci kloubů jako rotační matici. To opět neodpovídá standardu VRPN, který pracuje s orientacemi v kvaternionech. Rotační matici lze lehce převést na kvaterniony (viz. kód 3.8). Po transformaci dat může VRPN server vygenerovat zprávu a poslat jí klientské aplikaci. Zpráva se generuje pro každý kloub a je zarovnaná na hranici po osmi bytech. Zpráva má následující strukturu:

Zdrojový kód 3.7: Struktura zprávy zasílané VRPN serverem klientské aplikaci.

```
int      jointId;           /* unikatni cislo kloubu */
float    jointConfidence;   /* verohodnost kloubu */
double   pos[3];           /* pozice kloubu ve 3D v metrech */
double   quat[4];          /* orientace kloub v kvaternionenach */
timeval  timestamp;        /* casove razitko */

// -----
struct timeval {
    long tv_sec              /* sekundy casoveho razitka */
    long tv_usec            /* mikrosekundy casoveho razitka */
}
```

Celková velikost zprávy je tedy 72 bytů ( $1 \times \text{integer} - 4$  byty,  $1 \times \text{float} - 4$  byty,  $7 \times \text{double} - 56$  bytů,  $1 \times \text{timeval} - 8$  bytů). Takovouto zprávu přijme Kinect Recorder od každého VRPN serveru s frekvencí maximálně 30 Hz (maximum co jeden Kinect zvládne).

Zdrojový kód 3.8: Funkce umožňující převod mezi rotační maticí a kvaternionem.

```
void q_from_matrix(Quat4f &destQuat, const float srcMatrix[3][3])
{
    destQuat.w = sqrtf(max(0, 1 + srcMatrix[0][0] + srcMatrix[1][1]
        + srcMatrix[2][2])) / 2;
    destQuat.x = sqrtf(max(0, 1 + srcMatrix[0][0] - srcMatrix[1][1]
        - srcMatrix[2][2])) / 2;
    destQuat.y = sqrtf(max(0, 1 - srcMatrix[0][0] + srcMatrix[1][1]
        - srcMatrix[2][2])) / 2;
    destQuat.z = sqrtf(max(0, 1 - srcMatrix[0][0] - srcMatrix[1][1]
```

```

+ srcMatrix[2][2])) / 2;
destQuat.x *= sign(destQuat.x*(srcMatrix[2][1] - srcMatrix[1][2]));
destQuat.y *= sign(destQuat.y*(srcMatrix[0][2] - srcMatrix[2][0]));
destQuat.z *= sign(destQuat.z*(srcMatrix[1][0] - srcMatrix[0][1]));
}

```

### 3.3.2 C3D data

C3D (Coordinate 3D) formát [32] je standardním typem souboru, který umožňuje zaznamenávat třídimenzionální a analogová data společně s dalšími parametry. To vše je přitom zabaleno do jednoho souboru. Výhodou tohoto formátu je snadná rozšiřitelnost souboru o další parametry. Tento typ souboru se často používá v biomechanice, laboratořích chůze, ale i v počítačové animaci. Zpracování tohoto typu souboru je základní vlastností navržených aplikací Kinect Recorder a Gait Analyzer. Gait Analyzer umí data z C3D souboru pouze načítat. Oproti tomu Kinect Recorder umí data nejen načítat, ale i ukládat. Data v C3D souboru jsou uložena v binární formě. Výhodou tohoto řešení je, že práce s daty je dostatečně rychlá. Nevýhodou je naopak nutnost implementace nástroje, který s daty v této formě umí pracovat.

Proto byla implementována samostatná knihovna pro načítání z a ukládání do C3D souboru. Řešení odpovídá standardu C3D. Každý vytvořený C3D soubor se sestává z několika konzistentních bloků paměti, které mají velikost 512 bytů. Tyto bloky tvoří následující tři sekce:

Sekce hlavičky (jeden 512 bytový blok).
Sekce parametrů (jeden nebo více 512 bytových bloků).
Sekce dat (jeden nebo více 512 bytových bloků).

Tabulka 3.1: Základní struktura C3D souboru.

#### Sekce hlavičky

Hlavička je vždy prvním blokem a tvoří jí 512 bytů. Blok paměti lze interpretovat jako 256 16-ti bitových slov. Popis jednotlivých 16-ti bitových slov je uveden v tabulce 3.2. Pokud by si však čtenář prohlédl obsah vygenerovaného C3D souboru aplikací Kinect Recorder, možná by byl u hodnot některých parametrů zmaten. Například hodnota druhého slova, odpovídající počtu 3D bodů, je dvakrát větší než skutečný počet kloubů snímané osoby. Jako bylo v sekci popisující aplikaci Kinect Recorder zmíněno, ukládá se kromě 3D pozice každého kloubu i jeho orientace. Proto je hodnota druhého slova dvakrát větší. Dále stojí za zmínku, že struktura analogových dat byla využita pro účely zaznamenání časového razítka. Přestože součástí navrženého systému není žádné analogové zařízení, systém analogová data generuje. Analogová data mají dva kanály. Jeden kanál odpovídá sekundám a druhý mikrosekundám časového razítka.

Slovo	Typická hodnota	Popis
1	0x5002 hex	Byte 1: Ukazuje na první blok sekce parametrů. Byte 2: Musí odpovídat hodnotě 0x50h. Jinak se nejedná o C3D soubor.
2	nn	Počet 3D bodů uložených v C3D souboru (tzn. počet uložených trajektorií).
3	nn	Celkový počet analogových měření na jeden 3D snímek. Tzn. počet analogových kanálů vynásobených počtem vzorků na kanál.
4	1	Číslo prvního snímku 3D dat.
5	nn	Číslo posledního snímku 3D dat.
6	0	Maximální interpolační mezera v 3D snímcích.
7-8	nnnn	3D scale faktor (číslo s plovoucí desetinou čárkou). Pokud je záporné, pak jsou data uložena s plovoucí desetinou čárkou.
9	nn	DATA_START - jedná se o číslo prvního bloku, kde jsou uložena 3D a analogová data.
10	nn	Počet analogových vzorků pro jeden 3D snímek.
11 - 12	30.000	Snímková frekvence 3D dat v Hz (číslo s plovoucí desetinou čárkou).
13 - 256	0	Rezervováno pro budoucí použití.

Tabulka 3.2: Struktura sekce hlavičky.

### Sekce parametrů

První byte sekce hlavičky říká, na kolikátém 512-ti bytovém bloku se nachází sekce parametrů. V této sekci jsou uloženy informace vztahující se k 3D a analogovým datům. Jedná se o důležité informace, které používá software aplikace k přístupu a zpracování dat obsažených v C3D souboru. Sekce parametrů je vždy tvořena skupinami parametrů. Na začátku sekce parametrů je ještě hlavička, která nese informaci o počtu bloků této sekce a typu procesoru. Typ procesoru určuje vnitřní formát dat uvnitř souboru. Po hlavičce následují záznamy se skupinami parametrů a samotnými parametry. Struktura dat popisující skupinu parametrů je popsána v tabulce 3.3. Znaménko druhého bytu definuje, zda se jedná o skupinu či parametr skupiny. Pokud je znaménko kladné následuje struktura parametru skupiny, která je popsána v tabulce 3.4. V opačném případě jde o skupinu.

### Sekce dat

C3D soubor je navržen k uchování 3D a analogové informace. Třídídimenzionální data společně s analogovými mohou být synchronizována v čase. V tom případě jsou vzorky 3D dat prokládány snímek po snímku s analogovými daty (viz. tabulka 3.5). Začátek sekce s daty je uveden v hlavičce C3D souboru. Konkrétně se jedná o deváté slovo s parametrem *DATA\_START*.

Třídídimenzionální data jsou v souboru organizována následujícím způsobem. Nejprve jsou uloženy tři hodnoty X, Y, Z odpovídající souřadnicím pozice kloubu. Pokud jde o orientaci kloubu, pak tyto tři hodnoty představují rotaci v jednotlivých osách. Poté následuje infor-



Byte	Délka (bytů)	Popis
1	1	Počet znaků jména skupiny.
2	1	ID skupiny (-1 až -127... vždy záporné).
3	N	Jméno skupiny (ASCII znaky).
3 + n	2	Offset v bytech ukazující na další skupinu/parametr.
3 + n + 2	1	Počet znaků popisu skupiny.
3 + n + 3	M	Popis skupiny (ASCII znaky).

Tabulka 3.3: Struktura skupiny parametrů.

Byte	Délka (bytů)	Popis
1	1	Počet znaků jména parametru.
2	1	ID skupiny do které parametr patří (+1 až +127).
3	N	Jméno parametru (ASCII znaky).
3 + n	2	Offset v bytech ukazující na další skupinu/parametr.
3 + n + 2	1	Délka v bytech pro každý element dat -1 pro data tvořená znaky 1 pro data tvořená byty 2 pro data tvořená integery (16-bit) 4 pro data s plovoucí desetinou čárkou
3 + n + 3	1	Počet dimenzí (0-7) parametru. Nula pokud se jedná o skalár.
3 + n + 4	D	Dimenze parametru.
3 + n + 4 + d	T	Data parametru.
3 + n + 4 + d + t	1	Počet znaků popisu parametru.
3 + n + 4 + d + t + 1	M	Popis parametru (ASCII znaky).

Tabulka 3.4: Struktura parametru skupiny.

mace o věrohodnosti kloubu. Tímto způsobem jsou vždy zapsána všechna 3D data v jednom snímku. Poté následují analogová data.

Analogová data, tzn. časová razítka odpovídající času vygenerování pózy, jsou organizována trochu jiným způsobem než 3D data. Analogový záznam může pro každý vzorek 3D dat obsahovat jeden nebo více analogových vzorků, které mohou být tvořeny jedním nebo více analogovými měřeními (kanály). Organizace dat je popsána v tabulce 3.6, kde „n“ je počet analogových měření na jeden snímek 3D dat. Toto číslo je uloženo v hlavičce souboru jako slovo 10. Konkrétně v implementaci knihovny pro ukládání a načítání C3D souboru, náleží pro jeden snímek 3D dat dva analogové vzorky - časové razítko v sekundách a mikrosekundách. Číslo „n“ odpovídá počtu kloubů nasnímané kostry.

3D data pro snímek 1
Analogová data pro snímek 1
3D data pro snímek 2
Analogová data pro snímek 2
⋮
3D data pro snímek 'n'
Analogová data pro snímek 'n'

Tabulka 3.5: Struktura 3D / analogových dat.

Časové razítko v sekundách	Hodnota kloubu 1
	Hodnota kloubu 2
	⋮
	Hodnota kloubu 'n'
Časové razítko v mikrosekundách	Hodnota kloubu 1
	Hodnota kloubu 2
	⋮
	Hodnota kloubu 'n'

Tabulka 3.6: Organizace analogových dat pro jeden snímek.

### 3.3.3 Graf cyklu chůze

Aplikace Gait Analyzer zobrazuje data načtená z C3D souboru jako animaci a graf zobrazující cyklus chůze. Pro animaci se přímo používají trajektorie kloubů uvedené v C3D souboru. Graf vizualizuje průběh rotace kloubu v čase. Vodorovná osa grafu odpovídá procentům cyklu chůze, kdežto svislá osa odpovídá stupňům rotace kloubu. V C3D souboru jsou uvedeny orientace kloubů v jednotlivých snímcích. Nasnímaný cyklus chůze může mít různý počet snímků. Počet snímků se odvíjí od rychlosti chůze snímané osoby. Abychom mohli data porovnávat, je nutné je transformovat do procent cyklu chůze.

K transformaci dat na procenta cyklu chůze používá aplikace Gait Analyzer lineární interpolaci. Lineární interpolace je prováděna tak, jak je naznačeno v pseudokódu 3. Cyklus chůze se zobrazuje v rozsahu od 0 - 100%. Tzn. chceme získat 101 hodnot, které se zobrazí v grafu. Spočteme si velikost kroku, která závisí na počtu snímků. Každou hodnotu procenta cyklu chůze v intervalu  $\langle 0, 100 \rangle$  vynásobíme velikostí tohoto kroku. Výsledkem je typicky nějaké desetinné číslo, které zaokrouhlíme nahoru a také dolů, čímž získáme dvě hodnoty mezi kterými pak lineárně interpolujeme. Pomocí knihovny Flot je na závěr výsledná křivka vyhlazena kubickou interpolací, čímž je částečně odstraněn šum v datech.

---

**Pseudokód 3:** Lineární interpolace dat pro použití v grafu cyklu chůze.

---

**Vstup** : jointData, joint, frameCnt**Výstup:** result // Interpolovaná data připravená pro zobrazení v grafu.

```
1 begin
2   stepSize ← frameCnt / 101;
3   for frame = 0 to 100 do
4     step ← stepSize * frame;
5     x0 ← ⌊step⌋;
6     x1 ← ⌈step⌋;
7     result[frame] ← interpolate(x0, x1, jointData[joint, x0], jointData[joint, x1],
8                               step) ; // viz. pseudokód 4
9   end
10 end
```

---

---

**Pseudokód 4:** Funkce *interpolate* provádějící lineární interpolaci dat.

---

**Vstup** : x0, x1, f0, f1, step**Výstup:** Interpolovaná data.

```
1 begin
2   // Pokud souřadnice mezi kterými se interpoluje jsou stejné, pak
   nemusíme interpolovat.
3   if x0 == x1 then return f0;
4   else
5     return f0 + ((f1 - f0) / (x1 - x0)) * (step - x0);
6   end
7 end
```

---



## Kapitola 4

# Experiment

Tato kapitola je věnována experimentálnímu ověření navrženého systému. Průběh experimentu je popsán hned v následující sekci Návrh experimentu. Dále následuje sekce s naměřenými daty. Konec kapitoly je věnován diskusi a vyhodnocení výsledku.

### 4.1 Návrh experimentu

Cílem experimentu je nejen ověření funkčnosti a použitelnosti navrženého systému, ale především zhodnocení nasnímaných dat zařízením Kinect. Nejprve bude nutné naměřit tzv. zdravou bázi, která stanoví přípustné intervaly zdravé chůze. Zdravá báze vznikne zaznamenáním cyklu chůze jedenácti participantů (dospělých osob ve věku 24 až 60 let). Dále se zaznamená chůze participanta s poruchou hybnosti. Bohužel se v době experimentu nepodařilo domluvit účast žádného spastika, proto bude osobu s poruchou hybnosti simulovat zdravý jedinec. Simulace bude probíhat tak, že participant ponese na levém rameni sportovní tašku obsahující závaží o hmotnosti 10 kg. Nerovnoměrné zatížení musí notně ovlivnit participantovu chůzi. Od systému se očekává, že nasnímaná data takto handicapované osoby budou zobrazena v grafu cyklu chůze zřetelně mimo intervaly zdravé báze.

#### Experimentální prostředí

Laboratoř chůze bude simulována obývací místností o rozměrech  $6 \times 5$  m (viz. obrazová příloha B). Místnost je prosvětlena jedním oknem. Experiment proto bude proveden ve večerních hodinách, aby bylo omezeno parazitní osvětlení, způsobené především slunečním svitem přicházejícím skrz okno. K experimentu budou dostupná dvě zařízení Kinect. Během testování bylo zjištěno, že nejlepší výsledky jsou dosahovány, když jeden Kinect snímá frontální rovinu těla a druhý sagitální. Obě zařízení proto budou v místnosti rozestavena tak, aby jejich pohledy svíraly přibližně pravý úhel. Oba Kinecty budou připojeny ke stejnému počítači.

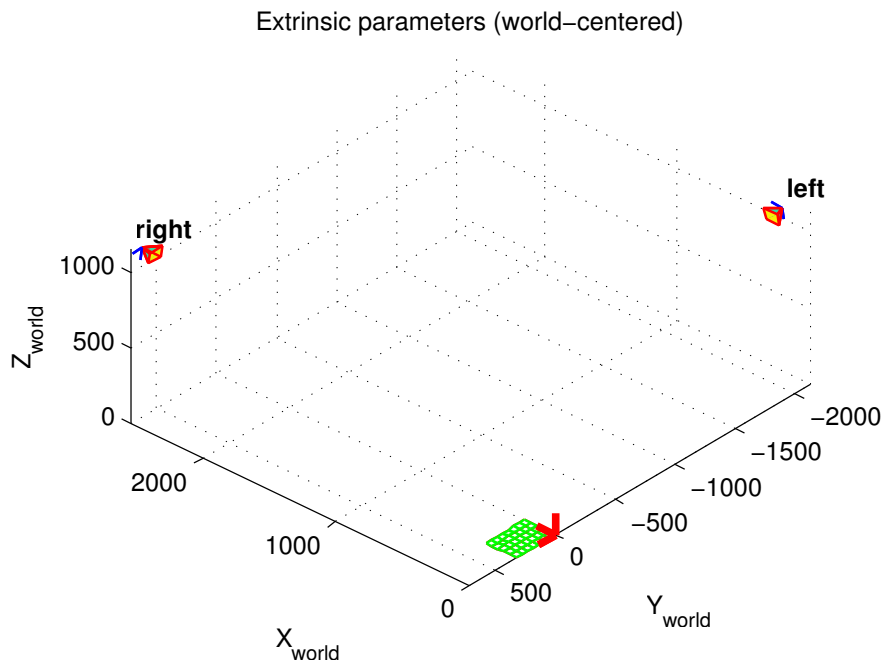
#### Scénář experimentu

V předchozí kapitole v sekci 3.2 *Snímání pomocí Kinectu* bylo zmíněno, že první krok před nasnímáním dat je kalibrace systému. Nejprve se tedy pomocí aplikace IRCapture zazname-

naří snímky infračerveného obrazu s kalibrační šachovnicí v různých pozicích. Tyto snímky umožní provést kalibraci systému nástrojem Camera Calibration Toolbox for Matlab. Z kalibrace se poté získají parametry pro slučování koster nasnímaných dvěma Kinecty. Po kalibraci se přistoupí k pořízení dynamického záznamu participantů. Pro každý Kinect se spustí jeden VRPN server a aplikací Kinect Recorder se zaznamená přibližně deset záznamů pro každého participanta (i pro simulujícího). Prvním úkolem participanta bude vstoupit do snímaného prostoru a zaujmout kalibrační pózu. Tím dojde ke zkalibrování participantovi virtuální kostry v systému. Dále bude mít participant za úkol udělat několik kroků v přímém směru tak, aby byl nasnímán kompletní cyklus chůze. Ze všech záznamů každého participanta se vybere jeden nejlepší. Nejlepším záznamem se rozumí záznam v jehož průběhu selhalo určení snímané pózy nejméněkrát. Na nejlepší záznam se pak aplikuje sloučení koster a výsledek se vyexportuje do C3D souboru. Dále se ze všech dat zdravé chůze vytvoří zdravá báze definována intervaly, jež jsou maximálními a minimálními hodnotami všech „zdravých“ dat. Na závěr se zdravá báze promítne v grafu cyklu chůze v aplikaci Gait Analyzer, ve které se načtou a vyhodnotí data chůze s deformitou.

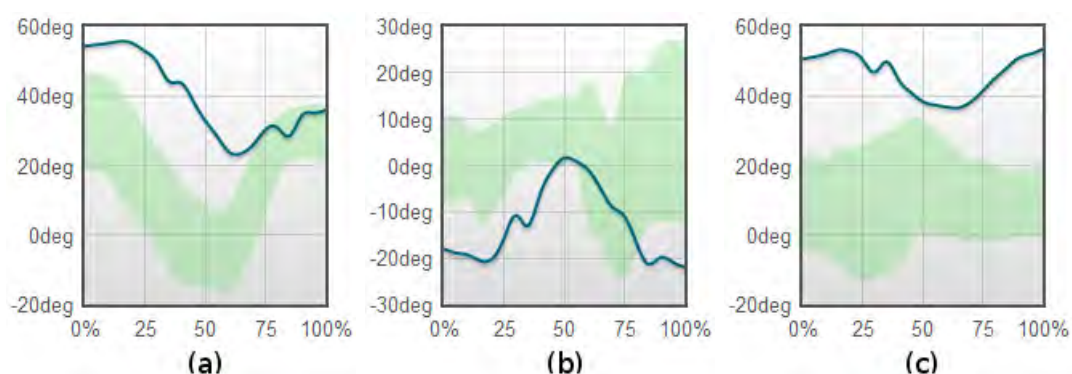
## 4.2 Naměřená data

Prvním krokem před nasnímáním dat byla kalibrace systému. Kalibrace se prováděla z šestnácti dvojic infračerveného obrazu. Výsledkem kalibrace jsou vnější parametry obou Kinectů vůči počátku souřadného systému, jak znázorňuje níže uvedený obrázek 4.1. Výsledek koreponduje s navrženým rozestavením Kinectů.

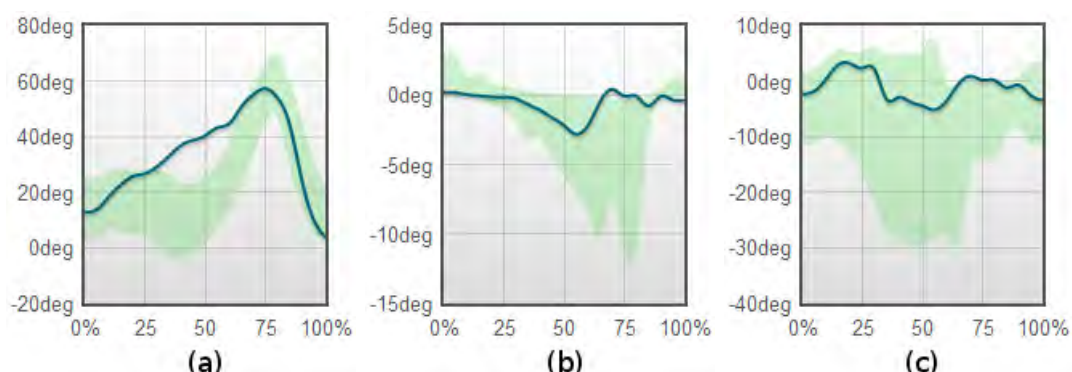


Obrázek 4.1: Pozice Kinectů vůči počátku získané po kalibraci systému.

Framework OpenNI umožňuje v dolní polovině lidského těla snímat pouze orientace kyčelního a kolenního kloubu. Orientace kotníku je pouze odhadována a nemusí proto odpovídat skutečné póze kloubu. Z tohoto důvodu jsou níže uvedena naměřená data pouze pro kyčelní a kolenní kloub. Pro demonstraci schopností systému jsou prezentovány naměřené hodnoty simulujícího participanta v porovnání se zdravou bází pouze pro levou dolní končetinu. To proto, že levá strana těla simulujícího participanta je zatížena závažím. Horní řada grafů odpovídá kinematickým proměnným kyčelního kloubu. Dolní řada odpovídá kolennímu kloubu. Zelená oblast odpovídá zdravé bázi. Modrá křivka odpovídá cyklu participanta s omezenou hybností. Vodorovná osa grafu koresponduje s procenty cyklu chůze. Pro připomenutí prvních cca 60% cyklu je fáze opory. Zbýlých cca 40% je fáze švihů. Svislá osa koresponduje se stupni natočení kloubu v dané rovině.



Obrázek 4.2: Cyklus chůze levé kyčle. (a) sagitální rovina, (b) transverzální rovina, (c) frontální rovina. Zelená oblast odpovídá zdravé bázi. Modrá křivka odpovídá cyklu chůze participanta s omezenou hybností.

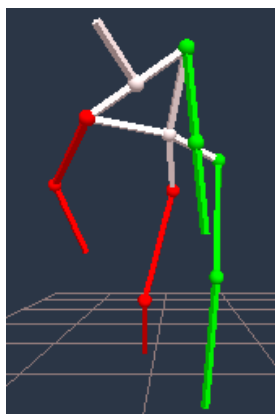


Obrázek 4.3: Cyklus chůze levého kolena. (a) sagitální rovina, (b) transverzální rovina, (c) frontální rovina. Zelená oblast odpovídá zdravé bázi. Modrá křivka odpovídá cyklu chůze participanta s omezenou hybností.

### 4.3 Vyhodnocení

Celý experiment proběhl podle navrženého scénáře. V průběhu experimentu se nevyskytly žádné problémy s funkčností ani s použitelností navrženého experimentálního systému. Návrh a realizaci komponent systému lze tedy shledat jako správný a v praxi fungující.

Z grafů na obrázcích 4.2 a 4.3 je zřetelně patrné, že průběh cyklus chůze simulujícího participanta se vychyluje z mezí zdravé báze. Vychýlení se koná především v cyklu chůze kyčle. Závaží na levém rameni způsobilo snahu o vyvážení, což vedlo k naklonění celé horní poloviny těla vpravo. Zejména z grafu kyčelní abdukce/addukce ve frontální rovině (obrázek 4.2 (c)) lze vyčíst k jak moc velkému vychýlení muselo dojít. To dokládá i obrázek 4.4 z animace nasnímané chůze. Vizualizací zřetelného vychýlení byl splněn požadavek na systém, aby nasnímaný cyklus chůze handicapované osoby byl zobrazen výrazně mimo intervaly zdravé báze.



Obrázek 4.4: Snímek z animace cyklu chůze participanta simulujícího poruchu hybnosti.

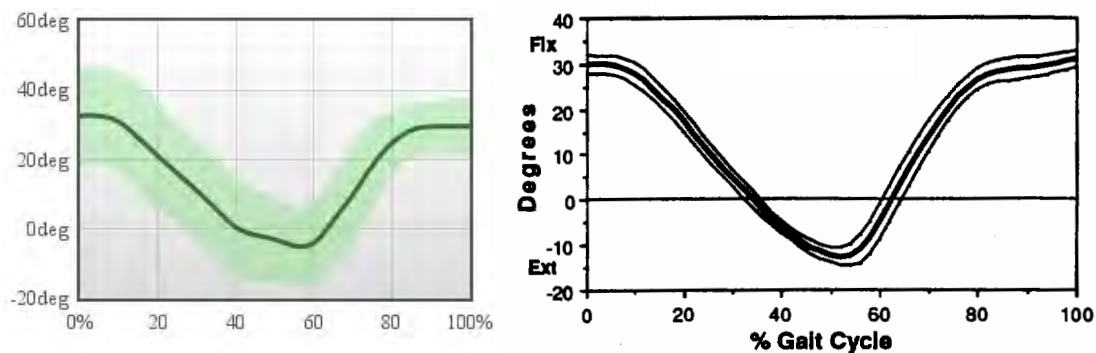
Obrázek 4.5: Participant simulující poruchu hybnosti s taškou obsahující 10 kg závaží.

Na tomto místě je dobré zhodnotit přesnost naměřených dat. Naměřenou zdravou bází porovnáme s jinou referenční zdravou bází, čímž získáme alespoň základní představu o přesnosti. K tomuto účelu dobře poslouží výsledky zveřejněné v článku [27], kde bylo 40 dospělých subjektů podrobena analýze chůze pomocí optického MoCap systému Vicon. Z naměřených dat byla sestavena zdravá báze. Na obrázcích 4.6 až 4.11 jsou pro porovnání vedle sebe umístěny dvojice grafů korespondujících zdravých bází. Levý graf popisuje naměřená data získaná navrženým systémem, pravý graf znázorňuje parametry chůze ze zmíněného článku. Křivka v levém grafu by se tedy měla co nejvíce blížit referenční křivce v pravém grafu. Z grafů je patrné, že především v sagitální rovině (tzn. při pohledu na chůzi z boku) se křivka naměřených dat nejvíce blíží referenční křivce zdravé báze. V této rovině jsou naměřené hodnoty celkem uspokojivé. Zdá se, že osa kloubu s největší volností pohybu je snímána celkem přesně. V transversální a frontální rovině jsou změny orientace jemnější a dosahují nižších hodnot. V těchto rovinách se projevuje nepřesnost systému Kinect. Podle referenční křivky by měl být průměrný rozsah pohybu v transversální a frontální rovině do přibližně patnácti stupňů. Navržený systém zaznamenal podobné rozsahy, avšak průběh křivky cyklu chůze ani dosažené stupně rotací kloubu se s referenční křivkou neshodují. Kinect bohužel není schopen přesně zaznamenat takovéto jemné niance. Nepřesnost zařízení je viditelná i na rozsazích

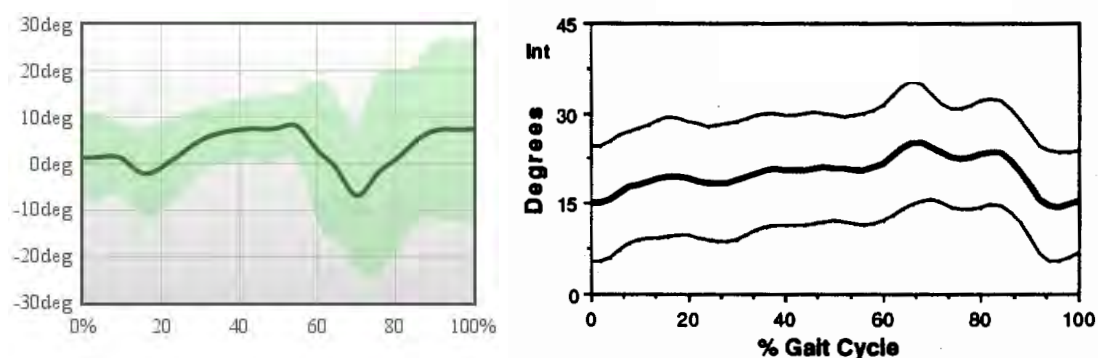


zdravé báze a to ve všech rovinách. I u sagitální roviny, kde je přesnost největší, je velikost oblasti zdravé báze mnohem větší než u referenční zdravé báze. Je však důležité zmínit, že prezentovaná data v korespondujících grafech nebyla získána na stejné skupině subjektů. Je tedy třeba počítat s určitou mírou zkreslení mezi korespondujícími grafy. Ovšem i tak by měly být průběhy cyklů chůze obdobné.

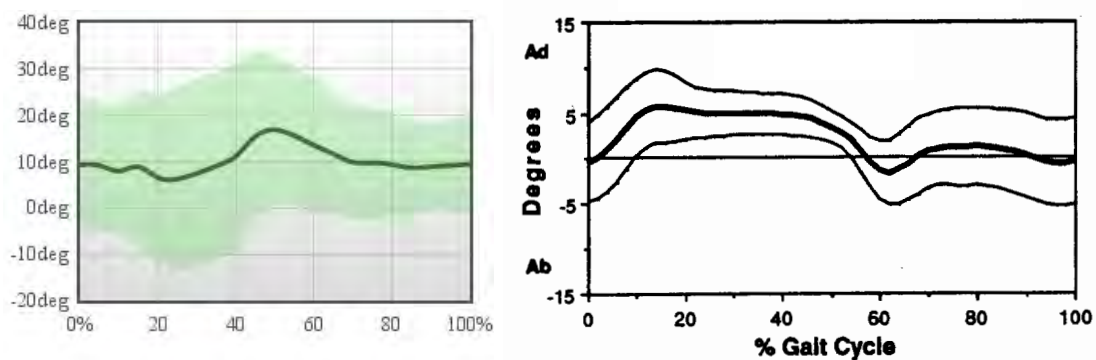
Z výše uvedených poznatků bohužel vyplývá, že zařízení Kinect není vhodné pro seriózní analýzu chůze. Zapojení více zařízení do systému nepřináší až takovou výhodu jak by se na první pohled čekalo. Více zařízení sice omezí problém sebe-zakrývání, ale přesnost nasnímaných dat zásadně nezvýší. Limitující se přitom nezdá ani tak snímková frekvence 30 Hz, ale spíše přesnost určení pózy z hloubkového obrazu. Ta je zásadní. Jak bylo zmíněno, nejpresněji jsou zaznamenány orientace kloubu v ose s největší volností pohybu. To odpovídá i primárnímu účelu zařízení Kinect - hraní her. U hraní her je důležité zachytit významné změny pózy hráčova těla. Zda se koleno v transversální rovině otočí o deset stupňů není až tak důležité. Ovšem v analýze chůze s aplikací v medicíně se jedná o podstatný detail, který může ovlivnit interpretaci dat a s tím i postup léčby pacienta s poruchou pohybového aparátu.



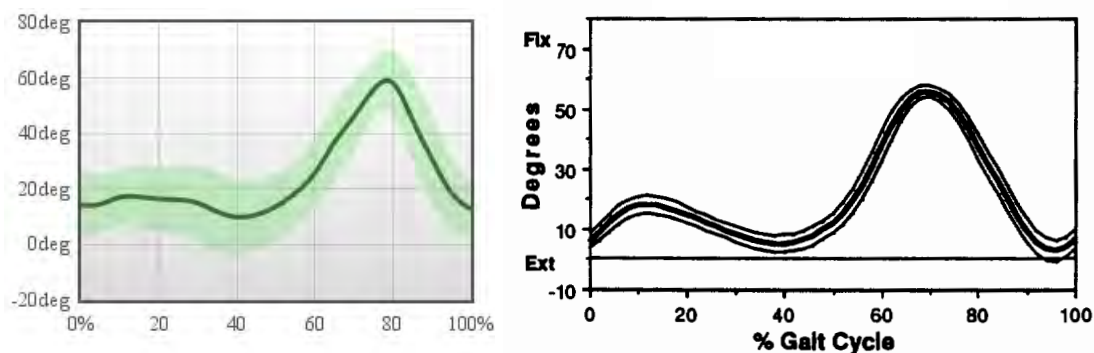
Obrázek 4.6: **Kyčelní flexe - extenze**. Sagitální rovina. Vlevo zdravá báze naměřená navrženým systémem, vpravo referenční zdravá báze (převzato z [27]). Tučná čára značí průměr, tenčí čára (respektive zelená plocha) značí standardní odchylku.



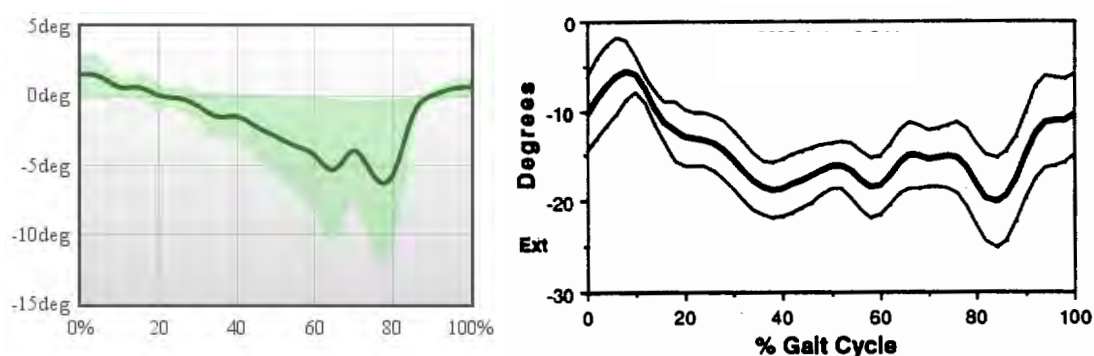
Obrázek 4.7: **Kyčelní rotace**. Transverzální rovina. Vlevo zdravá báze naměřená navrženým systémem, vpravo referenční zdravá báze (převzato z [27]). Tučná čára značí průměr, tenčí čára (respektive zelená plocha) značí standardní odchylku.



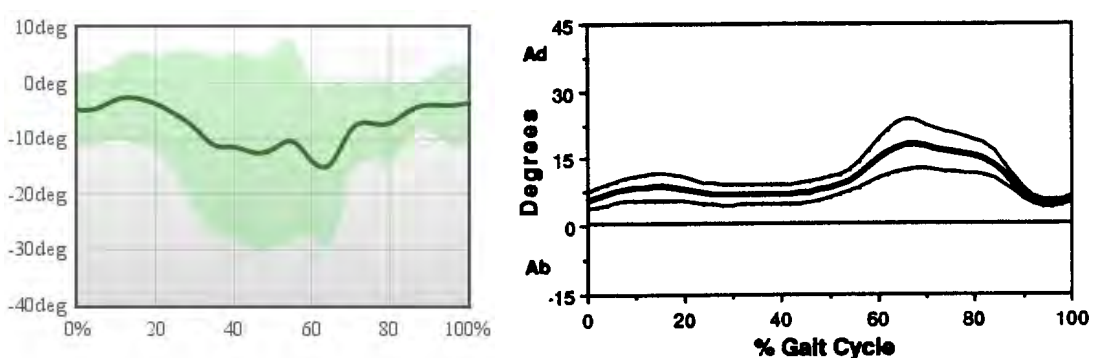
Obrázek 4.8: **Kyčelní abdukce - addukce**. Frontální rovina. Vlevo zdravá báze naměřená navrženým systémem, vpravo referenční zdravá báze (převzato z [27]). Tučná čára značí průměr, tenčí čára (respektive zelená plocha) značí standardní odchylku.



Obrázek 4.9: **Kolenní flexe - extenze**. Sagitální rovina. Vlevo zdravá báze naměřená navrženým systémem, vpravo referenční zdravá báze (převzato z [27]). Tučná čára značí průměr, tenčí čára (respektive zelená plocha) značí standardní odchylku.



Obrázek 4.10: **Kolenní rotace**. Transverzální rovina. Vlevo zdravá báze naměřená navrženým systémem, vpravo referenční zdravá báze (převzato z [27]). Tučná čára značí průměr, tenčí čára (respektive zelená plocha) značí standardní odchylku.



Obrázek 4.11: **Kolenní abdukce - addukce**. Frontální rovina. Vlevo zdravá báze naměřená navrženým systémem, vpravo referenční zdravá báze (převzato z [27]). Tučná čára značí průměr, tenčí čára (respektive zelená plocha) značí standardní odchylku.



# Kapitola 5

## Závěr

Diplomová práce se zabývá možnostmi využití systému Microsoft Kinect pro účely analýzy lidské chůze s aplikací v lékařství. Chůze je velice komplexní pohyb, který je dán souhrou centrálního nervového systému společně s pohybovým aparátem. Přesně zaznamenat a poté vizualizovat veškeré parametry je pro analýzu chůze klíčové. Na základě těchto požadavků byl celý problém snímání pohybu, společně s analýzou chůze rozebrán a v práci popsán. Díky tomu mohl být navržen a implementován systém umožňující záznam a analýzu chůze pomocí jednoho nebo více zařízení Microsoft Kinect. Funkčnost implementovaného systému byla na závěr experimentálně ověřena a získaná data vyhodnocena.

## Zhodnocení

V rámci práce se povedlo navrhnout a implementovat systém, který umožní zaznamenat chůzi a vizualizovat její parametry v takové podobě, která se standardně používá při její analýze. Realizovaný systém je schopen díky technologii VRPN pracovat s libovolným počtem zařízení Microsoft Kinect, přičemž byl testován s maximálně dvěma zařízeními. Pomocí realizovaného systému bylo úspěšně nasnímáno jedenáct participantů, z jejichž chůze byla stanovena referenční zdravá báze. Dále byl nasnímán participant simulující člověka s poruchou pohybového aparátu. Jeho cyklus chůze byl v grafu vizualizován zřetelně mimo meze zdravé báze. Tím systém splnil požadavek na rozlišení mezi normální a deformovanou chůzí.

Ze závěrů experimentu bohužel vyplývá, že zařízení Microsoft Kinect není vhodné pro seriózní analýzu chůze. Při porovnání dat pořízených navrženým systémem a dat uvedených v práci [27], využívající optickým systémem Vicon, bylo zjištěno, že průběh cyklu chůze je shodných vlastností pouze v sagitální rovině, kde jsou změny rotací kloubů největší. V ostatních rovinách jsou změny tak malé, že je algoritmus určení pózy není schopný z hloubkového obrazu přesně interpretovat. Navíc framework OpenNI s komponentou NITE neumožňuje nezávislé určení orientace kotníku. To je také velký problém, zejména v případě analýzy chůze.

Jediné co se nepodařilo uskutečnit bylo přímé porovnání navrženého systému se systémem ve skutečné laboratoři chůze. Zajistit takové porovnání je v českých podmínkách obtížné z důvodu malého počtu laboratoří. Analýza chůze se v klinické praxi v ČR používá pouze v jediné laboratoři a to v Brně. Jelikož pojišťovny stále nehradí vyšetření v laboratoři chůze,

musí se finance na provoz laboratoře složitě obstarávat. Což je také jeden z důvodů proč je obtížné se do laboratoře pro účely testování dostat.

## Možné pokračování práce

V době dokončování práce vydal Microsoft novou verzi svého Kinect SDK. Velkou novinkou oproti předchozí verzi je zpřístupnění infračerveného obrazu. Díky tomu bude možné provádět kalibraci více zařízení tak jako s řešením od OpenNI. Navíc Kinect SDK nabízí lepší skeletální tracking co se počtu snímaných kloubů týče. Především umožňuje snímat skutečné změny orientace kotníku. Bylo by tedy zajímavé porovnat přesnost stávajícího řešení s řešením od Microsoftu.

Pro precizní ověření přesnosti skeletálního trackování systému Kinect, by bylo také dobré porovnat nasnímaná data se systémem z reálné laboratoře chůze na stejném vzorku lidí.

Na konci roku 2013 by měla přijít na trh nová herní konzole Xbox 720. Proslýchá se, že s ní přijde na trh i druhá verze Kinectu. Na internetu se dokonce objevil údajný snímek z hloubkového obrazu s vyšším rozlišením, na němž je možné dokonale rozeznat jednotlivé prsty ale i obrysy oblečení. Pokud by Microsoft přišel s něčím takovým, byla by to senzace. S přesnějším hloubkovým senzorem by se jistě zvýšila i přesnost určení snímané pózy, což by mohlo vést k lepším výsledkům při analýze chůze.

# Literatura

- [1] About.com Inventors [online]. Citováno 18.12. 2012, <<http://inventors.about.com/>>.
- [2] AJAX - Asynchronous JavaScript and XML [online]. Citováno 18.12. 2012, <<http://www.w3schools.com/ajax/>>.
- [3] Belmont University, Gordon E. Inman College of Health Sciences and Nursing [online]. Citováno 18.12. 2012, <<http://www.belmont.edu/healthsciences/>>.
- [4] Flot - Attractive JavaScript plotting for jQuery [online]. Citováno 18.12. 2012, <<http://www.flotcharts.org/>>.
- [5] Inventor of the week archive [online]. Citováno 18.12. 2012, <<http://web.mit.edu/invent/iow/edgerton.html>>.
- [6] Meta motion [online]. Citováno 18.12. 2012, <<http://www.metamotion.com/>>.
- [7] Microsoft kinect [online]. Citováno 18.12. 2012, <<http://www.microsoft.com/en-us/kinectforwindows/>>.
- [8] OpenGL [online]. Citováno 18.12. 2012, <<http://www.opengl.org/>>.
- [9] OpenKinect [online]. Citováno 18.12. 2012, <<http://openkinect.org>>.
- [10] OpenNI [online]. Citováno 18.12. 2012, <<http://openni.org>>.
- [11] Phasespace [online]. Citováno 18.12. 2012, <<http://www.phasespace.com/>>.
- [12] PrimeSense Natural Interaction [online]. Citováno 18.12. 2012, <<http://www.primesense.com>>.
- [13] Qt [online]. Citováno 18.12. 2012, <<http://qt.digia.com/>>.
- [14] Unity 3d - game engine [online]. Citováno 18.12. 2012, <<http://unity3d.com/>>.
- [15] XBox 360 [online]. Citováno 18.12. 2012, <<http://www.xbox.com>>.
- [16] Xsens [online]. Citováno 18.12. 2012, <<http://www.xsens.com/>>.
- [17] Christian J. Barton, Pazit Levinger, Kay M. Crossley, Kate E. Webster, and Hylton B Menz. Relationships between the foot posture index and foot kinematics during gait in individuals with and without patellofemoral pain syndrome. *Journal of Foot and Ankle Research*, 2011.

- [18] Kai Berger, Kai Ruhl, Christian Brümmer, Yannic Schröder, Alexander Scholz, and Marcus Magnor. Markerless motion capture using multiple color-depth sensors. In *Proc. Vision, Modeling and Visualization (VMV) 2011*, pages 317–324, October 2011.
- [19] Bobby Bodenheimer, Chuck Rose, Seth Rosenthal, and John Pella. The Process of Motion Capture: Dealing with the Data. *Microsoft Research and Interactive Media Production Microsoft*, 1997.
- [20] Jean-Yves Bouguet. Camera Calibration Toolbox for Matlab [online], 2010. Citováno 18.12. 2012, <[http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)>.
- [21] Jan Buriánek. Vicon - nejlepší systém optického snímání pohybu. *Pixel 174*, 2009.
- [22] Jan Buriánek. Animazoo. *Pixel 191*, 2012.
- [23] M. Devy, V. Garric, and J.J. Orteu. Camera calibration from multiple views of a 2d object, using a global nonlinear minimization method. In *Intelligent Robots and Systems*, volume Volume 3, 1997.
- [24] Andrew Fitzgibbon, Jamie Shotton, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-Time Human Pose Recognition in Parts from Single Depth Images. *Microsoft Research Cambridge and Xbox Incubation*.
- [25] James R. Gage. Gait analysis - an essential tool in the treatment of cerebral palsy. *Clin Orthop.*, 1993.
- [26] Daniel Grest, Jan Woetzel, and Reinhard Koch. Nonlinear Body Pose Estimation from Depth Images. *Christian-Albrechts-University Kiel, Germany*.
- [27] M. P. Kadaba, H. K. Ramakrishnan, M. E. Wootten, J. Gainey, G. Gorton, and G. V. B. Cochran. Repeatability of Kinematic, Kinetic, and Electromyographic Data in Normal Adult Gait. *Jurnal of Orthopaedic Research*, pages 849 – 860, 1989.
- [28] Abhishek Kar. Skeletal Tracking using Microsoft Kinect. *Department of Computer Science and Engineering, IIT Kanpur*.
- [29] Hye-Jeong Kim and Kyoung-Mi Lee. Silhouette based human motion estimation for movement education of young children. *Child Game Research Center and Intelligent Multimedia Lab, Duksung Women's University, Korea*, 2006.
- [30] Joan Lasenby and Adam Stevenson. Using geometric algebra in optical motion capture, 2001.
- [31] Microsoft. *Kinect for Windows SDK documentation*, 2012.
- [32] Motion Lab Systems. *The C3D File Format - User Guide*, January 2008.
- [33] D. Pasparakis and N. Darras. Normal walking - principles, basic concepts, terminology. *EEXOT*, Volume 60, 2009.



- [34] Jan Poul, Jan Jadrný, Aneta Fedrová, Karel Urbášek, Jaroslava Bajerová, and Ludmila Kaiser-Šrámková. Vyšetření v laboratoři chůze (instrumentální pohybová analýza). *Pediatric pro praxi*, 2012.
- [35] Jan Smisek, Michal Jancosek, and Tomas Pajdla. 3D with Kinect. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011.
- [36] Petr Stolař. *Vyhodnocování biometrických měření laboratoře chůze*. České vysoké učení technické v Praze, 2010.
- [37] Russell M. Taylor, Thomas C. Hudson, Adam Seeger, Hans Weber, Jeffrey Juliano, and Aron T. Helser. Vrpn: a device-independent, network-transparent VR peripheral system. In *ACM Symposium on Virtual Reality Software & Technology 2011*, pages 55–61, 2011.
- [38] Kudo Tsunoda and Andrew Fitzgibbon. Kinect for Xbox 360 - The Innovation Journey. In *Microsoft Research Faculty Summit*, 2010.
- [39] Vicon. Oxford foot model. *Software Specification Oxford Foot Model*.
- [40] Youding Zhu and Kikuo Fujimura. Constrained Optimization for Human Pose Estimation from Depth Sequences. *The Ohio State University, Honda Research Institute USA*.
- [41] Youding Zhu and Kikuo Fujimura. A Bayesian Framework for Human Body Pose Tracking from Depth Image Sequences. *The Ohio State University, Honda Research Institute USA*, 2010.
- [42] Martin Švehlík, Ernst Bernhard Zwick, Gerhardt Steinwender, Tanja Kraus, and Wolfgang E. Linhart. Přístrojová analýza chůze u pacientů s dětskou mozkovou obrnou. *Neurologie pro praxi*, 2011.



## Příloha A

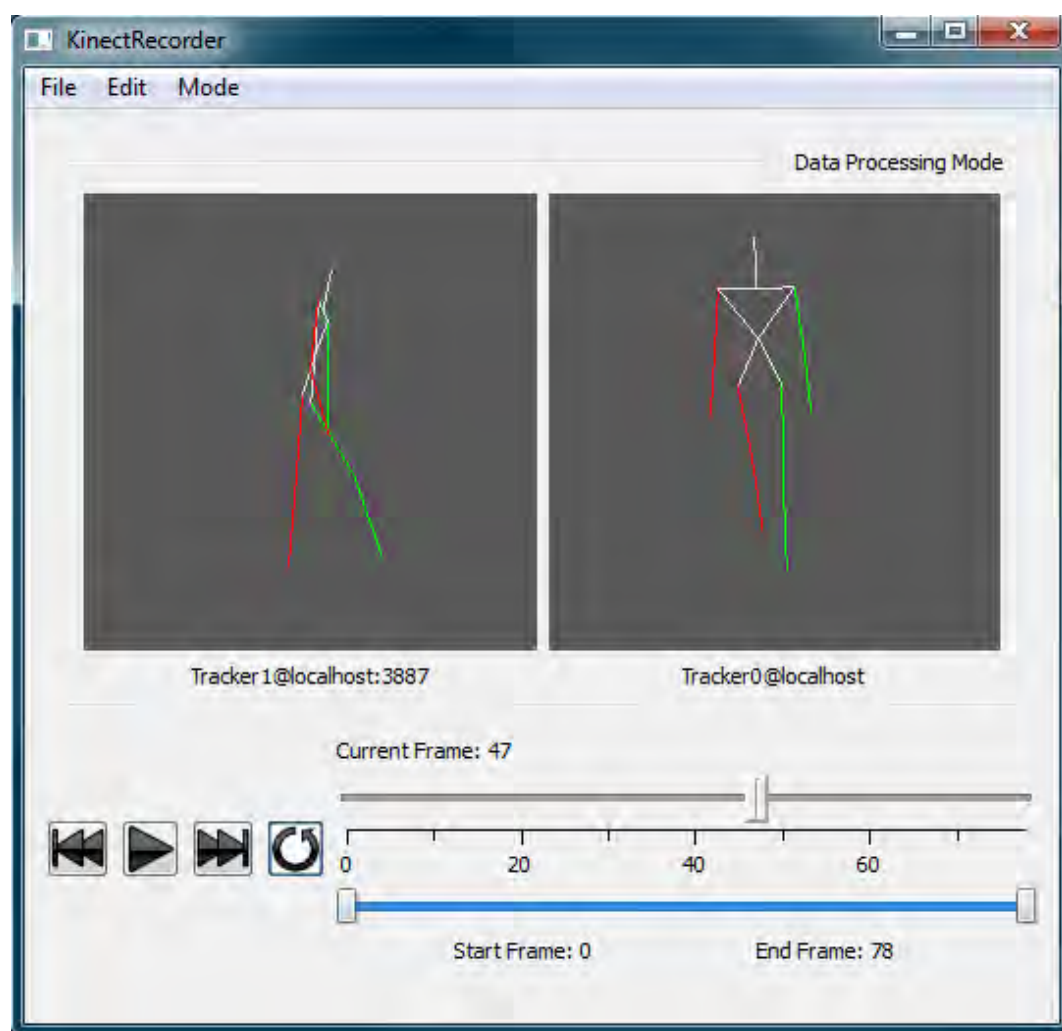
# Seznam použitých zkratek

<b>API</b>	Application Programming Interface
<b>DMO</b>	DirectX Media Object
<b>EMG</b>	Elektromyografie
<b>FPS</b>	Frames Per Second
<b>GUI</b>	Graphical User Interface
<b>ICP</b>	Iterative Closest Point
<b>IR</b>	InfraRed
<b>LED</b>	Light Emitting Diodes
<b>MoCap</b>	Motion Capture
<b>MS</b>	Microsoft
<b>NUI</b>	Natural User Interface
<b>OFM</b>	Oxford Foot Model
<b>PC</b>	Personal Computer
<b>RGB</b>	Red - Green - Blue
<b>SDK</b>	Software Development Kit
<b>s. s.</b>	souřadný systém
<b>USB</b>	Universal Serial Bus
<b>VRPN</b>	Virtual Reality Peripheral Network

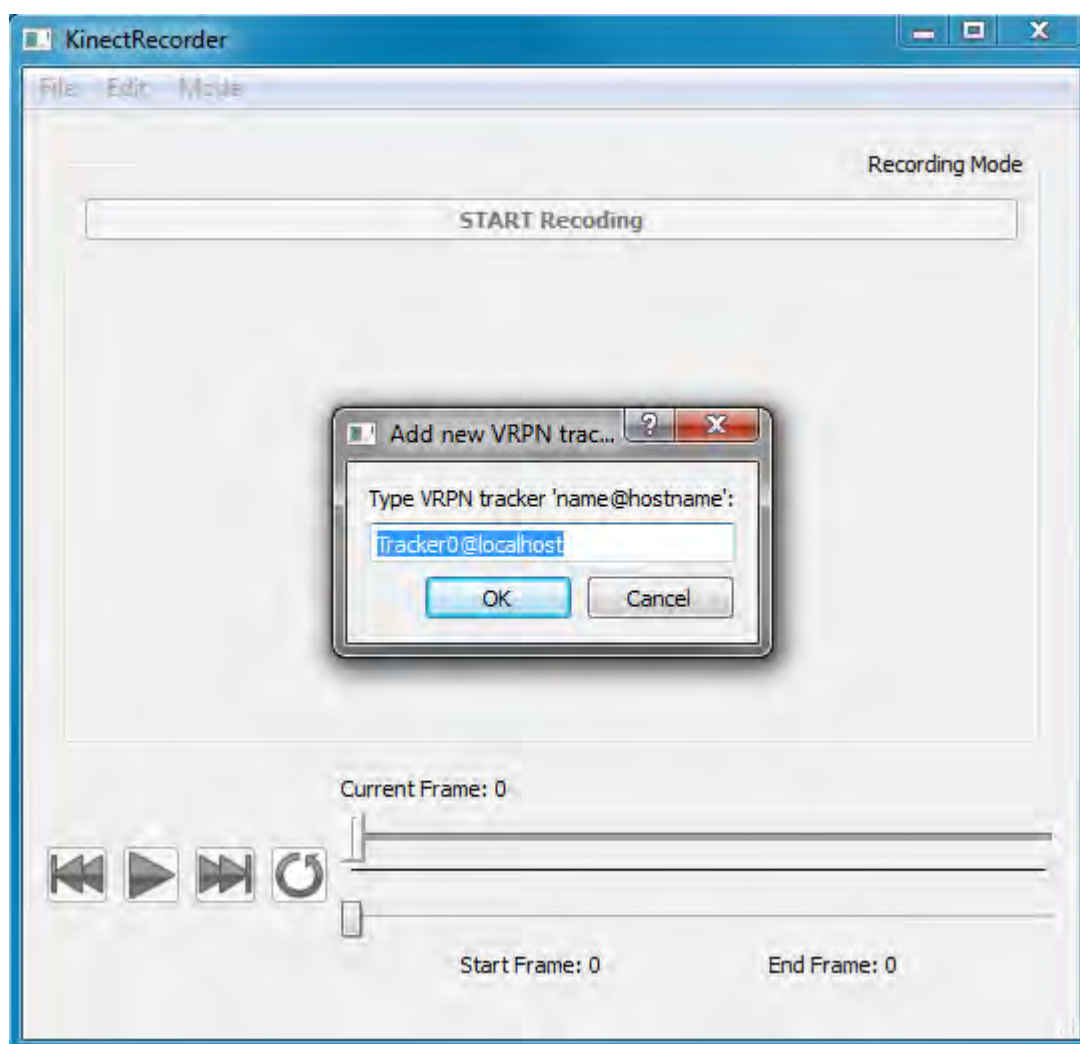


## Příloha B

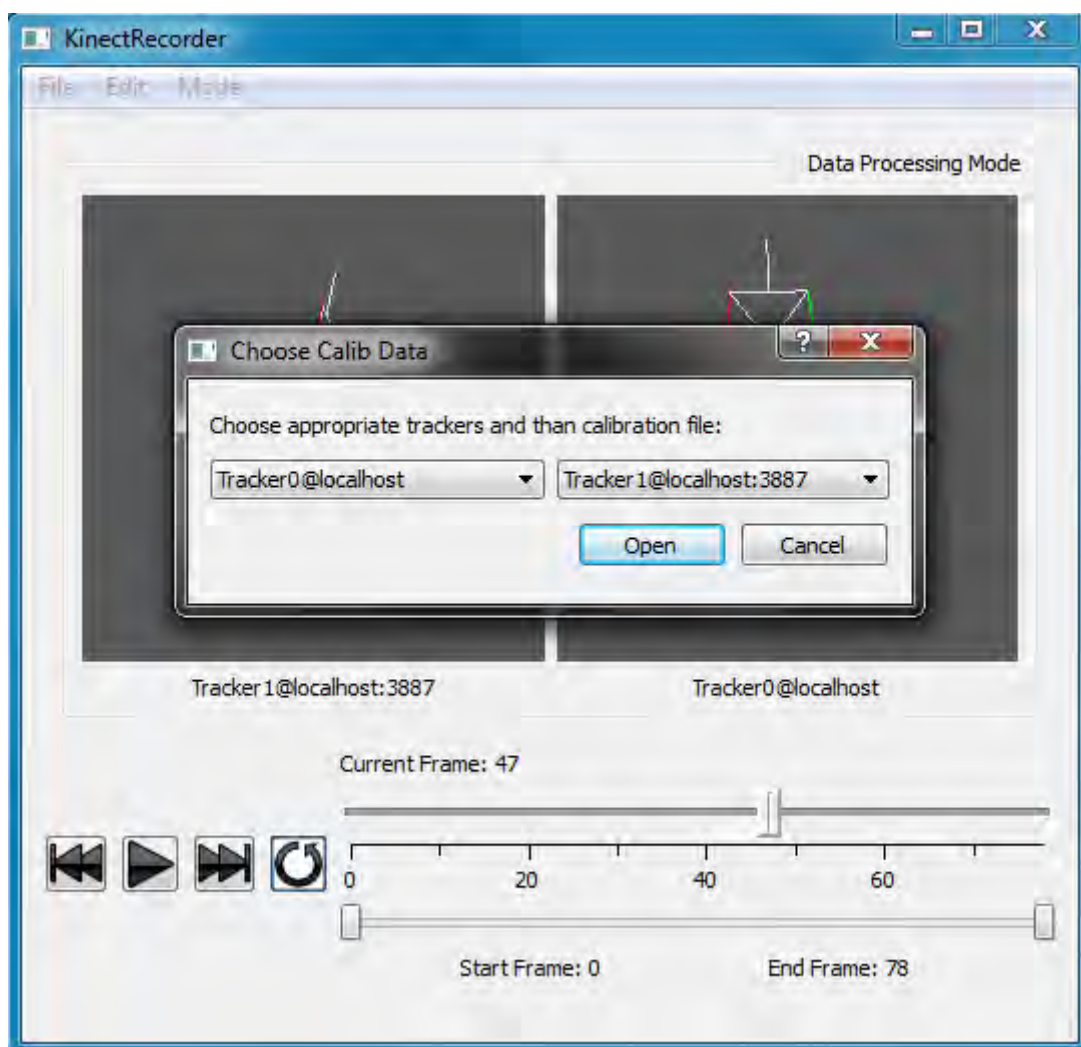
### Obrazová příloha



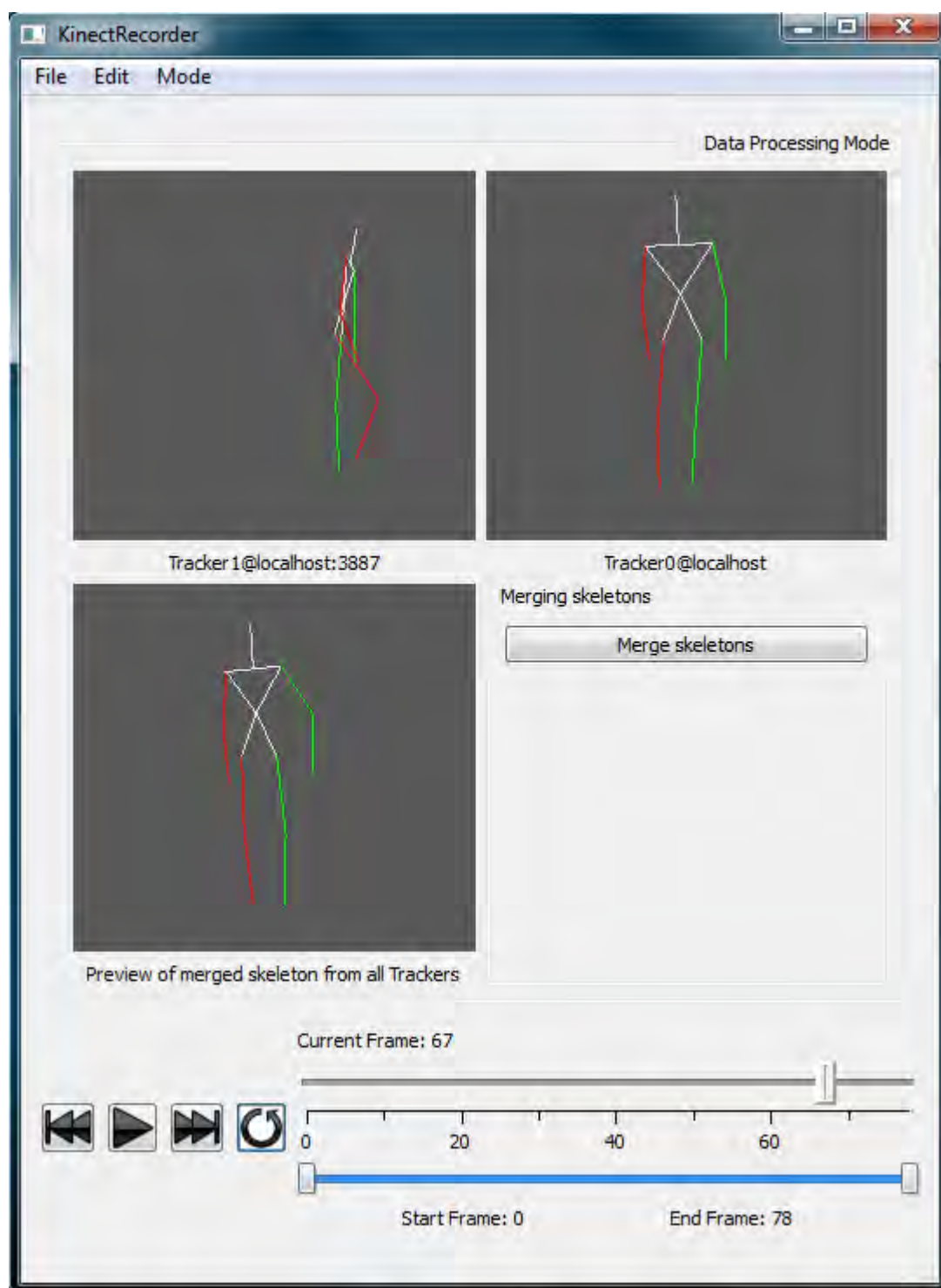
Obrázek B.1: Aplikace Kinect Recorder



Obrázek B.2: Aplikace Kinect Recorder - přidání nového VRPN trackeru. Tato operace způsobí vytvoření nového VRPN remote trackeru, který se připojí k uvedenému VRPN serveru.

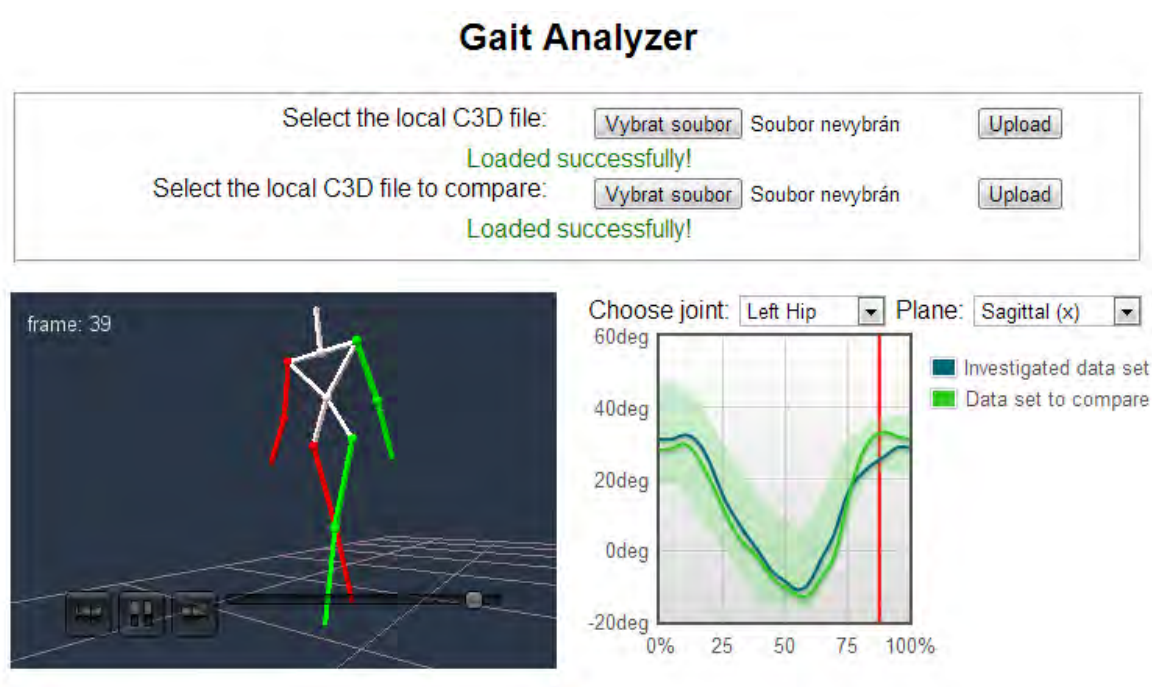


Obrázek B.3: Aplikace Kinect Recorder - definice pořadí Kinectů a načtení kalibračního souboru.



Obrázek B.4: Aplikace Kinect Recorder - sloučení nasnímaných koster do jediné výsledné.

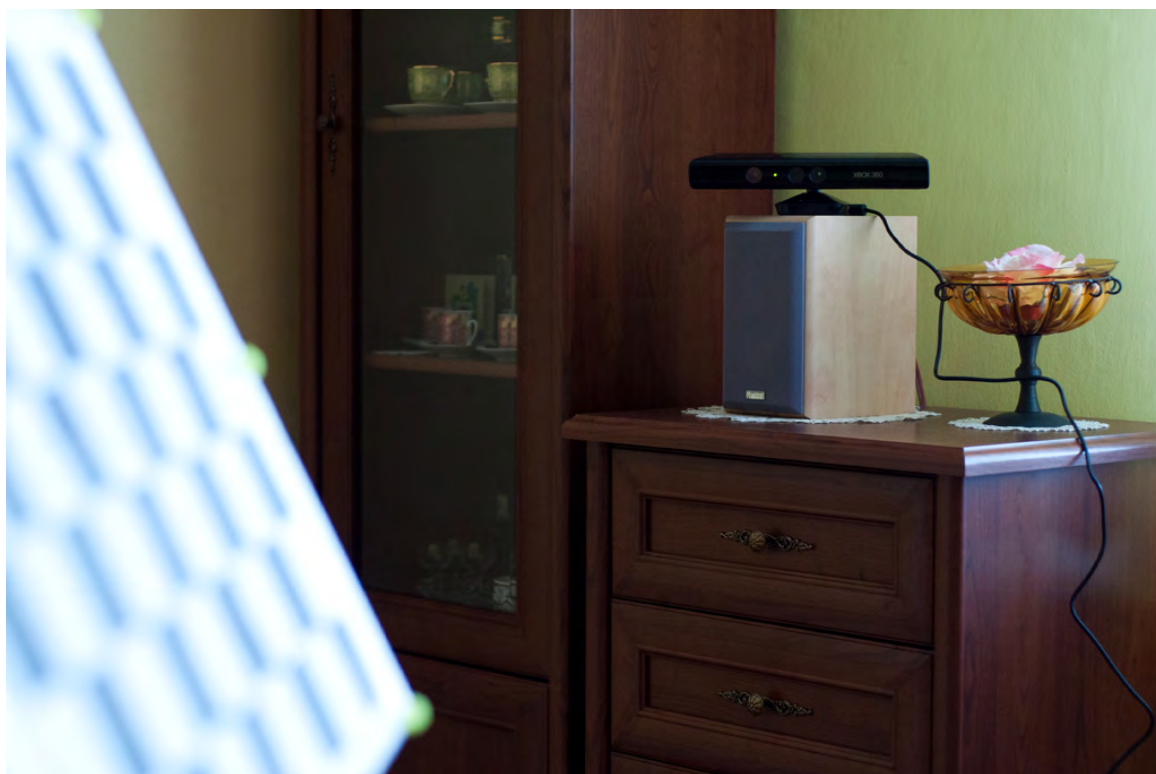




Obrázek B.5: Aplikace Gait Analyzer - porovnání dvou záznamů.



Obrázek B.6: Experiment - simulovaná laboratoř chůze.



Obrázek B.7: Experiment - simulovaná laboratoř chůze.



Obrázek B.8: Experiment - kalibrace systému.





Obrázek B.9: Experiment - dynamický záznam.



Obrázek B.10: Experiment - dynamický záznam.





Obrázek B.11: Experiment - participant simulující poruchu hybnosti. Na levém rameni nese tašku obsahující závaží o hmotnosti 10 kg, která způsobuje vychýlení těžiště těla, což má vliv i na samotnou chůzi.





## Příloha C

# Obsah přiloženého CD

·	
— exe	- adresář s přeloženými programy a exotickými .dll
— src	- adresář se zdrojovými kódy programů a exotickými knihovnami
— text	- adresář obsahující text DP v podobě LaTeXu a PDF
— install.txt	- instalační poznámky
— README.txt	- popis co ve kterém adresáři je + postup spuštění

Obrázek C.1: Obsah přiloženého CD