

MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY



Snímání a rekonstrukce pohybu postavy

BAKALÁŘSKÁ PRÁCE

Michal Vinkler

Brno, jaro 2009

Prohlášení

Prohlašuji, že tato práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Vedoucí práce: RNDr. Petr Beneš

Poděkování

Tímto bych chtěl poděkovat vedoucímu mé práce, RNDr. Petru Benešovi, za jeho cenné rady a připomínky při psaní této práce. Dále bych rád poděkoval všem účastníkům jMonkey Engine fóra, kteří mi trpělivě odpovídali na mé dotazy a pomohli mi tak překonat nejednu překážku. Děkuji také svým blízkým a přátelům, kteří mě při tom podporovali.

Shrnutí

Snímání pohybu je rychle se rozvíjející technologie, pomocí které lze digitálně zaznamenávat pohyby lidí, zvířat a neživých předmětů. Prostřednictvím této technologie je možné poměrně rychle, jednoduše a hlavně velmi detailně zaznamenat pohyby snímaného objektu a dále s nimi pracovat, například převést je na počítačem vytvořený model.

Cílem této práce bylo prostudovat techniky pro zaznamenávání a rekonstrukci pohybu objektů v prostoru, dále vytvořit ukázkovou aplikaci pro zařízení OptiTrack, která umožní snímání a rekonstrukci pohybu v reálném čase. Součástí aplikace je i nezávislá obecná knihovna zajišťující načítání vstupních dat ze zařízení OptiTrack pro možné využití v dalších aplikacích.

Klíčová slova

snímání pohybu, motion capture, virtuální realita, virtual reality, OptiTrack, Java, jMonkey engine, jME Physics 2, VRPN

Obsah

1	ÚVOD	1
2	SNÍMÁNÍ POHYBU	2
2.1	POJEM SNÍMÁNÍ POHYBU	2
2.2	VÝHODY A NEVÝHODY	3
2.3	OBLASTI POUŽITÍ	4
2.3.1	<i>Armáda</i>	4
2.3.2	<i>Lékařství</i>	4
2.3.3	<i>Virtuální realita</i>	5
2.3.4	<i>Filmová produkce a počítačové hry</i>	5
3	KLASIFIKACE SYSTÉMŮ PRO SNÍMÁNÍ POHYBU.....	6
3.1	KLÍČOVÉ VLASTNOSTI.....	6
3.1.1	<i>Latence</i>	6
3.1.2	<i>Přesnost</i>	6
3.1.3	<i>Obnovovací frekvence</i>	6
3.1.4	<i>Rozptyl</i>	6
3.2	DĚLENÍ SYSTÉMŮ SNÍMÁNÍ POHYBU.....	7
3.2.1	<i>Mechanické</i>	7
3.2.2	<i>Magnetické</i>	8
3.2.3	<i>Inerční</i>	9
3.2.4	<i>Optické</i>	10
4	POUŽITÉ TECHNOLOGIE	13
4.1	JMONKEY ENGINE.....	13
4.2	JME PHYSICS 2	14
4.3	VRPN.....	15

5	OMEZENÍ.....	16
5.1	KAMERY A PŘÍSLUŠNÝ SOFTWARE	16
5.2	POČET SNÍMANÝCH POZIC	19
5.3	INTELIGENTNÍ FILTR DAT	20
6	IMPLEMENTACE	23
6.1	PROSTŘEDÍ.....	23
6.2	VIRTUÁLNÍ POSTAVA	24
6.3	MAPOVÁNÍ SNÍMANÝCH POZIC NA POSTAVU	25
7	ZÁVĚR.....	26
7.1	MOŽNOSTI ROZŠÍŘENÍ	26
A	DOPLŇUJÍCÍ SNÍMKY	30
B	UŽIVATELSKÝ MANUÁL.....	35
B.1	ZAČÍNÁME	35
B.1.1	Hardwarové nároky.....	35
B.2	SPUŠTĚNÍ APLIKACE	36
B.2.1	NetBeans projekt.....	36
B.2.2	Zkompilovaný jar soubor.....	38
B.3	NASTAVENÍ GRAFICKÝCH PARAMETRŮ	39
B.4	POPIS APLIKACE MoCAP	39
B.4.1	Mapování kláves	39
B.4.2	Ladicí mód.....	42
B.4.3	Vyobrazení některých funkcí.....	43
B.5	NASTAVENÍ PROGRAMU RIGID BODY TOOL	44
B.6	ZNÁMÉ PROBLÉMY	44
C	OBSAH PŘÍLOŽENÉHO CD	45

1 Úvod

Snímání pohybu (Motion Capture) je zajímavou a rychle se rozvíjející technologií; používá se pro zaznamenávání pohybu herce nebo jiných objektů. Data takto zaznamenaná se pak nejčastěji použijí pro vytvoření 3D animace. Snímání pohybu se poslední dobou velmi rozšiřuje ve filmovém průmyslu, kde se nezapomenutelně zapsalo do paměti producentů a režisérů jako vynikající nástroj pro tvorbu animací počítačem vytvořených postav. Je však také velkou mírou využíváno v herním průmyslu, kde slouží jako pomůcka pro vytváření animovaných částí her nebo třeba i pohybů jednotlivých charakterů. Hojné využití nachází i ve vojenském či lékařském odvětví.

Snímání a rekonstrukce pohybu postav velice úzce souvisí s virtuální realitou. Tímto pojmem nazýváme počítačem vytvořené prostředí, které uživateli poskytuje iluzi, že se nachází v uměle vytvořeném (virtuálním) světě. Přesná definice pojmu Motion Capture, jeho výhody, nevýhody a možnosti použití (ať už ve virtuální realitě, nebo jiných odvětvích lidské činnosti) jsou popsány v [druhé kapitole](#).

[Kapitola třetí](#) pojednává o nejčastěji používaných způsobech snímání pohybu. Čtenář se zde může dočíst, na jakém principu pracují systémy mechanické, magnetické, inerční či optické. V této kapitole jsou také stručně popsány vlastnosti charakterizující tyto systémy, jako například latence, přesnost nebo obnovovací frekvence.

[Čtvrtá kapitola](#) se věnuje popisu technologií použitých u výsledného programu. Zaměřuji se na jMonkey Engine [\[01\]](#), jeho rozšiřující modul jME Physics 2 [\[02\]](#) a VRPN [\[03\]](#), rozhraní pro komunikaci mezi serverem (získávajícím pozice z kamerového systému) a klientskými aplikacemi.

Stejně jako u většiny jiných projektů, i zde se při implementaci objevila jistá omezení, ať už v podobě limitovaného počtu snímaných pozic, či příliš malého prostoru pokrytého kamerami. Těmito problémy a jejich případným řešením se zabývám v [páté kapitole](#).

V [kapitole šesté](#) se soustředím na vybrané implementační detaily. Je zde popsána virtuální postava, okolní prostředí a způsob mapování pozic získávaných z kamerového systému na postavu.

V [závěru](#) zmiňuji možnosti rozšíření a hodnotím přínos mé práce.

2 Snímání pohybu

2.1 Pojem snímání pohybu

Vymezit přesně pojem „snímání pohybu“ (anglicky „Motion Capture“, zkráceně „MoCap“) není jednoduché, v existující literatuře lze nalézt mnoho různých definicí. Například se můžeme dočíst, že Motion Capture je termín používaný pro popsání procesu zaznamenávání pohybu a přenesení tohoto pohybu na digitální model [04]. Maureen Furniss ve svém článku popisuje, že snímání pohybu zahrnuje měření pozice a orientace objektu ve fyzickém prostoru a následné zaznamenání těchto informací do počítačem zpracovatelné polohy [05].

Pokusme se zamyslet nad správným výkladem tohoto pojmu. Samotný název může být trochu matoucí – takto bychom Motion Capture mohli chápat jako mechanismus pro záznam pohybu nějakého objektu, který se hýbe ve fyzickém světě. Potom by však pouhé natáčení videokamerou mohlo být takto chápáno – videokamera „zaznamená pohyby“ všech osob nebo objektů v jejím zorném poli. Ze zřejmých důvodů tento způsob záznamu není běžně považován za snímání pohybu.

Co doopravdy vymezuje tento pojem, je fakt, že Motion Capture nejenom zaznamená pohyby, ale zároveň také abstrahuje od původu jejich vzniku. Dochází k vytvoření reprezentace, která odděluje pohyby od jejich výskytu; samotný pohyb je zapsán do podoby, která je vhodná pro další analýzy a zpracování.

Můžeme tedy říci, že snímání pohybu je vytváření takové reprezentace pohybu, kterou lze dále zpracovávat, a která vzniká ze snímání příslušného pohybu. To je stále velmi široký pojem; nespecifikovali jsme, jak má výsledná reprezentace vypadat, ani jaké druhy senzorů použijeme pro záznam. Oboje však záleží na tom, jaké výsledky chceme získat a jak budeme se zaznamenanými daty dále pracovat.

V laboratoři HCI (Human Computer Interaction, česky interakce člověka s počítačem) jsou informace o pohybu získávány z obrázků snímáných několika kamerami – používá se optický sledovací systém s pasivními značkami. Pro Motion Capture je to v současné době pravděpodobně nejčastěji používaná technologie. Představu o tom, jak optické snímání pohybu vypadá, lze získat při pohledu na [obrázek 2.1](#). Bližší informace o druzích systémů používaných pro sledování pohybu najdete v [třetí kapitole](#).

Výstupem procesu snímání pohybu nemusí nutně být animace, jak je někdy popisováno v literatuře [06]. Motion Capture je tedy čistě extrakce pohybových dat. Ta pak mohou být použita pro mnoho účelů – kromě již zmiňovaných animací například pro lékařské analýzy, ve strojírenství, nebo jako vstupní mechanismy pro interakci člověka s počítačem, používanou ve virtuální realitě. Oborům, kde se snímání pohybu používá, se věnuji dále v této kapitole.

Systémy pro snímání pohybu mohou být rozděleny do dvou základních kategorií. První z nich jsou online systémy, které dokážou produkovat data v dále zpracovatelné podobě v reálném čase.

Vzhledem k dostupnosti velkého výpočetního výkonu v dnešní době jsou tyto systémy stále více rozšířené. V druhé kategorii jsou offline systémy, u kterých se nejdříve zaznamenají pohybová data, a až po ukončení záznamu se z nich vypočítají informace o poloze a rotaci snímaných objektů.



Obrázek 2.1: Optický systém snímání pohybu společnosti Vicon. Převzato z [04].

2.2 Výhody a nevýhody

Při použití pro animaci modelů nabízí snímání pohybu mnoho výhod oproti klasickému 3D modelování:

- výsledná animace je získána mnohem rychleji, ve výkonných systémech může být dokonce vytvářena v reálném čase;
- množství vynaložené práce nevzrůstá úměrně složitosti snímané scény nebo délce animace oproti použití tradičních technik;
- je dosaženo komplexních pohybů a realistických fyzikálních interakcí.

Na druhé straně existují některé nevýhody:

- k získání a zpracování pohybových dat je potřeba speciální hardware a software s poměrně vysokou pořizovací cenou;
- snímání pohybu je omezeno pouze na určitý prostor;
- pohyby, které neakceptují zákony fyziky, obecně nemohou být zachyceny;
- pokud má počítačový model jiné proporce, než snímaný subjekt, mohou vznikat artefakty;
- snímaný pohyb se nemusí přenést na počítačový model tak, jak bylo očekáváno;
- systémy pro snímání pohybu mohou mít specifické potřeby na prostor a osvětlení, aj.

2.3 Oblasti použití

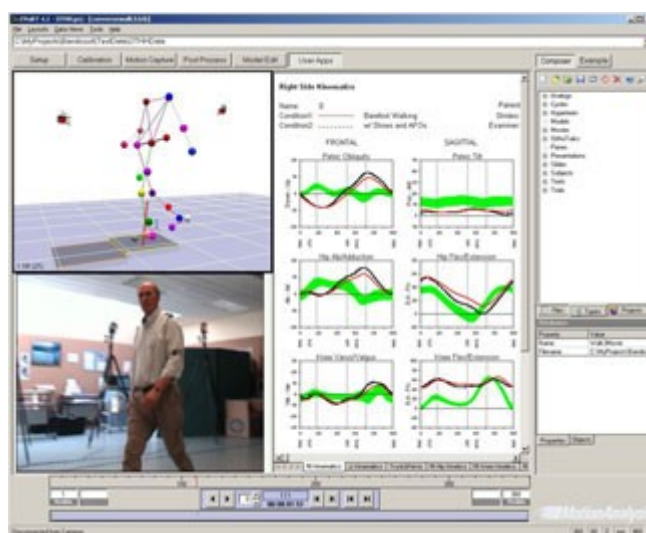
Snímání pohybu má široké spektrum použití. Přední oblastí samozřejmě zůstává zábavní průmysl, který potřebuje realistické ztvárnění pohybů ve hrách i počítačově vytvářených filmových sekvencích. Nicméně tyto oblasti nejsou jediné, nabízí se další uplatnění v armádě, biomedicině, virtuální realitě, atd. Detailnější informace o oblastech použití lze nalézt v [07], [08] a [09].

2.3.1 Armáda

První systémy snímání pohybu byly vyráběny pro armádní účely, počátky jejich používání spadají do druhé poloviny sedmdesátých let minulého století. Pomocí AC magnetického snímání (viz kapitola 3) byla určována pozice a orientace pilotovy helmy. Když pilot otočil hlavu, hlavně zbraňových systémů, řízené senzorem připevněným k jeho helmě, se otáčely s ním [10].

2.3.2 Lékařství

Technologie snímání pohybu jsou v dnešní době hojně používány pro klinické účely. Byly vyvinuty systémy, které analyzují chůzi a držení postoje pacienta. Cílem je identifikace abnormálních pohybů jednotlivých končetin nebo celého těla – ty by mohly znamenat přítomnost nemoci. Například kulhání u dítěte může být příznakem mozkové obrny nebo svalové degenerace [09]. Ukázkový snímek aplikace OrthoTrak¹, která provádí analýzu chůze, lze vidět na obrázku 2.2.



Obrázek 2.2: Aplikace OrthoTrak analyzující chůzi pacienta. Převzato z¹.

¹ OrthoTrack, <<http://www.motionanalysis.com/html/movement/gait.html>>

2.3.3 Virtuální realita

Systémy snímání pohybu, které pracují v reálném čase, umožňují uživateli ovládat virtuální bytost reprezentující uživatele (tzv. „avatař“) v počítačem vytvořeném prostředí. Tím je dosaženo přirozené interakce s virtuálním světem, narozdíl od omezených možností běžně dostupných vstupních zařízení (například klávesnice a myši). Mnou naprogramovaná aplikace se soustředí na možnost pohybovat s počítačem vytvořenou postavou – uživatelské pohyby jednotlivých částí těla jsou mapovány na tytéž části těla virtuální postavičky. Možnosti interakce s virtuálním prostředím mohou být rozšířeny v mnoha směrech, pro jednoduchou demonstraci je možno nechat před uživatelem objevit gumový míček, se kterým lze manipulovat.

2.3.4 Filmová produkce a počítačové hry

Zábavní sektor (zejména vývoj počítačových her) je bezpochyby největší oblastí, ve které se snímání pohybu používá. V herním průmyslu se Motion Capture uplatní jak při animaci jednotlivých pohybů postav, tak při vytváření video sekvencí, které ještě více umocňují zážitek ze hry.

V současnosti se stále častěji vytvářejí filmy, které jsou zcela vytvořeny počítačem. Pohyby herců jsou nasnímány a převedeny na jejich protějšky ve filmu; vznikají tak postavy, které působí velmi realistickým dojmem. Motion Capture se také používá v situacích, které by pro živé herce byly nepraktické nebo nebezpečné, jako například u postav padajících z vysoké výšky z lodi ve filmu Titanic [11]. Na obrázku 2.3 jsou zobrazeny scény z filmů Polar Express a Lord of the Rings, The Two Towers, které představují milníky v použití snímání pohybu pro potřeby filmu [12].



Obrázek 2.3: Použití technik snímání pohybu při tvorbě animovaných postav. Ukázky z filmů (a) Polar Express, (b) Lord of the Rings, The Two Towers. Převzato z [13].

3 Klasifikace systémů pro snímání pohybu

3.1 Klíčové vlastnosti

Všechna zařízení pro snímání pohybu se vyznačují určitými vlastnostmi, jako jsou latence, rozptyl, přesnost snímání, obnovovací frekvence. Tyto charakteristiky budou krátce popsány v následujících odstavcích, podrobné informace o vlastnostech i rozdělení jednotlivých systémů lze najít v [07], [08] a [14].

3.1.1 Latence

Zpoždění mezi změnou pohybu/rotace objektu a zaznamenáním této změny snímacím zařízením. U systémů pracujících v reálném čase musí být latence velmi malá (okem nepozorovatelná).

3.1.2 Přesnost

Udává maximální chybu, která může vzniknout při snímání polohy/rotace. U špičkových systémů se můžeme setkat s přesností měření na desetiny milimetru. Pro většinu zařízení není tato hodnota konstantní, ale je závislá na vzdálenosti senzorů a snímaných objektů.

3.1.3 Obnovovací frekvence

Určuje, kolikrát za jednu vteřinu dojde ke změření polohy/rotace snímaného objektu. Čím je obnovovací frekvence vyšší, tím jsou zaznamenané pohyby plynulejší. Polohovací systém OptiTrack², používaný v laboratoři HCI, má snímkovací frekvenci 100 Hz.

3.1.4 Rozptyl

Udává odchylku, se kterou kolísají hodnoty na výstupu snímacího zařízení pro objekt, který se nehýbe. Zatímco konstantní chyba v určování polohy/rotace nemusí být tak nápadná, velký rozptyl způsobuje chvění a nepříliš plynulé pohyby.

² OptiTrack, <<http://www.naturalpoint.com/optitrack/>>

3.2 Dělení systémů snímání pohybu

Snímání pohybu se v současnosti nejčastěji provádí opticky, magneticky, inerčně nebo mechanicky. Každý z těchto systémů má své výhody a nevýhody, nelze vybrat jedinou technologii, která je nejvhodnější pro použití ve všech případech.

3.2.1 Mechanické

Mechanické systémy nabízejí relativně rychlé a levné řešení, jak sledovat a zaznamenávat pohyby člověka. Pro účely snímání je na tělo člověka, jehož pohyby chceme snímat, přichycena „umělá kostra“ tvořená pevnými částmi, které jsou navzájem spojeny klouby. Jelikož je tato kostra vně lidského těla, je často označována pojmem „exoskeleton“ [04]. Exoskeleton je konstruován tak, aby pevné části co nejlépe odpovídaly jednotlivým končetinám a spojení mezi nimi korespondovalo s klouby člověka. Aby bylo dosaženo co nejlepších výsledků, je exoskeleton přizpůsobitelný různým výškám a tělesným proporcím. I přes neustálé zlepšování flexibility dochází u těchto systémů k určitému omezení pohybu uživatele. Jeden z nejmodernějších systémů tohoto druhu, Gypsy³, lze vidět na obrázku 3.1.



Obrázek 3.1: Moderní mechanický systém snímání pohybu Gypsy6. Převzato z³.

Pevné prvky exoskeletonu sledují pohyb jednotlivých částí těla a v ideálním případě tak mezi sebou svírají stejné úhly, jako jim odpovídající končetiny. Dané úhly se pak změní pomocí elektromechanických transduktorů – nejčastěji se pro tyto účely používají potenciometry [07]. Měření provedená transduktory jsou poslána do počítače; starší systémy musely být připojeny kabelem, zatímco v poslední době je stále častěji preferováno bezdrátové připojení. Pomocí relativně jednoduchých výpočtů, které mohou být snadno prováděny v reálném čase, se z těchto údajů vypočítají pozice a rotace jednotlivých končetin a určí se postoj celého těla.

³ Gypsy6, <http://www.inition.co.uk/inition/product.php?URL_=product_mocaptrack_animazoo_gypsy>

Problémem však zůstává určení absolutní pozice uživatele vůči okolnímu světu, kterou takto vypočítat nelze. Řešením může být připojení exoskeletonu k externímu pozičnímu systému nebo kombinace s jinou technologií pro sledování polohy (například použití akcelerometrů z inerčních systémů). Podrobnější popis tohoto problému lze nalézt v [15]. Dalším faktem je, že pomocí mechanického snímání pohybu lze sledovat pouze lidské tělo, nikoli například zvířata či předměty, které uživatel bere do rukou.

Nabízejí se však i výhody, které tyto systémy činí v porovnání s ostatními technologiemi velmi zajímavé. Mezi ně patří zpracování dat v reálném čase, relativně nízká cena, jednoduchost použití, nezávislost měření na okolních podmínkách (ať už světelných, nebo elektromagnetických). Pokud je použita bezdrátová technologie přenosu signálu, pak je prostor, ve kterém lze snímat pohyby, prakticky neomezený.

3.2.2 Magnetické

Magnetické systémy používají k určení polohy a rotace jednotlivých senzorů elektromagnetické pole generované stacionárním vysílačem. Sensory jsou umístěny na těle uživatele a jsou připojeny k řídicí jednotce – tato spojení jsou ve většině případů realizována pomocí kabelů [08]. Vysílač se skládá ze tří vzájemně ortogonálních cívek uzavřených ve feromagnetickém obalu. Tyto cívky jsou napájeny elektrickým proudem a generují tak tři ortogonální elektromagnetická pole; ta jsou buď alternující (pro AC systémy) nebo pulsující (pro DC systémy). Přijímač je v případě použití AC elektromagnetického pole složen rovněž ze tří malých ortogonálních cívek, v případě použití DC pole ze tří magnetometrů (alternativně ze tří senzorů Hallova jevu) [14]. Řídicí jednotka pak dokáže na základě měření napětí generovaného elektromagnetickou indukcí v senzorech určit relativní pozici a orientaci přijímače vůči vysílači a tato pohybová data předává počítači pro další zpracování.

Vzhledem k tomu, že ani lidská tkáň, ani nekovové předměty nebrání šíření elektromagnetického pole, nevzniká zde problém častý například u optického snímání, při kterém v důsledku zakrytí sledovaných bodů dojde na určitý čas k přerušení sledování pozice a rotace daného objektu – tzv. „line of sight issue“. Na druhou stranu tato technologie má problémy s elektromagnetickou vodivostí kovů umístěných poblíž vysílače. Při generování magnetického pole dochází v kovových předmětech vyskytujících se v bezprostřední blízkosti k indukci elektromagnetického náboje, což má za následek vytváření nových elektromagnetických polí, která interferují s polem vytvářeným vysílačem a zkreslují tak naměřené hodnoty. AC systémy jsou velmi citlivé na mosaz, hliník nebo měď, nejsou však příliš rušeny železem nebo ocelí. Interference nezpůsobují ani magnetické pole Země či elektrické vedení. Oproti tomu DC systémy mají problémy s materiály jako železo a ocel, ostatní kovy nevytvářejí téměř žádné rušení.

Vysílač v DC systému pracuje ve čtyřech fázích. V prvních třech generuje postupně navzájem ortogonální magnetická pole, která slouží k určení polohy přijímačů. Ve čtvrté fázi není generováno žádné pole, přijímače tak zachytí pouze magnetické pole Země a dalších kovových předmětů v blízkém okolí. Tato odchylka je pak použita ke korekci hodnot změřených v dřívějších fázích. Více informací o této problematice lze nalézt v [07], [08] a [15].

Pro snímání pohybů v HCI laboratoři se ještě donedávna používal DC magnetický systém Nest of Birds⁴. Umožňoval simultánní sledování až čtyř bodů v prostoru frekvencí 105 Hz, z nichž vždy dva byly umístěny v rukavicích zvaných Pinch Gloves. S velkou přesností tak bylo možno sledovat pohyby rukou. Pro snímání pohybu celé postavy však systém nebyl příliš vhodný – čtyři body pro

⁴ Nest of Birds, <www.hi-res800.com/Info/Nest%20of%20Birds.pdf>

pokrytí celé postavy nejsou dostačující a navíc jednotlivé senzory musejí být propojeny kabely k řídicí jednotce, což omezuje uživatele ve volnosti pohybu. Komponenty používané v systému Nest of Birds jsou vyobrazeny na [obrázku 3.2](#). V současné době se místo něj používá optický systém společnosti OptiTrack, jehož popis naleznete dále v této kapitole.



Obrázek 3.2: (a) Pinch Gloves, (b) řídicí jednotka, (c) generátor DC magnetického pole. Převzato z [16].

3.2.3 Inerční

Inerční systémy používají dva typy senzorů – gyroskopy a akcelerometry. Gyroskopy zaznamenávají změnu rychlosti úhlové (a tudíž jsou schopny určit, jak se změnila rotace sledovaného objektu), zatímco akcelerometry zaznamenávají změnu rychlosti pohybové (ze které se určí poloha objektu). Aby bylo možno zaznamenat orientaci ve všech třech osách, jsou použity tři gyroskopy umístěné v navzájem ortogonálních rovinách. Podobně pro výpočet pozice v trojrozměrném prostoru jsou zapotřebí tři akcelerometry, z nichž každý v ideálním případě leží ve stejné rovině, jako jednotlivé gyroskopy. Protože však dochází pouze ke sledování změn polohy, pro výpočet absolutní pozice v prostoru musíme znát počáteční souřadnice daného objektu.

Senzory v podobě malých čipů jsou připevněny přímo na těle uživatele, získaná pohybová data se pomocí kabelu nebo bezdrátově přenášejí do počítače pro další zpracování. Obrovskou výhodou oproti ostatním systémům je absence dalších zařízení nutných pro provoz systému, jako jsou například kamery či generátor magnetického pole. Odpadá také „line of sight“ problém nebo vlivy okolního prostředí na přesnost systému [08]. Senzory mají navíc velice nízkou latenci a disponují vysokými obnovovacími frekvencemi (řádově tisíce měření za vteřinu).

Použití těchto systémů má nicméně jednu závažnou nevýhodu, a to akumulaci chyb; odchylka mezi naměřenými a skutečnými hodnotami se zvyšuje lineárně s dobou měření. Pokud by měl jeden akcelerometr chybu měření pouze 1 mili-g, každé měření by vytvářelo průběžnou odchylku v akceleraci $0,0098 \text{ m}\cdot\text{s}^{-2}$. Po pouhých 30 vteřinách by se vypočítané hodnoty lišily od skutečných až o 4,5 m! [07] Inerční systémy však mají své neocenitelné výhody a používají se v kombinaci s jinými technologiemi pro snímání pohybu. Ukázka použití jednoho z nejdokonalejších inerčních systémů dostupných v současnosti, IGS-190⁵, je vidět na [obrázku 3.3](#).

⁵ IGS-190, <http://www.inition.co.uk/>



Obrázek 3.3: IGS-190 v akci, Royal Festival Hall, Londýn, duben 2005. Převzato z⁵.

3.2.4 Optické

Optické systémy používají ke zjištění pozice a orientace snímaného objektu kamery. Ty jsou u většiny systémů tohoto druhu umístěny ve fixních pozicích a uživatel se pohybuje uvnitř prostoru pokrytého zorným úhlem všech kamer. Jejich počet není pevně dán – lze použít typicky od čtyř po desítky, podle velikosti snímaného prostoru. V současné době většina komerčně dostupných systémů používá pro určování polohy CCD („Charge-Coupled Device“, česky „zařízení s vázanými náboji“) kamery [08]. Ty jsou umístěny tak, aby každá z nich zabírala snímaný objekt z jiné perspektivy.

Celý systém je nutno nejdříve zkalibrovat. Během tohoto procesu se určí přesná poloha a natočení každé kamery. Při snímání udělá každá z kamer v určitých časových intervalech (30 až 2000krát za vteřinu⁶) obraz scény. V těchto obrazech je následně určena 2D pozice každého ze snímaných bodů (tzv. značek); tyto souřadnice, různé pro každý obraz, jsou v kombinaci se znalostí přesného umístění dané kamery v prostoru dále použity pro výpočet 3D souřadnic sledovaných bodů. Pokud chceme sledovat nejen pozici určité části těla uživatele, ale i rotaci, je nutné definovat rovinu v prostoru alespoň třemi body a z nich vytvořit skupinu značek, která je nadále sledována jako celek. Kamery musejí být synchronizovány s vysokou přesností, aby snímaly obrazy ve stejný čas; i při malých odchylkách v synchronizaci může docházet k zásadním chybám v měření.

⁶ Schopnost snímat frekvencí 2000 Hz mají kamery Vicon T160, <<http://www.vicon.com/products/t160.html>>

Optické systémy sledování pohybu lze dále rozdělit na systémy používající značky a systémy bez značek. Obě kategorie budou popsány v následujících odstavcích.

3.2.4.1 Optické systémy používající značky

Značky jsou objekty umístěné na těle uživatele, jejichž poloha může být na zachycených snímcích lehce identifikována. U aktivních systémů tyto značky samy vyzařují světlo, u pasivních ho pouze odrážejí. Při práci se světlem v oblasti viditelného spektra vyvstává mnoho problémů s proměnlivými světelnými podmínkami během dne a s optickým šumem (vytvářeným objekty ve scéně) ztěžujícím identifikaci jednotlivých značek v obraze. Řešením je práce v infračerveném spektru – umělé zdroje osvětlení nezpůsobují poruchy měření (přímé sluneční světlo však ano) a snímání polohy může probíhat i ve tmavých prostředích. Na obrázcích zachycených kamerami pak snímané body vypadají jako světlé skvrny na tmavém pozadí.

Jako značky používají pasivní optické systémy nejčastěji kuličky obalené reflexivním materiálem [17]. Ty jsou relativně malé (od jednotek po desítky milimetrů v průměru) a nejsou mezi sebou nijak propojeny, což dává uživateli maximální volnost pohybu. Reflexní materiál na kuličkách odráží infračervené záření zpět směrem ke zdroji, infračervené reflektory jsou proto integrovány přímo v těle kamer nebo jsou umístovány v jejich těsné blízkosti.

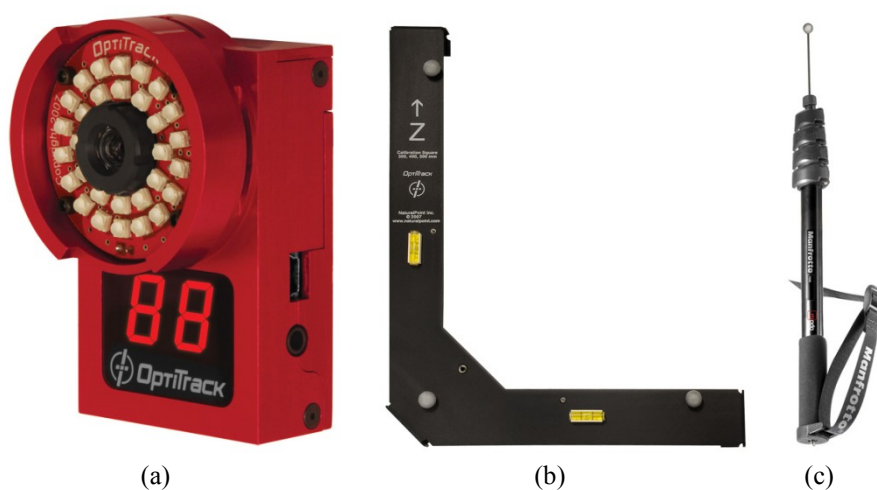
U aktivních optických systémů se jako značky používají infračervené LED diody, které aktivně vyzařují světlo. Tím lze dosáhnout několikanásobně větších vzdáleností, na které dokážou kamery snímané body rozpoznat. Výhodou je také možnost identifikace jednotlivých značek – diody synchronizované s kamerami blikají v určitém pořadí za sebou (u pasivních systémů od sebe jednotlivé značky nelze odlišit). Na druhou stranu aktivní značky jsou poměrně velké a potřebují vlastní zdroj energie.

V laboratoři HCI je v současnosti nainstalováno devět kamer OptiTrack FLEX:V100⁷ pracujících v infračerveném spektru s frekvencí 100 snímků za vteřinu. Na [obrázku 3.4](#) je zachycena kamera spolu s úhelníkem a tyčí používanými při kalibraci. Standardní reflexní kuličky lze sledovat až na vzdálenost šesti metrů a vytvořit tak snímací prostor o ploše do 37 m² [18]; v případě potřeby mohou být na větší vzdálenosti použity aktivní značky. Vzhledem k nedostačující rozloze laboratoře jsou v současné době pokryty pouze přibližně tři metry čtvereční, na kterých lze snímat pohyby.

3.2.4.2 Optické systémy bez značek

Nejnovější výzkumy v oblasti optického snímání pohybu vedou k technikám, které nevyžadují, aby měl uživatel na sobě připevněny jakékoliv značky. Byly vynalezeny speciální počítačové algoritmy, které analyzují optické vstupy a identifikují v nich lidskou podobu. Takto rozpoznaná postava člověka je pro účely sledování dále rozdělena na dílčí části. Některé systémy nevyžadují žádný speciální oblek, zatímco u jiných je potřeba barevného odlišení jednotlivých končetin. Tato technologie dobře zaznamenává velké pohyby, ale má problémy s rozpoznáváním pohybů prstů, mimiky obličeje, rotací zápěstí a malými pohyby obecně [04].

⁷ OptiTrack FLEX:V100, <<http://www.naturalpoint.com/optitrack/products/flex-v100/>>



Obrázek 3.4: (a) Kamera FLEX:V100, (b) kalibrační úhelník, (c) kalibrační tyč. Převzato z⁷ a⁸.

⁸ OptiTrack MoCap Accessories, <<http://www.naturalpoint.com/optitrack/store/accessories-mocap.html>>

4 Použité technologie

4.1 jMonkey Engine

jMonkey Engine (zkráceně jME) [01] je vysoce výkonné, robustní rozhraní pro programování aplikací, jehož autoři se inspirovali knihou 3D Game Engine Design od Davida Eberlyho [19]. Jeho vznik byl podmíněn absencí plnohodnotných grafických engineů napsaných v Javě. Díky přítomnosti abstraktní vrstvy je možno použít jakýkoliv systém pro vykreslování scény. V současné době jsou podporovány knihovny LWJGL⁹ i JOGL¹⁰. jME je projekt s otevřeným zdrojovým kódem (tzv. open source), který je distribuován pod BSD licenci¹¹. Je určen k volnému použití pro soukromé i komerční účely. Logo jMonkey Engine spolu s logem jME Physics 2 lze vidět na [obrázku 4.1](#).

jME používá architekturu grafu scény (tzv. scene graph architecture). Tento graf umožňuje organizaci herních dat ve stromové struktuře, ve které rodičovský uzel může obsahovat neomezený počet potomků, ale každý potomek má pouze jednoho rodiče. Tyto uzly jsou prostorově organizovány tak, aby mohly být lehce odpojeny celé větve stromu, které následně nebudou zpracovány.

Například, mějme graf scény, ve kterém rodičem všech objektů v pokoji je daný pokoj, rodičem všech pokojů v jednom podlaží je dané patro a rodičem všech pater je dům. Herní postava je v prvním pokoji prvního patra. Můžeme velice rychle odpojit uzly představující ostatní patra (což má za následek odpojení všech pokojů, které neleží v prvním patře, a všech předmětů v nich) a zpracovat pouze jediný uzel v této úrovni. V dalším kroku jsou všechny pokoje v prvním patře kromě toho s postavou také odpojeny a je zpracován pouze první pokoj se všemi jeho předměty.

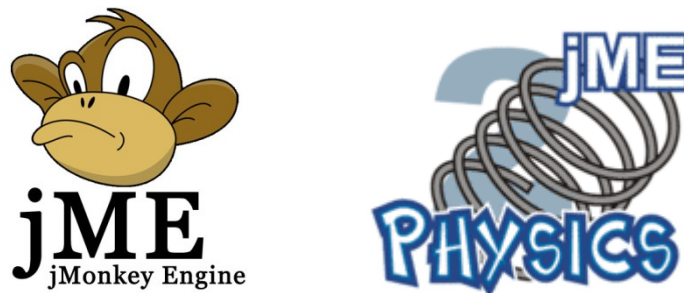
Termín „odpojení“ může představovat mnoho různých akcí, ale při programování grafických aplikací je nejdůležitější nevykreslování objektů, které nejsou v zorném poli kamery. To umožňuje rychlé renderování i velmi komplexních scén, protože v každém okamžiku je viditelná pouze jejich určitá část.

jME také podporuje široké množství grafických efektů, jako jsou bumpmapping, multitexturing či normal mapping. Jednoduše lze používat částicové systémy (simulace deště, sněhu, oblaků), stíny, odlesky, efekt vodní hladiny nebo mlhy. Kompletní seznam všech vlastností lze nalézt v [20].

⁹ LWJGL grafická knihovna, <<http://lwjgl.org/>>

¹⁰ JOGL grafická knihovna, <<https://jogl.dev.java.net/>>

¹¹ BSD licence, <<http://www.opensource.org/licenses/bsd-license.php>>



Obrázek 4.1: Logo jMonkey Engine a jME Physics2. Převzato z [01] a [02].

4.2 jME Physics 2

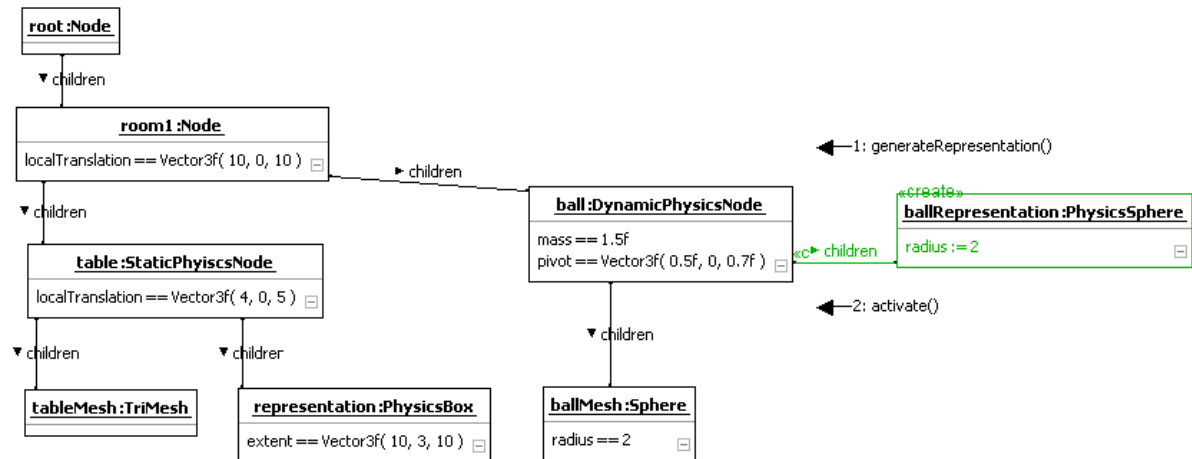
jME Physics 2 (zkráceně jPhysics) [02] představuje rozhraní pro jME, které je nezávislé na použitém fyzikálním enginu – lze použít například ODE¹², PhysX¹³ či ODEJava¹⁴. Poskytuje způsob, jak velice jednoduše implementovat fyzikální zákony do jakéhokoliv projektu postaveného na jMonkey Enginu. Cílem bylo poskytnout pokročilé možnosti použití fyzikálních interakcí – např. byla implementována podpora různých materiálů (led, guma, dřevo, kámen, atd.), takže objekty mohou reagovat odlišně podle typu povrchu, se kterým přišly do styku.

Mezi hlavní vlastnosti patří simulace gravitace, detekce a zpracování kolizí a možnost použít na objekty síly a točivé momenty. jME Physics 2 zcela využívá grafu scény vytvořeného jMonkey Enginem. Na [obrázku 4.2](#) lze vidět, jak jsou jednotlivé objekty v grafu scény specifikovány. Uzly, které jsou instancemi třídy DynamicPhysicsNode nebo StaticPhysicsNode, označují části grafu, které mají být kontrolovány fyzikálním enginem.

¹² ODE Physics Engine, <<http://www.ode.org/>>

¹³ PhysX Physics Engine, <http://www.nvidia.com/object/nvidia_physx.html>

¹⁴ ODEJava Physics Engine, <<https://odejava.dev.java.net/>>



Obrázek 4.2: Způsob výstavby grafu scény. Převzato z [02].

4.3 VRPN

VRPN („Virtual Reality Peripheral Network“, volný překlad „komunikační síť zařízení používaných ve virtuální realitě“) [03] bylo vyvinuto na Univerzitě Severní Karolíny v Chapel Hillu¹⁵. Tento software je používán ke sjednocení struktury informačních dat z velkého množství různých zařízení. Implementuje síťově transparentní rozhraní (nezávislé na konkrétních přístrojích) mezi aplikací a fyzickými zařízeními (systémy pro sledování pohybu, ovládací prvky, haptické senzory, atd.) používanými ve virtuální realitě.

Hlavní charakteristiky VRPN:

- přístup k širokému sortimentu zařízení pomocí rozšiřitelného rozhraní;
- síťová transparentnost;
- časové známky pro všechny zprávy přicházející z/do zařízení;
- synchronizace mezi klienty a servery umístěnými na různých počítačích;
- není omezen počet klientů, kteří se mohou připojit k témuž serveru.

¹⁵ The University of North Carolina at Chapel Hill, <<http://www.unc.edu/>>

5 Omezení

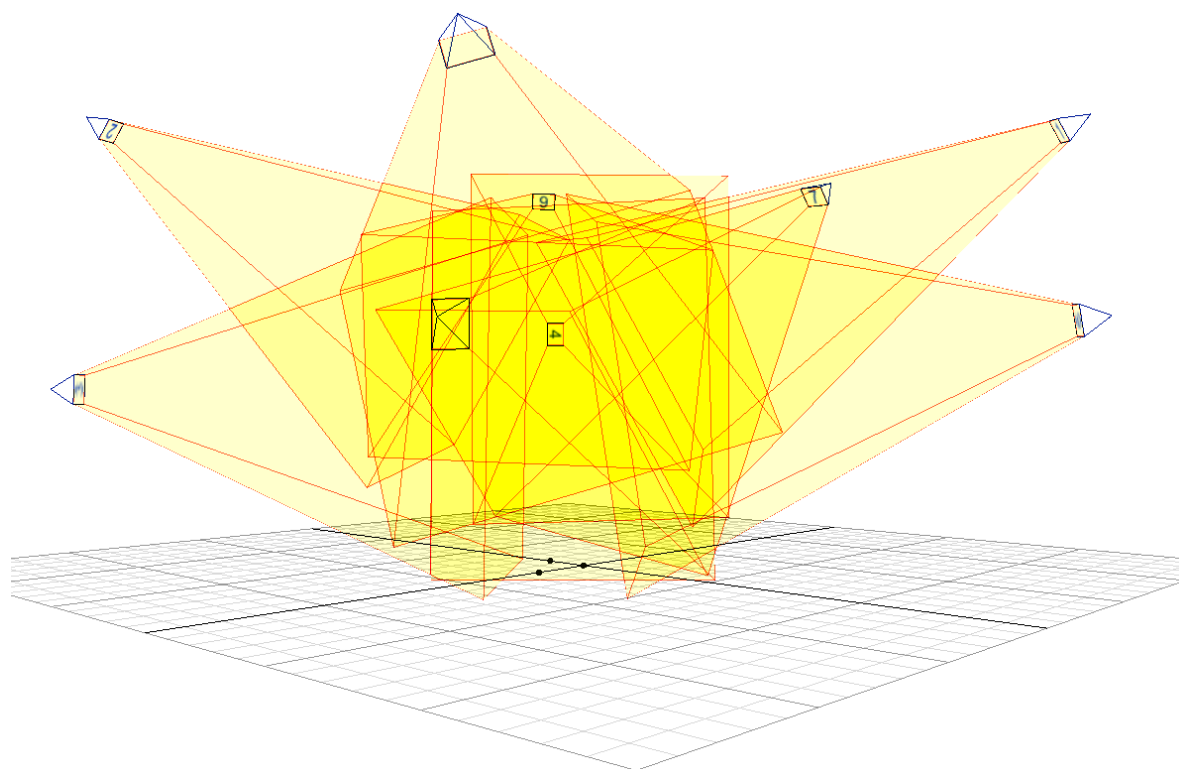
Idea rekonstrukce pohybu postavy je zřejmá – pozice jednotlivých kloubů snímáné v reálném čase a získávané pomocí VRPN jsou aplikovány na klouby virtuální postavy; ta následně kopíruje pohyby prováděné uživatelem. Při implementaci však vyvstávají jistá omezení, která jsou popsána v následujících odstavcích.

5.1 Kamery a příslušný software

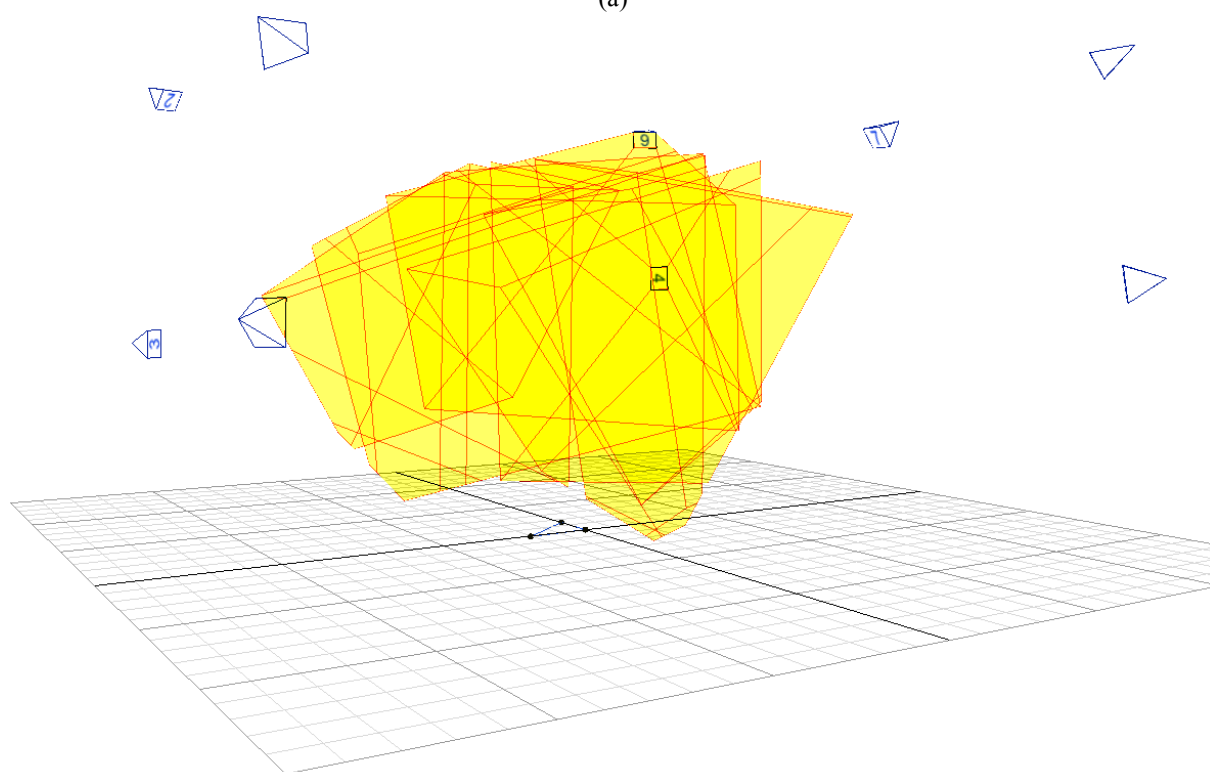
Jak již bylo zmíněno dříve, kamery OptiTrack FLEX:V100 jsou schopny sledovat pasivní značky až na vzdálenost 6 m, vytvářejíce tak prostor pro snímání o rozloze několika desítek metrů čtverečních [18]. Omezená velikost HCI laboratoře však dovoluje pokrýt snímací prostor o rozloze nejvýše 4 m². Tento fakt sám o sobě by nebyl pro demonstraci pohybů postavy přílišnou překážkou – rozpětí paží dospělého člověka nepřesahuje dva metry, pokud bychom tedy stáli přímo uprostřed snímaného prostoru a nedělali bychom úkroky do stran či dopředu a dozadu, všechny naše pohyby by měly být zaznamenány.

Bohužel v současné době jsou kamery orientovány tak, aby snímaly pohyby spíše horní části těla. Vzniká zde tedy omezení minimální výšky nad úrovní podlahy, pod kterou nejsou kamery schopny zaznamenávat pozice jednotlivých bodů. Tento problém však značně komplikuje snímání poloh dolních končetin, kdy polohu chodidel nelze snímat vůbec a pozice lýtek lze zaznamenávat jen s obtížemi. Takové umístění kamer má své opodstatnění. Při použití v aplikacích jinými studenty a pracovníky HCI laboratoře, zejména pak v aplikaci Modeller, vyvíjené Mgr. Jiřím Chmelíkem, je při kreslení potřeba precizního snímání pozic rukou, které se nejčastěji pohybují v úrovni hrudníku a hlavy.

Pokud dospělý člověk zvedne ruce nad hlavu, dosahuje průměrně výšky 230 cm. Vzhledem k malým vzdálenostem mezi kamerami a snímaným objektem vyvstává otázka, zda by při přeorientování kamer tak, aby byla posílena oblast nohou, nedošlo ke zhoršení viditelnosti kamer v horních částech snímaného prostoru. Současné umístění, orientace a zorné pole kamer znázorňuje [obrázek 5.1](#). V horním snímku (a) je žlutě vyobrazen prostor, který vidí každá z kamer. Dolní snímek (b) pak zachycuje tu část, kterou vidí vždy alespoň tři kamery (tzn. prostor, ve kterém lze snímat pozice těles). V obou případech si lze všimnout nepokrytého prostoru u podlahy.



(a)



(b)

Obrázek 5.1: Pokrytí snímaného prostoru kamerami. (a) Prostor viděný minimálně jednou kamerou, (b) prostor viděný minimálně třemi kamerami. Menší čtverce mají hranu 20 cm, větší 1 m.

Snížená schopnost optického systému rozpoznat jednotlivá tělesa v krajních oblastech snímaného prostoru je ještě dále umocněna nepříliš kvalitní kalibrací. Bohužel s dostupným kalibračním programem se nedaří dosáhnout lepších výsledků – počet vzorků, které se nasnímají při kalibračním procesu, je omezen na 300, což je relativně malé množství. Následný výpočet nad vzorky, který určí pozice jednotlivých kamer v prostoru, je nedeterministický; při vícenásobném spuštění výpočtu nad těmi samými daty se dočkáme různých výsledků. Toto je pravděpodobně marketingový tah výrobce, který nabízí propracovanější kalibrační program v rámci jiného softwarového balíčku, ale za mnohem vyšší prodejní cenu.

Chceme-li identifikovat určitý bod v prostoru tak, aby byl tento bod nezaměnitelný s nějakým jiným, je potřeba použít nejméně tři reflexních kuliček, které musí být vůči sobě nepohyblivě upevněny (jejich vzájemné vzdálenosti musí být neměnné) [21]. Aplikace Rigid Body Tool (zkráceně RBT) [18] dovolí identifikovat tři nebo více odrazivých bodů jako jedno těleso (tzv. Rigid body). Při vytváření dalších těles musíme použít unikátní vzájemnou polohu reflexních kuliček, aby během procesu rozpoznávání nedocházelo k záměně jednotlivých těles. Při streamování je pak po síti posílána rotace a pozice těžiště každého z nich. Na [obrázku 5.2](#) jsou znázorněny plastové držáky, umožňující různá upevnění tří reflexních kuliček tak, aby jejich vzájemné polohy byly unikátní.



Obrázek 5.2: Plastové držáky s odrazivými kuličkami. Převzato z⁸.

Streamování pozic jednotlivých odrazivých bodů není možné. Vzhledem k tomu, že u pasivních optických systémů jsou dvě různé reflexní značky navzájem nerozlišitelné (aktivní systémy toto omezení nemají, viz [kapitola 3](#)), aplikace pro rekonstrukci pohybu by dostávala pouze pole třírozměrných souřadnic a sama by musela určovat, která souřadnice odpovídá které části těla. To by vyžadovalo velice komplexní algoritmus a při použití nedostatečného počtu odrazivých bodů by mohlo snadno docházet k jejich vzájemné záměně. Nelze tedy říci, že nemožnost streamování jednotlivých bodů je příliš velkou nevýhodou.

Tou je však chování aplikace RBT, která občas na krátký časový úsek zamění pozice dvou odlišných těles, nebo mylně určí jejich rotaci. Další problém nastane, když se například těleso určující pozici lýtka uživatele dostane mimo zorný úhel kamer. Implementovaný algoritmus se jeho tvar následně snaží hledat ve tvarech jiných těles a na krátké okamžiky se mu to daří. Podobná situace nastane, když je jedna nebo více reflexních kuliček v zorném poli zakryta jinou částí těla a kamery tak přestanou dané těleso rozpoznávat. Lýtko virtuální postavy pak dostává nesprávné souřadnice udávající jeho pozici a snaží se na tyto souřadnice přemístit, což vede ke zkroucení virtuální postavičky do krkolomných pozic.

Tyto problémy by na straně snímacího zařízení byly s velkou pravděpodobností vyřešeny kvalitní kalibrací a zvětšením snímaného prostoru; oboje je však za současných podmínek téměř vyloučeno. Na straně aplikace přijímající pohybová data se řešením zdá být inteligentní filtr, který vezme v úvahu pouze správné souřadnice. Tato problematika je rozebírána dále v této kapitole.

Poměrně vysoký rozptyl (viz [kapitola 3](#)) optického systému se podařilo kompenzovat pomocí zprůměrování určitého počtu posledních pozic (typicky tři až deseti). Čím více pozic je průměrováno, tím více je pohyb virtuální postavy oproti uživateli zpomalen. Za tento nápad děkuji Mgr. Jiřímu Chmelíkovi, který používá stejnou techniku ve svém programu.

Také samotný RBT program bychom jen s těžší nazvali robustním. Aplikace má tendence přestat odpovídat při spuštění a neočekávaně ukončit svůj běh při přidání většího počtu těles; při minimalizaci okna programu pak dojde k přerušení streamování pohybových dat po síti. Všechny tyto nedostatky ztěžují vývoj a následné používání programů pracujících s pohybovými daty získávanými ze systému OptiTrack.

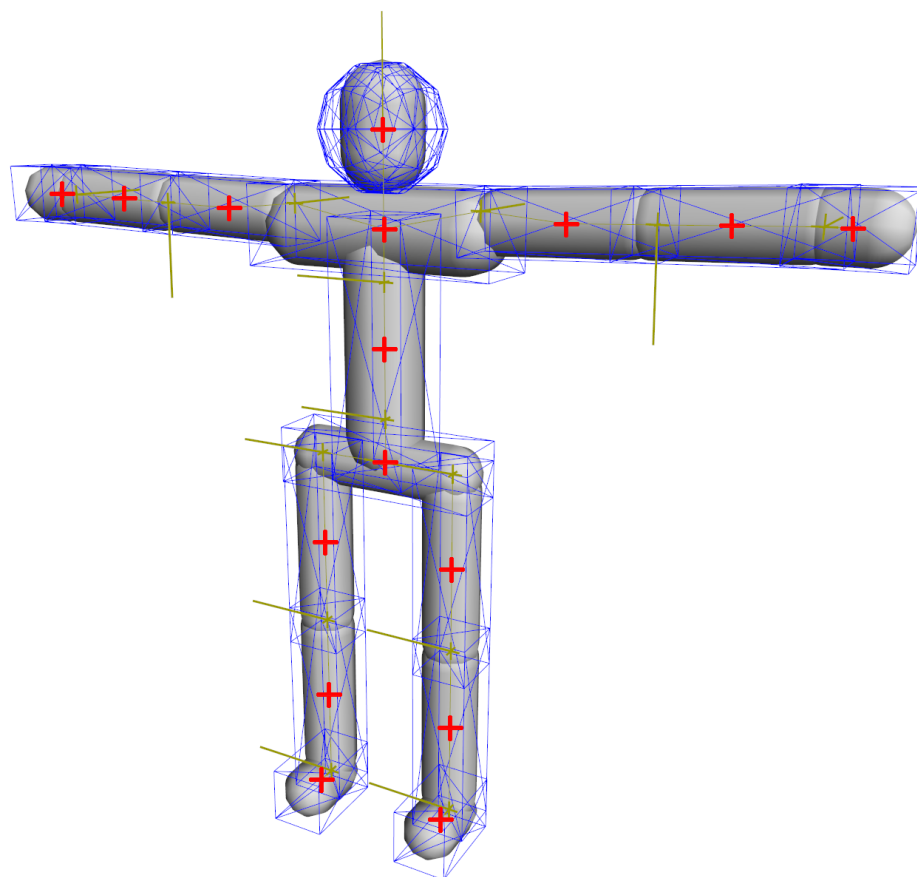
V době těsně před dokončením této práce byl vydán aktualizovaný balík programů Rigid Body Toolkit. Nová verze aplikace Rigid Body Tool, která je součástí tohoto balíku, s sebou přináší mnoho změn, a nutno říci, že k lepšímu. Mimo jiné byly odstraněny dříve zmiňované chyby, byla přidána možnost nastavení velkého množství různých parametrů, vývojáři zapracovali na vzhledu i uživatelské přívětivosti. Bohužel zabudovaný VRPN server se při streamování dat po síti chová mírně odlišně, takže mnou naprogramovaná aplikace MoCap není na tuto verzi programu Rigid Body Tool zcela přizpůsobena.

5.2 Počet snímaných pozic

Aby mohla virtuální postava dobře reflektovat uživatelské pohyby, musí mít stejný počet kloubů přibližně na stejných místech, jako má člověk. A aby mohly být pohyby rekonstruovány přesně, musíme pozice všech kloubů a dalších ohyblivých částí těla sledovat a přenášet je na virtuální postavu. Na [obrázku 5.3](#) je vyobrazen schematický model člověka, který jsem použil k rekonstrukci pohybů. I přes značné zjednodušení oproti skutečné postavě má tento model 12 kloubů, tři pohyblivá místa v trupu a dalším pohyblivým bodem je hlava. Profesionální systémy přitom používají standardně 18 a více sledovaných pozic [\[22\]](#). Vzhledem k problematickému uchycení jednotlivých těles přesně na klouby člověka jsem raději zvolil možnost snímání středů neohybatelných částí těla, na schematické postavě zobrazených jako červené křížky.

Počet sledovaných bodů není omezen, takže teoreticky by bylo možné určovat pozice všech šestnácti pohyblivých míst. Musíme však brát v potaz již dříve zmíněný fakt, že s přibývajícím počtem sledovaných těles vzrůstá pravděpodobnost, že na krátký okamžik dojde k záměně některých z nich. Další překážkou je nedostatek plastových držáků (viz [obrázek 5.2](#)) pro reflexní značky, problematické vymýšlení velkého počtu unikátních seskupení odrazivých značek a v neposlední řadě také určité nepohodlí uživatele – čím více sledovaných pozic, tím více těles musí mít na svém těle připevněno.

V současné době se pět sledovaných částí těla (hlava, dlaně, lýtka) tak zdá být maximem. Je tedy zřejmé, že pro pohyblivé části těla, u kterých nejsou určovány pozice (ať už úmyslně, nebo neúmyslně při zmizení jedné či více kuliček ze zorného pole kamer) musí být třidimenzionální souřadnice dopočítávány uměle. Zde jsem s výhodou použil jME Physics 2, engine starající se o implementaci fyziky. Podrobnější popis způsobu použití najdete v [šesté kapitole](#).



Obrázek 5.3: Schematický model člověka.

5.3 Inteligentní filtr dat

Pokud má program Rigid Body Tool zaregistrováno určité těleso a je zapnuto streamování, posílá 3D souřadnice těžiště tělesa. Pokud těleso není v záběru kamer, jsou VRPN serverem posílány souřadnice počátku souřadné soustavy. Klientská aplikace tedy takto může rozpoznat, zda je těleso v daném okamžiku sledováno nebo ne. Nelze ale přímo rozpoznat, zda jsou posílané pozice tělesa správné a nedošlo pouze k dočasné záměně.

Při snímkovací frekvenci 100 Hz jsou změny třídimenzionálních souřadnic jednotlivých bodů relativně malé i při velmi rychlých pohybech. Nabízí se tedy, že budeme kontrolovat rozdíly mezi posledními dvěma souřadnicemi; při velkých hodnotách tyto souřadnice budeme ignorovat, akceptovány budou pouze malé změny pozic oproti souřadnicím předchozím. Pokud tedy dojde na několik málo snímků k záměně jednoho tělesa za druhé, dostatečná vzdálenost mezi tělesy zajistí, že nesprávné souřadnice nebudou akceptovány. Toto řešení by bezchybně fungovalo v případě, že nedojde k přerušení sledování tělesa z dříve zmiňovaných důvodů. Tomu však v realitě nelze zabránit.

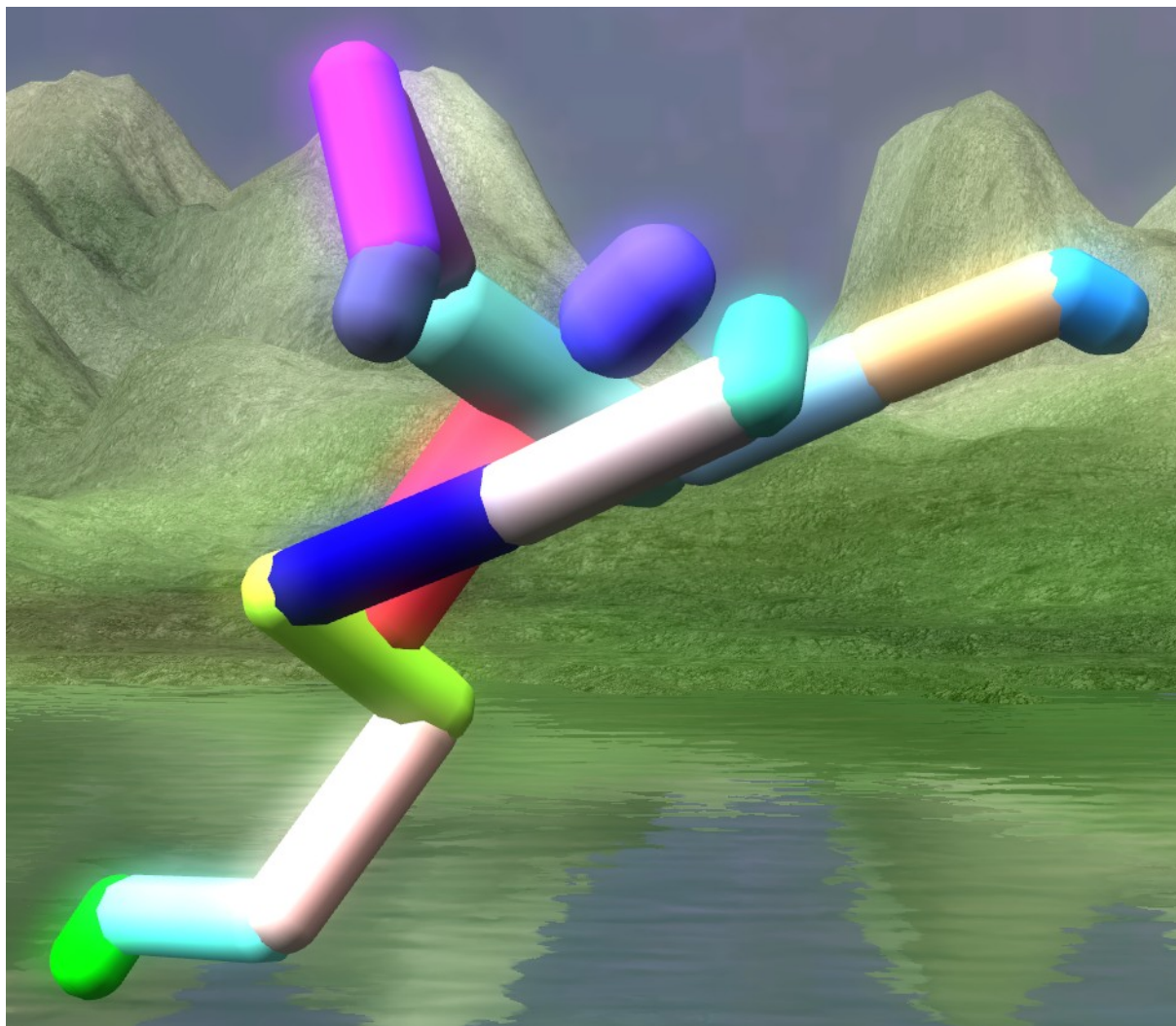
Pokud se tak stane, souřadnice počátku jsou samozřejmě ignorovány a řízení pohybu dané části těla získává plně na starost fyzikální engine. V momentě, kdy se obnoví proud souřadnic jiných než $[0;0;0]$, nelze na základě porovnání s posledními platnými souřadnicemi poznat, zda jsou tyto pozice správné – uživatel mohl určitou část těla v době, kdy její pozice nebyla rozpoznána, přemístit jinam.

Tento typ filtru je v mé aplikaci implementován. Rozdíly dvou pozic kontroluje pouze v případě, že předchozí pozice byla taktéž platná (tedy mimo jiné nebyla $[0;0;0]$). Pokud by filtr zahodil v řadě za sebou více než dvacet pozic, na následující souřadnice se neaplikuje – mohlo by se stát, že bychom mezi dvěma snímky kamer překonali kritickou vzdálenost a filtr by pak ignoroval veškeré příliš vzdálené pozice. Po dvaceti ignorovaných snímcích jsou tedy nové souřadnice prohlášeny za platné, i když by jinak byly zahozeny.

Největší problém, kdy aplikace Rigid Body Tool na okamžik rozezná seskupení reflexních kuliček, které není momentálně v zorném poli kamer, ve tvaru jiného tělesa, a začne tedy v tomto časovém úseku pro dané seskupení odrazivých bodů posílat nesprávné pozice, se však takto vyřešit nedá. Závažnost této problematiky si lze uvědomit, když uvážíme situaci, ve které například chodidlo virtuální postavy (doteď plně ovládané pomocí jPhysics) najednou začne dostávat souřadnice, které ve skutečnosti patří hlavě. Chodidlo tak má ve zlomku sekundy urazit vzdálenost přibližně dvou metrů. Fyzikální engine, který zajišťuje veškeré pohyby objektů pomocí silového působení na ně, tak vygeneruje vskutku impozantní sílu působící na danou část těla, která ji během několika málo setin vteřiny na požadované souřadnice přemístí. Ve správný okamžik je generována „protisíla“, která chodidlo zase zastaví. Toto silové působení je však distribuováno do zbytku celého těla, kde se projevuje prudkými pohyby zvláště těch částí těla, které jsou plně kontrolovány fyzikálním enginem (a jejich pozice není určována pomocí kamer). Nejedná se tedy jen o chvilkové statické přemístění chodidla na nesprávné místo. Obrovská dynamika volných částí těla, kterým určitou dobu trvá, než se uklidní, na uživatele působí rušivým dojmem (obrázek 5.4).

Dalším krokem, který připadá v úvahu při řešení tohoto problému, by bylo porovnávání nově přichozích souřadnic s pozicemi, které reprezentují jiný „díl“ stejné části těla – např. souřadnice chodidla by mohly být porovnávány se souřadnicemi lýtky a stehna; stejně tak platnost pozice dlaně lze určit pomocí pozic předloktí a ramene. Už samotný algoritmus provádějící toto porovnání by musel být velmi sofistikovaný (musíme vzít v potaz ohyblivost jednotlivých kloubů a na ní záležející akční rádius končetin). Avšak vzhledem k tomu, že v současnosti lze použít pro jednu končetinu maximálně jedno těleso, i správně implementovaný filtrační algoritmus by neměl k dispozici v podstatě žádná data, se kterými by mohl porovnávat.

Celý problém se mi nepodařilo úplně eliminovat; z části jej ale řeší zahození několika prvních pozic (typicky 50), které přicházejí po neplatných souřadnicích počátku souřadné soustavy. Pokud tedy aplikace Rigid Body Tool na dostatečně krátký čas „objeví“ tvar momentálně nesnímaného tělesa (řekněme například dlaně) v nějakém jiném seskupení značek, a po určitém čase zase správně rozezná pouze tvar původní a těleso reprezentující dlaň už ne, tato chyba se do pohybu postavy vůbec nezaznamená.



Obrázek 5.4: Distribuce silového působení postavou. Těleso určující pozice pravého lýtky bylo zaměněno za těleso patřící hlavě.

6 Implementace

6.1 Prostředí

Virtuální postava je umístěna do hornatého prostředí obklopeného vodními plochami ([obrázek 6.1](#)). Terén je generován na základě výškové mapy (při spuštění aplikace se náhodně načítá jedna z dostupných map), vodní hladina je znázorněna pohybující se texturou. Ve vodní hladině se také odráží okolní prostředí včetně postavičky – hladina horizontálně rozděluje prostředí na část, která se má zrcadlit, a část, která je pod vodou a tudíž se zrcadlit nemá. Toto rozdělení lze vidět ve dvou pomocných oknech při zapnutí ladicího módu. Výšku vodní hladiny lze dynamicky měnit, také si lze vybrat ze dvou barevných odstínů vody (popis ovládání se nachází v [dodatku B](#)).



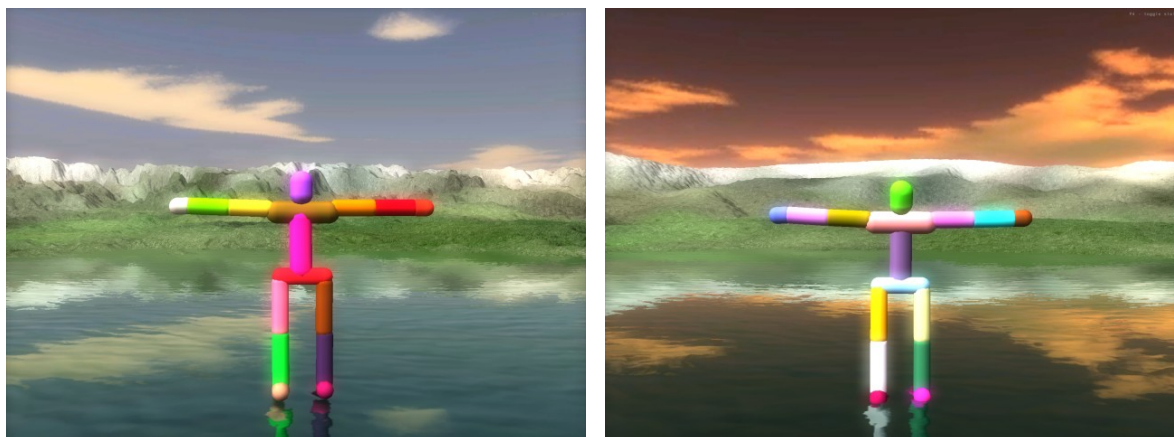
Obrázek 6.1: Prostředí, ve kterém je umístěna virtuální postava.

Pro základní demonstraci možností fyzikálního enginu a interakce člověka s předměty ve virtuální realitě lze nechat před uživatelem objevit jeden nebo více gumových míčků, které lze odkopnout či jinak s nimi pohybovat.

Ve scéně je umístěno jedno světlo směrové a jedno bodové. Bodové světlo lze snadno identifikovat pomocí efektu odlesků, které okolo světla vznikají. Vzdálené okolí je vytvořeno metodou Cube Mapping [23]. Na vnitřní strany pomyslné krychle jsou naneseny textury jednotlivých světových stran, nebe a země. Kamera je umístěna přesně uprostřed, při pohybu se pomyslná krychle hýbe s kamerou. Textury mají nastavenou stoprocentní průhlednost, takže skrz ně vidíme i objekty, které jsou ve větší vzdálenosti, než je poloměr krychle. Tento koncept vzbuzuje dojem, jakoby krajina vyobrazená na texturách byla velmi vzdálená. Při každém spuštění aplikace se náhodně vybere jiné pozadí, které lze však měnit i za běhu programu.

6.2 Virtuální postava

Jednotlivé části virtuální postavy (všechny tvořeny válci) jsou vůči sobě navzájem pohyblivě uchyceny. Spoje jsou limitovány v pohybu svých částí tak, aby co nejvěrněji kopírovaly ohybové možnosti jednotlivých kloubů člověka. Například spoj představující kolenní kloub tak umožní rotaci lýtkové části vůči stehnu pouze v úhlu 180° , a to ve směru, jakým se koleno přirozeně ohýbá u člověka. Jednotlivé části postavy jsou náhodně obarveny; za běhu programu lze generovat další náhodné obarvení, nebo vybrat jednu z dvanácti barev pro celou postavu. Ukázku náhodného výběru terénu, oblohy a náhodného zbarvení postavy lze vidět na [obrázku 6.2](#).



Obrázek 6.2: Náhodné generování barev a výběr okolí.

Třírozměrné souřadnice jsou mapovány na pomocné uzly, které jsou viditelné pouze při zapnutí zobrazování fyzikálních interakcí. Mezi těmito uzly a jím příslušnými částmi těla jsou vytvořena pružná spojení. Tato spojení pak při pohybu pomocných bodů způsobují, že se generují patřičné síly, které pohybují se samotnou postavou. V momentě, kdy začnou přicházet neplatné souřadnice, je dané spojení zrušeno a jemu příslušná část postavy už není přitahována k pomocnému uzlu. Pružnost spojení lze nastavit; obecně platí, že čím je spojení volnější, tím jsou pohyby samotné postavy plynulejší, ale také méně přesné oproti pohybům uživatele. Při velmi silných spojeních pohyb postavy naopak přestává být plynulý.

Různí lidé mají různou výšku a rozpětí rukou, nelze tedy použít stejnou konfiguraci pro dospělého člověka, jako pro dítě. Vzhledem k tomu, že virtuální postava má pevně danou velikost, musejí být patřičně upraveny přicházející souřadnice. Pro tuto potřebu se v ladicím módu zobrazují hodnoty tzv. multiplikátoru a výškového multiplikátoru, které lze dynamicky měnit. Multiplikátor násobí hodnoty všech přichozích souřadnic. Čím větší je hodnota multiplikátoru, tím menšího vzrůstu uživatel musí být, aby došlo k zachování proporcí. Při správném nastavení se tedy nestane, že u člověka s menším vzrůstem, který stojí vzpřímeně, bude mít virtuální postava mírně pokrčená kolena (obdobně s rukama při upažení). Výškový multiplikátor pak slouží k jemnému doladění velikosti y-hodnot (tedy výšky) vůči skutečnosti nezávisle na x-ové a z-ové souřadnici.

6.3 Mapování snímaných pozic na postavu

Aplikace je koncipována tak, že lze použít libovolný počet snímaných pozic v rozmezí od jedné (nepoužití žádné postrádá význam, ale principiálně je také možné) do šestnácti. Teoreticky lze tedy snímat každou (dále již neohebnou) část těla a získat tak dokonalou rekonstrukci uživatelových pohybů. Problémy objevujícími se v praxi, které použití velkého množství snímaných částí v podstatě vylučují, se zabývá [pátá kapitola](#). Mapování jednotlivých těles na části virtuální postavy je rozepsáno v [dodatku B](#).

Při zapnutí ladicího módu (je také vhodné zapnout pauzu, aby se postava přestala hýbat) lze však uzly libovolně přemapovat. Stačí vybrat uzel s požadovaným číslem a kliknout myší na tu část těla, kterou mu chceme přiřadit.

Kromě optického polohovacího systému lze jako další vstupní zařízení určující polohu jedné části virtuální postavy (nebo i gumových míčků) použít kurzor myši. Při umístění kurzoru na určitou část postavy a stlačení levého tlačítka myši se použijí souřadnice kurzoru myši a daná část se začne pohybovat (resp. začne sledovat kurzor myši) stejně, jako kdyby získávala pozice z kamerového systému.

7 Závěr

Výsledkem této práce je aplikace umožňující rekonstrukci pohybů uživatele snímaných pomocí kamerového systému OptiTrack. Lze ji použít jako jednoduchou a snadno pochopitelnou ukázkou využití optického systému pro snímání pohybu instalovaného v laboratoři HCI.

Aplikace samozřejmě není dokonalá, v řadě případů jsem musel přistoupit na kompromisy, které zcela nesplňovaly moje představy. Ve vývoji však hodlám intenzivně pokračovat, program se tak snad časem dočká dalších hodnotných rozšíření a možností využití.

7.1 Možnosti rozšíření

Výsledný program používá pouze schematický model znázorňující člověka. Pravděpodobně nejužitečnějším rozšířením by byla možnost použití složitějších modelů, více podobných člověku. V případě dodržení návrhu kostry a formátu modelu by dokonce případní zájemci nebyli limitováni počtem předpřipravených modelů a mohli by si vytvořit modely vlastní. Práce na možnosti použití velmi realistických modelů formátu md5 již byla započata; bohužel se objevily problémy při jejich zpracování jMonkey Enginem, které se do termínu odevzdání této práce nepodařilo vyřešit.

Dalším hodnotným rozšířením by byla možnost zapnout stereovizi (v HCI laboratoři dostupnou díky instalované stereoprojekci). Bohužel i zde zatím narážím na omezení jME, který momentálně nedovoluje vytvořit dvě nezávislá okna, ve kterých by se vykresloval obraz zvlášť pro levé a zvlášť pro pravé oko. O tomto problému je na jMonkey Engine fóru hojně diskutováno a je pouze otázkou času, než bude podpora stereovize implementována.

Při častém používání programu by uživatel jistě ocenil přívětivější grafické menu s možnostmi nastavení jednotlivých parametrů; ty by se pak měly ukládat do konfiguračního souboru, aby se program při dalším spuštění nemusel nastavovat znovu.

Jako perličku nakonec uvádím možnost lepší prezentace širokých grafických dovedností, kterých je jME schopen. Nabízí se například reálněji zpracovaná voda, použití měkkých stínů, normálových map nebo částicových systémů.

Literatura

- [01] Powell, Mark aj. *jMonkey Engine*, leden 2009.
Dostupné na URL <<http://www.jmonkeyengine.com/>> (15. května 2009)
- [02] Schneider, Christian aj. *jMonkey Physics 2*, prosinec 2008.
Dostupné na URL <<https://jmephysics.dev.java.net/>> (15. května 2009)
- [03] Taylor, Russell M. II aj. *VRPN: A Device-Independent, Network-Transparent VR Peripheral System*. Proceedings of the ACM Symposium on Virtual Reality Software & Technology 2001, VRST 2001. Banff Centre, Canada, listopad 2001.
Dostupné na URL <http://www.cs.unc.edu/~seeger/publications/2001taylor_vrpn.pdf> (15. května 2009)
- [04] Wikipedia contributors. *Motion capture*. Wikipedia, The Free Encyclopedia, květen 2009. Dostupné na URL <http://en.wikipedia.org/wiki/Motion_Capture> (15. května 2009)
- [05] Furniss, Maureen. *Motion capture*. MIT communications forum, prosinec 1999.
Dostupné na URL <<http://web.mit.edu/comm-forum/papers/furniss.html>> (15. května 2009)
- [06] Gleicher, Michael. *Animation From Observation: Motion Capture and Motion Editing*. University of Wisconsin, Madison, duben 2001.
Dostupné na URL <<http://www.cs.wisc.edu/graphics/Papers/Gleicher/MocapMusings/11Z-Gleicher-AnimObsArticle.pdf>> (15. května 2009)
- [07] Welch, Greg; Foxlin, Eric. *Motion tracking: No silver bullet, but a respectable arsenal*. IEEE Computer Graphics and Applications, Volume 22, Issue 6, Nov.-Dec. 2002, Page(s): 24-38.
Dostupné na URL <http://www.cs.unc.edu/~tracker/media/pdf/cga02_welch_tracking.pdf> (15. května 2009)
- [08] Menache, Alberto. *Understanding Motion Capture for Computer Animation and Video Games*. San Francisco, CA, USA, Morgan Kaufmann Publishers Inc, 1999.
- [09] King, Brett Allan; Paulson, Linda Dailey. *Motion Capture Moves into New Realms*. IEEE Computer, Volume 40, Issue 9, Sept. 2007, Page(s): 13-16. Dostupné na URL <<http://ieeexplore.ieee.org/iel5/2/4302594/04302605.pdf?arnumber=4302605>> (15. května 2009)

- [10] Anisfield, Nancy. *The Rise of a New Art Form – The Birth of MoCap*. Ascension Technology Corporation, říjen 2008. Dostupné na URL [<www.ascension-tech.com/docs/pdf/birth_of_mocap.pdf>](http://www.ascension-tech.com/docs/pdf/birth_of_mocap.pdf) (15. května 2009)
- [11] Dickson, Lee. *Motion Capture Uses*. Meta Motion, únor 2008. Dostupné na URL [<http://www.metamotion.com/motion-capture-uses/content-creation.html>](http://www.metamotion.com/motion-capture-uses/content-creation.html) (15. května 2009)
- [12] Thomson, Scott. *Motion Capture History*. Motion Capture Resources, leden 2008. Dostupné na URL [<http://www.motion-capture-system.com/resources/history.html>](http://www.motion-capture-system.com/resources/history.html) (15. května 2009)
- [13] Jenkins, Ward. *The Polar Express: A Virtual Train Wreck*, prosinec 2004. Dostupné na URL [<http://wardomatic.blogspot.com/2004/12/polar-express-virtual-train-wreck_18.html>](http://wardomatic.blogspot.com/2004/12/polar-express-virtual-train-wreck_18.html) (15. května 2009)
- [14] Burdea, Grigore C.; Coiffet, Philippe. *Virtual Reality Technology*. Edition II, Wiley-IEEE Press, červenec 2003.
- [15] Stanney, Kay Michael. *Handbook of virtual environments: design, implementation, and applications*. Lawrence Erlbaum Associates, 2002.
- [16] Brožková, Lucie. *Education program for calligraphy using HCI devices* [diplomová práce], Brno, Fakulta informatiky Masarykovy university, podzim 2008. Dostupné na URL [<http://is.muni.cz/th/98745/fi_m/thesis.pdf>](http://is.muni.cz/th/98745/fi_m/thesis.pdf) (15. května 2009)
- [17] Guerra-Filhol, Gutemberg B. *Optical motion capture: Theory and implementation*. Tutorial XVIII Brazilian Symposium on Computer Graphics and Image Processing 2005, RITA 2005. Volume 12, Issue 2, 2005. Dostupné na URL [<http://ranger.uta.edu/~guerra/Guerra-FilhoRITA05optical.pdf>](http://ranger.uta.edu/~guerra/Guerra-FilhoRITA05optical.pdf) (15. května 2009)
- [18] NaturalPoint developers. *Rigid Body Toolkit*, Natural Point, únor 2008. Dostupné na URL [<http://www.naturalpoint.com/optitrack/products/rigid-body.html>](http://www.naturalpoint.com/optitrack/products/rigid-body.html) (15. května 2009)
- [19] Eberly, David H. *3D Game Engine Design: A Practical Approach to Real-time Computer Graphics*. San Francisco, CA, USA, Morgan Kaufmann Publishers Inc, 2001.
- [20] Powell, Mark aj. *jMonkey Features List*, leden 2009. Dostupné na URL [<http://www.jmonkeyengine.com/wiki/doku.php?id=complete_features_list>](http://www.jmonkeyengine.com/wiki/doku.php?id=complete_features_list) (15. května 2009)
- [21] NaturalPoint developers. *NaturalPoint Tracking Toolkits (Point Cloud and Rigid Body) Users Manual*. Natural Point, únor 2008. Dostupné na URL [<http://www.naturalpoint.com/optitrack/support/manuals/point-cloud/PointCloud_users_manual.pdf>](http://www.naturalpoint.com/optitrack/support/manuals/point-cloud/PointCloud_users_manual.pdf) (15. května 2009)

- [22] Bodenheimer, Bobby; Rose, Charles; Rosenthal, Seth; Pella, John. *The Process of Motion Capture: Dealing with the Data*. In Computer Animation and Simulation '97, Eurographics Animation Workshop, D. Thalmann and M. van de Panne, eds., Springer-Verlag, Wien, Sept. 1997, Page(s) 3-18. Dostupné na URL
<<http://www.vuse.vanderbilt.edu/~bobbyb/pubs/dealingdata97.pdf>> (15. května 2009)
- [23] Wesley, Addison. *Environment Mapping Techniques*. In The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics, Randima Fernando and Mark J. Kilgard, Addison Wesley Professional, Feb. 2003, Chapter 7. Dostupné na URL
<<http://www.developer.com/lang/other/article.php/2169281>> (15. května 2009)

A Doplňující snímky



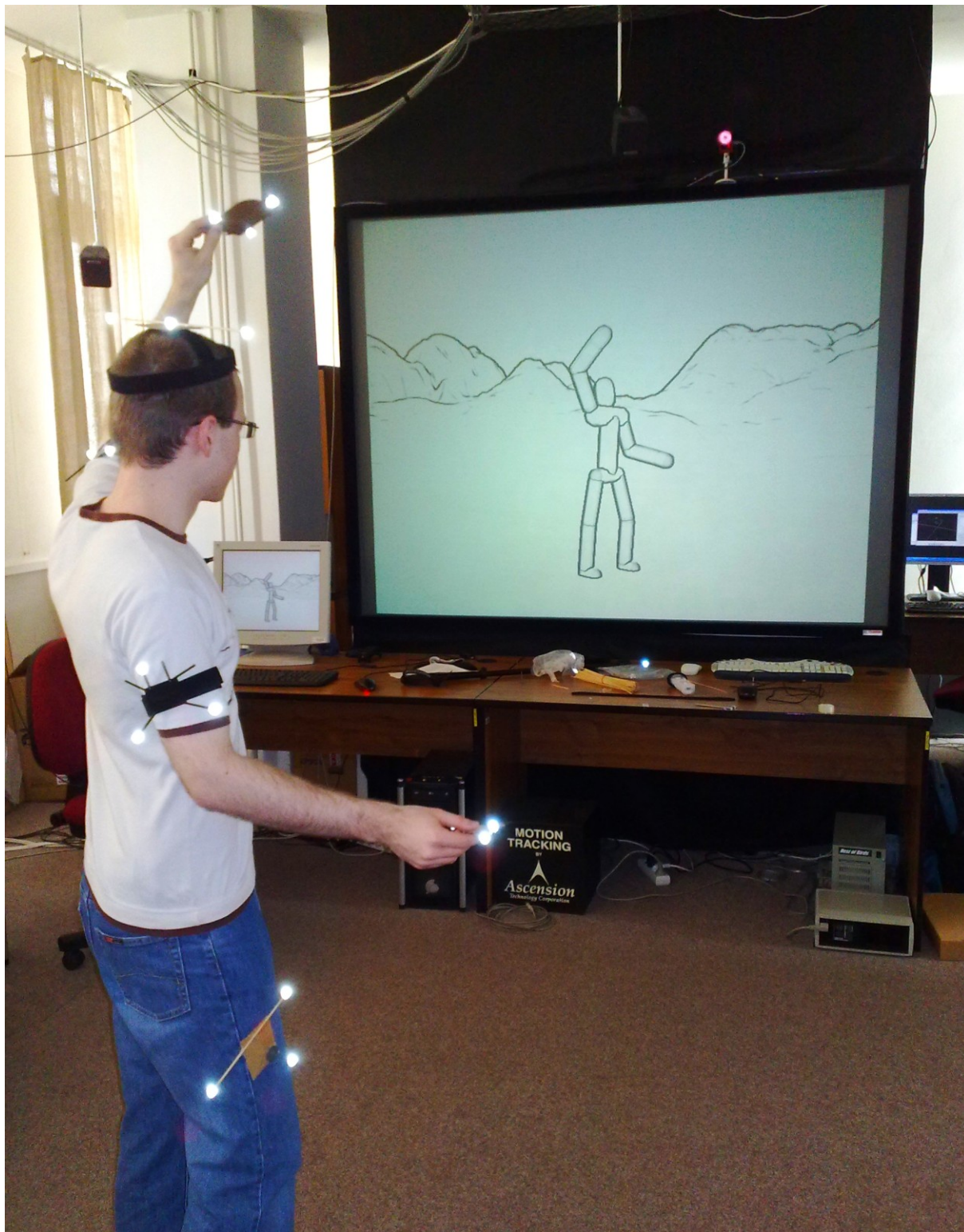
Obrázek A.1: Ukázka snímání a rekonstrukce pohybu postavy.



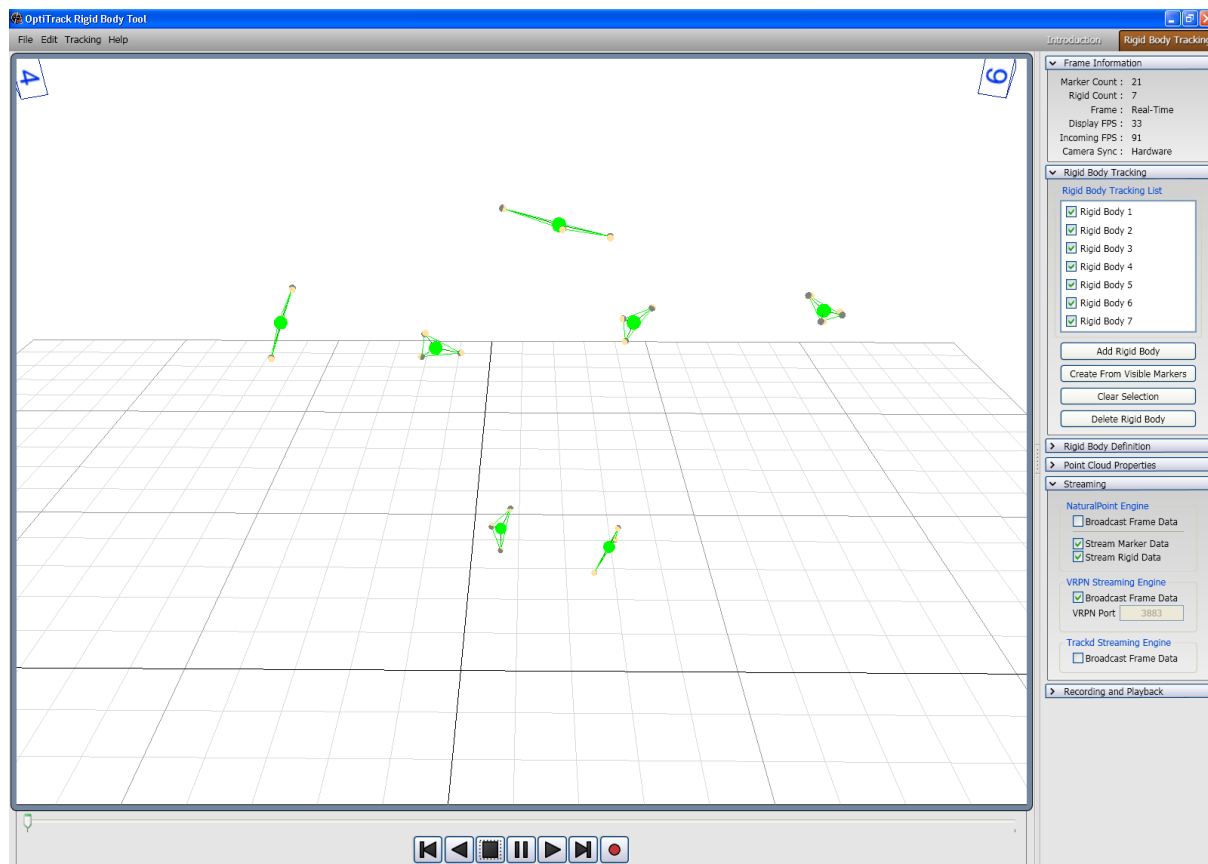
Obrázek A.2: Ukázka snímání a rekonstrukce pohybu postavy.



Obrázek A.3: Ukázka snímání a rekonstrukce pohybu postavy. Zapnut ladicí mód.



Obrázek A.4: Ukázka snímání a rekonstrukce pohybu postavy. Zapnut mód „Skica“.



Obrázek A.5: Snímek aplikace Rigid Body Tool při streamování pozic sedmi těles (Rigid Body 1 až Rigid Body 7).

B Uživatelský manuál

B.1 Začínáme

Tento manuál slouží k seznámení uživatele se mnou vytvořenou aplikací MoCap, jejím ovládáním a konfigurací software Rigid Body Tool pro potřeby této aplikace. Manuál je psán implicitně pro operační systém Microsoft Windows XP, s menšími změnami je však aplikovatelný i na jiné systémy.

B.1.1 Hardwarové nároky

Vzhledem k výkonnému hardware dostupnému v laboratoři HCI nebyl program nijak optimalizován pro běh na slabších počítačích. Pro správné vykreslování scény je zapotřebí grafické karty s plnou podporou OpenGL 2.1, pro funkčnost programu však není nutná. Pokud je aplikace spuštěna na kartě bez plné podpory OpenGL 2.1, ve spodním rohu se objeví nápis „GLSL není na tomto počítači podporován“, ve vodní hladině se nebude zrcadlit okolní prostředí, nebude dostupný efekt nasvícení povrchů (tzv. „bloom“) a mód „Skica“.

Testované sestavy:

konfigurace č. 1:

Intel Core 2 Quad CPU Q6600 @ 2.40 GHz
NVIDIA GeForce 8600GT 512MB VRAM
4 GB RAM, Microsoft Windows XP Professional SP3

nastavení programu:

rozlišení: 1600*1200
antialiasing: 4x
anisotropní filtrování: 16x

průměrný počet snímků za vteřinu: 20

konfigurace č. 1:

Intel Core 2 Quad CPU Q9550 @ 2.83 GHz
NVIDIA GeForce GTX280 1024MB VRAM
4 GB RAM, Microsoft Windows Vista Business SP1

nastavení programu:

rozlišení: 1600*1200
antialiasing: 4x
anisotropní filtrování: 16x

průměrný počet snímků za vteřinu: 140

Hardwarovou náročnost lze výrazně snížit vypnutím čtyřnásobného antialiasingu (v současné době je toto možné pouze zakomentováním příslušného řádku ve zdrojovém kódu aplikace) či zmenšením rozlišení (lze zvolit při každém spuštění aplikace).

B.2 Spuštění aplikace

Uživatel má k dispozici aplikaci jednak v podobě projektu se zdrojovými kódy pro vývojové prostředí NetBeans a jednak v podobě zkompilovaného jar souboru.

B.2.1 NetBeans projekt

Projekt byl vytvářen ve vývojovém prostředí NetBeans 6.5, předpokládá se kompatibilita i s novějšími verzemi.

Aby bylo možno spustit projekt, je potřeba mít nainstalováno:

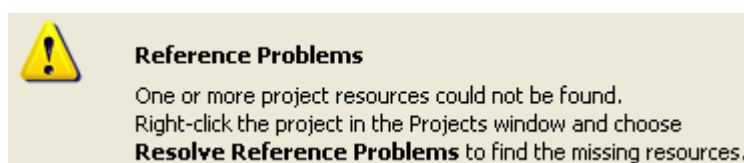
- NetBeans IDE (lze stáhnout na URL <<http://netbeans.org/>>);
- JDK (lze stáhnout na URL <<http://java.sun.com/javase/downloads/index.jsp>>).

B.2.1.1 Otevření projektu

- spustíme Netbeans, v nabídce [File] zvolíme [Open Project...], ve stromové struktuře nalezneme cestu k projektu, projekt vybereme;
- ponecháme zatrženo [Open as main project], zvolíme [Open Project].

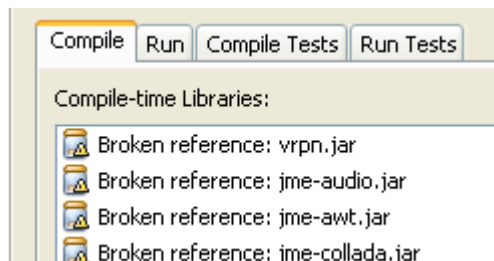
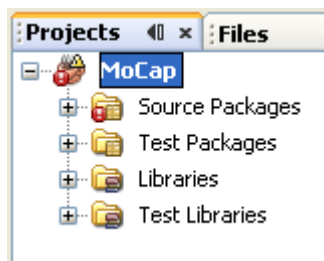
B.2.1.2 Kontrola knihoven

Pokud se u nově otevřeného projektu neobjeví žlutý či červený vykřičník, je vše v pořádku. Pokračujeme bodem 2.4. V opačném případě se pravděpodobně objeví chybová hláška:

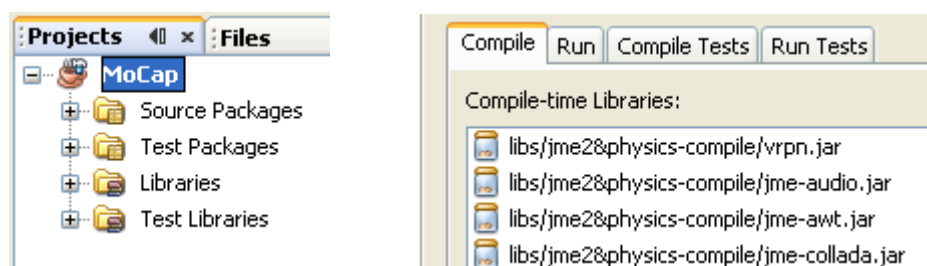


Zkontrolujeme cesty ke knihovnám:

- klikneme pravým tlačítkem myši na projekt, vybereme [Properties];
- vybereme kategorii [Libraries] vlevo, záložku [Compile] vpravo;
- zkontrolujeme, zda NetBeans našly všechny uvedené knihovny (u žádných není žlutý vykřičník);



- pokud tomu tak není, označíme všechny knihovny a pomocí tlačítka [Remove] je smažeme, dále klikneme na tlačítko [Add JAR/Folder], otevřeme adresář *jme2&physics-compile*, který se nachází ve složce *libs* v projektu MoCap, a zde vybereme všechny jar soubory a pomocí tlačítka [Open] je přidáme;
- obdobně postupujeme pro záložku Run, kde vybíráme adresář *jme2&physics-run*;
- potvrdíme stisknutím [OK], v tuto chvíli by měl výstražný trojúhelník zmizet.



B.2.1.3 Kompilace projektu

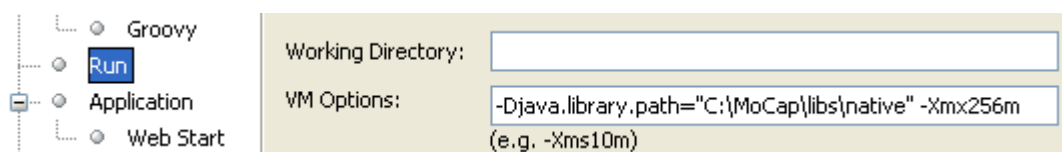
Klikneme pravým tlačítkem myši na projekt a zvolíme [Clean and Build]. Pokud proběhla kompilace bez chyb, je projekt téměř připraven ke spuštění. V opačném případě bude chyba nejspíše ve špatně definovaných knihovnách (viz část B.2.1.2).

B.2.1.4 Nastavení parametrů Virtual Machine

- klikneme pravým tlačítkem myši na projekt, vybereme [Properties];
- vlevo vybereme kategorii [Run];
- do pole [VM Options:] vložíme tento příkaz:

`-Djava.library.path="<cesta-nat>" -Xmx256m`

kde řetězec `<cesta-nat>` nahradíme cestou k nativním knihovnám umístěným ve složce *libs\native* v projektu Mocap. Pro konkrétní cestu pak příkaz může vypadat např. takto:



- potvrdíme stisknutím [OK].

B.2.1.5 Spuštění projektu

Klikneme pravým tlačítkem myši na projekt a zvolíme [Run]. Pokud je vše nastaveno správně, objeví se dialogové okno s logem a možnostmi zobrazení. Toto nastavení je rozebíráno v části B.3.

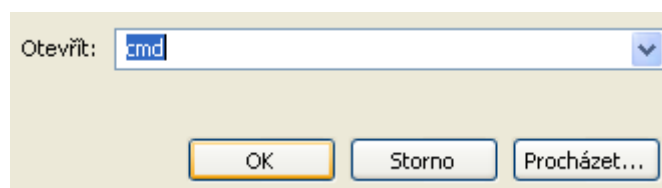
B.2.2 Zkompilovaný jar soubor

Aby bylo možno spustit aplikaci, je potřeba mít nainstalováno:

- JRE (lze stáhnout na URL [<http://java.sun.com/javase/downloads/index.jsp>](http://java.sun.com/javase/downloads/index.jsp)).

B.2.2.1 Spuštění z příkazového řádku

- klikneme na tlačítko [Start], z nabídky Start vybereme příkaz [Spustit...]
(alternativně použijeme klávesovou zkratku:  + R)
- do nově otevřeného dialogového okna napíšeme „cmd“, volbu potvrdíme



- do příkazového řádku napíšeme tento příkaz:

```
java -Djava.library.path="<cesta-nat>" -Xmx256m -jar "<cesta-proj>"
```

 (1)

kde řetězec *<cesta-nat>* nahradíme cestou k nativním knihovnám ve složce *native* v projektu MoCapCompiled, řetězec *<cesta-proj>* nahradíme cestou k souboru *MoCap.jar* ve složce projektu MoCapCompiled.

Příkaz s konkrétními cestami tedy může vypadat například takto:

```
java -Djava.library.path="C:\MoCapCompiled\native" -Xmx256m -jar "C:\MoCapCompiled\MoCap.jar"
```

Je důležité dodržet, aby parametr *-jar "<cesta-proj>"* byl uveden jako poslední (parametry za ním jsou ignorovány).

B.2.2.2 Spuštění pomocí zástupce

Spouštění zkompilovaného projektu lze usnadnit vytvořením zástupce.

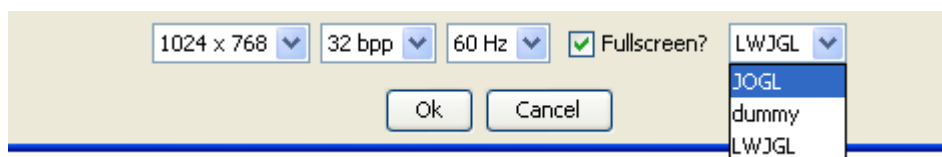
- klikneme pravým tlačítkem na plochu a vybereme možnost [Nový] → [Zástupce];
- v nově otevřeném dialogovém okně klikneme na [Procházet...] a najdeme cestu k souboru *MoCap.jar*;
- pokračujeme kliknutím na [Další] a [Dokončit];
- klikneme pravým tlačítkem na nově vytvořeného zástupce na ploše a vybereme [Vlastnosti];
- do pole „Cíl“ vložíme příkaz (1) popisovaný v části B.2.2.1; původní obsah tohoto pole tímto příkazem přepíšeme;
- potvrdíme stiskem tlačítka [OK].

Typ cíle:	Executable Jar File
Umístění cíle:	MoCapCompiled
Cíl:	<code>java -Djava.library.path="C:\MoCapCompiled\native" -Xmx256m -jar "C:\MoCapCompiled\Mocap.jar"</code>
Spustit v:	C:\MoCapCompiled
Klávesová zkratka:	není
Spustit:	V normálním okně ▼

Aplikaci lze poté spustit poklepnutím na zástupce.

B.3 Nastavení grafických parametrů

Při každém spuštění aplikace se nejprve objeví dialogové okno s logem aplikace MoCap, ve kterém lze nastavit rozlišení, barevnou hloubku, zobrazení na celé obrazovce/v okně, obnovovací frekvenci (pouze při zobrazení na celé obrazovce) a způsob vykreslování (JOGL/LWJGL).



B.4 Popis aplikace MoCap

V současné době je téměř veškeré ovládání aplikace řešeno pomocí klávesnice. Do budoucna je plánováno grafické menu, ve kterém by uživatel mohl nastavovat jednotlivé parametry pomocí posuvníků, checkboxů a tlačítek.

Aplikace má poměrně široké možnosti nastavení vzhledu prostředí a postavy, dále lze upravit mapování jednotlivých těles na příslušné části těla a multiplikátory měnící poměry velikostí mezi skutečným a virtuálním světem.

B.4.1 Mapování kláves

Pro stejné rozložení funkčních kláves (jako na obrázku dole) je nutné mít vybranou anglickou klávesnici. V opačném případě budou zaměněny pozice některých funkčních kláves na klávesnici (např. Z/Y, -/=, atd.)

B.4.1.1 Celkový pohled:



Legenda:

- pohyb kamery (+ pravé tlačítko myši)
- příkaz vypisující informace do konzole
- klávesa s přiřazenou funkcionalitou
- pro plynulou změnu daného parametru je klávesu nutné držet delší dobu
- obarvení postavy

B.4.1.2 Popis:

- A pohyb kamery doleva (nutno držet pravé tlačítko myši)
- C výpis pozice kamery na standardní výstup
- D pohyb kamery doprava (nutno držet pravé tlačítko myši)
- E zapnutí/vypnutí ladicího módu
- F přepínání anizotropního filtrování
- G znovunačtení shaderů vodní hladiny
- K zapnutí/vypnutí módu „skica“
- L zapnutí/vypnutí stínování
- N zapnutí/vypnutí zobrazování normálových vektorů pro povrchy
- P pauza (pozastavení fyzikálních interakcí, zejm. tedy pohybu postavy)
- Q pohyb kamery nahoru (nutno držet pravé tlačítko myši)
- R výpis informací o velikosti aktuální/volné/celkové paměti použité ve Virtual Machine
- S pohyb kamery dozadu (nutno držet pravé tlačítko myši)
- T zapnutí/vypnutí zobrazování drátěných modelů

V	zapnutí/vypnutí zobrazení geometrií a sil používaných fyzikálním enginem (červeně je vyobrazena gravitační síla, šedě součet ostatních sil působících na těleso)
W	pohyb kamery dopředu (nutno držet pravé tlačítko myši)
X	načtení následujícího pozadí (po posledním dostupném se znovu načte první)
Z	pohyb kamery dolů (nutno držet pravé tlačítko myši)
1..9	obarvení postavy barvami: růžová, žlutá, oranžová, červená, fialová, cyanová (modrozelená), modrá, zelená, hnědá (viz následující obrázek)

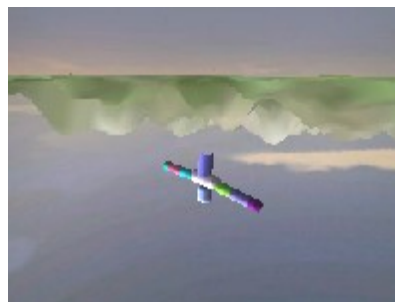
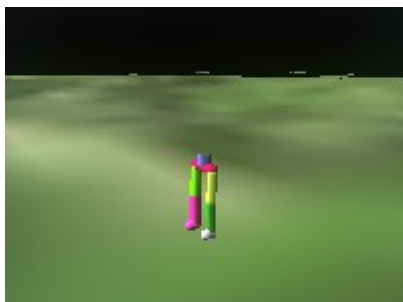


[	barva vodní hladiny (modravý nádech)
]	barva vodní hladiny (vínový nádech)
-	snížení intenzity svítivosti povrchů postavy (akt. hodnotu lze vidět v ladicím módu)
=	zvýšení intenzity svítivosti povrchů postavy (akt. hodnotu lze vidět v ladicím módu)
,	zmenšení zorného pole (aktuální hodnotu lze vidět v ladicím módu)
.	zvětšení zorného pole (aktuální hodnotu lze vidět v ladicím módu)
/	nastavení počáteční pozice kamery, nastavení zorného pole na výchozí hodnotu (45°)

Backspace	generování náhodného obarvení jednotlivých částí postavy
L Control	snížení vodní hladiny
L Shift	zvýšení vodní hladiny
mezerník	vytvoření nového gumového míčku
Delete	zmenšení multiplikátoru (aktuální hodnotu lze vidět v ladicím módu)
Insert	zvětšení multiplikátoru (aktuální hodnotu lze vidět v ladicím módu)
End	zmenšení výškového multiplikátoru (akt. hodnotu lze vidět v ladicím módu)
Home	zvětšení výškového multiplikátoru (akt. hodnotu lze vidět v ladicím módu)
Page Down	vybrat předchozí uzel pro přemapování (akt. vybraný uzel lze vidět v lad. m.)
Page Up	vybrat následující uzel pro přemapování (akt. vybraný uzel lze vidět v lad. m.)
Numpad -	snížení intenzity odlesků (aktuální hodnotu lze vidět v ladicím módu)
Numpad +	zvýšení intenzity odlesků (aktuální hodnotu lze vidět v ladicím módu)
Numpad *	nastavení výchozí hodnoty intenzity odlesků
F1	uložení snímku obrazovky do aktuální složky ve tvaru: <i>ScreenShot pořadové_číslo yyyy-MM-dd HH-mm-ss.png</i>
F2	zapnutí paralelní projekce
F4	zobrazení statistik
F6	přepnutí normálního/zrcadlového mapování pozic
F7	změna orientace souřadné osy x
F8	změna orientace souřadné osy z
Escape	ukončení aplikace

B.4.2 Ladicí mód

Při zapnutí ladicího módu se mimo jiné zobrazí dva pomocné pohledy, ve kterých lze vidět rozdělení scény vodní hladinou.



Obě okna znázorňují texture, které se (po určité deformaci) aplikují na rovinu hladiny. Tímto způsobem je realizován pohled na objekty pod vodou a zrcadlení okolí ve vodní hladině.

V levé horní části se zobrazí pomocné výpisy určitých parametrů, ty lze měnit příslušnými klávesami.

Vybrany uzel pro premapovani: 1 hlava
Vyskovy mltiplikator: 0.82

V tomto módu lze změnit výchozí mapování jednotlivých těles na části virtuální postavy. Každé těleso je v programu reprezentováno uzlem se stejným číslem. Každý uzel má implicitně přiřazenou určitou část těla. Při zapnutém ladicím módu (je také vhodné zapnout pauzu, aby se postava přestala pohybovat) lze však kterýkoliv uzel libovolně přemapovat.

B.4.2.1 Výchozí mapování uzlů

Namapování jednotlivých těles na části virtuální postavy jsou implicitně zvolena takto:

Název tělesa uvedený v programu RGB:	Číslo pomocného uzlu	Odpovídající část virtuální postavy:
Rigid Body 1	1	hlava
Rigid Body 2	2	pravá dlaň
Rigid Body 3	3	levá dlaň
Rigid Body 4	4	pravé lýtko
Rigid Body 5	5	levé lýtko
Rigid Body 6	6	pravá paže
Rigid Body 7	7	levá paže
Rigid Body 8	8	střední torzo
Rigid Body 9	9	pravé chodidlo
Rigid Body 10	10	levé chodidlo
Rigid Body 11	11	pravý loket
Rigid Body 12	12	levý loket
Rigid Body 13	13	pravé stehno

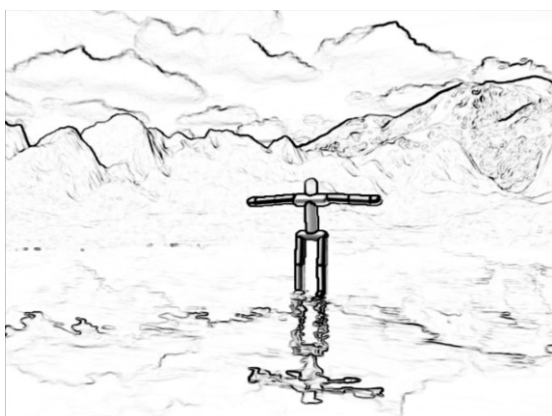
Rigid Body 14	14	levé stehno
Rigid Body 15	15	horní torzo
Rigid Body 16	16	dolní torzo

B.4.2.2 Přemapování uzlu:

- zapneme ladící mód (E), volitelně zapneme pauzu (P);
- vybereme požadovaný uzel pomocí Page Down/Page Up;
- myší klikneme na tu část těla, kterou chceme nově přiřadit danému uzlu (a tedy danému Rigid Body definovanému v aplikaci Rigid Body Tool), dojde k aktualizaci výpisu;
- od této chvíle jsou souřadnice daného tělesa aplikovány na nově zvolenou část těla.

Jelikož nově zvolená část těla měla již dříve určen nějaký uzel, je tento uzel přemapován na část těla, která je aktuálně bez uzlu. Nedojde tak k situaci, kdy by pozice jedné části těla byla současně určována více uzly.

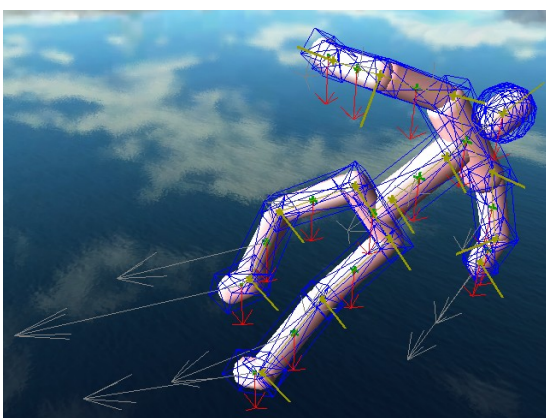
B.4.3 Vyobrazení některých funkcí



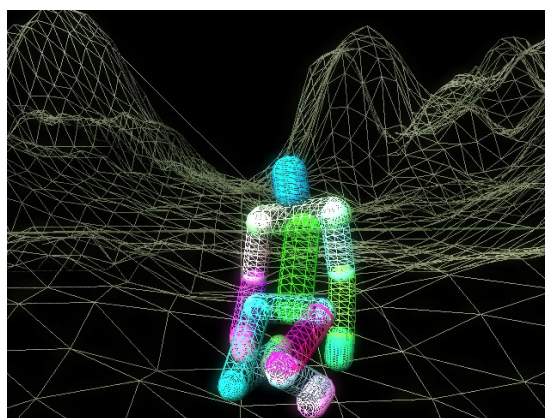
mód „Skica“ (K)



ladící mód (E)



zobrazení fyzikálních interakcí (V)



drátěný model (T)

B.5 Nastavení programu Rigid Body Tool

Program spustíme a načteme vhodnou kalibraci. Načteme dříve uložené definice jednotlivých těles, nebo je vytvoříme znovu (popis kalibrace kamerového systému a vytváření definic jednotlivých těles naleznete v [21]). Názvy jednotlivých těles musejí být ve tvaru Rigid Body 1 ... Rigid Body 16. Počet těles je v tomto rozmezí volitelný.

V záložce Streaming vpravo dole zaškrtneme kolonku [Broadcast Frame Data] (VRPN Port neměníme, ponecháme výchozí hodnotu 3883) a klikneme na tlačítko [Stop].



Program po spuštění implicitně nastaví levotočivý systém (tzv. left-handed system) souřadných os, který aplikace MoCap využívá. Nicméně program VRECKO používá pravotočivý systém, ve kterém je změněna orientace z-ové osy. Pokud bychom chtěli zároveň používat obě aplikace, je nutné systém souřadných os změnit.

Změna orientace souřadných os v aplikaci MoCap:

Osu x nastavíme tlačítkem F7 na reverzní, pokud uživatel otočený směrem k plátnu udělá úrok doleva a virtuální postava udělá úrok doprava (tedy na opačnou stranu, nikoli zrcadlově).

Osu z nastavíme tlačítkem F8 na reverzní, pokud uživatel otočený směrem k plátnu udělá krok dopředu a virtuální postava se posune o krok dozadu.

B.6 Známé problémy

Aplikace MoCap byla intenzivně testována a neměla by obsahovat žádné závažné chyby. V aplikaci byly ponechány i funkce, které mají spíše ladící charakter a přímo nesouvisí se snímáním pohybu či během programu jako takovým. Proto se v některých chvílích mohou objevit určité nesrovnalosti.

Například:

- při zapnutí paralelní projekce se kamera umístí mimo „skybox“ a ten je vidět jako průhledná krychle uprostřed scény;
- při zapnutí módu „Skica“ se ladící mód nezobrazuje správně.

C Obsah přiloženého CD

Přiložené CD obsahuje následující soubory:

- tato práce ve formátu pdf a doc;
- zdrojový kód aplikace MoCap ve formě projektu pro vývojové prostředí NetBeans;
- knihovny potřebné ke kompilaci programu MoCap;
- nezávislá obecná knihovna zajišťující načítání vstupních dat ze zařízení OptiTrack;
- zkompilovaná (multiplatformní) verze výsledného programu ve formátu jar.