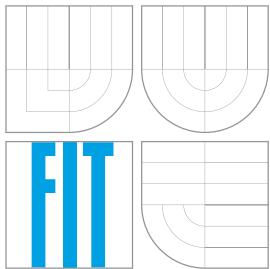


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PROCEDURÁLNÍ ANIMACE LIDSKÉ CHŮZE

PROCEDURAL ANIMATION OF HUMAN WALK

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PETR MOHELNÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. LUKÁŠ POLOK

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání diplomové práce

Řešitel: **Mohelník Petr, Bc.**

Obor: Počítačová grafika a multimédia

Téma: **Procedurální animace lidské chůze**

Procedural Animation of Human Walk

Kategorie: Počítačová grafika

Pokyny:

1. Seznamte se s principy skeletální animace a se stavbou lidské kostry a její funkcí při chůzi. Seznamte se s existujícími implementacemi a popište je.
2. Prostudujte knihovnu OpenGL a její rozšíření.
3. Navrhněte vhodný model pro animaci lidské chůze s nastavitelnými parametry jako pohlaví chodce, výška chodce, kolibavost a rychlosť chůze.
4. Implementujte navržený model a jeho vizualizaci pomocí OpenGL.
5. Zhodnoťte dosažené výsledky a navrhněte možnosti dalšího vývoje, zejména pak spojení s inverzní kinematikou pro chůzi po členitém terénu, překračování překážek, apod.
6. Vytvořte video s prezentací projektu.

Literatura:

- dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1 - 3, experimenty vedoucí ke splnění bodu 4.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese
<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Polok Lukáš, Ing., UPGM FIT VUT**

Datum zadání: 1. listopadu 2015

Datum odevzdání: 25. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
619 66 Brno, Božetěchova 2
L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Animace lidské chůze má velké využití v interaktivních aplikacích, převážně počítačových hrách. Existuje mnoho způsobů tvorby této animace, které se liší v kompromisu mezi přirozeností, kontrolou nad animací a výpočetním časem. Tato práce se zabývá implementací procedurální animace, která je vhodná pro chůzi po nerovném terénu. Pro manipulaci s modelem lidského těla se využívá skeletální animace. Dále je popsána a implementována inverzní kinematika, díky čemu může být animace přizpůsobována nerovnému terénu. A nechybí ani popis fází lidské chůze, abychom je mohli co nejvěrněji napodobit. Vytvořené řešení umožňuje změnu chůze jednoduše pomocí množství parametrů a je schopné se přizpůsobit okolnímu terénu. Výsledek má mít využití při tvorbě počítačových her a umožnit rychlé přidání specifické animace lidské chůze bez nutnosti takovou animaci ručně vytvářet.

Abstract

Animation of human walk is employed in many interactive applications, mostly in computer games. There are many ways to create such animation which differ in compromise among naturalness, control over animation and computing time. This work implements procedural animation which is applicable for walking on uneven terrain. Skeletal animation is used for manipulation with model of human body. Furthermore inverse kinematics is described and implemented. That allows for adaptation to uneven terrain. It also describes phases of human walk, so we can accurately approximate them. Proposed solution enables specification of walk using number of parameters and is able to adapt to surrounding terrain. The result should be usable in creation of computer games and should allow for creation of specific animation of human walk without need to create such animation manually.

Klíčová slova

počítačová grafika, procedurální animace, skeletální animace, lidská chůze, inverzní kinematika

Keywords

computer graphics, procedural animation, skeletal animation, human walk, inverse kinematics

Citace

MOHELNÍK, Petr. *Procedurální animace lidské chůze*. Brno, 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Polok Lukáš.

Procedurální animace lidské chůze

Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením Ing. Lukáše Poloka a uvedl jsem všechny zdroje, ze kterých jsem čerpal..

.....
Petr Mohelník
9. května 2016

Poděkování

Děkuji svému vedoucímu práce Ing. Lukáši Polokovi za odbornou pomoc poskytnutou při řešení této práce.

© Petr Mohelník, 2016.

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

Obsah

1	Úvod	2
2	Skeletální animace	3
2.1	Kostra	3
2.2	Skinning	4
2.3	Tvorba animace	6
3	Kinematiky	10
3.1	Jacobiho inverzní metoda	11
3.2	Cyclic Coordinate Descent	12
4	Lidská chůze	14
4.1	Fáze krokového cyklu	15
5	Existující implementace	18
5.1	Interactive Animation of Personalized Human Locomotion	18
5.2	Animation of Human Walking in Virtual Environments	19
6	Návrh a realizace řešení	21
6.1	Načtení modelu	21
6.2	Zobrazení	22
6.3	Animace	25
6.4	Nastavování parametrů	37
6.5	Měření rychlosti	38
7	Závěr	40
Literatura		41
Přílohy		43
Seznam příloh		44
A	Obsah CD	45
B	Manuál	46

Kapitola 1

Úvod

Počítačová animace má velké využití pro počítačové hry a filmy. Existuje množství způsobů tvorby a zobrazování animace s různými výhodami a nevýhodami. V této práci se zabýváme tvorbou animace lidské chůze - jedné z nejpoužívanějších a nejdůležitějších animací. Tato animace musí být co nejpřirozenější, protože lidé jsou velmi vnímaví na správnost lidské chůze a snadno rozpoznají, kdy je animace nepřirozená.

Přístupy k tvorbě animace lidské chůze se liší v kompromisu mezi přirozeností, kontrolou nad animací a výpočetním časem. Populární způsob tvorby animace je *motion capture*, kde se zachytává pohyb herce pomocí kamér. Ačkoliv tento způsob dosahuje velké přirozenosti, je velmi drahý. *Fyzikálně založená simulace* umožňuje velmi dobrou interakci s prostředím, ale je velmi výpočetně náročná. Tato práce se zabývá *procedurální animací*. Při tomto způsobu se využívá znalostí o *cyklu lidské chůze*, který je studován již od starověku. Procedurální systém umožňuje jednoduše vytvořit animaci pomocí specifikace množství parametrů. Je výpočetně nenáročná a vhodná pro interakci s prostředím, ale ne tak fyzikálně přesná jako fyzikálně založená simulace. Také není tak detailní jako v motion capture. V této práci si klademe za cíl vytvoření systému, který umožní rychlou a jednoduchou tvorbu specifické animace lidské chůze i uživatelům bez animačních znalostí a schopností, kterou budou moci použít např. v počítačové hře.

Druhá kapitola se zaměřuje na *skeletální animaci*. Přibližuje co je to *kostra* a jak se na ní připojí *kůže*. Dále obsahuje stručný přehled metod tvorby animace. Třetí kapitola se věnuje *kinematikám*. Definuje co je to *inverzní kinematika* a popisuje některé metody jejího řešení. Ve čtvrté kapitole je podrobně popsána lidská chůze a její fáze. Pátá kapitola obsahuje popis některých existujících implementací procedurální animace využívající tyto znalosti. Pátá kapitola nastínuje výslednou aplikaci a její řešení a popisuje implementované části v rámci semestrálního projektu.

Kapitola 2

Skeletální animace

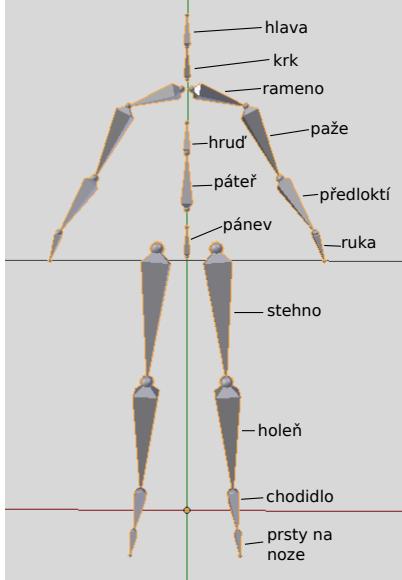
Skeletální animace je způsob animace, který využívá dvě základní komponenty: model (nejčastěji polygonální) nazýván *kůže* a *kostru*, na kterou je kůže připojena. Jestliže se pohně kost, pohně se také připojená kůže. V klasické animaci, je vytvořen model pro každý *klíčový snímek* animace a tyto modely se postupně zobrazují. Oproti tomu se skeletální animace skládá pouze z jednoho modelu a připojené kostry, pro kterou jsou uloženy pozice kostí v klíčových snímcích např. cyklu chůze. Každý tento snímek definuje pozici každé kosti v konkrétní čas. Díky tomu má mnohem nižší paměťovou náročnost. Také skeletální animace umožňuje vysokou interakci s prostředím. Kost se může zarazit o překážku apod. Skeletální animace také mohou být použité na více různých modelech. Někdy programy poskytují předpřipravenou kostru, okolo které se vytvoří kůže. Skeletální animace má také své nevýhody. Pohyby kostí a připojených vrcholů spotrebují výpočetní výkon. V animacích obecně se musí řešit interpolace mezi dvěma klíčovými snímkami, protože zobrazení snímku může proběhnout v libovolný čas.

2.1 Kostra

Kostra je abstraktní model lidského, zvířecího těla nebo jiného objektu. Jedná se o stromovou strukturu (obrázek 2.1). Uzly jsou nazývány *kosti* nebo *klouby*. Ke kostem jsou přiřazeny vrcholy. Kosti mají stupně volnosti, které určují v jakém směru a jak moc se můžou rotovat. Připojování modelu na kostru se nazývá *skinning*.

Každá kost má lokální transformaci: posunutí, rotaci a zvětšení. Tato transformace se může uložit jako matice a určuje transformaci vůči rodičovské kosti, pokud existuje. Pro realistické modely člověka se většinou používá pouze rotace. Výsledná globální transformace kosti se získá kombinací rodičovské globální transformace a vlastní lokální transformace. Globální transformace všech kostí se vypočítá procházením stromové struktury kostry a postupným násobením transformačních matic od kořene k listům. Pro kořenovou kost je lokální transformace její globální.

Před připojením kůže na kostru se kostra nastaví do pozice nazývané *bind pose* [3]. To je základní pozice kostry před jakýmkoliv pohybem. V tuto chvíli se na kostru připojí kůže. Následně jsou určeny a uloženy bind pose matice \mathbf{B} pro každou kost. Ty určují jejich transformaci z počátku souřadného systému kostry do bind pose. Matice \mathbf{B}_a každé kosti určuje její aktuální transformaci z počátku souřadného systému kostry. Před prvním pohybem kostry jsou shodné s \mathbf{B} . Chceme-li změnit pózu kostry (matici \mathbf{B}_a nějaké kosti) a tím i připojené kůže, musí se vrcholy transformovat ze souřadného systému kůže do



Obrázek 2.1: Kostra, která je vytvořena a použita v této práci.

souřadného systému kostry. K tomu se použije *inverzní bind pose matici* \mathbf{B}_i . Ta určuje transformaci kosti potřebnou pro přesun kosti z bind pose do počátku souřadného systému. Tedy invertuje transformaci, která byla na kůži aplikována v bind pose. Matematický zápis transformace \mathbf{T} (obrázek 2.2) určující novou pozici kosti je následující:

$$\mathbf{T} = \mathbf{B}_i \mathbf{B}_a \quad (2.1)$$

Poté se může použít \mathbf{T} na výpočet nové pozice vrcholů kůže. Vrchol bude pořád ve stejné relativní pozici vůči kosti. Transformování se často provádí na grafické kartě ve vertex shaderu.

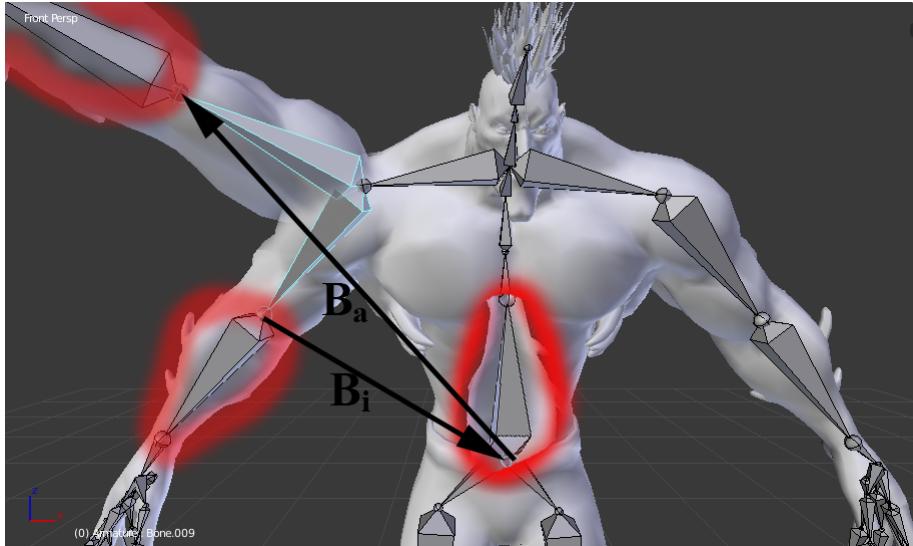
2.2 Skinning

Skinning [14] je nazýván proces přidělování kostí vrcholům kůže a určování jejich vah. Tento proces může probíhat algoritmicky nebo ručně v některém z modelovacích programů. Nejjednodušší způsob přiřazení kůže ke kostře je každé kosti přiřadit samostatný pevný objekt jako je např. válec. Na každý takový objekt se poté aplikuje transformace přiřazené kosti. Tento přístup není vhodný pro realistické animování.

Pokročilejší, ale stále základní metoda je *Jednoduchý skinning*, který již používá kůži, tedy jeden model pro celou kostru. Přiřazuje jeden vrchol k jedné kosti. Každý vrchol se poté transformuje podle přiřazené kosti. Je používán například ve starších hrách. Je dobré použitelný pro modely s nízkým počtem trojúhelníků. U detailnějších modelů způsobuje nepěkné deformace v kloubech a animace vypadá nepřirozeně.

2.2.1 Linear blend skinning

Linear blend skinning [14] je modernější algoritmus. Je nepublikovaný a vyskytuje se i pod jinými názvy jako je *skeleton-subspace deformation*, *enveloping* nebo pouze *skinning*. Každému vrcholu umožňuje přiřazení více kostí. Každá dvojice kost a vrchol má přiřazenou



Obrázek 2.2: Transformace předloktí. Prvně se aplikuje inverzní bind pose matice a poté aktuální transformace kosti. Model byl převzat z [5].

váhu. Váha určuje jak velký vliv má pohyb konkrétní kosti na výslednou pozici vrcholu. Na vstupu očekává následující data:

- Kůže v bind pose, typicky reprezentovaná jako polygonální model. Propojení mezi vrcholy se v průběhu nemění, pouze pozice vrcholů.
- Transformace kostí, reprezentované jako matice $\mathbf{T}_1, \dots, \mathbf{T}_m$ určené podle rovnice 2.1. Tyto matice jsou typicky jediná veličina, která se v průběhu animace může měnit.
- Váhy vrcholů, pro každý vrchol \mathbf{v}_i máme váhy $w_{i,1}, \dots, w_{i,m} \in \mathbb{R}$. Každá váha $W_{i,j}$ určuje vliv kosti j na vrchol i . Celkový součet všech vah je 1, přičemž žádná váha není záporná.

Množství kostí ovlivňující jeden vrchol se může významně lišit, např. od 1 do 10. Kvůli limitacím grafického hardwaru se často uvažuje že jich není více než 4. Transformovaná pozice vrcholu \mathbf{v}'_i se určí podle následující rovnice:

$$\mathbf{v}'_i = \left(\sum_{j=1}^m w_{i,j} \mathbf{T}_j \right) \mathbf{v}_i \quad (2.2)$$

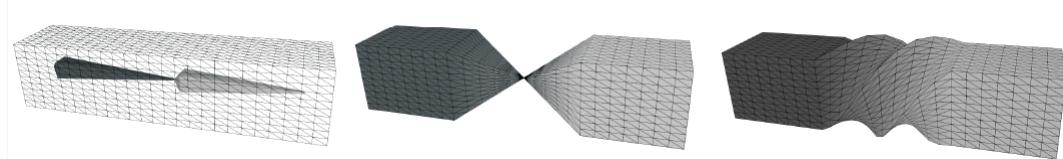
Tato metoda podává dobré výsledky, pokud míchané transformace nejsou příliš rozdílné. Problémy nastávají jestliže potřebujeme míchat transformace, které se významně liší v jejich rotaci. Z lineární kombinace rotací nevzniká rotace. To je důsledkem faktu, že Lieova grupa 3D transformací, $SO(3)$, není lineární prostor, ale zakřivená varieta. Pokud jsou si rotace blízké není to problém. Uvažujme ale tyto dvě rotace.

$$\mathbf{R}_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_2 \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matice \mathbf{R}_1 je maticí identity a matice \mathbf{R}_2 je rotace okolo osy Z o 180 stupňů. Lineární míchání $0.5 \cdot \mathbf{R}_1 + 0.5 \cdot \mathbf{R}_2$ nám dá za výsledek matici hodnosti 1, která promítá 3D prostor na osu Z. To má za následek ztrátu objemu modelu. Velké relativní rotace nejsou vzácné, protože klouby jako ramena a zápěstí mají velký rozsah pohybu. Tyto problémy je možné řešit zavedením více parametrů než má linear blend skinning. Takové metody se nazývají *multilineární*. Tyto metody ale nemusí být žádoucí kvůli nutnosti vytvářet a ukládat více vah pro každý vrchol.

2.2.2 Dual quaternion skinning

Velmi rozšířenou metodou je nelineární metoda *Dual quaternion skinning* [16], kde se místo matic míchají *duální kvaterniony*. Tato metoda zachovává objem modelu, řeší artefakty vznikající u linear blend skinning a je velmi rychlá. Používá se v rozšířených modelovacích programech jako je např. Blender. Klasické kvaterniony umožňují reprezentovat 3D rotaci okolo osy. Tato osa ale musí procházet počátkem souřadného systému. Oproti tomu duální kvaterniony toto omezení nemají - osa může být libovolná. Kromě toho duální kvaterniony umožňují popsat posunutí. Ale neumožňují popsat změnu měřítka. Proto nemůže docházet ke ztrátě objemu, jako u matic, kde násobením může vznikat zmenšení. Duální kvaterniony jsou také výhodné pro GPU hardware, protože obsahují méně hodnot než maticy. Pro porovnání metod viz obrázek 2.3.



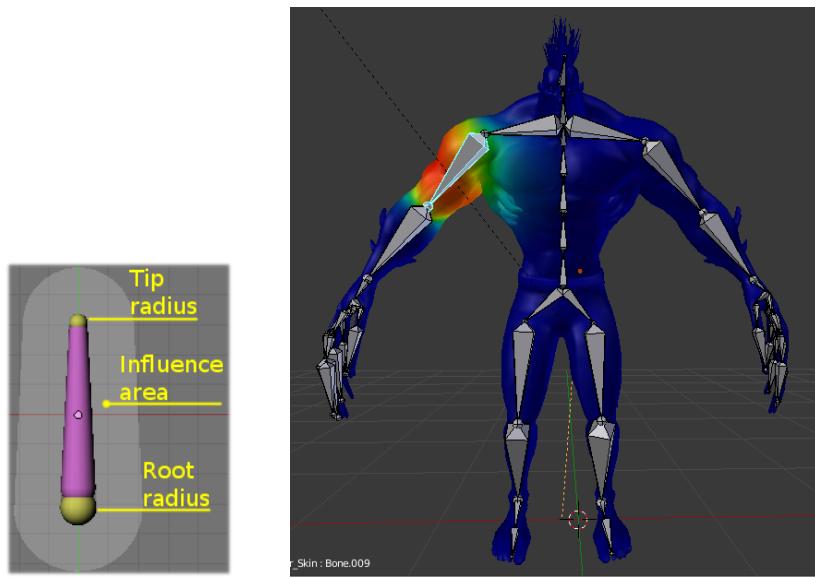
Obrázek 2.3: Zleva doprava: model v bind pose, linear blend skinning a dual quaternion skinning. Převzato z [14].

2.2.3 Metoda obálek

Přiřazení vah je možné automaticky *metodou obálek* [2]. Tato metoda je založena na blízkosti kostí a jejich geometrie. Každá kost má dvě oblasti vlivu. Vnitřní oblast, kde je geometrie plně ovlivněna touto kostí. A vnější oblast, kde je geometrie méně ovlivněna kostí, čím blíže je k okraji této oblasti. Pokud chceme mít váhy určené kvalitně, je nutné nastavit váhy ručně. To je možné v některém modelovacím programu, například Blender poskytuje *Weight Paint* mód, kde se pohybem štětce po modelu určují váhy vrcholů pro jednotlivé kosti. Tento proces je iterativní a náročný. Kreslením se odebírají nebo přidávají váhy vrcholům pro jednu aktivní kost. Poté se zkouší efekt na známé póze, upravují a vyhlažují váhy a tento proces se opakuje, dokud výsledek není dostatečně přirozený. Na obrázku 2.4 je ukázána obálka a model ve Weight Paint módu.

2.3 Tvorba animace

Animace člověka se vyskytuje ve velkém množství aplikací v různých prostředích a podmínkách. Tyto animace by proto měly být přizpůsobitelné prostředí v reálném čase. Také



Obrázek 2.4: Metoda obálek v programu Blender (převzato z [2]) a model [5] ve Weight paint módu v Blenderu.

by měly působit přirozeně a tím přidávat na reálnosti. Základní nástroje pro specifikaci pohybu jsou založeny na *kinematikách*. Existují dva způsoby popisu kinematické informace: *inverzní* a *přímá kinematika*, viz kapitola 3.

Welbergen a spol. [23] dělí techniky tvorby animace do tří skupin: *procedurální animace*, *fyzikálně založená simulace* a *editace pohybu*. Techniky se také mohou kombinovat dohromady a využívat vhodnější techniku pro danou situaci nebo část těla. Může se například použít technika editace pohybu dokud nedojde k interakci s prostředím a ta se může zpracovat pomocí fyzikálně založené simulace. Nebo pomocí procedurální animace vytvořit gesta rukou a fyzikální simulací zachovat fyzikálně přirozené balancování spodní části těla.

2.3.1 Procedurální animace

Procedurální animace popisuje animaci algoritmicky pomocí matematických vzorců. Její úspěch závisí na tom, jak hluboce jsme schopni pochopit a modelovat simulovaný pohyb. Metody mohou popisovat rotace kostí přímo nebo popisovat cesty koncových uzlů kostry (chodidla), potom se používá inverzní kinematika.

Procedurální animace umožňuje přesné časování a pozicování končetin a může využívat velké množství parametrů, pomocí kterých je možné upravit výstup na míru požadavkům. Dobře se adaptují na prostředí a jsou málo výpočetně náročné. Nicméně je často problematické zakomponovat detaily jako jsou u editace pohybu do matematického popisu. Také je nutné explicitně zachovávat fyzikální přirozenost v procedurálním modelu pohybu pro všechny možnosti parametrů. Je často používána pro animaci gest a mluvění, které vyžadují velké množství parametrů.

2.3.2 Fyzikálně založená simulace

V mnoha případech je vhodné počítat s *dynamikami* [20], aby byla zachována realističnost pohybu. Tyto případy mohou být:

- jestliže postava nese náklad
- jestliže postava musí reagovat na externí síly jako poryvy větru
- povrch na kterém se pohybuje je komplexní (schody atp.)

Dynamiky se používají u technik ze skupiny fyzikálně založené simulace. Dynamiky jsou založené na Newtonových pohybových zákonech. Spojují síly (resp. točivé momenty rotací) do výsledného pohybu (resp. rotace) podle rovnice:

$$f = m \cdot \ddot{x} \quad (2.3)$$

kde f je síla aplikovaná na objekt, m je jeho hmotnost, \ddot{x} je druhá derivace x podle času. Pro rotace vypadají rovnice podobně. Točivé momenty a úhlové pozice jsou spojeny pomocí:

$$t = i \cdot \ddot{\theta} + \dot{\theta} \times i \cdot \ddot{\theta} \quad (2.4)$$

kde t je točivý moment, i je matice setrvačnosti, $\dot{\theta}$ je úhlová rychlosť a $\ddot{\theta}$ je úhlové zrychlení. *Přímá dynamika* je aplikace těchto zákonů k výpočtu pohybu vygenerovaného danou silou. *Inverzní dynamika* se zabývá určováním síly, která by vygenerovala daný pohyb.

Ve fyzikálně založené simulaci je model typicky reprezentovaný jako systém pevných těles propojených klouby. Každé z těchto pevných těles má přiřazené fyzikální vlastnosti jako např. hmotnost. Pohyb je generován manipulací s točivými momenty kloubů s použitím přímé dynamiky. Pro umožnění kolizí je nutné geometricky reprezentovat pevná tělesa. Použití samotného povrchového modelu je příliš výpočetně náročné, proto se používají pro kolize approximace pomocí základních tvarů, např. válec.

Metody mohou fungovat na základě zadávání geometrických omezení jako parametrů animace. Tato omezení mohou být typicky splněna velkým množstvím různých točivých momentů. Proto se mohou zavést objektivní funkce pro preferenci jistých řešení. Tyto metody jsou pro animace v reálném čase velmi pomalé. Jiný způsob je používání kontrolérů, do kterých vstupuje žádaný stav systému. Výstupem je množina točivých momentů kloubů, které po aplikaci na systém vedou proměnné k jejich požadovaným hodnotám. Kontrolér využívá fyzikálních vlastností pohybujícího se těla. Úlohou kontroléra je minimalizace nesrovnaností mezi aktuálním a požadovaným stavem. Síly a točivé momenty určené kontrolérem, gravitace a síly způsobené externími vlivy jsou aplikované na fyzické tělo. To je poté pohnuto pomocí přímé dynamiky.

Fyzikálně založená simulace poskytuje fyzikálně realistický pohyb a interakci s prostředím. Umožňuje zachování parametrů pod vlivem externích sil. Nicméně přesné časování a umisťování končetin je problematické. Ačkoliv fyzikální simulace poskytuje fyzikálně správný pohyb, není často dostačující pro vytvoření přirozeného pohybu.

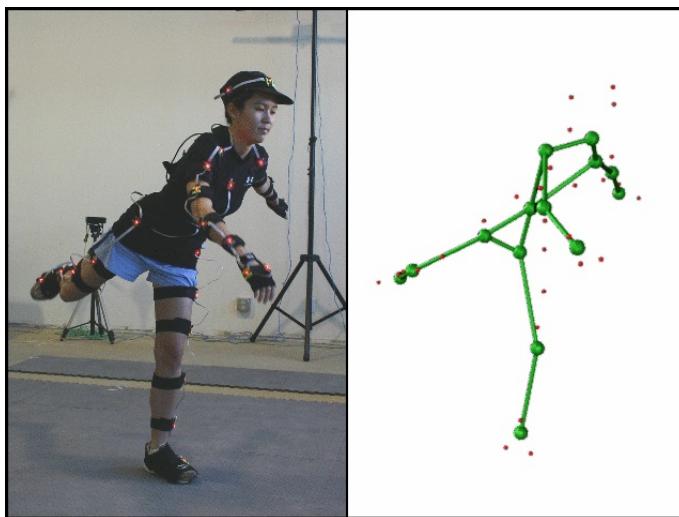
2.3.3 Editace pohybu

Metody editace pohybu vytváří pohyb na základně vzorů pohybu. Metody modifikace pohybu generují nová pohybová primitiva modifikací jednoho vzoru pohybu. Kombinační techniky vytváří pohybová primitiva pomocí databáze většího množství vzorových primitiv.

Pohyby se mohou vytvářet pomocí technik ze zpracování signálů, interpolací mezi vzory pohybů nebo pomocí statistických modelů.

Techniky editace pohybu zachovávají přirozenost vstupních vzorů pohybu, ale pouze jsou-li změny malé. Přirozenost i při větších změnách je zachována technikami používající více vzorů na jeden pohyb. Nicméně pro všechny techniky roste počet vzorů exponenciálně s počtem parametrů animace. Navíc techniky editace pohybu neposkytují fyzikální interakci s prostředím, ale jsou vázány na specifický kontext. Jsou vhodné pro vytváření animací v předstihu pro neinteraktivní aplikace, např. filmy. Pro interaktivní (např. videohry) je potřeba velká databáze pohybů. Pohyby se mohou vytvářet pomocí technik ze zpracování signálů, interpolací mezi vzory pohybů nebo pomocí statistických modelů.

Vstupní data mohou být být tvořená animátorem ručně, ale mohou být získána i např. pomocí *motion capture*. Motion capture [17] jsou techniky pro nahrávání pohybů člověka, zvířete nebo jiného subjektu. Mají velké využití ve filmech a videohrách. Nejčastěji se používá optických metod pro zachycení pohybu. V Motion capture má herc s okolo každého kloubu značky, které jsou zaznamenávány a sledovány kamerami, viz obrázek 2.5. Výstupem je množina lokací bodů v čase. Těmi se poté proloží kostra.



Obrázek 2.5: Automatická rekonstrukce kostry podle pozic značek. Převzato z [17].

Kapitola 3

Kinematiky

Kinematiky v počítačové grafice slouží pro manipulaci s kostrou. Jsou rozděleny na dvě části: přímá a inverzní kinematika. V přímé kinematici se hýbe s každou kostí v řetězci kostí a podle toho se určí výsledná pozice posledního článku - *koncového efektoru*. Výsledky při ručním vytváření animace jsou velmi závislé na schopnostech animátora, protože musí nastavovat pozici kostí od ruky. V této technice je obtížné omezovat pohyb, například že chodidlo nemůže proniknout do země.

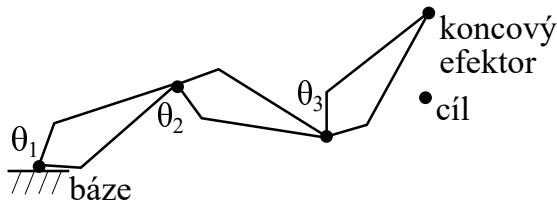
Máme-li řetězec *kloubů*, kde každý kloub i má přiřazen úhel θ_i , který určuje jeho rotaci oproti jeho předkovi, viz obrázek 3.1. Potom stav řetězce je určen stavovým vektorem $\boldsymbol{\theta}(\theta_1, \dots, \theta_n)$. Označme pozici koncového efektoru \mathbf{P} . Přímá kinematika řeší následující rovnici:

$$\mathbf{P} = f(\boldsymbol{\theta}) \quad (3.1)$$

Každý kloub může mít více než jeden *stupeň volnosti*, tedy být definován více úhly. Tím je určena složitost struktury kloubů, která je definována jako součet stupňů volnosti všech prvků.

Oproti tomu v inverzní kinematici [9, 24] se hýbe pouze s koncovým efektorem a pohyb ostatních kloubů se dopočítá, tak aby byl tento pohyb umožněn. To je matematicky zapsáno následovně:

$$\boldsymbol{\theta} = f^{-1}(\mathbf{P}) \quad (3.2)$$



Obrázek 3.1: Příklad řetězce tří kloubů.

Řešení inverzní kinematiky není tak jednoduché jako přímé. Funkce f je nelineární a ačkoliv v rovnici 3.1 existuje unikátní namapování z $\boldsymbol{\theta}$ na \mathbf{P} , pro inverzní mapování v rovnici 3.2 může být nekonečné množství $\boldsymbol{\theta}$ pro jedno konkrétní \mathbf{P} . Analytické řešení pro libovolný řetězec kloubů neexistuje. Problém se řeší pomocí numerických metod pro řešení

systémů nelineárních rovnic. Z nekonečného množství řešení je snaha vybírat to nejvhodnější řešení. Různé metody preferují různé řešení a proto se metody vybírají na základě konkrétně řešeného problému. V případě lidské kostry má každý kloub určené stupně volnosti, aby rotovali jenom v určitých směrech a také je definován maximální úhel rotace. Například koleno rotuje pouze v jednom směru a nemůže být otevřeno více než 180 stupňů a zavřeno méně než 30 stupňů. Inverzní kinematika poskytuje lepší kontrolu nad koncovým efektem, jehož cílová pozice nás většinou nejvíce zajímá. V modelovacím programu můžeme hýbat např. pouze chodidlem a nemusíme také pohybovat stehenní a lýtkovou kostí. Toho se také využije v procedurální animaci lidské chůze, kde budeme určovat trajektorie koncových efektorů. Techniky založené na kinematikách využívají empirickou a biomechanickou znalost pohybu pro generování animace.

Nyní si ukážeme některé způsoby řešení inverzní kinematiky.

3.1 Jacobiho inverzní metoda

Jacobiho inverzní metoda [9, 24, 7] je iterativní metoda využívající *Jacobiho matice* k aproximaci řešení. Vztah mezi kartézským prostorem koncového efektoru \mathbf{P} a prostoru kloubů úhlů $\boldsymbol{\theta}$ je:

$$\dot{\mathbf{P}} = J(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \quad (3.3)$$

kde tečka značí první derivaci podle času. Jacobiho matice J je $m \times n$ matice:

$$J(\boldsymbol{\theta}) = \left(\frac{\partial P_i}{\partial \theta_j} \right)_{i,j} \quad (3.4)$$

kde m je počet dimenzí pozice koncového efektoru \mathbf{P} , n je počet dimenzí $\boldsymbol{\theta}$, $i = 1, \dots, m$ a $j = 1, \dots, n$. J je matice částečných derivací celého řetězce relativně ke koncovému efektoru. J mapuje změny proměnných kloubů $\boldsymbol{\theta}$ na změny v pozici koncového efektoru. Sloupec i v J reprezentuje inkrementální změnu pozice koncového efektoru způsobenou inkrementální změnou proměnné θ_i . Neznámou pro inverzní kinematiku je $\dot{\boldsymbol{\theta}}$, proto potřebujeme inverzi Jacobiho matice. Vztah je následující:

$$\dot{\boldsymbol{\theta}} = J^{-1}(\boldsymbol{\theta})\dot{\mathbf{P}} \quad (3.5)$$

Poté co určíme Jacobiho matici, hledáme hodnotu $\Delta\boldsymbol{\theta}$ pro inkrementaci $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = \boldsymbol{\theta} + \Delta\boldsymbol{\theta} \quad (3.6)$$

Změna v pozici koncového efektoru určená touto změnou je:

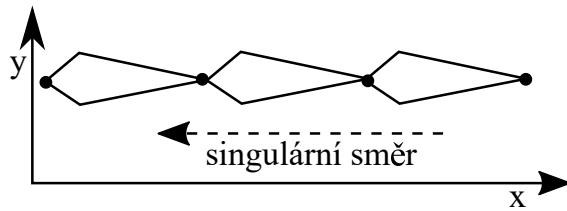
$$\Delta\vec{s} \approx J\Delta\boldsymbol{\theta} \quad (3.7)$$

Hodnota $\Delta\boldsymbol{\theta}$ by měla být vybrána tak, aby $\Delta\vec{s}$ přibližně odpovídala \vec{e} - rozdílu mezí cílovou a aktuální pozicí koncového efektoru. Potom přímá kinematika může být přepsána jako $\vec{e} = J\Delta\boldsymbol{\theta}$ a inverzní jako $\Delta\boldsymbol{\theta} = J^{-1}\vec{e}$.

Jacobiho matice ale není vždy invertibilní, tzn. není čtvercová a zároveň singulární. To nastane když konfigurace kloubů je redundantní nebo když prochází skrz nebo blízko singulární konfigurace.

Řetězec kloubů je redundantní jestliže obsahuje více stupňů volnosti, než je potřeba pro specifikaci cíle koncového efektoru. Je-li cíl koncového efektoru určen pozicí v 3D prostoru,

potom jakýkoliv řetězec obsahující více než 3 stupně volnosti je redundantní a tedy existuje nekonečně mnoho řešení. Jacobiho matice potom má více sloupců než řádků. V takových případech se J^{-1} nahradí generalizovanou inverzí J^\dagger , většinou *Moore-Penroseovou pseudoinverzí*. Jestliže existuje nekonečné množství řešení, je možné toho využít pro splnění nějakého sekundárního kriteria, např. vyhnutí se překážkám. Matice je singulární jestliže dva nebo více řádků jsou lineárně závislé, příklad takové konfigurace je vidět na obrázku 3.2. Jacobiho matice pro takovou konfiguraci bude obsahovat nuly v prvním řádku a nemůže být invertována. V takovém případě je možné použít pseudoinverzní matici, která poskytne řešení v singulární konfiguraci, nicméně v okolí singulární konfigurace bude nevhodně oscilovat.



Obrázek 3.2: Singulární konfigurace řetězce kloubů.

3.2 Cyclic Coordinate Descent

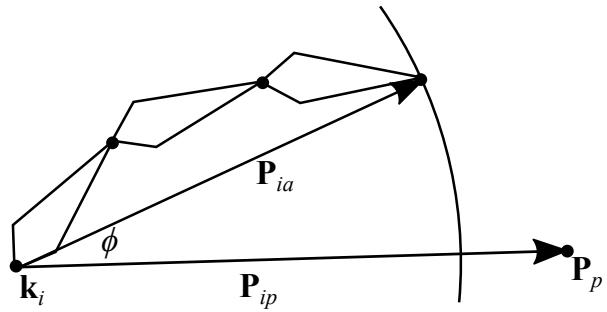
Cyclic Coordinate Descent (CCD) [24] je řešení inverzní kinematiky založené na iterativním heuristickém vyhledávání, které se snaží o minimalizaci rotační chyby zpracováváním jednoho kloubu v jeden okamžik. Každá iterace provede jednotlivě pohyb všech článků od nejposlednějšího k prvnímu. To se opakuje dokud se nedosáhne požadované pozice koncového efektoru a nebo dokud se nepřesáhne maximální počet cyklů. K určení úhlu rotace se používají dva vektory. Protože metoda pracuje s jedním kloubem v jeden okamžik, zvýhodňuje klouby na začátku. Oproti metodám používajícím Jacobiho matice provádí více iterací. Ačkoliv je algoritmus jednoduchý, je často nutné ho dále přizpůsobovat, aby bylo dosaženo vhodných výsledků.

Mějme iteraci i (viz obrázek 3.3) a pozici kloubu \mathbf{k}_i pro i -tý kloub od konce řetězce. Potom vektor \mathbf{P}_{ia} je vektor od pozice kloubu \mathbf{k}_i k aktuální pozici koncového efektoru a \mathbf{P}_{ip} je vektor od \mathbf{k}_i do požadované pozice koncového efektoru. Budeme-li rotovat vektor \mathbf{P}_{ia} o úhel ϕ , dostaneme nový vektor $\mathbf{P}'_{ia}(\phi)$. Jestliže úhel bude nabývat libovolných hodnot, $\mathbf{P}'_{ia}(\phi)$ začne tvořit kruh se středem \mathbf{k}_i . Bod na tomto kruhu nejbližší k požadované pozici \mathbf{P}_p , je bod, na kterém kruh protíná přímku určenou vektorem \mathbf{P}_{ip} . Proto budeme měnit rotaci kloubu \mathbf{k}_i tak, aby se \mathbf{P}_{ip} a $\mathbf{P}'_{ia}(\phi)$ zarovnaly. Úhel ϕ který hledáme, je úhel, který maximalizuje následující funkci:

$$g(\phi) = k_1(1 - \cos \phi) + k_2 \cos \phi + k_3 \sin \phi \quad (3.8)$$

kde k_1 , k_2 a k_3 jsou konstantní koeficienty, pro detaily viz [24]. Funkce má maximum v intervalu $-\pi \leq \phi \leq \pi$, když je první derivace nula a druhá derivace je záporná. Z první podmínky dostaváme rovnici:

$$\phi = \tan^{-1} \frac{k_3}{k_2 - k_1} \quad (3.9)$$



Obrázek 3.3: Ukázka jedné iterace metody CCD.

která vymezí kandidátní hodnotu ϕ_c v rozsahu $\frac{-\pi}{2} \leq \phi_c \leq \frac{\pi}{2}$. Protože tan je periodické, musíme uvažovat i dvě další kandidátní hodnoty: $\phi_c + \pi$ a $\phi_c - \pi$. Z kandidátních hodnot se vybere taková, která spadá do intervalu $-\pi \leq x \leq \pi$ a jejíž druhá derivace je záporná. Je-li jich více, funkce se vyhodnotí nad všemi a vybere se z nich maximum. Poté co jsme získali úhel ϕ , můžeme ho vynásobit váhou určující tuhost kloubu a přičíst ho k aktuální rotaci r_i kloubu i . Poté se pokračuje s iterací $i + 1$.

Kapitola 4

Lidská chůze

Lidská chůze [6] se skládá z opakujících se *krokových cyklů* neboli *dvojkroků*, viz obrázek 4.1. Cyklus je sekvence pohybů od doby kdy se jedna noha dotkne země do doby kdy se stejná noha dotkne země znovu. Každý krokový cyklus má dvě základní fáze: *stojnou fázi* a *švihovou fázi*. Protože je cyklus shodný pro obě nohy, budeme popisovat pouze jednu.

Stojná fáze, je fáze, kdy je noha v kontaktu se zemí. Zabírá 60% dvojkroku a začíná s *úderem paty* (*počáteční kontakt*) a končí s *odrazem palce*. Je dále rozdělena do podfází:

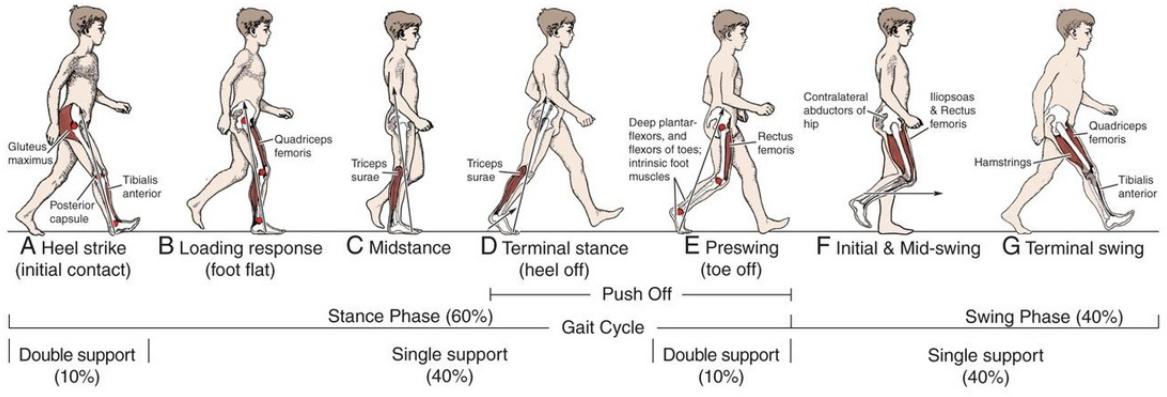
- *stádium zatěžování*
- *mezistoj*
- *konečný stoj*
- *předšvihová fáze*

Po počátečním kontaktu pravé nohy nastává perioda *dvojí opory*, která končí při odrazu palce levé nohy. Následuje *jedno-oporová fáze* do té doby, než se levá dostane do počátečního kontaktu. Nastává druhá perioda dvojí opory, až do odrazu palce pravé nohy. V každé fázi dvojí opory je jedna noha vpředu, právě se dotkla země. A druhá je vzadu připravena na zvednutí se ze země. Přední noha je ve stadiu zatěžování a zadní je v konečném švihu. V každém krokovém cyklu jsou dvě periody dvojí opory a dvě periody jedno-oporové fáze. Dvojí opora zabírá přibližně 10% dvojkroku, ale její délka se mění s rychlosťí chůze. S větší rychlostí se prodlužuje švihová fáze a zkracuje fáze dvojí opory. Když dvojí opora úplně vymizí z chůze se stává běh. Mezi kroky běhu je fáze letu, kdy není žádná noha na zemi.

Švihová fáze nastává když je chodidlo ve vzduchu a trvá 40% délky dvojkroku. Začíná s *úderem paty* a končí s druhým odrazem palce. Je dále rozdělena do podfází:

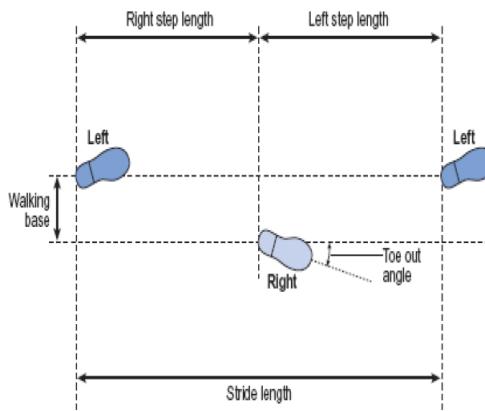
- *počáteční švih*
- *mezišvih*
- *konečný švih*

Nyní si popíšeme některé termíny a parametry specifikující chůzi (obrázek 4.2). *Délka dvojkroku* je vzdálenost mezi dvěma po sobě jdoucími kroky stejné nohy. Skládá se ze dvou délek kroku, levé a pravé, každá s nich určuje vzdálenost o kterou se jedna noha přesune před druhou. Může být např. nulová, když se levá noha přesune na úroveň pravé a ne přední. *Šířka dvojkroku* je vzdálenost mezi přímkami určenými středy pat obou nohou při chůzi.



Obrázek 4.1: Krokový cyklus a jeho fáze. Převzato z [10].

Úhel vytočení chodidla určuje úhel mezi zmíněnou přímkou a přímkou procházející středem chodidla. *Kadence* je počet kroků v daném čase. V jednom krokovém cyklu jsou dva kroky a tedy kadence je měřítko půl-cyklů. Čas dvojkroku c v sekundách se určí jako $c = 120/kad$, kde kad je kadence v krocích za minutu. *Rychlosť chůze* je vzdáenosť ušlá v čase. Okamžitá rychlosť je variabilní v průběhu kroku, ale průměrná rychlosť je výsledkem kadence a délky dvojkroku. Rychlosť chůze r se spočítá jako $r = d/c$, kde d je délka dvojkroku. Rychlosť chůze tedy závisí na délce kroku, které závisí na délce švihové fáze.



Obrázek 4.2: Délka, šířka kroku a úhel vytočení chodidla. Převzato z [6].

4.1 Fáze krokového cyklu

Tato sekce obsahuje podrobný popis jednotlivých fází krokového cyklu. Každá z osmi fází má funkční objektiv. Sekvenční kombinace fází umožňuje končetině dosáhnout tří základních úkolů. Těmi jsou přesun hmotnosti těla, jednooporová fáze a posun končetiny ve švihové fázi. Pro tuto podkapitolu bylo čerpáno z [6].

4.1.1 Počáteční kontakt

V této fázi je kyčel ohnutá, koleno natažené a kotník v neutrální poloze. Kontakt se zemí je dosažen pomocí paty, která se stává středem otáčení. Druhá noha je na konci konečného stoje. Tato fáze obsahuje moment kdy právě nastane dotyk nohy se zemí. Držení těla v tuto chvíli určuje pohyb končetiny ve stádiu zatěžování.

4.1.2 Stádium zatěžování

Stádium zatěžování je fáze, kde je váha těla přenesena na přední nohu. Koleno je pokrčeno pro absorbování šoku. Chodidlo se dostává do plného kontaktu se zemí. Protilehlá noha je v předšvihové fázi. Tato fáze je počátkem fáze dvojí opory. Pokračuje dokud nezačne být druhá noha zvedána ke švihu.

4.1.3 Mezistoj

V mezistoji dochází k posunutí dolní končetiny přes zafixované chodidlo, zatímco koleno a kotník se propínají. Protilehlá noha se nachází uprostřed mezišvihové fáze. Je to první polovina intervalu stoje na pouze jedné noze. Trvá od zvednutí druhé nohy dokud váha není přesunuta po chodidle do oblasti přednoží.

4.1.4 Konečný stoj

Během druhé poloviny intervalu stoje na jedné noze se zvedne pata a střed otáčení stojné končetiny se přesune do přední části nohy. Koleno zvýší své propnutí a poté se začne lehce ohýbat, kyčel se propíná. Druhá noha je v konečném švihu. Začíná se zdvihem paty a končí v okamžiku kontaktu paty druhé nohy se zemí. Váha těla je přesunuta před nohu.

4.1.5 Předšvihová fáze

Kontakt se zemí druhé nohy začal další fázi dvojí opory. Referenční noha zvýší ohyb v kotníku a v koleni, kyčel již není propnutá. Protilehlá noha je ve stadiu zatěžování. Tato poslední podfáze stojné fáze je druhým (a posledním) intervalom dvojí opory v krokovém cyklu. Začíná s počátečním kontaktem druhé nohy a končí s odrazem palce referenční nohy. Váha je uvolněna z referenční nohy a přenesena na nohu druhou. Uvolněná noha využije volnosti k přípravě na švihovou fázi, k čemuž slouží všechny pohyby a svalové akce odehrávající se v tuto dobu.

4.1.6 Počáteční švih

Chodidlo je zvednuto a přesunuto vpřed pomocí ohybu v kyčli a koleni. Kotník částečně dorziflexuje (ohyb kotníku směrem k hřbetu nohy). Druhá končetina je na začátku mezišvihu. Tato fáze je přibližně třetina švihové periody. Začíná když noha opustí podložku a končí v okamžiku maximálního propnutí v koleni.

4.1.7 Mezišvih

Přesun nohy dopředu je získán dalším ohybem v kyčli. Koleno se propíná a kotník dorziflexuje do neutrální polohy. Druhá noha je na konci mezistojce. Tato druhá fáze švihové periody začíná když je švihová noha naproti stojné noze. Končí když je holení kost ve vertikálním postavení - ohyb kolene a kyčle je shodný.

4.1.8 Konečný švih

Přesun chodidla je ukončen propnutím kolene a přesunem holeně před stehno. Kyčel si zachovává ohyb z předchozí fáze. Kotník zůstává v neutrální poloze. Druhá noha je v konečném stoji. Tato poslední fáze švihu začíná s vertikální holení kostí a končí při kontaktu nohy se zemí. Touto fází je přesun nohy ukončen.

Kapitola 5

Existující implementace

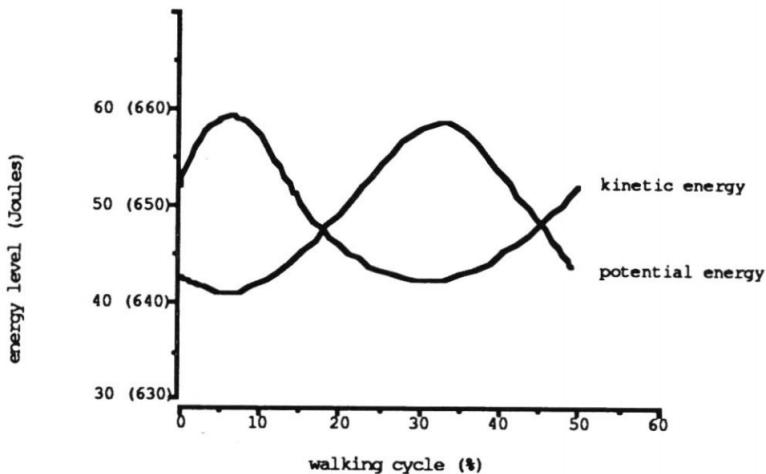
V této kapitole jsou popsány některé existující implementace procedurální animace lidské chůze. Je ukázáno jak se využívá znalostí o lidské chůzi a jak je možné animaci ovlivňovat pomocí parametrů.

5.1 Interactive Animation of Personalized Human Locomotion

V článku Interactive Animation of Personalized Human Locomotion [8] je popsán systém pro tvorbu lidského pohybu pomocí procedurální animace. Umožňuje specifikaci tří parametrů: délka kroku, kadence kroků a rychlosť. A dalších 15 atributů jako je např. rotace a náklon pánve. Díky těmto atributům je umožněna individualizace pohybu pro starého člověka, malé dítě apod. Tento systém je plně kinematický.

Pro každý krok jsou vypočítány tři omezení ze specifikovaných parametrů a atributů: doba trvání fází kroku (stojná a švihová), úhly nohy na konci kroku a kontrolní body určující pohyb kyčle stojné nohy během kroku. Určují se čtyři kontrolní body zvlášť pro vertikální a horizontální komponent. První a poslední bod se určí z délky kroku a úhlů nohy na začátku a konci kroku. Druhý a třetí kontrolní bod se určí na základě znalosti, že vertikální přesun je nejnižší uprostřed dvojí opory a nejvyšší uprostřed švihové fáze. Zatímco horizontální přesun má maximum a minimum opačně. To je v korelací s výdajem energie běhen jednoho dvojkroku, jak je vidět na obrázku 5.1, kde potenciální energie je identická s vertikálním přesunem a kinetická s horizontálním. Mezi body z každé množiny jsou proloženy interpolační spline křivky a tím je určen pohyb kyčle pro stojnou nohu během kroku. Pozice prstů nohy je statická během stojné fáze. Úhly zbylých kloubů nohy jsou dopočítány pomocí inverzní kinematiky.

Tím je určen pohyb stojné nohy. Dále se určí rotace, úklon a náklon pánve. Rotace pánve je na maximu při úderu paty na zem a na minimu v mezistoji. Úklon pánve má minimum při úderu paty a maximum na konci dvojí opory. Náklon pánve je způsobem tím, že se tělo vždy nakloní na stranu nohy nesoucí váhu. Je minimální při úderu paty a maximální uprostřed stojné fáze. Pozice kyčle nohy ve švihu je určena interpolací těchto hodnot. Pohyb nohy ve švihu je rozdělen do tří fází a určen lineární interpolací. Pohyb horní poloviny těla je určen v závislosti na spodní, např. ruka se hýbe dopředu s protilehlou nohou. Pohyb ruky a rotace ramene je určen nastavitelnými atributy.



Obrázek 5.1: Aproximace kinetických a potenciálních energií horní poloviny těla pro průměrný krok. Převzato z [8].

5.2 Animation of Human Walking in Virtual Environments

Kontrolní systém pro animaci lidské chůze po nerovném terénu je prezentován v článku Animation of Human Walking in Virtual Environments [12]. Používá se model člověka s 18 klouby a celkem 36 stupni volnosti.

Kontrola pohybu na nejvyšší úrovni umožňuje uživateli animovat pohyby s malým množstvím parametrů. Uživatel specifikuje cestu pohybu pomocí kubických splinů v horizontální rovině. Z parametrů výšky člověka a rychlosti chůze je vygenerována délka a frekvence kroku. Pro normální chůzi může být vztah určen jako v rovnici:

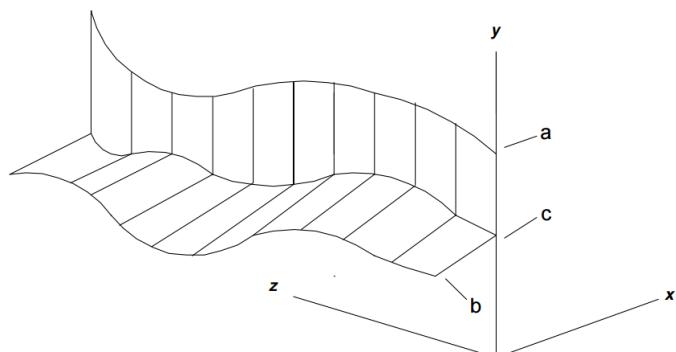
$$l = \sqrt{0.004 \times s \times h} \quad (5.1)$$

kde l je délka kroku, s je rychlosť trupu a h je výška člověka. Výpočet pozice chodidla na terénu je určen jako funkce délky kroku, změny směru globální cesty a stavu terénu. Prvně jsou pozice chodidel rovnoměrně rozloženy po křivce cesty v horizontální rovině. Jestliže přesáhne rotace chodidla prahovou hodnotu, je krok zkrácen, ale doba provedení kroku zůstane zachována. Stav terénů ve 3D je zkонтrolován pro určení bezkolizních trajektorií nohou.

Na střední úrovni kontroly jsou poskytnuty atributy pro kontrolu spodní části těla. Prvně se odhadnou úhly dopadu a zvednutí chodidla stojící nohy. Z těchto odhadnutých hodnot a podle přibližné trajektorie pánevně jsou pomocí inverzí kinematiky dopočítány úhly kolene a kyče. Poté jsou přesně dopočítány úhly dopadu a zvednutí chodidla. Trajektorie chodidla se určí lineární interpolací mezi položeným chodidlem na zemi a úhly dopadu a zvednutí chodidla. Tím je určena trajektorie stojící nohy, jednoho z koncových efektorů pro řetězec inverzní kinematiky. Nyní je potřeba určit pohyb kořene řetězce, pánevně.

Observace ukázaly, že trajektorie pánevně přibližně odpovídá sinusoidě, viz obrázek 5.2. Pro interpolaci jsou použity kubické spliny, konkrétně Beziérova křivka. Pánev prochází vertikálním maximem uprostřed stojné fáze, tato hodnota se určí podle velikosti ohýbu kolene. Pánev prochází minimem uprostřed intervalu dvojí opory, kdy obě nohy jsou v kontaktu se zemí. Pozice v této fázi je určena tak, aby byla minimalizována suma úhlových zrychlení

všech kloubů podpírající nohy. Uživatel může těmto kloubům nastavovat váhy. Výsledná Beziérova křivka je poté nerovnoměrně vzorkována, aby bylo dosaženo nejrychlejšího pohybu během dvojí opory a nejpomalejšího uprostřed stojné fáze. Pro definici trajektorie



Obrázek 5.2: 3D trajektorie (c) je určena z vertikálního (a) a horizontálního (b) posuvu. Převzato z [12].

chodidla nohy ve švihové fázi se použijí Beziérovky obdobně jako při pohybu pánev. Spolu se znalostí pozice kořene (pánev) kinematického řetězce se dopočítají pozice ostatních kloubů. Trajektorie se určí tak, aby se chodidlo vyhnulo kolizím a aby byla spotřebována co nejmenší energie. Na nejnižší úrovni je poté možné nastavit další atributy, např. rotace a náklon pánev.

Kapitola 6

Návrh a realizace řešení

Řešení je implementováno v jazyce C++. 3D grafika je vykreslena pomocí rozhraní OpenGL. OpenGL je multiplatformní rozhraní, podporováno množstvím programovacích jazyků, které umožňuje vykreslování 2D a 3D grafiky. Komunikuje s grafickým procesorem pro hardwarově akcelerované vykreslování. Pro zpracování vstupů a vytvoření kontextu OpenGL je využita knihovna SDL 2.0. SDL je multiplatformní knihovna umožňující nízkoúrovňový přístup ke klávesnici, myši, zvuku a grafickému hardwaru. Funguje nativně v C++.

Aplikace bude umožňovat načtení předpřipraveného modelu s připojenou kostrou. Případně uživatel může využít poskytnutou předpřipravenou kostru a na ní připojit vlastní model a ten poté vyexportovat v Blenderu do formátu *COLLADA*. Aplikace model zobrazí za využití Linear blend skinning (kapitola 2.2.1) algoritmu. Uživatel bude mít možnost zobrazení pouze kůže modelu, nebo kostry, nebo obojího zároveň. Model bude rozhýbán animací chůze. Bude možné upravit animaci pomocí parametrů. K tomu bude využito grafické uživatelské rozhraní. Animace bude schopna se přizpůsobit nerovnému terénu.

6.1 Načtení modelu

Model je načítán z formátu *COLLADA*, který podporuje uložení geometrie, efektů, fyziky, animace, kinematik scény aj. Využívá XML schéma. Při načítání uvažujeme, že model do tohoto formátu byl exportován z Blenderu.

Začne se načtením geometrie modelu z XML elementu `library_geometries`. Uloží se pole s pozicemi vrcholů, normálami a texturovacími koordináty. Načtou se seznamy polygonů, které jsou určeny odkazy do polí načtených v předchozím kroku. Očekává se použití pouze trojúhelníků. Těchto seznamů může být více, liší se použitým materiálem a v naší aplikaci jeden tento seznam tvoří jeden *Mesh* (kód 6.1). Název materiálu je nastavený v Blenderu a při načítání modelu se očekává přiložení textury ve formátu png se shodným názvem a příponou `_D` pro difuzní texturu.

Následuje načtení vah a přiřazení kostí vrcholům z elementu `library_controllers`. První se přečeťte *bind shape* matici. Ta popisuje transformaci geometrie do správného souřadného systému pro použití s klouby. Poté je načten seznam názvů kostí, seznam inverzních bind pose matic \mathbf{B}_i (viz rovnice 2.1) pro každou kost a seznam vah vrcholů. Následuje seznam, který určuje pro každý vrchol počet kostí, které ho ovlivňují a stejný počet odkazů do seznamu kostí a vah.

Nyní se načte samotná hierarchie kostry z XML elementu `library_visual_scenes`. První uzel obsahuje kořenovou transformaci, od které se odvíjí další transformace všech

```

class Mesh {
protected:
    std::vector<glm::vec3> v; //pozice vrcholu
    std::vector<glm::vec3> n; //normaly
    std::vector<glm::vec2> t; //texturovaci koordinaty
    std::shared_ptr<Material> m; //material
public:
    /* ... */};

class WeightedMesh : public Mesh {
private:
    std::vector<glm::vec4> weights; //vahy prirazenyh kosti
    std::vector<glm::ivec4> jointIndices; //prirazene kosti
public:
    /* ... */};

```

Kód 6.1: Třída modelu.

kostí. Kosti jsou hierarchicky zanořené, obsahují název a lokální transformační matici, viz kód 6.3. Kosti se ukládají do 2D pole, kde každá kost má uložený index nadřazeného uzlu, tedy rodičovské kosti a $0 \dots n$ indexů na své potomky, viz kód 6.2. Kostem se následně přiřadí inverzní bind pose maticy vynásobené bind shape maticí, ta se dále neukládá. Pro správné zobrazení kostry je nutné určit velikost kostí, ta se určí jako rozdíl pozice dané kosti a jejího potomka. Pokud ale kost nemá žádného potomka nemůžeme zjistit její velikost. Proto uvažujeme poloviční velikost než je velikost rodičovské kosti.

Nakonec jsou naplněny objekty třídy `WeightedMesh` (kód 6.1). Abychom mohli použít `vec4` pro vstup kostí do vertex shaderu, musí mít každý vrchol přiřazené maximálně čtyři kosti. Pro model lidského těla jsou čtyři kosti dostatečné a pokud je jich více, mají některé velmi nízké váhy. Proto jsou pro každý vrchol seřazeny všechny jeho kosti sestupně podle vah a uloženy pouze první čtyři. Váhy ostatních jsou rozloženy rovnoměrně mezi ostatní, aby zůstala zachována podmínka o celkovém součtu všech vah (viz 2.2.1).

6.2 Zobrazení

Před každým zobrazením snímku je nutné spočítat globální transformační matice pro všechny kosti kostry. Proto procházíme pole s kostmi a pro každou kost vynásobíme globální matici rodiče s její lokální maticí. Poté získáme transformační matici \mathbf{T} podle rovnice 2.1. Následně nastavíme kořenovou transformační matici jako transformační matici modelu. Dále ve vertex shaderu získáme pozici vrcholu kůže násobením maticí \mathbf{T} a odpovídající vahou pro čtyři přiřazené kosti. Přeypočítáváme také normálu, viz kód 6.4.

Následuje zobrazení kostry. Pro každou kost se použije shodný model. Pro kosti se používají globální matice shodné jako při zobrazování kůže, ale je v nich započítané zvětšení, aby na sebe kosti navazovaly. Výsledná transformační matici kosti je kořenová transformace vynásobená s touto globální maticí kosti. Před zobrazením je vyčištěn hloubkový buffer, aby se kostra překreslila přes kůži. Pro zobrazení jsou použity difuzní textury a Phongův osvětlovací model. Výsledný výstup je na obrázku 6.1.

```

struct Bone {
    glm::mat4 localMat; //lokalni transformace
    glm::mat4 inverseBindMatrix; //inverzni bind pose matice
    int parent; //odkaz na rodicovskou kost
    std::vector<int> childs; //odkaz na potomky
    float scale; //velikost kosti, slouzi pro zobrazeni kosti
};

class Skeleton {
private:
    std::vector<Bone> bones; //pole kosti
    glm::mat4 rootTransform; //korenova transformacni matice
    std::vector<glm::mat4> globalMat; //globalni transformace
public:
    /* ... */ };

```

Kód 6.2: Třída kostry.

```

<library_visual_scenes>
    <visual_scene id="Scene" name="Scene">
        <!-- ... -->
        <node id="metarig" name="metarig" type="NODE">
            <matrix sid="transform">1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1</matrix>
            <node id="hips" name="hips" sid="hips" type="JOINT">
                <matrix sid="transform">
                    1 0 0 0 0 -0.16 -0.98 0.05 0 0.98 -0.167 1.01 0 0 0 1
                </matrix>
                <node id="spine" name="spine" sid="spine" type="JOINT">
                    <matrix sid="transform">
                        1 0 0 0 0 0.99 0.11 0.17 0 -0.11 0.99 1.49e-8 0 0 0 1
                    </matrix>
                    <node id="chest" name="chest" sid="chest" type="JOINT">
                        <matrix sid="transform">
                            1 0 0 0 0 0.99 0.14 0.15 0 -0.14 0.99 7.45e-9 0 0 0 1
                        </matrix>
                        <!-- ... -->
                    </node>
                </node>
                <!-- ... -->
            </node>
        </node>
        <!-- ... -->
    </visual_scene>
</library_visual_scenes>

```

Kód 6.3: Ukázka uložení hierarchie kostry ve formátu COLLADA.

```

vec4 pos = vec4(0.0);
pos += T[v_joints.x] * vec4(v_pos, 1.0) * v_weights.x;
/* ... */
vec3 norm = vec3(0.0);
norm += ti_T[v_joints.x] * v_norm * v_weights.x;
/* ... */
gl_Position = mvp * vec4(pos.xyz, 1.0);

```

Kód 6.4: Ukázka z vertex shaderu pro animovaný model. Výpočet pro pozici a normálu se provede 4×, pro komponenty x, y, z a w vektorů `v_weights` a `v_joints`.

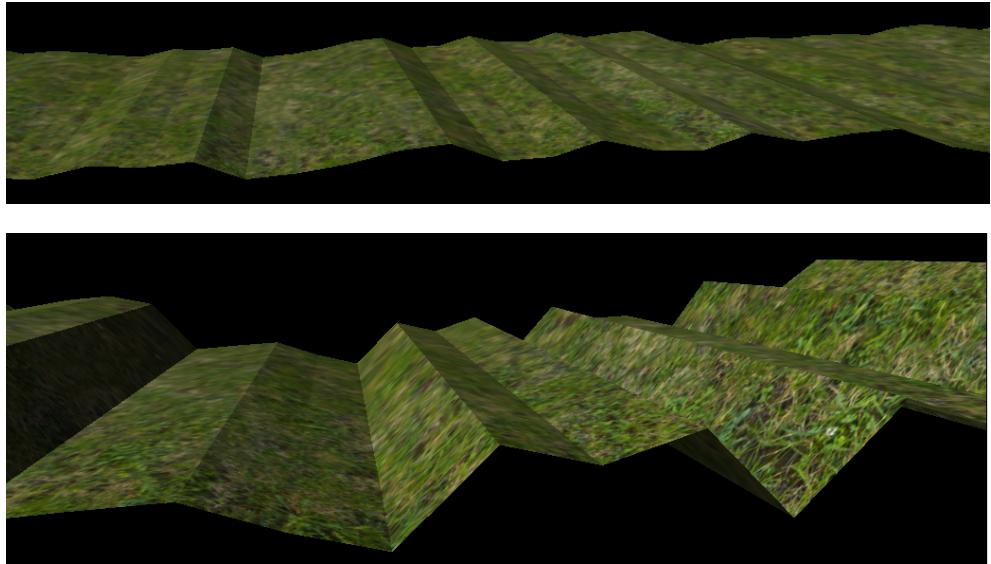


Obrázek 6.1: Grafický výstup aplikace pro dva různé modely. Model vlevo je převzat z [22] a vpravo z [5]

6.2.1 Terén

Jako terén pro demonstraci animace se využívá *výškové pole*. Výškové pole se skládá ze čtverců spojených v jedné řadě, jejichž spojné úsečky mohou mít různou výšku. Terén může mít tři varianty. Bud je celý rovný, nebo má schodovitý tvar a nebo je náhodně vygenerovaný.

Pro náhodně vygenerovaný terén je použit *Midpoint* algoritmus v jedné dimenzi [19]. Algoritmus začíná s jednou horizontální úsečkou. Opakováně nalezne prostřední bod úsečky a změní jeho výškovou souřadnici o náhodnou hodnotu a přitom sníží rozsah této náhodné hodnoty pro další použití. Algoritmus je implementovaný rekurzivně. To jak bude terén hladký nebo naopak hrubý je určeno konstantou $0.0 \leq H \leq 1.0$. Rozsah náhodné hodnoty je pro každé zanoření vynásoben 2^{-H} , tzn. pro velké hodnoty H bude terén velmi hladký a pro malé H bude zase velmi hrubý, viz obrázek 6.2.



Obrázek 6.2: Terén vygenerovaný pomocí Midpoint algoritmu – nahoře pro $\mathbf{H} = 0.3$ a dole pro $\mathbf{H} = 0.1$.

6.3 Animace

Animace je vytvářena nad kostrou z obrázku 2.1, tedy model musí mít připojenou právě tuto kostru, aby animace fungovala. Popis animace je rozdělen na pohyb nohou, pánev a horní poloviny těla. Ve všech těchto částech je potřeba určovat rotace a trajektorie kloubů. K tomu slouží Bézierovy křivky. Prvně je určen pohyb pánev a poté se za využití inverzní kinematiky nastaví klouby nohou. Horní končetiny se hýbou opačně k dolním končetinám. Při popisu animace budeme využívat referenčních rovin zobrazených na obrázku 6.3.

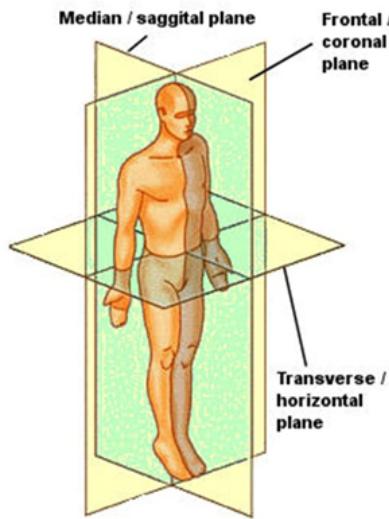
6.3.1 Bézierovy křivky

Bézierovy křivky [15] jsou parametrické interpolační křivky, které nám umožní přesnou approximaci s malým množstvím kontrolních bodů. Zvolili jsme kubické Bézierovy křivky, které používají čtyři kontrolní body, kde křivka prochází pouze krajinami. Dva prostřední body určují tvar křivky. Segmenty těchto křivek je možné skládat za sebe. Jestliže chceme plynulou návaznost, je nutné zajistit aby třetí bod předchozího segmentu a druhý následujícího byly kolineární. Máme-li parametr t potom se pozice \mathbf{B} na křivce určí následovně:

$$\mathbf{B}(t) = (1-t)^3 \mathbf{P}_0 + 3(1-t)^2 t \mathbf{P}_1 + 3(1-t)t^2 \mathbf{P}_2 + t^3 \mathbf{P}_3 \quad (6.1)$$

kde $0 \leq t \leq 1$ a $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2$ a \mathbf{P}_3 jsou kontrolní body křivky.

Křivky, které chceme approximovat jsou ve 2D prostoru. Potřebujeme pro konkrétní hodnotu x získat odpovídající hodnotu y . Bézierovy křivky ale vrací hodnotu podle parametru t , který se nepohybuje po křivce lineárně. Proto musíme pro naše x určit odpovídající t a to poté použít pro získání y . To je možné řešit tak, že se určí větší množství bodů na křivce (např. 100) a pro každý se získá hodnota x . Když je potom potřeba určit t pro nějaké x , najde se vhodný interval a t se určí lineární interpolací. Protože se ale naše křivky budou měnit s nastavováním parametrů a nerovností terénu, toto řešení není vhodné, protože vy-



Obrázek 6.3: Referenční roviny pohybu – horizontální, frontální a sagitální.¹

žaduje při každé změně přepočítat všechny hodnoty. Proto jsme zvolili řešení numerickou metodou.

Zvolili jsme metodu Regula Falsi, která slouží pro řešení jedné nelineární rovnice. Ta konverguje vždy a v našem případě kdy řeší hladké a monotónní křivky i velmi rychle. Funguje na principu dělení intervalu $\langle a, b \rangle$ na dvě obecně různé části. Dělícím bodem intervalu je průsečík osy x s úsečkou spojující $[a, f(a)]$ a $[b, f(b)]$ určený následovně:

$$x_k = b_k - \frac{b_k - a_k}{f(b_k) - f(a_k)} f(b_k) \quad (6.2)$$

Výpočet končí jestliže $|x_k - x_{k-1}| < \epsilon$. Tato metoda najde v našem případě odpovídající parametr t s dostatečnou přesností na cca 4 iterace.

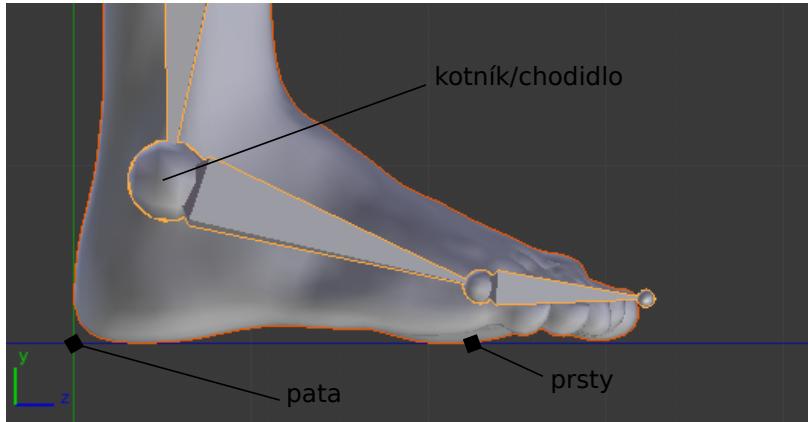
6.3.2 Pohyb nohou

O pohyb nohou se stará třída `Leg` jejíž hlavní metodou, která se volá každý snímek poté co je již pohnutá pánev je metoda `update`. Ta využívá jediný parametr a tím je čas, o který se animace posunula. Jeden krokový cyklus nohy je nastaven na dobu trvání jedné vteřiny. Zrychlení nebo zpomalení kroku je dosaženo úpravou vstupního času. Tento vstupní čas je poté přičten k proměnné t , která se při přetečení přes 1 vteřinu vrací o vteřinu zpět. Podle t se poté určí v jaké fázi krokového cyklu se noha nachází a podle toho je určen další postup.

Nemusíme rozlišovat všech sedm fází cyklu, ale budou nám stačit čtyři. Stádium zatěžování a mezistoj jsou nezměněny. Konečný stoj a předsvihová fáze jsou spojeny do jedné, protože z pohledu animace se tyto fáze neliší, v obou je prováděn stejný úkon – zvedání paty. Poslední tři fáze (počáteční švih, mezišvih a konečný švih) jsou také spojeny do jedné, budeme ji říkat pouze švihová. Tyto fáze je možné spojit, protože celá švihová fáze může být popsána spojité pomocí trajektorie od odrazu palce do počátečního kontaktu.

¹Převzato z <http://sme-chinoises-euronext.typepad.fr/artbot/2014/09/a-study-of-walking-in-order-to-design-a-biped-robot.html>

Pro potřeby správného a přesného určování pohybu nohou potřebujeme znát pozici paty a prstů nohy. Přesně se tyto pozice nedají získat pouze z pozic kostí. Tyto pozice jsou závislé na pozici a rotaci chodidla. Jsou určeny podle umístění kostry v prostoru před samotným provedením animace, viz obrázek 6.4. Takto získané pozice paty a prstů nohy jsou vynásobeny inverzní transformační maticí chodidla. Tím jsou získány relativní pozice vůči chodidlu, z kterých je později možné získat přesně pozici paty a prstů v závislosti na aktuální pozici a rotaci chodila. Stačí vynásobit aktuální transformační maticí chodidla a tyto relativní pozice.



Obrázek 6.4: Vstupní pozice modelu a jeho kostry. Pozice paty má shodné x s kotníkem, ale nulové souřadnice na ose z a y . Pozice prstů nohy má nulovou souřadnici na ose y .

Důležitou podmínkou pro chůzi po nerovném terénu je možnost získat informace o terénu na kterém se noha nachází. Terén je objekt, který je sdílený všemi součástmi těla, kteří s ním potřebují pracovat. Pro nohu je důležité získat výšku terénu pod patou nohy a úhel mezi chodidlem a terénem. Tento úhel se určí pomocí skalárního součinu mezi vektorem směřujícím od paty k prstům nohy a vektorem mezi pozicemi na terénu pod patou a pod prsty.

Stádium zatěžování

Stádium zatěžování nastane pro $t < 0.12$. Před prvním vykonáním této fáze se dokončí švihová fáze. Protože předchozí snímek se vykonával pro nějaké $t < 1.0$ a v aktuálním snímku se přesunul z švihové fáze do stadia zatěžování s nějakým $t \geq 0.0$ je potřeba prvně dokončit předchozí fázi jako by čas byl roven 1. Kdybychom to neudělali animace by nebyla plynulá, jak moc by záviselo na velikosti časového kroku. Také je získán počáteční úhel mezi chodidlem a terénem.

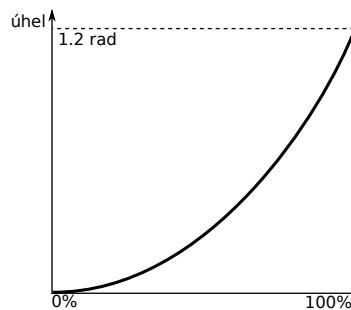
Tato fáze se zabývá zachováním pozice paty na terénu a sklápěním chodidla. Pánev se v každém snímku pohně první, proto je noha na jiné pozici než byla v předchozím snímku. Rotace chodidla se v této fázi určuje lineární interpolací mezi počátečním úhlem s terénem a nulovým úhlem. Protože je pozice paty závislá na rotaci chodidla, musíme prvně natočit chodidlo do požadovaného úhlu se zemí pro aktuální t . Nyní se může určit rozdíl mezi aktuální pozicí paty a předchozí a z toho určit požadovanou pozici kotníku a pomocí inverzní kinematiky kotník do této pozice nastavit. Inverzní kinematika hýbe nohou a tím i mění rotaci chodidla vůči terénu. Proto je po jejím provedení potřeba znova natočit chodidlo do správného úhlu.

Mezistoj

Následující fáze mezistoj nastává pro $0.1 \leq t < 0.31$ a má za úkol udržet chodilo položeno na zemi na stejně pozici. Je vykonávána podobně jako předchozí fáze. Před prvním vykonáním je tedy prvně dokončeno stádium zatěžování. Rozdílem je, že pohyb nevychází z paty, ale vychází z kotníku. Proto nám stačí si na začátku zapamatovat pozici kotníku a tu použít pro inverzní kinematiku. Není tedy před jejím provedením nutné nastavovat novou rotaci chodidla pro určení pozice paty. Rotace je nastavena až po provedení inverzní kinematiky tak, aby úhel mezi terénem byl nulový.

Zvedání paty

Zvedání paty nastává pro $0.31 \leq t < 0.62$. Pro čas do $t = 0.5$ při zvedání paty zůstává koleno převážně propnuté, ve zbylém čase se koleno ohýbá. Aby tohoto bylo dosaženo není vhodné zvedat patu v celé fázi lineárně. Také není vhodné ji rozdělit na dvě samostatné části, potom by zvedání paty nepůsobilo plynule. Nejvhodnějším řešením se zdá použití Bézierovy křivky, kde v první části se zvedá pata méně a v druhé více a celá fáze plynulá. Konkrétní tvar křivky (obrázek 6.5) byl určen experimentálně tak aby tato fáze byla dostatečně přirozená. Také je u této křivky důležitý sklon u konce fáze, který při vhodném zvolení zajistí co nejplynulejší přechod do švihové fáze, tzn. je důležité aby rychlosť pohybu paty na konci této fáze a na začátku švihové byly co nejpodobnější. Rychlosť pohybu nohy ve švihové fázi je závislá na délce kroku. Proto je celkový úhel o který je pata zvednutá zvětšován se zvětšující se délkou kroku. Mimo správnou návaznost je toto zvětšování také nutné pro zvýšení efektivního dosahu nohy, aby byl vůbec delší krok umožněn. Křivka úhlu zvedání paty je vytvořena pro nejdélší možný dvojkrok, který je 1.8 metru dlouhý. Pro jiné délky kroku je vynásobena $c = (dvojkrok/1.8)^2$, tato hodnota byla také zjištěna experimentálně, aby tato fáze vypadala přirozeně.



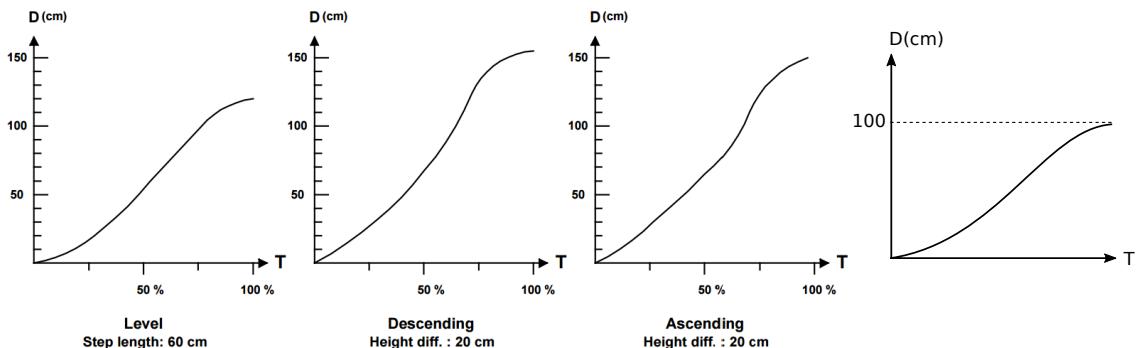
Obrázek 6.5: Rotace chodidla vůči zemi při fázi zvedání paty pro délku dvojkroku 1.8 metru.

Samotné zvedání paty probíhá následovně. Na začátku fáze se uloží pozice prstů, která se bude udržovat. Následně se chodidlo nastaví do úhlu se zemí určeného z křivky zvedání paty. Dále se může určit pozice prstů nohy a posunout kotník o rozdíl původní a aktuální pozice prstů. Tím se kotník dostane do požadované polohy, ale prsty v původní pozici nebudou, protože inverzní kinematika změnila rotaci chodidla vůči zemi a proto je chodidlo znova natočeno na správný úhel. Nakonec zbývá jenom natočit prsty nohy, aby byly v nulovém úhlu se zemí, tzn. natočit na rotaci odpovídající úhlu mezi chodidlem a zemí.

Švihová fáze

Švihová fáze nastává pro $0.62 \leq t \leq 1.0$. Jejím cílem je přesunutí nohy z aktuální pozice na pozici dalšího kroku. V této fázi se využívají čtyři Bézierovy křivky. Při prvním vykonání této fáze se dokončí fáze zvedání paty a zapamatuje se pozice paty, která bude sloužit jako výchozí bod pro švihovou fázi. V průběhu této fáze je také potřeba vrátit prsty na noze do původní polohy, protože při zvedání paty byly otočeny. V první pětině švihové fáze jsou prsty rotovány do pozice určené lineární interpolací mezi otočenou pozicí a původní nataženou pozicí.

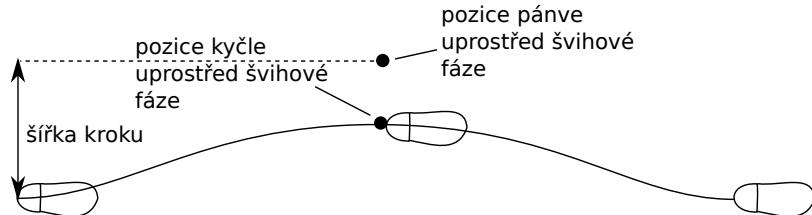
Podle [12] se chodidlo pohybuje vpřed s měnící se rychlostí. Rychlosť je nejvyšší uprostřed švihové fáze a nejpomalejší na začátku a na konci. K tomu slouží Bézierova křivka určující posun chodidla v horizontální rovině, viz obrázek 6.6. Tato křivka je vytvořena pro posun o délce 1 metr. Potom pro různé délky stačí před použitím křivky její kontrolní body vynásobit požadovanou délkou. Jako délku nemůžeme použít aktuální délku dvojkroku, to by se noha přesouvala před požadovanou pozici. Je nutné vzít v úvahu, že při zvedání paty se již pata posune vpřed. Noha zná pozici paty na začátku aktuálního kroku a pozici na kterou chceme aby pata dopadla při konci kroku, tedy začátku dalšího. Pata zůstává na stejně pozici při stádiu zatěžování a při mezistojí. Ale při zvedání paty se již pohybuje v horizontální rovině a délka tohoto pohybu se musí odečíst od délky dvojkroku, abychom získali přesný zbyvající posun nohy na cílovou pozici pro vynásobení Bézierovy křivky. Pro posun vpřed v horizontální rovině poté stačí zjistit hodnotu křivky a přičíst ji k pozici po ukončení zvedání paty.



Obrázek 6.6: Vlevo jsou naměřené grafy posunu chodidla v čase pro chůzi po rovině, z kopce a do kopce převzaté z článku [12]. Vpravo je vytvořena generická Bézierova křivka, která je použita pro posun nohy v horizontální rovině.

V průběhu chůze může být nastavena různá šířka kroku. Není přirozené, aby si chodidlo v průběhu celé švihové fáze udržovalo stejnou vzdálenost do šířky od páne. Působí mnohem věrohodněji, pokud se chodidlo po odrazu od země přesune na šířku kyčle, které dosáhne v polovině švihové fáze a poté se začne přesouvat zpět na šířku kroku, které dosáhne v okamžiku dopadu paty. To by bylo možné vyřešit lineární interpolací mezi šířkou kroku a pozicí kyčle. Tímto způsobem ale animace není plynulá kvůli náhlému zrychlení nohy při odrazu paty a zastavení při dopadu. Proto jsme vytvořili další Bézierovu křivku, která se použije pro plynulé zrychlení a zpomalení tohoto pohybu, viz obrázek 6.7. Časový parametr lineární interpolace se použije pro získání hodnoty této křivky, která se použije místo něho.

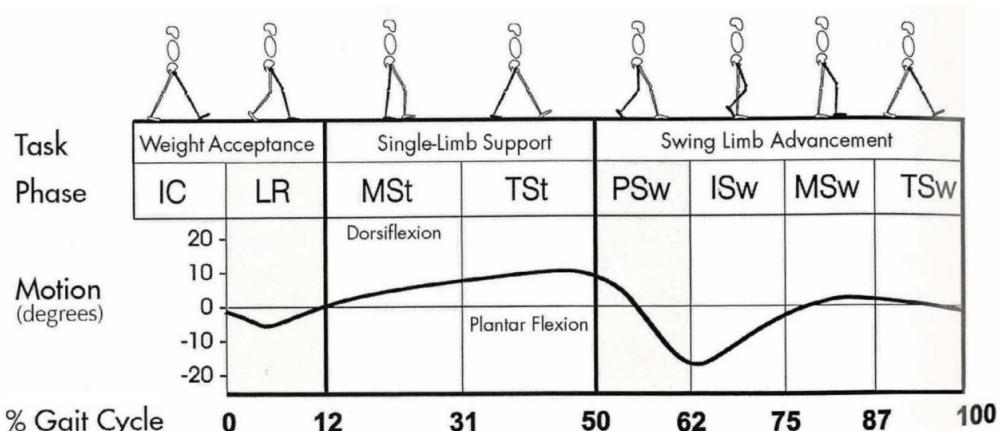
Takto jsou získány dva údaje ze čtyř, které jsou potřeba pro určení pozice chodidla ve



Obrázek 6.7: Přesun chodidla během švihové fáze v závislosti na šířce kroku.

švihové fázi. Máme pozici paty do šířky, vpřed a dále potřebujeme výšku a rotaci chodidla. Když máme všechny tyto čtyři údaje můžeme nastavit správnou rotaci v kotníku. A poté z ní určit novou pozici kotníku tak, aby pata byla na správné pozici podle naší určené šířky, výšky a posunu vpřed. S touto novou pozicí kotníku se spočítá inverzní kinematika. Zbylé dva údaje rotace a výšky chodidla je prvně nutné přizpůsobit nerovnému terénu.

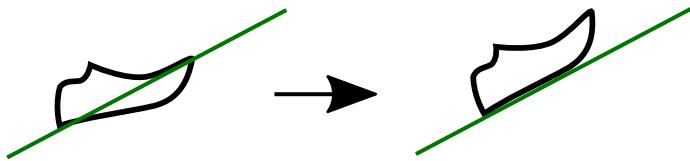
Pro rotaci chodidla v průběhu švihové fáze vycházíme z obrázku 6.8, kde je uvedena rotace v kotníku v průběhu celého cyklu. Část toho grafu ve švihové fázi je approximována pomocí Bézierovy křivky. Tato křivka začíná pro 15 stupňů plantární flexe, tedy propnutí kotníku z neutrální polohy, a končí u přibližně neutrální polohy. Tuto křivku je nutné přizpůsobit propnutí kotníku na začátku švihové fáze, které se může lišit v závislosti na délce kroku a sklonu terénu. Kontrolní body křivky jsou proto vynásobeny koeficientem $c = aktRot/15^\circ$, kde $aktRot$ je aktuální propnutí v kotníku.



Obrázek 6.8: Rotace kotníku v průběhu krokového cyklu. Převzato z [1].

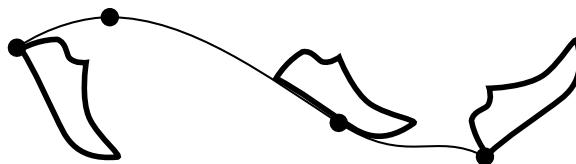
Nyní je potřeba vyřešit úhel dopadu chodidla na terén. Jedna možnost je mít pevně daný úhel, pod kterým bude chodidlo vždy dopadat. Zdá se ale přirozenější, nechat úhel podle naší křivky a upravit ho jenom v nevhodných případech. Tímto případem je možnost průchodu chodidla skrz terén při sklonu do kopce (obrázek 6.9), ve kterém je rotace upravena tak, aby při dopadu byl sklon chodidla shodný s terénem.

Pro úpravu rotace chodidla je nutné před samotným provedením kroku vyzkoušet pozici na konci kroku při dopadu paty. To vyžaduje určení transformační matice párnve při dopadu paty a posunutí nohy na cílovou pozici pomocí inverzní kinematiky. Poté je zkонтrolován úhel mezi chodidlem a terénem a jestliže je záporný, tzn. chodidlo se nachází pod terénem je upravena křivka rotace. Potřebujeme upravit pouze koncový úhel propnutí kotníku a proto jsou o potřebný úhel posunuty poslední dva kontrolní body křivky.



Obrázek 6.9: Pozice chodidla na terénu při dopadu paty s původní rotací a po úpravě rotace.

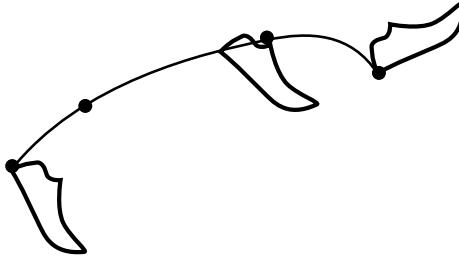
Poslední co je potřeba určit je výška chodidla. K tomu slouží poslední Bézierova křivka použitá pro pohyb nohy, která je approximována podle [13], viz obrázek 6.10. Tato křivka určuje výšku paty a skládá se ze tří segmentů, to nám umožňuje křivku dobře tvarovat při přizpůsobování se nerovnému terénu. Obdobně jako při rotaci chodidla je tato křivka vytvořena pro jeden případ a poté upravena. Konkrétně je vytvořena pro patu nacházející se 20 centimetrů nad terénem při počátku švihové fáze. Pro případy kdy se pata nachází výše nebo níže jsou všechny kontrolní body vynásobeny koeficientem, který zajistí aby výška paty odpovídala a švihová fáze plynule navazovala na zvedání paty.



Obrázek 6.10: Výška paty při švihové fázi pro chůzi po rovném terénu.

Kvůli přizpůsobení křivky terénu se před samotným provedením kroku krok odsimuluje. K tomu je potřeba zvolit vhodný časový rozestup. Zvolili jsme že se výpočet provede 50krát, to je dostatečně přesné i rychlé. V každém kroku simulace se určí pozice pánve a pomocí inverzní kinematiky se noha přesune na novou pozici obdobným způsobem jako při samotném provádění kroku chůze. Poté se zkонтroluje zda požadovaná pozice, určená mimo jiné právě křivkou výšky paty, byla v dosahu nohy, tzn. vzdálenost od kyčle k požadované pozici je větší než délka stehenní a holenní kosti. Jestliže není, je křivka v daném místě zvednuta. Poté co je již pozice v dosahu, je zkontovalo, zda se pata nebo prsty nohy nenachází pod úrovní terénu. Jestliže ano je křivka znova adekvátně zvednuta. Tím jeden krok simulace končí.

Před samotnou simulací je křivka upravena tak, aby výška, tedy souřadnice na ose y , posledního kontrolního bodu byla posunuta o h , kde h je rozdíl výšky terénu aktuálního a následujícího kroku. Toho je dosaženo vynásobením souřadnice y koeficientem $c = x \times h$, kde x je souřadnice na ose x a nabývá hodnot od 0 do 1. Tím je křivka přizpůsobena změně výšky terénu, je zachován její tvar a i plynulost díky zachování kolinearita kontrolních bodů při navazování segmentů. Další úprava křivky nastane, pokud při simulaci musí být křivka pro danou pozici na ose x zvednuta. Při této úpravě se prvně naleze segment, do kterého spadá x . Budou upravovány okrajové kontrolní body tohoto segmentu (pokud nejsou okrajové pro celou křivku) a k nim přilehlé kontrolní body. Celkem tedy budou upravovány tři nebo jeden kontrolní bod na každé straně segmentu. Těmto bodům je k souřadnici y přičtena hodnota určující o kolik je potřeba křivku zvednout. Jestliže se x nachází v prostředním resp. druhém segmentu, je hodnota rozložena na obě strany tak, že čím více vlevo je x , tím jsou více zvedány levé kontrolní body a čím více vpravo, tím jsou více zvedány pravé kontrolní body. Při této úpravě je také zachována plynulost křivky. Příklad upravené křivky je na obrázku 6.11.



Obrázek 6.11: Křivka určující výšku paty při švihové fázi přizpůsobená nerovnému terénu.

Inverzní kinematika

Kinematický řetězec který potřebujeme řešit obsahuje pouze dva klouby – kyčel a koleno. Pro takový řetězec existuje rychlé analytické řešení. Pozici kyčle známe a požadovanou cílovou pozici koncového efektoru (kotníku) také. Kromě toho známe i délku stehenní a holenní kosti. Rotace kyčle a kolene se poté dopočítá jako úhly v trojúhelníku pomocí kosinové věty.

Tímto ještě není inverzní kinematika vyřešená, protože může být nekonečně mnoho řešení s různými rotacemi okolo osy mezi kyčlí a kotníkem, viz obrázek 6.12. Pomocí kosinové věty jsme pro stehenní kost určili úhel α_1 , který určuje rotaci mezi vektorem \mathbf{a} a stehenní kostí \mathbf{v} . Abychom nastavili kost do správné pozice, je potřeba vektor \mathbf{v} natočit do \mathbf{a} . K tomu je sestavena transformační matice, jejímž vynásobením se \mathbf{v} dostane do \mathbf{a} . Poté je již možné rotovat kyčel o úhel α_1 a rotovat holeň do úhlu α_2 .

K tomu abychom určili výslednou rotaci okolo vektoru \mathbf{a} slouží referenční vektor [4]. Výsledné řešení se poté vybere takové, aby úhel δ mezi referenčním vektorem a stehenní kostí byl co nejmenší. Proto budeme rotovat vektor \mathbf{v} o úhel θ do nového vektora \mathbf{v}' . Minimalizace úhlu δ je ekvivalentní maximalizaci $\cos \delta$ (resp. $\mathbf{v}' \cdot \mathbf{k}$) a to nastane když je první derivace rovna nule:

$$\frac{d}{d\theta} \mathbf{v}' \cdot \mathbf{k} = 0 \quad (6.3)$$

Po úpravě dostaneme:

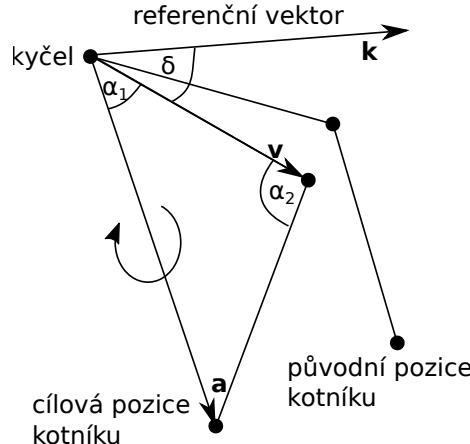
$$\theta = \tan^{-1} \frac{2B}{C - A} \quad \vee \quad \theta = \pi + \tan^{-1} \frac{2B}{C - A} \quad (6.4)$$

kde $A = (\mathbf{a} \cdot \mathbf{v})\mathbf{a} \cdot \mathbf{k} - ((\mathbf{a} \times \mathbf{v}) \times \mathbf{a}) \cdot \mathbf{k}$, $B = (\mathbf{a} \times \mathbf{v}) \cdot \mathbf{k}$ a $C = \mathbf{v} \cdot \mathbf{k}$. Abychom určili, které ze dvou řešení maximalizuje $\cos \delta$ musíme určit druhou derivaci $\cos \delta$:

$$\frac{d^2}{d\theta^2} \cos \delta = \frac{A - C}{2} \cos \theta - B \sin \theta \quad (6.5)$$

Úhel δ je minimální, když je druhá derivace menší než nula.

Referenční vektor směruje vpřed ve směru chůze a leží v horizontální rovině. Důležité je, že tento vektor nevychází z kyčle, protože by byl ovlivněn rotacemi páne v průběhu krokového cyklu a nohy by se vytáčely vně a dovnitř a jejich natočení by nebylo konzistentní. To, že je vektor v horizontální rovině, umožňuje se stehnu natočit vpřed i vzad do pravého úhlu, než se dostane nad vektor a začne se nekonzistentně natáčet. Tato pozice je vhodná, protože noha se takto vysoko v průběhu chůze dostává velmi zřídka. Kdyby se referenční vektor určoval v závislosti na aktuální pozici nohy a vždy předcházel stehnu a konstantní úhel, tento problém by v okrajových případech nenastával, ale v průběhu celého cyklu by



Obrázek 6.12: Pro dosažení cílové pozice v kotníku se pomocí kosinové věty určí úhly v kyčli a v koleni. Rotace okolo osy mezi kyčlí a kotníkem se určí tak, aby úhel δ byl co nejmenší.

natočení nohy nebylo konzistentní. Otáčením vektoru okolo svislé osy se poté dá dosáhnout různých úhlů vytočení chodidla.

Řešení inverzní kinematiky rotuje holeň okolo její lokální osy x a tedy aby výsledná pozice kotníku odpovídala té požadované, noha nemůže být libovolně natočena vůči své rodičovské kosti (stehnu). Vstupní model může mít koleno různě natočené a proto před samotným začátkem animace je nutné lokální matici holeně zbavit všech rotací a zachovat pouze translaci.

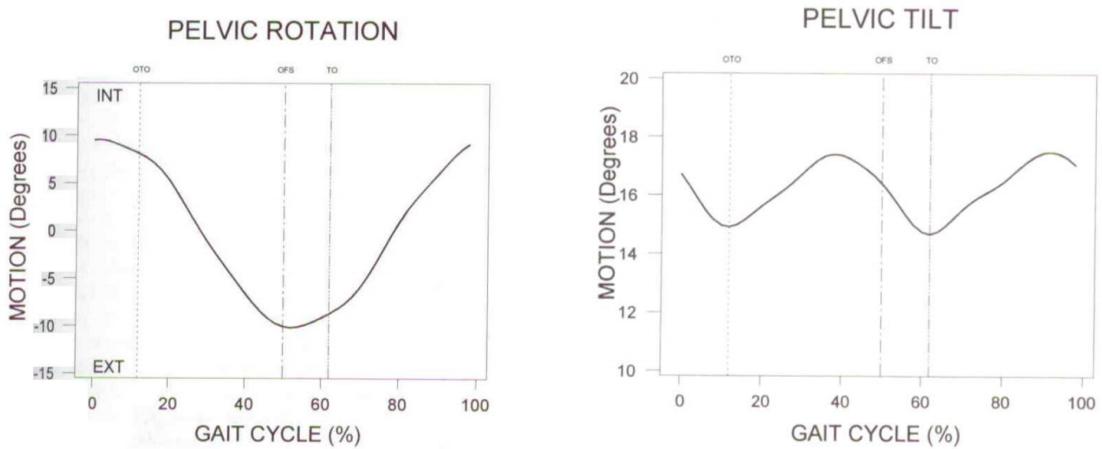
6.3.3 Pohyb pánev

Pánev slouží jako kořen kinematického řetězce. Odvíjí se od ní nejen pohyb nohou, ale i horní poloviny těla. Její pohyb je implementován ve třídě `Skeleton`, která zapouzdřuje celou animaci. Umožňuje nastavovat parametry animace a poskytuje transformační matice kostí pro zobrazování. Na začátku tato třída inicializuje kostru do počáteční pozice. To znamená nastavení počátečních rotací a posunutí pánev, trupu a hlavy. Dále vytvoření objektů pro levou a pravou nohu a ruku. Levé a pravé končetiny se liší nastaveným počátečním časem, který je posunutý o půl vteřiny a díky tomu se končetiny střídají. Obdobně jako třída `Leg` tato třída obsahuje metodu `onUpdate`, které je předáván čas od předchozího snímku animace. Tento čas je omezen na 0.04 sekundy, tzn. jeden krokový cyklus nemůže mít méně než 25 snímků. Kdyby čas nebyl omezen, při velkých časových krocích by na sebe nenavazovaly jednotlivé části krokového cyklu.

Pánev rotuje ve všech třech rovinách – frontální, horizontální a sagitální. K tomu jsou využity tři Bézierovy křivky. Ve frontální rovině je úklon pánev approximován podle [18] a je k vidění na obrázku 6.18. Pro rotaci v horizontální rovině a náklon v sagitální rovině je čerpáno z [1] (obrázek 6.13).

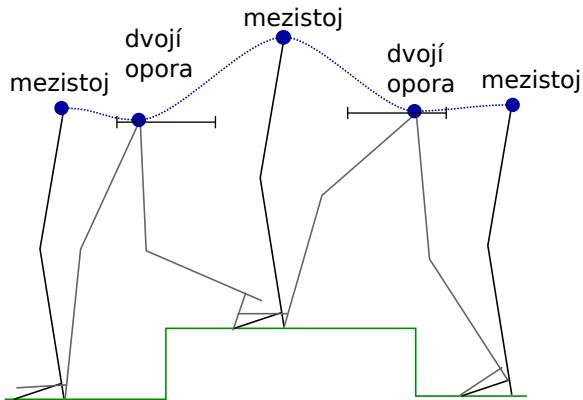
Pánev se vpřed nepohybuje po přímce, ale v horizontální a sagitální rovině její trajektorie připomíná sinusoidu [18]. V sagitální rovině je maximální vertikální posun uprostřed stojné fáze a minimální ve fázi dvojí opory. Pro umožnění chůze po nerovném terénu budeme přizpůsobovat právě vertikální posun pánev. Použití Bézierových křivek nám umožní proložit krajinářské body segmentů maximem a minimem této trajektorie.

Při chůzi po nerovném terénu vertikální maximum pánev upravíme podle pozice stojné nohy, aby propnutí kolena bylo nezměněno. Rozdíl ve výšce terénu v pozici předchozího



Obrázek 6.13: Rotace a náklon pánve v průběhu krokového cyklu. Převzato z [1].

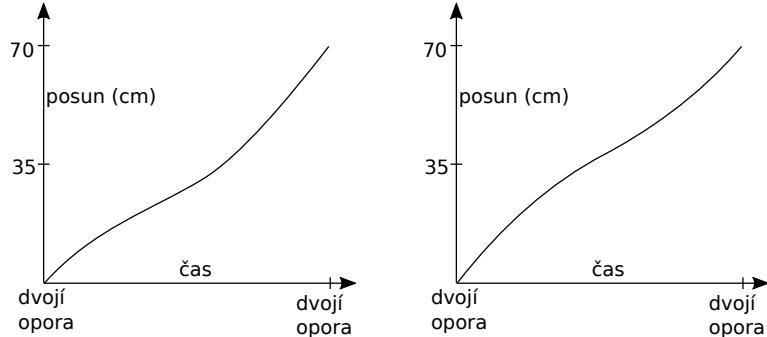
a následujícího umístění chodidla určuje změnu ve vertikálním maximu. Jestliže výška terénu je pro další krok nižší, snížíme o tuto hodnotu i vertikální minimum. Díky tomu může noha dosáhnout na cílovou pozici. Pokud je následující pozice nohy výše než předchozí, vertikální minimum neměníme, protože při chůzi do kopce se noha pokrčí v kolenu a nemá problém dosáhnout na cílovou pozici. Výškový rozdíl pánve poté vyrovnáme až při narovnávání stojné nohy, viz obrázek 6.14.



Obrázek 6.14: Změna vertikální pozice pánve při chůzi po nerovném terénu. Obrázek také znázorňuje možné pozice pánve v dvojí opoře.

Podobně jako při švihové fázi nohy se pánev pohybuje vpřed s variací v rychlosti. Pohyb se zrychluje se snižováním vertikální pozice a je nejrychlejší uprostřed dvojí opory, kdy je kinetická energie nejvyšší. Nejpomaleji se pánev pohybuje ve vertikálním maximu uprostřed stojné fáze. Posun pánev vpřed určujeme podle Beziérových křivky, kterou jsme vytvořili podle grafů uvedených v [12]. Grafy se liší pro chůzi po rovině, z kopce, nebo do kopce. Vytvořili jsme jednu křivku pro chůzi po rovině a pro chůzi po nerovném terénu ji budeme upravovat podle pozice pánev ve dvojí opoře, viz obrázek 6.15. Vertikální minimum je v čase vždy uprostřed dvojí opory. Pozice pánev ve dvojí opoře se ale může lišit. Pozici pánev se dá manipulovat změnou křivky posunu, je nutné, aby pozice pánev v mezistojí byla vždy

shodná. Z toho důvodu se křivka skládá ze dvou samostatných křivek. První určuje posun od dvojí opory k mezistoji a je upravena podle vzdálenosti mezi pozicemi pánve v těchto dvou okamžicích. Druhá určuje posun od mezistoje k dvojí opoře a je upravena obdobně.



Obrázek 6.15: Posun pánve v čase mezi dvojími oporami. Vlevo je chůze po rovině, vpravo je křivka upravena podle pozice pánve při chůzi z kopce.

Pro určení pozice pánve ve vertikálním minimu je použit optimální přístup využívající rovnici z článku [12]. Vybírá se taková pozice pánve, aby úhlová akcelerace kloubů stojné nohy byla minimální. Protože chceme mít určenou trajektorii pánve již před určováním trajektorie švihové nohy, určíme ji v okamžiku dvojí opory, než se švihová noha odrazí od země.

Prvně se v rozsahu možných pozic určí úhlová akcelerace na pěti místech a poté se okolí místa s nejnižší akcelerací rozdělí na 5 dalších pozic, ze kterých se zase určí ta s minimální akcelerací a ta je již finální. Výpočet úhlové akcelerace pro jednu pozici pánve vyžaduje následující úkony. Upravení jedné křivky posunu na vzdálenost mezi následujícím mezistojem a další dvojí oporou. A upravení druhé křivky posunu pro vzdálenost mezi další dvojí oporou a jí následujícím mezistojem. Dále je nutné určení trajektorie pánve v sagitální rovině v závislosti na nerovnosti terénu. Poté se simuluje celý krok – celkem 50krát. Z toho polovina simulačních kroků se provede pro část mezi mezistojem (MS) a dvojí oporou (DO). Pro aktuální stojnou nohu se pomocí inverzní kinematiky a podle aktuální simulované fáze kroku vypočítá pohyb kloubů nohy, obdobně jako při samotné animaci chůze. V dvojí opoře se stojná noha změní na druhou a pro tuto druhou nohu se provede dalších 25 simulačních kroků. Při simulaci pohybu kloubů nohou je určován koeficient c podle následující rovnice:

$$c = \sum_{i=1}^3 \mathbf{W}_i \int_{t_{MS}}^{t_{DO}} f_i(t) d^2 t + \sum_{j=1}^3 \mathbf{W}_j \int_{t_{DO}}^{t_{MS}} f_j(t) d^2 t \quad (6.6)$$

kde $f_i(t)$ je rotace i -tého kloubu v kinematickém řetězci pro danou pozici pánve a chodidla a čas t , \mathbf{W}_i je váha i -tého kloubu.

Nakonec se z těch pozic, pro které stojná noha v průběhu celého kroku dosáhne na svoji pozici na zemi, vybere ta s nejmenším c a je použita pro následující krok. V článku [12] určují také vertikální pozici pánve. Tu my máme určenou parametrem velikosti amplitudy v sagitální rovině. Díky tomu máme menší množství pozic, které musíme prozkoumat, máme lepší kontrolu nad vertikálním posunem, ale pohyb není z hlediska akcelerace kloubů tak optimální.

Důvod proč se ve dvojí opoře neprovádí simulace mezi aktuální pozicí a další dvojí oporou, ale provádí se od následujícího mezistoju k dalšímu mezistoji, je kvůli tomu jaká

noha je stojná. V prvním případě by celou simulaci byla jedna stojná noha a zvolená pozice pánve byla nevhodná, protože by nebrala v úvahu druhou nohu. Zatímco v druhém případě se stojná noha vystřídá a výsledná pozice tedy vyjadřuje kompromis mezi oběma nohami a pozice pánve je poté mnohem vhodnější. Simulace se neprovádí až ve chvíli, kdy je pánev v mezistojí, protože musíme mít určenou pozici pánve v další dvojí opore již od začátku švihové fáze, která začíná při odrazu nohy na konci dvojí opory. Zatímco mezistoj je až v polovině švihové fáze.

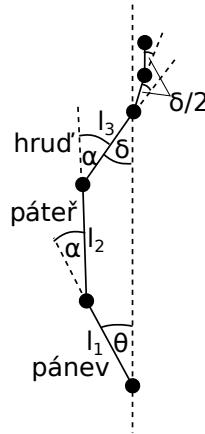
6.3.4 Pohyb horní poloviny těla

Lidský trup při chůzi vyrovnává rotaci pánve a natáčí ramena do opačného směru než pánev. Hlava směřuje vpřed a výrazně se nehýbe. V horizontální rovině je rotace ramen podobná rotaci pánve. Má opačnou amplitudu a dosahuje maxima před fází dvojí opory a během ní zůstává nezměněna [11]. Tato rotace nevychází pouze z ramen, ale z celého trupu. V našem případě rotaci rovnoměrně rozkládáme mezi páteř, hrud' a ramena. Hlava je rotována v opačném směru a tato rotace je rozdělena mezi hlavu a krk tak, že hlava směřuje vpřed.

Ve frontální rovině rotaci pánve vyrovnáváme tak, aby pozice krku byla ve svislé poloze s páneví (obrázek 6.16). Dále je natočena hlava podle úhlu δ , aby byla rovnoběžná se svislou osou. Úhel potřebné rotace páteře a hrudi α pro danou rotaci pánve θ je získán vyřešením následující rovnice:

$$l_1 \sin(\theta) = l_2 \sin(\theta + \alpha) + l_3 \sin(\theta + 2\alpha) \quad (6.7)$$

Tato rovnice je řešena numericky pomocí metody Regula Falsi.

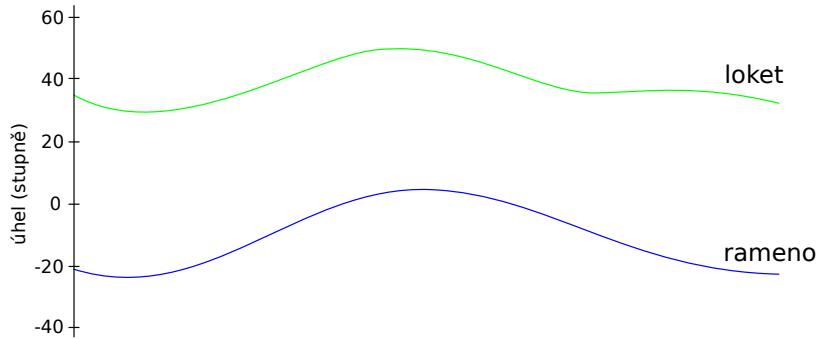


Obrázek 6.16: Pohyb trupu a hlavy v závislosti na rotaci pánve ve frontální rovině. Trup je rotován tak, aby pozice krku byla v ose nad páneví.

Pohyb rukou

O pohyb rukou se stará třída `Arm`. Obdobně jako třída `Leg` obsahuje metodu `update`, které je předáván čas od posledního snímku animace. Pohyb rukou je ale výrazně jednodušší. Inverzní kinematika je využita pouze jednou při inicializaci pro nastavení rukou do výchozí pozice podél těla. Inverzní kinematika je řešená stejně jako pro nohy, ale místo stehenní kosti se používá paže a místo holenní kosti předloktí.

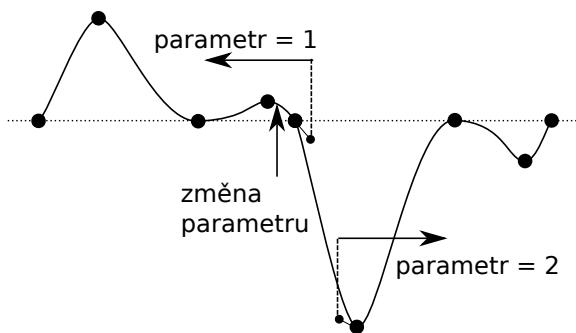
Dále je každý snímek rotována paže a předloktí pomocí Bézierových křivek. Prvně je paže rotována okolo svislé osy o aktuální rotaci pánve, díky tomu bude ruka směrovat vpřed. Poté je vyrovnaný úklon pánve, aby ruka směřovala dolů. Nakonec je rotována paže a loket podle křivek na obrázku 6.17. Ruce se pohybují v opačném směru než nohy, tzn. jsou v největším rozpětí ve fázi dvojí opory.



Obrázek 6.17: Křivky pro rotaci v rameni a v lokti, grafy jsou approximovány podle [11].

6.4 Nastavování parametrů

Systém umožňuje nastavování parametrů chůze v reálném čase. Pro parametry jako je pozice a rotace pánve a rukou je možné zvětšovat amplitudu Bézierových křivek. Aby byl pohyb při změně parametrů plynulý, musíme zajistit návaznost segmentů. Toho dosáhneme tak, že změníme hodnotu třetího a čtvrtého kontrolního bodu následujícího segmentu od aktuální pozice, viz obrázek 6.18. A dále změníme hodnoty všech následujících kontrolních bodů.



Obrázek 6.18: Bézierova křivka určující úklon (rotaci ve frontální rovině) pánve. Při změně parametru jsou změněny kontrolní body začínající třetím kontrolním bodem následujícího segmentu.

Na nejvyšší úrovni kontroly umožňujeme specifikovat rychlosť chůze. Výzkum [21] ukazuje, že se zvyšující se rychlosť chůze se snižuje pohyb pánve v horizontální rovině a zvyšuje ve vertikální rovině. Také se zvyšuje délka kroku, počet kroků za minutu a šířka kroku. Aby nohy dosáhly na požadovanou délku kroku, snižujeme i maximální výšku pánve, přestože se v článku nezmiňuje.

Dále se dají parametry nastavovat samostatně. Pro pohyb pánve je možné měnit amplitudy křivek pro rotaci, úklon, náklon pánve, ale také horizontální a vertikální posun. Kromě

toho je možné určit maximální výšku pánve, předklon trupu a pozici pánve v mezistojí, tedy posun pánve vůči chodidlu stojné nohy v maximu vertikálního pohybu pánve.

Pro pohyb nohou je možné nastavit délku a šířku kroku. Dále úhel vytočení chodidla, který se mění rotací referenčního vektoru při řešení inverzní kinematiky. Výška zvedání kolena se dá měnit zvyšováním středu trajektorie švihové nohy. Obdobně jako při úpravě trajektorie kdy se noha nachází pod terénem. Je také možné změnit dobu trvání kroku.

Pro pohyb rukou je možné měnit amplitudy křivek pro rotaci v rameni a v lokti. Dále je možné měnit rozpřáhnutí rukou a maximální napnutí loktu. A konečně je možné měnit váhy pro kyčel, koleno a kotník při určování vhodné pozice pánve v dvojí opoře.

Pro nastavování parametrů slouží grafické uživatelské rozhraní. Pro jeho tvorbu je využita knihovna ImGui. Pro změnu parametrů slouží posuvníky. Pomocí tlačítka je možné vybrat některou z přednastavených chůzí. Dále je možné přecházet mezi různými modely a terény. Pomocí grafického uživatelského rozhraní jsou také zobrazeny zajímavé křivky, které se přizpůsobují nerovnému terénu. Na obrázku 6.19 je možné vidět výslednou aplikaci i s grafickým uživatelským rozhraním.



Obrázek 6.19: Výsledná aplikace s grafickým uživatelským rozhraním při chůzi po nerovném terénu.

6.5 Měření rychlosti

Aplikaci jsme testovali na notebooku Asus K53SV s Intel Core i5 procesorem a GeForce GT 540M grafickou kartou. Vykreslení jednoho snímku trvá průměrně 5ms. Tato doba je nezávislá na komplexnosti terénu. V průběhu jednoho kroku mají dva snímky výrazně delší dobu výpočtu – když se určuje trajektorie pánve a upravuje trajektorie nohy ve švihové fázi. Při nedostatku výpočetního výkonu může v těchto snímcích docházet k neplynulosti animace. V tabulce 6.1 je uvedeno kolik procent z celkového času zabírají některé části aplikace.

Pro porovnání podobný systém [12], který je také procedurální a řeší chůzi po nerovném terénu, je schopen vypočítat 3.75 kroků za vteřinu s 10 snímky na krok na procesoru Pentium-II 300 MHz. Tento systém neřeší horní polovinu těla a zobrazování trojrozměrného modelu.

Tabulka 6.1: Podíl částí řešení na celkovém času, který aplikace stráví na CPU. Části řešení se nevylučují.

Funkce	Procento celkového času
Vykreslování scény	25
Vykreslování GUI	35
Řešení inverzní kinematiky	5
Určení pozice pánve	10
Vyhodnocování křivek	3
Řešení rukou	0.6
Řešení nohou	5
Řešení páteře	0.08

Kapitola 7

Závěr

V této práci jsme se zabývali implementací animace lidské chůze s nastavitelnými parametry a možností chůze po nerovném terénu. Byla představena skeletální animace, metody tvorby animace a nastíněny možné přístupy ke tvorbě animace lidské chůze. Definovali jsme inverzní kinematiku a popsali některé metody jejich řešení. Dále byly popsány fáze lidské chůze a dvě existující řešení procedurální animace chůze. Následně bylo popsáno načtení a zobrazení modelu a samotná implementace animace. Nakonec byla změřena rychlosť aplikace.

Podařilo se nám za využití interpolačních křivek a inverzní kinematiky vytvořit realistickou animaci lidské chůze. K tomu jsme se inspirovali existujícími řešeními pro pohyb pánve a využili vlastní řešení pro pohyb horní poloviny těla a nohou. Vytvořená animace umožňuje chůzi po nerovném terénu, kde ale může při velkém sklonu terénu docházet k neprirozenému ohybu v kotníku. Výpočet animace je rychlý díky využití analytického řešení inverzní kinematiky. Pomocí parametrů chůze se dají nastavit velmi odlišné a specifické animace chůze.

Výsledky práce by mohly být použity při tvorbě počítačových her. V takovém případě by bylo vhodné implementovat animace iniciace a zastavení chůze a případně i animace běhu. Také by po implementaci exportu animace do souborového formátu bylo možné práci využít pro tvorbu animace klíčových snímků.

Literatura

- [1] Biomechanika chůze. [online].
URL http://ftk.upol.cz/fileadmin/user_upload/FTK-katedry/biomechanika/BIAP_Chuze.pdf
- [2] Blender Wiki: Skinning to Shapes. [online].
URL
<http://wiki.blender.org/index.php/Doc:2.4/Manual/Rigging/Skinning/ObData>
- [3] Vertex Skinning. [online].
URL <http://ruh.li/AnimationVertexSkinning.html>
- [4] Inverse Kinematics (two joints) for foot placement. [online], 2012.
URL http://www.gamasutra.com/view/news/129168/Inverse_Kinematics_two_joints_for_foot_placement.php
- [5] Venom joker rigged. [online], 2012.
URL <http://tf3dm.com/3d-model/venom-joker-rigged-36682.html>
- [6] A. Kharb, Y. K. J., V. Saini; Dhiman, S.: A review of gait cycle and its parameters. *Int. J. Comput. Eng. Manag.*, ročník 13, 2011: s. 78–83.
- [7] Aristidou, A.; Lasenby, J.: Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver. Technická Zpráva CUEDF-INFENG, TR-632, Department of Information Engineering, University of Cambridge, September 2009.
- [8] Bruderlin, A.; Calvert, T.: Interactive animation of personalized human locomotion. In *Proceedings of Graphics Interface '93*, Canadian Information Processing Society, Toronto, Ontario, Canada: Canadian Information Processing Society, 1993, str. 17–23.
- [9] Buss, S. R.: Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods. 2009.
- [10] Cain, A.: Physiopedia: Gait in prosthetic rehabilitation. [online].
URL http://www.physio-pedia.com/Gait_in_prosthetic_rehabilitation
- [11] Ceccato, J.-C.; De Sèze, M.; Azevedo, C.; aj.: Comparison of trunk activity during gait initiation and walking in humans. *PLoS One*, ročník 4, č. 12, 2009: str. e8193.
- [12] Chung, S.-k.; Hahn, J. K.: Animation of Human Walking in Virtual Environments. In *Proceedings of the Computer Animation, CA '99*, Washington, DC, USA: IEEE Computer Society, 1999, ISBN 0-7695-0167-2, s. 4–.

- [13] Dadashi, F.; Mariani, B.; Rochat, S.; aj.: Gait and Foot Clearance Parameters Obtained Using Shoe-Worn Inertial Sensors in a Large-Population Sample of Older Adults. *Sensors*, ročník 14, č. 1, 2014: str. 443, ISSN 1424-8220, doi:10.3390/s140100443.
- [14] Jacobson, A.; Deng, Z.; Kavan, L.; aj.: Skinning: Real-time Shape Deformation. In *ACM SIGGRAPH 2014 Courses*, 2014.
- [15] Kamermans, M.: A Primer on Bézier Curves. [online], 2011-2016.
URL <http://pomax.github.io/bezierinfo/>
- [16] Kavan, L.; Collins, S.; Zara, J.; aj.: Geometric Skinning with Approximate Dual Quaternion Blending. *ACM Trans. Graph.*, ročník 27, č. 4, 2008: str. 105.
- [17] Kirk, A.; O'Brien, J.; Forsyth, D.: Skeletal parameter estimation from optical motion capture data. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, ročník 2, June 2005, ISSN 1063-6919, s. 1185 vol. 2-, doi:10.1109/CVPR.2005.327.
- [18] Lin, Y.-C.; Gfoehler, M.; Pandy, M. G.: Quantitative evaluation of the major determinants of human gait. *Journal of biomechanics*, ročník 47, č. 6, 2014: s. 1324–1331.
- [19] Martz, P.: Generating Random Fractal Terrain. [online].
URL <http://www.gameprogrammer.com/fractal.html>
- [20] Multon, F.; France, L.; Cani-Gascuel, M.-P.; aj.: Computer Animation of Human Walking: a Survey. 1999.
- [21] Orendurff, M. S.; Segal, A. D.; Klute, G. K.; aj.: The effect of walking speed on center of mass displacement. *Journal of rehabilitation research and development*, ročník 41, č. 6A, 2004: str. 829.
- [22] Paul Chen: Male Base Mesh. [online], 2015.
URL <http://tf3dm.com/3d-model/male-base-mesh-6682.html>
- [23] Van Welbergen, H.; Van Basten, B. J. H.; Egges, A.; aj.: Real Time Animation of Virtual Humans: A Trade-off Between Naturalness and Control. *Computer Graphics Forum*, ročník 29, č. 8, 2010: s. 2530–2554, ISSN 1467-8659, doi:10.1111/j.1467-8659.2010.01822.x.
- [24] Welman, C.: *Inverse kinematics and geometric constraints for articulated figure manipulation*. Diplomová práce, Simon Fraser University, 1993.

Přílohy

Seznam příloh

A Obsah CD	45
B Manuál	46

Příloha A

Obsah CD

Adresář tex Obsahuje zdrojový tvar písemné zprávy v L^AT_EXu. Překlad je možný pomocí `$make`. Případně v Texmakeru pomocí pdflatex + bibtex + pdflatex + pdflatex. Adresář `fig` obsahuje obrázky použité v písemné zprávě.

`proceduralni_animace_lidske_chuze.pdf` Přeložená písemná zpráva ve formátu pdf.

Adresář human_walk Obsahuje solution pro Visual Studio 2013 se zdrojovými kódy aplikace. Ve Visual studiu se aplikace přeloží pomocí Build/Build Solution (F7) a spustí pomocí Debug/Start Debugging (F5). Aplikace vyžaduje podporu OpenGL 3.1. Zdrojové kódy jsou v podadresáři `human_walk/source`. V rámci práce nebyly vytvořeny soubory `imconfig`, `imgui`, `imgui_demo`, `imgui_draw`, `imgui_impl_sdl_gl3`, `imgui_internal`, `stb_rect_pack`, `stb_textedit`, `stb_truetype` a `lodepng`.

Adresář Release Obsahuje přeloženou aplikaci pro Windows spustitelnou pomocí `human_walk.exe` a další soubory potřebné pro její běh.

`video.mp4` Video s prezentací projektu. Dostupné také z <https://youtu.be/Z5Qckdr9yDI>

Příloha B

Manuál

Ovládání aplikace

L - vypnutí/zapnutí úrovně detailu
F - přepínání mezi flat a smooth stínováním
B - vypnutí/zapnutí skyboxu
T - vypnutí/zapnutí terénu
C - zapnutí/vypnutí zobrazení křivky definující dráhu
V - zapnutí/vypnutí zobrazení rozdelení křivky na segmenty
K - zapnutí/vypnutí barevného odlišení úrovní detailu
E - zapnutí/vypnutí zobrazení dráhy jako drátového modelu
N - spustí normovaný průchod
O - zapnutí/vypnutí zobrazení dráhy pomocí přechodů mezi úrovněmi. Od nejvyšší úrovně k nejnižší a zpět k nejvyšší, bez ohledu na pozici kamery.
↑ / ↓ - pohyb po dráze vpřed/vzad
WSAD - volný pohyb kamery
+/- - oddálení/přiblížení kamery
2/8 - rotace kamery okolo osy X
4/6 - rotace kamery okolo osy Y
SHIFT - Urychlí pohyb kamery při souběžném držení s klávesou pro pohyb kamery

Změna parametrů aplikace Parametry je možné měnit v souboru `params.txt`. Je nutné zadat všechny níže popsané parametry. Tvar je **parametr=hodnota** bez mezer.

SEED - seed pro generování náhodných čísel
MAP_SIZE - velikost mapy
MAP_VERTICES - počet vrcholů mapy
SPEED - rychlosť pohybu kamery po dráze
TRIANGLES_1 - počet trojúhelníků původního modelu
TRIANGLES_2/3/4 - počet trojúhelníků pro vytvoření nižších úrovní detailu
MAX_TRIANGLES - maximální počet trojúhelníků při použití úrovně detailu
INCREMENT_1/2/3 - inkrementy pro určení intervalů vzdáleností jednotlivých úrovní detailu
N - počet oblastí v jednom kvadrantu pro generování dráhy