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

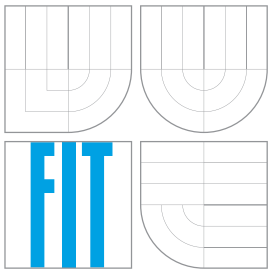
SEMESTRÁLNÍ PROJEKT
TERM PROJECT

AUTOR PRÁCE
AUTHOR

PETR MOHELNÍK

BRNO 2016

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

SEMESTRÁLNÍ PROJEKT
TERM PROJECT

AUTOR PRÁCE — PETR MOHELNÍK
AUTHOR

VEDOUCÍ PRÁCE — Ing. LUKÁŠ POLOK
SUPERVISOR

BRNO 2016

Abstrakt

Animace lidské chůze má velké využití v interaktivních aplikacích, převážně počítačových hrách. Existuje množství způsobů tvorby této animace. Tyto způsoby se liší v kompromisu mezi přirozeností, kontrolou nad animací a výpočetním časem. Tato práce se zabývá implementací procedurální animace. Dále jsou popsány metody řešení inverzní kinematiky a je rozebrána lidská chůze a její fáze.

Abstract

Animation of human walk is employed in many interactive applications, mostly computer games. There are many ways to create such animation. Techniques differ in compromise between naturalness, control over animation and computing time. This work implements procedural animation. It also describes methods to solve inverse kinematics problem and analyze human walk and its phases.

Klíčová slova

počítačová grafika, skeletální animace, lidská chůze, inverzní kinematika

Keywords

computer graphics, skeletal animation, human walk, inverse kinematics

Citace

Petr Mohelník: Procedurální animace lidské chůze, semestrální projekt, Brno, FIT VUT v Brně, 2016

Procedurální animace lidské chůze

Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením Ing. Lukáše Poloka,

.....
Petr Mohelník
2. ledna 2016

Poděkování

Zde je možné uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc.

Obsah

Kapitola 1

Úvod

Počítačová animace má velké využití pro počítačové hry a filmy. Existuje množství způsobů tvorby a zobrazování animace. Tato práce se zabývá tvorbou animace lidské chůze. Lidská chůze je jedna z nejpoužívanějších a nejdůležitějších animací. Lidé jsou velmi vnímaví na správnost lidské chůze a snadno rozpoznají, kdy je animace nepřirozená. To klade velký důraz na přesnost.

Existují různé přístupy k tvorbě animace lidské chůze. Tyto přístupy 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 kamer. Ačkoliv tento způsob dosahuje velké přirozenosti, je velmi drahý. Fyzikální simulace umožňuje velmi dobrou interakci s prostředím, ale je velmi výpočetně náročná. Tato práce se zabývá procedurální tvorbou. 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, ale není příliš vhodná pro interakci s prostředím. 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 použit 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. Také obsahuje popis některých existujících implementací procedurální animace využívající tyto znalosti.

Kapitola 2

Skeletální animace

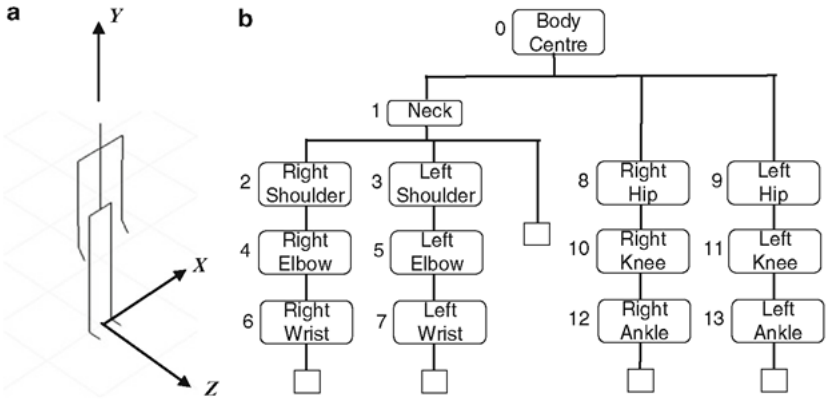
Skeletální animace je způsob animace, který využívá dvě základní komponenty: model (nejčastěji polygonální) nazýván kůže a kostru, na kterou je kůže připojena. Jestliže se pohne kost, pohne se také připojená kůže. V klasické animaci, je vytvořen model pro každý klíčový snímek animace a tyto modely se za 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 model. Skeletální animace má také své nevýhody. Pohyby kostí a připojených vrcholů spotřebovávají výpočetní výkon. V animacích obecně se musí řešit interpolace mezi dvěma klíčovými snímky, 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. 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 připojeného modelu (kůže), musí se vrcholy transformovat ze souřadného systému modelu

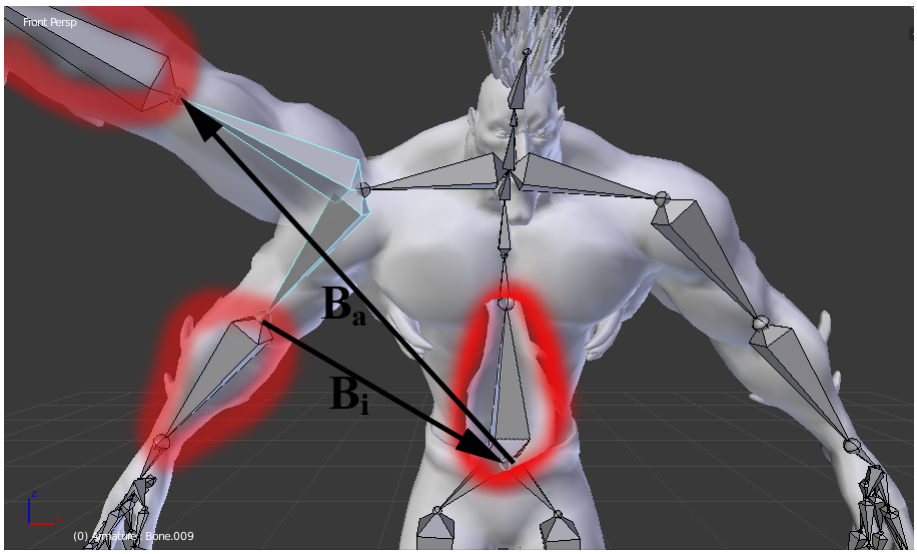


Obrázek 2.1: Příklad hierarchie kostry

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

$$\mathbf{T} = \mathbf{B}_i\mathbf{B}_a \tag{2.1}$$

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.



Obrázek 2.2: Transformace předloktí. Prvně se aplikuje inverzní bind pose matice a poté aktuální transformace kosti.

2.2 Skinning

Skinning [9] 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 dobře 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 je modernější algoritmus. Každému vrcholu umožňuje přiřazení více kostí. Každá dvojice kost a vrchol má přiřazenou 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 $\mathbf{T}_1, \ldots, \mathbf{T}_m$ 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 $w_{i,1}, \ldots, w_{i,m} \in \mathbb{R}$. Každá váha $W_{i,j}$ určuje vliv kosti j na vrchol i. Celkový součet všech vah je 1, přičemž žádná váha není záporná.

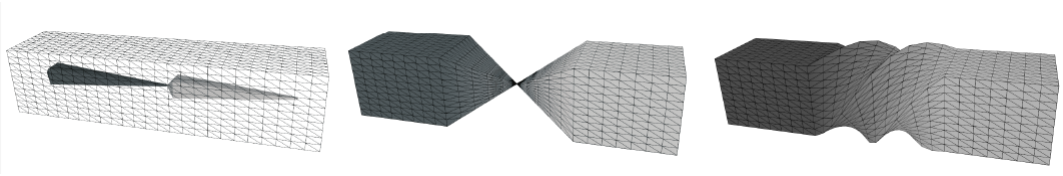
Množství kostí ovlivňující jeden vrchol se může významně lišit, např. od 1 do 10. Kvůli limitacím grafického hardwaru se často uvažuje že jich není více než 4. Transformovaná pozice vrcholu $\mathbf{v}'_i$ se určí podle následující rovnice:

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

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.

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

$\mathbf{R}_1$ je maticí identity a $\mathbf{R}_2$ je rotace okolo osy Z o 180°. 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



Obrázek 2.3: Zleva doprava: model v bind pose, linear blend skinning a dual quaternion. Převzato z [9].

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.

Velmi rozšířenou metodou je nelineární metoda Dual quaternion skinning [10], kde se místo matic míchají duální kvaterniony. Tato metoda řeší artefakty vznikající u linear blend skinning, zachovává objem modelu 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 matice.

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 vyhlazují váhy a tento proces se opakuje, dokud výsledek není dostatečně přirozený.

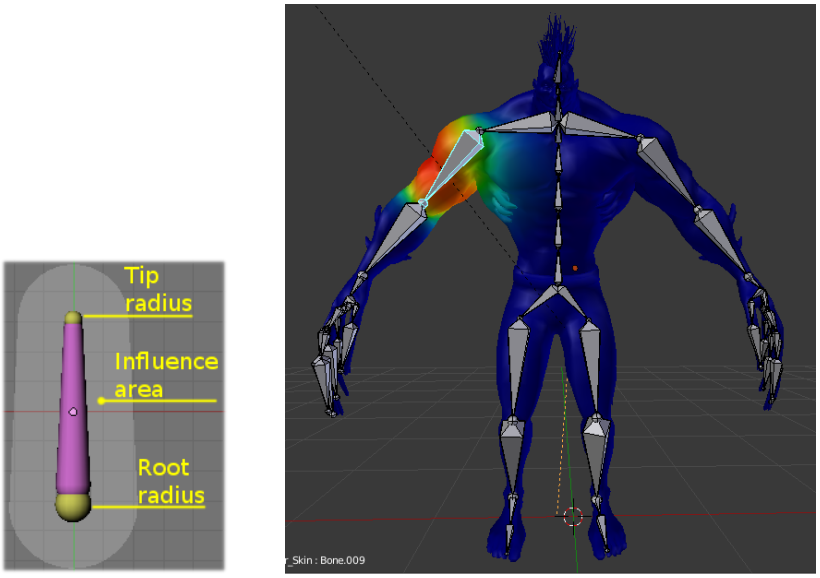
2.3 Tvorba animace

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 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. [13] dělí techniky tvorby animace do tří skupin: procedurální simulace, fyzikální 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 fyzikální simulace. Nebo pomocí procedurální simulace vytvořit gesta rukou a fyzikální simulací zachovat fyzikálně přirozené balancování spodní části těla.



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

2.3.1 Procedurální simulace

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

Procedurální simulace 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 matematických formulí. Také je nutné explicitně zachovávat fyzikální přirozenost v procedurálním modelu pohybu pro všechny možnosti parametrů. Je často používána pro animaci gest a mluvení, které vyžadují velké množství parametrů.

2.3.2 Fyzikální simulace

V mnoha případech je vhodné počítat s dynamikami [12], 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 fyzikální simulace. Dynamiky jsou založené na Newtonových pohybových zákonech. Spojují síly (resp. točivé momenty rotací) do vý-

sledného pohybu (resp. rotace) podle rovnice:

$$f = m \cdot \ddot{x} \tag{2.3}$$

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:

$$t = i \cdot \ddot{\theta} + \dot{\theta} \times i \cdot \ddot{\theta} \tag{2.4}$$

kde t je točivý moment, i je matice setrvačnosti, $\dot{\theta}$ je úhlová rychlost 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 fyzikální 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 aproximace pomoci 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 kontroléru je minimalizace nesrovnalostí 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.

Fyzikální 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.

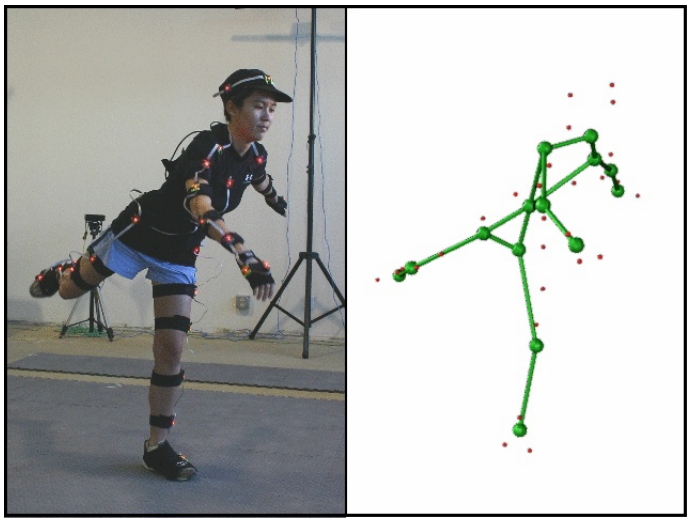
2.3.3 Editace pohybu

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 [11] 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 herec okolo každého kloubu značky, které jsou zaznamenávány a sledovány kamerami. Výstupem je množina lokací bodů v čase. Těmi se poté proloží kostra.



Obrázek 2.5: Automatická rekonstukce kostry podle pozic značek. Převzato z [11].

Kapitola 3

Kinematiky

Kinematiky v počítačové grafice slouží pro manipulaci s kostrou. Jsou rozděleny na dvě části: přímá a inverzní kinematika. V přímé kinematice 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 pozice kostí od ruky. V této technice je obtížné omezovat pohyb, například že chodidlo nemůže proniknout do země.

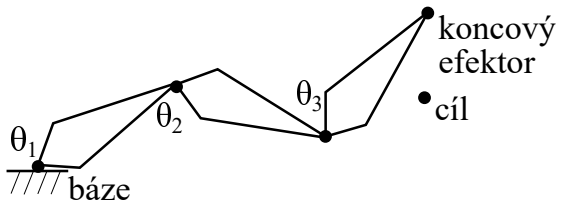
Máme-li řetězec kloubů, kde každý kloub i má přiřazen úhel θ_i, který určuje jeho rotaci oproti jeho předkovi, viz obrázek 3.1. Potom stav řetězce je určen stavovým vektorem $\boldsymbol{\theta}(\theta_1, \ldots, \theta_n)$. Označme pozici koncového efektoru $\mathbf{P}$. Přímá kinematika řeší následující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

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

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 kinematice [7][14] se hýbe pouze s koncovým efektorem a pohyb ostatních kloubů se dopočítá, tak aby byl tento pohyb umožněn. To je matematicky zapsáno následovně:

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



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 ze 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, které se pro řešení použijí 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° a zavřeno méně než cca 30°. Inverzní kinematika poskytuje lepší kontrolu nad koncovým efektorem, 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 [7][14][5] je iterativní metoda využívající Jacobiho matice k aproximaci řešení. Vztah mezi kartézským prostorem koncového efektoru $\mathbf{P}$ a prostoru kloubů úhlů $\boldsymbol{\theta}$ je:

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

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

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

kde m je počet dimenzí pozice koncového efektoru $\mathbf{P}$, n je počet dimenzí $\boldsymbol{\theta}$, $i = 1, \ldots, m$ a $j = 1, \ldots, n$. 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:

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

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

$$\boldsymbol{\theta} = \boldsymbol{\theta} + \Delta\boldsymbol{\theta} \tag{3.6}$$

Změna v pozici koncového efektoru určená touto změnou je:

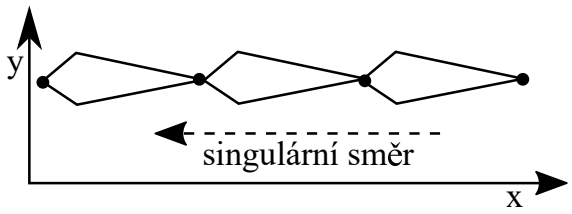
$$\Delta\vec{\mathbf{s}} \approx J\Delta\boldsymbol{\theta} \tag{3.7}$$

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

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 $J^{\dagger}$, vetšinou 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

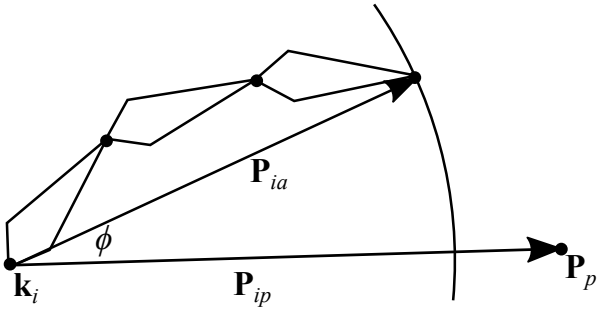
Cyclic Coordinate Descent (CCD) [14] je řešení inverzní kinematiky založené na iterativním heuristickém vyhledávání, které se snaží o minimalizaci rotační chyby zpracováváním jednoho kloubu v jeden okamžik. Každá iterace provede jednotlivě pohyb všech článků od nejposlednějšího k prvnímu. To se opakuje 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 používající 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 nejblíže 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:

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

kde k_1, k_2 a k_3 jsou konstantní koeficienty, pro detaily viz [14]. Funkce má maximum v intervalu $-\pi \leq \phi \leq \pi$, když je první derivace nula a druhá derivace je záporná. Z první podmínky dostáváme rovnici:

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



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 [4] 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:

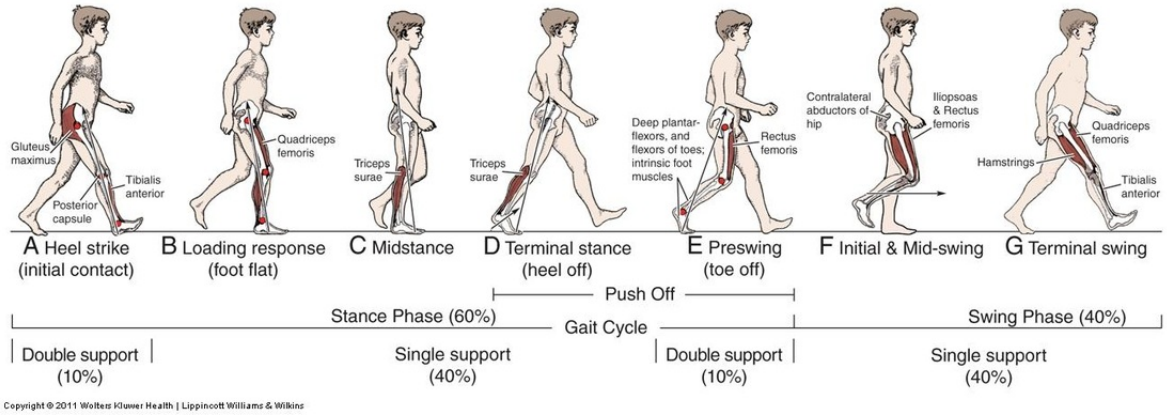
- stadium zatěžování
- mezistoj
- konečný stoj
- předšvihová fáze

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 rychlostí 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:

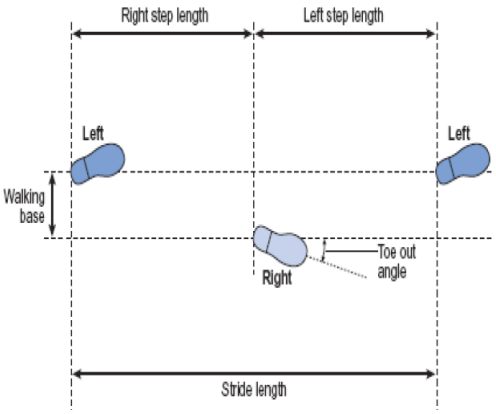
- počáteční švih
- mezišvih
- konečný švih

Nyní si popíšeme některé termíny a parametry specifikující chůzi. 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 před ní. Šířka dvojkroku je vzdálenost mezi přímkami určenými středy pat obou nohou při chůzi. Úhel



Obrázek 4.1: Krokový cyklus a jeho fáze. Převzato z [1].

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. Rychlost chůze je vzdálenost ušlá v čase. Okamžitá rychlost je variabilní v průběhu kroku, ale průměrná rychlost je výsledkem kadence a délky dvojkroku. Rychlost chůze r se spočítá jako $r = d/c$, kde d je délka dvojkroku. Rychlost chůze tedy závisí na délce kroku, které závisí na délce švihové fáze.



Obrázek 4.2: Délka, šířka kroku a úhel vytočení chodidla. Převzato z [4].

4.1 Fáze krokového cyklu

Následuje podrobnější popis jednotlivých fází krokového cyklu [4].

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 stadiu zatěžování.

4.1.2 Stadium zatěžování

Stadium zatěžování je fáze, kde je váha těla přenesena na přední nohu. Koleno je pokrčeno pro absorbci 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) intervalem 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 mezistoje. 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.

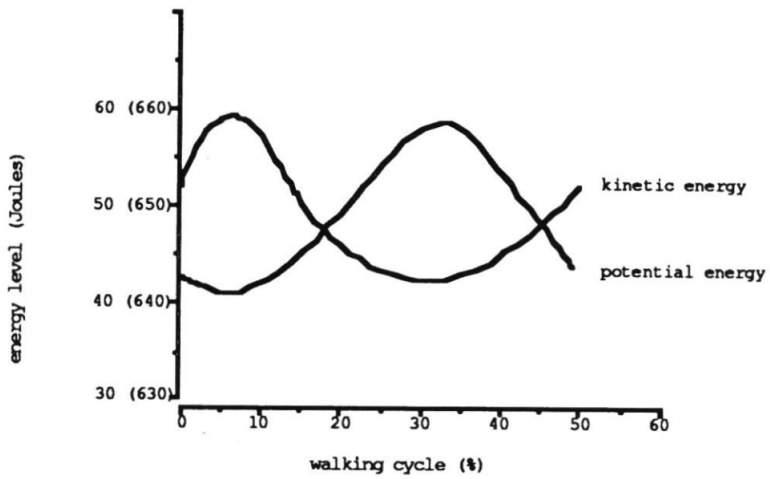
4.1.8 Konečný švih

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.

4.2 Interactive Animation of Personalized Human Locomotion

V článku Interactive Animation of Personalized Human Locomotion [6] je popsán systém pro tvorbu lidského pohybu pomocí procedurální animace. Umožňuje specifikaci tří parametrů: délka kroku, kadence kroků a rychlost. A 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 korelaci s výdajem energie během jednoho dvojkroku, jak je vidět na obrázku 4.3, 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.



Obrázek 4.3: Aproximace kinetických a potenciálních energií horní poloviny těla pro průměrný krok. Převzato z [6].

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.

4.3 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 [8]. Používá se model člověka s 18 klouby a celkem 36 stupni volnosti.

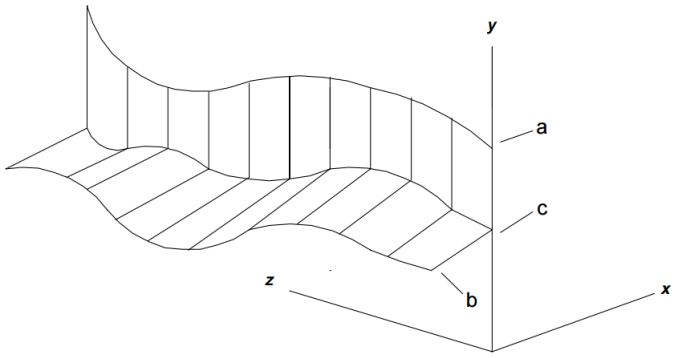
Kontrola pohybu na nejvyšší úrovni umožňuje uživateli animovat pohyby s malým množstvím parametrů. Uživatel specifikuje cestu pohybu pomocí kubických splinů v horizontální rovině. Z parametrů výšky člověka a rychlosti chůze je vygenerována délka a frekvence kroku. Pro normální chůzi může být vztah určen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

$$l = \sqrt{0.004 \times s \times h} \tag{4.1}$$

kde l je délka kroku, s je rychlost 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. Přesáhne-li rotace chodidla prahovou hodnotu, je krok zkrácen, ale doba provedení kroku zůstane zachována. Stav terénů ve 3D je zkontrolován 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 pánve jsou pomocí inverzí kinematiky dopočítány úhly kolene a kyčle. 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, pánve.

Observace ukázaly, že trajektorie pánve přibližně odpovídá sinusoidě, viz obrázek 4.4. Pro interpolaci jsou použity kubické spliny, konkrétně Beziérova křivka. Pánev prochází vertikálním maximem uprostřed stojné fáze, tato hodnota se určí podle velikosti ohybu 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 chodidla nohy ve švihové fázi se použijí Beziérovy křivky obdobně jako při pohybu pánve. Spolu se znalostí pozice kořene (pánve) 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 pánve.



Obrázek 4.4: 3D trajektorie (c) je určena z vertikálního (a) a horizontálního (b) posuvu. Převzato z [8].

Kapitola 5

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, umožňující 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 umožňuje načtení předpřipraveného modelu s připojenou kostrou. Případně uživatel může využít poskytnutou předpřipravenou kostru a na ní si připojit vlastní model. A ten poté vyexportovat v Blenderu do formátu COLLADA (.dae). Aplikace model zobrazí za využití Linear blend skinning (kapitola 2.2.1) algoritmu. Uživatel má možnost zobrazení pouze kůže modelu, nebo kostry, nebo obojí zároveň. Model je animován. Je možnost pomocí několika parametrů upravit animaci chůze. K tomu slouží grafické uživatelské rozhraní. Animace může být vyexportována a následně použita například v Unity enginu.

5.0.1 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 5.1). Název materiálu je nastavený v Blenderu a při načítání modelu se očekává přiložení textury se shodným názvem a příponou "_D"pro difuzní texturu.

```
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 prirazenych kosti
  std::vector<glm::ivec4> jointIndices; //prirazene kosti
public:
  /* ... */ };
```

Kód 5.1: Třída modelu.

Následuje načtení vah a přiřazení kostí vrcholům z XML elementu `library_controllers`. První se přečte bind shape matice. 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 kloubů, které ho ovlivňují a stejný počet odkazů do seznamu kloubů 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 kostí. Kosti jsou hierarchicky zanořené, obsahují název a lokální transformační matici, viz kód 5.3. 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 obrázek kód 5.2. Kostem se následně přidají inverzní bind pose matice 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. Dále jsme zavedli možnost přidat kost, která se použije pouze pro určení velikosti a poté se odstraní, taková kost musí ve svém názvu obsahovat "empty".

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

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

Kód 5.2: Třída kostry.

Nakonec jsou naplněny objekty třídy `WeightedMesh` (kód 5.1). Protože z hardwarových důvodů je vhodné mít pro každý vrchol kůže přiřazeny pouze čtyři kosti a jejich váhy, abychom mohly použít `vec4` ve vertex shaderu. Musí být některé kosti zanedbány, pokud jich vrchol používá více. Čtyři kosti jsou dostatečné a pokud je jich více, mají některé velmi nízké váhy. Proto jsou pro každý vrchol všechny jeho kosti seřazeny sestupně podle vah a

ukládají se pouze první čtyři. Váhy ostatních jsou rozloženy rovnoměrně mezi tyto čtyři kosti, aby zůstala zachována podmínka o celkovém součtu všech vah (viz 2.2.1).

```
<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 5.3: Ukázka uložení hierarchie kostry ve formátu COLLADA.

5.0.2 Zobrazení

Před každým zobrazením snímku je nutné spočítat globální 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 **T** podle rovnice 2.1. Vezmeme kořenovou transformační matici a nastavíme jí jako model matici do vertex shaderu. Následně ve vertex shaderu získáme pozici vrcholu násobením maticí **T** a odpovídající vahou pro čtyři přiřazené kosti. Přepočítáváme také normálu, viz kód 5.4.

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

Kód 5.4: 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`.

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 model matice kosti je kořenová transformace vynásobená s touto globální maticí kosti. Před zobrazením je vyčištěn hloubkový buffer, aby se kostra překreslila přes model.

Pro zobrazení jsou použity difuzní textury a Phongův osvětlovací model. Výsledný výstup viz obrázek 5.1.

Obrázek 5.1: Grafický výstup aplikace pro dva různé modely. Model vpravo má rotované některé kosti (ručně, uvnitř kódu).

Kapitola 6

Závěr

Bylo implementováno načtení modelu člověka spolu s připojenou kostrou a jeho zobrazení. Výsledná aplikace má … řádků kódu. Byly popsány některé existující implementace procedurální animace a metody řešení inverzní kinematiky. V rámci dalšího vývoje bude potřeba implementovat systém tvorby animace a její vyexportování do souboru.

Literatura

[1] Gait in prosthetic rehabilitation. http://www.physio-pedia.com/Gait_in_prosthetic_rehabilitation.

[2] Skinning to Shapes. http://wiki.blender.org/index.php/Doc:2.4/Manual/Rigging/Skinning/ObData.

[3] Vertex Skinning. http://ruh.li/AnimationVertexSkinning.html.

[4] 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.

[5] 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.

[6] 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.

[7] Buss, S. R.: Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods. 2009.

[8] 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–.
URL http://dl.acm.org/citation.cfm?id=791217.791557

[9] Jacobson, A.; Deng, Z.; Kavan, L.; aj.: Skinning: Real-time Shape Deformation. In *ACM SIGGRAPH 2014 Courses*, 2014.

[10] Kavan, L.; Collins, S.; Zara, J.; aj.: Geometric Skinning with Approximate Dual Quaternion Blending. *ACM Trans. Graph.*, ročník 27, č. 4, 2008: str. 105.

[11] 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.

[12] Multon, F.; France, L.; Cani-Gascuel, M.-P.; aj.: Computer Animation of Human Walking: a Survey. 1999.

[13] 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.
URL http://dx.doi.org/10.1111/j.1467-8659.2010.01822.x

[14] Welman, C.: *Inverse kinematics and geometric constraints for articulated figure manipulation*. Diplomová práce, Simon Frasor University, 1993.