

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

AUTHOR

PETR MOHELNÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. LUKÁŠ POLOK

BRNO 2016

Abstrakt

Výtah (abstrakt) práce v českém jazyce.

Abstract

Výtah (abstrakt) práce v anglickém jazyce.

Klíčová slova

skeletální animace, lidská chůze, inverzní kinematika

Keywords

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 tuto bakalářskou práci vypracoval samostatně pod vedením pana ...

.....

Petr Mohelník
23. prosince 2015

Poděkování

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

© 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
2.4	Kinematiky	8
2.4.1	Jacobiho inverzní metoda	9
2.4.2	Cyclic Coordinate Descent	10
2.5	Lidská chůze	11
2.5.1	Interactive Animation of Personalized Human Locomotion	12
2.5.2	Animation of Human Walking in Virtual Environments	13
3	Závěr	15

Kapitola 1

Úvod

Počítačová animace má velké využití pro počítačové hry a filmy. Lidé jsou velmi vnímaví na správnost lidské chůze a snadno rozpoznají, kdy je animace nepřírozená. Proto je vylepšování tvorby této animace stále aktuálním tématem. Lidská chůze je jedna z nejpoužívanějších a nejdůležitějších animací. Existují různé přístupy k tvorbě animace lidské chůze. Tyto přístupy se liší v kompromisu mezi přirozeností a kontrolou nad animací. Tato práce se zabývá procedurální tvorbou. Procedurální systém umožňuje jednoduše vytvořit animaci pomocí specifikace několika parametrů.

Kapitola 2

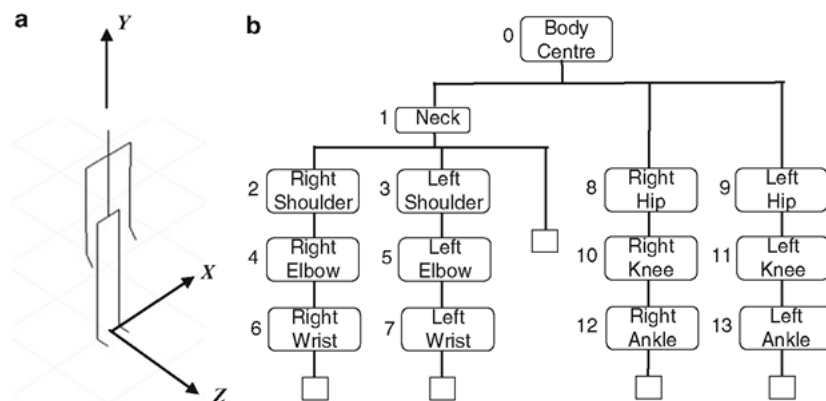
Skeletální animace

Skeletální animace je způsob animace, který využívá dvě základní komponenty: model (nejčastěji polygonální) nazýván kůže a kostru, na kterou je kůže připojena. Pohyb kosti způsobí pohyb připojené kůže. V klasické animaci, je vytvořen mesh pro každý klíčový snímek animace a tyto meshe se za sebou 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. Ty mohou být předpočítané. 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 nebo zvířecího těla. Jedná se o stromovou strukturu. Uzly jsou nazývány kosti nebo klouby. Ke kostem jsou přiřazeny vrcholy. Každá kost má 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 se většinou používá pouze rotace. Výsledná globální transformace kosti \mathbf{B} se získá kombinací rodičovské globální transformace a vlastní lokální transformace. To se provede procházením řetězce přímé kinematiky, kde se postupně aplikují transformace od kořene k listům. Pro kořenovou kost je lokální transformace její globální.

Na začátku se kostra nastaví do bind pose [1]. To je základní pozice kostry před jakýmkoliv pohybem, na kterou je připojena kůže. Když je kostra připojena na kůži, jsou určeny bind pose matice \mathbf{B} pro každou kost. Ty určují jejich transformaci z počátku souřadného systému kostry. Chceme-li změnit pózu kostry (matici \mathbf{B} nějaké kosti) a tím i připojeného modelu, musí se vrcholy transformovat ze souřadného systému modelu do souřadného systému kostry. K tomu se použije inverzní bind pose matice \mathbf{B}_i . Ta určuje transformaci kosti, tak aby byla umístěna v počátku. Tedy invertuje transformaci, která byla na model

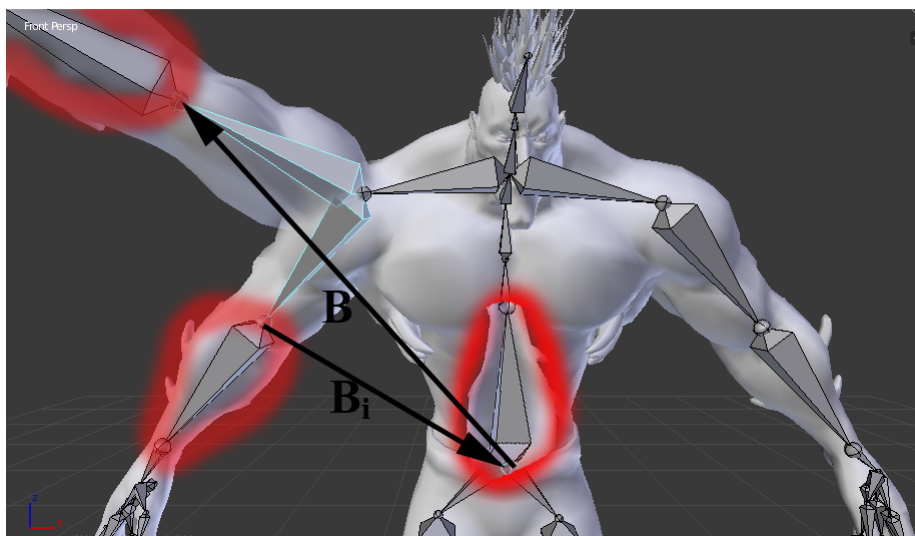


Obrázek 2.1: Příklad hierarchie kostry

aplikována v bind pose. Matematický zápis transformace \mathbf{T} do nové pozice je následující:

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

Poté se může použít nová transformace kosti. 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 inverse bind pose a poté nová bind pose

2.2 Skinning

Skinning [2] je proces přidělování kostí vrcholům a určování jejich vah. Tento proces může probíhat algoritmicky nebo ručně v některém z modelovacích programů.

Nejjednodušší způsob přiřazení modelu kostře je každé kostře přiřadit samostatný pevný

objekt jako je např. válec. Na každý takový objekt se poté aplikuje transformace odpovídající kosti. Tento přístup není vhodný pro realistické animování.

Simple skinning je základní metoda, která již používá kůži, tedy jeden model pro celou kostru. Přiřazuje jeden vrchol k jedné kosti. Každý vrchol se poté transformuje transformací odpovídající kosti. Je používána například ve starších hrách. Je použitelná pro modely s nízkým počtem trojúhelníků. U detailnějších modelů způsobuje nepěkné deformace v kloubech a animace vypadá nepřírodně.

Linear blend skinning je pokročilejší algoritmus. Každému vrcholu se může přiřadit více kostí. Každá kost ovlivňuje vrchol různou silou, vahou. Váha určuje jak velký vliv má pohyb kosti na výslednou pozici vrcholu. Očekává na vstupu následující data.

- Rest pose shape, typicky reprezentovaný jako polygonální model. Propojení meshe se v průběhu nemění, pouze pozice vrcholů.
- Transformace kostí, reprezentované jako matice $\mathbf{T}_1, \dots, \mathbf{T}_m$ určené podle rovnice 2.1.
- 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.

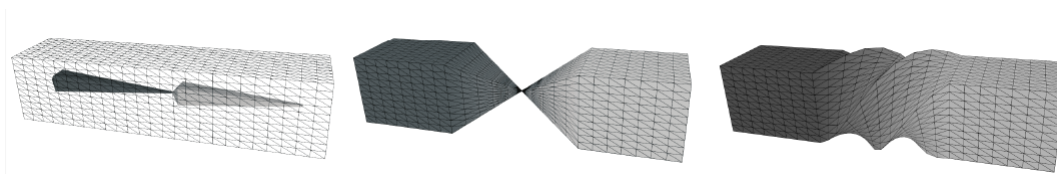
Množství kostí ovlivňující jeden vrchol se může významně lišit, např. od 1 do 10. Kvůli limitacím grafického hardwaru se často uvažuje že jich není více než 4. Transformovaná pozice vrcholu \mathbf{v}'_i se určí podle následující rovnice:

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

Tato metoda podává dobré výsledky, pokud míchané transformace nejsou příliš rozdílné. Problémy nastávají jestliže potřebujeme míchat transformace, které se významně liší v jejich rotaci. Je známým faktem, že 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 problémem. Uvažujme ale tyto dvě rotace.

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

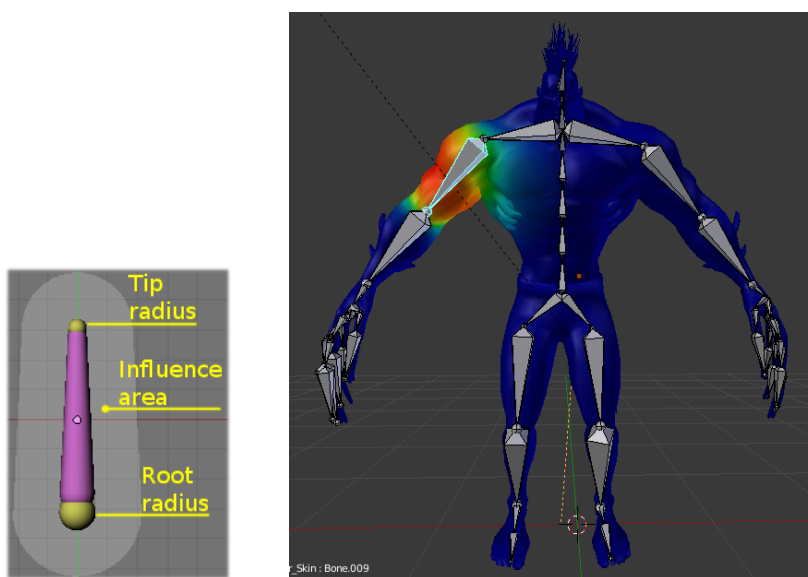
\mathbf{R}_1 je maticí identity a \mathbf{R}_2 je rotace okolo osy Z o 180° . Lineární míchání $0.5 \cdot \mathbf{R}_1 + 0.5 \cdot \mathbf{R}_2$ nám dá za výsledek matici hodnoty 1, která promítá 3D prostor na osu Z. Dojde ke ztrátě objemu modelu. Velké relativní rotace nejsou vzácné, protože klouby jako ramena a zápěstí mají velký rozsah pohybu. Tyto problémy je možné řešit zavedením více parametrů než má lineární blend skinning. Takové metody se nazývají multilineární. Tyto metody nemusí být žádoucí kvůli nutnosti vytvářet a ukládat více vah pro každý vrchol. Velmi rozšířenou metodou je nelineární metoda Dual quaternion skinning [9], kde se místo matic míchají duální kvaterniony. Tato metoda řeší artefakty vznikající u LBD, 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 dual kvaterniony toto omezení nemají - osa může být libovolná. Kromě toho duální kvaterniony ještě umožňují popsat posunutí. Ale neumožňují popsat změnu měřítka. Proto nemůže docházet ke ztrátě objemu,



Obrázek 2.3: Zleva doprava: model v rest pose, linear blend skinning a dual quaternion

jako u matic, kde násobením může vznikat zmenšení. Dual kvaterniony jsou také výhodné pro GPU hardware, protože obsahují méně hodnot než matice.

Přiřazení vah je možné automaticky metodou obálek [3]. 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, jak se vzdaluje k okraji této oblasti. Pro kvalitní model je nutné nastavit váhy ručně, nakreslit. Například v Blenderu je "Weight Paint" mode, 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 konkrétní kost. Poté se zkouší efekt na známé póze, upravují a vyhlazují váhy a tento proces se opakuje, dokud výsledek není dostatečně kvalitní.



Obrázek 2.4: Metoda obálek v programu Blender a model ve Weight paint mode

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 dvě techniky popisu kinematické informace: inverzní a přímá kinematika, viz 2.4.

Welbergen a spol. [12] dělí techniky tvorby animace na tři základní: procedurální simulace, fyzikální simulace a editace pohybu.

Procedurální simulace popisuje animaci algoritmicky pomocí matematických formulí. 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). Výstupy těchto metod se ovlivňují změnami hodnot parametrů, díky čemuž je možné upravit výstup na míru požadavků. Jsou dobré pro adaptaci prostředí a jsou také málo výpočetně náročné. Nedosahují takové přirozenosti jako např. editace pohybu nebo fyzikálně simulované. Umožňují přesné časování a pozicování kostí a mohou využívat velké množství parametrů. Nicméně je obtížné popsat pohyby matematicky. Musí se explicitně zachovávat přirozenost v procedurálním modelu pohybu pro všechny možnosti parametrů.

V mnoha případech je vhodné počítat s dynamikami [11], 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 přístupů založených na fyzikální simulaci. Dynamiky jsou založené na Newtonových pohybových zákonech. Spojují síly (resp. točivé momenty rotací) do výsledného pohybu (rotace) podle rovnice:

$$f = m \cdot x'' \quad (2.3)$$

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

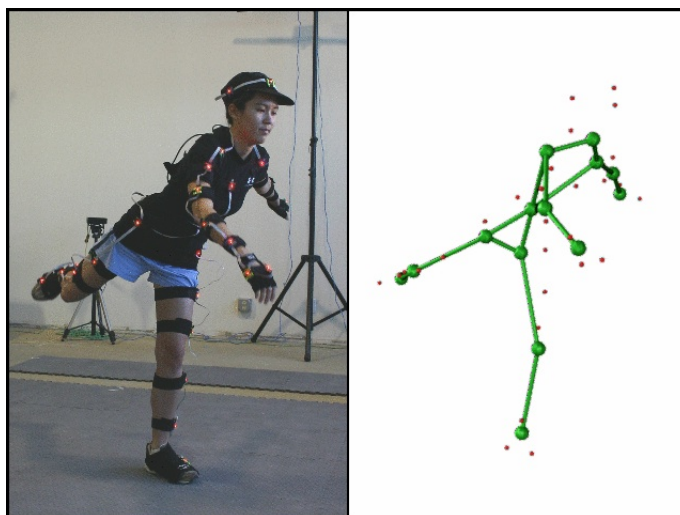
$$t = i \cdot \theta'' + \theta' \times i \cdot \theta' \quad (2.4)$$

kde t je točivý moment, i je matice setrvačnosti, θ' je úhlová rychlost a θ'' je úhlové zrychlení. Přímá(?) dynamika je aplikace těchto zákonů k výpočtu pohybu vygenerovaného danou silou. Inverzní dynamika jsou metody určující sílu, která by vygenerovala daný pohyb.

Animace založená na fyzikální simulaci aplikuje na kosti točivý moment a vypočítává pohyb pomocí přímé kinematiky. Přiřazuje objektům fyzikální vlastnosti jako např. hmotnost. U tohoto typu animace je typicky model tvořen systémem pevných těles. Jsou velmi dobré na adaptování se prostředí, ale jsou výpočetně náročné. Metody mohou fungovat na základě zadávání omezení pohybů kostí jako parametrů animace, kde se spočítají pohyby pomocí kinematik a poté se zkontroluje pomocí dynamik jejich vhodnost. Nebo zadávání požadovaného cílového stavu animace.

Metody editace pohybu vytváří pohyb na základě vstupu několika vzorů pohybu. Vstupní data mohou být získána pomocí např. motion capture, ale mohou být tvořena i animátorem ručně. V Motion capture [10] má herec okolo každého kloubu značku, které jsou zaznamenávány a sledovány. Z pozic a úhlů mezi těmito značkami se určí pohyb, který se poté aplikuje na kostru modelu. Ačkoliv tyto techniky vypadají nejvíce přirozeně, jsou vázány na specifický kontext. Zachovávají přirozenost vstupních vzorů pohybu, ale pouze jsou-li změny malé. Techniky používající více vzorů na jeden pohyb zachovávají přirozenost

i při větších změnách. Počet vzorů roste exponenciálně s počtem parametrů animace. Neposkytují fyzikální interakci s prostředím. Vhodné pro vytváření předem pro neinteraktivní aplikace jako filmy. Pro interaktivní, např. hry je potřeba velká databáze pohybů. Pohyby se mohou vytvářet pomocí technik ze zpracování signálů, interpolací mezi vzory pohybů nebo pomocí statistických modelů.



Obrázek 2.5: Rekonstrukce kostry podle pozic značek. Převzato z [10]

Techniky se také mohou kombinovat dohromady a využívat vhodnější techniku pro danou situaci. Může se například použít technika editace pohybu dokud nedojde k interakci s prostředím a ta se může zpracovat pomocí fyzikální simulace.

2.4 Kinematiky

Kinematiky jsou v počítačové grafice většinou rozděleny na dvě části: přímá a inverzní kinematika. Přímá kinematika je metoda, kde 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, jako 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 2.6. Potom stav řetězce je určen stavovým vektorem $\theta(\theta_1, \dots, \theta_n)$. Označme pozici koncového efektoru \mathbf{P} . Přímá kinematika řeší následující rovnici:

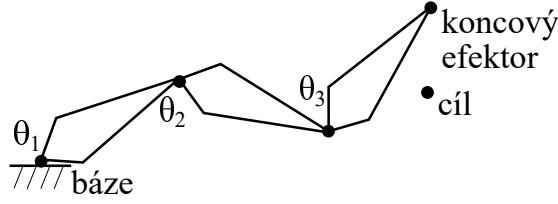
$$\mathbf{P} = f(\theta) \quad (2.5)$$

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][4] se hýbe pouze s koncovým efektozem a pohyb ostatních kloubů se dopočítá, tak aby byl tento pohyb umožněn. To je matematicky zapsáno

následovně:

$$\theta = f^{-1}(\mathbf{P}) \quad (2.6)$$



Obrázek 2.6: Příklad řetězce tří kloubů

Řešení inverzní kinematiky není tak jednoduché jako přímé. Funkce f je nelineární a ačkoliv v rovnici 2.5 existuje unikátní namapování z θ na \mathbf{P} , pro inverzní mapování v 2.6 může být nekonečné množství θ pro jedno konkrétní \mathbf{P} . Analytické řešení pro libovolný řetězec kloubů neexistuje. Problém se řeší pomocí numerických metod pro řešení systémů nelineárních rovnic. 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 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 používající kinematiku používají empirickou a biomechanickou znalost pohybu pro generování animace. Nyní si ukážeme některé způsoby řešení inverzní kinematiky.

2.4.1 Jacobiho inverzní metoda

Jacobiho inverzní metoda [7][4][6] je iterativní metoda využívající Jacobiho matice k aproximaci řešení. Vztah mezi kartézským prostorem koncového efektoru \mathbf{P} a prostoru kloubů úhlů θ je:

$$\dot{\mathbf{P}} = J(\theta)\dot{\theta} \quad (2.7)$$

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

$$J(\theta) = \left(\frac{\partial P_i}{\partial \theta_j} \right)_{i,j} \quad (2.8)$$

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

$$\dot{\theta} = J^{-1}(\theta)\dot{\mathbf{P}} \quad (2.9)$$

Poté co určíme Jacobiho matici, hledáme hodnotu $\Delta\theta$ pro inkrementaci θ :

$$\theta = \theta + \Delta\theta \quad (2.10)$$

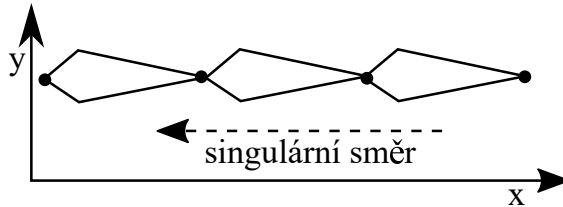
Změna v pozici koncového efektoru určená touto změnou je:

$$\Delta\vec{s} \approx J\Delta\theta \quad (2.11)$$

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

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 2.7. 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.



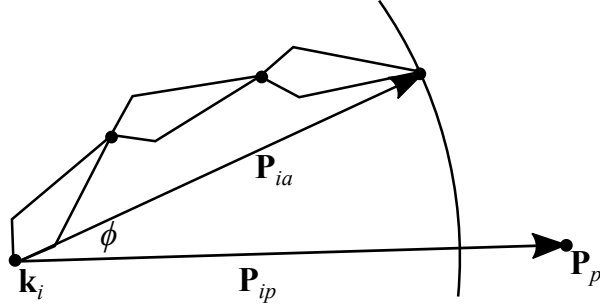
Obrázek 2.7: Singulární konfigurace řetězce kloubů

2.4.2 Cyclic Coordinate Descent

Cyclic Coordinate Descent (CCD) [4] je technika založená na iterativním heuristickém vyhledávání, která se snaží o minimalizaci rotační chyby zpracováváním jednoho kloubu v jeden okamžik. Každá iterace provede pohyb nejposlednějšího článku a postupuje přes celý řetězec a to se opakuje dokud se nedosáhlo požadované pozice a nebo se nepřesáhl maximální počet cyklů. K určení úhlu rotace se používají dva vektory. Protože pracuje s jedním kloubem v jeden okamžik, zvýhodňuje klouby na začátku. Oproti Jakobianským metodám provádí více iterací. Ačkoliv je algoritmus jednoduchý, je často nutné ho dále přizpůsobovat, aby bylo dosaženo vhodných výsledků.

Mějme pozici kloubu \mathbf{k}_i pro iteraci a kloub i . 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} okolo osy o o úhel ϕ , dostaneme nový vektor $\mathbf{P}'_{ia}(\phi)$. Jestliže úhel bude nabývat libovolných hodnot, výsledné pozice $\mathbf{P}'_{ia}(\phi)$ začnou 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} zarovnaly. Úhel ϕ který hledáme, je úhel, který maximalizuje



Obrázek 2.8: Ukázka jedné iterace metody CCD.

následující funkce:

$$g(\phi) = k_1(1 - \cos \phi) + k_2 \cos \phi + k_3 \sin \phi \quad (2.12)$$

kde k_1 , k_2 a k_3 jsou konstantní koeficienty, pro detaily viz [4]. Funkce má maximum v intervalu $-\pi \leq \phi \leq \pi$, když je první derivace nula a druhá derivace je záporná. Z první podmínky dostáváme:

$$\phi = \tan^{-1} \frac{k_3}{k_2 - k_1} \quad (2.13)$$

kteřá 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. Když jsme získali úhel ϕ můžeme ho vynásobit vahou určující tuhost kloubu a přičteme ho k aktuální rotaci r_i kloubu i . Poté se pokračuje s iterací $i + 1$.

2.5 Lidská chůze

Lidská chůze se skládá z opakujících se krokových cyklů, neboli dvojkroků. 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 levé nohy nastává perioda dvojí opory, která končí při odrazu palce pravé nohy. Následuje jedno-oporová fáze do té doby, než se pravá dostane do počátečního kontaktu. Nastává druhá perioda dvojí opory, až do odrazu palce levé 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 švih. 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

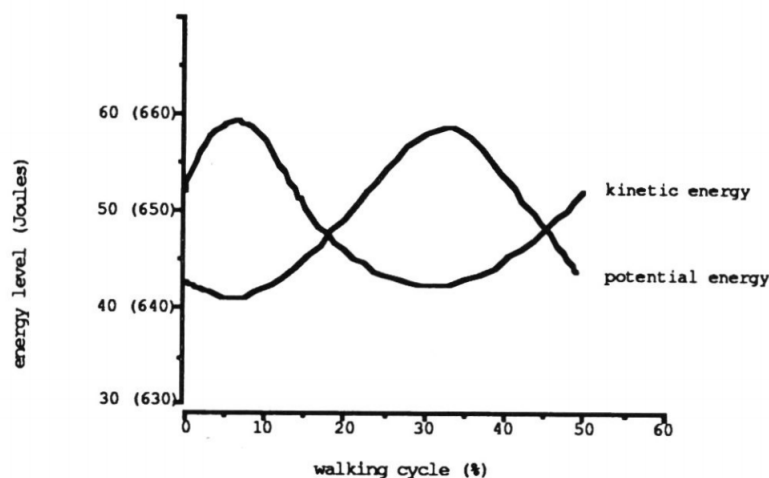
Nyní si popíšeme některé termíny a parametry specifikující chůzi. Délka dvojkroku je vzdálenost mezi dvěma po sobě jdoucími kroky stejné nohy. Skládá se ze dvou délek kroku, levé a pravé, každá s nich určuje vzdálenost o kterou se jedna noha přesune před druhou. Může být např. nulová, když se levá noha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

2.5.1 Interactive Animation of Personalized Human Locomotion

V článku Interactive Animation of Personalized Human Locomotion [5] je popsán systém pro tvorbu lidského pohybu pomocí procedurální animace. Umožňuje specifikaci tří parametrů: délka kroku, frekvence 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.

Pohyb je určený dvěma symetrickými kroky (levým a pravým), které dohromady tvoří jeden stride, který se cyklicky opakuje. Proto se můžeme zabývat pouze jedním krokem. Pro každý krok jsou vypočítány tři omezení ze specifikovaných parametrů a atributů. Doba trvání fází kroku (stání a švih), úhly nohy na konci kroku a kontrolní body určující pohyb kyčle stojící 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 fáze, kde jsou obě nohy na zemi a nejvyšší uprostřed švih. Zatímco horizontální přesun má maximum a minimum opačně. To je v korelaci s výdajem energie během jedné stride, jak je vidět na obrázku 2.9, 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 stojící nohu během kroku. Pozice prstů nohy je statická během fáze stání. Úhly zbylých kloubů nohy jsou dopočítány pomocí inverzní kinematiky.



Obrázek 2.9: Aproximace kinetických a potenciálních energií horní poloviny těla pro průměrný krok. Převzato z [5]

Tím je určen pohyb stojící nohy. Dále se určí rotace, náklon pánve a lateral displacement!! natočení těla do boku. Rotace pánve je na maximum při dopadu paty na zem a na minimum jsou-li obě nohy souběžně. Náklon pánve má minimum a maximum opačně. Náklon těla do boku je způsobem tím, že se tělo vždy nakloní na stranu nohy nesoucí váhu. Je minimální při dopadu paty a maximální uprostřed stání. 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.

2.5.2 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 [8]. Používá se model člověka s 18 klouby a celkem 36 stupni volnosti.

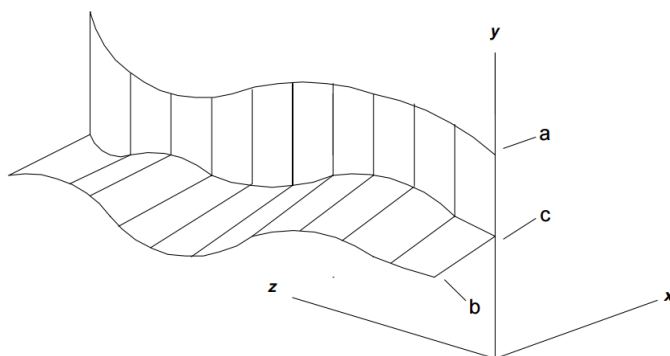
Kontrola pohybu na nejvyšší úrovni umožňuje uživateli animovat pohyby s malým množstvím parametrů. Uživatel specifikuje cestu pohybu pomocí kubických splinů v horizontální rovině. Z parametrů výšky člověka a rychlosti chůze je vygenerována délka a frekvence kroku. Pro normální chůzi může být vztah určen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

$$l = \sqrt{0.004 \times s \times h} \quad (2.14)$$

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 těmito úhly. 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 2.10. Pro interpolaci jsou použity kubické spliny, konkrétně Beziérova křivka. Pánev prochází vertikálním maximem uprostřed fáze podpírající nohy, tato hodnota se určí podle velikosti ohybu kolene. Pánev prochází minimem uprostřed fáze podpory obou nohou, která se nachází uprostřed mezi dvěma fázemi podpírající nohy. Pozice v této fázi je určena tak, aby byla minimalizována suma úhlových zrychlení všech kloubů podpírající nohy. Uživatel může těmito kloubům nastavovat váhy. Výsledná Beziérova křivka je poté nerovnoměrně vzorkována, aby bylo dosaženo nejrychlejšího pohybu během fáze podpory obou nohou a nejpomalejšího uprostřed fáze podpírající nohy. Pro definici trajektorie chodidla nohy ve



Obrázek 2.10: 3D trajektorie(c) je určena z vertikálního (a) a horizontálního (b) posuvu. Převzato z [8]

švihů se použijí Beziérovky 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.

Kapitola 3

Závěr

Literatura

- [1] <http://ruh.li/AnimationVertexSkinning.html>.
- [2] <http://skinning.org/direct-methods.pdf>.
- [3] <http://wiki.blender.org/index.php/Doc:2.4/Manual/Rigging/Skinning/ObData>.
- [4] http://graphics.ucsd.edu/courses/cse169_w04/welman.pdf.
- [5] <http://graphicsinterface.org/wp-content/uploads/gi1993-3.pdf>.
- [6] 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.
- [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] Kavan, L.; Collins, S.; Zara, J.; aj.: Geometric Skinning with Approximate Dual Quaternion Blending. *ACM Trans. Graph.*, ročník 27, č. 4, 2008: str. 105.
- [10] 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.
- [11] Multon, F.; France, L.; Cani-Gascuel, M.-P.; aj.: Computer Animation of Human Walking: a Survey. 1999.
- [12] 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>