

## Information technologies and multimedia Informacinės technologijos ir multimedija

# PROCEDŪRINĖS ANIMACIJOS TECHNOLOGIJŲ TAIKYMAS HUMANOIDINIAMS ŽAIDIMŲ AGENTAMS

Kristina BUILISOVA <sup>\*</sup>

*Vilniaus Gedimino technikos universitetas, Vilnius, Lietuva*

Gauta 2023 m. balandžio 11 d.; priimta 2023 m. birželio 1 d.

**Santrauka.** Straipsnyje analizuojami atvirkštinės kinematikos algoritmai ir pasirinkto algoritmo taikymas „Unity“ žaidimo variklyje. Pastebėta, kad „Unity“ variklio siūlomi sprendikliai neturi funkcionalumo, reikalingo atvaizduoti anatomiškai teisingą humanoidinio agento judesį. Išrinkti ir palyginti keli atvirkštinės kinematikos algoritmai, tokie kaip *CCD*, *FABRIK*, ir trianguliacijos algoritmas – dėl kelių galinių vykdymo įtaisų palaikymo ir optimizacijos būdų išrinktas algoritmas *FABRIK* tolesniam užduočių vykdymui ir sprendiklio „FABRIK IK“ kūrimui. Palyginus sprendiklių funkcionalumo lygį, galima teigti, kad funkcionaliai „FABRIK IK“ sprendiklis yra pranašus „Two Bone IK“ ir „Chain IK“ sprendikliams analizuotose srityse. Pamatavus našumą kadravimo dažniu, matoma, jog šių atvirkštinės kinematikos sprendiklių taikymas „Unity“ aplinkoje neišreiškia didelio skirtumo (nuo –8,29 % iki 5,89 % skirtumo). Demonstracinis ėjimo ciklas, taikantis „FABRIK IK“, veikia demonstracinėje aplinkoje ir juda anatomiškai ir fiziškai tiksliau nei ėjimo ciklas, kuriam pritaikytas „Chain IK“ sprendiklis; judėjimo anatominis tikslumas sutampa su „Two Bone IK“ implementacijos demonstracija.

**Reikšminiai žodžiai:** procedūrinė animacija, „Unity“ žaidimų variklis, atvirkštinės kinematikos algoritmas, sprendiklis, *FABRIK*, humanoidinis personažas.

## Įvadas

Šiais laikais kompiuterinių žaidimų industrija yra didelė ir besiplečianti pramogų rinkos dalis. Kompiuteriniai žaidimai daro įtaką ne tik informatikos mokslams, bet ir mokslams, susijusiems su pramogų industrija, kuri tampa vis didesne modernaus gyvenimo dalimi (Seller, 2019; Aliaga et al., 2016; Bhatti, 2019). Žmogaus atvaizdas matomas ir dvimatis, ir trimatis erdves vaizduojančiose medijose. Žmogaus judesio animacija yra sudėtinga, todėl ji reikalauja nemažai resursų: laiko, technologinės pažangos, patyrusių specialistų, finansų. Kadangi šių laikų technologijos siūlo animuojamų personažų modeliams suteikti „kaulų“ armatūrą (angl. *armature* arba *skeleton*) (Wampler, 2016), procedūrinė animacija tampa sprendimu, kaip tokius personažus animuoti greičiau ir pigiau. Taip pat procedūrinė animacija leidžia pasitelkti atvirkštinės kinematikos (angl. *inverse kinematics*) technologijas, siekiant leisti kompiuterinio žaidimo personažui tiksliai ir tiesiogiai reaguoti į aplinkos ypatybes (nelygią žemę, atsitiktinius objektus personažo judėjimo trajektorijoje, pastatus ir kitas kliūtis). Atvirkštinės kinematikos uždaviniai yra sudėtingi uždaviniai, kurie turi sprendimus – atvirkš-

tinės kinematikos algoritmus (Erleben & Andrews, 2019; Kenwright, 2018). Atvirkštinė kinematika yra viena dažniausiai pasitaikančių procedūrinės animacijos metodikų kompiuteriniuose žaidimuose, kuriuose dalyvauja humanoidiniai personažai, sąveikaujantys su aplinka.

Šiame darbe bus nagrinėjama procedūrinė animacija ir jos pritaikymas humanoidiniams žaidimų agentams, įgyvendinant atvirkštinės kinematikos sprendimą pagal pasirinktą algoritmą. „Unity“ variklio siūlomi sprendikliai neturi viso funkcionalumo, reikalingo išpildyti anatomiškai taisyklingą humanoidinio agento judesį. Šio darbo tikslas yra pagal pasirinktą atvirkštinės kinematikos algoritmą sukurti „Unity“ žaidimų varikliui pritaikytą sprendiklių humanoidiniam agentui, kuris galėtų taikyti atvirkštinės kinematikos algoritmą ir suteikti galimybę naudotojui pritaikyti funkcijas, padedančias lengviau animuoti humanoidinius agentus „Unity“ žaidimo variklyje.

## 1. Atvirkštinės kinematikos technologijos

Kinematika – tai mokslas, apibrėžiantis judančius objektus neįskaitant juos veikiančių jėgų ar judesio priežasčių (Stenlund, 2019; Aristidou et al., 2018). Kompiuterinėje grafikoje

\*Autorius susirašinėti. El. paštas [kristina.builisova@stud.vilniustech.lt](mailto:kristina.builisova@stud.vilniustech.lt)

kinematika naudojama animacijoje, ir yra apibrėžiami du pagrindiniai metodai – tiesioginė kinematika (angl. *forward kinematics*) ir atvirkštinė kinematika (angl. *inverse kinematics*). Abu šie metodai yra susiję tuom, jog abu leidžia judinti ir deformuoti trimačius modelius su skeletais, ir abu šie metodai taikomi trimačių modelių animacijoje. Užduotims atlikti reikalinga išanalizuoti atvirkštinės kinematikos komponentą procedūrinėje animacijoje. Šio komponento pritaikymo rezultatas – skeleto kaulų grandys, kurios gali prisitaikyti prie 3D modelių supančios aplinkos arba aplinkoje esančių objektų; galūnių taškai yra apskaičiuojami pirmiausia, o pagal juos apskaičiuojama galutinė kūno pozicija pagal fizikinį modelį (Bhatti et al., 2013). Kompiuteriniuose žaidimuose tai matoma kaip humanoidinių ir nehumanoidinių personažų galūnių tikslus kontaktas su supančios aplinkos objektais: kojos realiuoju laiku prisitaiko prie sudėtingų grindų, žemės arba laiptų, rankos atpažįsta griebiamo objekto formą ir tiksliai ją apglėbia, laisvai krentantis kūno modelis su armatūra atsitrenkia į aplinkos objektus tiksliai be prasiskverbimų (angl. *clipping*) ar sklindymų ore. Atvirkštinė kinematika yra aktuali ne tik kompiuterinių žaidimų srityje, bet ir robotikos srityje (Aristidou & Lasenby, 2009). Kadangi sričių yra daugiau nei viena, algoritmai, siekiantys išspręsti atvirkštinės kinematikos uždavinį, yra įvairialypiai. Pagal Muller-Cajar ir Mukundan, dauguma algoritmų taiko skaitines ir iteruojančias procedūras, nes jomis naudojantis galima išvengti uždaros formos išraiškų (tai išraiškos, kurios išreiškiamos naudojant baigtinį skaičių standartinių operacijų) (Muller-Cajar & Mukundan, 2007); kiti algoritmai naudoja Jakobio matricas ir jomis manipuliuoja. Šis metodas reikalauja daug resursų, ypač didelėms matricoms.

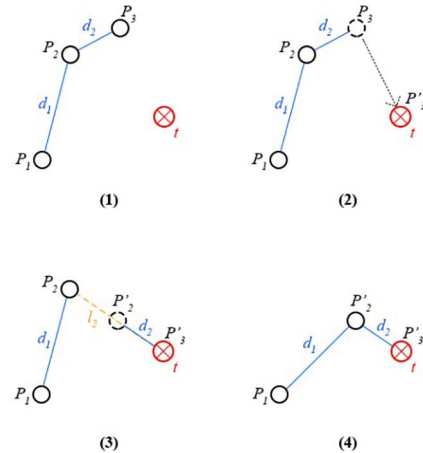
*FABRIK* – tai heuristinės metodologijos algoritmas, kurio veikimo principas – tai prieš tai apskaičiuotų sąnarių pozicijų naudojimas naujiems pasukimams skaičiuoti iteruojant pirmyn ir atgal kinematine grandimi (Aristidou & Lasenby, 2009). Vienas iš skirtumų nuo kitų algoritmų – tai, jog kiekvieno sąnario pasisukimas apskaičiuojamas atskirai. Algoritmo veikimui vaizduoti pateiktas pavyzdys pagal Aristidou ir Lasenby: pateikiamos sąnarių pozicijos, įvardintos  $P_1, \dots, P_n$ , čia  $P_1$  – pradinis sąnarys,  $P_n$  – galinis vykdymo įtaisas (Aristidou & Lasenby, 2011). Taikinyms simbolizuojamas  $t$ , pradinė grandies pozicija –  $b$ . Pirma apskaičiuojami atstumai tarp kiekvieno sąnario:

$$d_i = |P_{i+1} - P_i|, \quad i = 1, \dots, n - 1. \quad (1)$$

Tuomet atliekamas patikrinimas, ar taikinyms gali būti pasiekiamas matuojant atstumą nuo pradinio sąnario iki taikinio. Jei atstumas nuo pirmo sąnario iki taikinio yra mažesnis nei atstumų tarp sąnarių suma (kinematinė grandis pakankamai ilga pasiekti taikinį), pradedamos pilnos dvi iteracijos stadijos. Pirmoje stadijoje algoritmas nuspėja kiekvieno sąnario poziciją pradedant nuo galinio vykdymo įtaiso ir baigiant grandies pradžia  $P_1$ . Tada galinis vykdymo įtaisas įgauna tą pačią poziciją kaip ir taikinyms:

$$P'_n = t. \quad (2)$$

Kitas sąnarys,  $P_{n-1}$ , paslenkamas galinio vykdymo įtaiso link išlaikant buvusį atstumą tarp  $P_n$  ir  $P_{n-1}$ . Ta pati operacija taikoma kiekvienam sąnariui, kol pasiekiamas pirmasis sąnarys  $P_1$ . Baigus šiai iteracijai, galinis vykdymo įtaisas pasislenka arčiau taikinio ir susiję sąnariai pasislenka atitinkamai. Iteruojama daugiau kartų, kol galinis vykdymo įtaisas bus pakankamai arti taikinio, tik antra iteracija vykdoma atvirkščia tvarka – nuo pradinio sąnario  $P_1$  iki galinio vykdymo įtaiso  $P_n$ . Tai yra vienas išskirtinių *FABRIK* algoritmo bruožų – iteracijos vyksta pirmyn ir atgal kinematine grandine (1 pav.).



1 paveikslas. *FABRIK* algoritmo veikimo principas  
Figure 1. The backward reaching stage of the *FABRIK* algorithm

## 2. Procedūrinės animacijos metodika „Unity“ variklyje

Procedūrinė animacija „Unity“ variklyje gali būti vykdoma C# kodo pagrindu – kadangi šio tipo animacija yra paremta procedūrinio kodu, naudotojams yra suteikiama galimybė programuoti norimą procedūrinę animaciją variklio aplinkoje. Šiam tikslui įgyvendinti turi būti apibrėžiama nemažai savybių variklio objektui: norint atvaizduoti ir judinti objektą scenoje, naudotojas apibrėžia objektui grafinį vaizdavimą, suteikia transformaciją grafinio vaizdavimo slinkimui, rotacijai bei mastelio kitimui, suteikia fizikines savybes ir kita.

### 2.1. „Animation Rigging“ paketo atvirkštinės kinematikos sprendikliai

„Animation Rigging“ paketas yra „Unity Technologies“ bendrovės tiekiamas sprendimas nuo 2018 metų. Šis paketas yra papildantis bazinį variklį įrankiais, kurie padeda apdoroti ir modifikuoti modelius, turinčius skeletą. Pagal dokumentaciją išskiriami keturi branduoliniai paketo naudojami įrankiai: animatorius, skeleto įrangos (angl. *rig*) statytojas, skeleto įranga ir apribojimai.

Tarp skeleto įrangos apribojimų aptinkami du atvirkštinės kinematikos taikymo pavyzdžiai – „Two Bone IK“ ir „Chain IK“. Šie du sprendimai yra vykdomi

modifikuojamu C# kodo pagrindu ir manipuliuoja „GameObject“ objektais scenoje.

### 2.1.1. „Two Bone IK“ ypatumai

Šis būdas skirtas kinematinėms grandims, sudarytoms iš ne daugiau kaip trijų sąnarių. Tipiškai naudojamas, kai animuojama grandis susidaro tik iš trijų sąnarių. „Two Bone IK“ palaiko orientacijos taikinį ir grandies taikinį. Komponente būtina nurodyti visus tris sąnarius, atitinkamai pavadintus „root“, „mid“ ir „tip“. Taip pat pateikiamos parinkty taikiniui bei orientacijos taikiniui.

### 2.1.2. „Chain IK“ ypatumai

Ši metodika palaiko kinematinę grandį, sudarytą iš neapibrėžto skaičiaus sąnarių (Unity Technologies, 2020). „ChainIK“ naudoja FABRIK algoritmą. Pateikiami „root“ ir „tip“ sąnariai, kuriuos būtina nurodyti. Atvirkščiai, nei „Two Bone IK“, „Chain IK“ nepriima orientacijos taikinio, bet turi panašias parinktis taikiniui.

### 2.1.3. Kuriamo „FABRIK IK“ sprendiklio ypatumai

Sudaromi punktai, atspindintys situacijas, kurias turėtų gebėti spręsti būsimas sprendiklis:

- Nustatyti grandies savybes nurodant dalyvaujančius sąnarius bei apribojantį sąnarį ir vykdymo įtaisą bet kurioje grandies dalyje.
- Nustatyti grandies vykdymo įtaiso taikinį.
- Nustatyti grandies orientacijos taikinį ir priklausomumo laipsnį nuo orientacijos taikinio kiekvienam sąnariui.
- Nurodyti grandies kryptį.
- Leisti pėdoms prisitaikyti prie vaikšto paviršiaus.

## 3. Darbo rezultato testavimas

Sudaromas testavimo planas šiam darbui:

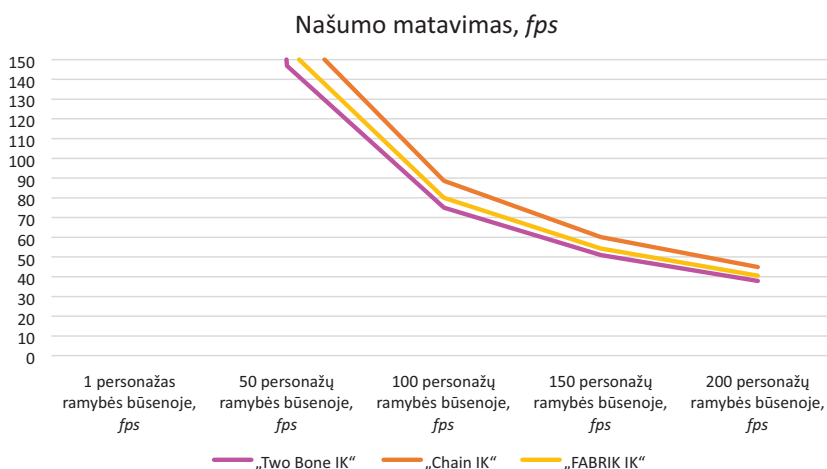
- Veikimo našumo palyginimas pagal kadravimo dažnį tarp „Unity“ siūlomų sprendimų „Two Bone IK“, „Chain IK“ ir sukurto FABRIK sprendiklio.

1 lentelė. Funkcionalumo palyginimas  
Table 1. Comparison of functionality between the chosen solvers

	„Two Bone IK“	„Chain IK“	„FABRIK IK“
Priklausomybė nuo „Animation Rigging“ paketo	Taip	Taip	Ne
Papildomas įrengimas be sprendiklio komponento įrangos	Taip	Taip	Ne
Neribotas kiekis sąnarių grandyje	Ne	Taip	Taip
Taikinio nustatymas	Taip	Taip	Taip
Orientacijos taikinių nustatymas	Dalinis	Ne	Taip
Galinio vykdymo įtaiso sąnario paskyrimas	Ne	Ne	Taip
Grandies krypties paskyrimas	Ne	Ne	Taip
Pėdų adaptavimas ant sudėtingų paviršių	Taip	Taip	Taip
Priklausomybės laipsnio slankikliai	Taip	Taip	Ne

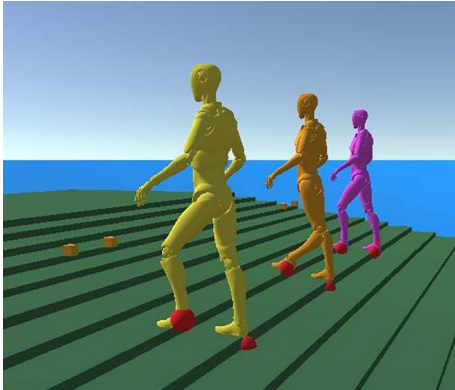
- Palyginime sukurti scenoje vieną, penkiasdešimt, šimtą, šimtą penkiasdešimt bei du šimtus humanoidų modelių su pritaikytais sprendimais ir sukurti ciklinę animaciją.
  - Parodyti funkcinis panašumus ir pranašumus tarp „Unity“ siūlomų sprendimų ir sukurto sprendiklio.
  - Parodyti, kaip humanoidiniai personažai su taikytais trimis sprendimais juda sudėtingoje erdvėje.
  - Erdvė susideda iš plokštumos, laiptų ir įkalmės.
- Pagal funkcionalumo įvertinimą matomi rezultatai 1 lentelėje.

Pagal rezultatus matyti 2 pav., jog „Two Bone IK“ ir „Chain IK“ sprendimai nesudaro didelio skirtumo nuo „FABRIK IK“ sprendiklio – skirtumas yra 5,89 % tarp „Two Bone IK“ ir „FABRIK IK“, ir 8,29 % tarp „FABRIK IK“ ir „Chain IK“.

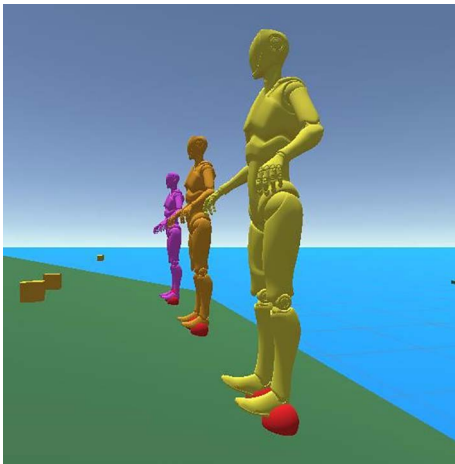


2 paveikslas. Našumo palyginimas pagal kadravimo dažnį  
Figure 2. Performance measurements according to framerate in a Unity scene

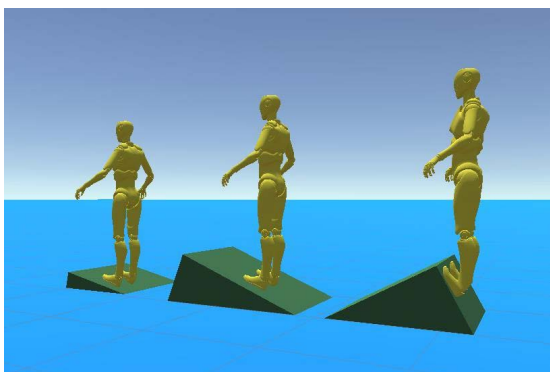
Šie trys paviršiai rodo, kaip atvirkštinės kinematikos sprendiklis, pritaikytas ant humanoidinio personažo, veikia sudėtingoje aplinkoje, kai ne tik grandis ir galinio vykdymo įtaisai turi prisitaikyti prie aplinkos, bet ir turi vykti atitinkamos modelio transformacijos (kaip modelio geometrijos transformacijos) (3, 4, 5 ir 6 pav.).



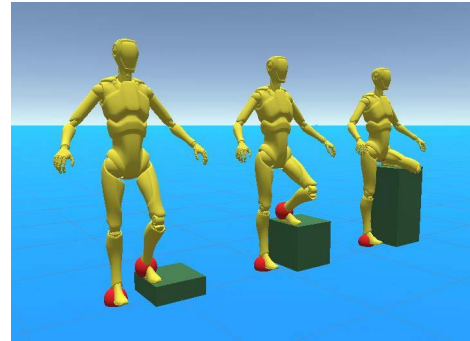
3 paveikslas. Ėjimo laiptais demonstracija  
Figure 3. Demonstration of walking up stairs



4 paveikslas. Ėjimo įkalne iliustracija  
Figure 4. Demonstration of walking uphill



5 paveikslas. Pėdų pasukimo kampo lūžio vizualizacija  
Figure 5. Visualization of ankle rotation according to the surface the model is standing on



6 paveikslas. Kojos susislinkimas į save esant per aukštai platformai

Figure 6. Visualization of a limb collapsing when the target is too close to the humanoid model

## Išvados

- Atlikus analizę nustatyta, kad procedūrinė animacija palaiko atvirkštinės kinematikos algoritmus ir yra lankstesnė nei raktinių kadro ar hibridinės animacijos metodikos.
- Atlikus „Unity“ variklio siūlomų atvirkštinės kinematikos sprendiklių analizę nustatyta, kad jie turi aiškius apribojimus: naudotojas privalo rinktis tarp sprendiklio, leidžiančio naudoti vieną orientacijos taikinį, bet ribojančio grandies ilgį, arba sprendiklio, leidžiančio surinkti bet kokio ilgio grandį, bet neleidžiančio taikyti orientacijos taikinio.
- Atlikus egzistuojančių atvirkštinės kinematikos algoritmų analizę pritaikymui ir praplėtimui pasirinktas FABRIK algoritmas, nes jis gali būti taikomas realiojo laiko programose ir palaiko bet kurio ilgio grandis.
- „FABRIK IK“ sprendiklis patobulintas su galimybe taikyti orientacijos taikinius kiekvienam grandies sąnariui ir nustatyti orientacijos taikinio priklausomybės laipsnį. Taip pat kūrimo metu pritaikyta galimybė nustatyti bet kurį grandies sąnarį kaip galinį vykdymo įtaisą.
- Našumas tarp „FABRIK IK“ ir „Two Bone IK“ su „Chain IK“ nežymiai skiriasi. „FABRIK IK“ elgesys testavimo scenarijuose atrodo anomiškai taisyklingai. „FABRIK IK“ sprendiklis pranoksta abu „Two Bone IK“ ir „Chain IK“ sprendiklius funkcinėmis galimybėmis.

## Literatūra

- Aliaga, D. G., Demir, I., Benes, B., & Wand, M. (2016, July 24). Inverse procedural modeling of 3D models for virtual worlds. In *ACM SIGGRAPH 2016 Courses*. <https://doi.org/10.1145/2897826.2927323>
- Aristidou, A., Lasenby, J., Chrysanthou, Y., & Shamir, A. (2018). Inverse kinematics techniques in computer graphics: A survey. *Computer Graphics Forum*, 37(6), 35–58. <https://doi.org/10.1111/cgf.13310>
- Aristidou, A., & Lasenby, J. (2009). *Inverse Kinematics: A review of existing techniques and introduction of a new fast iterative solver*. <http://andreasaristidou.com/publications/papers/CUEDF-INFENG,%20TR-632.pdf>



- Aristidou, A., & Lasenby, J. (2011). FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models*, 73(5), 243–260. <https://doi.org/10.1016/j.gmod.2011.05.003>
- Bhatti, Z. (2019). Oscillator driven central pattern generator (CPG) system for procedural animation of quadruped locomotion. *Multimedia Tools and Applications*, 78(21), 30485–30502. <https://doi.org/10.1007/s11042-019-7641-1>
- Bhatti, Z., Shah, A., Karabasi, M., & Mahesar, W. (2013). Expression driven trigonometric based procedural animation of quadrupeds. In *2013 International Conference on Informatics and Creative Multimedia* (pp. 104–109). IEEE. <https://doi.org/10.1109/ICICM.2013.25>
- Erleben, K., & Andrews, S. (2019). Solving inverse kinematics using exact Hessian matrices. *Computers & Graphics*, 78, 1–11. <https://doi.org/10.1016/j.cag.2018.10.012>
- Kenwright, B. (2018, December 4). Neural network in combination with a differential evolutionary training algorithm for addressing ambiguous articulated inverse kinematic problems. In *SIGGRAPH Asia 2018 Technical Briefs, SA 2018* (pp. 1–4). <https://doi.org/10.1145/3283254.3283262>
- Muller-Cajar, R., & Mukundan, R. (2007, December 5–7). Triangulation - A new algorithm for inverse kinematics. In *Image and Vision Computing New Zealand (IVCNZ) 2007 Conference* (pp. 181–186). <http://ir.canterbury.ac.nz/handle/10092/743>
- Seller, M. (2019). *Performing animation, animating performance*. <https://necs.org/node/117209>
- Stenlund, P. (2019). *Simulating realistic ragdoll behaviour in physical situations*. <https://www.diva-portal.org/smash/get/diva2:1328413/FULLTEXT01.pdf>
- Unity Technologies. (2020). *Animation rigging 1.0.3: ChainIK constraint*. <https://docs.unity3d.com/Packages/com.unity.animation.rigging@1.0/manual/constraints/ChainIKConstraint.html?q=chainIK>
- Wampler, K. (2016). Fast and reliable example-based mesh IK for stylized deformations. *ACM Transactions on Graphics*, 35(6), 1–12. <https://doi.org/10.1145/2980179.2982433>

## PROCEDURAL ANIMATION TECHNOLOGY APPLICATION FOR HUMANOID GAME AGENTS

K. Builiso

### Abstract

This article analyzes inverse kinematics algorithms and demonstrates the application of the chosen algorithm to the Unity game engine. It has been found that Unity inverse kinematics solvers do not offer functionality necessary to apply anatomically correct animation to a humanoid agent. A few inverse kinematics algorithms were chosen for comparison, such as Cyclic Coordinate Descent, FABRIK and triangulation; due to the support of multiple end effectors and possible optimizations, FABRIK was chosen to be implemented in the “FABRIK IK” solver. After comparing the functionality of “FABRIK IK” to Unity’s “Two Bone IK” and “Chain IK” solvers, it was found that “FABRIK IK” has more built-in functionality than “Two Bone IK” and “Chain IK” in the analyzed areas. The measurement of framerate has shown that the application of either solver has not posed a significant difference in performance on Unity (the difference in performance ranges from –8.29% to 5.89%). The visual demonstration that shows the application of “FABRIK IK” demonstrates an anatomically sound and accurate walking cycle, especially compared to “Chain IK”; the accuracy of the animation is comparable to “Two Bone IK”.

**Keywords:** procedural animation, Unity game engine, inverse kinematics algorithm, solver, FABRIK, humanoid character.