

## ВИКОРИСТАННЯ БАГАТОЯДЕРНИХ ПРОЦЕСОРІВ ДЛЯ ПРОСТОРОВОЇ АНІМАЦІЇ ЖЕСТОВОЇ МОВИ

*Ю.Г. Кривонос, Ю.В. Крак, О.В. Бармак, Б.А. Троценко*

Інститут кібернетики імені В.М. Глушкова НАН України,  
03680, Київ, МСП, проспект Академіка Глушкова, 40,

Запропоновано технологію використання багатоядерних процесорів для розрахунків при відтворенні просторової анімації жестової мови. Приведена реалізація, що використовує зазначений підхід.

A technology of 3d animation for the sign language representation that benefits from the usage of multicore processors is suggested. Description of the implementation of the suggested technology is presented.

### Вступ та постановка задачі

Велика кількість людей з вадами слуху з однієї сторони, та стан сучасної науки, комп'ютеризація суспільства, використання мультимедійних та Інтернет-технологій з іншої сторони, створили достатні умови для розробки комп'ютерних систем комунікації цих людей у формах і образах близьких і зрозумілих для них і для оточуючого середовища.

Люди з вадами слуху для спілкування між собою використовують жестову мову. Основними візуальними засобами, що відтворюють цю мову є фотографічні та відео зображення жестів. Створювати за допомогою цих засобів сучасні навчальні та комунікаційні комп'ютерні системи досить складно і проблематично. Фотографічне зображення не відтворює необхідної динаміки жестів, а відеозображення досить громіздке і в ньому відсутня потрібна інтерактивність (не можливо подивитися на людину, яка відтворює жест з інших ракурсів крім того, в якому проведений відеозапис). Ці суттєві обмеження спонукають до розробки кібернетичних технологій за допомогою яких можна було б створювати нові комп'ютерні системи навчання та комунікації для людей з вадами слуху. Авторами запропонована концепція [1, 2] інформаційної технології невербального спілкування людей з вадами слуху. Комплексна інформаційна технологія включає в себе функціональність за синтезом: рухів жестової мови глухих, дактильної абетки, міміки промовляння на просторовій моделі людини.

Для реалізації можливості генерації анімації процесу мовлення за допомогою жестової мови з використанням віртуальних моделей людей, розроблені відповідні інформаційні та математичні моделі [3]. Основними вимогами до реалізації процесу анімації жестової мови за допомогою просторових моделей є:

- 1) досить висока розмірність цих моделей (особливо обличчя – тому, що потрібно відтворювати необхідні для жестової мови емоційні та артикуляційні складові);
- 2) модель має бути скелетною – тобто має бути можливість реалізації скінінгу для генерації всіх можливих рухів;
- 3) програмне забезпечення, що реалізує запропоновану технологію, має працювати на комп'ютерах зі звичайними мультимедійними вимогами до апаратної частини.

Дослідження існуючих засобів відтворення анімації [4] показали, що основна їх частина призначена для ігрової індустрії. Основними недоліками існуючих підходів є те, що для оптимізації за часом процесу рендерінгу сцени застосовують наступні підходи:

- 1) зменшення розмірності моделі;
- 2) зменшення кількості кісток у скелеті моделі;
- 3) більш високі вимоги до апаратної частини.

Специфіка жестової мови не дозволяє застосувати ці підходи. Виходячи з цього розглядається наступна постановка задачі.

Потрібно запропонувати алгоритм та реалізацію оптимального за часом розрахунку стану просторової моделі за умови її високої розмірності, наявності скелета, близького до скелета людини, та звичайних мультимедійних вимог до апаратної частини комп'ютера. Під оптимальним за часом розрахунком мається на увазі час, що відповідає частоті зміни кадрів за секунду 30 і більше.

### Модель для фіксації одиниць жестової мови

Процес відтворення жесту об'ємною моделлю людини можна вважати анімацією з відповідною частотою різних станів скелетної моделі людини.

Скелетна модель людини спрощено відтворює скелет живої людини. Її можна формалізувати як ієрархічну структуру, що складається з поєднаних кінематичних пар, які відтворюють основні кістки скелета людини. Сучасні пакети тривимірного моделювання (Poser, 3D Studio Max) вміють генерувати анімацію за допомогою віртуальної статичної моделі та інформації про зміну відповідних кутів скелета.

Отже для формального опису процесу фіксації жесту можна використовувати множини, які відображають спрощений скелет людини та зміни відповідних кісток цього скелета в часі:  $H = \mathbb{H}_i : H_i = \mathbb{K}_i, d_i, M_i \in \mathbb{M}$  – спрощений скелет людини (ієрархія кісток) та значення кутів, на які змінюються координати кісток у часі для відтворення анімації жесту, де  $H_i$  –  $i$ -та кістка скелета ( $i=0, \dots, N-1$ ,  $N$  – кількість кісток у скелеті);  $k$  – індекс кістки-предка;  $d_i = \mathbb{I}_i, y_i, z_i, \tau$  – координати точки – кінця кістки у системі координат, яка пов'язана з початком цієї кістки;  $M = \mathbb{M}_i : M_i = \text{Order}_i, \theta_i$  – значення кутів та порядок застосувань обертань для кожної кістки з плином часу.

### Модель для відтворення анімації жестів та міміки

Для синтезу процесу анімації жестів та міміки просторовою моделлю людини запропоновано наступний формальний опис, який використовує відповідні множини параметрів та алгоритми роботи з ними. Тривимірною моделлю людини, на якій буде реалізовуватись процес анімації жестів та міміки, має наступні атрибути:

$V = \mathbb{V}_i : v_i = \{x, y, z\}$  – множина вершин трикутників для триангуляції поверхні моделі людини;

$N = \mathbb{N}_i : n_i = \{x, y, z\}$  – множина нормалей до вершин;  $T = \mathbb{T}_i : t_i = \{t, s\}$  – множина текстурних координат до вершин;

$V^{ind} = \mathbb{V}_i^{ind} : V_i^{ind} = \{k_1, k_2, k_3\}$  – множина індексів, які вказують порядок побудови трикутників з множини вершин;

$I = \mathbb{I}_i : I_i = \text{img}$  – множина фотографічних зображень елементів моделі – текстури.

Для моделювання скелетної анімації потрібно вміти розраховувати нові значення вершин трикутників ( $V$ ). Для цього пропонується використовувати механізм скінінга. Під скінінгом будемо розуміти алгоритм прив'язки множини вершин трикутників, які визначають поверхню моделі до значень кутів скелета.

Тоді просторову модель скелетної анімації можна формалізувати наступним чином:

$MH = \mathbb{MH}_i : MH_i = \mathbb{K}_i \{l_1, \dots, l_m\}, d_i, Glb, Order_i$  – опис спрощеного скелета людини (ієрархія кісток) для реалізації скелетної анімації, де  $MH_i$  –  $i$ -та кістка скелета ( $i=0, \dots, N-1$ ,  $N$  – кількість кісток у скелеті);  $k$  – індекс кістки-предка;

$\mathbb{K}_i \cdot l_m$  – множина індексів нащадків кістки,  $d_i = \mathbb{I}_i, y_i, z_i, \tau$  – координати точки – кінця кістки у системі координат, яка пов'язана з початком цієї кістки;  $Glb$  – вектор для визначення координат кістки у глобальній системі координат,  $Order_i$  – порядок застосування обертання.

Позначимо:  $Skin = \mathbb{Skin}_i : Skin_i = \langle \text{IndexVertex}_i, \text{Weight}_i, \dots \rangle$  – множини вершин, які впливають на поточну вершину при зміні кутів.

Скінінг розраховується для кожної вершини  $V$  наступним чином:

$$v_j = \sum_{i=0}^N \langle \mathbb{K}_i * IBM_{H_i} * JM_{H_i} \rangle JW_{H_i} \quad (1)$$

де  $n$  – кількість кісток, пов'язаних з вершиною  $v$ ;  $IBM_{H_i}$  – інверсна bind-pose матриця для кістки  $H_i$ ;  $JM_{H_i}$  – матриця переміщення для кістки  $H_i$ ;  $JW_{H_i}$  – ваговий коефіцієнт для застосування впливу точок кістки  $H_i$  на вершину  $v$ .

Для моделювання анімації промовляння запропоновано використовувати механізм морфінгу. Морфінг можна визначити як алгоритм плавного переходу від одного стану об'єкта до іншого. При використанні морфінгу беруться лише опорні стани, за допомогою яких розраховуються проміжні стани і моделюється процес анімації. Відображення або побудова міміки на обличчі просторової моделі досягається при застосуванні відносного (сегментного) морфінгу до моделі голови і мімічного процесу.

Формула відносного морфінгу для  $M$  морфів у формалізмі моделі матиме наступний вигляд:

$$V' = V + \sum_{m=1}^M w_m \cdot TM_m, \quad (2)$$

де  $w_m$  – вагові коефіцієнти,  $V$  – меш базової моделі без морфінгу,  $TM_m$  – вхідний меш (морф) для блендінгу (blending). Результатом операції є лінійна комбінація мешів моделі та мімік.

Метод сегментного морфінгу дає змогу: формувати декілька виразів обличчя на основі невеликої кількості морфів та змінювати стан обличчя при анімації промовляння. Додатковою перевагою сегментного морфінгу є те, що можна анімувати щелепу незалежно від губ та робити анімацію очей (кліпання), незалежно від емоційного прояву на обличчі.

### Технологія оптимального відтворення анімації просторовою моделлю

У загальному випадку процес анімації показано на рис. 1. Виходячи з постановки задачі, необхідно запропонувати технологію розрахунку поточного стану моделі яка б дозволила відтворювати процес анімації за необхідний для нормального показу час. Беручи до уваги ті обставини, що:

- сучасні комп'ютери комплектуються багатоядерними процесорами (поки в кількості 2-х, але перспективи збільшення вже реальні) які дозволяють розпаралелювати обчислювальні процеси;
- математична та інформаційна моделі як процесу збереження жесту, так і самої просторової моделі побудовані так, що наступний стан не залежить від попереднього;
- стає можливим запропонувати розпаралелювання розрахунку станів моделі, що достатнє для прискорення процесу.

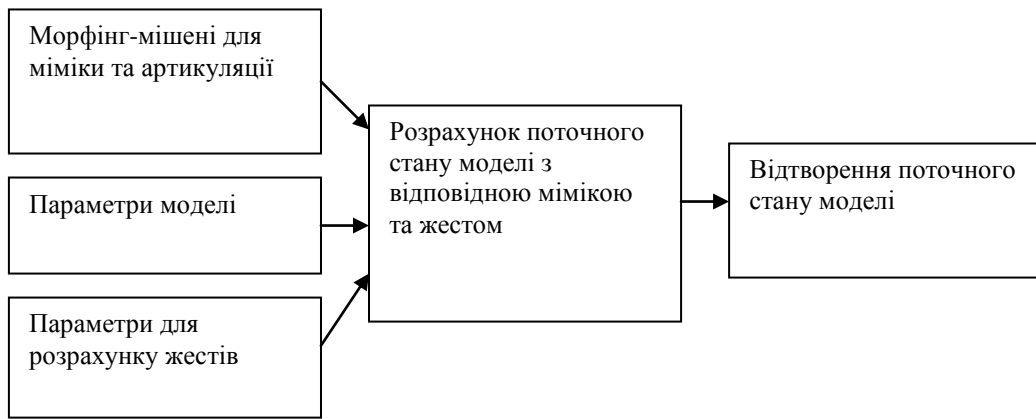


Рис. 1. Процес анімації стану моделі з відповідним жестом і мімікою

Для реалізації розпаралелювання обчислень розроблено алгоритм, що складається з  $2+n$  потоків, де  $n$  – кількість ядер процесора. Використовується факт, що обчислення різних кадрів є рівносильною задачею у термінах процесорного часу незалежно від параметрів моделі. Тому обчислення  $k$  кадрів можна розбити на  $n$  потоків максимально рівномірно таким способом: кожний  $i$ -й потік обробляє кадри з тим номером, залишок від ділення якого на  $n$  дорівнює  $i$ .

Основний потік (потік інтерфейсу користувача) ініціює створення  $n$  потоків обчислень і потоку анімації (який запускається призупиненим). Потік анімації показує із заданим проміжком підготовлені кадри та звільняє пам'ять після їх показу.

Вирішено проблему, коли операційна система виділяє нерівні кількості процесорного часу потокам (це відбувається, наприклад, при паралельному обчисленні інших задач). Кожен потік проасоційовано з одним семафором, який ініціалізовано одиницею. На початку обчислення кадру потік зменшує свій семафор на одиницю, після завершення кадру він збільшує семафор наступного потоку на одиницю (останній потік збільшує семафор нульового потоку). Таким чином, обчислення кадрів одного потоку не випереджує решту більше ніж на одиницю.

Особливу роль має потік з нульовим індексом: він збирає статистику обчислень інших потоків, прогнозує час завершення обробки кадрів, у разі, коли часу вдосталь, він запускає потік анімації.

Загальна схема підходу показана на рис. 2, де

- кружком з крапкою позначено початок виконання алгоритму;
- стрілками – перехід виконання;
- шестикутником – семафор; нуль всередині – його початкове значення;
- стрілка 1 означає, що перед початком переходу семафор буде зменшено на одиницю;
- стрілка 2 означає, що блок розрахунку кадрів збільшує семафор на одиницю;
- стрілки 3, 4, 5 показують запис та читання з області пам'яті;
- стрілка 6 означає передачу даних у інший блок і виклик методу в ньому.

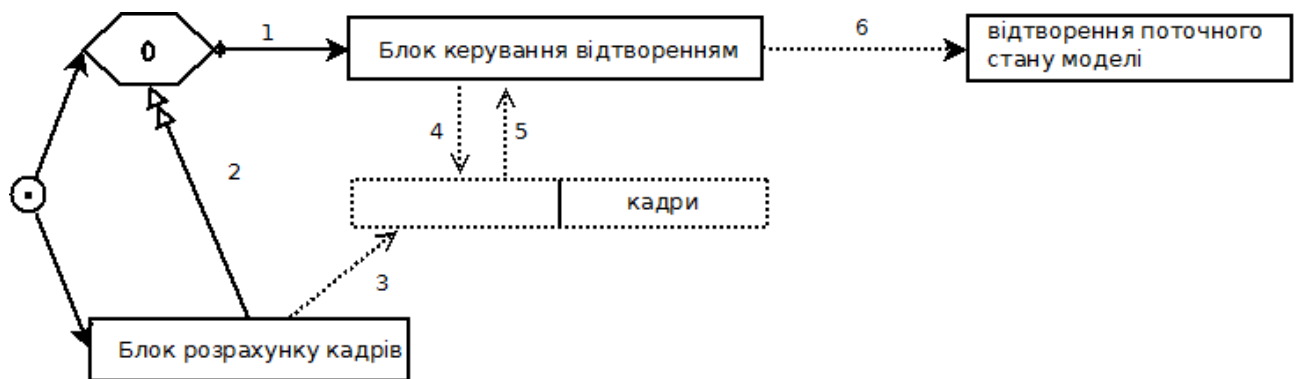


Рис. 2. Загальна схема розпаралелювання обчислень

Схема блоку розрахунку кадрів показана на рис. 3, де

- шестикутником позначено семафор; одиниця всередині – його початкове значення;
- суцільна стрілка з чорним кінцем означає, що для виконання обчислення семафор, з якого веде стрілка, зменшується на одиницю;
- суцільна стрілка з подвійним трикутником на кінці означає, що семафор, на який вказує стрілка, після виконання операції збільшується на одиницю;
- штрихова стрілка показує напрямок передачі обчислених даних.

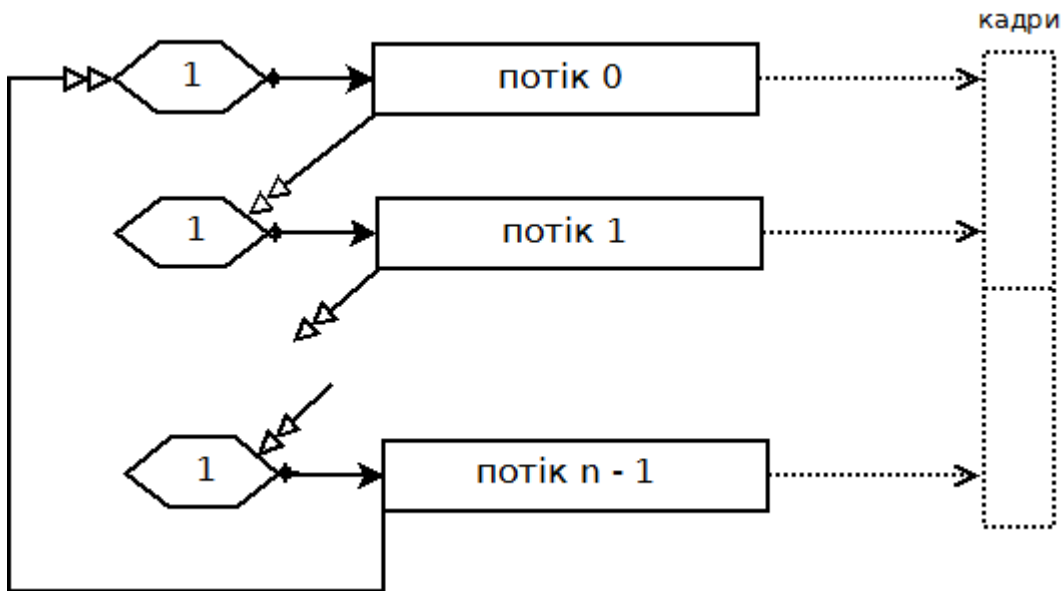


Рис. 3. Схема блоку розрахунку кадрів

### Реалізація технології

Запропоновану технологію реалізовано в застосуванні, що відтворює методику викладання жестової мови у спеціальних загальноосвітніх школах для глухих дітей. За основу було взято рекомендовану міністерством освіти програму [5] для початкових класів (рис. 4).

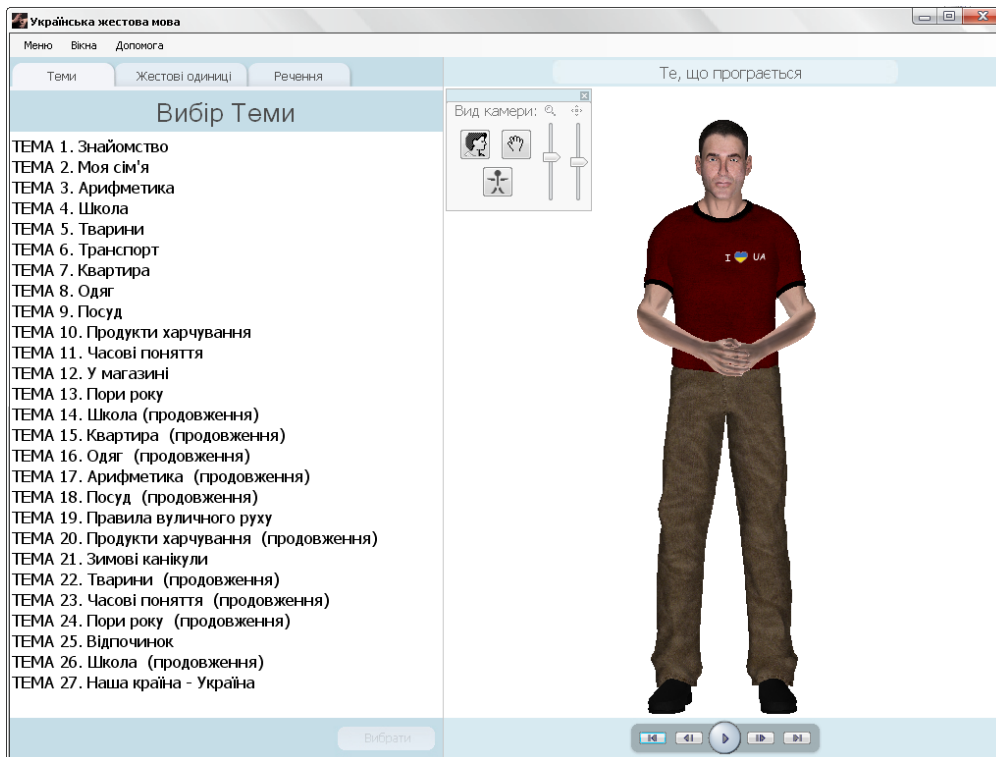


Рис. 4. Програма «Українська жестова мова»

Функціонально програма складається з трьох інформаційних блоків (теми, слова та речення) та блоку відтворення жесту віртуальною моделлю. Основним блоком є блок «Теми». У ньому сконцентровані основні методичні відомості, які розглядаються на уроці: що потрібно за цією темою сформувати в учнів, які навички потрібно засвоїти, які особливості морфології жестової мови та які види синтаксичних конструкцій містить матеріал. Наведений список нових жестів за темою, що розглядається. Наведені речення з цими словами, які містять граматичні та синтаксичні конструкції (рис. 5–7).

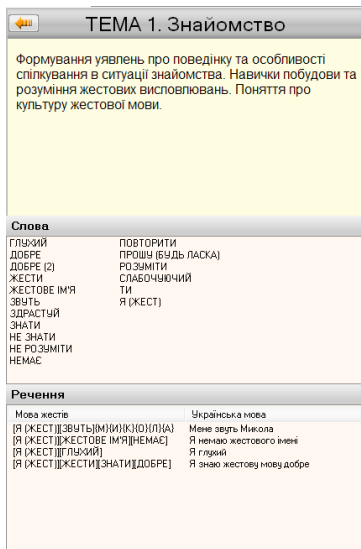


Рис. 5. Блок «Теми», тема «Знайомство»

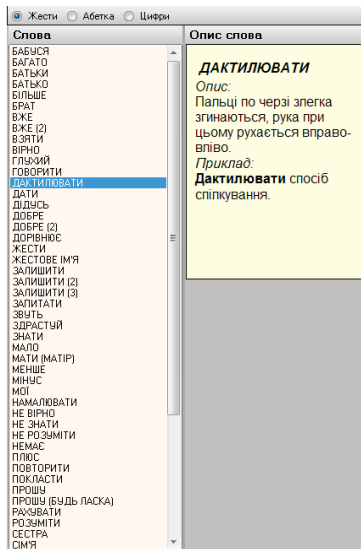


Рис. 6. Блок «Слова»

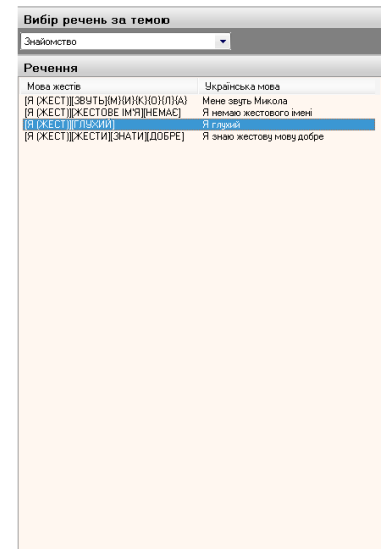


Рис. 7. Блок «Речення»

Блоки «Слова» та «Речення» носять допоміжну функцію. В них сконцентровані всі жести та приклади речень (які використовують розглянуті жести).

Блок відтворення жесту віртуальною моделлю у цій реалізації несе особливу функцію. Саме за допомогою нього стало можливим демонструвати в навчальному процесі динаміку жесту.

Для реалізації технології розроблено клас *Mu3DEngineCollada*, в якому інкапсульовані поля і методи, наведені в табл. 1.

Таблиця 1

Назва та сигнатура методів	Пояснення
<i>AnimationInProgress : bool</i> ,	Змінна, яка є істинною під час анімації. Використовується для блокування другої команди показу, якщо анімації ще триває.
<i>FramesPerSecond : float</i>	Задає кількість кадрів на секунду для анімації
<i>semStartAnimation : Semaphore</i>	Семафор, що регламентує запуск роботи потоку керування відтворенням
<i>ewhFrames : EventWaitHandle[]</i>	Масив подій, які вказують на готовність конкретного кадру
<i>lDataPrefetched : List&lt;float[,]&gt;</i>	Список підготовлених кадрів
<i>PrefetchThreadCount : int</i>	Кількість потоків, які проводять обчислення кадрів
<i>semsPrefetchSync : Semaphore[]</i>	Масив семафорів, які регламентують активність відповідних потоків обчислення кадрів
<i>thPrefetchers : Thread[]</i>	Потоки обчислення кадрів
<i>PrefetchedFramesInMemoryCount : int</i>	Лічильник загальної кількості обчислених кадрів

Метод *Play*, який запускає потоки обчислення кадрів і потік анімування, ініціалізує семафори, події, масиви семафорів тощо.

Метод *PrefetcherWorker*, що визначає потоки обчислення кадрів. Як параметр, приймає ціле число – *k*, яке визначає залишок від ділення, з яким працюватиме. Тобто, потік обчислюватиме кадри, які при діленні на *PrefetchThreadCount* дають у залишку *k*.

Задача розподілення потоків на кожне ядро виконується операційною системою. Але часто бувають ситуації, коли ядра зайняті сторонніми процесами. Це призвело б до того, що потоки передобробки отримали різні кванти процесорного часу, а як наслідок, кадри обчислювались нерівномірно.

Наведено приклад ситуації при використанні двоядерного процесора. Стан розрахунку наведено в табл. 2.

Таблиця 2

Номер кадру	0	1	2	3	4	5	6	7	...
Стан	+	+	+	-	+	-	+	-	...

У даній таблиці стан «+» означає, що кадр підготовлено, «-» – не підготовлено.

Цієї ситуації можна уникнути, якщо призупинити потік, який виконує більше обчислень, ніж інші. Для цього кожному потоку співставлено семафор, який визначає, скільки кадрів може обчислити потік. Такий семафор кожного потоку метод `Play` встановлює в одиницю. Перед обчисленням кадру потік зменшує виділений йому семафор. Після обчислення кадру – збільшує семафор наступного потоку. В ідеальній ситуації семафор кожного потоку буде збільшуватись тоді, коли даний потік зайнятий обчисленням свого кадру. Якщо він свій кадр обчислив, а його семафор збільшено не було, то він випереджає інші потоки, ядра яких зайняті іншими задачами. Тому цей потік буде операційною системою призупинений, а інший потік, який може виконуватись, буде перенесено на вільне ядро.

Додаткову роботу виконує потік, параметром якого є нуль. На початку своєї роботи він фіксує момент початку обчислень. Після того, як п'ять кадрів було обчислено, він також прогнозує момент часу, коли буде закінчено обчислення всіх кадрів на основі загальної кількості обчислених кадрів і тривалості обчислень. Якщо до кінця обчислення залишається менше часу, ніж тривалість самої анімації, він збільшує `semStartAnimation`; від цього моменту анімація й обчислення кадрів проходять одночасно, відомо, що на момент показу конкретного кадру його вже буде підготовлено.

Метод `AnimationWorker` зображає потік, який із заданою частотою показує кадри на екрані, звільняє пам'ять після показу кадру.

Цей потік працює з підвищеним пріоритетом, щоб зміна кадрів відбувалась вчасно.

## Висновки

Запропонована технологія використання багатоядерних процесорів для реалізації процесу просторової анімації жестової мови показала свою ефективність. На звичайному мультимедійному комп'ютері (Intel® Core™ 2 Duo CPU E8200 @ 2/66GHz, 2 Гб ОЗУ) для моделі що складається з 90 тисяч вертексів та 56 кісток скелета процес розрахунку стану моделі становив 15 мс, що відповідає частоті – 60 кадрів на секунду.

Подальші дослідження будуть направлені на розробку оптимальних алгоритмів для розрахунку скінінга та морфінга.

1. *Кривонос Ю.Г., Крак Ю.В., Бармак О.В., Тернов А.С.* Інформаційна технологія невербального спілкування людей з вадами слуху // Штучний інтелект. – 2008. – № 3. – С. 325–331.
2. *Крак Ю.В., Бармак О.В., Ганджа А.С., Тернов А.С., Шатковский Н.Н.* Компьютерная система виртуального общения людей с проблемами слуха. // *Advanced Studies in Software and Knowledge Engineering, International Book Series "INFORMATION SCIENCE & COMPUTING"*. Supplement to the International Journal "INFORMATION TECHNOLOGIES & KNOWLEDGE", 2008. – Vol. 2. – № 4. – P. 161–165.
3. *Кривонос Ю.Г., Крак Ю.В., Бармак О.В., Тернов А.С., Троценко Б.А.* Інформаційна технологія для укрїнської мови жестів // Штучний інтелект. – 2009. № 3. – С. 186–198.
4. *Чеканов Д.* Анатомия игровых движков. [Електронний ресурс] // 3D News. Daily Digital Digest – Режим доступу до журн.: <http://www.3dnews.ru/games/engines/>
5. *Грищенко С.С., Стьопкін В.В.* Українська жестова мова 1–4 класи. – К.: «Богдана», 2004.