

Київський національний торговельно-економічний університет

Кафедра інформаційних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка програмного забезпечення для управління кіберрукавичкою»

(за матеріалами Inc «Raccoon Technologies» м. Київ)

Студентки 2м курсу, 7 групи,
факультету обліку, аудиту та інформаційних
систем,
денної форми навчання
спеціальності
122 «Комп'ютерні науки»

Радовіченко
Олександра
Олегівна

(підпис студента)

Науковий керівник
канд фіз.-матем. н.
доцент

Нагірна Алла
Миколаївна

(підпис наукового
керівника)

Гарант освітньої програми
доктор т.н.
професор

Краскевич
Валерій
Євгенович

(підпис гаранта
освітньої
програми)

Київ 2018

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ПЕРЕМІЩЕННЯ ТВЕРДИХ ТІЛ.....	7
1.1 Кіберрукавички: історія та визначення.....	7
1.2 Переваги використання кіберрукавичок.....	10
1.3 Огляд існуючих алгоритмів для переміщення твердих тіл	21
Висновки до розділу 1.....	28
РОЗДІЛ 2. МАТЕМАТИКА ПЕРЕМІЩЕННЯ ТВЕРДИХ ТІЛ.....	29
2.1 Алгебра кватерніонів	30
2.2 Кінематика твердих тіл	37
2.3 Можливості кватерніонів у розробці програмного забезпечення.....	53
Висновки до розділу 2.....	59
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	60
3.1 Опис інтерфейсу користувача програми RASCOON.Clip.....	61
3.2 Опис програмного коду	70
3.3 Аналіз отриманих результатів	76
Висновки до розділу 3.....	79
ВИСНОВОК.....	80
РЕКОМЕНДАЦІЇ	81
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	82

					КНТЕУ-122-2018		
Зм.	Аркуш	№ документа	Підпис	Дата	Розробка програмного забезпечення для управління кіберрукавичкою Переміщення твердих тіл	Сторінка	Сторінок
Зав. кафедрою	Краскевич В.Є.					2	83
Керівник	Нагірна А.М.					Кафедра інформаційних технологій ОІ-2м-7	
Гарант	Краскевич В.Є.						
Розробив	Радовіченко О.О.						
Перевірів	Нагірна А.М.						

Анотація

“Розробка програмного забезпечення для управління кіберрукавичкою”

У даній роботі описується принцип створення пристрою для визначення місця розташування об'єкта без його прив'язки до супутникових або сотовим системам за допомогою доступних модулів і програмного забезпечення.

Кінематика обертального руху твердого тіла побудована на використанні апарату кватерніонів - чотиривимірних гіперкомплексних чисел зі спеціальними правилами множення.

Кватерніони мають ряд практичних переваг у порівнянні з іншими способами опису обертального руху твердого тіла. За допомогою кватерніонів ефективно вирішуються завдання на визначення переміщення кіберрукавички.

Результати наведені у вигляді розробленої програми, а також 3D графіка, що наочно відображає переміщення об'єкта в просторі.

Ключові слова: КВАТЕРНІОНИ, МАТРИЦІ ОБЕРТІВ, КУТИ ЕЙЛЕРА, КІБЕРРУКАВИЧКА, КОНТРОЛЕРИ, ПЕРЕМІЩЕННЯ ТВЕРДИХ ТІЛ, КУТОВІ ОБЕРТИ, КІНЕМАТИКА ТВЕРДИХ ТІЛ

“Software development for cyberglove management”

This work describes the principle of creating a device to determine the location of an object without its binding to satellite or cellular systems using accessible modules and software.

Kinematics of the rotational motion of a solid is based on the use of quaternion apparatus - four-dimensional hypercomplex numbers with special rules of multiplication.

Quaternions have a number of practical advantages over other methods of describing the rotational motion of a solid. With the help of quaternions, the tasks for determining the displacement of cyberglove are effectively solved.

The results are presented in the form of a developed program, as well as 3D graphics, which clearly displays the movement of the object in space.

Key Words: QUATERNIONS, ROTATION MATRICES, EULER ANGLES, CYBERGLOVE, CONTROLLERS, RELOCATION OF A SOLID, ANGULAR TURNOVERS, KINEMATICS OF SOLIDS

ВСТУП

Сучасні трекінг системи проходять серйозну перевірку часом в умовах постійних інноваційних трансформацій. Швидка зміна умов функціонування, вплив зовнішнього середовища, необхідність внутрішніх перетворень зумовлюють постійне удосконалення існуючих трекінг систем і стимулює розробників впроваджувати більш ефективні методи і засоби проектування. Це стосується і моделювання руху твердих тіл в просторі. Використання кватерніонів дає можливість уникнути багатьох обмежень кутових переміщень, простіше реалізовує комбінування обертів і дозволяє обійти проблеми, пов'язані з відсутністю можливості повороту навколо осі, що не залежить від обертань по іншим осях.

Об'єкт дослідження – кіберрукавички та інші тверді тіла.

Предмет дослідження – є кватерніони, що використовуються для переміщення твердих тіл.

Метою роботи є обґрунтування теоретичних і науково-методологічних положень щодо доцільності використання кватерніонів під час розрахунку переміщення твердих тіл, а також розробка програмного забезпечення, що наочно відображає їх використання для управління кіберрукавичкою.

Теоретичною та методологічною основою даної дипломної роботи стали фундаментальні положення зарубіжної та вітчизняної теорії функціонування та використання кватерніонів. Для досягнення визначеної мети залежно від складності наукових завдань застосовувався комплекс загальнонаукових методів: аналіз і узагальнення, аналогія та моделювання; системний та порівняльний аналіз.

					КНТЕУ-122-2018		
Зм.	Аркуш	№ документа	Підпис	Дата	Розробка програмного забезпечення для управління кіберрукавичкою Переміщення твердих тіл	Сторінка	Сторінка
Зав. кафедрою	Краскевич В.Є.					5	83
Керівник	Нагірна А.М.					Кафедра інформаційних технологій ОІ-2м-7	
Гарант	Краскевич В.Є.						
Розробив	Радовіченко О.О.						
Перевірив	Нагірна А.М.						

Значний внесок у розробку питань теорії і практики використання кватерніонів здійснили зарубіжні вчені: Полак Л.С., Джон Х.Конвей, Дерек А.Сміт, Фу К., Гонсалес Р., Э. Садбері та інші.

Зазначеній тематиці присвячені праці вітчизняних вчених-економістів: Гамільтон В.Р., Ільїн В.А., Мальцев А.И, Бранец В.Н., Шмиглевський І.П., Дубровин Б.А., Новиков С.П., Фоменко А.Т., Вебер Г..

Для досягнення поставленої мети були вирішені наступні завдання:

- розгляд сучасних методів визначення положення об'єкта в просторі. Оцінка переваг і недоліків кожного методу;
- проектування архітектури програмної системи;
- вибір технології для реалізації програмного продукту;
- реалізація алгоритмів, необхідних для визначення місця розташування кіберрукавички;
- створення 3D-моделі при пересуванні об'єкта;
- розробка програмних модулів і компоновка їх в готовий проект;
- тестування розробленого програмного забезпечення.

Розроблене програмне забезпечення знайде застосування в багатьох галузях виробництва і сферах життя, зокрема у реабілітології, геймінгу, в навігаційному обладнанні тощо.

					КНТЕУ-122-2018		
Зм.	Аркуш	№ документа	Підпис	Дата			
Зав. кафедрою	Краскевич В.С.				Розробка програмного забезпечення для управління кіберрукавичкою	Сторінка	Сторінка
Керівник	Нагірна А.М.					6	83
Гарант	Краскевич В.С.				Переміщення твердих тіл	Кафедра інформаційних технологій ОІ-2м-7	
Розробив	Радовіченко О.О.						
Перевірив	Нагірна А.М.						

РОЗДІЛ 1. ПЕРЕМІЩЕННЯ ТВЕРДИХ ТІЛ

1.1 Кіберрукавички: історія та визначення

Кіберрукавичка - це пристрій введення даних для взаємодії людини з комп'ютером у вигляді рукавички. Різні датчики вловлюють рухи згинів пальців і всієї кисті. Активуючи сенсори певними рухами, можна виконувати команди, пов'язані з ними. Потім ці рухи пальців і всієї руки розшифровуються за допомогою програмного забезпечення, що йде разом з рукавичкою. Кожен рух і обертання пальцем або всією рукою означає певну команду. При використанні кіберрукавички користувачеві не доведеться переводити погляд з монітора комп'ютера на клавіатуру або мишу. Також кіберрукавичка може бути корисна для людей з обмеженими можливостями, тому що комп'ютер зможе аналізувати команди за допомогою мови жестів. Дорогі високотехнологічні кіберрукавички можуть забезпечити зворотний тактильний зв'язок, спрямований на дотик людини, тим самим роблячи рукавичку пристроєм не тільки введення даних, а й виведення. Найчастіше кіберрукавички використовуються в ігровому середовищі, так як пристрій можна запрограмувати на виконання безлічі дій в комп'ютерній грі. Наприклад, повернути великий палець вниз - атакувати противника. Експерти відзначають, що особливо зручні кіберрукавички для онлайн-стратегій з великою кількістю користувачів.

Виникнення кіберрукавичок датується 1977 роком. Перша кіберрукавичка має назву Sayre Glove і була розроблена Electronic Visualization

					КНТЕУ-122-2018		
Зм.	Аркуш	№ документа	Підпис	Дата	Розробка програмного забезпечення для управління кіберрукавичкою Переміщення твердих тіл	Сторінка	Сторінок
Зав. кафедрою		Краскевич В.Є.				7	83
Керівник		Нагірна А.М.				Кафедра інформаційних технологій ОІ-2м-7	
Гарант		Краскевич В.Є.					
Розробив		Радовіченко О.О.					
Перевірив		Нагірна А.М.					

Laboratory.

Перша провідна рукавичка була заснована на ідеї недорогої, легкої рукавички для контролю руху рук. Sayre Glove забезпечувала ефективний метод багатовимірного контролю, такий як імітація комплекту слайдерів. Цей пристрій використовував світло на основі датчиків з гнучкими трубками з джерелом світла на одному кінці та фотоелементом на іншому. Коли пальці були зігнуті, кількість світла, що потрапляла в фотоелементи була різноманітною, що забезпечувало міру розгинання пальців. В основному ця кіберрукавичка використовувалась для маніпулювання слайдерами, але була легкою і недорогою.

У 1983 році Томас Г. Циммерман подав заявку на отримання патенту (патент США 4542291) на оптичний гнучкий датчик, встановлений в рукавичці, який фіксує рухи згинів пальців. Також Циммерман працював з Джарон Ланье над ультразвуковими і магнітними датчиками, які ловлять положення рук (патент США 4988981).

Однією з перших кіберрукавичок, доступних для використання вдома, стала Nintendo Power Glove, що з'явилася в 1987 році. Це був офіційний продукт, але Nintendo не брала участь в розробці та виробництві аксесуара. Розробкою Power Glove займалися компанії Mattel в США і PAX в Японії. Коштувала рукавичка майже 80 \$. В рукавичку вбудований звичайний джойстик і кілька додаткових кнопок, призначених для набору програмного коду ігор, які не були офіційно адаптовані для Power Glove. На пальці рукавички нанесені карбонові чорнило, передають стиснення і розтягування. Підключення відбувається в роз'єм для першого джойстика. В рукавичці є два ультразвукових динаміка (трансмітера), передають звук на частоті 40 kHz, що не чути для людського вуха. У кріпленні, розташованому на телевізорі, вбудовані три мікрофони (ресивера), які беруть звуки від рукавички і вираховують її місце розташування в просторі по осях X, Y, Z і кут нахилу.

					КНТЕУ-122-2018	Аркуш
						8
Зм.	Аркуш	№ документа	Підпис	Дата		

У зв'язку з тим що позиція руки визначається за допомогою звуку, грати в галасливій обстановці не представляється можливим, тому що мікрофони часто помиляються, тому використання такої кіберрукавички було не зручним.

Після кіберрукавички Nintendo була кіберрукавичка CyberGlove, створена Virtual Technologies Inc. в 1990 році. У 2009 році змінився власник кіберрукавички і відтоді розробкою, виготовленням та сбутом стала займатись CyberGlove Systems. CyberGlove Systems позиціанує себе як компанія, що розробляє апаратно-програмні технології, які дозволяють користувачам взаємодіяти з комп'ютерами, що надають їм відчуття дотику. CyberGlove використовує фірмову резистивну технологію згинання, що дозволяє точно перетворити рухи рук і пальців в цифровий стиснутий кут даних у режимі реального часу. Система CyberGlove була використана в різноманітних реальних програмах, включаючи віртуальну реальність, анімації, цифрові прототипи, біомеханіка, і реального часу захоплення руху. Окрім CyberGlove були розроблені три інші рукавички: CyberTouch, яка вібрує на кожному пальці рукавички при дотику об'єкту в віртуальній реальності; CyberGrasp, яка фактично імітує стиснення та наближення до твердого та м'якого об'єкту, а також пристрою CyberForce, яка робить все перераховане вище, а також вимірює точне рух всієї руки користувача.

У 2002 році був випущений P5 Glove. Рукавичка Essential Reality P5 Glove розроблялась і створювалась, в першу чергу, для домашнього використання і для комп'ютерних ігор. Принцип її дії: у базовій станції розміщуються 4 фотодіоди (2 пари), встановлені в невеликому поглибленні. Одна пара контролює горизонтальні переміщення, інша - вертикальні. При переміщенні світлодіодів на корпусі P5 Glove, частина світла, що випромінюється ними, буде блокуватися краями поглиблення, і в результаті фотодіоди будуть виробляти меншу кількість електричного струму.

					КНТЕУ-122-2018	Аркуш
						9
Зм.	Аркуш	№ документу	Підпис	Дата		

Абсолютно однакові фотодіоди при попаданні на них світла видають однаковий сигнал, і щоб розрізнити їх, один фотодіод частково закритий від світла. Завдяки цьому мікропроцесор розрізняє лівий і правий фотодіоди. Порівнюючи сигнали від світлодіодів, система обчислює положення рукавички по горизонтальній осі. Мінуси: існує не так вже й багато додатків, в яких підтримується P5 Glove. На сьогоднішній день такою підтримкою володіють лише кілька ігор - Hitman 2, Tiger Hunt, Beach Head 2000, Serious Sam 2 і Black & White. Три з них поставляються в комплекті з рукавичкою, на інші доведеться ставити патчі. У будь-яких інших іграх P5 Glove можна використовувати в режимі емуляції мишки. Працювати буде, але замість шести ступенів свободи залишаться дві - вліво / вправо і вперед / назад. Рукавичка P5 сумісна з Microsoft Windows XP і Mac OS версії 9 або нижче. Також існують неофіційні драйвери для Linux. Ця рукавичка отримала безліч позитивних відгуків у пресі, але широко використовуватися так і не почала.

Зараз існує досить велика кількість різних кіберрукавичок. Gloveone, Avatar VR, Noitom Hi5 VR Glove, Manus VR Glove, Senso Glove, Oculus Gloves.

Але використання більшості з них потребує встановлення додаткових камер чи трекінг систем, що є досить затратно.

1.2. Актуальність кіберрукавички у наші дні

У сучасному світі новітні технології є рушійною силою. Вони використовуються будь де і це являє собою логічною еволюцією людини. Якщо раніше на перший план виходили ремесло, фізична праця, то зараз цю ланку займають технології.

					КНТЕУ-122-2018	Аркуш
						10
Зм.	Аркуш	№ документа	Підпис	Дата		

Використання кіберрукавичок найрізноманітніше:

- геймінг;
- проведення медичних операцій;
- розробка дрібної моторики рук;
- управління марсоходом;
- розумний дім;
- авіація;
- інтерактивні тренажери та реабілітація;
- vr/ar технології;
- управління роботами тощо.

Розглянемо сферу VR-технологій. Останнім часом все більше доводиться стикатися з завданнями, що стосуються взаємодії у віртуальній реальності. Тема цікава і актуальна, так як прямим чином пов'язана з функціональними можливостями у віртуальному середовищі і впливає на призначений для користувача досвід в цілому нарівні з технологіями самих VR-шоломів, розширюючи межі рівня занурення.

Як тільки з'явилася перша версія очок Oculus Rift DK1 на нову хвилю інтересу до VR відразу ж піднялися і питання систем введення - клавіатура і миша виявилися фактично марними інструментами, а управління через наведення поглядом - недостатнім способом взаємодії з елементами оточення і інтерфейсу.

У мобільному VR відмінно прижився тачпад, який можна побачити в моделях гарнітур Samsung Gear VR, а також бездротові джойстики. Але в стаціонарному VR, крім використання класичних джойстиків (без цих пристроїв не обійтися, так як в цілому ряді ігрових жанрів вони є оптимальним рішенням і в VR), існують набагато більші можливості для систем введення.

					КНТЕУ-122-2018	Аркуш
						11
Зм.	Аркуш	№ документа	Підпис	Дата		

Зараз всі три великих гравці - Oculus VR, HTC / Valve і Sony -

використовуватимуть разом зі своїми окулярами контролери, чутливі до руху. Положення самих очок і контролерів при цьому можуть визначатися в просторі.

Для початку давайте зрозуміємо, що в першу чергу потрібно від системи управління в VR.

І якщо спробувати описати вимоги одним словом, то думаю це буде слово «природність». Система управління в віртуальній реальності повинна бути максимально наближена до природного взаємодії з навколишнім нас об'єктивною дійсністю, з об'єктами якої ми взаємодіємо переважно за допомогою рук.

VR цікава тим, що ми можемо не тільки відтворювати реальність в віртуальності, а й розширити межі звичайної дійсності і це, звичайно, може ставитися і до UI, але в даному випадку дещо виходить за позначену тему.

Природність також має на увазі низький поріг входу за рахунок інтуїтивного розуміння принципів роботи.

Є ще синхронізація - вона пов'язана з темою природності і надзвичайно важлива. Мова про синхронізацію дій користувача в реалі і в віртуалі. Цей аспект стосується не тільки UI, скільки в першу чергу UX, самосприйняття людини у віртуальній реальності.

І цьому, нарівні з якісною реалізацією роботи VR-окулярів, сприяють кіберрукавички та інші гаджети для рук. Чим більші можливості для трекінгу доступні, тим більший ефект занурення можна отримати і тим більше можливості для взаємодії з оточенням. Але синхронізація - це не тільки питання можливостей і точності трекінгу, але також і його стабільної роботи.

І проста відповідь на питання чому великі гравці пропонують використовувати зазначені ними системи визначення об'єкту в просторі, кіберрукавички на поточний момент часу дозволяють створювати найбільш універсальне, природне, інтуїтивно зрозуміле і стабільну взаємодію у

					КНТЕУ-122-2018	Аркуш
						12
Зм.	Аркуш	№ документу	Підпис	Дата		

віртуальному середовищі, яке можна позиціонувати як продуктове рішення для

використання вже зараз.

Взаємодія. Якщо говорити про пристрої, які пропонують самі власники платформ, то на перший погляд може здатися, що два контролера з кнопками, які потрібно обов'язково тримати в руках можуть погіршити процес природного взаємодії з оточенням.

Але контролер в руці є ніщо інше, як інструмент, такий, як молоток, кермо автомобіля або комп'ютерна миша. На певному рівні цей інструмент стає частиною самого користувача в процесі взаємодії з ним. Наприклад, коли людина вже деякий час керує автомобілем, він починає фактично відчувати його габарити, відчуття його тіла розширюється до кордонів корпусу машини. У певні моменти ти забуваєш, що тримаєш контролер і просто оперуєш рукою віртуальними об'єктами сцени. Контролер просто стає частиною тебе, як щось само собою зрозуміле.

Тепер спробуємо розібратися в існуючих альтернатив та в тому, так вони гарні з позиції природності і синхронізації - і це важливий момент - саме для кінцевого широкого кола споживачів.

Оптичні датчики захоплення дрібної моторики.

В даному випадку є кілька проблем використання подібного рішення, ось одні з найбільш критичних:

Для взаємодії з віртуальним середовищем руки потрібно тримати завжди перед датчиком - перед обличчям користувача в досить обмеженій області. Це вкрай неприродно, а також фізично важко протягом тривалого періоду роботи.

Поточні алгоритми визначення кистей рук далекі від ідеалу - під деякими кутами руки взагалі фактично перестають визначатися - високий поріг входу через постійні проблеми з трекингом сильно погіршують досвід використання

					КНТЕУ-122-2018	Аркуш
						13
Зм.	Аркуш	№ документа	Підпис	Дата		

рішення в цілому.

Одна з важливих проблем, яка є в поточних VR системах вводу - питання

переміщення людини у віртуальному просторі. В цілому, крім обмеженого простору для переміщення у HTC Vive, всі інші системи не дозволяють синхронізувати реальне переміщення тіла людини і віртуальне. А саме відсутність подібної синхронізації веде до так званого сенсорного дисонансу через якого і виникає ефект запаморочення при переміщенні в VR.

Проблема настільки серйозна, що в багатьох проектах, спеціально розроблених під VR, мало не єдиним варіантом виявилось повністю відмовитися або сильно переробити можливість вільного переміщення в просторі.

Платформи переміщення. Звичайно, є різні технічні рішення, що дозволяють імітувати природне переміщення. Наприклад, Virtuix Omni. Свою аудиторію ці пристрої знайти цілком зможуть, але в поточному вигляді всі існуючі платформи погано придатні для саме широкого кола споживачів, щоб бути частиною продуктової версії самих очок - громіздка установка, вимоги знімати взуття або мати спеціальне взуття для використання на платформі, час на настройку перед грою, досить вузька сфера застосування пристрою, ось лише кілька найочевидніших причин, чому широке поширення ці платформи в поточному вигляді навряд чи отримають.

Кіберрукавички. Звичайно, у багатьох ідея кіберрукавичок для управління в VR викликає ностальгію за часами розквіту класичного кіберпанку в літературі і кіно, але в практичному плані рукавички можуть вирішити багато питань з датчиками вище. Тому використання кіберрукавичок досить доречно, тим паче, що людині такий пристрій буде достатньо звичним, щоб вона вирішила їм користуватись.

Наукові дослідження в трекінг системах. Позиційний трекінг (англ. Positional tracking) - одна з технологій віртуальної реальності, що лежить в основі

					КНТЕУ-122-2018	Аркуш
						14
Зм.	Аркуш	№ документу	Підпис	Дата		

взаємодії людини з віртуальним світом. Призначена для визначення позиції і орієнтації реального об'єкта (наприклад, руки, голови або спеціального пристрою) в віртуальному середовищі за допомогою декількох ступенів свободи. Як правило, трьох координат його розташування (x, y, z) і трьох кутів, які задають його орієнтацію в просторі («крен», «тангажу», «нишпорення» або кути Ейлера).

Компанія Raccoon.World - це технологічна компанія, яка створює розумні пристрої для тіла та складні системи для взаємодії з цифровими, віртуальним, доповненим та реальними світами.

Raccoon.Clip (рис.1) - простий і зрозумілий контролер, який покликаний стати універсальним інструментом для взаємодії з віртуальною реальністю (простіше і функціональніше існуючих) і зручною альтернативою класичним контролерам для відеоігор (мишка / клавіатура, джойстик).



Рис.1.1. Raccoon.Clip

Raccoon.Glove (рис.1.2) - складний пристрій, що дозволяє повноцінно взаємодіяти з віртуальним світом. Завдяки винайденим технологіям та рішенням за допомогою Raccoon.Glove можна фактично відчувати віртуальні предмети - їх форму, текстуру, пружність та ін.

					КНТЕУ-122-2018	Аркуш
						15
Зм.	Аркуш	№ документа	Підпис	Дата		

Raccoon Glove - a glove that allows you to read and transfer the fine motor skills on to a virtual hand or remote robo - hand.



Рис.1.2 Raccoon.Glove

Щоб зробити свій гаджет "ідеальним для всіх геймерів світу", Raccoon.World опитав п'ять тисяч користувачів. Вони розповіли про те, які якості повинні бути у ідеального контролера. Виявилося, що він повинен: бути універсальним і сумісним з будь-якою грою; забезпечувати повну свободу пересування; працювати з будь-яким пристроєм і операційною системою (ОС); бути зручними в експлуатації; не вимагати змін в програмному забезпеченні (ПЗ) і додаткових інструментів. Так і виникають гаджети Raccoon.World.

Перш за все, гаджет компанії Raccoon.World - універсальний пристрій. Тоді як більшість інших контролерів працюють тільки з однією конкретною платформою або гарнітурою.

На відміну від мишки і клавіатури, пристрої Raccoon.World не прив'язують ігровий процес до поверхні. У порівнянні з джойстиками - залишає руки вільними. Вони працюють з будь-яким смартфоном або планшетом, на відміну від інших контролерів для мобільного VR / AR.

					КНТЕУ-122-2018	Аркуш
						16
Зм.	Аркуш	№ документа	Підпис	Дата		

На даний момент існуючі контролери для VR / AR мають ряд значних недоліків. Вони незручні у використанні, займають руки, вимагають особливих умов - додаткових кабелів, камер, освітлення. Гаджети Raccoon.World легко кріпляться на руку без необхідності тримати її, працює по бездротовому з'єднанню в будь-яких умовах освітлення, а також набагато точніше передає орієнтацію руки в порівнянні з існуючими контролерами.

Крім цього, використання універсального контроллера допоможе підняти рівень взаємодії будь-якого мобільного VR-шолома до рівня DayDream (VR-шолом з контролером від Google). Розробників буде простіше підштовхнути до створення контенту відразу для декількох рішень, а сам проект Google посприє просуванню індустрії в цілому.



Рис.1.3 Використання гаджетів Raccoon.World

					КНТЕУ-122-2018	Аркуш
						17

Зм.	Аркуш	№ документа	Підпис	Дата	
-----	-------	-------------	--------	------	--

Контроллери Raccoon.World можуть повноцінно замінити існуючі контролери для відеоігор, запропонувавши користувачам покращений досвід від їх улюбленого проведення часу. В ході досліджень ми виявили ряд проблем, які вирішує пристрій. Потрібно використовувати обидві руки і робити величезну кількість рухів і дій для управління традиційними контролерами. Крім цього, тривалі ігри викликають болі в руках і тунельний синдром. Тривала ж гра з Raccoon.World не втомлює, так як положення руки може бути абсолютно будь-яке - витягнута в повітрі, зігнута перед грудьми, що лежить на коліні або на столі, тощо. В залежності від типу гри можна вибрати найбільш відповідний стан руки, в процесі змінюючи його, щоб уникнути затікання або втоми. Але головне - контроллери абсолютно автономні і не потребують доопрацювань на стороні виробника гри. Вона може використовуватися як HID-пристрій в будь-яких програмах, де використовуються миша / клавіатура / джойстик.

Подальший розвиток можливо в таких напрямках: індустрія для дорослих, реабілітологія, медицина, віддалене маніпулювання за допомогою роборуки, навчальні системи.

Наразі як найголовнішою сферою діяльності Raccoon.World було обрано медицина, а саме реабілітологія. Рішення Raccoon.Recovery дає можливість людям після травм і інсульту розробити моторику рук. Програма підбере найбільш ефективний курс і буде сигналізувати, коли пацієнт робить неправильні рухи. У додатку можна дивитися на поточний результат і скільки залишилося до одужання. Але, найприємніше для пацієнта це те, що весь курс реабілітації складається з гри у відеоігри або використання комп'ютеру, чи приставки на свій розсуд.

Протестували 256 пацієнтів в 7 реабілітаційних центрах в ЄС, США та України. Отримали грант Horizon 2020 на розмірі 50 тисяч євро на імплементацію рішення. Команда пройшла в Startupbootcamp Digital Health

					КНТЕУ-122-2018	Аркуш
						18
Зм.	Аркуш	№ документу	Підпис	Дата		

Berlin і з вересня 2018 починала тестування в Берліні.

Справа в тому, що після травм рук і інсульту мільйони людей потребують ефективної фізичної реабілітації. Лікарі обмежені місцем і часом навчити пацієнтів знову користуватися руками. Існуючі рішення не контрольовані реабілітологом і нудні, тому пацієнти проходять реабілітацію довго і один на один із самим собою.

Рішення Rassoon.Recovery дозволяє побудувати найбільш ефективний курс реабілітації для конкретного пацієнта і стежити за динамікою в даний момент часу. В короткий проміжок часу можна спрогнозувати скільки триватиме курс реабілітації. І з точки зору грошей, реабілітаційні центри зможуть отримувати дохід після того як пацієнт іде на домашнє лікування, так як пацієнту вручають контролер, який фіксує рухи, поки пацієнт грає у відео ігри або просто використовує персональний комп'ютер.

Лікарі використовують кустарні методи (масаж, моделювання пластиліном, головоломки), дорогі громіздкі установки, які не можна використовувати поза лікарнею (InMotion ARM™, Gloreha) або дорогі мобільні рішення з обмеженим програмним забезпеченням (Hand Tutor, MusicGlove). Є кілька нових рішень (Xtensor® Hand Exerciser, Mind-controlled device), які знаходяться тільки в тестовому режимі в лабораторіях, але в них є також велика кількість проблем.

Існуючі методи лікування не тільки незручні для лікарень, але також недоступні для пацієнтів і пригнічують їх. Рішення Rassoon.World буде корисно для клінік і мотивує пацієнтів. Це набагато ефективніше, ніж кустарні методи. Продукт Rassoon.World вирішує питання наявності контенту для тренувань, мобільності за адекватні гроші, можливість збору даних, аналітики, програми для реабілітації. Легко інтегрується і використовується як в лікарні, так і вдома.

					КНТЕУ-122-2018	Аркуш
						19
Зм.	Аркуш	№ документу	Підпис	Дата		

Рішення Raccoon.Recovery може використовуватися з будь-яким існуючим програмним забезпеченням - лікар може вибрати гру для тренування конкретних м'язів відповідно до переваг пацієнта. Це робить процес реабілітації захоплюючим і різноманітним. Мобільний додаток дозволяє пацієнтові стежити за прогресом і показувати досягнення, щоб зробити реабілітацію привабливішою.

Продукт допоможе лікарям збирати статистику про пацієнта, поставити більш точний діагноз і підібрати курс лікування. Програма збирає дані про пацієнта, навіть якщо він не в лікарні. Людина може грати в комп'ютерну гру вдома, а потрібна інформація зберігається в хмарі, де лікуючий лікар може її подивитися (рис.1.4).



Рис.1.4 Збирання даних пацієнта

Ідея зробити такий продукт з'явилася, коли Raccoon.World в минулому році представляли на TechCrunch Disrupt в Сан-Франциско. Тоді до стенду компанії підійшов хлопчик, який через проблеми з руками не міг впоратися з мишкою або геймпадом. Продукт української компанії дозволив дитині вперше

					КНТЕУ-122-2018	Аркуш
						20
Зм.	Аркуш	№ документа	Підпис	Дата		

відчути, як це - грати у відеоігри.

За даними Всесвітньої Організації Охорони здоров'я, щорічно 17 млн людей у всьому світі страждають від церебрального паралічу. Через нестачу медичних програм тільки половина з них отримують повну реабілітацію.

Перемоги в конкурсах і інші нагороди:

- Horizon2020 2018 - грант від Європейського Союзу в розмірі 50 тисяч євро
- Vernadsky challenge 2017 - виграли приз глядацьких симпатій з призовим фондом в 200 000 грн
- CES (Las Vegas) - Raccoon виграв можливість участь у виставці електроніки CES в Лас-Вегасі [15]
- Стартап Батл Інтерконтинентальний - 1-місце [16]
- Стартап Батл Україна - Польща - 1-е місце [17]

Проходження інкубаційних / акселераційних програм:

- Стартап кухня [18]
- У вас не вийде [19]
- Crowd.Inc 2018
- SABIT 2018
- Startupbootcamp Digital Health Berlin 2018

Також Raccoon.World потрапив в ТОП-10 стартапів в області робототехніки і VR на Techstars Boston, переміг в International Startup Battle і отримав звання «Кращий проект-2017» за результатами голосування глядачів на Vernadsky Challenge.

1.3 Математичні моделі для переміщення твердих тіл

У будь-яких сучасних кіберрукавичок у склад входять акселерометр та

					КНТЕУ-122-2018	Аркуш
						21
Зм.	Аркуш	№ документу	Підпис	Дата		

гіроскоп. Акселерометр використовують, оскільки у більшості випадків нахили різних об'єктів визначаються з допомогою гравітаційної сили Землі. Акселерометр - це пристрій, що вимірює проекцію прискорення на його чутливу вісь. За значення вимірної проекції визначають кут нахилу. Але у реальному часі окрім гравітаційної сили Землі на об'єкт діють й інші сили, які можуть виникати під час обертів, тряски чи будь-якої зовнішньої дії. Сила гравітації має постійну величину, тому будь-які інші сили, що діють на об'єкт, можуть змінювати дані, що входять до акселерометра. Тому при розрахунку кута нахилу можуть виникати помилки. Щоб отримувати коректні дані потрібно спочатку обробити вхідний сигнал.

Що стосується гіроскопа, то його використання обумовлено тим, що акселерометр не може вимірювати оберт навколо вертикальної осі. На практиці використовують три підходи до обчислення таких кутів:

- Кути Ейлера;
- Матриці обертів;
- Кватерніони.

Кути Ейлера. Кути Ейлера – це кути, що описують поворот об'єкту в тривимірному евклідовому просторі. При цьому розглядаються дві прямокутні системи координат, які мають загальний центр: нерухома система і рухома, що пов'язана з об'єктом [14]. На рис.1 нерухома система координат має позначення XYZ (вона нахилена), а рухлива система позначена як хуз. Кути Ейлера є кути, на які повертається рухома система координат, що пов'язана з об'єктом, до суміщення з нерухомою системою. У класичному варіанті перший поворот відбувається на кут α навколо осі z, пов'язаної з об'єктом, до тих пір, поки не відбудеться збіг осі x, пов'язаної з об'єктом, з площиною XY нерухомої системи. Такий збіг відбудеться по лінії перетину площин XY і ху (лінія N на рис. 1.5). Наступний поворот здійснюється на кут β навколо нового положення осі x, пов'язаної з об'єктом, до того часу, поки не сполучаться осі аплікату обох

					КНТЕУ-122-2018	Аркуш 22
Зм.	Аркуш	№ документа	Підпис	Дата		

прямокутних систем. При цьому вісь y , пов'язана з об'єктом, виявиться в площині xu нерухомої системи координат XYZ . Останній поворот проводиться на кут γ навколо нового положення осі аплікат рухомий системи координат (вона буде збігатися з такою ж віссю нерухомої системи), після чого осі координат XU і xu сполучаться.

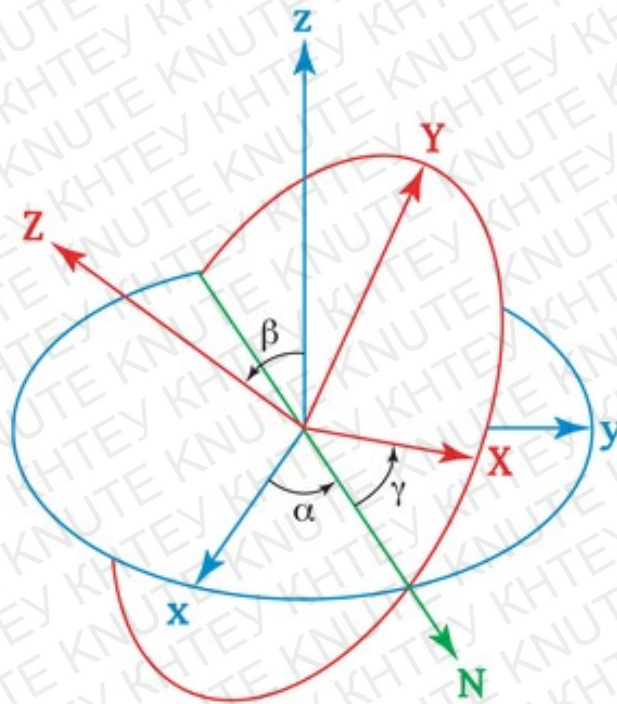


Рис. 1.5 Кути Ейлера

Такі повороти некомутативними, і кінцеве положення рухомої системи координат залежить від порядку, в якому відбуваються повороти.

Якщо відомі координати вектора R (r_x , r_y , r_z) в рухомий системі координат $X'Y'Z'$ і відомі кути Ейлера (α , β , γ) рухомої системи координат xu відносно нерухомої, то можна обчислити координати цього вектора в нерухомій системі координат XYZ . Для цього слід побудувати матриці трьох послідовних поворотів на кути α , β і γ :

					КНТЕУ-122-2018	Аркуш
						23
Зм.	Аркуш	№ документа	Підпис	Дата		

$$T_1 = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{bmatrix}, \quad T_3 = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Перемноживши ці матриці в зворотному порядку, отримаємо підсумкову ортогональну матрицю:

$$T = T_3 \times T_2 \times T_1, \quad (1.1)$$

яка перетворює координати вектора R (r_x, r_y, r_z) рухомої системи координат в координати вектора N (n_x, n_y, n_z) такої ж довжини в нерухомій системі координат:

$$N = T \times R, \quad (1.2)$$

де N і R - матриці-стовпці відповідних координат.

Кути Ейлера є найбільш природними і зрозумілими при виконанні різних операцій обертання об'єктів, оскільки вони відповідають обертанням об'єкта, які спостерігаються в видових вікнах тривимірних графічних систем. Однак їх використання в системах комп'ютерної анімації стикається з низкою труднощів. Перш за все, це необхідність вибору певної послідовності поворотів об'єкта щодо осей системи координат. Якщо повернути об'єкт спочатку навколо осі X , потім навколо осі Y і, нарешті, навколо осі Z , то це буде зовсім не той поворот, якби повернути цей об'єкт на ті ж кути, але в іншій послідовності [7].

Розглянемо ще один мінус - застосування ейлерових обертань в програмах комп'ютерної графіки, обмежується одночасним обертанням на кут, що не

					КНТЕУ-122-2018	Аркуш
						24
Зм.	Аркуш	№ документа	Підпис	Дата		

перевищує 360° . Однак це не заважає створювати анімацію вручну за екраном дисплея.

Інша проблема кутів Ейлера полягає в наявності Gimbal lock, або шарнірного замку. Його поява залежить від вибору порядку поворотів об'єкта. Наприклад, повернемо об'єкт спочатку навколо осі Z на кут 140° , потім навколо осі X на кут 90° , а потім на кут 130° навколо осі Y (рис. 1.6).

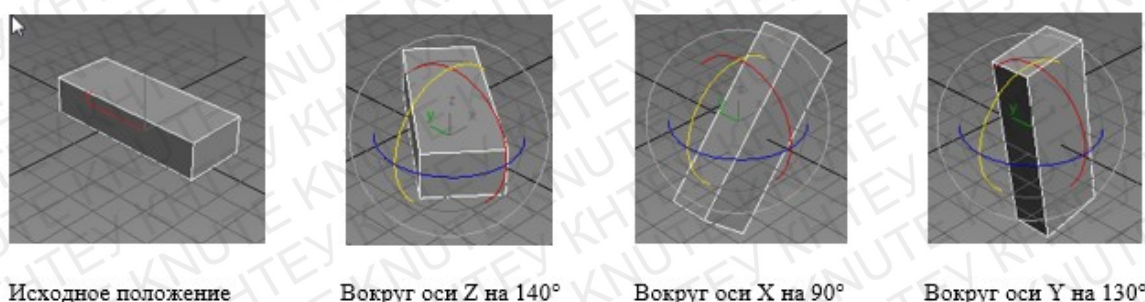


Рис.1.6 Послідовні повороти об'єкта

Якщо тепер заново виконати ту ж послідовність поворотів, наприклад, на кути 10° навколо осі Z, потім на 90° навколо осі X, а потім на 0° навколо осі Y, то отримаємо той же результат. Проблема полягає в тому, що коли обертання навколо осі X стає рівним 90° або -90° , то локальна вісь обертання Y стає паралельною осі Z, але зі зворотним напрямком, і тому обертання навколо неї вступає в конфлікт з попереднім обертанням навколо осі Z.

Отже, використання кутів Ейлера для анімації або переміщення твердих тіл має значну кількість недоліків. Найбільш негативний момент - це поява шарнірного замка. Саме тому використання кутів Ейлера при певних умовах може завести в глухий кут.

Матриці обертів. Наступний спосіб, який дозволяє описати переміщення твердого тіла в просторі - це матриця оберту. Задані матрицям оберти можна

					КНТЕУ-122-2018	Аркуш
						25
Зм.	Аркуш	№ документа	Підпис	Дата		

комбінувати звичайним множенням матриць. Застосувавши оберт до вектору можна помножити його на матрицю повороту [2].

Матрицю повороту розмірністю 3×3 можна визначити як матрицю перетворення тривимірного вектора положення в Евклідовому просторі, переводячи його координати з перевернутої системи відліку OUVW в абсолютну систему координат OXYZ. На рис.1.7 показані дві праві прямокутні системи координат: системи координат OXYZ з осями OX, OY, OZ і система OUVW з осями OU, OV, OW. Початки цих систем збігаються і розташовані в точці O. Система OXYZ фіксована в тривимірному просторі і прийнята за абсолютну, а система координат OUVW обертається щодо абсолютної системи OXYZ.

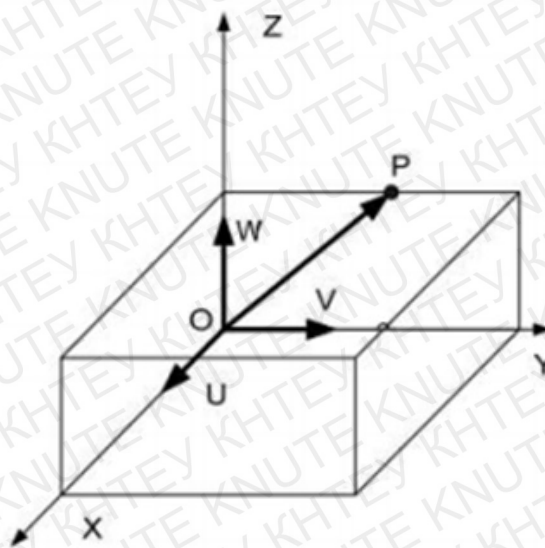


Рис.1.7 Абсолютна і пов'язана системи координат

Фізично система OUVW може розглядатися як пов'язана система координат. Це означає, що вона жорстко пов'язана з твердим тілом і рухається разом з ним. Будемо вважати (i_x, j_y, i_z) і (i_u, j_v, i_w) одиничними векторами, які орієнтовані вздовж осей систем OXYZ і OUVW відповідно. Деяку точку p в

					КНТЕУ-122-2018	Аркуш
						26
Зм.	Аркуш	№ документа	Підпис	Дата		

просторі можна охарактеризувати координатами щодо будь-якої із зазначених систем. Для наочності будемо вважати, що очка p фіксована і нерухома в системі відліку OUVW. Тоді в системах координат OUVW і OXYZ точка буде мати відповідно координати: $P_{xyz}=(i_x, j_y, i_z)$, $P_{uvw}=(i_u, j_v, i_w)$, де P_{xyz} і P_{uvw} характеризують стан однієї і тієї ж точки P відносно різних систем відліку. Завдання полягає в тому, щоб визначити матрицю R розмірністю 3×3 , яка перетворює координати P_{uvw} в координати вектора P в системі OXYZ після того, як система OUVW буде повернута, тобто наступним чином:

$$P_{xyz} = R * P_{uvw} \quad (1.3)$$

Фізична точка p обертається разом з системою координат OUVW. З визначення компонент вектора одержуємо:

$$P_{uvw} = P_u * i_u + P_v * i_v + P_w * i_w \quad (1.4)$$

Тут P_u , P_v , P_w - складові вектора P уздовж осей OU, OV, OW відповідно, або проекція вектора P на ці осі. З визначення скалярної похідної отримуємо

$$\begin{aligned} P_x &= P * i_x = i_x * i_u * P_u + i_x * j_v * P_v + i_x * k_w * P_w \\ P_y &= P * i_y = i_y * i_u * P_u + i_y * j_v * P_v + i_y * k_w * P_w \\ P_z &= P * k_z = k_z * i_u * P_u + k_z * j_v * P_v + k_z * k_w * P_w \end{aligned} \quad (1.5)$$

Матрична форма цього запису буде мати вигляд:

					КНТЕУ-122-2018	Аркуш
						27
Зм.	Аркуш	№ документа	Підпис	Дата		

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} i_x * i_u & i_x * j_v & i_x * k_w \\ j_y * i_u & j_y * j_v & j_y * k_w \\ k_z * i_u & k_z * j_v & k_z * k_w \end{bmatrix} * \begin{bmatrix} P_u \\ P_v \\ P_w \end{bmatrix} \quad (1.6)$$

З врахуванням цього матриця R приймає вигляд:

$$R = \begin{bmatrix} i_x * i_u & i_x * j_v & i_x * k_w \\ j_y * i_u & j_y * j_v & j_y * k_w \\ k_z * i_u & k_z * j_v & k_z * k_w \end{bmatrix} \quad (1.7)$$

Аналогічним чином, координати P_{uvw} можна отримати з координат P_{xyz}

$$P_{uvw} = Q * P_{xyz} \quad (1.8)$$

$$\begin{bmatrix} P_u \\ P_v \\ P_w \end{bmatrix} = \begin{bmatrix} i_x * i_u & i_x * j_v & i_x * k_w \\ j_y * i_u & j_y * j_v & j_y * k_w \\ k_z * i_u & k_z * j_v & k_z * k_w \end{bmatrix} * \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (1.9)$$

Даний спосіб позбавлений головного недоліку кутів Ейлера - «шарнірного замку». Шарнірний замок" - це назва проблеми, яка виникає при використанні ейлерови кутів через те, що кінцевий результат серії обертань залежить від порядку проміжних обертань, іноді трапляється, що обертання навколо однієї осі відображається на обертання навколо іншої осі. Навіть гірше, може бути неможливо обертати об'єкт навколо бажаної осі.

Однак такі матриці, також як і кути Ейлера не можна безпосередньо інтерполювати. Матриця повороту є відображенням повороту навколо довільного одиничного вектора на довільний кут.

					KHTEU-122-2018	Аркуш
						28
Зм.	Аркуш	№ документа	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 1

Отже, на сучасному етапі розвитку технологій, використання кіберрукавичок стає ще більш актуальним (в робототехніці, медицині, геймінгу, обладнання розумного дому, тощо).

Однак існують певні проблеми, що пов'язані з переміщенням твердих тіл. Некоректні данні призводять до необхідності попередньо обробляти вхідний сигнал.

Але використовуючи кути Ейлера, можуть виникати помилки через «шарнірний замок», а також їх не можна інтерполювати. Матриці обертів не мають головного недоліку кутів Ейлера - «шарнірного замку». Проте їх також не можна напряду інтерполювати.

					КНТЕУ-122-2018	Аркуш
						29
Зм.	Аркуш	№ документу	Підпис	Дата		

РОЗДІЛ 2. МАТЕМАТИКА ПЕРЕМІЩЕННЯ ТВЕРДИХ ТІЛ

2.1. Алгебра кватерніонів

Поняття кватерніон (від лат. Quaterni, по чотири) став вживати в своїх дослідженнях ірландський математик Вільям Гамільтон [3]. Анрі Пуанкаре писав про кватерніони: «Їх поява дала потужний поштовх розвитку алгебри. Виходячи від них, наука пішла шляхом узагальнення поняття числа, прийшовши до концепції матриці і лінійного оператора, що пронизує сучасну математику. Це була революція в арифметиці, подібна той, яку зробив Лобачевський в геометрії» [4].

Кватерніонні числа (від лат. Quaterni, по чотири) - система гіперкомплексних чисел, що утворює векторний простір розмірністю чотири над полем дійсних чисел.

Інтерес до кватерніонних чисел істотно зріс в останні два десятиріччя з приходом нового покоління теоретиків, які відчували в кватерніонів високий нерозкритий потенціал.

Кватерніони вживають для запису обертань в робототехніці, ігрових двигунах, програмного забезпечення для моделювання і взагалі скрізь, де не потрібні проблеми з кутами Ейлера або матрицями. Кватерніони розширюють поняття обертання в трьох вимірах на обертання в чотирьох вимірах. Вони вирішують проблему шарнірного замку і дозволяють виконати плавне і безперервне обертання. Насправді вони можуть вважатися доповненням до кутів обертання в сферичних координатах, широті, довготі і кута обертання.

					КНТЕУ-122-2018		
Зм.	Аркуш	№ документа	Підпис	Дата	Розробка програмного забезпечення для управління кіберрукавичкою Математика переміщення твердих тіл	Сторінка	Сторінок
Зав. кафедрою	Краскевич В.Є.					30	83
Керівник	Нагірна А.М.					Кафедра інформаційних технологій ОІ-2м-7	
Гарант	Краскевич В.Є.						
Розробив	Радовіченко О.О.						
Перевірив	Нагірна А.М.						

Ці чотиривимірні гіперкомплексні числа записуються виразами наступного виду [11]:

$$\Lambda = \lambda_0 i_0 + \lambda_1 i_1 + \lambda_2 i_2 + \lambda_3 i_3, \quad (2.1)$$

де $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ - довільні дійсні числа і є компонентами кватерніону Λ , а i_0, i_1, i_2, i_3 - кватерніонні одиниці.

Кватерніонне складання визначається за правилами звичайної векторної алгебри, тобто при складанні двох кватерніонів складаються їх відповідні компоненти:

$$\Lambda + M = \sum_{k=0}^3 (\lambda_k + \mu_k) * i_k \quad (2.2)$$

Кватерніонне множення позначається знаком « $\circ$ » і визначається так, що при множенні кватерніона на скаляр, множаться на цей скаляр всі його компоненти, тобто:

$$a \circ \Lambda = \Lambda \circ a = \sum_{k=0}^3 (\lambda_k) * i_k \quad (2.3)$$

а правила множення кватерніони одиниць визначаються наступною таблицею:

$$\begin{aligned} i_0 \circ i_0 &= i_0, i_0 \circ i_k = i_k \circ i_0, i_k \circ i_k = -1, k=1,2,3. \\ i_1 \circ i_2 &= i_3, i_2 \circ i_3 = i_1, i_3 \circ i_1 = i_2, \\ i_2 \circ i_1 &= -i_3, i_3 \circ i_2 = -i_1, i_1 \circ i_3 = -i_2. \end{aligned} \quad (2.4)$$

Відповідно до цих правил можна використовувати таку інтерпретацію

					КНТЕУ-122-2018	Аркуш
						31
Зм.	Аркуш	№ документа	Підпис	Дата		

кватерніонів, при якій елемент i_0 ототожнюється з речовинною одиницею, а елементи i_1, i_2, i_3 - з одиничними векторами, що утворюють в тривимірному просторі праву ортогональну трійку. Тоді за аналогією з комплексними числами кватерніон Λ можна представити у вигляді формальної суми скалярної частини λ_0 і векторної частини λ :

$$\Lambda = \lambda_0 + \lambda_1 i_1 + \lambda_2 i_2 + \lambda_3 i_3 = \lambda_0 + \lambda, \quad (2.5)$$

а правила (2.4) множення базисних векторів записати через скалярну і векторну похідну наступною формулою:

$$i_k \circ i_j = -(i_k \cdot i_j) + i_k \times i_j, \quad k, j=1, 2, 3. \quad (2.4^*)$$

Звідси, аксіоматизуючи властивість дистрибутивності

$$\Lambda \circ (M+N) = \Lambda \circ M + \Lambda \circ N, \quad (2.6)$$

отримуємо для кватерніонів похідні векторів λ і μ (кватерніонів з нульовою скалярною частиною) формулу

$$\lambda \circ \mu = -(\lambda \cdot \mu) + \lambda \times \mu \quad (2.7)$$

Формула для похідних двох «повноцінних» кватерніонів $\Lambda = \lambda_0 + \lambda$ і $M = \mu_0 + \mu$ приймає з урахуванням (2.2) наступний вид:

					КНТЕУ-122-2018	Аркуш
						32
Зм.	Аркуш	№ документа	Підпис	Дата		

$$\Lambda o M = \lambda_0 \mu_0 - (\lambda \cdot \mu) + \lambda_0 \mu + \mu_0 \lambda + \lambda \times \mu. \quad (2.8)$$

З наведених правил додавання і множення слідує, що алгебра кватерніонів включає в себе алгебру речових і комплексних чисел. При розгляді кватерніонів з нульовим векторними частинами ($\Lambda = \lambda_0$) отримуємо алгебру речовинних чисел, а в разі, коли векторна частина кватерніонів представлена одним виміром ($\Lambda = \lambda_0 + \lambda_1 i$) – алгебру комплексних чисел.

Зазначимо основні властивості множення кватерніонів.

1. Множення кватерніонів володіє дистрибутивними по відношенню до складання властивостями (2.6).

2. Множення кватерніонів асоціативно, тобто

$$\Lambda o Mo N = \Lambda o (Mo N) = (\Lambda o M) o N \quad (2.9)$$

3. Кватерніонне множення не має властивість коммутативності, тобто $\Lambda o M \neq Mo \Lambda$. Цей факт є наслідком некомутативності векторної похідної $\lambda \times \mu$, що входить до формулу (2.8). Рівність $\Lambda o M = Mo \Lambda$ має місце тільки в тому випадку, коли векторні частини співмножників колінеарна, тобто $\lambda \times \mu = 0$.

4. Скалярная частина твору кватерніонів не змінюється при циклічною перестановці співмножників, тобто

$$\text{sqal}(\Lambda o Mo N) = \text{sqal}(No \Lambda o M) \quad (1.10)$$

Властивість (2.9) встановлюється безпосередньою перевіркою з

					КНТЕУ-122-2018	Аркуш
						33
Зм.	Аркуш	№ документа	Підпис	Дата		

використанням аксіом додавання і множення, а властивість (2.10) впливає з властивості (2.9) і формули (2.8), в силу якої скалярна частина твору двох кватерніонів не залежить від порядку співмножників:

$$\text{sqal}[(\Lambda \circ M) \circ N] = \text{sqal}[N \circ (\Lambda \circ M)] \quad (2.11)$$

За аналогією з комплексними числами для кватерніона $\Lambda = \lambda_0 + \lambda$ визначається пов'язаний кватерніон $\bar{\Lambda}$:

$$\|\Lambda\| = \Lambda \circ \bar{\Lambda} = \lambda_0^2 + (\lambda \cdot \lambda) = \sum_{k=0}^3 \lambda_k^2 \quad (2.12)$$

Таким чином, норма кватерніона є скаляром і інваріантна по відношенню до вибору базису i_1, i_2, i_3 , в той час як компоненти $\lambda_1, \lambda_2, \lambda_3$ векторної частини кватерніона залежать від вибору базису.

Кватерніон Λ називається нормованим, якщо $\|\Lambda\| = 1$

Правила обчислення сполученого значення і норми від похідної двох кватерніонів легко встановлюються за допомогою формули множення (2.8). Так, для похідної двох кватерніонів $M = \mu_0 - \mu$ і $\Lambda = \lambda_0 - \lambda$ маємо

$$M \circ \Lambda = \lambda_0 \mu_0 - \lambda \cdot \mu - \lambda_0 \mu + \mu_0 \lambda + \lambda \times \mu = \Lambda \circ M \quad (2.13^*)$$

Звідси випливає, що поєднане значення від похідної двох кватерніонів дорівнює добутку їх пов'язаних значень, узятих в зворотному порядку:

$$\overline{(\Lambda \circ M)} = \bar{M} \circ \bar{\Lambda} \quad (2.13)$$

					КНТЕУ-122-2018	Аркуш
						34
Зм.	Аркуш	№ документа	Підпис	Дата		

Отримане співвідношення дозволяє в свою чергу знайти вираз для норми похідної двох кватерніонів:

$$\|\Lambda^\circ M\| = \Lambda^\circ M^\circ \overline{\Lambda^\circ M} = \|\Lambda\| \cdot \|M\|, \quad (2.14)$$

тобто норма похідної двох кватерніонів дорівнює добутку норм співмножників. Звідси випливає, що похідна нормованих кватерніонів є також нормований кватерніон.

Методом індукції легко показати, що правила (2.13) і (2.14) поширюються на випадок довільного числа співмножників, тобто

$$\overline{(\Lambda_1^\circ \dots \Lambda_n)} = \overline{\Lambda_n}^\circ \dots \overline{\Lambda_1} \quad (2.15)$$

$$\|\Lambda_1^\circ \dots \Lambda_n\| = \|\Lambda_1\| \cdot \dots \cdot \|\Lambda_n\| \quad (2.16)$$

Важливою властивістю алгебри кватерніонів є існування операції ділення, яка визначається як операція множення на зворотний кватерніон.

Кватерніонами, зворотним до Λ називається кватерніон Λ^{-1} , визначається з умови:

$$\Lambda^\circ \Lambda^{-1} = \Lambda^{-1}^\circ \Lambda = 1 \quad (2.17)$$

Вираз для зворотного кватерніона можна знайти безпосередньо з цього визначення, розглядаючи його як рівняння щодо невідомого Λ^{-1} . Помноживши обидві частини рівняння $\Lambda^\circ \Lambda^{-1} = 1$ на $\overline{\Lambda}$ зліва, отримаємо формулу

					КНТЕУ-122-2018	Аркуш
						35
Зм.	Аркуш	№ документа	Підпис	Дата		

З наведеного визначення випливає, що для векторів $\mathbf{r}_b$ і $\mathbf{r}_c$, що з'єднують довільну точку O тіла з точками B і C , виконуються умови:

$$(r_b - r_c)^2 = r_b^2 + r_c^2 - 2(r_b \cdot r_c) = \text{const} \quad (2.21)$$

$$(r_b \cdot r_c) = \text{const}, \quad (2.22)$$

Відзначимо, що наведене визначення твердого тіла не є повним. Всілякі дзеркальні відображення простору також задовольняють умові (2.22), але вони не включаються в математичну модель твердого тіла. Повне визначення твердого тіла описується умовою (2.22), доповненим умовою незмінності змішаного добутку векторів, що з'єднують довільні пари точок в тілі:

$$((r_b \times r_c) \cdot r_d) = \text{const}, \quad (2.23)$$

					КНТЕУ-122-2018	Аркуш
						37
Зм.	Аркуш	№ документу	Підпис	Дата		
З умови (2.22) випливає, що якщо в якийсь момент часу з деякими точками тіла зв'язати ортонормований базис $Oe_1e_2e_3$, то цей базис						

залишатиметься ортонормованим в будь-який інший час після цього, а розкладання вектора r довільної точки тіла в цьому базисі буде незмінно.

У свою чергу умова (2.23) означає незмінність взаємної орієнтації базисних векторів в тому сенсі, що права трійка векторів залишається правою, а ліва - лівою. Цим умовою дзеркальні відображення виключаються з математичної моделі твердого тіла.

Завдання визначення положення твердого тіла в деякій системі відліку полягає у визначенні такого набору параметрів, через які однозначно виражається положення кожної точки тіла [8].

Розглянемо довільний рух твердого тіла щодо системи відліку $Ai_1i_2i_3$, задається правим ортонормованим базисом з початком в точці A (рис.2.1). Виберемо в тілі жорстко пов'язаний з ним правий ортонормований базис $Oe_1e_2e_3$, з початком в точці O . В цьому базисі радіус-вектор r довільної точки тіла має незмінні компоненти r_k ($k=1, 2, 3$), а положення цієї точки в системі $Ai_1i_2i_3$ дається співвідношенням

$$R = R_0 + r = R_0 + \sum_{k=1}^3 r_k e_k, \quad (2.24)$$

Звідси випливає, що для однозначного визначення положення кожної точки тіла в системі $Ai_1i_2i_3$ досить задати положення базису $Oe_1e_2e_3$ щодо базису $Ai_1i_2i_3$.

					КНТЕУ-122-2018	Аркуш
						38
Зм.	Аркуш	№ документи	Підпис	Дата		



Рис.2.2 Рух базису $Oe_1e_2e_3$ щодо $Oi_1i_2i_3$

Описана схема лежить в основі всіх існуючих способів завдання положення твердого тіла. У кожному з них в якості трьох параметрів вибираються координати вектора R_O деякої точки O тіла, а різниця в способах

					КНТЕУ-122-2018	Аркуш
						39
Зм.	Аркуш	№ документа	Підпис	Дата		
полягає в різному виборі інших параметрів, які задають орієнтацію тіла						

(положення базису $Oe_1e_2e_3$ щодо $Oi_1i_2i_3$).

Наведемо основні способи завдання орієнтації твердого тіла.

Матриці напрямних косинусів.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad a_{kj} = (i_k \cdot e_j) \quad (2.25)$$

Елементи матриці (2.25) являють собою коефіцієнти в розкладанні ортов пов'язаного з тілом базису $Oe_1e_2e_3$ по ортам системи відліку $Oi_1i_2i_3$:

$$e_j = \sum_{k=1}^3 a_{kj} i_k, \quad j = 1, 2, 3 \quad (2.26)$$

З ортонормірованності базисів $Oe_1e_2e_3$ і $Oi_1i_2i_3$ слідує, що матриця напрямних косинусів ортогональна, тобто

$$AA^T = I, \quad (2.27)$$

де I - одинична матриця, «Т» - знак транспонування.

Умова (2.27) визначає 6 рівнянь, що зв'язують коефіцієнти матриці A , тому тільки три з дев'яти напрямних косинусів є незалежними. Звідси випливає, що положення твердого тіла з нерухомою точкою задається трьома незалежними параметрами, а положення вільного тіла - шістьма незалежними параметрами [2].

Матриця (2.25) задовольняє умові $\det(A) = 1$, впливаючому з того, що базиси $Oe_1e_2e_3$ і $Oi_1i_2i_3$ є правими трійками векторів. Зі співвідношень (2.25) і

					КНТЕУ-122-2018	Аркуш
						40
Зм.	Аркуш	№ документа	Підпис	Дата		

(2.26) слідує також, що має місце взаємно однозначна відповідність між положеннями тіла і елементами матриці напрямних косинусів.

Нехай r - вектор-стовпець, складений з координат r_j деякої точки тіла в пов'язаному з тілом базисі $Oe_1e_2e_3$. В силу співвідношень (2.26) вектор-стовпець r , складений з координат r_k цієї точки в системі відліку $Oi_1i_2i_3$, визначається формулою

$$r = Ar, \quad (2.28)$$

Якщо прийняти, що в початковий момент часу орти пов'язаного з тілом базису збігаються з однойменними ортами системи відліку ($e_k(0) = i_k$), то положення кожної точки тіла $r(t)$ буде визначатися через її початкове положення $r(0)$ формулою $r(t) = A(t)r(0)$. Звідси в силу ортогональності матриці A випливає, що рух твердого тіла можна розглядати як залежне від часу ортогональне перетворення в тривимірному просторі.

Кути кінцевого обертання. Положення твердого тіла з нерухомою точкою можна задавати також різними системами кутів - Ейлера, Крилова та ін. Суть такого опису ґрунтується на тому, що два правих ортонормованих базиса $Oe_1e_2e_3$ і $Oi_1i_2i_3$ можна поєднати послідовними поворотами навколо трьох некомпланарних осей на деякі кути [14].

Розглянемо систему кутів Ейлера (рис. 2.3). нехай базис $Oe_1e_2e_3$ займає довільне положення. Всі вектори i_k базису $Oi_1i_2i_3$ можна поєднати з базисними векторами e_k за допомогою наступних трьох поворотів:

1. Оберт навколо осі i_3 на кут ψ до суміщення вектора i_1 з лінією вузлів i_1 , тобто з лінією перетину площин векторів i_1, i_2 та e_1, e_2 .

					КНТЕУ-122-2018	Аркуш
						41
Зм.	Аркуш	№ документа	Підпис	Дата		

2. Поворот навколо лінії вузлів $\hat{i}_1$ на кут θ до суміщення орта i_3 з ортом e_3 .
3. Поворот навколо осі e_3 на кут φ до повного суміщення базисів.
- 4.

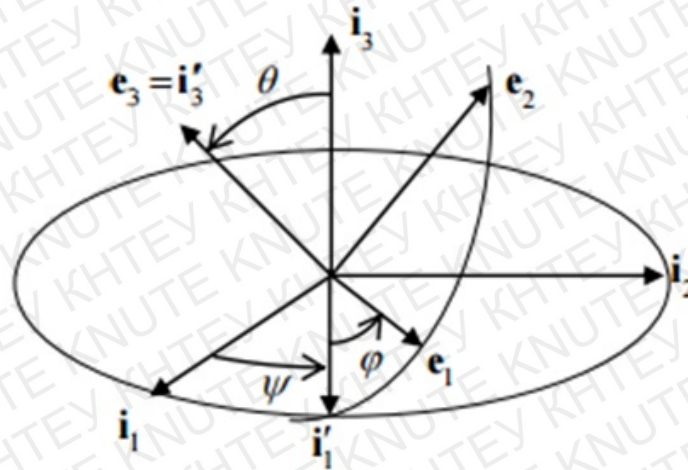


Рис.2.3 Поєднання всіх векторів i_k базису $Oi_1i_2i_3$ з базисними векторами e_k

Сукупність зазначених поворотів переводить базис $Oi_1i_2i_3$ в базис $Oe_1e_2e_3$ і являє собою послідовність поворотів на ейлерови кути ψ (кут прецесії), θ (кут нутації), і φ (кут власного обертання).

Зв'язок між напрямними косинусами (2.1.5) і кутами Ейлера визначається наступними співвідношеннями:

$$\alpha_{11} = \cos\psi \cos\varphi - \sin\psi \sin\varphi \cos\theta,$$

$$\alpha_{12} = -\cos\psi \sin\varphi - \sin\psi \cos\varphi \cos\theta,$$

$$\alpha_{13} = \sin\psi \sin\theta,$$

$$\alpha_{21} = \sin\psi \cos\varphi + \cos\psi \sin\varphi \cos\theta,$$

$$\alpha_{22} = -\sin\psi \sin\varphi + \cos\psi \cos\varphi \cos\theta,$$

$$\alpha_{23} = -\cos\psi \sin\theta,$$

					КНТЕУ-122-2018	Аркуш
						42
Зм.	Аркуш	№ документа	Підпис	Дата		

$$\alpha_{31} = \sin\varphi\sin\theta,$$

$$\alpha_{32} = \cos\varphi\sin\theta,$$

$$\alpha_{33} = \cos\theta.$$

Кватерніонний спосіб завдання орієнтації твердого тіла заснований на наступній теоремі.

Теорема 1. Довільний положення твердого тіла з нерухомою точкою задається нормованим кватерніоном Λ за формулами [13]:

$$e_k = \Lambda^\circ i_k \circ \bar{\Lambda}, \quad k=1, 2, 3 \quad (2.29)$$

При цьому кожному положенню тіла відповідає два значення кватерніона Λ , що відрізняються знаком.

Знайдемо кватерніон Λ з співвідношень (2.29), розглядаючи їх як систему рівнянь щодо невідомого Λ . Рішення системи (2.29) будемо шукати у вигляді:

$$\Lambda = \lambda_0 + \lambda, \quad \text{де } \lambda_0^2 + |\lambda|^2 = 1 \quad (2.29^*)$$

Введемо наступні позначення для різниць і сум однойменних базисних векторів:

$$r_k = (e_k - i_k), \quad s_k = (e_k + i_k); \quad k=1, 2, 3. \quad (2.30)$$

Для цих векторів виконуються наступні рівності:

$$(s_k \cdot r_k) = 0, \quad (s_j \cdot r_k) = -(s_k \cdot r_j) \quad (2.31)$$

					КНТЕУ-122-2018	Аркуш
						43
Зм.	Аркуш	№ документа	Підпис	Дата		

Записуючи рівняння (2.29) у вигляді $e_k \circ \Lambda = \Lambda \circ i_k$ отримаємо систему

$$\begin{cases} e_k \circ \lambda = \lambda \circ i_k \\ \lambda_0 e_k + e_k \times \lambda = \lambda_0 i_k - i_k \times \lambda \end{cases} \Rightarrow \begin{cases} r_k \cdot \lambda = 0 \\ \lambda_0 r_k = \lambda \times s_k \end{cases} \quad (2.32) \quad (2.33)$$

Розглядаючи спочатку випадок, коли базиси співпадають, тобто $i_k = e_k$, отримаємо з (2.33) $\lambda \times i_k = 0 \Rightarrow \lambda = 0$, звідки випливає $\Lambda = \lambda_0 = \pm 1$, що відповідає твердженням теореми.

У загальному випадку, коли базиси $i_1 i_2 i_3$ в базис $e_1 e_2 e_3$ не збігаються, знайдеться, принаймні два не рівних нулю вектора r_k і r_j . Нехай для визначеності $r_1 \neq 0$ і $r_2 \neq 0$. Тоді з співвідношень (2.32) отримаємо, що векторна частина шуканого рішення може бути записана у вигляді:

$$\lambda = x(r_1 \times r_2), \quad (2.34)$$

де x - деякий скаляр. Підставляючи цей вираз в (2.33) і враховуючи (2.31), отримаємо:

$$\begin{cases} \lambda_0 r_1 = -x s_1 \times (r_1 \times r_2) = -x r_1 (s_1 \cdot r_2) \\ \lambda_0 r_2 = -x s_2 \times (r_1 \times r_2) = -x r_2 (s_2 \cdot r_1) \end{cases} \Rightarrow \begin{cases} \lambda_0 = -x(s_1 \cdot r_2) \\ \lambda_0 = x(s_2 \cdot r_1) \end{cases}$$

В силу (2.31) отримані два рівняння тотожно збігаються, і з них і (2.34) випливає, що дані рішення системи (2.29) має вигляд:

$$\Lambda = x(s_2 \cdot r_1 + r_1 \times r_2) \quad (2.35)$$

З урахуванням умови нормування $\lambda_0^2 + |\lambda|^2 = 1$ остаточно отримуємо формулу:

					КНТЕУ-122-2018	Аркуш
						44
Зм.	Аркуш	№ документа	Підпис	Дата		

$$\Lambda = \pm \frac{r_1 \cdot s_2 + r_1 \times r_2}{\sqrt{(r_1 \cdot s_2)^2 + (r_1 \times r_2)^2}}, \quad (2.36)$$

яка визначає два нормованих кватерніона, що відрізняються один від одного знаком.

Перетворення $\Lambda^\circ()^\circ\bar{\Lambda}$, яке фігурує в формулах (2.29), називається приєднаним відображенням.

Таким чином, в кватерніонному способі для завдання орієнтації твердого тіла використовуються чотири параметри: $\lambda_0, \lambda_1, \lambda_2, \lambda_3$, які підпорядковані одному рівнянню зв'язку:

$$\|\Lambda\| = \lambda_0^2 + \lambda_1^2 + \lambda_2^2 + \lambda_3^2 = 1 \quad (2.37)$$

На підставі формул перетворення базисів (2.29) можна встановити зв'язок між компонентами кватерніонами в різних базисах. Нехай положення базису $E(Oe_1e_2e_3)$ відносно базису $I(Oi_1i_2i_3)$ задається кватерніоном Λ . Розглянемо деякий кватерніон R . Позначимо через R_k^E R_k^I його компоненти в базисі E і I відповідно, де $R_0^I = R_0^E = R_0$ (скалярна компонента кватерніона не залежить від базису). Тоді цей кватерніон можна записати у вигляді наступних двох виразів:

$$R = R_0 + \sum_I^3 R_k^I i_k = R_0 + \sum_I^3 R_k^E e_k \quad (2.38^*)$$

Звідси з урахуванням формул перетворення базисів (2.29) і рівності $\Lambda^\circ\bar{\Lambda} = 1$ отримаємо:

$$R_0 + \sum_I^3 R_k^I i_k = \Lambda_0 (R_0 + \sum_I^3 R_k^E i_k)^\circ \bar{\Lambda} \Rightarrow R = \hat{\Lambda}^\circ R^E \bar{\Lambda}, \quad (2.38)$$

					КНТЕУ-122-2018	Аркуш
						45
Зм.	Аркуш	№ документа	Підпис	Дата		

де R^I і R^E - вектори-стовпці, складені з компонент кватерніона R в базисі I і E відповідно.

Співвідношення (2.38) являє собою формулу перетворення компонент кватерніона при заміні базису. З неї випливає, що вектор-стовпець $\vec{r}$, складений з координат довільної точки тіла в базисі I , виражається через вектор стовпець r її координат в базисі E формулою:

$$\vec{r} = \Lambda_0 \circ r \circ \bar{\Lambda} \quad (2.39)$$

Надалі домовимось вважати початковим становищем твердого тіла таке його положення, коли орт пов'язаного з тілом базису збігаються з однойменними ортами системи відліку, тобто $e_k(0) = i_k$. Тоді формула (2.38) буде визначати зв'язок між кінцевим положенням $\vec{r}$ довільної точки тіла і її початковим становищем r [7].

Теорема 2 (теорема Ейлера про кінцевий повороті). Будь-яке становище твердого тіла з нерухомою точкою може бути отримано з початкового положення одним поворотом навколо осі $e = \lambda/|\lambda|$ на кут $\vartheta = 2 \arccos \lambda_0$, де $\Lambda = \lambda_0 + \lambda$ - нормований кватерніон, що задає положення тіла.

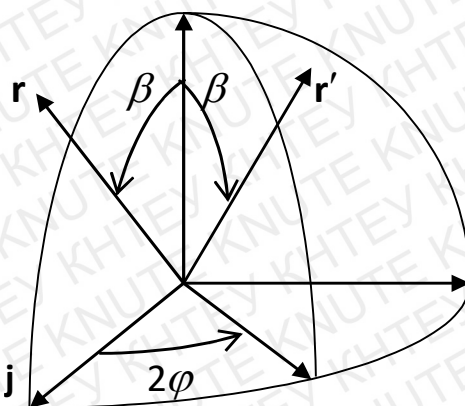


Рис 2.4. Поворот навколо осі

					KHTEU-122-2018	Аркуш
						46
Зм.	Аркуш	№ документа	Підпис	Дата		

Запишемо кватерніон Λ в тригонометричній формі:

$$\Lambda = \cos \varphi + \lambda \sin \varphi \mathbf{e} ; \cos \varphi = \lambda_0, \mathbf{e} = \lambda / |\lambda| \quad (2.40)$$

і досліджуємо перетворення (2.38). Доповнимо вектор $\mathbf{e}$ одиничними векторами $\mathbf{j}$ і $\mathbf{k}$ до правої ортогональної трійки ($\mathbf{e} \circ \mathbf{j} = \mathbf{k}$) таким чином, щоб вектор $\mathbf{r}$ виявився в площині векторів $\mathbf{e}$ і $\mathbf{j}$ (рис. 2.4). Тоді вектор $\mathbf{r}$ запишеться у вигляді:

$$\mathbf{r} = |\mathbf{r}|(\mathbf{e} \cos \beta + \mathbf{j} \sin \beta) \quad (2.38^*)$$

З огляду на, що з умови ортогональності векторів $\mathbf{e}$ і $\mathbf{j}$ слідує рівність $\mathbf{j} \circ \bar{\Lambda} = \Lambda \circ \mathbf{j}$ отримаємо з формули (2.38):

$$\dot{\mathbf{r}} = |\mathbf{r}|(\Lambda \circ \mathbf{e} \circ \bar{\Lambda} \cos \beta + \Lambda \circ \mathbf{j} \circ \bar{\Lambda} \sin \beta) = |\mathbf{r}|(\mathbf{e} \cos \beta + \Lambda^2 \circ \mathbf{j} \sin \beta) = |\mathbf{r}|(\mathbf{e} \cos \beta + (\mathbf{j} \cos 2\varphi + \mathbf{k} \sin 2\varphi) \sin \beta) \quad (2.38^{**})$$

З цього виразу випливає, що кінцеве положення $\dot{\mathbf{r}}$ кожної точки тіла виходить з початкового положення $\mathbf{r}$ поворотом навколо осі $\mathbf{e}$ на кут $\vartheta = 2\varphi = 2 \arccos \lambda_0$.

Надалі на підставі теореми 2 будемо говорити, що кватерніон Λ задає поворот з базису $\mathbf{I}$ в базис $\mathbf{E}$, якщо цей кватерніон пов'язує вектори цих базисів формулами (2.29). Зауважимо, що зворотне перетворення задається зворотним кватерніонів $\bar{\Lambda}$, оскільки формули цього перетворення мають вигляд:

$$\mathbf{i}_k = \bar{\Lambda} \circ \mathbf{e}_k \circ \Lambda, \quad k=1, 2, 3. \quad (2.41)$$

					КНТЕУ-122-2018	Аркуш
						47
Зм.	Аркуш	№ документа	Підпис	Дата		

Нехай кватерніон Λ задає поворот тіла з базису I в базис $\bar{I}$, а кватерніон M - поворот з базису $\bar{I}$ в базис I'' (Рис. 5). Найдём кватерніон N результуючого повороту з базису I в базис I'' . В силу (2.38) в результаті зазначених двох поворотів початкове положення r довільної точки тіла перетворюється в кінцеве положення r'' за формулою:

$$r'' = M \circ r \circ \bar{M} = M \circ \Lambda \circ r \circ \bar{\Lambda} \circ \bar{M} = N \circ r \circ \bar{N} \quad (2.42)$$

Звідси випливає, що кватерніон результуючого повороту визначається виразом:

$$N = M \circ \Lambda \quad (2.43)$$

У разі n послідовних поворотів, що задаються кватерніонами $\Lambda_1, \Lambda_2, \dots, \Lambda_n$, формула складання поворотів має вид:

$$\Lambda = \Lambda_n \circ \Lambda_{n-1} \circ \dots \circ \Lambda_1, \quad (2.44)$$

Формули (2.43) і (2.44) придатні для практичного використання в тому випадку, коли кватерніони складових поворотів задані своїми компонентами в одному і тому ж базисі. Якщо ж кватерніони складових поворотів задані в різних базисах, то виникає необхідність перетворення зазначених формул до такого виду, в якому векторні частини всіх співмножників будуть записані в ортах того базису, в якому потрібно знайти результуючий кватерніон (як правило, цим базисом є нерухомий базис I) [10].

Розглянемо випадок, коли складові повороти задаються параметрами

					КНТЕУ-122-2018	Аркуш
						48
Зм.	Аркуш	№ документа	Підпис	Дата		

Родріга-Гамільтона.

Параметрами Родріга-Гамільтона називаються компоненти кватерніона в базисі, що перетворюються цим кватерніоном. Надалі, щоб відрізнити параметри Родріга-Гамільтона від компонент кватерніона в інших базисах, будемо позначати їх символами із зірочкою (за винятком скалярних компонент, що не залежать від базису).

У задачі з двома поворотами (рис.2.5) параметрами Родріга-Гамільтона для першого і результуючого поворотів є компоненти Λ і N в базисі I , а для другого повороту – компоненти M в базисі I' [4]. Тому кватерніони цих поворотів виражаються через параметри Родріга-Гамільтона співвідношеннями:

$$\Lambda = \lambda_0 + \sum_{k=1}^3 \lambda_k \cdot i_k, \quad N = \eta_0 + \sum_{k=1}^3 \eta_k \cdot i_k, \quad M = \mu_0 + \sum_{k=1}^3 \mu_k \cdot i_k \quad (2.45)$$

У разі n послідовних поворотів матимемо

$$\Lambda_j = \lambda_{j0} + \lambda_j = \lambda_{j0} + \sum_{k=1}^3 \lambda_{jk} \cdot i_k^{(j-1)}, \quad i_k^{(0)} = i_k \quad (2.45^*)$$

Тут $i_k^{(j-1)}$ – орти базису I^{j-1} одержуваного в результаті перших $j - 1$ поворотів.

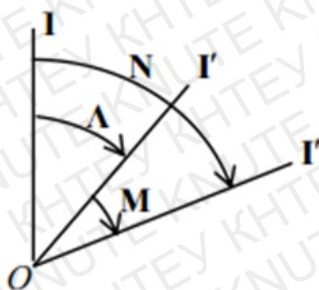


Рис.2.5 Задача з двома поворотами

					КНТЕУ-122-2018	Аркуш
						49
Зм.	Аркуш	№ документа	Підпис	Дата		

Оскільки кватерніони (2.45) задані в різних базисах, то формула (2.43) не може бути безпосередньо використана для обчислення результуючого кватерніона. Для отримання формули додавання поворотів в параметрах Родріга-Гамільтона кожному повороту поставимо у відповідність власний кватерніон, який визначається виразом:

$$\Lambda_j^* = \lambda_{j0} + \lambda_j^* = \lambda_{j0} + \sum_{k=1}^3 \lambda_{jk}^* i_k, \quad (2.46)$$

Цей кватерніон виходить з (2.45) заміною ортов базису $I^{(j-1)}$ ортами нерухомого базису I . Іншими словами, кожен власний кватерніон (2.46) являє собою заданий в нерухомому базисі вектор-стовпець, складений з параметрів Родріга-Гамільтона. При цьому для першого і результуючого поворотів маємо:

$$\Lambda_1 = \Lambda_1^*, \quad \Lambda = \Lambda^*$$

У задачі з двома поворотами (рис.2.5, 2.6) власні кватерніони визначаються виразами

$$\Lambda^* = \Lambda, \quad N^* = N, \quad M^* = \mu_0 + \mu^* = \mu_0 + \sum_{k=1}^3 \mu_k^* i_k \quad (2.47)$$

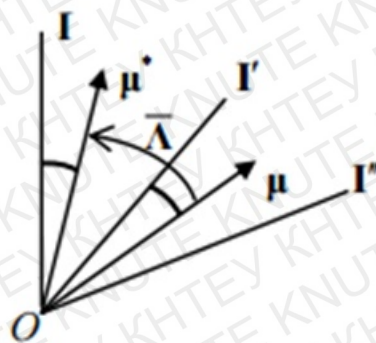


Рис.2.6 Задача з двома поворотами

					КНТЕУ-122-2018	Аркуш
						50
Зм.	Аркуш	№ документа	Підпис	Дата		

З формул перетворення базисів $\bar{I}_k = \Lambda^\circ i_k^\circ \bar{\Lambda}$ слідує співвідношення $M = \Lambda^\circ M^* \bar{\Lambda}$ підстановка якого в (2.40) дає:

$$N = M^\circ \Lambda = \Lambda^\circ M^* \bar{\Lambda}^\circ \Lambda = \Lambda^\circ M^* \rightarrow N^* = \Lambda^* \circ M^* \quad (2.48)$$

В отриманій формулі (2.47) всі кватерніони записані в ортах одного і того ж базису I, а їх компонентами є параметри Родріга-Гамільтона. Тому дана формула пов'язує параметри Родріга-Гамільтона результуючого повороту і складових поворотів.

Для випадку n поворотів формула складання в параметрах Родріга-Гамільтона набирає вигляду:

$$\Lambda = \Lambda^* = \Lambda_1^* \circ \Lambda_2^* \circ \dots \circ \Lambda_n^* \quad (2.49)$$

Звернемо увагу, що у формулі (2.49) власні кватерніони перемножуються в зворотному порядку по відношенню до порядку множення кватерніонів у формулі (2.43).

Використовуємо формулу (2.49) для визначення зв'язку між параметрами Родріга-Гамільтона, які задають положення твердого тіла, і кутами Ейлера (рис. 3).

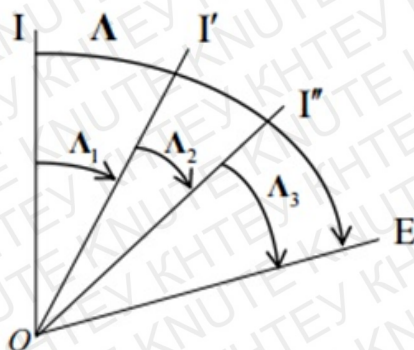


Рис.2.7 Результат трьох поворотів

					КНТЕУ-122-2018	Аркуш
						51
Зм.	Аркуш	№ документи	Підпис	Дата		

Кінцеве положення пов'язаного з тілом базису E виходить з початкового положення I в результаті трьох поворотів (Рис.2.7). Першим є поворот з положення I навколо осі i_3 на кут ψ . Кватерніон цього повороту Λ_1 і власний кватерніон Λ_1^* мають вигляд:

$$\Lambda_1 = \cos \frac{\psi}{2} + i_3 \sin \frac{\psi}{2} = \Lambda_1^* \quad (2.50)$$

Другий поворот здійснюється з положення I навколо осі i_1 на кут θ . Тому маємо:

$$\Lambda_2 = \cos \frac{\theta}{2} + i_1 \sin \frac{\theta}{2} ; \quad \Lambda_2^* = \cos \frac{\theta}{2} + i_1 \sin \frac{\theta}{2} \quad (2.51)$$

Для третього повороту, здійснюваного з положення I'' навколо осі i_3'' на кут φ , отримуємо:

$$\Lambda_3 = \cos \frac{\varphi}{2} + i_3'' \sin \frac{\varphi}{2} ; \quad \Lambda_3^* = \cos \frac{\varphi}{2} + i_3 \sin \frac{\varphi}{2} \quad (2.52)$$

Обчисливши тепер відповідно до формули (2.1.26) твір $\Lambda^* = \Lambda_1^* \circ \Lambda_2^* \circ \Lambda_3^*$, знайдемо зв'язок між параметрами Родріга-Гамільтона і кутами Ейлера:

$$\begin{aligned} \lambda_0^* &= \cos \frac{\theta}{2} \cos \frac{\psi+\varphi}{2}; & \lambda_1^* &= \sin \frac{\theta}{2} \cos \frac{\psi-\varphi}{2}; \\ \lambda_2^* &= \sin \frac{\theta}{2} \sin \frac{\psi-\varphi}{2}; & \lambda_3^* &= \cos \frac{\theta}{2} \sin \frac{\psi+\varphi}{2}; \end{aligned} \quad (2.53)$$

Співвідношення (2.1.27) дають можливість визначити вісь і кут кінцевого повороту твердого тіла як функції кутів Ейлера, а саме, напрямок осі кінцевого

					КНТЕУ-122-2018	Аркуш
						52
Зм.	Аркуш	№ документа	Підпис	Дата		

повороту по відношенню до нерухомого базису I визначається компонентами $\lambda_1^*, \lambda_2^*, \lambda_3^*$, а кут кінцевого повороту обчислюється за формулою:

$$\alpha = \arccos(\lambda_0^*) = 2 \arccos \left[\cos \frac{\theta}{2} \cos \frac{\psi + \varphi}{2} \right] \quad (2.54)$$

Окрім цього, кватерніони є числами, які дозволяють однозначно визначити оберт у трьохмірному просторі [12].

2.3 Можливості кватерніонів для розробки ПЗ

Викорстання кватерніонів для обертів та переміщення твердих тіл є найбільш ефективним. Але більшість розробників не вважають за потрібне розібратись, як з ними працювати, обирають матриці обертів чи кути Ейлера, незважаючи на їх явні недоліки. Для того щоб краще зрозуміти, як використовувати кватерніони і що це дійсно не важко, розглянемо приклад. Обертання за допомогою кватерніонів зводиться до множення чисел, що дуже просто в програмній реалізації.

В даному прикладі ми обертаємо кватерніон навколо кватерніона, який обертається навколо третього кватерніона, який обертається навколо четвертого кватерніона для того щоб продемонструвати простоту складних обертань через гіперкомплексні числа.

Реалізована функція Quaternion rotate (double x, double y, double z, double i, double j, double k, float angle), яка приймає координати обертання точки, координати вектора, навколо якого обертаємо точку і кут.

					КНТЕУ-122-2018	Аркуш
						53
Зм.	Аркуш	№ документа	Підпис	Дата		

```
private Quaternion rotate(
    double x, double y, double z,
    double i, double j, double k,
    float angle) {
    double phi = Math.PI/180.0*angle;
    double cosphi = Math.cos(phi/2.0);
    double sinphi = Math.sin(phi/2.0);

    Quaternion q = new Quaternion(cosphi, i*sinphi, j*sinphi, k*sinphi).unit();
    Quaternion p = new Quaternion(0, x, y, z);
    return q.mul(p.mul(q.inverse()));
}
```

Рис.2.8. Функція Quaternion rotate

Імпортовані класи, які ми будемо використовувати для вирішення даного прикладу, це: org.lwjgl.opengl.GL11. *, org.lwjgl.util.glu.GLU. *, java.nio.ByteBuffer, org.lwjgl.opengl. *, org.lwjgl.input. *, org.lwjgl. *.

Алгоритм вирішення:

Вхідні дані:

- 1) кут θ на який проводиться обертання.
- 2) координати x, y, z точки оберт.
- 3) координати i, j, k направляючого одиничного вектора, навколо якого відбувається обертання.

На виході алгоритм видає кватерніон, що містить координати точки, яка зробила оберт навколо заданого вектора за годинниковою стрілкою його напрямку.

					КНТЕУ-122-2018	Аркуш
						54
Зм.	Аркуш	№ документа	Підпис	Дата		

```

public class QuaternionRotation {
    public static final int WIDTH = 640;
    public static final int HEIGHT = 360;
    private int delta;

    public static void main(String[] args) {
        QuaternionRotation qr = new QuaternionRotation();
        qr.start();
    }

    private void start() {
        initDisplay();
        initOpenGL();
        updateDelta();
        while (!Display.isCloseRequested()) {
            render();
            processInput();
            Display.sync(60);
            Display.update();
            updateDelta();
        }
    }
}

```

Рис.2.9 Оберт навколо заданого вектора

Визначимо, яку структуру ми будемо вважати кватерніонами в подальших лістингах:

```

Struct Quaternion{
    float x;
    float y;
    float z;
    float w;
};
typedef struct Quaternion Quaternion;

```

Рис 2.10 Структура кватерніона

					КНТЕУ-122-2018	Аркуш
						55
Зм.	Аркуш	№ документа	Підпис	Дата		

Тепер запишемо приклад реалізації нашої бієкції для отримання кватерніона з кутів Ейлера. Якщо вам необхідно задати якийсь просте обертання, наприклад стартове, то наочніше додати кути Ейлера, ніж за допомогою лише кватерніонів. Отже, треба знайти бієкцію між кутами Ейлера і нормалізованими кватерніонами.

Для початку згадаємо що $p = [\sin(\alpha/2)u, \cos(\alpha/2)w]$, де u - уявна частина кватерніона і відповідна осі обертання, а w - дійсна частина кватерніона відповідна куту повороту.

```
Quaternion QuaternionFromEuler (Vector axis, float angle) {
    Quaternion result;
    float sinAngle;

    angle *= 0.5f;
    VectorNormalize(&axis);
    sinAngle = sin(angle);
    result.x = (axis.x * sinAngle);
    result.y = (axis.y * sinAngle);
    result.z = (axis.z * sinAngle);
    result.w = cos(angle);
    return result;
}
```

Рис.2.11 Бієкції для отримання кватерніона з кутів Ейлера

Трохи нижче буде реалізована функція перемноження для кватерніонів, яка дозволить повністю отримати реалізацію нашого відображення, а не тільки навколо конкретної осі.

Далі нам знадобитися отримати зв'язок між кватерніонами і матрицями повороту. Будемо вважати, що матриця трансформації у нас 4x4 при чому в додаткові рядок і стовпець виставлені 0, за винятком перетину там буде 1. Для початку отримаємо матрицю повороту з існуючого кватерніона. Скористаємося формулою:

					КНТЕУ-122-2018	Аркуш
						56
Зм.	Аркуш	№ документа	Підпис	Дата		

$$\begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.3.1)$$

Тоді реалізація такого співвідношення буде виглядати наступним чином:

```
void QuaternionToMatrix(float m[4][4], const Quaternion * quat)
{
    float wx, wy, wz, xx, yy, yz, xy, xz, zz, x2, y2, z2;
    x2 = quat->x + quat->x;
    y2 = quat->y + quat->y;
    z2 = quat->z + quat->z;
    xx = quat->x * x2;    xy = quat->x * y2;    xz = quat->x * z2;
    yy = quat->y * y2;    yz = quat->y * z2;    zz = quat->z * z2;
    wx = quat->w * x2;    wy = quat->w * y2;    wz = quat->w * z2;

    m[0][0]=1.0f-(yy+zz); m[0][1]=xy-wz;        m[0][2]=xz+wy;
    m[1][0]=xy+wz;        m[1][1]=1.0f-(xx+zz); m[1][2]=yz-wx;
    m[2][0]=xz-wy;        m[2][1]=yz+wx;        m[2][2]=1.0f-(xx+yy);

    m[0][3] = m[1][3] = m[2][3] = 0;
    m[3][0] = m[3][1] = m[3][2] = 0;
    m[3][3] = 1;
}
```

Рис.2.12 Матриця повороту з існуючого кватерніона

Тепер побудуємо зворотнє відображення аї матриця повороту, і $w^2 + x^2 + y^2 + z^2 = 1$, тоді:

					КНТЕУ-122-2018	Аркуш
						57
Зм.	Аркуш	№ документу	Підпис	Дата		

$$\begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} w^2 \\ x^2 \\ y^2 \\ z^2 \end{pmatrix} = \begin{pmatrix} a_{11} \\ a_{22} \\ a_{33} \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} w^2 \\ x^2 \\ y^2 \\ z^2 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{22} \\ a_{33} \\ 1 \end{pmatrix}$$

$$\text{sign}(w) = +1$$

$$\text{sign}(x) = \text{sign}(a_{32} - a_{23})$$

$$\text{sign}(y) = \text{sign}(a_{13} - a_{31})$$

$$\text{sign}(z) = \text{sign}(a_{21} - a_{12})$$

Тепер, коли ми розібралися, з API, які підтримують тільки матриці повороту, ми можемо працювати тільки з кватерніонами.

Для початку реалізуємо функцію множення кватерніонів.

```
Quaternion QuaternionMultiply(Quaternion * quat1, Quaternion * quat2) {
    Vector vector1, vector2, cross;
    Quaternion result;
    float angle;

    vector1.x = quat1->x;
    vector1.y = quat1->y;
    vector1.z = quat1->z;
    vector2.x = quat2->x;
    vector2.y = quat2->y;
    vector2.z = quat2->z;
    angle = ((quat1->w * quat2->w) - (VectorDot(vector1, vector2)));

    cross = VectorCross(vector1, vector2);
    vector1.x *= quat2->w;
    vector1.y *= quat2->w;
    vector1.z *= quat2->w;
    vector2.x *= quat1->w;
    vector2.y *= quat1->w;
    vector2.z *= quat1->w;

    result.x = (vector1.x + vector2.x + cross.x);
    result.y = (vector1.y + vector2.y + cross.y);
    result.z = (vector1.z + vector2.z + cross.z);
    result.w = angle;

    return result;
}
```

Рис 2.13 Множення кватерніонів

Зм.	Аркуш	№ документа	Підпис	Дата	КНТЕУ-122-2018	Аркуш
						58

Тепер отримаємо функцію зворотного кватерніона:

```
void QuaternionInvert(Quaternion * quat) {
    float length;

    length = (1.0f / ((quat->x * quat->x) +
                     (quat->y * quat->y) +
                     (quat->z * quat->z) +
                     (quat->w * quat->w)));

    quat->x *= -length;
    quat->y *= -length;
    quat->z *= -length;
    quat->w *= length;
}
```

Рис 2.14 Функція зворотного кватерніона

І нарешті, функція, яка дозволяє повернути вектор на оберт заданий кватерніонами:

```
Vector QuaternionMultiplyVector(Quaternion * quat, Vector * vector) {
    Quaternion vectorQuat, inverseQuat, resultQuat;
    Vector resultVector;

    vectorQuat.x = vector->x;
    vectorQuat.y = vector->y;
    vectorQuat.z = vector->z;
    vectorQuat.w = 0.0f;

    inverseQuat = *quat;
    QuaternionInvert(&inverseQuat);
    resultQuat = QuaternionMultiply(&vectorQuat, &inverseQuat);
    resultQuat = QuaternionMultiply(quat, &resultQuat);

    resultVector.x = resultQuat.x;
    resultVector.y = resultQuat.y;
    resultVector.z = resultQuat.z;

    return resultVector;
}
```

Рис 2.15 Функція, що дозволяє повернути вектор на оберт заданий кватерніонами

					КНТЕУ-122-2018	Аркуш
						59
Зм.	Аркуш	№ документа	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 2

Отже, кватерніон є чотирьохвимірним комплексним числом, що складається з трьох мнимох одиниць та однією реальною, які всі між собою є взаємно перпендикулярні. Саме тому їх можна представляти у вигляді одиничного вектора, що задає базис чотирьохмірного простору. Тому кватерніони можуть використовувати властивості як вектора, так і гіперкомплексного числа. Це дозволяє застосовувати до кватерніонів різні операції векторної графіки та комплексного аналізу.

З наведеного опису, можна визначити, що він відображає вільну вісь. А це значить, що він позбавлен основного недоліку кутів Ейлера – «шарнірного замку». Ще однією ключовою перевагою яка є у кватерніонах і не має у кутів Ейлера і напряду у матриць оберту - це інтерполяція. Вона дозволяє знайти найкоротший кут.

					КНТЕУ-122-2018	Аркуш
						60
Зм.	Аркуш	№ документу	Підпис	Дата		

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Опис інтерфейсу користувача програми RACCOON.Clip

Програма RACCOON.Clip призначена для:

- ✓ заміни комп'ютерної миші;
- ✓ заміна клавіатури;
- ✓ заміна трекінг систем, у тому числі кіберрукавичок;
- ✓ коригування чутливості зовнішніх пристроїв.

Головне вікно запущеної програми Raccoon.Clip:

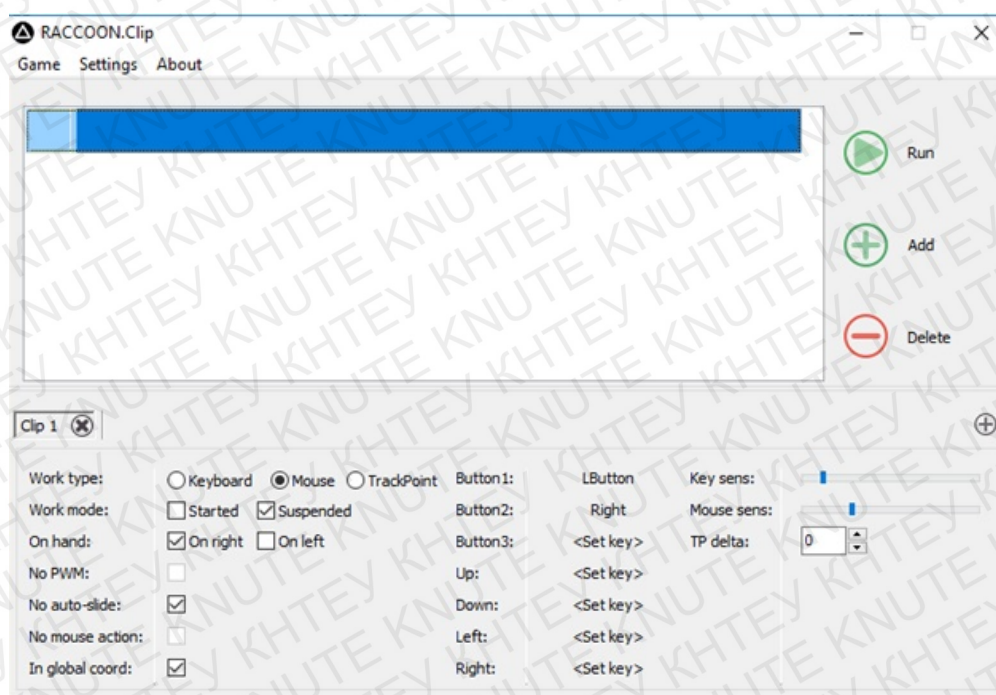


Рис.3 Головне вікно

					КНТЕУ-122-2018		
Зм.	Аркуш	№ документа	Підпис	Дата	Розробка програмного забезпечення для управління кіберрукавичкою	Сторінка	Сторінок
Зав. кафедрою	Краскевич В.Є.					61	83
Керівник	Нагірна А.М.					Кафедра інформаційних технологій ОІ-2м-7	
Гарант	Краскевич В.Є.						
Розробив	Радовіченко О.О.						
Перевірив	Нагірна А.М.				Розробка програмного забезпечення		

Обговоримо структуру верхньої частини вікна програми.

У самому верху вікна розташована назва Raccoon.Clip (1 на рисунку 31). У правому верхньому куті вікна є звичні три кнопки управління розміром вікна (2 на рисунку 31): закрити (косий хрестик), розгорнути в повний розмір / відновити зменшений розмір (квадратик або два "накладених" квадрата відповідно), згорнути (знак підкреслення):

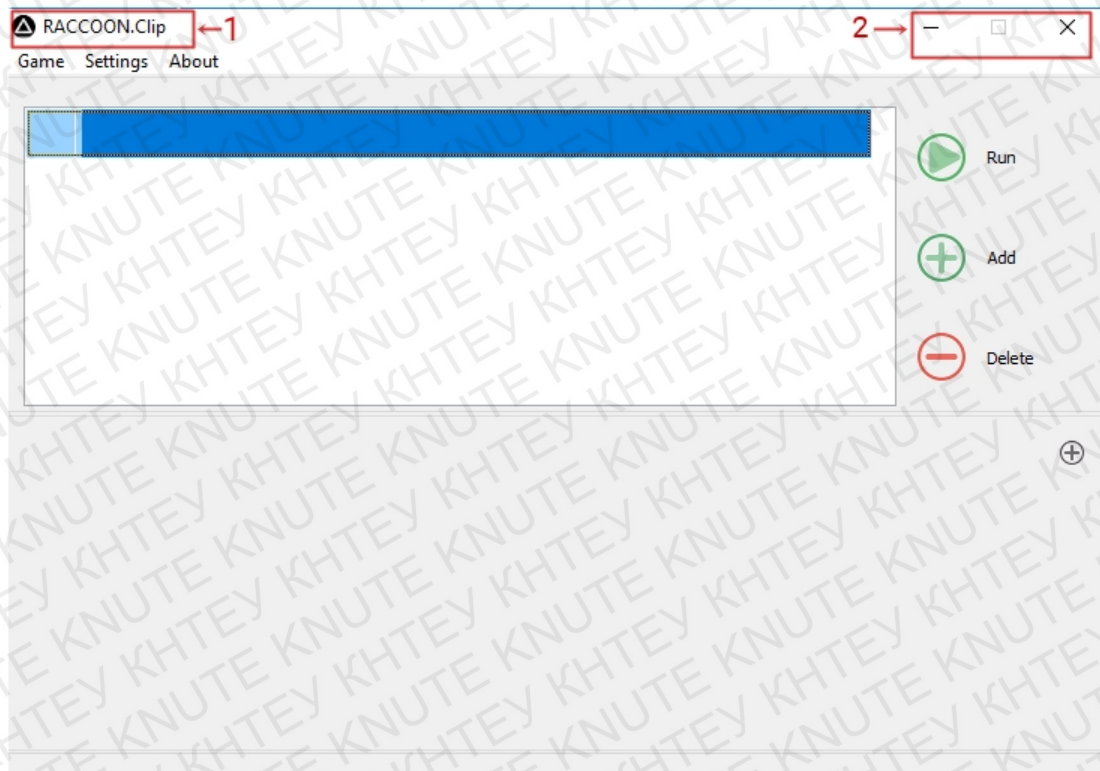


Рис.3.1 Рядок заголовку

Нижче рядка заголовка знаходиться рядок меню, яке складається з 3-х пунктів (1 на рисунку 3.2): Game, Settings та About.

					КНТЕУ-122-2018	Аркуш
						62
Зм.	Аркуш	№ документа	Підпис	Дата		

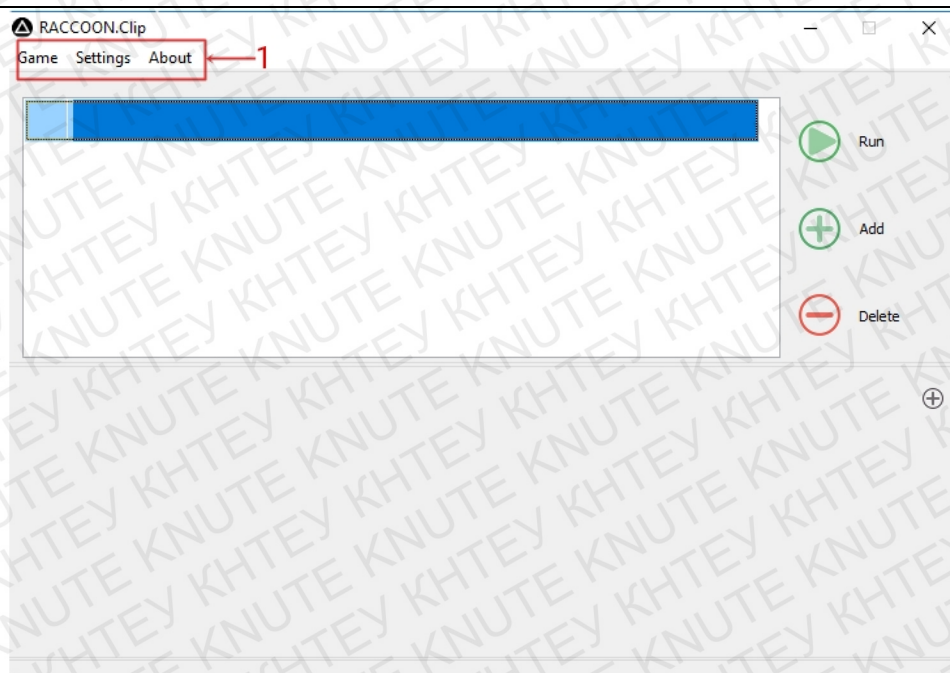


Рис.3.2 Рядок меню

При натисканні кнопки Game з меню, з'являється спливаюче меню, що складається з трьох пунктів (рис.3.3): Run, Add та Delete. Ці кнопки відповідають за додавання, видалення та запуску програм.

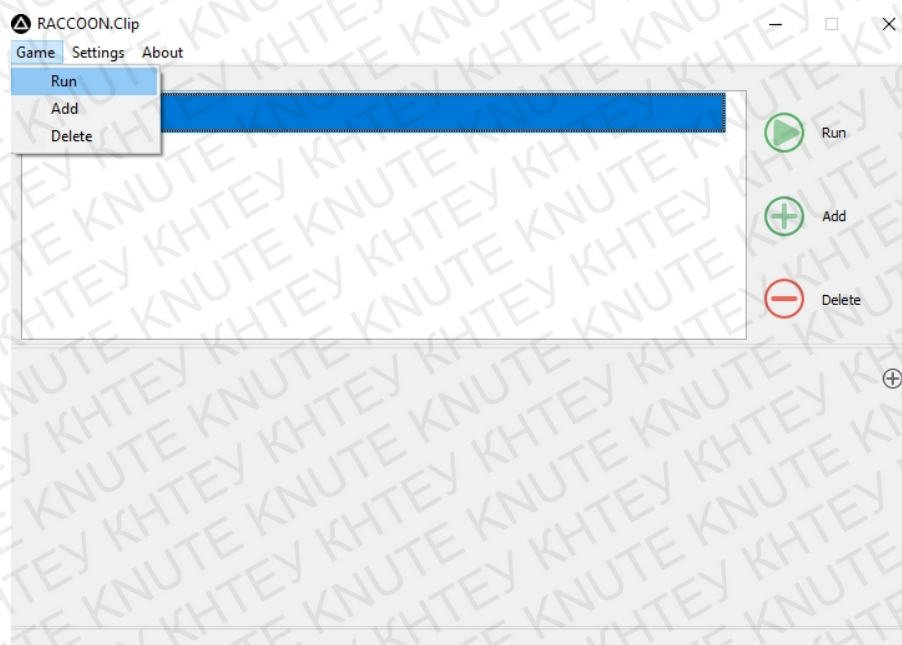


Рис.3.3 Пункт меню Game

					КНТЕУ-122-2018	Аркуш
						63
Зм.	Аркуш	№ документа	Підпис	Дата		

Наступною кнопкою в меню йде кнопка Settings. При її натисканні спливає меню з двох пунктів (рис.3.4): Clip info та Run game.

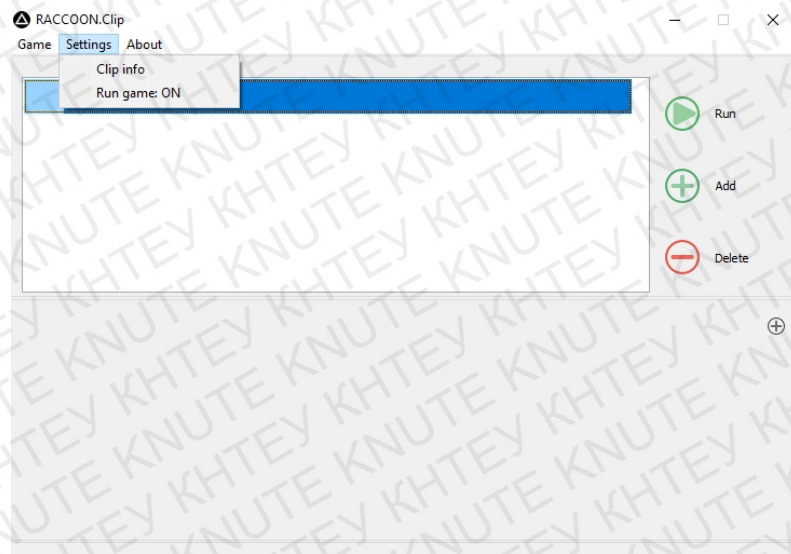


Рис. 3.4 Пункт меню Settings

Тут є важливою кнопка Clip info, оскільки при її натисканні ми отримуємо інформацію про підключення пристрою (кіберрукавички, кліпси, тощо) і координати її переміщення (рис.3.5).

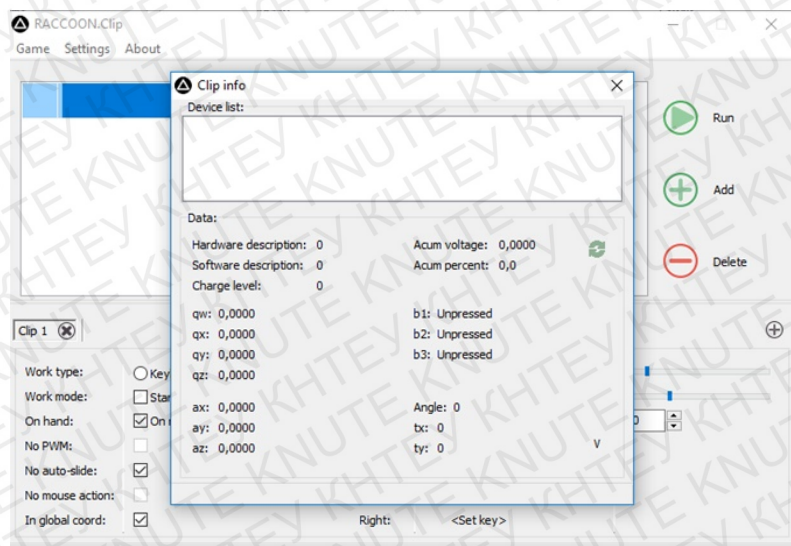


Рис 35. Підпункт меню Clip info

					КНТЕУ-122-2018	Аркуш
						64
Зм.	Аркуш	№ документа	Підпис	Дата		

Наступна кнопка в меню – About. При її натисканні впливає вікно, що ця програма призначена не для комерційного використання, а також, що за допомогою кнопок F1 та F2 можна активувати обробку потоків/ призупинити обробку потоків (рис. 3.6).



Рис.3.6 Пункт меню About

У центральній частині знаходиться область задач. Справа області задач знаходяться три кнопки, як і в пункті меню Game: Run, Add і Delete. В основну частину області задач за допомогою кнопки Add.

					КНТЕУ-122-2018	Аркуш
						65
Зм.	Аркуш	№ документа	Підпис	Дата		

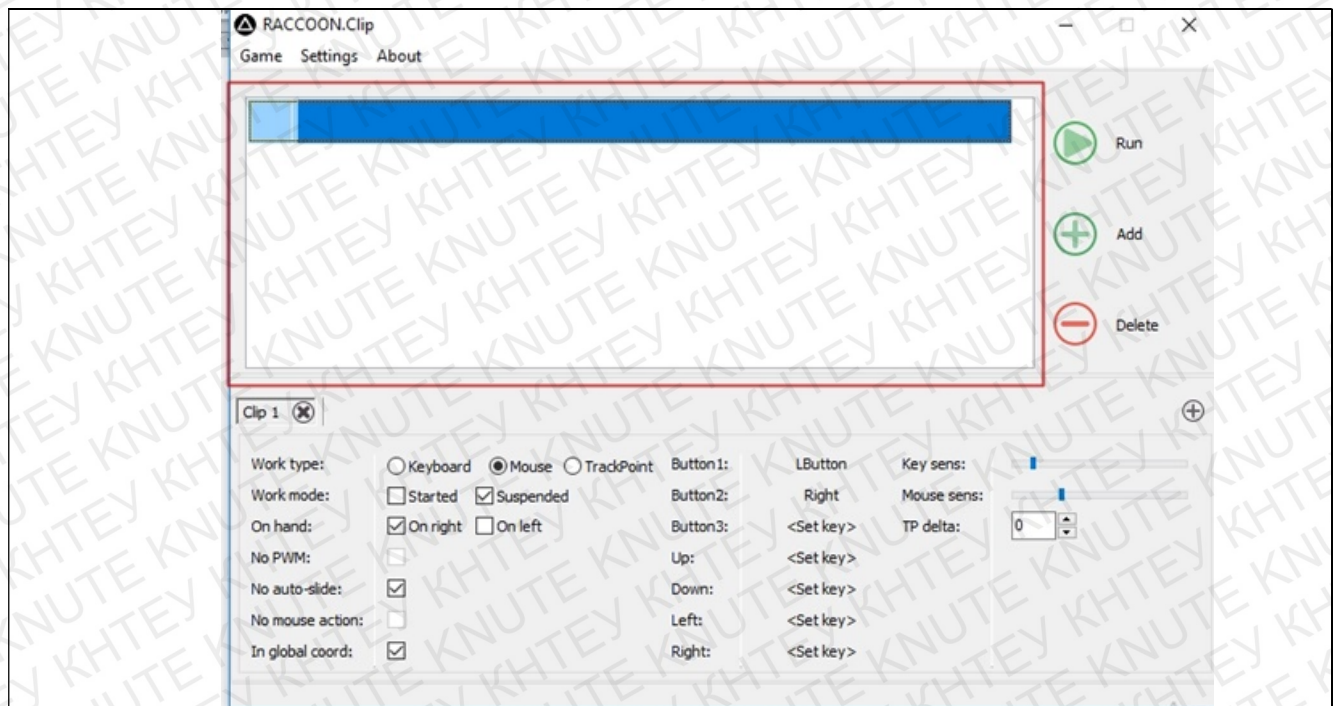


Рис. 3.7 Основне вікно області задач

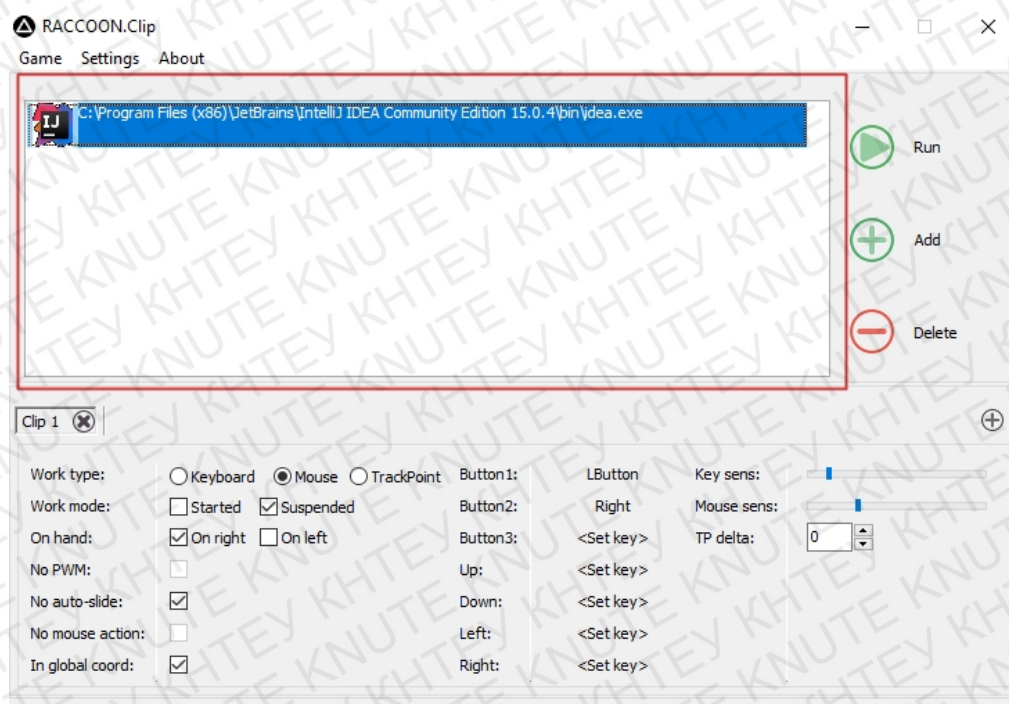


Рис.3.8 Основне вікно області задач після додавання обраної програми

					КНТЕУ-122-2018	Аркуш
						66
Зм.	Аркуш	№ документа	Підпис	Дата		

Нижче області задач знаходиться функціональна панель, яка має 17 настроюваних функціональних полів.

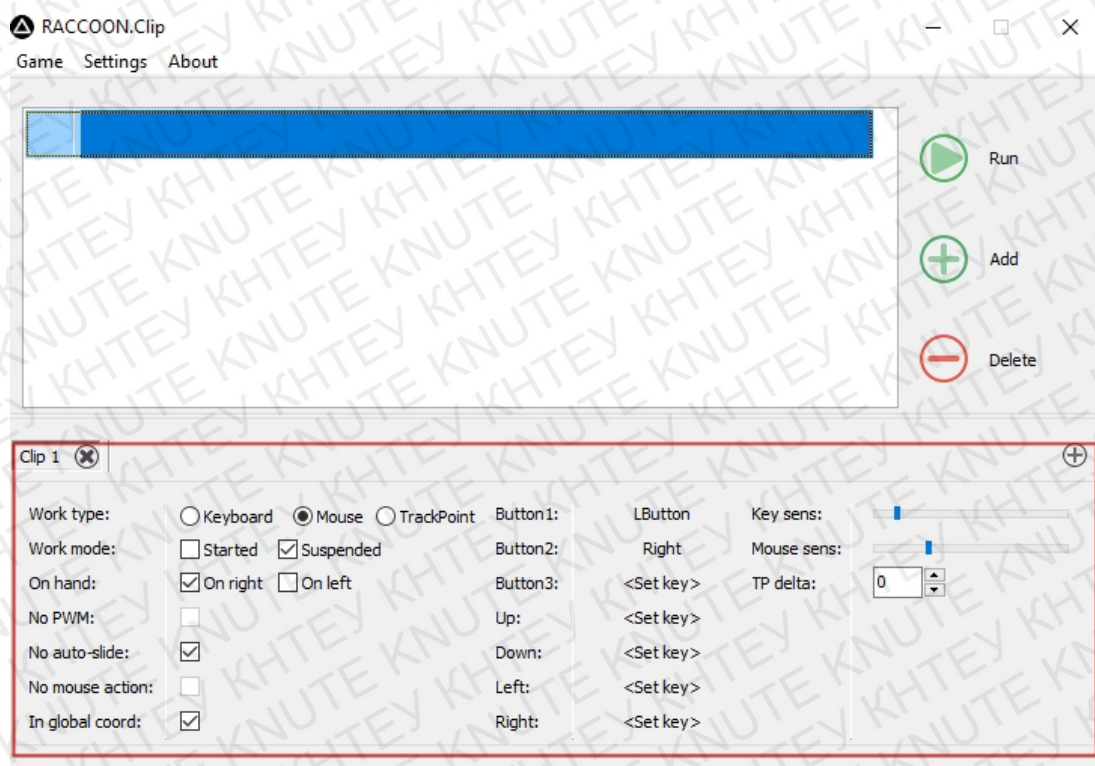


Рис.3.9 Функціональна панель

Перший критерій Wor type. Обираємо потрібне, в залежності від того, що в програмі використовується: клавіатура, мишка, трекінг система (кіберрукавичка).

Наступний критерій work mode. Обираємо режим роботи. Тобто як знаходиться пристрій в підвищеному стані, чи ні.

Далі критерій On hand. На якій руці знаходиться пристрій: на правій, на лівій чи одразу на двох руках.

Нижче знаходяться критерії: No PWM, no auto-slide, no mouse action, in global coord. Обираємо потрібні.

					КНТЕУ-122-2018	Аркуш
						67
Зм.	Аркуш	№ документа	Підпис	Дата		

Якщо на кіберрукавичці знаходяться кнопки, то їх також можна запрограмувати на потрібні дії (рис.3.10).

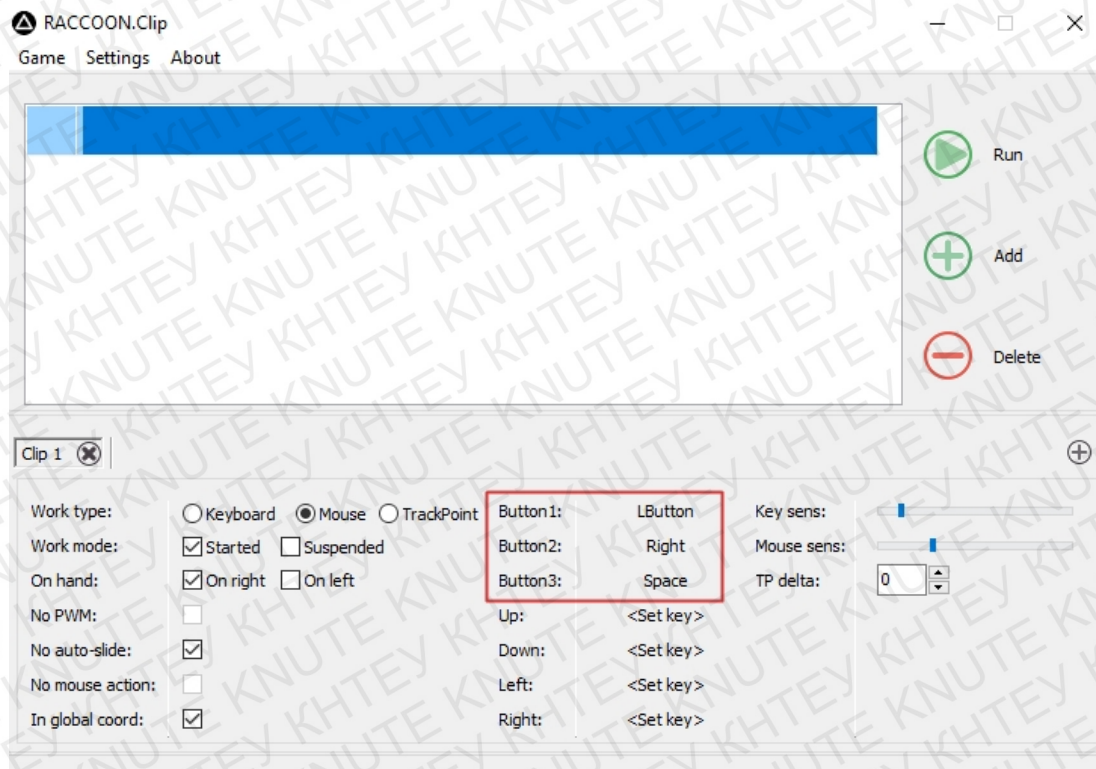


Рис.3.10 Функціональні кнопки на кіберрукавички

Нижче знаходиться блок, який допомагає запрограмувати, що буде відбуватись при нахилах руки вгору, вниз, вліво, вправо (рис.3.11). Це значить, що кожну дію руки можна запрограмувати.

					КНТЕУ-122-2018	Аркуш
						68
Зм.	Аркуш	№ документа	Підпис	Дата		

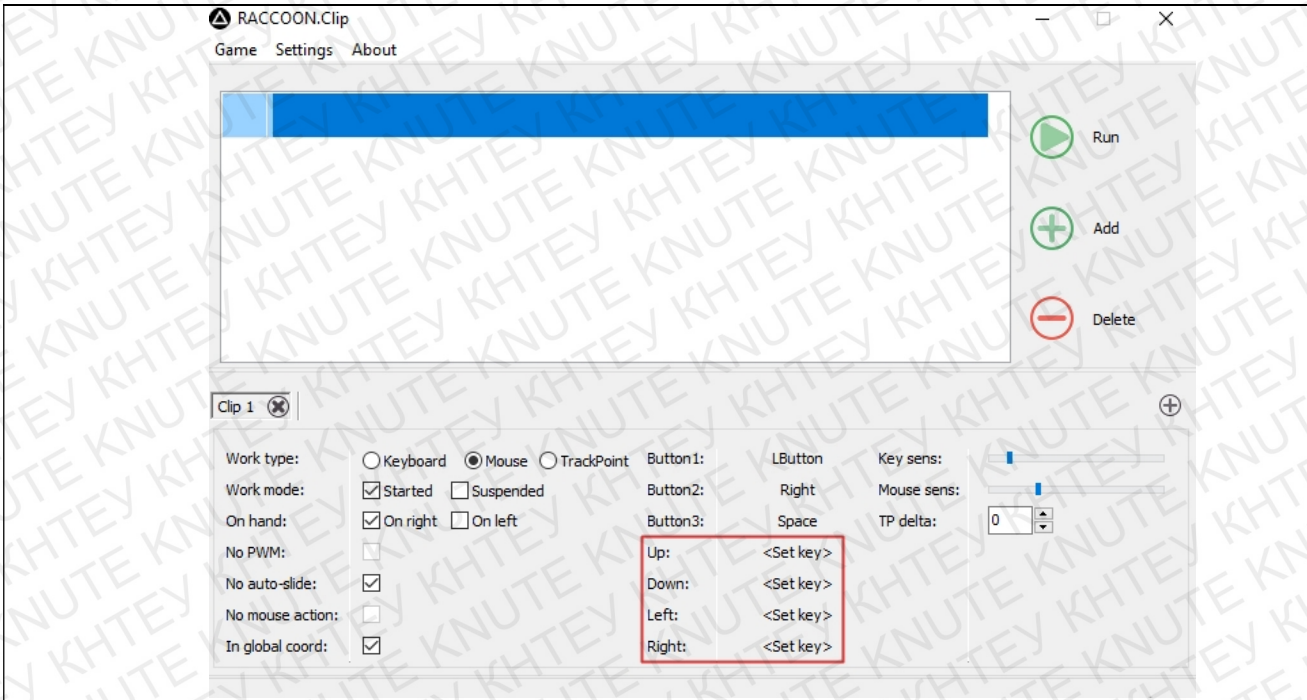


Рис.3.11 Блок дій кіберрукавичкою

І останній блок, який регулює чутливість (рис.3.12)

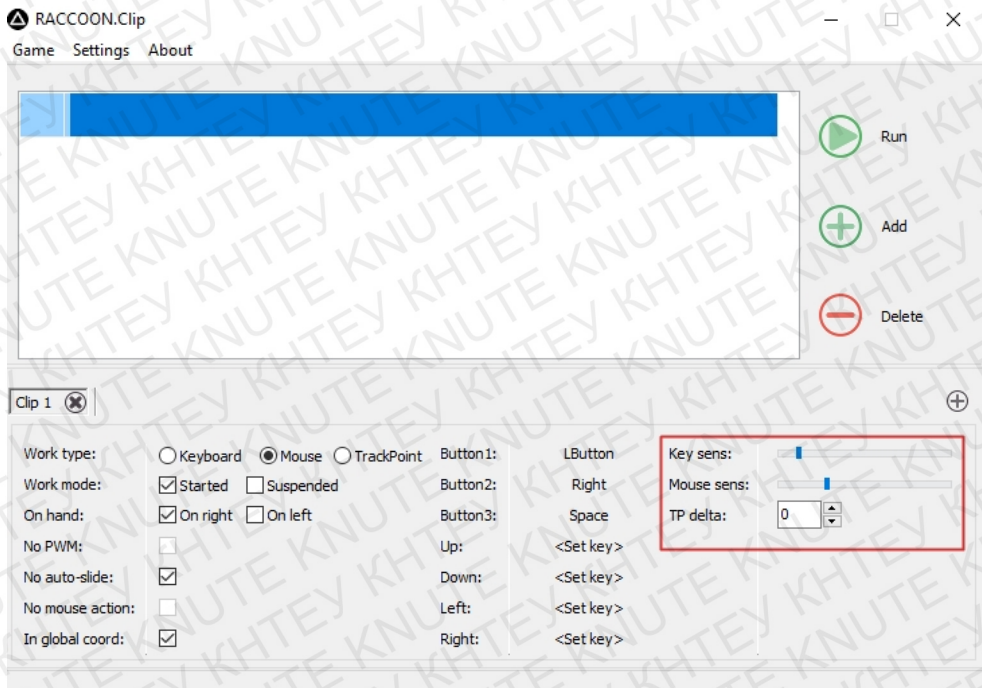


Рис.3.12 Блок чутливості

					КНТЕУ-122-2018	Аркуш
						69
Зм.	Аркуш	№ документу	Підпис	Дата		

3.2 Опис програмного коду

Розглянемо конвертування кватерніона у матрицю оберту. Часто для завдання обертань використовують тільки кватерніони одиничної довжини, але це не обов'язково і іноді навіть не ефективно. Різниця між конвертацією одиничного і неединичного кватерніонів становить близько 6-ти множень і 3-х складань, зате позбавить в багатьох випадках від необхідності нормувати (приводити довжину до 1) кватерніони. Якщо шматок коду критичний по швидкості і ви користуєтеся тільки кватерніонами одиничної довжини тоді можна скористатися фактом що норма його дорівнює 1 (рис.3.13).

```
/**
 * set the rotation to matrix
 */
inline void to_matrix( matrix33& m )const {
    float wx, wy, wz, xx, yy, yz, xy, xz, zz, x2, y2, z2;
    float s = 2.0f/norm(); // 4 mul 3 add 1 div
    x2 = x * s;   y2 = y * s;   z2 = z * s;
    xx = x * x2;  xy = x * y2;  xz = x * z2;
    yy = y * y2;  yz = y * z2;  zz = z * z2;
    wx = w * x2;  wy = w * y2;  wz = w * z2;

    m.m[0][0] = 1.0f - (yy + zz);
    m.m[1][0] = xy - wz;
    m.m[2][0] = xz + wy;

    m.m[0][1] = xy + wz;
    m.m[1][1] = 1.0f - (xx + zz);
    m.m[2][1] = yz - wx;

    m.m[0][2] = xz - wy;
    m.m[1][2] = yz + wx;
    m.m[2][2] = 1.0f - (xx + yy);
}
```

Рис.3.1.3 Конвертування кватерніона у матрицю оберту

Як ми бачимо, при конвертації використовується тільки множення і складання, що є безперечною перевагою на сучасних процесорах.

					КНТЕУ-122-2018	Аркуш
						70
Зм.	Аркуш	№ документа	Підпис	Дата		

```

inline void unit_from_matrix( const matrix33& mtx )
{
    typedef float mtx_elm[3][3];
    const mtx_elm& m = mtx.m;
    float tr = m[0][0] + m[1][1] + m[2][2]; // trace of matrix
    if (tr > 0.0f) { // if trace positive than "w" is biggest component
        set( m[1][2] - m[2][1], m[2][0] - m[0][2], m[0][1] - m[1][0], tr+1.0f );
        scale( 0.5f/(float)sqrt( w ) ); // "w" contain the "norm * 4"
    }else // Some of vector components is bigger
    if( (m[0][0] > m[1][1]) && (m[0][0] > m[2][2]) ) {
        set( 1.0f + m[0][0] - m[1][1] - m[2][2], m[1][0] + m[0][1],
            m[2][0] + m[0][2], m[1][2] - m[2][1] );
        scale( 0.5f/(float)sqrt( x ) );
    }else
    if ( m[1][1] > m[2][2] ) {
        set( m[1][0] + m[0][1], 1.0f + m[1][1] - m[0][0] - m[2][2],
            m[2][1] + m[0][2], m[2][0] - m[0][2] );
        scale( 0.5f/(float)sqrt( y ) );
    }else{
        set( m[2][0] + m[0][2], m[2][1] + m[1][2],
            1.0f + m[2][2] - m[0][0] - m[1][1], m[0][1] - m[1][0] );
        scale( 0.5f/(float)sqrt( z ) );
    }
}

```

Рис.3.14 Зворотна конвертація з матриці в кватерніон

Зворотне конвертація з матриці в кватерніон виконується не менш ефективно, але в підсумку ми отримаємо кватерніон неодиначної довжини. Звичайно, його можна нормалізувати, але можна обчислити довжину трохи ефективніше, а ось від квадратного кореня позбутися не вдасться(рис.3.14).

Досить прості перетворення між кватерніонами і Axis Angle представленням (рис.3.15).

					КНТЕУ-122-2018	Аркуш
						71
Зм.	Аркуш	№ документу	Підпис	Дата		

```

/**
 * create a unit quaternion rotating by axis angle representation
 */
inline void unit_from_axis_angle(const vector3& axis, const float& angle) {
    vector3 v(axis);
    v.norm();
    float half_angle = angle*0.5f;
    float sin_a = (float)sin(half_angle);
    set(v.x*sin_a, v.y*sin_a, v.z*sin_a, (float)cos(half_angle));
};

//-----
/**
 * convert a quaternion to axis angle representation,
 * preserve the axis direction and angle from -PI to +PI
 */
inline void to_axis_angle(vector3& axis, float& angle) const {
    float vl = (float)sqrt( x*x + y*y + z*z );
    if( vl > TINY )
    {
        float ivl = 1.0f/vl;
        axis.set( x*ivl, y*ivl, z*ivl );
        if( w < 0 )
            angle = 2.0f*(float)atan2(-vl, -w); //-PI,0
        else
            angle = 2.0f*(float)atan2( vl,  w); //0,PI
    }else{
        axis = vector3(0,0,0);
        angle = 0;
    }
};

```

Рис.3.15 Перетворення між кватерніонами і Axis Angle представленням

Як ми зазначали раніше довжина векторного і скалярного компонентів дорівнюють відповідно синусу і косинусу половини кута повороту. Часто зручно розглядати кватерніон як одиничний, помножений на якийсь скаляр, тоді багато операцій можна вивести для одиничних кватерніонів, а потім доповнити довжиною. Деякі операції виконуються з одиничними кватерніонами швидше, а деякі повільніше, наприклад, конвертація в матрицю одиничного кватерніона швидше, ніж довільного. Але конвертація з матриці в одиничний кватерніон більш повільна операція, ніж конвертування в непоодинокий. Саме для одиничних кватерніонів існують ефективні способи інтерполяції. Багато математичних бібліотек використовують тільки поодинокі кватерніони.

Нам необхідна більш точна інтерполяція обертань, ми можемо скористатися наближеними обчисленнями з дуже малою похибкою. Те ж саме

					КНТЕУ-122-2018	Аркуш
						72
Зм.	Аркуш	№ документа	Підпис	Дата		

можна сказати і про плавну інтерполяції сплайнами. Якщо є набір ключових орієнтацій можна плавно змінювати орієнтацію без різких змін в кутовій швидкості.

При реалізації інтерполяції рекомендуємо рознести інтерполяцію на два етапи - ініціалізація, яка може бути зроблена один раз для пари кватерніонів і, безпосередньо, інтерполяція параметра, це може помітно прискорити процес. Ініціалізація використовує досить ресурсномісткий acos , який і можна обчислити тільки раз для пари кватерніонів.

```
class q_lerper
{
    xxquaternion q1, q2;
public:

    inline void setup_from_unit(const xxquaternions& q1_, const xxquaternions& q2_){
        q1 = q1_;
        q2 = q2_;
        float inner = inner_product( q1, q2 );
        if(inner < 0 ) q2 = -q2;
        q2 -= q1;
    }

    inline void setup(const xxquaternions& q1_, const xxquaternions& q2_){
        q1 = q1_;    q1.normalize();
        q2 = q2_;    q2.normalize();
        setup_from_unit( q1, q2);
    }

    inline void interpolate( float t, xxquaternions& result_q){
        result_q = q1 + q2*t;
    }
};
```

Рис.3.16 Лінійна інтрополяція

					КНТЕУ-122-2018	Аркуш
						73
Зм.	Аркуш	№ документи	Підпис	Дата		

```

class q_slerper
{
    xxquaternion q1, q2;
    float omega;

public:
    inline void setup_from_unit(const xxquaternion& q1_, const xxquaternion& q2_) {
        q1 = q1_;
        q2 = q2_;
        float cos_omega = inner_product(q1, q2);
        if(cos_omega < 0)
        {
            cos_omega = -cos_omega;
            q2 = -q2;
        }
        if(cos_omega > 0.9999f) cos_omega = 0.9999f;
        omega = (float)acos( cos_omega );
        float inv_sin_omega = (float)(1.0/sin( omega ));
        q1.scale(inv_sin_omega);
        q2.scale(inv_sin_omega);
    }

    inline void setup(const xxquaternion& q1_, const xxquaternion& q2_) {
        q1 = q1_;
        q1.normalize();
        q2 = q2_;
        q2.normalize();
        setup_from_unit(q1, q2);
    }

    inline void interpolate( float t, xxquaternion& result_q ) {
        result_q = q1*(float)sin( (1.0 - t)*omega ) + q2*(float)sin( t*omega );
    }
};

```

Рис.3.17 Сферична лінійна інтрополяція

Цікавою операцією є створення кватерніона повороту з одного напрямку в інший по дузі, утвореної цими напрямками. Причому це буде найкоротша дуга між напрямками, звана "Shortest arc". Ще ефективніше можна отримати кватерніон, що обертає на дві таких дуги. Це кватерніон, утворений з векторного і скалярної похідної цих векторів, або кватерніон, що вийшов з твору двох кватерніонів, утворених з векторів. Поворот на "Shortest arc" можна отримати з суміші кватерніона подвійного повороту і ідентичності (нульового обертання). Просто склавши кватерніон подвійного повороту і кватерніона ідентичності, отримаємо кватерніон повороту по найкоротшій дузі (вони повинні бути однакової довжини).

Так, наприклад, можна орієнтувати камеру в напрямку об'єкта або об'єкт

					КНТЕУ-122-2018	Аркуш
						74
Зм.	Аркуш	№ документа	Підпис	Дата		

об'єкт у напрямку до камери (Sprite), легко реалізується контролер обертання за допомогою комп'ютерної миші, контролера, кіберрукавички (Track Ball).

За допомогою "Shortest arc" можна орієнтувати кіберрукавичку в напрямку повороту, причому її повороти будуть виглядати природно (розворот за найкоротшою дузі). Алгоритм дуже простий, на кожному кроці беремо попередній вектор напрямку. Будуємо "Shortest arc" від нього до поточного напрямку і повертаємо об'єкт на отриманий кватерніон. Якщо ми застосуємо повороти за допомогою "Shortest arc" при русі по безперервній кривій (наприклад, по сплайну), ми реалізуємо дуже корисний метод "parallel transport frame". Цей метод дає нам як би орієнтацію каната протягнутого по цій кривій і мінімізує скручування "twist" каната.

Для вирішення завдання інверсної кінематики, коли по заданому напрямку треба знайти необхідний поворот "Shortest arc" буде дуже доречним (наприклад у комп'ютерній грі).

```
inline void shortest_arc(const vector3& from, const vector3& to )
{
    vector3 c( from*to );
    set( c.x, c.y, c.z, from*to );
    normalize(); // if "from" or "to" is not unit, normalize it
    w += 1.0f; // reducing angle to halfangle
    if( w <= TINY ) // angle close to PI
    {
        if( ( from.z*from.z ) > ( -from.x*from.x ) )
            set( 0, from.z, - from.y, w ); //from*vector3(1,0,0)
        else
            set( from.y, - from.x, 0, w ); //from*vector3(0,0,1)
    }
    normalize();
}
```

Рис.3.18 Інверсна кінематика

					КНТЕУ-122-2018	Аркуш
						75
Зм.	Аркуш	№ документа	Підпис	Дата		

3.3 Аналіз отриманих результатів

У сучасний час кватерніони широко використовуються для вирішення задач пов'язаних з керуванням космічними апаратами, обчислювальної математики, при створенні тривимірної графіки тощо.

Ми створили алгоритми, які відстежують активність пристрою. Якщо програми не працюють або пристрій залишається в одному місці протягом тривалого часу, кілька процесів обробки даних можуть бути припинені, щоб знизити споживання енергії.

Майже усі сучасні кіберрукавички мають вбудований акселерометр і гіроскоп. Це дозволяє програмному забезпеченню зрозуміти відповідну орієнтацію і положення кіберрукавички без будь-яких додаткових камер або особливих умов освітлення. Відстеження зап'ястя здійснюється шляхом перенесення тривимірної сферичної системи координат в тривимірну матрицю. Після цього ми виберемо необхідну двовимірну проекцію з цієї матриці. Вихід на екран виконується завдяки алгоритмам.

У розробленій програмі були використані бібліотека для датчиків. Для того щоб отримати дані, достатньо просто запустити програму та пристрій. Збережемо отриманні дані у документ Microsoft Excel (рис. 3.19).

					КНТЕУ-122-2018	Аркуш
						76
Зм.	Аркуш	№ документа	Підпис	Дата		

	A	B	C	D	E	F	G	H	I
1	ax	ay	az	cx	cy	cz	gx	gy	gz
2	131	-2	7	-530	203	1	0,63	-2,5	2,16
3	136	1	3	-523	201	10	-30,05	16,97	-9,18
4	124	3	-54	-401	203	278	-34,92	3,41	-1,11
5	71	14	-109	-214	161	456	-28,52	-9,67	-4,94
6	53	28	-119	-122	65	457	-30,19	-2,43	-4,31
7	9	41	-124	116	-37	447	-25,67	-4,31	-2,71
8	-28	27	-124	235	48	436	-17,81	13,08	-13,98
9	-54	13	-117	372	125	371	-22,54	19,97	-2,5
10	-98	-21	-81	528	276	212	-43,41	24,97	7,03
11	-118	-39	-55	594	362	22	-21,15	4,8	1,46
12	-137	-41	-8	602	384	-140	-34,78	19,27	2,02
13	-117	-37	-29	575	345	-327	-18,64	7,26	3,97
14	-117	-32	52	532	319	-407	-23,03	8,42	8,9
15	-99	-24	70	455	274	-502	-15,93	19,97	0,7
16	-88	-18	84	394	241	-548	-10,43	0,9	3,2
17	-82	-8	93	352	195	-580	-12,59	5,22	11,76
18	-64	12	104	278	112	-618	-9,53	5,01	7,17
19	-61	19	107	263	88	-627	-2,23	3,55	-0,7
20	-59	23	109	261	70	-626	-7,17	4,94	5,36
21	-46	32	101	208	22	-637	-4,8	3,76	4,8
22	-40	35	78	173	6	-641	-4,73	0,42	1,6
23	-61	30	34	277	43	-619	72,63	-13,43	-27,9

Рис.3.19 Координати з датчиків

де координати ax, ay, az отримані з акселерометра, gx, gy, gz з гіроскопа, а cx, cy, cz дані з компаса (тут отримані координати при обертанні датчика на 180°).

Отримані координати представимо у вигляді тривимірному графіку.

					КНТЕУ-122-2018	Аркуш
						77
Зм.	Аркуш	№ документа	Підпис	Дата		

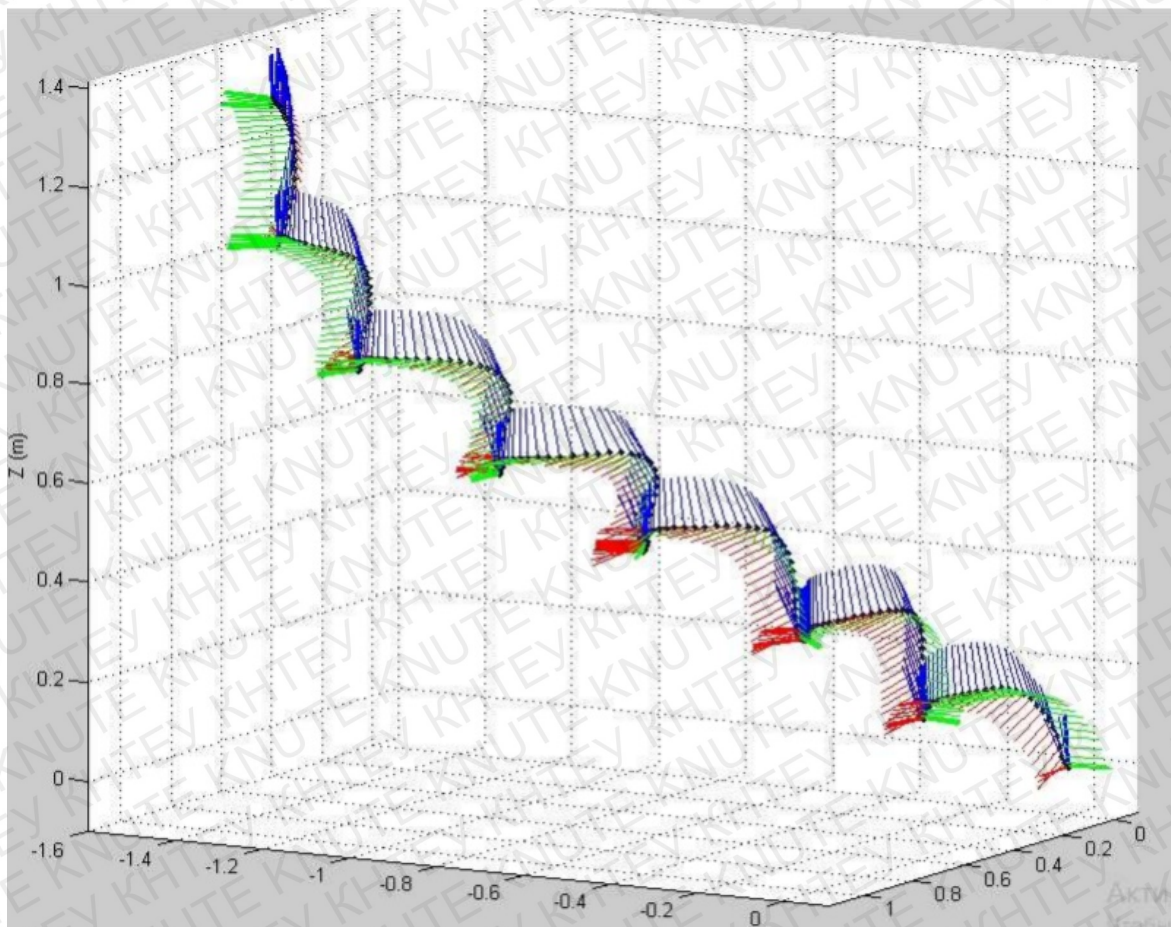


Рис 3.20 Анімований графік в тривимірному просторі

На рис.3.20 представлений графік пересування руки в тривимірному просторі. Графік показує анімацію пересування руки напівковговими рухами до низу. Кожен етап це рух руки людини. Як видно з графіка, це дуже точне відображення, так як видно детально кожен рух руки.

Як видно з графіку, кватерніони дають дуже наочний результат (визначення місця розташування об'єкта). З цього випливає, що кватерніони доцільно використовувати для переміщення твердих тіл.

Зм.	Аркуш	№ документа	Підпис	Дата	КНТЕУ-122-2018		Аркуш
							78

ВИСНОВКИ ДО РОЗДІЛУ 3

Проаналізувавши отримані результати можна зробити висновок, що кватерніони, які використовують матриці поворотів і кути Ейлера, є найбільш оптимальним рішенням, ніж використання кожного методу окремо.

Незважаючи на складність для розуміння, при роботі з поворотами кватерніони забезпечують кілька очевидних переваг в порівнянні з матрицями і кутами Ейлера.

- Інтерполяція кватерніонів надає спосіб плавної інтерполяції між орієнтаціями в просторі.
- Конкатенація поворотів за допомогою кватерніонів виконується швидше, ніж об'єднання поворотів, виражених в матричному вигляді.
- Для поодиначних кватерніонів зворотна величина повороту береться вирахуванням векторної частини кватерніона. Обчислення зворотної величини матриці повороту значно повільніше, якщо матриця не ортонормована.
- Перетворення кватерніонів в матриці трохи швидше, ніж для кутів Ейлера.
- Для опису повороту кватерніонами потрібно всього 4 числа (3, якщо вони нормалізовані. Речову частину можна обчислювати під час виконання програми), в той час як матрицями необхідно не менше 9 значень.

					КНТЕУ-122-2018	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		79

ВИСНОВКИ

У магістерській роботі були детально проаналізовані існуючі підходи, які використовують для розрахунку переміщення твердих тіл. Якщо порівнювати використання кватерніонів для переміщення твердих тіл з кутами Ейлера та матрицями, кватерніони мають низку переваг. Кути Ейлера для анімації переміщення твердого тіла можна взагалі не розглядати, вони мають «шарнірний замок», який відсутній у матрицях і кватерніонах. Запис кватерніона є трохи компактніший, ніж матриці, чотири числа проти дев'яти. Інтерполяція кватерніонів простіша. Рано чи пізно накопичуються помилки обчислень над дробом і потрібно від них позбавлятися, нормалізувати кватерніон теж простіше, ніж ортогоналізувати матрицю. Тому використовувати кватерніони набагато зручніше, особливо якщо буде велика кількість обертів.

Висунили і підтвердили гіпотезу щодо доцільності використання кватерніонів під час розрахунку переміщення будь-якого твердого тіла, зокрема кіберрукавички. А також був розроблений програмний додаток, який вже впровадила одна компанія, що свідчить про актуальність даного програмного забезпечення.

					КНТЕУ-122-2018		
Зм.	Аркуш	№ документа	Підпис	Дата	Розробка програмного забезпечення для управління кіберрукавичкою	Сторінка	Сторінок
Зав. кафедрою	Краскевич В.Є.					80	83
Керівник	Нагірна А.М.					Кафедра інформаційних технологій ОІ-2м-7	
Гарант	Краскевич В.Є.						
Розробив	Радовіченко О.О.						
Перевірив	Нагірна А.М.				Переміщення твердих тіл		

РЕКОМЕНДАЦІЇ

Висока точність одержуваних даних дозволить використовувати дану програму в різних областях:

- геймінг;
- проведення медичних операцій;
- розробка дрібної моторики рук;
- управління марсоходом;
- розумний дім;
- авіація;
- інтерактивні тренажери та реабілітація;
- vr/ar технології;
- управління роботами.

План подальшого покращення додатку:

- покращення інтерфейсу;
- додати функцію збору і відображення отриманих даних;
- довести програму до кросплатформеності.

					КНТЕУ-122-2018		
Зм.	Аркуш	№ документа	Підпис	Дата	Розробка програмного забезпечення для управління кіберрукавичкою Переміщення твердих тіл	Сторінка	Сторінок
Зав. кафедрою	Краскевич В.Є.					81	83
Керівник	Нагірна А.М.					Кафедра інформаційних технологій ОІ-2м-7	
Гарант	Краскевич В.Є.						
Розробив	Радовіченко О.О.						
Перевірів	Нагірна А.М.						

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Наукова література та періодичні видання

1. Баженов В. А. Будівельна механіка. Комп'ютерні технології та моделювання / В. А. Баженов, А. В. Перельмутер, О. В. Шишов. - М.: Изд-во СКАД СОФТ, Изд-во Асоціації будівельних вузів, 2014. - 912 с.
2. Борисов А. В. Динаміка твердого тіла / А. В. Борисов, І. С. Мамаєв. - М. : Іжевськ, НДЦ «Регулярна і хаотична динаміка», 2001.- 384 с.
3. Гамільтон У. Р. Про кватерніонів, або про нову систему уявних величин в алгебрі / У. Р. Гамільтон // Вибрані праці. Оптика. Динаміка. Кватерніони. - М.: Наука, 1994. - С. 345 - 391.
4. Єфремов А. П. Кватерніони: алгебра, геометрія і фізичні теорії. /А. П. Єфремов // Гіперкомплексні числа в геометрії та фізиці. - 2004. - № 1. -С. 111-127.
5. Каратаєв Е. А. Перетворення Гіперкомплексні чисел / Е. А. Каратаєв.- М.: Солон-прес, 2016. - 300 с.
6. Лазарєв Ю. Ф.Кінематика твердого тіла : навчальний посібник. – К.: НТУУ "КПІ", 2014. – 61 с.
7. Лорд Е. Е. Нова геометрія для нових матеріалів / Е. Е. Лорд, А. Л. Маккей, С. М. Ранганатана; пер. з англ. канд. хім. наук Л. П. Мезенцевої; під ред. акад. В. Я. Шевченко і д-ра фіз.-мат. наук В. Е. Дмитрієнко. - М.: Фізматліт, 2010. -263 с.
8. Людковська С. В. некомутативними квазіконформне інтегральні перетворення над кватерніонами і октоніонов / С. В. Людковська // Сучасна математика та її застосування. - Тбілісі, 2008. - Том 52. - С. 3-54.

					КНТЕУ-122-2018		
Зм.	Аркуш	№ документа	Підпис	Дата			
Зав. кафедрою	Краскевич В.Є.				Розробка програмного забезпечення для управління кібerrукавичкою	Сторінка	Сторінок
Керівник	Нагірна А.М.					82	83
Гарант	Краскевич В.Є.				Переміщення твердих тіл	Кафедра інформаційних технологій ОІ-2м-7	
Розробив	Радовіченко О.О.						
Перевірів	Нагірна А.М.						

9. Молоденков А. В. Про визначення орієнтації твердого тіла по його кутовий швидкості / А. В. Молоденков // Вісник Саратовського державного технічного університету. - 2007. - №1 (21). - Випуск 1. - С. 26-31.
10. Побегайло А. П. Застосування кватерніонів в комп'ютерній геометрії і графіку / А. П. Побегайло. - Мінськ: БДУ, 2010. - 216 с.
11. Садбері Ентоні. Кватерніони аналіз / Ентоні Садбері // Гіперкомплексні числа в геометрії та фізиці. - 2004. - № 2
12. Свешніков А. Г. Теорія функцій комплексної змінної / А. Г. Свешніков, А. Н. Тихонов. - М.: Фізматліт, 2010. - 336 с.
13. Челноков Ю. Н. кватерніонами моделі і методи динаміки, навігації та управління рухом / Ю. М. Челноков. - М.: Фізматліт, 2011. - 560 с
14. Чуб В. Ф. Про можливості застосування однієї системи Гіперкомплексні чисел в інерціальної навігації / В. Ф. Чуб // Известия Російської академії наук. Механіка твердого тіла. - 2002. - № 6. - С. 3-23.

Інтернет джерела

15. <https://investgazeta.info/news/item/1077-sformirovana-ukrainskaya-sektsiya-v-ramkakh-ses-2017>
16. <http://btl.startup.network/e785/>
17. <https://startup.ua/news/5-0-sbornaya-startapov-ukrainy-s-razgromnym-schetom-pobedila-sbornuyu-polshi>
18. <https://www.youtube.com/watch?v=NZ3683RYBHU>
19. <https://www.youtube.com/watch?v=Q7MYNtztKI4>

					<i>КНТЕУ-122-2018</i>	<i>Аркуш</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>		83