

УДК 004.75

DOI 10.30837/bi.2019.2(93).09

**Л.Е. Чала¹, Д.І. Карпенко², О.О. Берестовий³**

¹ Кандидат технічних наук, доцент кафедри штучного інтелекту,
Харківський національний університет радіоелектроніки,
larysa.chala@nure.ua, ORCID iD: 0000-0002-9890-4790

² Студент кафедри штучного інтелекту, Харківський національний університет радіоелектроніки,
daniil.karpenko@nure.ua

³ Студент кафедри штучного інтелекту, Харківський національний університет радіоелектроніки,
oleksandr.berestovyi@nure.ua

РОЗРОБКА ІНТЕРАКТИВНОГО МУЗЕЙНОГО ДОДАТКУ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

Наведено результати розробки інтерактивного музейного додатку з використанням технології доповненої реальності, що базується на інструментах середовища Vuforia та графічного ядра Unity. Додаток дає можливість здійснювати аналіз та розпізнавання реальних об'єктів (наприклад, музейних експонатів), які раніше були додані в сервіси Vuforia для реєстрації зображень. Додаток протестовано для низки різноманітних мобільних пристроїв на платформах iOS та Android. Визначено перспективи використання запропонованого додатку.

ДОПОВНЕНА РЕАЛЬНІСТЬ, ІНТЕРАКТИВНИЙ ДОДАТОК, VUFORIA, FAST-ДЕТЕКТОР, ЯКІСНЕ НАВЧАННЯ

Приведены результаты разработки интерактивного музейного приложения с использованием технологии дополненной реальности, основанной на инструментах среды Vuforia и графического ядра Unity. Приложение дает возможность осуществлять анализ и распознавание реальных объектов (например, музейных экспонатов), которые ранее были добавлены в сервисы Vuforia для регистрации изображений. Приложение протестировано для ряда различных мобильных устройств на платформах iOS и Android. Определены перспективы использования предложенного приложения.

ДОПОЛНЕННАЯ РЕАЛЬНОСТЬ, ИНТЕРАКТИВНОЕ ПРИЛОЖЕНИЕ, VUFORIA, FAST-ДЕТЕКТОР, КАЧЕСТВЕННОЕ ОБРАЗОВАНИЕ

The results of the development of an interactive museum application using augmented reality technology based on tools from the Vuforia environment and the Unity graphic core are presented. The application makes it possible to analyze and recognize real objects (for example, museum exhibits) that were previously added to Vuforia services for registering images. The application has been tested for different mobile devices on iOS and Android platforms. The prospects for using the proposed application are determined.

AUGMENTED REALITY, INTERACTIVE APPLICATION, VUFORIA, FAST-DETECTOR, QUALITY EDUCATION

Вступ

Технології доповненої реальності (AR — augmented reality) — це технології, що дозволяють доповнювати зображення реальних об'єктів різними об'єктами комп'ютерної графіки, а також поєднувати зображення, отримані від різних джерел [1]. Доповнена реальність, на відміну від віртуальної, лише вносить окремі штучні елементи в сприйняття світу реального. Доповнена реальність додає до відчуттів користувачів з реального світу уявні об'єкти, зазвичай допоміжно-інформативного характеру. AR — технології мають забезпечувати додаткам, що створюються на їх основі такі властивості як: комбінування реального та віртуального світу; інтерактивність; тривимірне представлення об'єктів [2].

До найбільш перспективних галузей використання технологій доповненої реальності, в першу чергу, слід віднести: створення тренажерних систем (в дистанційному освітньому процесі та в промисловості); проектування та дизайн; картографія і туристична галузь; використання QR кодів тощо [3–7].

Наприклад, у туристичній галузі доповнена реальність отримала актуальність в зв'язку з широким

розповсюдженням мобільних пристроїв. Так, подібні системи можуть ідентифікувати навколишні об'єкти (зокрема об'єкти з музейних колекцій), дозволяючи користувачу оперативно орієнтуватися в навколишньому просторі. У зв'язку з цим актуальною є задача створення додатків, які дозволяли б отримати додаткову інформацію про музейний об'єкт після наведення на нього камери телефону або іншого мобільного пристрою.

Такі додатки мають здійснювати функцію автоматичної ідентифікації зображень в відеопотоці з можливістю подальшої прив'язки до них інтерактивних об'єктів.

Перспективним слід вважати використання для реалізації цієї функції платформи Vuforia [8, 9], що є програмним забезпеченням для мобільних пристроїв, яке дозволяє створювати додатки доповненої реальності. Воно використовує технологію комп'ютерного зору для того, щоб розпізнавати і відстежувати плоскі зображення і прості 3D-об'єкти в режимі реального часу. Ця можливість реєстрації зображень дозволяє визначати розташування й орієнтації віртуальних

об'єктів, таких як 3D-моделі, в реальному світі, коли вони розглядаються через камеру мобільного пристрою. Положення і орієнтація віртуального об'єкта відстежується в реальному часі, так що точка зору глядача на об'єкт співвідноситься з їх точкою зору на зображення, що створює ілюзію належності віртуального об'єкту до реальної сцени. Vuforia надає можливість використовувати бібліотеки OpenCV (ComputerVision), які обгорнуті в зручний інтерфейс для розробників на C++, Java, Objective-C і .Net через інтеграцію з міжплатформенним середовищем розробки комп'ютерних ігор Unity. Це дає нам можливість використовувати системи когнітивного розвитку, які пов'язані з аналізом реальних образів.

У даній роботі досліджується завдання розробки інтерактивного музейного додатку з використанням технології доповненої реальності, що базується на інструментах середовища Vuforia та дає можливість аналізувати і розпізнавати реальні об'єкти, які раніше були додані в один із сервісів Vuforia для реєстрації зображень (маркерів). В результаті система надає користувачам дані про відповідність реального об'єкта і маркера, внаслідок чого виникає можливість зміни віртуального оточення реальних об'єктів. Наприклад, користувач може завантажити в веб сервіс зображення якої-небудь картини і використовуючи розроблений додаток, визначити збіг картини з частиною реального світу, що потрапила в фокус камери, з маркером, який розташований в сервісі, і в разі необхідності здійснити певні маніпуляції в віртуальному світі, наприклад, вивести опис про цю картину на екран.

1. Принципи побудови доповненої реальності

Можна виділити два головних принципи побудови доповненої реальності: на основі маркера і на основі координат місця розташування користувача.

Безмаркерні технології часто застосовуються в мобільних пристроях і будуються за допомогою спеціальних датчиків: акселерометр, гіроскоп, магнетометр, GPS-приймач. Розглянемо завдання побудови доповненої реальності за допомогою маркерів і алгоритмів комп'ютерного зору.

Під маркером будемо розуміти об'єкт, розташований в навколишньому просторі, який визначається та аналізується спеціальним програмним забезпеченням для подальшого відтворення віртуальних об'єктів. На основі інформації про стан маркера в просторі програма може досить точно спроектувати на нього віртуальний об'єкт, від чого буде досягнуто ефект його фізичної присутності в навколишньому просторі. Використовуючи додаткові графічні фільтри і високоякісні моделі, віртуальний об'єкт може стати важко відмінним від інших елементів інтер'єру або екстер'єру. Як маркери можуть використовуватися аркуш паперу з деяким спеціальним зображенням,

геометричні фігури, тривимірні об'єкти і навіть очі та обличчя людей.

Технології комп'ютерного зору є основоположними для розвитку технологій доповненої реальності і перш за все в галузі використання маркерів. Основний напрямок даної дисципліни — це аналіз і обробка статичних зображень та відеопотоків. Алгоритми комп'ютерного зору дозволяють виділяти ключові особливості на зображенні (кути, межі області), проводити пошук фігур і об'єктів в реальному часі, виконувати 3D реконструкцію за кількома фотографіями тощо. Приклади об'єктів, що можуть використовуватися як маркери, наведено на рис. 1.

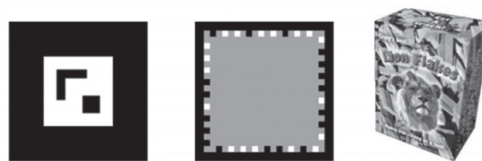


Рис. 1. Приклади маркерів

В області доповненої реальності алгоритми комп'ютерного зору використовуються для пошуку в відеопотоці спеціальних маркерів. Залежно від завдання маркерами можуть бути визначені як спеціально сформовані зображення, так і особи людей. Після знаходження маркера в відеопотоці і обчислення його розташування з'являється можливість побудови матриці проєкції і позиціонування віртуальних моделей. За допомогою цих моделей можна накласти віртуальний об'єкт таким чином, що буде досягнутий ефект присутності. Основна складність полягає в тому, щоб знайти маркер, визначити його місце розташування в кадрі і спроектувати відповідним чином віртуальну модель.

За останнє десятиліття була створена теоретична база в сфері обробки зображень та пошуку в ньому різних об'єктів [10]. Перш за все це стосується методів template matching, генетичних алгоритмів і feature detection. З точки зору побудови доповненої реальності часто використовується feature detection.

Методи feature detection, що найчастіше застосовуються в системах комп'ютерного зору, націлені на обчислення абстракції зображення і виділення на ньому ключових особливостей (наприклад, ізольованих точок, а також кривих або пов'язаних областей). того, що таке ключова особливість зображення. Кожен з конкретних алгоритмів, які реалізують концепцію feature detection, може мати свої специфічні ключові особливості зображення (кути, межі, області тощо).

Згідно з завданням даної роботи будемо для пошуку маркерів виконувати пошук і порівняння зображень по ключових точках, що є певними ділянками фрагменту, характерного для заданого зображення. Приклад такої ключові точки на зображенні наведено на рис. 2.

Для знаходження ключових точок для ідентифікації зображення в відеопотоці використовуються три складові:

- детектор (здійснює пошук ключових точок на зображенні безпосередньо (feature detector));
- дескриптор (здійснює опис знайдених ключових точок, оцінюючи їх позиції через опис навколишніх областей);
- матчер (здійснює побудову відповідностей між двома наборами точок).



Рис. 2. Приклади ключових точок на зображенні

Спочатку за допомогою детектора здійснюється пошук ключових точок шаблонного (шуканого) зображення, які потім описуються за допомогою дескриптора. Дана інформація зберігається в окремий файл (або базу даних), щоб не виконувати цей процес повторно. При обробці відеопотоку з метою пошуку заданого шаблону описаний процес виконується для кожного кадру (за винятком збереження даних). Для встановлення відповідності між ключовими точками і дескрипторами застосовується матчер. Приклад встановлення відповідності точок наведено на рис. 3.



Рис. 3. Встановлення відповідностей між точками шаблону та тестового зображення

Природно припустити, що різні алгоритми працюють з різною швидкістю та ефективністю. В умовах їх застосування для побудови доповненої реальності

необхідно використовувати тільки ті, які показують високу швидкість роботи при досить хорошій якості відстеження позицій ключових точок. В іншому випадку ми можемо отримати помітні відставання у відеоданих, що знімаються. Для підвищення швидкості роботи алгоритмів роботи feature points detection застосовуються різні способи фільтрації точок, щоб мінімізувати їх число і відсіяти невдалі поєднання. Таким чином можна домогтися підвищення як швидкості роботи, так і якості трекінгу маркерів.

2. Алгоритми пошуку особливих точок зображення та створення їх дескрипторів

Як було зазначено, для реалізації технології доповненої реальності використовуються маркери, які допомагають визначати положення камери в просторі. Це обмежує ефективність її використання, так як поперше, маркери повинні бути постійно в кадрі, а по-друге, їх необхідно спочатку підготувати.

Працювати відразу з усіма точками зображення для мобільних платформ досить складно, тому будемо виділяти окремі особливі точки на зображенні і стежити за їх переміщеннями. Для цього необхідно попередньо здійснити процес препроцесінгу маркерів.

Для реалізації поставлених в даній роботі завдань доцільним є застосування методу QCAR, що послідовно здійснює пошук особливих точок зображення та створення їх дескрипторів, інваріантних до масштабу і обертання [10]. Така інваріантність означає, що опис ключової точки буде однаковим, навіть якщо зразок змінить розмір і буде повернений. Крім того, сам пошук ключових точок теж має бути інваріантним, щоб повернений об'єкт сцени мав той же набір ключових точок, як і зразок.

В базовому алгоритмі QCAR з використанням FAST-детектора (Features from Accelerated Test) розглядається коло з 16 пікселів навколо точки кандидата. У цьому алгоритмі будується дуга кола для першого квадранта, а координати точок кола для інших квадрантів задаються симетрично. На кожному кроці алгоритму розглядаються три пікселя, з яких вибирається найбільш придатний шляхом порівняння відстаней від центру до обраного пікселя з радіусом кола. Далі вибирається точка зображення p з яскравістю I_p , для якої буде вирішуватися, чи є вона особливою чи ні. Нехай I_p — яскравість точки. Для цього вибирається значення порога t та розглядається коло з 16 пікселів навколо обраної точки. Точка p вважається кутом, якщо серед 16-ти пікселів кола існує n пікселів, кожен з яких яскравіше, ніж $I_p + t$, або n пікселів, кожен з яких темніше, ніж $I_p - t$ (n зазвичай вибирається рівним 12). Відомо, що найменше значення n , при якому точки починають стабільно виявлятися, дорівнює 9.

Висока швидкість роботи даного алгоритму обумовлена тим, що спочатку перевіряється інтенсивність

лише чотирьох точок з кола, під номерами 1,5, 9 і 13. Якщо хоча б для трьох з них виконується зазначена вище умова нерівності, тоді проводиться перевірка решти 12-ти пікселів. В іншому випадку вибирається наступна точка і алгоритм повторюється для неї. Однак ефективність даного алгоритму залежить від порядку обробки зображення і розподілу пікселів (поблизу деякого околу може виявитися кілька особливих точок). В [10] було запропоновано використання машинного навчання для виправлення цього недоліку, згідно з чим було отримано модифікований алгоритм технології QCAR:

- вибирається набір зображень для навчання;
- алгоритм FAST застосовується для пошуку особливих точок на кожному із зображень;
- для кожної особливої точки зберігається 16 пікселів навколо неї у вигляді вектора. Після виконання даного кроку для кожного зображення всі отримані вектори об'єднуються в один вектор P ;
- кожен з 16-ти пікселів S_p відносять до одного з трьох станів згідно з наступними умовами:

$$S_{p \rightarrow x} = \begin{cases} d, I_{p \rightarrow x} \leq I_p - t (\text{темніше}) \\ s, I_p - t < I_{p \rightarrow x} < I_p + t (\text{схожий}) \\ b, I_p + t \leq I_{p \rightarrow x} (\text{світліше}) \end{cases} \quad (1)$$

- в залежності від станів, наведених вище, P розділяється на три підмножини P_d, P_s, P_b ;
- в розгляд вводиться булева змінна K_p , яка дорівнює 1, якщо p – кут та 0 у зворотному випадку;
- використовується алгоритм ID3 (класифікатор дерев рішень) [11,12]. Він вибирає x , який дає більше інформації про те, чи є піксель кутом чи ні, обчисленої за допомогою ентропії K_p ;
- ця процедура повторюється рекурсивно, до тих пір, поки ентропія K_p не дорівнюватиме нулю;
- створене таким чином дерево рішень використовується для більш швидкого детектування в інших зображеннях.

Для того, щоб уникнути знаходження декількох особливих точок поблизу деякого околу, в розроблюваному додатку будемо використовувати процедуру пошуку локальних максимумів, що полягає в реалізації таких операцій:

- обчислюється значення функції V для кожної зі знайдених особливих точок (V – сума різниць інтенсивностей кожного з 16-ти навколишніх пікселів);
- розглядаються дві сусідні особливої точки і порівнюються їх значення V ;
- точка з меншим значенням V виключається з розгляду.

3. Обчислення координат камери і положення особливих точок

Графічна інтерпретація процедури визначення координат наведена на рис. 4. Ідея цієї процедури

полягає в тому, що при зміні кута, з якого направлено камеру користувача, відповідним чином змінюється розмір проекції.

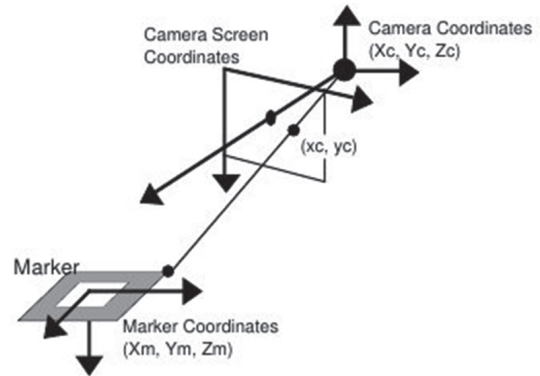


Рис. 4. Принцип визначення координат

Для визначення положення камери в просторі за особливими точками спочатку здійснюється ініціалізація по двом сусіднім кадрам та обчислюються координати камери в цих кадрах і положення особливих точок. Далі, по відомим тривимірним координатам точок можна обчислювати положення камери в кожному наступному кадрі відеопотоку. Для більш стабільної роботи в процес спостереження додаються нові точки та складається карту простору в зоні спостереження камери. Для початку переведемо тривимірні координати особливих точок в екранні. Для цього використовується наступна формула:

$$\begin{pmatrix} screen[i]_x \\ screen[i]_y \\ screen[i]_z \end{pmatrix} = mProj * mWorld * World[i], \quad (2)$$

де $World[i]$ – особливі точки в світових координатах, невідомих до ініціалізації;

$screen[i]$ – компоненти x та y – координати особливої точки на зображенні;

z – глибина зображення відносно камери;

$mProj$ – матриця проекції, що має вигляд:

$$mProj = \begin{bmatrix} pf_x & 0 & -pc_x \\ 0 & pf_y & -pc_y \\ 0 & 0 & 1 \end{bmatrix},$$

pf – фокальна відстань камери в пікселях;

pc – оптичний центр камери в пікселях (центр зображення);

$mWorld$ – матриця, що описує трансформацію точок, розміром 3 на 4 (світова матриця з якої прибрали останній рядок (0 0 0 1)).

У матриці $mWorld$ міститься інформація про переміщення і повороті камери на кожному кадрі.

Екранні координати вже будуть використовуватися на кожному кадрі відео потоку.

Спростимо формулу (2), позбувшись від матриці $mProj$ шляхом введення коефіцієнту $c[i]$ наступного вигляду:

$$c[i] = mProj^{-1} * \begin{pmatrix} screen[i]_x \\ screen[i]_y \end{pmatrix}. \quad (3)$$

З урахуванням (3) отримуємо:

$$c[i] * (screen[i]_z = mWorld * (world[i], 1). \quad (4)$$

Ініціалізація відбувається за двома кадрами. Позначимо їх як A і B . Точки $c[i]$ на кадрах A і B позначимо як $cA[i]$ і $cB[i]$

Таким чином з'являються матриці $mWorldA$ (одиначна матриця) та $mWorldB$, яка обчислюється за допомогою точок, розташованих на одній площині.

Для цих матриць є дійсним таке рівняння:

$$\begin{pmatrix} cB_x * screenB_z \\ cB_y * screenB_z \\ screenB_z \end{pmatrix} = \begin{pmatrix} H_{00} * cA_x + H_{01} * cA_y + H_{02} \\ H_{10} * cA_x + H_{11} * cA_y + H_{12} \\ H_{20} * cA_x + H_{21} * cA_y + H_{22} \end{pmatrix}$$

Це справедливо тільки для точок, розташованих на одній площині. В загальному випадку, у циклі з заздалегідь заданою кількістю ітерацій виконуються наступні дії: випадковим чином вибираються чотири пари точок кадрів A і B ; знаходиться матриця H ; визначається, скільки точок дають помилку менше заданого значення; вибирається H на тій ітерації, для якої було отримано більше точок.

З матриці H необхідно отримати матрицю переходу положення кадру A до кадру B (матрицю $mMotion$). Для цього виконується сингулярне розкладання матриці H згідно з наступними перетвореннями:

$$s = [U] * [V]; \quad (5)$$

$$w1 = |W_{00}|; w2 = |W_{11}|; w3 = |W_{22}|; \quad (6)$$

$$x2 = 0; x3 = \sqrt{(w2^2 - w3^2) / (w1^2 - w3^2)}. \quad (7)$$

Використовуючи перетворення (5) – (7), можна отримати матрицю повороту $R[i]$:

$$R[i] = U * \begin{bmatrix} \cos A & 0 & -\sin A * e3[i] \\ 0 & 1 & 0 \\ \sin A & 0 & \cos A * e3[i] \end{bmatrix} * V^T * e3[i],$$

де

$$\sin A = (w1 - w3 * e3[i]) * x1 * x3 * e1[i] * \frac{e2[i]}{w2};$$

$$\cos A = (w3 * x1^2 + w1 * x3^2 * e3[i]) * \frac{1}{w2};$$

$$e1 = [1, -1, 1, -1, 1, -1, 1, -1];$$

$$e2 = [1, 1, -1, -1, 1, 1, -1, -1];$$

$$e3 = [1, 1, 1, 1, -1, -1, -1, -1];$$

Крім того, необхідно визначити вектор зміщення:

$$t[i] = U * (((w1 - w3 * e3[i]) * x1 * e1[i]), 0, (w3 - w1 * e3[i]) * x3 * e2[i])).$$

Отримані значення $R[i]$ та $t[i]$ дозволяють побудувати 8 можливих варіантів матриці $mMotion$:

$$mMotion[i] = \begin{bmatrix} R[i]_{00} & R[i]_{01} & R[i]_{02} & t[i]_x \\ R[i]_{10} & R[i]_{11} & R[i]_{12} & t[i]_y \\ R[i]_{20} & R[i]_{21} & R[i]_{22} & t[i]_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Можна показати, що:

$$mWorld_B = mMotion[i] * mWorld_A. \quad (9)$$

Залишається вибрати одну з восьми матриць $mMotion[i]$. Якщо промені виходять з точок першого і другого кадру, то вони повинні перетинатися, причому перед камерою, як в першому, так і в другому випадку. Тому слід підрахувати кількість точок перетину перед камерою в першому і в другому кадрі, використовуючи отримані $mMotion[i]$, та відкинути варіанти, при яких кількість точок буде меншою.

В результаті вибирається та з матриць, яка дає менше помилок. Таким чином, використовуючи отримані матриці $mWorldA$ та $mWorldB$, можна знайти світові координати точок по проєкціях.

Після проведених дій можна обчислити положення камери використовуючи відомі світові координати точок. Для цього використовуємо ітеративний алгоритм, початковим наближенням якого є попередній результат визначення. На кожній ітерації вводимо точки $b[i]$:

$$b[i] = mWorld[i] * (World[i], 1). \quad (10)$$

Ці точки мають послідовно наближатися до визначених вище точок $c[i]$ з відповідною помилкою:

$$e[i] = \left(c[i]_x - \frac{b[i]_x}{b[i]_z}, c[i]_y - \frac{b[i]_y}{b[i]_z} \right). \quad (11)$$

Далі підраховується вектор помилки і вирішується система рівнянь методом найменших квадратів з остаточним отриманням шуканих координат.

4. Опис розробки інтерактивного музейного додатку та його структури

В розробленому інтерактивному музейному додатку для підтримки AR – технологій з забезпеченням функції трекінгу було використано засоби середовища Vuforia. Модуль Vuforia підтримує різні 2D і 3D типи мішеней, включаючи маркерні Image Target, що дозволяє використовувати вкладені функції для реалізації плавного треку маркерів, які можуть бути реалізовані у вигляді деяких реальних графічних зображень музейних колекцій. Наступним кроком став вибір графічного рушія, який дозволив би перенести додаток на мобільну платформу, оскільки відеопотік має трансляватися безпосередньо через відеокамеру мобільного пристрою.

Інструментом для розробки додатків, що працюють під операційною системою Windows, обрано мову C # 7.0, яка легко інтегрується в графічне ядро Unity. З використанням цієї мови було реалізовано модуль візуалізації особливих точок зображення за методом FAST технології QCAR. Приклад такої реалізації наведено на рис. 5.

У даній реалізації алгоритм ID3 використовується для оптимізації порядку, в якому аналізуються пікселі, що дозволило побудувати найбільш ефективний

детектор особливостей. Для отримання проміжних результатів розроблено допоміжний алгоритм, який виводить початкові зображення і зображення, що пройшли препроцесінг (з візуалізованими на них особливими точками).



Рис. 5. Приклад візуалізації особливих точок за методом FAST

Після успішної інтеграції бази даних аналізованих зображень в проект Unity необхідно, використовуючи модуль Vuforia, співвіднести маркери і відповідні їм віртуальні об'єкти. У розробленому додатку як віртуальні об'єкти розглядаються тривимірні тексти з координатами, що відповідають локальним координатам маркера. Можливості інтерактивного характеру для кожного віртуального об'єкта реалізуються за допомогою виведення додаткової інформації при натисканні на ідентифікований в відеопотоці маркер.

Основними перевагами Unity є наявність візуальної середовища розробки, міжплатформенної підтримки і модульної системи компонентів. До недоліків зазвичай відносять появу складнощів при роботі з багатокомпонентними схемами і труднощі при підключенні зовнішніх бібліотек, однак для розроблюваного додатку ця умова не є критичною.

Головний інтерфейс розробки музейного інтерактивного додатку поділено на 5 зон:

- зона 1, де знаходяться всі фізичні об'єкти, з якими в подальшому працює додаток;
- зона 2 (сцена), на якій здійснюється відображення усіх фізичних об'єктів;
- зона 3 (файлова структура);

— зона 4, через яку можна працювати з файлами, які знаходяться в зоні 3;

— зона 4, що містить усі характеристики об'єктів, які обрано в зоні 1.

Коренева структура розробленого додатку має 3 основних директорії, а саме: Modules, StreamingAssets та Mechanisms.

У директорії Modules знаходяться три основних Third-Party модулі: Demigant, TextMesh Pro, Vuforia.

Demigant — модуль для взаємодії функціонального пересування, контролю на налаштування ротації та змінення розмірів об'єктів у реальному часі. Цей модуль необхідний для полегшення роботи при написанні механізмів візуалізації.

TextMesh Pro — допоміжний модуль для роботи зі шрифтами та UI текстовими компонентами, що має можливість роботи в мультимовному режимі. У додатку він використовується для візуалізації описів ідентифікованих картин.

Vuforia — допоміжний модуль для реалізації взаємодії трекінгу та додаткових даними.

У директорії StreamingAssets знаходиться база даних маркерів, що дають можливість ідентифікувати зображення.

Директорія Mechanisms містить наступні механізми: Bootstrap; DescriptionManager; MuseumManager; TrackingSystem; UIController.

Для розробки бази даних зображень та описів до них був реалізований модуль ScriptableObject DescriptionDataStorage, що серіалізує ліст класу ArtDescription, який містить назву зображення, спрайт зображення, рік створення та код трекінгу. Разом з тим реалізовано механізм GetArtByCode, що здійснює пошук необхідних даних по трекінговому коду для серіалізованого об'єкту.

Для доступу до бази даних використовується менеджер DescriptionManager, що представлений статичним класом та має в диспозиції набір методів доступу до бази даних.

Для видачі інформації про ідентифіковане зображення та меню основної та додаткової інформації у додатку використано універсальний контролер UI.

Важливим кроком реалізації додатку є обробка подій детекції зображень та їх виходу з зони пошуку. Для реалізації цих функцій використано клас компонентів CustomTrackableEventHandler, що є спадкоємцем класів MonoBehaviour та інтерфейсу ITrackableEventHandler. При ініціалізації цього компоненту проходить реєстрація підписок TrackableEventHandler компоненту та ініціалізація TextMesh, тобто генерація об'єкту з компонентом тексту за заданим набором параметрів відображення до кожного із об'єктів трекінгу. Даний алгоритм необхідний для того, щоб при ідентифікації об'єкту у відеопотоці на ідентифікований об'єкт можна було зпроєціювати його назву.

При реєстрації події трекінгу маркеру викликається метод `OnTrackingFound`, що в свою чергу викликає метод `UpdateTextMesh`, який виводить на ідентифіковане зображення необхідний текст та викликає метод `OnItemTracked` класу `MuseumManager`. При загрузці зображення у зоні пошуку викликається метод `OnTrackingLost`, що відключає текст на зображенні та викликає метод `OnItemLost`, який відключає процес отримання додаткової інформації на екрані.

Для реалізації основних модулів додатку необхідно заповнити базу даних зображеннями та дескрипторами для них з застосуванням механізмів `Vuforia`.

Додаток передбачає можливість роботи з хмарною базою даних. Для реалізації в додатку цієї можливості використано клас `CloudRecoBehaviour`.

5. Апробація інтерактивного музейного додатку

Додаток був протестований для низки різноманітних мобільних пристроїв на платформах `iOS` та `Android`. Результати тестування показали, що додаток працює стабільно і не викликає системних помилок, що призводять до закриття чи збоїв.

Реалізований продукт відповідає усім вимогам, що були поставлені у розрізі аналізу області та маркетингового огляду: стабільна робота на різноманітних пристроях; кросплатформеність; можливість роботи без підключення до інтернету; безперервна робота у режимі відеопотоку; можливість отримувати назву зображення при фокусі на ньому; можливість отримати базову додаткову інформацію про зображення, не виходячи з додатку; можливість отримати розширену інформацію про додаток з виходом у браузер.

Базовий інтерфейс додатку є відеопотоком камери на пристрої та інтерактивною кнопкою у активному стані та кнопкою-підказкою у пасивному. У пасивному стані (стані пошуку) текст на кнопці підказує користувачу, що треба знайти картину. Приклад візуалізації інтерфейсу у цьому стані представлено на рис. 6.



Рис. 6. Інтерфейс додатку у стані пошуку

Коли у рамках відеопотоку з'явиться зображення, яке можливо ідентифікувати, то кнопка змінить свій стан та стане інтерактивною, внаслідок чого текст на ній зміниться на `Get Info`, а під ідентифікованим зображенням з'явиться його назва. Приклад візуалізації цього стану представлено на рис. 7.

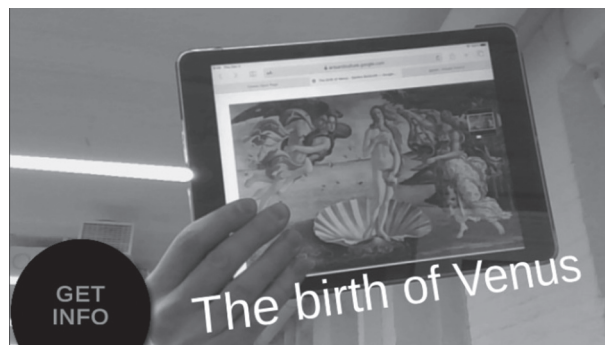


Рис. 7. Інтерфейс додатку в активному стані

Якщо користувач натисне на кнопку `Get Info`, то у додатку запуститься вікно додаткової інформації, що містить у собі зображення у кращій якості, назву картини, прізвище автора та роки її створення. Візуалізація меню додаткової інформації представлена на рис. 8.



Рис. 8. Меню додаткової інформації про картину

На цьому меню також присутні кнопка виходу та кнопка пошуку інформації в інтернеті. При натисканні на кнопку виходу додаток повернеться до стану обробки відеопотоку. При натисканні на кнопку пошуку інформації в інтернеті відкриється вікно браузера із запитом про дану картину.

Оскільки розробка додатку зорієнтована на мобільні платформи, що передбачає необхідність взаємодії додатку з відеопотоком та рендерінгом, важливо дотримуватись високої загальної швидкодії продукту та високої швидкодії його роботи з відеопам'ятю.

Тестування швидкодії додатку по `CPU` та аналіз навантаження на відеопроцесор підтвердили відсутність проблем зі швидкодією при використанні додатку в музейних приміщеннях в реальному режимі часу.

Крім того, тестування дозволило отримати оцінки мінімальних затрат фізичної пам'яті, необхідних для стабільної роботи додатку, що реалізується на різних на мобільних платформах.

Висновки

У статті досліджено завдання розробки інтерактивного музейного додатку з використанням технології доповненої реальності, що базується на інструментах середовища `Vuforia` та графічного ядра `Unity`. Додаток дає можливість здійснювати аналізу та розпізнавання

реальних об'єктів (наприклад, музейних експонатів), які раніше були додані в один із сервісів Vuforia для реєстрації зображень (маркерів). В результаті система надає користувачам дані про відповідність реального об'єкта і маркера, внаслідок виникає можливість зміни віртуального оточення реальних об'єктів. Положення і орієнтація віртуального об'єкта відстежується в реальному часі, так що точка зору користувача на об'єкт співвідноситься з його точкою зору на зображення, що створює ілюзію належності віртуального об'єкту до реальної сцени. Наприклад, користувач може завантажити в веб сервіс зображення картини і використовувати розроблений додаток, визначити збіг картини з частиною реального світу, що потрапила в фокус камери, з маркером, який розташований в сервісі, і в разі необхідності здійснити певні маніпуляції в віртуальному світі, наприклад, вивести опис про цю картину на екран. Результати експериментального тестування підтверджують відсутність проблем зі швидкодією при використанні додатку в музейних приміщеннях в реальному режимі часу.

Перспективним розвитком додатку є подальше дослідження напряму розвитку його функціональних можливостей з урахуванням додаткових метаданих щодо зображень об'єктів та створення відповідних програмних модулів. Додаток можна використовувати під час навчання людей с обмеженими можливостями, розробивши модуль, який відтворюватиме звук.

Список літератури:

- [1] Website Architecture Planning - Augmented Reality (AR) [Електронний ресурс]. — URL: <https://dynamapper.com/blog/15-augmentedreality-ar>. — Назва з екрана.
- [2] Доповнена реальність у мобільному додатку Smartum! [Електронний ресурс]. — Режим доступу: <https://smartum.com.ua/news/dopolnennaya-real-nost-v-mobilnomprilozhenii/>. — Назва з екрана.
- [3] Tim Pedure. (2017). Applications of Augmented Reality [Електронний ресурс]. URL: <https://www.lifewire.com/applications-of-augmented-reality-2495561>. — Назва з екрана.
- [4] V. Interrante, T. Höllerer and A. Lécuyer, «Virtual and Augmented Reality,» in IEEE Computer Graphics and Applications, vol. 38, no. 2, pp. 28-30, Mar./Apr. 2018. doi: 10.1109/MCG.2018.021951630 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8336832&isnumber=8336820>. — Назва з екрана.
- [5] Qingfeng Zhang, Weilong Chu, Changhong Ji, Chengyuan Ke, Yamei Li. (2015). An implementation of generic augmented reality in mobile devices [Електронний ресурс]. URL: <http://ieeexplore.ieee.org.libaccess.sjlibrary.org/document/7065112/>. — Назва з екрана.
- [6] Модло Є. О. Використання технології доповненої реальності у мобільно орієнтованому середовищі навчання ВНЗ / Є. О. Модло, Ю. В. Єчкало, С. О. Семеріков, В. В. Ткачук. [Електронний ресурс]. — URL: <https://phm.cuspu.edu.ua/ojs/index.php/NZPMFMTO/article/viewFile/1115/1094>. — Назва з екрана.
- [7] Слюсар В.І. Концепція стандартизації тактичних засобів доповненої реальності / В.І. Слюсар //Тези доповідей Міжнародної науково-технічної конференції «Перспективи розвитку озброєння та військової техніки Сухопутних військ». —Львів: Національна академія сухопутних військ імені гетьмана Петра Сагайдачного.— 2018 — С. 63 — 64.
- [8] Чубукова О.Ю. Інноваційні технології доповненої реальності для викладання дисциплін у вищих навчальних закладах України / О.Ю. Чубукова, І.В. Пономаренко // Проблеми інноваційно- інвестиційного розвитку. — 2018. — № 16 —С. 20 — 27.
- [9] Берестовий О.О. Системи когнітивного розвитку з використанням платформи Vuforia / О.О. Берестовий, Д.І. Карпенко, С.Г. Удовенко// Матеріали 5 Міжнародної науково-практичної конференції «Обчислювальний інтелект» — 15-21.04.19 р. Україна . — Ужгород. — 2019. — С. 165-166.
- [10] Mikolajczyk K. An Affine Invariant Interest Point Detector / K. Mikolajczyk, C. Schmid // European Conference on Computer Vision, — 2002. — P. 128 —142.
- [11] Ronald T. Azuma. A Survey of Augmented Reality / T. Azuma Ronald // [Електронний ресурс]. — URL: <https://www.cs.unc.edu/~azuma/ARpresence>
- [12] Четвериков Г.Г. Многозначные структуры (анализ, сравнение, синтез, обобщение). Ч.1: Учебное пособие.— К.: ИСМО, 1997.— 192 с.

Надійшла до редколегії 3.12.2019