

天主教輔仁大學資訊工程研究所論文

指導教授:梅 興 博士

Mobile AR Cache Management

- A Case Study with Vital Tree

行動 AR 快取管理

-以生命之樹為例

研究生:歐育綺 撰

中華民國一百零九年七月

天主教輔仁大學資訊工程研究所論文

指導教授:梅 興 博士

Mobile AR Cache Management

– A Case Study with Vital Tree

行動 AR 快取管理

-以生命之樹為例



研究生:歐育綺 撰

中華民國一百零九年七月

摘要

隨著科技迅速演進，許多場域中已使用將虛擬物件與真實環境結合的擴增實境 (Augmented Reality, AR) 技術。行動擴增實境 (Mobile AR) 應用雖隨著行動裝置能力增強而逐漸普遍，但 AR 應用使用時，裝置上卡頓、閃退等問題層出不窮。經過不流暢體驗，使用者極可能會卸載此應用程式。主要原因是 AR 應用需要較多運算、儲存、與網路資源。

本論文提出一個行動 AR 快取管理機制。透過在遊戲設計中常用到的預存 (Prefetch) 技術，預先存置某些未來可能會用到的物件，並參考作業系統用於虛擬記憶管理的 LRU (Least Recently Used) 快取替換演算法來增加儲存效能。本研究考量使用者行為與行動 AR 需求，提出一個改良 LRU 快取替換演算法。有別於傳統 LRU 演算法只考慮時間因素，以最久未被使用的物件為替換目標。改良 LRU 演算法還考量裝置與目標物件的距離，再排出替換物件的優先權重。在「輔大醫院生命之樹瓷浮雕壁畫」案例中，提升了 25% 快取命中率。除了解決個別案例，本研究的快取管理機制也期望能有效應用在一般有行動需求的 AR 應用中。

關鍵字

擴增實境、快取替換、LRU 演算法、裝置資源管理

Abstract

Owing to the rapidly evolving information and communication technologies, Augmented Reality (AR) that combines virtual objects with real environments has been used in many applications. Although Mobile AR applications are becoming more common with the enhancement of mobile device capabilities, issues such as freezing and flashbacks on the device are constantly appearing in AR users' experience. As continuous delays or interrupts occur, users are likely to uninstall this application. The main reason of unsmooth operations is that AR applications require more computing, storage, and network resources.

This research proposes a cache management mechanism for Mobile AR applications. Prefetching technology, which is widely used in game development, is used to store AR objects ahead the users' operation. The LRU(Least Recently Used) cache replacement algorithm, which is commonly used in operating systems, is also referenced to enhance the AR object storage performance. This research considers both user behavior and mobile AR requirements, and proposes a modified LRU cache replacement algorithm. Different from the traditional LRU cache replacement algorithm which considers only the time factor, replaces the least recently used object out of cache storage. The modified LRU algorithm combines both time and distance parameters to prioritize the replacement order. In the case of "Tree of Life", a porcelain relief murals in the Fu Jen Catholic University Hospital, the cache hit rate was increased by 25%. Not only solving a single AR application, this cache management mechanism is expected to be effectively for general Mobile AR applications.

Keywords

Augmented Reality, Cache Replacement, LRU Algorithm, Device Resource Management

目錄

摘要	IV
Abstract.....	V
附表目錄	VIII
附圖目錄	IX
第一章 緒論	1
1-1 研究動機與目的.....	1
1-2 論文架構.....	3
第二章 背景與文獻探討	4
2-1 擴增實境.....	5
2-2 快取管理.....	8
2-2-1 快取替換	8
2-2-2 預存機制	9
2-3 生命之樹瓷浮雕壁畫.....	10
第三章 AR 物件快取機制	13
3-1 改良 LRU 快取替換演算法.....	14
3-1-1 案例說明	16
3-2 預存機制.....	19

第四章 實驗與分析	22
4-1 實驗設計.....	22
4-2 實驗分析.....	28
第五章 結論及未來展望	33
5-1 結論.....	33
5-2 未來展望.....	35
參考文獻	36



附表目錄

表格 一，預存及快取機制兩種處理方式	20
表格 二，預存及快取機制四種情境	20
表格 三，傳統 LRU 快取替換演算法的優先權重	23
表格 四，改良 LRU 快取替換演算法的優先權重	24
表格 五，傳統 LRU 快取替換演算法	26
表格 六，改良 LRU 快取替換演算法	26
表格 七，距離參數固定的參數分析(樣本數量:100000 組).....	29
表格 八，時間參數固定的參數分析(樣本數量:100000 組).....	29



附圖目錄

圖 一，擴增實境的定位	5
圖 二，生命之樹瓷浮雕壁畫	10
圖 三，生命之樹瓷牆	11
圖 四，生命之樹 AR 動畫.....	11
圖 五，改良 LRU 快取替換演算法流程圖	15
圖 六，範例 AR 物件大小.....	16
圖 七，範例 AR 物件間距離.....	16
圖 八，範例 AR 物件序列及時間.....	16
圖 九，傳統 LRU 快取替換演算法	17
圖 十，改良 LRU 快取替換演算法	18
圖 十一，傳統 LRU 替換快取演算法	25
圖 十二，優化 LRU 替換快取演算法	25
圖 十三，實驗一快取命中率比較	30
圖 十四，實驗二快取命中率比較	31

第一章 緒論

1-1 研究動機與目的

在這個虛擬實境、擴增實境逐漸成為新風潮的時代，許多遊戲公司、新興工廠、文創相關展覽、各式導覽活動皆引進 VR(Virtual Reality)、AR(Augmented Reality) 作為新技術，運用虛實整合帶來的獨特體驗，藉此吸引不同族群用戶。其中擴增實境部分，在實體環境中疊加虛擬影像，配合定位等技術，營造有別於 VR 更真實的體驗模式，配合上發展日新月異的行動裝置，讓使用者更著重於虛實之間細微的感受，相對的在呈現部分就需以更高標準對待。

AR 應用雖逐漸廣泛，裝置效能也持續地進步，在許多遊戲、導覽中不乏看見各式相關 AR 體驗，但也經常在使用時出現不定期卡頓，甚至閃退的問題。行動 AR 受限於裝置資源有限，不順暢的流程對使用者體驗的影響更為強烈。經我們發現，普通使用者在經歷二至三次發生延遲或閃退問題後，即會立即卸載應用程式，不僅造成使用率下降，更是給開發者帶來極大困擾。

在實際行動 AR 應用中，使用者會因場域大小、行動路徑、個人體驗等不同因素，不斷移動並產生各種選擇的物件序列。難以預測的使用者體驗，再加上有限的裝置資源，就產生延遲的問題。因此本研究以系統效能為考量，以遊戲中常用到的預存技術為出發點，配合改良的 LRU 快取替換演算法[1]，將預存及快取技術[2]應用在 AR 環境中。除了時間參數之外加入距離參數，排出替換儲存的優先權重，設計一

個整合機制，有效地對裝置進行動態資源分配。

最後以程式模擬傳統及改良 LRU 快取替換演算法，並蒐集生命之樹行動 AR 應用的數據後，以生命之樹及隨機物件序列進行實驗並綜合對比。採用快取命中率[3]比較在不同距離、時間參數是否變動時的差異。並分析說明當物件個數、裝置快取總容量、AR 物件大小相同，距離參數與時間參數不同時帶來的實驗分析。也期望在不同 AR 應用個案當中，皆能以本研究方法做為參考依據，降低體驗時的延遲問題。



1-2 論文架構

第一章，緒論，敘述本論文動機與目的。

第二章，背景與文獻探討，首先探討現有擴增實境與遇到問題，接著探討本論文相關快取管理、快取替換、預存機制，與生命之樹瓷浮雕壁畫介紹。

第三章，AR 物件快取機制，詳細說明本研究之改良 LRU 快取替換演算法與預存機制。

第四章，實驗與分析，說明本研究實驗設計，分析及提出建議。

第五章，結論及未來展望，提出總結與未來研究方向。



第二章 背景與文獻探討

近年來 AR 在創新領域內潛力無窮，且具有高度國際競爭力。在許多展覽、遊戲、設計、電商等產業內都迅速發展應用。國發會在計畫內的幾家新創公司，也與多家國際團隊一同榮獲「尋找十年後的 Google，改變世界的 100 家公司」殊榮[4]，顯現 AR 富有的龐大實力。

此外，在現行社會中行動 AR 應用也漸趨廣泛，人手一支的手機就能輕鬆支援 AR。但想要有更高臨摹感的 AR 應用體驗，就需要解決目前行動 AR 可能產生的效能問題。本章節將介紹何謂擴增實境(AR)，當使用 AR 應用程式時常會遇到的問題。本研究為了解決此問題會應用到的預存及快取管理，以及實驗相關的生命之樹瓷浮雕壁畫。



2-1 擴增實境

1997 年 Ronald Azuma 提出了擴增實境必備的三個面向[5]:

1. 結合虛擬與現實
2. 可以即時互動
3. 存在於三維空間中

由此定義可探討 AR 在虛實間的定位，若現實空間全部皆為真實物體；虛擬實境為電腦模擬產生三維空間的虛擬世界，AR 則存在於兩端點之中，獨特的在現實世界上疊加了虛擬物件，如圖一所示。



圖一，擴增實境的定位

以另一種角度來看待 AR，AR 應用程式通常包含三項特點：掃描偵測、呈現、使用互動[6]。掃描偵測以物體在三維空間中的六個自由度來定位，基於偵測的場景來呈現 3D 模型等…，最後在使用互動上允許用戶與實體空間中的虛擬物件傳遞數位訊息。

AR 不論在教育[7]、工程[8]、娛樂[9]、建築規劃等產業，皆利用虛擬物體嵌於實體物理空間中的特性，提升各行業的蓬勃發展。從 AR 設備的產生(Google glass、

Microsoft Hololens、 Magic Leap Magic Leap One)、適用不同平台的開發套件 (ARCore、 ARKit)、webAR[10]的出現，逐漸地發展至今日。前幾年引起全球熱潮的適地性 AR 遊戲 Pokemon Go，在公開發布後的短短八週內，全球下載量已超過了 5 億次[11]，也讓大眾慢慢認知到行動 AR 的重要性。行動 AR 的技術與設備追求輕量、移動性且功能齊全，可在一台具有所需要傳感功能的行動裝置中，邊移動邊使用 AR 帶來的特殊體驗，也成為了近年來行動 AR 應用如此普遍的原因。

比起虛擬實境，擴增實境讓人們融入了真實世界，並產生更深的連結。擴增實境在多媒體應用中，並非是讓人們迷惘在科技裡，反倒是運用一種新興方式，使科技退居幕後，輔助我們更專注在真實生活。



擴增實境問題

由於在多數行動 AR 應用中，使用者要求能有流暢的體驗感受，使用上才能更為全神貫注。然而經我們發現，因每臺行動裝置差異性，及不同 AR 應用的物件差別，在應用執行的不同階段均可能發生畫面延遲、程式閃退等問題[12]。此外，裝置記憶體空間固定，要能同時容納所有物件也增加了解決問題的複雜性。因此本論文的研究議題是設計一個預存及快取管理，在有限的裝置資源內，進行有效的動態資源分配[13]。如何在各式應用環境下將不同 AR 物件進行有效分類，減低因使用者在體驗時無法預測的動向導致耗費行動 AR 裝置龐大的儲存及運算資源，就成為本論文需要解決的議題。



2-2 快取管理

為了提高整體應用程式與檔案儲存的效率，需要一個儲存速度較快的記憶體輔助。當 CPU 處理大量資料時，就會先到快取記憶體尋找，對曾經使用過而先被讀取進快取記憶體的檔案，系統就不需要再次進入速率較慢的主記憶體內進行搜索，漸少不必要的資源浪費。而在具有大量儲存資料的行動 AR 應用程式中，因過多大型檔案造成的延遲問題更是嚴重。因此本研究提出了一個快取管理機制，將之後高機率會使用到的物件先行替換進快取中，達到資源的有效配置。

2-2-1 快取替換

在快取管理中，快取處理速度雖快，儲存空間卻十分有限。當有限的記憶體空間被占用殆盡，無法容納程式執行所有需要的資料，就需要開始選擇替換某些優先次序低的檔案。因此，當有新的檔案想要替換進快取時，需要根據一定的原則來取代掉已在快取記憶體的檔案。快取替換的方法有許多種，常見的 FIFO 演算法、LFU 演算法、LRU 演算法[14,15]……，這些演算法皆是基於不同標準而訂立[16]。本論文所參考的是 LRU 演算法(Least Recently Used)，此演算法相信，若某一檔案最近被訪問過，未來再度被訪問的機率也會高。因此基於被訪問的時間記錄排列優先次序，替換掉優先次序低的檔案，達成替換最長時間未被使用檔案的策略。

不同的是，因本研究應用範圍為室內行動 AR 裝置，使用者會不斷移動位置，也與環境中不同 AR 物件產生相對距離。傳統快取替換演算法沒有考慮到行動 AR 中

使用者與 AR 物件距離的變化性，最長時間未被使用的檔案未必不會馬上再被使用。為了解決此問題，本研究在參數部分除了考量基本 AR 物件大小、裝置快取大小、時間參數等數據，還添加了距離參數，成為一個針對室內行動 AR 形成的改良 LRU 快取替換演算法。本研究也透過此快取替換方式，有效提高室內行動 AR 的快取命中率。

2-2-2 預存機制

不管是在當紅的手遊或網路遊戲中，我們經常看到遊戲加載進度條呈現在畫面中間。進度條產生的緣由來自於遊戲開始時資源的載入。在遊戲切換場景過程，往往需要載入多樣的物件，若使用者同步下載與遊玩，極可能造成等待時間過長。因此多媒體遊戲開發者常使用一種在遊戲程式執行中非同步加載資源的方式，預先存置某些預期未來可能會使用到的資源[17]。在本論文中也使用類似的方法，以 AR 應用為出發點分類預存機制。透過預存機制與本研究改良的 LRU 演算法配合，期望能從系統效能作為考量，在使用體驗上帶來好的影響。

2-3 生命之樹瓷浮雕壁畫



圖 二，生命之樹瓷浮雕壁畫

坐落於輔大醫院一樓主電梯旁的生命之樹瓷浮雕壁畫(圖二)，記載著聖經中生命樹十個天上牧者與人間典範的故事(圖三)。仔細觀察可發現藏於其中的三十八隻迷你青蛙，潺潺流水與巍然屹立的樹幹則代表了旺盛的生命力。生命之樹瓷浮雕壁畫意味著輔大醫院如同生命樹般照拂與眷顧著每一位輔大醫院的民眾，本研究也將會以此藝術瓷牆衍生的 AR 應用程式做為分析題材。



圖 三，生命之樹瓷牆



圖 四，生命之樹 AR 動畫

其中，此 AR 應用程式包含了五個天上牧者及一位人間典範的動畫故事(圖四)。

使用此應用程式流程應為掃描至瓷牆區域，接著呈現相對應 AR 動畫。但由於一個 AR 動畫即囊括了大小物件及場景，規模較大的 AR 動畫可達到 1.4GB，而小動畫則低至 224MB。初步判斷大量不同的物件大小，會造成物件不斷無效率地移入並移出快取，導致效能的低落與延遲問題。

在本研究中特別的以室內行動 AR 應用為例，行動 AR 區分為室內與室外兩種。

而生命之樹是一款室內行動 AR 應用程式。室內與室外行動 AR 皆會受到使用者移動而影響，不同的也正是因为使用者在室外具有更廣闊的移動範圍[18]，還需要考量到相較室內行動 AR 不穩定的網路等…因素。因此本研究將透過預存及改良 LRU 快取演算法，著重於室內的行動 AR 應用，提出適合的行動 AR 物件快取管理。



第三章 AR 物件快取機制

本研究以一物件快取管理機制來有效減緩行動 AR 體驗中的延遲問題。該物件快取管理機制透過應用在遊戲領域中的預存技術，以及經常在作業系統中使用的 LRU 快取替換演算法做改良，添加距離參數，使 AR 物件有效地進入快取。當用戶使用室內行動 AR 應用程式時，體驗感受更加優化。

本研究之預存與快取機制考量以下參數：

- AR 物件大小
- AR 物件個數
- 裝置快取容量
- 物件序列
- 時間:物件距離上次被使用的時間/目前總花費時間
- 距離:物件與目標物件距離/物件間總距離



3-1 改良 LRU 快取替換演算法

當目標 AR 物件在預存考量範圍內，物件大小超出裝置快取容量許可範圍時，需要透過改良的 LRU 快取替換演算法進行置換。本論文因應現行 AR 行動裝置運算能力，除了參考傳統 LRU 快取替換演算法，以時間為基準選擇替換最長時間未被使用檔案外，額外添加距離參數，形成改良的 LRU 快取替換演算法。改良 LRU 快取替換演算法流程圖如圖五。

如圖五所示，當此目標 AR 物件不在快取中，且快取空間不足以置放目標 AR 物件時，就需要改良 LRU 快取替換演算法的配合。有別於傳統 LRU 演算法的是，傳統 LRU 快取替換演算法為替換最久時間未被使用物件，只以時間參數做為考量。而在現實行動 AR 使用體驗中，使用者會因場域大小、行動路徑、個人體驗等不同因素，不斷移動並產生各種選擇的物件序列。因此本研究特別地加入了距離參數做紀錄，以『改良 LRU 的優先權重』作為改良 LRU 快取替換演算法標準[19]，目的為透過距離及時間兩大參數的相乘，得出一個相較於傳統 LRU 快取替換演算法更適用於室內行動 AR 裝置的改良 LRU 演算法。『改良 LRU 的優先權重』公式如下：

$$P = (U/T) \times (D/S)$$

其中 U 為 AR 物件距離上次被使用的時間(sec)；T 為目前總花費時間(sec)；D 為各 AR 物件與目標物件距離(m)；S 為物件間總距離(m)。計算出的結果為 P，意指此改良 LRU 快取替換演算法的替換優先權重。藉此方法可以得知當裝置剩餘快取

容量不足時，應該從快取中被替換的 AR 物件優先順序。P 數值愈高表示會在需要時優先被替換，將最長時間未被使用及與目標物件距離越遠的資源離開快取，以此達到行動 AR 資源的有效配置。

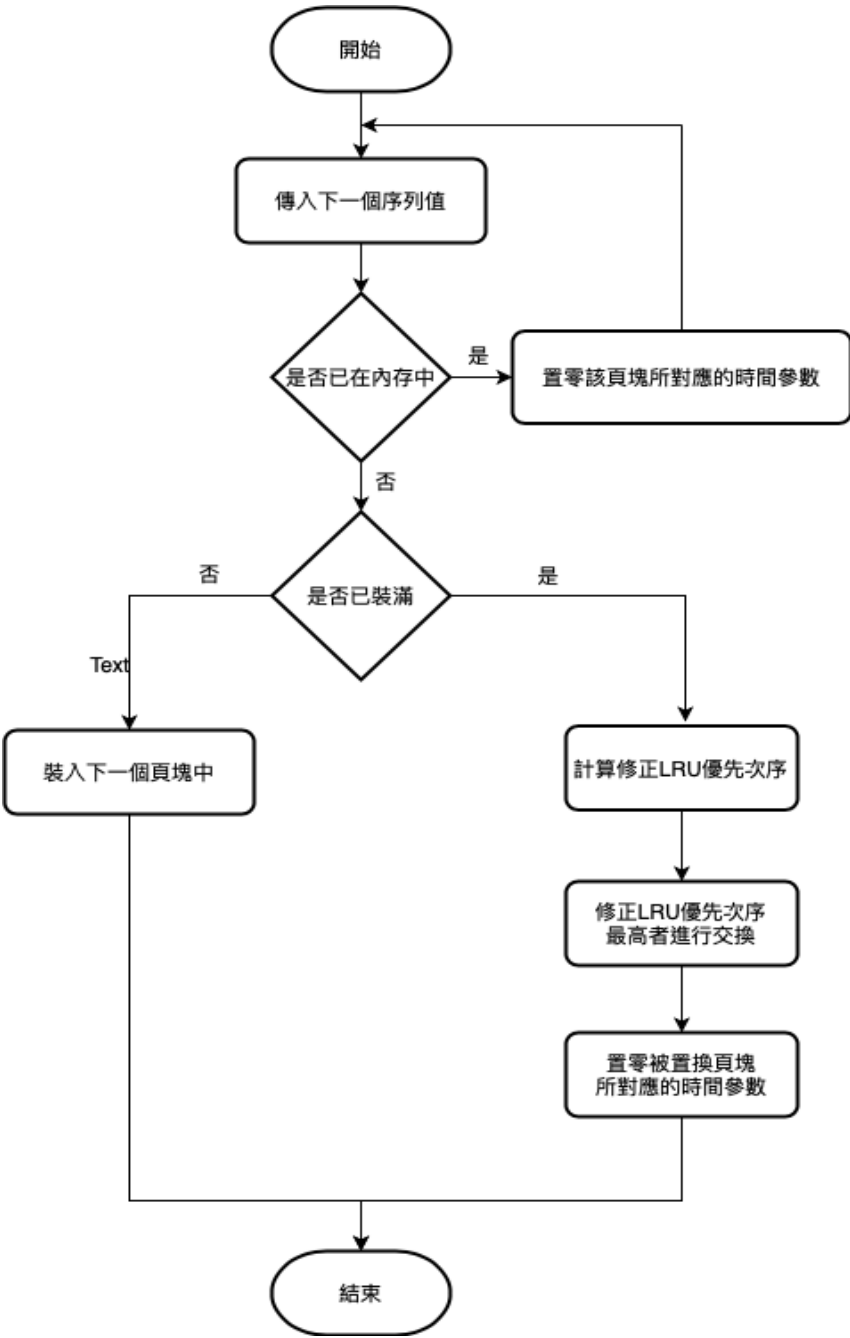


圖 五，改良 LRU 快取替換演算法流程圖

3-1-1 案例說明

本研究使用的改良 LRU 快取替換演算法增加距離相關參數，配合傳統 LRU 快取替換演算法的時間參數，有效提高行動 AR 裝置的快取命中率。AR 物件大小等數據從輔大醫院生命之樹 AR 應用程式中獲取，並按照本研究中設計的改良 LRU 快取替換演算法進行模擬。首先透過圖表呈現傳統 LRU 替換演算法與本研究設計之改良 LRU 替換演算法的實際替換情況[20]與差別。範例 AR 物件大小及裝置快取總容量如圖六、範例 AR 物件間距離則如圖七、範例 AR 物件序列及時間如圖八所示。

AR 物件	1	2	3	4	5	6
物件大小(MB)	477	920	302	224	300	1000




圖 六，範例 AR 物件大小

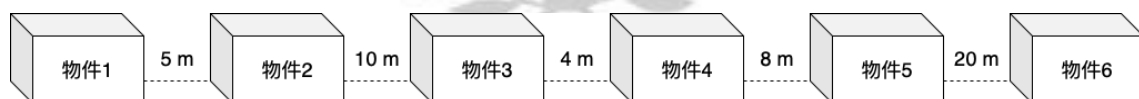


圖 七，範例 AR 物件間距離

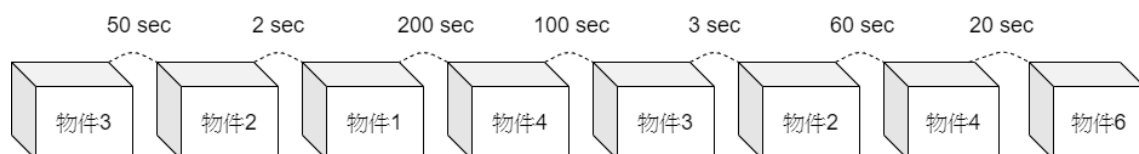


圖 八，範例 AR 物件序列及時間

傳統 LRU 快取替換演算法：

傳統 LRU 快取替換演算法只以時間參數作為判斷標準，使最久未被使用的物件為快取演算法替換目標。圖九展示 AR 物件替換序列:物件 3->物件 2->物件 1->物件 4->物件 3->物件 2->物件 4->物件 6。由圖九可看出傳統 LRU 快取替換演算法替換情形。

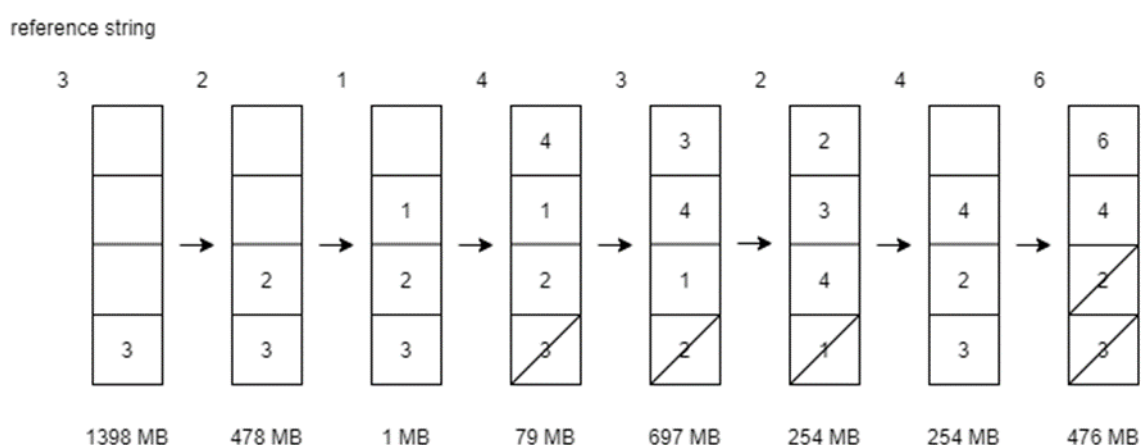


圖 九，傳統 LRU 快取替換演算法

圖九中，當快取剩餘容量不足物件 4 置入時，傳統 LRU 快取替換演算法以時間參數為考量，將最久未被使用到的物件 3 替換出快取。傳統 LRU 快取替換演算法雖將最久未被使用的物件替換，但考慮到在行動 AR 應用中，大多為使用公用裝置及使用者行走路徑不按照固定順序的情況。若使用者偶發地繞回了某個剛被替換的物件前，對記憶體而言需再重複放進將剛替換的 AR 物件，效能也因此未達到預期成果。

因此，本研究考量使用者行為與應用需求等相關因素，提出的改良 LRU 快取替

換演算法如圖十。

改良 LRU 快取替換演算法：

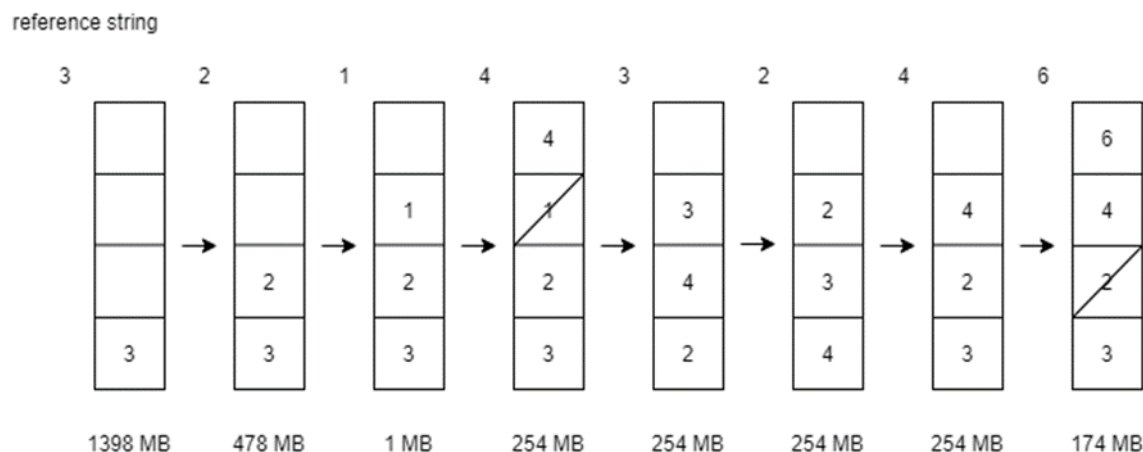


圖 十，改良 LRU 快取替換演算法

圖十為改良 LRU 快取替換演算法，此處使用了與傳統 LRU 快取替換演算法(圖八)相同的 AR 物件序列。以距離與時間參數作為考量進行模擬。當物件 4 欲加入快取之中，因考量物件 4 與物件 3 距離相近，物件 3 被使用者點擊機率大增，因此不將最久未被使用的物件 3 替換出快取，改為替換掉距離與時間參數比較下排第二順位的物件 1。同樣物件序列中，也能看出改良的 LRU 快取替換演算法，因替換的物件不同，使後面的物件序列也避免了不必要的重複置換。此案例在難以預測使用者動向的室內行動 AR 應用之中能有效減低因欠缺距離考量導致系統效能減低的情形。同時也在 4-1 實驗設計中討論在不同情形下，該以何種方式有效應用改良的 LRU 快取替換演算法在普遍行動 AR 裝置中。

3-2 預存機制

除了改良 LRU 快取替換演算法，也探討了預存機制，期望能有效配置 AR 資源。

此預存機制根據 AR 物件大小、裝置快取大小、時間參數、行動裝置與 AR 物件距離進行分類。應用環境設定為使用公用行動裝置。

在進入改良 LRU 快取替換演算法前，先初步透過使用者手持行動裝置的位置進行分類。在該使用者尚未進入預存考量範圍內，即不進行預存動作。倘若使用者進入可能預存範圍內，會依據裝置剩餘快取容量決定是否進行預存動作。當剩餘快取容量足夠時，直接進行預存並加入快取；當剩下快取容量不足時，AR 物件配合改良的 LRU 快取替換演算法進行替換。因此分析出兩種不同處理方法：I.進行預存，並加進快取中 II.不進行預存，不加進快取，與四種情境：一、AR 物件在預存範圍外時，不預存及快取；二、AR 物件在預存範圍內，剩餘快取容量足夠，預存並加進快取中；三、AR 物件在預存範圍內，剩餘快取容量不足，改良 LRU 快取替換演算法判斷下，預存並替換進快取；四、AR 物件在預存範圍內，剩餘快取容量不足，改良 LRU 快取替換演算法判斷下，不預存及不替換進快取。表格如下：

表格 一，預存及快取機制兩種處理方式

處理方式	
1	不加進快取，不進行預存
2	加進快取，並且進行預存

表格 二，預存及快取機制四種情境

情境	處理方式
一、AR 物件在預存範圍外	1
二、AR 物件在預存範圍內，剩餘快取容量足夠	2
三、AR 物件在預存範圍內，剩餘快取容量不足，改良 LRU 演算法判斷	1
四、AR 物件在預存範圍內，剩餘快取容量不足，改良 LRU 演算法判斷	2

快取管理中，預存機制屬於應用開發環節所探討的技術，開發者會透過預存機制決定哪些是需要提前被存置的資源。而快取替換為系統管理中重要的一部分，能有

效提升系統運行速度。因此在本論文中，除了在此小節對預存技術進行分類外，實驗中假設不考慮預存機制，只考慮行動 AR 的快取替換，以傳統及改良 LRU 快取替換演算法進行系統的分析研究。



第四章 實驗與分析

本論文以程式模擬預存及改良 LRU 快取替換演算法，分別以輔大醫院生命之樹案例及隨機案例模擬不同場域中多變的 AR 物件大小、裝置移動順序……。為了驗證本研究提出的公式，實驗也用生命之樹走過的實際物件序列與隨機模擬的 100000 組物件序列，並用快取命中率[21]進行分析，比較不同時間及距離參數下的改良 LRU 快取替換演算法與傳統 LRU 快取替換演算法結果。

4-1 實驗設計

本研究使用輔大醫院生命之樹瓷浮雕壁畫衍生的 AR 應用程式做為實驗題材，Xcode 作為實驗平台，Swift 程式語言撰寫模擬改良的 LRU 快取替換演算法。以此個案實驗改良 LRU 快取替換演算法，並解決行動 AR 裝置在應用執行中因資源配置不均產生的延遲問題。除了順利優化此行動 AR 個案之外，也讓此快取管理機制應用至大部分室內行動 AR 裝置。

為了讓本研究設計的行動 AR 快取管理能有效應用在普遍室內行動 AR 裝置中，實驗部分也推衍出以時間與距離參數兩者作為考量的改良 LRU 演算法替換優先權重公式。用快取命中率比較傳統 LRU 演算法及改良 LRU 演算法差別，並以不同實驗說明在相同物件個數、物件大小、物件間距離、物件序列時間差、裝置快取容量，不同物件序列情形的可能性。

實驗一

A. 傳統 LRU 快取替換演算法：

傳統 LRU 演算法如圖九所示，AR 物件序列為 3->2->1->4->3->2->4->6，以表格三呈現有物件替換時，只受時間參數影響的優先權重。優先權重最高者與目標物件進行替換。

表格 三，傳統 LRU 快取替換演算法的優先權重

		AR 物件距離 上次被使用時間(sec)	LRU 替換優先權重
欲放置物件 4	物件 1	200	1
	物件 2	202	2
	物件 3	252	3
欲放置物件 3	物件 1	300	2
	物件 2	302	3
	物件 4	100	1
欲放置物件 2	物件 1	303	3
	物件 3	3	1
	物件 4	103	2
欲放置物件 6	物件 2	80	2
	物件 3	83	3
	物件 4	20	1
快取命中率			12.5%

B. 改良 LRU 快取替換演算法：

同樣實驗在使用預存及改良 LRU 快取替換演算法時，如圖十所示。以表格四呈現套用本研究設計的改良優先權重公式計算情形。有別於傳統 LRU 快取替換演算法，此考量距離的改良 LRU 快取替換演算法降低了替換 AR 物件的次數。原因為加入了距離參數，使得在傳統 LRU 替換演算法中最久未被使用的物件 3 不被替換，改成替換物件 1。當使用者受習慣的移動路線影響，欲接著拿取距離較近的資源(物件 3)時，便能在快取中直接找到。

表格 四，改良 LRU 快取替換演算法的優先權重

		U / T	D / S	LRU 替換優先權重
欲放置物件 4	物件 1	200sec/252sec	19m/47m	0.321
	物件 2	202sec/252sec	14m/47m	0.238
	物件 3	252sec/252sec	4m/47m	0.085
欲放置物件 6	物件 2	80sec/432sec	42m/47m	0.165
	物件 3	83sec/432sec	32m/47m	0.131
	物件 4	20sec/432sec	28m/47m	0.028
快取命中率			37.5%	

實驗二

A. 傳統 LRU 快取替換演算法：

實驗二以隨機物件序列 1->1->2->5->3->1->1->3 為例，圖十一、表格五為使用傳統 LRU 替換演算法，各 AR 物件與快取的情形。圖十二、表格六為使用改良 LRU 快取替換演算法，各 AR 物件與快取的情形。

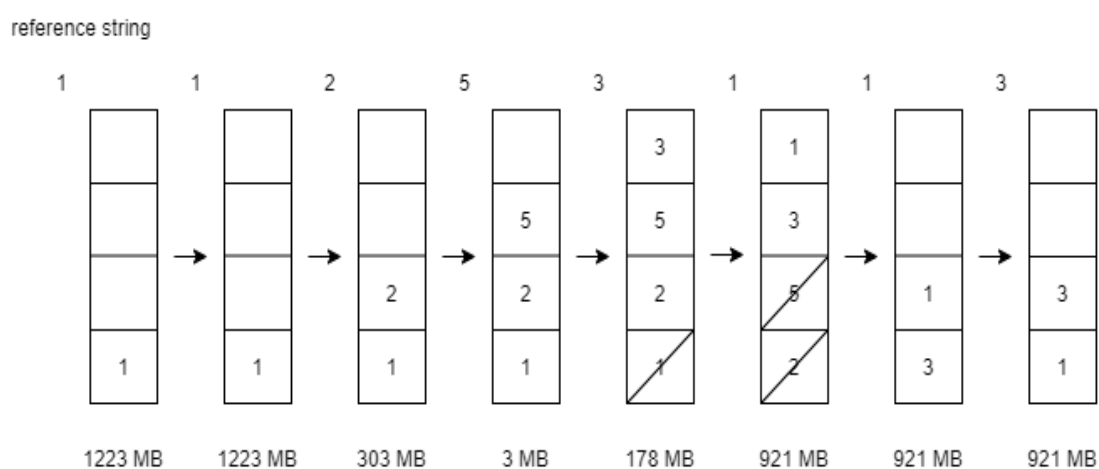


圖 十一，傳統 LRU 替換快取演算法

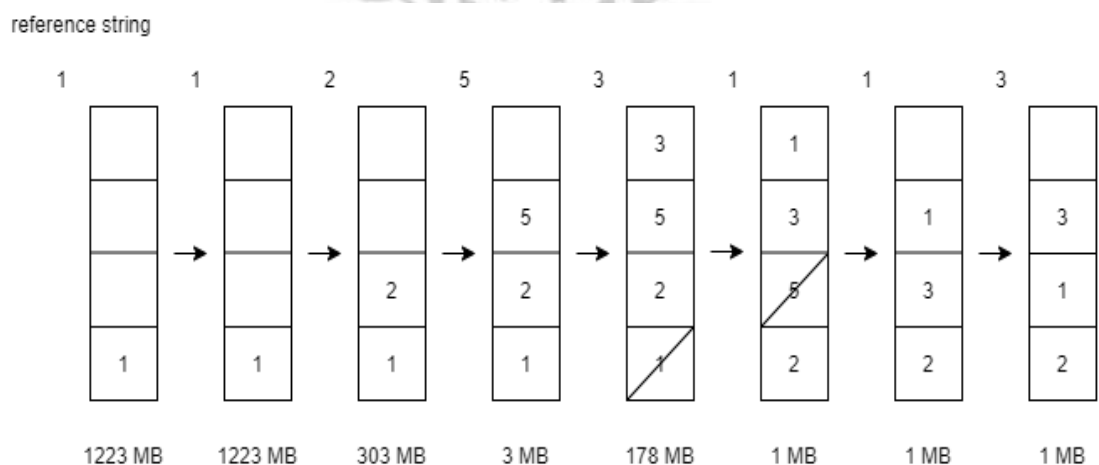


圖 十二，改良 LRU 替換快取演算法

表格 五，傳統 LRU 快取替換演算法

		AR 物件距離上次被使用時間(sec)	LRU 替換優先權重
欲放置物件 3	物件 1	302	3
	物件 2	300	2
	物件 5	100	1
欲放置物件 1	物件 2	303	3
	物件 3	3	1
	物件 5	103	2
快取命中率		37.5%	

表格 六，改良 LRU 快取替換演算法

		U / T	D / S	LRU 替換優先權重
欲放置物件 3	物件 1	302sec/352sec	15m/47m	0.274
	物件 2	300sec/352sec	10m/47m	0.181
	物件 5	100sec/352sec	12m/47m	0.073
欲放置物件 1	物件 2	303sec/355sec	5m/47m	0.091
	物件 3	3sec/355sec	15m/47m	0.003
	物件 5	103sec/355sec	27m/47m	0.167
快取命中率		37.5%		

比較實驗一、二在傳統及改良 LRU 快取替換演算法使用公式計算，可以發現當物件個數、裝置快取總容量、AR 物件大小、物件間距離、物件序列間時間相同，物件序列不同時，改良的 LRU 快取替換演算法在快取命中率會有兩種表現：I.有提升(案例一：提升 25%) II.沒有影響(案例二：提升 0%)。在 4-2 實驗分析部分詳細說明。



4-2 實驗分析

本章節採用傳統 LRU 快取替換演算法、改良 LRU 快取替換演算法的快取命中率綜合比較距離參數、時間參數是否變動時的差異。以 4-1 提到的實驗一(較好的物件序列)、實驗二(較壞的物件序列)兩種不同的物件順序進行分析。說明在物件個數、裝置快取總容量、AR 物件大小相同時，距離參數與時間參數不同帶來實驗測試的結果。

A、B 為距離參數固定、時間參數改變下，以實驗一(較好的物件序列)為基準，用程式隨機模擬執行 100000 組不同的時間參數所得到之樣本。其中在大多數情形(樣本比例為 76%)下會產生分析結果 A，而在少數情形(樣本比例為 24%)下會產生分析結果 B。而 C、D 則為距離參數改變、時間參數固定下，同樣以實驗一(較好的物件序列)為基準，讓程式隨機模擬執行 100000 組不同的距離參數所得到之樣本。其中在大多數情形(樣本比例為 93%)下會產生分析結果 C，而在少數情形(樣本比例為 7%)下會產生分析結果 D。

以下可透過表格七、八看到在距離參數與時間參數不同時，用程式隨機模擬執行的樣本比例。

表格 七，距離參數固定的參數分析(樣本數量:100000 組)

	距離參數	時間參數	樣本比例
A	固定	改變	76%
B	固定	改變	24%

表格 八，時間參數固定的參數分析(樣本數量:100000 組)

	距離參數	時間參數	樣本比例
C	改變	固定	93%
D	改變	固定	7%

實驗一：

採用輔大醫院生命之樹 AR 應用程式的物件序列，物件 3->物件 2->物件 1->物件 4->物件 3->物件 2->物件 4->物件 6。圖十三呈現實驗一在傳統 LRU 快取替換演算法、改良 LRU 快取替換演算法的快取命中率差別。

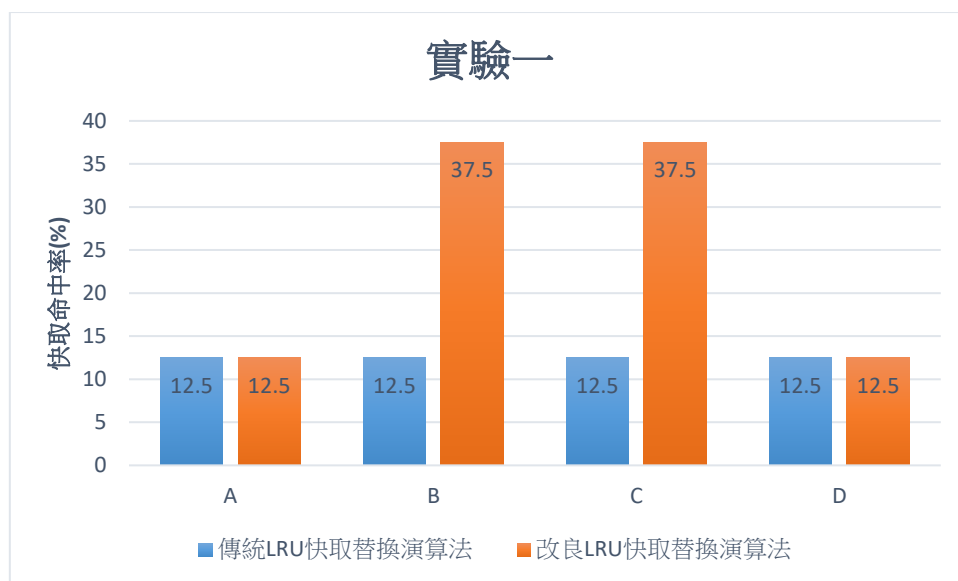


圖 十三，實驗一快取命中率比較

參照表格七、八與圖十三可見，在此較好的物件序列中，在樣本比例較多(A、C)時，因距離參數的改變導致快取命中率有顯著的進步(C)；而在樣本比例較少(B、D)時，會因時間參數的改變使得快取命中率進步(D)。

實驗二：

採用模擬程式隨機產生的其中一個物件序列，物件 1->物件 1->物件 2->物件 5->物件 3->物件 1->物件 1->物件 3。圖十四呈現的是實驗二在傳統 LRU 快取替換演算法、改良 LRU 快取替換演算法的快取命中率差別。

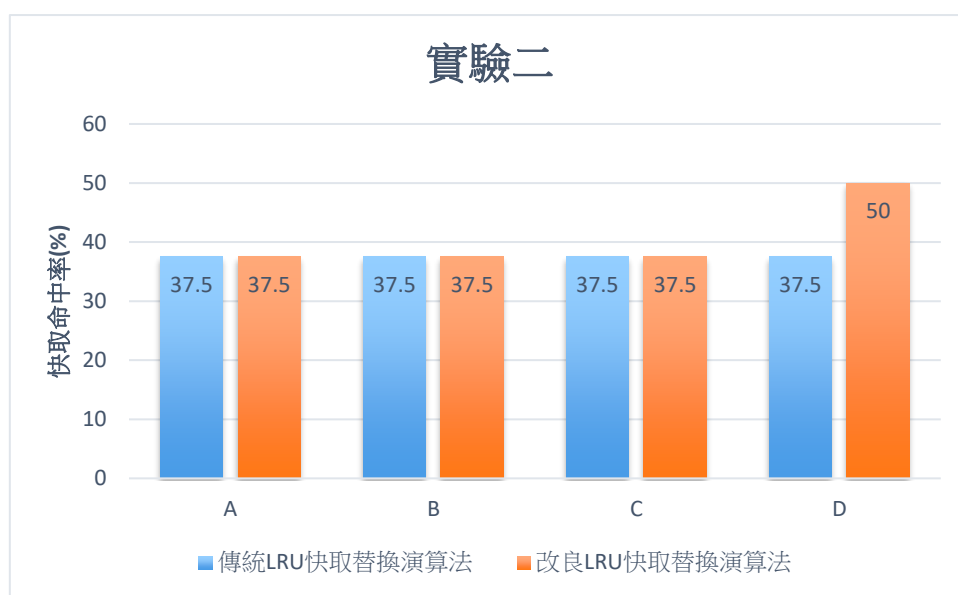


圖 十四，實驗二快取命中率比較

參照表格七、八與圖十四可見，在此較不好的物件序列中，只改變時間參數(A、B)並不會給改良 LRU 的快取命中率帶來影響，主要是因為物件序列緣故，時間參數 = (物件距離上次被使用的時間/目前總花費時間)，此參數受序列影響。故在實驗二中，時間參數並不會造成快取命中率提升。在樣本比例較少(B、D)時，會因距離參數的改變使得快取命中率顯著提升(D)。

在模擬執行 100000 組結果後，本研究取實驗一、二進行結果分析。此兩種實驗結果中也可見，在本實驗所進行的模擬改良 LRU 快取替換演算法中，最差的快取命中率結果也會與傳統的 LRU 快取替換演算法數值相同。顯示在本實驗模擬結果中，改良 LRU 演算法優於傳統 LRU 演算法的表現。許多案例中也可發現改良 LRU 快取替換演算法相較傳統 LRU 快取替換演算法在快取命中率上有明顯的提升。



第五章 結論及未來展望

5-1 結論

在現代行動 AR 技術逐漸普遍應用在遊戲、文創、導覽等領域下，因 AR 本身耗費較多運算資源，而使用者在行動 AR 體驗時動向往往無法預測。行動裝置能力雖不斷增強，但仍須負荷起龐大 AR 應用帶來的儲存及運算問題。如何搶先存置未來可能會用到的 AR 資源，降低使用上延遲的負面體驗，就成為現階段重要的議題。

本論文以「輔大醫院生命之樹瓷浮雕壁畫」為例，設計一預存及改良 LRU 替換快取管理機制。蒐集瓷浮雕壁畫 AR 應用程式中實際物件相關參數並模擬改良 LRU 快取替換機制。特別在改良 LRU 快取替換演算法中加入距離相關參數，並以模擬程式同步執行傳統與改良 LRU 快取替換演算法。了解在同一 AR 應用情況中，本機制對行動 AR 體驗帶來快取命中率的提升。

此外，本研究也提出能普遍應用於室內行動 AR 裝置的優先權重替換公式。在室內行動 AR 應用中，常因使用者習慣、動向不一致等難以預估的行動，產生變化多端的距離與時間。在此複雜的情形下，有極大可能需要較適合動態的改良 LRU 快取替換演算法，可有效降低因最久時間未被使用被置換，但使用者卻因距離近而欲重新拿取該資源的耗費情形。此預存及改良 LRU 快取管理機制不只適用於「輔大醫院生命之樹瓷浮雕壁畫」AR 應用程式，更因應於不同場域下多變的室內行動 AR 體驗。透過動態資源配置，未來高機率可能使用的物件不需重複載入，使用效能因

此提升。



5-2 未來展望

應用在更多實際 AR 應用程式：

本研究目前以「輔大醫院生命之樹瓷浮雕壁畫」應用程式為案例實驗，也提出了一適用於室內行動 AR 的預存與快取管理機制。除了在此案例中有效提升快取命中率，並解決行動 AR 的延遲問題外。也期望此行動 AR 快取管理能實際應用在未來多變的行動 AR 體驗中，在不同情境下以相同模式，使效能有效提升，降低延遲情形。

預測使用者動向：

室內與室外行動 AR 差別甚大。使用者動向的未知對 AR 應用程式運算儲存帶來很大的負擔。期望未來能透過數據分析室內行動 AR 的使用者行為，了解不同類型應用中使用者的不同移動方式，藉此在改良演算法中，優化出基於使用者體驗更佳的效能提升方法。

改善體驗延遲負面影響：

因使用者受延遲問題而卸載 AR 應用程式情形屢見不鮮。不論在遊戲、導覽、亦或其他行動 AR 體驗中，皆希望能準確了解應用正在延遲。期望給使用者更多元的體驗，藉此改善無法操作帶來的負面影響。

參考文獻

- [1] H. Ghasemzadeh, S. Mazrouee, and M. R. Kakoei, "Modified pseudo LRU replacement algorithm," 13th Annual IEEE International Symposium and Workshop on Engineering of Computer-Based Systems (ECBS'06), Potsdam, pp. 6-376, 2006.
- [2] R. Pendse and R. Bhagavathula, "Performance of LRU block replacement algorithm with pre-fetching," 1998 Midwest Symposium on Circuits and Systems (Cat. No. 98CB36268), Notre Dame, IN, USA, pp. 86-89, 1998.
- [3] M. P. Pamungkas, S. A. Ekawibowo, and N. R. Syambas, "Priority Based Multilevel Cache LRU On Named Data Network," 2019 IEEE 5th International Conference on Wireless and Telematics (ICWT), Yogyakarta, Indonesi, pp. 1-4, 2019.
- [4] National Development Council, " AR/VR 團隊屢獲國際關注 為臺灣新創開啟世界大門",
[Online].https://www.ndc.gov.tw/News_Content.aspx?n=114AAE178CD95D4C&s=04E0607E964532EE.
- [5] Ronald T. Azuma, "A Survey of Augmented Reality," Teleoperators and Virtual Environments, pp. 355-385, 1997.
- [6] J. Grubert, T. Langlotz, S. Zollmann, and H. Regenbrecht, "Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality," IEEE Transactions on Visualization and Computer Graphics, pp. 1706-1724, 2017.
- [7] A Annafi1, D L Hakim1, and D Rohendi1, "Impact of using augmented reality applications in the educational environment," Annual Conference of Science and Technology, Malang, Indonesia, Volume 1375, 2018.
- [8] M. Lorenz, S. Knopp, and P. Klimant, "Industrial Augmented Reality: Requirements for an Augmented Reality Maintenance Worker Support System," 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct, Munich, Germany, pp. 151-153, 2018.
- [9] A. Costa, R. Lima, and S. Tamayo, "Eva: A Virtual pet in Augmented Reality," 2019 21st Symposium on Virtual and Augmented Reality (SVR), Rio de Janeiro, Brazil,

pp. 47-51, 2019.

- [10] X. Qiao, P. Ren, S. Dustdar, and J. Chen, "A New Era for Web AR with Mobile Edge Computing," *IEEE Internet Computing*, pp. 46-55, 2018.
- [11] Rachel Swatman, "Pokémon Go catches five new world records",
[Online].<https://www.guinnessworldrecords.com/news/2016/8/pokemon-go-catches-five-world-records-439327>.
- [12] Z. I. Bhutta, S. Umm-e-Hani, and I. Tariq, "The next problems to solve in augmented reality," 2015 International Conference on Information and Communication Technologies (ICICT), Karachi, pp. 1-4, 2015.
- [13] E. Alwagait and S. Ghandeharizadch, "A comparison of alternative Web service allocation and scheduling policies," *IEEE International Conference on Services Computing*, Shanghai, China, pp. 319-326, 2004.
- [14] S. Du, C. Li, X. Mao, and W. Yan, "The Optimization of LRU Algorithm Based on Pre-Selection and Cache Prefetching of Files in Hybrid Cloud," 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Guangzhou, pp. 125-132, 2016.
- [15] C. C. Kavar and S. S. Parmar, "Improve the performance of LRU page replacement algorithm using augmentation of data structure," 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Tiruchengode, pp. 1-5, 2013.
- [16] T. Ma, Y. Hao, W. Shen, Y. Tian, and M. Al-Rodhaan, "An Improved Web Cache Replacement Algorithm Based on Weighting and Cost," *IEEE Access*, pp. 27010-27017, 2018.
- [17] C. Wu, A. Jaleel, M. Martonosi, S. C. Steely, and J. Emer, "PACMan: Prefetch-Aware Cache Management for high performance caching," 2011 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), Porto Alegre, pp. 442-453, 2011.
- [18] L. Wu and X. Yu, "Outdoor Navigation with Handheld Augmented Reality," 2018

International Conference on Audio, Language and Image Processing (ICALIP), Shanghai, pp. 237-241, 2018.

- [19] N. C. Nachiappan, A. K. Mishra, M. Kandemir, A. Sivasubramaniam, O. Mutlu, and C. R. Das, "Application-aware prefetch prioritization in on-chip networks," 2012 21st International Conference on Parallel Architectures and Compilation Techniques (PACT), Minneapolis, MN, pp. 441-442, 2012.
- [20] Mr.C.C.Kavar and Mr. S.S.Parmar, "Performance Analysis of LRU Page Replacement Algorithm with Reference to different Data Structure," International Journal of Engineering Research and Applications (IJERA), Wadhwan,Gujarat,India, pp.2070-2076, 2013.
- [21] Weng Meizhen, Shang Yanlei, and Tian Yue, "The design and implementation of LRU-based web cache," 2013 8th International Conference on Communications and Networking in China (CHINACOM), Guilin, pp. 400-404, 2013.

