

國立臺灣師範大學應用電子科技學系

碩士論文

指導教授：高文忠 教授

基於自我運動速度估測分析之數位影像穩定器

Digital Image Stabilizer Based on Ego-motion Velocity Analysis



研究生：黃紹綱 撰

中華民國一零二年八月

基於自我運動速度估測分析之數位影像穩定器

學生：黃紹綱

指導教授：高文忠

國立臺灣師範大學應用電子科技學系碩士班

摘 要

數位影像穩定器，主要為了移除影片中非預期的運動向量，是應用在手持相機錄影或行車紀錄器上一個理想相機必須擁有的功能。優秀的數位影像穩定器應能濾除抖動運動向量，但仍能保留所欲的運動向量。

在本文中，我們基於多個幀之間運動向量算得的速度來進行分析，並提出一個新的數位影像穩定器。透過更準確的自我運動向量估計演算法，我們處理的影片成果能比其他已提出方法有更加的視覺品質。

關鍵字：數位影像穩定器、自我運動向量估計演算法

Digital Image Stabilizer Based on Ego-motion Motion Velocity Analysis

Student: Shao-Kang Huang

Advisors : Dr. Wen-Chung Kao

Department of Applied Electronics Technology
National Taiwan Normal University

ABSTRACT

The digital image stabilization (DIS), which removes unwanted motions, is a desirable camera capability for taking videos with a hand-held camera or a recorder installed on moving vehicles. A good DIS should be able to remove jitter motions but still remains the intentional ego-motion. In this thesis, we present a new DIS based on motion velocity analysis among several successive frames. By relying on a more accurate ego-motion estimation algorithm, the processed video shows better visual quality than other published works.

Key word: digital image stabilizer, ego-motion estimation.

誌 謝

誠摯地感謝指導教授高文忠博士，給予我學業和研究上的指導。特別是在做學問的態度和研究方法上，讓我的知識和人生經驗得以在碩士生這兩年間突飛猛進，進步神速。老師嚴謹治學的做事態度是學生們最佳的典範，我以身為其指導學生為榮。此外我要感謝口試中給我指導與建議的李佩君教授和陳彥霖教授，使本論文可以更加完善。還要感謝國立台灣師範大學的老師們，特別是吳順德教授和其指導的研究生吳求文同學，感謝他們在訊號處理部分的幫助讓我獲益匪淺。

感謝葉嘉安、劉人瑞學長與黃湘婷、劉奕珣學姊，給我學習上的照顧與關懷。感謝張維德、劉書呈、王泓淳同學帶給我充實有趣的研究生活以及學術上的切磋學習和相互砥礪。同時，我還要謝謝學弟妹，劉建輝、吳昇儒、鄞詩堯、陳星羽、柳英浩、邱大洲、廖重涓、杜曉玫、吳禎庭、李家宜與助理陳思婷、劉麗燕在研究上的諸多協助。

我也要感謝我的家人，父母親還有弟弟的支持與愛，有了你們的鼓勵，我才能在攻讀碩士期間無後顧之憂地精實自我。當然還要感謝所有支持我的親朋好友們，在研究生活中帶給我充實的人生經歷與生活樂趣。最後再次感謝所有曾經幫助過我的人，紹綱非常謝謝你們的支持和鼓勵，必定會在未來人生和學術研究更加努力。

黃紹綱 謹誌
于國立台灣師範大學系統晶片實驗室
2013年7月

目 錄

中文摘要	ii
英文摘要	iii
誌 謝	iv
目 錄	v
圖目錄	vii
表目錄	ix
第一章 緒論	1
1.1 研究動機及背景	1
1.2 相關問題概述	5
1.3 本文研究方法	8
第二章 相關研究探討	9
2.1 影像穩定器系統基本架構與流程	9
2.2 區域運動向量估計研究	10
2.3 全域運動向量估計研究	15
2.4 補償值估計演算法探討	17
2.5 影像補償之簡介	22
第三章 數位影像穩定器演算法系統架構	24
3.1 系統流程方塊圖	24
3.2 背景區域之篩選	25
3.3 權重式移動平均系統簡介	26
第四章 系統各模組演算法探討	28

4.1	區域運動向量估計	28
4.2	全域運動向量估計	31
4.3	運動向量補償值估計	36
第五章	研究結果	55
5.1	實驗環境及說明	55
5.2	影像穩定成果比較	56
第六章	結論與未來展望	75
6.1	結論	75
6.2	未來展望	76
參考文獻	77
附錄	80

圖目錄

圖 1-1	光學影像穩定器產品範例之 Nikon VR 系統結構.....	3
圖 1-2	鏡頭內嵌式光學影像穩定器結構.....	4
圖 1-3	數位影像穩定器輸出解析度下降示意圖.....	6
圖 2-1	影像穩定器架構.....	9
圖 2-2	區塊比對法示意圖.....	11
圖 2-3	區域運動向量估計後的各類相機鏡頭運動狀態.....	16
圖 2-4	抖動成分被當作自我運動之現象.....	19
圖 2-5	回授型 PI 控制器.....	22
圖 3-1	本文之數位影像穩定器演算法系統.....	24
圖 3-2	本文之全域運動向量估計法輸入資訊方塊圖.....	25
圖 4-1	本文之區域運動向量估計法流程圖.....	28
圖 4-2	拱門狀區域設計示意圖.....	29
圖 4-3	本文之全域運動向量估計法流程圖.....	32
圖 4-4	本文之移動平均式區段結構.....	37
圖 4-5	本文之重疊區段結構設計.....	39
圖 4-6	本文之補償值運動向量估計演算法.....	40
圖 4-7	各時刻區段個數計算之演算法流程.....	42
圖 4-8	區段預期速度和代表性偏移演算法方塊圖.....	43
圖 4-9	代表點取得之演算法流程.....	44

圖 4-10	非邊界取樣點之代表值計算.....	47
圖 4-11	區段之預期速度與代表性偏移計算流程.....	50
圖 4-12	區段不穩定值計算範例.....	52
圖 4-13	區段不穩定值之設計意義.....	53
圖 5-1	模擬實驗之實驗環境設計.....	56
圖 5-2	實驗拍攝用之相機.....	57
圖 5-3	拍攝相機與手推車架設.....	57
圖 5-4	本文之位置—幀二維影像穩定結果截圖.....	58
圖 5-5	抖動向量與理想位置差異造成的偏移現象.....	60
圖 5-6	後處理與預測型穩定器所擁有的全域資訊範圍差異.....	61
圖 5-7	有加速度補償的本文方法穩定結果.....	71
圖 5-8	無加速度補償的本文方法穩定結果.....	72
圖 A-1	區段代表性偏移最接近當下時間點之全域運動向量資訊.....	81
圖 A-2	區段代表性偏移最遠離當下時間點之全域運動向量資訊.....	82
圖 B-1	模擬實驗下本文方法影像穩定影片之截圖範例一.....	84
圖 B-2	模擬實驗下本文方法影像穩定影片之截圖範例二.....	85
圖 B-3	實景拍攝下本文方法影像穩定影片之截圖範例.....	86

表目錄

表 1-1	影像穩定器分類.....	2
表 5-1	各補償值計算方法模擬實驗之 RMSE 值結果表.....	59
表 5-2	各補償值計算方法模擬實驗之平均 RMSE 值比較表.....	60
表 5-3	各補償值計算方法於模擬實驗一號影片之 SI 值結果表.....	62
表 5-4	各補償值計算方法模擬實驗之平均 SI 值比較表.....	62
表 5-5	各補償值計算方法模擬實驗之 SI 值結果表.....	63
表 5-6	各補償值計算方法真實拍攝之平均 SI 值比較表.....	63
表 5-7	各補償值計算方法真實拍攝之 SI 值結果表.....	64
表 5-8	各補償值計算方法以平滑訊號輸入之平均 RMSE 值比較表.....	64
表 5-9	各補償值計算方法以平滑訊號輸入之平均 SI 比較表.....	65
表 5-10	各補償值計算方法之執行速度比較表.....	66
表 5-11	各補償值演算法整體比較表.....	66
表 5-12	本文補償值演算法不同參數下之 RMSE 值比較表.....	68
表 5-13	本文補償值演算法不同參數下之 SI 值比較表.....	69
表 5-14	各補償值演算法模擬實驗之 RMSE 值結果表.....	72
表 5-15	各補償值演算法模擬實驗之 SI 值結果表.....	73
表 5-16	各補償值演算法之整體模擬總比較表.....	73

第一章 緒論



1.1 研究動機及背景

自人類開始使用影片紀錄和傳遞訊息，諸如 DV、數位相機、手機相機等手持錄影設備推陳出新，對手持錄影設備影像品質的需求逐年升高。科技的飛速進步和商業上的激烈競爭，使得人們追求手持錄影影片的高解析度(Resolution)之餘，同時要求消費性電子產品能隨著摩爾定律(Moore's law)的影響能更加輕量化，讓手持錄影設備逐漸朝著輕薄短小的方向不斷演進。然而，使用者手持錄影設備進行影片的錄製，相對於相機腳架等輔具使用的情況，純粹手持使用的拍攝過程難免會因為雙手穩定性不夠，讓錄製的影片有非預期的抖動發生。逐年輕量化手持錄影設備會加劇物理環境搖晃在攝影過程的影響，造成輸出影片的晃動程度上升。

早期攝影師面對這樣的問題以使用三腳架、調整快門速度、或使用低感光度底片等多種技巧來迴避抖動造成的問題，但是一般使用者不一定具備這些特殊拍攝技巧或輔具。除此之外，輔具的使用本身便與追求電子產品輕量化的設計初衷相違背。舉例來說，若使用手機拍照或錄影時還要額外攜帶腳架來安裝對於使用者而言是相當不方便的負擔，何況臨時或緊急拍攝不可能還要使用者考慮這些額外設備的有無。我們希望以不額外增加相機本身之外的輔具為前提，解決手持抖動的問題。因此，研究的範圍將琢磨於在手持攝影器材上以軟硬體方式濾除抖動現象。本文所研究的對象—影像穩定器(Image stabilizer, IS)，就是為了濾除抖動現象而出現並逐漸受到學界和業界的重視。

影像穩定器的功能主要在於濾除所錄製影片中因為相機系統外在物理環境造成的非預期抖動運動向量(Jitter motion)，並盡可能還原使用者自我運動向量(Ego motion)。這個濾除抖動現象的穩定器系統，主要輸入參數為影片中幀(Frame)之間的數位影像資訊或外部裝置測得的物理性三軸變化。根據這些資訊得出相機可能的運行軌跡，我們可以透過硬體機構或是數位影像方式來濾除屬於抖動成分的運動向量並還原預期影片。

近年來，影像穩定器依照不同的輸入結構分成三個大類，分別為光學影像穩定器(Optical Image Stabilizer, OIS)[1][2][3][4][5][6][7]、電子影像穩定器(Electronic Image Stabilizer, EIS)[8]、數位影像穩定器(Digital Image Stabilizer, DIS)[9][10][11][12][13][14][15][16][17][18][19]。這三類中，OIS 本身擁有一個機械結構，該結構可動態調整入射光線折射後的成像位置來穩定成像。OIS 相機系統，尚可再搭配上可判讀本體三維運動變化的陀螺儀(Gyro sensor)，運用陀螺儀分析的三維變化資訊，經量化後傳遞給該可調整式機械結構改變相機鏡頭組。鏡頭組將依位移資訊而改變入射光在感測器上的成像位置來補償影像的非預期抖動；電子影像穩定器則相較 OIS 缺少這類可調式機械結構，主要是利用相機感測器傳來的數位影像資訊，配合陀螺儀感測器得出的三維空間數據當作參考值進行影像穩定。最後，本文研究的數位影像穩定器，並沒有使用任何外部輔助工具，屬於較低成本的影像穩定器，全然以影片中切割的各個幀(Frame)影像資訊為輸入來進行影像穩定。

表1-1 影像穩定器分類

Type	Gyro-sensor	Adjustment mechanism
OIS	Supported	Supported
EIS	Supported	Not-supported
DIS	Not-supported	Not-supported

市售產品中，最早的防手震系統為 1994 年 Nikon 的穩定器功能系統 Vibration Reduction (VR)，VR 以 35mm 大小的光學影像穩定器鏡頭裝置使用在該公司旗下商品相機 Nikon zoom 700VR(銷往美洲的名稱為 Zoom-Touch 105 VR)，亦為第一個光學影像穩定器在消費型電子上的應用實例。但是當時 Nikon zoom 700VR 製造不易，且市場上因為其產品的高單價讓銷售情況不佳，Nikon 因而較晚繼續推出後續光學影像穩定器鏡頭等相關影像穩定器功能商品。

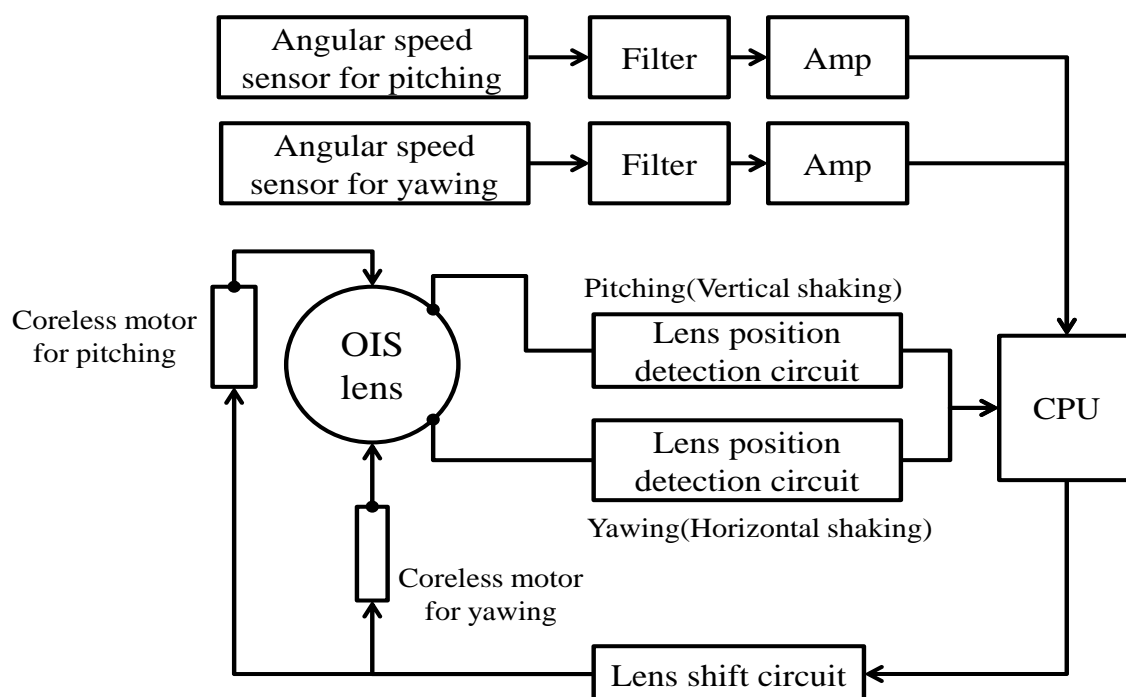


圖1-1 光學影像穩定器產品範例之 Nikon VR 系統結構[5]

隔年(1995 年)，佳能 Canon 首次推出直接將光學影像穩定器功能做在變焦鏡頭上的商品 Canon EF75 - 300 mm f/4-5.6 IS USM 大受市場好評，奠定其在影像穩定器功能相機的市場地位。光學影像穩定器本身在鏡頭結構中有一影像穩定單位(IS unit)，影像穩定單位面對鏡頭在水平垂直向位置變化而改變其空間位置讓成像能補回視覺上的連續穩定影像，目前市售商品常以「光學防手震功能」稱之。光學影像穩定器主要可分為彈性稜鏡式(Flexible prism)[1][2]、鏡頭內嵌式(In-lens)[4][5][6]、相機內嵌式(In-camera)。彈性稜鏡式是在以一個填滿液體且光

滑平面的稜鏡當作補償機構，當相機偏移時稜鏡會產生形變造成入射光線重新匯聚補回預期成像。由於此法不做偏移資訊的主動偵測，而僅被動地接受三維軸變化所帶來的形變，所以被稱為「被動式光學影像穩定器」；鏡頭內嵌式，鏡頭模組中夾入一片可移動的穩定單元。相機在空間中因應感測器所偵測角度和位移的不同，系統會主動計算後得出穩定單元要補償的相應偏移資訊並調整機構進行穩定，Nikon 的 VR 系統即為此型；相機內嵌式則是同鏡頭內嵌式的原理，只是補償機構更改為移動影像感測器本身而非使用穩定單元。

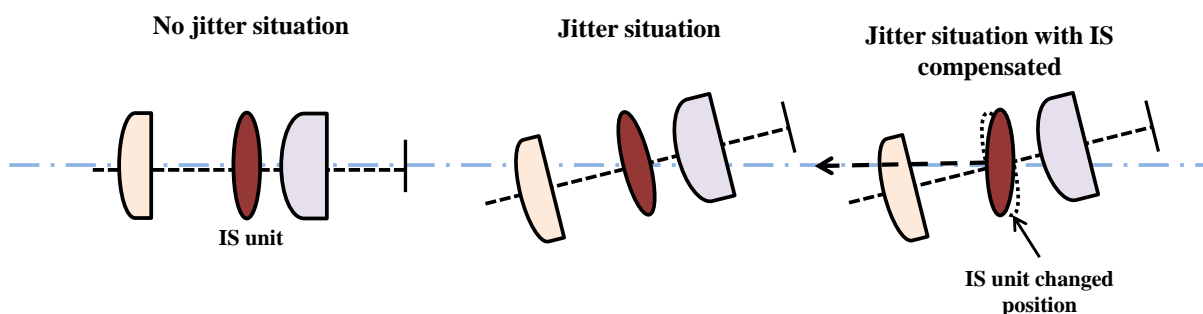


圖1-2 鏡頭內嵌式光學影像穩定器結構[4]

數位影像穩定器(DIS)採不使用其他元件的設計方式，以純粹影像感測器擷取的影像資訊進行影像穩定。數位影像穩定器具有較低製造成本而被廣泛研究，學術上極為重視數位影像穩定器的設計。其中不論何種維度空間的數位影像穩定器，大多數的設計集中於兩部分的研究：全域運動向量估計(Global motion estimation)和影像補償(Image compensation)。前者是將感測器擷取的影像取得其空間上相機整體偏移的估計，後者則是將抖動影像補正的演算法設計。

根據數位影像穩定器在這兩部分的設計原理和應用，尚能依照不同類別而有下列分類：

- 一、依照搜尋物體移動的最大空間維度分類，數位影像穩定器可區分為處理二維或三維空間的影像穩定器。二維處理主要是針對相機在物

理空間的水平面或垂直面偏移量分析，而三維則額外增加廣角縮放與旋轉變化，其中廣角探討上會面對補償上的色彩插補考量、旋轉則會需要額外處理運算量的增加和旋轉中心如何預估等問題。

二、依演算法複雜度或所需資訊的時域範圍區分出即時處理(Real-time)式或後處理式(Off line)數位影像穩定器。高複雜度和需要大量當下時間點之後的資訊才能進行補償的數位影像穩定器演算法，各會因為硬體處理速度耗時和資訊受限而無法進行即時處理，因此會被歸類於後處理的方法，僅能預錄影片後再進行穩定器處理。相對之下，低複雜度和不需要當下時間點之後資訊就能進行補償的演算法能實現即時運算，亦較有機會在嵌入式系統(Embedded system)上隨感測器讀入影片的 Frame 資訊同時進行影像穩定。

三、依實施影像穩定的時刻分類可分出編碼端影像穩定器和解碼端影像穩定器。編碼端影像穩定器為每當感測器完成新的一張 Frame 資訊時進行影像穩定功能，因而能在錄影行為終止時刻，以連續視覺穩定的 Frame 資訊完成影片。相對的，解碼端影像穩定器則為將影片端切成 Frame 資訊後再進行影像穩定，這種方式會要考量演算法是否同時刻涉及不同取樣時間的 Frame 資訊做運算，但會因此有多出影片切割和重新壓製的耗時負擔。

1.2 相關問題概述

數位影像穩定器的問題可分為輸出解析度與系統流程兩類。第一類，數位影像穩定器本身仰賴影像感測器接收資訊進行處理和補償，輸入和輸出的資訊都受限於相同來源，造成輸出影片解析度下降。為了適當保留部分感測器接收的像

素資訊留待補償影像時運用，數位影像穩定器所輸出影片的解析度一定會下降，這是所有數位影像穩定器的共同限制。下圖中，假設原先有抖動的 Frame (寬 W 、高 H) 是較大的外圍黑實線區域，且應將該輸入 Frame 以中心位置為基準偏移 CMV 向量才為穩定狀態的 Frame 位置，為了用原有資訊補償回穩定影片，用較小的黑虛線橙色區域(寬 W_i 、高 H_i) 的像素資訊取代正中間較小的黑實線區域並輸出這樣大小的 Frame，因此輸出影片解析度必定會下降。

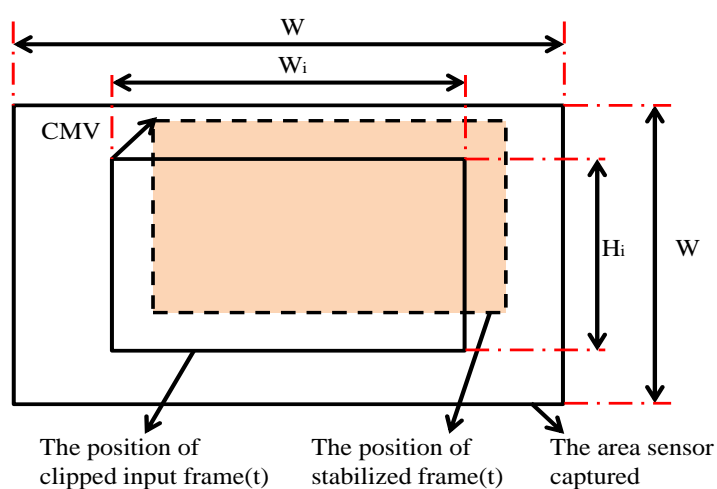


圖1-3 數位影像穩定器輸出解析度下降示意圖[28]

第二類，是數位影像穩定器系統穩定流程中要解決的問題。數位影像穩定器是由許多功能模組共同完成輸入影像的穩定過程，一個影像穩定器的品質則取決於處理後影片的結果。穩定過程從感測器隨時間統整外部訊息製成 Frame 開始，進行一系列的影像穩定處理流程。這個流程首先所要面對的是如何由 Frame 中擷取或辨識出運動向量資訊。最為常見的處理手法為區塊比對法 (Block-matching)[20][21]，我們固定前一時刻 Frame 運用目前時刻 Frame 進行相對偏移後比對像素資訊差值，再搜尋出最小差值的偏移，這個偏移量即可代表為背景運動向量。但是，整張畫面都進行比對運算會造成計算耗時嚴重，因此有些研究僅取用 Frame 中部分區塊的資訊進行比對。然而，在一張有限大小的

Frame 中，設計區域的分布也是值得探討的課題，通常針對多寡和排列方式做變化。假設區域設置越多，計算複雜度上升但是取樣較多，可以降低錯誤判斷的風險；反之，區域設置越少，計算複雜度低但是取樣較少，運動向量計算的容錯率也相對較低。區域排列上，基於考量到鄰近 Frame 的四個邊角的部分較可能是背景資訊所在，所以有上緣兩邊角排列、拱門型排列、四邊角排列等方式。還有些演算法使用所抓取特徵點的方式獲得區域運動向量資訊，哈氏角點偵測法 (Harris corner detection)、LKT 特徵搜尋法 (Lucas-Kanade-Tomashi feature tracker)[22] 即為取得圖像中較具特徵意義的像素位置點再由該點在 Frame 之間的相對位移得出運動向量。通常特徵點的計算複雜度不見得會小於區塊比對法，提升估計運動向量的準確性才是這類方法的目的。因此，針對複雜度和準確性的取捨是運動向量擷取步驟的設計者需要面臨的抉擇。

其次，搜集完區域運動向量後，我們要從眾多向量篩選出目前代表背景的全域運動向量 (Global motion vector, GMV)。篩選出全域運動向量的原因是為了要提取抖動的現象和簡化問題，而這個步驟是基於假設僅有少部分的運動向量與其他不同為前提。理想的情況下，全域運動判斷方式是直接取區域運動向量的眾數當作全域資訊。實際拍攝過程中，拍攝時旋轉發生的情況會使得單純取眾數當作求解的方法不見得完全準確，我們尚需先用演算法判斷旋轉成分出現的可能與否。所以常見的處理流程是先分析目前鏡頭狀態後排除可能有移動物體干擾的區域，再用這些區域得出 GMV 值。這當中會面對的難題在於區域運動向量影響鏡頭狀態的判定，這個部分會需要設計函數來區分出那些區域較可靠，並用它們做後續分析。

獲得 GMV 值以後，我們要給原訊號計算相應補償值以進行抖動現象平滑化的流程。這個流程主要區分為預測型和後處理型，前者僅取用當下時刻之前讀取的 GMV 資訊來做補償值估計，使用者得以邊錄影邊穩定，可以運用即時處理

(Real-time processing)，但是容易有穩定延遲現象和受極端值影像較大等問題；後者，後處理型顧名思義是影片中所有 GMV 都讀完後才做影像穩定，通常運用數學方式轉換為頻域分析再濾除雜訊，不適用於即時處理，但是穩定性較佳，且不易受極端值影響。最後，我們要用算得的補償值回饋給原訊號，讓影像能校正回預期的穩定狀態。理想上，我們直接用補償值偏移影像足以完成影像修正。但是若是涉及場景的移動物體，我們直接使用背景補償值套用會與移動物體的本身位移產生遷移，造成影像穩定誤差。此外，修補影像勢必造成輸出圖像的部分資訊遭到捨棄，影像補償步驟尚能探討影像的插補和銳利化等額外影像處理步驟以提高輸出影片的品質。計算複雜度和輸出影像品質的比重分配則是影像穩定器設計者要做的選擇。

1.3 本文研究方法

我們所提出的數位影像穩定器系統在全域運動向量的估計中運用一個可靠度演算法排除較不適宜的運動向量，在穩定抖動的演算法上利用權重式移動平均(Weighted moving average)的概念做一個重疊區段(Overlapped section)，接著進行各時間區段資訊的提取，最後線性組合區段資訊以估算出自我運動，完成影像穩定。以下簡述本文架構：第一章介紹本文研究動機和概述穩定器的相關問題，第二章將討論數位影像穩定器的基本結構和其相關的演算法研究。第三章為本文所提出的數位影像穩定器演算法系統架構以及所使用觀念；第四章為本演算法各個模組的細節探討，我們將更深入了解本系統的所用公式和處理步驟；第五章是實驗結果，我們將展示實驗環境、實驗結果與其他 DIS 方法的比較。最後，第六章為結論和本研究的未來展望。

第二章 相關研究探討

本章會介紹影像穩定器的基本結構並對組成系統的四個功能模組逐一探討。在 2.1 節介紹系統架構和流程區，2.2 節介紹如何擷取畫面中的區域向量，2.3 節介紹如何篩選區域向量資訊並求出全域向量，2.4 節介紹如何將抖動的全域向量平滑化，2.5 簡述影像補償補回穩定畫面。

2.1 影像穩定器系統基本架構與流程

基本上傳統影像穩定器的系統架構主要包含四項功能模組，分別是：區域運動向量估計(Local motion vector estimation)、全域運動向量估計(Global motion vector estimation)、運動向量補償值估計(Compensative motion vector estimation)、和影像修補(Image inpainting)等。以下，我們將先介紹系統流程，再逐步介紹各個功能模組。

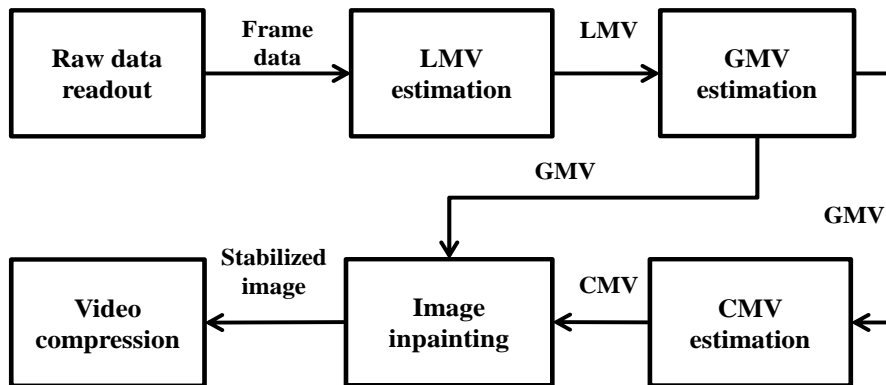


圖2-1 影像穩定器架構

影像穩定器的流程如下所示：首先，影像幀(Frame)資訊由感測器隨時間讀入外部資訊後產生；第二，接著針對 Frame 與 Frame 之間的進行 Frame 中特定區域的相對偏移量計算，此步驟稱為區域運動向量估計(Local motion vector

estimation)；第三，讀取多個區域運動向量後，我們將用全域運動向量估計演算法(Global motion vector estimation)分出代表背景的全域運動向量(Global motion vector, GMV)；第四、補償值運動向量估計(Compensative motion vector estimation)將含抖動成分的 GMV 平滑化，並算出相應的補償值；最後，用補償值修正原始影像完成影像穩定並將 Frame 資訊製成影片。

2.2 區域運動向量估計研究

區域向量估計的功能在於使用者讀入 Frame 資訊後，將所接收的影像資訊在 Frame 中特定區域裡進行運動向量的特徵擷取。區域運動向量的估計涉及整張 Frame 資訊的利用率，通常是造成系統運算複雜度提升的主因之一。根據演算利用率的的不同而分成利用率最高的全域搜尋法(Full search)和相對較低利用率的幀比對法(Frame match)。全域搜尋法顧名思義，將整張 Frame 資訊的像素資訊都納入計算區域向量的考慮範圍；幀比對方法，僅取整張 Frame 中的特定區域都做比對，所以設計者尚需設定的其區域範圍，並將各個區域中隱含的運動向量求出。以下我們將主要介紹四種 LMV 擷取方法：區塊比對法、二進制葛雷碼法、LKT 特徵搜尋法和哈氏角點偵測法。

區塊比對法(Block-matching)[20][21]是最常見的估計方法，其運用像素差值關係來獲取相對偏移向量。首先，我們在 Frame 中訂出固定的大小和位置的微區塊(Macro block, MB)區域。在相鄰時間中的兩 Frame 中，以固定其中一 Frame 中的 MB 區域為準軸，在另一 Frame 中的同區域位置附近擷取同樣大小的像素點資訊。在擷取過程裡，每做一種偏移，便能用準軸區域和偏移擷取區域的差值計算出一個絕對差異總和(Sum of absolute differences, SAD)資訊。由這些不同偏移量計算的像素差異值總和中，我們從中找出擁有最小差異值的偏移向量。該偏移向量，即為該兩 Frame 之間該 MB 區域最可能的區域運動向量。我們可以在一

張 Frame 上設定多個 MB 區域並由各個 MB 的區域得出各自的運動向量，這些小區域中代表特定區域偏移趨勢的運動向量，即為區域運動向量(Local motion vector, LMV)。下圖[圖 2-2]即為區塊比對法的示意圖，由 Frame(t)與 Frame(t-1)兩張 Frame，以 C(x)位置為基準區域在 Frame(t-1)同區域進行一個偏移後所得的 P(x+i)區域，將 C(x)與 P(x+i)的各個像素做差值量總和，即代表該偏移量的絕對差異總和值，而不同偏移量能算得各自的絕對差異總和值。

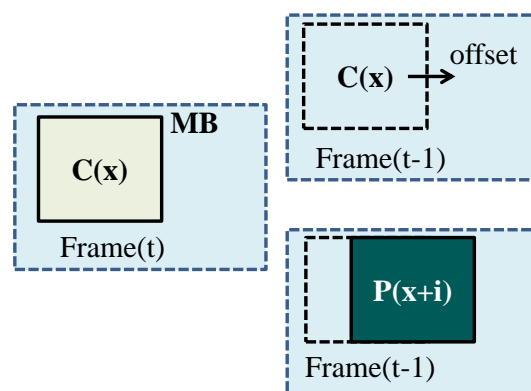


圖2-2 區塊比對法示意圖

區塊比對法公式如下所示：

$$SAD_{(i,j)} = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} |C(x,y) - P(x+i,y+j)| \dots\dots\dots (2-1)$$

$$MB_motion(i,j) = \arg \min(SAD_{(i,j)}) \dots\dots\dots (2-2)$$

其中 $SAD_{(i,j)}$ 為水平偏移 i 分量而垂直偏移 j 分量的 SAD 資訊， $C(x,y)$ 值代表目前 Frame 的像素位置資訊， $P(x+i,y+j)$ 代表前一時刻 Frame 水平偏移 i 分量而垂直偏移 j 分量的像素位置資訊， M 為方形 MB 的邊長， $MB_motion(i,j)$ 為所得 MB 區域的運動向量。公式將前後兩 Frame 的依照特定偏移，將整個 MB 大小的範圍像素資訊進行差值量總和，接著取得最小差值量總和來代表該為 MB 區域的運動向量。

Ko[17][18]用二進制葛雷碼(Gray code)將區塊比對法的差值總和用互斥或邏輯運算 (Exclusive or, XOR) \oplus 取代，運用布林函數(Boolean function)的方式大幅降低了區塊比對法的計算複雜度。首先，將 Frame 的某處影像資訊以 2^K 個灰階表示：

$$F(x,y) = H_{K-1}2^{K-1} + H_{K-2}2^{K-2} + \dots + H_12^1 + H_02^0 \dots \dots \dots (2-3)$$

其中 $F(x,y)$ 指 Frame 中位置在 (x,y) 的灰階二進制表示法， H_k 為二進制碼， $0 \leq k \leq K-1$ 。如此能將 Frame 中每一像素點以二進制表示出來，而一張 Frame 中的位置可以依照位置順序表示為一段有限長的二進制碼。接著，我們再應用剛才所得各像素的葛雷碼序列資訊，將整張 Frame 表示為 K -位元的葛雷碼。 K -位元葛雷碼的表示公式如下：

$$G_{K-1} = H_{K-1} \dots \dots \dots (2-4)$$

$$G_k = H_k \oplus H_{k+1} \dots \dots \dots (2-5)$$

其中 $0 \leq k \leq K-2$ ，因此整張 Frame 可以表示為 $G_{K-1}G_{K-2} \dots G_1G_0$ 。接著，我們藉由兩 Frame 的葛雷碼序列進行相對關係式 $E(i,j)$ 的計算：

$$E(i,j) = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} G_k^t(x,y) \oplus G_k^{t-1}(x+i,y+j) \dots \dots \dots (2-6)$$

$G_k^t(x,y)$ 為第 t 時刻 Frame、第 K 組葛雷碼序列。我們計算前後兩 Frame 之間葛雷碼序列相對關係的不匹配個數，並將不匹配個數越少的偏移量視為 LMV 資訊。雖然目前的硬體已經可以快速計算 SAD，葛雷碼方法不見得會有顯著的執行速度優勢。但是我們不能忽視葛雷碼方法在數位影像穩定器在嵌入式硬體執行上仍有降低複雜度的優勢。

Lucas-Kanade-Tomashi algorithm(LKT)[22]利用光流(Optical flow)趨勢變化來得出運動向量。這個方法僅需取用 Frame 中的固定區域做輸入資訊來得出運動向量值，但是該區域特性上要滿足三項假設：第一，亮度恆定性，Frame 與 Frame 之間的代表同目標區域的亮度資訊是相同或維持不變的。第二，時間連續性，隨時間變化的運動速度要低於一定範圍，亦即 Frame 之間目標位移要夠小。第三，空間一致性，同目標像素的鄰近像素與目標像素有相似的位移運動。本法利用這些假設將亮度與時間關係的導數配合迭代方式得出目標點的運動向量。首先，亮度恆定使得亮度 I 裡影像水平向 x 像素值和垂直向 y 像素值所對應的時間導數為零(t 為時間)：

$$I(x(t), y(t), t) = I(x(t + \Delta t), y(t + \Delta t), t + \Delta t) = 0 \dots\dots\dots (2-7)$$

接著由第二假設的時間連續特性，用鏈結公式(Chain rule)將隱含在時間的兩方向參數表示成偏微分形式，可得出二維空間光流速度的恆等式：

$$\left. \frac{\partial I}{\partial x} \right|_t \cdot \frac{\partial x}{\partial t} + \left. \frac{\partial I}{\partial y} \right|_t \cdot \frac{\partial y}{\partial t} + \left. \frac{\partial I}{\partial t} \right|_{x(t), y(t)} = I_x u + I_y v + I_t = 0 \dots\dots\dots (2-8)$$

因為這個恆等式的解空間將是一條直線，我們會應用最後一個假設——空間一致性，將相鄰像素點的亮度偏微分資訊納入考量。因此得到一個矩陣形式：

$$\begin{bmatrix} I_x(P_1) & I_y(P_1) \\ I_x(P_2) & I_y(P_2) \\ \vdots & \vdots \\ I_x(P_n) & I_y(P_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} I_t(P_1) \\ I_t(P_2) \\ \vdots \\ I_t(P_n) \end{bmatrix} \Rightarrow A\eta = B \dots\dots\dots (2-9)$$

其中 $I_x(P_1)$ 為某像素點 P_1 水平向亮度偏微分值、 $I_y(P_1)$ 為某像素點 P_1 垂直向亮度偏微分值。當 (AA^T) 可逆時，我們可用逆矩陣關係式求解：

$$\eta = (AA^T)^{-1} A^T B \dots\dots\dots (2-10)$$

(AA^T)可逆情況發生於滿足滿秩現象或是有兩個較大的特徵向量，亦即該區域有兩個方向對均勻梯度不會反應。該處的解(u,v)^T即為所求區域運動向量。

Harris-corner detection[19]將 Frame 中屬於特徵的哈式角點(Harris corner)，視為在兩方向梯度不受旋轉和移動改變的參考點，並用來在兩 Frame 之間界定運動向量。同樣也是找出影像在灰階時，依照水平與垂直兩方向進行類似 Hessian 矩陣 H(x,y) 二階導數的原理：

$$H(x,y) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \dots\dots\dots (2-11)$$

$$M(x,y) = \begin{bmatrix} \sum_{-K \leq i, j \leq K} \omega_{i,j} \cdot I_x^2 & \sum_{-K \leq i, j \leq K} \omega_{i,j} \cdot I_x I_y \\ \sum_{-K \leq i, j \leq K} \omega_{i,j} \cdot I_x I_y & \sum_{-K \leq i, j \leq K} \omega_{i,j} \cdot I_y^2 \end{bmatrix} \dots\dots\dots (2-12)$$

$$I_x = I_x(x+i, y+j) \dots\dots\dots (2-13)$$

$$I_y = I_y(x+i, y+j) \dots\dots\dots (2-14)$$

而取得 Harris corner 的方法是利用每點以固定範圍大小 K，搜尋屬於該點二階導數的自相關矩陣 M(x,y)，公式中 x、y 各為水平與垂直向位置，i、j 各為水平與垂直向偏移，ω_{i,j} 是讓函數歸一化的權重，也可用作產生特定搜尋範圍樣式的權重配置。利用該矩陣在滿秩情況或有兩最大特徵值情況會有解的原則，取得一組解。這可表示以該點為中心軸在固定搜尋範圍內找到水平垂直兩種方向的梯度交會，而交會的該點即被視為角點(Corner)，也就是適合追蹤的特徵點。

其餘計算區域向量的參考資訊，尚有使用均方根誤差(Mean square error)、平均絕對差值(Mean absolute differences)等統計方法。區域運動向量估計主要是特徵運動向量值的擷取，包含水平、垂直面的估計。特徵取得功能除了本節取得二維成分之外，在全域運動向量估計中尚能在整張 Frame 上額外探討旋轉和縮放的參數搜尋。

2.3 全域運動向量估計研究

全域運動向量(GMV)由判斷鏡頭狀態、篩選干擾因素和公式化問題求解等三個步驟組成。影像穩定器本身是針對錄影操作者所欲的方向進行影像資訊的補償，所以本步驟由上步驟所算得的區域運動向量(LMV)分析，估計出最近似攝影機自我運動的 GMV。理想情況下，我們拍攝時能記錄到完整的背景資訊，可以直接選擇 LMV 群中出現次數最多的運動向量視為全域運動向量。然而現實情況下，一般使用者攝影會需要處理鏡頭狀態的偵測和非背景干擾成分的排除，以降低區域運動計算的誤差和避免估計出與預期運動截然不同運動向量的錯誤產生，進而提升全域性運動向量估計的穩定性。所以，本步驟功能在於判別鏡頭狀態、排除相關干擾成分後再用公式求解 GMV。

首先，將上一階段取得的多個區域運動向量當作輸入值開始判定鏡頭狀態。我們可以由 LMV 的方向上來巨觀判斷各種目前的相機狀態而選用相應的處理模式，如：平移模式(Panning mode)、靜止模式(Fix mode)、廣角模式(Zoom mode)、旋轉模式(Rotation mode)。假設僅取 Frame 中四個邊角進行區域向量計算，若四個 LMV 皆有相同方向，應為平移模式；若 LMV 向量大小不大且方向不一致，可能為靜止模式；若 LMV 方向指向 Frame 中心位置，應為廣角模式；若四個 LMV 方向呈現順時針或逆時針，應為旋轉模式。全域運動向量的篩選應根據不同模式選擇不同演算法進行計算。

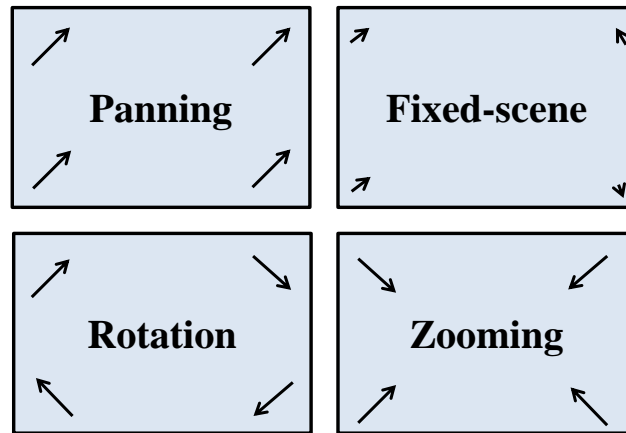


圖2-3 區域運動向量估計後的各類相機鏡頭運動狀態[28]

其次，隨著處理層級和視角廣度逐漸上升的過程中，我們需要一個篩選演算法來過濾資訊，依照各個 LMV 所在區域干擾成分的多寡，計算相應的可靠度值。干擾因素大致上分為三類：移動物體(Moving object)、低對比度區域(Low contrast area)、或重複式圖像(Repeated pattern)。區域中的移動物體會造成擷取到的 LMV 為物體行為而非背景資訊，低對比度區域的像素資訊不穩定且不易得出 LMV 資訊，重複式圖像有機會算出多種 LMV，這三類干擾因素存在的區域皆不適合納入 GMV 的估計。本篩選步驟主要根據該 LMV 所在的區域特性來決定一個可靠度值，可靠度值越高的 LMV 屬於背景的可能性越大。反之，當該區域的低對比度、移動物體、斑斕式圖案等成分越多的 LMV，可靠度值應該越低：低對比區域的雜訊成分高，所算出的穩定運動成分的可靠度低，移動物體的運動成分本身不代表相機使用者預期的成分，重複式圖像算出的運動會因為鄰近圖案相似度高而產生多種可能的運動向量。可靠度演算法的設計理念是可靠度值低的 LMV 在估計 GMV 時所佔的參考比重要相對較低，而可靠度值高的 LMV 所佔的參考比重要相對較高。

最後，計算完各個 LMV 的可靠度後，我們將公式化 GMV 的估計並求解。我們鎖定較高可靠度 LMV 所在的區域取得位置代表點(Representative point of

position)，位置代表點通常是使用該區域的中心位置，接著理想上 GMV 估計即可變成一個最佳化的過程。假設區域 i 前時刻的代表點為 $\hat{p}_i = [\hat{x}_i, \hat{y}_i]^T$ 而本時刻的代表點為 $p'_i = [x'_i, y'_i]^T$ ，則我們由前一時刻代表點位置做相應的位移變化以求出最接近本時刻的代表點位置。以讓誤差 Δ 最小的基準上，我們期望求出參數：旋轉角度 θ 、旋轉軸中心位置 (x_c, y_c) 、水平位移 a 和垂直位移 b 等資訊。

$$\hat{p}_i = \begin{bmatrix} \hat{x}_i \\ \hat{y}_i \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x_i - x_c \\ y_i - y_c \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \dots\dots\dots (2-15)$$

$$\Delta = \sum_{i=1}^m \|p'_i - \hat{p}_i\|^2 = \sum [(x'_i - \hat{x}_i)^2 + (y'_i - \hat{y}_i)^2] \dots\dots\dots (2-16)$$

GMV 值主要代表兩 Frame 之間的位移關聯性，在決定 GMV 時我們根據各個區域所算出的 LMV 賦予相應大小的權重以進行 LMV 篩選，並從篩選後的 LMV 中提取 GMV。舉例來說，全部 LMV 值中假設超過半數以上的 LMV 方向相似度非常高且位置分配平均，則該類 LMV 是 GMV 值的可能性越大，演算法計算得的可靠度值應越大，而 GMV 的決定應從這些高可靠度的 LMV 中做篩選。常見的手法有：運用權重值線性組合、低通濾波器(Low pass filter)、直方圖統計(Histogram)或重心的概念取得 GMV 值。GMV 值估計演算法客觀上是根據當下的相機狀態來決定如何篩選 LMV 值，這個演算法通常包含可靠度值公式和全域運動向量篩選法則的設計。

2.4 補償值估計演算法探討

獲得全域運動向量(GMV)後，補償值估計演算法讓我們將抖動的 GMV 訊號平滑化成穩定的訊號。本步驟的功能在於濾除全域運動向量中的抖動成分(Jitter motion)，並盡可能保留穩定的自我運動(Ego-motion)。其中，自我運動亦可稱之

為預期平移運動向量(Panning motion vector, PMV)。全域運動向量與自我運動向量之間的差異即為本步驟要求得的補償值運動向量(Compensative motion vector, CMV)。一個影像穩定器的品質著重於是否能輸出最接近使用者想要的所欲穩定影像，也就是說系統用來校正的補償值即為影像穩定器輸出成果的關鍵。所以，本步驟相當於影像穩定器的核心。

補償值演算法不論是預測型演算法或是後處理型演算法，皆以所使用的 GMV 輸入資訊來做分類。S. Erturk[9]由本步驟所使用的 GMV 物理特性對補償值演算法做分類：直接使用任兩影像初次擷取的運動向量，不能從中看出與第一張 Frame 之間的位置位移關係，屬於 Frame 與 Frame 之間相對偏移量的演算法，稱其為運動向量積分型(Motion vector integration, MVI)；另一種方式則多一個累加的暫存器，當新的資訊讀入時將這些 MVI 型 GMV 資訊進行累加以求出時間軸上的絕對位置資訊，因而能得知與第一張 Frame 的位置關係。這樣的意義在於能直接針對絕對位置進行校正演算法，稱其為幀位置平滑化型(Frame position smoothing, FPS)，優點在於能有較平滑的輸出結果。MVI 型 DIS 使用相對偏移向量進行 GMV 抖動成分平滑化，因為直接取用擷取資訊而適合即時(Real-time)運算，但是針對向量做穩定會使得輸出結果慢於自我運動；而 FPS 方式的 DIS，輸入值是針對絕對位置，所以輸出結果會較穩定，但會有誤將抖動運動向量當作所欲運動向量曲線的可能性，取樣率會需要另外考慮。

Konstantinos Ioannidis[10]將處理訊號使用的經驗模態分解法(Empirical Mode Decomposition, EMD)再配合希爾伯特·黃轉換(Hilbert-Huang Transform, HHT)處理影像穩定。該方法為後處理型數位影像穩定器，套用代表抖動的小波動乘載於代表自我運動的大波動訊號觀念，將全域運動向量資訊依據經驗模態分解法不斷地抽取出近似規律弦波訊號的本質模態函數(Intrinsic mode function, IMF)。IMF 需要滿足兩項條件：(1)區域最大值(Local maxima)與區域最

小值(Local minima)的總和個數必須和零交越線(Zero crossing)的個數相等或容許僅差一個，(2)在任何時間點，上包絡線行經的值與下包絡線行經的值的兩者平均將為零。因此，IMF 是近乎規則弦波但是可以有不規則週期的訊號。隨著原始訊號被抽取 IMF 的深度增加，抽出的規律函數頻率將越來越低。接著，將抽取出的 IMF 和殘餘訊號(Residual signal)求出希爾伯特·黃轉換的能量值。最後依據能量的巨量變化當作分界，將殘餘訊號與後抽出的多個 IMF 組合出自我運動訊號。但是，用 EMD 方法進行抖動現象平滑化會出現兩種問題：第一種，混模(Mode mixing)問題。這個現象是當混合波中的訊號頻率或振幅相差過大導致，EMD 無法分離兩者造成部分抖動現象無法濾除。第二種，邊界效應(Boundary effect)，EMD 會利用包絡線(Envelope line)來進行 IMF 搜尋。包絡線定義最模糊的地方在於首末兩處的歸屬，因為資訊不足無法判斷該處為上包絡線或是下包絡線，而只要判斷與實際有誤差，則輸出的曲線會與預期完全不同或產生巨幅波動，進而影響穩定器的輸出結果。前者可用的解決方式是改用最佳化 EMD(Oblique-extrema EMD, OEMD)的方法，將原本抽取 IMF 的參考線由零準位變更為原訊號反曲點的連線，這樣取得 IMF 的條件不至於太嚴格，原本純粹 EMD 無法抽取 IMF 的區段也能抽取 IMF。但是，降低抽取 IMF 的條件是一把雙面刃，使用 OEMD 的副作用在於若是該區段 IMF 抽取的深度不夠，會將抖動成分當作自我運動訊號，這對影像穩定器而言是個致命傷。

OEMD CMV Error when depth of IMF is not enough

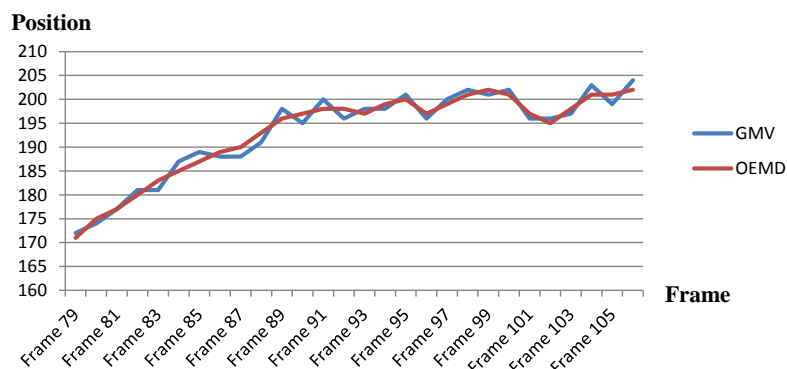


圖2-4 抖動成分被當作自我運動之現象

Sheng-Che Hsu[11]的影像穩定器隨時間紀錄並累加過去補償值(CMV)，屬於累加補償值法(Accumulative compensation method, ACM)。每次進行該時刻的補償值計算時，會參考上一時刻補償值和累加的補償值紀錄。而參照的方式是以希望補償值的數值絕對值不要過大，來進行演算法的補償。此外，該作者還增加了 Clipper function(2-19)避免 CMV 結果超過其在 LMV 擷取步驟所設定的搜尋上限，詳細公式如下所示：

$$CMV(t) = \alpha \cdot CMV(t-1) + GMV(t) - \beta \cdot CMV_I(t-1) \dots\dots\dots (2-17)$$

$$CMV_I(t) = CMV_I(t-1) + CMV(t) \dots\dots\dots (2-18)$$

$$CMV(t) = Clipper(CMV(t)) = (|CMV(t) + L| - |CMV(t) - L|) / 2 \dots\dots (2-19)$$

其中 $0 < \alpha < 1$ 且 $0 < \beta < 1$ ， t 代表時間。公式(2-17)(2-18)中 $CMV(t)$ 為第 t 時刻 Frame 的補償值。 $CMV_I(t)$ 為第 t 時刻 Frame 的累加過去補償值的暫存器， L 為搜尋範圍上限。該公式意義用「以上一個穩定後的 Frame 位置為基準，以衰減上次補償值量值做本次修正」，以及「累積的補償值應當平衡，補償值應在座標軸零附近擺盪」為準則，將累加補償值紀錄來做為回授，以達到目前時間點要使用的補償值最能讓下次累加值的絕對值更接近零。最後運用 Clipper 函數確保補償值不超過設定的搜尋上限。Yeon Geol Ryu[12]曾提及，由於以平衡補償值為準則，純粹公式可能會造成的問題主要在於參數設定的取捨和錯誤的補償值累積。首先，在於參數設定上要考量公式的 α 與 β 值。 α 設定若太小會比較不在意 $CMV(t-1)$ 並提高對於平衡至零準則的影響，造成輸出結果過度反應(Over-reaction)而讓輸出的自我運動位置曲線鋸齒現象嚴重；同理， β 設定若過接近一，會提高對於平衡至零準則的影響。因此 ACM 法的缺點在於參數設定需要動態調整且 CMV_I 值需要定期清空，避免錯誤 CMV 累積造成後續判斷錯誤。此外，在純粹等速度拍攝下做 ACM，該仍有週期性擺幅出現的可能。

Chuntao Wang[13] 運用卡爾曼濾波器 (Kalman filter)[23] 配合光流 (Lucas-Kanade Tomasi optical flow) 演算法進行影像穩定。光流法篩選出兩 Frame 之間的特徵點，依照特徵點判斷是否有場景發生變化，決定是否要繼續更新卡爾曼濾波器並補償抖動運動向量部分直至影片終止。卡爾曼濾波器本身是利用先驗誤差的大小決定校正用的卡爾曼常數，配合當下值回授計算出本時刻的後驗誤差值，再用後驗誤差推測出下一時刻的先驗誤差並重覆這個循環，卡爾曼濾波器利用當下修正後進行補償而擁有適合長期錄影的優點，但是卡爾曼濾波器通常會因為校正值設定產生問題。校正參數設的較多，回授較迅速但對劇烈變化反應較大，穩定結果有時會有抖動現象；較正值設計過小，則回授較慢，可能造成輸出影片的延遲現象。

Yeon Geol Ryu[12] 同樣用 LKT 法抓特徵點軌跡，並將各個特徵點軌跡所得的原始訊號做卡爾曼濾波器平滑化。作者認為補償值是介於原始訊號和用卡爾曼濾波器的平滑訊號軌跡之間，所以針對原始訊號和卡爾曼濾波後的平滑訊號做隨機抽樣一致性演算法(Random sample consensus, RANSAC)。 $\hat{P}_{i,K}$ 為卡爾曼濾波平滑後第 i 個特徵軌跡資訊中第 K 時刻的點特徵值， $P_{i,k}$ 則為同點特徵軌跡資訊中第 K 時刻的原始資訊點特徵值。

$$\hat{P}_{i,K} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} P_{i,k} + \begin{bmatrix} e \\ f \end{bmatrix} \dots\dots\dots (2-20)$$

一張 Frame 中 LKT 法可以得出多組特徵資訊，且每一個時刻的特徵資訊皆可讓我們能近似出一組六變數{a,b,c,d,e,f}的補償值解，這組解可與其他情況的解計算出誤差值。設定一個權重值，任取三組點特徵所計算得的誤差值若是小於所設計的權重值，則該組解可做為真正的補償值。若是沒有則繼續重複任選三組點的步驟，若迭代次數終止前都沒辦法得解，則適度調整誤差權重值。

Chin-Teng Lin[15]以離散 PI 控制器(Discrete proportional integrated controller)配合回授系統動態調整累積補償值的方法進行影像穩定。該控制器以補償值為輸出和回授，M 以全域運動向量值輸入，L 為在取樣週期下前次補償值與本次補償值的誤差值，K 區塊是累積補償值型的累加器。該系統以 PI 控制器對每單位取樣時刻的補償值依據上回補償值的差距進行校正，使得該回的補償值能更貼近理想上的平滑度。

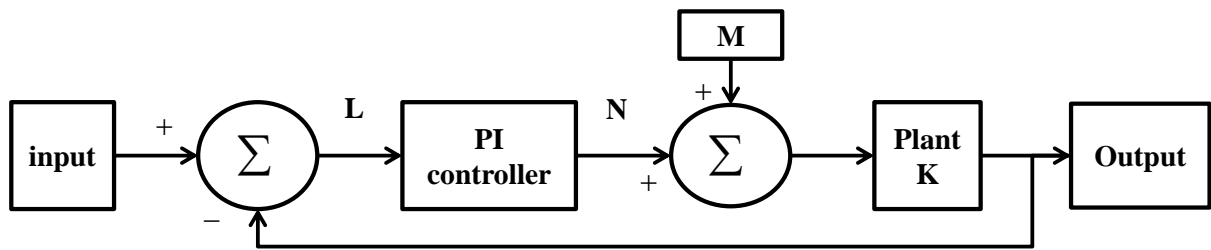


圖2-5 回授型 PI 控制器[15]

Angelos A. Amanatiadis[14]使用獨立成分分析(Independent component analysis, ICA)法。首先，將每張 Frame 上切出兩區域並算出相應的區域運動向量。其次，由於訊號上只有抖動與所欲運動向量，將獨立成分分析寫法表示區域運動矢量為抖動和所欲運動的線性組合。接著用統計學上的峰度(Kurtosis)識別直方圖有否出現非高斯分布(Non-Gaussian distribution)。接著，利用中央極限定理和拉格朗日乘數得出最不像高斯分布的矩陣以獲得抖動與所欲運動向量兩訊號。最後，以演算法分出何者為所欲運動向量並得出補償值完成影像穩定。

2.5 影像補償之簡介

最基本的影像修補在於直接依據二維補償值資訊將原影像做二維偏移並剪裁影像，使得Frame之間的影像位置連續性能以自我運動向量為依歸。當影片所需解決的移動問題面相過多，則本部分演算法尚須考量到其他因素，諸如旋轉、模糊、和影像延伸。影片因相機抖動或場景中移動物體造成的模糊，是因

為感測器受劇烈搖動讓資訊蒐集過程造成影響；影像廣角的延伸會需要面臨決定參照哪些像素點和色彩像素插捕點的問題。真實拍攝環境或行車紀錄器的拍攝場景並非全為簡單的二維問題，所以影像修補演算法的好壞影響結果影片的空間感和穩定性。

第三章 數位影像穩定器演算法系統架構

本章將介紹本文所提出的影像穩定器處理流程和特色，3.1 節介紹系統方塊圖、3.2 與 3.3 節分別介紹背景區域篩選法則以及用移動平均方式來平滑化全域向量的特色與原因。

3.1 系統流程方塊圖

我們的系統主要分成區域運動向量估計、全域運動向量估計和運動向量補償值估計。首先，區域運動向量估計是用區塊比對法以拱門型分布的有限區域進行區域向量計算；其次，全域運動向量則在於使用一個背景區域篩選演算法將區域中各個 LMV 分出高可靠度的 LMV，再配合兩種可能的 GMV 情況來估計 GMV 值；最後，我們所提出演算法主要針對運動向量補償值估計上，以估計自我運動向量(Ego motion vector, EMV)的速度來獲得所需補償值。每時刻所要估計的 EMV 是以有限個數鄰近時刻的各個自我運動速度來計算得出，而算得各個時間區間自我運動速度的方法是利用權重式移動平均的觀念。本章我們會基本介紹背景區域的篩選和權重式移動平均觀念的簡介。

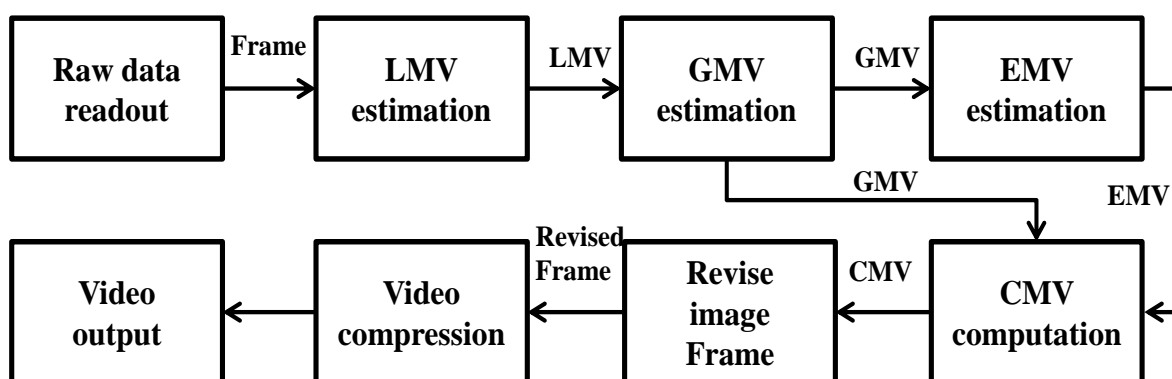


圖3-1 本文之數位影像穩定器演算法系統

3.2 背景區域之篩選

全域運動向量(GMV)是否屬於背景的準確性會影響影像穩定器的輸出結果，因此在區域運動向量和全域運動向量估計法篩選出真正背景成分的運動向量成為穩定器系統的重要課題之一。

本文濾出背景運動成分的方式可分成兩個篩選部分，區域運動向量篩選和全域運動向量篩選。區域運動向量篩選在於從一個區域所容納的九個(16X16 像素大小)微區塊中分出代表該區域的向量值，這個向量值可能不是正確的背景資訊，但是足以表現該區域主要的運動方向。全域運動向量則要考量三種背景可能存在的情況，第一種包含於 LMV 之中，我們可以由各區域中蒐集到的眾多 LMV 值，篩選出較可能存在 GMV 運動成分或是區域特性上干擾成分較少的 LMV。第二種，則是在靜止拍攝情況的零向量變化。靜態模式下拍攝，若場景存在移動物體則可能出現多種不同大小及方向的 LMV 值，此時我們可將位移不動的情況納入考量。第三種，等速度情況，我們承接影像穩定器大多用於等速度運動的假設，將上一時刻的 GMV 資訊也納入本時刻 GMV 的候選資訊。

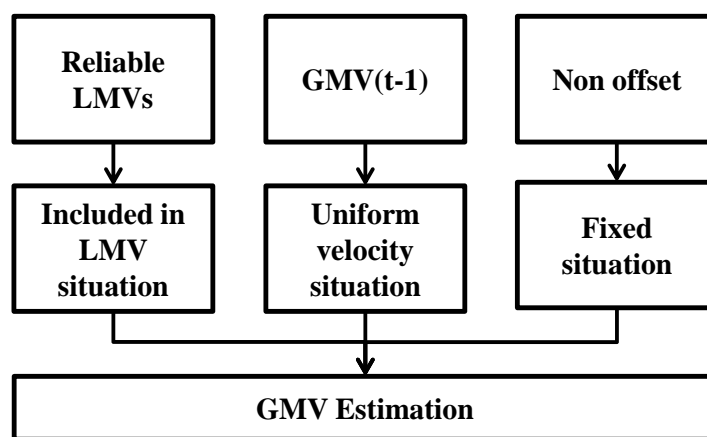


圖3-2 本文之全域運動向量估計法輸入資訊方塊圖

3.3 權重式移動平均系統簡介

移動平均系統(Moving average system)滿足當下取樣時刻 n 的輸出值是僅由該時刻 n 之前的取樣資訊做平均求得，而不會有超過該時刻的資訊被當作輸入值。而權重式移動平均系統(Weighted moving average system)除了承接僅用該時刻之前的取樣資訊的特性，他的輸出是運用輸入資訊做線性組合而求得。同理可知，權重式移動平均包含著移動平均系統的結果，其公式如下所示：

$$y[n] = \sum_{k=0}^{N-1} W_k \cdot x[n-k] / N \dots\dots\dots (3-1)$$

第 n 時刻的取樣輸出為 $y[n]$ ，輸入為 $x[n]$ ， N 為總取樣數，主要由權重 W_k 配合輸入值進行線性組合來得出。其中，因為移動平均並非累加器，當中隨著時間移動的所參考的取樣值個數 K 是有限的。移動平均的功用在於辨識數據片段的隱含變化，而套用不均等的加權至移動平均則能萃取出特定區段的資訊。

本文所提出的方法，將權重式移動平均觀念納入補償值計算的設計方向。有別於一些演算法作者為考量即時運算而使 CMV 值演算法僅參考目前與前一刻的時間點的運動向量資訊，我們使用權重式移動平均的方法將過去一定時段的資訊都納入考量可以降低極端值資訊影響。相較僅僅參考前後兩三張 Frame 之間的資訊，選擇參考一段時間的資訊更具客觀性。移動平均將一段時間的資訊，對其進行類似有限脈衝響應濾波器(Finite impulse response, FIR)的濾波。

移動平均使用在穩定器的優勢在於當有抖動情況發生，亦即有極端值出現時，能因為決定補償值的參考點數較多而有效降低對輸出訊號的影響，使得穩定器演算法能進行更客觀的判斷。該演算法的處理類似訊號處理的方式以低通濾波方式過濾連續全域運動向量資訊。

移動平均運用在平滑化抖動的取捨在於所設計的移動平均視窗(Moving average window)長度，視窗長度除了代表濾波的有效範圍，尚應考量哪種讀取隱含區段的方式最有價值。移動平均的方法讓預測型穩定器因為擁有更多的資訊而提升客觀性，但若將移動平均窗格的長度不斷延伸至全影片過程，可能會讓平均後的資訊產生延遲甚至變成後處理型穩定器；反之，長度設計過短則會降低濾波效果，在功能上無異於不使用移動平均。本文考量到影像延遲問題(詳細請參考附錄 A)，設計以五張 Frame 為做為移動平均的一個單位長度。此外，我們的演算法能針對每個移動平均區段依照穩定度來設計相應的權重值。該權重值能讓穩定度較高的移動平均區段，在自我運動估計步驟中有較高的參考比重，此部分將於第四章作詳細介紹。

第四章 系統各模組演算法探討

本章介紹本文 DIS 系統各個功能模組的詳細作法，4.1 節介紹 LMV 的篩選法，4.2 介紹 GMV 的篩選法，4.3 介紹自我運動向量的估計與補償值的取得方法。

4.1 區域運動向量估計

我們的區域運動向量估計演算法(Local motion vector estimation)類型屬於區塊比對法。區塊比對法的區塊(Block)泛指一張幀(Frame)中設計者所劃分的特定像素區塊，稱之為區域(Region)。Region 通常由多個固定大小和數目的微區塊(Macro block, MB)(16X16 單位像素大小)組成，而本步驟主要是要求出 Region 所代表的運動向量。

本演算法的做法由 MB 層級所獲得的運動向量往 Region 層級濾出區域運動向量(LMV)。首先，在 MB 層級，由每個 Region 的九個 MB 篩選出各代表該 MB 的九個運動向量，我們稱之為候選區域運動向量(Candidate LMV, CLMV)；接著在 Region 層級，由候選區域運動向量篩選出該區域所代表的區域運動向量。以下本文將逐步介紹區塊比對法的區域設定、候選區域運動向量篩選法和區域運動向量估計演算法。

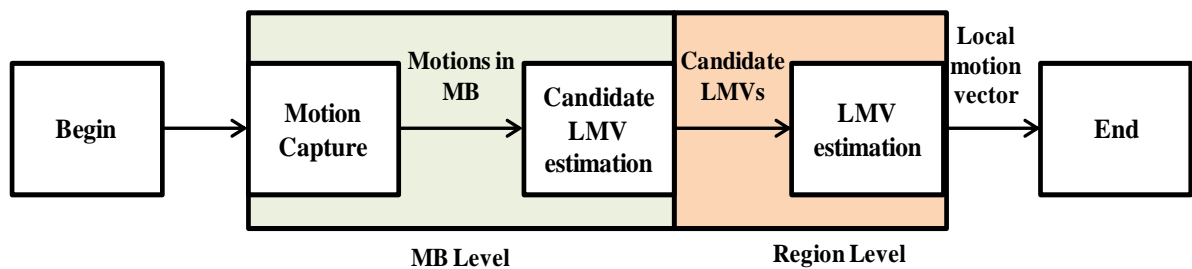


圖4-1 本文之區域運動向量估計法流程圖

4.1.1 區塊比對法的區域設定

區塊比較演算法的一個 Region 包含著固定個數像素點的資訊，Region 的分布是 Frame 資訊利用率和計算量兩者間的取捨。本文所設計的 Region 分布，在 640 X 480 像素單位的 Frame 中，取固定大小的 19 個正方形區域以拱門狀樣式排列，而每個 Region 都由 3 行 3 列排列的 9 個 MB (16 X 16 像素單位大小) 組成。由於背景資訊出現在 Frame 四周的拱門狀區域中的可能性較高，所以本文選擇使用拱門狀排列的區域型式。

本文的拱門狀區域型式上因應 640 X 480 像素單位的 Frame，共有 19 個 Region。包含上緣最多 9 個 Region，左右兩側最多 6 個 Region。本文所設定的 Region 個數是因應 Frame 大小而動態變化，並非固定 19 個 Region。拱門狀區域個數的決定方式在於「Region 與 Frame 上邊緣和左右邊緣差距 28 像素距離單位」，「Region 與 Region 之間差距 8 像素距離單位」兩指標做基準。因此，我們的 Region 設定是能因應 Frame 大小而動態變化個數。

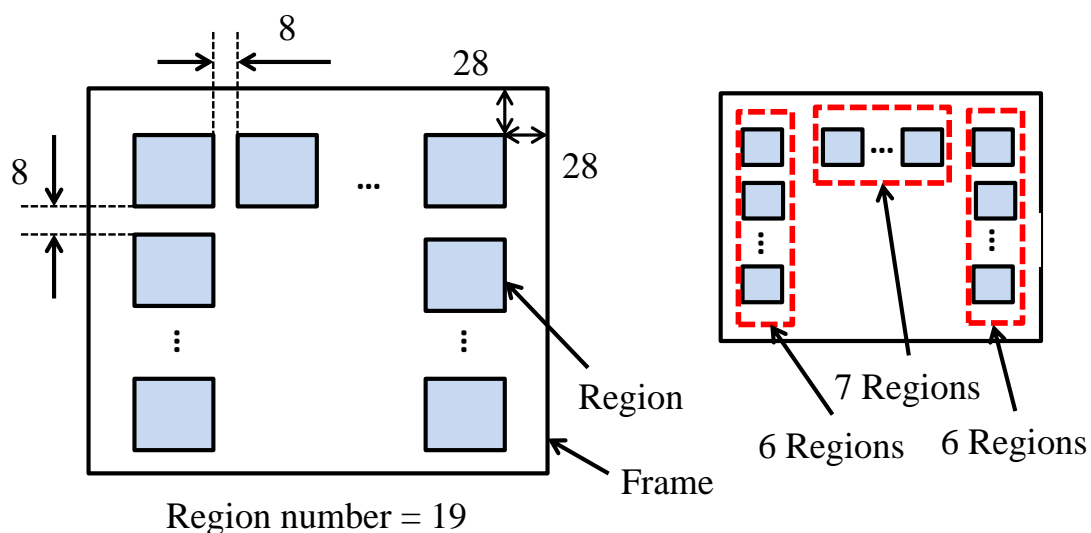


圖4-2 拱門狀區域設計示意圖

4.1.2 候選區域運動向量篩選法

每個正方形 Region 含 9 個 16 X 16 像素大小的 Macro block(MB)，以 MB 為基準，配合上下左右各八個像素的搜尋範圍建立 Sum-of-absolute differences (SAD) 資訊，所以每 MB 共兩百八十九(=17×17)個各偏移量算出的 SAD 資訊。我們由每個 MB 的兩百八十九個 SAD 資訊選擇出現擁有最小 SAD 的偏移量當作該 MB 的候選區域運動向量(Candidate local motion vector, CLMV)。本步驟演算法公式，如下所示：

$$SAD(i_k, j_k) = \sum_{x=0}^{15} \sum_{y=0}^{15} |C_k(x, y) - P_k(x+i_k, y+j_k)| \dots\dots\dots (4-1)$$

$$\forall(i, j) = ((i, j) | -8 \leq i \leq 8, -8 \leq j \leq 8) \dots\dots\dots (4-2)$$

$$g = \arg \min_k (SAD(i_k, j_k)) \dots\dots\dots (4-3)$$

$$CLMV_R(i_m, j_m) = (i_g, j_g) \dots\dots\dots (4-4)$$

k 為同 Region 中 MB 的編號，由於有九個 MB，所以 k 值範圍為 0 至 8。 x 、 y 分別為水平與垂直位置。 $SAD(i_k, j_k)$ 為一個 Region 中編號 k 的 MB 其在偏移了 (i_k, j_k) 後與上一時刻 Frame 的亮度差值總和，代表橫向偏移的 i 與縱向偏移的 j 範圍限制在 -8 和 8 之間。 $C_k(x, y)$ 為第 k 個 MB 在本時刻 Frame 在位置 (x, y) 的亮度資訊， $P_k(x, y)$ 為第 k 個 MB 在上一時刻 Frame 在位置 (x, y) 的亮度資訊。 $CLMV_R(i_m, j_m)$ 為第 R 個 Region 中偏移量為 (i_m, j_m) 的候選 LMV。公式(4-1)中將同 Region 中的 MB 的兩百八十九種偏移 motion 的各個 SAD 值求出。接著，公式(4-3)，再由 MB 階層篩選的兩百八十九個 SAD 值，選出擁有最小 SAD 的運動向量編號。最後，將該編號的運動向量當作該區域的候選 LMV(Candidate LMV, CLMV)。

4.1.3 區域運動向量估計演算法

同個 Region 內九個 MB 皆各自篩出其候選區域運動向量(CLMV)後，我們將由這九個 CLMV 利用運動向量方向種類和上階段算出的 SAD 值紀錄，決定出代表該 Region 的區域運動向量(LMV)。其中，每 Region 的 LMV 值將由其容納的九個 MB 各自持有的 CLMV 來決定。由於九個 CLMV 的種類可能不止一種，我們選擇篩選 LMV 的判斷式為：若有存在某種類 CLMV 個數最多時，視為該種類的向量為 LMV；反之，若上述狀況無法成立時，也就是沒有特別一種運動向量在數量上佔優勢，則選擇 SAD 值最小的 CLMV 為 LMV。

演算法公式：

$$h = \arg \max_m (N(i_m, j_m)) \dots\dots\dots (4-5)$$

$$f = \arg \min_m (SAD_{CLMV_R(i_m, j_m)}) \dots\dots\dots (4-6)$$

$$LMV(i_R, j_R) = (h \notin \phi) ? (i_h, j_h) : (i_f, j_f) \dots\dots\dots (4-7)$$

$N(i_m, j_m)$ 為 CLMV 中運動向量種類為 (i_m, j_m) 的出現個數。公式(4-5)求出在 CLMV 中出現率最高的運動向量編號，公式(4-6)求出 CLMV 中 SAD 最小的運動向量編號。公式(4-7)表示當無法從 CLMV 中分出出現率最高的編號時，選擇 SAD 最小的運動向量當作 LMV；若能分出 CLMV 中出現率最高的編號時，將該編號的運動向量種類當作本 Region 的 LMV。

4.2 全域運動向量估計

經過上回算出各個 Region 所代表的 LMV 後，我們將設計由這些 LMV 來判斷 GMV 值的演算法。本階段的演算法有兩部分，包含一個可靠度演算法和 GMV 篩選演算法。前者，在於篩出那些 LMV 較可靠，我們用一個可靠度值來

量化可靠性大小；後者，將較可靠的 LMV 和較可能含有背景資訊的向量估算出 GMV 值。

詳細步驟如下圖[圖 4-3]所示，首先，將各個 Region 依照可靠度演算法計算相應的可靠度值以篩選出哪些 Region 可靠度較高。高可靠 Region 的 LMV 會被視為可靠的運動向量(Reliable local motion vector, RLMV)，並作為下一階段決定 GMV 判斷式的輸入資訊之一。RLMV 與其平均運動向量、上一時刻的 GMV 值和零偏移向量為所有較可能代表本時刻 GMV 的運動向量，我們稱之為候選全域運動向量(Candidate global motion vector,CGMV)。最後利用 CGMV 的各個運動向量，依演算法篩選出本時刻 GMV。以下我們將逐步介紹 LMV 的可靠度函式(Reliability function)、候選全域運動向量估計(Candidate GMV estimation)和全域運動向量估計演算法(GMV estimation)演算法。

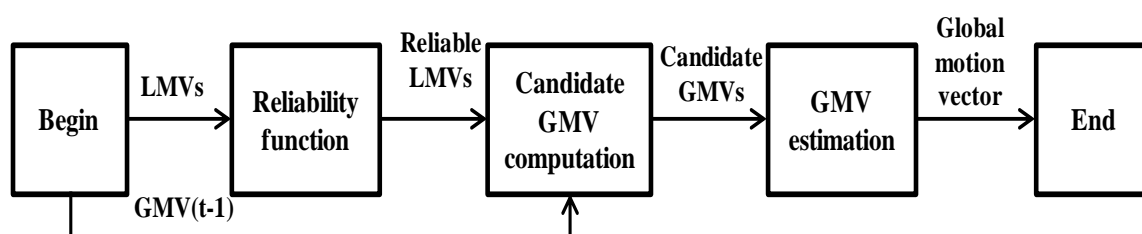


圖4-3 本文之全域運動向量估計法流程圖

4.2.1 區域運動向量的可靠度函式

本文使用可靠度函式(Reliability function)來進行各個 Region 資訊的可靠度篩選。此可靠度函式的計算步驟，使得各個 Region 會攜帶的資訊包含：LMV 的水平偏移 x 值和垂直偏移 y 值以及可靠度 s 值。針對每個 Region 區域計算可靠度 s 值，該值越大則可信度越高；越小可信度越低。我們選擇觀察 LMV 所在的區域，將各種偏移變化計算的 SAD 值作分析，並將 SAD 數值分布較平均的 Region

視作低可靠度 Region。低可靠度 Region 獲得的 LMV 資訊，將不被用於預估 GMV 的輸入資訊。

演算法公式：

$$\bar{R}_c = [x_c \quad y_c \quad s_c]^T \dots\dots\dots (4-8)$$

$$s_c = \frac{1}{N-1} \sum_{\forall(i,j)} (SAD(i,j) - \hat{a})^2 \dots\dots\dots (4-9)$$

$$\hat{a} = \sum_{\forall(i,j)} SAD(i,j) / N \dots\dots\dots (4-10)$$

$$\forall(i,j) = ((i,j) | -8 \leq i \leq 8, -8 \leq j \leq 8) \dots\dots\dots (4-11)$$

N 為搜尋區段(Search section)的位移變化總數，本處取橫向 i 左右偏移八個像素單位大小的十七種變化配合縱向 j 上下偏移八個像素單位的十七種變化，所以一共兩百八十九種可能性。 $SAD(i,j)$ 為第 c 個 Region 在本時刻 Frame 位置中心偏移 (i,j) 後與上時刻 Frame 算得的 SAD 值。

這部分公式意義在於取 Region 各種偏移產生的 SAD 值，由多種 SAD 資訊算出 Region 區域中 SAD 值的標準差，藉著標準差超過某一門檻值(Threshold)的情況可判定為可靠度高的區域。在實際運用上應在整張 Frame 進行全域性判斷時才決定門檻值(Threshold)，並濾除可靠度較低的 Region 不列入計算。當算完所有 Region 的可信度 s 值時，用所有 Region 的 s 值取得一個平均 s 值，並選擇擁有高於平均 s 值的 Region 所算出的 LMV 才被視為可靠的區域運動向量(Reliable local motion vector, RLMV)並有資格納入候選全域運動向量(CGMV)。

本可靠度演算法的意義在於將各個 LMV 所代表的 Region 進行一次區域性的 SAD(Sum-of-Absolute-Differences)分布的分析。仰賴求得各區域 SAD 的標準差，

我們得以排除過低對比度和重複式圖案的 Region，並將可靠的 LMV 納入 CGMV 中。以下，我們將介紹 CGMV 中的其他候選運動向量和如何從 CGMV 中決定出 GMV 值。

4.2.2 候選全域運動向量估計

演算法判斷 GMV 所用的輸入資訊是候選全域運動向量(CGMV)，也是本階段演算法的結果。CGMV 的組成有四類，包含第一類有多個向量資訊，這些向量是由上階段所篩選出的可靠區域運動向量(RLMV)，第二類是一個向量，由 RLMV 取得其平均的向量資訊，該資訊可稱為可靠運動平均向量(Average reliable motion vector, ARMV)，第三類則是一個向量，為上一時刻的 GMV 資訊。最後一類，則是偏移為零的情況，可稱為零向量(Zero motion vector, ZMV)。演算法公式如下：

$$ARMV(i, j) = \sum_{R=0}^{P-1} (RLMV(i_R, j_R)) / P \dots\dots\dots (4-12)$$

$$CGMV = \{RLMV, ARMV, GMV(t-1), ZMV\} \dots\dots\dots (4-13)$$

P 為 RLMV 的總個數，候選全域運動向量(CGMV)為包含著 ARMV、RLMV 和 GMV(t-1)等三類運動向量資訊的集合。我們接著會把候選全域運動向量當作下一步驟演算法的輸入值，獲得出該時刻 Frame 的全域運動向量(GMV)。本步驟的功用在於層層篩選出較可能為背景的成分，我們區分的四類向量涵蓋了較可能決定 GMV 值的三種來源。第一種，RLMV 和 ARMV 兩類皆屬於區域資訊中存在 GMV 的情況，用高可靠度的區域資訊來找到背景；第二種，GMV(t-1)則代表等速度情況，影像穩定的假設情況是在等速運動的追蹤拍攝，所以等速度情況可能性較高；最後一種，ZMV 代表目前為不動的情況，靜止拍攝的情況景物大部分為移動物體。不動情況的場景中運動向量大小和方向不定，這在一般估計方式沒有確切方式可得知，所以不動情況也值得考慮。

4.2.3 全域運動向量估計演算法

獲得本時刻所有候選全域運動向量(CGMV)後，假設上述共有 K 種 CGMV，我們計算每個 Region 中心位置在這些 K 種 CGMV 偏移下所得 SAD 之總和，這會產生 K 個 SAD 總和值。最後，從中取最小總和值所代表的 CGMV 資訊當作本 Frame 時刻的 GMV 值。演算法公式如下所示：

$$SAD_{R_c,k,t} = \sum_{(x,y) \in R_c} |I(x,y,t) - I(x+x_k, y+y_k, t-1)| \dots\dots\dots (4-14)$$

$$0 < k \leq 21$$

(最多 22 種情況：19 RLMV + ARMV+GMV(t-1)+ZMV)

$$0 \leq c < 19 \text{ (Region 0 ~ 18)}$$

$$U_k = \sum_{c=0}^{18} SAD_{R_c,k,t} \dots\dots\dots (4-15)$$

$$D = \arg \min_k (U_k) \dots\dots\dots (4-16)$$

$$GMV = CGMV_D \dots\dots\dots (4-17)$$

x、y 分別為水平與垂直位置。公式(4-14)為把第 k 個 CGMV 的運動向量資訊當作偏移量，並依該偏移情況在 R_c Region 區域中進行 SAD 資訊的計算結果。k 為 CGMV 的編號範圍，其介於 0 至 21 之間，c 為 Region 編號值，其介於 0 至 18 之間。 $I(x, y, t)$ 為 t 時刻在 R_c 在區域的位置(x,y)的亮度值。 (x_k, y_k) 為 CGMV 在 x 與 y 方向的值。 U_k 為依照第 k 個 CGMV 的運動像量值在 19 個 Region 中計算得的 SAD 總和值。不同 CGMV 會在不同 Region 算得 $SAD_{R_c,k,t}$ ，每個 CGMV 在各 Region 算得的 SAD 值總和後，將擁有最小總和值的 CGMV 的運動像量資訊當作 GMV 值。

4.3 運動向量補償值估計

本文提出一個 FPS 型補償值估計演算法，分開處理輸入的二維 GMV 資訊(水平向 x 值和垂直向 y 值)進行影像穩定。本演算法是預測型 DIS，是使用目前時間以前鄰近時間區段內的 GMV 資訊估算 Ego-motion 的速度，以延遲一張 Frame 的方式進行影像補償值計算。我們的補償值估計演算法介紹可分成三大部分：第一，重疊區段(Overlapped section)的結構，我們設定時間軸上所記錄的連續五個 GMV 當作一個區段(Section)，在 t 時刻計算補償值時會參考一定個數且時間所占 GMV 資訊彼此部分重疊的 Section，我們稱為重疊區段結構。第二，代表點與區段內的資訊取得，我們會以 Section 為單位，由各個 Section 所含的 GMV 資訊取得數個代表點，並利用代表點計算出該 Section 所隱含的資訊。第三，自我運動估計演算法(Ego-motion vector estimation)，我們運用 Section 算出的資訊，搭配適當的權重設計，線性估測出該時刻 EMV。EMV 最後將和原輸入決定出補償值運動向量(CMV)並完成影像穩定。另外，演算法本身因為使用移動平均窗的觀念而有延遲現象出現，我們採取以延遲一張 Frame 的方式降低延遲影響。

本預期運動向量估計演算法的參考輸入值，是由重疊區段內每個區段當下時刻的 GMV 資訊決定。以下我們逐步介紹重疊區段的設計結構、代表點與區段內的資訊取得、自我運動估計演算法。

4.3.1 全域運動向量之重疊區段結構

首先，將介紹本文運用 GMV 資訊的方式。本文運用一個重疊區段的結構，該結構因延遲特性考量，且為了限制演算法僅需延遲一張 Frame，在 30 fps 切割比下，我們設定區段(Section)暫存自目前時間點以前五張 Frame 之內的 GMV 資訊，如圖[圖 4-4]所示，第 t 時刻的區段儲存著 GMV(t)以前五個時間點(GMV($t-4$)至 GMV(t))的資訊。

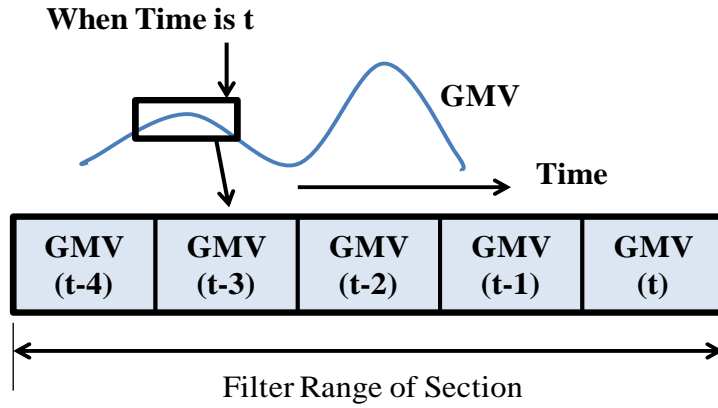


圖4-4 本文之移動平均式區段結構

區段結構公式：

$$\vec{G}_t = [GMV_{t-4}, GMV_{t-3}, GMV_{t-2}, GMV_{t-1}, GMV_t]^T \dots \quad (4-18)$$

重疊區段結構設計

固定大小的區段暫存器存放全域運動向量(GMV)資訊，每當新 Frame 資訊隨感測器讀入都會更新區段暫存器內的 GMV 資訊，並為該時間點計算該時刻擁有的特徵向量(Feature vector)資訊並將其存入重疊區段暫存器(Overlapped section register, OSR)。OSR 儲存著固定同於區段數目的特徵向量，該特徵向量擁有的資訊為(1)區段包含的 GMV 資訊(GMV Data)、(2)區段預期速度(Section panning velocity, SPV)、(3)區段代表性偏移(Section representative offset, SRO)、(4)該時刻的高斯函數(Gaussian)值和(5)區段不穩定性(Section instability)。

OSR 結構演算法公式：

$$\vec{R}_t = [\vec{X}_{t-V+1}, \vec{X}_{t-V+2}, \dots, \vec{X}_t]^T \dots \quad (4-19)$$

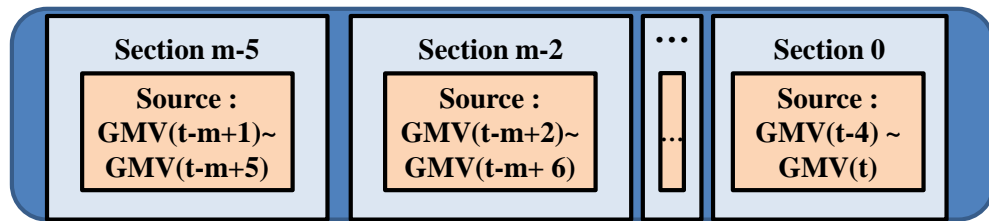
$$\vec{X}_t = [\vec{G}_t, S_t, P_t, L_t, I_t]^T \dots \quad (4-20)$$

$$\bar{G}_t = [GMV_{t-4}, GMV_{t-3}, GMV_{t-2}, GMV_{t-1}, GMV_t]^T \dots\dots\dots (4-21)$$

\bar{R}_t 為 t 時刻的 OSR 內資訊，V 為 OSR 最多納容納的 Section 個數，在我們的演算法設定 V = 10， \bar{X}_t 為 t 時刻的各個 Section 所擁有的全部資訊， \bar{G}_t 為 t 時刻 Section 所包含的 GMV 資訊， S_t 為該 Section 所代表的瞬時速度值， P_t 為為該 Section 的代表性偏移， L_t 為 t 時刻該 Section 的高斯函數值， I_t 為 Section 的不穩定值。根據移動平均移動窗的特性，OSR 隨著時間移動更新各個 Section 內的資訊。

每 Section 的特徵向量固定儲存著五種資訊，而每次演算法決定補償值運動向量(CMV)時，理應參考鄰近 Section 的資訊以實現 DIS。其中，決定每個時間點 CMV 要參考 Feature vector 的總數和影響比重。此兩者分別由 Section 中高斯函數值和不穩定性資訊來決定。此外，OSR 的範圍大小將決定計算當下補償值時最多要參考多少 Section。

在決定當下的補償值時，Section 算得的 SPV 和 SRO 提供估計出自我運動的資訊，而高斯函數值和不穩定性數值影響 Section 在估計自我運動時的參考比重。高斯函數值，使得演算法決定 CMV 時只仰賴當下時間點之前一些時段的 Section 資訊並遺忘更早之前的舊資訊，而不穩定性數值使得在決定 CMV 的判斷式中，越不穩定的 Section 其資訊會越不受重視，越穩定的 Section 則被提升參考比重。本文所提出的方法利用這兩項影響參數，模擬出人類對位置—時間座標圖中，由抖動的 GMV 位置曲線要判斷自我運動時可能參照的判斷依據。



- (1) GMV Data
- (2) Section panning velocity
- (3) Section representative offset
- (4) Gaussian value
- (5) Instability

圖4-5 本文之重疊區段結構設計

設計 OSR 內 Section 的長度主要考量的面向包含所濾抖動的週期範圍、前期放棄做 DIS 的 Frame 數和延遲現象(附錄 A)。由於特徵向量的 SPV 和 SRO 資訊是依據 Section 固定 Frame 大小所容納的 GMV 資訊進行計算而來，且抖動成分的濾除需要講究該補償值估計演算法將何種週期長度之下的起伏視為抖動，則 DIS 的行為近似於對充斥抖動成分的訊號進行低通濾波動作，所以 Section 的範圍意義同等於微幅濾波範圍(Filter Range, FR)。另外，Section 的長度設計需考量前期放棄 DIS 的 Frame 數。當該時間點未能填滿一個 Section 大小的 GMV 資訊，則演算法無法進行 CMV 計算，讓前期 Frame 無法 DIS。因此，當 Section 範圍設定過小，Section 微幅濾波功能被限制在一個很小的範圍；Section 設定過大，使得 DIS 前期需要放棄不做穩定的 Frame 個數上升且 OSR 的長度因為要包含住足夠個數的 Section 被迫亦須提升更長，使得記憶量的耗費提升。

本文演算法中估計自我運動的輸入參數是運用重疊區段結構內 Section 中 GMV 位置的趨勢資訊，這樣的優勢在於除了將不穩定的 GMV 值域轉換至另一較穩定維度，亦降低了極端值 GMV 的造成的影響。

4.3.2 補償值估計演算法結構

我們所提出的補償值估計流程分成四個步驟，第一步，DIS 系統會先隨時間讀取新的 GMV 資訊來放入 Section 暫存器，並判斷目前 Section 內 GMV 資訊量有否大於等於 Section Range(SR)的五張 Frame。若否，表示暫存器內資訊不滿足創造新 Section 的條件，則等待下次 GMV 資訊繼續填入 OSR 暫存器；若有大於等於 5，表示 GMV 資訊足以填滿一個 Section，則計算決定該時刻 CMV 要參考的 Section 個數。第二步，由演算法獲得 Section 範圍內 GMV 位置所隱含的資訊，包含區段預期速度(Section panning velocity, SPV)、區段代表性偏移(Section representative offset, SRO)和該 Section 的不穩定性(Instability)。第三步，為了衰減降低過去 Section 的重要性而更新與之相依的權重值(Weight)。最後，由該時刻的 OSR 所有 Section 資訊計算決定出前一時刻應輸出的 CMV 值。

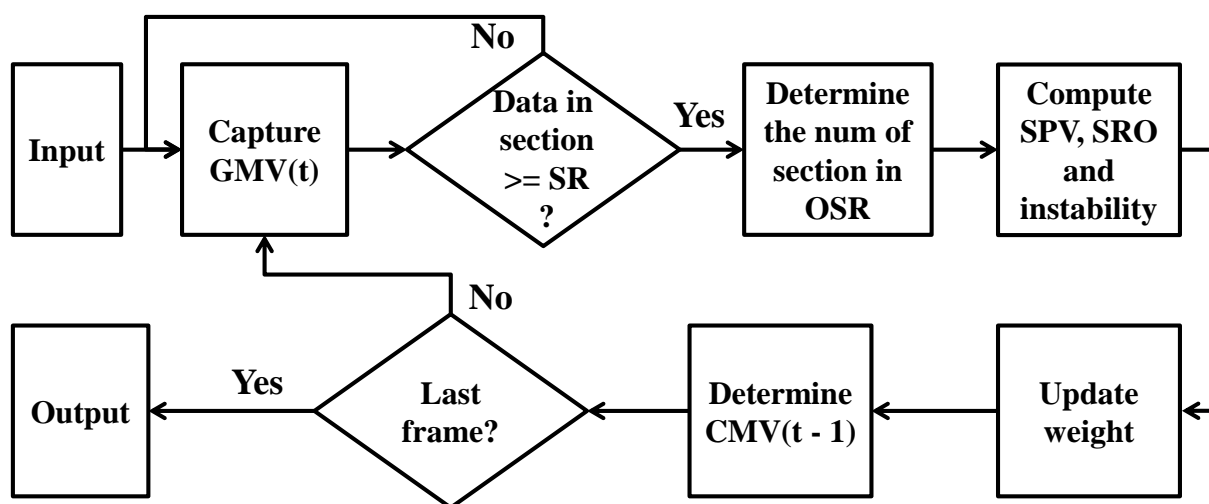


圖4-6 本文之補償值運動向量估計演算法

4.3.3 區段個數估計

由於我們最終判斷 CMV 的方式是用該時刻 OSR 內所有 Section 特徵資訊，本階段的工作是要計算每個時刻所要考量的 Section 個數。決定每個時刻所要考量的最多 Section 個數的演算法公式如下所示：

$$j = (t - W + 2) \cdot (\mu[t - S_1] - \mu[t - S_2]) + V \cdot \mu[t - S_2] \dots\dots\dots (4-22)$$

$$S_1 = W - 1 \dots\dots\dots (4-23)$$

$$S_2 = V + W - 2 \dots\dots\dots (4-24)$$

$$\mu[n] = \begin{cases} 1, n \geq 0 \\ 0, n < 0 \end{cases}$$

j 為 t 時刻的 Section 個數， $t = 0$ 時刻尚未有 GMV 值讀入，所以不用考量區段個數。 W 為區段範圍(Section range)，本演算法此設定為五張 Frame。 V 為 OSR 最多納容納的 Section 個數， $\mu[i]$ 為單位步階序列。

演算法方塊圖

我們決定 DIS 開始各時期 Overlapped section register (OSR)內存在 Section 個數的演算流程如下[圖 4-7]：首先，當 DIS 開始時初始化 Section 個數為零，接著依時間讀入 GMV 資訊。

其次，第一個判斷式會判斷當下有否填滿 Section 暫存器，若無，會繼續等待下次讀入 GMV，若滿足則進入第二個判斷式。第二個判斷讀入的 Frame 的編號是否大於等於五(Section range 為五張 Frame)，若有，則表示需要 FIFO 更新 Section register，若無則不進行針對 Section register 的 FIFO 更新。第三個判斷式，確認上回的特徵向量個數是否不等於 OSR 範圍，若是，則表示上回的 OSR 內的 Section 數已達上限，故本回需進行對 OSR 內的 Section 資訊進行 FIFO 更新，並直接套用上個迴圈的 Section 個數作為本次個數結果；若不是，則本回的 Section 個數比上個迴圈的 Section 多增加 1 個。

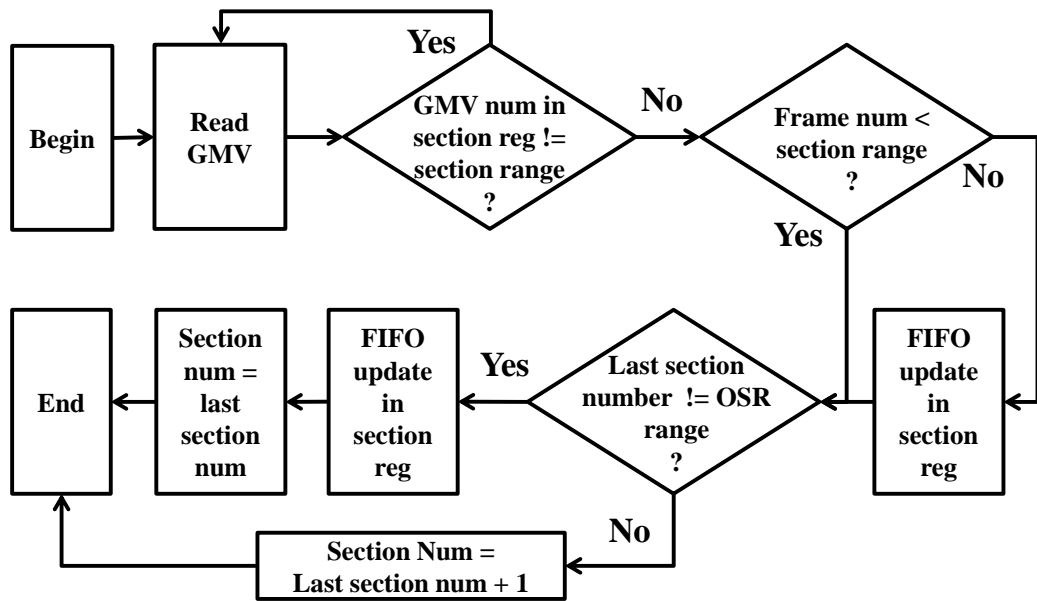


圖4-7 各時刻區段個數計算之演算法流程

4.3.4 代表點與區段內資訊的取得

當 Section Register 儲存的 GMV 數恰好滿足五張 Frame 而進入本步驟，我們在位置和時間的二維平面中進行代表區段預期速度(Section panning velocity, SPV)和區段代表性偏移(Section representative offset, SRO)的計算。

上述二者將代表 Section 資訊的速度和位置趨勢。計算這兩項資訊之前，會需要先進行取代表點的步驟，接著利用包絡線(Envelope line)對 GMV 資訊進行屬於上點(Upper point)或下點(Lower point)的分類，再依不同但相鄰的兩點取平均值的方法來取得代表此時刻 Section 內 GMV 趨勢的點集合，我們稱之為「Section 之 GMV 代表點集合」(Representative point set of GMVs in section)，以下統一簡稱代表點(Representative point, RP)。先進行取代表點步驟的原因，是因為本研究演算法設定代表點的意涵在於能組合出該 Section 暫存器中 GMV 位置資訊的趨勢和平均位置，也就是 SPV 和 SRO 的組成元素。代表點的數量可能是零至四(= Section Range - 1)不等，視 Section 的 GMV 資訊而定，但我們的演算法(More-than-one-RP algorithm)會將代表點的數量控制在大於或等於二的情況以利

斜率趨勢的取得。取完代表點後，本研究將用一個演算法算出該時刻下該 Section 代表預期速度的 SPV 和代表位置的 SRO 資訊。以下筆者將逐步介紹代表點的取得演算法、區段預期速度和預期位置演算法。

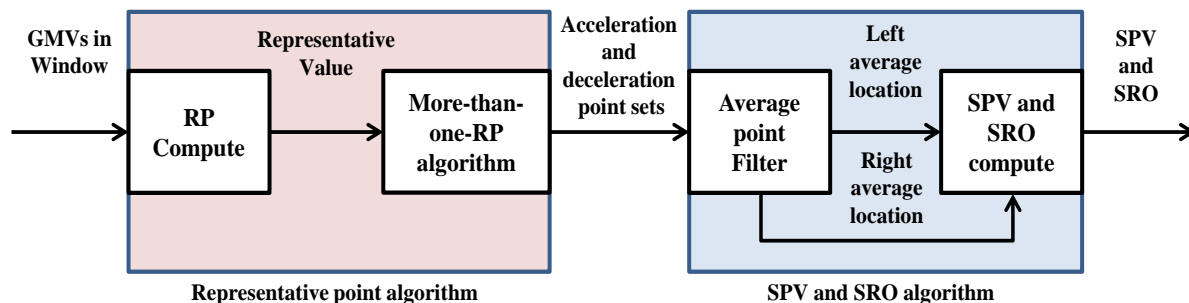


圖4-8 區段預期速度和代表性偏移演算法方塊圖

步驟 1：代表點演算法設計

首先如圖[圖 4-9]所示，本文在位置和時間的二維平面中將 Section register 中五張 Frame 的 GMV 位置資訊連線(Connection line of GMVs)，再對該連線取上下包絡線(Upper and lower envelope)並將 GMV 資訊分兩種類，分別為：屬於上包絡線的上點(Upper point)，或稱減速點(Deceleration point)；反之，屬於下包絡線的下點(Lower point)，或稱加速點(Acceleration point)；接著，將所有上點與其相鄰的下點取兩者的平均位置，其在位置—時間二維座標系中即為本文所認定的代表點(representative point)。以下為了介紹方便，將簡稱「Section 中的五個 GMV 位置資訊取樣點」為「取樣點」。

首先，為避免上下包絡線邊界取捨問題和計算複雜度，本文的設計有三項要點：第一，Section 暫存器首項和末項在上下包絡線歸屬是參考各自鄰近的上點或是下點來決定其性質。意即在位置資訊上，若距離首項或末項取樣點最鄰近擁有非零代表值的取樣點為上點，則該取樣點(首項或末項)歸屬於下點(Lower point)；反之，若首或末項取樣點最鄰近擁有非零代表值的取樣點為下點，則該點歸屬於上點(Upper point)。第二，因為演算法僅用不同類別(Upper point 和

Lower point)的兩點取平均來獲得代表點，所以我們的上下包絡線不需要也不用以傳統三次曲線近似法(Cubic spline)，避免了包絡線的邊界問題。第三，假如首項和末項取樣點之間都沒有受到分類的上點或下點存在，意即代表點算出來只有一點以下，此時的情況為Section內GMV資訊連線相當接近水平或斜直線，我們的對應方法是直接將首項和末項取樣點兩點當作代表點。

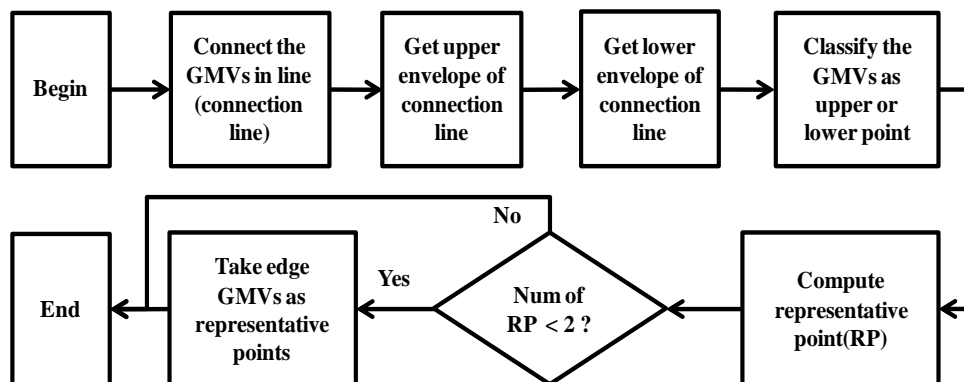


圖4-9 代表點取得之演算法流程

代表點取得流程分成兩步驟：步驟一，先取得各取樣點的代表值(Representative value, RV)，步驟二，依照RV結果判斷這些取樣點是上點(Upper point)或下點(Lower Point)，由相鄰上下兩點的中間點位置當作一個代表點，並控制代表點數量在大於一。

步驟 1-1：代表值計算(Representative value compute, RV compute)

$$\vec{T}_t = [T_{t-4} \quad T_{t-3} \quad T_{t-2} \quad T_{t-1} \quad T_t]^T \dots \quad (4-25)$$

$$T_k = \{x \mid x \in \{1, 0, -1\}\}, \quad t-4 \leq k \leq t$$

$$S_{prev,t} = GMV_{i,t} - GMV_{i-1,t} \dots \quad (4-26)$$

$$S_{curr,t} = GMV_{i+1,t} - GMV_{i,t} \dots \quad (4-27)$$

$$H_{i,t} = S_{curr,t} \cdot S_{prev,t} \dots \quad (4-28)$$

\bar{T}_i 為一個 Section 內對應五個時刻(t-4 至 t)各個取樣點的代表值集合，其中代表上點的代表值將以數值"1"表示，代表下點的代表值將以數值"-1"表示，皆非二者的點將以數值"0"表示。 $S_{prev,t}$ 為 t 時刻 Section 中第 i 時刻與第 i-1 時刻 GMV 資訊的斜率。 $S_{curr,t}$ 為 t 時刻 Section 中第 i+1 時刻與第 i 時刻 GMV 資訊的斜率。我們設定的上點代表區域範圍內的位置最高點，而下點代表區域範圍內的位置最低點。區分一個取樣點是否為區域的極端值最簡易的方式是參考鄰近點的資訊，亦即我們評斷上下點仰賴參考前後時刻的資訊。因為一個區段中滿足前後兩時刻都有資訊的取樣點只有區段中非邊界的部分，所以我們將代表值的取得方法分成邊界取樣點和非邊界取樣點兩種。分開處理邊界與非邊界的同時，也解決一般純粹使用包絡線方式界定上下時會面臨的邊界條件問題。

非邊界取樣點的代表值計算(t-3 ~ t-1)

$$A_{i,t} = \mu[S_{prev,t}] \dots\dots\dots (4-29)$$

$$B_{i,t} = \mu[S_{curr,t}] \dots\dots\dots (4-30)$$

$$\mu[n] = \begin{cases} 1, n \geq 0 \\ 0, n < 0 \end{cases}$$

$$T_i = [(S_{prev,t} == 0) \&\& (S_{curr,t} != 0)] ? (-2 \cdot B_{i,t} + 1) : \\ [(S_{prev,t} != 0) \&\& (S_{curr,t} == 0)] ? (2 \cdot A_{i,t} - 1) : \dots\dots\dots (4-31) \\ (H_{i,t} < 0) ? (2A_{i,t} - 1) : 0 \\ t - 3 \leq i \leq t - 1$$

公式(4-26)(4-27)(4-28)，是在處理 Section 的第(t-3)時刻取樣點至第(t-1)時刻取樣點。此三個取樣點的 GMV 資訊都擁有前後時刻 GMV 資訊，所以要與 Section 頭尾的(t)和(t-4)兩時刻取樣點分別處理。此外，邊界取樣點的代表值是參照非邊界的代表值，所以非邊界的代表值需優先計算。

首先，將 Section 中非邊界取樣點的每點 i 取「與前取樣點時刻的斜率」和「與後取樣點時刻的斜率」(4-26)(4-27)。接著，我們的演算法針對兩斜率中三種情況：某一斜率是零而另一斜率非零、皆非零斜率且相異號、皆非零且為同號的情況來決定該時刻的代表值為多少。所以要先取得兩斜率的乘積(4-28)和各自是否為正值或負值零等資訊。最後，根據前後各個情況，決定其代表值 T 為多少。

舉圖[圖 4-10]Case 1 為例，當前時刻斜率為零，而後時刻斜率非零。表示前兩點間的連線是平直線，所以應由後取樣點時刻斜率 B 來決定代表值 T ，因此運用代表值 $T = -2B + 1$ 的公式決定代表值。當後取樣點斜率為正值，表示本取樣點應為下點所以 T 值為-1；當後取樣點斜率為負值，則表示本取樣點應為上點所以 T 值應為+1。

邊界取樣點的代表值計算($t, t - 4$)

邊界取樣點計算必須在非邊界取樣點之後，公式(4-32)處理首末取樣點代表值的計算。我們將離邊界取樣點最近的非零代表值為基準，讓首末兩點的代表值與最鄰近的非零代表值相異號。這樣讓首末兩點與最接近的上點或下點之間各會取一個代表點。本處代表值是參照非邊界取樣點的代表值，所以邊界取樣點的計算必須是在非邊界取樣點代表值計算之後。

$$T_i = (T_{nearest, non-zero} > 0) ? -1 : 1 \dots\dots\dots (4-32)$$

$$i = t, t - 4$$

本步驟計算區段的取樣點相應的代表值，其中首末取樣點和中間三個取樣點分別使用不同演算法避開了包絡線的邊界效應問題，使得演算法在任何情況的計算上都有穩定解存在。接著，我們將運用代表值進行代表點的計算。

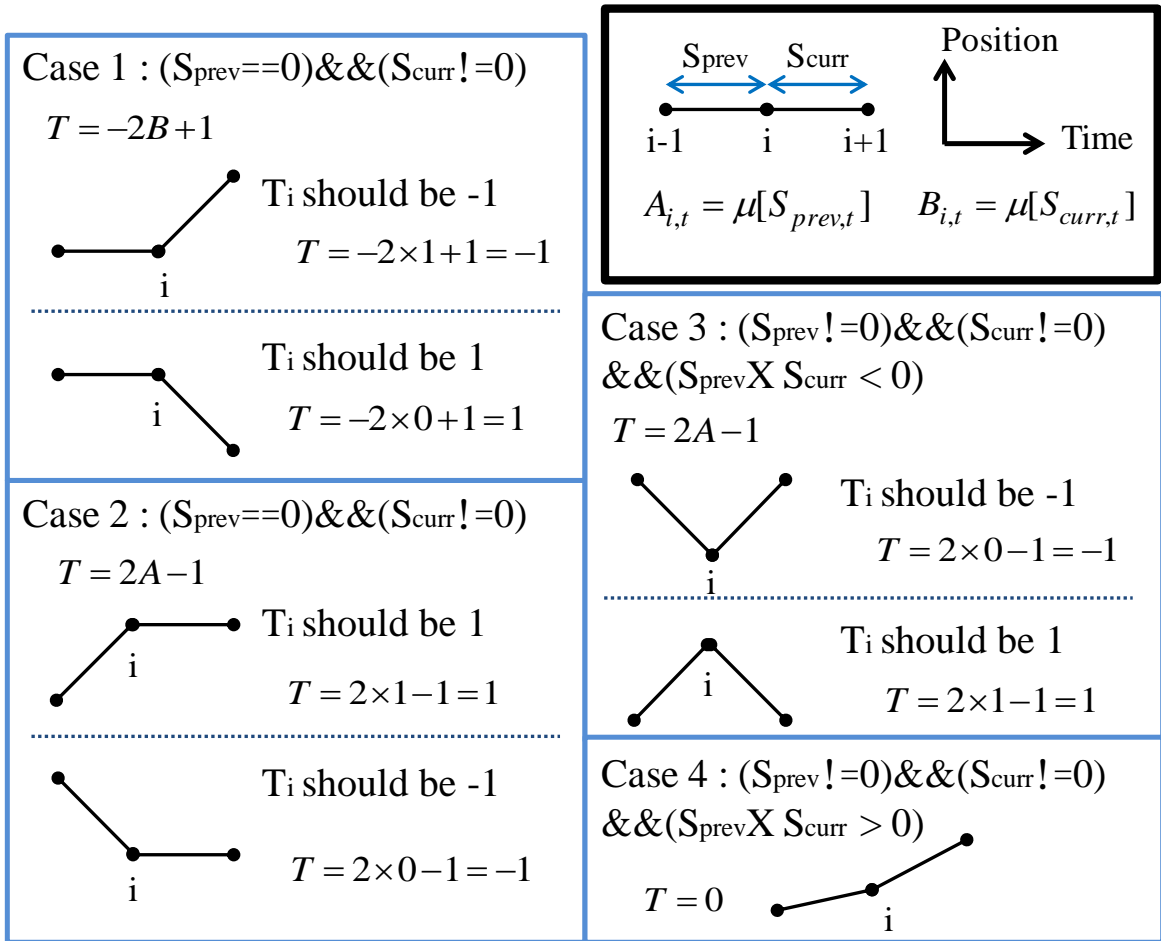


圖4-10 非邊界取樣點之代表值計算

步驟 1-2：代表點數量多於一演算法(More-than-one-RP algorithm)

本階段主要因應步驟一求得數個取樣點的代表值後，進行代表點的判斷。將所有上點(代表值：+1)與其相鄰的下點(代表值：-1)，取其平均，此平均位置點及相應位置資訊即為一個代表點的位置和時間資訊。此外，本階段尚須將輸出代表點數控制在大大於一的情況，以利 SPV 和 SRO 的計算。以下為演算法公式：

$$\bar{R}_t = [\bar{C}_0 \quad \bar{C}_1 \quad \dots \quad \bar{C}_{\alpha-1}]^T \dots\dots\dots (4-33)$$

$$\bar{C}_i = [C_i^P \quad C_i^T]^T \dots\dots\dots (4-34)$$

$$C_i^P = (GMV_{not-zero, T_t} + GMV_{dif, nearest T_t}) / 2 \dots\dots\dots (4-35)$$

$$C_i^T = (Tap_{not-zero, T_t} + Tap_{dif, nearest T_t}) / 2 \dots\dots\dots (4-36)$$

if ($\alpha < 2$)
 $\bar{R}_t = [\bar{C}_0 \quad \bar{C}_1], \bar{C}_0 = [GMV_{t-4} \quad 0]^T, \bar{C}_1 = [GMV_t \quad 4]^T, \alpha = 2 \dots\dots\dots (4-37)$

α 為代表點總個數。 \bar{R}_t 為 t 時刻代表點集合，每個代表點 \bar{C}_i 包含代表位置資訊的 C_i^P 和代表時間資訊的 C_i^T 。 $GMV_{not-zero, T_t}$ 為擁有非零代表值的取樣點其對應的 GMV 資訊， $Tap_{not-zero, T_t}$ 擁有非零代表值的取樣點時間編號。

公式(4-35)可知，由「擁有非零代表值的取樣點其所對應的 GMV 位置資訊」與「鄰近但擁有非零異號代表值的取樣點所對應的 GMV 位置資訊」取平均；同理，將上述兩取樣點的時間編號資訊取平均，此平均位置和平均時間即可組成一組代表點資訊。公式(4-37)，控制代表點數量。當小於二時，取首末兩取樣點資訊當作本時刻的兩個代表點。

步驟 2：區段內資訊取得 – SPV 與 SRO 演算法

取代表點的步驟結束後開始進入 Section panning velocity(SPV)和 Section representative offset(SRO)的計算過程。上回代表點數量因演算法的控制，個數只有大於或等於二，我們的判斷式會針對這種情況進行計算。

SPV 與 SRO 演算法公式：

$$[C_l^P \quad C_l^T]^T = \frac{1}{\alpha-1} \cdot \sum_{i=0}^{\alpha-2} [C_i^P \quad C_i^T]^T \dots\dots\dots (4-38)$$

$$[C_r^P \quad C_r^T]^T = \frac{1}{\alpha-1} \cdot \sum_{i=1}^{\alpha-1} [C_i^P \quad C_i^T]^T \dots\dots\dots (4-39)$$

$$S_t = \frac{Motion}{Time} = \frac{C_r^P - C_l^P}{C_r^T - C_l^T} \dots\dots\dots (4-40)$$

$$P_t = \frac{C_r + C_l}{2} \dots\dots\dots (4-41)$$

α 代表代表點總個數。在位置和時間的二維平面中， C_i^P 代表第 i 個代表點的位置資訊， C_i^T 代表第 i 個代表點的時間資訊， C_l^P 與 C_l^T 分別為最左代表點集合平均點的位置和時間資訊， C_r^P 與 C_r^T 分別為右代表點集合平均點的位置和時間資訊。 S_t 為 t 時刻的 SPV， P_t 為 t 時刻的 SRO。

當代表點個數大於二，利用公式(4-38)(4-39)(4-40)(4-41)，則將最左邊代表點與除了最右代表點之外其餘代表點做平均(4-38) 得一點(C_l^P, C_l^T)，視其為左代表點集合的平均；同理，再將最右代表點與除了最左代表點之外其餘代表點做平均(4-39)得點(C_r^P, C_r^T)，視其為右代表點集合的平均。將與二點取中間斜率得

SPV(4-40)，而兩者位置值的平均視為 SRO 之解(4-41)；當代表點個數為二，則直接將 Section 首項 GMV 位置資訊當作公式(4-38)的 (C_i^P, C_i^T) ，同理將 Section 末項 GMV 位置資訊當作公式(4-39)的 (C_r^P, C_r^T) 。SPV 為同樣為兩平均集合代表點間的斜率(4-40)，SRO 為兩點的平均(4-41)。

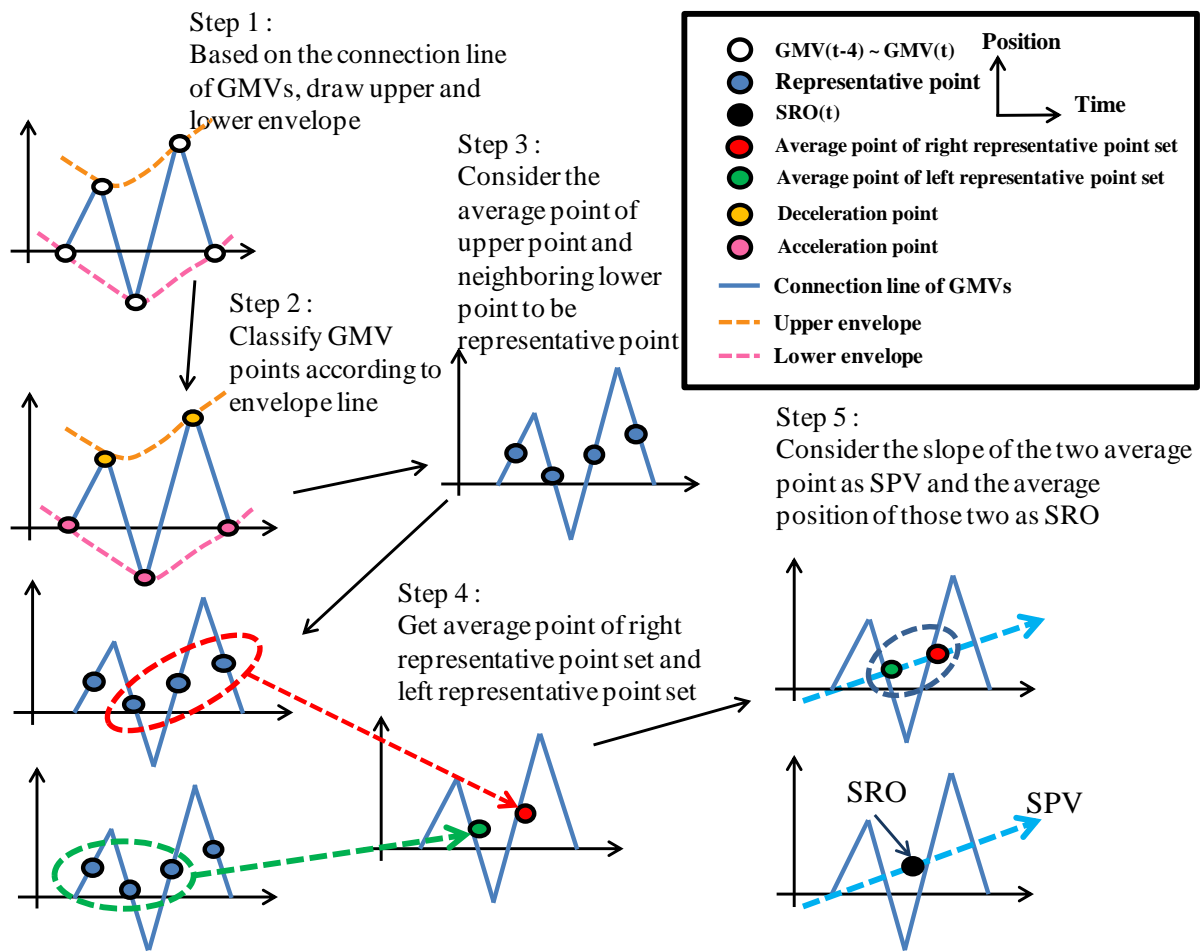


圖4-11 區段之預期速度與代表性偏移計算流程

4.3.5 權重值計算

權重 $\omega(t)$ 的設定(4-42)，包含了兩項參數，不穩定性和高斯函數成分。前者，滿足我們能將越不穩定的 Section 賦予越小的權重，而後者滿足越接近當下時刻的 Section 資訊權重越大。我們的不穩定性以兩兩 GMV 之間的加速度量值總和來決定，而高斯函數則是以累積常態分配函數(Cumulative distribution function)方式來配給。詳細公式如下所示：

$$\omega_t(T) = \Omega_t(T) / \kappa_t \dots\dots\dots (4-42)$$

$$\kappa_t = \sum_{i=0}^{W-2} |GMV_{t-i} - 2 \cdot GMV_{t-1-i} + GMV_{t-2-i}| \dots\dots\dots (4-43)$$

$$\Omega_t(T) = \frac{\delta}{\sigma\sqrt{2\pi}} \int_{i=-\infty}^{i=t-T} \exp\left(\frac{-(4-i)^2}{2 \cdot \sigma^2}\right) dt \dots\dots\dots (4-44)$$

$$\omega_t(T) = \frac{1 - \frac{\delta}{\sigma\sqrt{2\pi}} \int_{i=-\infty}^{i=t-T} \exp\left(\frac{-(O(i))^2}{2 \cdot \sigma^2}\right) dt}{\sum_{i=0}^{W-2} |GMV_{t-i} - 2 \cdot GMV_{t-1-i} + GMV_{t-2-i}|} \dots\dots\dots (4-45)$$

$\omega_t(T)$ 為 t 時刻移動平均窗內第 T 個 Section 的權重值，而 $\Omega_t(T)$ 為 t 時刻第 T 個 Section 的高斯函數值。W 為 Section 範圍，數值為 5。 κ_t 為 t 時刻移動平均窗內最新 Section 的不穩定值。 δ 參數是高斯函數最大值，本實驗設計值為 5，等同 Section 範圍。標準差 σ 的參數大小為 1.25，確保第 t-9 時刻取樣值權重持有的最小機率夠小 (6.87×10^{-4})，而第 t 時刻取樣值權重持有的最大機率足夠接近 1 (其值為 0.99996)。

不穩定性的公式意義

Section 暫存器儲存著五張 Frame 之內的 GMV 變化，共有五個 GMV 位置資訊。各張 Frame 之間 GMV 用兩兩 Frame 間的 GMV 相減可以算出的相對斜率，意義即為速度；根據微分關係，若再將速度之間進行相減，即為物理上的加速度量值。加速度總和值為急劇變化的象徵，數值越大表示 Section 內 GMV 的抖動越劇烈，將來該 Section 的參考價值應越低；反之，加速度總和若越小，表示 GMV 資訊等速度的可能性較高，或該 Section 未受抖動影響而顯示 EMV 成分較多的可能性也較高，因此該 Section 在決定 CMV 步驟的參考比重應越高，此為不穩定性函數如此定義的原因。下圖[圖 4-12]代表一個 Section 暫存器內的情況，水平軸方向為 Frame、垂直方向為 GMV 位置變化。圖中可看出區段包含不同的五個 GMV 值輸入情況和相應的不穩定值計算結果，GMV 穩定度越高(視覺判斷即可看出)的 Section 套用我們所設計的不穩定性函數產生的不穩定性值必定越低。

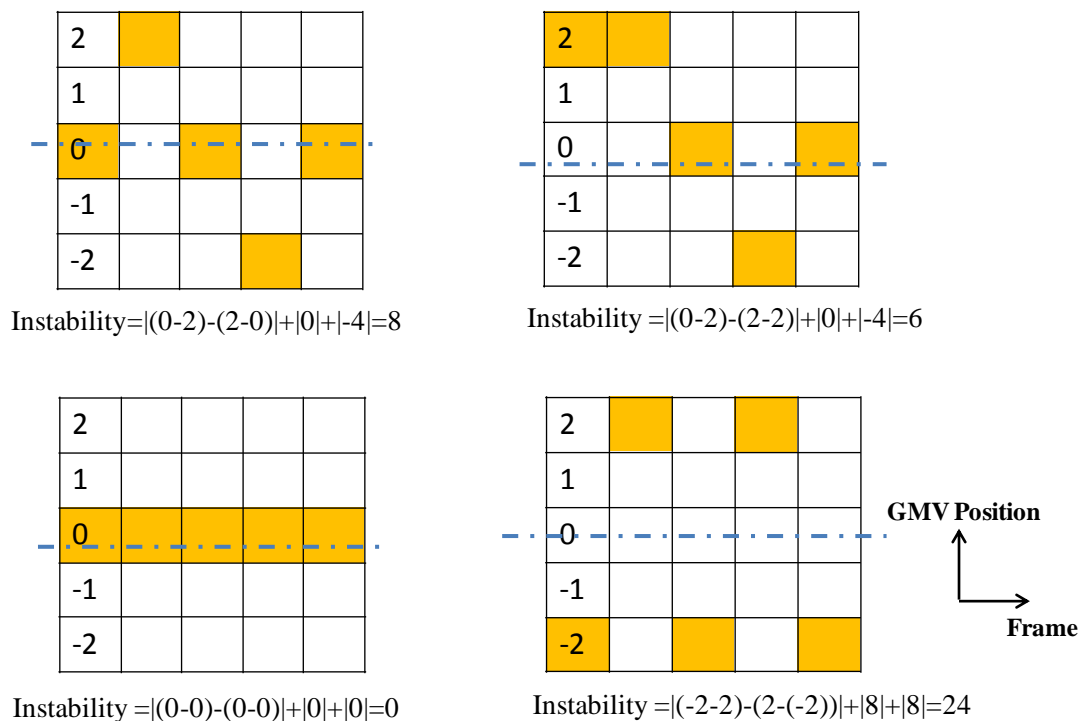


圖4-12

區段不穩定值計算範例

不穩定性函數的設定目標是遇到越平滑的 GMV 資訊情況，不穩定值應越小。預測型影像穩定器本身會考量前時刻的 GMV 值做本時刻的校正，所以當決定本時刻的 CMV 時，我們由該時刻的 GMV 資訊的穩定度來分配「決定 CMV 值時，該時刻 Section 資訊在判斷式中所應佔的比重」。此時越穩定平滑的 Section 在決定 CMV 比重時佔的影響力應該越大；反之，越不穩定的 Section，影響力理應越低。另外，用自我運動向量成分的比重來區分，Section 計算得到的不穩定值越小，表示該線段未受抖動影響而顯示自我運動成分較多的可能性也會越大。因此，不穩定性函數能利用 Section 內 GMV 的平滑度和自我運動的成分比重兩個原因指出各個 Section 的重要性，這也是我們補償演算法需要不穩定性函數系統的原因。

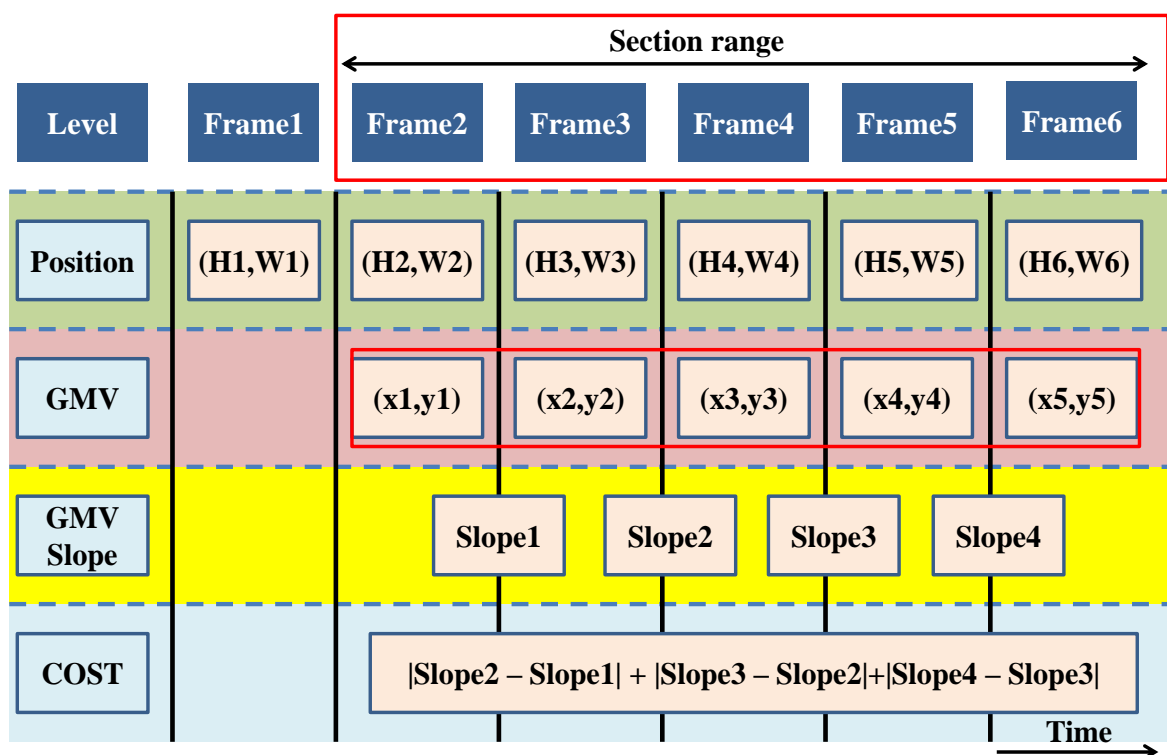


圖4-13 區段不穩定值之設計意義

4.3.6 補償值運動向量之計算

演算法中 CMV 補償值的決定方法(4-46)(4-47)由所估計的 EMV 與原訊號相減而來。EMV 的估計主要考量本時刻代表性偏移(SRO)資訊與上一回 EMV 的線性組合，再加上當下重疊區域暫存器(OSR)內每個 Section 的區域預期平移速度(SPV)值以線性組合方式來決定速度的大小。其演算法公式如下所示：

$$\bar{\omega} = \sum \omega(s) \dots\dots\dots (4-46)$$

$$EMV(t) = (1 - \alpha)EMV(t-1) + \alpha \cdot p(t) + \frac{1}{\bar{\omega}} \sum \omega(s) \cdot v(s) \dots\dots\dots (4-47)$$

$$0 < \alpha < 1$$

$$CMV(t) = GMV(t) - EMV(t) \dots\dots\dots (4-48)$$

$EMV(t)$ 為 t 時刻的 EMV， $GMV(t)$ 為 t 時刻的 GMV， $CMV(t)$ 為 t 時刻的 CMV， s 為該時刻 OSR 內的 Section 數， β 係數控制和上一回 EMV 資訊線性組合的比率，我們的實驗結果建議使用 0.45。 $\bar{\omega}$ 為該時刻所有 Section 各自權重的總和。 $p(t)$ 為 t 時刻的 SRO。 $PMV(t)$ 是計算前時刻 CMV 和 PMV 值和利用該時刻 section 位置 $p(t)$ 配合線性組合的速度值 $v(s)$ 得出結果。

此外，如前述部分所提，由於本演算法是套用目前時間點以前的五張 Frame 的 GMV 資訊來獲得 SRO 資訊故校正出的 CMV 資訊必定需要慢一張 Frame(該現象於附錄 A 解釋之)。故本時間點補償用的 CMV 資訊要等算完下一張 Frame 時間的 Feature vector 資訊再計算出該時刻的 CMV 值。由於僅為二維影像穩定，我們算得 CMV 值後，將補償值回授給原影像，再將修正後的原影像壓製成影片後輸出，完成影像穩定。

第五章 研究結果

本章介紹本文 DIS 系統的實驗環境與探討實驗結果，5.1 介紹實驗環境與設備，5.2 分別針對模擬和實際拍攝兩種實驗影片在準確性、平滑度、穩定向量輸入情況和執行時間等四種數據上分析實驗結果並做深入探討。

5.1 實驗環境及說明

數位影像穩定器的驗證主要是由混合著抖動向量訊號(Jitter motion)和自我運動向量訊號(Ego-motion)的輸入訊號與經過影像穩定後的輸出訊號(Stabilized motion)的差異程度來判別，從兩訊號之間的相似度可以判斷一個 DIS 的品質。為了能夠精確的獲知輸入和輸出，我們設計一種能客觀擷取出輸入與輸出的模擬實驗環境。首先，我們建構的實驗環境以固定場景(Fixed scene)拍攝(解析度 1920 X 1080)的影片，第二，將設定抖動向量訊號和自我運動向量訊號混合並輸入，第三，由輸入的混合訊號當作位置資訊，在原始高解析度(解析度 1920 X 1080)的 Frame 中抽出所對應較低解析度(解析度 640 X 480)的 Frame，接著將所有低解析度 Frame 所合成的影片套至 DIS 內並直接由程式內部穩定訊號後輸出結果。

由於我們的影像穩定器系統分開處理水平向變化 x 值和垂直向變化 y 值，可以分別設定不同情況來測試。我們輸入影片設定自我運動向量資訊的水平偏移 x 值包含加速度和方向轉折等情況，而垂直偏移 y 值則是固定於零的情況；抖動向量訊號則是在正負三像素單位之間隨機變化。如此設定的結果，使得 y 部分值因為抖動向量可以充分發揮，受到抖動影響較大； x 值則因為有速度和加速度變化有機會抵消抖動向量，受到影響相對較小。

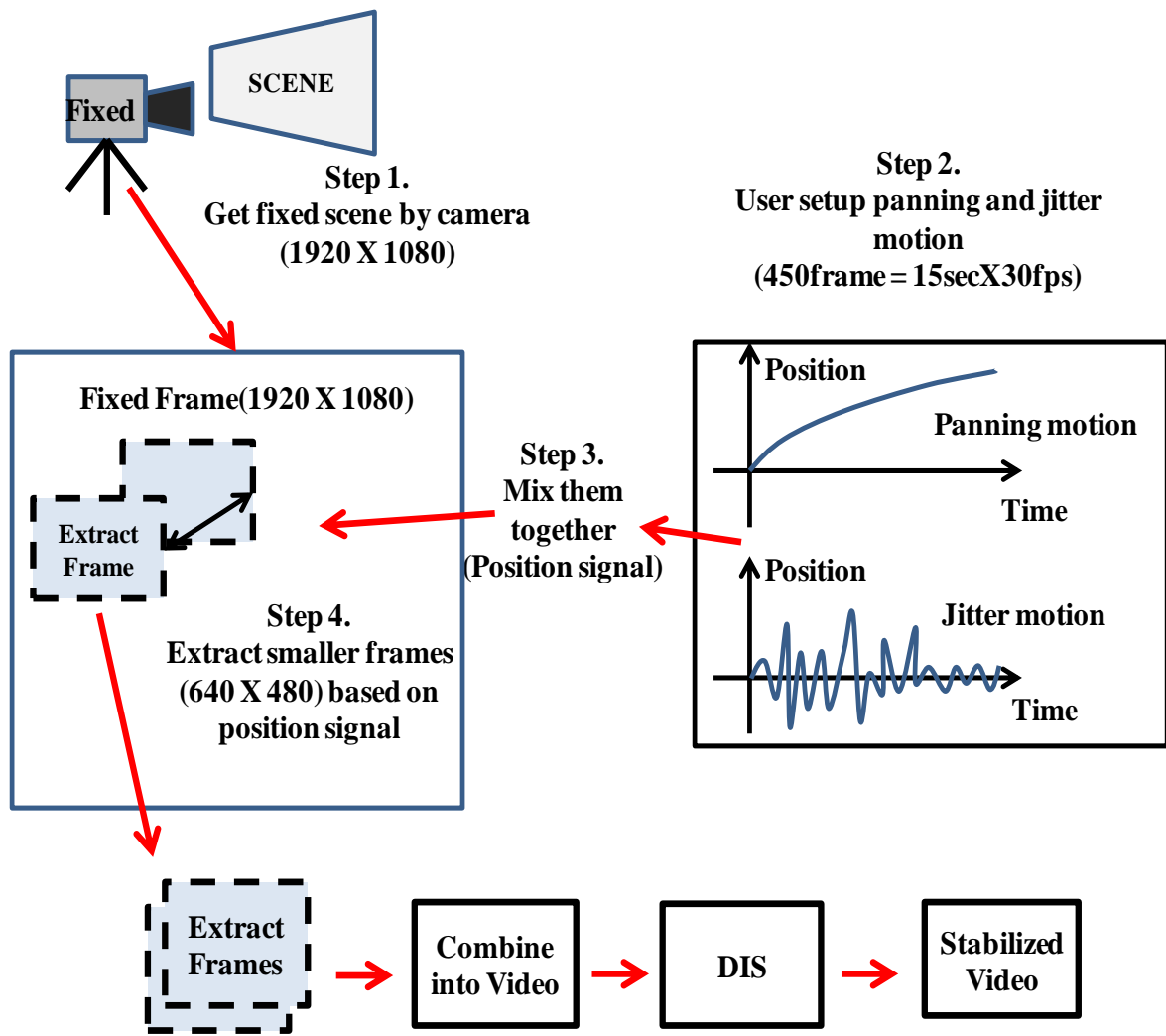


圖5-1 模擬實驗之實驗環境設計

5.2 影像穩定成果比較

實驗模擬結果主要有四類檢驗，包含準確率、平滑度、穩定曲線輸入的結果和執行速度。以下，我們將依序於各小節(5.2.1~5.2.5)中展示與討論。實驗所使用的設備上，我們的拍攝相機為 JVC GZ-HM960BU 相機，該相機感測器為 CCD 感測器。CCD 感測器，有別於 CMOS 感測器擷取像素資訊是一排排掃描來獲得，其優勢在於像素資訊擷取是一張 Frame 同時刻擷取，使得抖動情況發生時，不會造成擷取資訊的形變。若是 CMOS 感測器一排排掃描非常容易讓影像部分物體形變造成實際拍攝後進行 DIS 的結果不佳。



圖5-2 實驗拍攝用之相機

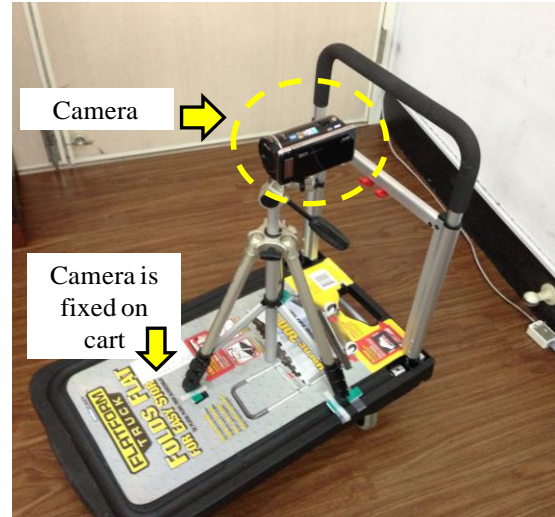


圖5-3 拍攝相機與手推車架設

我們在真實拍攝的實驗是將相機架設於手推車上，使其因為路面不平造成的抖動進行影片錄製。我們將手推車進行前進、後退與左右橫向錄影，讓實驗能在各種使用狀況下驗證演算法的強健性。實驗過程中所出現的影片抖動將來自三個部份：沒有懸吊系統校正的手推車行走於柏油路面傳來的抖動、腳架與手推車的連結的不穩定性與移動手推車的使用者在推動方向上的不確定性。因此，我們的影像穩定器實驗並不純粹僅解決來自路面的抖動，而是更全面的針對整體實驗環境產生的抖動。

5.2.1 影像穩定結果圖

首先，以位置一幀二維座標圖呈現結果，本文的抖動現象平滑化方法在模擬實驗中，針對四百五十張 Frame 內取其中速度轉折處一百張 Frame 的影像穩定結果如下圖[圖 5-4]所示：在位置一幀二維座標圖中充斥抖動成分的 GMV 值(黑色線部分)，通過我們的補償值系統確實能將其曲線中抖動的成分平滑化(紅色線部分)。圖中僅擷取在水平軸在第三百二十五張 Frame 中 GMV 明顯轉折處的結果，是為了模擬鏡頭平移方向發生巨幅改變的狀況，我們的補償值演算法對於巨幅轉折有略為延遲的現象，這是所有預測型演算法都會面對的狀況。

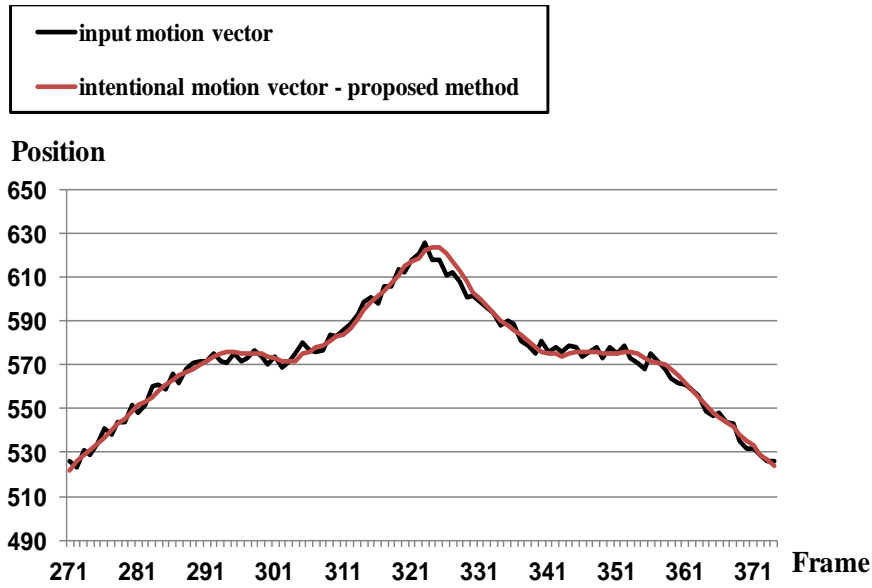


圖5-4 本文之位置一幀二維影像穩定結果截圖

5.2.2 均方根誤差值比較

在進行驗證穩定的準確性方面，我們發現若 Jitter motion 有影響前幾張 Frame 資訊，且當 Ego-motion 輸入訊號在該時期恰為不動(Fixed)情況，則當初設定的 Ego-motion 成分必定會與擷取的 GMV 位置資訊的理想穩定情況有些微偏移差距。這個狀況會造成常使用的檢驗方法均方根誤差(Root mean square error, RMSE)有很大的差距。RMSE 為比較 DIS 穩定後的數值和理想數值的方均根誤差值，若是 RMSE 越小，則穩定結果越接近 Ego-motion。

$$RMSE = \frac{\sum_{i=0}^N \sqrt{(x_i - x_{ego})^2 + (y_i - y_{ego})^2}}{N} \dots\dots\dots (5-1)$$

x_i 為該 DIS 方法所估計的自我運動的水平 x 方向資訊， x_{ego} 為預設輸入的 Ego-motion 的 x 方向資訊；同理， y_i 為該 DIS 方法所估計之自我運動的垂直 y 方向資訊， y_{ego} 為預設輸入 Ego-motion 的 y 方向資訊。N 為總處理 Frame 數。

表5-1 各補償值計算方法模擬實驗之 RMSE 值結果表

RMSE result of simulation			
Video	Video#1	Video#2	Video#3
ACM	4.3394	4.8326	4.3784
EMD	3.5204	4.2209	3.7176
Proposed method	4.2166	4.6027	4.0367
Video	Video#4	Video#5	Video#6
ACM	4.3550	6.9584	138.0421
EMD	3.6272	6.4523	138.0427
Proposed method	4.1135	7.4125	134.4124

如表[表 5-1]所示，是由 30 Frame per second(fps)下四百五十張 Frame 扣除演算法前端不處理狀況的 Frame 數後(前後共扣了三十張 Frame，剩四百二十張 Frame 做比較)，在本表比較三種方法：(a)累積補償值型 ACM[11]、(b)後處理 EMD[10]以及(c)本文所提出的方法，所計算的 RMSE 結果。此外，前述所提之擷取結果偏移現象，我們以下圖表示[圖 5-5]。早期設定 Ego-motion 為位置由零開始，但是因為抖動影響整體擷取到的 GMV 絕對位置與 Ego-motion 有明顯偏移。本例 Ego-motion 是較慢於擷取結果所擷取的 GMV 絕對位置資訊，所以我們會需要人工方式調整理想上的 Ego-motion 使其較貼近 GMV 的絕對位置線，使得 RMSE 數值更為客觀。

然而，除了初始值影響 GMV 資訊的擷取造成偏移之外，尚需考量各個影片中的干擾因素，諸如移動物體、重複圖案等狀況會造成的 GMV 資訊擷取失敗。舉 Video#6 為例，其 RMSE 數值普遍高於其他影片的原因是因為影片中可能因為有相關干擾因素使得 GMV 資訊的擷取不如預期，造成偏離於預設的 GMV 資訊過大，讓 DIS 後的自我運動位置曲線仍舊與原先所設大相逕庭，這是本研究模擬實驗可能面對的難題。

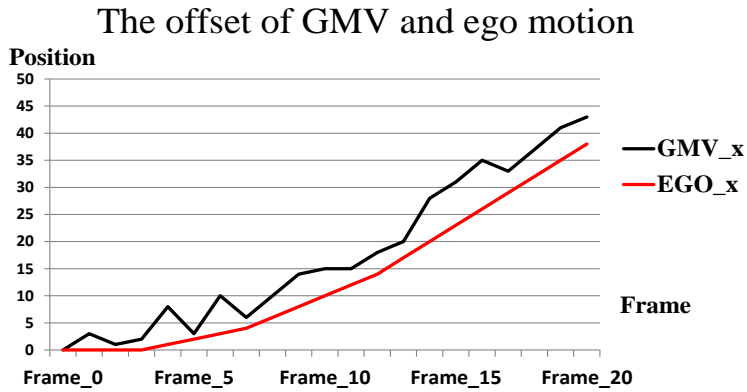


圖5-5 抖動向量與理想位置差異造成的偏移現象

表5-2 各補償值計算方法模擬實驗之平均 RMSE 值比較表

Final average result of RMSE	
Method	Average
ACM	27.1477
EMD	26.5969
Proposed method	26.4657

我們將六部影片所計算得的 RMSE 值取其總平均後比較個方法，由上表所示。由結果得知，我們的穩定方法在方均根誤差的總平均比較上優於同為預測型 ACM(RMSE 值 27.1477)的演算法，而對比於後處理型的 EMD 方法(RMSE 值 26.5969)，我們的方法(RMSE 值 26.4657)正確率仍為最佳但是差距不大。RMSE 值的結果會顯示 EMD 方法比我們的方法更穩定，這是因為我們的演算法是預測型演算法，並沒有如 EMD 計算當下補償值時能參考當下時間點之後的大量 GMV 資訊。當系統讀入的 GMV 範圍能分析出該區段訊號為抖動時才會將正確路徑讀出，這是所有預測型 DIS 的共同限制。但是正因為預測型的演算法僅用當下時間點前的 GMV，才有支援即時運算的能力，這使得預測型 DIS 在應用消費型電子的重要性遠大於後處理型 DIS。

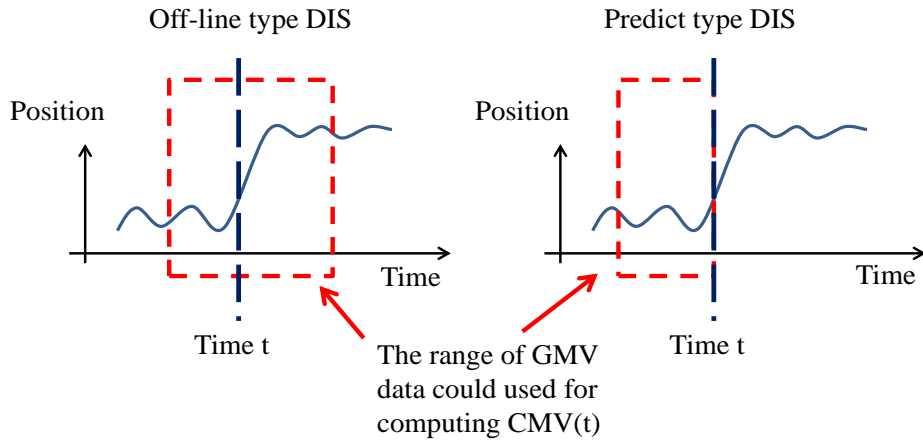


圖5-6 後處理與預測型穩定器所擁有的全域資訊範圍差異

因此，我們的DIS穩定器雖須延遲一張Frame為代價，輸出結果上仍能比僅參考前時刻資訊的預測型演算法有更好的準確性。

5.2.3 平滑度值比較檢測

另一種檢驗方法：Smoothness index(SI)[26]由過去文獻所使用，該值是以下列公式求得：

$$IFMV(t) = EP(t) - EP(t-1) \dots\dots\dots (5-2)$$

$$SI = \frac{1}{N-2} \sum_{i=2}^N M \cdot |IFMV(t) - IFMV(t-1)| \dots\dots\dots (5-3)$$

$EP(t)$ 代表 t 時刻的自我運動位置(Ego-motion position, EP)， $IFMV(t)$ 為 t 時刻交互幀的運動向量(Inter-frame motion vector, IFMV)，是由穩定器系統輸出的已平滑化全域運動位置相鄰 Frame 的位置差值，SI 為所求平滑度， N 為總資訊個數。 M 為每張 Frame 的比重(Mass of frame)，通常可由使用者設固定常數值或零與一之間的值，本文設定該值為一。 N 為總 Frame 數。 i 為 Frame 的編號，代表目前為第幾張 Frame。

由於 SI 定義上只使用二維(位置軸, 時間軸)的數據, 我們期望能比較三維(x 軸,y 軸,時間軸)便選擇將 x 值與 y 值各取其 SI 值, 後將兩者取平均。我們再度比較三種方法, 而下表[表 5-3]實驗結果顯示我們的平均平滑程度優於其於二者。在與 CMV 平衡型比較, 我們兩軸的平滑程度都較佳; 在與 EMD 比較, 我們在 x 軸的平滑度較優, 但是在 y 值的平滑度上 EMD 較佳。

互有優缺的比較結果中, 我們能在 x 軸表現較佳的原因, 是因為實驗環境上 x 方向設定的轉折和加速度使得抖動成分影響較低, 而存在 Mode mixing 問題的 EMD 方法會在此種情況將抖動成分當作平滑曲線的機會因此提升; y 軸部分, 則是 Fixed mode 的輸入訊號使得抖動成分影響較大。因為 EMD 後處理的特性的大局觀會明顯優於預測型的方法, 我們在這 y 部分的數值略遜於 EMD, 但是平均平滑度上仍舊為三種方法中最佳的。

表5-3 各補償值計算方法於模擬實驗一號影片之 SI 值結果表

Result of smoothness index(SI)			
Method	x-axis	y-axis	Average(x,y)
ACM	0.7911	0.7400	0.7656
EMD	1.1311	0.2089	0.6700
Proposed method	0.6844	0.4933	0.5889

為了檢視各個 DIS 方法所輸出的自我運動在平滑度上的強健性, 我們錄製六個實驗模擬用的影片, 其以各方法處理後的平滑度比較結果如下表所示:

表5-4 各補償值計算方法模擬實驗之平均 SI 值比較表

Final average result of smoothness index(SI)	
ACM	0.7713
EMD	0.6939
Proposed method	0.5628

表5-5 各補償值計算方法模擬實驗之 SI 值結果表

Result of smoothness index(SI)			
Video	Video#1	Video#2	Video#3
ACM	0.7656	0.7367	0.7500
EMD	0.6700	0.6722	0.6800
Proposed method	0.5889	0.5378	0.5556
Video	Video#4	Video#5	Video#6
ACM	0.7667	0.7522	0.8567
EMD	0.6667	0.6700	0.8044
Proposed method	0.5422	0.5522	0.6000

針對上述上述六個模擬實驗的影片，我們可以得出其總平均的 SI 值，其結果比較於上表。我們的演算法的平均平滑度(SI 值 0.5628)領先 ACM(SI 值 0.7713)和 EMD(SI 值 0.6939)方法。此外，平滑度公式使用上相當便利，有不需原始理想值資訊的優點。當使用穩定器於實際拍攝時，平滑度公式不需要數據化的自我運動資訊，而是僅以影片輸出的平滑性為探討依據。因此，我們接著在真時拍攝下進行實驗。

在真實拍攝下的八個影片中，我們使用 640 X 480 的解析度進行實際拍攝後輸入影像穩定器。我們的演算法在實際拍攝情況下，平滑度值的結果仍是三種方法中最佳的。影片中包含往復錄影與水平錄影。將八部影片的 SI 值取其總平均 SI 值於下表，結果顯示我們所提出的平滑化演算法以 SI 值 0.7153 占據優勢。如此可見，我們的方法在平滑度上，除了模擬實驗之外，在真實拍攝下也有優秀的平滑化結果。

表5-6 各補償值計算方法真實拍攝之平均 SI 值比較表

Final average result of smoothness index(SI)	
ACM	0.8873
EMD	1.3895
Proposed method	0.7153

表5-7 各補償值計算方法真實拍攝之 SI 值結果表

Result of smoothness index(SI)				
Video	Video#1	Video#2	Video#3	Video#4
ACM	1.1944	0.9878	0.8489	0.8111
EMD	1.3867	1.7267	1.2411	1.3056
Proposed method	0.7744	0.6944	0.6656	0.6000
Video	Video#5	Video#6	Video#7	Video#8
ACM	1.0300	0.8744	1.4511	1.3522
EMD	1.7400	0.9500	1.3189	1.4467
Proposed method	0.6878	0.6056	0.8822	0.8122

5.2.4 穩定運動向量輸入的比較

大部分影像穩定器設計者在文中並未提到其在理想 Ego-motion 情形下的穩定結果，這是因為一般設計者的實驗環境沒有能夠擁有理想上的 Ego-motion，意即沒有絕對正確的對照依據。我們建立的實驗環境是由高解析度 Frame 上依照設定好的 Ego-motion 和 Jitter motion 混和訊號作為依據來抽取低解析度的 Frame 來做實驗輸入，所以可以進行這項驗證。這種驗證是通常在加速度的運動向量才可看出演算法之間的明顯不同，因為演算法通常會在加速度部分設計後驗估計的補償值，以利於下一 Frame 的 CMV 預測；反之，完全等速度的情況，大部分演算法都不會有多餘補償值的問題，其結果通常與真實 Ego-motion 接近程度較高。我們再度比較三種方法：

表5-8 各補償值計算方法以平滑訊號輸入之平均 RMSE 值比較表

RMSE when input smoothed motion	
Method	RMSE
ACM	1.0667
EMD	0.0000
Proposed method	0.6889

表5-9 各補償值計算方法以平滑訊號輸入之平均 SI 比較表

Smoothness Index(SI) (Mass = 1)			
Method	x-axis	y-axis	Average(x,y)
ACM	0.4244	0.0000	0.2122
EMD	0.1333	0.0000	0.0667
Proposed method	0.3689	0.0000	0.1844

RMSE 比較結果 [表 5-8]中，後處理型 DIS 的 EMD 方法，其大局觀較強的優點非常明顯，其在輸出結果上與原始輸入的 Ego-motion 相似度最高。但我們所提出的演算法在純粹 Ego-motion 輸入的 RMSE 值比較上，結果仍舊比同為預測型的 ACM 好。這表示我們的演算法方法在平滑訊號輸入的情況下有較少的額外抖動，而使用累積資訊來估計補償值的方法可能會因為錯誤累積而產生較多額外抖動。

SI 平滑度上[表 5-9]，由於 EMD 方法在 RMSE 表現上已經完全貼近預設的 Ego-motion 故其 SI 值在客觀上來說等同於 Ego-motion 的平滑度。在垂直位移 y 軸等速度情況上，三種演算法都能完全準確；在水平位移 x 軸加速度情況上，我們提出的演算法相較於 ACM，仍舊獲得較佳的結果。這顯示我們把隨時間讀入的 GMV 資訊以 Section 方式做微濾波功能在 DIS 輸出結果上，能比僅用本時刻和上一時刻兩點資訊來判斷 CMV 的演算法設計具有更高的平滑度。

5.2.5 執行速度結果與分析

我們模擬程式用的電腦配備為處理器 3.3GHz Intel® core(TM)i5-2500 CPU、3.17RAM。實驗將四百五十張 Frame 的 GMV 資訊分別比較 ACM、EMD、本研究方法，結果於下表所示：

表5-10 各補償值計算方法之執行速度比較表

Execution time of 450 Frames	
Method	Time(second)
ACM	0.016
EMD	0.078
Proposed method	0.018

ACM 方法因為演算法容易，在執行時間最短，而本研究所提出的補償值演算法計算次之。使用 EMD 方法最為耗時，因為 EMD 需要搜尋有效線段和分段迭代等耗時步驟。所以，我們的方法與其他方法的整體比較表：

表5-11 各補償值演算法整體比較表

Table of method comparison			
Method	Accuracy (RMSE)	Execution time (second)	Smoothness (SI)
ACM	4.9728	0.016	0.7656
EMD	4.3077	0.078	0.6700
Proposed method	4.8764	0.018	0.5889

5.2.6 本研究方法的貢獻探討

我們所提出的方法記憶 t 時刻之前的 GMV 資訊，以 Section 為單位進行微幅濾波找出 t 時刻以前各時段速度，以加速度絕對值總和 (Sum-of-absolute acceleration) 所算得的 Instability 和 Gaussian 函數兩者決定 t 時刻各 Section 的參考價值，最後以上述資訊預估出 t 時刻可能的自我運動後修正影像，完成影像穩定。接著，我們在與其他 DIS 方法的特色作比較探討。

運用卡爾曼濾波器的 DIS 其主要是利用上時刻預期的速度配合本時刻所接收的資訊做校正並估計下時刻的值，運用該功能於 GMV 平滑化所造成的問題將有兩點。第一，校正資訊的選擇，由於影像穩定的原始輸入資訊皆為含有抖動成

分的 GMV 資訊，通常校正資訊的選用方式有直接取用 GMV 或是用其他方式轉化 GMV 至抖動成分較少的值域後再用於卡爾曼校正。若直接運用 GMV 做影像平滑化，表示設計者想借用卡爾曼濾波器的微幅校正特性進行平滑化。然而卡爾曼校正是一個緩慢的過程，這容易造成在速度變化處發生延遲現象。舉例來說，當本回合 GMV 位置在零位置，當校正資訊因為突發性抖動而出現在 K 位置時，卡爾曼校正後所給予的位置必定會座落於零至 K 位置之間；用其他方式轉化 GMV 則涉及新方法的穩定性和耗時性增加。第二點，除非下一回合的校正速度仍然相同，否則卡爾曼方法的輸出位置曲線會無法避免地產生些微抖動，特別是在急促抖動變化時尤其明顯。相較於卡爾曼濾波器，我們的方法因為 Section 的微幅濾波特性舒緩了 GMV 的抖動特性，讓演算法輸入值並不會太過極端造成需要設計出低靈敏度的演算法。

ACM 法主要面對的問題在於易受到極端值影響，這個現象起因於「判斷當下補償值僅使用前一刻 GMV 資訊」和「補償值累積遇上極端震動」兩特性。假設抖動為一正弦波半波信號，前者，讓抖動週期大於一張 Frame 的情況會讓補償值計算結果誤判；後者，讓自我運動方向改變時會有錯誤的累積值出現，且原始 ACM 沒有歸零動作，無法有效修正該問題。相對地，本研究採用移動平均的方式進行補償值計算，而非直接取用含有抖動成分的 GMV 資訊。因為我們的方法以 Section 為單位將抖動成分多的 GMV 資訊投影到抖動成分較低的空間域，而使得補償值計算上得以降低上述兩現象造成的問題。此外，Section 的延長能增加穩定的巨觀性，提升穩定品質但是會需要額外考慮延遲現象。Section 數為五張 Frame 內的 GMV 者和 Section 數為十張 Frame 內的 GMV，在處理穩定影像上後者能有相對較高的客觀性，但是要延遲的 Frame 數會上升。

EMD 方法最大的問題在於抽取本質模態函數(IMF)的部分，能否抽到頻率最低但不會使穩定結果曲解自我運動的 IMF 是決定該方法結果的關鍵因素。此

外，純粹 EMD 會因為邊界包絡線的選擇而受到 Boundary effect 影響，且完整 Frame 的位置資訊不一定能滿足抽取 IMF 的條件，所以實際運用上會要切割區段分別進行 EMD。我們在實驗上改進文獻[10]上的方法，將五十張 Frame 做 EMD 後取得中間二十張 Frame 的補償值輸出作為結果，確實改善 Boundary effect 的影響。前處理型 DIS 的方法相較於 EMD 主要可能的劣勢在於客觀性無法比擬後處理的 EMD，舉例來說，假設抖動信號是一個週期 T 的正弦波而當下進行到時段 0 至 T/2 之中，後處理的 EMD 可能會因為可視範圍較廣而獲得較為巨觀的結果；而前處理型會因為僅參照當下時刻之前的 GMV 資訊而無法認出抖動現象，我們的方法即使 Section 範圍沒有會覆蓋到整個抖動現象，重疊區段結構(OSR)能補足這方面的缺失。EMD 方法最容易失敗的地方在於速度轉折之處，因為該情況曲線不容易抽出 IMF，反之速度正是本方法計算補償值所參考的依據，我們在速度轉折的處理必定能優於 EMD。

本法不適用於將 OSR 的長度設定過長，也不需要過多提供處理時間點後方 GMV 資訊。舉例來說，若將處理時間點 t 後一至五個 GMV 資訊(GMV(t+1) ~ GMV(t+5))給予我們的 Section 結構，亦即延遲一至五張 Frame 的資訊輸出結果，本方法的結果如下表所示。我們針對模擬實驗 Video#2 做延遲三種情況和改變 OSR 考量範圍，探討均方根誤差值與平滑度值的變化：

表5-12 本文補償值演算法不同參數下之 RMSE 值比較表

RMSE result of different parameters			
	OSR 30 Frames	OSR 40 Frames	OSR 50 Frames
Delay 1 Frame	3.8597	3.9246	4.0059
Delay 3 Frames	6.5913	6.6394	6.6782
Delay 5 Frames	10.0855	10.1288	10.1174

表5-13 本文補償值演算法不同參數下之 SI 值比較表

SI result of different parameters			
	OSR 30 Frames	OSR 40 Frames	OSR 50 Frames
Delay 1 Frame	0.5822	0.5933	0.58
Delay 3 Frames	0.5767	0.5911	0.5789
Delay 5 Frames	0.5833	0.5989	0.58

OSR 的範圍大小是決定一個補償值中最多會考慮的 GMV 資訊個數，舉例來說，若 OSR 範圍為三十張 Frame，表示記憶體內用來決定一個補償值所儲存的有限個數 GMV，是運用在三十張 Frame 範圍內的資訊。由上表可知，改變 OSR 的範圍或是擷取更後方的 GMV 資訊，無益於讓本方法的結果更優異。在 RMSE 的部分，若是提升 OSR 資訊在 RMSE 部分，並沒有對降低誤差值有較明顯的現象，反而在延遲一張 Frame 時有升高誤差的現象。這是因為本方法考量的瞬時速度會因為過早判讀而使得所估計的自我運動位置曲線，在速度改變處會過早趨緩或過早上升，讓位置曲線偏離 GMV 資訊過多並讓 RMSE 誤差值提升。SI 值部分，則不論參數如何改變數值都介於 0.57 至 0.59 之間，表示本方法在抖動曲線平滑化功能相當穩定，無須使用特定參數讓輸出平滑度提升。

本方法為一預測型 DIS，提升擷取處理時間點後的 GMV 資訊量或拉長 OSR 涵蓋範圍皆無益於本方法且會讓記憶量增加。此外，上述兩種參數變換對輸出影像而言，可能造成輸出的自我運動與理想影像有明顯位置偏移的風險。因此，本法在平滑化功能不論是模擬實驗或是真實拍攝下使用的穩定度都相當高，且有無須設定多餘參數的優點。

5.2.7 加速度參數的補償

由於純粹利用區段預期速度(SPV)來進行自我運動的估計容易在加速度變化處產生延遲反應讓穩定結果的 RMSE 誤差值上升，我們可以選擇將加速度參數融入 Section 所計算的 SPV 以修正該現象。我們利用 OSR 內所有能利用到的 SPV 等資訊，算得加速度後修正原始 SPV 完成補償。假設我們的 OSR 範圍為三十張 Frame 內的 GMV 資訊(GMV(t-29)~GMV(t))，則表示在一次補償值計算會考慮到 26(=30 - 5 + 1)個 SPV。考量兩兩速度僅能算得一個加速度，我們只補償後 25 個 SPV。

$$A_s = v_s - v_{s-1} \dots\dots\dots (5-4)$$

變數 A_s 表示由第 s 個 Section 所算得的瞬時加速度， v_s 表示第 s 個 Section 的 SPV。我們接著將 25 個瞬時加速度依時間順序(s 範圍從 1 到 25)作簡化卡爾曼濾波器的輸入資訊，以估計出當下時刻的加速度。算得最後的瞬時加速度估計值後，該值將用來修正當下所有 Section 的 SPV。我們用簡化的卡爾曼濾波器公式估計加速度，公式如下所示：

$$\hat{E}_{s+1}^- = \hat{E}_s \dots\dots\dots (5-5)$$

$$P_{s+1}^- = P_s + Q \dots\dots\dots (5-6)$$

$$K_s = P_s^- / (P_s^- + R) \dots\dots\dots (5-7)$$

$$\hat{E}_s = \hat{E}_s^- + K_s (A_s - \hat{E}_s^-) \dots\dots\dots (5-8)$$

$$P_s = (1 - K_s) \cdot P_s^- \dots\dots\dots (5-9)$$

公式 \hat{E}_s 為區段 s 的後驗估計，初始值為首項加速度資訊， \hat{E}_s^- 為區段 s 的先驗估計， P_s 為估計誤差共變異數(Error covariance)， P_s^- 為先驗估計誤差共變異數(A priori error covariance)， Q 與 R 分別為猜測誤差(本文設 0.00001)與系統誤差(本文設 0.001)， K_s 為卡爾曼增益值。前兩個公式(5-5)(5-6)是屬於時間更新狀態，為估計功能，後三個公式(5-7) (5-8) (5-9)是屬於測量更新狀態，為校正功能。將 25 個加速度 A_s 值當作測量的校正資訊，依序帶入公式後，進行 24 次重複迭代所算出的 \hat{E}_s 即為所求。最後，用 \hat{E}_s 值修正當下原本的區段預期速度 v_s 並用修正後區段預期速度 v'_s 取代 v_s 來估計自我速度。

$$v'_s = v_s - \hat{E}_s \dots\dots\dots (5-10)$$

為了比較 RMSE 與 SI 的變化，我們選以模擬實驗的 Video#1 進行實驗，並特別擷取 GMV 速度轉折處的變化在位置—幀二維關係圖中比較結果。由下圖結果可知當補償值考慮加速度參數時，RMSE 誤差值由原本的 4.2166 下降至 3.6695，自我運動能夠更貼近理想值。因此，本方法加入加速度參數確實能讓穩定結果的誤差下降。

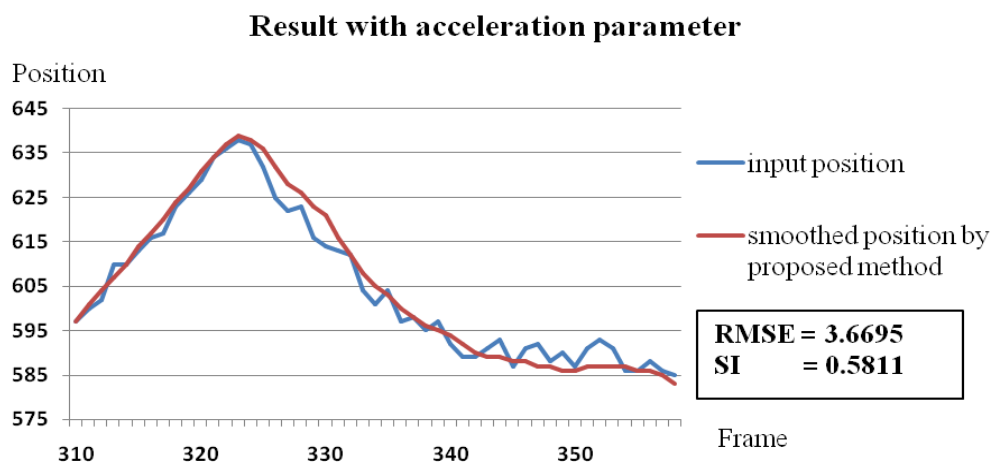


圖5-7 有加速度補償的本文方法穩定結果

Result with no acceleration parameter

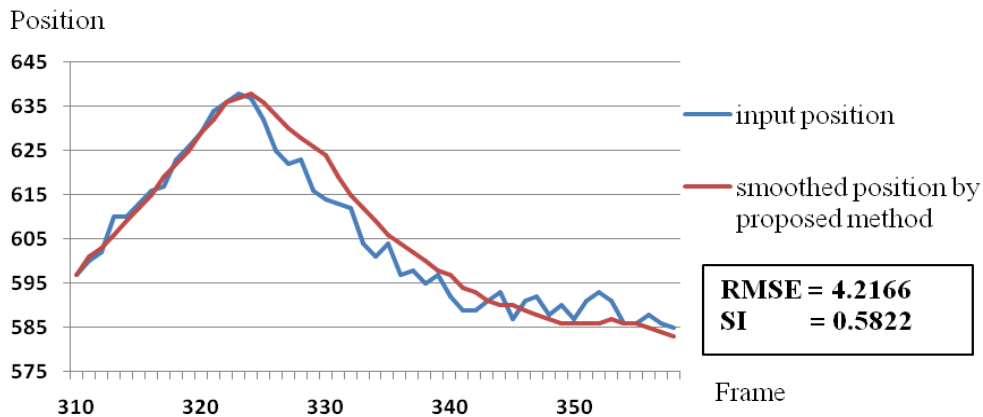


圖5-8 無加速度補償的本文方法穩定結果

表5-14 各補償值演算法模擬實驗之 RMSE 值結果表

RMSE result of simulation			
Video	Video#1	Video#2	Video#3
ACM	4.3394	4.8326	4.3784
EMD	3.5204	4.2209	3.7176
Proposed method	4.2166	4.6027	4.0367
Proposed method with acceleration	3.6695	4.2963	3.6869
Video	Video#4	Video#5	Video#6
ACM	4.3550	6.9584	138.0421
EMD	3.6272	6.4523	138.0427
Proposed method	4.1135	7.4125	134.4124
Proposed method with acceleration	3.8433	7.4093	134.7456

接著，我們再度就模擬實驗的六部影片進行誤差值與平滑度測試。結果顯示出當混合加速度參數時，整體平均誤差值將由 26.4657 下降至 26.2752。但是平滑度部分，由 0.5889 上升至 0.6069。因此，當加速度參數考慮時，有犧牲部分平滑度而降低誤差值的現象，執行時間亦會微幅上升。

表5-15 各補償值演算法模擬實驗之 SI 值結果表

Result of smoothness index(SI)			
Video	Video#1	Video#2	Video#3
ACM	0.7656	0.7367	0.7500
EMD	0.6700	0.6722	0.6800
Proposed method	0.5889	0.5378	0.5556
Proposed method with acceleration	0.5811	0.5822	0.6022
Video	Video#4	Video#5	Video#6
ACM	0.7667	0.7522	0.8567
EMD	0.6667	0.6700	0.8044
Proposed method	0.5422	0.5522	0.6000
Proposed method with acceleration	0.5878	0.5911	0.6967

表5-16 各補償值演算法之整體模擬總比較表

Table of method comparison			
Method	Accuracy (RMSE)	Execution time (second)	Smoothness (SI)
ACM	27.1477	0.016	0.7656
EMD	26.5969	0.078	0.6700
Proposed method	26.4657	0.018	0.5889
Proposed method with acceleration	26.2752	0.024	0.6069

因此，透過觀察模擬拍攝與實驗拍攝的輸出結果，本文提出的方法誤差值加上加速度參數後(RMSE 值 26.2752)比後處理型的 EMD 方法(RMSE 值 26.5969)還要好。此外，在輸出影片的平滑度上，不論是有否增加加速度參數數值都是最佳的，表示本方法在人眼視覺的感受上有最佳的結果。此外，ACM 和未加加速度參數的本方法的 RMSE 數值都相對較高，這個結果的原因在於 RMSE 誤差值主要上升的原因來自於估計的自我運動是否足夠貼近原始 GMV 數值。而影響該值

的因素來自於 DIS 界定當下 GMV 是「抖動狀態」或是「速度變化狀態」的準確性。一般型 DIS 因為只能獲得當下時刻之前的 GMV 資訊，讓 GMV 速度變化的早期會被 DIS 誤認為抖動成分而不反應，所以在速度變化部分的穩定常有延遲現象出現，進而造成 RMSE 值較高；後處理型的 EMD 則會因為有當下時刻之後的大量 GMV 資訊補足這部分的延遲，讓 RMSE 值較低。而我們的方法在加入加速度參數修正後，能補償預測型 DIS 在速度變化處的延遲現象，提升準確性。相較之下，平滑度公式較能反應人眼視覺感受，因為該值主要計算兩兩 Frame 之間的滑順度，這在結果影片與數值上皆能佐證本研究的優勢。所以，前者公式追求的方向在於學術理論上，後者追求的方向在於消費型電子應用上，二者皆有參考的意義與價值，本文提出的方法以一個預測型數位影像穩定器演算法而言，在兩項檢驗公式上都能展現優勢。

第六章 結論與未來展望

6.1 結論

我們提出一個使用權重式移動平均捲積觀念實現預測型 DIS 的方法，以估計區段所分析的預期移動速度和加速度得出自我運動並獲得補償值，進而成功地完成影像穩定。文中所提的演算法並沒有使用累積補償值的方式補償，因而避免速度變化時錯誤累積現象，所以可以進行長時期影像穩定。此外，加速度參數的補償，可以有效提升穩定器的準確性和解決預測型 DIS 的延遲現象

我們設計的 DIS 是分別處理垂直和水平的影像穩定，主要用途在於手持相機等低速相機拍攝，本演算法的限制主要有兩項。第一，若要將演算法套用至高速攝影，如高速行駛的行車紀錄器等應用，尚須增加 GMV 在 Zoom-in 或 Zoom-out 等情況的判別定義和相應的影像插補方法。行駛的車子所記錄的影像在前後運動狀態下會有消失點和物體形變等狀況發生，而單純使用垂直和水平位置的補償無法支援這些情形，這是本研究的限制之一。第二，平滑後的影片會因為穩定器修正整張 Frame 的位置，讓同一時刻輸出影片與原始影片在影像位置上有失真或不一致的感覺，這是所有二維 DIS 共同的限制。

本文提出的 DIS 能以延遲一張 Frame 為代價，即時處理影像抖動問題，且除了要設定高斯函數參數大小之外，沒有多餘參數需要設定。此外，本研究所提出的移動平均容納五個全域運動向量的設計，主要是針對震盪半週期在 1/6 秒(五張 Frame / 30fps)以內的抖動成分，若要增加抖動濾除範圍，則可用低於 30 fps 擷取頻率進行本演算法，再配合曲線近似法，來補回 30 fps 下該擷取頻率中未進行處理而缺失的補償值資訊，這樣我們的設計足以濾除各種頻率下的抖動成分。因

此，本文的主要貢獻在於設計一套預測型 DIS 系統，以延遲一張 Frame 為代價，以速度和加速度資訊分析自我運動求得預期移動變化，並有機會實現即時穩定的功能。

6.2 未來展望

未來探討的方向上，本研究可進行的方向可分成五類：第一，不穩定性參數的重新定義，本研究以速度與加速度為主要的穩定性依據，這部分可運用其他演算法算得新的 Cost 值來重新評估各個 Section 的重要性；第二，自我運動向量估計演算法的公式變化，相對於本研究以固定 Section 來進行演算，尚可用動態變化 Section 長度的想法來進行改進或創新，以期望能應對不同週期長度或振幅大小的抖動情況。第三，本文運用固定的 Region 大小來進行 GMV 估計演算法，且尚未特別針對移動物體發生的情境做處理，未來可以在 Region 上動態調整大小或個數來進行 LMV 的估計和移動物體的偵測。這其中會要用演算法考量所要假設的最大移動物體大小和如何適時動態改變 Region 的大小，以期能偵測出正確的 GMV 資訊。第四，嘗試設計新的穩定器檢驗公式也是一重要探討對象。由穩定後的絕對或相對位置關係，探討數值之間的連續性、誤差或標準差狀況並設計出新穎的公式來評斷過去的演算法。最後，針對穩定器處理層級進行設計，本法為二維穩定器，僅考量水平分量與垂直分量，如何上升穩定器的處理維度以應對更複雜的影像情況也是重要課題之一。所謂的高維數位影像穩定器，主要是針對高速拍攝或是以 CCD 感測器相機進行拍攝造成 Frame 擷取畫面有形變現象等問題做處理。其中，如何定義相機往復運動情況的 GMV 資訊以及穩定影像對於形變影像的插補方法仍有待研究。數位影像穩定器的出現是重視資源節省和數位化時代的產物，相信未來可有更多有創意的想法繼續拓展這個領域。

参 考 文 献

- [1] K. Sato, S. Ishizuka, A. Nikami, and M. Sato, "Control techniques for optical image stabilizing system," *IEEE Trans. Consum. Electron.*, vol. 39, no. 3, pp. 461-466, Aug. 1993.
- [2] Canon, "What is vari-angle prism image stabilizer (VAP-IS)." Available online:
<http://www.canon.com/bctv/faq/vari.html>.
- [3] Canon, "What is optical shift image stabilizer." Available online:
<http://www.canon.com/bctv/faq/optis.html>.
- [4] Canon, "Optical image stabilizer." Available online:
http://www.usa.canon.com/cusa/consumer/standard_display/Lens_Advantage_IS.
- [5] Nikon, "Vibration reduction (VR) technology." Available online:
<http://imaging.nikon.com/history/scenes/16/>.
- [6] Panasonic, "Panasonic Mega O.I.S." Available online:
<https://panasonic.ca/english/audiovideo/camerascamcorders/digitalstill/megaOIS.asp>.
- [7] H. J. Chang, P. J. Kim, D. S. Song, J. Y. Choi, "Optical image stabilizing system using multirate fuzzy PID controller for mobile device camera," *IEEE Trans. Consum. Electron.*, vol. 55, no. 2, pp. 303-311, May 2009.
- [8] H. R. Pourreza, M. Rahmati, F. Behazin, "An electronic digital image stabilizer based on stationary wavelet transform (SWT)," in *Proc. 2003 Int. Conf. on Image Process.*, vol. 2, Sep. 2003, pp. II-383-386.
- [9] S. Ertürk, "Image sequence stabilisation: motion vector integration (MVI) versus frame position smoothing (FPS)," in *Proc. 2nd Int. Symp. on ISPA*, 2001, pp. 266-271.
- [10] K. Ioannidis and I. T. Andreadis, "A digital image stabilization method based on the

- Hilbert–Huang transform,” *IEEE Trans. Instrum. Meas.*, vol. 61, no. 9, pp. 2446-2457, Sep. 2012.
- [11] S. C. Hsu, S. F. Liang, K. W. Fan, and C. T. Lin, “A robust in-car digital image stabilization technique,” *IEEE Trans. Syst. Man Cybern.*, vol. 37, no. 2, pp. 234-247, Mar. 2007.
- [12] Y. G. Ryu and M. J. Chung, “Robust online digital image stabilization based on point-feature trajectory without accumulative global motion estimation,” *IEEE Signal Process. Lett.*, vol. 19, no. 4, pp. 223-226, 2012.
- [13] C. Wang, J. H. Kim, K. Y. Byun, J. Ni, and S. J. Ko, “Robust digital image stabilization using the Kalman filter,” *IEEE Trans. Consum. Electron.*, vol. 55, no. 1, pp. 6-14, 2009.
- [14] A. A. Amanatiadis and I. Andreadis, “Digital image stabilization by independent component analysis,” *IEEE Trans. Instrum. Meas.*, vol. 59, no. 7, July 2010.
- [15] C. T. Lin, C. T. Hong, and C. T. Yang, “Real-time digital image stabilization system using modified proportional integrated controller,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 427-431, Mar. 2009.
- [16] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.Y. Shum, “Full-frame video stabilization with motion inpainting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1150 – 1163, July 2006.
- [17] S. J. Ko, S. H. Lee, and K. H. Lee, “Digital image stabilizing algorithms based on bit-plane matching,” *IEEE Trans. Consum. Electron.*, vol. 44, no. 3, pp. 617-622, Aug. 1998.
- [18] S. J. Ko, S. H. Lee, S. W. Jeon, and E. S. Kang, “Fast digital image stabilizer based on gray-coded bit-plane matching,” *IEEE Trans. Consum. Electron.*, vol. 45, no. 3, pp. 598-603, Aug. 1999.
- [19] Q. Zeng, H. Wang, and X. Ma, “On-line digital image stabilization for translational and rotational motion,” *Int. Conf. on IASP*, 21-23 Oct. 2011, pp. 266-270.

- [20] C. H. Cheung and L. M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1168-1177, Dec. 2002.
- [21] L. Xu and X. Lin, "Digital image stabilization based on circular block matching," *IEEE Trans. Consum. Electron.*, vol. 52, no. 2, pp. 566-574, May 2006.
- [22] J. Shi, C. Tomasi, "Good Features to Track," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 1994, pp. 593-600.
- [23] G. Welch and G. Bishop, "An introduction to the Kalman filter," in *SIGGARCH2001 Course Note*, 2001 : Univ. North Carolina.
- [24] M. Bertalmio, G. Sapiro, V. Casellas, and C. Ballester, "Image inpainting," in *Proc. of the Int. Conf. on Computer Graphics and Interactive Techniques*, New Orleans, LA, USA, July 2000, pp.417-424.
- [25] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Washington, D.C., USA, June/July 2004, vol. 1, pp. 120-127.
- [26] S. C. Hse, S.F. Liang, and C. T. Lin, "A robust digital image stabilization technique based on inverse triangle method and background detection," *IEEE Trans. Consum. Electron.*, vol. 51, no. 2, pp. 335-345, May 2005.
- [27] W. C. Kao, S. H. Chen, P. Y. Hsiao, "Real-time image stabilization for digital video cameras," in *Proc. of the IEEE Asia Pacific Conf. on Circuits and Systems*, Singapore, Dec. 2006, pp. 1651-1654.
- [28] Rastislav Lukac, *Single-sensor imaging methods and applications for digital cameras*, CRC Press Taylor & Francis Group, Boca Raton 2009.

附 錄

A. 演算法延遲代價現象的解釋

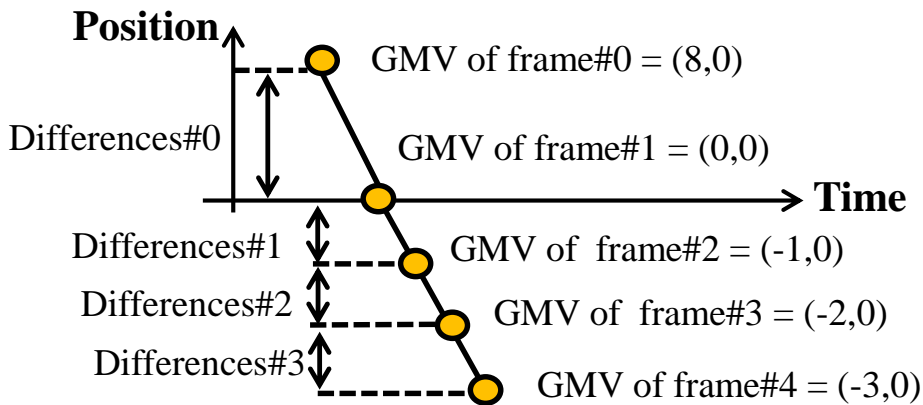
本演算法為什麼會有要延遲一張 Frame 的代價？這是因為我們演算法判斷自我運動是運用 Section panning velocity(SPV)和 Section representative offset(SRO)兩資訊。前者，代表該時間點 Section 內 GMV 資訊的概略走勢；後者，代表該時間點 Section 內 GMV 資訊的代表性位置。這個兩個參數的特性在 Section 中的 GMV 位置資訊是相異方向的鋸齒狀抖動時確實能有濾波功能，但是當目前時刻 Section 暫存器中的 GMV 位置趨勢為同方向同斜率位置變化時，這種濾波功能會造成對位置趨勢的延遲預估。

此外，本文決定自我運動的方法是考量所有 Section 的 SPV 和當下 Section 的 SRO 資訊。假若前者 SPV(速度)和後者 SRO(位置)相互以相加方式組合產生的新位置結果比 Section 暫存器中當下時間點的 GMV 位置還要慢，那便有延遲現象出現。因為 SRO 本身參考處會較偏重於 Section 中心(第三個 GMV 位置)，且速度最高的首末 GMV 距離也僅出現在完全平滑線上，所以延遲現象是幾乎是必然會發生。以下將證明為什麼本文所設計的五張 Frame 的 Section 能僅延遲一張 Frame 而不是延遲至二或三張 Frame 的原因。

本文的證明方法為(1)證出在同方向位置變化下當 SRO 在最左端 (算出的 SRO 會最遠離當下時間點 GMV 位置資訊)的情況，配合上 SPV 斜率算得的位置是否能超過 Section 暫存器正中間 GMV 的位置。若能超過，表示至少 GMV(t-2) 資訊。(2)接著，再證明在同方向位置變化下 SRO 在最右端 (算出的 SRO 會最鄰近於當下時間點 GMV 位置資訊)的情況下，SPV 配合 SRO 的組合位置是否慢於最新 Frame 的 GMV 位置。若會較慢，則表示即使此時為最接近當下時刻的 SRO

情況，在計算上的新位置仍舊慢於當下時刻 Section 最新的 GMV 位置。只要證明這兩者成立，表示在 t 時刻 Section 的 SPV 和 SRO 組合出的位置大都限制在「快於 GMV(t-2)」和「慢於 GMV(t)」位置，則證實五張 Frame 為長度的 Section 設計用於 DIS 功能僅需延遲一張 Frame，也為最好的設計。

當區域代表性偏移最接近當下時間點的情況



圖A-1 區段代表性偏移最接近當下時間點之全域運動向量資訊

如圖所示，假設 Section 當下五張 Frame 的 GMV 資訊所算出的 SRO 是最靠左端的狀況：Section = { Frame#0, Frame#1, Frame#2, Frame#3, Frame#4 }，SRO 最靠近 Frame#0。這樣的條件我們可以設計兩項參數(1)Frame#0 至 Frame#1 的斜率和 Frame#1 至 Frame#4 有相異但同號斜率，本例設皆為負號，(2)此時 Frame#0 至 Frame#1 的運動向量差距 Differences#0 為了滿足最靠近 Frame#0 的條件，設為最大搜尋上限的八個像素單位，其餘兩兩 Frame 之間 Differences 設為最小搜尋下限一個像素單位。若假設 Frame#1 的 GMV 位置恰在零且每 Frame 之間的時間軸上為一的情況，此時：

$$\begin{bmatrix} C_i^p & C_i^t \end{bmatrix}^T = [4 \quad 0.5]^T$$

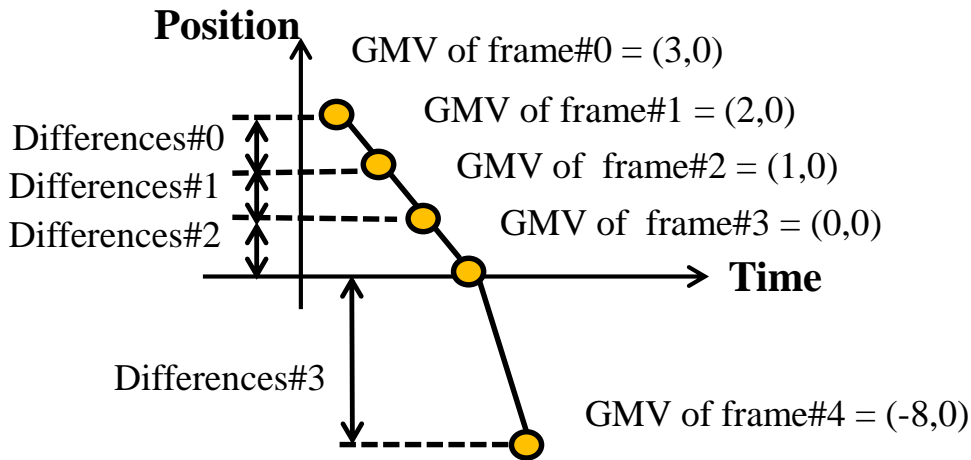
$$[C_r^P \quad C_r^T]^T = [-1.5 \quad 2.5]^T$$

$$SRO = \text{average}(4, -1.5) = 1.25$$

$$SPV = (-1.5 - 4) / (2.5 - 0.5) = -2.75$$

SPV 配合 SRO 得值為 $1.25 - 2.75 = -1.55$ ，結果大約快於 Frame#2 的 GMV 的位置資訊(-1)。由於本次假設為 SRO 最左極端狀況，則上一時刻的 Section 資訊必定因為與本時刻 Section 的部分資訊有接續情況而不可能同為 SRO 最左極端狀況，故速度的線性組合只會更遠離 Frame#2 而接近 Frame#3 的位置。得證，在同方向位置變化下當 SRO 在最左端，則與 SPV 算得的位置能快於 GMV(t-2)。

當區域代表性偏移最遠離當下時間點的情況



圖A-2 區段代表性偏移最遠離當下時間點之全域運動向量資訊

再來證明算出的 SRO 最遠離當下時間點的情況，上圖[圖 A-2]所示，假設 Section 暫存器當下五張 Frame 的 GMV 資訊算出的 SRO 是在最右端狀況：Section Register{ Frame#0, Frame#1, Frame#2, Frame#3, Frame#4}，SRO 最靠近 Frame#4。這樣的條件我們可以設計兩項參數(1)Frame#0 至 Frame#3 的斜率和 Frame#3 至

Frame#4 有相異但同號斜率，本例設皆為負號，(2)此時 Frame#3 至 Frame#4 的運動向量差距 Differences#3 為了滿足最靠近 Frame#4 的條件，設為最大搜尋上限的八個像素單位，其餘兩兩 Frame 之間 Differences 設為最小搜尋下限一個像素單位。若假設 Frame#3 的 GMV 位置恰在 0 且每張 Frame 之間的時間軸上為一的情況，此時：

$$[C_l^P \quad C_l^T]^T = [1.5 \quad 1.5]^T$$

$$[C_r^P \quad C_r^T]^T = [-4 \quad 3.5]^T$$

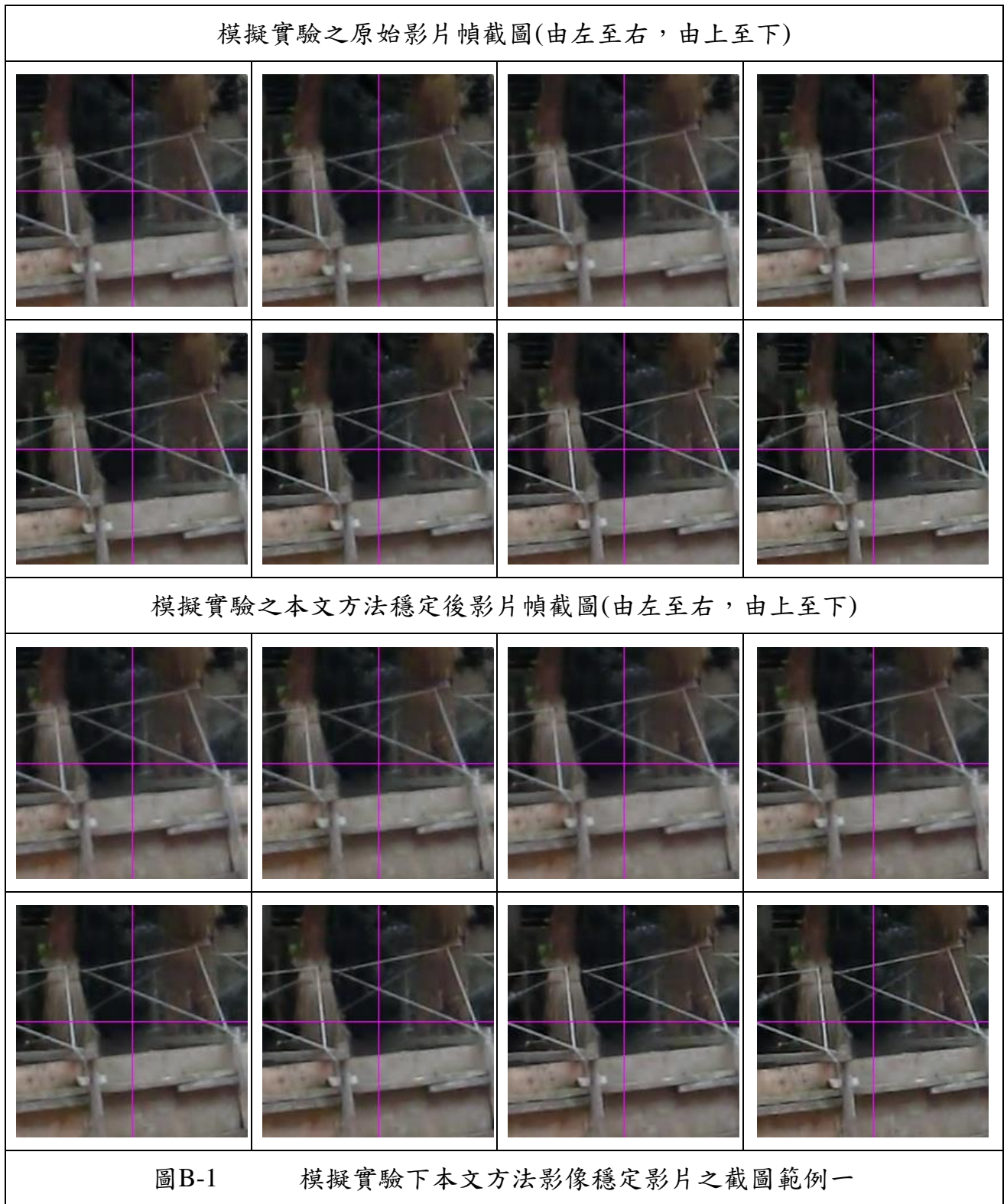
$$SRO = \text{average}(1.5, -4) = -1.25$$

$$SPV = (-4 - 1.5) / (3.5 - 1.5) = -2.75$$

SPV 配合 SRO 得值為 $-1.25 - 2.75 = -4$ ，結果位置界於 Frame#3 至 Frame#4 之間，慢於 Frame#4 的 GMV 位置資訊(-8)。由於本次假設為 SRO 最右極端狀況，則上一時刻的 Section 資訊必定因為與本時刻 Section 的部分資訊有接續情況而不可能同為 SRO 最右極端狀況，故速度的線性組合只會使結果位置越遠離 Frame#4 而接近 Frame#3 的位置。得證，在同方向位置變化下當 SRO 在最右端，與 SPV 算得的位置會慢於 GMV(t)。

綜上所證，若當計算 Section(t)時推測的自我運動位置範圍，在最極端的狀況就是在快於 Frame(t-2)或慢於 Frame(t)，造成在 Frame(t-1)附近較能近似準確值。本演算法基於平衡考量，選擇延遲一張 Frame 時間點的方式來輸出 CMV 值。此外，當 Section 的暫存器長度提升超過五張 Frame，尚需另外計算延遲的範圍和面對延遲範圍過大的問題，這也是為什麼我們選擇 Section 長度為五張 Frame 的原因。

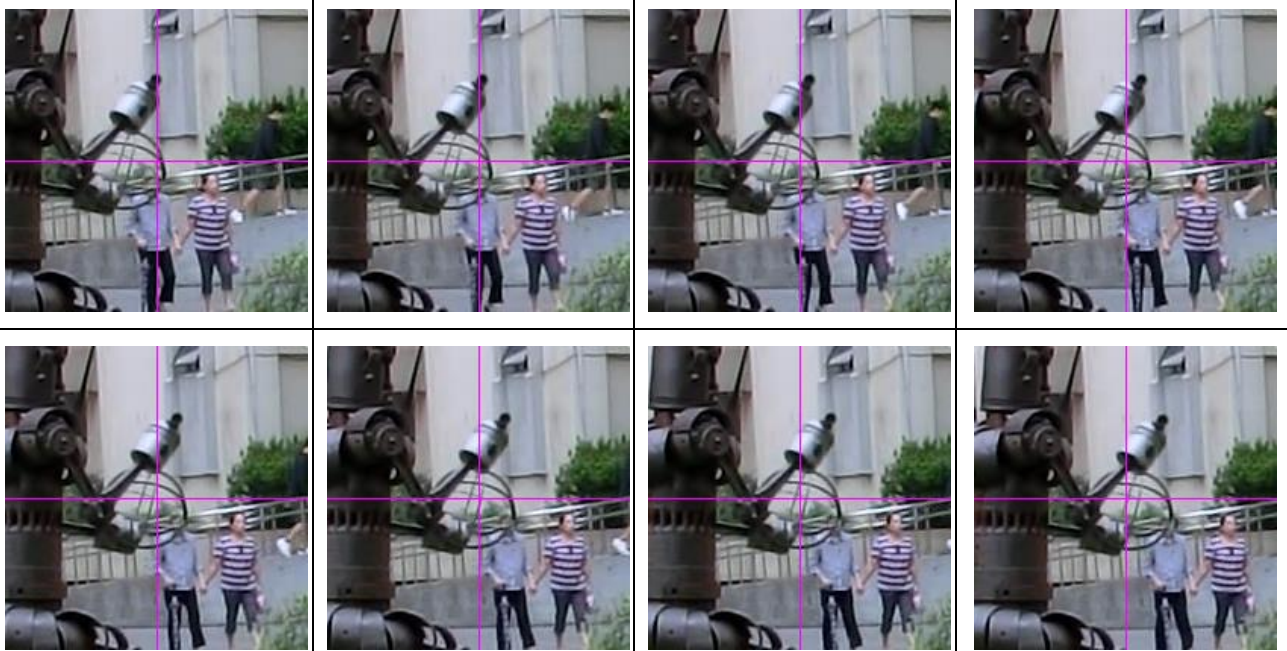
B. 輸出影片截圖



模擬實驗之原始影片幀截圖(由左至右，由上至下)

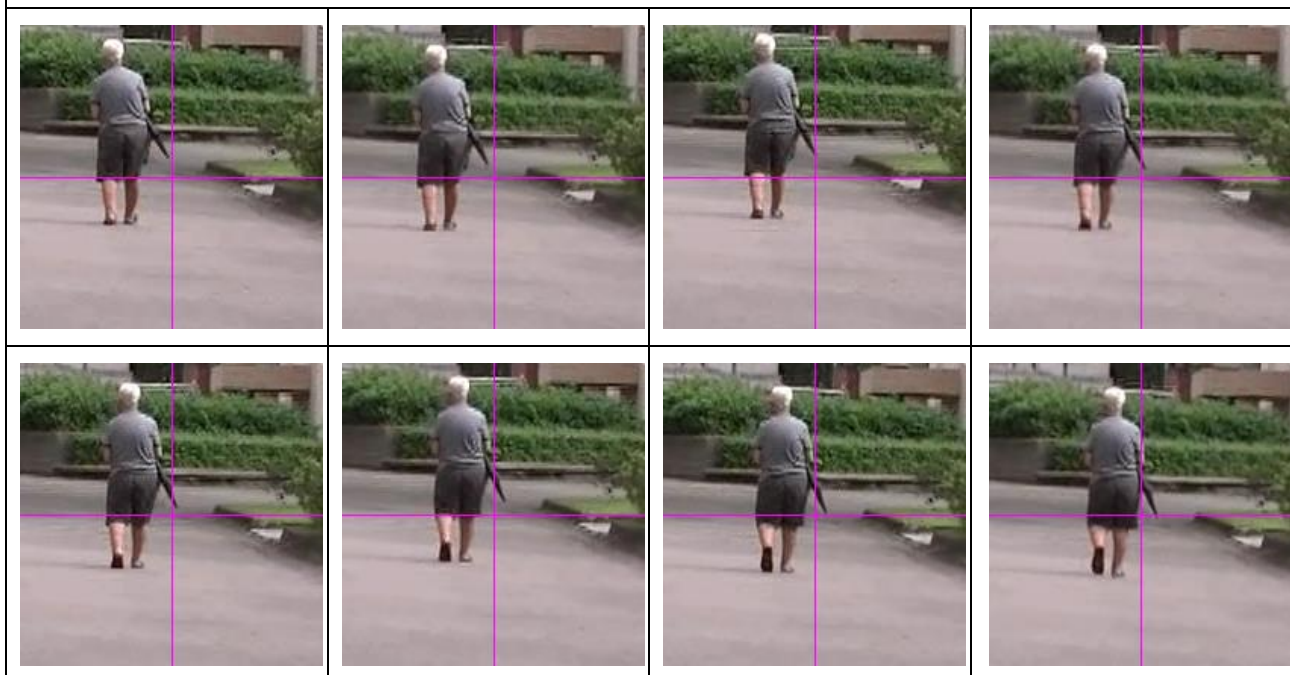


模擬實驗之本文方法穩定後影片幀截圖(由左至右，由上至下)

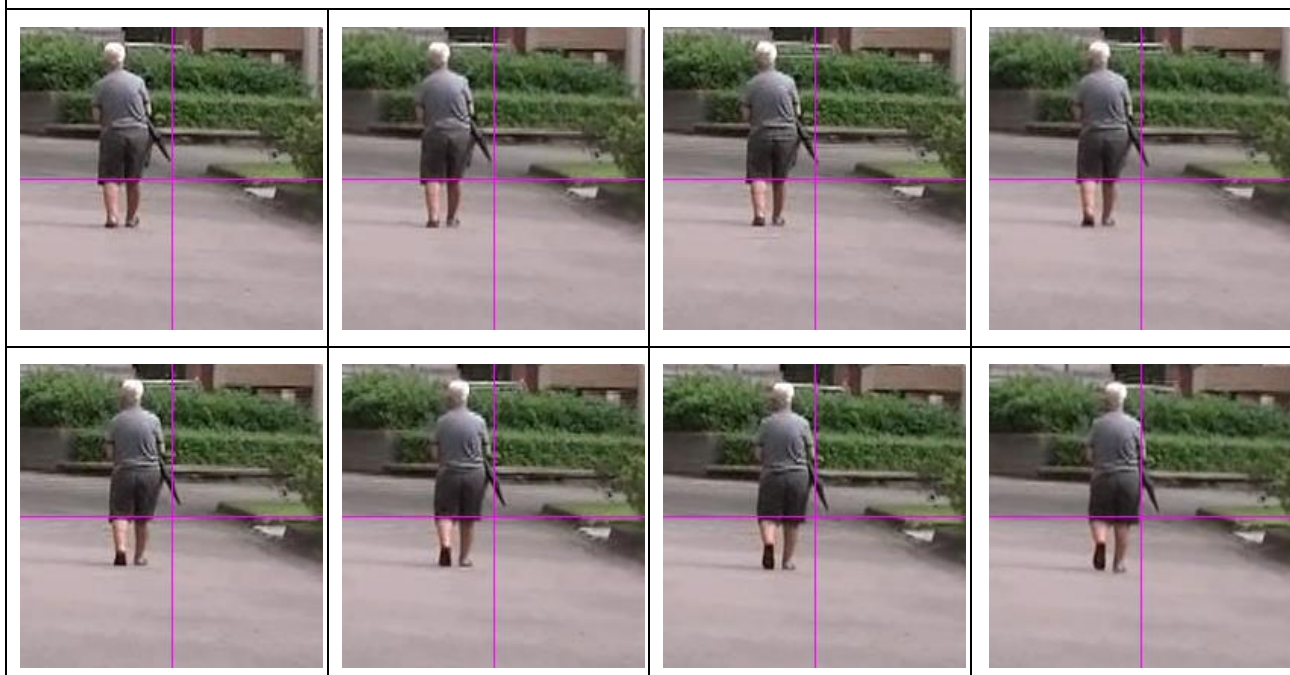


圖B-2 模擬實驗下本文方法影像穩定影片之截圖範例二

實景拍攝之原始影片幀截圖(由左至右，由上至下)



實景拍攝之本文方法穩定後影片幀截圖(由左至右，由上至下)



圖B-3 實景拍攝下本文方法影像穩定影片之截圖範例