

一個使用於移動估算的適應性非對稱樣本搜尋演算法

梁廷宇

國立高雄應用科技大學電機工程系
lty@mail.ee.kuas.edu.tw

郭柏賢

國立高雄應用科技大學電機工程系
kps@hpds.ee.kuas.edu.tw

摘要

在這篇論文中，我們為移動估算設計一個新的快速方塊匹配演算法，稱為適應性非對稱樣本搜尋演算法。這個搜尋演算法的最大特色在於其所使用的搜尋樣本是非對稱式的，而且有預測方向及動態調整搜尋範圍的特性。藉由這些特性，這個演算法可以有效地減少方塊搜尋的點數，進而提高移動估算的速度。在本篇論文，我們已將此演算法應用在MPEG-4的視訊編碼中，並與其它知名的搜尋演算法做效能上的比較。實驗結果顯示我們所提出的演算法比鑽石搜尋演算法的速度改善率提高了56.448%至65.805%，而與適應性十字形樣本搜尋演算法比起來，對於動態的畫面亦有將近20%的速度改善率。另一方面，亦同時維持很高的PSNR值。

關鍵字詞：移動估算、快速方塊匹配演算法、非對稱式、MPEG-4、PSNR。

1. 簡介

對於許多移動補償(Motion compensation)為基礎的視訊編碼技術標準像ISO MPEG-1[1]、MPEG-2[3]、MPEG-4[2]以及ITU-T H.261[5]、H.262[3]、H.263[17]來說，方塊匹配移動估算(Block-matching motion estimation, BMME)是一個很重要的處理步驟，它可以有效地幫助移動補償編碼去除連續畫面間冗餘的資料。對於方塊移動估算，最直接的方法就是完全搜尋(Full search, FS)，它能夠保證在搜尋範圍中找到最準確、誤差最小的匹配方塊。但是完全搜尋有一個最大的缺點，就是必須花相當冗長的時間做搜尋，因此近十年來許多著名的快速方塊匹配演算法被提出來。這些演算法都有一個通用的準則就是在最少失真的情況下盡可能地提高其搜尋效能。例如：最早被廣泛使用的三步搜尋法(Three step search, TSS)[18]是以一種逐步縮小樣本範圍間距4-2-1的搜尋方式，所需的搜尋點數為常數共25個點。相較於FS，它大量減少搜尋點數。但是TSS的搜尋方向是由外而內，第一回合的搜尋範圍過大而且稀疏，因此可能將搜尋方向帶領到一個錯誤的位置去。新三步搜尋法(New Three step search, NTSS)[14]採用一個以中央集中(center-biased)的搜尋範圍取代之前TSS均勻分佈(distribution uniformly)搜尋的方法，提高了搜尋速度及準確度。此後中央集中特性的觀念被廣泛應用於之後的各種搜尋法。例如：四步搜尋演算法(Four

step search, 4SS)[9]亦採用中央集中特性，並縮小第一回合搜尋的樣本大小，因此改善了NTSS最差的情況。鑽石搜尋法(Diamond search, DS)[15]有著較緊密且重疊性高的樣本結構，使得搜尋的效率或者是準確度都優於之前的各種演算法，因此成為最普遍採用的演算法。其它像二維對數搜尋演算法(2-D logarithmic search, LOGS)[8]、十字搜尋演算法(Cross search, CS)[11]以及十字鑽石搜尋法(Cross-diamond search, CDS)[4]等，都是採用比較固定式的搜尋樣本，亦即其搜尋樣本依照特定的規則來使用或替換。

相較前面的方法，適應性十字形樣本搜尋演算法(Adaptive rood pattern search, ARPS)[19]是採用非固定式的樣本大小來提高第一回合搜尋範圍的命中率。由於畫面中屬於相同物件的方塊必有很高的空間相關性，所以ARPS以目前待搜尋方塊之周圍方塊做為預測的支援區域(Region of support, ROS)。此支援區域的所有移動向量的平均移動向量，稱為目標預測移動向量(Prediction of the Target MV)。對於支援區域的選擇通常有Type A, B, C, D四種(如圖1所示)，而ARPS所用的支援向量為Type D。其原因是Type D的效能和其他三種差不多，而且較容易實現。除此之外，ARPS還將目標預測移動向量的長度做為搜尋樣本大小的參考。經由實驗證明，ARPS的效能比DS提高了2倍左右。另外，由於畫面中的方塊有很高機率為靜止不動或移動的幅度微小，因此ARPS還使用了零移動預先判決(Zero-motion prejudgment, ZMP)的技術，對於方塊誤差和(Sum of absolute difference, SAD)小於一個臨界值T的目標區塊，就視為其移動向量(i.e. $MV=[0, 0]$)為零，如此便可以有效節省搜尋的點數。對於靜態的連續畫面，整個ARPS-ZMP可更進一步地提高移動估算的速度，而且亦同時維持相近的PSNR值。

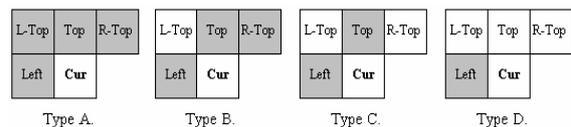


圖1 支援區域(Region of support, ROS)

根據前面的敘述得知，在移動估算裡採用預測目標移動向量與大小可調整的搜尋樣本，可以減少相當多的搜尋點數。然而，我們的實驗結果顯示搜尋樣本的點數仍有進一步減少的空間。如表1顯示支援區域Type D所提供目標預測移動向量與真

正移動向量相差在正負90度內的機率高達79.457%至98.202%。此結果意味著我們不需採用完整的十字型搜尋樣本，因此我們提出了一個以非對稱性樣本為基礎的演算法稱為AAPS (Adaptively Asymmetric Pattern Search)。經由我們的實驗顯示在較動態的畫面中，AAPS較APRS提高了10~20%移動估算的速度。本篇論文在第2章中將詳細介紹本論文所提出的這個AAPS演算法。第3章將討論AAPS整體效能評估結果。最後第4章則為本篇論文之結論與我們未來的工作。

$$P = \begin{cases} X \text{ 軸}(y=0) & \begin{cases} 1 \cup 4 & , X > 0 \\ 2 \cup 3 & , X < 0 \end{cases} \\ Y \text{ 軸}(x=0) & \begin{cases} 1 \cup 2 & , Y > 0 \\ 3 \cup 4 & , Y < 0 \end{cases} \\ p & , p=1,2,3,4 \end{cases} \quad (1)$$

表1 搜尋區域之測預區域命中率，經由full search 對所有標準測試樣本實驗結果

測試樣本 (Image sequence)	預測區域命中率(%)
	±90 度
Coastguard	0.92717
Container	0.93743
Garden	0.91820
Football	0.79457
News	0.94421
Salesman	0.98202

2. 適應性非對稱樣本搜尋演算法(AAPS)

2.1 搜尋樣本

AAPS演算法主要使用二種搜尋樣本：(1)適應性非對稱樣本(Adaptively asymmetric pattern, AAP)，(2)動態適應性十字形樣本(Dynamically adaptive rood pattern, DARP)。下面是這兩個搜尋樣本的描述。

(1) 適應性非對稱樣本

AAP 也是一個具有適應性的搜尋樣本。比較不同的是它為一非對稱性搜尋樣本，而且藉由選擇特定的搜尋方向來減少搜尋的點數。如前所述，支援區域Type D所提供目標預測移動向量與真正移動向量相差在正負90度內的機率很高，根據此特性我們可以使用一個非對稱十字形搜尋樣本來取代完整十字形搜尋樣本，因此AAP根據目標預測移動向量的象限來選擇所搜尋的方向。其中所選擇的方向依相位分別有第1、2、3、4象限，以及第1∪2、2∪3、3∪4、4∪1象限共八種，如圖2。此八種相位根據目標預測移動向量的方向來決定，當目標移動向量預測落在第1至4象限其一時，AAP則為一個90度角搜尋範圍(圖3(a))。如果目標預測移動向量落在水平之X軸或是垂直之Y軸上，則AAP為一個180度角搜尋範圍(圖3(b))。AAP選擇相位P取決如下式(1)：

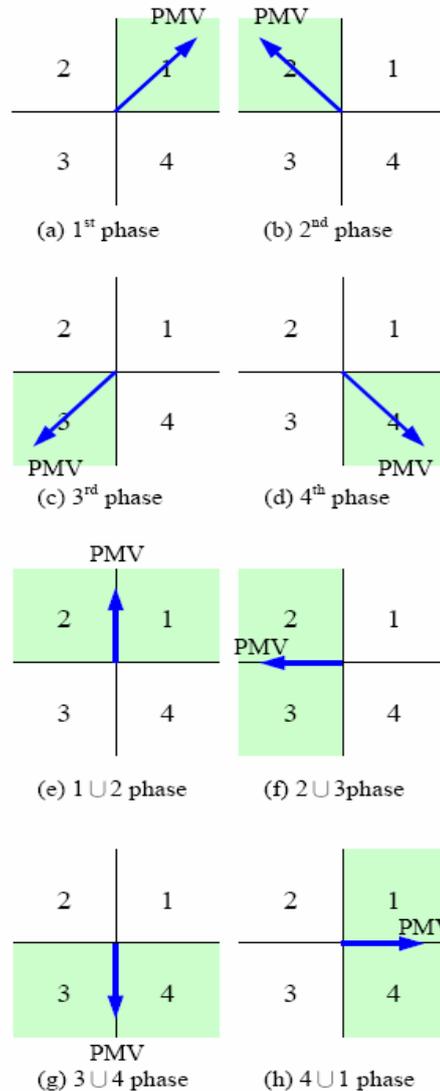


圖2 AAP使用的八種搜尋區域

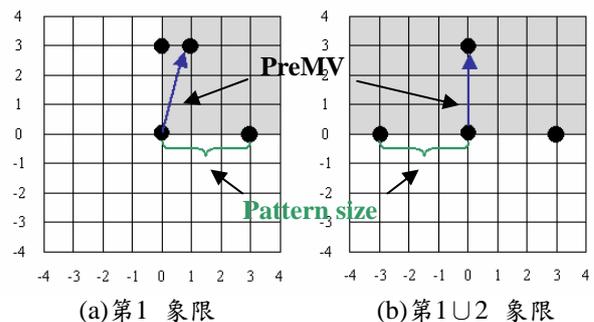


圖3 適應性非對稱樣本(AAP)

AAP的大小則是根據目標預測向量座標的最大值來定義AAP範圍長度 Γ ：

$$\Gamma = \text{Max} \left\{ |MV_{\text{Predicted}}(x)|, |MV_{\text{Predicted}}(y)| \right\} \quad (2)$$

決定樣本的搜尋方向及大小後，便可建構出包含原目標預測移動向量總共3個搜尋點做為AAP之搜尋樣本。其中，假若目標預測移動向量剛好為0，則此時只需搜尋中心點1個。圖2顯示出AAP與ARP比較可節省1至2個搜尋點數。

(2) 動態適應性十字形搜尋樣本

一般情況下經過AAP第一次搜尋後，搜尋的區域被帶到一個目標物件移動向量可能發生的區域，為了能在本地區域找出一個最精確、誤差最小的方塊，一個範圍較小且緊密的修正搜尋樣本是必要的。相反地，面對畫面中物件鄰界的方塊或者變動不定之物件，搜尋範圍有限的AAP有可能將搜尋的區域帶到「錯誤」位置。此時如果使用一範圍較小且大小固定之樣本，必會花費相當多的搜尋點數才能找到最匹配的巨集方塊。為了將搜尋的區域快速地帶回到正確的目標物件移動位置，必須利用一個範圍大且稀疏的搜尋樣本來修正。為了因應這兩種不同的考量，AAPS在第一回合搜尋後，便改用了可動態變動大小的樣本DARP來繼續以後的搜尋。DARP為一包含中心共5個點之對稱性十字型樣本，其動態大小適應共分二種，(a)範圍大小 $\Gamma=1$ ，(b)範圍大小 $\Gamma=2$ ，如圖4。DARP適應範圍大小 Γ 依預測修正係數(Predicted fixed coefficient)改變，如式(3)。其中預測修正係數($Fixed_Coef_{\text{Predicted}}$)為支援區域Type D之DARP搜尋回合數。

$$\Gamma = \begin{cases} 2, & Fixed_Coef_{\text{Predicted}} > 0 \\ 1, & otherwise \end{cases} \quad (3)$$

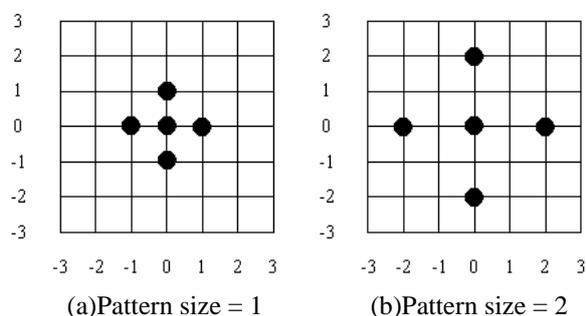


圖4 動態適應性十字形搜尋樣本

此外由於在畫面編碼處理通常採列方塊優先的循序計算方式(由左至右、由上至下)，因此在畫面中最左邊方塊並沒有參考向量。根據真實世界的畫面，巨集方塊的移動向量落於水平或垂直方向的機率高於其它的位置[16]，而且經過FS對許多測

試畫面所得實驗結果顯示，有52.76%~98.7%的移動向量落在與中心距離為2的像素範圍內[15]。有鑑於此，AAPS在畫面最左邊的巨集方塊使用一間距為2的DARP。這將有利於快速地搜尋到水平或垂直位置的向量。另外，位於畫面邊界的巨集方塊大多屬於背景物件，它們通常有較小的移動距離，所以亦不適宜採用較大的搜尋樣本。

2.2 演算法

整個AAPS 步驟如下：

步驟1：首先判斷待搜尋巨集方塊是否為畫面之左邊界巨方塊。若為左邊界巨集方塊則使用範圍大小 $\Gamma=2$ 且含中心點共5個搜尋點的DARP，找其最小方塊誤差(Minimum Block Distortion, MBD)位置後，跳至步驟3；若非為畫面之左邊界巨集方塊則做步驟2。

步驟2：根據目標預測移動向量，並參考式(1)及式(2)來建構出所使用的AAP，並在此AAP 中找出其MBD位置，然後做步驟3。

步驟3：此步驟使用一個DARP，依預測修正係數($Fixed_Coef_{\text{Predicted}}$)決定其範圍大小(如為畫面左邊界方塊則預測修正係數為0)，計算其MBD。若其MBD 落於中心點，則此對應向量為最佳匹配方塊位置之移動向量，並記錄DARP 搜尋回合次數做為下一個巨集方塊參考的預測修正係數；不然就將預測修正係數減1 並重覆此步驟。

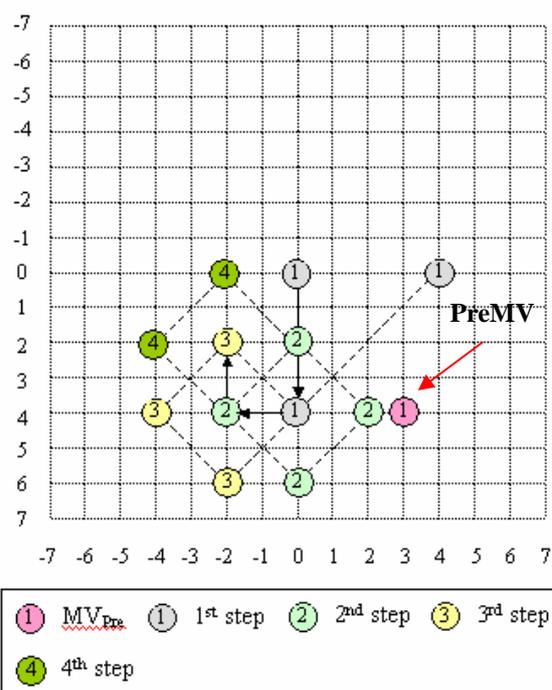


圖5 AAPS搜尋路徑圖例

圖5為一個AAPS搜尋路徑簡單圖例，其搜尋視窗大小為15×15，搜尋方塊為非畫面中左邊界方塊。其中1st step、2nd step, ..., nth step代表移動估計每一回合所搜尋的位置，而MV_{Pre}表示支援區塊Type D中目標預測移動向量的位置(3, 4)，預測修正係數(Fixed_Coef_{Predicted})為5，最後找到之移動向量為(-2, 2)。這個例子顯示一個方向預測錯誤發生的情形，原目標預測移動向量位於第四象限，但最後所搜尋的移動向量落於第三象限，由此可看出一個動態的DARP是必須的。

3. 實驗結果

在本文實驗結果中，我們將AAPS應用於MPEG-4編碼器上，所採用的所有標準測試樣本如下表2，共6種。畫面編碼結構採取IPPP...IPPP，每一張I畫面間隔距離為30張。巨集方塊大小為16×16，方塊搜尋範圍(Search range)為16(i.e.繞著中心[-16, 16]像素的搜尋視窗大小)。匹配準則在這裡採用平均絕對值誤差(MAD)準則，因為它有比平均平方根誤差(MSE)準則有更小的計算複雜度[10]。對於快速方塊匹配移動估算，其計算複雜度則取決於每一個移動向量估計所需花費的平均搜尋點數。

在此對本文所提出的AAPS分別與TSS、DS，以及ARPS...等快速方塊匹配演算法比較。實驗結果分別記錄在表3至表5中。其中表3為各種演算法之方塊平均搜尋點數，表4則為搜尋效能增益之比較，表5為重建後畫面之PSNR值。由表3顯示出方塊平均搜尋點數AAPS < ARPS < DS < 3SS < FS，本論文所提出的AAPS與其它的演算法比起有最少的平均搜尋點數。其中AAPS與效能較好的ARPS比

較，整體效能改善率(SIR)提高了1.964%至19.170%不等。這是由於在較靜態的畫面中，預測目標移動向量大多為靜止或移動微小，此時對於AAP與ARP這二種搜尋樣本之搜尋點數皆僅為原點1點，因此AAPS選擇方向搜尋的AAP其優越性比起搜尋每個方向的ARP並沒有太大的差異。相反地，在動態的畫面中相鄰的巨集區塊經常有類似的移動方向，因此AAP往往能在每個方塊中節省1個搜尋點，而且一個可變的DARP亦能減少搜尋的回合數進而有效提高搜尋效能。與DS比較起來其SIR亦提高了56.448%至65.805%不等。另外與TSS比較，其SIR更有提高61.214%至81.821%的高改善率。由效能增益來看，AAPS與FS比較，有著105.60至189.10大幅的改善，並同時維持少於0.08dB 誤差的重建品質。

圖6至圖11分別為本實驗中六種測試樣本之平均搜尋點數曲線圖。從圖中可看出本文提出的AAPS與ARPS比較，對於一些比較動態的畫面如"Coastguard"、"Garden"、"Football"等有著較明顯的效能改善率。

表2 標準測試畫面樣本

測試樣本 (Image sequence)	解析度 (Frame size)	張數 (Length)
Coastguard	352×288	300
Container	352×288	300
Garden	352×240	115
Football	352×240	125
News	176×144	300
Salesman	176×144	449

表3 方塊平均搜尋點數

Image sequence	FS	3SS	DS	ARPS	AAPS
Coastguard	1024.00	25.00	20.34	10.13	8.39
Container	1024.00	25.00	13.29	5.53	5.42
Garden	1024.00	25.00	21.40	10.31	9.05
Football	1024.00	25.00	22.26	12.00	9.70
News	1024.00	25.00	13.47	6.00	5.70
Salesman	1024.00	25.00	13.09	5.54	5.43

表4 搜尋效能增益

Image sequence	FS	3SS	DS	ARPS	AAPS
Coastguard	1.00	40.96	50.35	101.08	122.05
Container	1.00	40.96	77.05	185.18	189.10
Garden	1.00	40.96	47.84	99.28	113.21
Football	1.00	40.96	45.99	85.36	105.60
News	1.00	40.96	76.03	170.59	179.76
Salesman	1.00	40.96	78.21	184.79	188.60

表5 畫面平均PSNR(dB)

Image sequence	FS	3SS	DS	ARPS	AAPS
Coastguard	31.31	31.32	31.33	31.30	31.30
Container	33.94	33.92	33.92	33.92	33.92
Garden	28.96	29.04	28.96	28.96	28.95
Football	29.56	29.66	29.66	29.64	29.64
News	33.28	33.29	33.28	33.27	33.27
Salesman	33.07	33.07	33.07	33.07	33.07

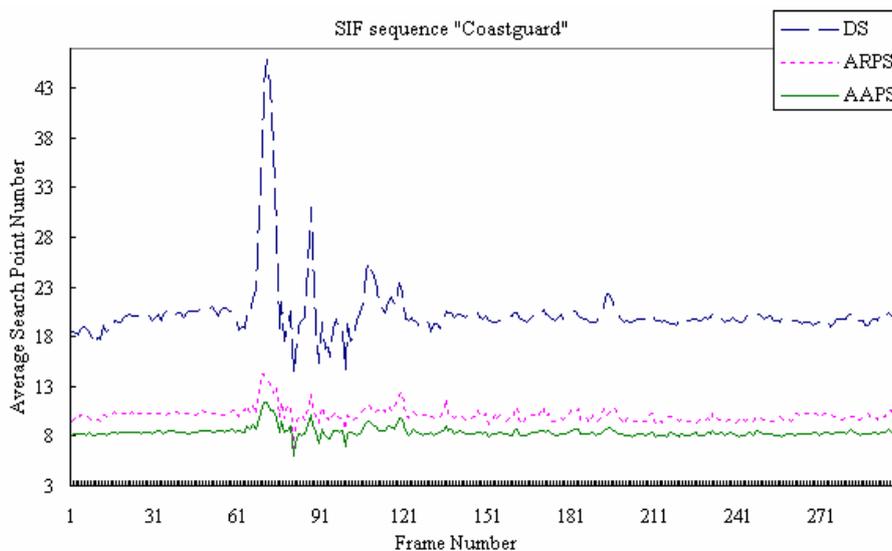


圖6 方塊平均搜尋點數(“Coastguard”)

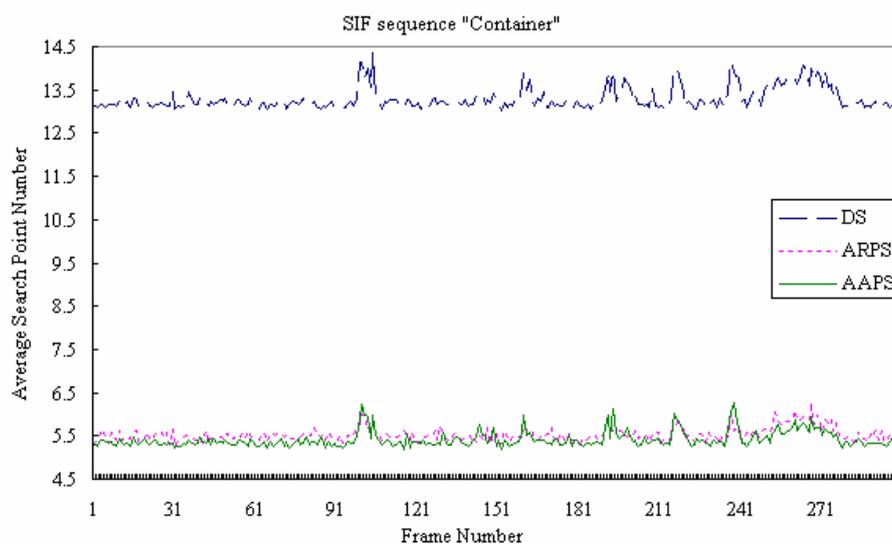


圖7 方塊平均搜尋點數(“Container”)



圖8 方塊平均搜尋點數("Garden")



圖9 方塊平均搜尋點數("Football")

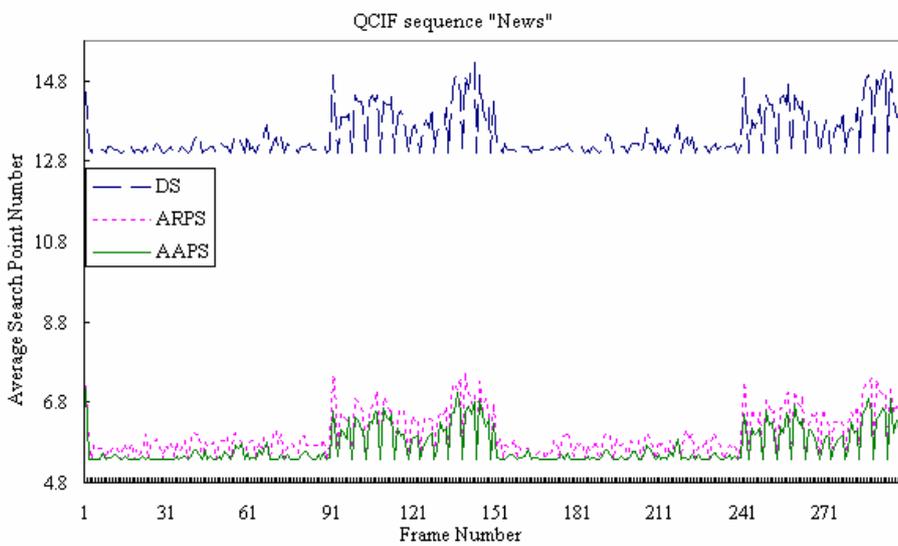


圖10 方塊平均搜尋點數("News")

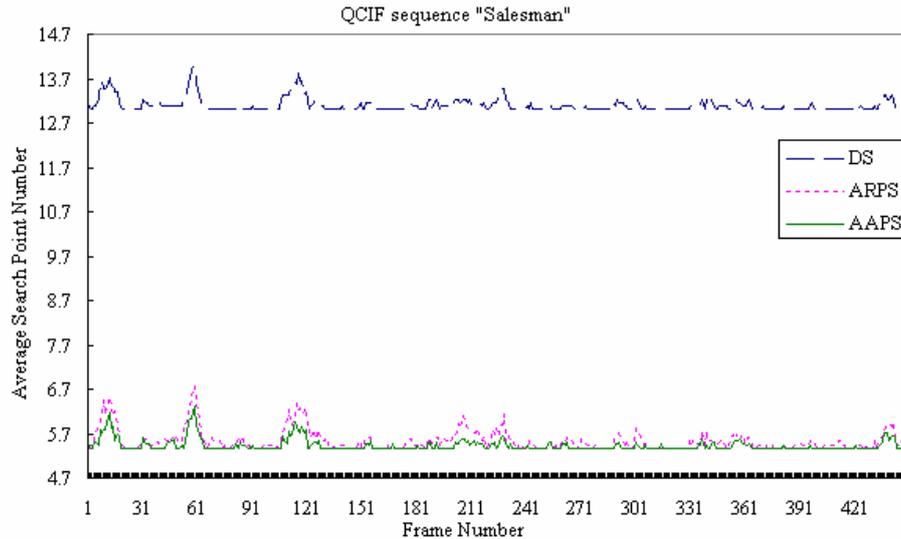


圖11 方塊平均搜尋點數(“Salesman”)

4. 結論與未來工作

在這篇論文中，我們提出了一個新的快速方塊匹配演算法，稱為適應性非對稱樣本搜尋演算法(AAPS)。AAPS利用目標預測移動向量與目標移動向量相差正負90度內的機率極高的特性，以選擇特定搜尋方向的方式來減少搜尋的點數。在這種選擇特定搜尋方向的方式來說，亦有預測錯誤或正確二種情況。若目標移動向量落於選擇的相位上，對於畫面中快速移動物件的方塊，DARP可依修正係數來快速達到目標位置以加快搜尋效能或做精密的搜尋。對於預測錯誤的情況發生時，使用DARP亦能夠改善多餘的搜尋點數。由此可知道對於AAPS，DARP有其使用的重要性。目前我們已經將AAPS應用在MPEG-4視訊編碼中，實驗的數據顯示證明，相較於TSS、DS、ARPS，AAPS有較少的方塊平均搜尋點數，而其所重建的影像亦有很高的PSNR值，所以AAPS是一個相當有效率的快速方塊匹配演算法。

近年來一些視訊的應用如影像電話、訪客對講系統，以及隨選視訊服務(Video on demand, VOD)如隨選影片(Movies on demand, MOD)[13]、遠距教學服務(Distance learning)[6]、視訊會議服務(Video conference)[7]等等都必須經由壓縮的處理來減少資料的儲存以及網路傳輸量。對於MPEG-4這種以移動補償為基礎的視訊編碼技術標準，在方塊匹配移動估算運用BMAs雖能加快編碼效能，但是當視訊編碼對象為一現場直播的高解析度(如640×480或1024×768)畫面時，以目前的PC或工作站的機器對高解析度的影像進行壓縮，仍不足以達到即時性的編碼。解決的方法可利用平行運算的技術。我們未來將在計算叢集上設計與實現一個即時性的MPEG-4平行編碼器[12][20]。

參考文獻

- [1] “Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s: Video”, ISO/IEC 11 172-2 (MPEG-1 Video), 1993.
- [2] “Information Technology—Generic Coding of Audio-Visual Objects” Part 2: Visual, ISO/IEC 14 496-2 (MPEG-4 Video), 1999.
- [3] “Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video”, ISO/IEC 13 818-2-ITU-T Rec. H.262(MPEG-2 Video), 1995.
- [4] C. H. Cheung, L. M. Po, “A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation”, *Transactions on Circuits System for Video Technology*, Vol. 12, No.12, Dec. 2002. pp.1168-1177.
- [5] CCITT SGXV, “Description of reference model 8 (RM8)”, Document 525, Working Party XV/4, Specialists Group on Coding for Visual Telephony, June 1989.
- [6] G. Bressan, R.M. Silveira and W.V. Ruggiero, “A Framework for Distance Learning Employing Video on Web Environment”, *International Conference on Technology and Distance Education*, Fort Lauderdale -Florida, June/1999.
- [7] ITU-T Recommendation T.120. Introduced to Audiographics and Audiovisual Conferencing.1996.
- [8] J.R. Jain and A.K. Jain, “Displacement measurement and its application in inter-frame image. Coding”, *IEEE Transactions on Communication*, Vol. COM-29, pp.1799-1806, Dec. 1981.
- [9] L. M. Po, W. C. Ma, “A novel four-step search algorithm for fast block motion estimation”,

- IEEE Transactions. Circuits System for Video Technology*, Vol. 6, pp. 313–317, June 1996.
- [10] Lai-Man Po, Chok-Kwan Cheung, “A new center-biased orthogonal search algorithm for fast block motion estimation”, *TENCON '96. Proceedings. 1996 IEEE TENCON. Digital Signal Processing Applications*, Vol. 2, 26-29 Nov. 1996 pp. 874 – 877 Vol. 2
- [11] M. Ghanbari, “The cross-search algorithm for motion estimation”, *IEEE Transactions on Communications*, 38 (7), 950–953, Aug. 1990.
- [12] Miguel Ribeiro, Oliver Sinnen, Leonel Sousa, ”MPEG-4 Natural Video Parallel Implementation on a Cluster”, *Image and Video Coding, RECPAD2002*, Page:3
- [13] P. Venkat Rangan, Harrick M. Vin, Srinivas Ramanathan. “Designing an On-Demand Multimedia Service.”, *IEEE Communications*, Vol 30, No. 7, July 1992.
- [14] R. Li, B. Zeng, and M. L. Liou, “A new three-step search algorithm for block motion estimation”, *IEEE Transaction. Circuits System for Video Technology*, Vol. 4, pp. 438–442, Aug. 1994.
- [15] S. Zhu and K.-K. Ma, “A new diamond search algorithm for fast block-matching motion estimation,” in *Proc. 1997 Int. Conf. Information, Communications and Signal Processing (ICICS)*, Vol. 1, Sept.9–12, 1997, pp. 292–296.
- [16] S. Zhu, K.-K. Ma, “A new star search algorithm for fast block-matching motion estimation”, in *Proc. Workshop on Very Low Bitrate Coding (VLBV)*, Oct. 8–9, 1998, pp. 173–176.
- [17] Standardization Sector of ITU, Video Coding for Low Bitrate Communication, ITU-T Rec. H.263, Mar. 1996.
- [18] T. KOGA, K. IINUMA, A. HIRANO, Y. IJIMA, T. ISHIGURO, “Motion-compensated interframe coding for video conferencing”, in *Proc. NTC 81*, pp. C9.6.1-9.6.5, New Orleans, LA, Nov. Dec. 1981.
- [19] Yao Nie, Kai-Kuang Ma, “Adaptive rood pattern search for fast block-matching motion estimation”, *IEEE Transactions on Image Processing*, Vol. 11, Issue 12, Dec. 2002 pp. 1442 - 1449
- [20] Yong He, Ahmad, T., Liou, M.L., ”MPEG-4 based interactive video using parallel processing”, *Proceedings. IEEE International Conference on Parallel Processing 1998*, 10-14 Aug. 1998, pp.329 – 336