國 立 中 正 大 學

電 機 工 程 研 究 所

視 訊 處 理 作 業 一

實作動作估計並使用Rate Distortion Optimization計算分割區塊成本與EBMA比較

研 究 生：傅冠豪

學號：612415078

任課教授：江瑞秋 博士

中 華 民 國 一 百 一 十 二 年 十 一 月 五 日

1. **簡介**
   1. **動作估計(Motion Estimation)**

動作估計 (Motion estimation)在Video Processing System中扮演著關鍵的角色。它不僅用於前處理已進行三維結構的動作估計以預測前後兩幀的差異，還廣泛用於影片壓縮、影片取樣率轉換等等。

在當前的課程，專注於探討動作估計於影片壓縮中的應用。這涉及到通過比較前後兩幀來來估計物體的運動得到移動向量 ( Motion Vector , MV )，利用這些向量進行動作補償生成預測影像(Predicted frame)，以此可以將移動向量和預測誤差的位元總數和最小化，提升視訊編碼效率。

在本文中，我們專注於動作估計基於區塊進行動作估測 ( Block-based motion estimation )，並引入了Rate Distortion Optimization(RDO)，目標是保證視訊品質的前提下，盡量減少壓縮所需要的位元數，通過優化位元率(Bit-Rate)和失真(Distortion)之間的權衡找到最佳的編碼策略，比起只單純利用Displacement Frame Difference(DFD)所計算的誤差，本文的方法更具有優勢，因為能更好地平衡視訊品質和壓縮效率，以提供更高圖像品質的壓縮同時保持更低的位元率。

* 1. **區塊匹配演算法 (Block Matching Algorithm , BMA )**

所有的動作估測方法中，最為人所知的就是基於區塊進行的BMA，此演算法將一個frame完整分割了數個block，並將區塊遍歷周圍的區塊，找出最小誤差的區塊位置，只會產生一組各自區塊獨立的Motion Vector，用於預測整個區塊。

現今有非常多的演算法被開發，如最初的全局搜索區塊匹配法 ( Exhaustive block match algorithm , EBMA)就是本文所使用的Baseline，雖然可以透過窮舉法取得精確的移動向量MV，但花費的計算量大，且也沒有考慮到壓縮所需要的位元數，因此本文透過將Block分割利用RDO比較兩者之間所需要的壓縮成本(cost)，將MV以及Residual一併考慮，雖增加了計算量卻提升了Predicted Frame的品質。

* 1. **光流 ( Optical flow )(TO Do list)**

在二維的影像中判斷動作的方法，主要基於時間序列影像中同一物體只是位置改變而像素點強度不變的概念。如 (1.3.1)式的位移幀差 ( Displaced Frame Difference Equation , EDFD )等式，就是根據這個概念推導出的。

*E*DFD(**a**) = (1.3.1)

而光流等式 (Optical Flow Equation, EOF )，也是基於相同概念的假設 (1.3.2)式進行推倒，將其進行泰勒展開得到 (1.3.3)式，比較(1.3.2)式與(1.3.3)式後能得到 (1.3.4)式，並將表示式用梯度與向量替換表示成 (1.3.5)式。最後，將物體匹配問題利用上述公式轉成最小化問題 (1.3.6)式。

(1.3.2)

(1.3.3)

(1.3.4)

(1.3.5)

(1.3.6)

* 1. **RDO (TO Do list)**

1. **方法**

本次作業1將會利用光流等式 (Optical Flow Equation, EOF )實作全局搜索區塊匹配法 ( Exhaustive block match algorithm , EBMA)。作業2將會實作二維對數搜索法 ( Two-dimensional logarithmic Search Method, 2D-log)、三步搜索法 ( Three-Step Search Method , TSS)、新三步搜索法 ( New Three-Step Search Method , NTSS)和四步搜索法 ( Four-Step Search , 4SS)並與EBMA進行比較。

1. **EBMA**

全局搜索區塊匹配法 ( Exhaustive block match algorithm , EBMA)，為最基礎的BMA，首先，演算法會在目前幀 ( Current frame ) 選取一個目前要匹配的區塊 ( Current block )，將current block對目標幀 ( Target frame )搜索區域 ( Search region )內的每個區塊進行搜索，計算光流等式 (Optical Flow Equation, EOF )，找出最小值並進行匹配，得到移動向量 ( Motion Vector , MV )。如圖2.1範例所示。

EBMA能找到較精準的MV，但是會花費鉅大的計算量。

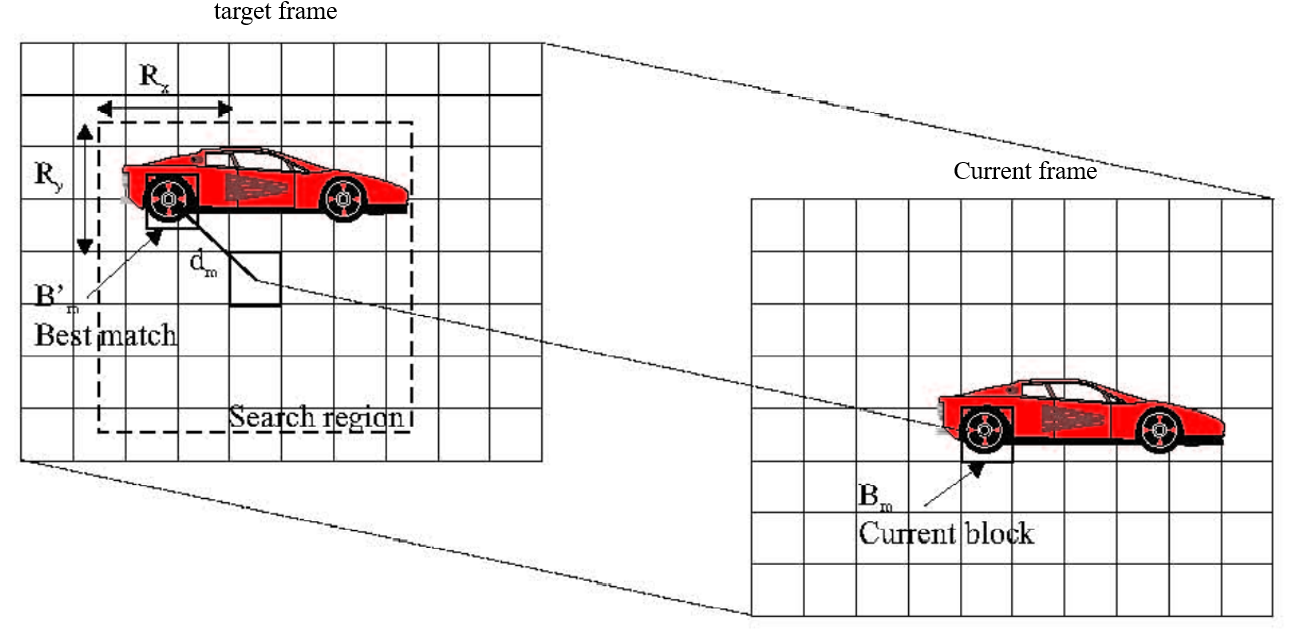


圖2.1

1. **2D-log**

二維對數搜索法 ( Two-dimensional logarithmic Search Method, 2D-log)是一種快速搜索演算法，其名字源自初始搜索步長的設定(stepsize )。初始stepsize由(2.2.1)式和(2.2.2)式得出，其中R為搜索大小，n為搜索步長。一開始會搜索中心和距離中心搜索步長的上下左右五個點，尋找最小的區塊失真測量(block distortion measure , BDM)，若是BDM最小值的搜索點在中心或是搜索邊界，則step size會減半，沒有則會繼續同樣的模式搜索，當stepsize等於1時，改為搜索中心點周圍的八個點，並選擇最小值的搜索點作為最終的匹配點。搜索範例如圖2.2，此範例搜索範圍R為6，用數字表示每一步的搜索點，用圓圈框出該步的最佳點，並用\*標示最終的匹配點。

(2.2.1)

(2.2.2)

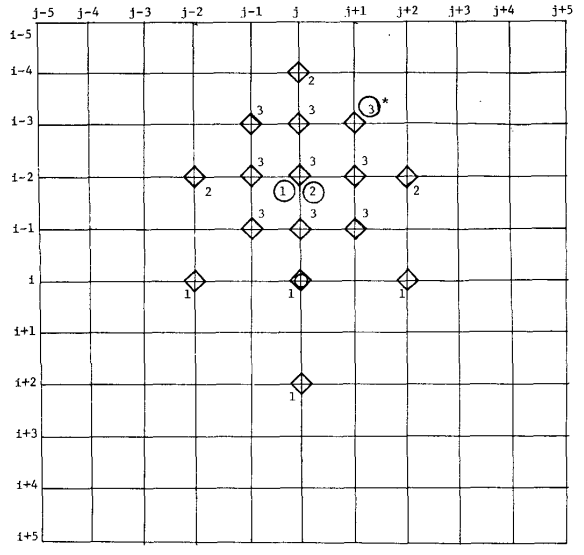
****

圖2.2

1. **TSS**

三步搜索法 ( Three-Step Search Method , TSS)是一種快速搜索演算法，顧名思義，此演算法在搜索範圍R為6或7的時候最多只需三步即可完成搜索。當搜索範圍不同時，會需要不同的步數，實際步數L可由 ( 2.3.1 )式計算，(2.3.2)式可計算最大步數時需要的總搜索點數量N。初始stepsize通常設為大於等於R的一半。

(2.3.1)

(2.3.2)

此演算法在每一步都會搜索九個點，如圖2.3所示，每一步搜索完都會將stepsize減半，直到stepsize等於1，並選取BDM最小值作為匹配點，計算該區塊的MV。

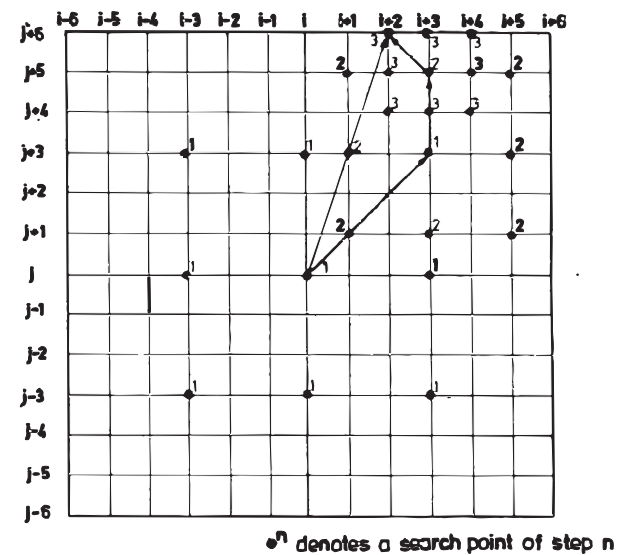


圖2.3

1. **NTSS**

新三步搜索法 ( New Three-Step Search Method , NTSS)是TSS的改良版，雖然TSS因他的簡單性和有效性十分適合用於low bit-rate的影像壓縮，但他在第一步使用固定分配的檢查點，導致其對於小動作的動作估計十分低效。因此NTSS在第一步就同時對中心區塊和外部區塊等17個點進行偵測，如搜索範例圖2.4.1中的黑色圓點與方形點，若是BDM最小值在中心點，則停止搜索，如果最小值在中心點鄰居的八個點內，則進行第二步，根據在角落找五個點或是邊緣找三個點，如圖2.4.1中的三角形點。若是最小值在最外圍的八個點內，則做TSS的步驟。詳細流程可參閱圖2.4.2

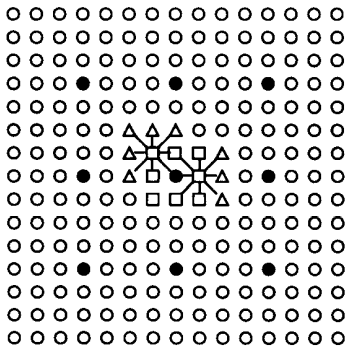


圖2.4.1

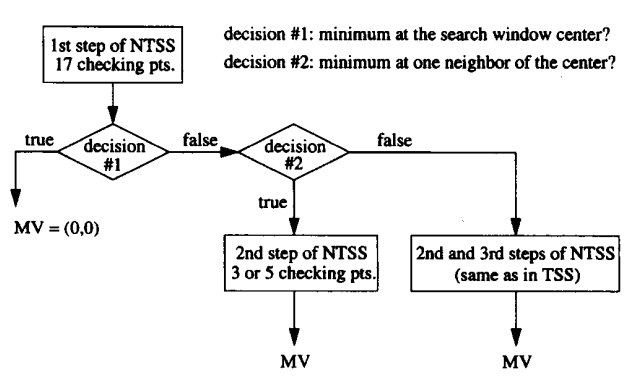


圖2.4.2

1. **4SS**

四步搜索法 ( Four-Step Search , 4SS)是基於現實世界影像序列具有中心偏移移動向量分布的特徵所提出的方法，此演算法的搜索步數為2~4步，總搜索檢查點為17~27個，與NTSS有相似的誤差，但4SS將最壞情況的檢查點從33個減少到27個，並將平均檢查點從21個減少到19個檢查點。

此演算法的搜索範例如圖2.5，初始搜索步長設定為2。首先找尋中心的九個點，若不是BDM最小值不在中心點，則根據角落或是邊緣分別搜索3或5個點，重複兩次，若是找到最小值在中心點，則將步常改為1，尋找最小值當作翠加匹配點。

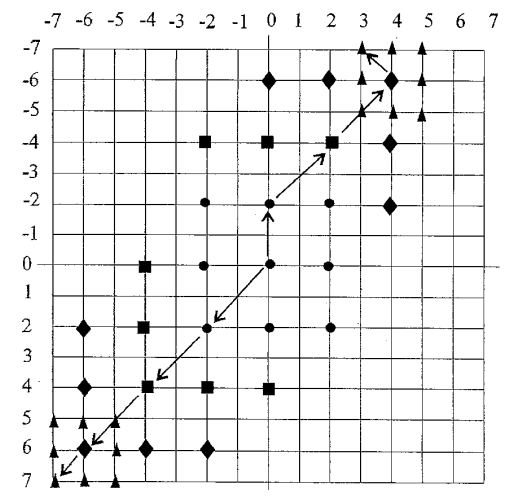


圖2.5

1. **實驗結果與分析**
2. **使用光流等式實作動作估計**

此實驗根據作業規定，將使用(3.1)式做為區塊失真測量(block distortion measure , BDM)，並將block size設為4x4。本次實驗使用第t幀影像當作current frame，t+1幀當作target frame進行預測。

1. **比較不同大小的搜索範圍 ( search range , R )**

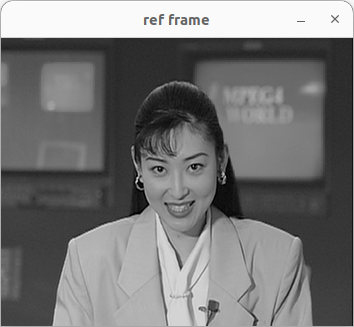
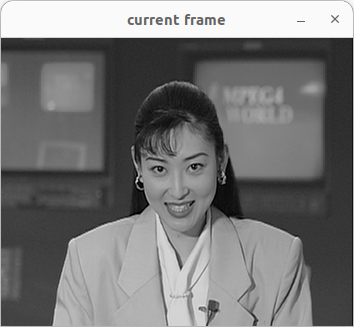


圖3.1.1 current frame 圖3.1.2 target frame

此處R設置為8

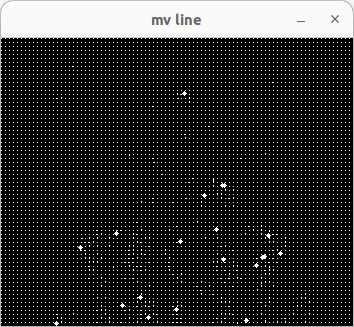
 

圖3.1.3 motion field 圖3.1.4 predicted frame

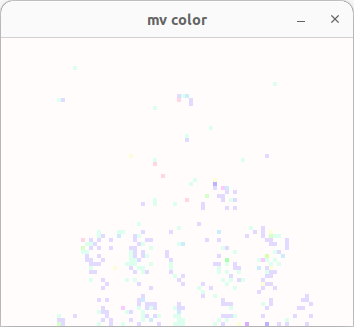
 

圖3.1.5 color space 圖3.1.6 color motion field

此處R設置為16

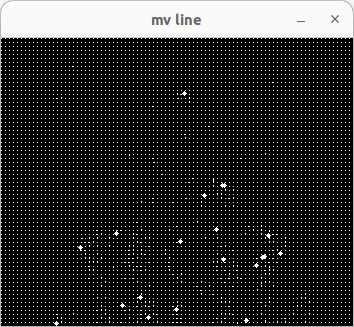
 

圖3.1.5 color space 圖3.1.6 color motion field

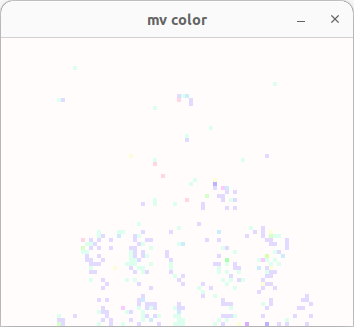
 

圖3.1.5 color space 圖3.1.6 color motion field

此處R設置為32

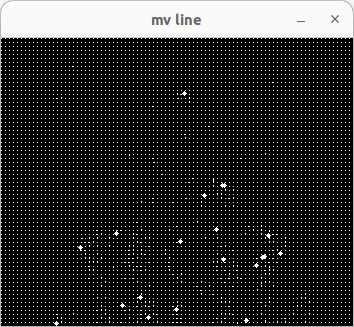
 

圖3.1.3 motion field 圖3.1.4 predicted fram

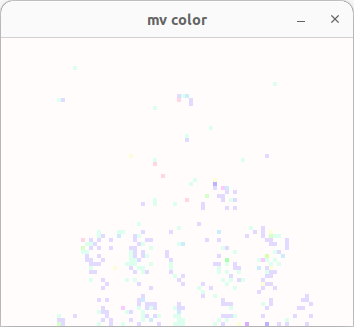
 

圖3.1.5 color space 圖3.1.6 color motion field

1. **比較EOF和EDFD**

比較相同參數下，使用EOF和EDFD對於predicted frame和target frame間的每個像素均方誤差 ( Mean Square Error , MSE )有何影響。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **EOF** | **EDFD** | **EOF** Time(sec/frame) | **EDFD** Time(sec/frame) |
| Akiyo | 0.222 | 0.390 | 0.7777 | 0.5555 |
| Bus | 10.796 | 16.311 | 0.8888 | 0.6666 |
| Container | 0.416 | 0.672 | 0.8888 | 0.6666 |
| stefan | 5.356 | 9.324 | 0.8888 | 0.6666 |

表3.1

由表3.1可看出經**EOF**計算後的MSE較小，但是時間會花費比較久。

1. **實作快速搜索動作估計並與EBMA比較**

將( Exhaustive block match algorithm , EBMA)。二維對數搜索法 ( Two-dimensional logarithmic Search Method, 2D-log)、三步搜索法 ( Three-Step Search Method , TSS)、新三步搜索法 ( New Three-Step Search Method , NTSS)和四步搜索法 ( Four-Step Search , 4SS)進行比較。

1. **比較搜索點數量及搜索步數**

**在R=8的情況**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **搜索演算法** | **搜索點的數量** | | **搜索步數** | |
|  | **最小值** | **最大值** | **最小值** | **最大值** |
| **EBMA** | **255** | **255** | **1** | **1** |
| **2D-log** | **13** | **32** | **2** | **7** |
| **TSS** | **25** | **25** | **3** | **3** |
| **NTSS** | **17** | **33** | **2** | **3** |
| **4SS** | **17** | **27** | **2** | **4** |

表3.2.1

能從表3.2.1觀察到每個演算法在最糟的情況和最好的情況下分別會做多少搜索點。

1. **比較各資料集的MSE**

本實驗是使用第t幀圖預測第t+1幀圖，在計算參考的圖與預測的圖的MSE，故沒有第1幀的資料。且由作業一中了解EOF較能提升精準度，故在此使用EOF。

圖3.2.1

圖3.2.2

圖3.2.3

圖3.2.4

能從圖3.2.1到圖3.2.4中觀察到，在此參數下，雖然每個演算法各有差異，但預測出的都差不多。

1. **相對EBMA，各演算法提升速度**

本實驗各演算法皆使用R=8，block size = 4x4，EOF當作衡量標準，紀錄各演算法在不同資料集的平均每幀花費的時間，以秒為單位。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | EBMA | 2D-log | TSS | NTSS | 4SS |
| Akiyo | 0.7777 | 0.2222 | 0.2222 | 0.2222 | 0.2222 |
| Bus | 0.8888 | 0.2222 | 0.2222 | 0.2222 | 0.2222 |
| Container | 0.8888 | 0.2222 | 0.2222 | 0.2222 | 0.2222 |
| stefan | 0.8888 | 0.2222 | 0.2222 | 0.2222 | 0.2222 |

表3.2.2

可從表3.2.2觀察到搜索範圍小且區塊小的情況，快速搜尋法花費的時間都差不多。

1. **各演算法的預測圖**

本實驗各演算法皆使用R=8，block size = 4x4，EOF當作衡量標準，比較Bus資料集，用不同演算法的預測結果。

圖3.2.5 current frame 圖3.2.6 target frame

EBMA

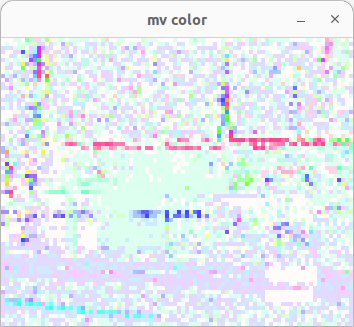
 

圖3.2.7 predicted frame 圖3.2.8 color motion field

2D-log

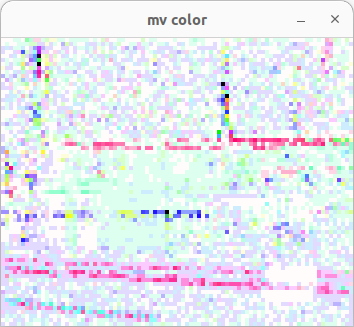
 

圖3.2.9 predicted frame 圖3.2.10 color motion field

TSS

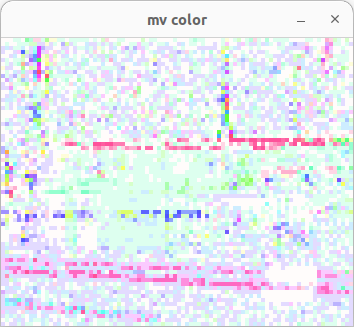
 

圖3.2.11 predicted frame 圖3.2.12 color motion field

NTSS

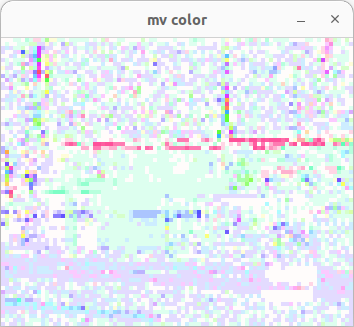
 

圖3.2.13 predicted frame 圖3.2.14 color motion field

4SS

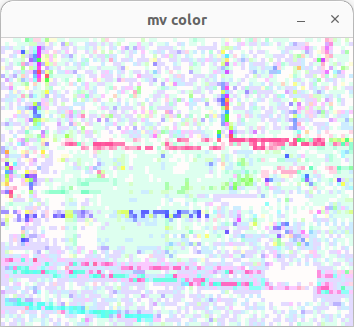
 

圖3.2.15 predicted frame 圖3.2.16 color motion field

能從圖3.2.7到圖3.2.16觀察到雖然會有顏色的深淺，但是大致上移動向量都差不多，每種演算法的效能沒有顯著的差異。

1. **結論**

從本次作業中，學實際將這些演算法寫出來，並在查找資料的過程中更加的了解運作原理。而從我的目前的實作結果可以觀察到，EOF和EDFD都有其優缺點，就看要用要到什麼場合。而本次比較的快速演算法都比EBMA還要快許多並且都有差不多的效能，我想可能是因為搜索範圍和區塊大小的原因，也許下次可以調整這部分參數再深入研究。

**參考文獻**

[1] Y. W. and J. O and Y. Q. Z. VIDEO PROCESSING AND COMMUNICATIONS, Prentice-Hall ,2002.

[2] J. R. Jain and A. K. Jain. Displacement measurement and its application in interframe image coding. *IEEE Trans. Commun.,* Com-29:1799-1808, Dec. 1981.

[3] T. Koga eg al. Motion-compensated interframe coding for video conferencing. In *Proc. Nat. Telecommun. Conf., pages* G5.3.1-G5.3.5, New Orleans, LA, Nov.1981.

[4] R. Li , B. Z. and Ming L. Liou, senior Member. A New Three-Step Search Algorithm for Block Motion Estimation. *IEEE Trans.* Aug 1994

[5] Lai-Man Po and Wing-Chung Ma. A Novel Four-Step Search Algorithm for Fast Block Motion Estimation. *IEEE Trans.* JUNE 1996