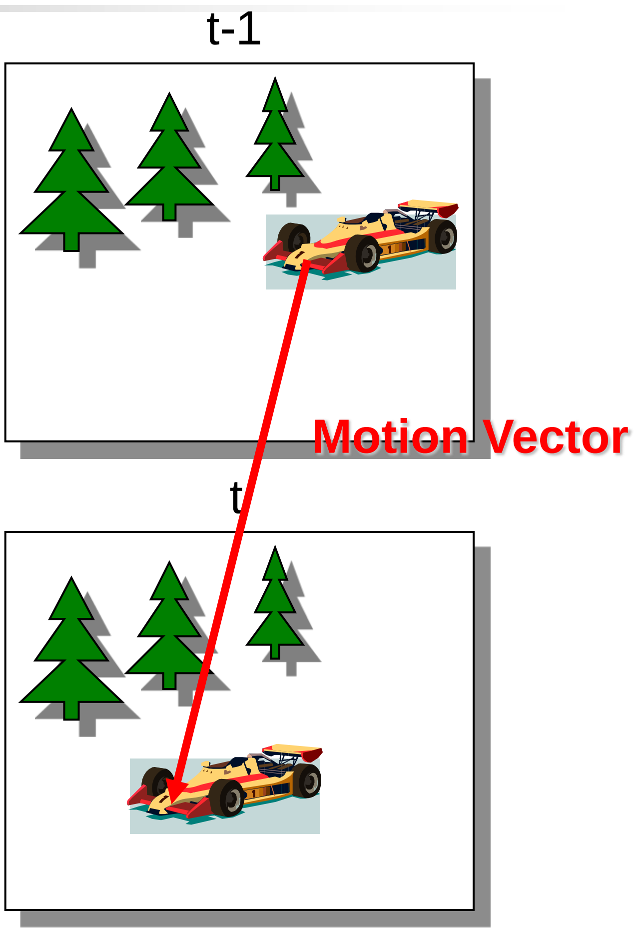
Motion Estimation and Motion Compensation

Motion Estimation (ME)

- To derive the motion information between frames
- Or, find the **corresponding points** between frames



Motion Estimation

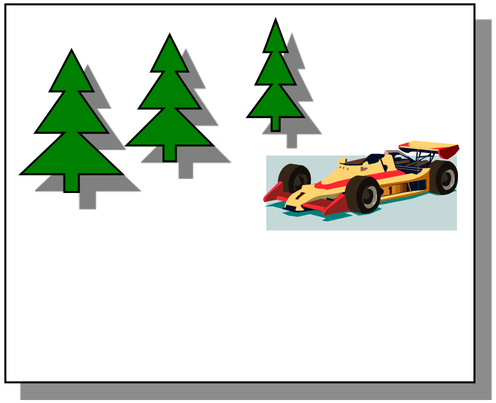
- Objective
  Predict current frame from neighboring frames
- Motion Estimation Algorithm
  - Pixel-based method (Pel-Recursive Algorithm, Optical Flow)
    - Large computation overhead
  - Block-based method
    - Regularity and simplicity
    - Suitable for hardware implementation
  - Object-based method (content-based)
  - Frame-based method
  - …
- Crucial to make possible a high degree of video compression (CR from tens to hundreds)

Motion Compensation (MC)

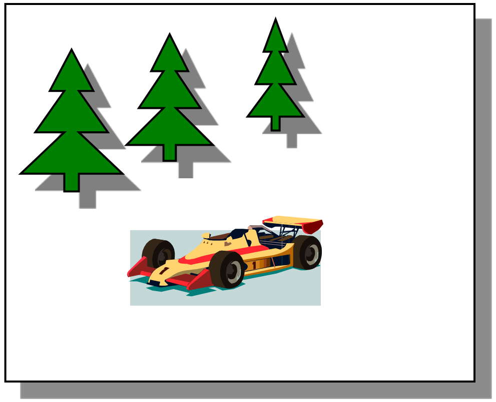
- To compensate the displacement between frames based on motion vectors

- Or, **align or warp** one frame to the other frame

t-1



t



Motion Compensation/Estimation

Original

Reconstructed

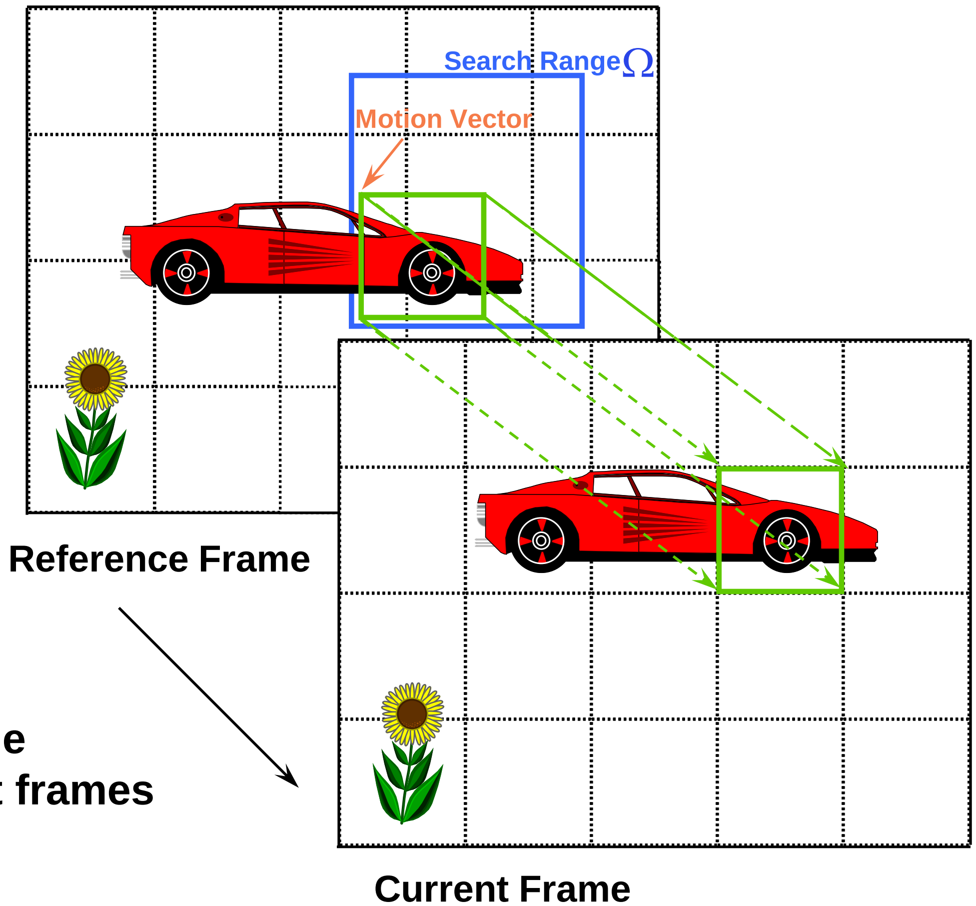
Similar to DPCM

Difference

Motion Estimation

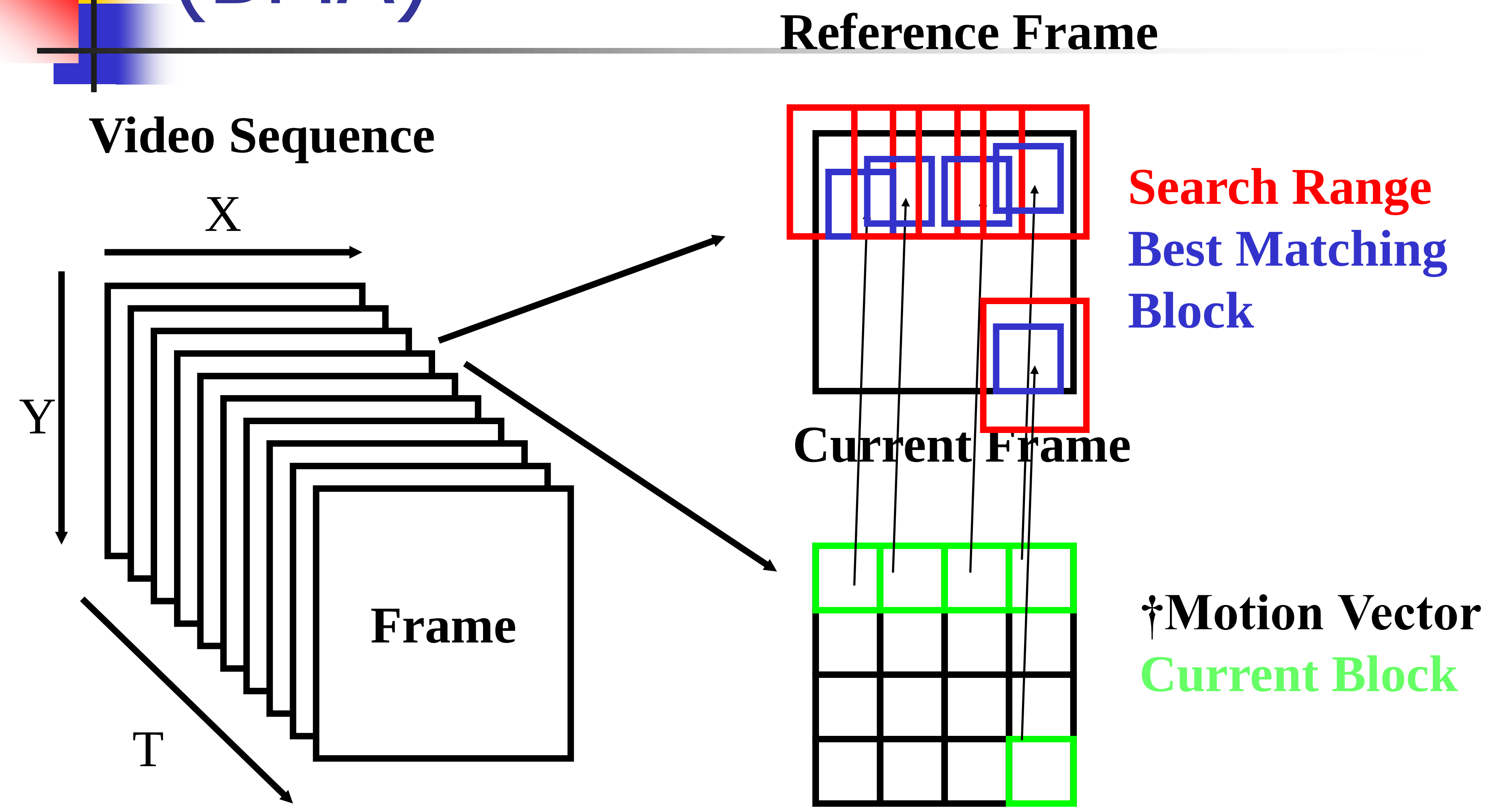
- Pixel-based method
  - Pel-recursive algorithm
  - Optical flow
- Block-based method
  - Full-search block matching algorithm
  - Fast motion estimation
  - More improvements on motion estimation

Block-Matching Motion Estimation



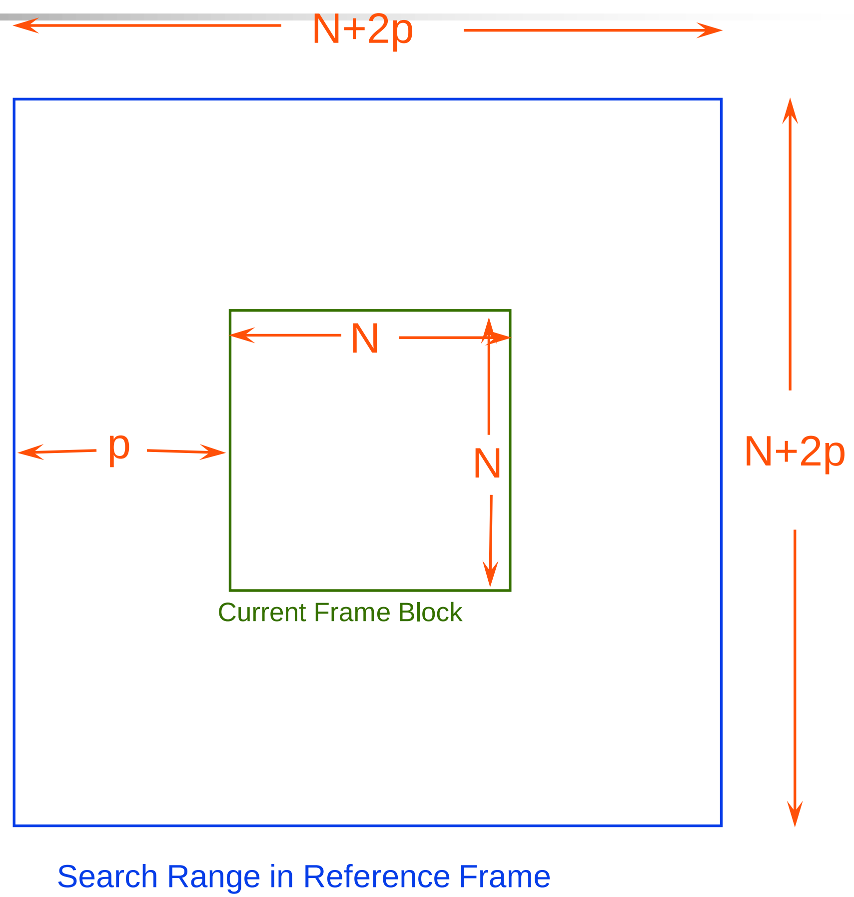
Motion Vector
V_t(p.q) =(Veci,Vecj)
the location in the search range Ω
that has the maximum correlation value
between blocks in temporally adjacent frames

Block Matching Algorithm (BMA)



Factors of Affecting BMA

- Search algorithm
- Matching criterion
  - SSD (sum of squared pixel difference, mostly used in software)
  - SAD (sum of absolute pixel difference, mostly used in hardware)
- Search range [-p,+p]



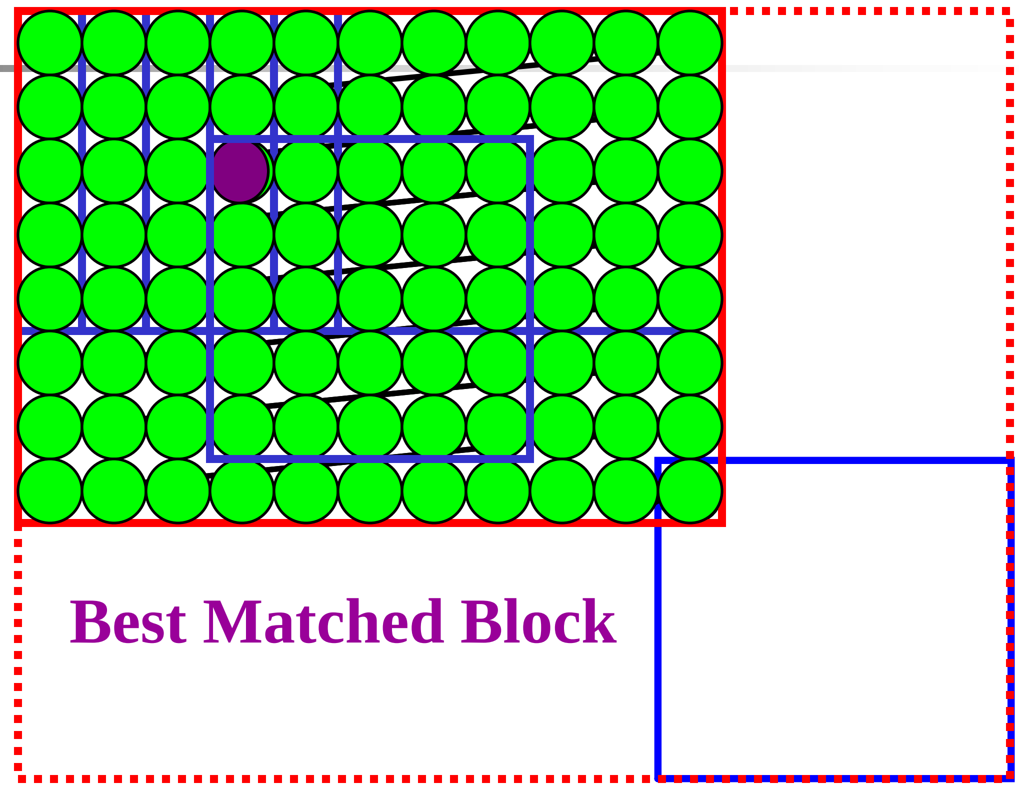
Full-Search Block Matching Algorithm

Current Block

Search Range

Reference Block (Candidate Block)

Candidate Search Position (Search Location)



$$SAD(i,j)=\sum_{k=1}^{N}\sum_{l=1}^{N}\left|x_t(k,l)-x_{t-1}(k+i,l+j)\right|$$

Computation Complexity

```
Loop 1: For m= 0 to (width/blocksize)-1
Loop 2:    For n= 0 to (height/blocksize)-1
Loop 3:       For i = -d to d-1
Loop 4:          For j = -d to d-1
Loop 5:             For k = 0 to N-1
Loop 6:                For l = 0 to N-1
                             SAD(i,j) = SAD(i,j) + |X(k,l)-Y(k+i,l+j)|
                       End (Loop 6)
                    End (Loop 5)
                 End (Loop 4)
              End (Loop 3)
           End (Loop 2)
        End (Loop 1)
```

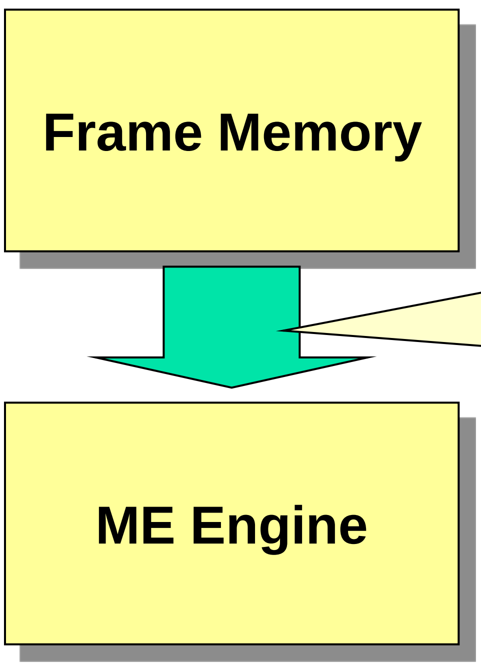
For each macroblock

For each candidate search position

Calculate the distortion, and chose the smallest one

Ultra Large Complexity and Memory Bandwidth

- For example, for a 720x576 30fps video, block size=16x16, SR=[-64, 63]

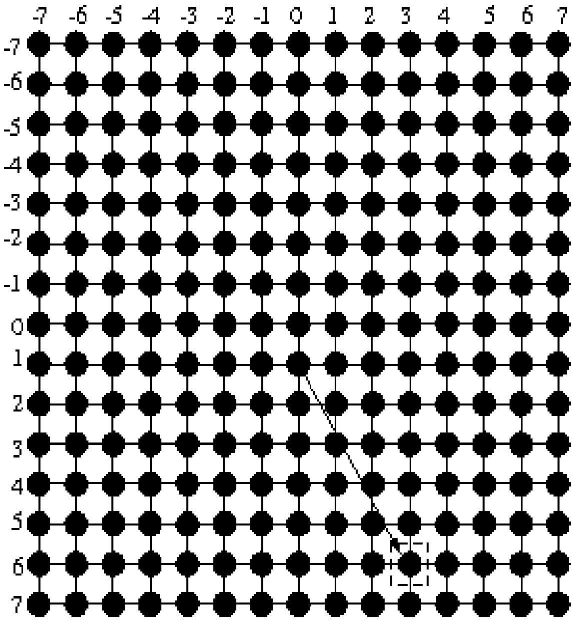


$$30 \times \frac{720}{16} \times \frac{576}{16} \times 128 \times 128 \times 16 \times 16 \times 2 = 408\text{GByte/s}$$

$$30 \times \frac{720}{16} \times \frac{576}{16} \times 128 \times 128 \times 16 \times 16 = 204\text{GOPS}$$

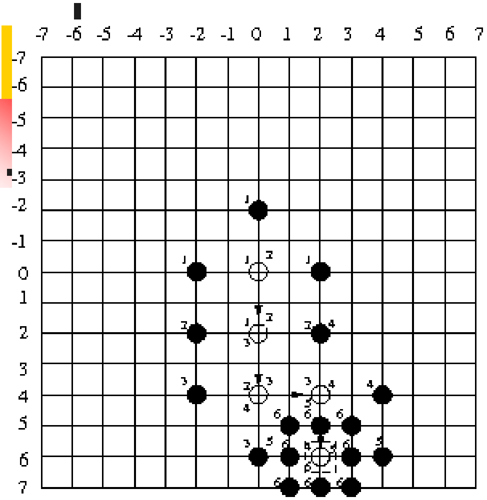
Review of Block Matching Algorithms

- Two-Dimensional Full Search(2DFS) (FSBMA)
  - Exhaustive search
  - Extremely Large Computation
- Fast Search
  - Assumption: Monotonic Distortion Function
  - Reduce computation at the expense of accuracy
  - 2-D Logarithmic Search
  - Modified 2-D Logarithmic Search
  - Three-Step Hierarchical Search
  - Cross Search
  - One-at-a-time Search
  - Parallel Hierarchical One-Dimensional Search
  - One-Dimensional Full Search

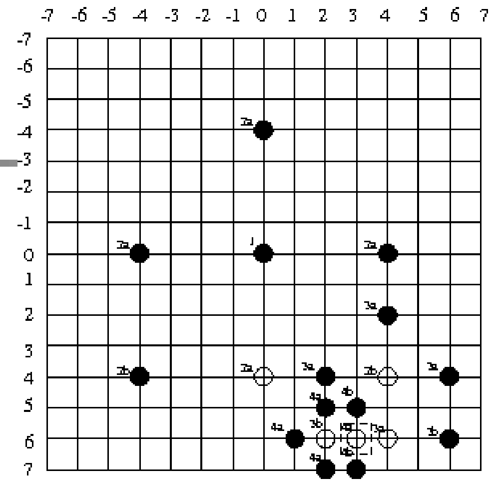


Two-Dimensional Full Search Procedure
Search Range -7 to +7
Motion Vector (3,6) in this case

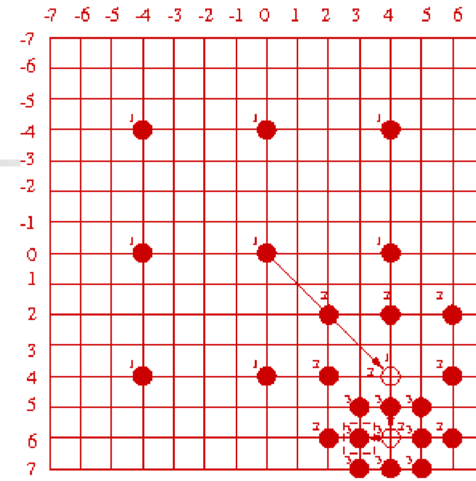
Fast Search Algorithms



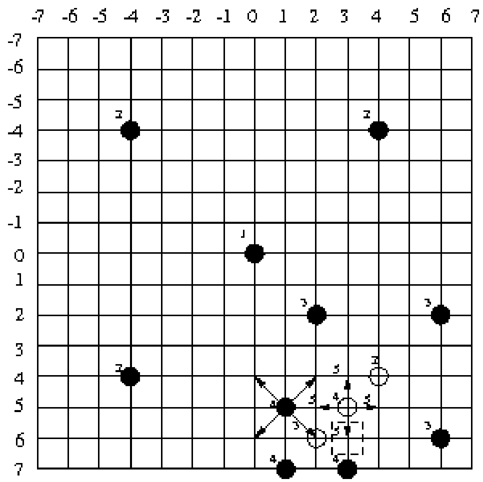
2-D Logarithmic Search Procedure
Search Range -7 to +7
Motion Vector (3,6) in this case



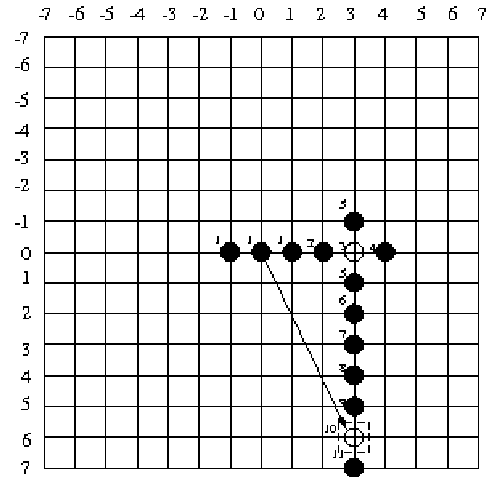
Modified 2-D Logarithmic Search Procedure
Search Range -7 to +7
Motion Vector (3,6) in this case



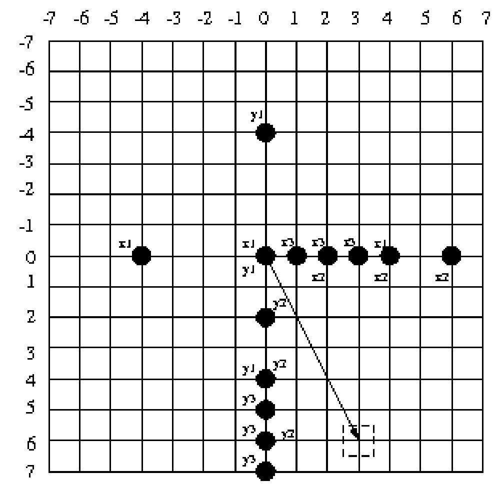
Three-Step Hierarchical Search Procedure
Search Range -7 to +7
Motion Vector (3,6) in this case



Cross-Search Procedure
Search Range -7 to +7
Motion Vector (3,6) in this case



One-at-a-time Search(OTS) Procedure
Search Range -7 to +7
Motion Vector (3,6) in this case



Parallel Hierarchical One-Dimensional Search Procedure
Search Range -7 to +7
Motion Vector (3,6) in this case

Fast Search Algorithms

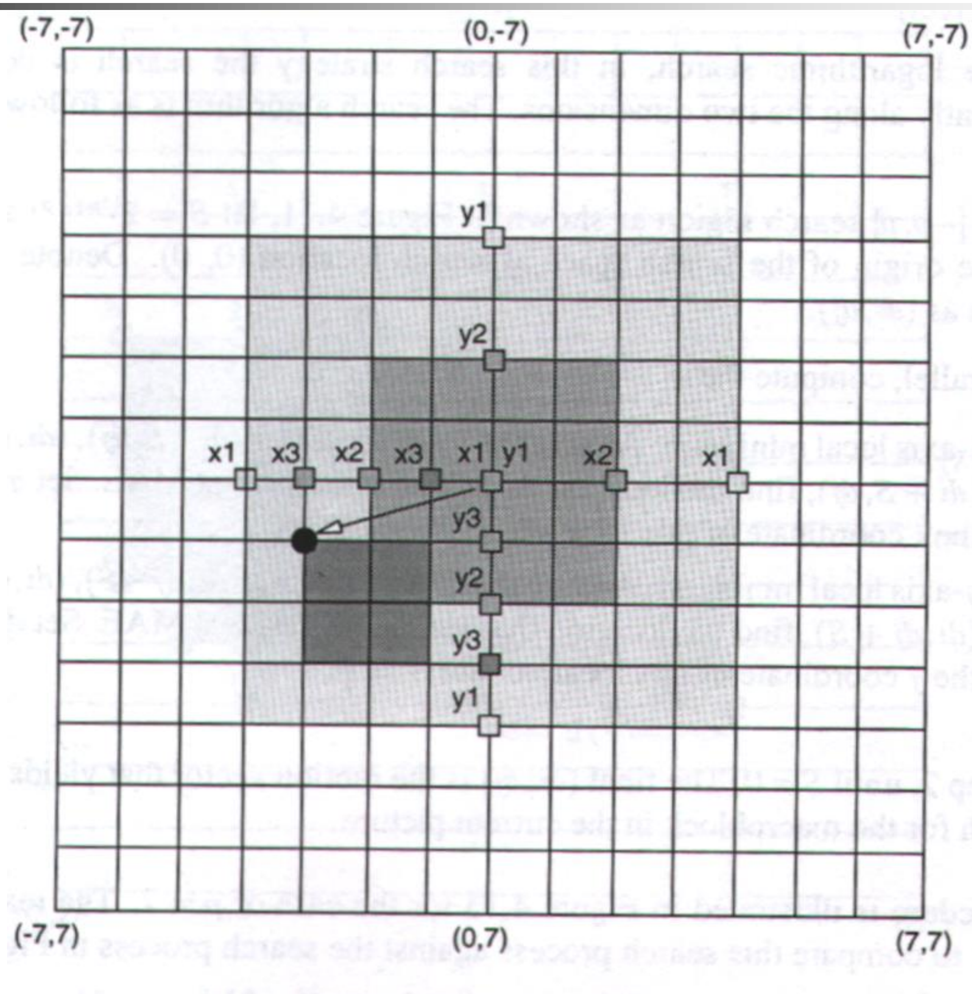


Figure 4.13 Example of PHODS strategy.

Fast Search Algorithms
- One-Dimensional Full Search

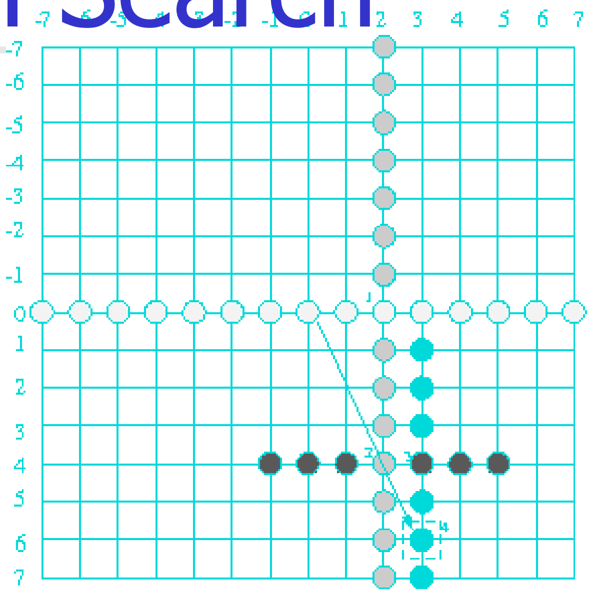
For (i= -P to +P)   Veci'= i | min D(i,j)

For (j= -P to +P)   Vecj'= j | min D(Veci',j)

For (i= -P/2 to +P/2)   Veci = i | min D(i+Veci',Vecj')

For (j= -P/2 to +P/2)   Vecj = j | min D(Veci,j+Vecj')

Motion Vector = (Veci,Vecj)

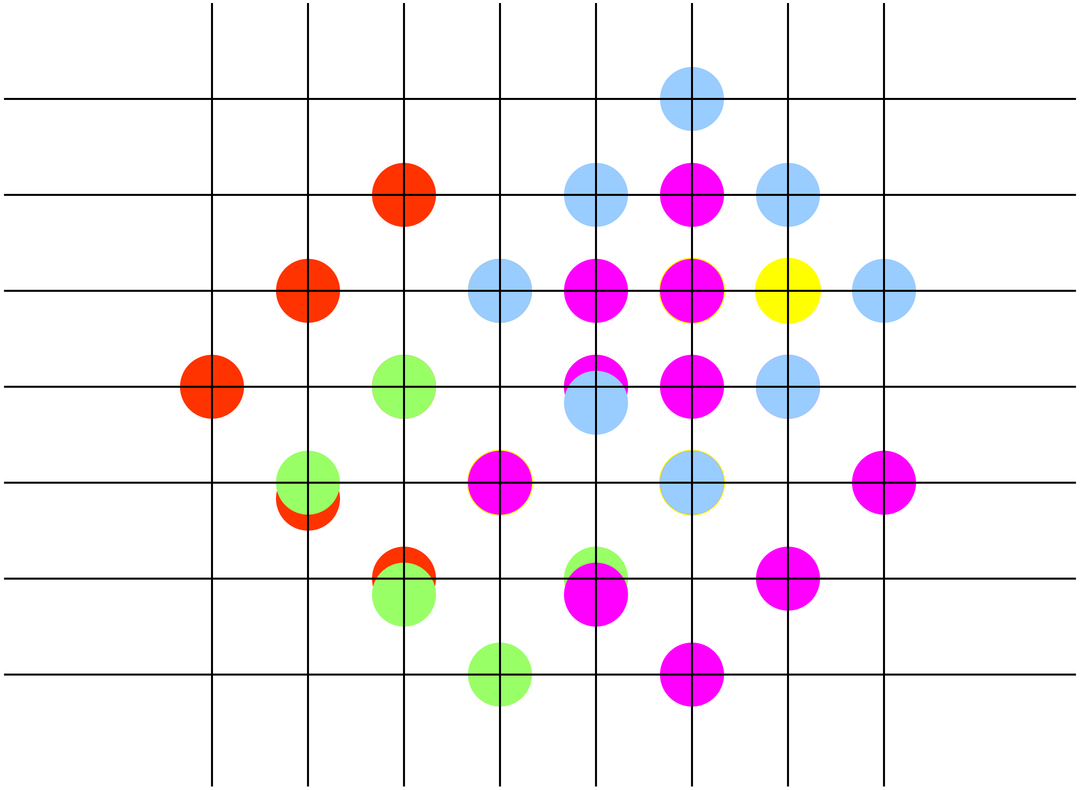


One-Dimensional Full Search Procedure
Search Range -7 to +7
Motion Vector (3,6) in this case

- ***Hardware-Oriented***
- ***Full Data Reuse***
- ***Regular Data Flow***
- ***Fixed-Step Operation***
- ***Good Performance***

Ref: Mei-Juan Chen, Liang-Gee Chen and Tzi-Dar Chiueh, "One-Dimensional Full Search Motion Estimation Algorithm for Video Coding", IEEE Transactions on Circuits and Systems for Video Technology, Vol.4, No.5, October 1994.

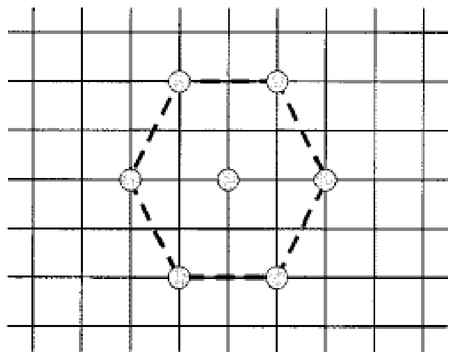
Diamond Search

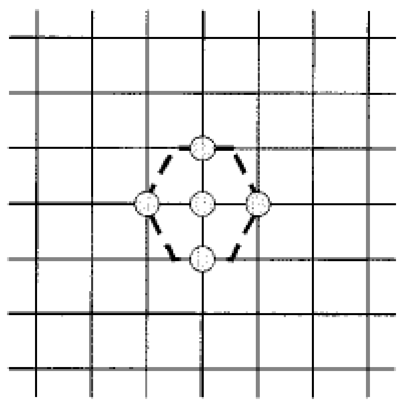


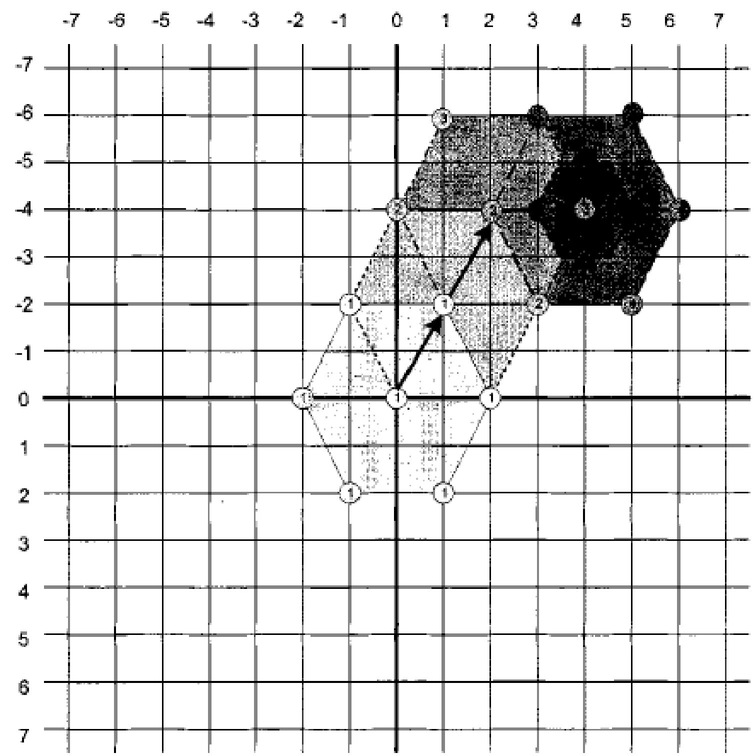
MV

Ref: Shan Zhu and Kai-Kuang Ma, "A new diamond search algorithm for fast block matching motion estimation," *Proceedings of ICICS, 1997 International Conference on Information, Communications and Signal Processing.*

Hexagon-Based Search (HEXBS)



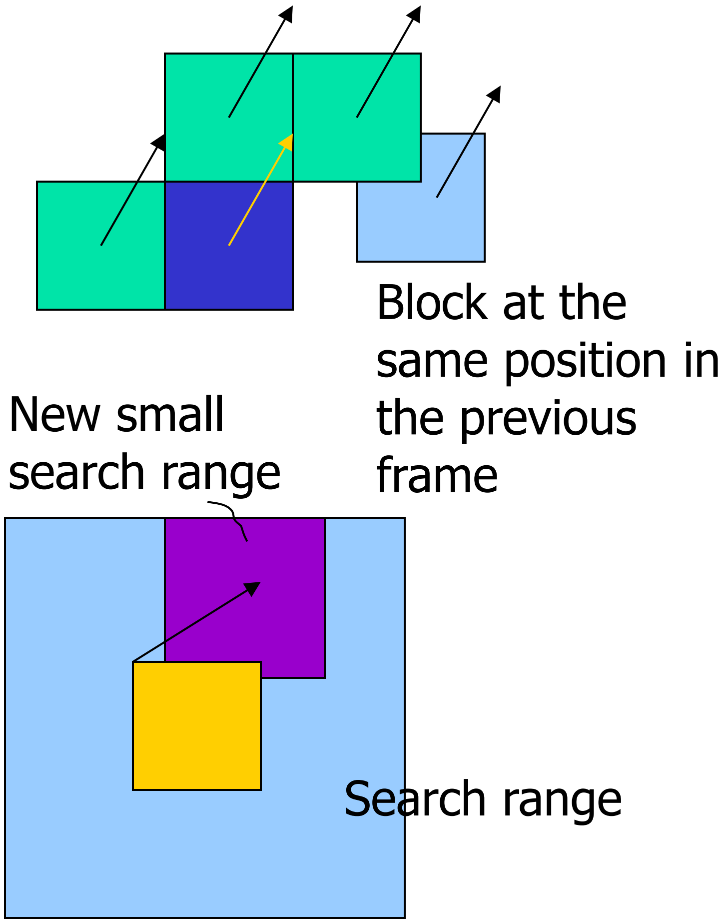




Ce Zhu, Xiao Lin, and Lap-Pui Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 5, pp. 349--355, May 2002.

Motion Vector Prediction

- Prediction: guess the motion vector of the current block from the neighbors
- Good prediction can
  - Improve the performance of fast motion estimation
  - Can achieve similar quality with smaller search range



Matching Criteria

- Cross-Correlation Function (CCF)
- Mean Squared Error (MSE)
- Mean Absolute Error (MAE)
- Pel Difference Classification (PDC)
- Minimized Maximum Error (MiniMax)

$x_t(k,l)$: luminance for the location (k,l) in $X_t(p,q)$

$x_{t-1}(k+i,l+j)$: luminance for the shifted location by i pels and j lines within the search range

Matching Criteria

Cross-Correlation Function(CCF)

$$\text{CCF(i,j)=}\ \frac{\sum_{k=1}^{N}\sum_{l=1}^{N} x_t(k,l)\cdot x_{t-1}(k+i,l+j)}{\left[\sum_{k=1}^{N}\sum_{l=1}^{N} x_t^2(k,l)\right]^{\frac{1}{2}}\left[\sum_{k=1}^{N}\sum_{l=1}^{N} x_{t-1}^2(k+i,l+j)\right]^{\frac{1}{2}}}$$

(Veci,Vecj)=(i,j) | $_{\text{max CCF(i,j)}}$

Mean Squared Error(MSE)

$$\text{MSE(i,j)=}\ \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}\left[x_t(k,l)-x_{t-1}(k+i,l+j)\right]^2$$

(Veci,Vecj)=(i,j) | $_{\text{min MSE(i,j)}}$

Matching Criteria

Mean Absolute Error(MAE) ➡ Most Widely Used in hardware

$$\text{MAE(i,j)}= \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}\left|x_t(k,l)-x_{t-1}(k+i,l+j)\right|$$

(Veci,Vecj)=(i,j) | $_{\text{min MAE(i,j)}}$

Pel Difference Classification(PDC)

$$T(k,l,i,j)=1, \left|x_t(k,l)-x_{t-1}(k+i,l+j)\right| \leq Threshold$$

$$0, otherwise$$

$$\text{G(i,j)}= \sum_{k=1}^{N}\sum_{l=1}^{N} T(k,l,i,j)$$

(Veci,Vecj)=(i,j) | $_{\text{max G(i,j)}}$

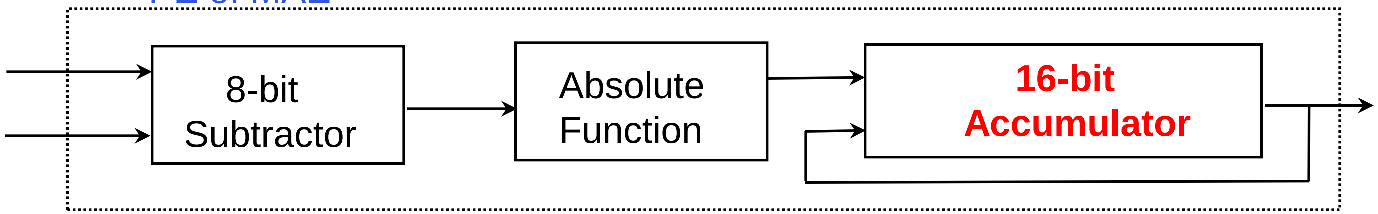
Matching Criteria

Minimized Maximum Error (MiniMax) → Hardware Reduction

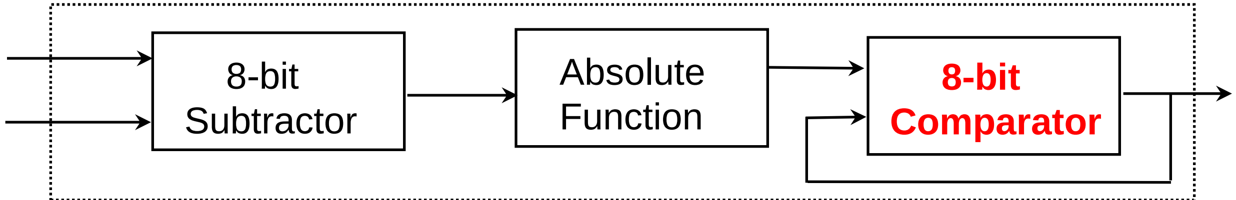
$$G(i,j) = \max \mid \left| x_t(k,l) - x_{t-1}(k+i, l+j) \right|$$

$$(Veci, Vecj) = (i,j) \mid_{\min G(i,j)}$$

PE of MAE



PE of MiniMax

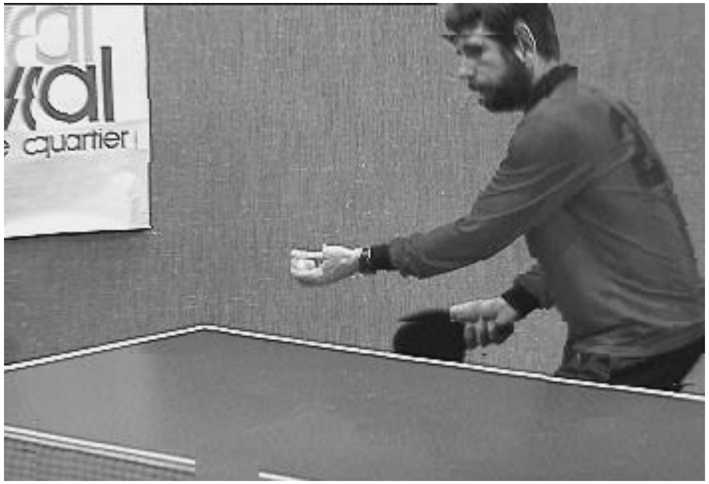


Subjective Quality

Min-max | MAE

1DFS'

Table Tennis I



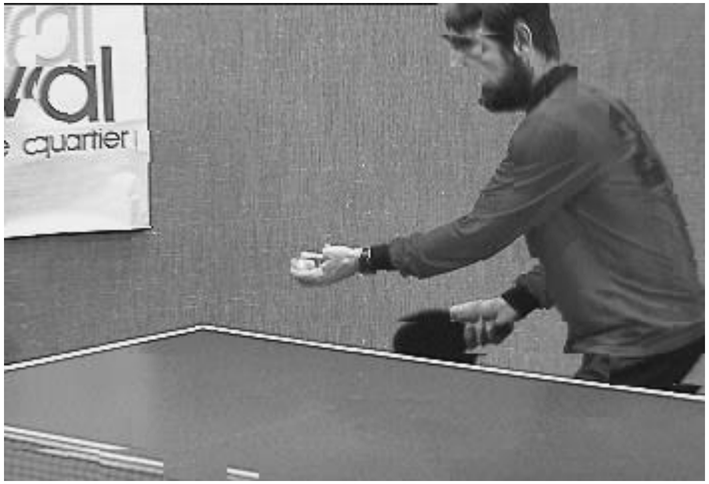
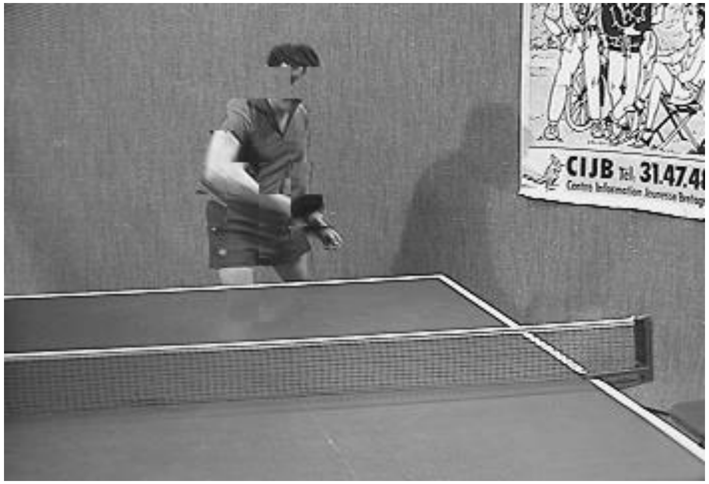


Table Tennis II



Subjective Quality

	Min-max	MAE
Salesman		
Train & Calendar		

Other Approaches
- Pixel Subsampling for MAE calculations

1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4
1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4
1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4
1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4

Matching block size 16x16 → 8x8
Pixel decimation for block matching in an 8x8 block.

- For each MAE value , use only 1/4 of the pixels.
- But every pixel in the block will be used.
- It minimizes the possibility of not considering one-pixel-wide horizontal , vertical and diagonal lines .

Projections for MAE calculations

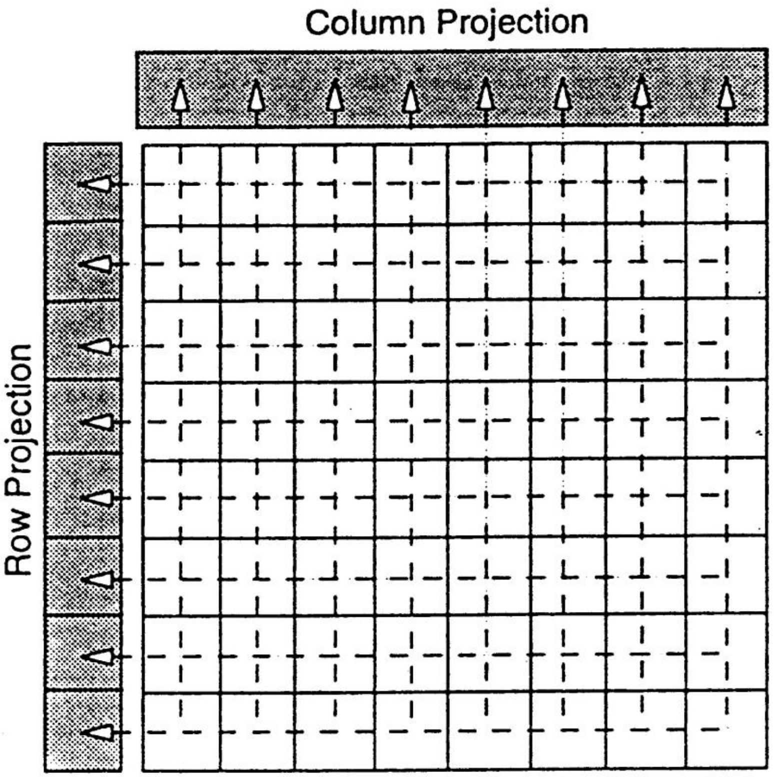
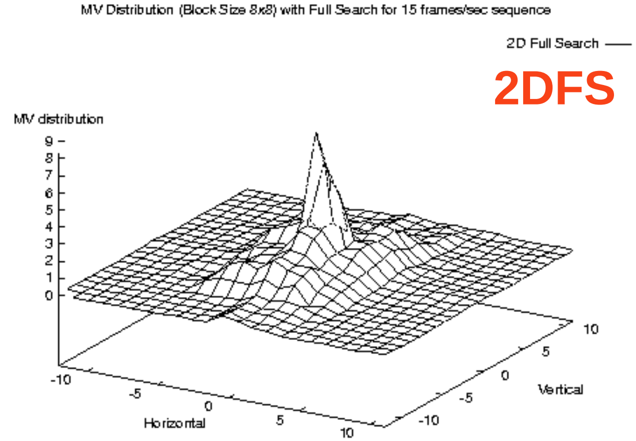


Figure 4.16 Row and column projection of pixels in an 8 × 8 block.

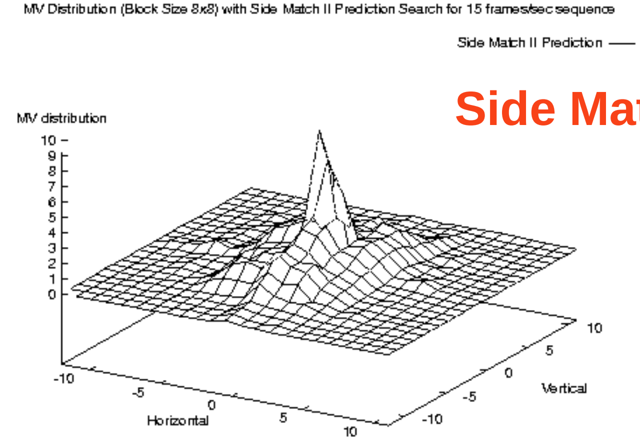
Motion Vector Distribution

block size 8x8 **15 frames/sec**
Table Tennis **Sequence**

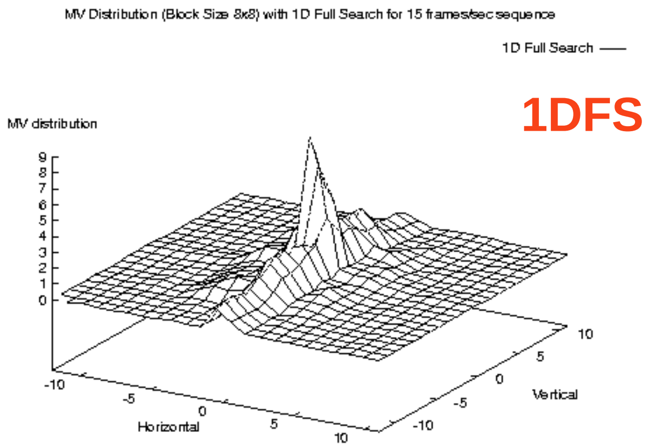
2DFS



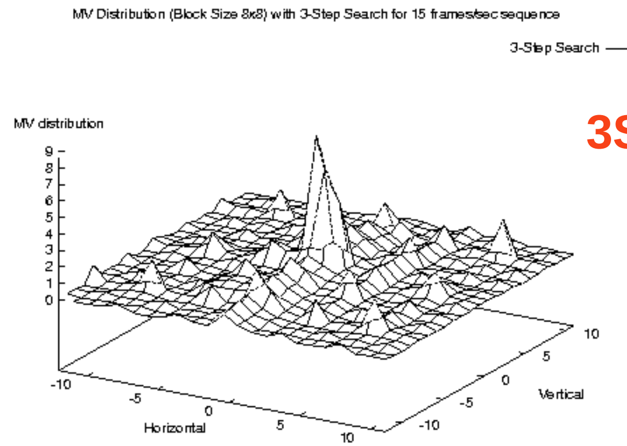
Side Match



1DFS



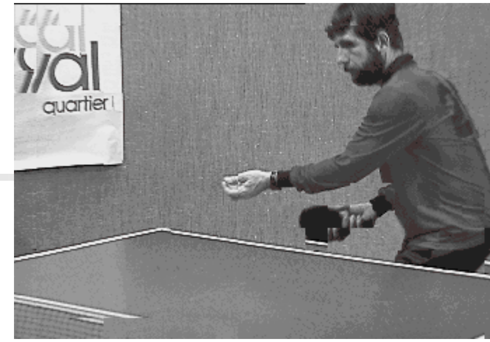
3Step



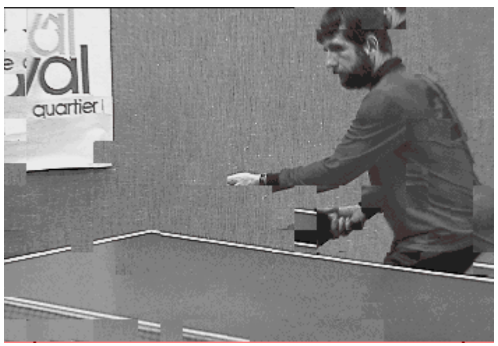
Subjective Quality

16x16 blocks
15 frames/sec Table Tennis Sequence

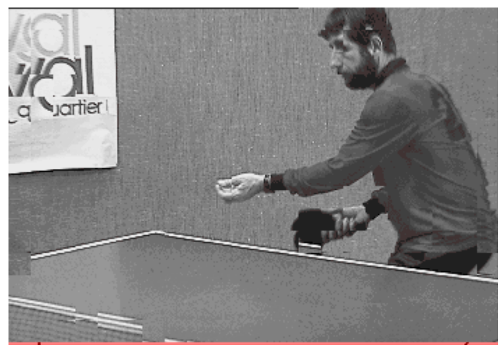
Original



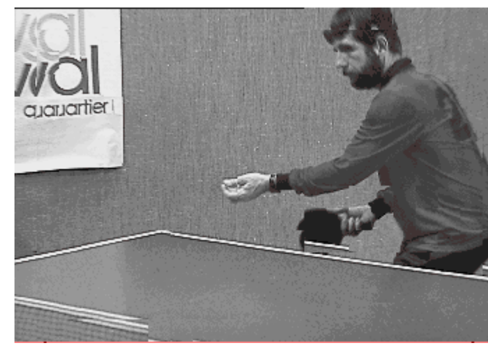
2DFS



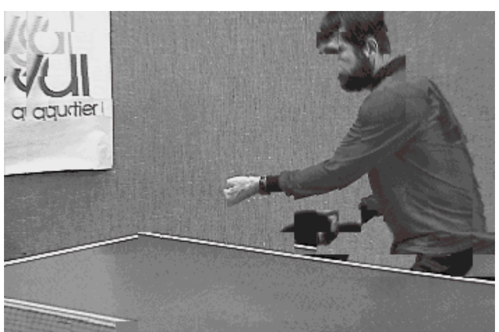
Inter-Frame



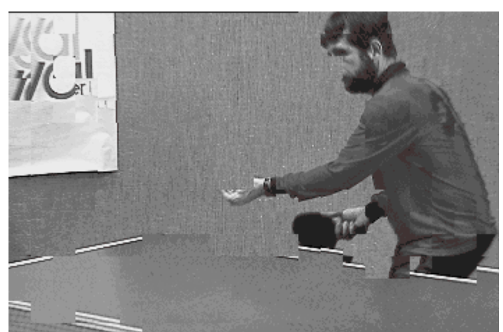
Inter-Block



Boundary Match



1DFS



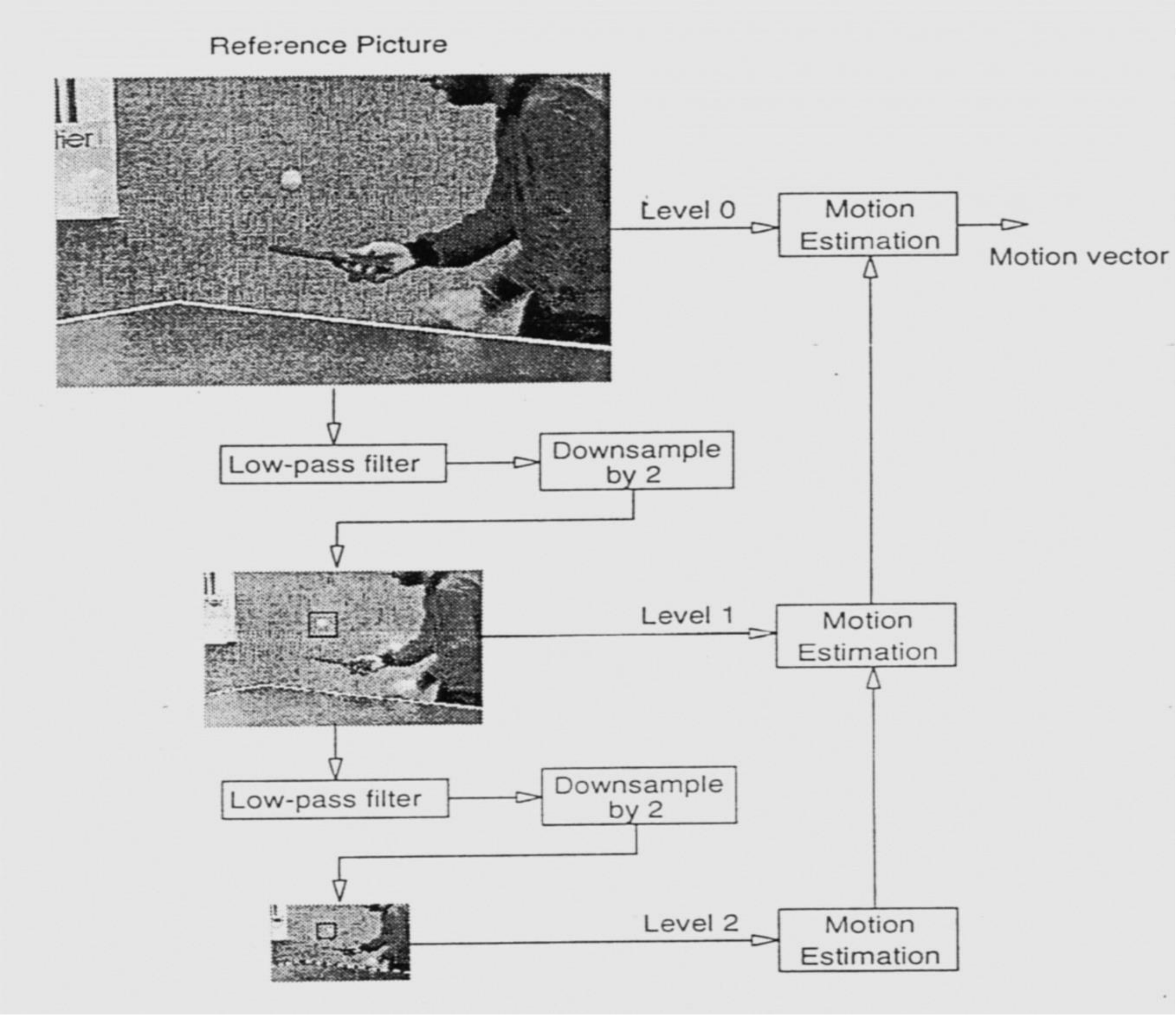
3Step

Side Match

Improvements of Motion Estimation

- Hierarchical motion estimation
- Multigrid block matching
- Overlapped block matching
- Motion estimation with fractional precision
- Bidirectional prediction
- Lossless motion estimation

Hierarchical Motion Estimation



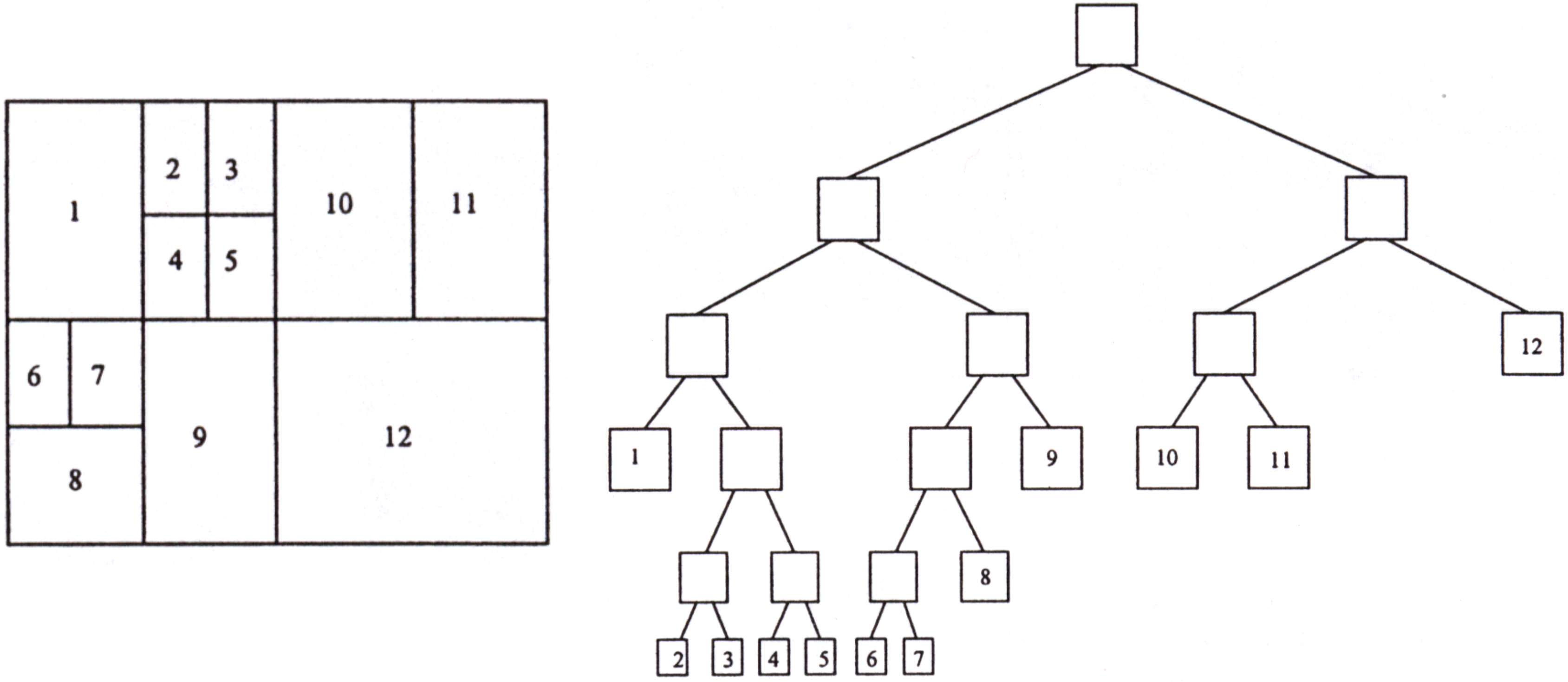
Hierarchical Motion Estimation

- Summary:
  - It requires increased storage to keep pictures at different resolutions.
  - Motion vector may be inaccurate for regions containing small objects .
  - Low-pass filter may be necessary to reduce noise.

Search Method	Operations per Macroblock	Operations for pictures 720 x 40 at 30 fps	
		$p = 15$	$p = 7$
Full-search	$(2p+1)^2 NM3$	29.89 GOPS	6.99 GOPS
Logarithmic	$(8\lceil log_2 p\rceil + 1)NM3$	1.02 GOPS	777.60 MOPS
PHODS	$(4\lceil log_2 p\rceil + 1)NM3$	528.76 MOPS	404.35 MOPS
Hierarchical	$[(2\lceil \frac{p}{4}\rceil + 1)^2 + 180]\ \frac{NM}{16}3$	507.38 MOPS	398.52 MOPS

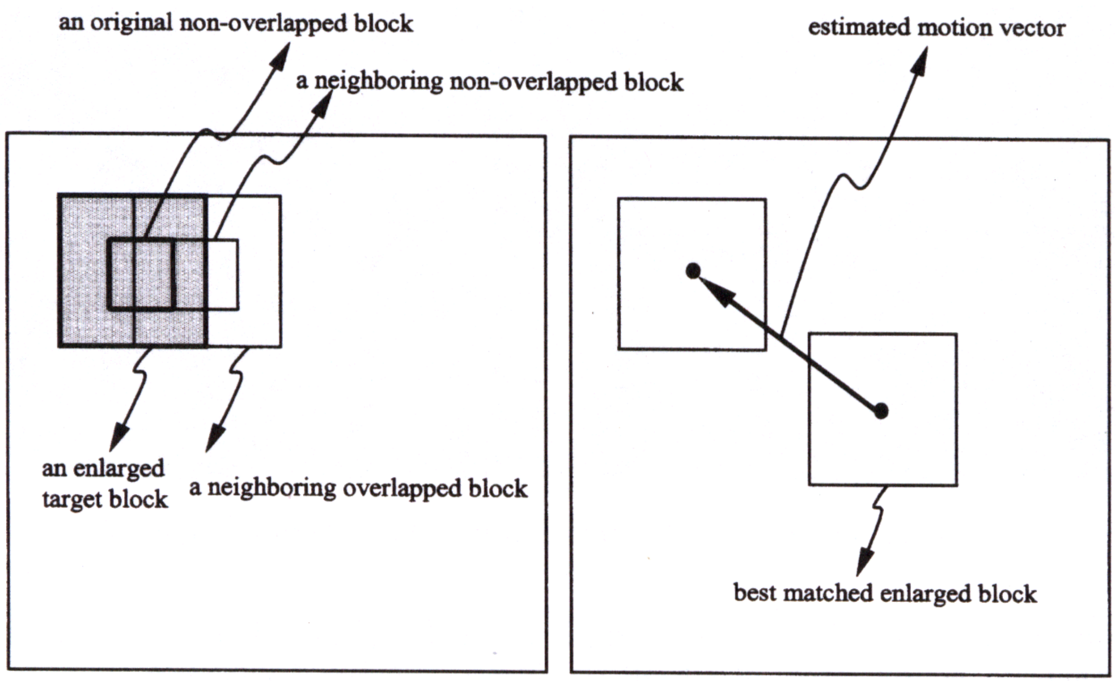
Table 4.1 Computational complexity and MOPS requirements for various motion-estimation algorithms using the MAE criterion and a [-p, p] search range.

Multigrid Block Matching



Overlapped Block Matching (1)

- Relax the restriction of a non-overlapped block partition



(a) frame at t_n (b) frame at t_{n-1}

The motion estimation procedure is the same except for using **enlarge blocks** and a **window function**

Overlapped Block Matching (2)

- Overlapped block motion estimation

$$E_{v_i}(x,y) = P_{v_i}(x,y) - T(x,y)$$

$$WE_{v_i}(x,y) = E_{v_i}(x,y) \times W(x,y)$$

$$\text{Block matching : to minimize } MAD = \frac{1}{l^2}\sum_{x=1}^{l}\sum_{y=1}^{l}\left|WE_{v_i}(x,y)\right|$$

$T(x,y)$: enlarged current block

$P_{v_i}(x,y)$: corresponding block in reference with motion vector v_i

$W(x,y)$: window function

- Overlapped block motion compensation (OBMC)

$$WP_{v_i}(x,y) = P_{v_i}(x,y) \times W(x,y)$$

Sub-Pixel-Accurate Motion Estimation (Fractional Precision)

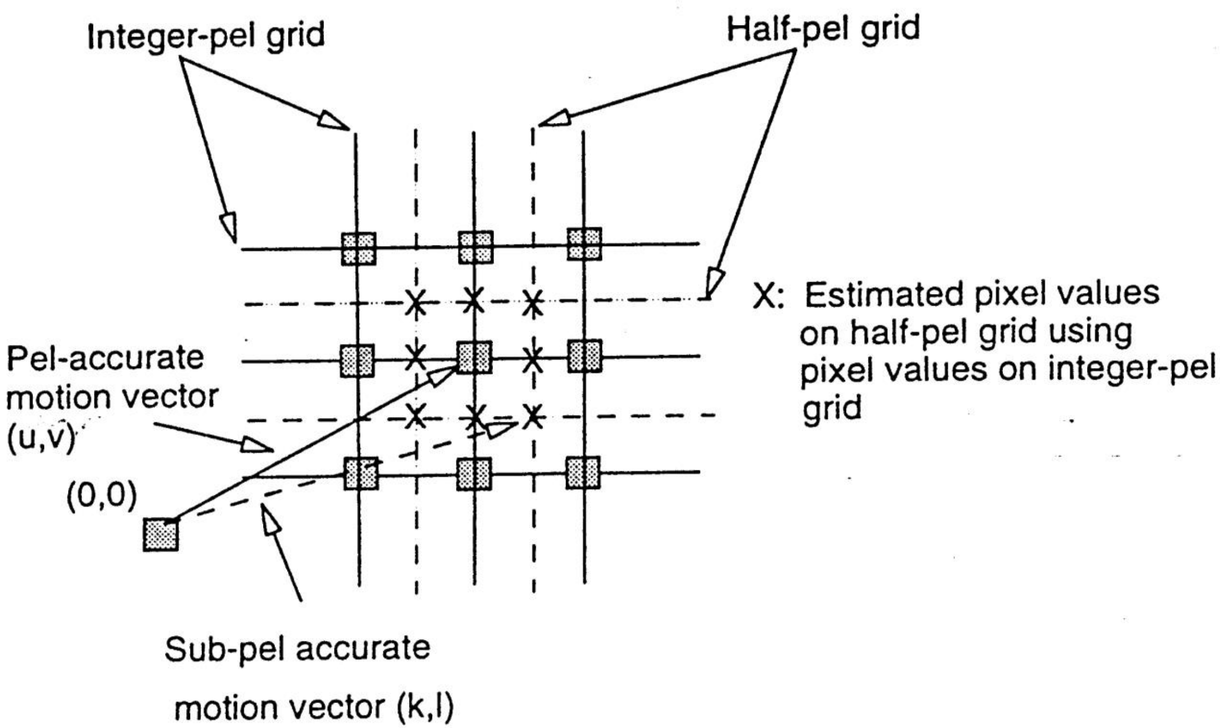
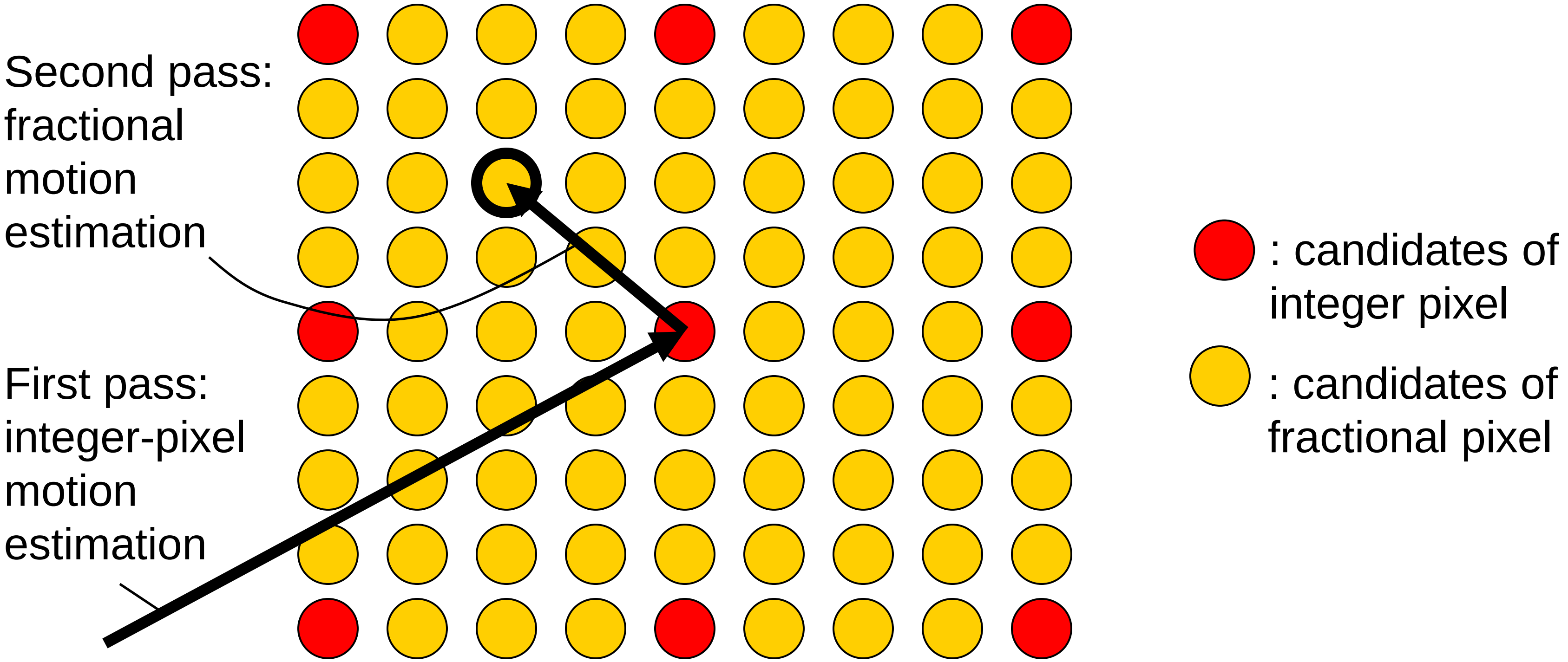


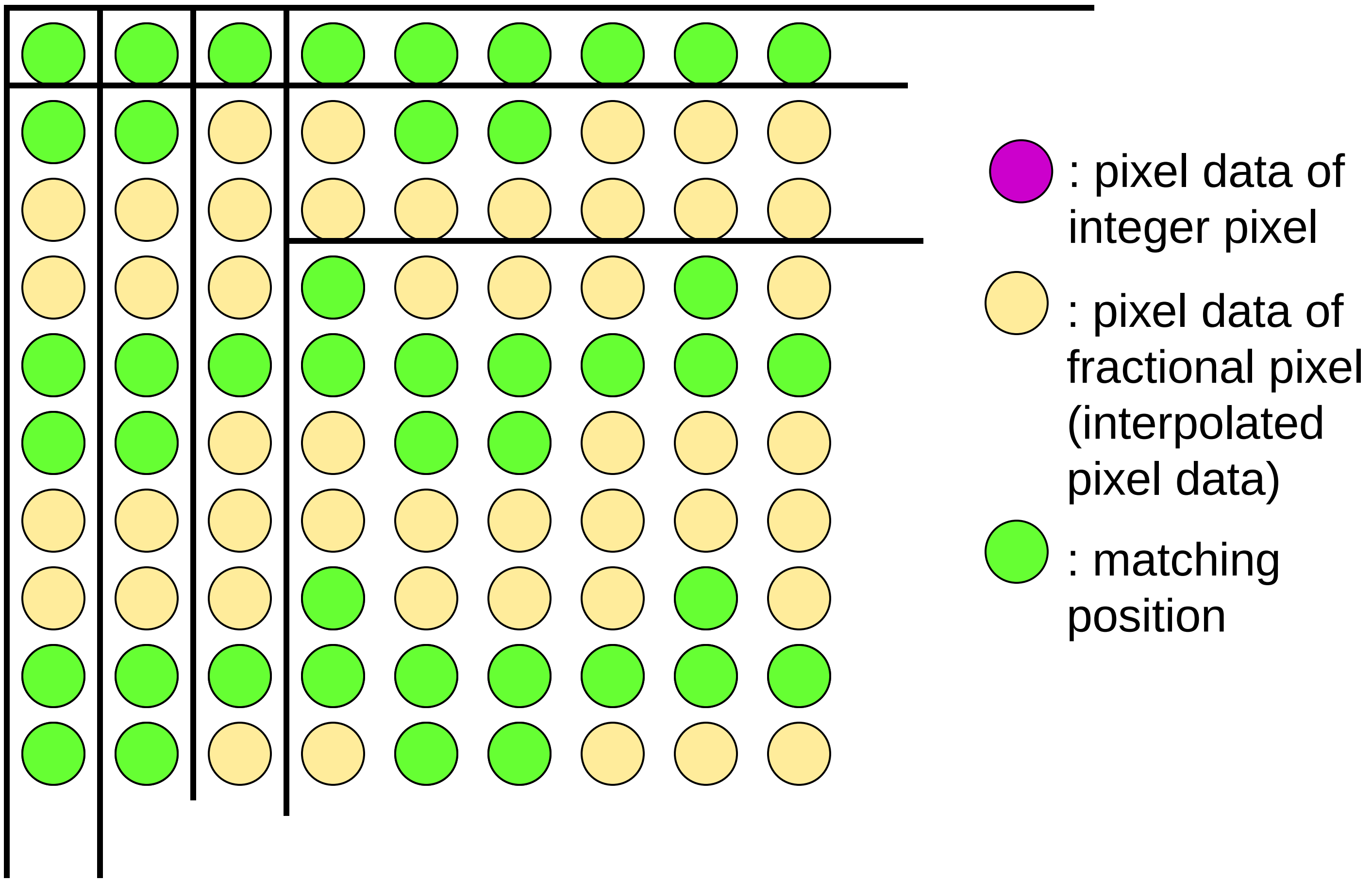
Figure 4.19 Half-pel accurate motion vector estimation.

Motion Estimation with Fractional Precision (1/2)

- Quarter-pel precision

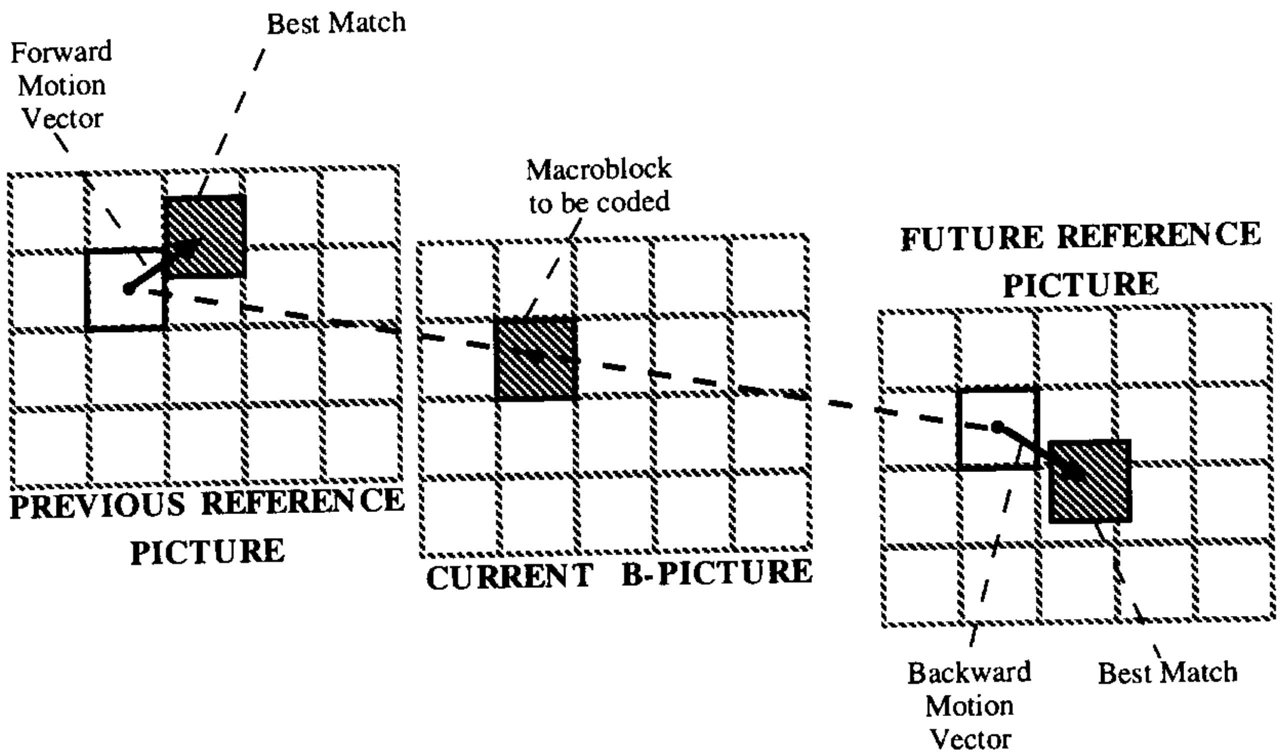


Motion Estimation with Fractional Precision (2/2)



Bidirectional Temporal Prediction for Progressive Video

- Reference frame must be I or P-frame
- B-picture
- 2 MV

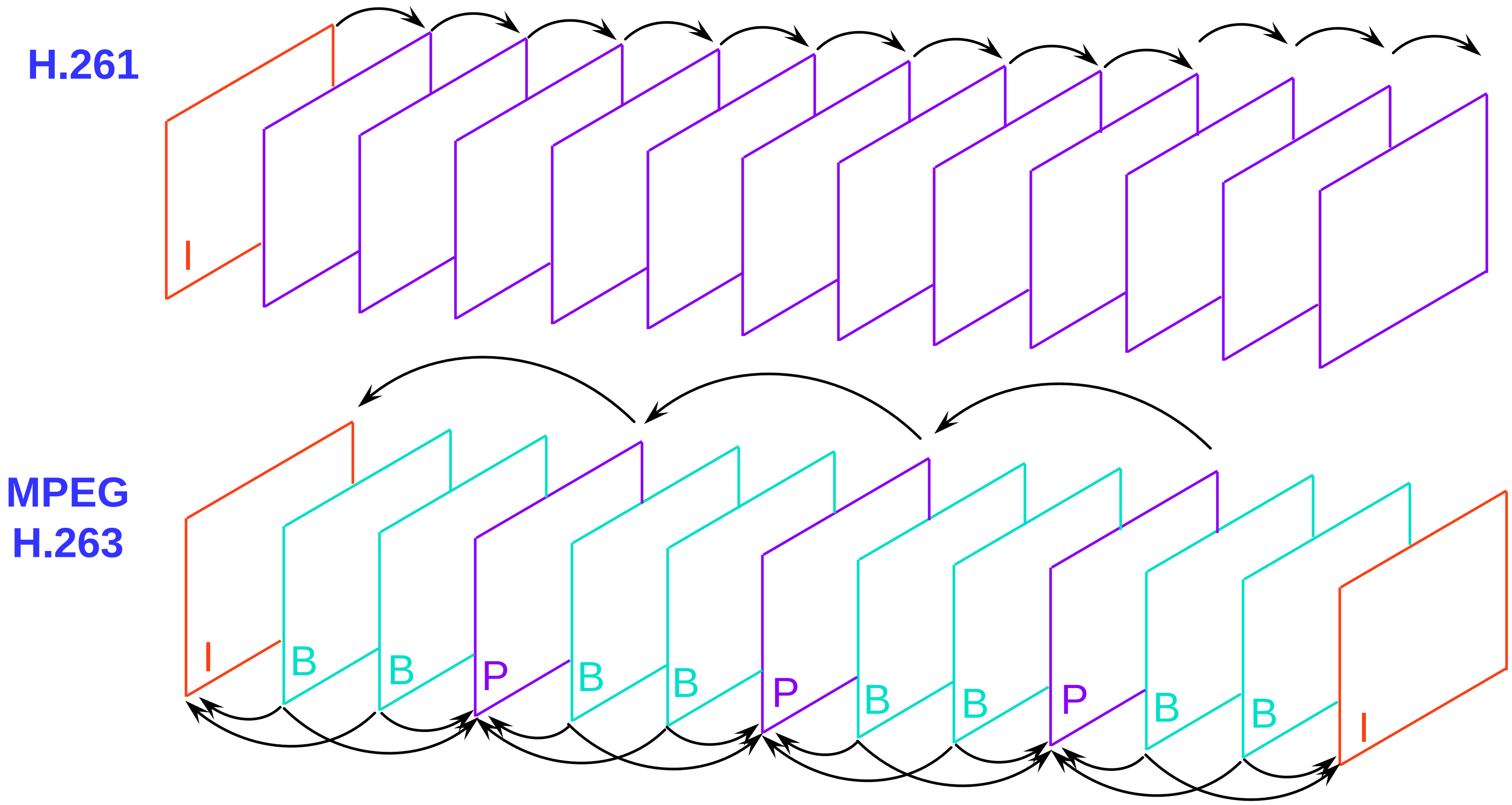


Coding of Moving Pictures

I: Intra Frame

P: Predict Frame

B: Bi-directional Frame



Lossless Motion Estimation

- Partial Distortion Elimination (PDE)
  - In the loop of calculating SAD, if the current SAD is larger than the minimum SAD value, then quit the loop.
- Successive Elimination Algorithm (SEA)

By the following inequality:

$$\sum |A-B| \geq \sum |A| - \sum |B|$$

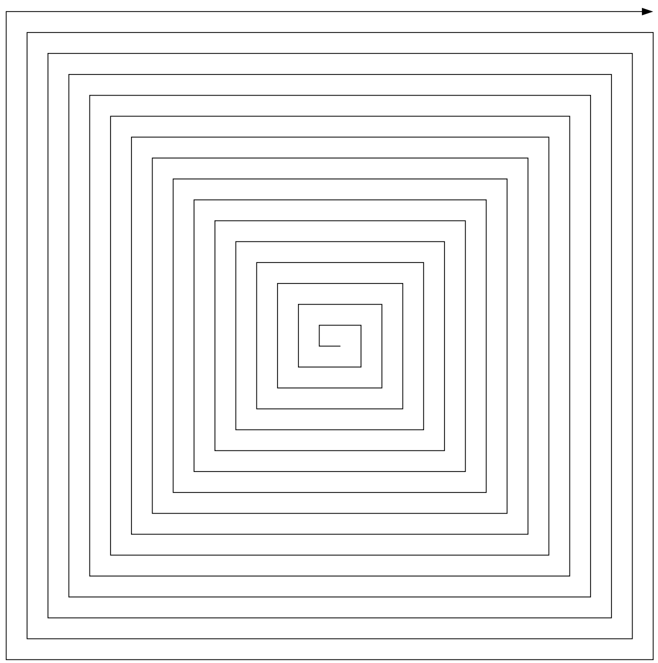
So

$$\sum_{j=0}^{N-1}\sum_{i=0}^{N-1}\left|c(i,j)-p(i+u,j+v)\right| \geq \sum_{j=0}^{N-1}\sum_{i=0}^{N-1}\left|c(i,j)\right| - \sum_{j=0}^{N-1}\sum_{i=0}^{N-1}\left|p(i+u,j+v)\right| = C - \sum_{j=0}^{N-1}\sum_{i=0}^{N-1}\left|p(i+u,j+b)\right| = K$$

if K > current minimum SAD then skip this reference point

Spiral Scan

- Provide a good initial guess for PDE and SEA



Lossy Motion Estimation

- Variable/Dynamic Search Range Techniques
- Half-stop Techniques Using Threshold of Matching Distortion
- Lower Bit-resolution Techniques (Using Quantization)
  - Pixel-truncation
- Subsampling Techniques of Matching Block