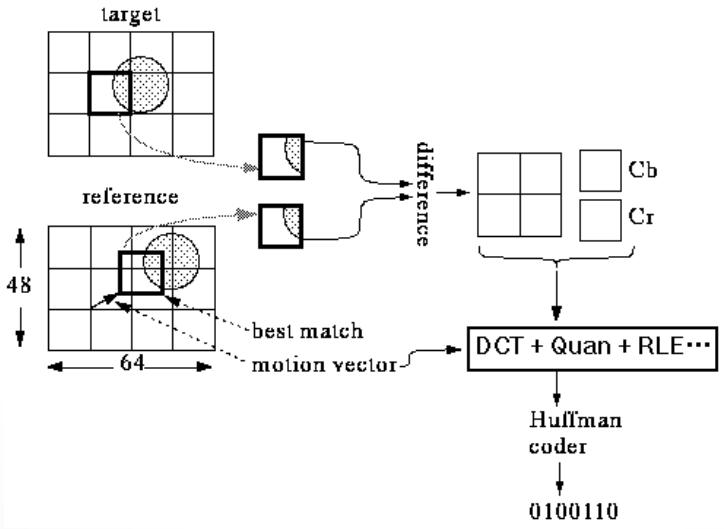
Block Distortion Measure

- The range of motion is predicted based on the correlation between the block distortion measure (BDM) of origin checking point and motion vector magnitude.



Matching Criteria

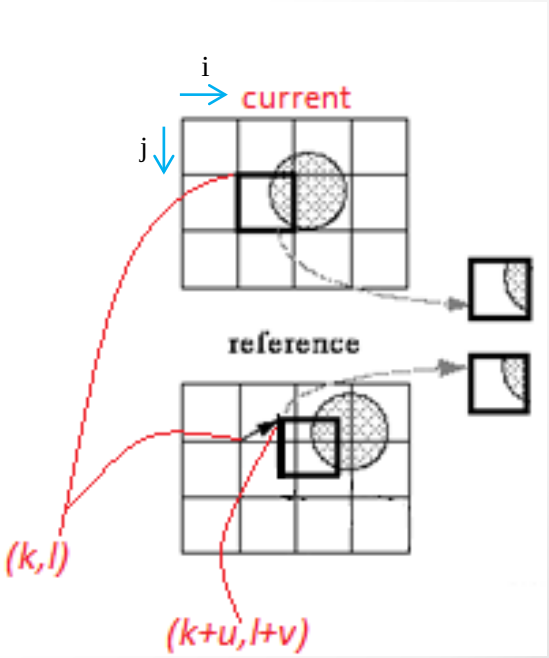
- In order to measure the similarity between the block of the current frame and a candidate block of the reference frame, various matching criteria such as mean square error (MSE), mean absolute error (MAE) and maximum matching pel count (MPC) are defined.
- Suppose the top left corner of the searching block of frame n is (k,l), the displacement of the matching block of frame n-1 is (u,v) and the block size equals to 8x8 , the MSE, MAE and MPC matching criteria are defined as follows:

best

Mean Square Error (MSE)

$$MSE(k,l;u,v)=\frac{1}{64}\sum_{i=0}^{7}\sum_{j=0}^{7}\left(I_n(k+i,l+j)-I_{n-1}(k+u+i,l+v+j)\right)^2$$

- The square operation in the equation makes hardware implementations more complex.
- Instead implementing the MSE criterion, many video encoders employ the MAE as their matching criterion in block matching.



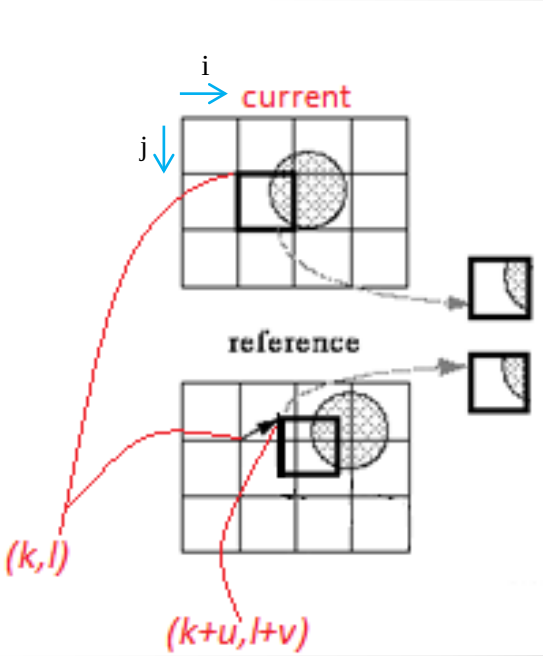
minimum ~~MAE~~ MSE corresponding u, v motion vector

Mean Absolute Error (MAE)

common use

$$MAE(k,l;u,v) = \frac{1}{64}\sum_{i=0}^{7}\sum_{j=0}^{7}\left|I_n(k+i,l+j) - I_{n-1}(k+u+i,l+v+j)\right|$$

- The MAE criterion is the most popular choice for hardware implementations.
- Its hardware implementation complexity is much lower than that of the MSE criterion.
- Another matching criterion is the MPC criterion which counts the number of matching pixels.



Maximum Matching Pel Count (MPC)

$$MPC(k,l;u,v) = \sum_{i=0}^{7}\sum_{j=0}^{7} T(k+i,l+j;u,v)$$

$$T(x,y;u,v) = \begin{cases} 1 & \text{if } |I_n(x,y)-I_{n-1}(x+u,y+v)| <= t \\ 0 & \text{otherwise} \end{cases}$$

- where ***t*** is a predetermined threshold.
- Different from MSE and MAE, the best match block found using this matching criterion is the one which gives the largest MPC value.

PSNR

- PSNR (peak signal to noise ratio) is used to evaluate the subjective quality of a reconstructed video sequence of a video codec quantitatively.

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}\right) = 10\log_{10}\left(\frac{255^2}{\text{MSE}}\right)$$

- The value 255 is due to the peak value of a 8-bit quantized pixel.
- N is number of pixel in the picture.
- The PSNR criterion is widely used to evaluate the quality performance of a video codec.

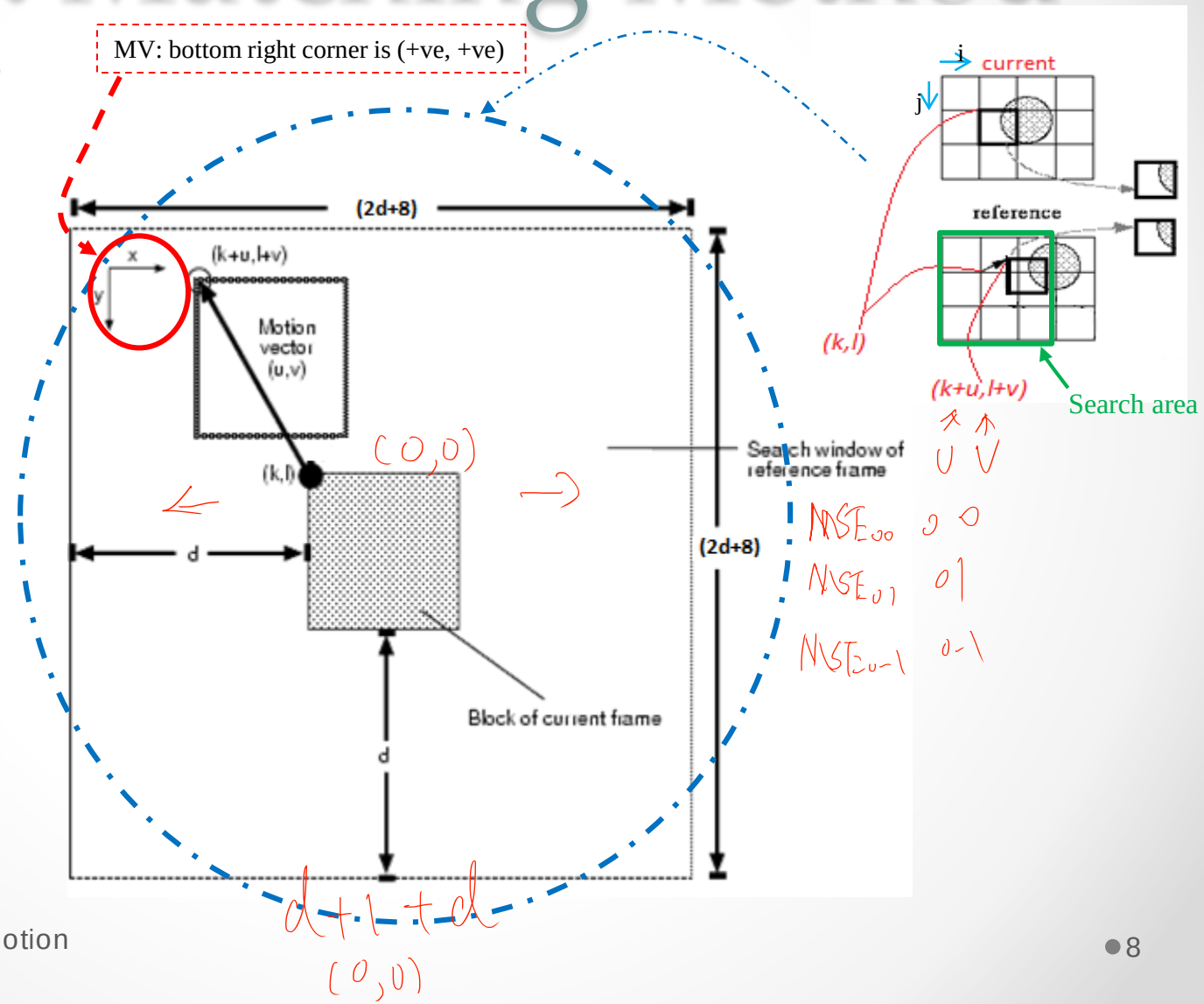
Motion Estimation

- Motion compensation is an predictive technique for exploiting the temporal redundancy between successive frames of video sequence.
- Block matching is a simple and effective motion estimation method to obtain the motion compensated prediction.
- By dividing each frame into rectangular blocks, motion vectors are obtained via the block matching algorithms (BMA).
- The full search algorithm (FS) is a traditional BMA. It searches all possible locations inside the search window in the reference frame to provide an optimal solution. However, its high computational complexity makes it often not suitable for real-time implementation.

full search → best match block

Block Matching Method

- Suppose the block size is pixels and the maximum displacement of a motion vector is d in both horizontal and vertical directions. A motion vector (***u***,***v***) is obtained by finding a match block within a search window of size in the reference frame. The center of the search window is equal to the location of the searching block of the current frame.
- Number of Block comparison = (2*d+1)*(2*d+1)

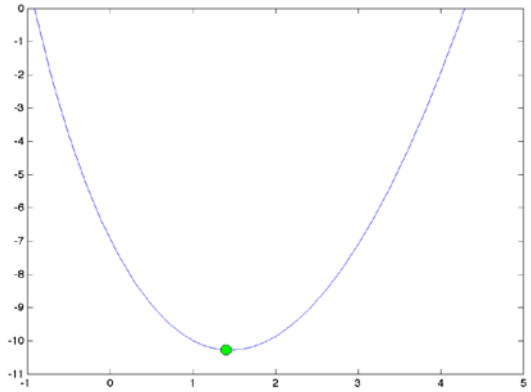


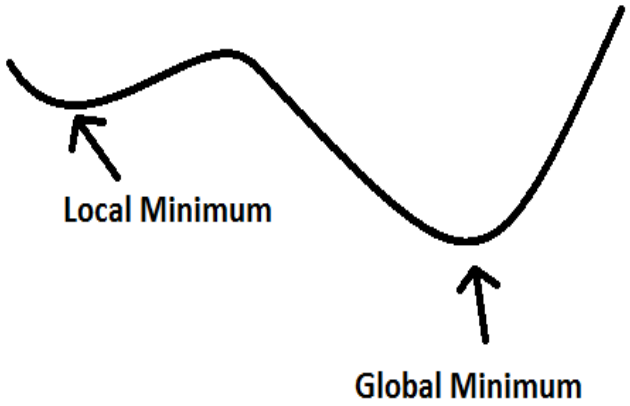
Block Matching Method

- The FS algorithm finds a global optimal motion vector from all the candidate motion vectors (CMV) inside the search window.
- It requires $(2d+1)^2$ BDM calculations. This high computational complexity makes it often not suitable for real-time implementations.
- Many fast BMAs are proposed to reduce computational complexity.

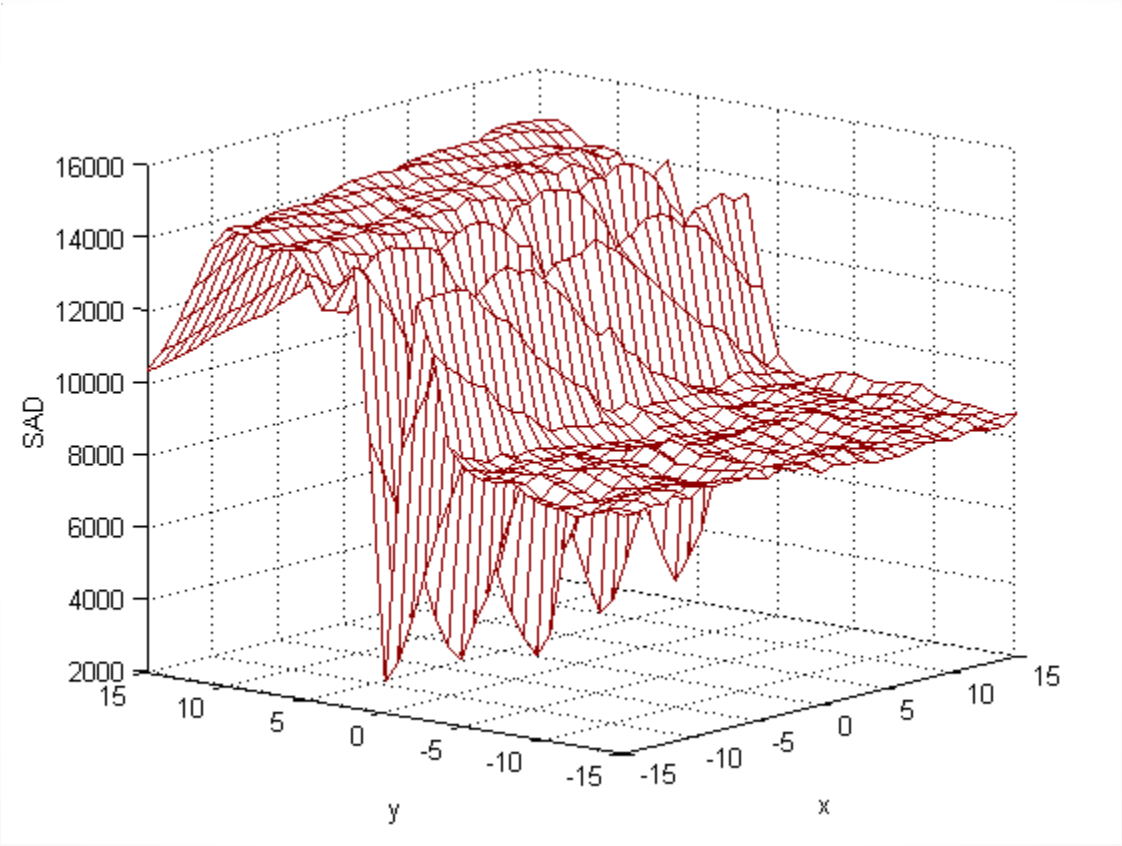
Block Matching Method

- Block matching methods assume that the block of pixels have the same translational motion from frame to frame. Moreover, for the fast BMAs described in the later sections, they assume that BDM increases monotonically as the checking point moves away from the global minimum BDM point. Thus, in each searching step, those fast BMAs would choose the direction such that the BDM of the checking point in that direction is the minimum. Figure depicts the basic principle of block matching. The current frame is divided into small rectangular blocks. For each block of the current frame, a motion vector is obtained by finding the displaced coordinate of a match block within the search window of the reference frame.



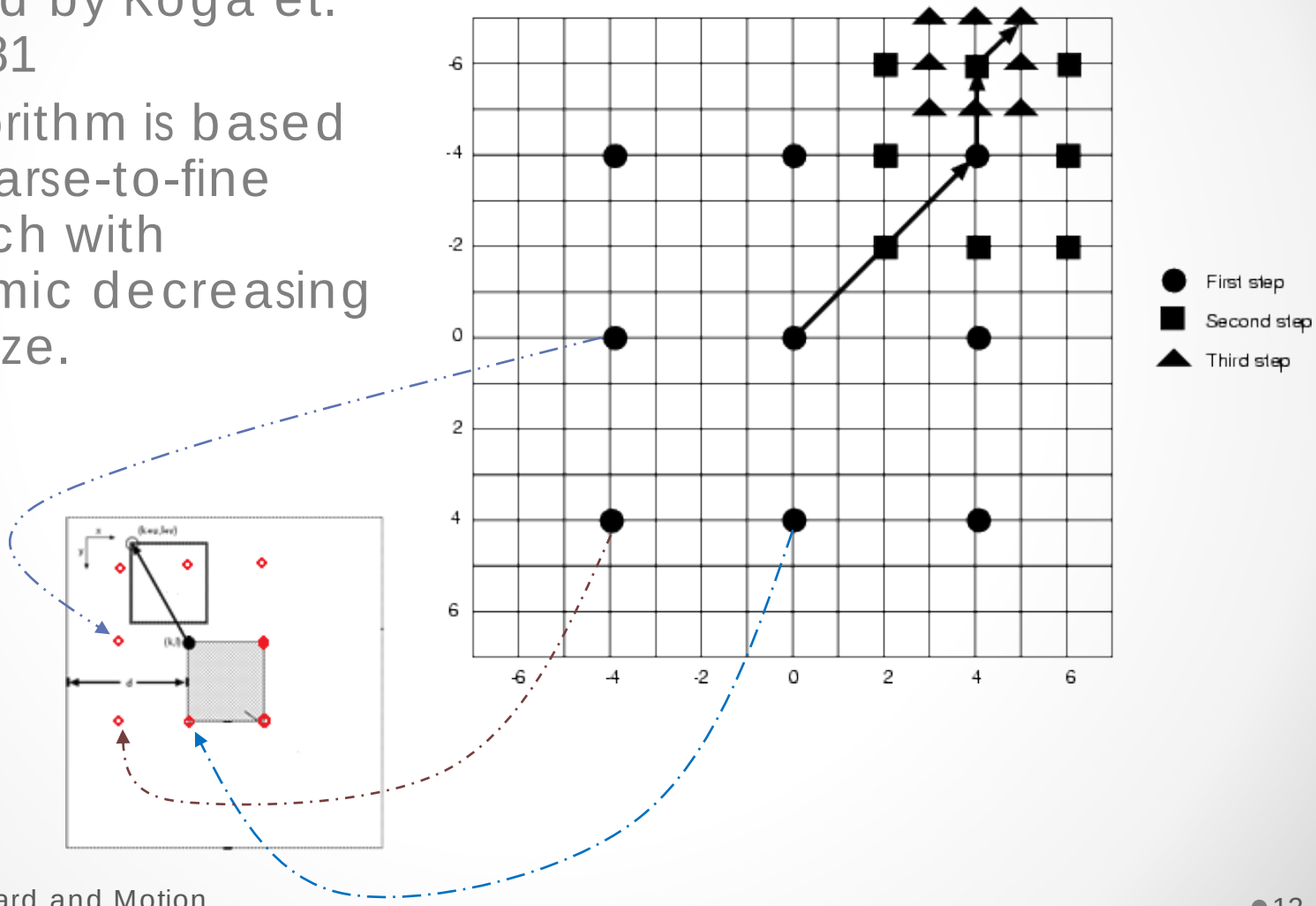


Multiple local minima within a search window



Three-Step Search

- Proposed by Koga et. al. in 1981
- This algorithm is based on a coarse-to-fine approach with logarithmic decreasing in step size.



Three-Step Search

- The initial step size is half of the maximum motion displacement ***d*** (i.e. where is the upper integer truncation function).
- For each step, nine checking points are matched and the minimum BDM point of that step is chosen as the starting center of the next step.
- For ***d***=7, the number of checking points required is (9+8+8)=25. For larger search window (i.e. larger ***d***), 3SS can be easily extended to n-steps using the same searching strategy with the number of checking points required equals to $[1+8*\text{ceil}(\log_2(d+1))]$.

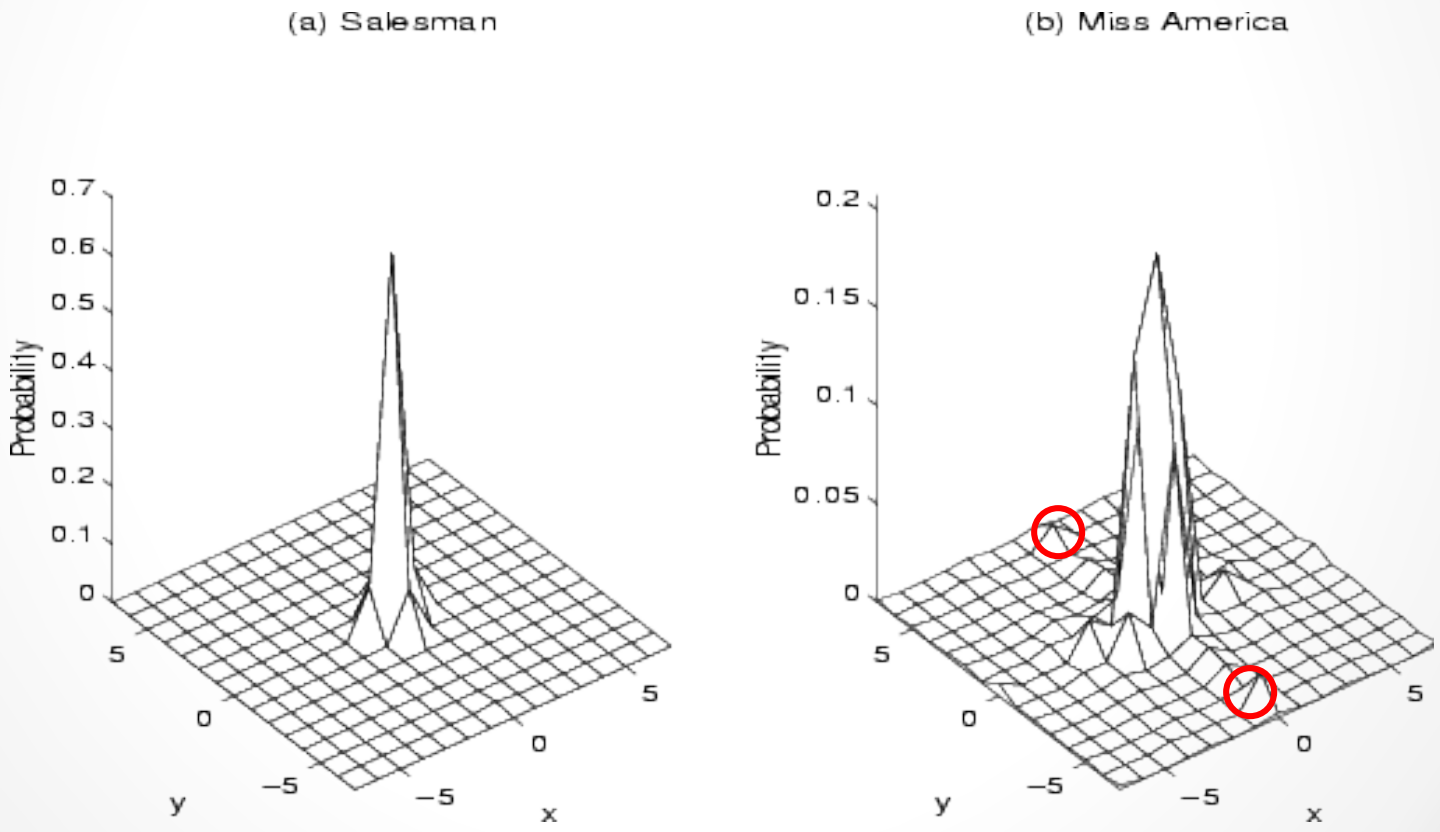
New Three-Step Search

- Proposed by Li, Zeng and Liou in 1994
- A modified version of the three-step search algorithm for searching small motion video sequences.
- The head and shoulder video sequences occurring the most in video conferencing applications.
- The motion vector distributions are highly center-biased.

close to origin

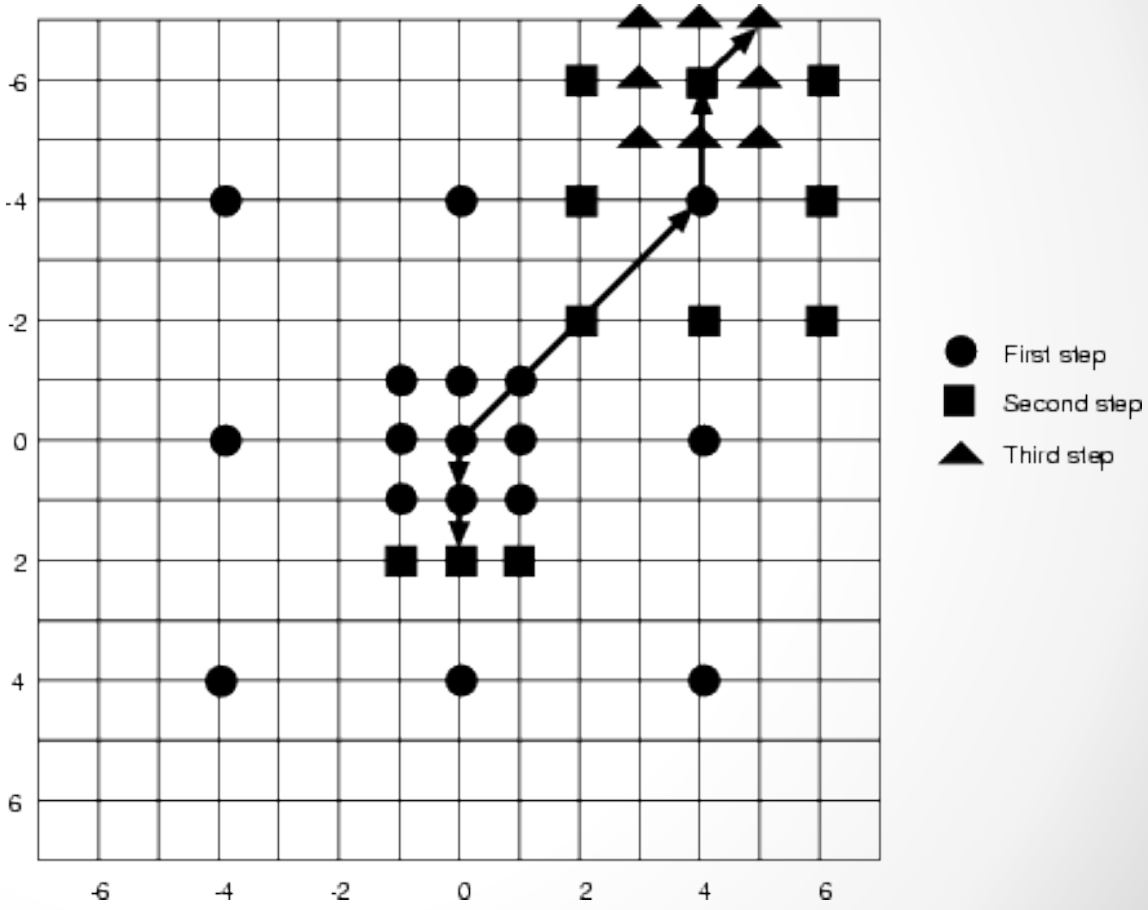
(0,0)

New Three-Step Search



New Three-Step Search

- Figure shows two search paths with ***d***=7.



New Three-Step Search

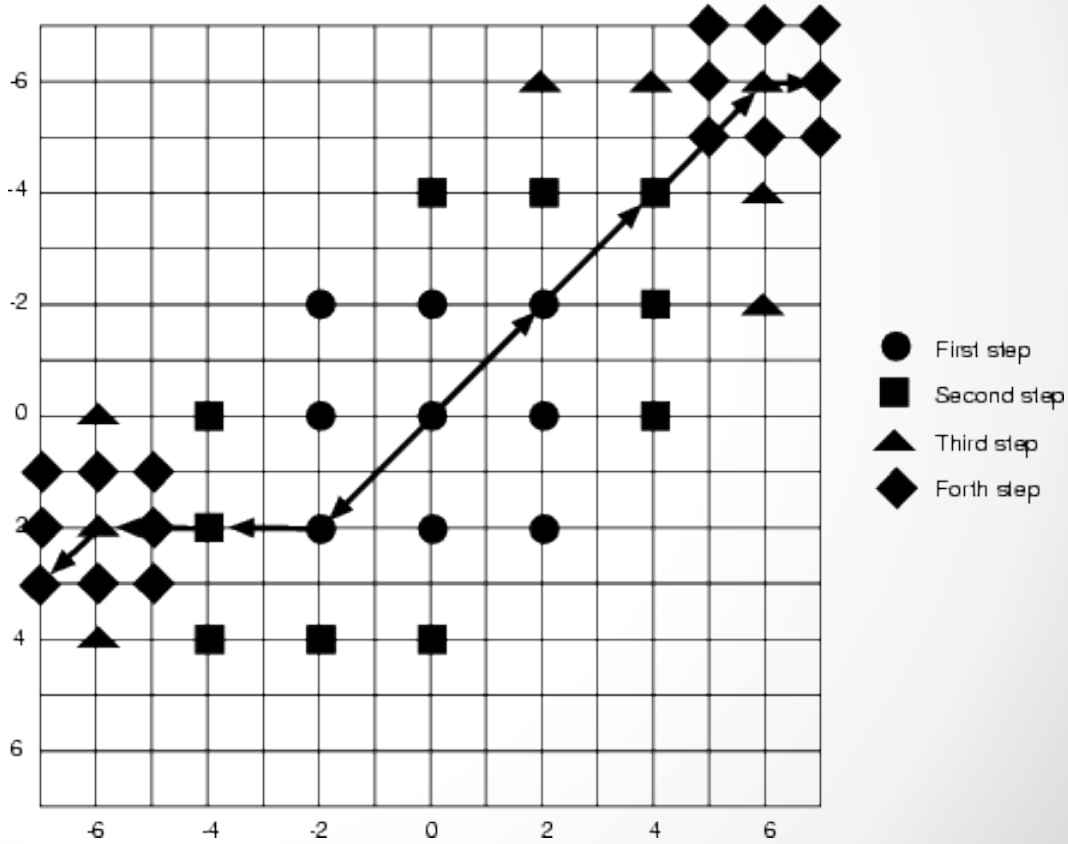
- The center path shows the case of searching small motion. In this case, the minimum BDM point of the first step is one of the 8 neighbor checking points.
- The search is halfway-stopped with matching three more neighbor checking points of the first step's minimum BDM point. The number of checking points required is (17+3)=20.
- The upper-right path shows the case of searching large motion. In this case, the minimum BDM point of the first step is one of the outer eight checking points.
- Then the searching procedures proceed the same as the 3SS algorithm. The number of checking points required is (17+8+8)=33.

Four-Step Search

- Proposed by L. M. Po and W. C. Ma in 1996
- Exploits the center-biased characteristics of the real world video sequences by using a smaller initial step size compared with 3SS.
- The initial step size is fourth of the maximum motion displacement ***d*** (i.e. ceil(d/4)).
- Due to the smaller initial step size, the 4SS algorithm needs four searching steps to reach the boundary of a search window with ***d***=7.
- Same as the small motion case in the N3SS algorithm, the 4SS algorithm also uses a halfway-stop technique in its second and third step's search.

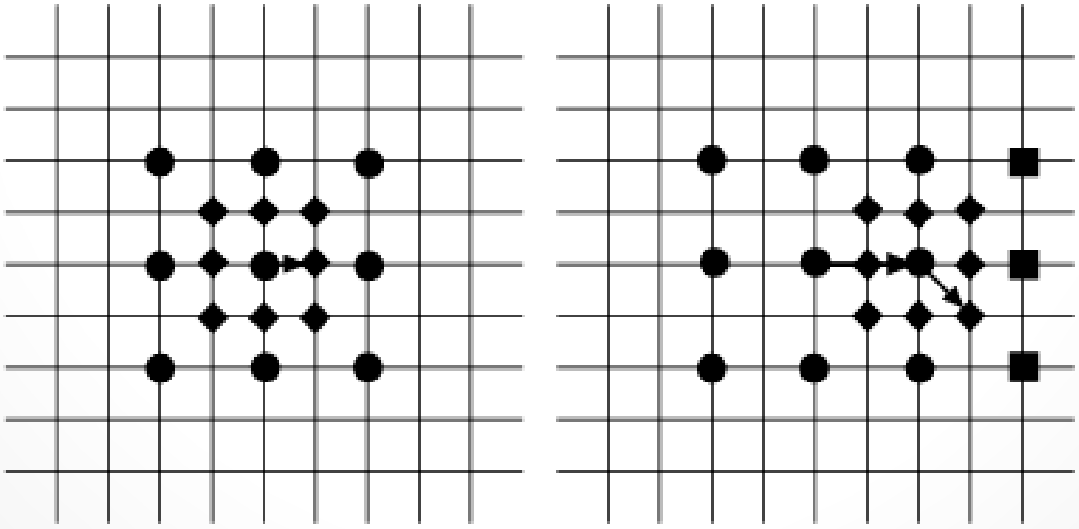
Four-Step Search

- Figure shows two search paths of 4SS for searching large motion.
- For the lower-left path, it requires (9+5+3+8)=25 checking points.
- For the upper-right path, it requires (9+5+5+8)=27 checking points that is the worse case of the algorithm for ***d***=7.



Four-Step Search

- Figure shows two search paths of 4SS for searching small motion. For the left path, it requires (9+8)=17 checking points. For the right path, it requires (9+3+8)=20 checking points.

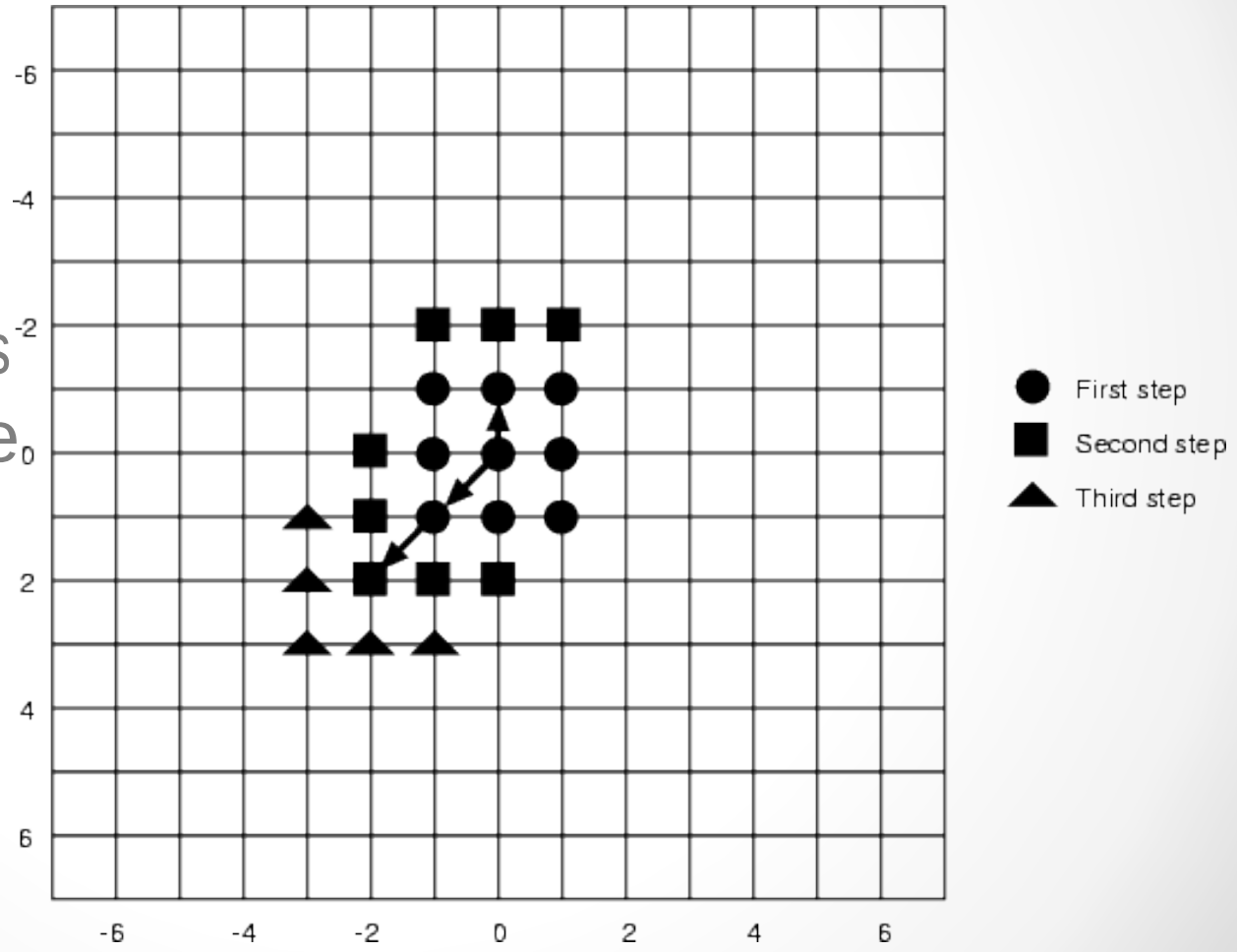


Block-based Gradient Descent Search

- Proposed by L. K. Liu and E. Feig in 1996
- This algorithm uses a very center-biased search patterns of 9 checking points in each step with step size of one.
- It does not restrict the number of searching steps but it is stopped when the minimum checking point of the current step is the center one or it is reached the search window boundary.
- There are also overlapped checking points between adjacent steps. The BBGDS algorithm performs better in searching small motions.

Block-based Gradient Descent Search

- Two small motion search paths of BBGDS are shown.



Performance comparisons

Performance comparisons for CCIR601 sequence “Football” – large motion

BMA	Window size d=7			Window size d=15		
	MSE	SearchPT	SpeedUp	MSE	SearchPT	SpeedUp
FS	257.83	225.00	1.00	145.06	961.00	1.00
3SS	276.22	25.00	9.00	186.04	33.00	29.12
4SS	289.72	20.56	10.94	193.54	29.80	32.25
N3SS	276.70	23.81	9.45	192.82	25.68	37.42

Performance comparisons

Performance comparisons for CIF sequence “Susie” – small motion

BMA	Window size w=7			Window size w=15		
	MSE	SearchPT	SpeedUp	MSE	SearchPT	SpeedUp
FS	11.89	225.00	1.00	11.82	961.00	1.00
3SS	12.52	25.00	9.00	12.71	33.00	29.12
4SS	12.29	17.37	12.95	12.37	25.51	37.67
N3SS	11.93	17.96	12.53	12.00	17.92	53.63

Half-Pel Accurate Motion Estimation

- In the real world video sequences, the true frame-to-frame displacements are unrelated to the integer pixel sampling grids.
- The motion prediction accuracy is limited to integer-pel accuracy.
- The motion prediction accuracy can be improved by half-pel accurate motion estimation. That is to determine motion vectors with half-pel accuracy.
- The half-pel accurate motion estimation could achieve noticeable prediction performance gain in the motion estimation process.
- Many video coding standards such as MPEG-1/2 and H.262/263 employ it to improve their compression efficiency.

Half-Pel Accurate Motion Estimation

- Half-pel accurate motion vectors can be found by interpolating (double the frame size) the current frame and reference frame by a factor of two and then using any of the BMAs for motion estimation.
- However, this method uses excessive storage requirements.
- An alternate approach is usually preferred and it is as follows:

Step 1.

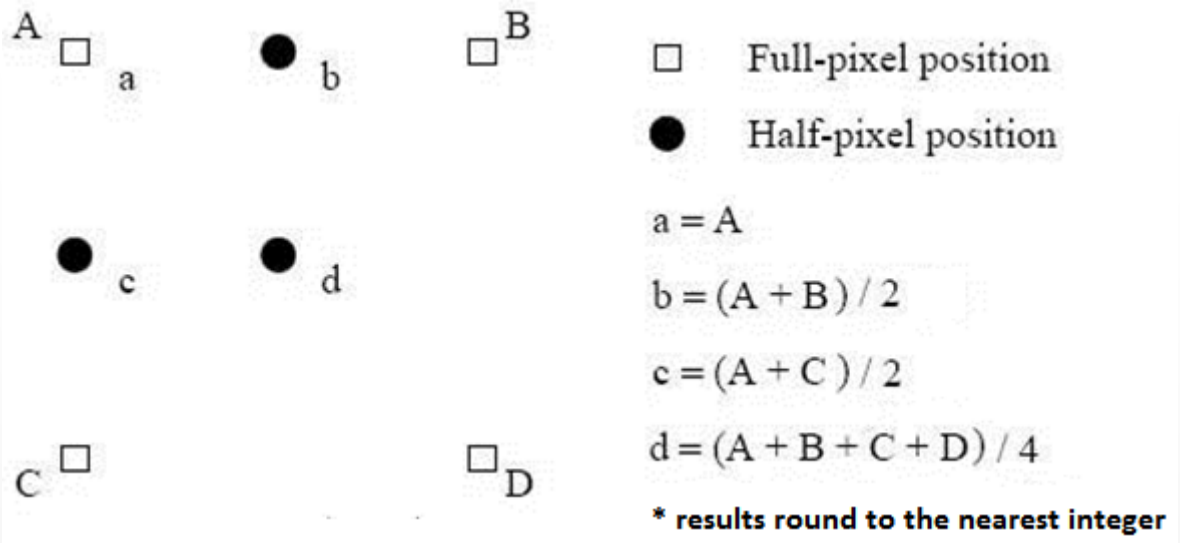
Find the motion vector with integer-pel accuracy using the BMAs to obtain a motion vector (u,v).

Step 2.

Refine the motion vector (u,v) to half-pel accuracy as follows:

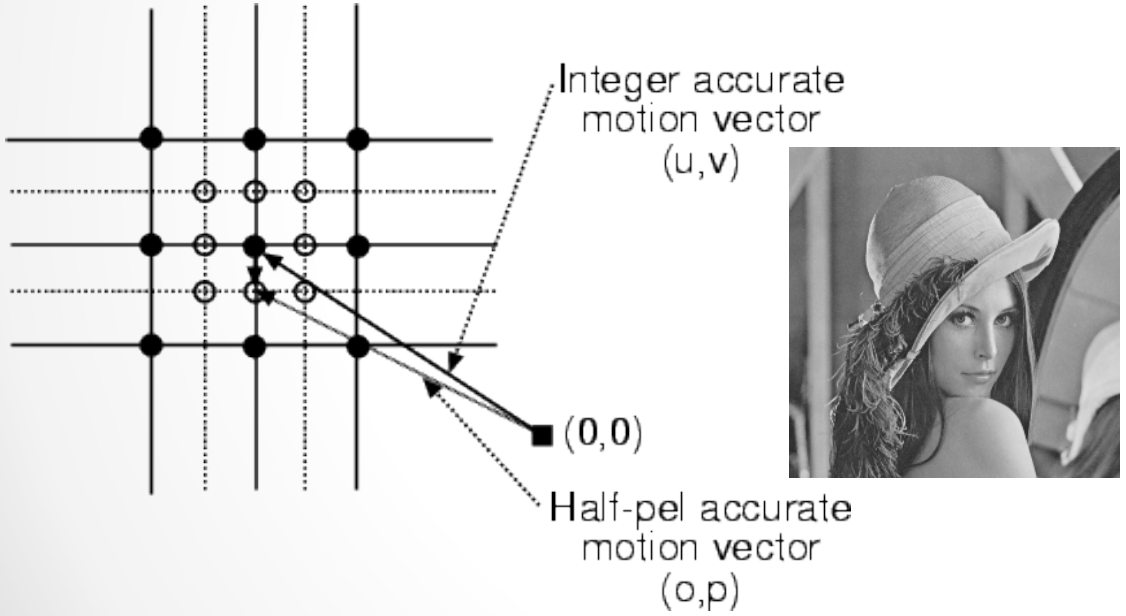
Half-Pel Accurate Motion Estimation

- A) Obtain the 8 surrounding half-pel blocks using bilinear interpolation as shown in figure.

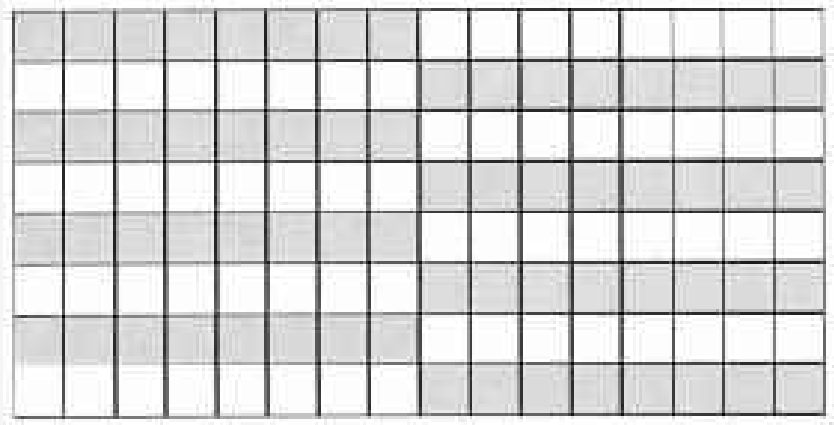


Half-Pel Accurate Motion Estimation

- B) Compute the BDMs of all the 8 half-pel checking points and compared them with the one of (u,v). The minimum one is the output half-pel accurate motion vector as shown in figure.



Pseudo-checkerboard pattern



- Instead of calculating the BDMs over all pixels in the macroblock, we use a pseudo-checkerboard pattern.

- For a 16x8 field macroblock, this checkerboard consists of calculating the BDMs over the 8 pixels on the left-half side of the even numbered lines and 8 pixels on the right-half side of the odd number line.
- The BDMs are calculated only over 64 pixels within the 16x8 field macroblock.