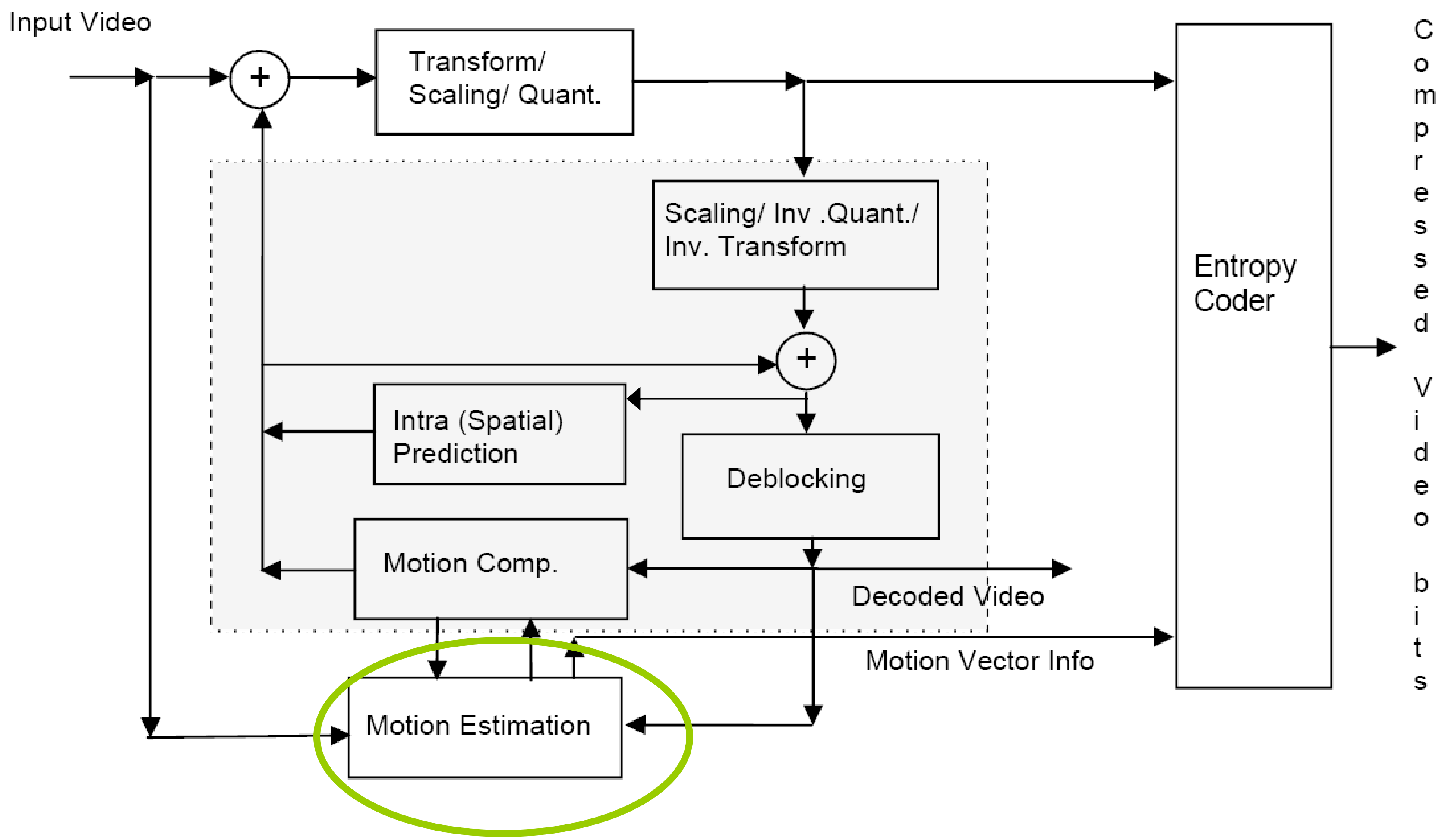
Motion Estimation and Intra Frame Prediction in H.264/AVC Encoder

Rahul Vanam
University of Washington

Motion Estimation

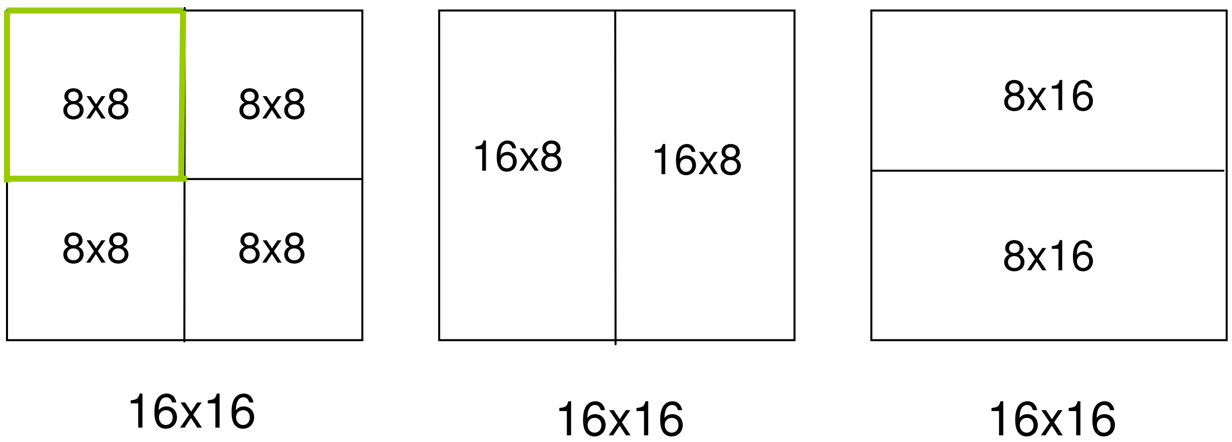


H.264/AVC Encoder [2]

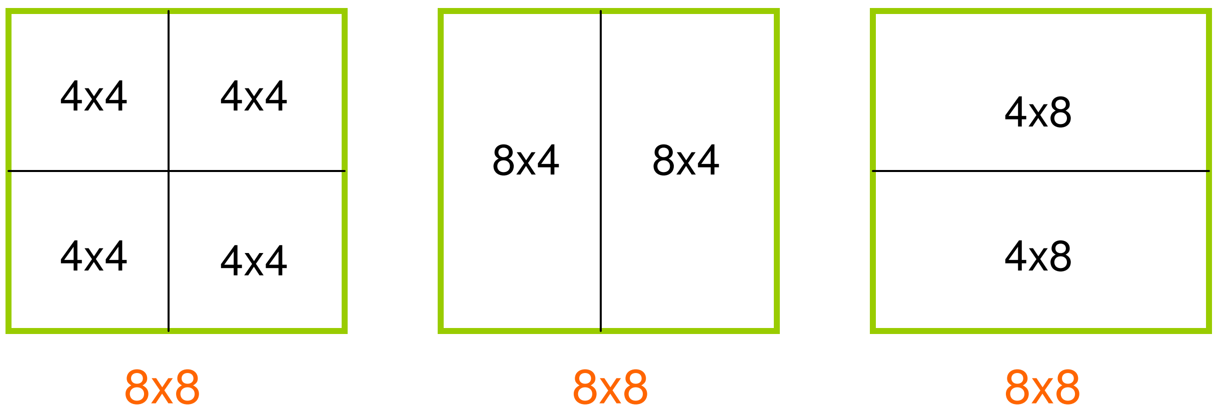
Motion Estimation

- H.264 does block based coding.
- Each frame is divided into blocks of 16x16 pixels called macroblocks (MB).
- Each MB can be encoded using blocks of pixels that are already encoded within the current frame - Intra frame coding.
- MBs can be coded using blocks of pixels in previous or future encoded frames - Inter frame coding.
- The process of finding a match of pixel blocks in inter frame coding is called Motion Estimation.

Macroblock Partitions



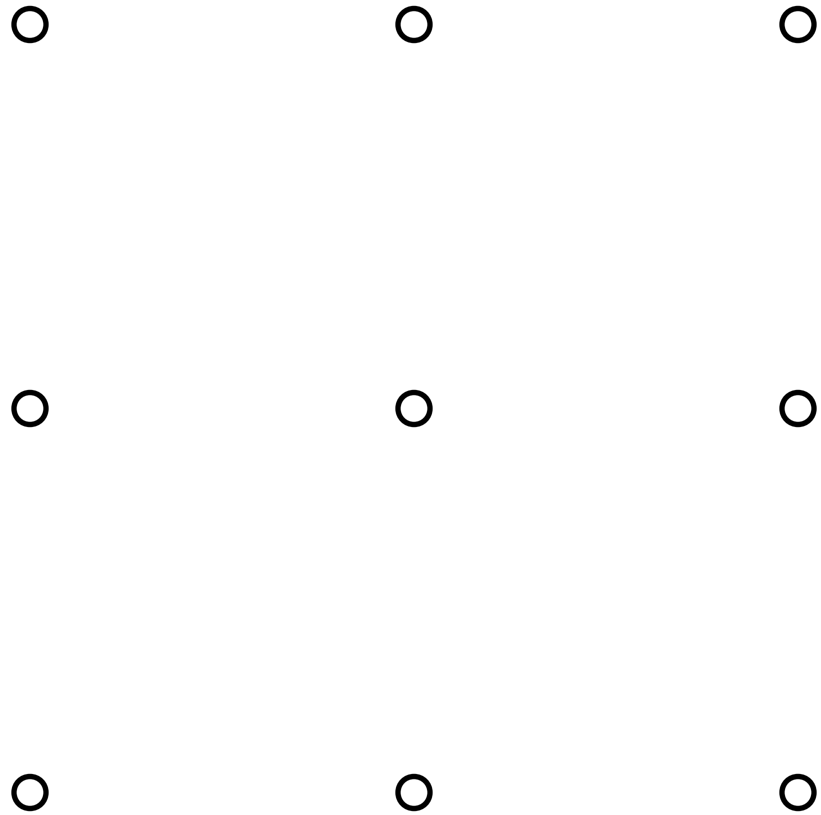
16x16 blocks can be broken into blocks of sizes 8x8, 16x8, or 8x16.



8x8 blocks can be broken into blocks of sizes 4x4, 4x8, or 8x4.

Sub-pixel Motion Estimation

Integer pixel search

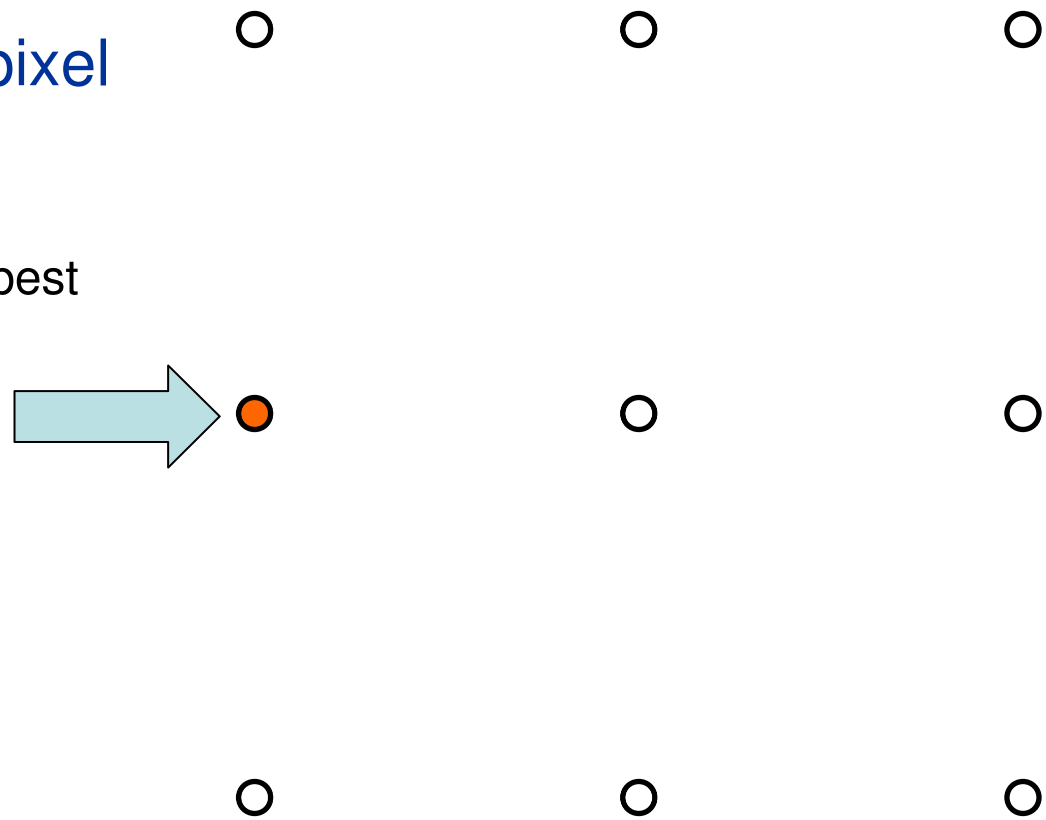


- Compute the block distortion at each pixel position within the search window

Sub-pixel Motion Estimation

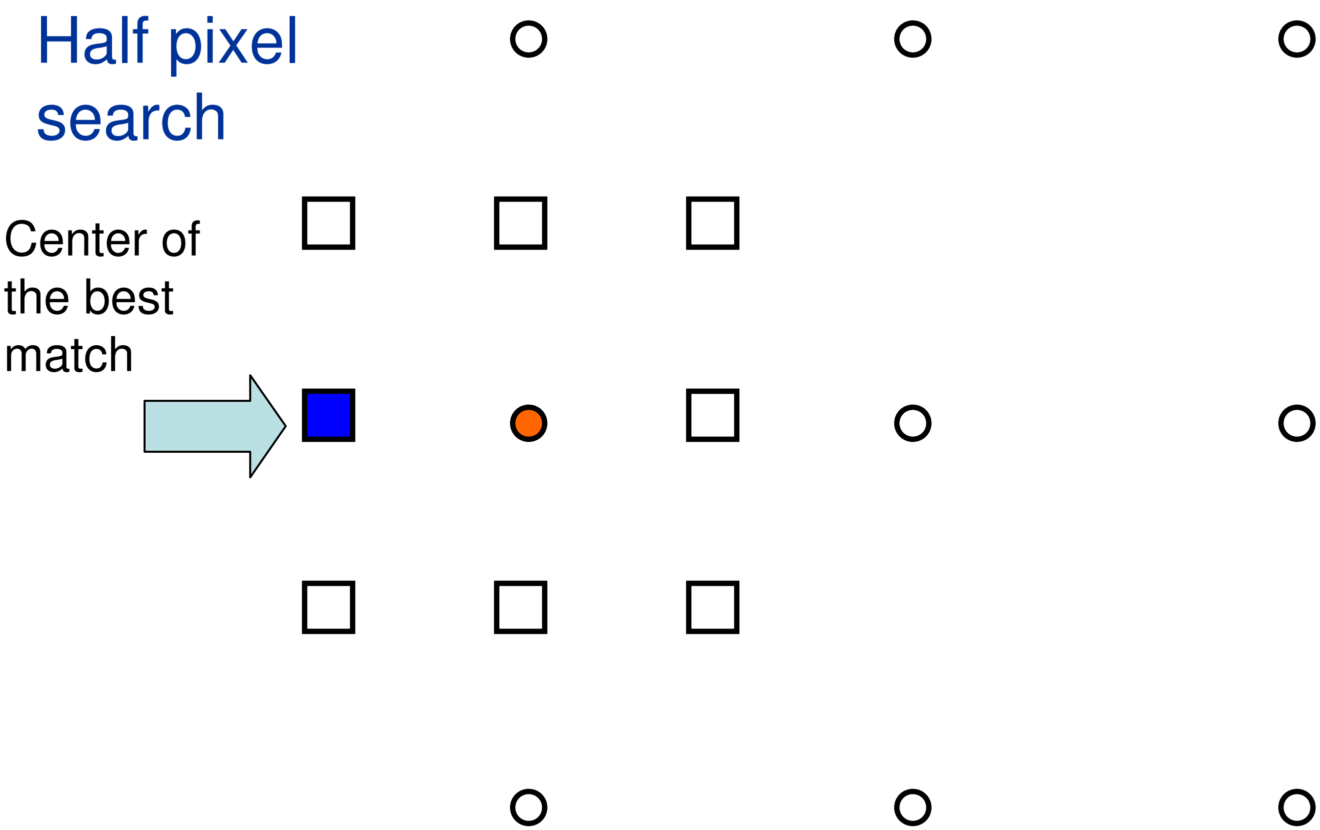
Integer pixel search

Center of best match



• Find the position corresponding to the minimum block distortion

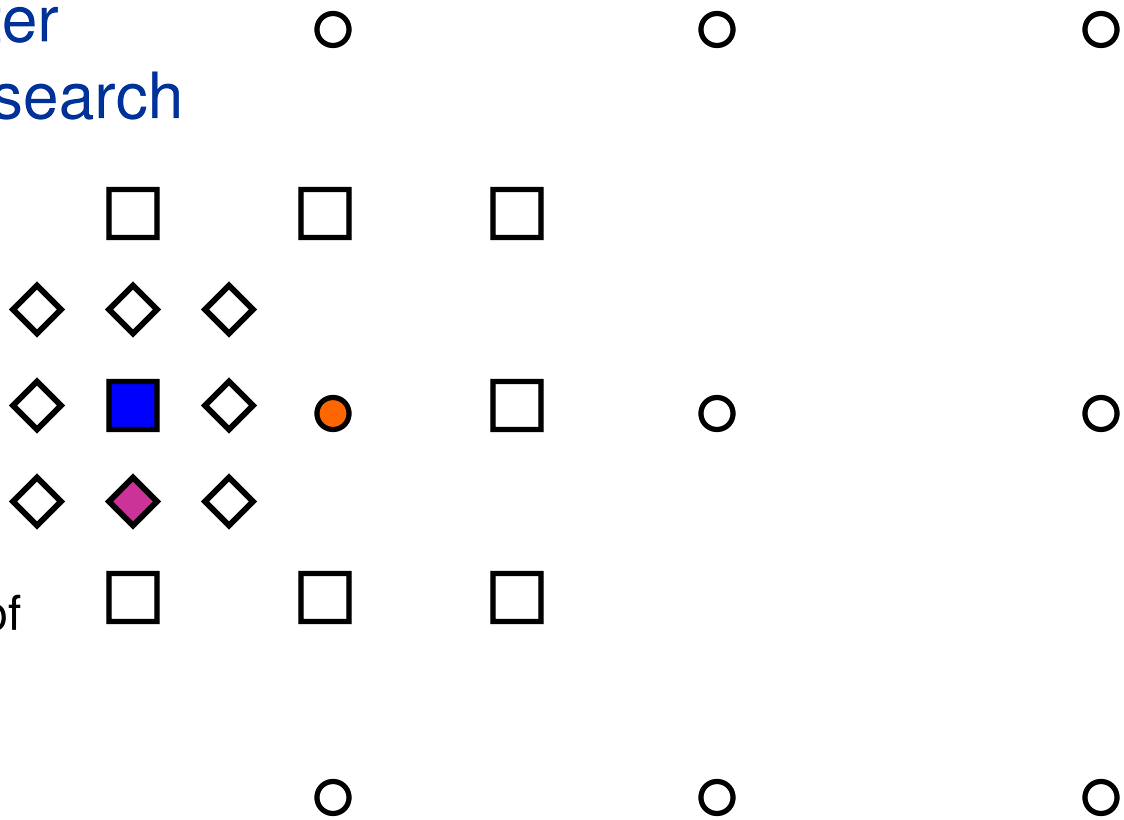
Sub-pixel Motion Estimation



• Half pixel motion estimation is then done where the best match was found in the integer pixel search step.

Sub-pixel Motion Estimation

Quarter pixel search



Center of the best match

• Finally, quarter pixel motion estimation is done where the best match was found from the half pixel search step, giving us the final motion vector.

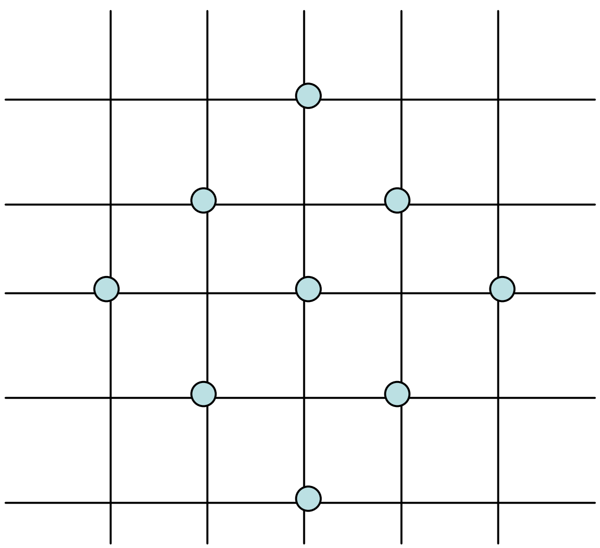
Motion Estimation

- Motion estimation is computationally expensive since
  - search is done at every pixel position
  - over different reference frames
- There are several different fast integer search methods – diamond search, hexagon search, Simplified Uneven Multihexagon search (UMH), etc.

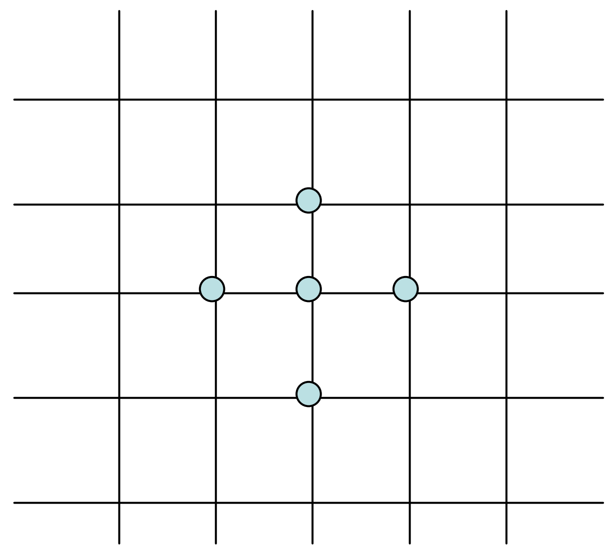
Diamond Search Algorithm

S. Zhu and K. K. Ma, IEEE CSVT 2000

- It uses large diamond search pattern of radius 2 and small diamond search pattern of radius 1.

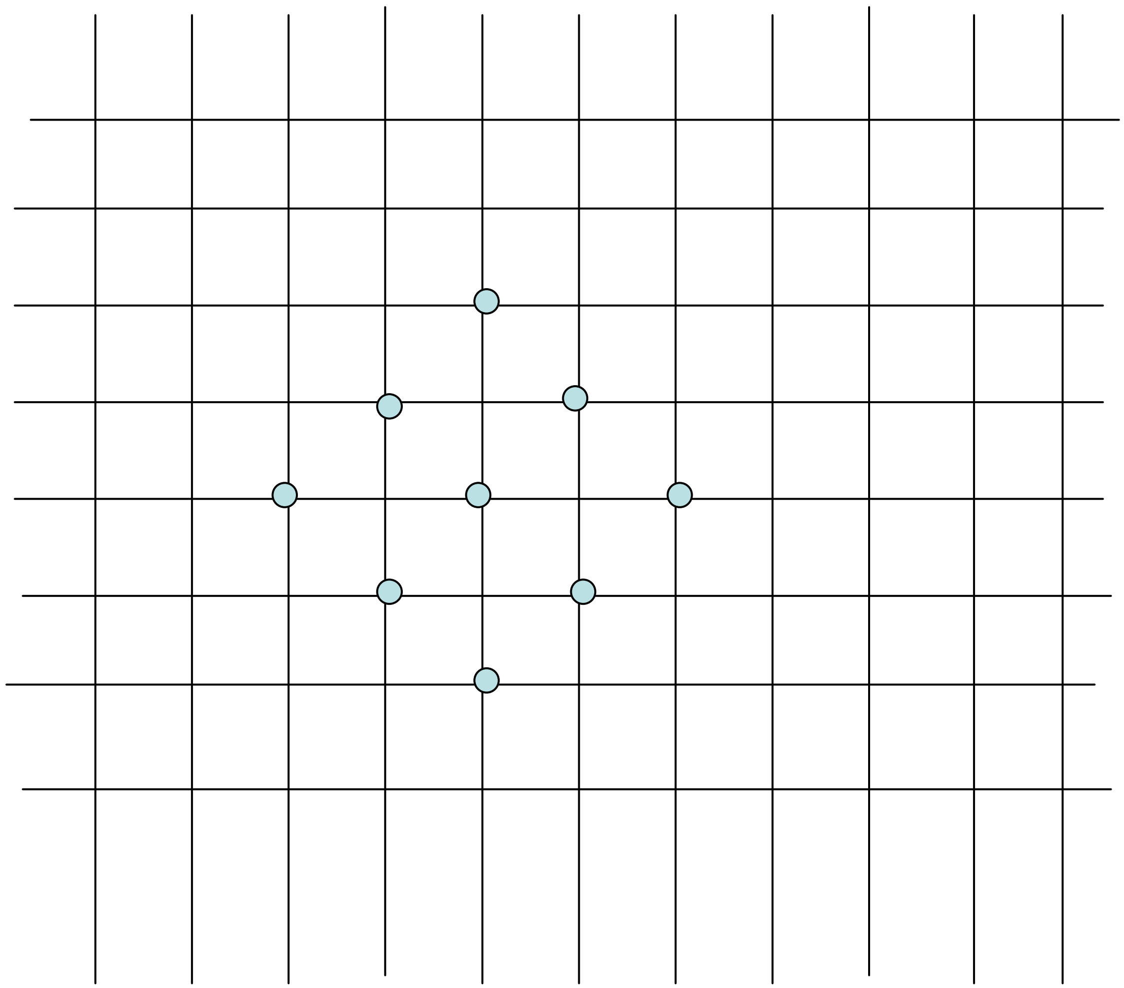


Large diamond search pattern



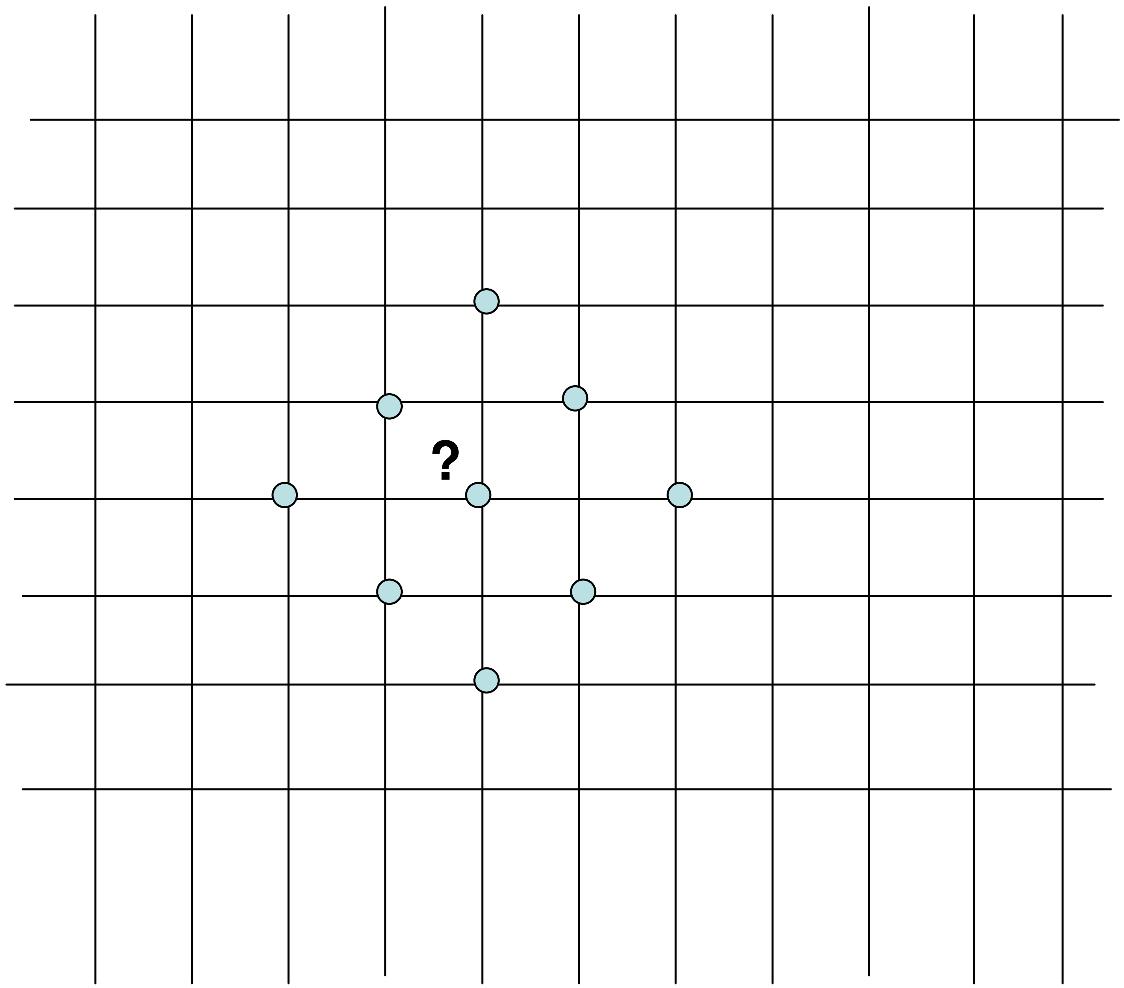
Small diamond search pattern

Diamond Search Algorithm



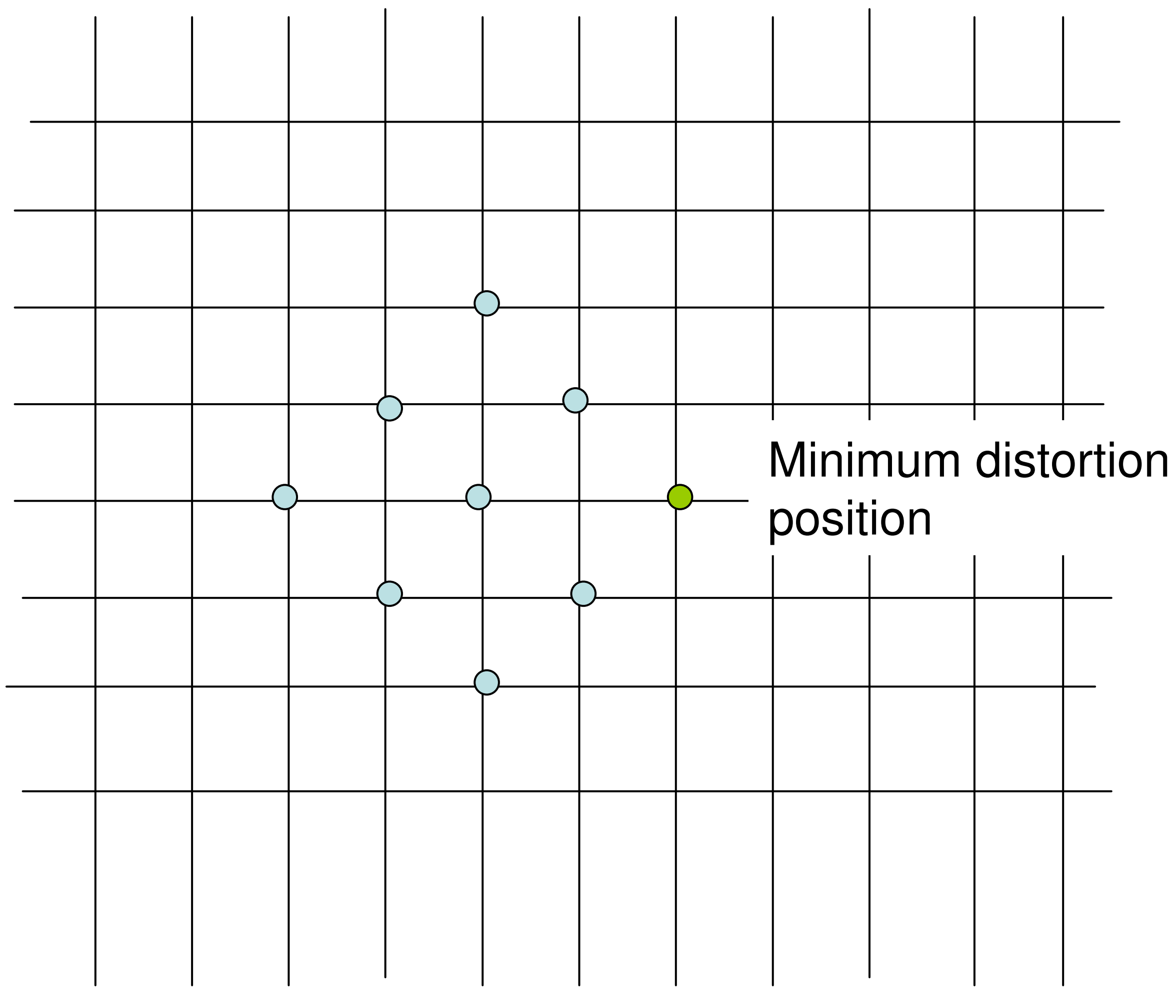
Apply large diamond to the center of the search window

Diamond Search Algorithm

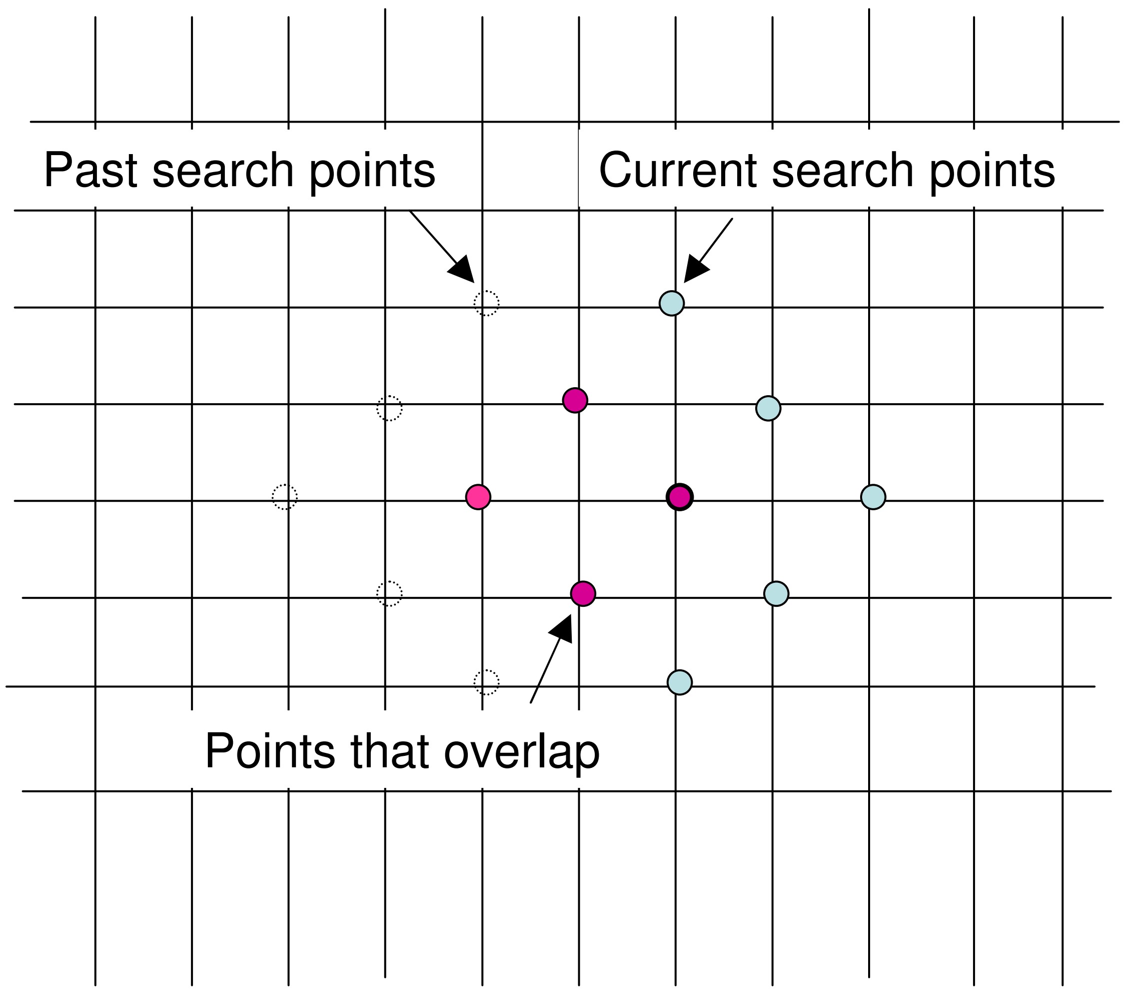


Compute the block distortions corresponding to all positions and check if the center position has the minimum distortion.

Diamond Search Algorithm



Diamond Search Algorithm



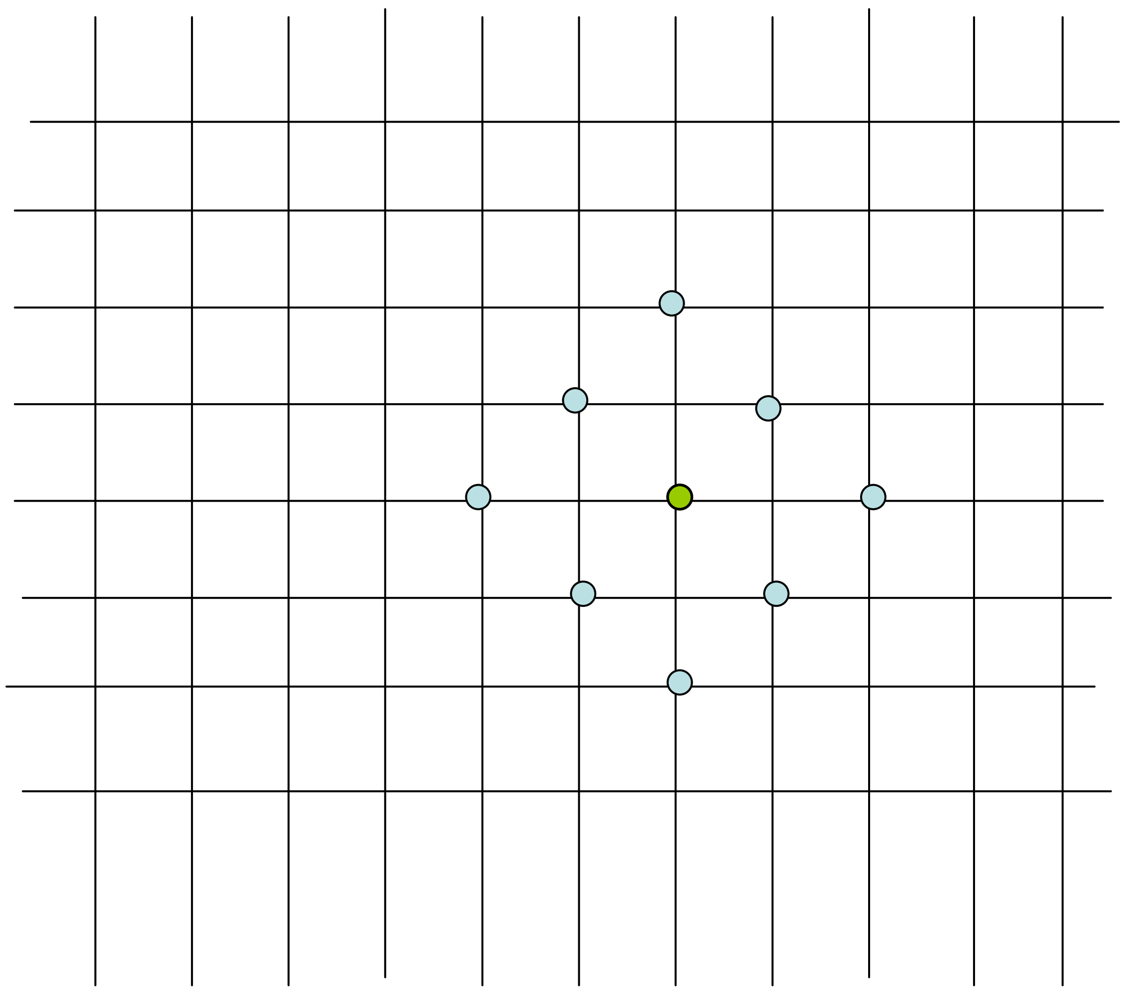
The minimum point in the previous step is the center position of this step.

Apply large diamond to the new center position.

Find the new minimum block distortion.

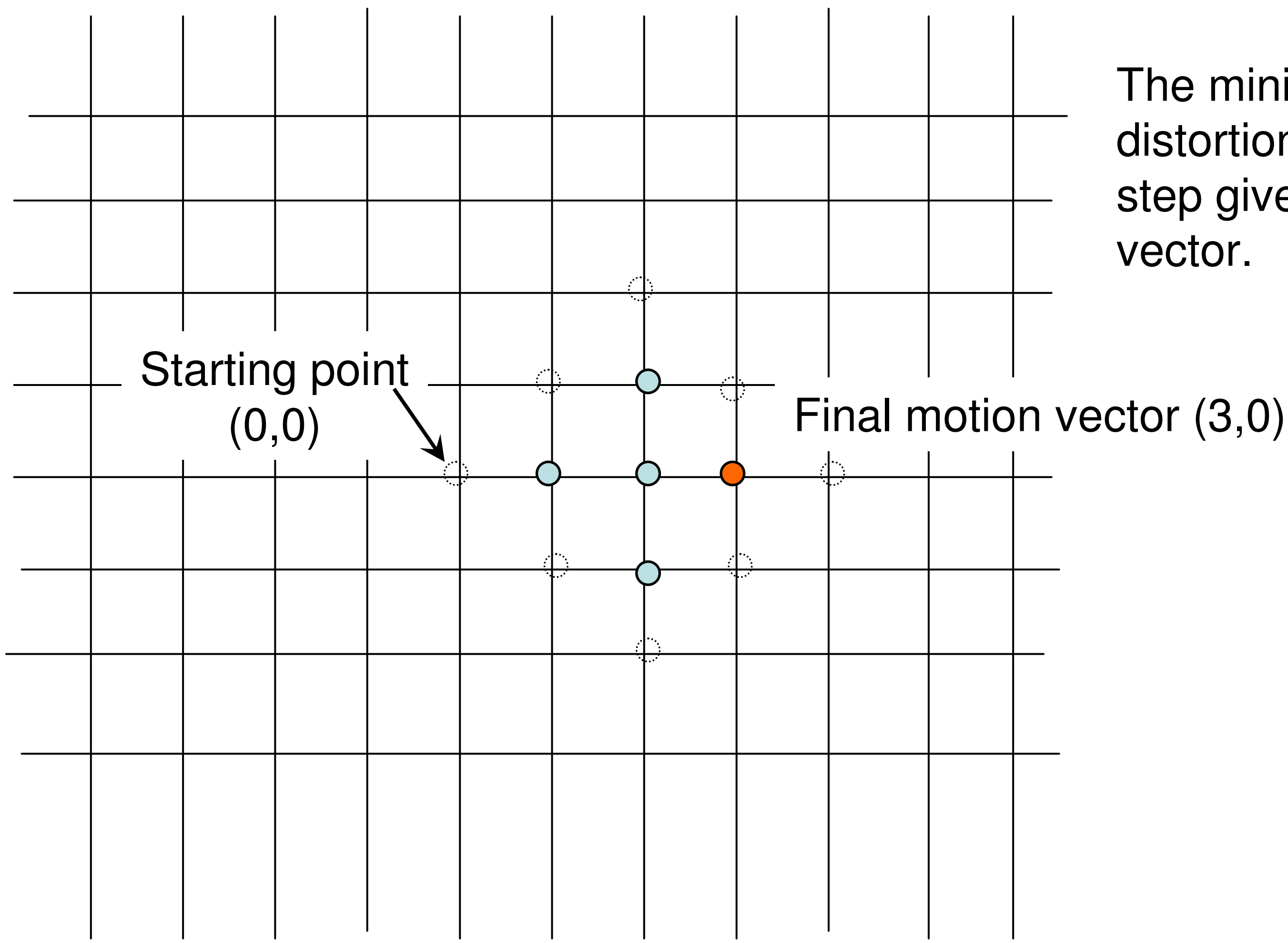
If the center is not the minimum, move the center to the minimum point and reapply the large diamond pattern.

Diamond Search Algorithm



If the center is the minimum block distortion position, then apply a small diamond.

Diamond Search Algorithm

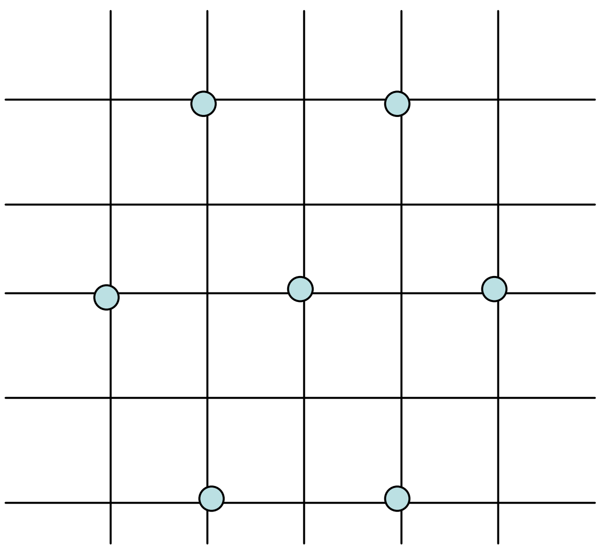


The minimum block distortion position in this step gives the final motion vector.

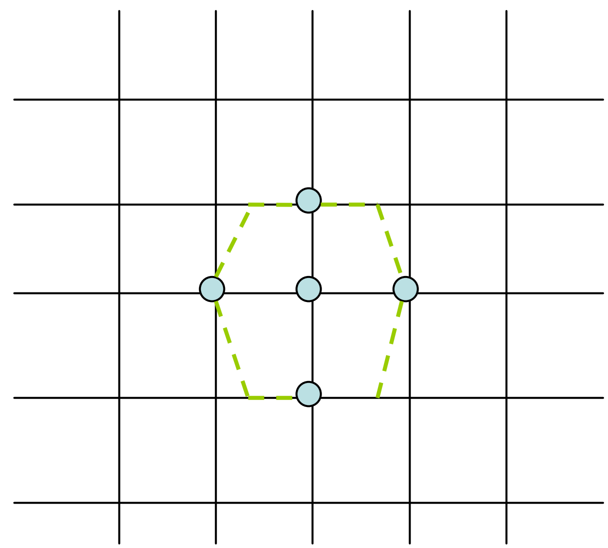
Hexagon Search Algorithm

C. Zhu, X. Lin and L-P. Chau, IEEE CSVT, 2002

- It consists of large hexagon pattern of radius 2 in horizontal and vertical direction, and small hexagon (or diamond) pattern of radius 1.
- The search approach is similar to Diamond search.

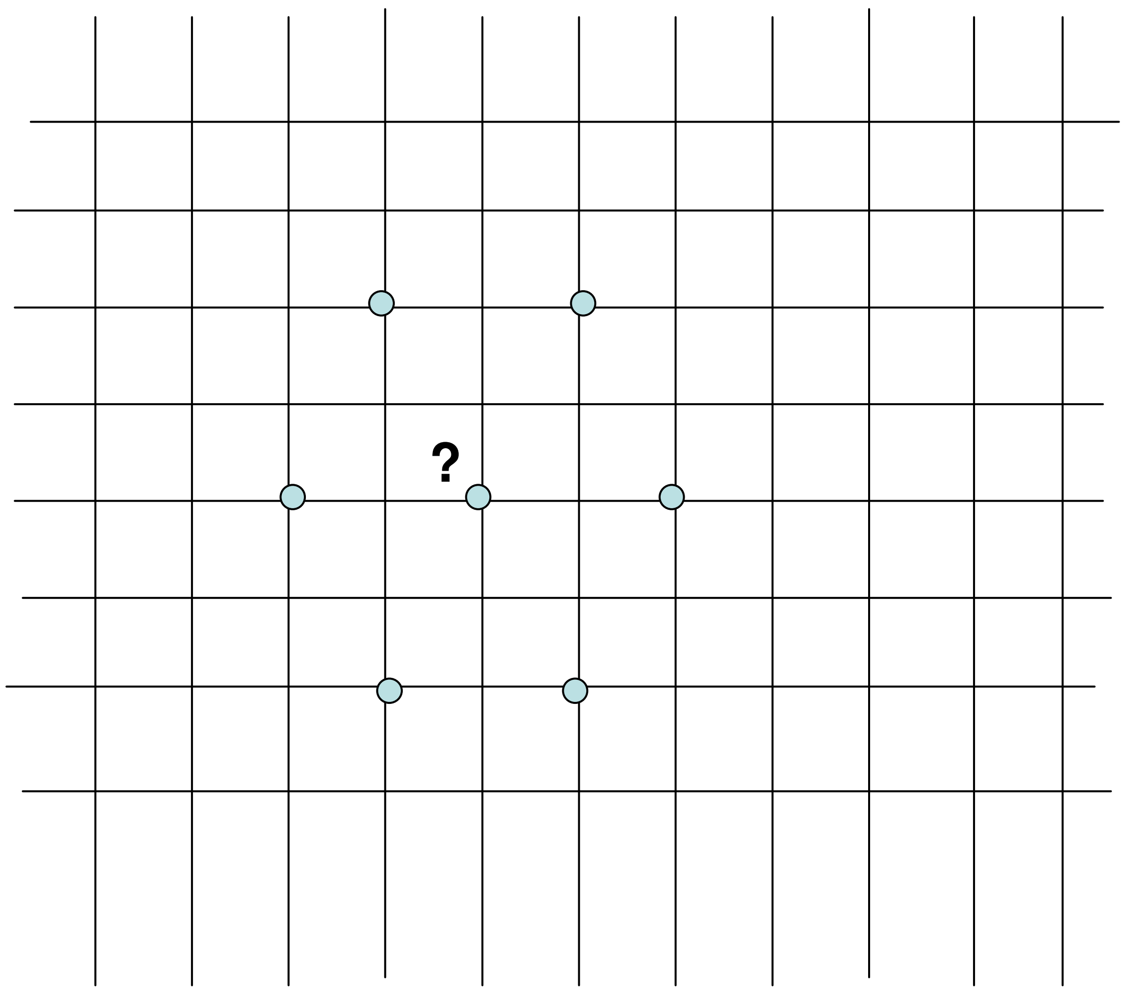


Large hexagon pattern



Small hexagon pattern

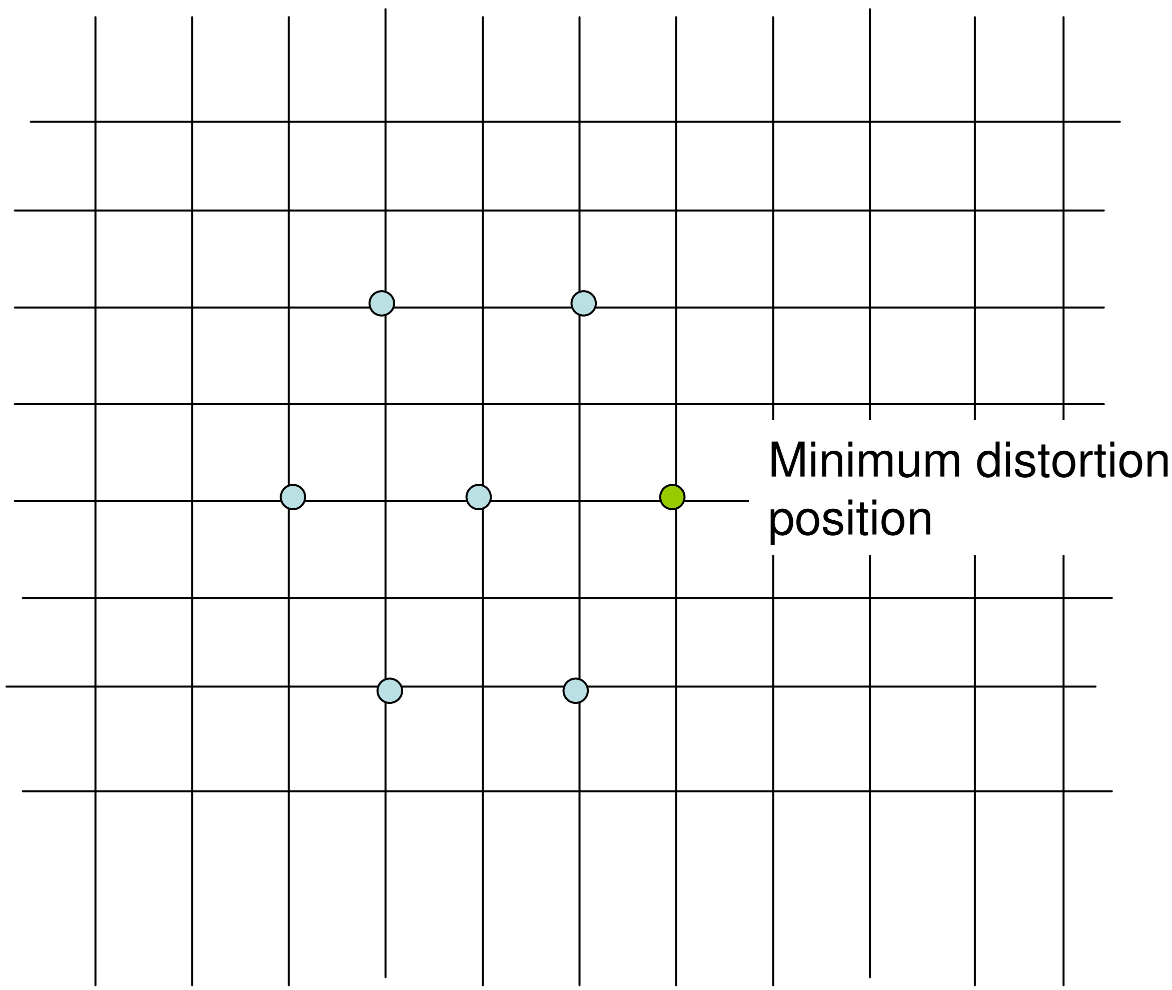
Hexagon Search Algorithm



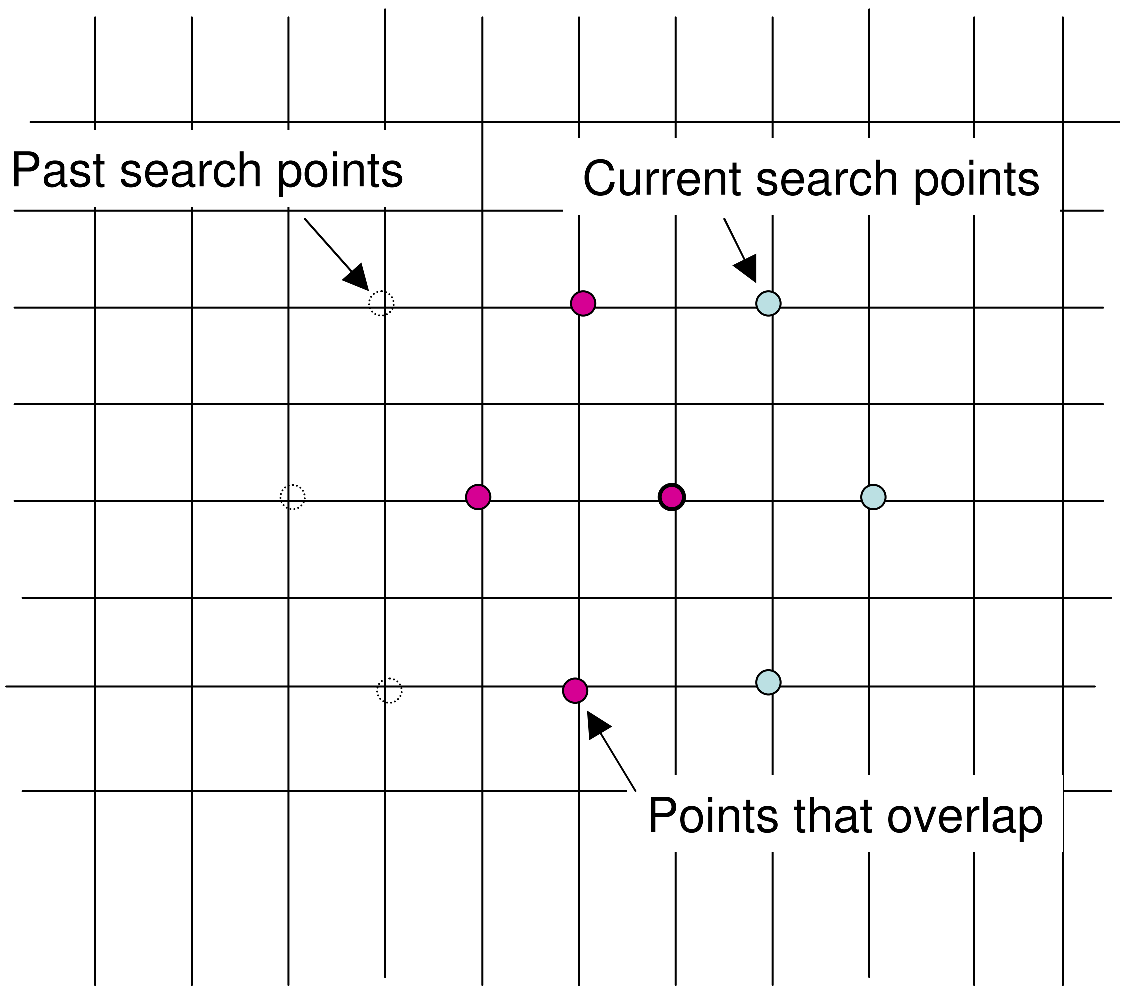
Apply large hexagon to the center of the search window.

Compute the block distortions corresponding to all positions and check if the center position has the minimum distortion.

Hexagon Search Algorithm



Hexagon Search Algorithm



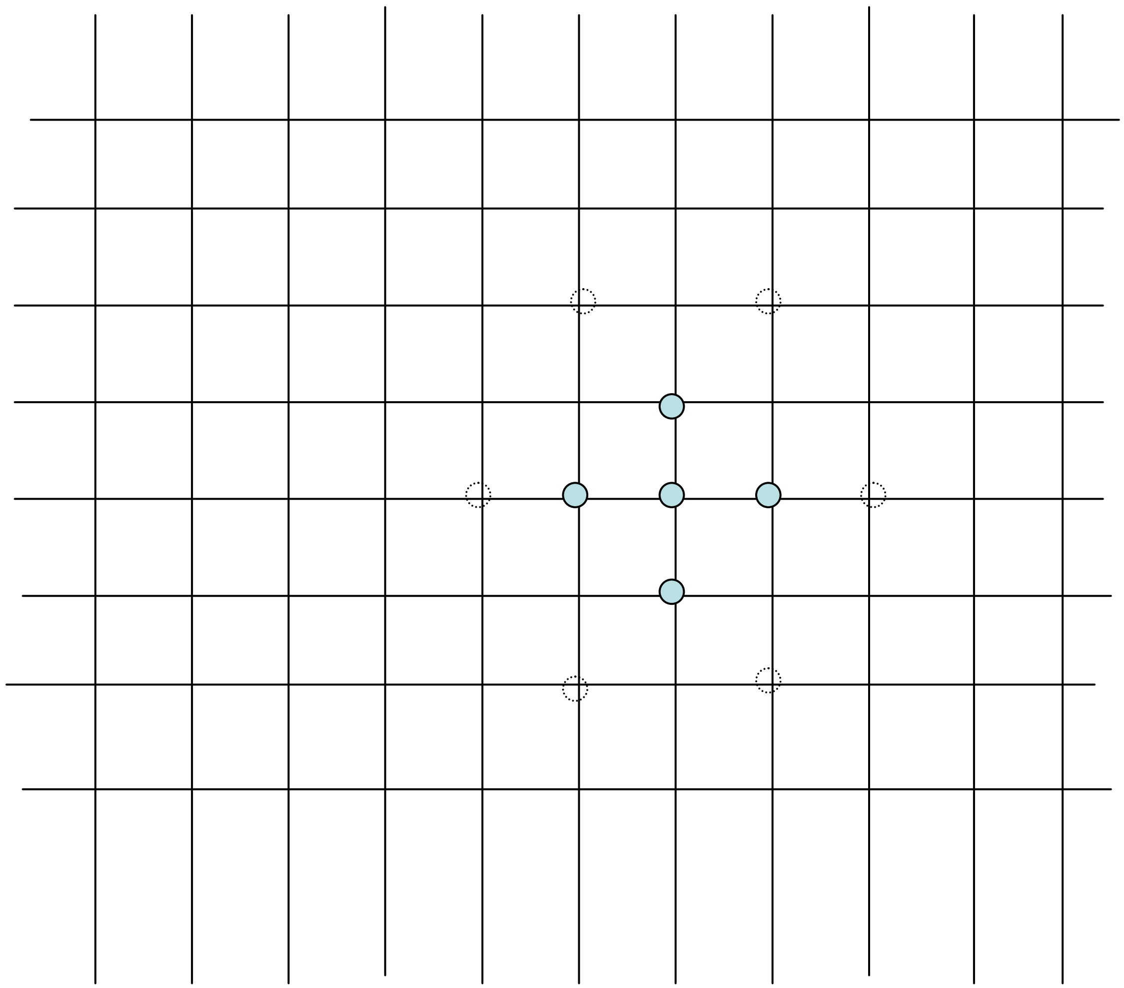
The minimum point in the previous step is the center position of this step.

Apply large hexagon to the new center position.

Find the new minimum block distortion.

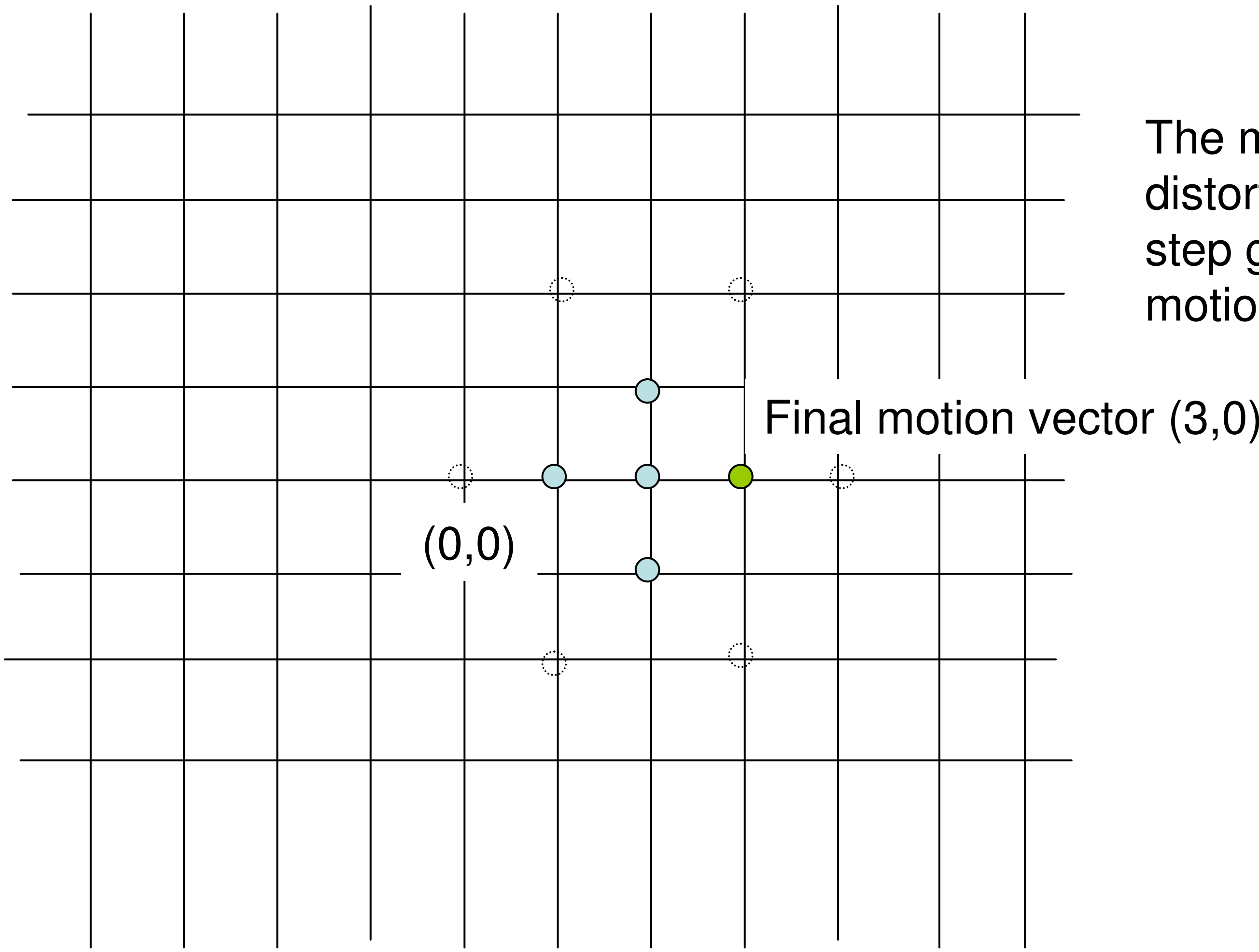
If the center is not the minimum, move the center to the minimum point and reapply the large hexagon pattern.

Hexagon Search Algorithm



If the center is the minimum block distortion position, then apply a small hexagon.

Hexagon Search Algorithm



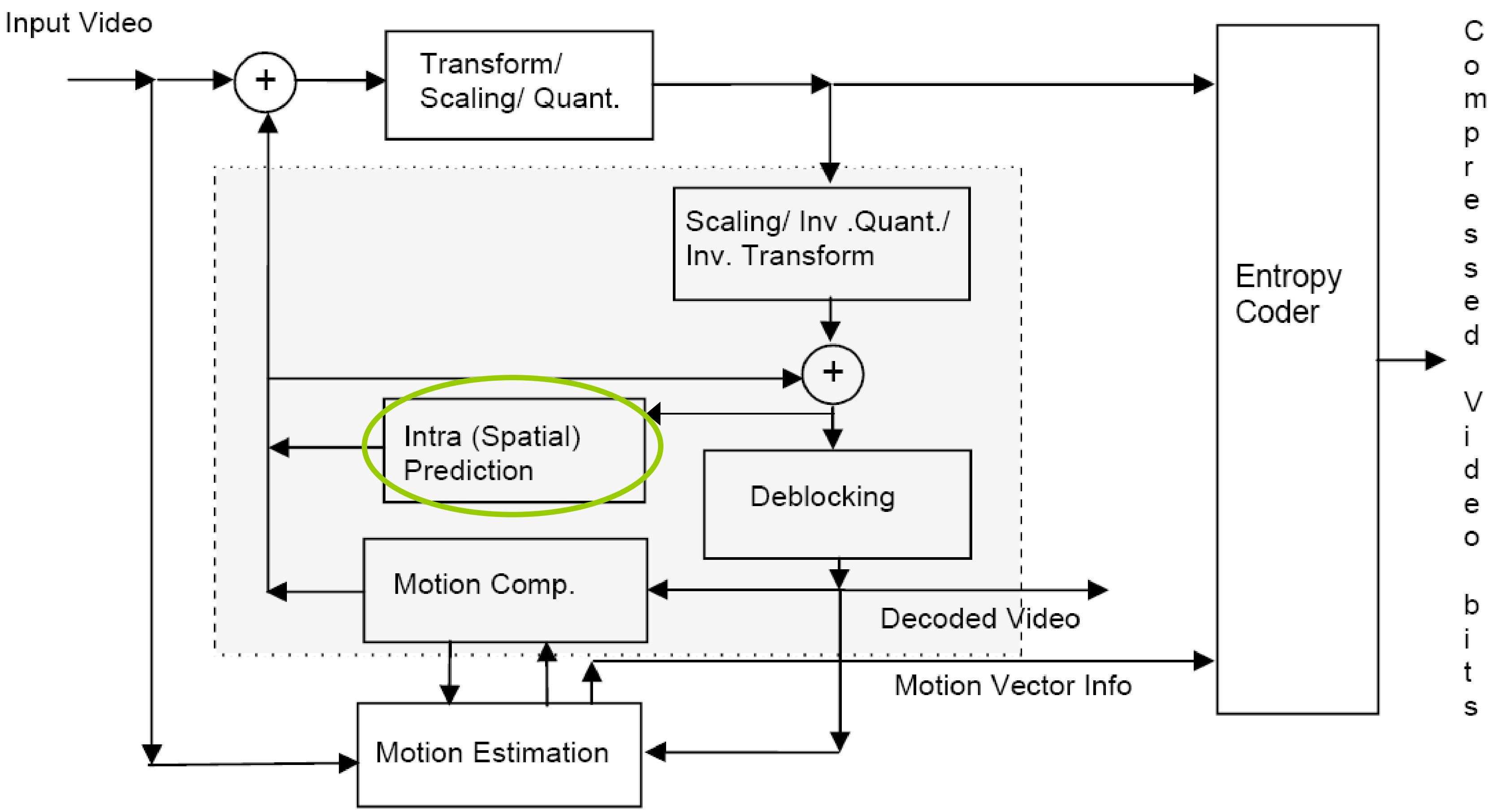
The minimum block distortion position in this step gives the final motion vector.

Hexagon Search Algorithm (HEXS)

Comparison with Diamond search (DS)

- HEXS uses fewer search points compared to diamond search (DS). In our example, HEXS requires 14 search points while DS requires 18 search points.
- HEXS gives higher savings in searches for larger motion vectors.
- HEXS results in slightly higher mean distortion compared to DS.

Intra-Frame Prediction



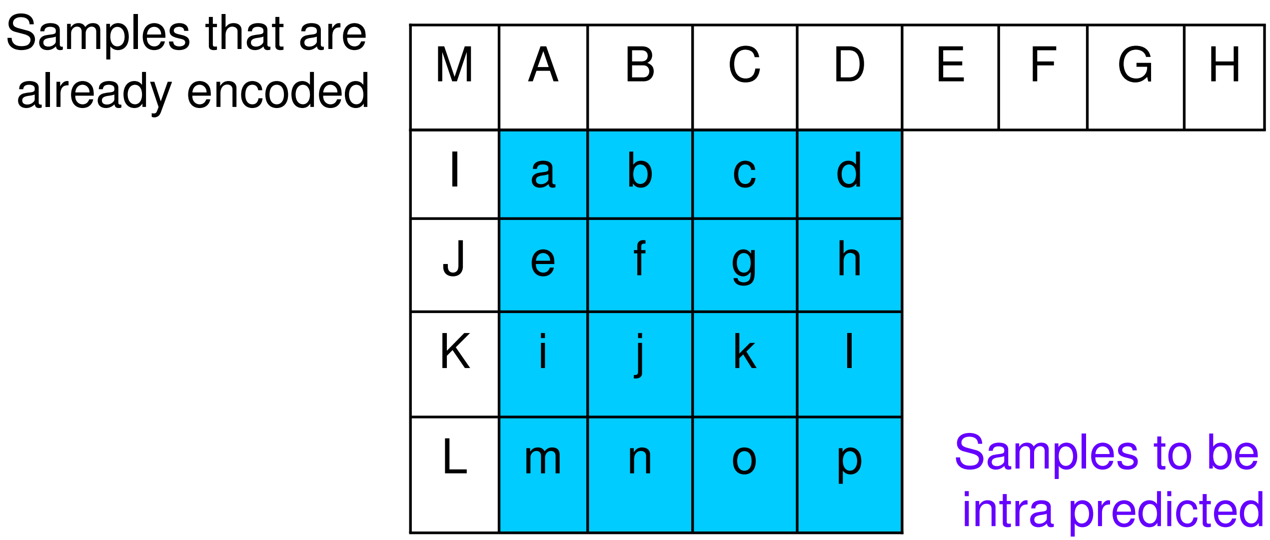
H.264/AVC Encoder [2]

Intra-Frame Prediction

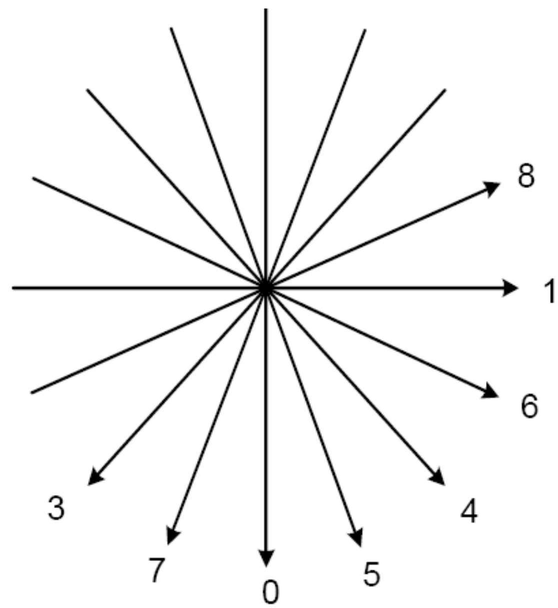
- Intra modes for Luma samples
  - 9 modes for 4x4 blocks
  - 4 modes for 8x8 blocks
- Intra modes for Chroma samples
  - 4 modes for 8x8 blocks

Intra Luma Prediction for 4x4 blocks

- Samples a, b, …, p are predicted from samples A,…, M that have been encoded previously.



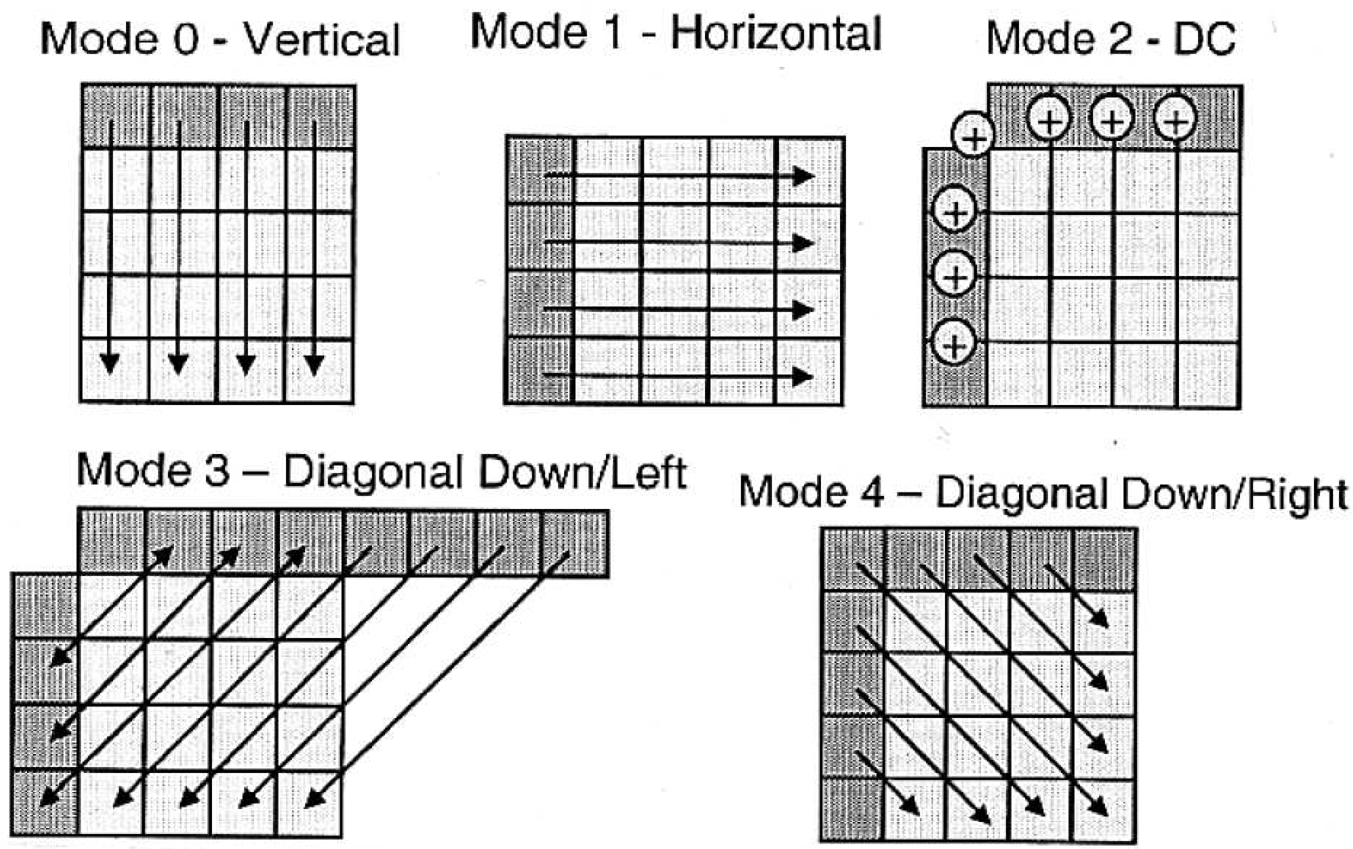
Intra Luma Prediction for 4x4 blocks



• The direction of prediction for 8 modes are shown above [2].

• In Mode 2, the samples a,..,p are predicted using average of samples A,..,D and I,…,L.

Intra Luma Prediction for 4x4 blocks



Five of the 9 intra 4x4 modes [3]

Intra Luma Prediction for 16x16 blocks

- There are 4 modes
  - Mode 0: vertical prediction
  - Mode 1: horizontal prediction
  - Mode 2: DC prediction
  - Mode 4: Plane prediction
- Intra chroma prediction has the same modes as above, but prediction is done for 8x8 chroma blocks.

Mode Decision

16x16 luma Macroblock

Intra Modes (For all frames)

- Nine 4x4 Modes
- Four 16x16 Modes

Inter Modes (Only for P and B-frames)

- Macroblock partitions: 16x16,16x8,8x16, 8x8,8x4,4x8,4x4
- Use of reference frames
- Use of integer, half and quarter pixel motion estimation

- Each mode (inter or intra) has an associated Rate-Distortion (RD) cost.
- Encoder performs mode decision to select the mode having the least RD cost. This process is computationally intensive.

References

1. I.E.Richardson, “H.264 and MPEG-4 video compression,” Wiley, 2003.
2. G. J. Sullivan, P. Topiwala, and A. Luthra, “The H.264/AVC Advanced Video Coding Standard:Overview and Introduction to the Fidelity Range Extensions,” SPIE Conference on Applications of Digital Image Processing XXVII, August, 2004
3. T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, “Overview of the H.264/AVC Video Coding Standard,” IEEE CSVT, Vol.13, pp. 560-576, July 2003.