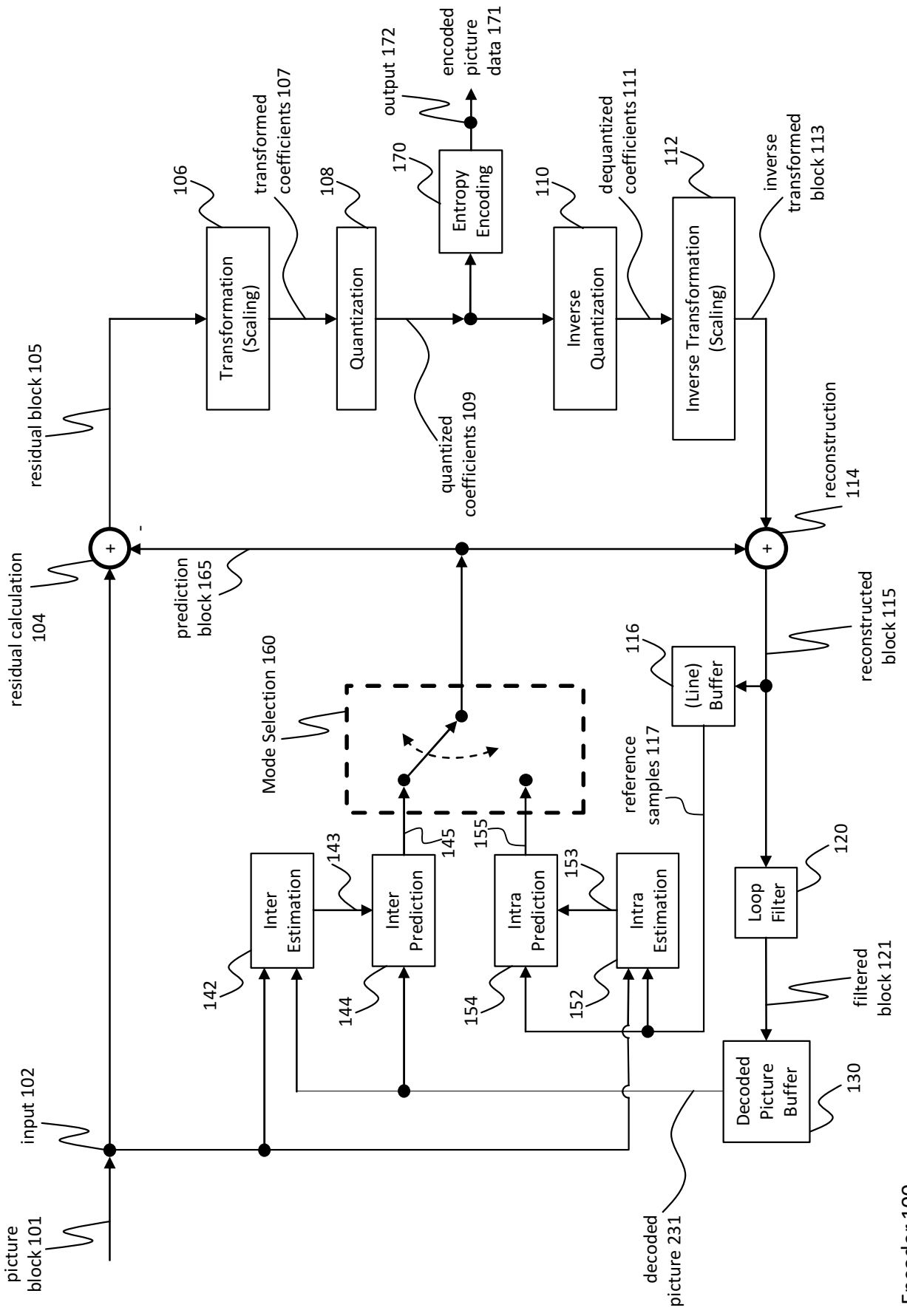
picture block 101
input 102
residual calculation 104
residual block 105
Transformation (Scaling)
106
transformed coefficients 107
Quantization
108
quantized coefficients 109
Entropy Encoding
170
output 172
encoded picture data 171
Inverse Quantization
110
dequantized coefficients 111
Inverse Transformation (Scaling)
112
inverse transformed block 113
reconstruction 114
reconstructed block 115
(Line) Buffer
116
reference samples 117
Loop Filter
120
filtered block 121
Decoded Picture Buffer
130
decoded picture 231
prediction block 165
Mode Selection 160
Inter Estimation
142
Inter Prediction
143
144
145
Intra Prediction
154
Intra Estimation
152
153
155
Encoder 100

Fig. 1

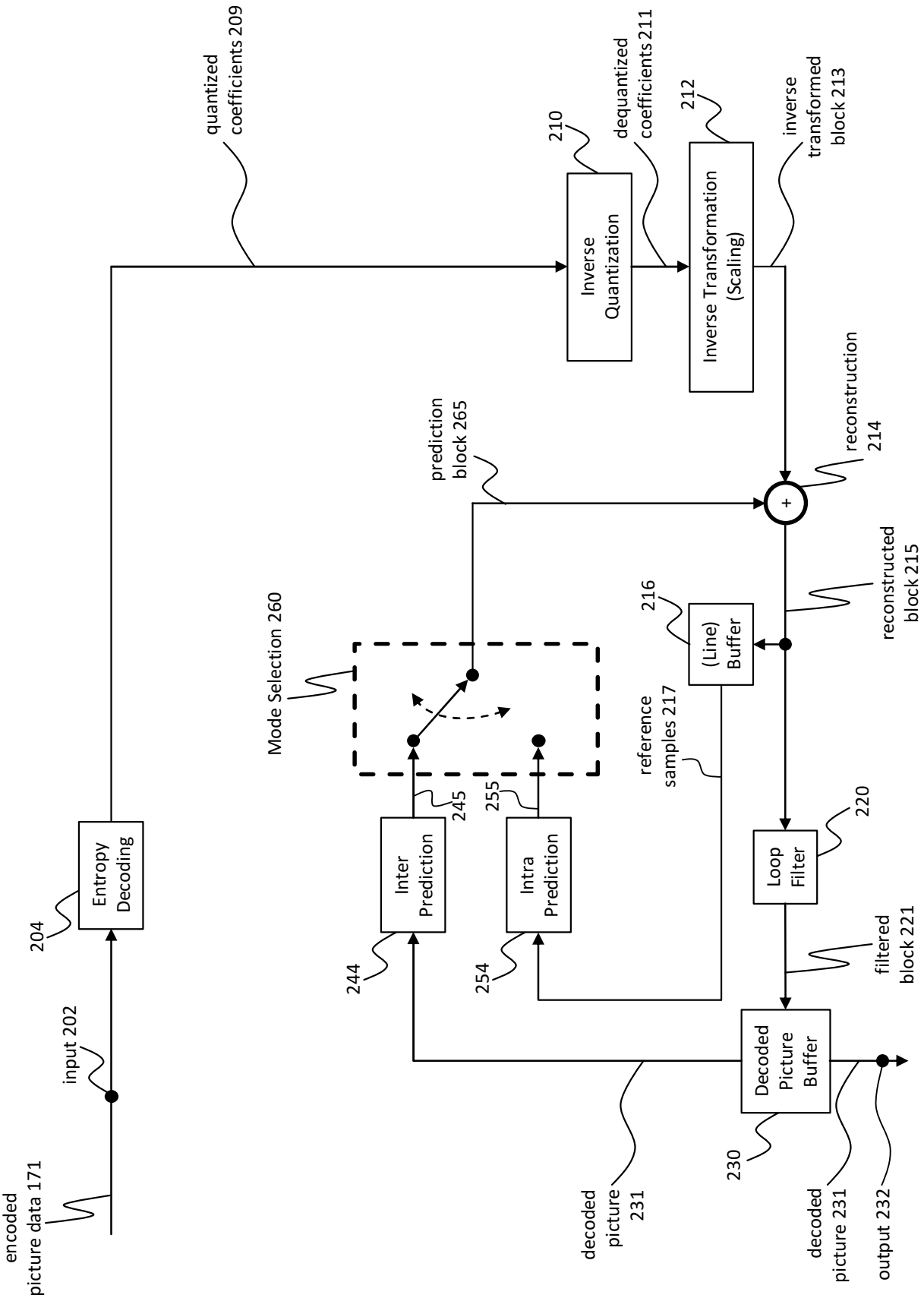
encoded picture data 171
input 202
204
Entropy Decoding
quantized coefficients 209
210
Inverse Quantization
dequantized coefficients 211
212
Inverse Transformation (Scaling)
inverse transformed block 213
reconstruction 214
prediction block 265
Mode Selection 260
Inter Prediction
244
245
Intra Prediction
254
255
216
(Line) Buffer
reference samples 217
reconstructed block 215
Loop Filter
220
filtered block 221
Decoded Picture Buffer
230
decoded picture 231
decoded picture 231
output 232
Decoder 200

Fig. 2

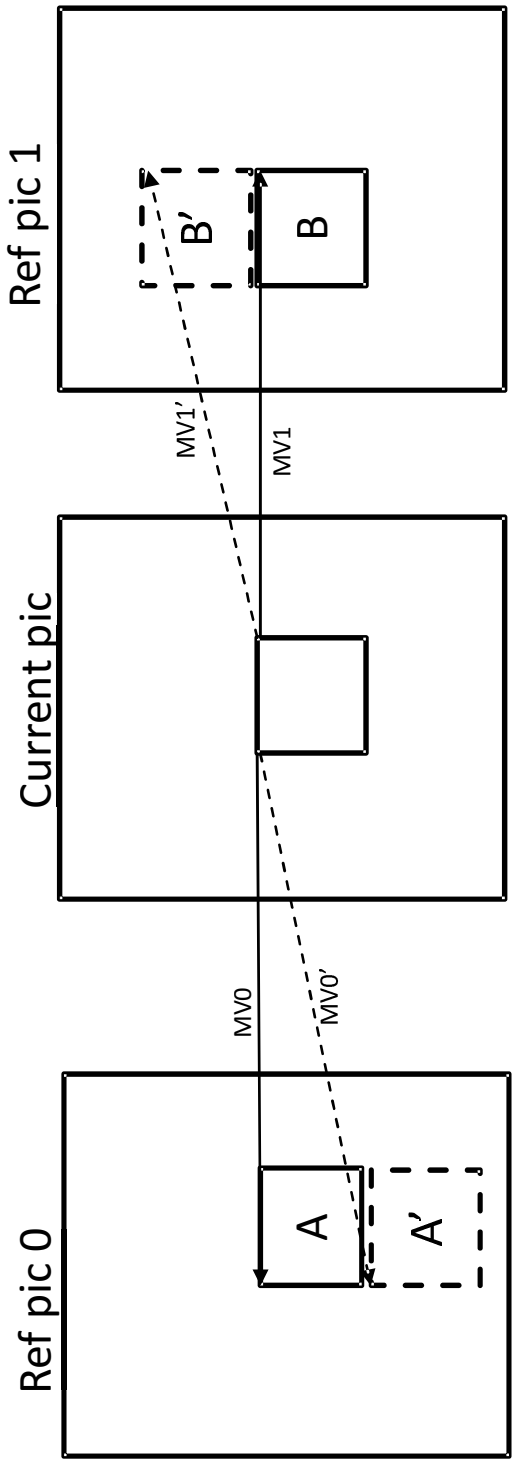
Ref pic 1
B'
B
MV1'
MV1
Current pic
MV0
MV0'
Ref pic 0
A
A'

Fig. 3

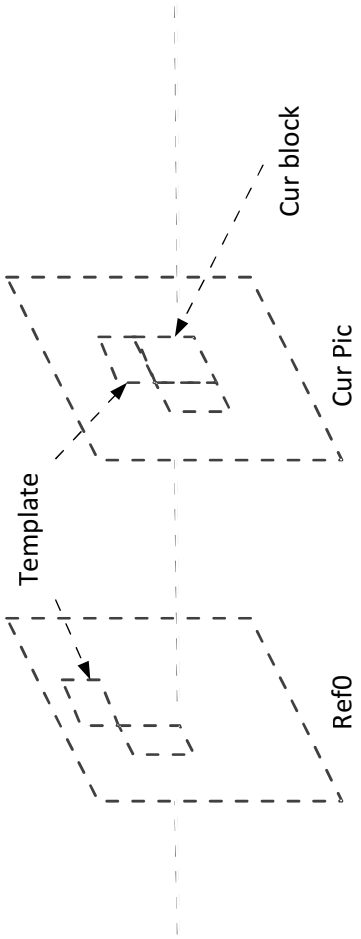
Cur block
Cur Pic
Template
Ref0

Fig. 4

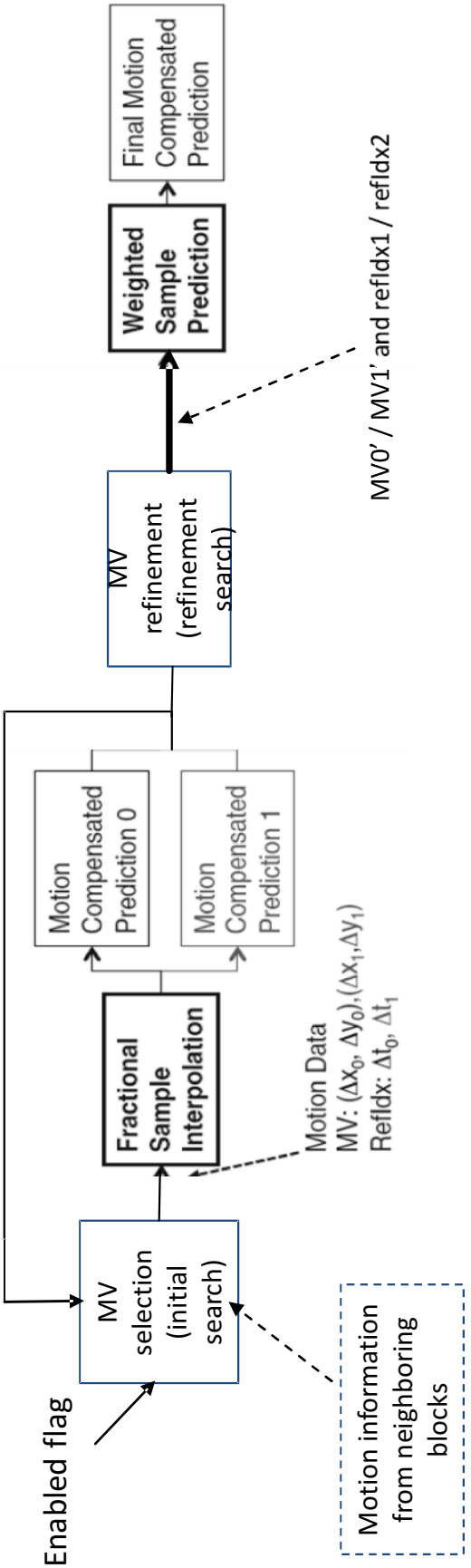
Enabled flag
MV selection (initial search)
Motion information from neighboring blocks
Fractional Sample Interpolation
Motion Data
MV: $(\Delta x_0, \Delta y_0),(\Delta x_1,\Delta y_1)$
RefIdx: $\Delta t_0, \Delta t_1$
Motion Compensated Prediction 0
Motion Compensated Prediction 1
MV refinement (refinement search)
MV0' / MV1' and refIdx1 / refIdx2
Weighted Sample Prediction
Final Motion Compensated Prediction

Fig. 5

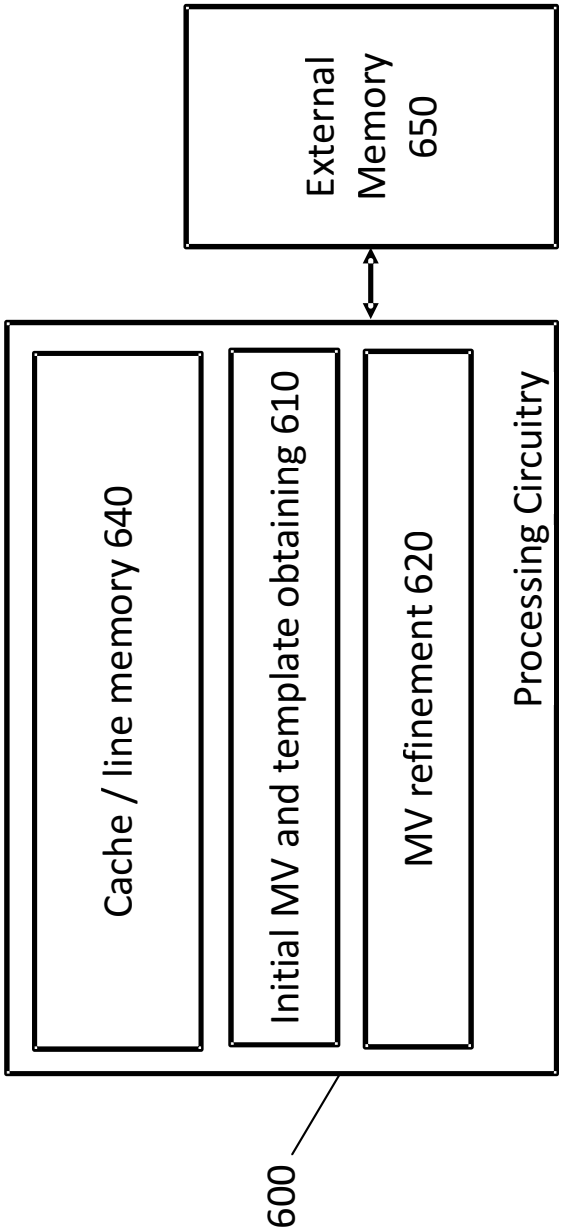
External Memory 650
Cache / line memory 640
Initial MV and template obtaining 610
MV refinement 620
Processing Circuitry
600

Fig. 6

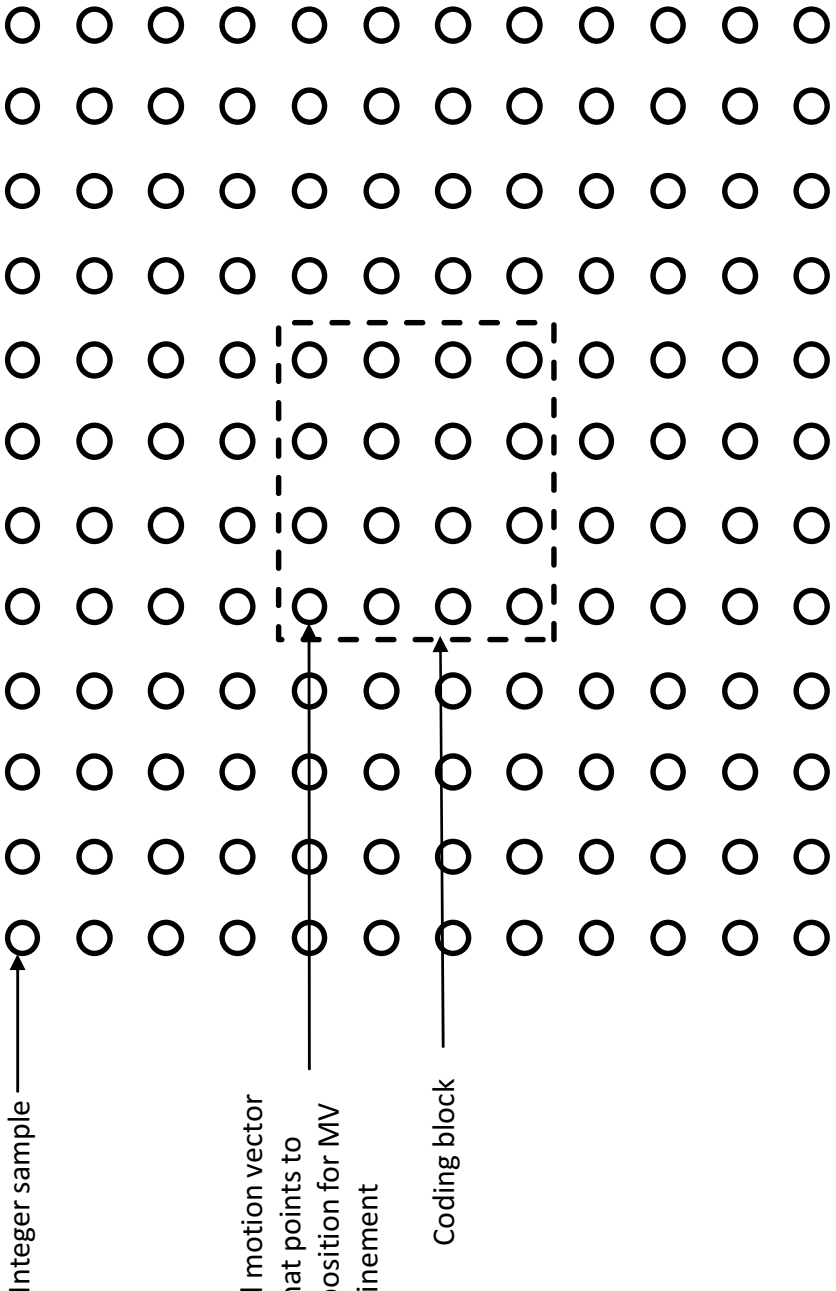
Integer sample
Unrefined motion vector
MV_0 that points to
starting position for MV
refinement
Coding block

Fig. 7

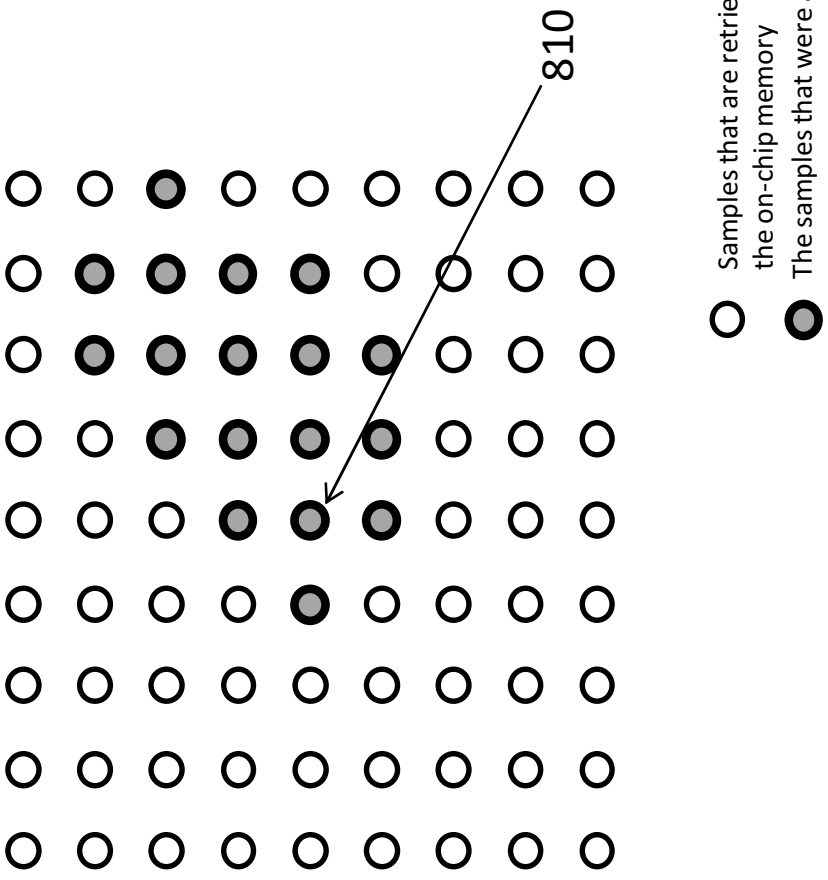
810
Samples that are retrieved from the external memory to the on-chip memory
The samples that were actually used by the computation

Fig. 8

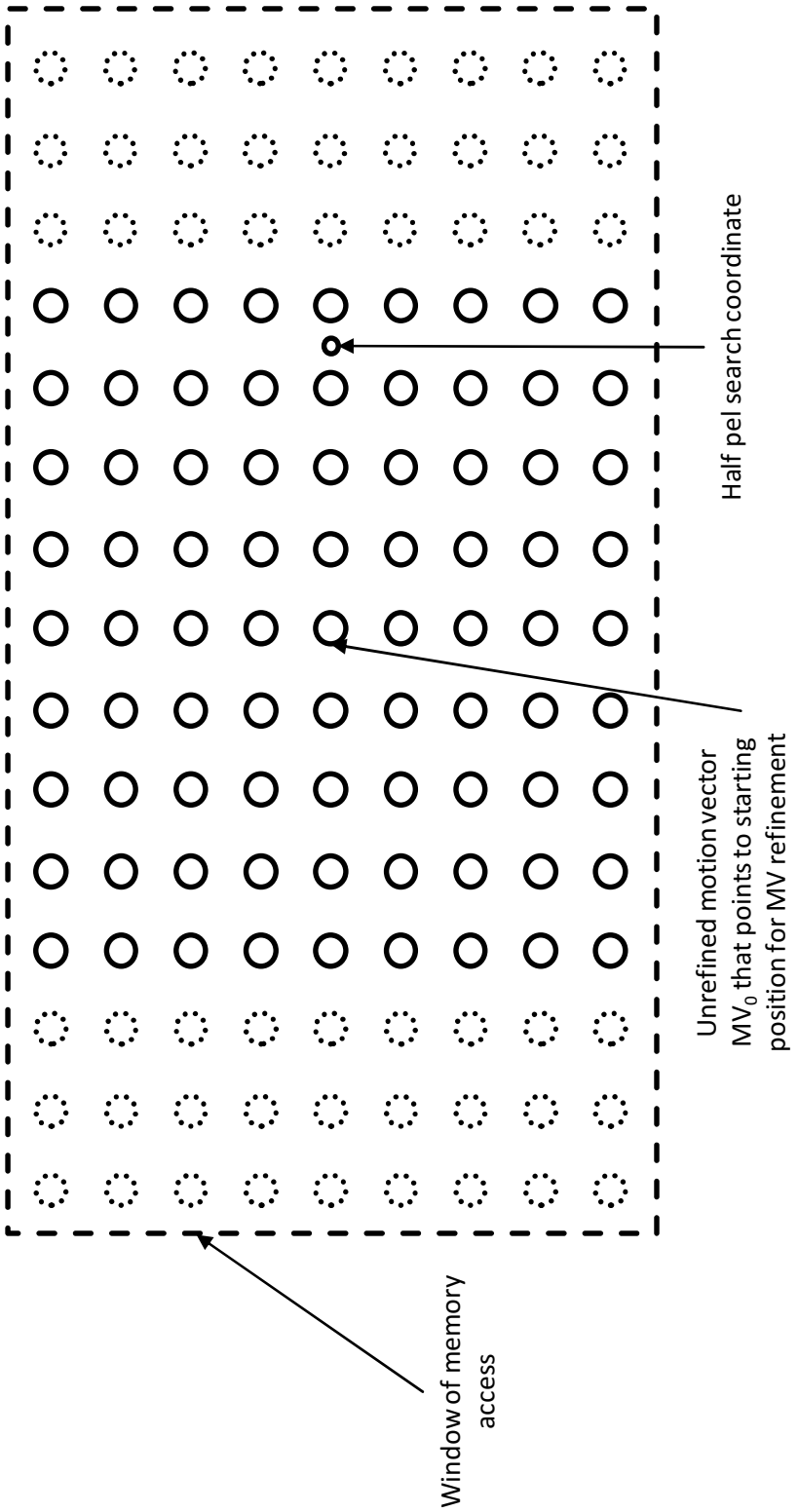
Half pel search coordinate
Unrefined motion vector
MV_0 that points to starting
position for MV refinement
Window of memory
access

Fig. 9

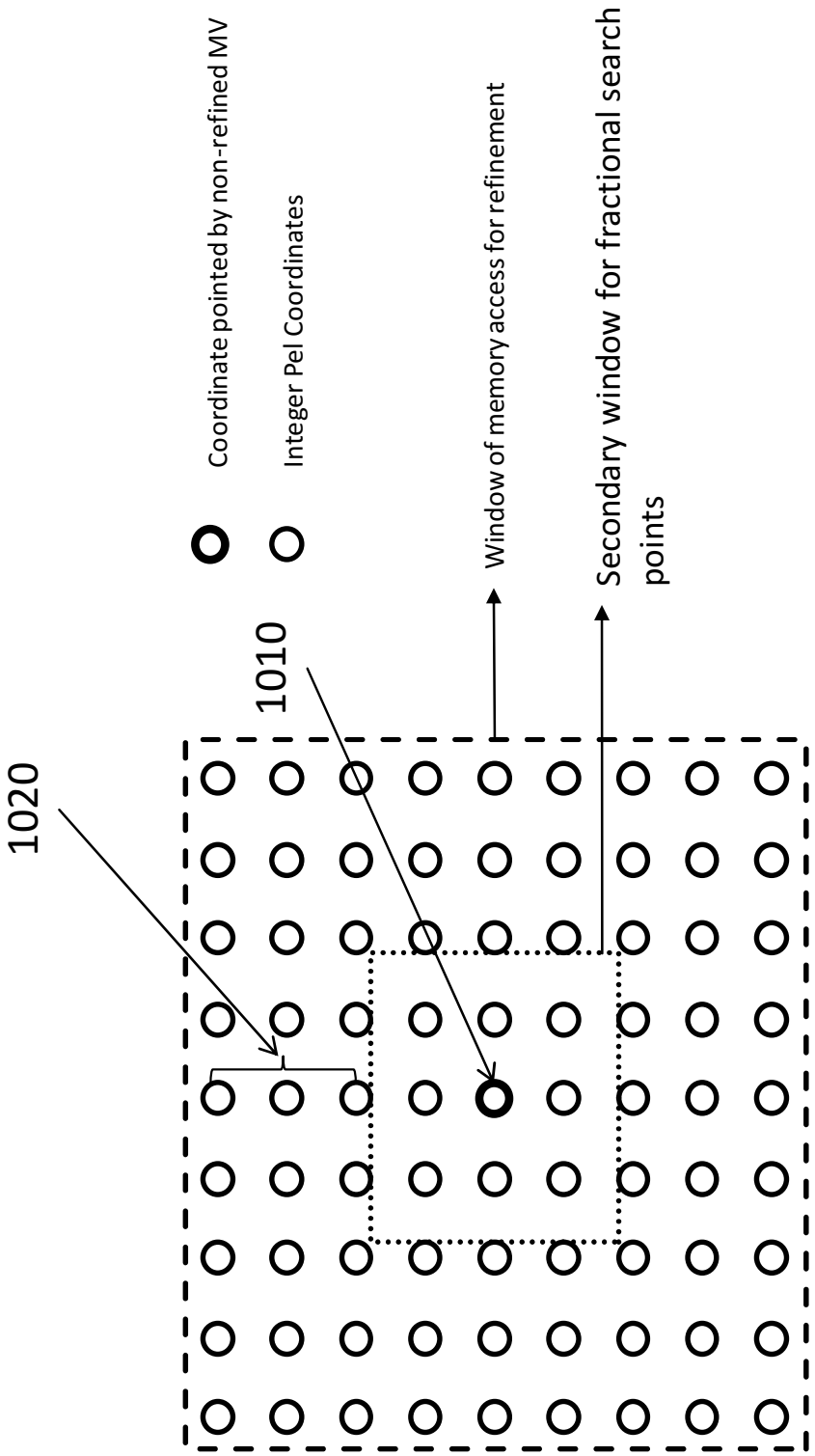
Coordinate pointed by non-refined MV
Integer Pel Coordinates
Window of memory access for refinement
Secondary window for fractional search points
1010
1020

Fig. 10

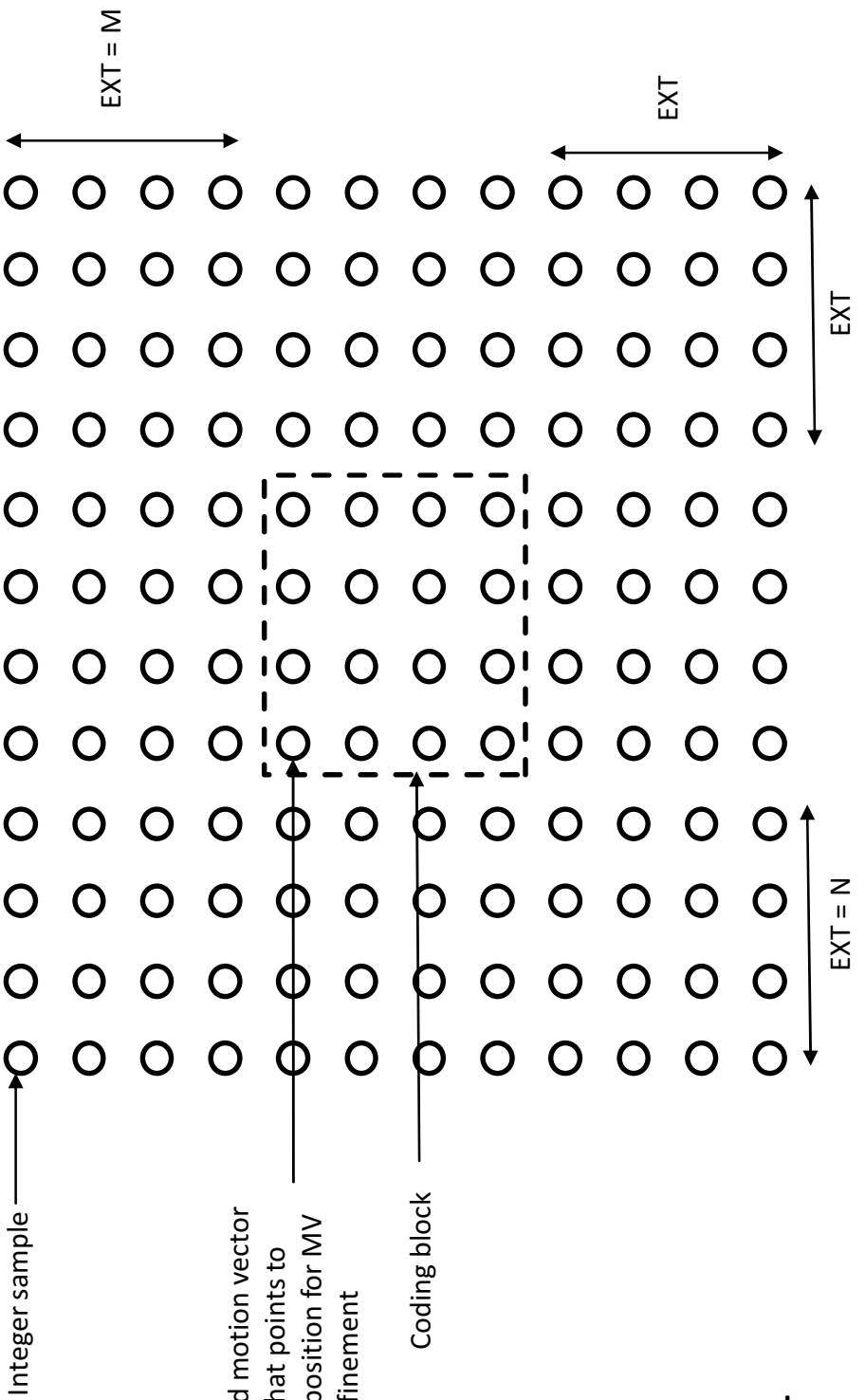
EXT = M
EXT
EXT
EXT = N
Integer sample
Unrefined motion vector
MV_0 that points to
starting position for MV
refinement
Coding block

Fig. 11

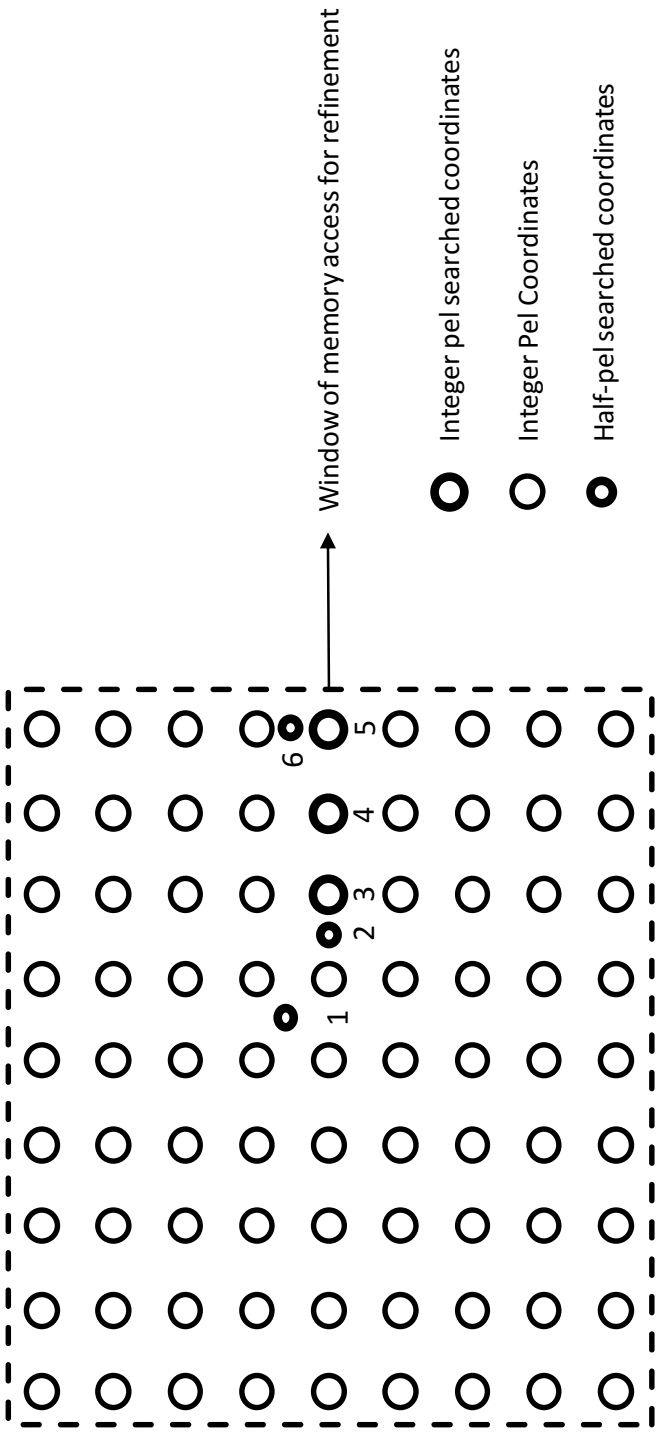
Window of memory access for refinement
Integer pel searched coordinates
Integer Pel Coordinates
Half-pel searched coordinates
1
2
3
4
5
6

Fig. 12

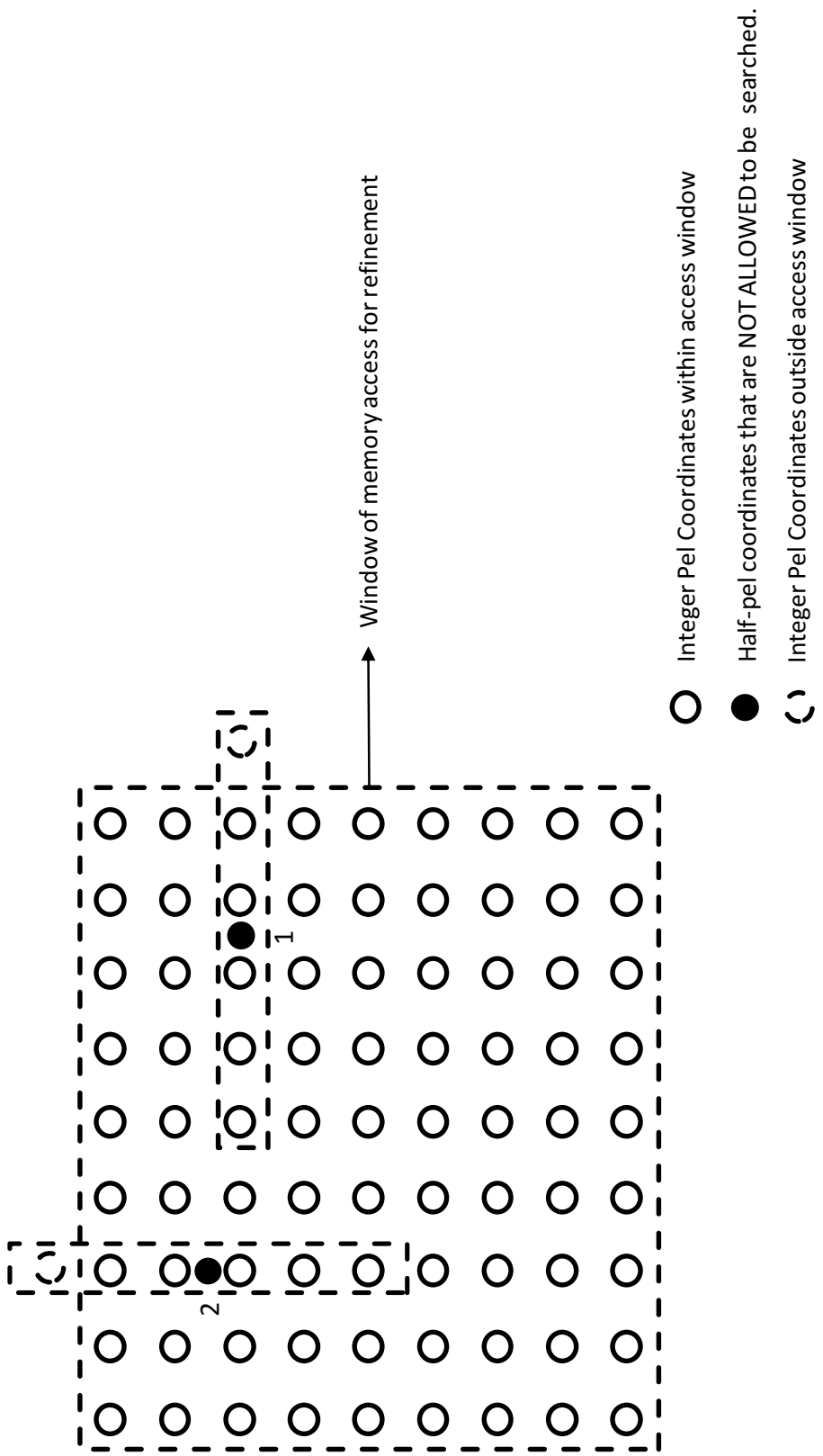
Window of memory access for refinement
Integer Pel Coordinates within access window
Half-pel coordinates that are NOT ALLOWED to be searched.
Integer Pel Coordinates outside access window
1
2

Fig. 13

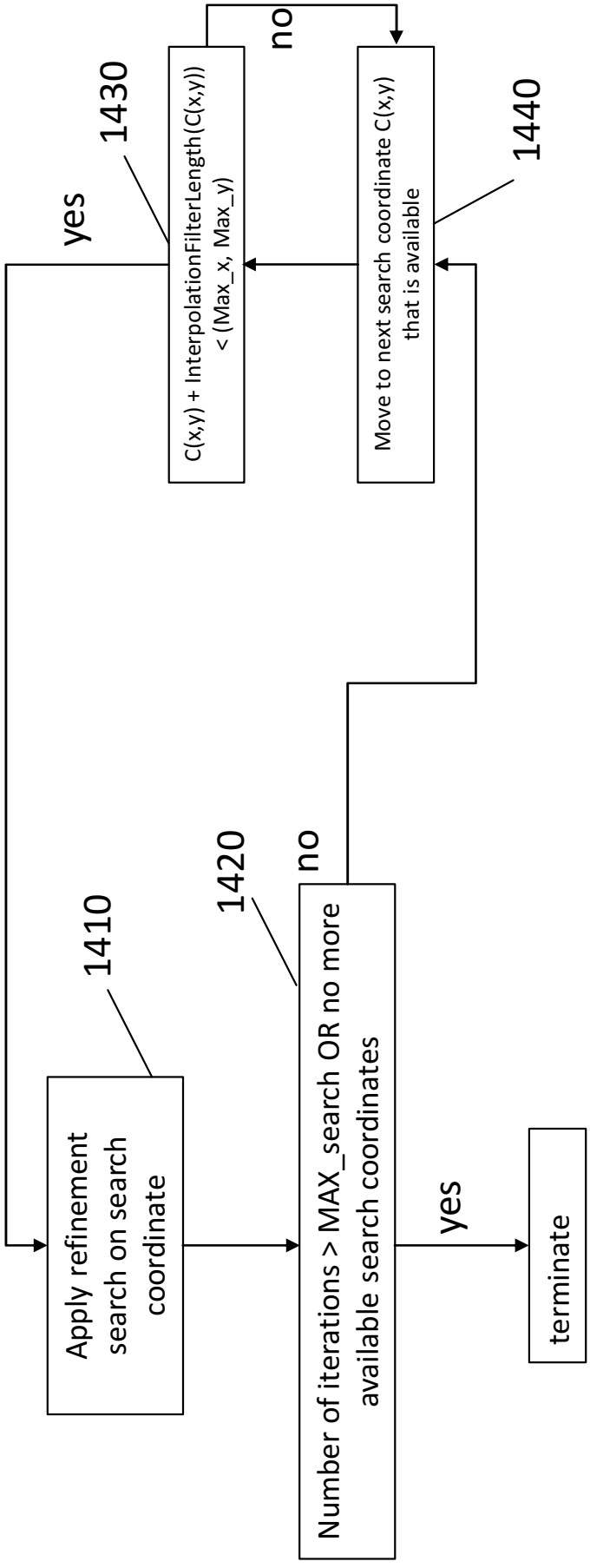
Apply refinement search on search coordinate
1410
Number of iterations > MAX_search OR no more available search coordinates
1420
no
yes
terminate
C(x,y) + InterpolationFilterLength(C(x,y)) < (Max_x, Max_y)
1430
yes
no
Move to next search coordinate C(x,y) that is available
1440

Fig. 14

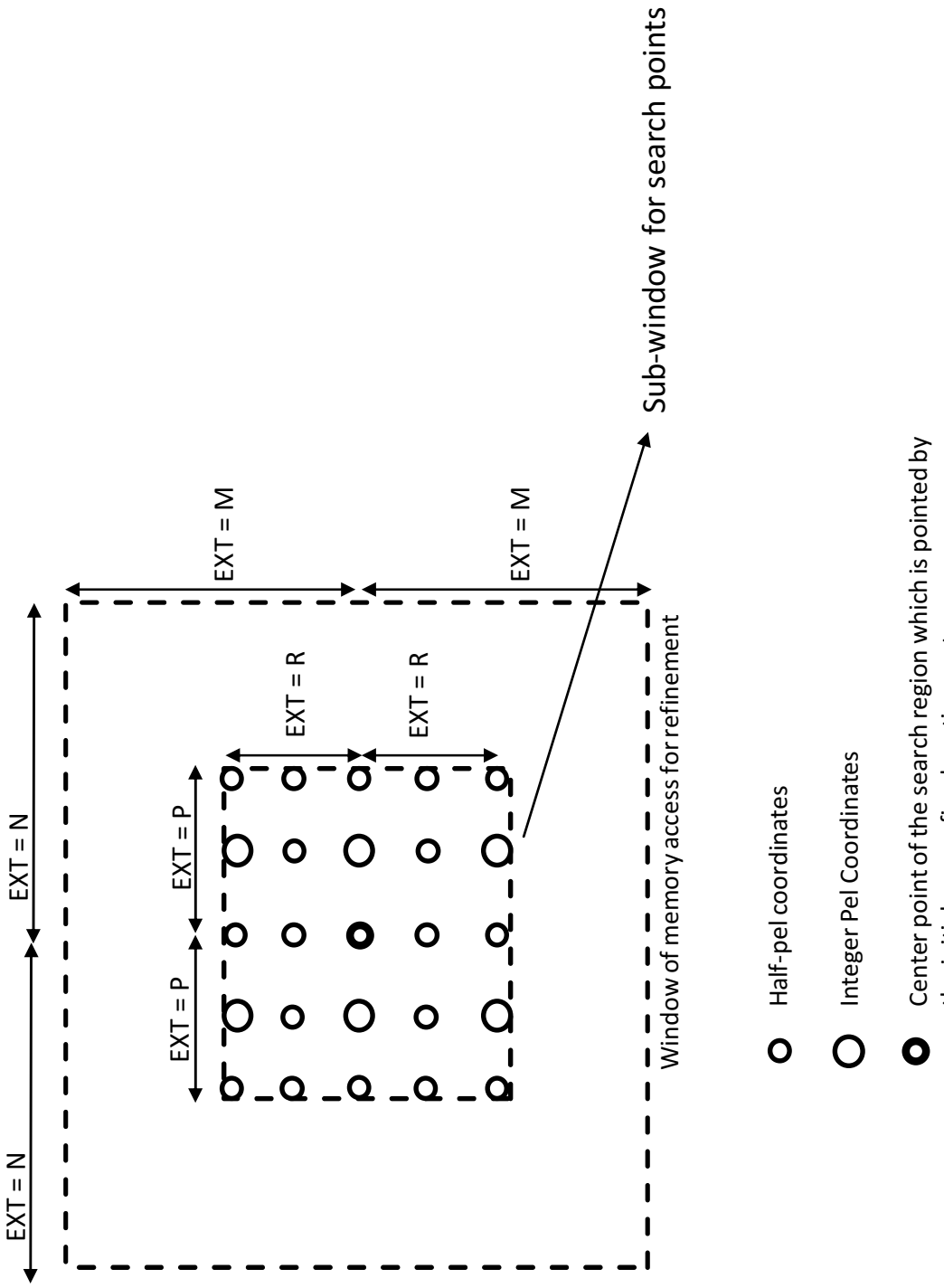
EXT = N
EXT = N
EXT = P
EXT = P
EXT = R
EXT = R
EXT = M
EXT = M
Window of memory access for refinement
Sub-window for search points
Half-pel coordinates
Integer Pel Coordinates
Center point of the search region which is pointed by the initial non-refined motion vector

Fig. 15

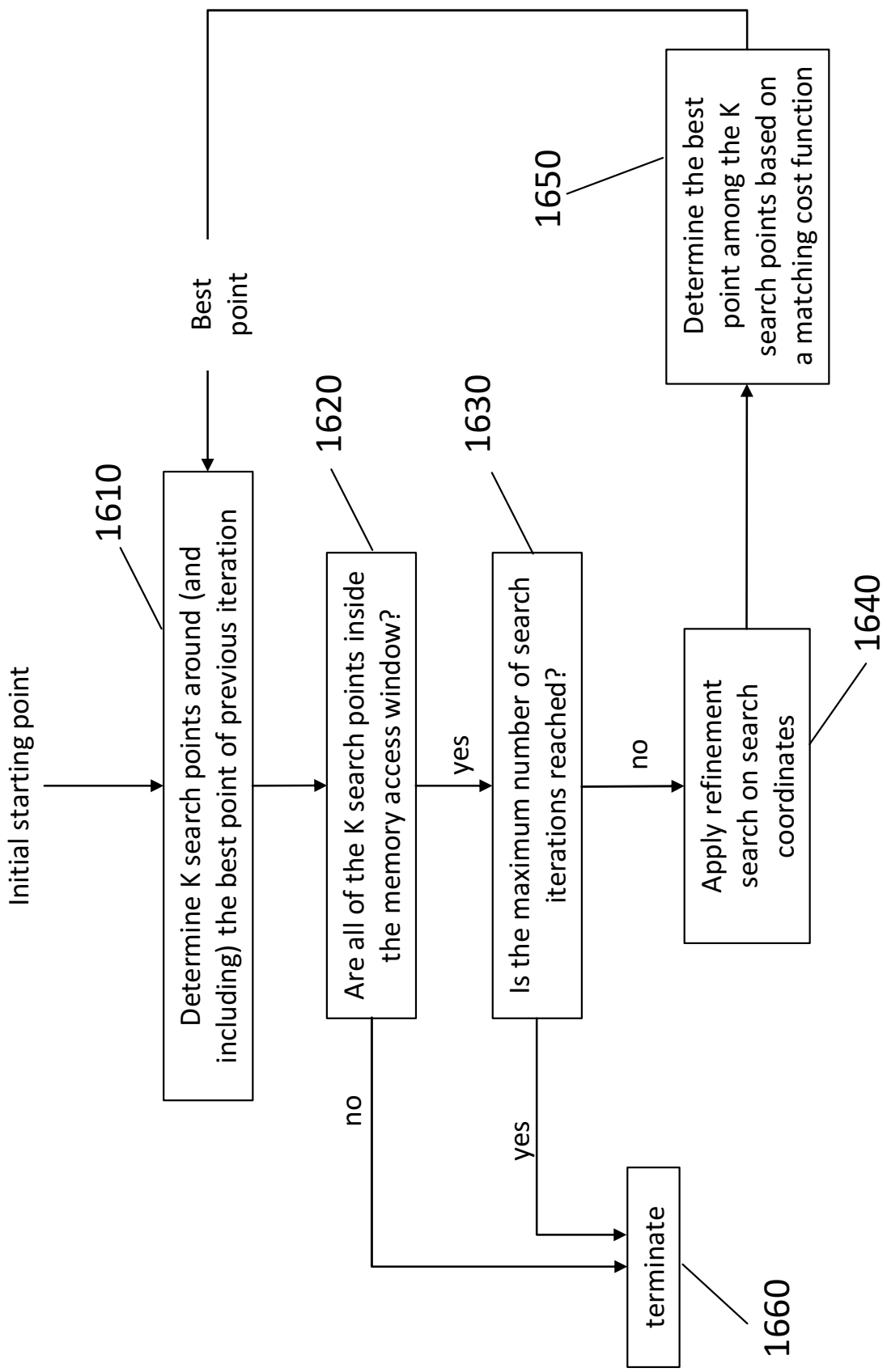
Initial starting point
Determine K search points around (and including) the best point of previous iteration
1610
Best point
Are all of the K search points inside the memory access window?
1620
yes
no
Is the maximum number of search iterations reached?
1630
yes
no
Apply refinement search on search coordinates
1640
Determine the best point among the K search points based on a matching cost function
1650
terminate
1660

Fig. 16

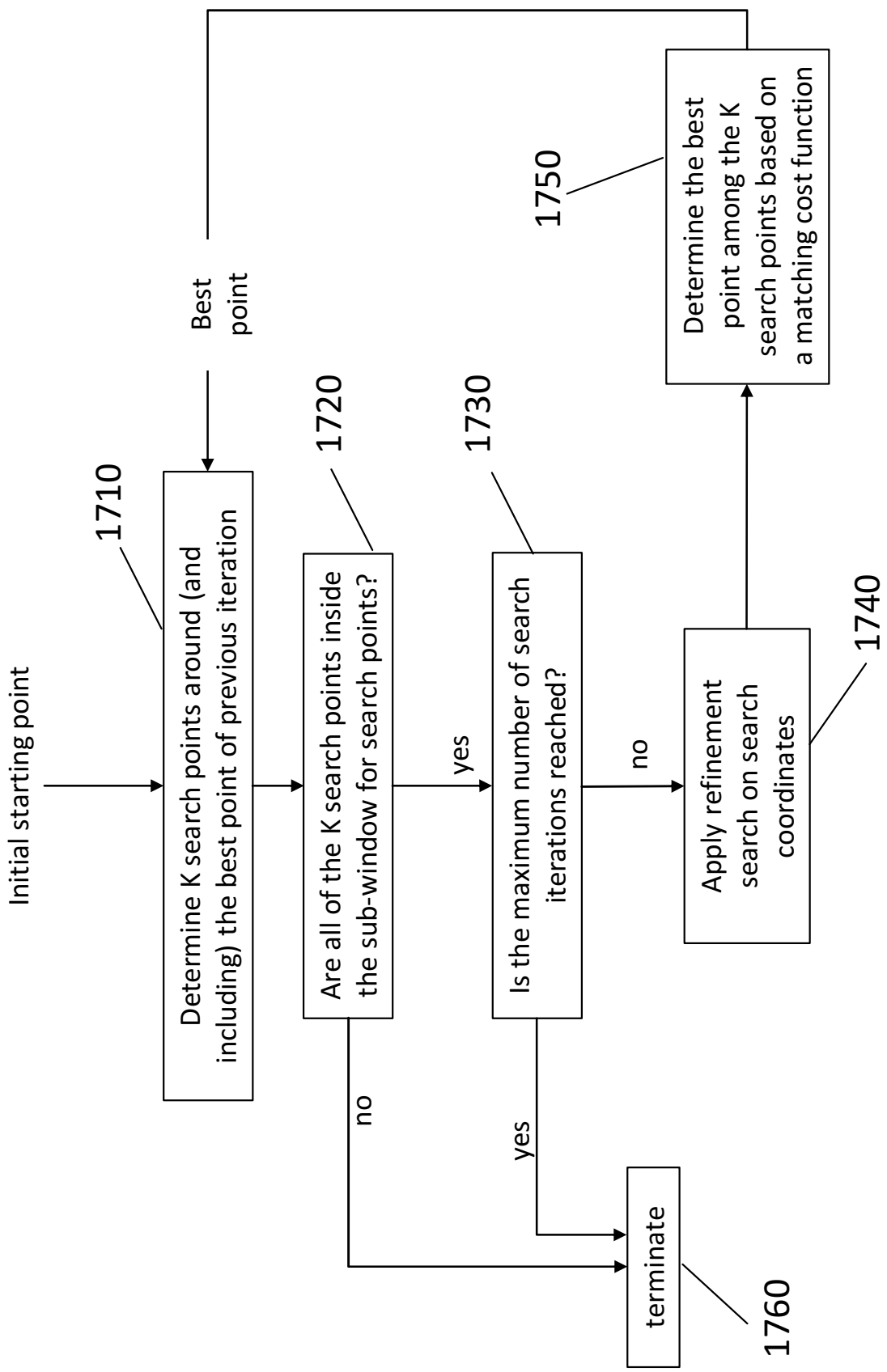
Initial starting point
Determine K search points around (and including) the best point of previous iteration
1710
Are all of the K search points inside the sub-window for search points?
1720
yes
no
Is the maximum number of search iterations reached?
1730
yes
no
Apply refinement search on search coordinates
1740
Determine the best point among the K search points based on a matching cost function
1750
Best point
terminate
1760

Fig. 17