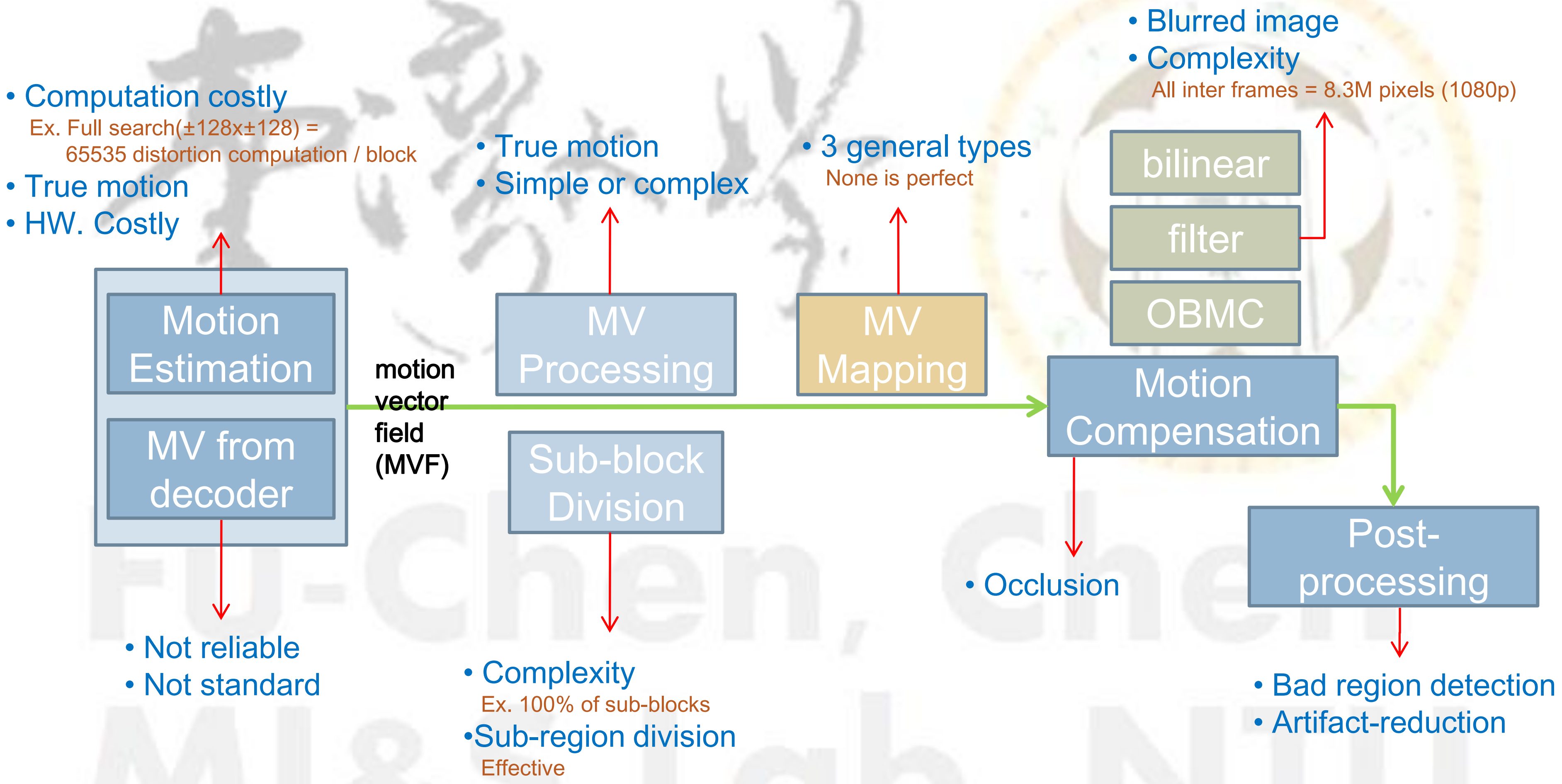
Algorithm Analysis & Challenge



- Computation costly
  Ex. Full search(±128x±128) = 65535 distortion computation / block
- True motion
- HW. Costly

Motion Estimation

MV from decoder

- Not reliable
- Not standard

motion vector field (MVF)

- True motion
- Simple or complex

MV Processing

Sub-block Division

- Complexity
  Ex. 100% of sub-blocks
- Sub-region division
  Effective

- 3 general types
  None is perfect

MV Mapping

- Blurred image
- Complexity
  All inter frames = 8.3M pixels (1080p)

bilinear

filter

OBMC

Motion Compensation

- Occlusion

Post-processing

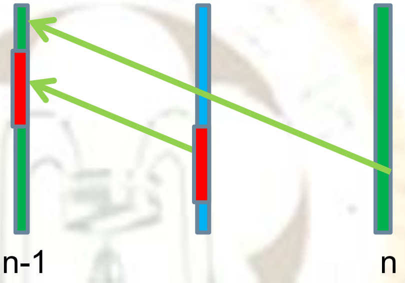
- Bad region detection
- Artifact-reduction

* HW. = hardware
* BW. = bandwidth

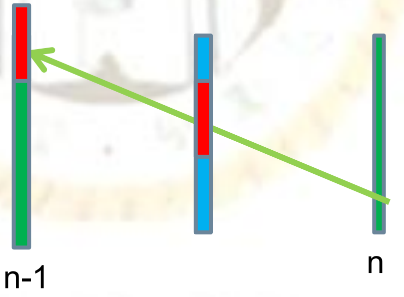
Algorithm

Algorithm Analysis & Challenge

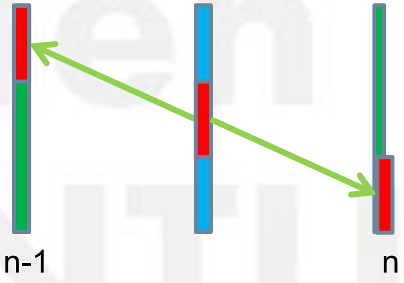
- MV Mapping
  - Tradition
    - ME on the existing frames
    - Copy the MV of corresponding position for MC
    - Simple but **wrong in time domain**



  - Through
    - Similar to tradition, but MC through exist MV's direction
    - **Overlap & Hole problem**
    - **Non-block based MC**



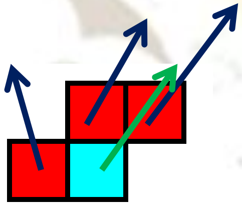
  - Bilateral
    - ME on inter-frames
    - **# of ME = # of inter-frames**
    - **Often failed at flat region**

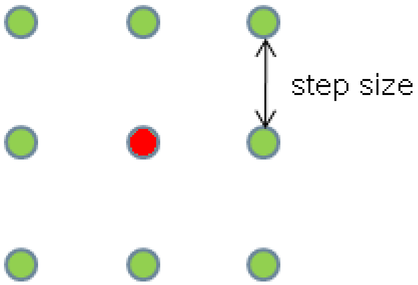


  - No one is perfect !

Algorithm - ME
Predictive square search

- Similar to predictive diamond search but using square pattern (SP)
- Procedure
  - 1. Set MV = median of 3 neighboring blocks’ MVs
  - 2. Apply 4-step SP on MV
    - If ε is at center or ε < threshold
      - Apply 2-step & 1-step SP for converge
    - Else
      - Set MV = origin, go to step 3
  - 3. Apply 8-step SP on MV
    - If ε is not at center
      - Set MV = ε’s position, repeat step 3
    - Else
      - Apply 4-step, 2-step & 1-step SP for converge

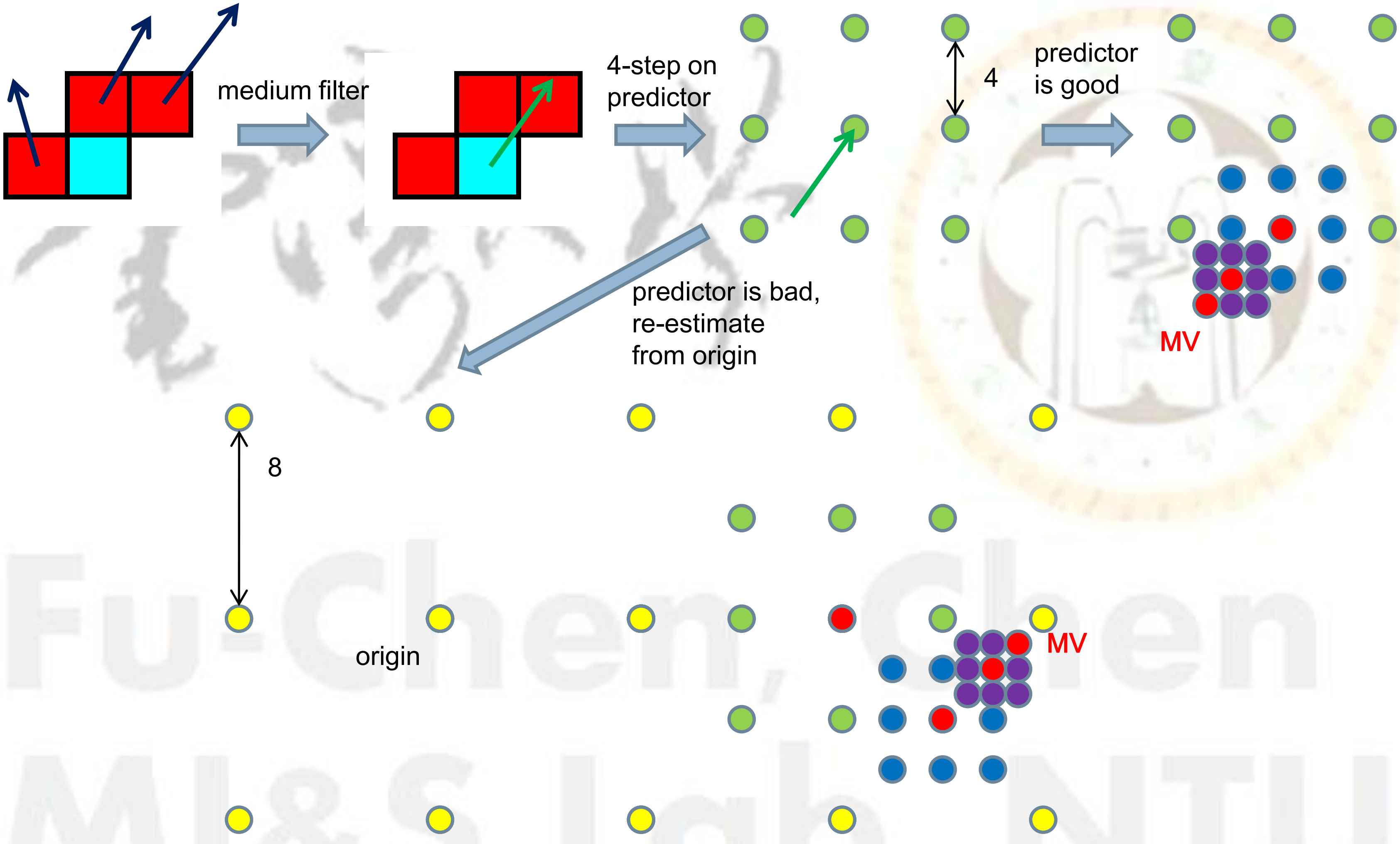




Square pattern

* ε = min. distortion

Algorithm - ME
Predictive square search

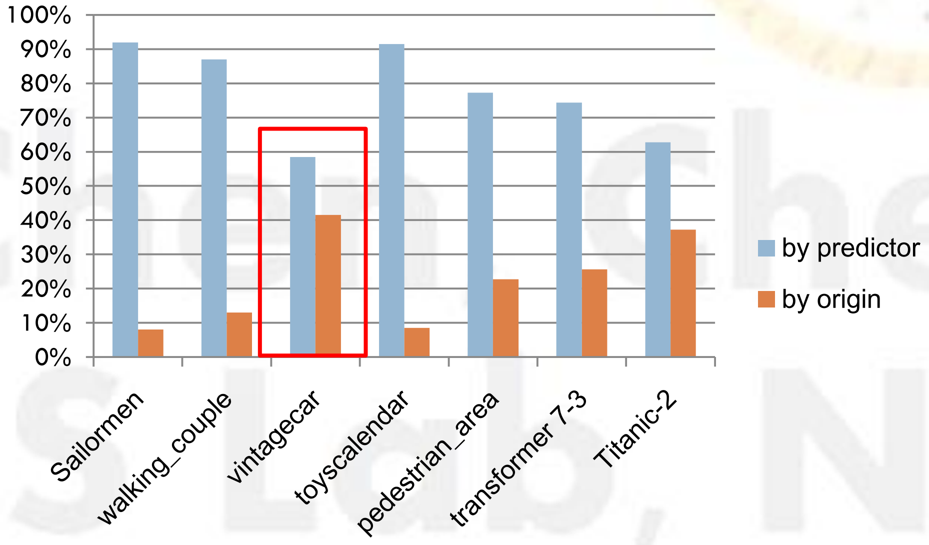


Algorithm - ME
Predictive square search

[TrueME 07]
Huska, J., Kulla, P., “A New Recursive Search with Multi Stage Approach for Fast Block Based True Motion Estimation,” International Conference Radioelektronika 2007

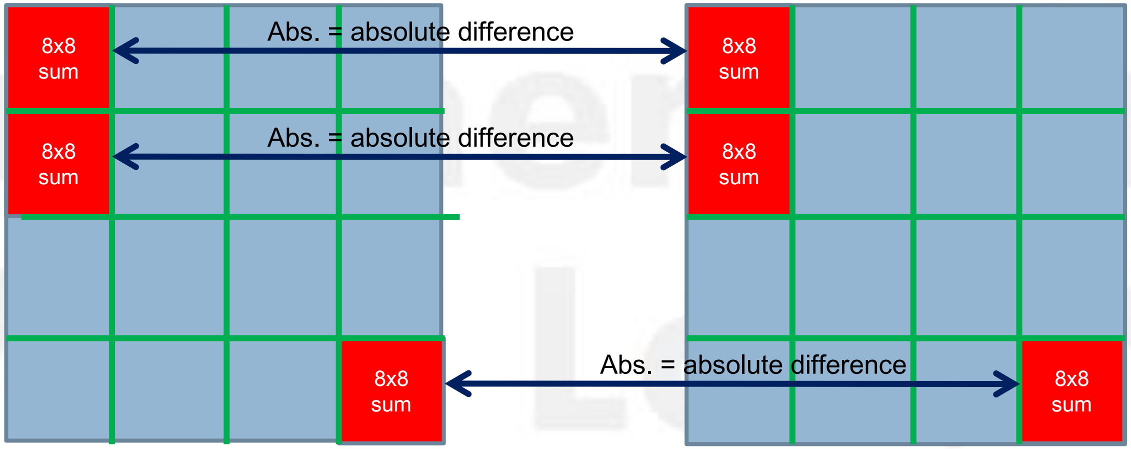
- The reasons
  - Similar to many true motion estimation algorithms **[TrueME 07]**
    - Spatial coherence of MVF
  - Can reject predictor & re-estimation
  - Very cost efficient
    - For complex sequences, near **60%** blocks converge around predictor (25 distortion computation)

Percentage of block’s converge type
(worst cases for each sequence)



Algorithm - ME
Predictive square search

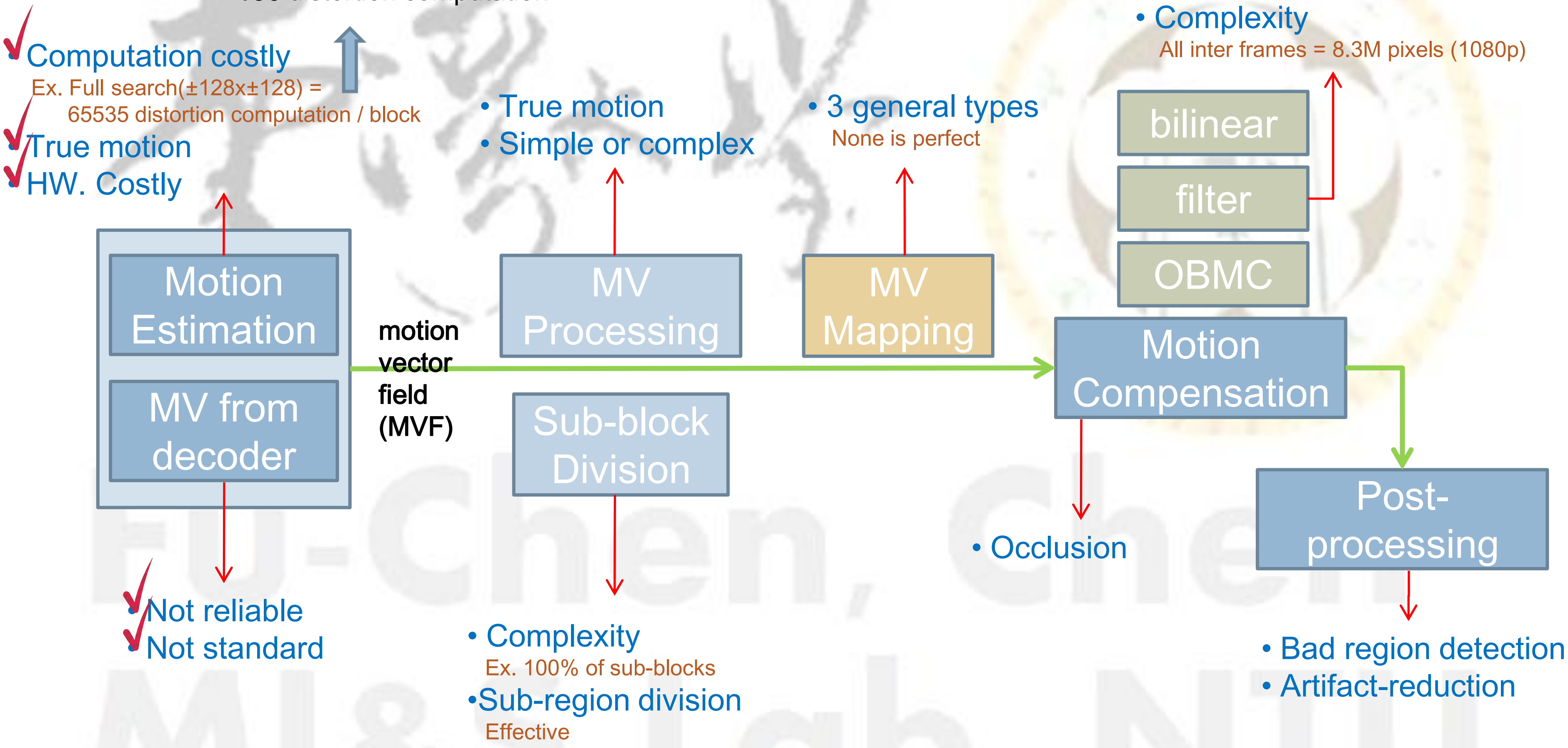
- Bigger block size : 32x32 for 1080p
  - For aperture problem
    - Ex. OBME
  - More efficient than merging smaller block for MV unity
- Distortion criterion : 8x8 MSEA
  - Down-sample version of SAD
    - Without down-sample MV value
    - Almost the same result comparing to SAD
  - To co-operate with square pattern, it reduces lots of **computation & bandwidth cost** in hardware
- Search range
  - By experiments, the max. MV value is about 128 (for 24 Hz)
  - Set search range = **±128 x ±128** for hardware implementation


8x8 sum
Abs. = absolute difference
8x8 sum
8x8 sum
Abs. = absolute difference
8x8 sum
8x8 sum
Abs. = absolute difference
8x8 sum

MSEA = Σ Abs.

Algorithm - ME
Predictive square search

60% blocks : **25** distortion computation
40% blocks : at most 25 + (128/8)*5 + 25
= **130** distortion computation



- ✓ Computation costly
  Ex. Full search(±128x±128) =
  65535 distortion computation / block
- ✓ True motion
- ✓ HW. Costly

Motion Estimation

MV from decoder

- ✓ Not reliable
- ✓ Not standard

motion vector field (MVF)

- True motion
- Simple or complex

MV Processing

Sub-block Division

- Complexity
  Ex. 100% of sub-blocks
- Sub-region division
  Effective

- 3 general types
  None is perfect

MV Mapping

- Blurred image
- Complexity
  All inter frames = 8.3M pixels (1080p)

bilinear

filter

OBMC

Motion Compensation

- Occlusion

Post-processing

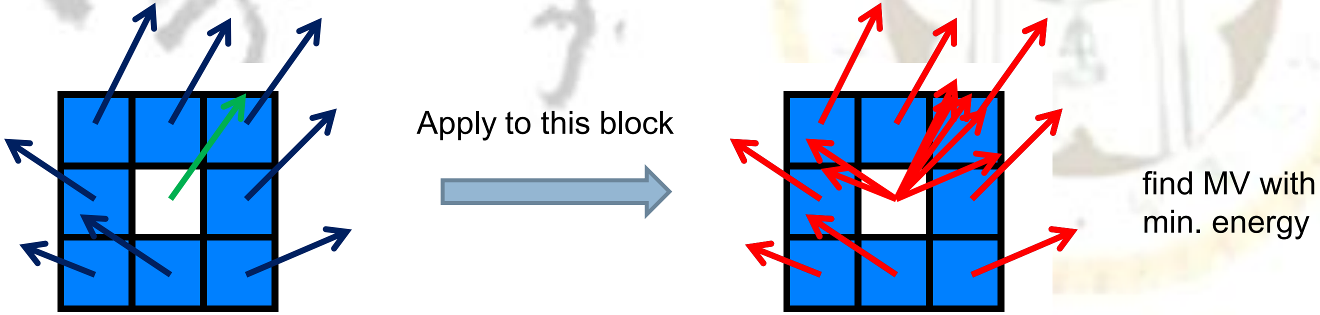
- Bad region detection
- Artifact-reduction

* HW. = hardware
* BW. = bandwidth

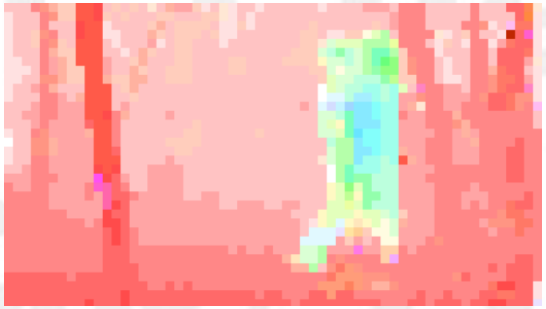
Algorithm - MV Processing
Markov random field correction

- Select 8-neighbor MV and MV of itself as candidates
- Apply those 9 candidates to this block, choose one with min. MRF energy
  - Like <u>ICM minimization</u> for MRF with selected candidates

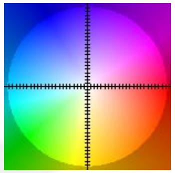
MRF energy = $MSEA_{candidate} + weight * \sum_{neighbor} |MV_{candidate} - MV_{neighbor}|$



- Run 3 iterations on whole frame



* Color wheel of motion



Algorithm - MV Processing
Markov random field correction

- Why choosing MRF modeling
  - Theoretical modeling based on Bayesian framework

$$E(f) = \underbrace{-\sum_{p} \ln g_{p}(O|f_{p})}_{\text{Data term (sensor noise)}} + \underbrace{\sum_{(p,q)} V_{(p,q)}(f_{p}, f_{q})}_{\text{Smoothness term (MRF prior)}}$$

  - Frequently applied on flow or motion estimation **[Lim 02]**
- Why choosing ICM for energy minimization
  - With lower cost
    - Ex. Belief propagation hardware with **633K gate count** & **1.88MByte SRAM [Liang 09]**
- Why choosing only 9 neighboring candidates
  - True motion comes from nearby blocks with very high probability **[Huang 10]**
  - Preventing over-smoothing
  - Low cost
    - Tradition ICM : 65536 MRF energy computation
    - With selected candidates : 9 MRF energy computation

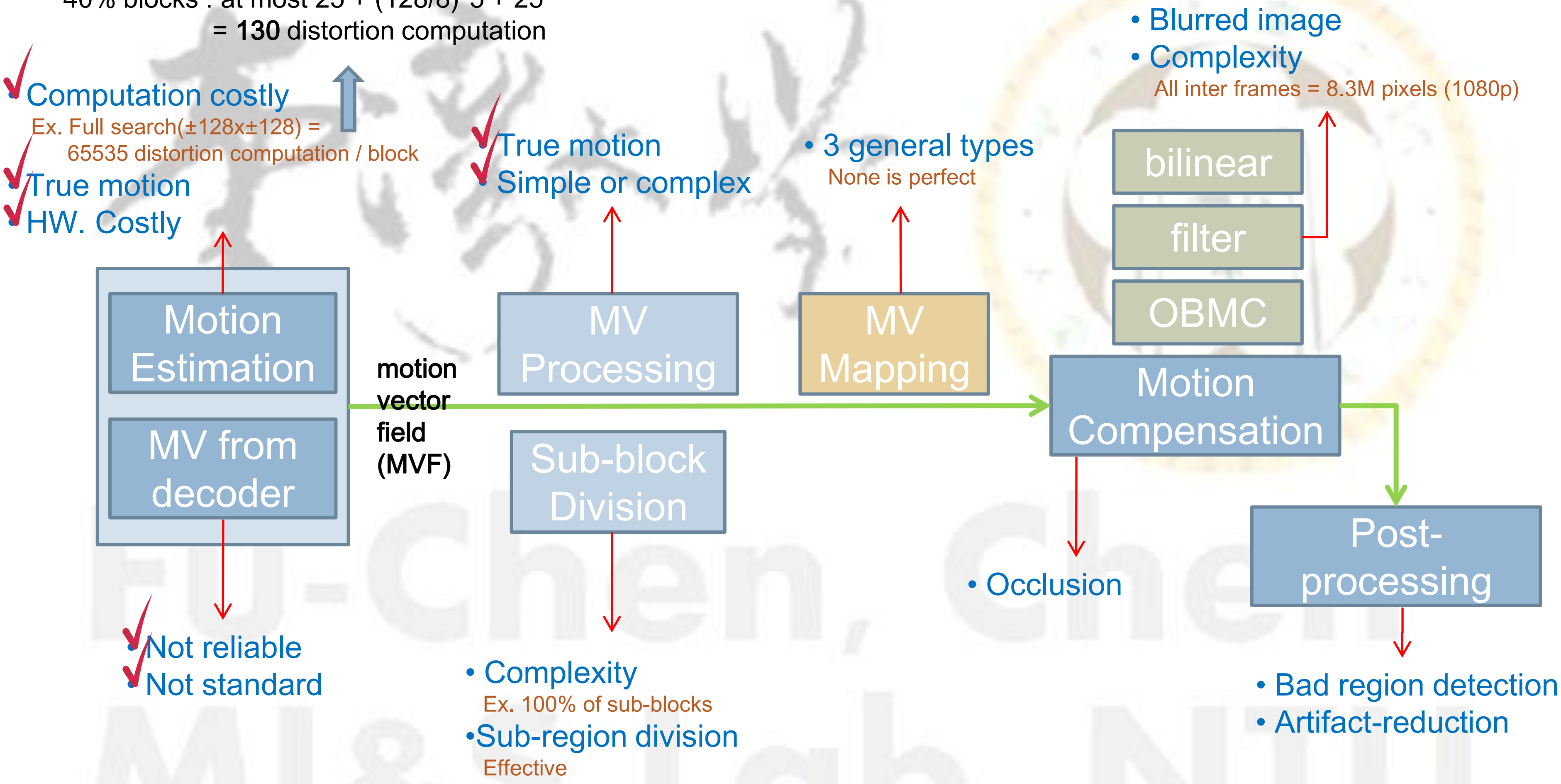
[Lim 02]
Keng Pang Lim, Das, A., Man Nang Chong, "Estimation of occlusion and dense motion fields in a bidirectional Bayesian framework," IEEE Transactions on Pattern Analysis and Machine Intelligence, May 2002, pp. 712-718

[Liang 09]
Chia-Kai Liang, Chao-Chung Cheng, Yen-Chieh Lai, Liang-Gee Chen, Homer H. Chen, "Hardware-Efficient Belief Propagation," CVPR 2009

[Huang 10]
Yung-Lin Huang, Yi-Nung Liu, and Shao-Yi Chien, "MRF-based True Motion Estimation Using H.264 Decoding Information," SiPS 2010

Algorithm - MV Processing
Markov random field correction

60% blocks : **25** distortion computation
40% blocks : at most 25 + (128/8)*5 + 25
= **130** distortion computation



- ✓ Computation costly
  Ex. Full search(±128x±128) = 65535 distortion computation / block
- ✓ True motion
- ✓ HW. Costly

- ✓ True motion
- ✓ Simple or complex

- 3 general types
  None is perfect

- Blurred image
- Complexity
  All inter frames = 8.3M pixels (1080p)

- ✓ Not reliable
- ✓ Not standard

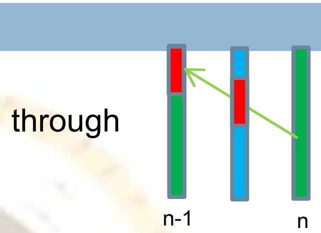
- Complexity
  Ex. 100% of sub-blocks
- Sub-region division
  Effective

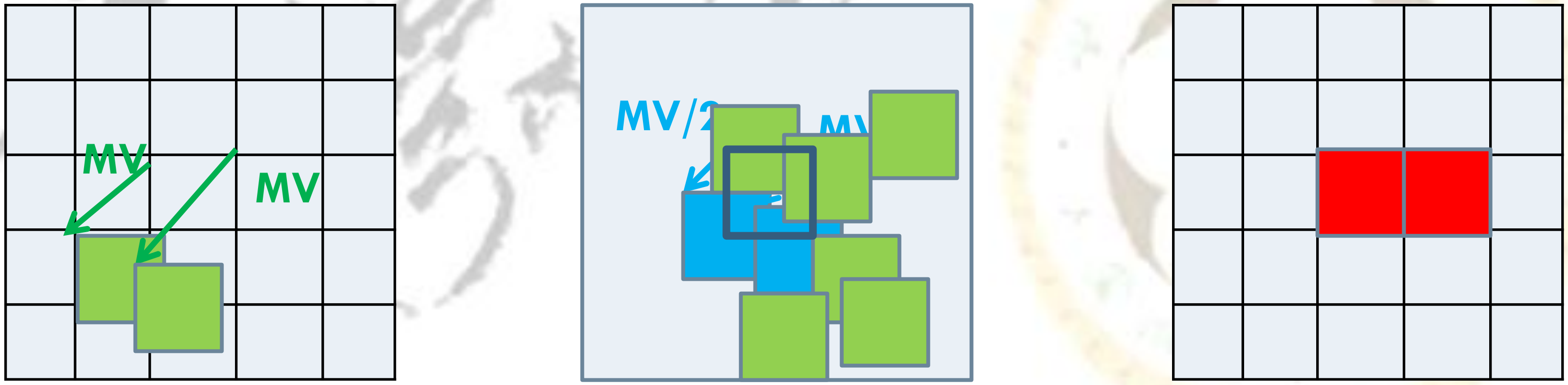
- Occlusion

- Bad region detection
- Artifact-reduction

* HW. = hardware
* BW. = bandwidth

Algorithm - MV Mapping & MC
Block-based through interpolation





Reference frame

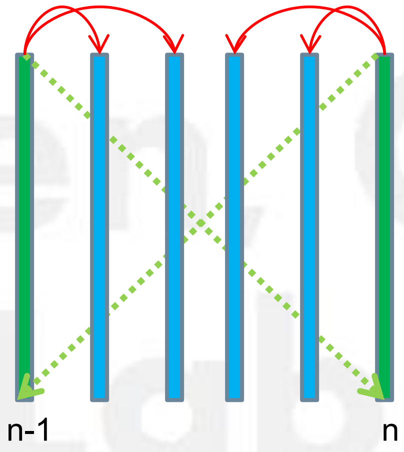
Inter frame

Current frame

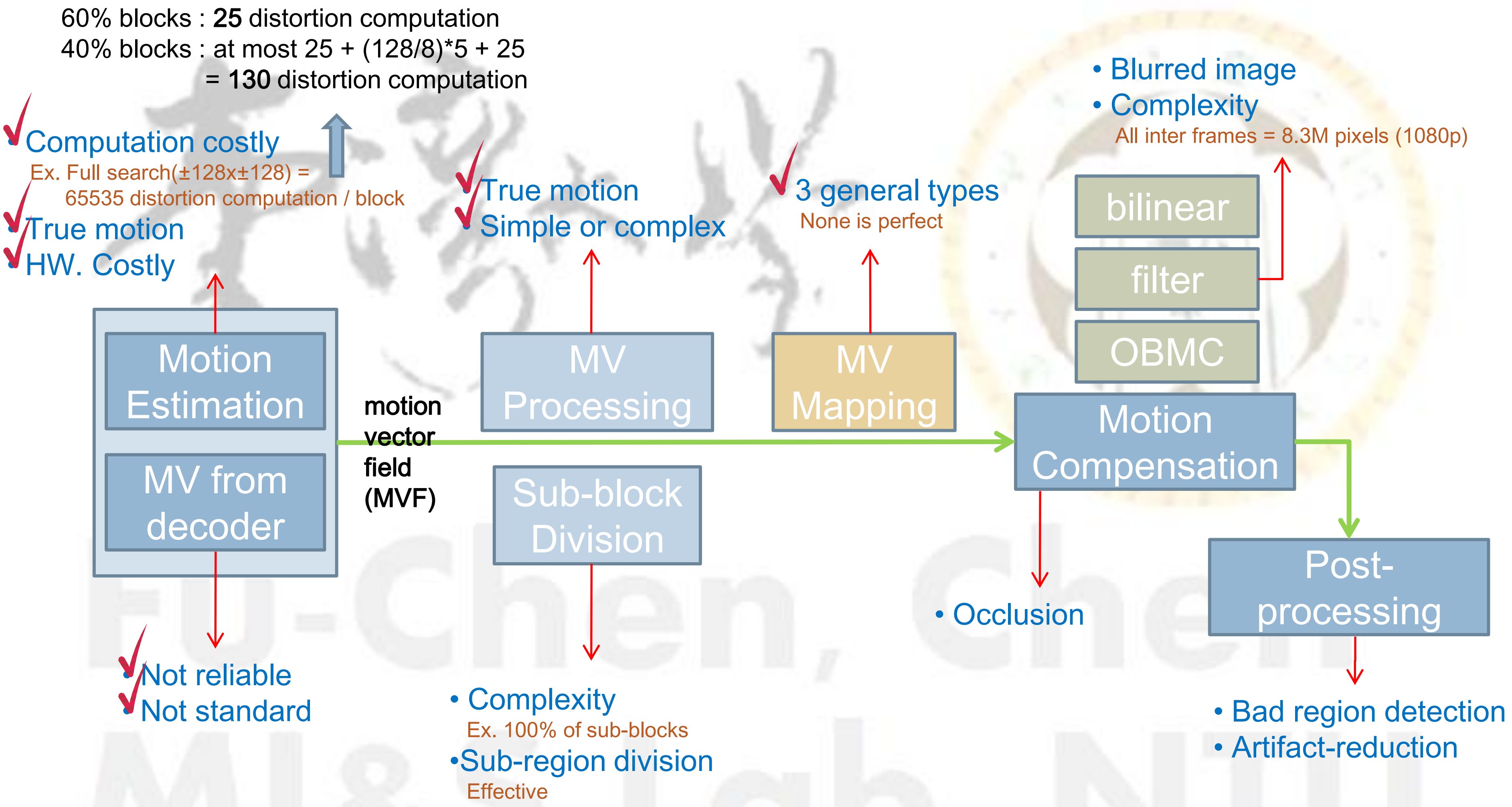
- Accumulate the total overlapped area of each MV
  - If max. area > ½ of block size
    - MV of inter frame's block = MV with max. area
  - Else
    - MV of inter frame's block = MV of co-located block

Algorithm - MV Mapping & MC
Block-based through interpolation

- Combine the advantages of tradition & through MV mapping
- For multi-frame MC
  - ME twice for two direction MVF
  - To get pixels from the nearest frame
  - Uni-directional interpolation to prevent being blur
    - Complexity is also lower



Algorithm - MV Mapping & MC
Block-based through interpolation



60% blocks : **25** distortion computation
40% blocks : at most 25 + (128/8)*5 + 25
= **130** distortion computation

- ✓ Computation costly
  Ex. Full search(±128x±128) = 65535 distortion computation / block
- ✓ True motion
- ✓ HW. Costly

Motion Estimation

MV from decoder

- ✓ Not reliable
- ✓ Not standard

motion vector field (MVF)

- ✓ True motion
- ✓ Simple or complex

MV Processing

Sub-block Division

- Complexity
  Ex. 100% of sub-blocks
- Sub-region division
  Effective

- ✓ 3 general types
  None is perfect

MV Mapping

- Blurred image
- Complexity
  All inter frames = 8.3M pixels (1080p)

bilinear

filter

OBMC

Motion Compensation

- Occlusion

Post-processing

- Bad region detection
- Artifact-reduction

* HW. = hardware
* BW. = bandwidth

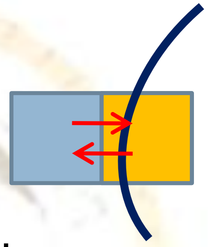
Algorithm - Sub-block Division
Precise artifact detection

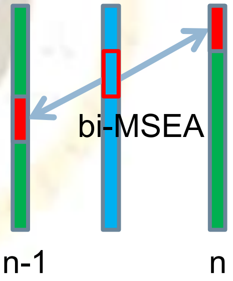
- Typical method
  - Sub-block dividing on exist frames
  - But artifact doesn’t appear on exist frames
- Our method
  - Sub-block dividing on **inter frames** where artifact appears

- The appearance of block artifact
  - Always appears when neighboring block’s MV is not continuous
  - Just detect the block where it’s MV is not continuous with the others
    - Simple & precise

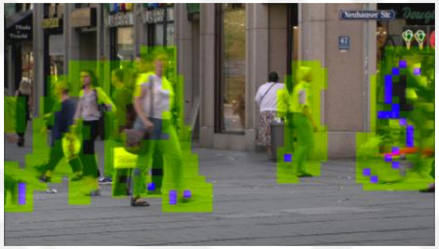
Algorithm - Sub-block Division
Precise artifact detection

- Artifact detecting condition
  - Only MV value
    - $|\text{MV\_x}_{cur} - \text{MV\_x}_{4\text{-neigh.}}|$ or $|\text{MV\_y}_{cur} - \text{MV\_y}_{4\text{-neigh.}}| >$ threshold
    - Will detect **2 blocks** sharing the same MV discontinuity boundary
      - Only one of them should be detected
  - MV value & bilateral MSEA (bi-MSEA) comparison
    - $|\text{MV\_x}_{cur} - \text{MV\_x}_{4\text{-neigh.}}|$ or $|\text{MV\_y}_{cur} - \text{MV\_y}_{4\text{-neigh.}}| >$ threshold & bigger bi-MSEA
    - Detect **only one block** of the same MV discontinuity boundary
    - More computation for bi-MSEA, but less # of sub-blocks

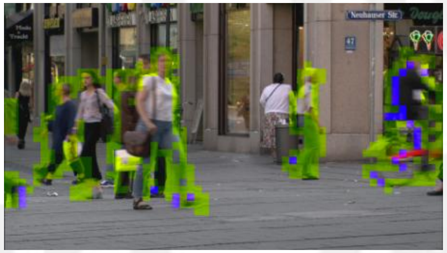




Detected 32x32 blocks (for 1080p)



Only MV value

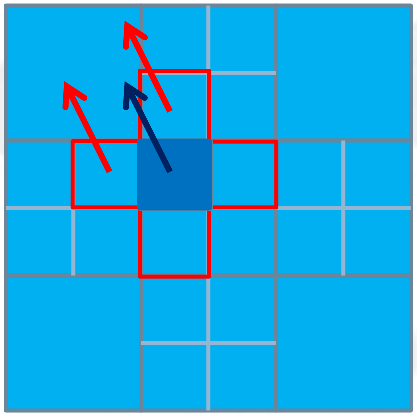


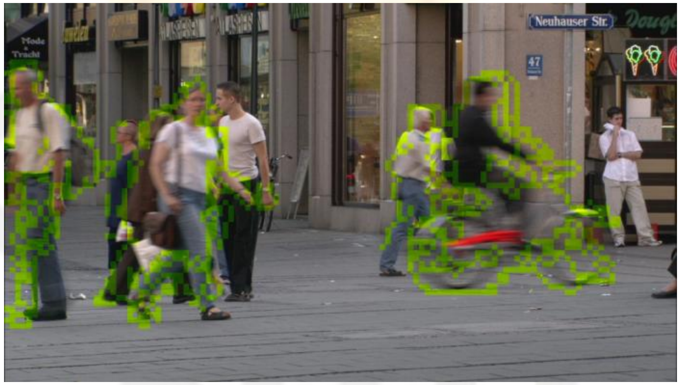
MV value & bi-MSEA

* threshold = 2 (for 1080p)
* neigh. = neighbor

Algorithm - Sub-block Division
Precise artifact detection

- Sub-block division & initial MV assignment
  - Regard current & neighboring blocks as 16x16 sub-blocks
  - Let $W_{4\text{-neigh.}}$ = 1 if detecting condition is satisfied
  - If $\Sigma W_{4\text{-neigh.}} > 0$
    - Divide & label this sub-block for next refinement
    - If bi-MSEA > threshold
      - Initial MV = $\Sigma\,(W_{4\text{-neigh.}} * MV_{4\text{-neigh.}}) / \Sigma W_{4\text{-neigh.}}$
    - Else
      - Initial MV = original MV





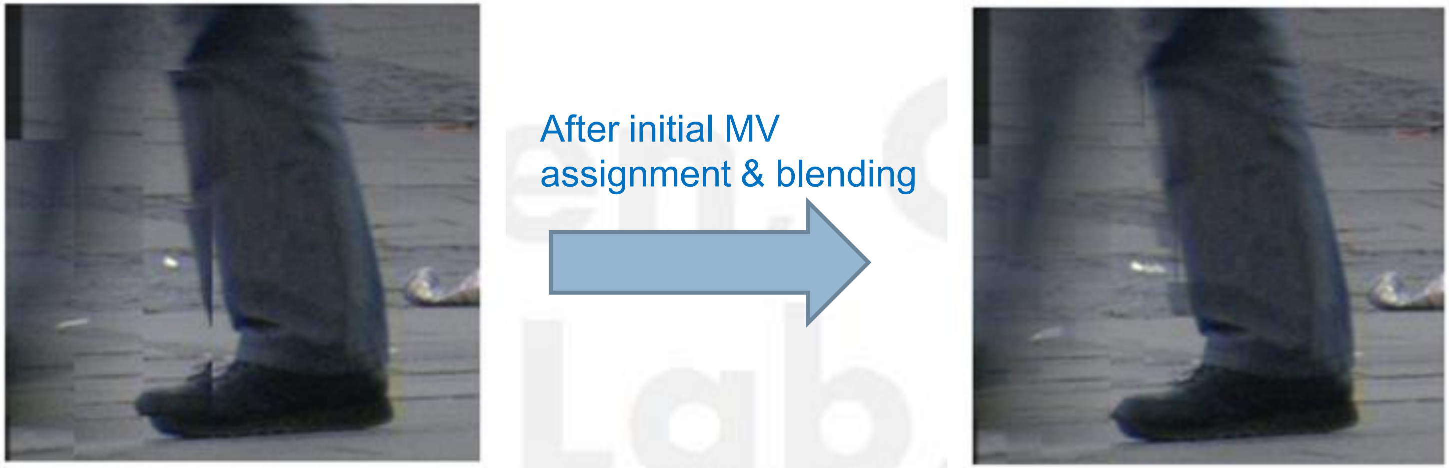
Labeled 16x16 sub-blocks (for 1080p)

Algorithm - Sub-block Division Precise artifact detection

- # of labeled sub-blocks
  - About 12% for the worst case
- Why assign initial MV
  - Instead of re-estimate sub-blocks
    - Lower the sub-block MV searching cost
  - True motion comes from nearby blocks

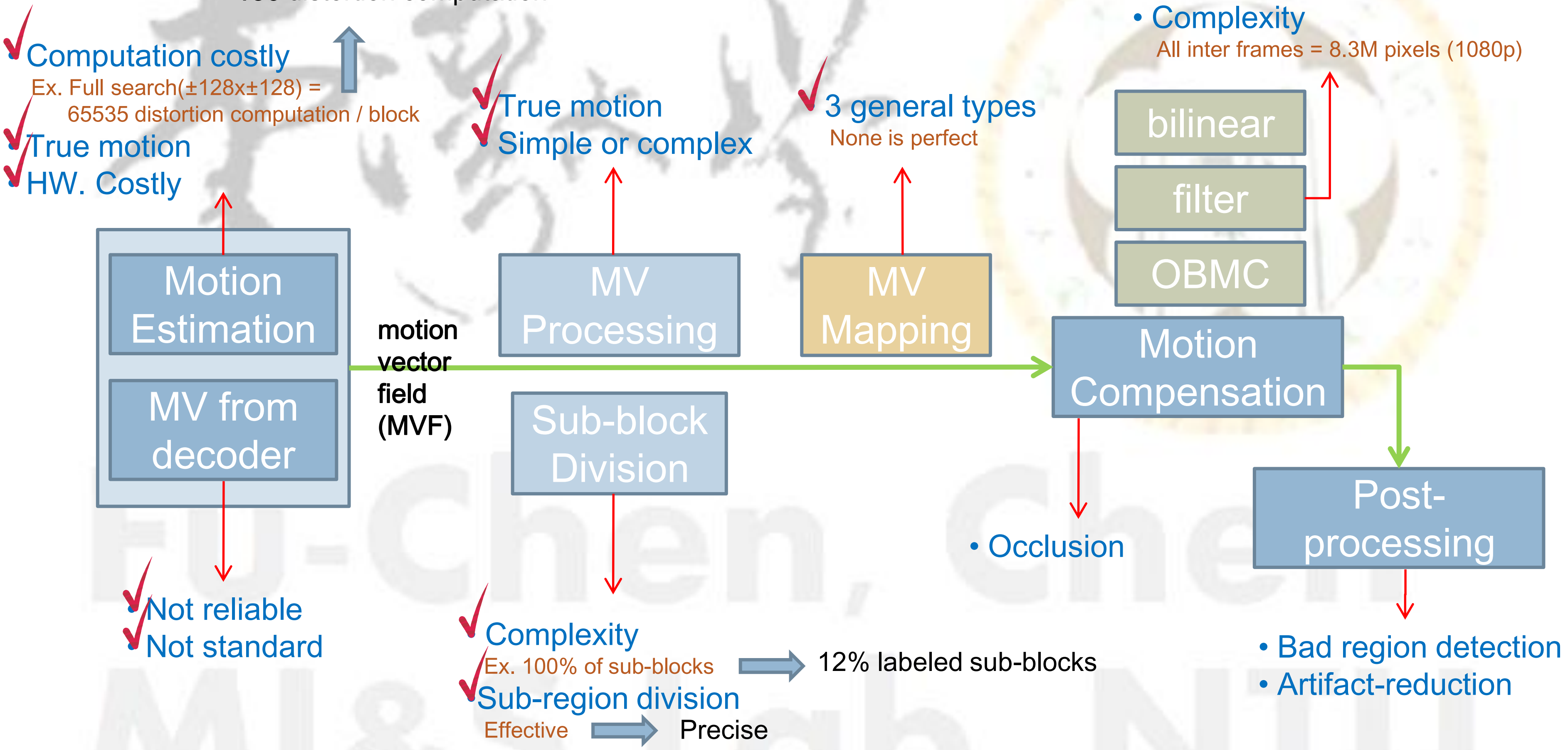
# of labeled sub-blocks
(worst cases for each sequence)

total # of sub-blocks	32640	100.00%
pedestrian_area	2787	8.54%
Titanic-2	1820	5.58%
vintagecar	3074	9.42%
ducks_take_off	1340	4.11%
park_joy	2474	7.58%
tractor	1969	6.03%
transformer 7-3	3947	12.09%



Algorithm - Sub-block Division
Precise artifact detection

60% blocks : **25** distortion computation
40% blocks : at most 25 + (128/8)*5 + 25
= **130** distortion computation



✓ Computation costly
Ex. Full search(±128x±128) = 65535 distortion computation / block
✓ True motion
✓ HW. Costly

Motion Estimation

MV from decoder

✓ Not reliable
✓ Not standard

motion vector field (MVF)

✓ True motion
✓ Simple or complex

MV Processing

Sub-block Division

✓ Complexity
Ex. 100% of sub-blocks ⇒ 12% labeled sub-blocks
✓ Sub-region division
Effective ⇒ Precise

✓ 3 general types
None is perfect

MV Mapping

- Blurred image
- Complexity

All inter frames = 8.3M pixels (1080p)

bilinear

filter

OBMC

Motion Compensation

- Occlusion

Post-processing

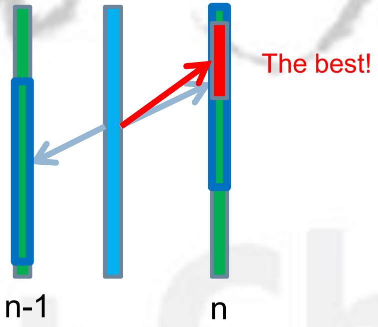
- Bad region detection
- Artifact-reduction

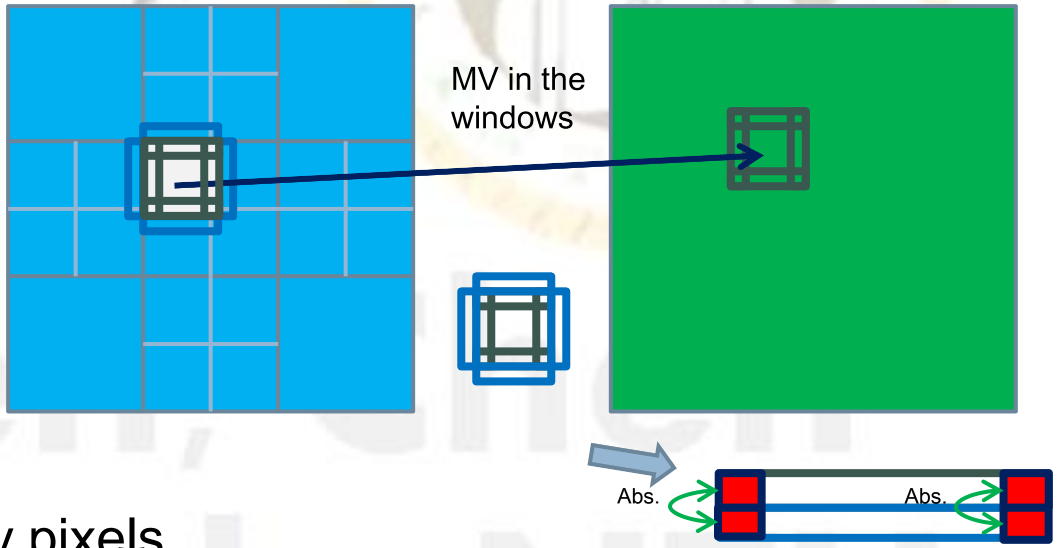
* HW. = hardware
* BW. = bandwidth

Algorithm - Post-processing
Bilateral sub-block refinement

* Abs. = absolute difference
* BE. = boundary error

- Find new MV for all labeled sub-blocks
  - Around the initial MV & its opposite direction
    - Open two search windows : ±8 x ±8
    - Compare **BE.** , find the best MV



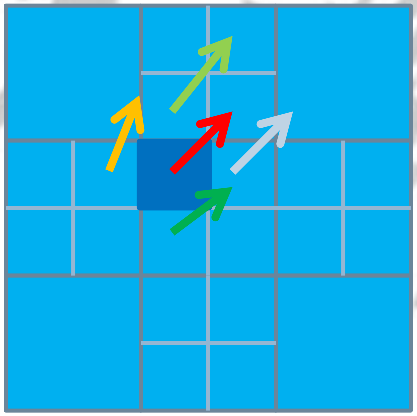


BE. = Σ Abs.

  - Interpolate the boundary pixels
    - For next sub-block's comparison

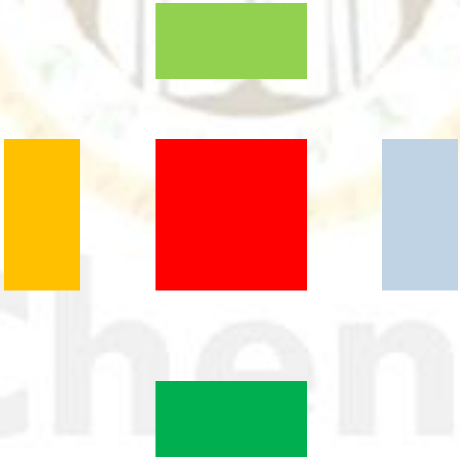
Algorithm - Post-processing
Bilateral sub-block refinement

□ OBMC on all labeled sub-blocks



2	3	4	5	6	7	8	9	9	8	7	6	5	4	3	2
3	4	5	6	7	8	9	10	10	9	8	7	6	5	4	3
4	5	6	7	8	9	10	11	11	10	9	8	7	6	5	4
5	6	7	8	9	10	11	12	12	11	10	9	8	7	6	5
6	7	8	9	10	11	12	13	13	12	11	10	9	8	7	6
7	8	9	10	11	12	13	14	14	13	12	11	10	9	8	7
8	9	10	11	12	13	14	15	15	14	13	12	11	10	9	8
9	10	11	12	13	14	15	16	16	15	14	13	12	11	10	9
9	10	11	12	13	14	15	16	16	15	14	13	12	11	10	9
8	9	10	11	12	13	14	15	15	14	13	12	11	10	9	8
7	8	9	10	11	12	13	14	14	13	12	11	10	9	8	7
6	7	8	9	10	11	12	13	13	12	11	10	9	8	7	6
5	6	7	8	9	10	11	12	12	11	10	9	8	7	6	5
4	5	6	7	8	9	10	11	11	10	9	8	7	6	5	4
3	4	5	6	7	8	9	10	10	9	8	7	6	5	4	3
2	3	4	5	6	7	8	9	9	8	7	6	5	4	3	2

Weighted sum

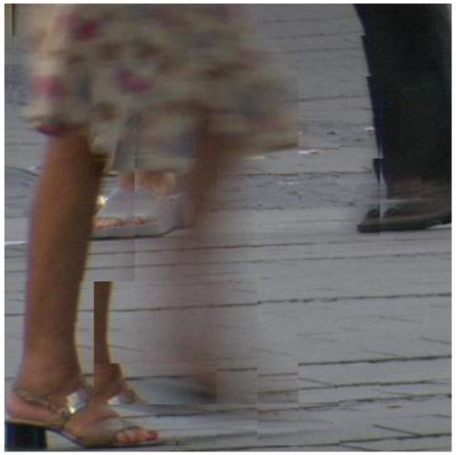


7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

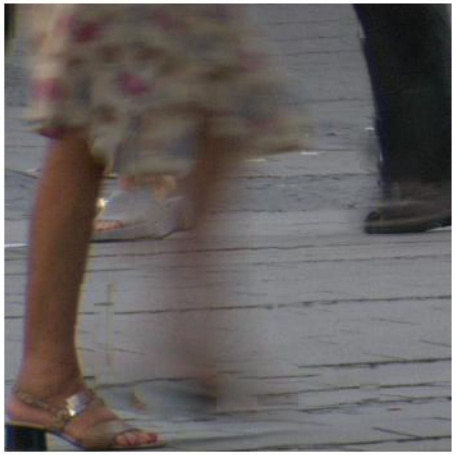
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7

Algorithm - Post-processing
Bilateral sub-block refinement

- The reasons
  - Open a window in opposite direction
    - For occlusive area
  - Boundary error comparison
    - To find the MV with the least block artifact
  - OBMC on all labeled sub-blocks
    - To further reduce block artifact
    - Will not cause blur when neighboring MVs are similar

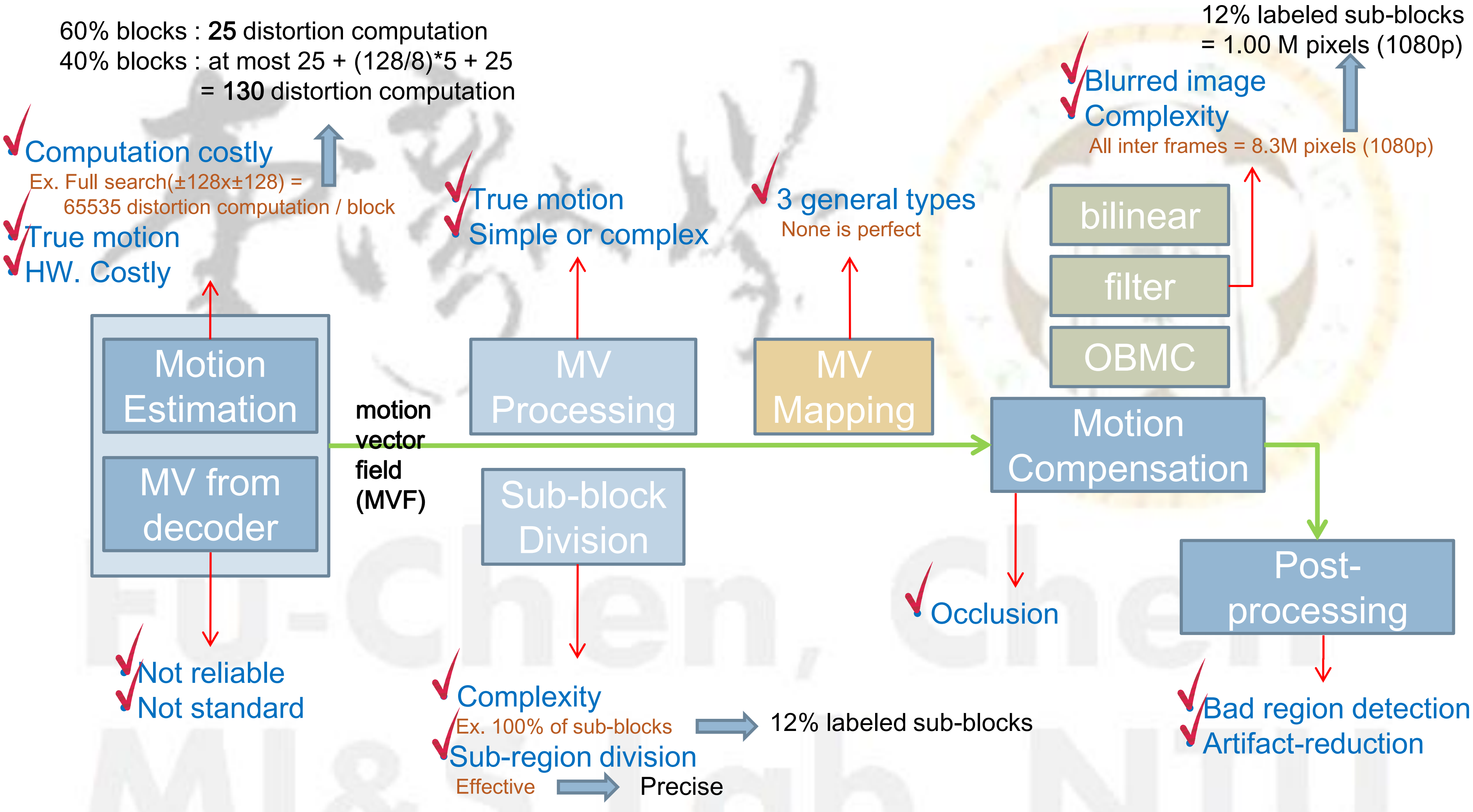


before
post-processing



after
post-processing

Algorithm - Post-processing
Bilateral sub-block refinement



60% blocks : **25** distortion computation
40% blocks : at most 25 + (128/8)*5 + 25
= **130** distortion computation

12% labeled sub-blocks
= 1.00 M pixels (1080p)

✓ Blurred image
✓ Complexity
All inter frames = 8.3M pixels (1080p)

✓ Computation costly
Ex. Full search(±128x±128) =
65535 distortion computation / block
✓ True motion
✓ HW. Costly

✓ True motion
✓ Simple or complex

✓ 3 general types
None is perfect

bilinear
filter
OBMC

Motion Estimation
MV from decoder

motion vector field (MVF)

MV Processing

MV Mapping

Motion Compensation

Sub-block Division

✓ Occlusion

Post-processing

✓ Not reliable
✓ Not standard

✓ Complexity
Ex. 100% of sub-blocks → 12% labeled sub-blocks
✓ Sub-region division
Effective → Precise

✓ Bad region detection
✓ Artifact-reduction

* HW. = hardware
* BW. = bandwidth