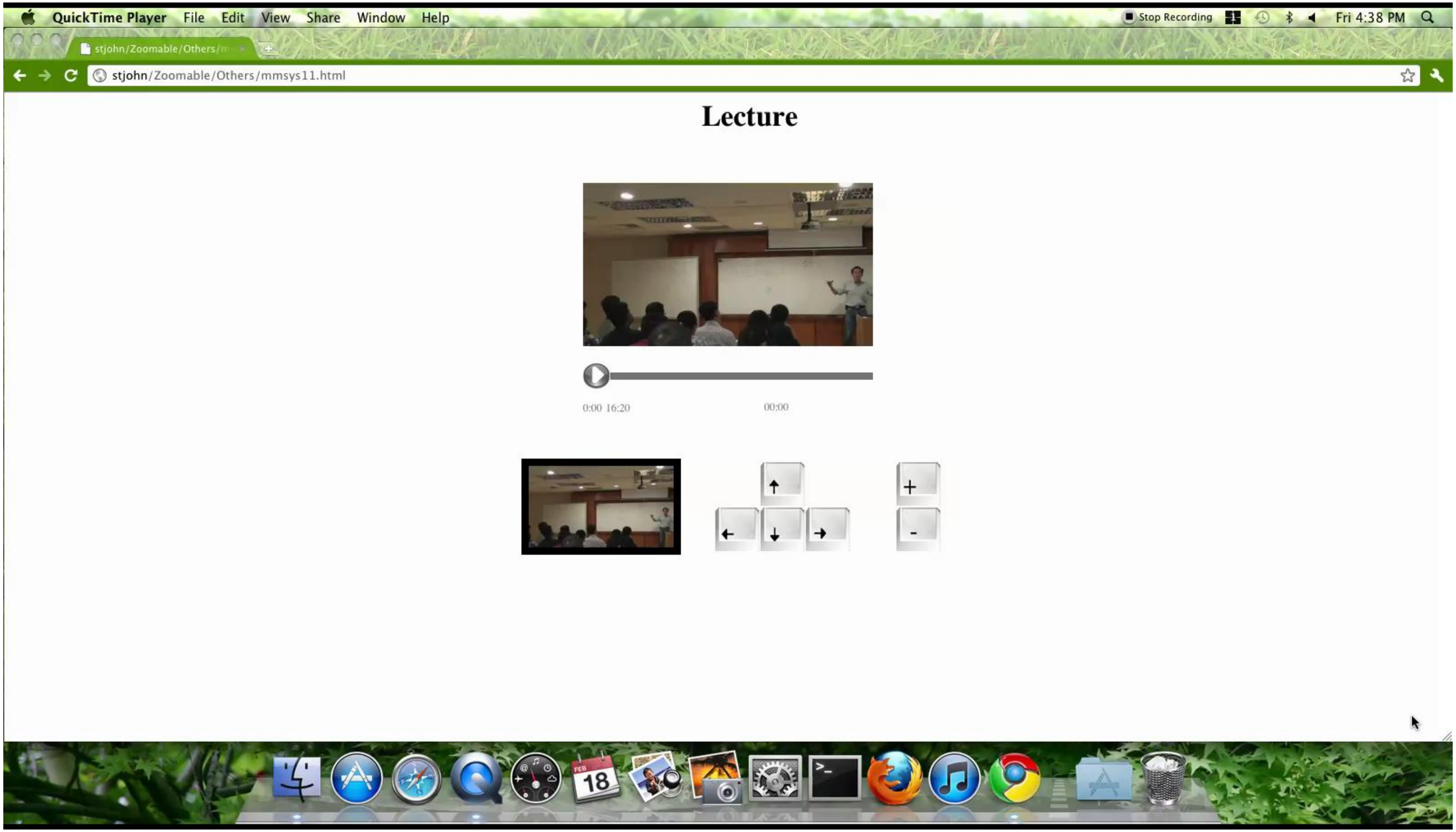
Adaptive Encoding of Zoomable Video Streams based on User Access Pattern

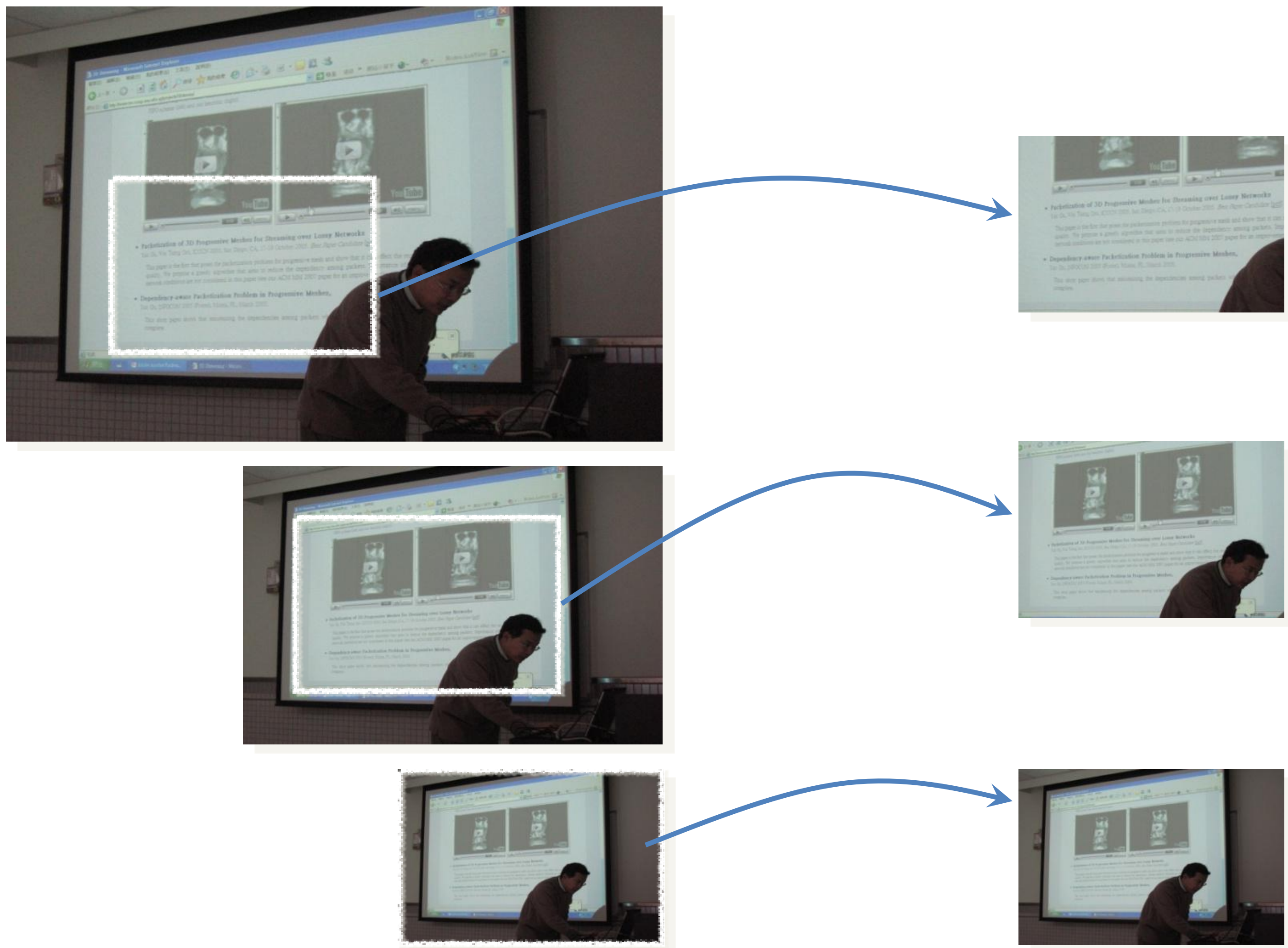
Ngo Quang Minh Khiem
Guntur Ravindra
Wei Tsang Ooi

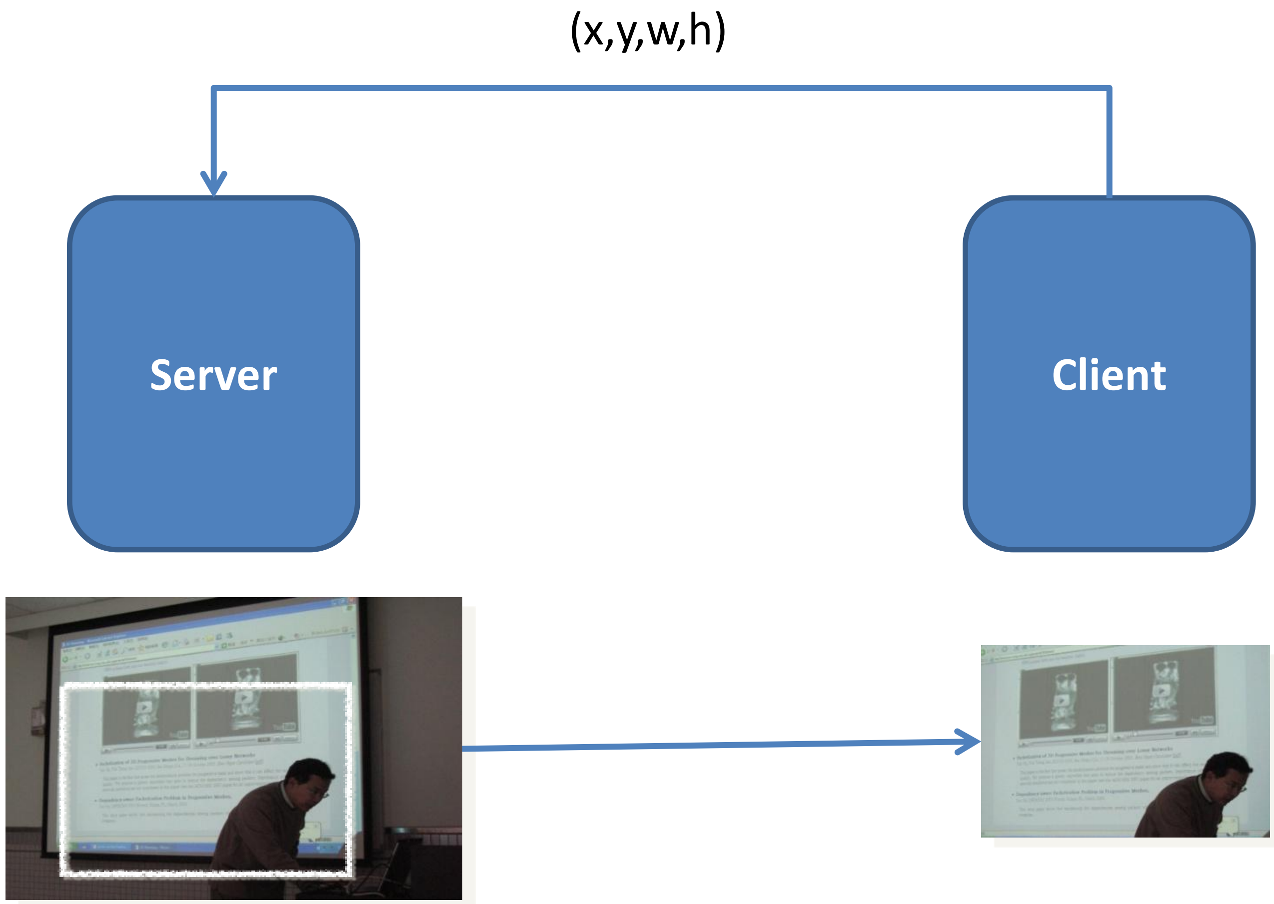
National University of Singapore

Zoomable Video



Zoomable Video with Bitstream Switching





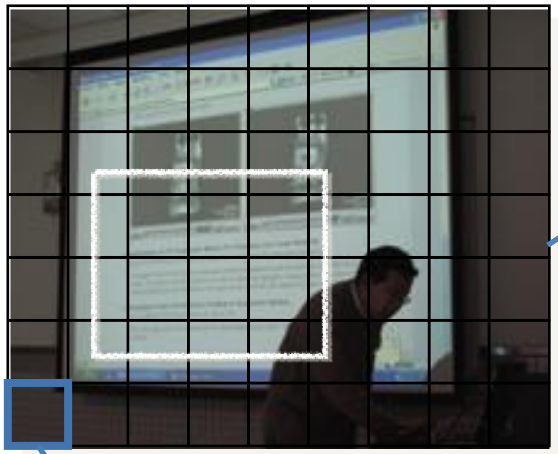
GOAL: Minimize bandwidth to transmit RoIs

Dynamic Cropping of ROI

Encode video once
Support any RoI cropping

Tiled Streaming (TS)

Monolithic Streaming (MS)

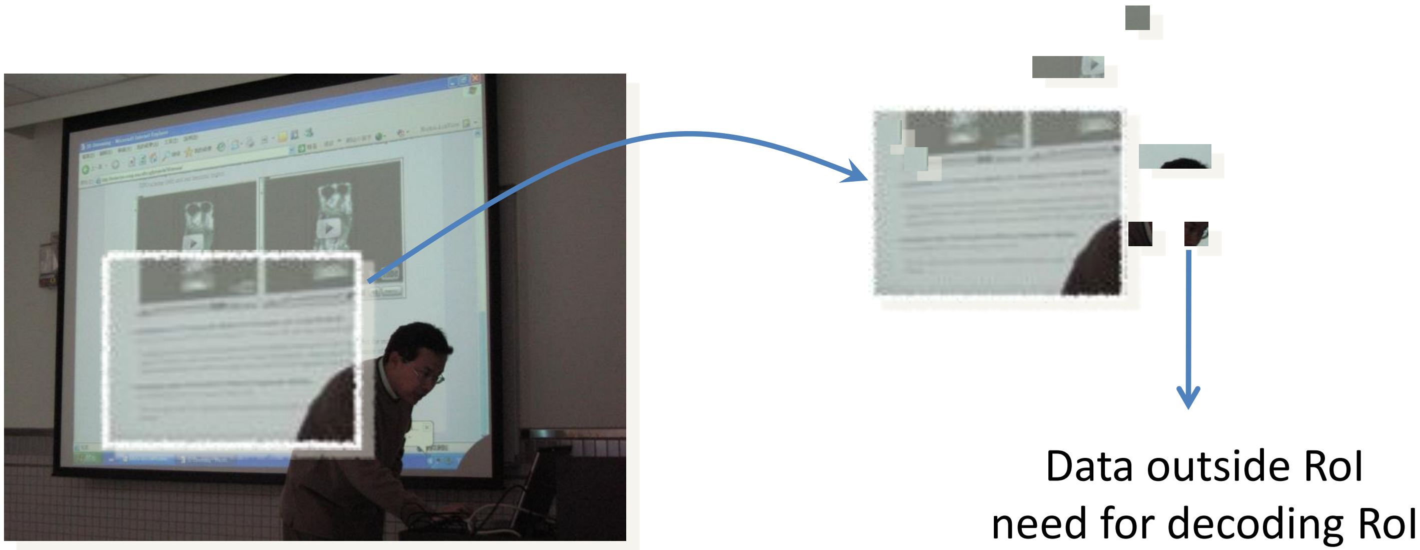


One tile = k x k macroblocks

Tiles overlapping with
the RoI are
transmitted

Encode each
tile as independantly
decodable video streams

Tiled Streaming



Single monolithic video

Monolithic Streaming

Trade-offs with TS and MS

TS

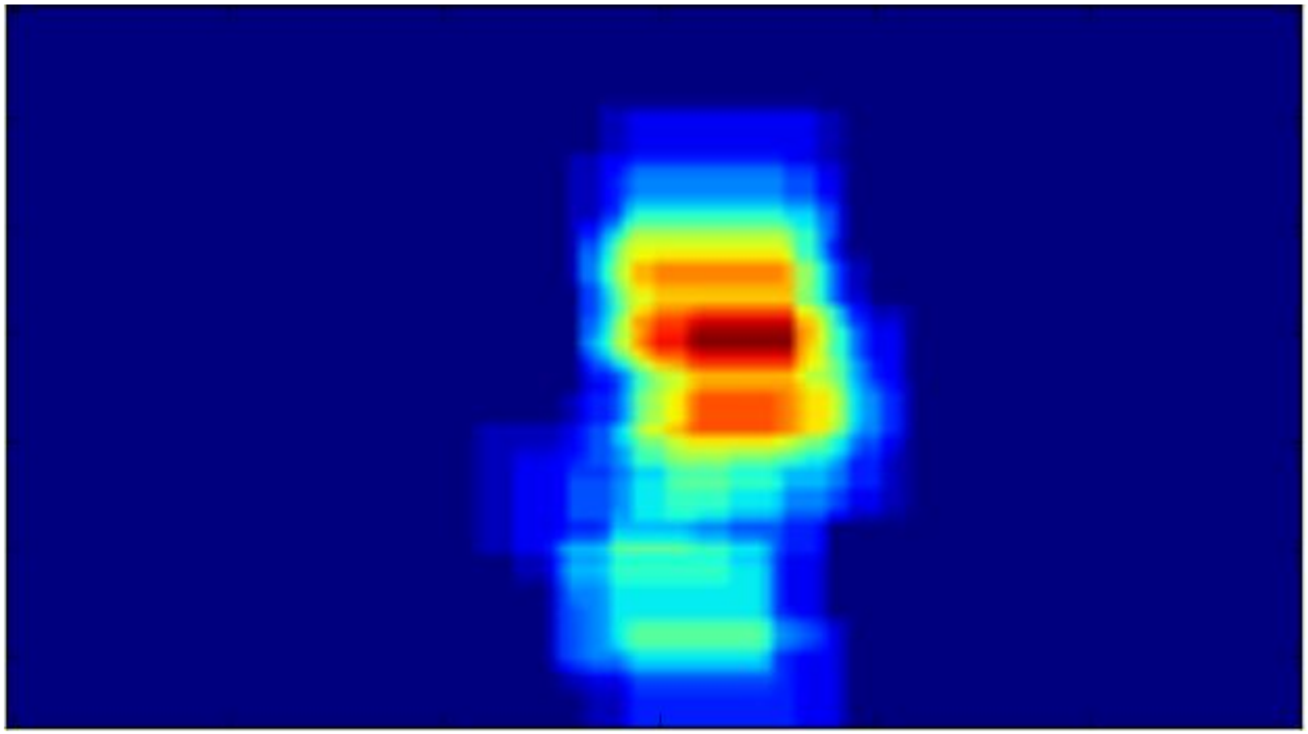
Bigger tile → More waste →**More bits**

Smaller tile → Less compression →**More bits**

MS

- **Longer MV** → More dependency →**More bits**
- **Shorter MV** → Less compression →**More bits**

RoI Access Pattern



Reduce bandwidth further, given RoI access statistics?

Questions in this paper

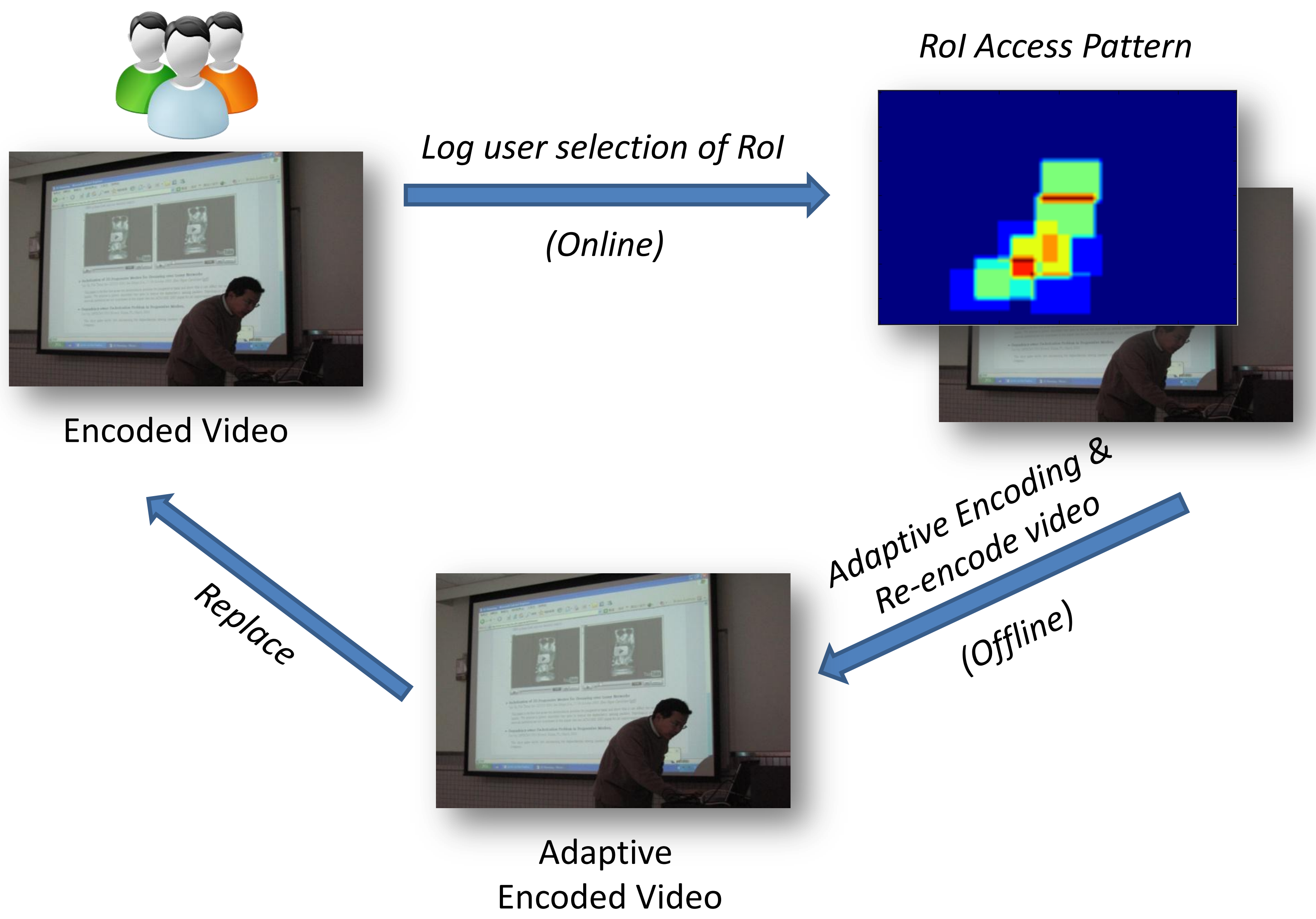
- Tiled Streaming
  - Different tile size in the same frame?
- Monolithic Streaming
  - Different motion search range?
- How?

Adaptive Encoding

Given RoI access statistics, adapt the encoding parameters such that the *expected bandwidth E* needed to transmit a RoI is minimized

$$E = \sum_{r \in R} c(r)\, p(r)$$

c(r): compressed size of RoI r
p(r): access probability of RoI r


RoI Access Pattern
Log user selection of RoI
(Online)
Encoded Video
Adaptive Encoding &
Re-encode video
(Offline)
Replace
Adaptive
Encoded Video

Adaptive Encoding

Adaptive Tiling (AT)

Monolithic Streaming with RoI-aware Coding (MS-PB)

Adaptive Tiling

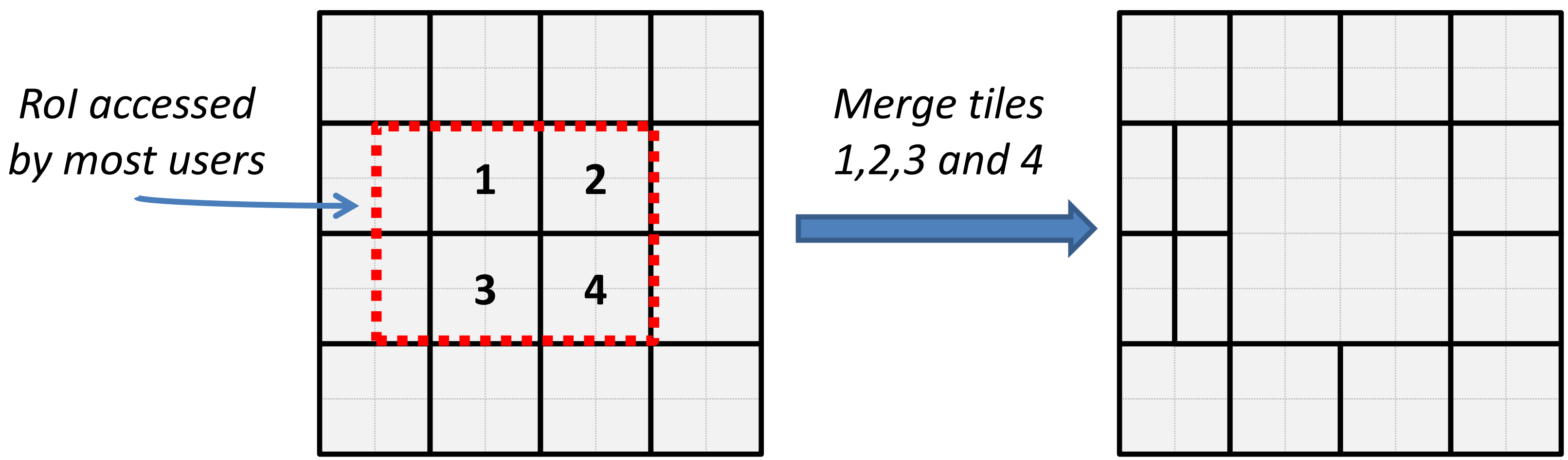
Given RoI access pattern, tile the video such that E is minimized

$$E = \sum_{t \in T} c(t)\, p(t)$$

c(t): compressed size of tile t
p(t): access probability of tile t

Intuition

Allowing tiles of different sizes can reduce bandwidth

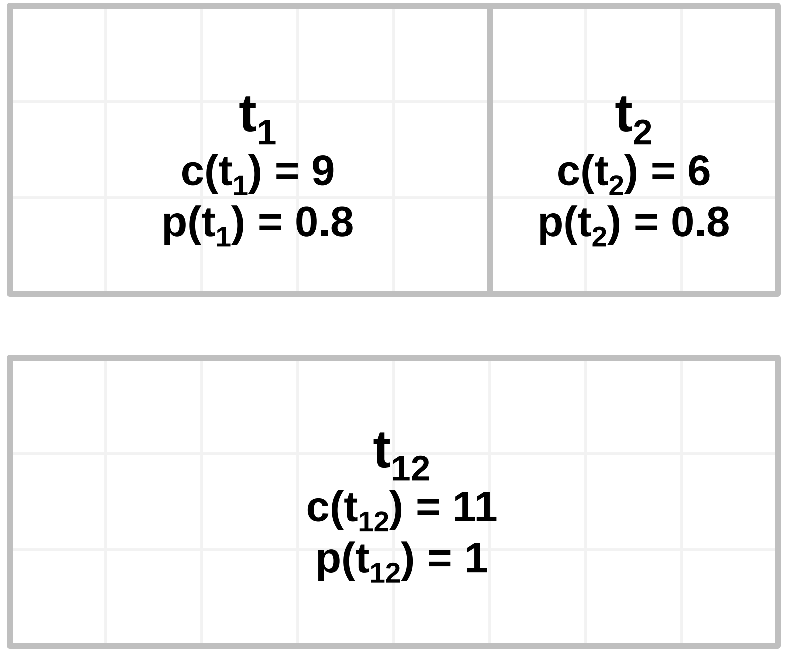


Regular tiling with 2x2 tiles

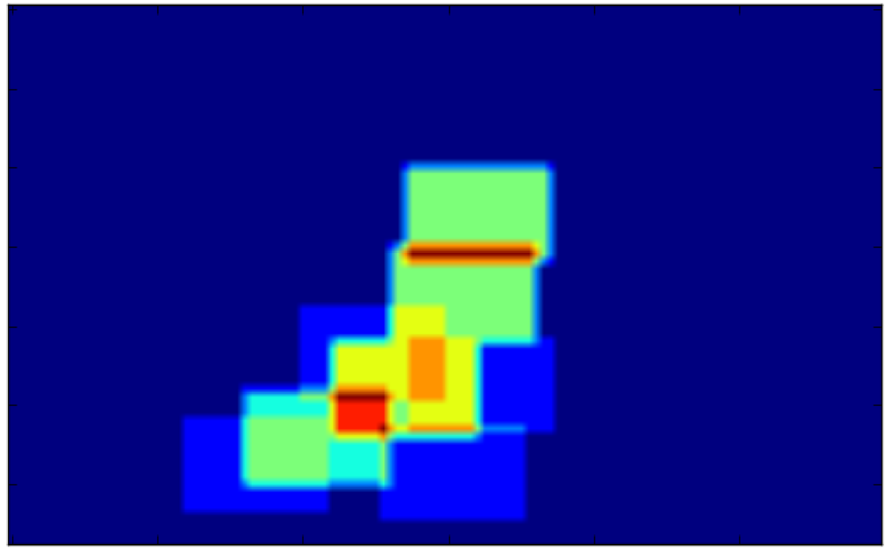
Adaptive tiling

Greedy Heuristic Tiling

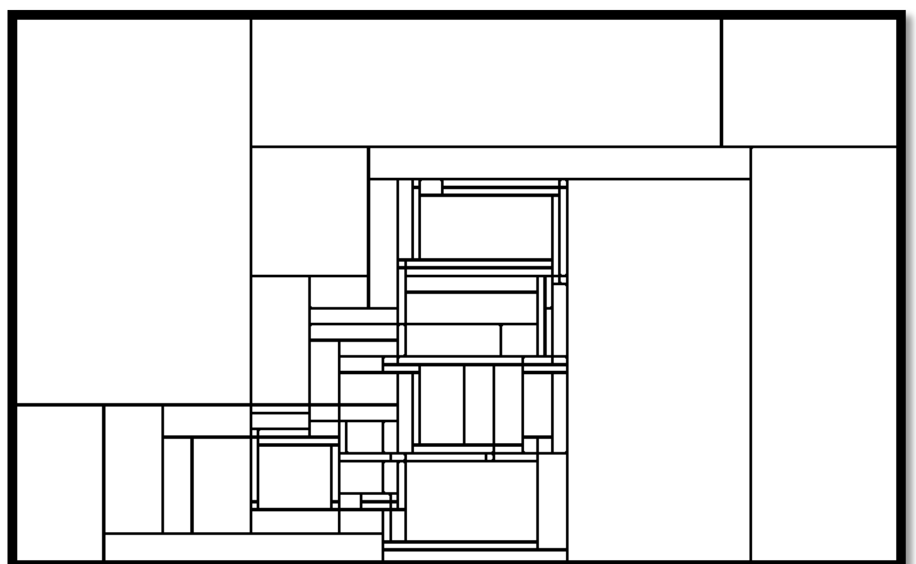
- Start with regular 1x1 tiles
- Merge a tile with its neighbors if expected bandwidth is reduced
- Merge newly-formed tile with its neighbors bandwidth is reduced



$$p(t_1)c(t_1) + p(t_2)c(t_2) \geq p(t_{12})c(t_{12})$$



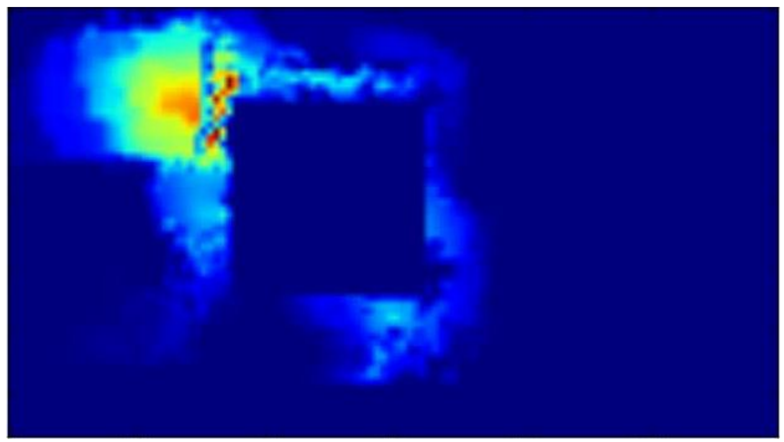
RoI Access Pattern



Resulting tile map

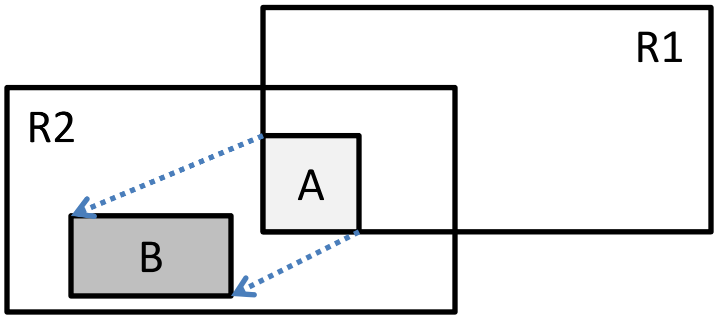
Monolithic Streaming with RoI-aware Coding

- Referenced MBs form large region outside RoI



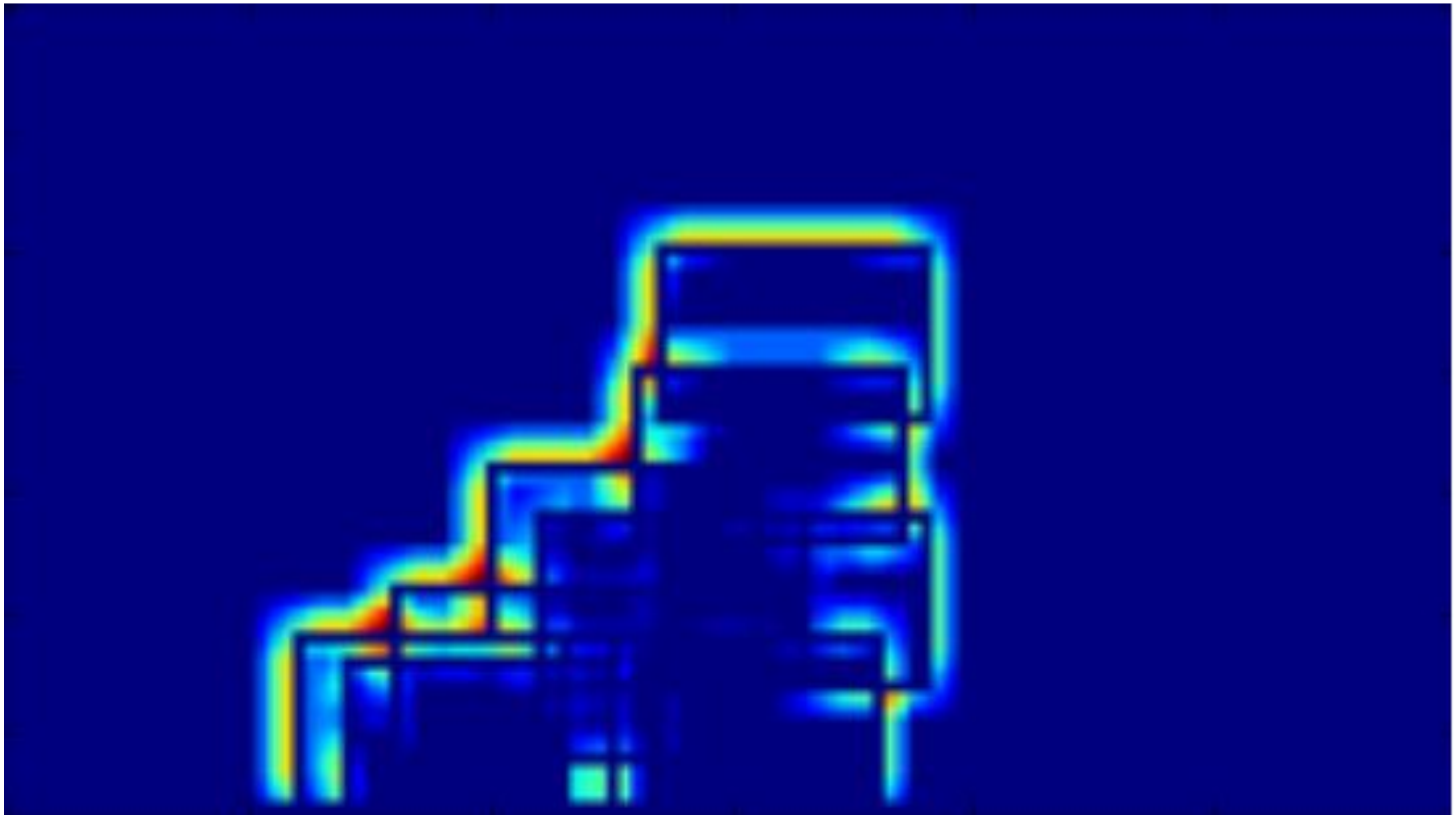
- Short motion vector: less bandwidth efficient
- Probabilistic boxing motion vector (MS-PB)

Intuition



P(A), P(B): sending A, B
P(AB) : A and B in same RoI
P(A) – P(AB): sending A independent of B

- $P(A) - P(AB) > P(B)$
  - Increase in size of A when sending R2 is marginal
- $P(A) - P(AB) < P(B)$
  - Increase in size of A when sending R2 is higher
- $[P(A)\text{-}P(AB)]\ \overline{S(A)} > P(B)\ S(B)$



Motion Vector Spread after MS-PB

Evaluation

- Evaluate AT and MS-PB in terms of
  - Bandwidth efficiency
  - Compression efficiency
- Benchmark methods
  - Per-RoI
  - Tiled Streaming
  - Monolithic Streaming

Video Sequences

Rush-Hour (500 frames)

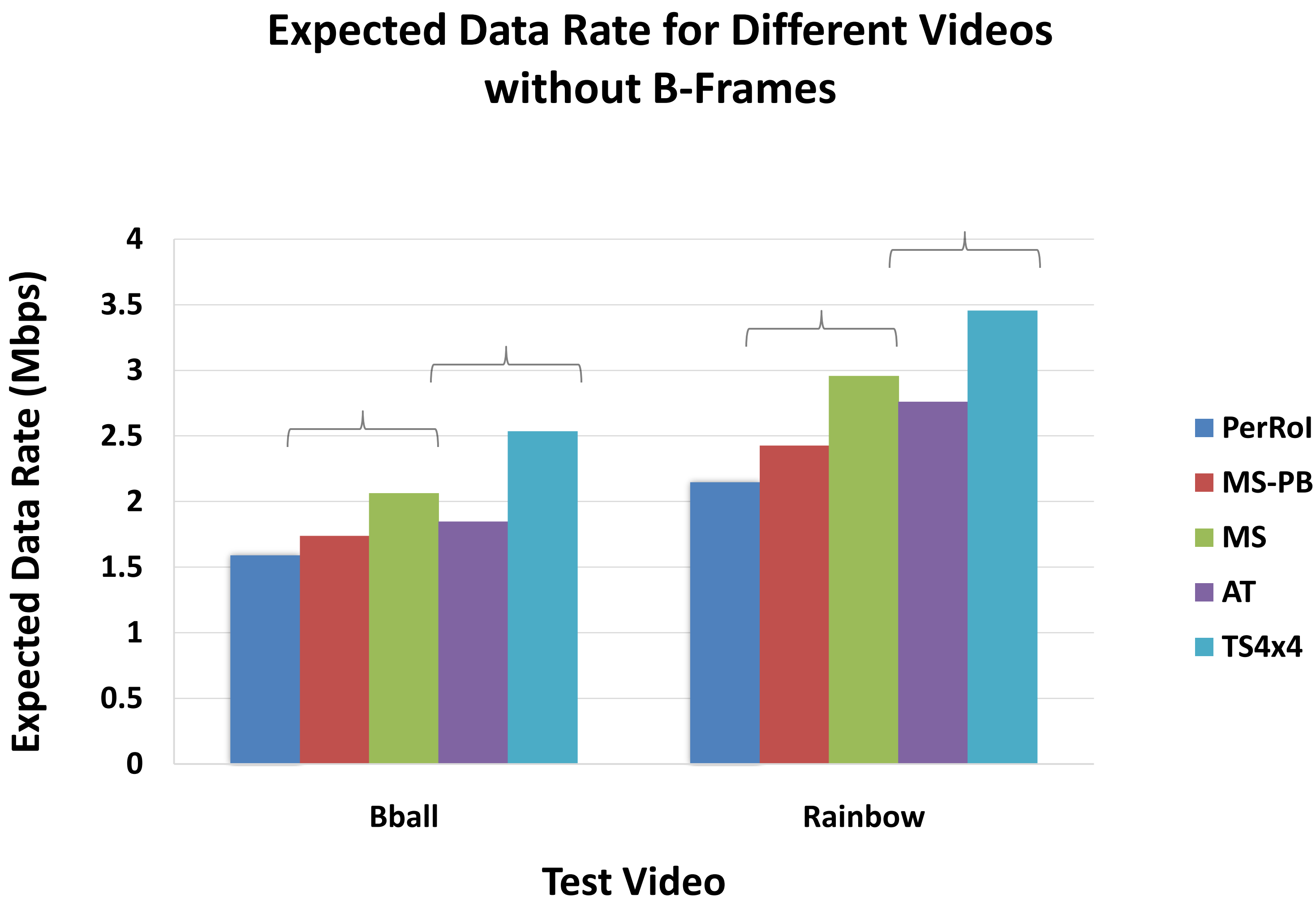
Tractor (688 frames)

Bball (200 frames)

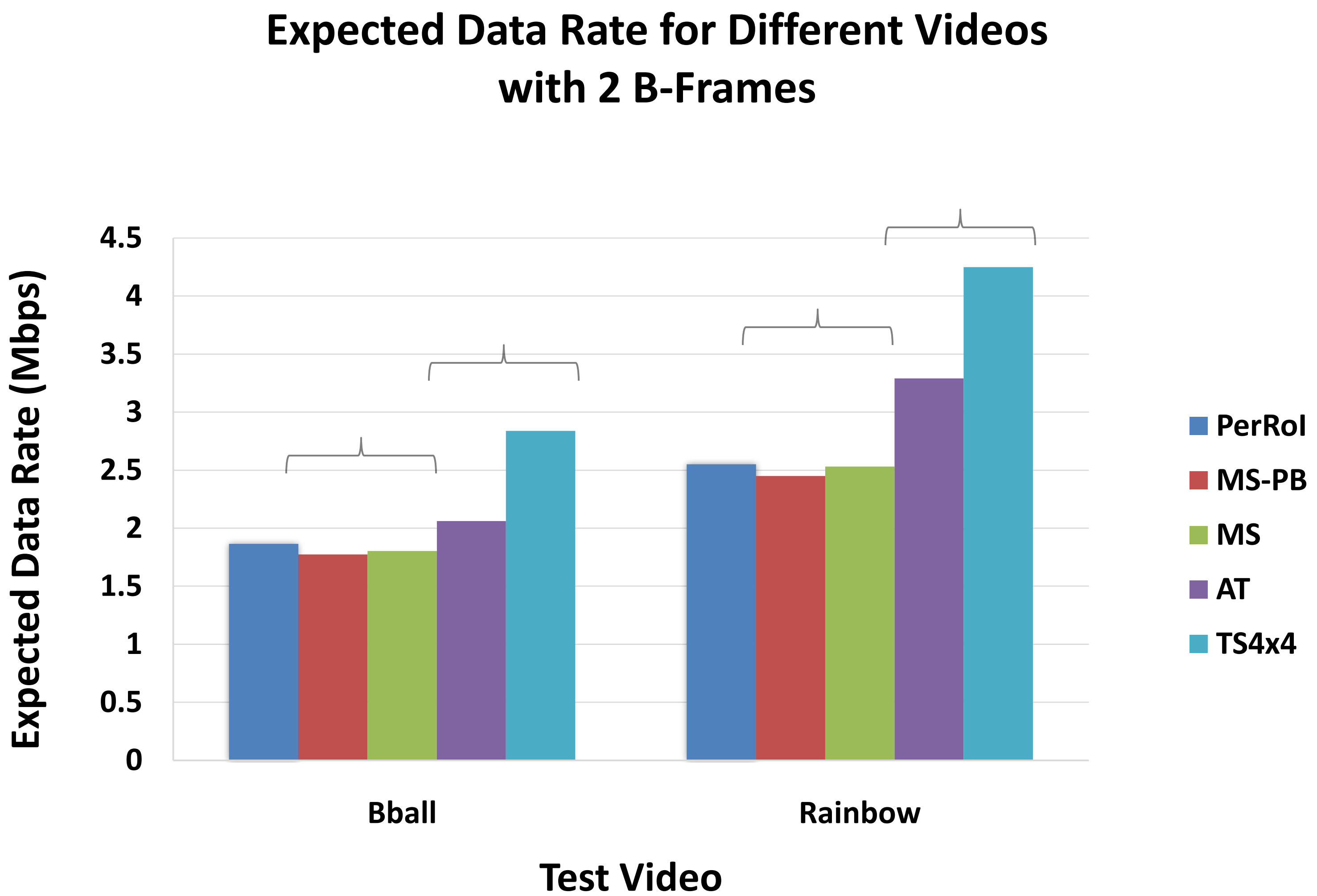
Rainbow (350 frames)

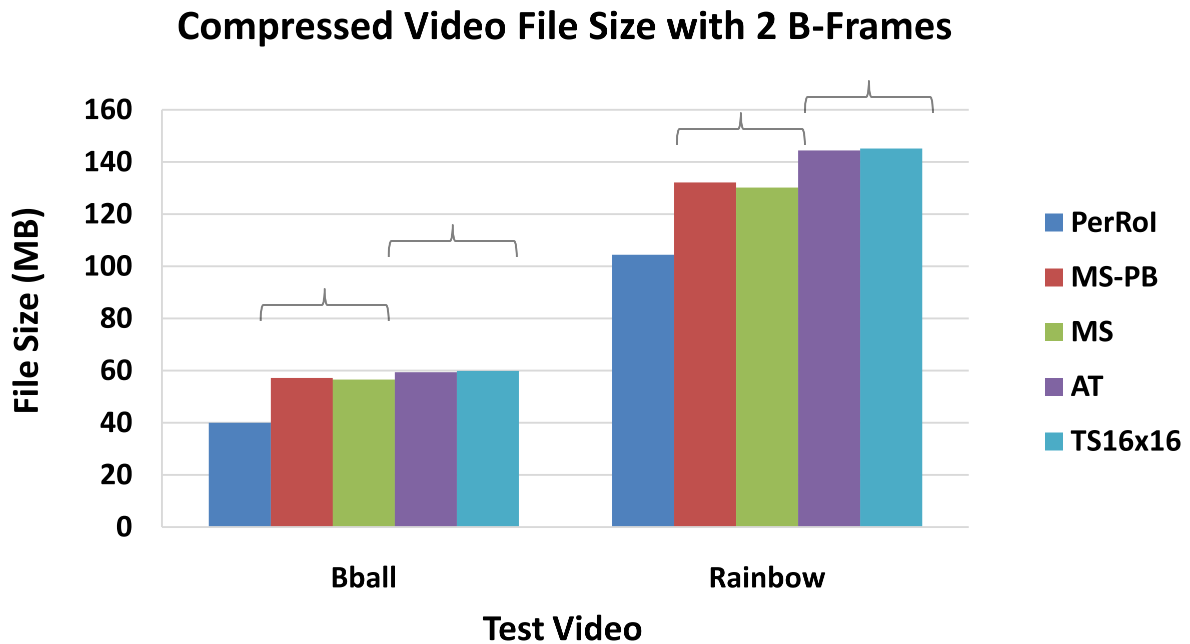
Experiment Setup

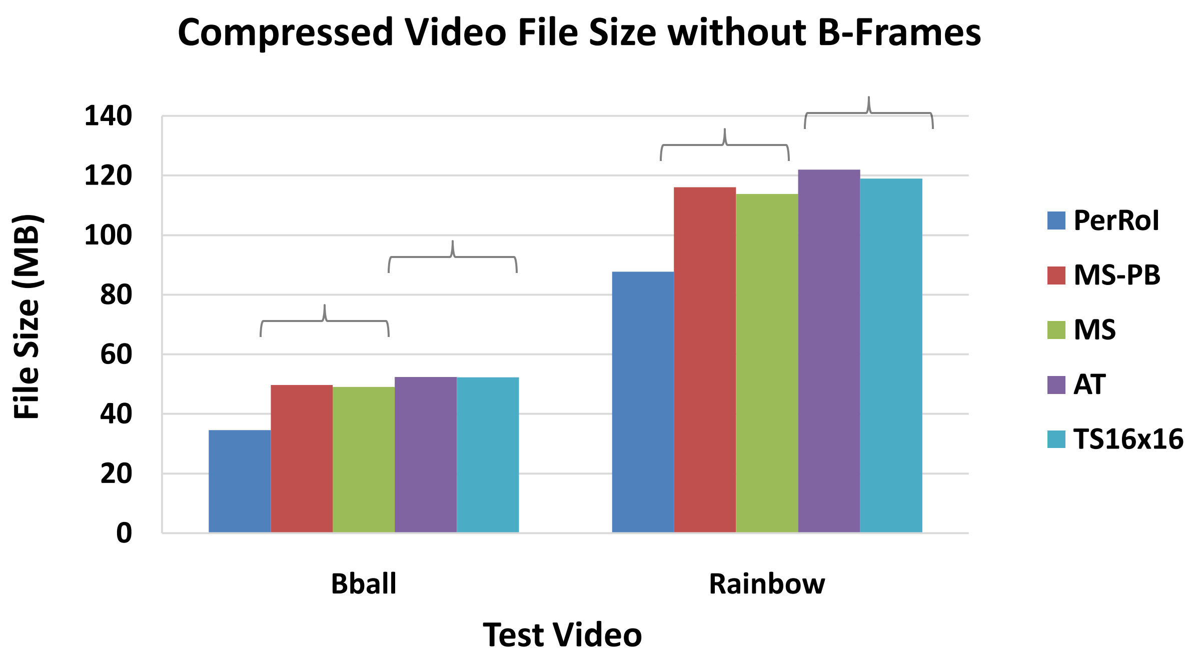
- RoI size: 320x192 pel
- Video resolution 1920x1080 pel
- Evaluation is conducted by a training-testing framework
  - Training and test sets have the same distribution
- One training and test set for each GoP



Expected Data Rate for Different Videos
without B-Frames
Expected Data Rate (Mbps)
4
3.5
3
2.5
2
1.5
1
0.5
0
Bball
Rainbow
Test Video
PerRoI
MS-PB
MS
AT
TS4x4


Expected Data Rate for Different Videos
with 2 B-Frames
Expected Data Rate (Mbps)
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
Bball
Rainbow
Test Video
PerRoI
MS-PB
MS
AT
TS4x4

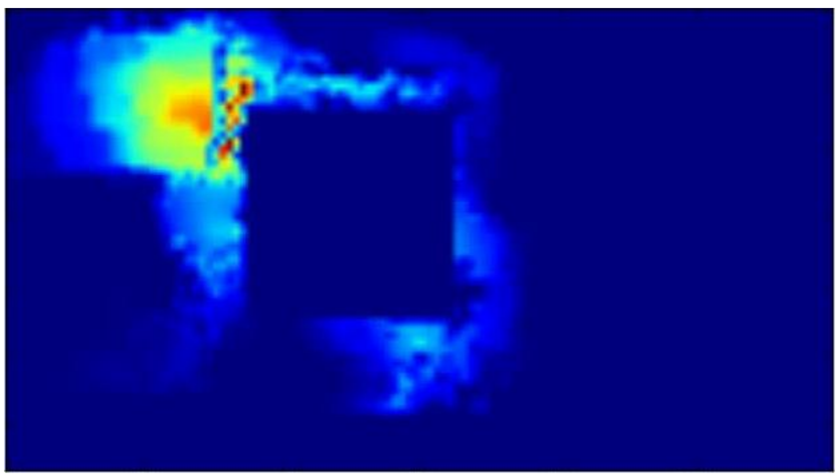

Compressed Video File Size with 2 B-Frames
File Size (MB)
0
20
40
60
80
100
120
140
160
Bball
Rainbow
Test Video
PerRoI
MS-PB
MS
AT
TS16x16


Compressed Video File Size without B-Frames
File Size (MB)
0
20
40
60
80
100
120
140
Bball
Rainbow
Test Video
PerRoI
MS-PB
MS
AT
TS16x16

Presence of B-frame

Without B-frame

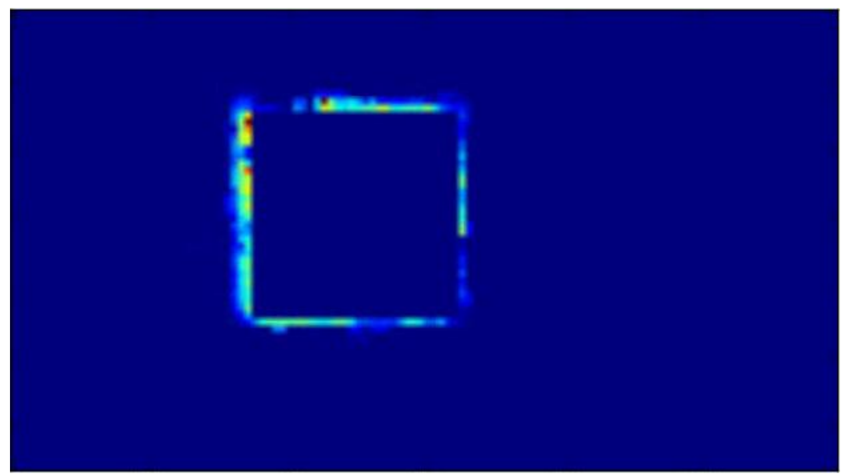
MS-PB < MS



Motion Vector Spread without B-frame

With B-frame

MS-PB ≈ MS



Motion Vector Spread with 2 B-frame

Conclusion & Future Work

- Propose an adaptive encoding approach based on user access patterns
- Reduce bandwidth by 21% (MS-PB) and 27% (AT)
- Limiting motion vector is beneficial to zoomable video with wide spread of dependency
- Future work:
  - Computational complexity
  - Diverse user interest of RoI
  - Frequency of Adaptation

Thank you

- Questions?
- Feedback/Suggesetion?