Zoomable Video

黃瑋璨

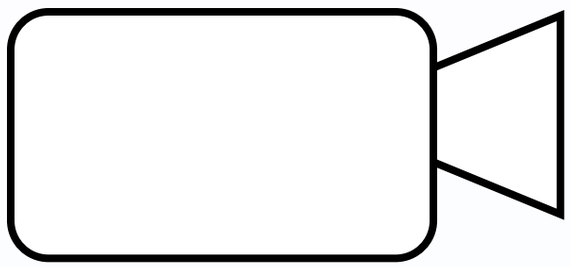
新加坡國立大學

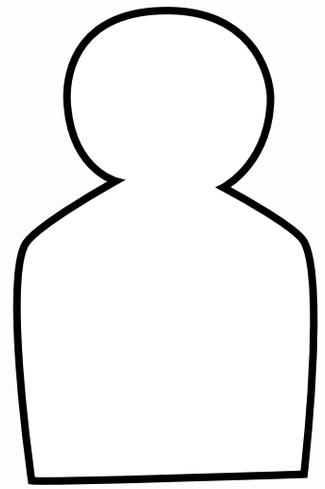
Wei Tsang **Ooi**

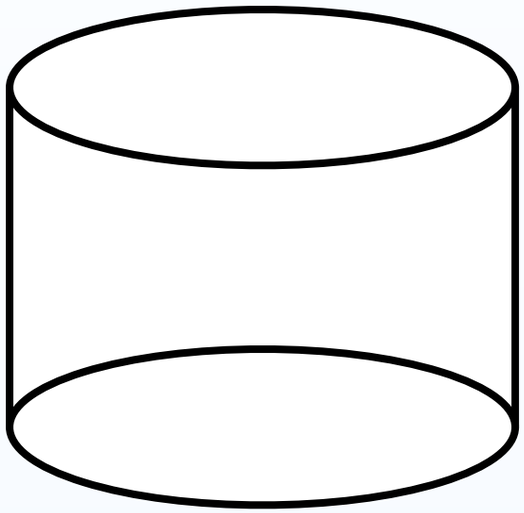
National University of Singapore

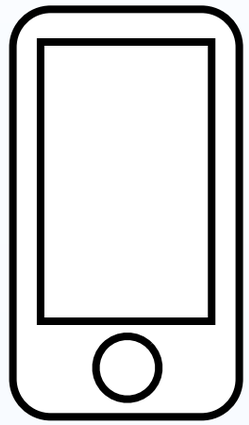
2009 - now

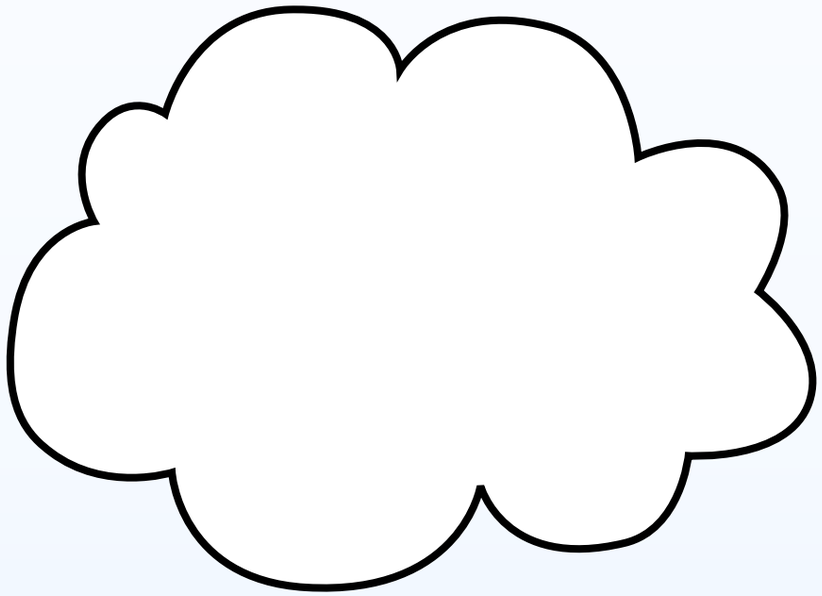
Guntur Ravindra, NUS
Ngo Quang Minh Khiem, NUS
Liu Feipeng, NUS
Axel Carlier, NUS / U of Toulouse
Arash Shafiei, NUS / U of Toulouse
Vincent Charvillat, U of Toulouse
Geraldine Morin, U of Toulouse
Romulus Grigoras, U of Toulouse



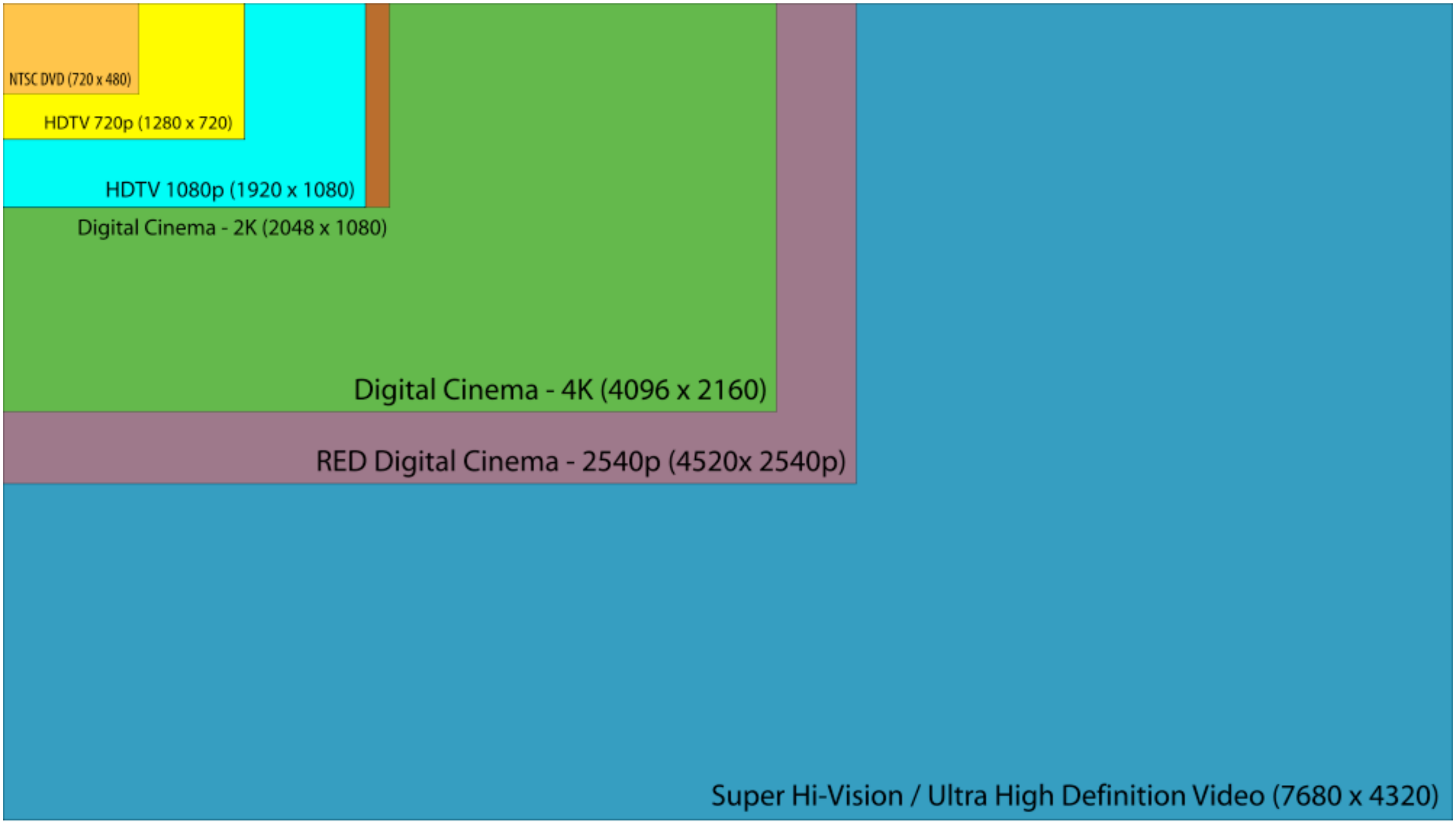






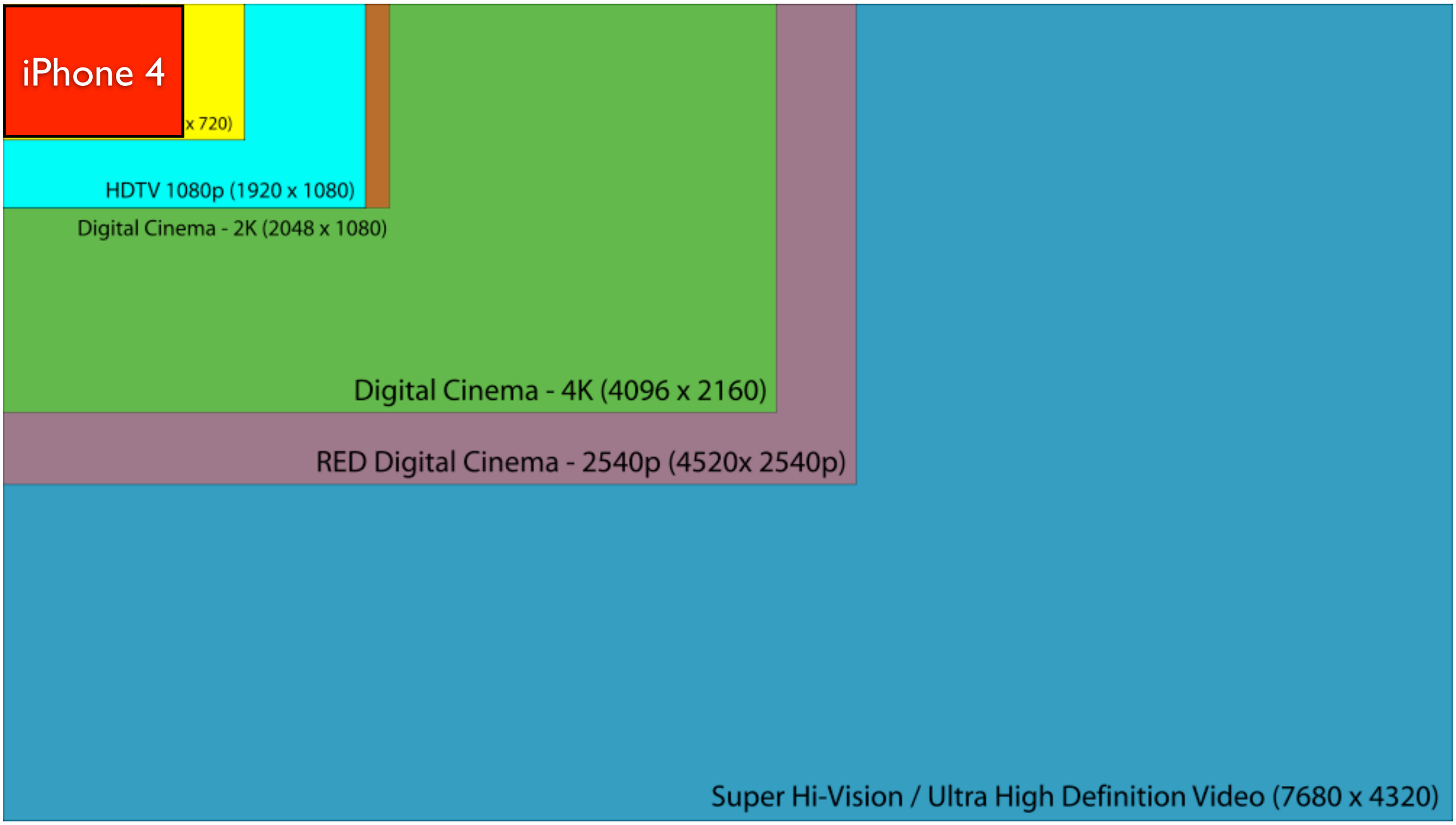


Why Zoomable Video?



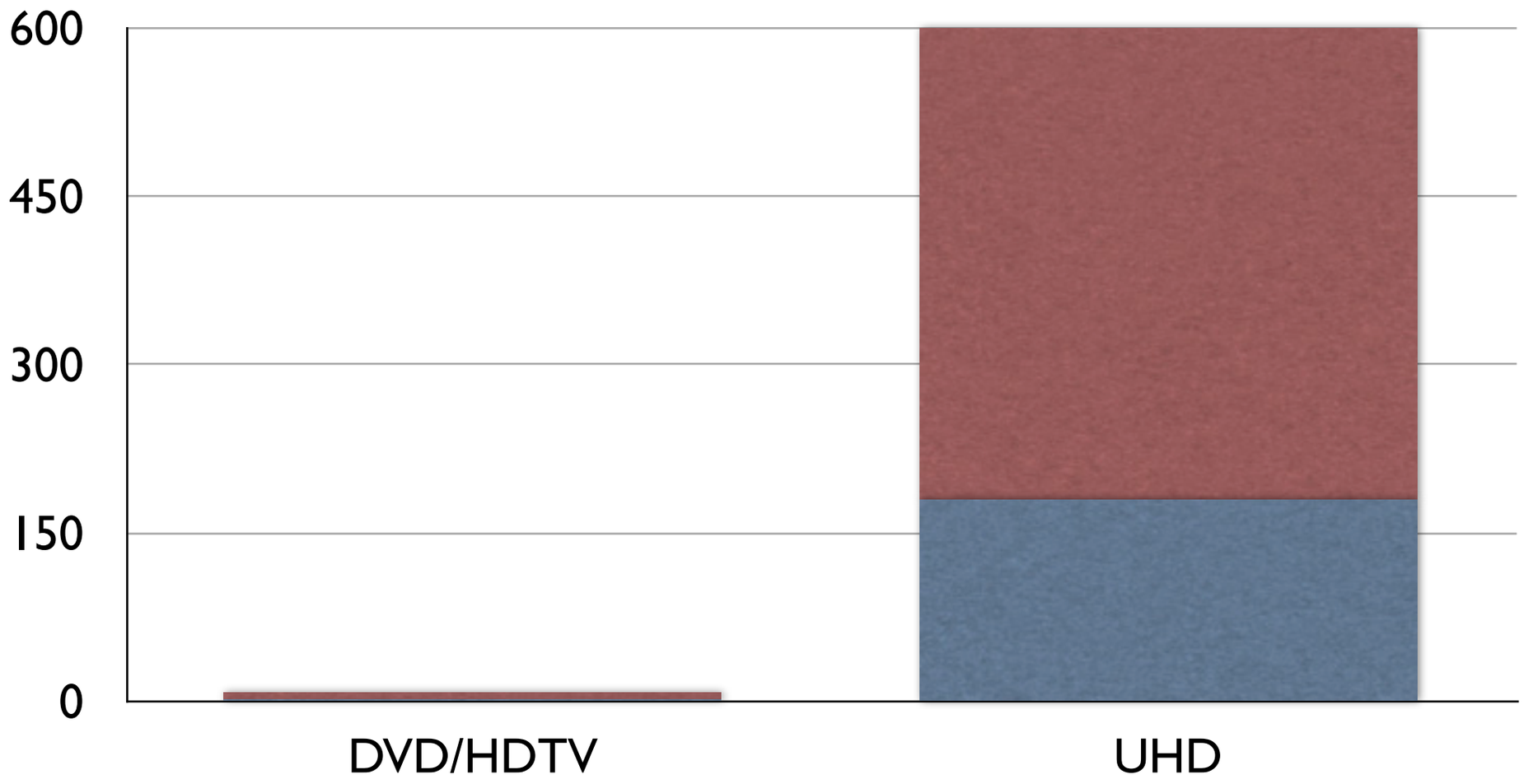
http://en.wikipedia.org/wiki/File:UHDV.svg

960 x 640

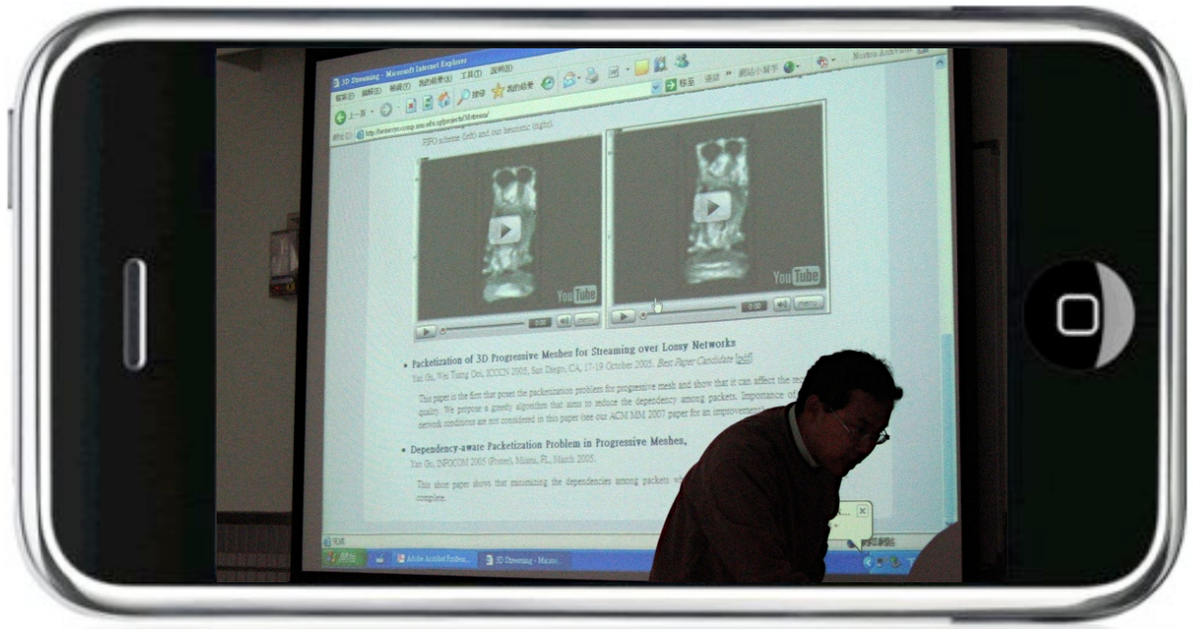


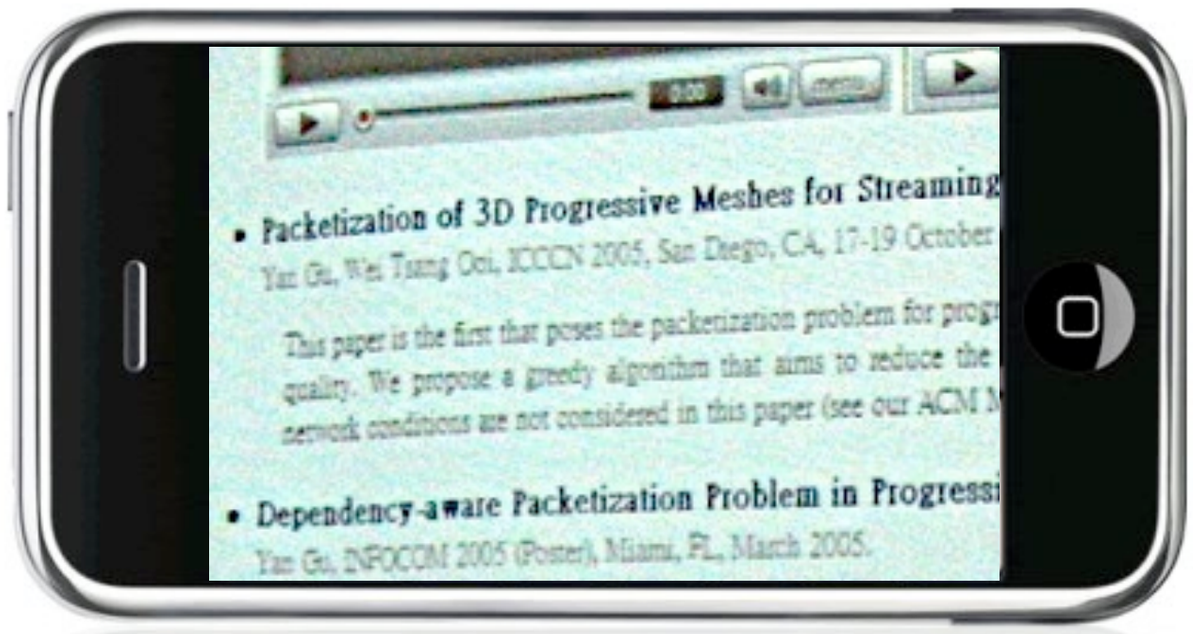
http://en.wikipedia.org/wiki/File:UHDV.svg

Bandwidth Required (Mbps)

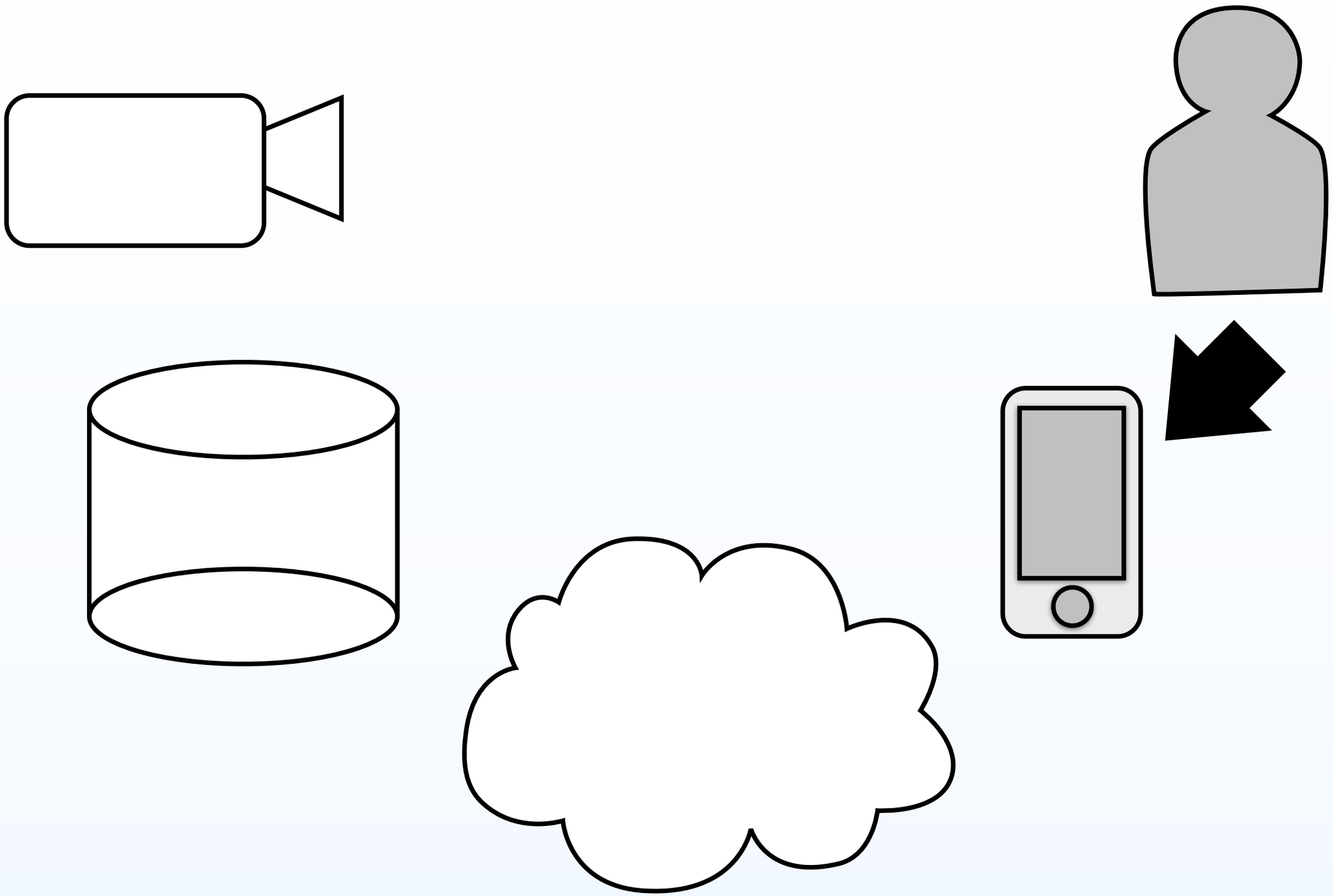


What is zoomable video?


• Packetization of 3D Progressive Meshes for Streaming over Lossy Networks
• Dependency-aware Packetization Problem in Progressive Meshes,


• Packetization of 3D Progressive Meshes for Streaming
Yan Gu, Wei Tsang Ooi, ICCCN 2005, San Diego, CA, 17-19 October
This paper is the first that poses the packetization problem for progr
quality. We propose a greedy algorithm that aims to reduce the
network conditions are not considered in this paper (see our ACM M
• Dependency-aware Packetization Problem in Progressi
Yan Gu, INFOCOM 2005 (Poster), Miami, FL, March 2005.

How would users use zoomable video?



HTML5-based web player

0:00 / 3:00 00:00

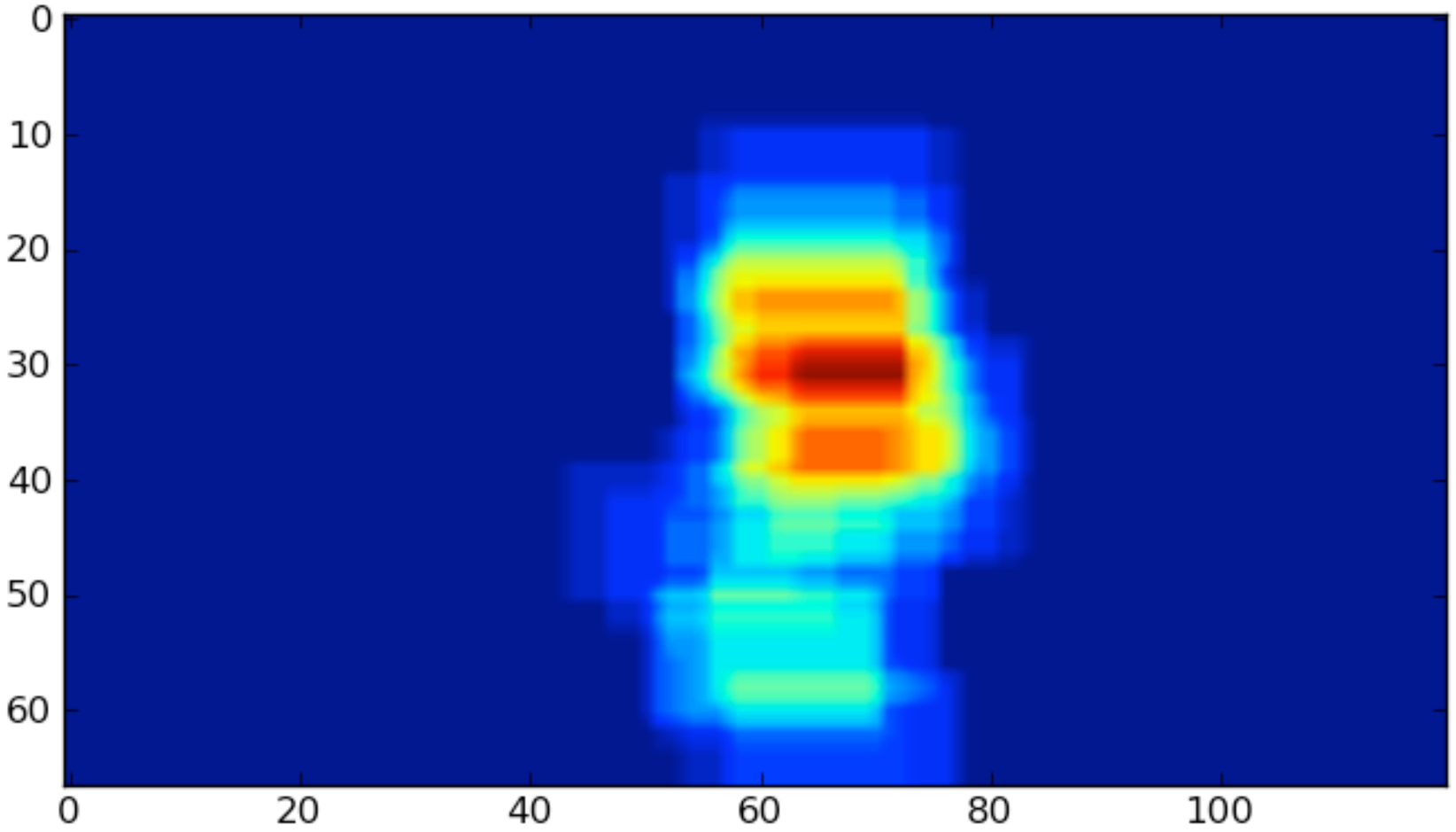
4 video clips

37 viewing sessions

log all interactions

(after a tutorial)

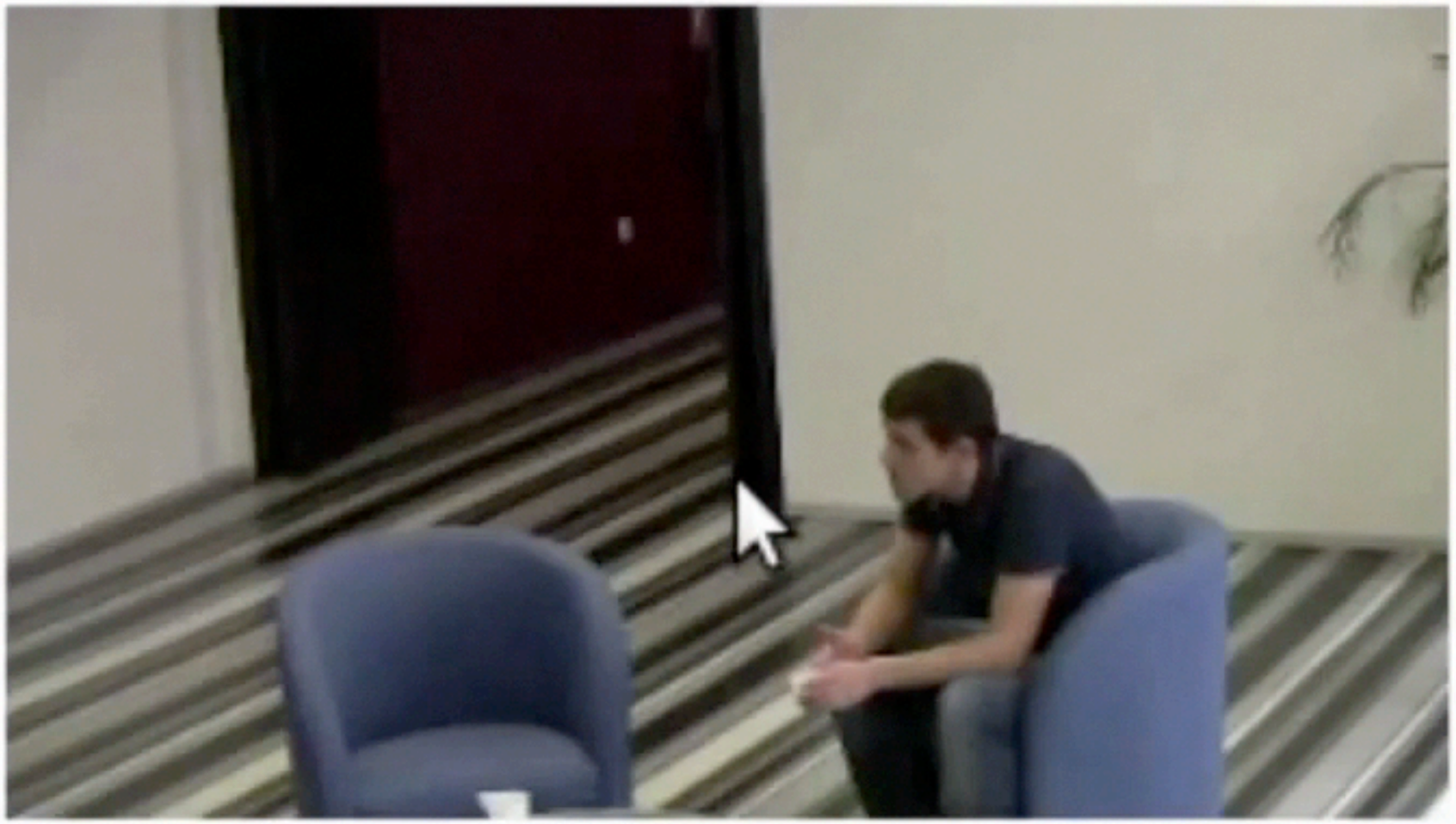
lots of interaction
spend time zooming in
next action hard to predict
similar ROIs


0
10
20
30
40
50
60
0
20
40
60
80
100

How to reduce the number of interactions?

zoom and pan are tedious
(clicking and dragging)

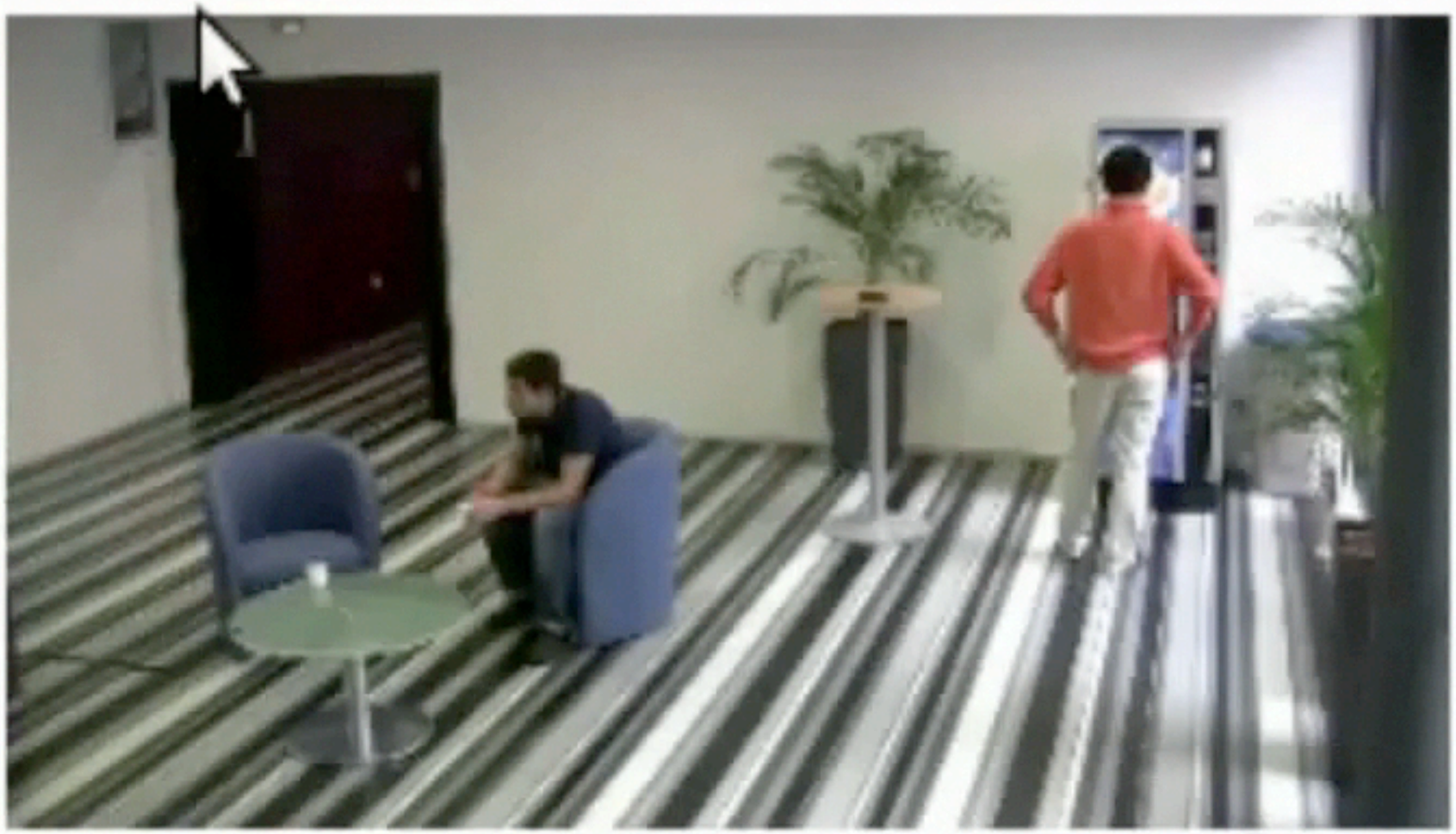
especially if the interested
object is moving



0:33

can we help the user?

idea: one-click
zoom, pan, and track.



0:33

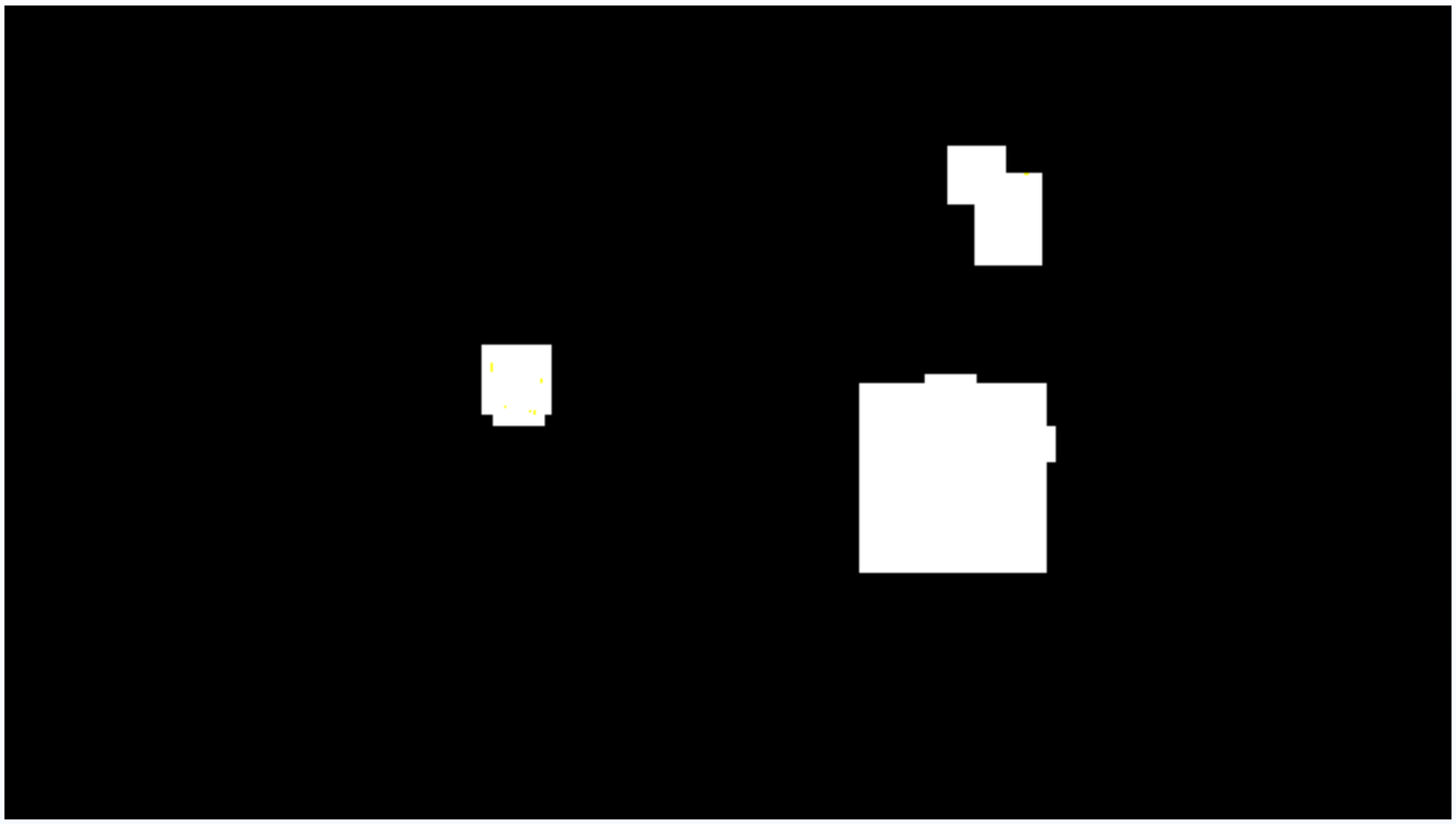
recommended viewport:
highlight what could be interesting to users

what is interesting?

1. content analysis

Saliency

Motion

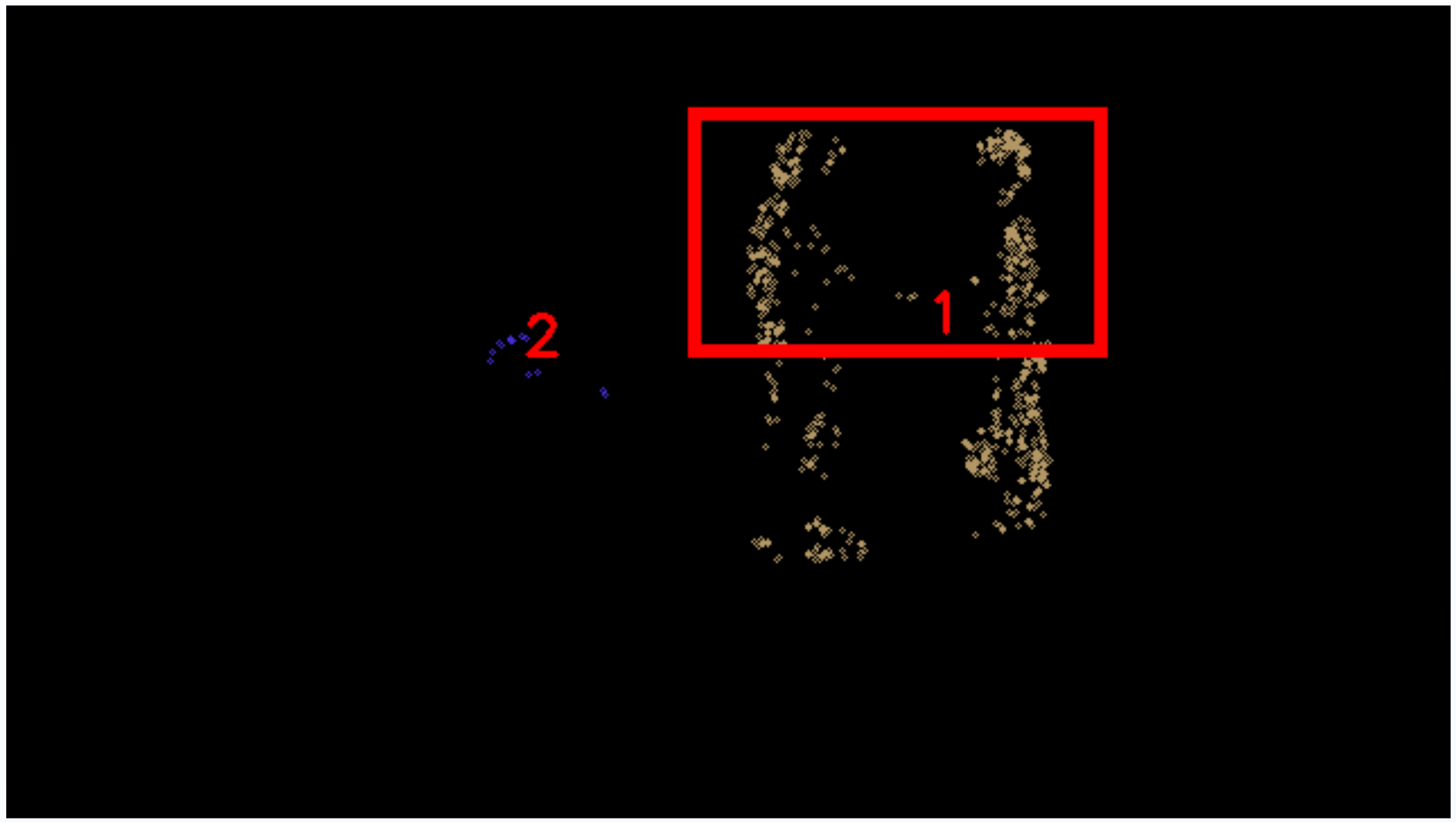


Face

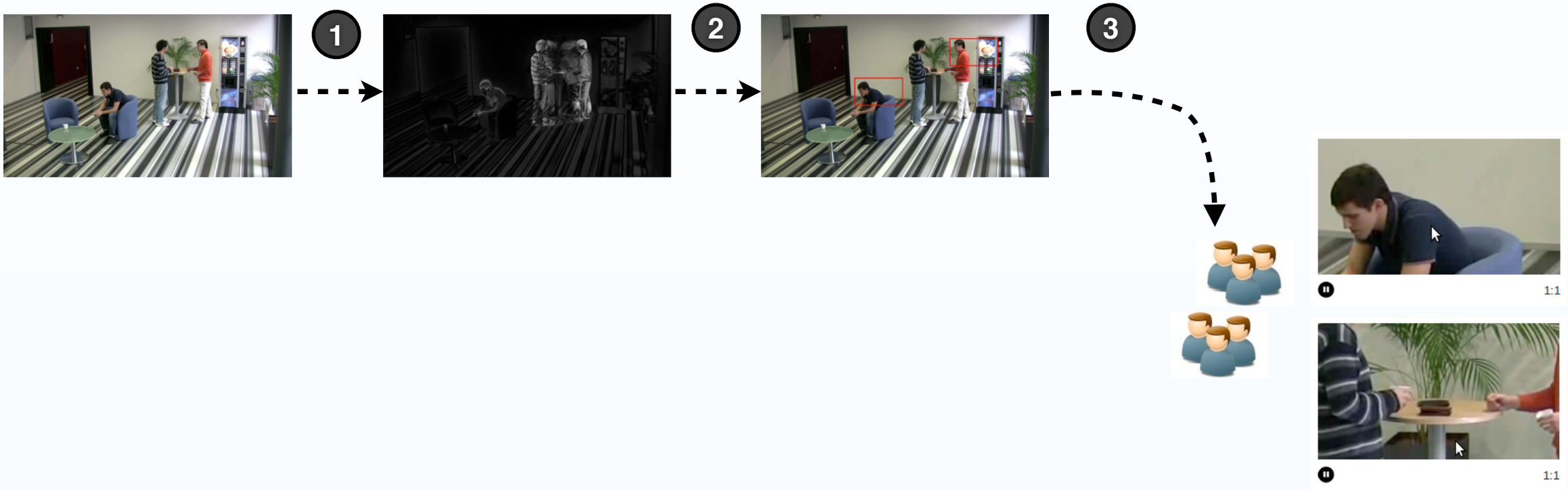
Linear Combination

cluster important pixels using mean-shift

find viewports with min cut



Cut (with preference to keep the top)


1
2
3
1:1
1:1

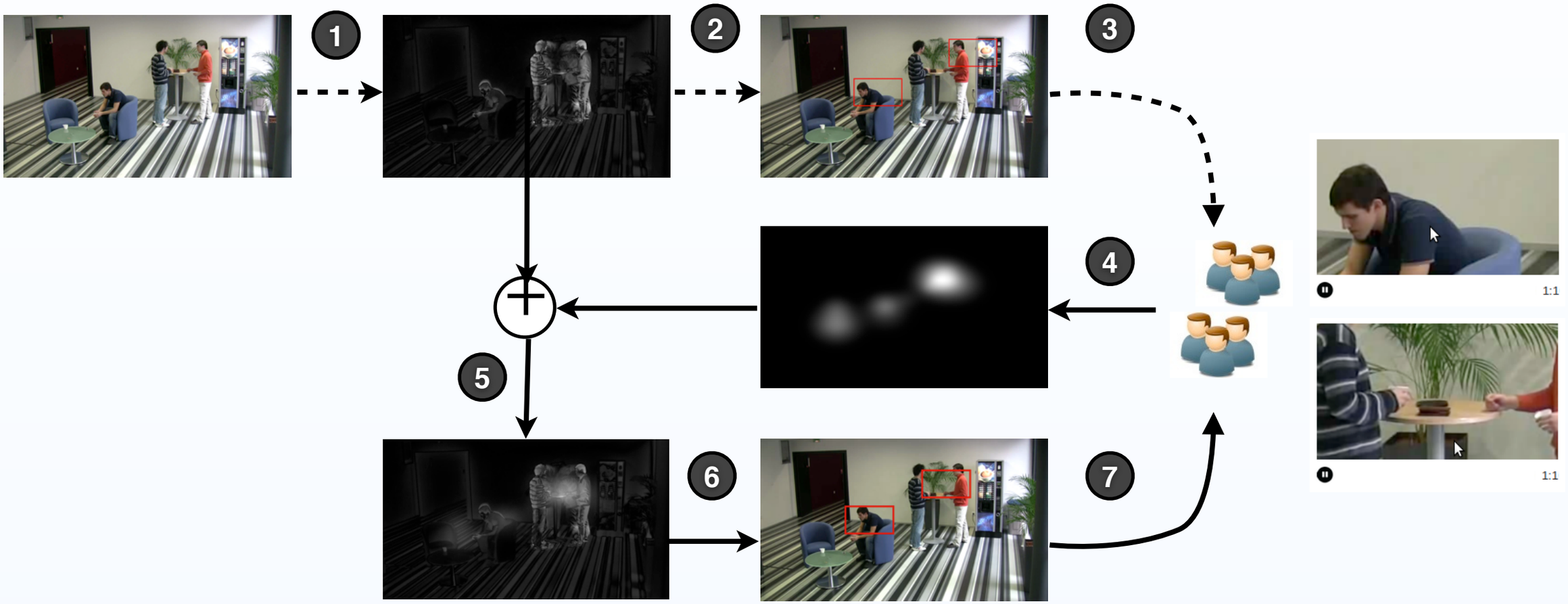
show video, ask question

not all recommendations are clicked

many click on non-recommended regions

what is interesting?

1. content analysis
2. crowdsourcing


1
2
3
4
5
6
7
1:1
1:1

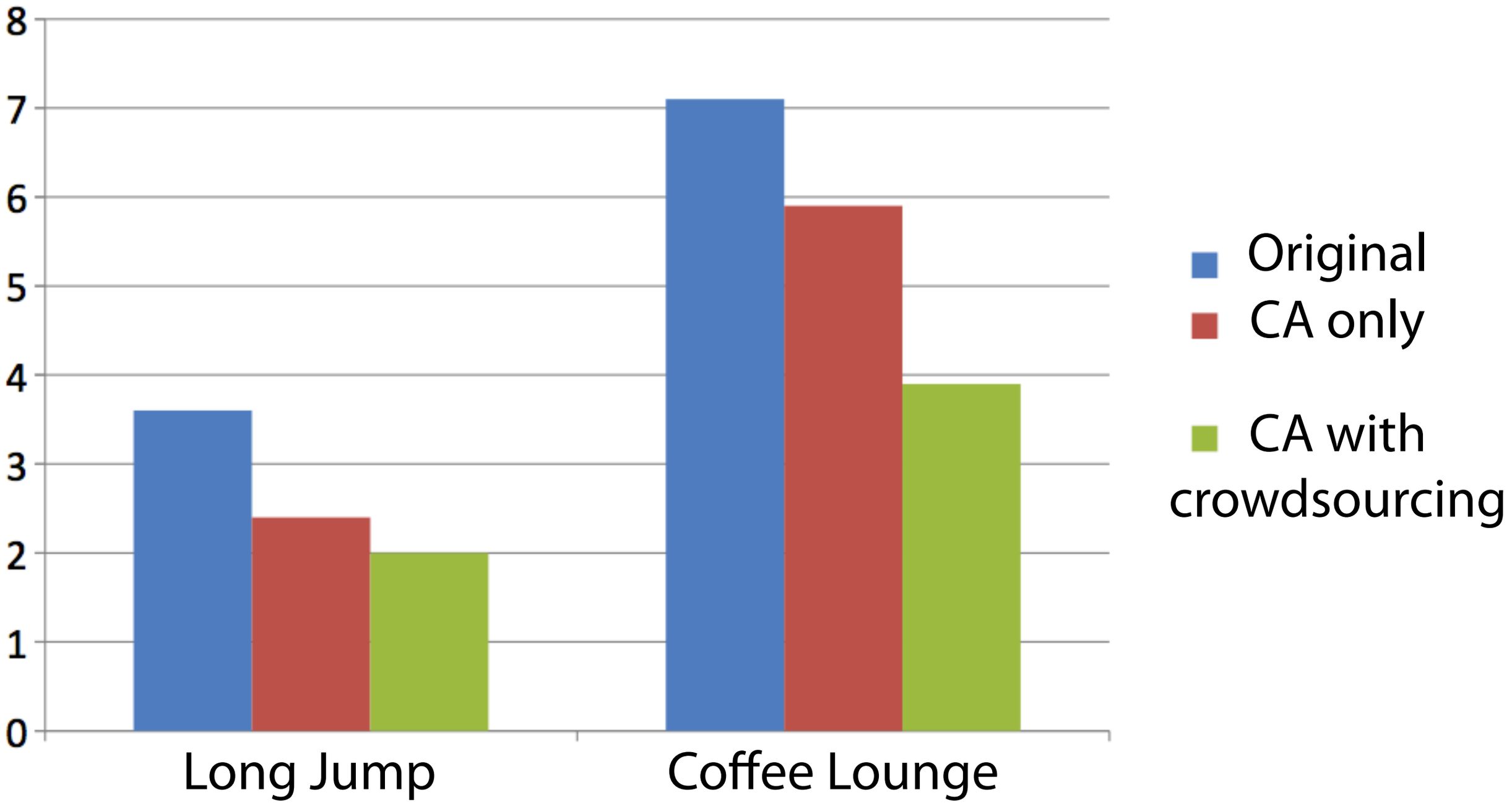
Results

70 users

better understanding of video

fewer interactions

Average number of pans per user



what is interesting?

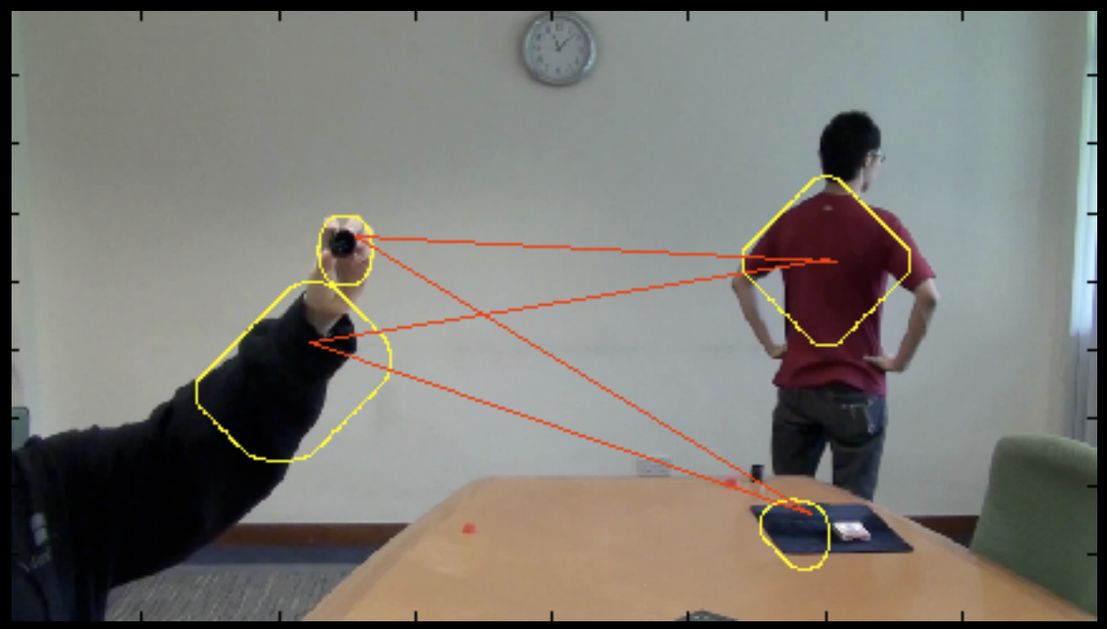
1. content analysis
2. crowdsourcing

application to:

video retargeting

Sample Video

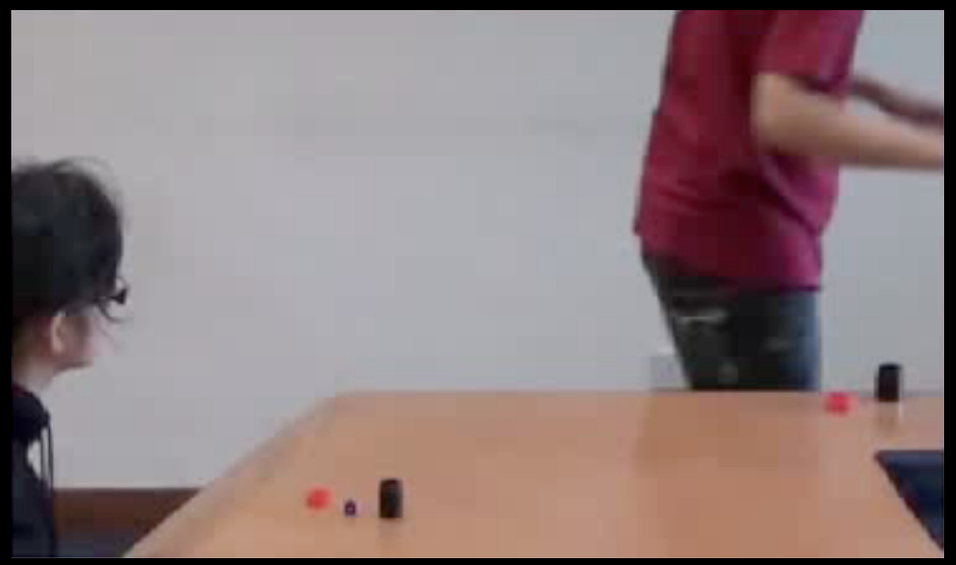
Using Saliency Map



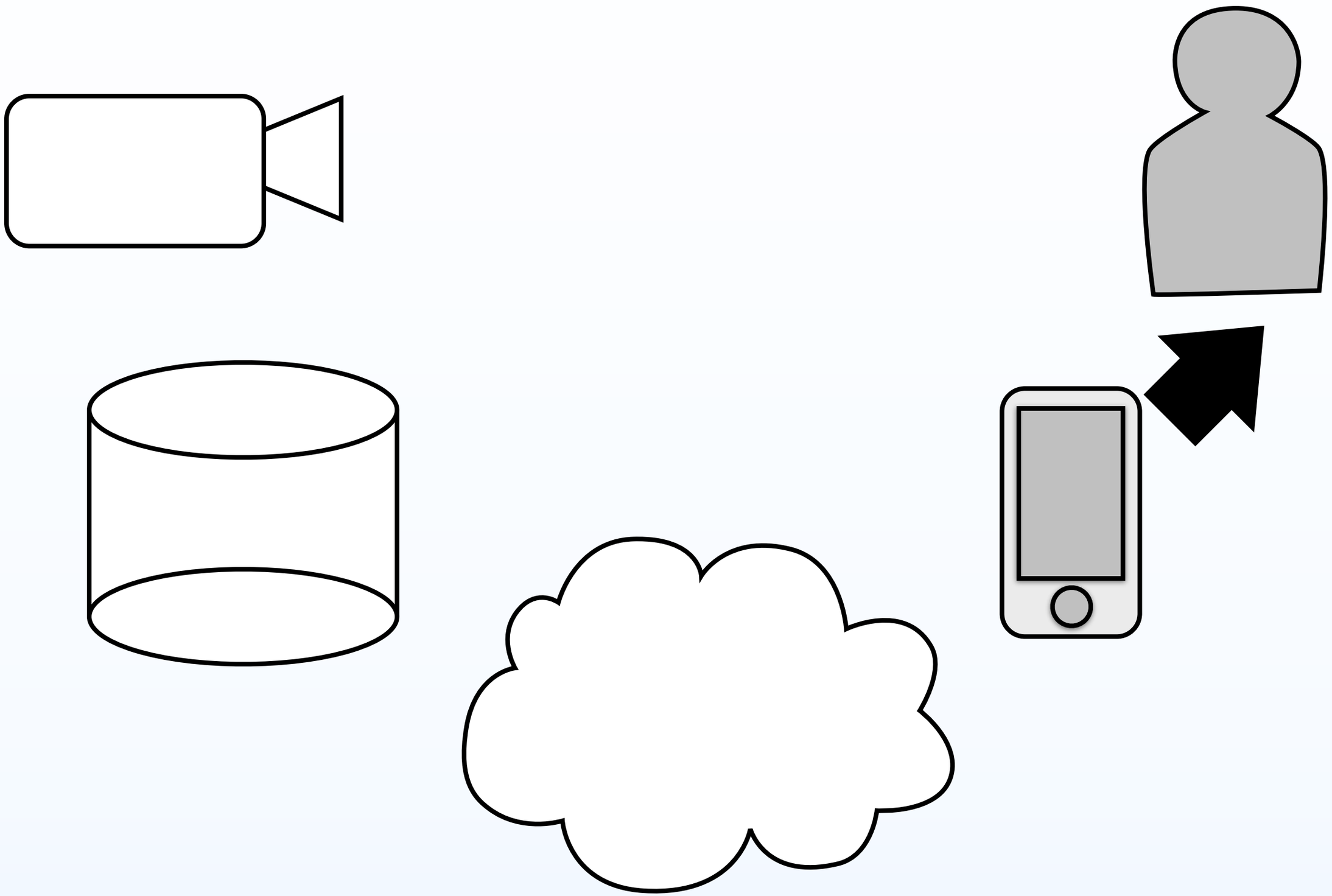
Using Saliency Map

Voting

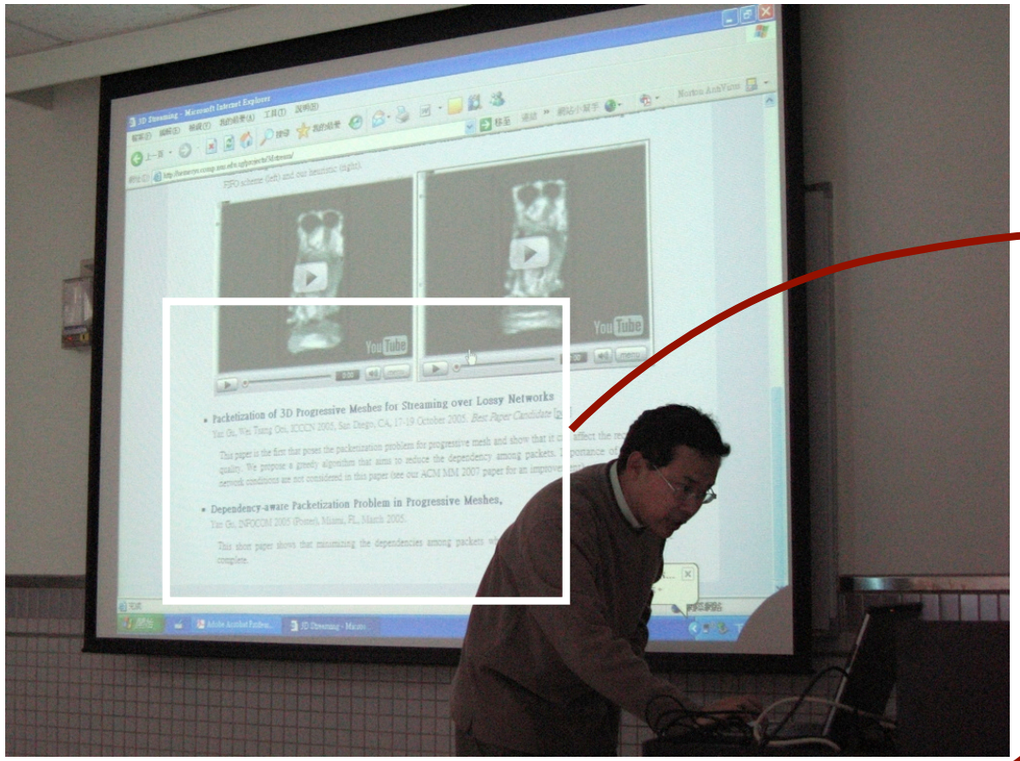
Voting + Cinematography Rules

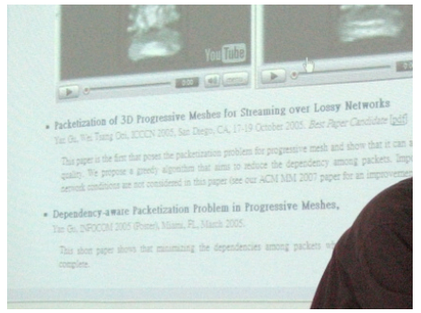


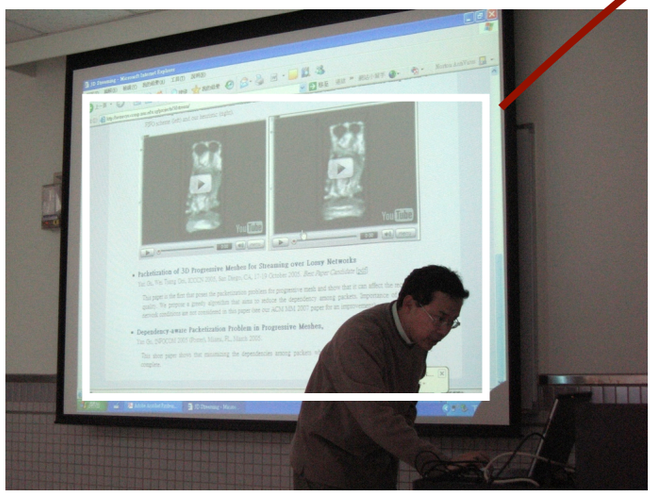
Voting + Cinematography Rules

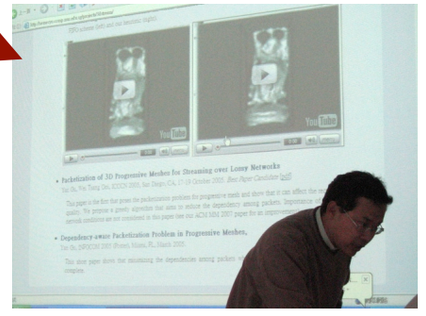


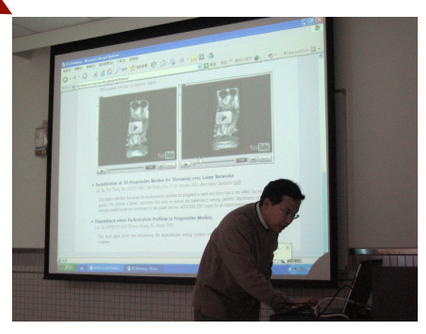
How to implement zoomable video?







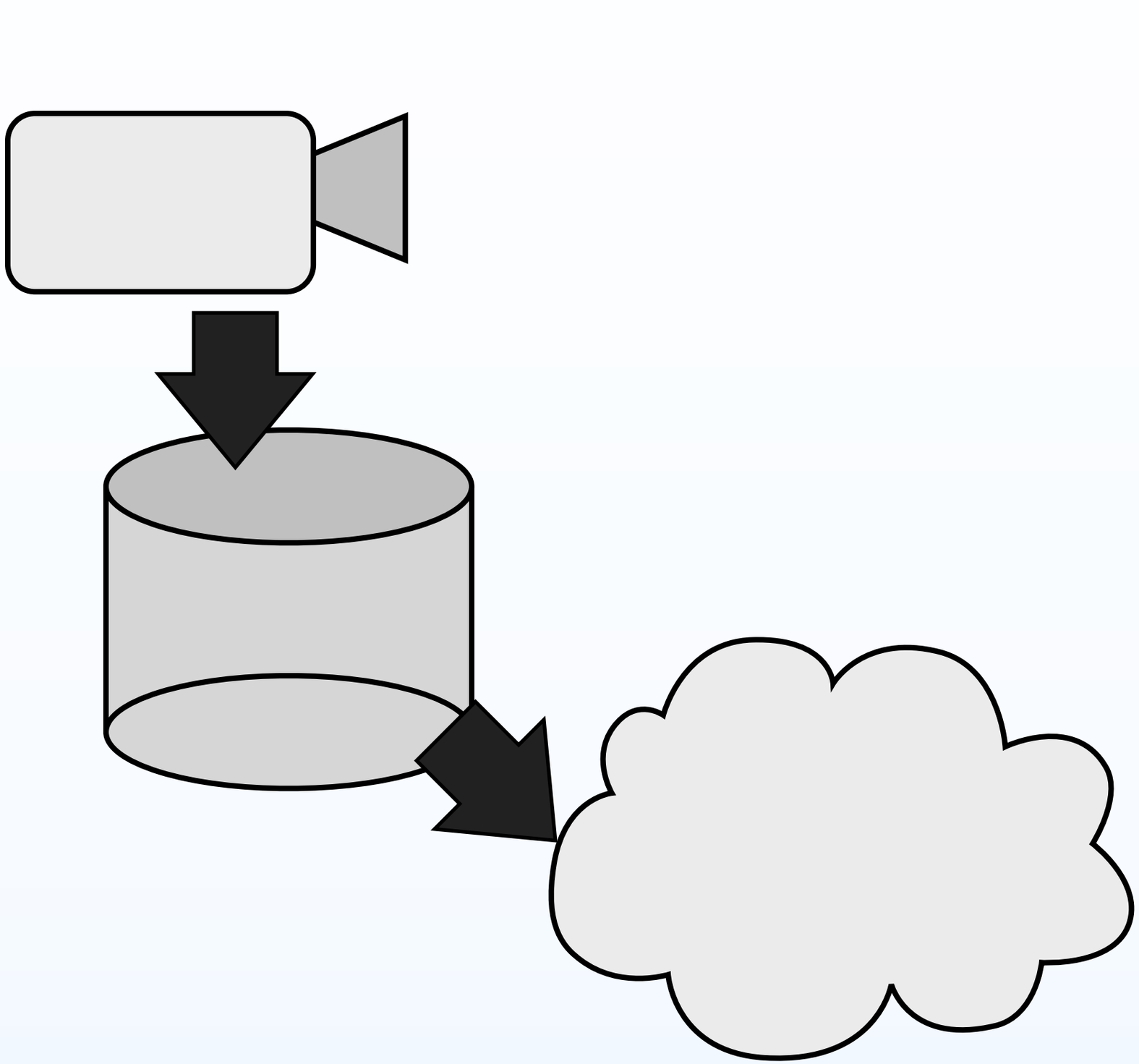




zoom

Dynamic ROI Cropping

(which is not supported by standard video codec)



Local playback: need not decode the whole frame

Remote playback: need not send the whole frame

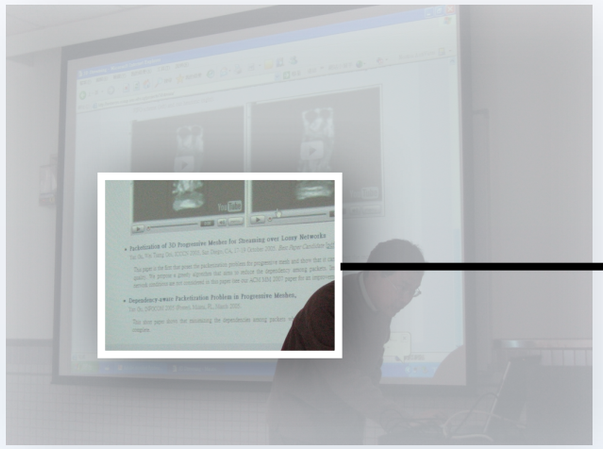
Method 1:

Dynamic Encoding

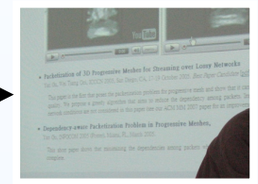
(x, y, w, h)

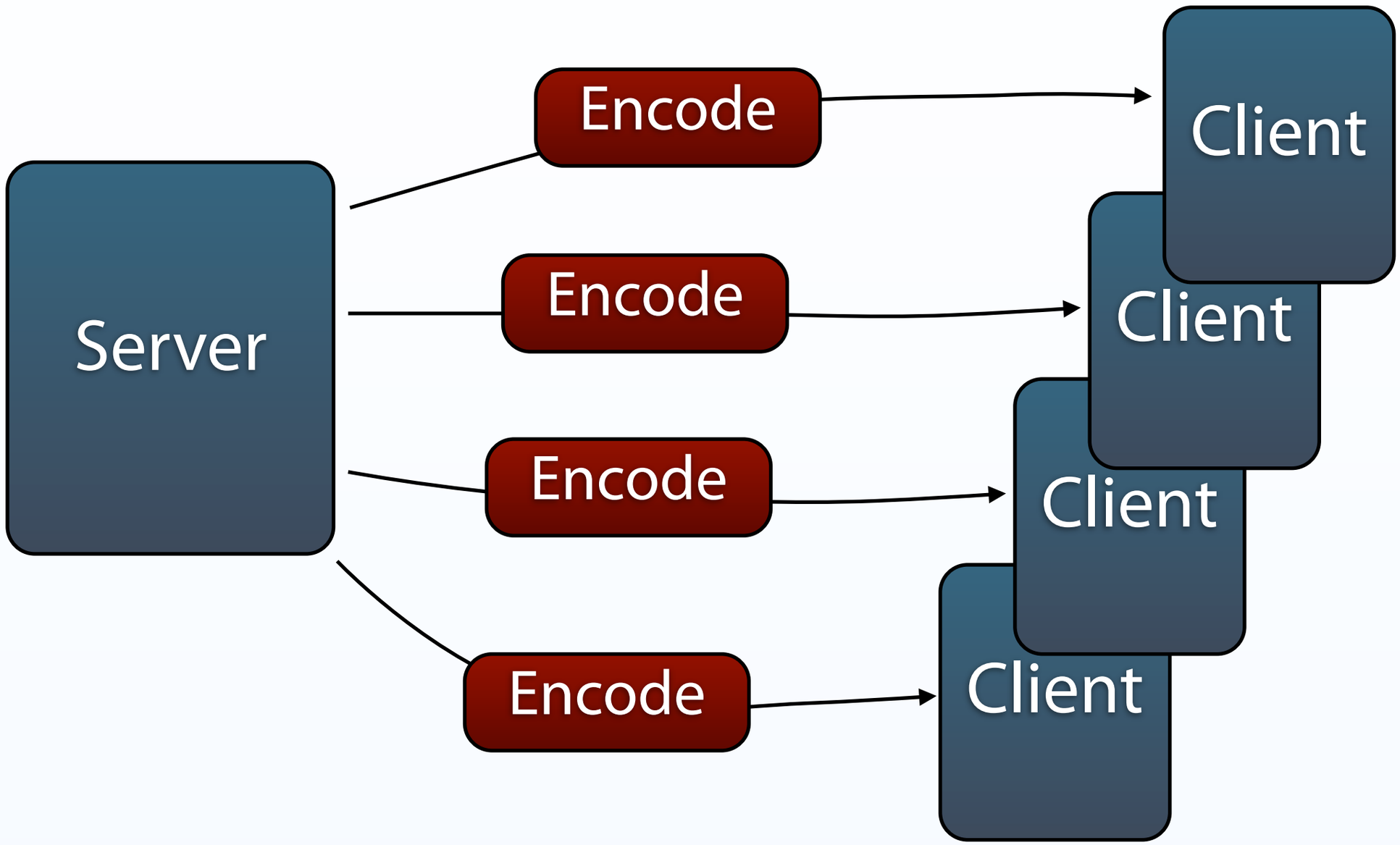
Server

Client



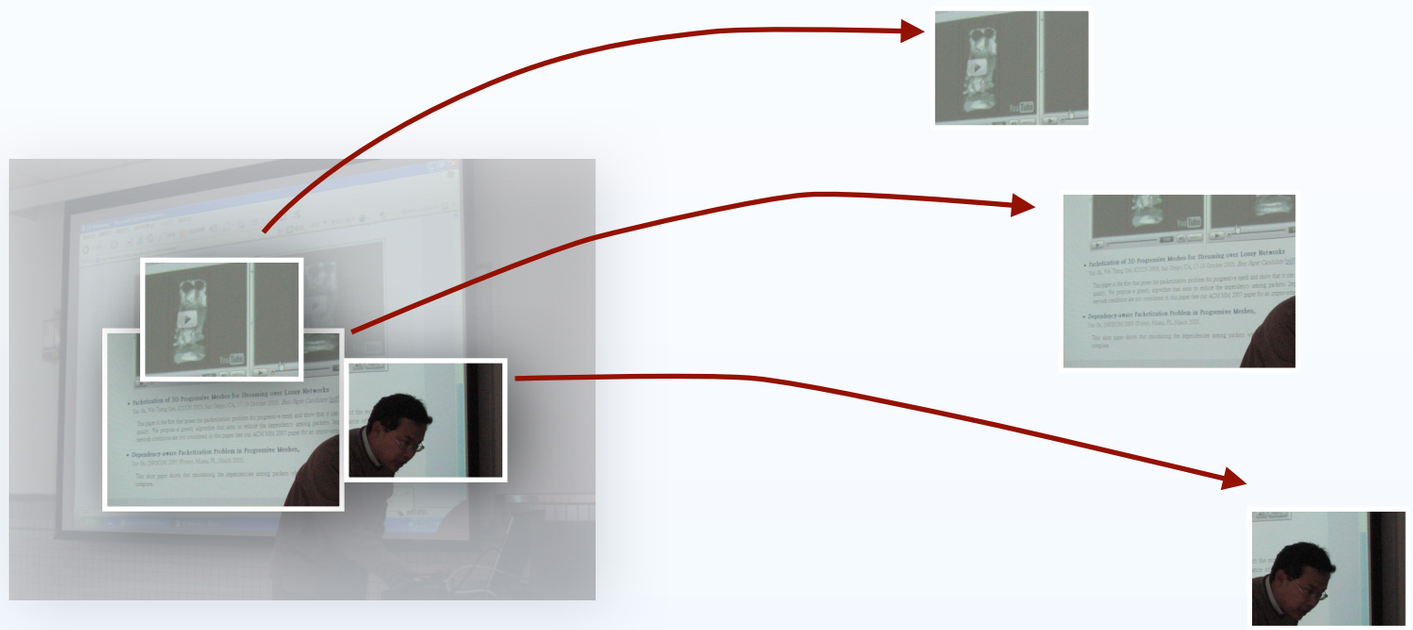
Encode





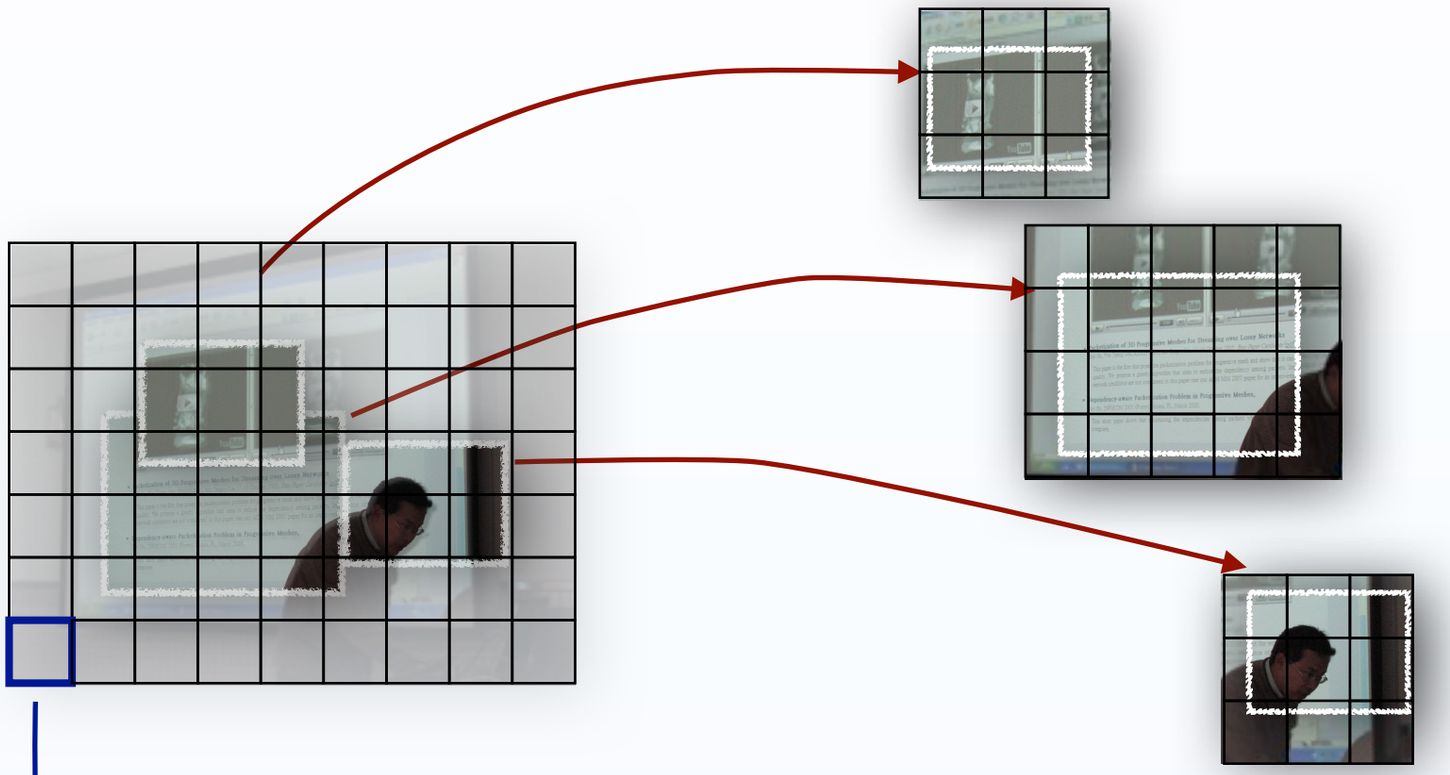
Not scalable

encode once
multiple ROIs



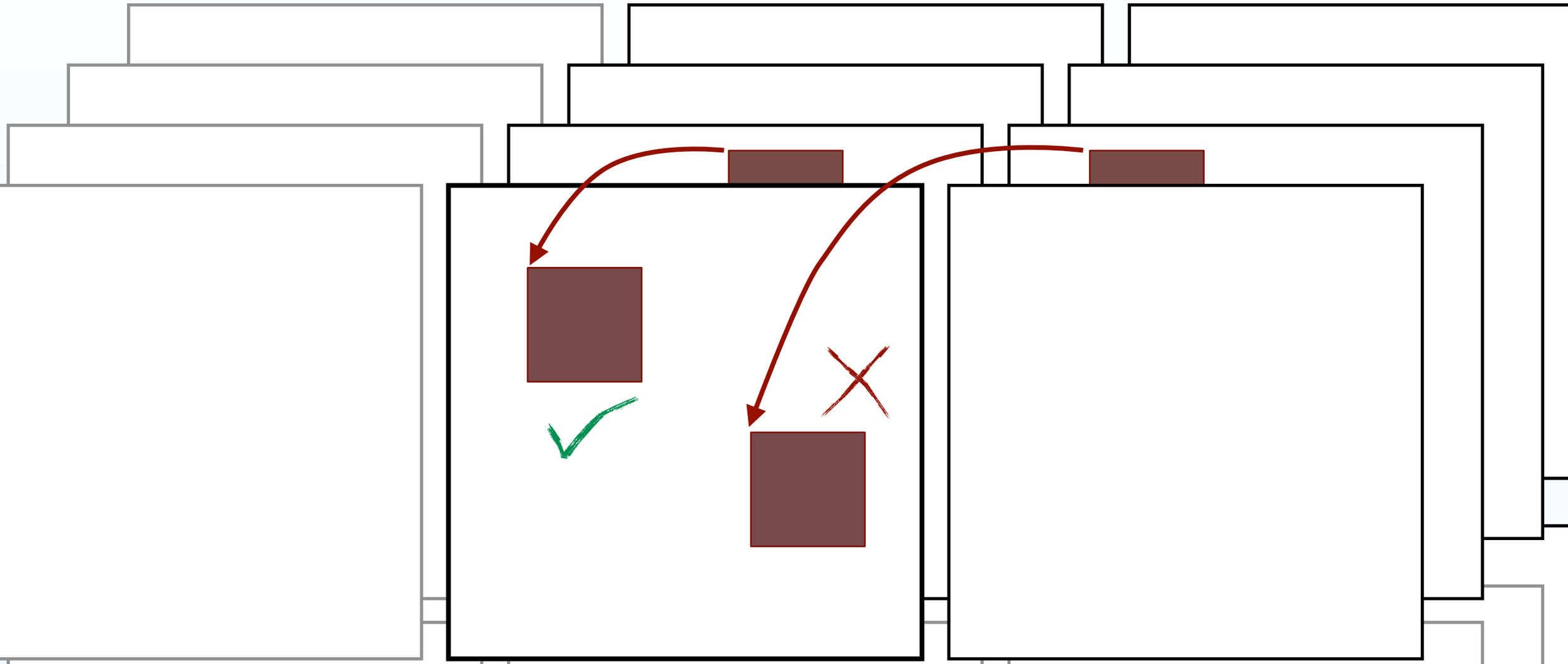
Method II:

Tiled Streams

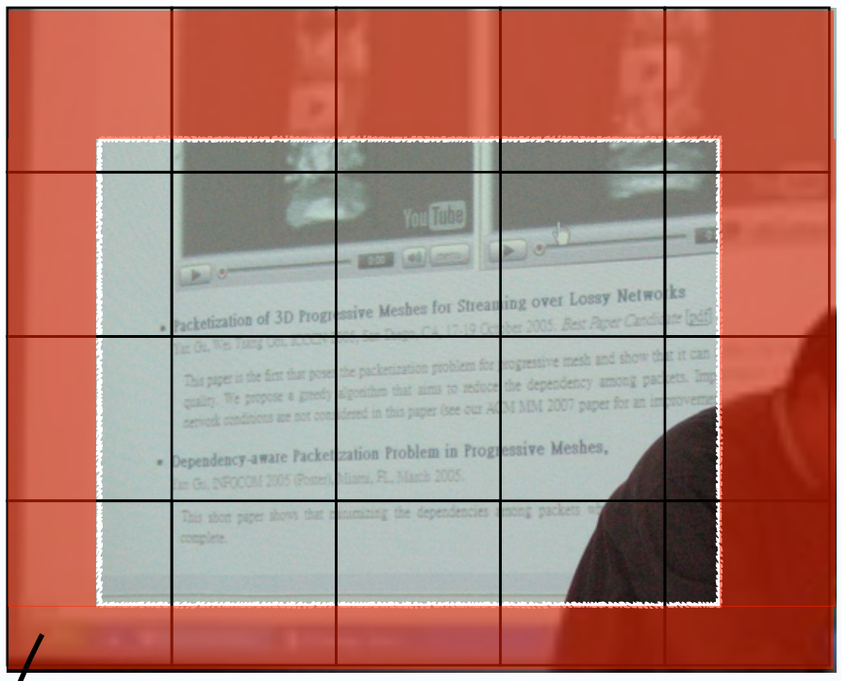


1 tile = *k* x *k* macroblocks

Motion vectors are constrained within a tile.
Each tile stream is independently decodable.



Big tile or small tile?



sent but not displayed (wasted bits)

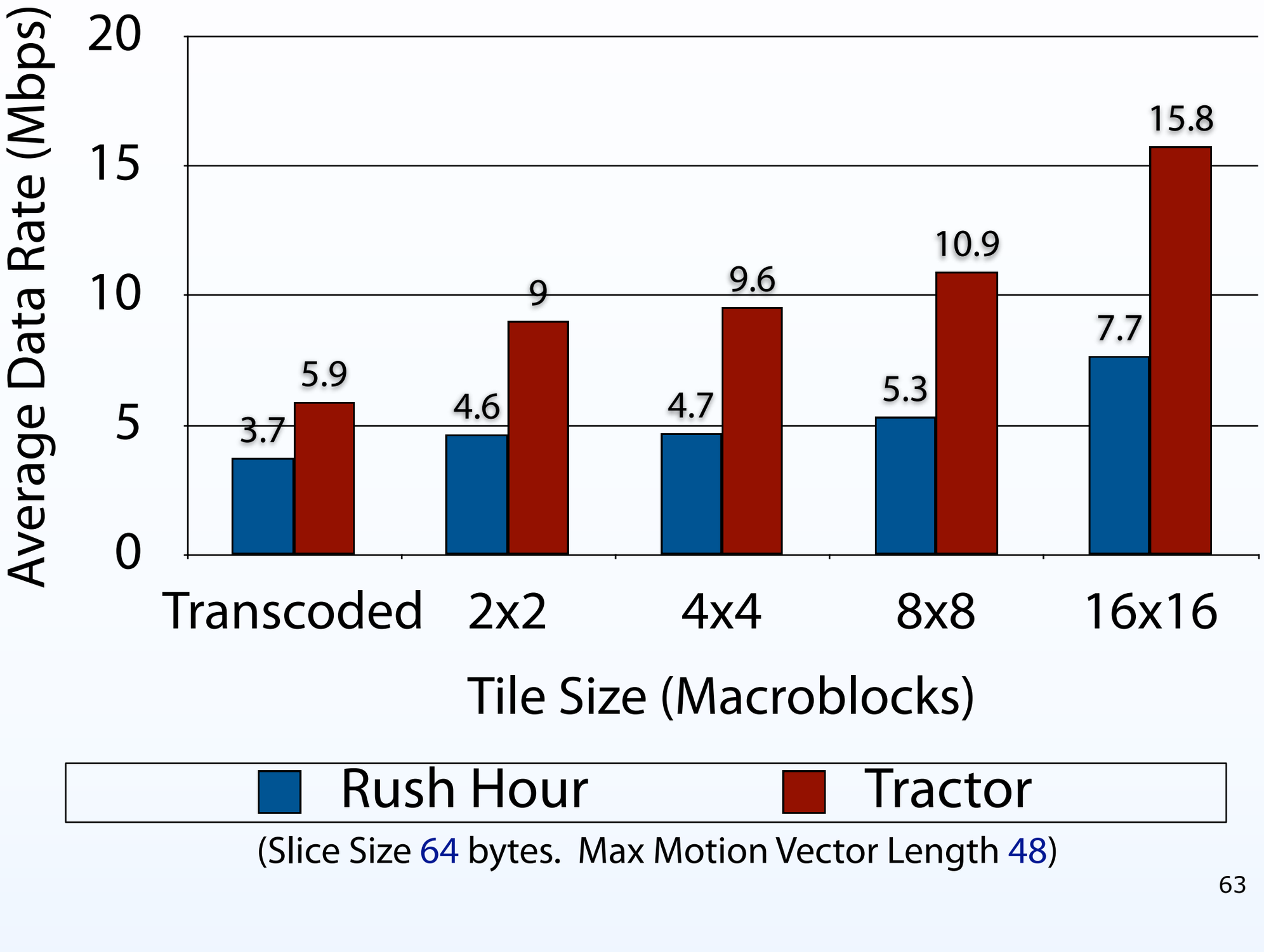
bigger tile → **more waste**

→ **more bits**

smaller tile → less compression

→ more bits

Average Data Rate When Transmitted ROI of 30x30 Macroblocks



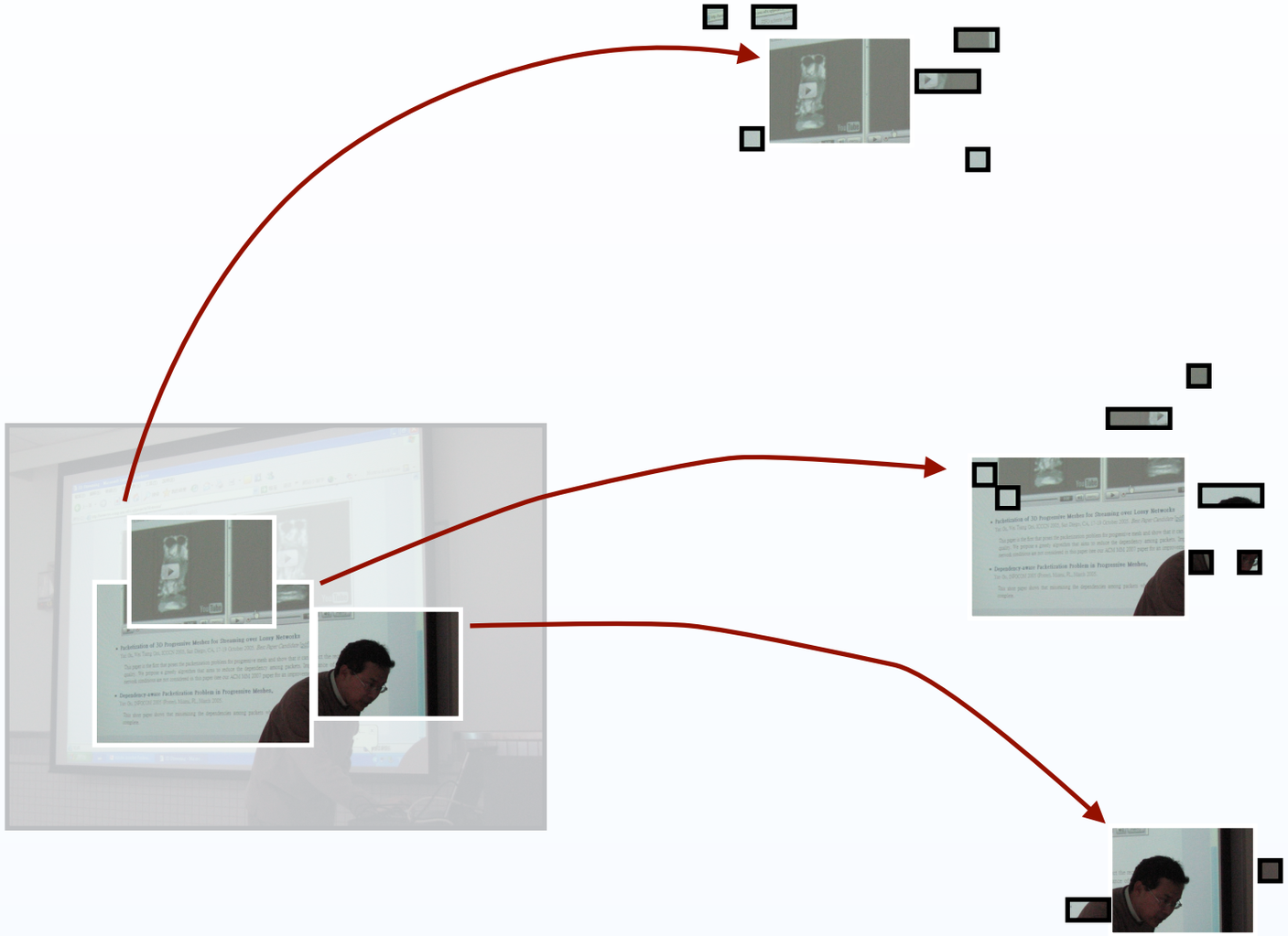
(Slice Size 64 bytes. Max Motion Vector Length 48)

Gain in compression is less significant than lost in wasted bits

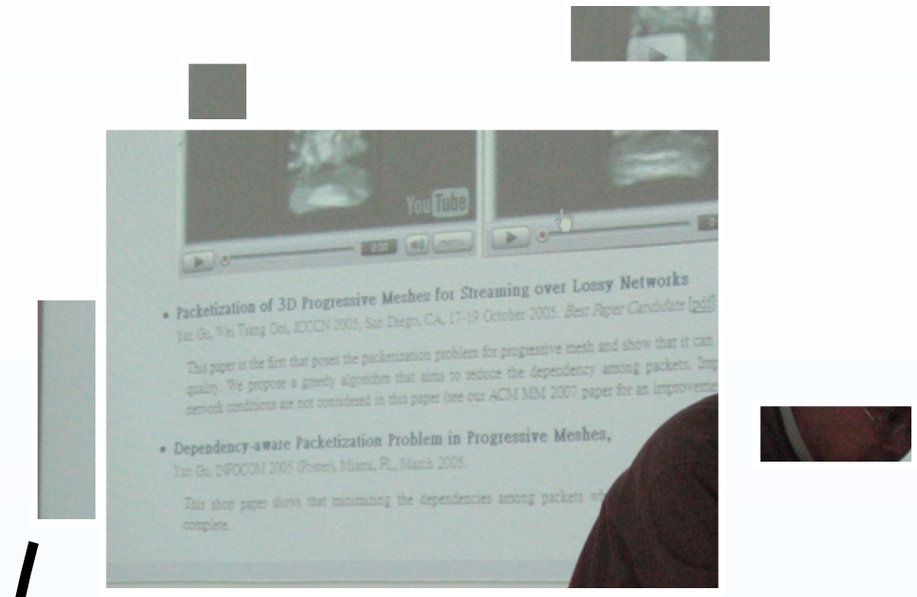
Can we reduce wasted bits?

Method III:

Monolithic Stream

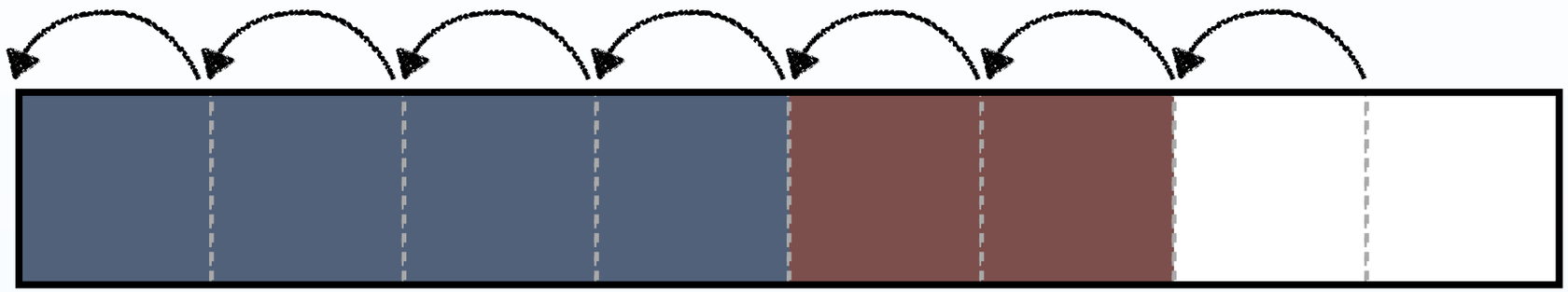


send the ROI, plus any extra bits needed to decode the pixels within ROI



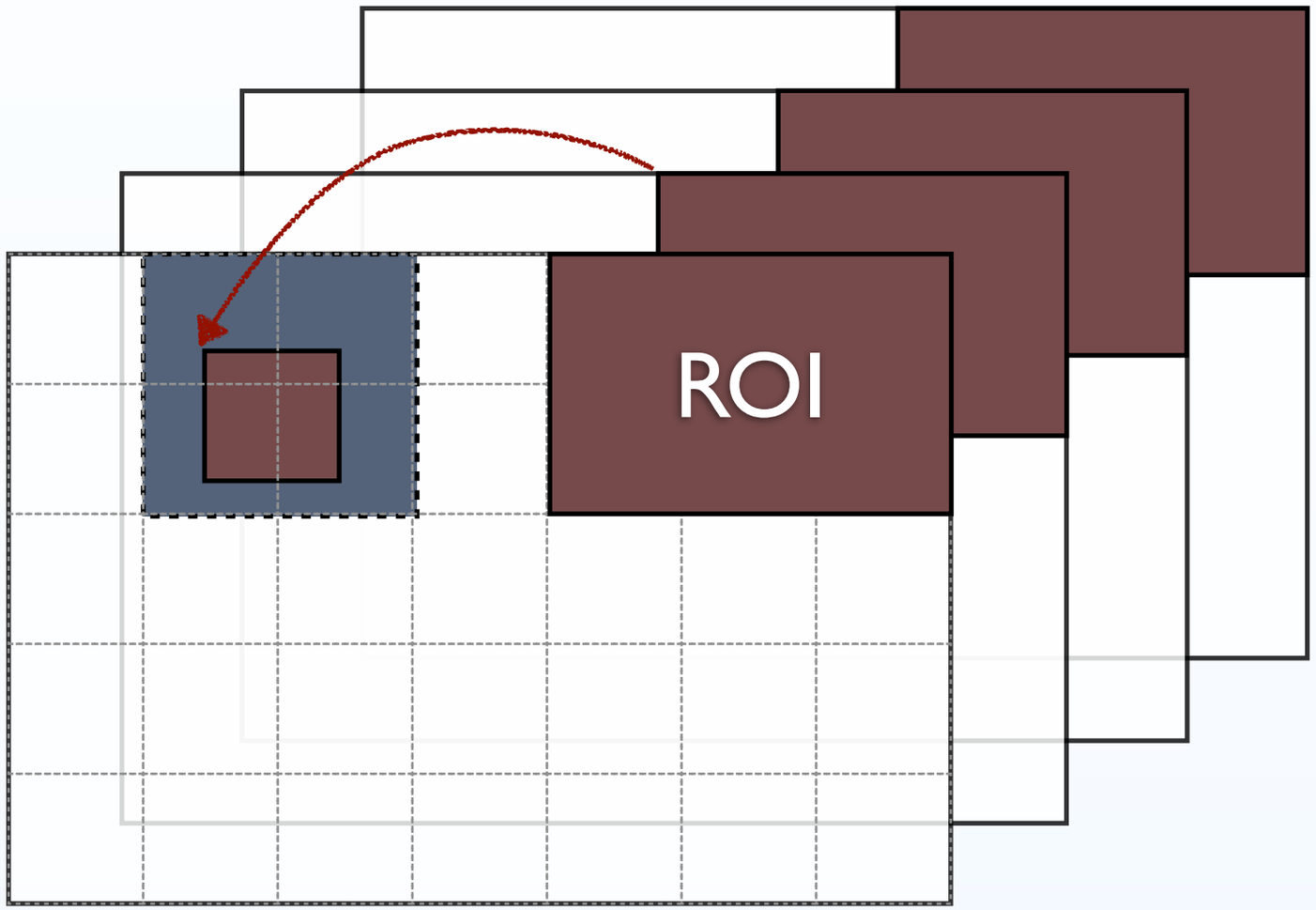
some macroblocks
within ROI depends on these

VLC dependency in a slice

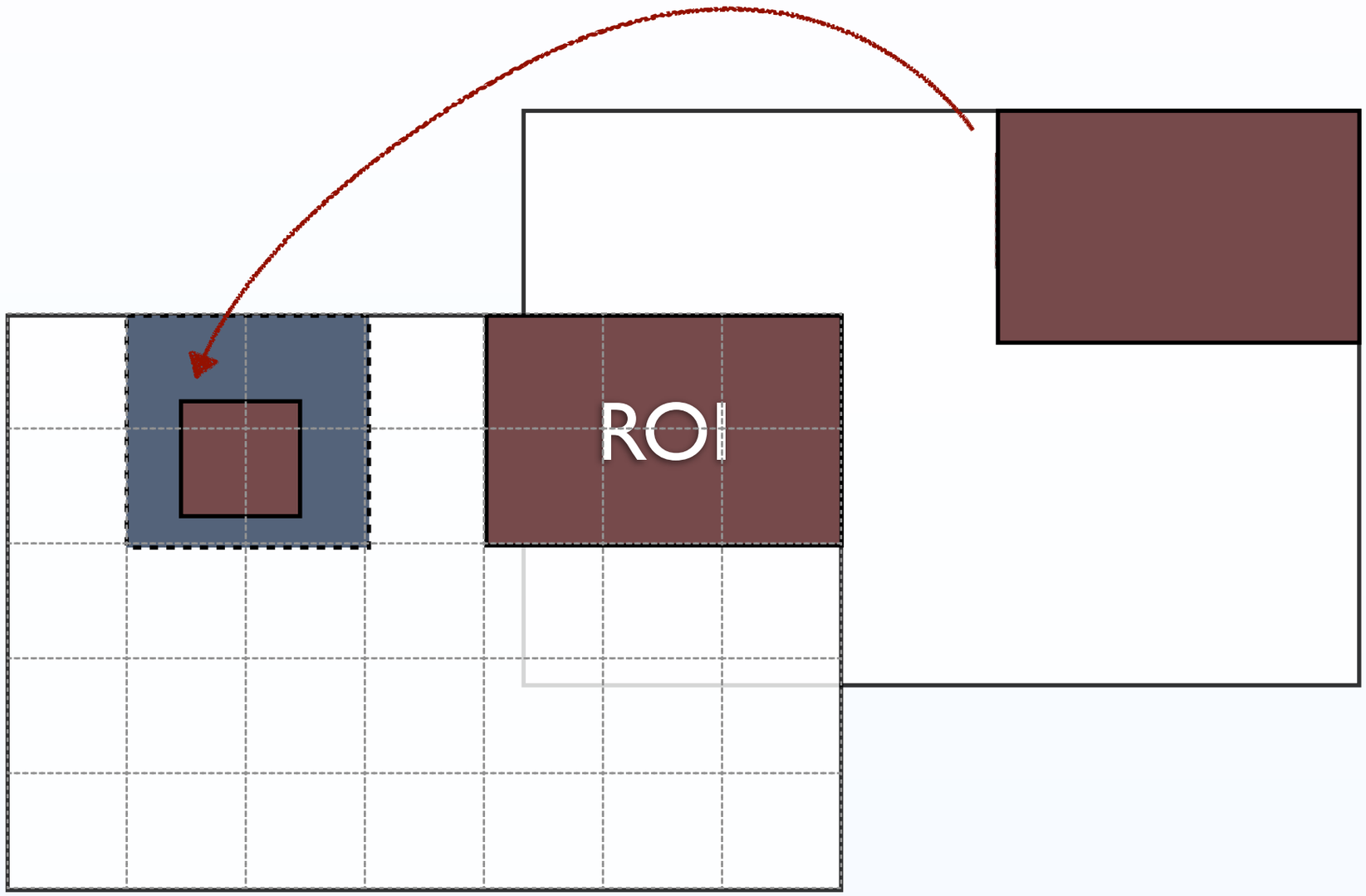


Within a slice, preceeding macroblocks ■ need to be parsed to access macroblocks in the middle ■ (no random access to macroblocks)

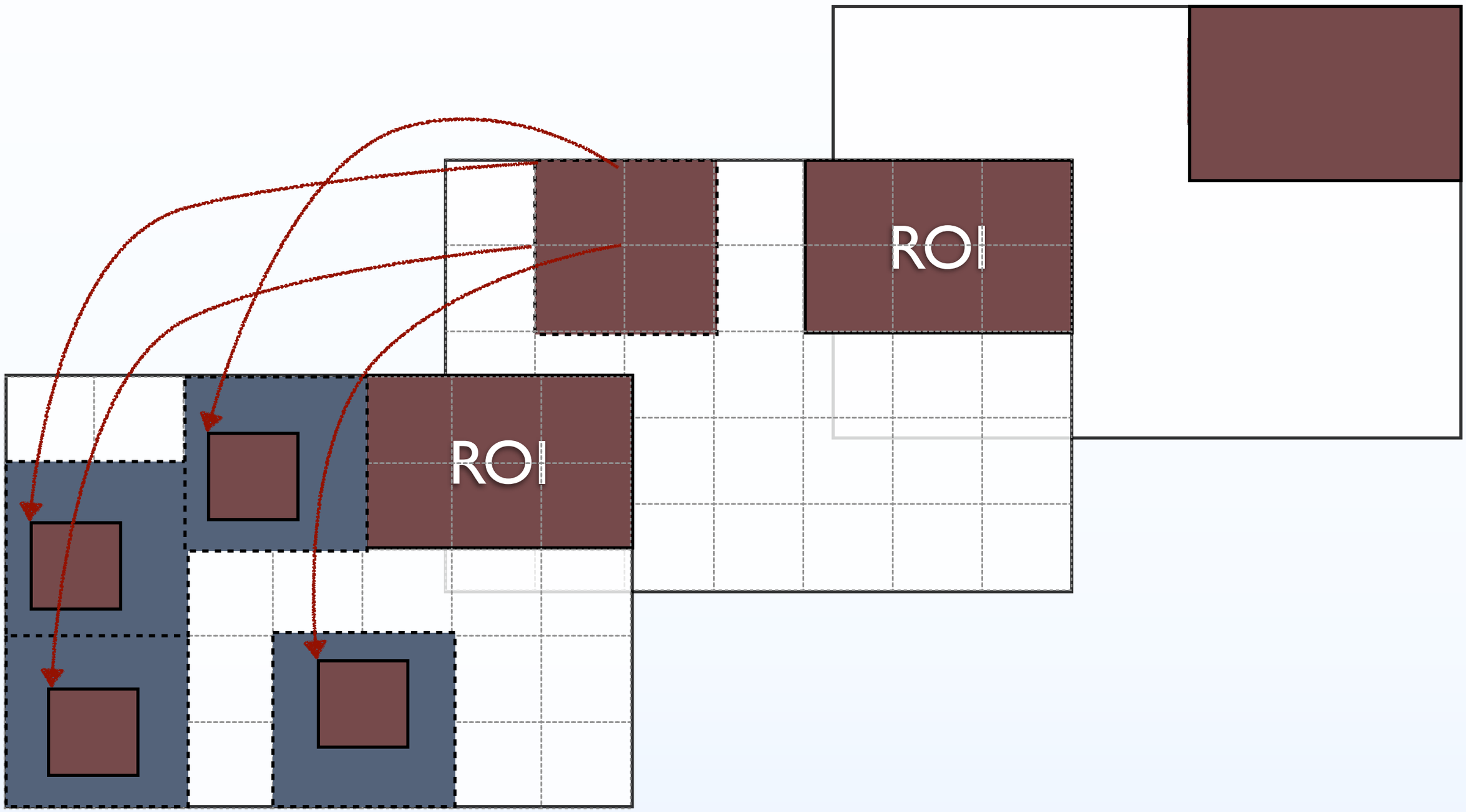
motion dependency across frames

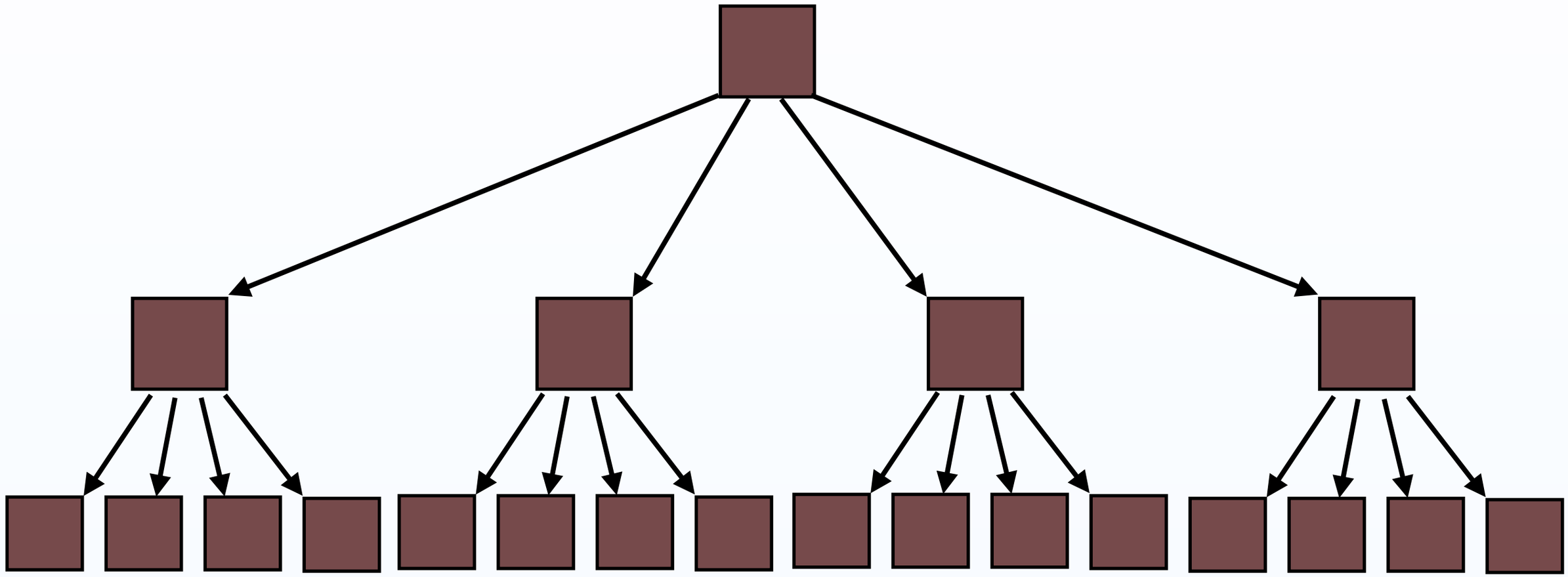


motion dependency propagates

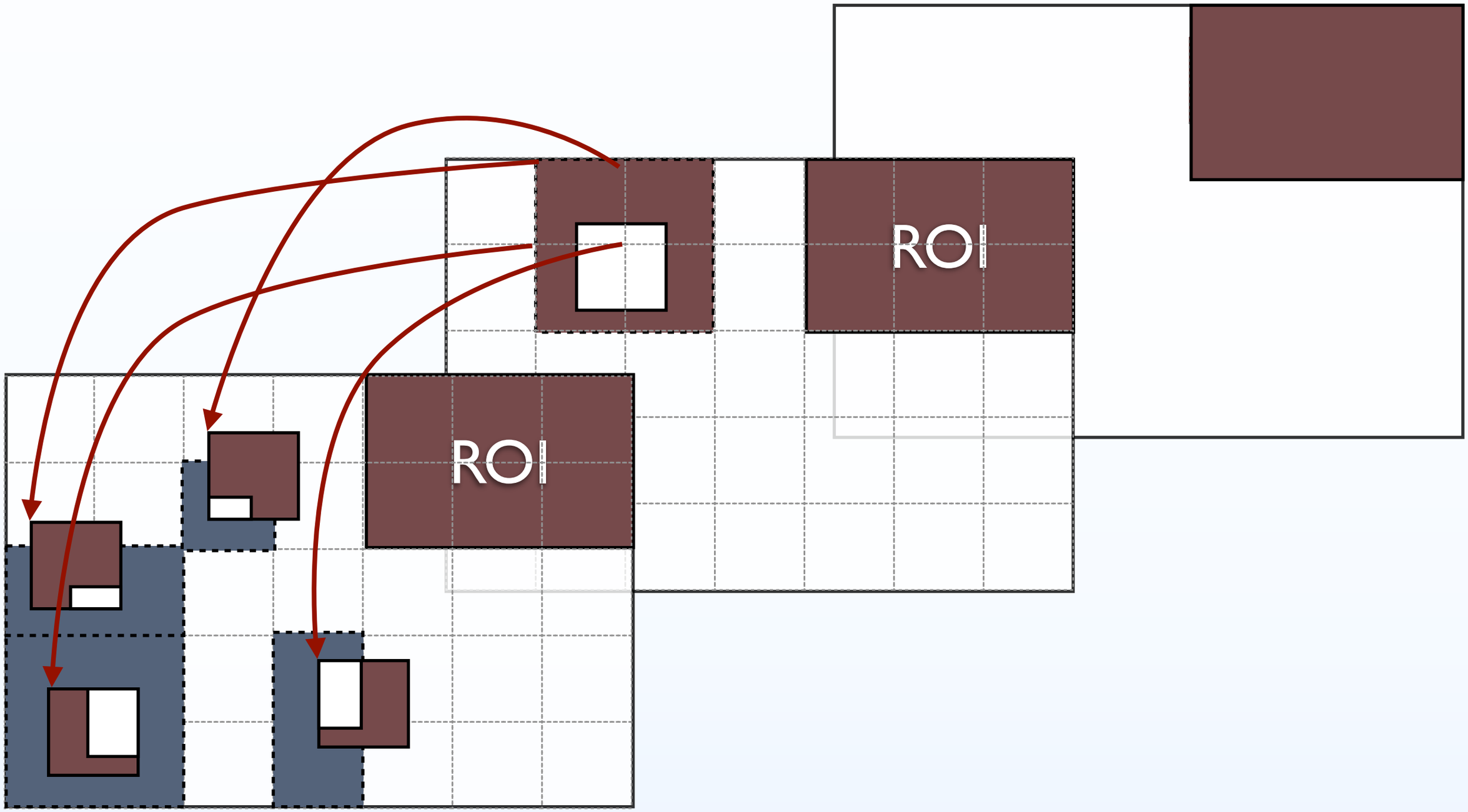


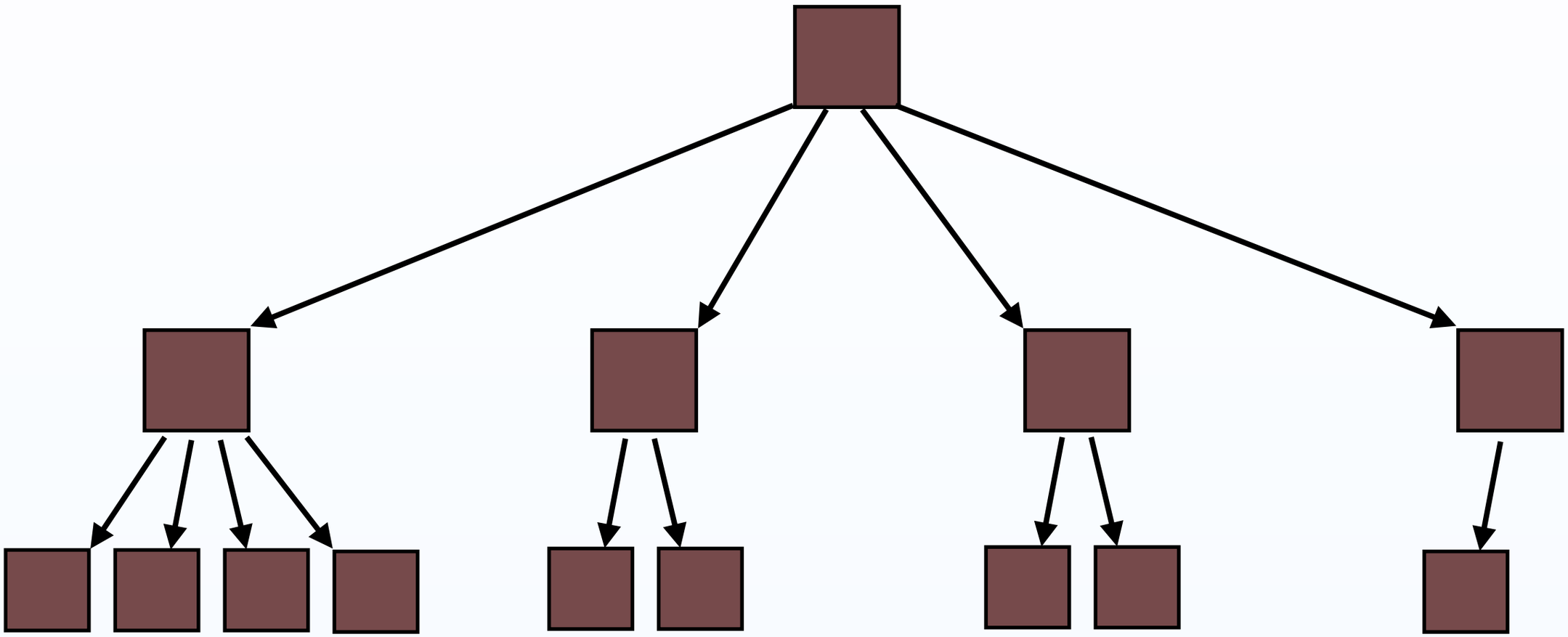
motion dependency propagates





careful optimization can reduce the dependency

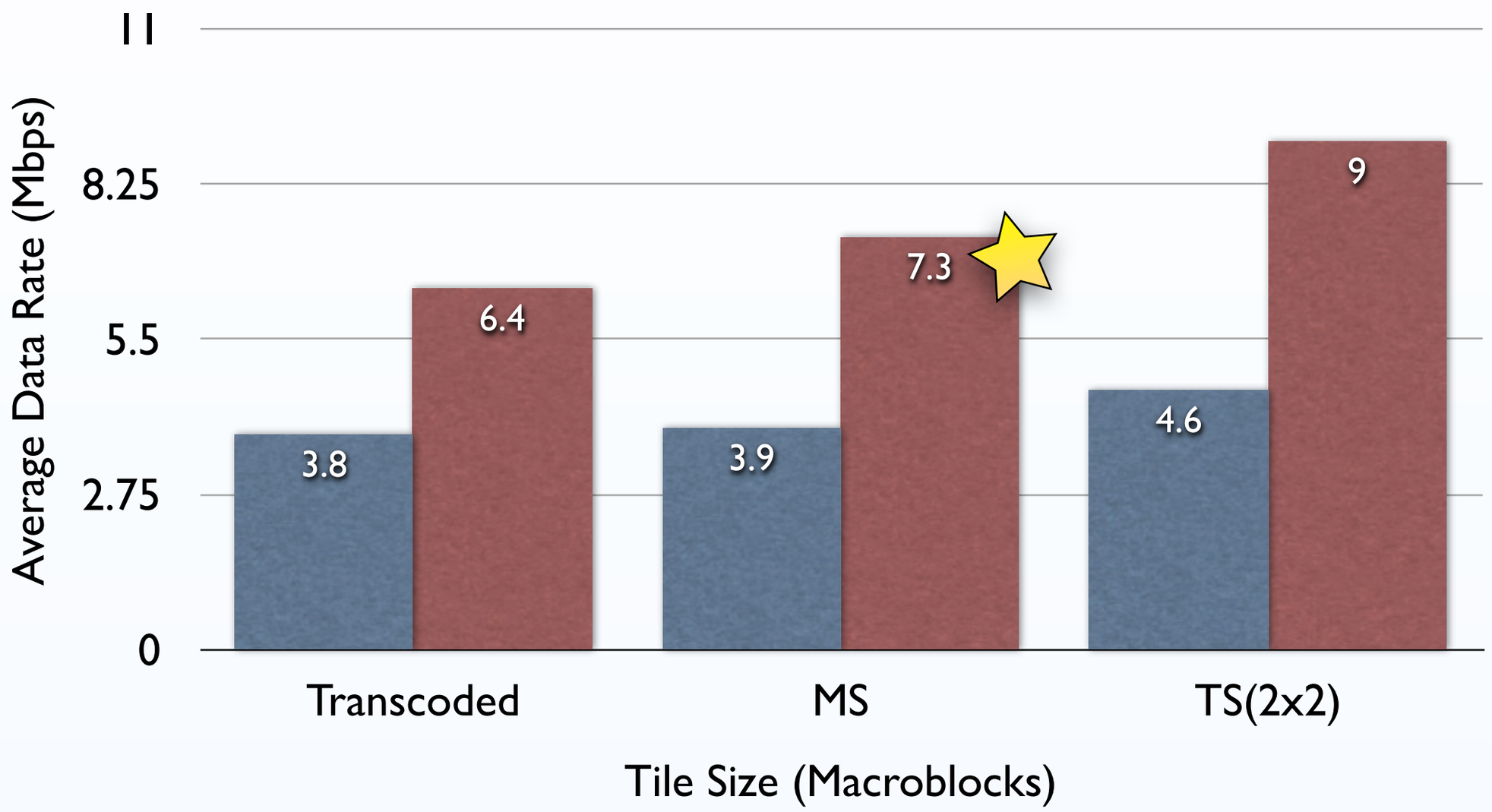




data structure: given a macroblock *m*, is there another macroblock *m'* inside the ROI that depends to *m*?

average 0.44 ms per frame

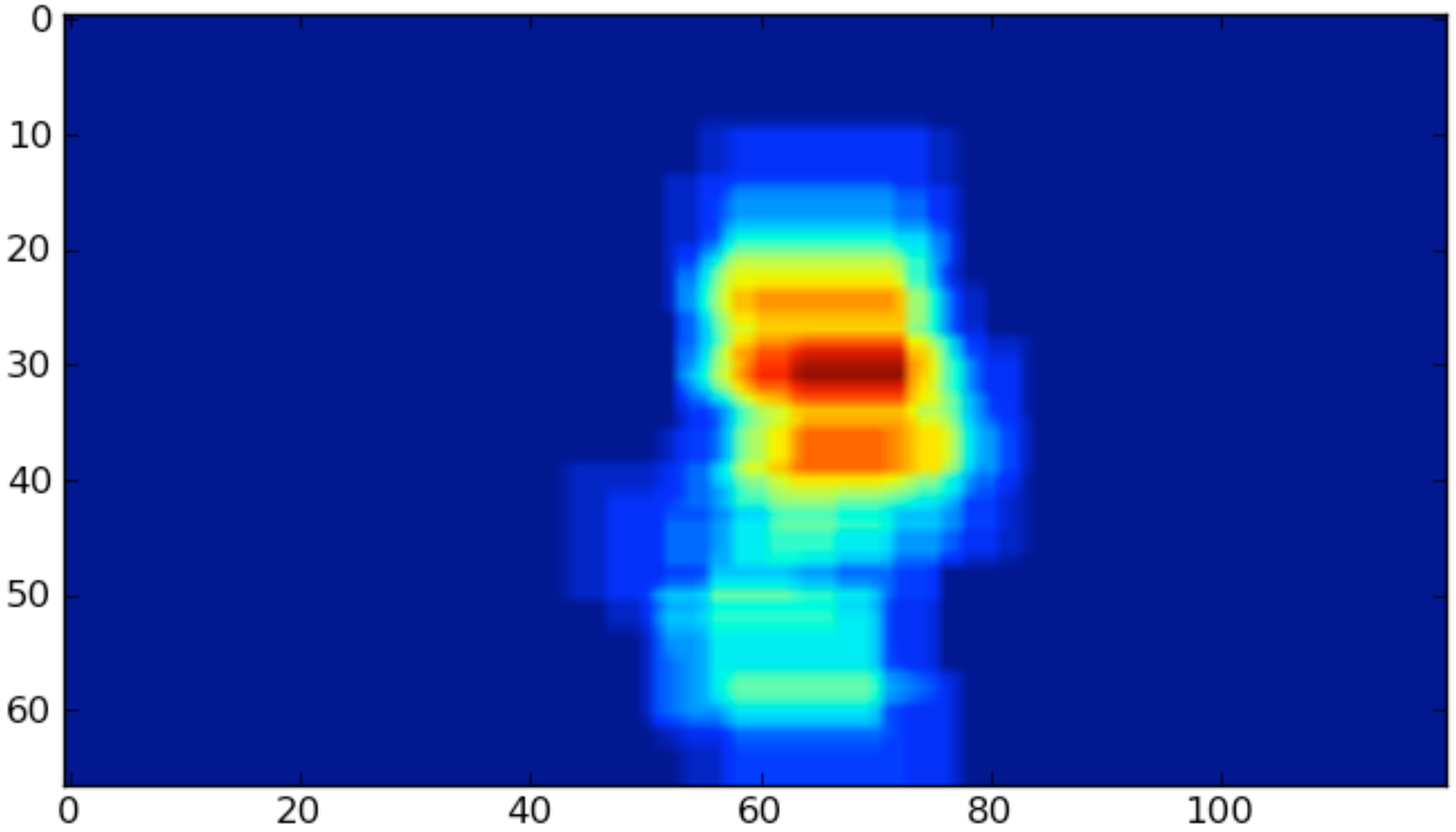
Average Data Rate When Transmitted ROI of 30x30 Macroblocks



(Slice Size **64** bytes. Max Motion Vector Length 16)

Monolithic	Tile
less bandwidth if encoding parameters carefully chosen	higher bandwidth
standard encoder can be used	need to modify encoder
much metadata needed	little metadata needed
no prefetching	prefetches surrounding areas

Can we adapt the encoding parameters based on user access patterns?


0
10
20
30
40
50
60
0
20
40
60
80
100

Observation 1

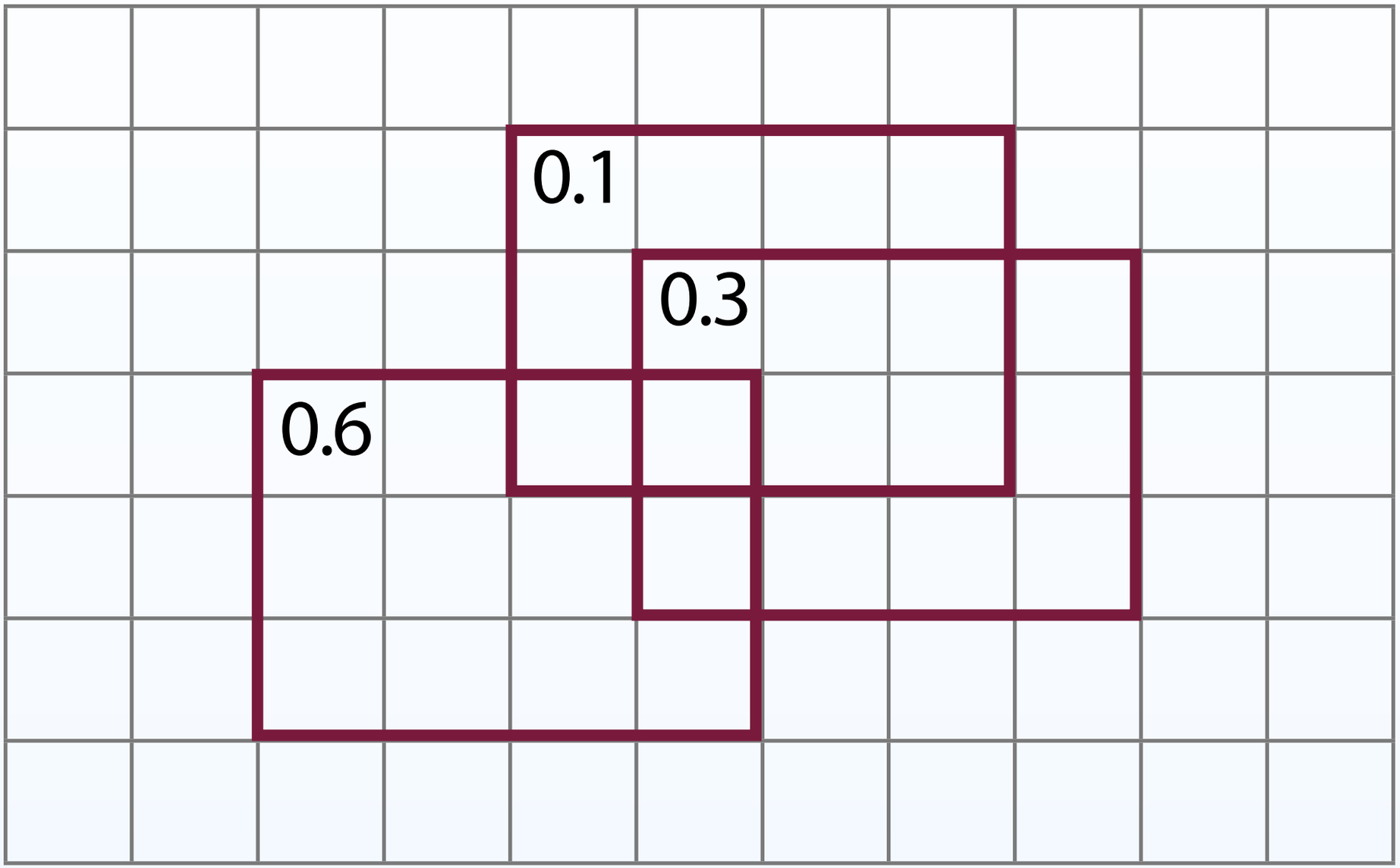
Areas with small access probability can have more dependencies and larger tiles

Observation 2

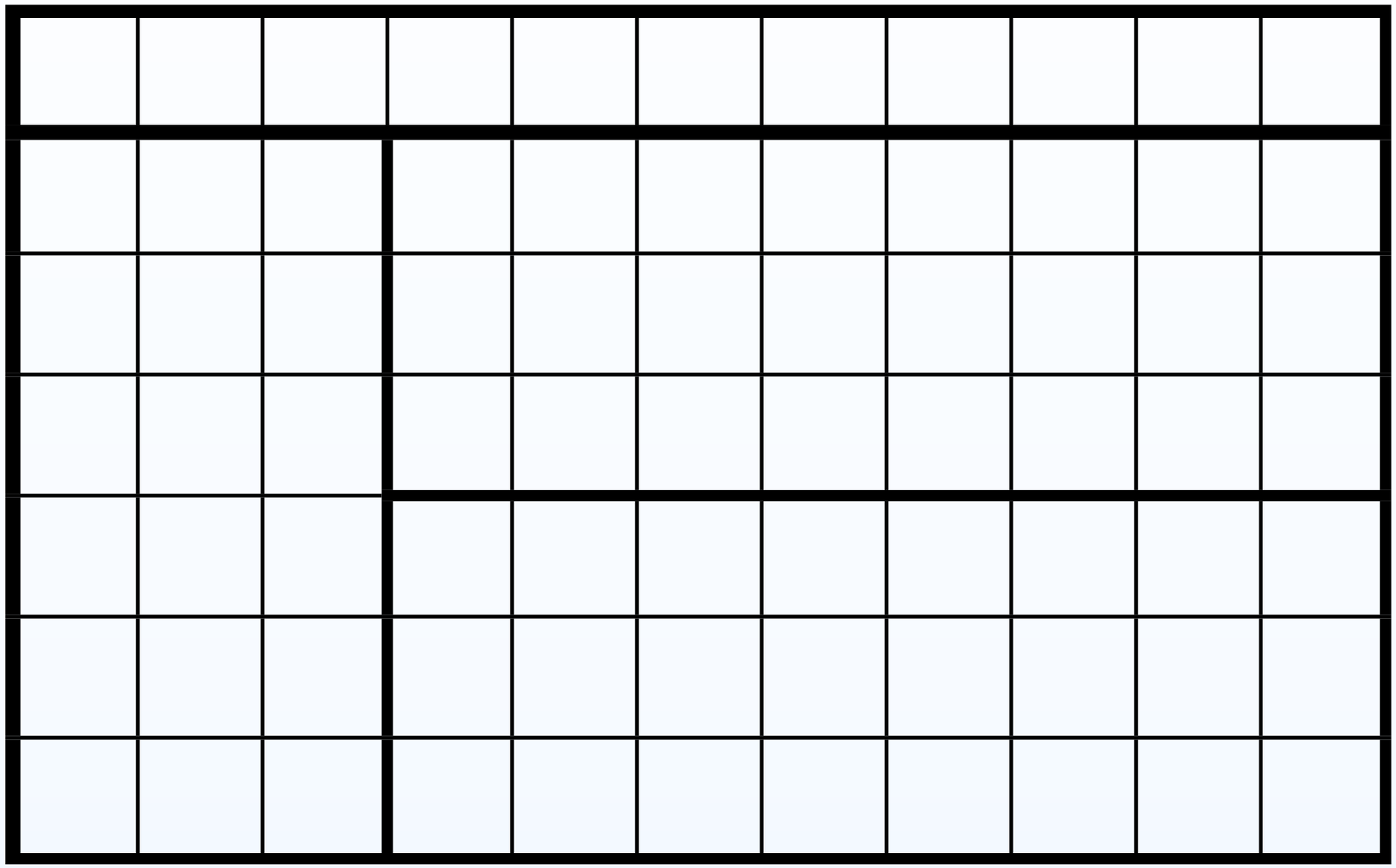
Areas that are accessed together can be put in the same tile

Given access pattern, find the best way to tile the video to reduce the expected bandwidth.

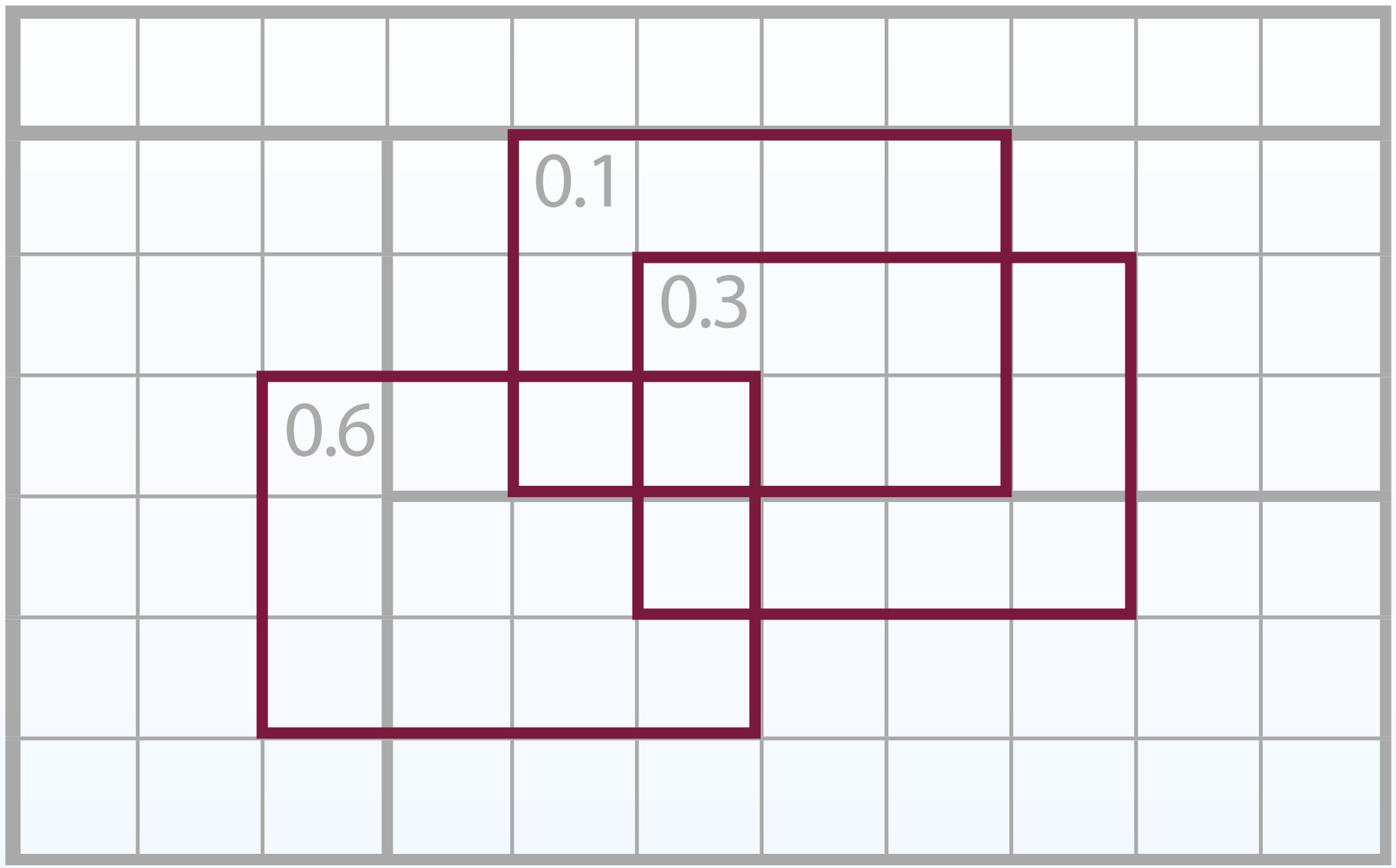
Access Probability



cost = size (in bytes) x total probability



cost = size of tile x total probability



A greedy heuristic:

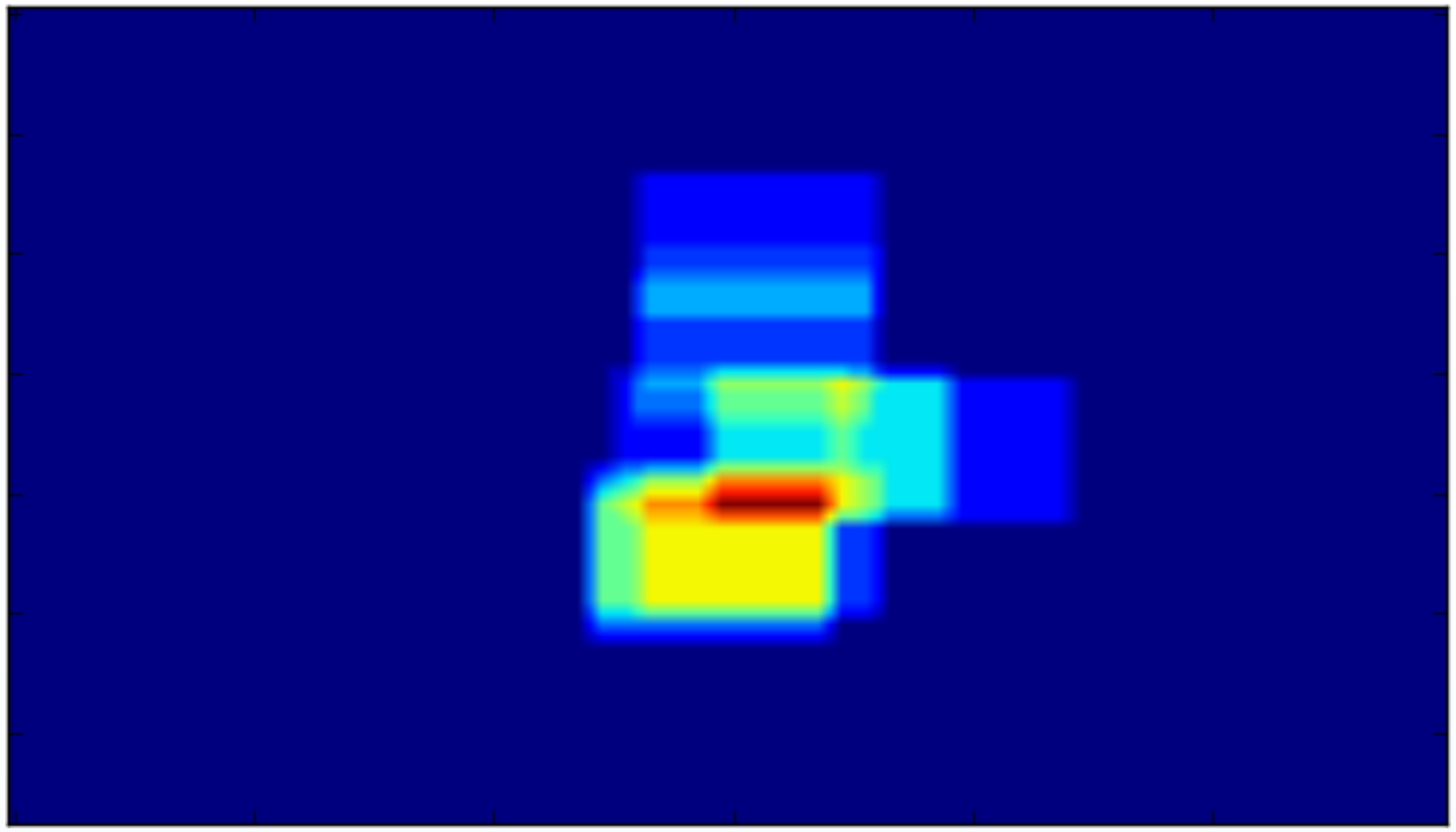
from left to right, top to bottom

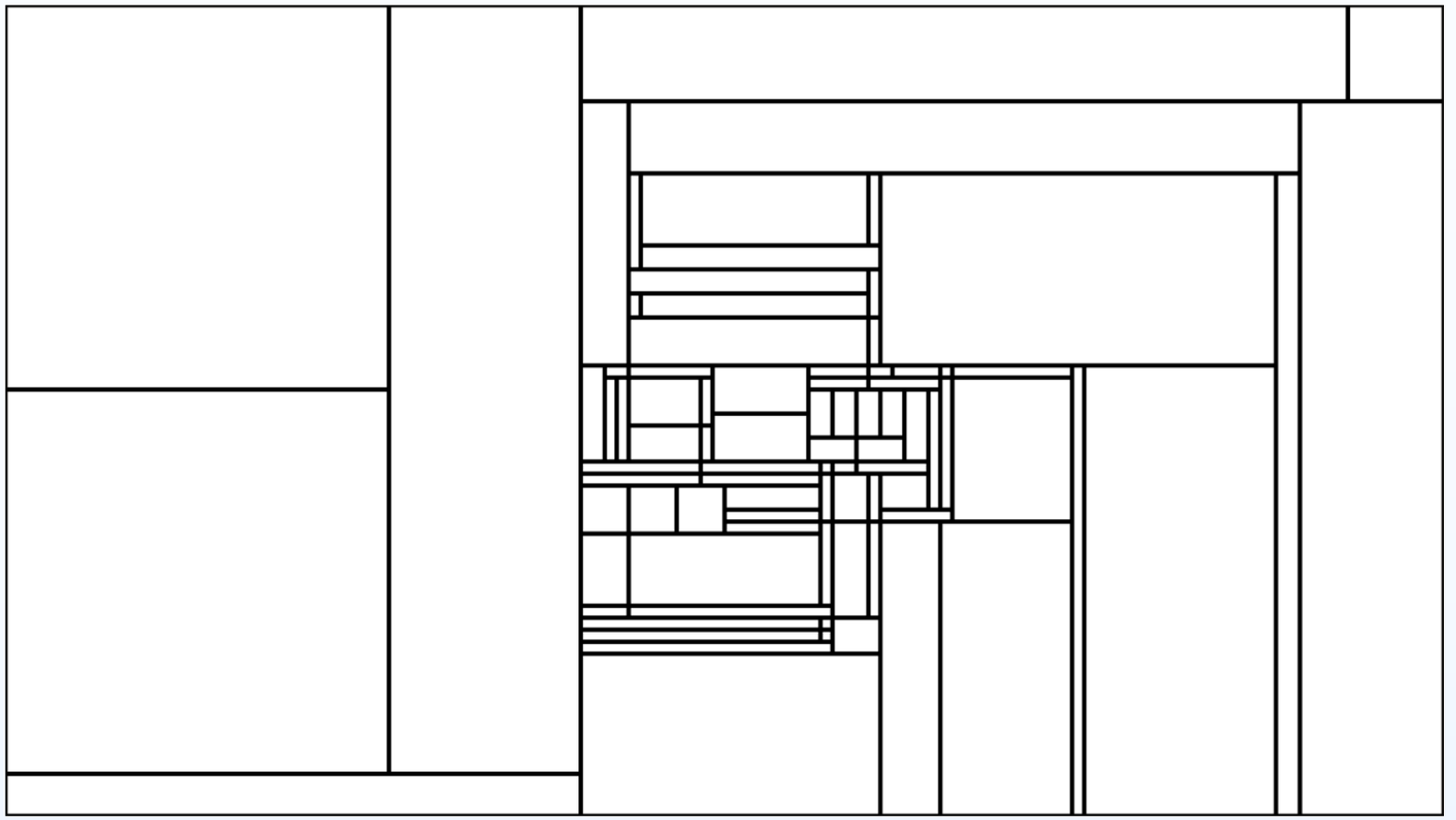
merge with neighbors if the expected size is reduced

cost = 9
p = 0.7

cost = 5
p = 0.8

cost = 12
p = 1

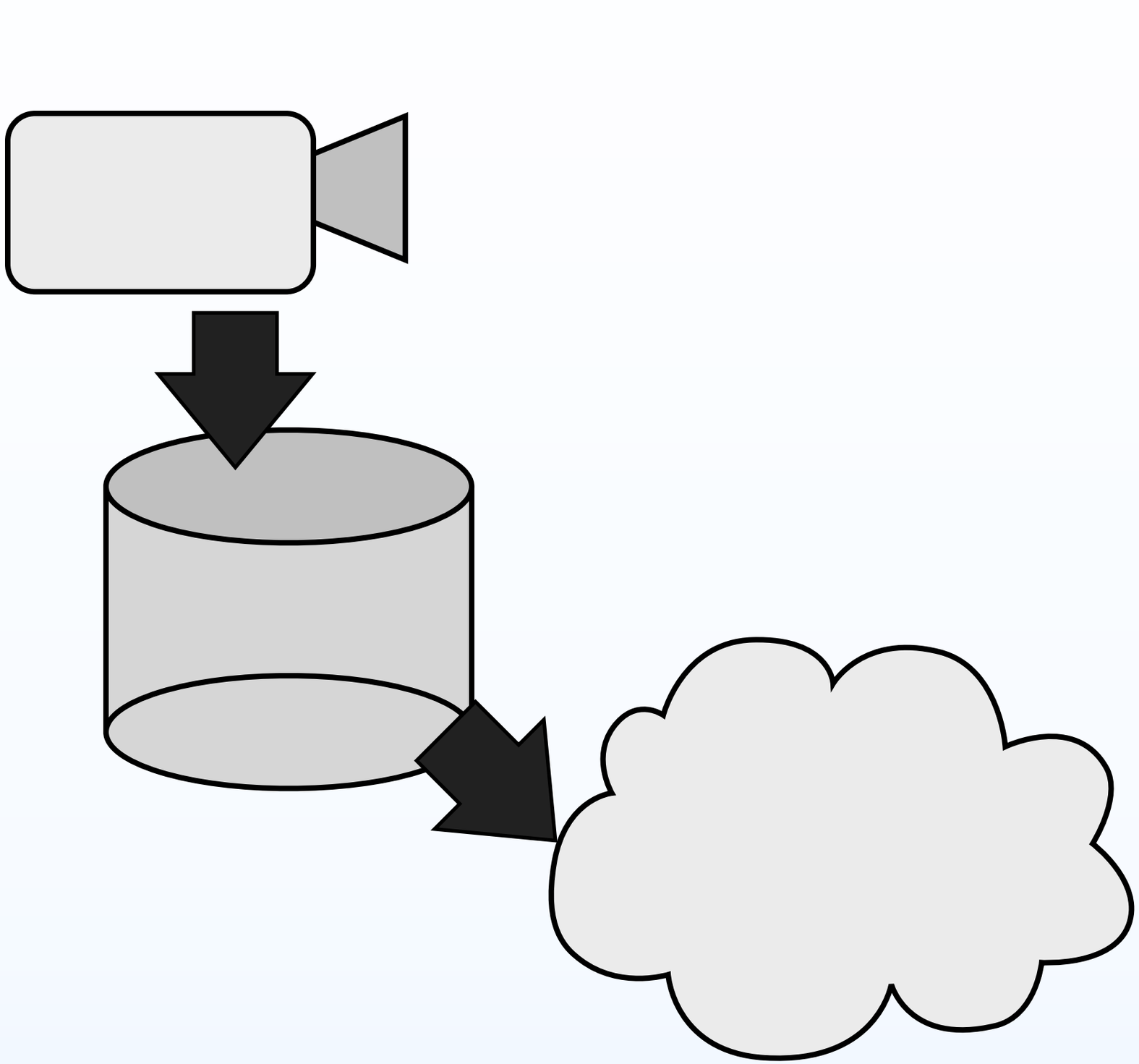




20 - 27%

Savings in Bandwidth

(even better than Monolithic Streaming)



Papers

MMSYS'10: Encoding

MM'10: Video retargeting

SAPMIA'10: User behavior

MMSYS'11: Adaptive Coding

MM'11: Better UI

MM'11: Tolerance to delay

Current Work

1. Live zoomable video

2. Streaming to many users

謝謝

歡迎發問及指教