Distributed Video Encoding

Austin Dworaczyk Wiltshire

Dr. Chris Lupo

Video Encoding In General

- Currently, it’s a one machine per video kind of world.
- Video encoders such as X264 are being optimized everyday for a *single machine* system, but not for massive scaling.
- Attack with cloud (or GPU) ninjas!

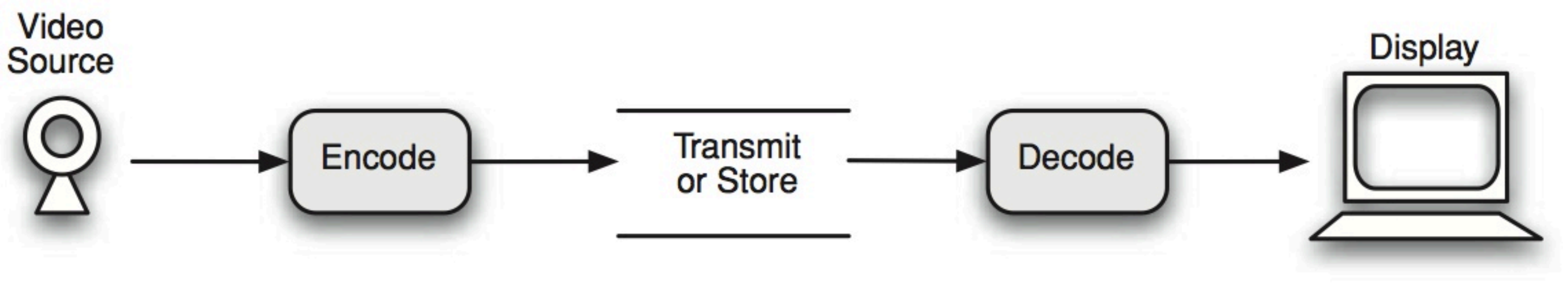
Past Work

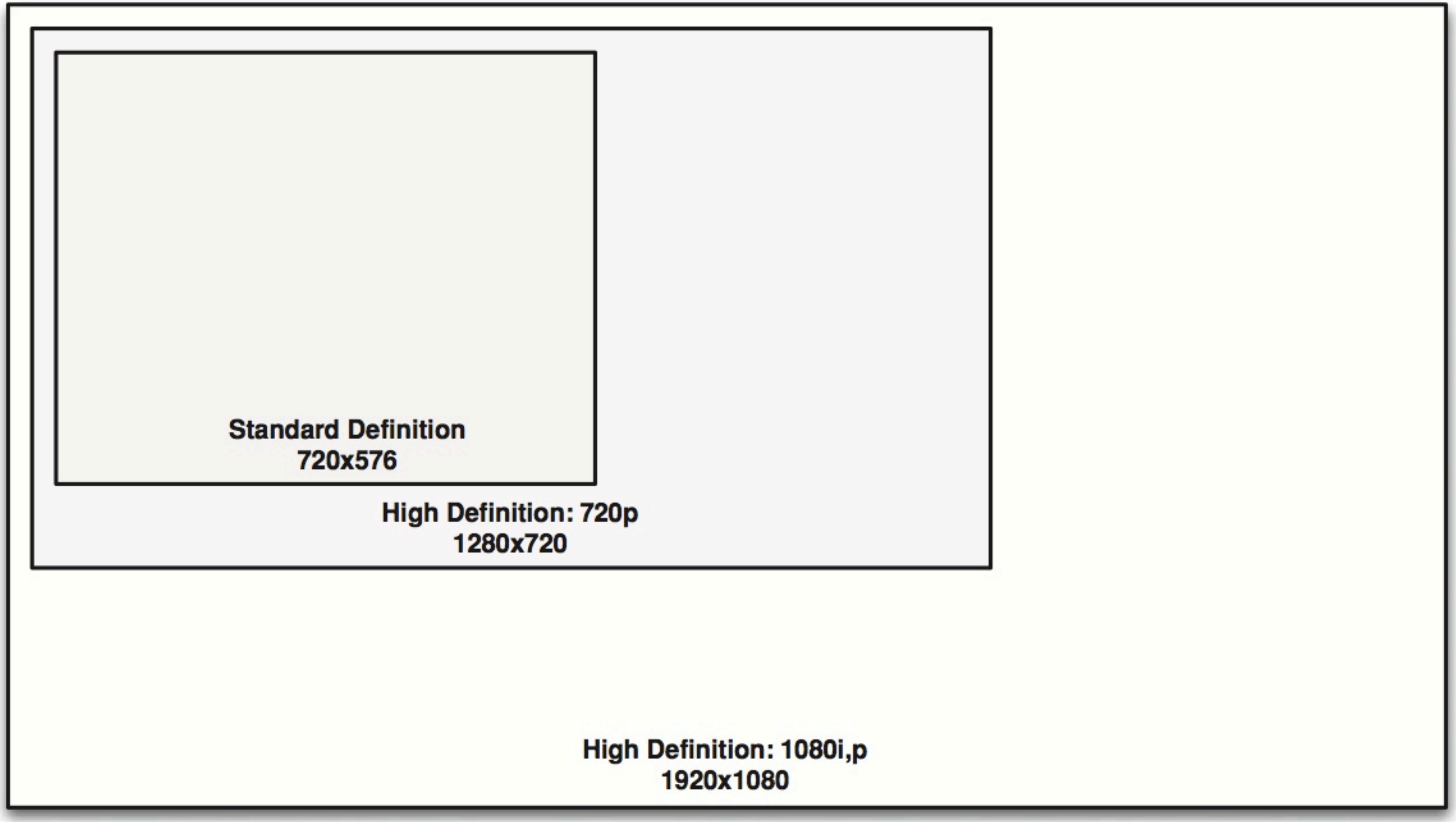
- Not a whole lot of work done (at least, any that is published).
- Mostly limited to educational papers with little to no resulting code.
- Most examples of such systems are closed source and/or dead. (RipBot, x264farm)
- Some examples of parallelizing specific motion search algorithms, but not the whole process.

Modern Rendition

- Ideally create an open source project for distributed encoding.
- Use the modern H264 standard, implemented with the X264 encoder.
- Determine the optimal design structure (shared memory, segment passing, etc).
- Possibly enhance using GPUs.

A bit of background...



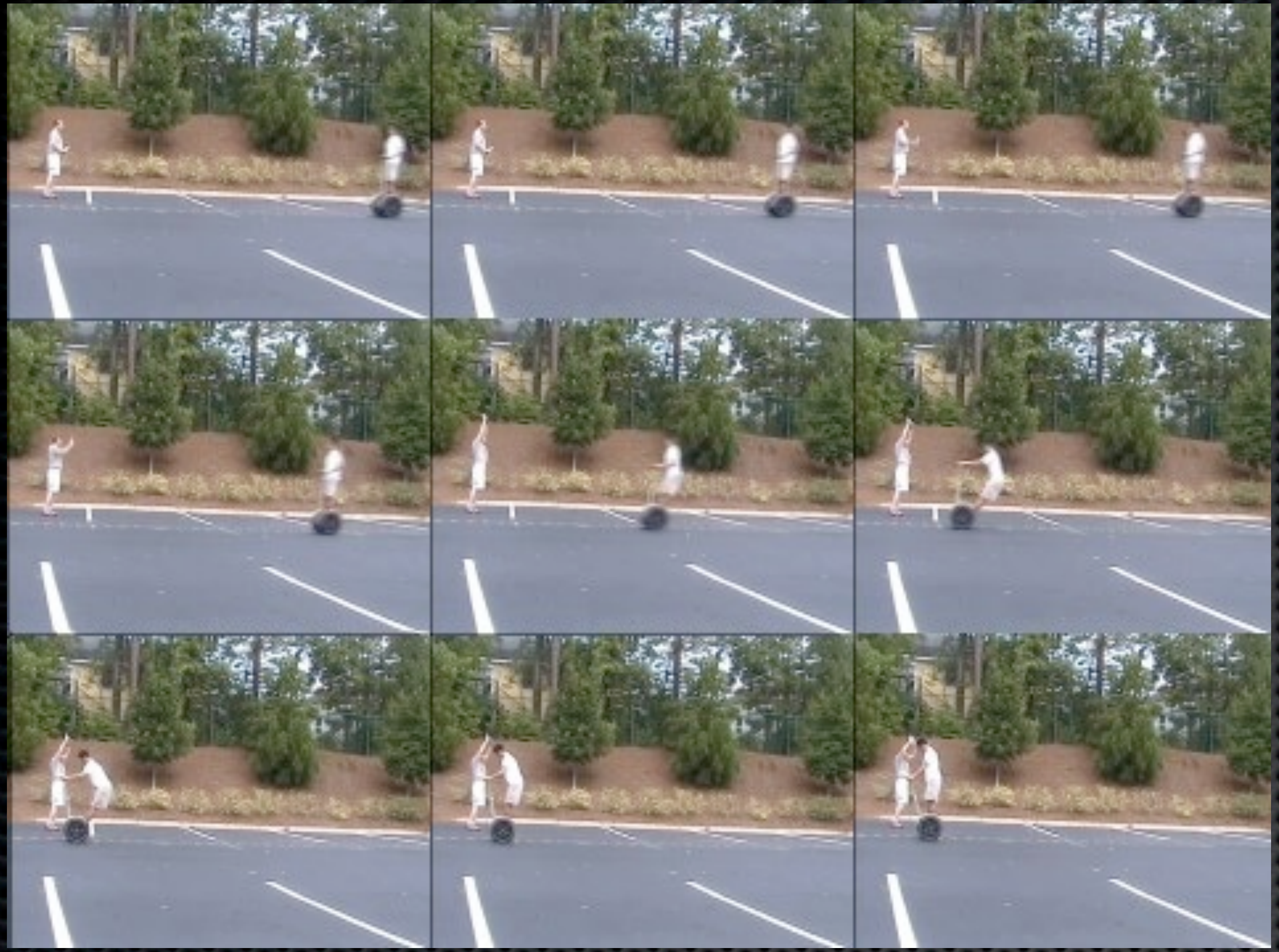


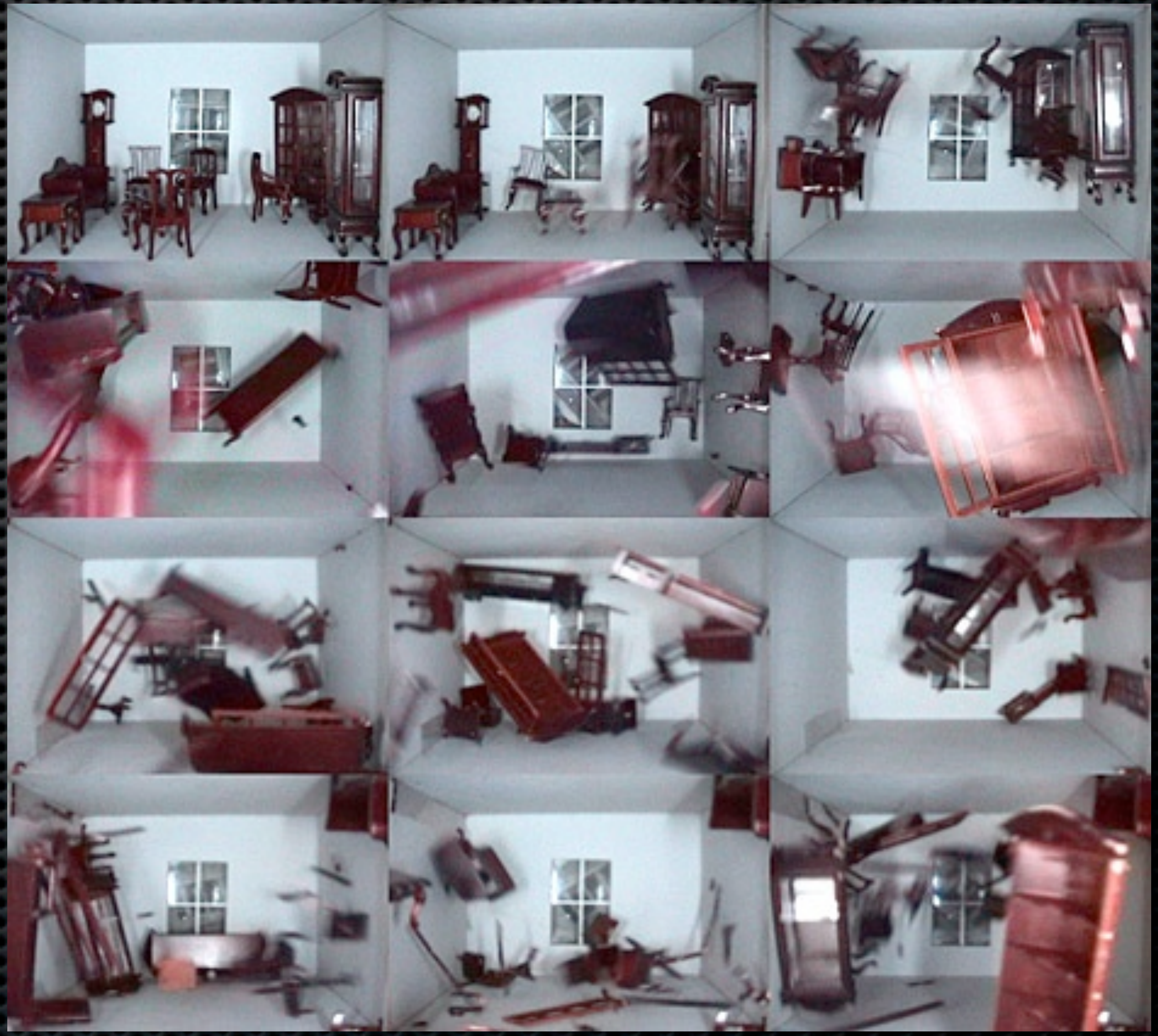
Standard Definition
720x576

High Definition: 720p
1280x720

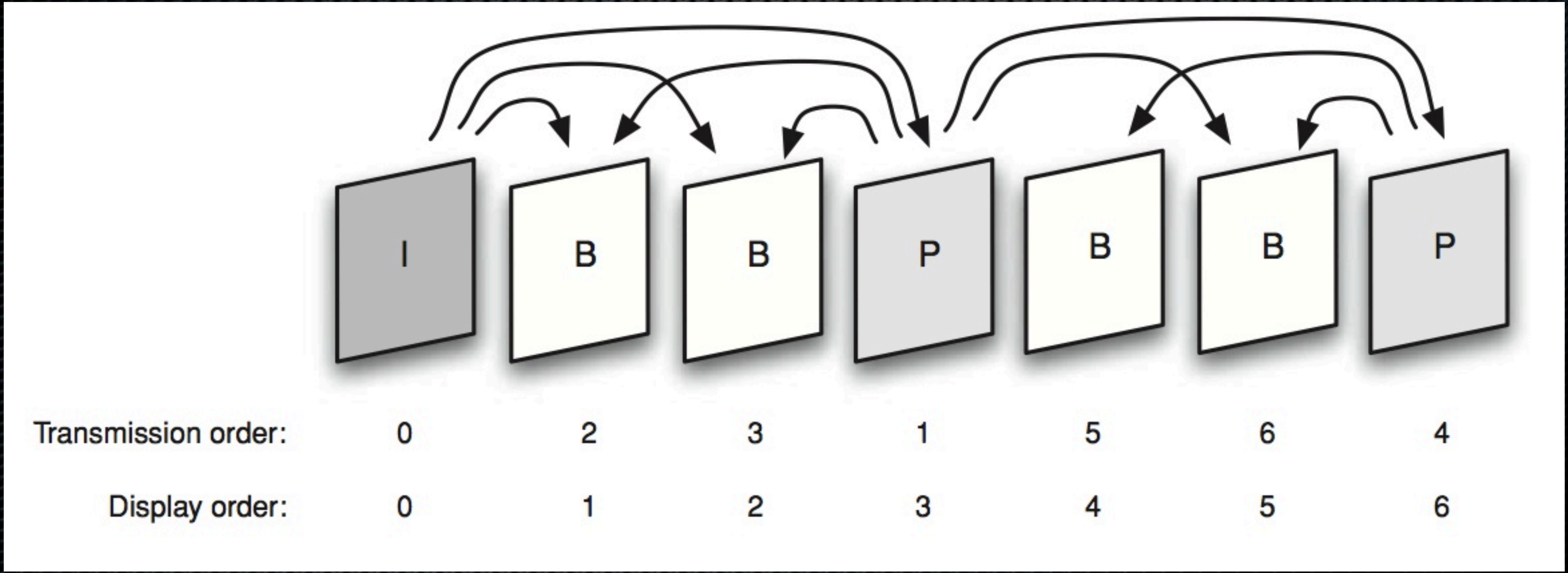
High Definition: 1080i,p
1920x1080

Break it into chunks..



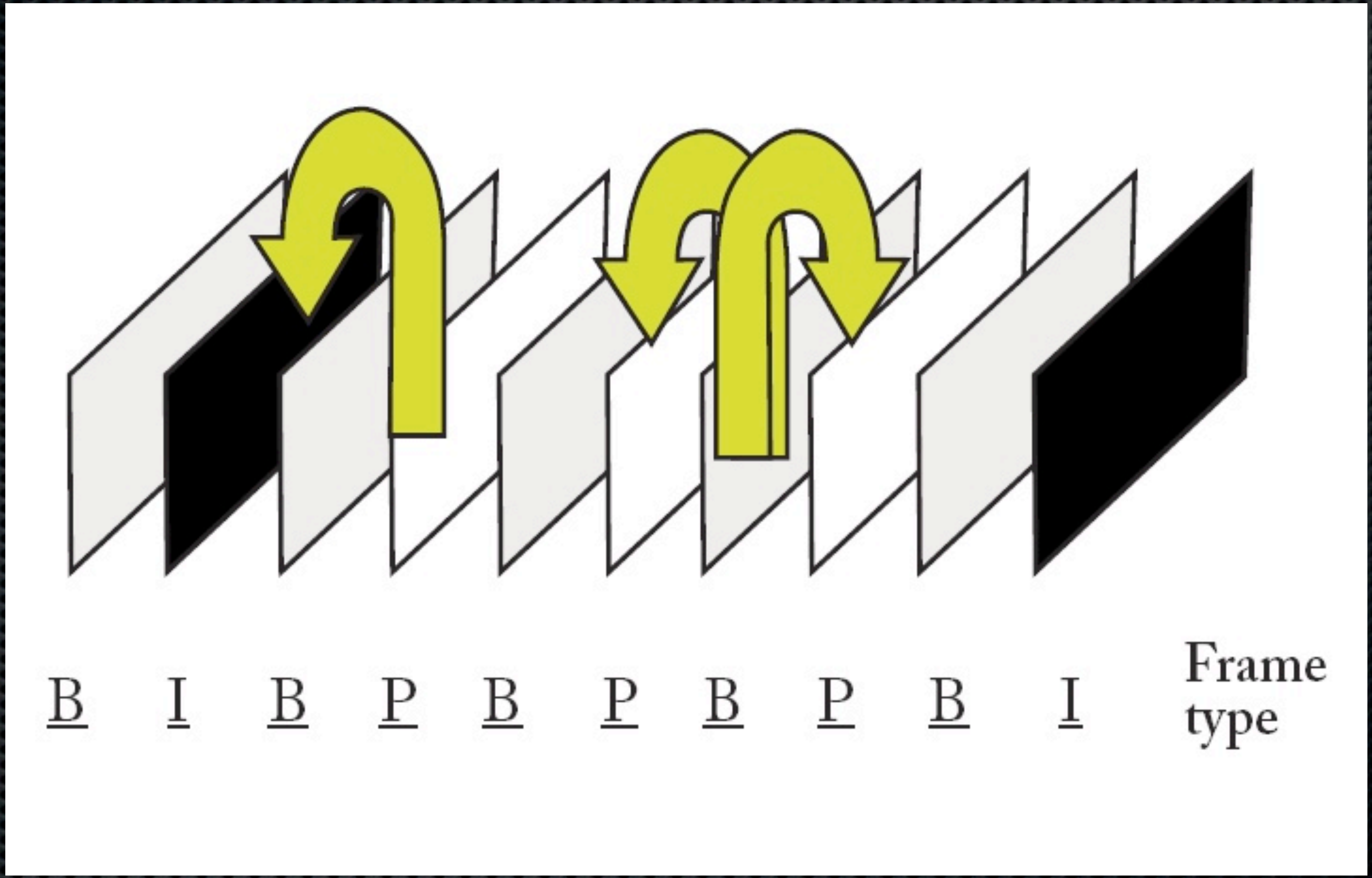


Ordering Matters



	I	B	B	P	B	B	P
Transmission order:	0	2	3	1	5	6	4
Display order:	0	1	2	3	4	5	6

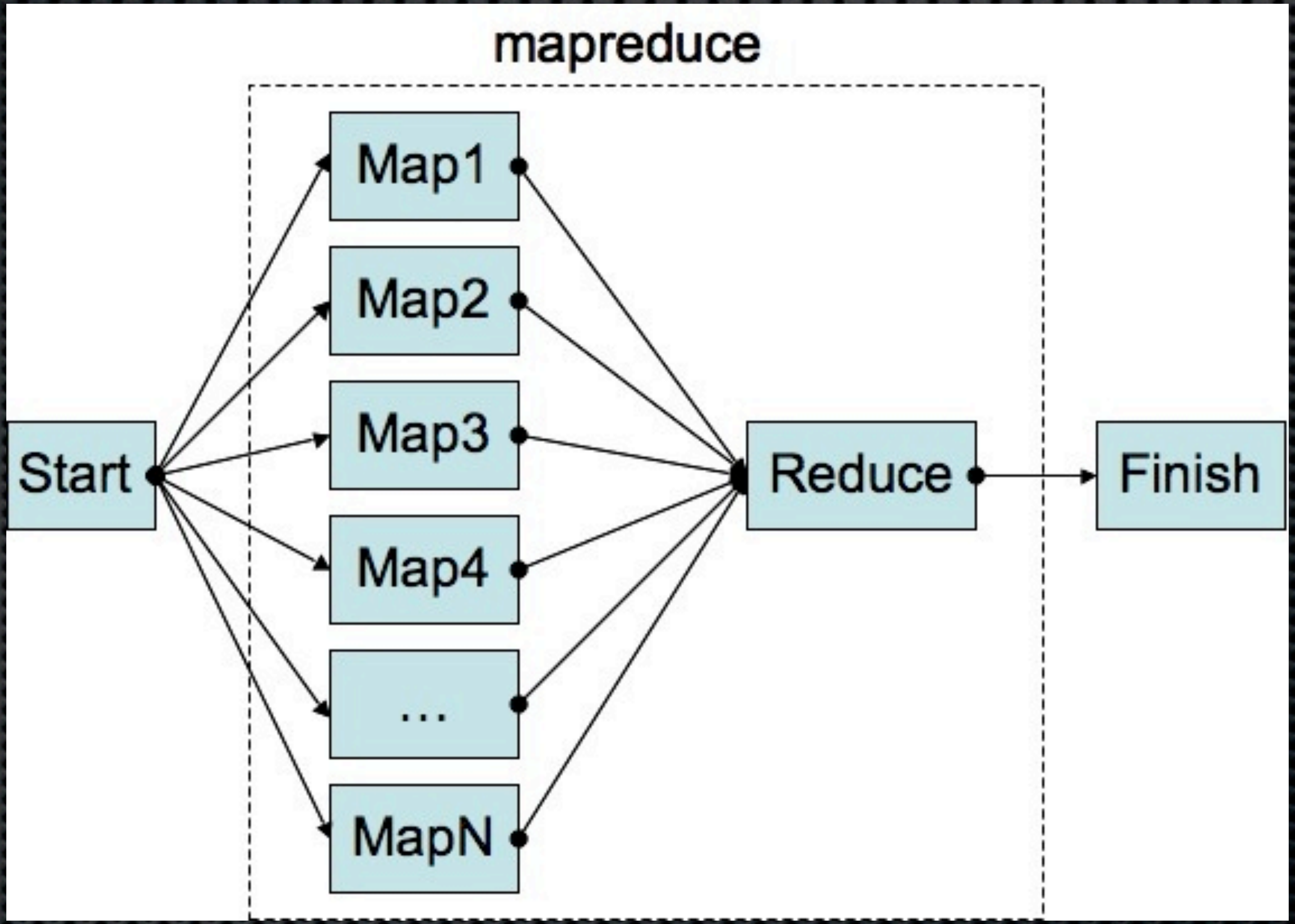
Another Example



A quick algorithm

```
identify key-frames
open chunk
for each frame in input video
    if frame is key-frame and chunkSize > GOP
        add frame to chunk
        close chunk
        open new chunk
    add frame to chunk
close chunk
```

Map Reduce... sorta



Just one problem...

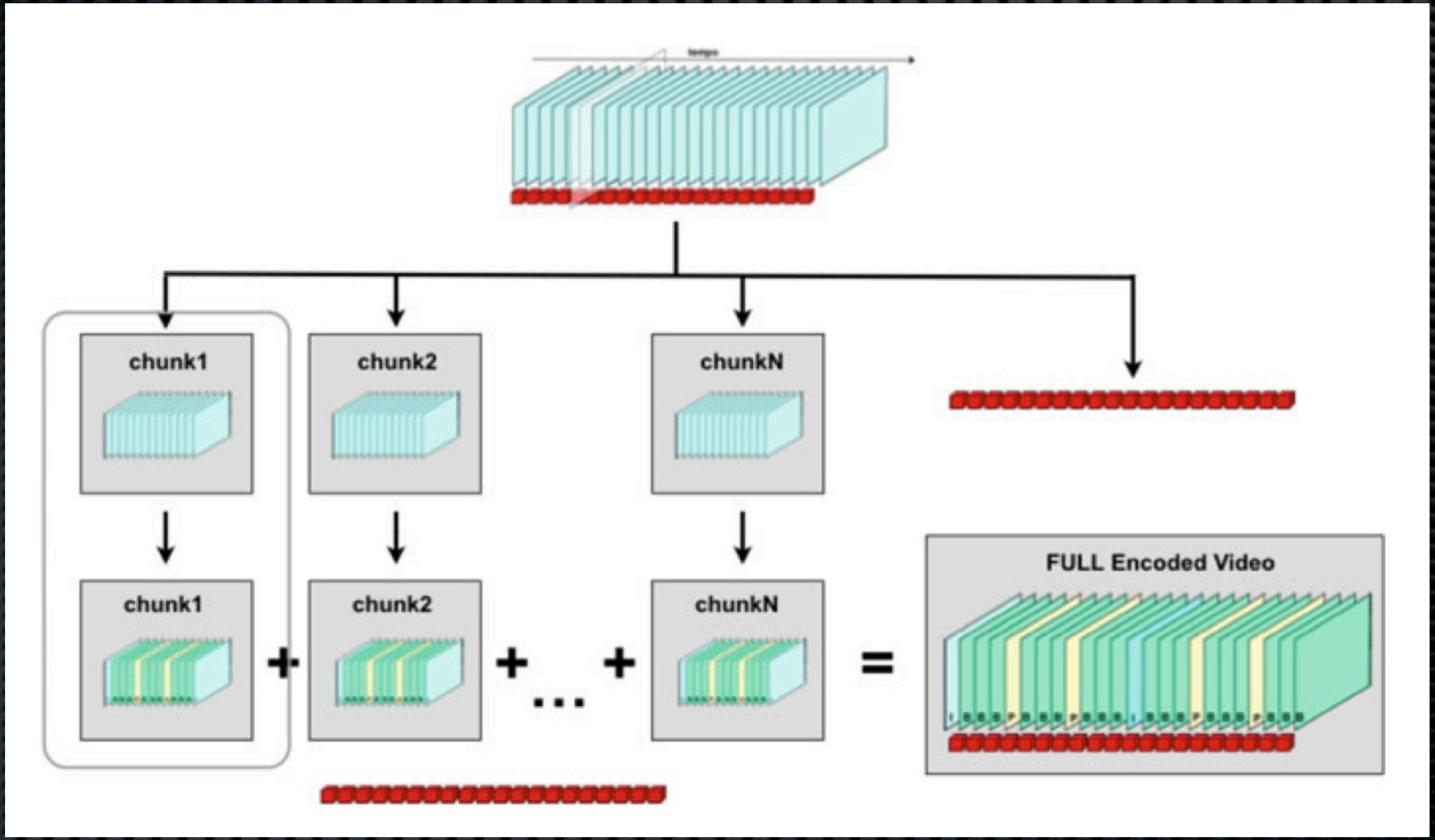
Just one problem...

Audio

A few other problems

- Order
  - Frame Dependence
  - Overall sequence ordering

Split 'n Merge



tempo
chunk1
chunk2
chunkN
chunk1
chunk2
chunkN
+
+
...
+
=
FULL Encoded Video