RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

YALE

Carnegie Mellon University

High-Rate Flicker-Free Screen-Camera Communication with Spatially Adaptive Embedding

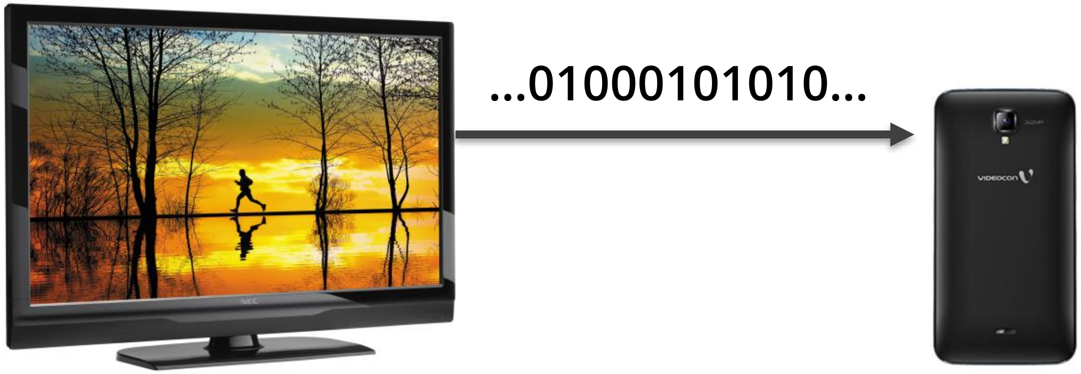
Viet Nguyen*, Yaqin Tang*, Ashwin Ashok §, Marco Gruteser*, Kristin Dana*, Wenjun Hu#, Eric Wengrowski*, Narayan Mandayam*

* Rutgers University

§ Carnegie Mellon University

# Yale University

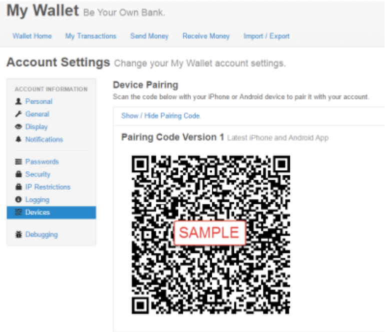
Screen-Camera communication



Watch demo videos on your phone by scanning this tag with your QR scan reader.

Or text the word "Touch" to 70734.

Watch demo videos on your phone by scanning this tag with your QR scan reader. Or text the worD Touch to 70734.


My Wallet Be Your Own Bank.

Account Settings Change your My Wallet account settings.

Device Pairing

Scan the code below with your iPhone or Android device to pair it with your account.

Show / Hide Pairing Code.

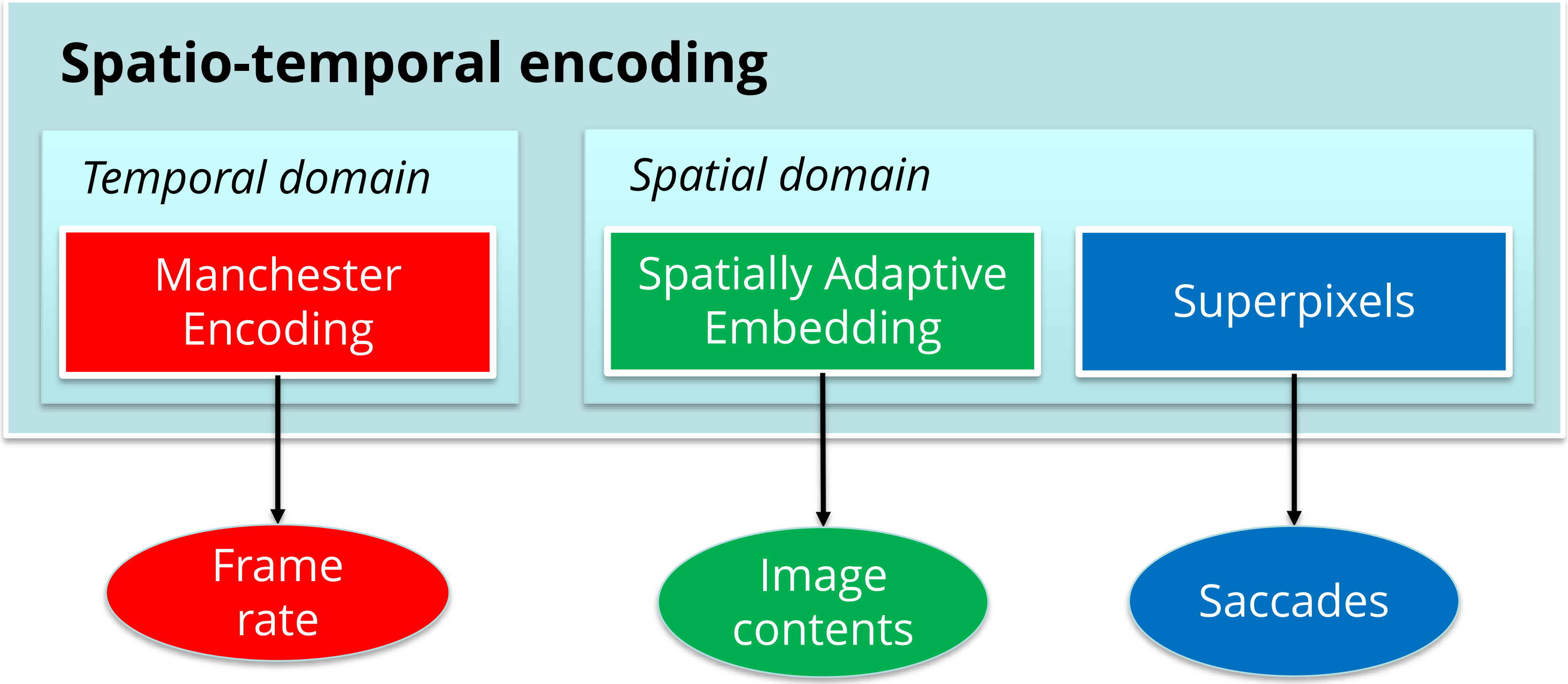
Pairing Code Version 1 Latest iPhone and Android App

- ❌ Occlusive
- ❌ Static data
- ❌ Low throughput

Embedded Screen-Camera communication

- Experience normal full-screen contents
- Provide high throughput data communication, but imperceptible to viewers

TextureCode: Design



- ➢ Reduce flicker
- ➢ Combine multiple dimensions of coding opportunities
- ➢ Increase communication throughput

Flicker perception

- ➢ **Frame rate**
- ➢ Image contents
- ➢ Saccades

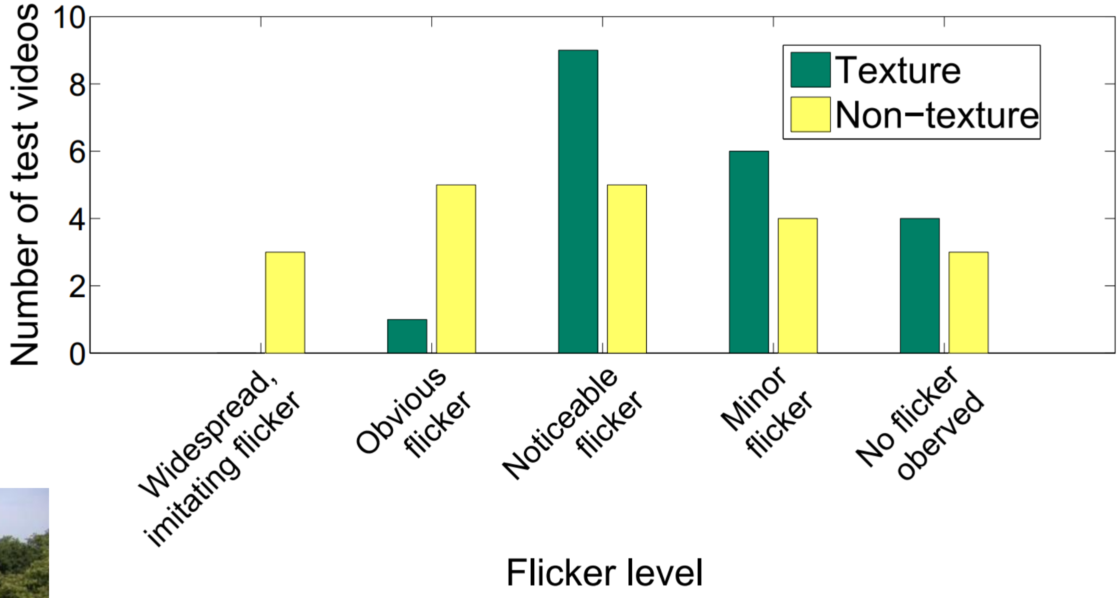
Fixed intensity *Changing intensity*

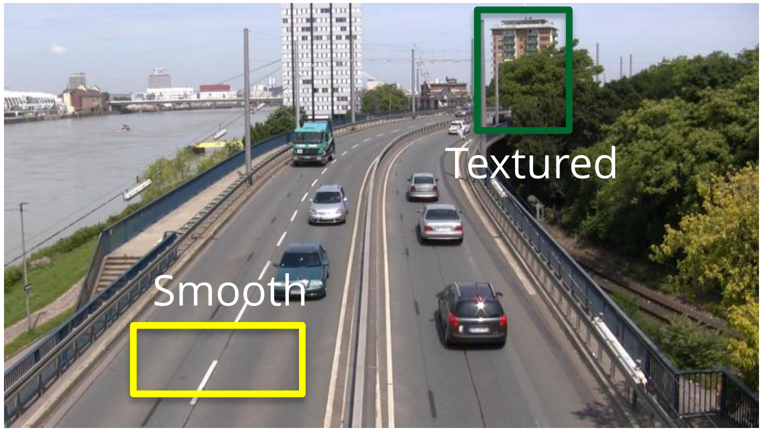
- At 120fps, no difference between two blocks
- Human’s eyes cannot perceive high speed change of light

→ Use high frame rate videos to embed messages

Flicker perception

- Frame rate
- **Image contents**
- Saccades



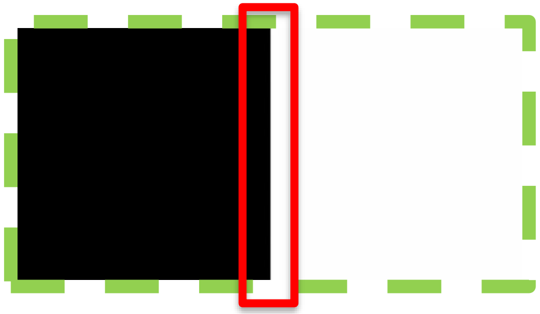


- Intensity modifications in smooth regions are more likely to cause flicker than textured region.

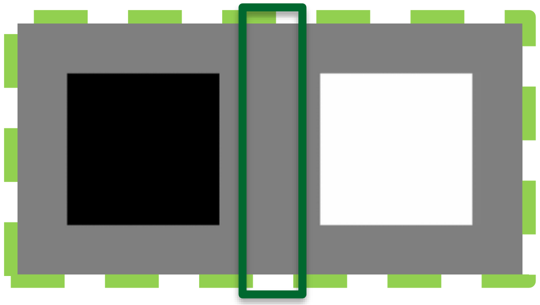
Flicker perception

- Frame rate
- Image contents
- **Saccades**

  (rapid eye movements)

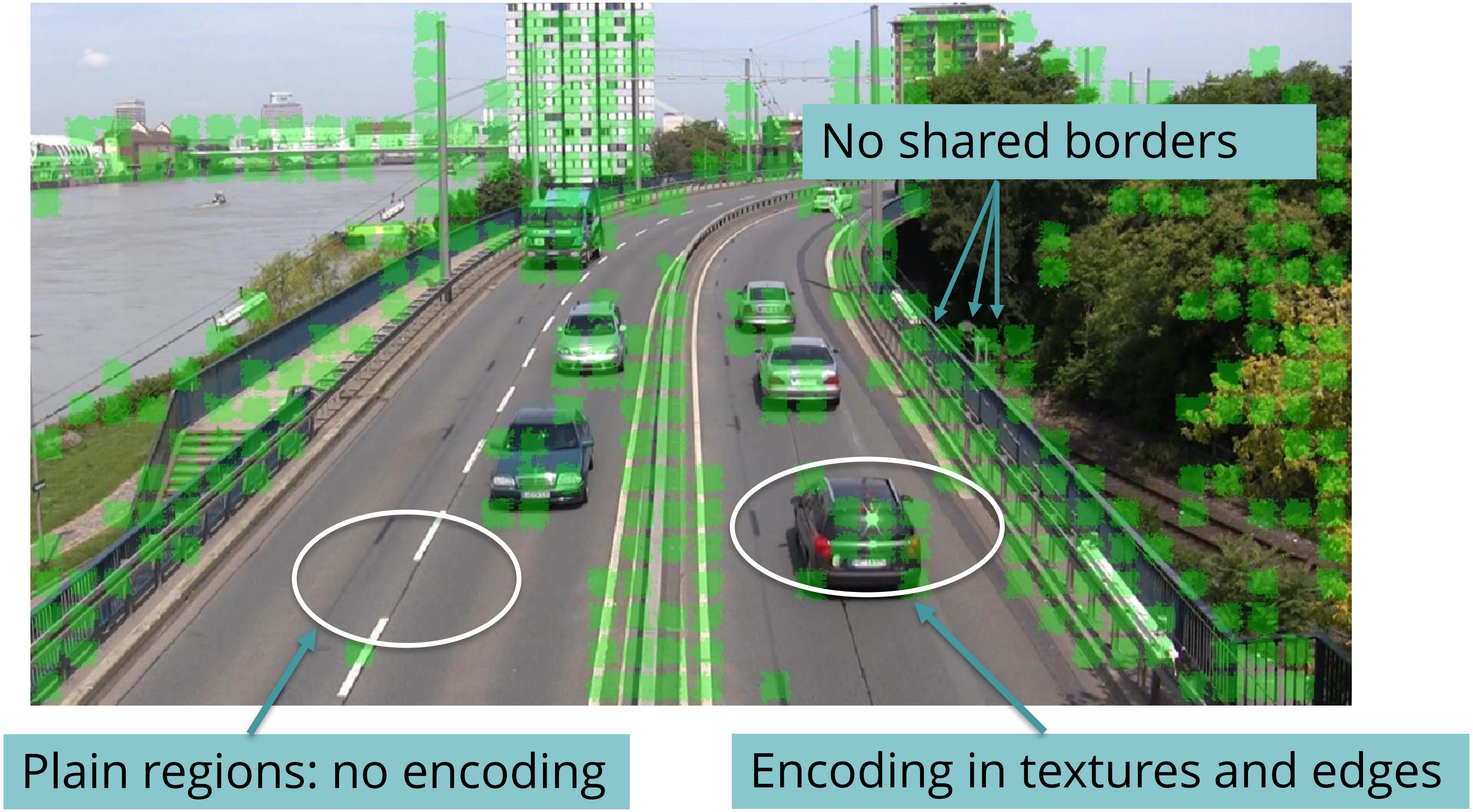


noticeable flicker,
even at high frequency



separating the blocks with some distance can reduce the flicker effects

TextureCode: An example

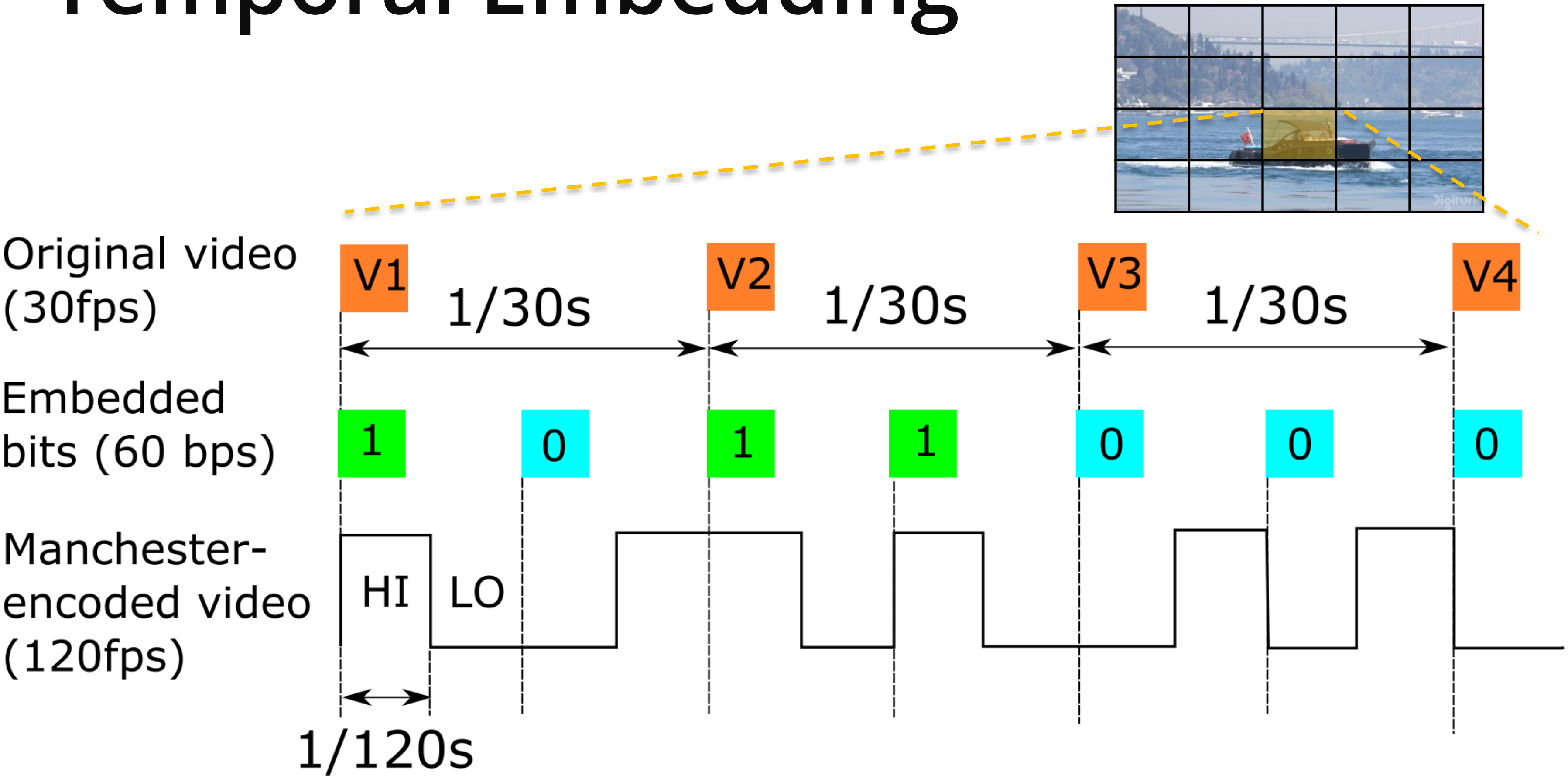


No shared borders

Plain regions: no encoding

Encoding in textures and edges

Temporal Embedding

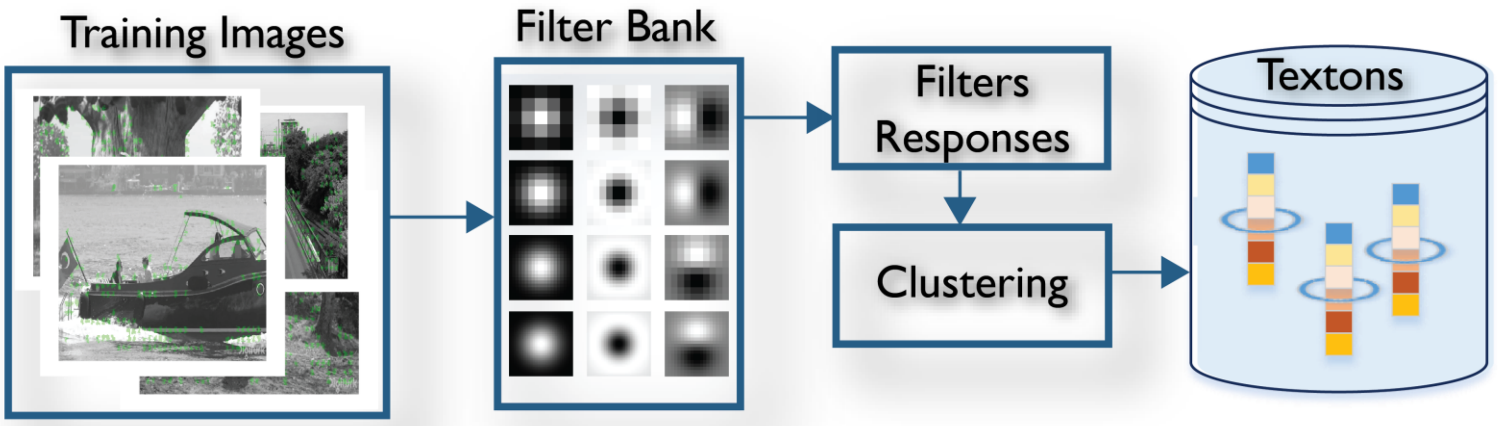


HI = original pixel intensity + α (brighter)

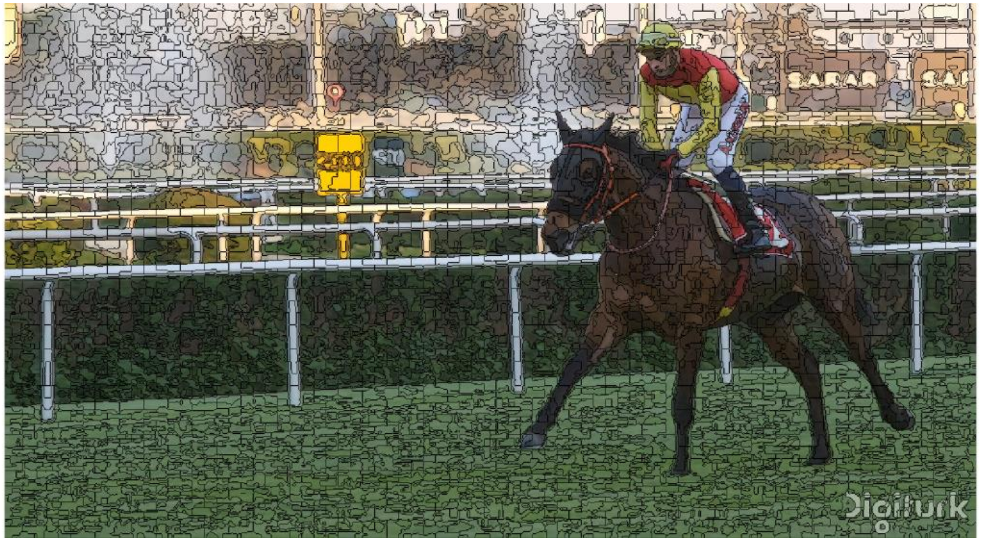
LO = original pixel intensity – β (darker)

Spatially Adaptive Embedding

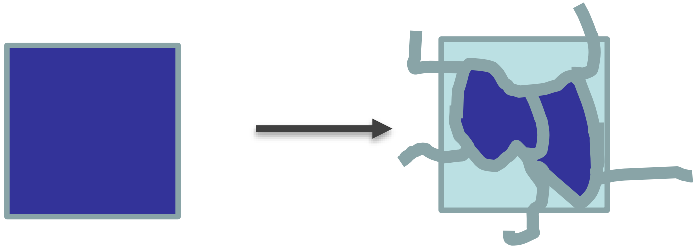
- Find low-flicker blocks when encoding by Manchester
- Two proposed methods
  - Texton analysis → Static videos
  - Pixel-based texture analysis → Dynamic videos



Superpixels



Find natural edges inside frame



all pixels

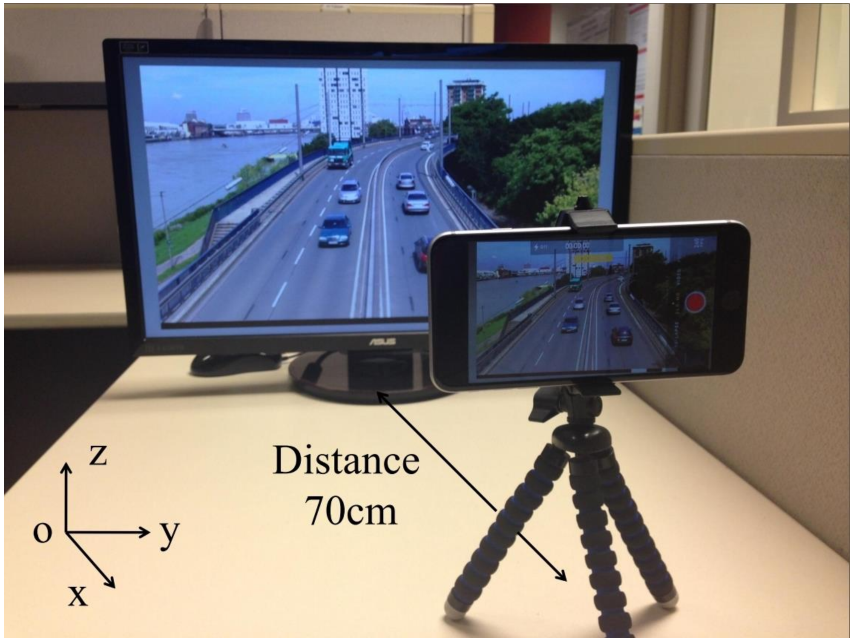
only superpixels inside block

Block–superpixels hybrid encoding

- Remove boundaries between encoded units
- Preserve block-based decoding method

Experiment setup

Environment	Office settings
Monitor refresh rate	120Hz
Video resolution	1280x720
Video encoding rate	120fps
Camera receiver	iPhone 6
Receiver frame rate	240fps (Slo-mo)



- *The transmitter and receiver work offline*

Evaluation metrics

- Bit error rate
- Goodput

$$\text{Goodput} = \sum_{\text{all frames}} \frac{D}{t}$$

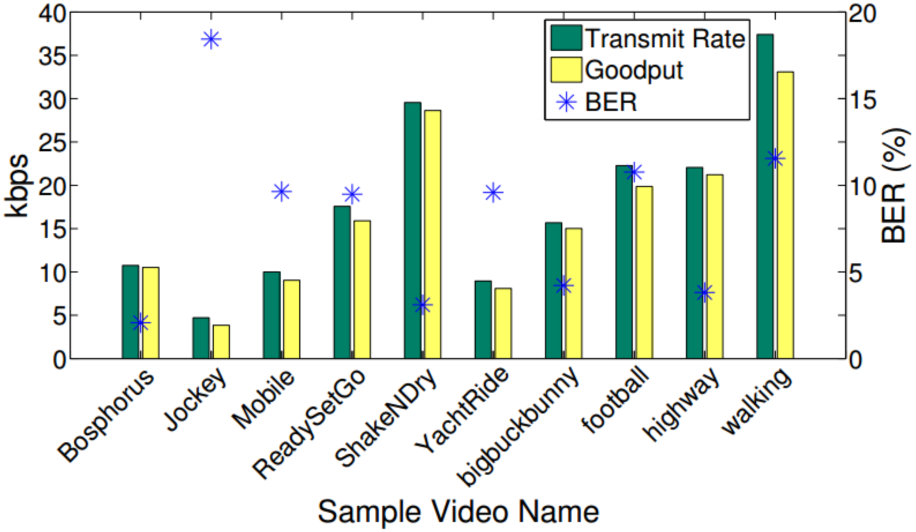
D - # correctly decoded bits
t - transmission time

- Transmit rate

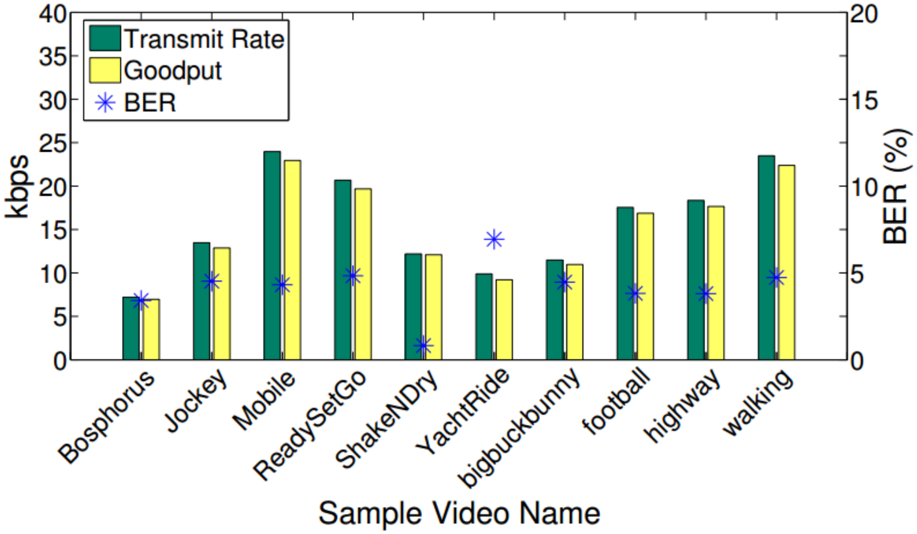
$$\text{Transmit Rate} = \sum_{i=1}^{N} \frac{B_i \times b \times V}{N \times F}$$

B_i - # encoded blocks in frame i
b - # bits encoded in each block
V - video frame rate
N - # frames in the video
F - # frames to encode one bit

Transmit rate – Goodput - BER



Dynamic videos (16kbps average)



Static videos (15kbps average)

Comparison between schemes

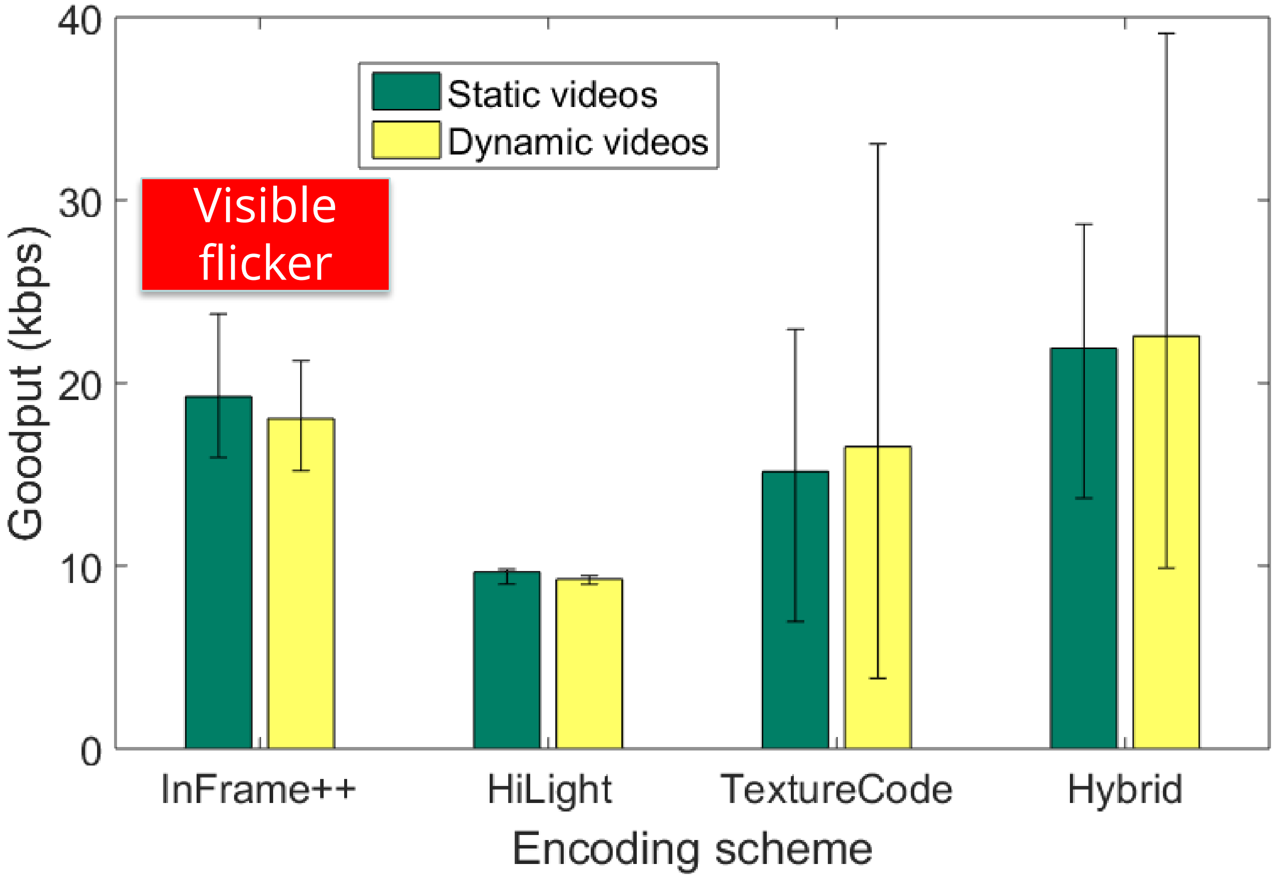
- ***InFrame++***
  - Spatial-Temporal complementary frames
  - Each block deliver multiple bits → boost data throughput
- ***HiLight***
  - Alpha channel
  - Binary Frequency Shift Keying (BFSK)
- ***TextureCode***
  - Temporal Embedding + Spatially Adaptive Embedding + Superpixels
- ***Hybrid***
  - ***TextureCode*** in high texture blocks, ***HiLight*** in plain texture blocks.

1. *"InFrame++: Achieve simultaneous screen-human viewing and hidden screen-camera communication"* – Mobisys 2015
2. *"Real-time Screen-camera communication behind any screen"* – MobiSys 2015

Comparison: Flicker perception

- Subjective assessment
- **TextureCode**, **HiLight** and **Hybrid**: no sign of flicker
- **InFrame++**
  - some residual flicker at 70cm distance, with block size 32x32
  - smaller block size (12x12) could help reduce flicker level, but the communication range reduces
- The design of **TextureCode** inherently reduces flicker
  - Block boundaries are aligned with edges
  - Block boundaries are separated

Comparison: Goodput



Hybrid improves goodput of TextureCode by 45% and HiLight by 125%

(block size = 32x32)

TextureCode - Summary

- Spatially adaptive encoding: more goodput, near-zero flicker to embedded screen-camera communication.
- Show potential to improve goodput of embedded screen-camera communication by combining multiple dimensions of embedding, up to 22kbps while remaining flicker-free.

Thank you!