Content Adaptive Encoding Method for High Frame Rate Screen-Camera Communication

M.S. Defense Presentation

Yaqin Tang

yaqin.tang@rutgers.edu

Advisor(s): Prof. Marco Gruteser, Prof. Kristin Dana, Prof. Narayan Mandayam

January 13th 2016

WINLAB, Rutgers University
Screen-camera Communication

Transmitter (screen)  
Signal modulation

Receiver (camera)  
Image analysis  
Signal demodulation
Motivations and Objectives

Key considerations:

User experience
  Invisibility

Communication performance
  Goodput (accurate bits per unit time)

Existing works

Our Goal

Flicker / distortion

noticeable → unnoticeable

Goodput (kbps)
Outline

• User experience: flicker perception factors
• System design
  - content-adaptive encoding method;
  - signal amplitude tracking decoding method;
• Prototype implementation
• System performance evaluation
• Conclusion and future work
Outline

• User experience: flicker perception factors
• System design
  - content-adaptive encoding method;
  - signal amplitude tracking decoding method;
• Prototype implementation
• System performance evaluation
• Conclusion and future work
Flicker Perception

**Definition:** apparent fluctuation and change in the brightness of the displaying surface.

**Affecting factors:**

- Frame rate
- Modulation amplitude
- Edge effect
- Viewer’s field of view
- Image content
Flicker Perception – frame rate

No obvious flicker if:
Brightness change frequency > 100 Hz.

* Note: this is a simplified experiment setup from D.H. Kelly, “Sine wave and flicker fusion”.
Flicker Perception – modulation amplitude

Signal amplitude experiment:
- $x$: original brightness;
- $\alpha$, $\beta$: alteration amplitude.

![Diagram](image-url)
Flicker Perception – modulation amplitude

Brute force method:
• check brightness from \((0 \sim 255) \pm (1 \sim 10)\).

<table>
<thead>
<tr>
<th>(\alpha, \beta)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>...</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>\star</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• \((+2 / -3)\) win for flicker perception and camera detection.
Flicker Perception – edge effect
viewer’s field of view

Observation:
• Along the edges $\rightarrow$ more obvious flicker.
• Smaller field of view in retina $\rightarrow$ less flicker.
  (combine viewing distance and display block size)

Gray block experiment
Flicker Perception – image texture

- Attribute representing spatial arrangement of gray levels of the pixels in a region of image.
- Texture regions give less flicker.
Flicker Perception – brightness and contrast

Image brightness:
- Visual perspective, color in R, G, B space or gray scale from 0 to 255 pixel intensity value.

Image contrast:
- Visual concept defined by the difference in the color and brightness of the object.

Both are minor factors!
Flicker Perception Factors

- **Frame rate** -> greater than 100 Hz.
- **Modulation amplitude** -> (+2 / -3) brightness alternation.
- **Edge effect** -> more flicker along edges.
- **Viewer’s field of view** -> smaller area in retina, less flicker
- **Image content** -> image texture (major factor)
  image brightness / contrast (minor)
Outline

- User experience: flicker perception factors
- System design
  - content-adaptive encoding method;
  - signal amplitude tracking decoding method;
- Prototype implementation
- System performance evaluation
- Conclusion and future work
Content Adaptive Encoding Method

Temporal domain encoding:

- Message bit-stream assigned to each frame;
- Image modulated as brightness change.

Consecutive same bits reduce displaying frequency!
Content Adaptive Encoding Method

Temporal domain encoding:
- Manchester code ensures minimum frequency at 60 fps.
- Bit 1, brightness increase 2; bit 0, decrease 3.

Original video (30fps)
Actual Message (60fps)
Displaying video (120fps)

1/120s

F1 1/30s F2 1/30s F3 1/30s F4
Content Adaptive Encoding Method

Spatial domain encoding:
- **Checkerboard** on top of each frame.
  - improve throughput;
  - decease field of view in retina.
- Texture and edges analysis.
Content Adaptive Encoding Method

Image texture analysis:
- Texture range filter.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
</tbody>
</table>

Original image matrix sample

Texture range value: $15 - 3 = 12$

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>7</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>7</td>
<td>5</td>
</tr>
</tbody>
</table>

Corresponding texture range value
Content Adaptive Encoding Method

Image texture analysis:
- Texture range filter.
- Choose a band pass filter to:
  - get texture region & avoid edges
Content Adaptive Encoding Method

Original image

Texture range matrix

Spatial domain embedding

Temporal domain embedding

Block analysis

Time ...
## Content Adaptive Encoding Method

### Flicker and goodput oriented:

<table>
<thead>
<tr>
<th>Our encoding method</th>
<th>Reasons</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Temporal encoding</strong></td>
<td>Frame rate</td>
</tr>
<tr>
<td>• Display at 120fps;</td>
<td>Modulation amplitude</td>
</tr>
<tr>
<td>• Brightness change +2, -3 with</td>
<td></td>
</tr>
<tr>
<td>Manchester code;</td>
<td></td>
</tr>
<tr>
<td><strong>Spatial encoding</strong></td>
<td>Viewer’s field of view</td>
</tr>
<tr>
<td>• Checkerboard size 32*32 pixel²;</td>
<td></td>
</tr>
<tr>
<td>• Texture range analysis and</td>
<td>Edge effect</td>
</tr>
<tr>
<td>bandpass filter.</td>
<td>Image texture</td>
</tr>
</tbody>
</table>
Outline

• User experience: flicker perception factors
• System design
  - content-adaptive encoding method;
  - signal amplitude tracking decoding method;
• Prototype implementation
• System performance evaluation
• Conclusion and future work
Decoding Method

Assumption, receiver knows:
• Starting frame of the message;
• Original video resolution;
• Checkerboard size.
• Encoded checkerboard patterns;

Algorithm:
• Track temporal brightness change, if
  - High to low → “bit 1”
  - Low to high → “bit 0”
Decoding Method

- **Captured frame**
  - Warped and cropped frame; Divide checkerboard pattern.
  - Frame 1

- **Captured frame**
  - Warped and cropped frame; Divide checkerboard pattern.
  - Frame 2

**Average brightness decode:**
- **Bit 1:** HI -> LO
- **Bit 0:** LO -> HI
Frame Alignment Issue

120 fps video on display

| 1 | 0 | 0 | 1 | 0 | ...

Synchronization issue if capturing at 120 fps

| 0.5 | 0 | 0.5 | 0.5 | ...

Sequence capturing at 240 fps

| 1 clean | tainted | 0 clean | tainted | 0 clean | tainted | 1 | 0 | ...

Odd set of the frames

| 1 | 0 | 0 | 0 | 1 | 0 | ...

Even set of the frames

| 0.5 | 0 | 0.5 | 0.5 | ...

Clean frames!
Frame Alignment Issue

Histogram method to extract clean frames.

![Histogram Graph](image)

- Larger peak distance

26
Encode Pattern Detection

Recap: - content-adaptive encoding
- selected blocks encoded message.
Algorithm: find large brightness change blocks.

\[
\begin{array}{ccccccc}
-1 & 0 & 0 & 1 & 0 & & \\
1 & 0 & -1 & -1 & 1 & & \\
0 & -1 & 1 & 1 & 0 & & \\
1 & 0 & 0 & 1 & 0 & & \\
1 & 0 & 0 & -1 & -1 & & \\
\end{array}
\begin{array}{ccccccc}
-1 & 1 & 1 & 0 & 1 & & \\
0 & 1 & -1 & -1 & 0 & & \\
1 & -1 & 0 & 0 & 1 & & \\
0 & 1 & 1 & 0 & 1 & & \\
0 & 1 & 1 & -1 & -1 & & \\
\end{array}
= \begin{array}{ccccccc}
0 & 1 & 1 & 1 & 1 & 1 & \\
1 & 1 & 0 & 0 & 1 & & \\
1 & 0 & 1 & 1 & 1 & & \\
1 & 1 & 1 & 1 & 1 & & \\
1 & 1 & 1 & 0 & 0 & & \\
\end{array}
\]

Frame 1 (subtract) frame 2 (equals) difference matrix (in absolute value)
Outline

• User experience: flicker perception factors
• System design
  - content-adaptive encoding method;
  - signal amplitude tracking decoding method;
• Prototype implementation
• System performance evaluation
• Conclusion and future work
Prototype Implementation

- 120fps video displayed using glvideoplayer.
- Iphone6 recording at 240 fps.

Video resolution 1280-by-720
Camera fixed auto-exposure; in Slo-Mo mode

Distance 70cm
Experiment Videos

- bigbuckbunny
- Bosphorus
- ReadySetGo
- ShakeNDry
- football
- highway
- walking
- YachtRide
- Jockey
- Mobile
Outline

• User experience: flicker perception factors
• System design
  - content-adaptive encoding method;
  - signal amplitude tracking decoding method;
• Prototype implementation
• System performance evaluation
• Conclusion and future work
Pattern Detection Evaluation

- Static scene, color videos
Sample Error Markers

**Red**
not encoded blocks as encoded;

**Green**
bite 0 blocks as not encoded;

**Blue**
bite 1 blocks as not encoded;

**Gray**
original encoded blocks without error.
Basic Decoding Algorithm Evaluation

• Static scene, color videos

Goodput: correct bits per unit time.
Basic Decoding Algorithm Evaluation

- Dynamic scene, color videos

Goodput: correct bits per unit time.

[Graph showing transmit rate, goodput, and BER for different video names: Bosphorus, Jockey, Mobile, ReadySetGo, ShakeNDry, YachtRide, bigbuckbunny, football, highway, walking.]
Conclusions and Future Works

• Explored factors contributing to flicker perception;
• Proposed content-adaptive encoding method to achieve flicker-free screen-camera communication as well as high communication capacity and accuracy;
• Identified reasons causing system error;

• Combine pattern detection to decoding algorithm;
• Applications like screen identification etc.
Thank you!