REAL-TIME MORSE CODE COMMUNICATION APP

Supervisor: Prof. LYU Rung Tsong Michael

Students: LUO Xin (1155026046)
          ZOU Lei (1155026057)
AGENDA

Introduction
Relative topics study
Design and Implementation
Experiments and testing
Conclusion
Project demo
MOTIVATION

Present Morse code apps in Android market
OBJECTIVES

• Encoding Morse code and playing it by flashlight;
• Decoding Morse code of light pattern;
• Allowing users to change transmission rates;
• Decoding messages with any transmission rate in some range;
• Bi-directional communication in the standard way;
• Template database in case of emergency, for example, SOS;
• Saving words or sentences used frequently to the template database.
AGENDA

Introduction
Relative topics study
Design and Implementation
Experiments and testing
Conclusion
Project demo
INTRODUCTION TO MORSE CODE

Coding rule

<table>
<thead>
<tr>
<th>Element</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dot (●)</td>
<td>1 time unit</td>
</tr>
<tr>
<td>Dash (        )</td>
<td>3 time units</td>
</tr>
<tr>
<td>Inter-element gap</td>
<td>1 time unit</td>
</tr>
<tr>
<td>Short gap between letters</td>
<td>3 time units</td>
</tr>
<tr>
<td>Medium gap between words</td>
<td>7 time units</td>
</tr>
</tbody>
</table>
INTRODUCTION OF MORSE CODE

Example:

F: •••
Di-di-dah-dit

Y: ——•
Dah-di-dah-dah

P: •—••
Di-dah-dah-dit.
INTRODUCTION TO OPENCV

In the past

Human eyes to see things
Image processed in our brain

Now

Machine eyes to see things
Real time image processed in machine
INTRODUCTION TO OPENCV

Open source library

- Computer Vision algorithms
- Machine learning algorithms

Usage

- Face detection
- Camera’s movements trace
- Human actions’ classification
INTRODUCTION TO OPENCV

Simple Android application

Face Detection
INTRODUCTION TO OPENCV

Simple **Android application**

Puzzle game
INTRODUCTION TO OPENCV

OpenCV and our App

• Detection part mainly depends on OpenCV

• Real time image

• High speed image processing
AGENDA

Introduction
Relative topics study
Design and Implementation
Experiments and testing
Conclusion
Project demo
ENCODING

<table>
<thead>
<tr>
<th>Index</th>
<th>Code[index]</th>
<th>Corresponding symbol</th>
<th>Index</th>
<th>Code[index]</th>
<th>Corresponding symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[1, 3]</td>
<td>A/a</td>
<td>27</td>
<td>[1, 3, 3, 3]</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>[3, 1, 1, 1]</td>
<td>B/b</td>
<td>28</td>
<td>[1, 3, 3, 3]</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>[3, 1, 3, 1]</td>
<td>C/c</td>
<td>29</td>
<td>[1, 1, 1, 3]</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>[3, 1, 1]</td>
<td>D/d</td>
<td>30</td>
<td>[1, 1, 1, 3]</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>[1]</td>
<td>E/e</td>
<td>31</td>
<td>[1, 1, 1, 1]</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>[1, 1, 3, 1]</td>
<td>F/f</td>
<td>32</td>
<td>[3, 1, 1, 1]</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>[3, 3, 1]</td>
<td>G/g</td>
<td>33</td>
<td>[3, 1, 1, 1]</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>[1, 1, 1, 1]</td>
<td>H/h</td>
<td>34</td>
<td>[3, 3, 1, 1]</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>[1, 1]</td>
<td>I/i</td>
<td>35</td>
<td>[3, 3, 3, 1]</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>[1, 3, 3, 3]</td>
<td>J/j</td>
<td>36</td>
<td>[1, 3, 1, 3, 1]</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>[3, 1, 3]</td>
<td>K/k</td>
<td>37</td>
<td>[3, 1, 1, 3, 3]</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>[1, 3, 1, 1]</td>
<td>L/l</td>
<td>38</td>
<td>[1, 1, 3, 1, 1]</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>[3, 3]</td>
<td>M/m</td>
<td>39</td>
<td>[1, 3, 3, 3, 1]</td>
<td>'</td>
</tr>
<tr>
<td>13</td>
<td>[3, 1]</td>
<td>N/n</td>
<td>40</td>
<td>[3, 1, 1, 1, 3]</td>
<td>!</td>
</tr>
<tr>
<td>14</td>
<td>[3, 3, 3]</td>
<td>O/o</td>
<td>41</td>
<td>[3, 1, 1, 3, 1]</td>
<td>/</td>
</tr>
<tr>
<td>15</td>
<td>[1, 3, 3, 1]</td>
<td>P/p</td>
<td>42</td>
<td>[3, 1, 3, 3, 1]</td>
<td>(</td>
</tr>
<tr>
<td>16</td>
<td>[3, 3, 1, 3]</td>
<td>Q/q</td>
<td>43</td>
<td>[3, 1, 3, 1, 3]</td>
<td>)</td>
</tr>
<tr>
<td>17</td>
<td>[1, 3, 1]</td>
<td>R/r</td>
<td>44</td>
<td>[1, 1, 1, 1, 1]</td>
<td>&amp;</td>
</tr>
<tr>
<td>18</td>
<td>[1, 1, 1]</td>
<td>S/s</td>
<td>45</td>
<td>[3, 3, 1, 1, 1]</td>
<td>:</td>
</tr>
<tr>
<td>20</td>
<td>[1, 1, 3]</td>
<td>U/u</td>
<td>47</td>
<td>[3, 1, 1, 1, 3]</td>
<td>=</td>
</tr>
<tr>
<td>21</td>
<td>[1, 1, 1, 3]</td>
<td>V/v</td>
<td>48</td>
<td>[1, 3, 1, 3, 1]</td>
<td>+</td>
</tr>
<tr>
<td>22</td>
<td>[1, 3, 3]</td>
<td>W/w</td>
<td>49</td>
<td>[3, 1, 1, 1, 3]</td>
<td>-</td>
</tr>
<tr>
<td>23</td>
<td>[3, 1, 1, 3]</td>
<td>X/x</td>
<td>50</td>
<td>[1, 1, 3, 1, 3]</td>
<td>-</td>
</tr>
<tr>
<td>24</td>
<td>[3, 1, 3, 3]</td>
<td>Y/y</td>
<td>51</td>
<td>[1, 3, 1, 1, 3]</td>
<td>&quot;</td>
</tr>
<tr>
<td>25</td>
<td>[3, 3, 1, 1]</td>
<td>Z/z</td>
<td>52</td>
<td>[1, 1, 1, 1, 3]</td>
<td>$</td>
</tr>
<tr>
<td>26</td>
<td>[3, 3, 3, 3]</td>
<td>0</td>
<td>53</td>
<td>[1, 3, 3, 1, 3]</td>
<td>@</td>
</tr>
</tbody>
</table>
cm[]: the array of message inputted.
len: length of cm[].
‘ ‘: space
index: the index of Morse code array.
Code[][]: The Morse code array.

Example: “A E”
Morse code: “· — ·”
cm[] = {‘A’, ‘ ‘, ‘E’}
len = 3

Index = code_index(cm[i])
i = 0
index = 0, code[0][] = {1, 3}
i = 1
index = 54
i = 2
index = 4, code[2][] = {1}
Open the camera

- Create the preview
- Get each frame of the real time image
- Convert frame to RGBA32

Set parameters for camera

- Keep the screen on
- Enable the view

Process frame values
• Draw rectangle in the image

**RGBA** of the input frame

\[
\begin{bmatrix}
m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\
m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\
m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} \\
m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{bmatrix}
\]

Get sub matrix

\[
\begin{bmatrix}
w_{11} & w_{12} & w_{13} \\
w_{21} & w_{22} & w_{23} \\
\vdots & \vdots & \vdots \\
w_{n1} & \vdots & \vdots \\
\end{bmatrix}
\]

**RGBA** of the **Rectangle**
DECODING — PROCESS FRAME VALUES

- Threshold value of each pixel’s Light ON/OFF
  
  Step1: Light fully fill the rectangle
  
  Step2: Sum of all elements in the rectangle
  
  Step3: Average value

Threshold value for each channel:

\[
T(R) = 210 \quad T(G) = 210 \quad T(B) = 210
\]

\[
V(R) > 210 \land V(G) > 210 \land V(B) > 210 \quad \Rightarrow \quad \text{Pixel is Light ON}
\]

\[
\frac{\text{Light ON pixels}}{\text{all the pixels in the rectangle}} > 10\% \quad \Rightarrow \quad \text{Rectangle is Light ON}
\]
• Duration of Light ON/OFF
DECODING — PROCESS FRAME VALUES

• Duration of Light ON/OFF

\[
\text{newTime} = \begin{cases} 
\text{dot} & 0.7 \times \text{dot} < \text{realTime} < 1.3 \times \text{dot} \\
3 \times \text{dot} & 2.5 \times \text{dot} < \text{realTime} < 3.5 \times \text{dot} \\
7 \times \text{dot} & 6.5 \times \text{dot} < \text{realTime} < 7.5 \times \text{dot}
\end{cases}
\]

• Decode

Make use of duration of Light ON/OFF
Match them to the Morse code pattern
AGENDA

Introduction
Relative topics study
Design and Implementation
Experiments and testing
Conclusion
Project demo
EXPERIMENTS AND TESTING — LIGHT ON/OFF

Actual Light OFF condition
EXPERIMENTS AND TESTING — LIGHT ON/OFF

Actual Light ON condition while...
EXPERIMENTS AND TESTING

Light ON duration testing
EXPERIMENTS AND TESTING

Symbol testing: 26 letters and numbers
EXPERIMENTS AND TESTING

Symbol testing: Punctuations

The code is: ;=+-= "$@ ,?! /()&c:
Minimum emitting rate --- determined by dot duration

Dot duration < 0.5 s  ===>  errors happened in decoding

**Rule:** emitting rate < receiving rate / 2

Suppose receiving frequency = FPS

Emitting rate = 1 / (Dot duration) Hz

The max(emitting rate) = FPS / 2.

Our camera FPS = 8 fps  \[\text{max(emitting rate)} = \frac{8}{2} \text{ Hz}\]

min(dot duration) = \[\frac{2}{8} \text{ s} = 0.25\text{s}\]
AGENDA

Introduction
Relative topics study
Design and Implementation
Experiments and testing
Conclusion
Project demo
CURRENT LIMITATIONS

- Separated apps
- Unchangeable transmission rate
- Nonautomatic decoding
- Disturbance of environmental light
- Low accuracy under high transmission rate.
- Unchangeable parameters of the environmental light, e.g. exposure value
DIFFICULTIES

- For the whole project:
  - Without any knowledge about Android programming and Java

- For the encoding part:
  - Flashlight control
  - Timing control

- For the decoding part:
  - No idea about OpenCV
  - RGBA values process
  - ON/OFF duration process
FUTURE DEVELOPMENT

- For the whole project:
  - Combination of the two apps
  - User Interface optimization
  - Accuracy improvement
  - Bi-directional communication in the standard way
FUTURE DEVELOPMENT

- For the encoding part:
  - Changeable transmission rate

- For the decoding part:
  - Longer distance decoding
  - Higher transmission rate decoding
  - Auto-detection
  - Changeable parameters, e.g. brightness, exposure
AGENDA

Introduction
Relative topics study
Design and Implementation
Experiments and testing
Conclusion
Project demo
Supervisor: Prof. LYU Rung Tsong Michael

Students: LUO Xin (1155026046)  
ZOU Lei (1155026057)