Department of Computer Science and Engineering
The Chinese University of Hong Kong

REAL-TIME MORSE CODE COMMUNICATION APP

FINAL YEAR PROJECT FALL2013 LYU1305

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

Element	Time
Dot (●)	1 time unit
Dash (▬)	3 time units
Inter-element gap	1 time unit
Short gap between letters	3 time units
Medium gap between words	7 time units

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

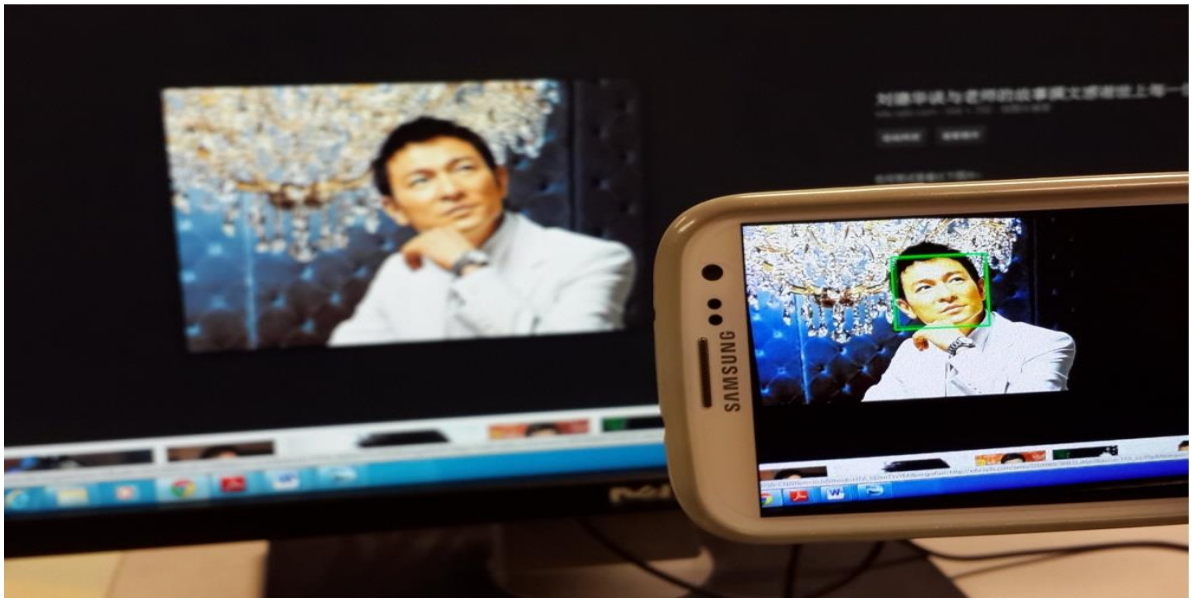
Usage

- Face detection
- Camera's movements trace
- Human actions' classification
- ●●●●●

INTRODUCTION TO OPENCV

Simple **Android application**

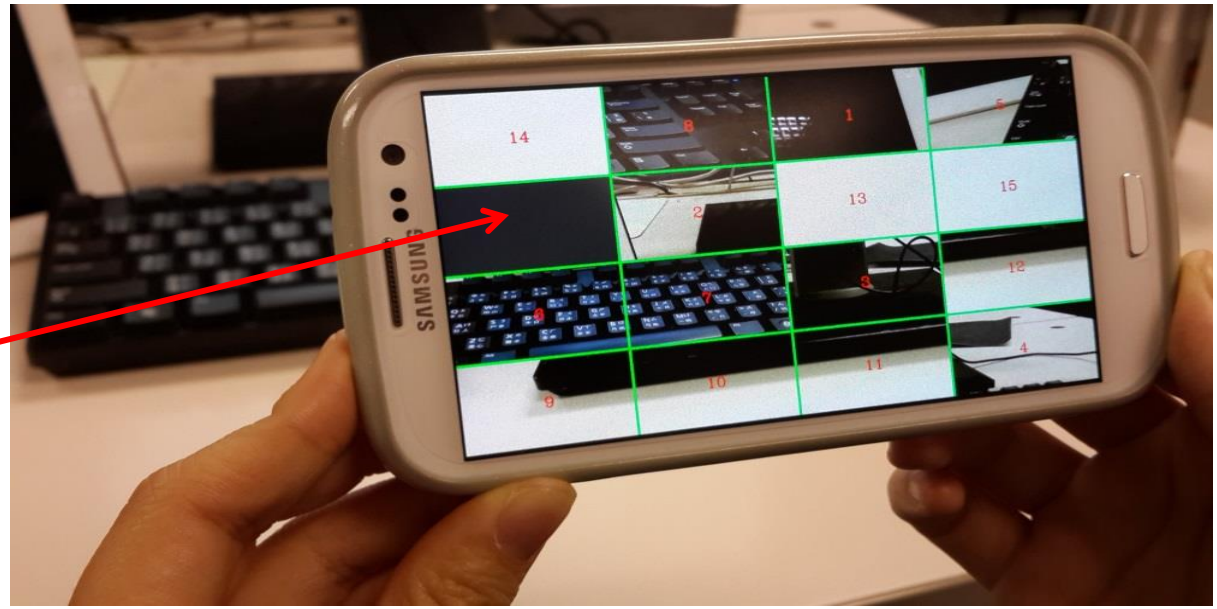
Face Detection



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

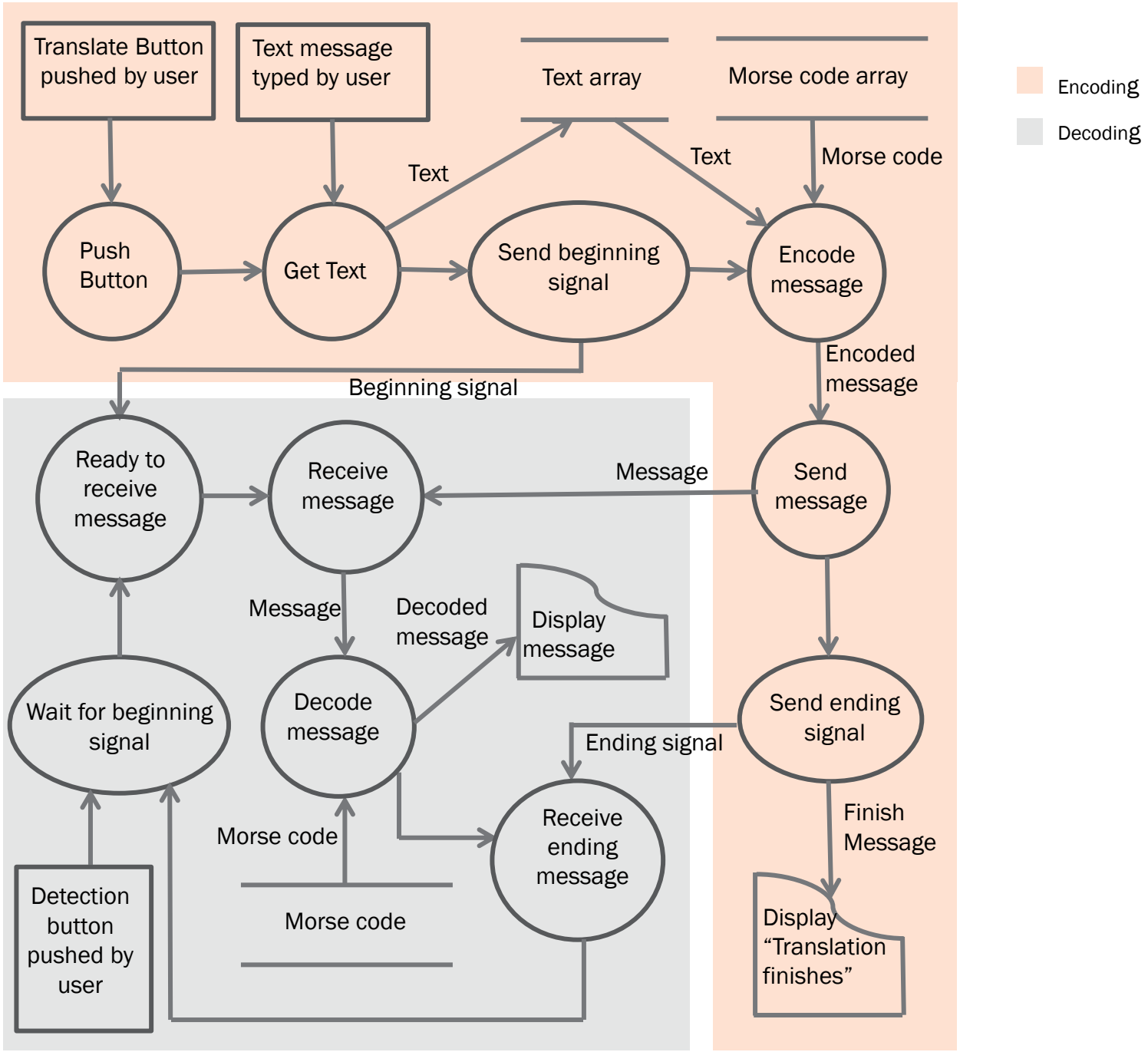
Relative topics study

Design and Implementation

Experiments and testing

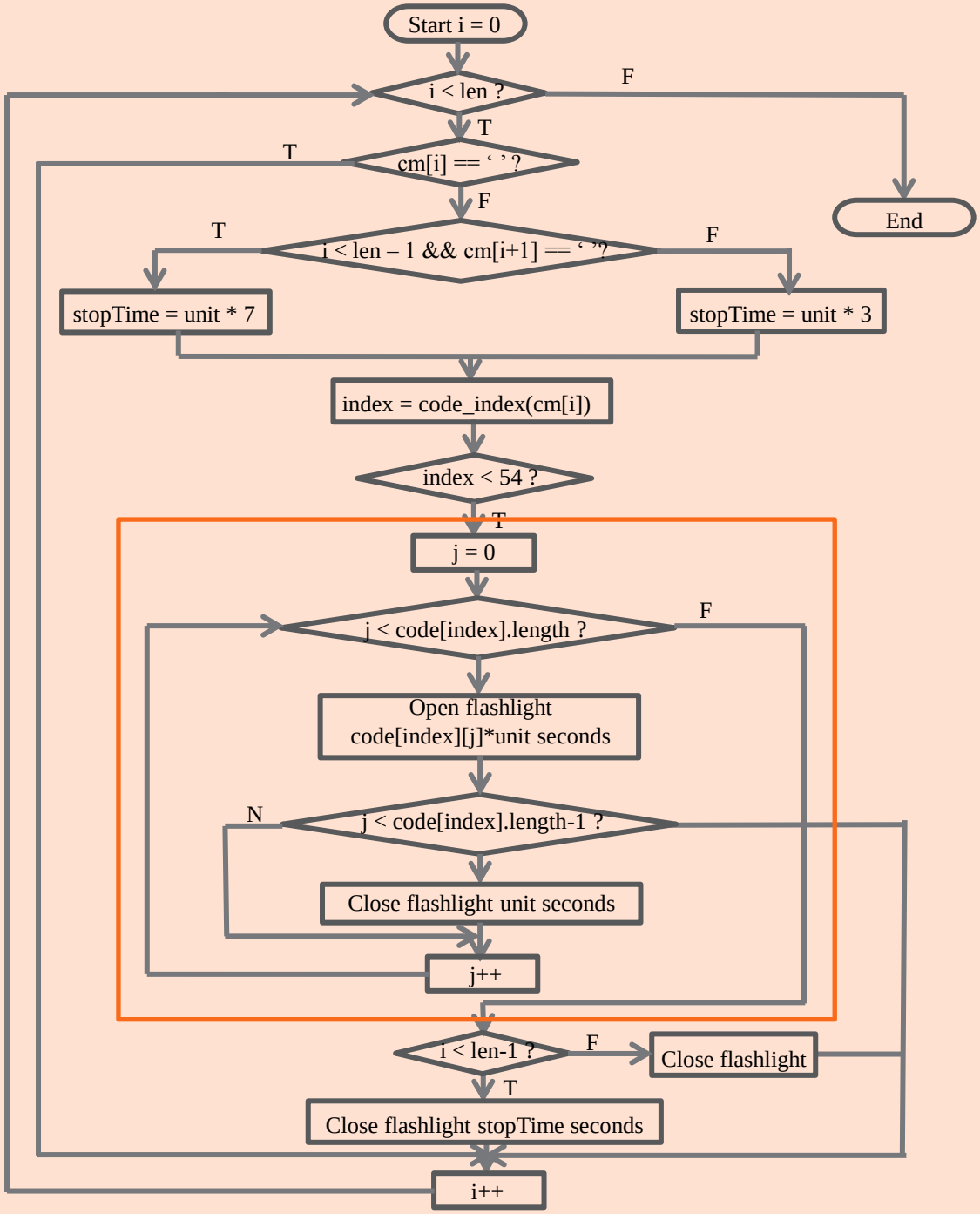
Conclusion

Project demo


Translate Button pushed by user
Text message typed by user
Text array
Morse code array
Encoding
Decoding
Text
Text
Morse code
Push Button
Get Text
Send beginning signal
Encode message
Encoded message
Beginning signal
Ready to receive message
Receive message
Message
Send message
Message
Decoded message
Display message
Wait for beginning signal
Decode message
Send ending signal
Ending signal
Morse code
Receive ending message
Finish Message
Detection button pushed by user
Morse code
Display “Translation finishes”

ENCODING

Index	Code[index]	Corresponding symbol	Index	Code[index]	Corresponding symbol
0	{1, 3}	A/a	27	{1, 3, 3, 3, 3}	1
1	{3, 1, 1, 1}	B/b	28	{1, 1, 3, 3, 3}	2
2	{3, 1, 3, 1}	C/c	29	{1, 1, 1, 3, 3}	3
3	{3, 1, 1}	D/d	30	{1, 1, 1, 1, 3}	4
4	{1}	E/e	31	{1, 1, 1, 1, 1}	5
5	{1, 1, 3, 1}	F/f	32	{3, 1, 1, 1, 1}	6
6	{3, 3, 1}	G/g	33	{3, 3, 1, 1, 1}	7
7	{1, 1, 1, 1}	H/h	34	{3, 3, 3, 1, 1}	8
8	{1, 1}	I/i	35	{3, 3, 3, 3, 1}	9
9	{1, 3, 3, 3}	J/j	36	{1, 3, 1, 3, 1, 3}	.
10	{3, 1, 3}	K/k	37	{3, 3, 1, 1, 3, 3}	,
11	{1, 3, 1, 1}	L/l	38	{1, 1, 3, 3, 1, 1}	?
12	{3, 3}	M/m	39	{1, 3, 3, 3, 3, 1}	’
13	{3, 1}	N/n	40	{3, 1, 3, 1, 3, 3}	!
14	{3, 3, 3}	O/o	41	{3, 1, 1, 3, 1}	/
15	{1, 3, 3, 1}	P/p	42	{3, 1, 3, 3, 1}	(
16	{3, 3, 1, 3}	Q/q	43	{3, 1, 3, 3, 1, 3}	)
17	{1, 3, 1}	R/r	44	{1, 3, 1, 1, 1}	&
18	{1, 1, 1}	S/s	45	{3, 3, 3, 1, 1, 1}	:
19	{3}	T/t	46	{3, 1, 3, 1, 3, 1}	;
20	{1, 1, 3}	U/u	47	{3, 1, 1, 1, 3}	=
21	{1, 1, 1, 3}	V/v	48	{1, 3, 1, 3, 1}	+
22	{1, 3, 3}	W/w	49	{3, 1, 1, 1, 1, 3}	-
23	{3, 1, 1, 3}	X/x	50	{1, 1, 3, 3, 1, 3}	_
24	{3, 1, 3, 3}	Y/y	51	{1, 3, 1, 1, 3, 1}	”
25	{3, 3, 1, 1}	Z/z	52	{1, 1, 1, 3, 1, 1, 3}	$
26	{3, 3, 3, 3, 3}	0	53	{1, 3, 3, 1, 3, 1}	@



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}

DECODING

Open the camera

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

Set parameters for camera

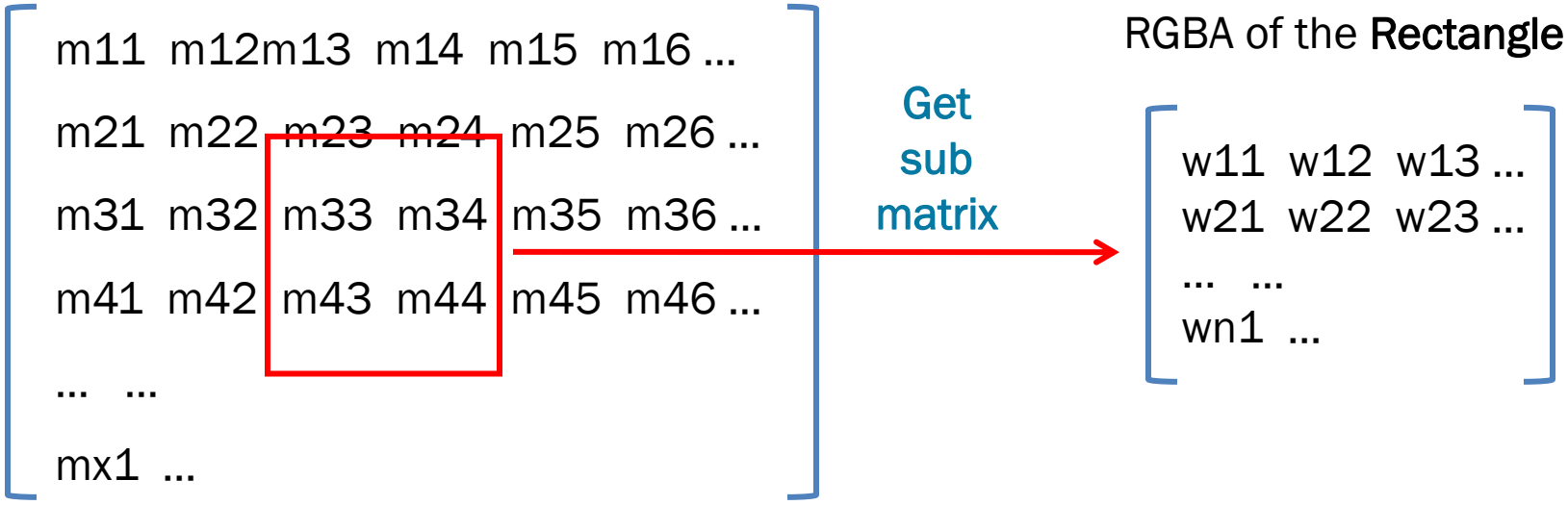
- Keep the screen on
- Enable the view

Process frame values

DECODING — PROCESS FRAME VALUES

- Draw rectangle in the image

RGBA of the input frame



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:

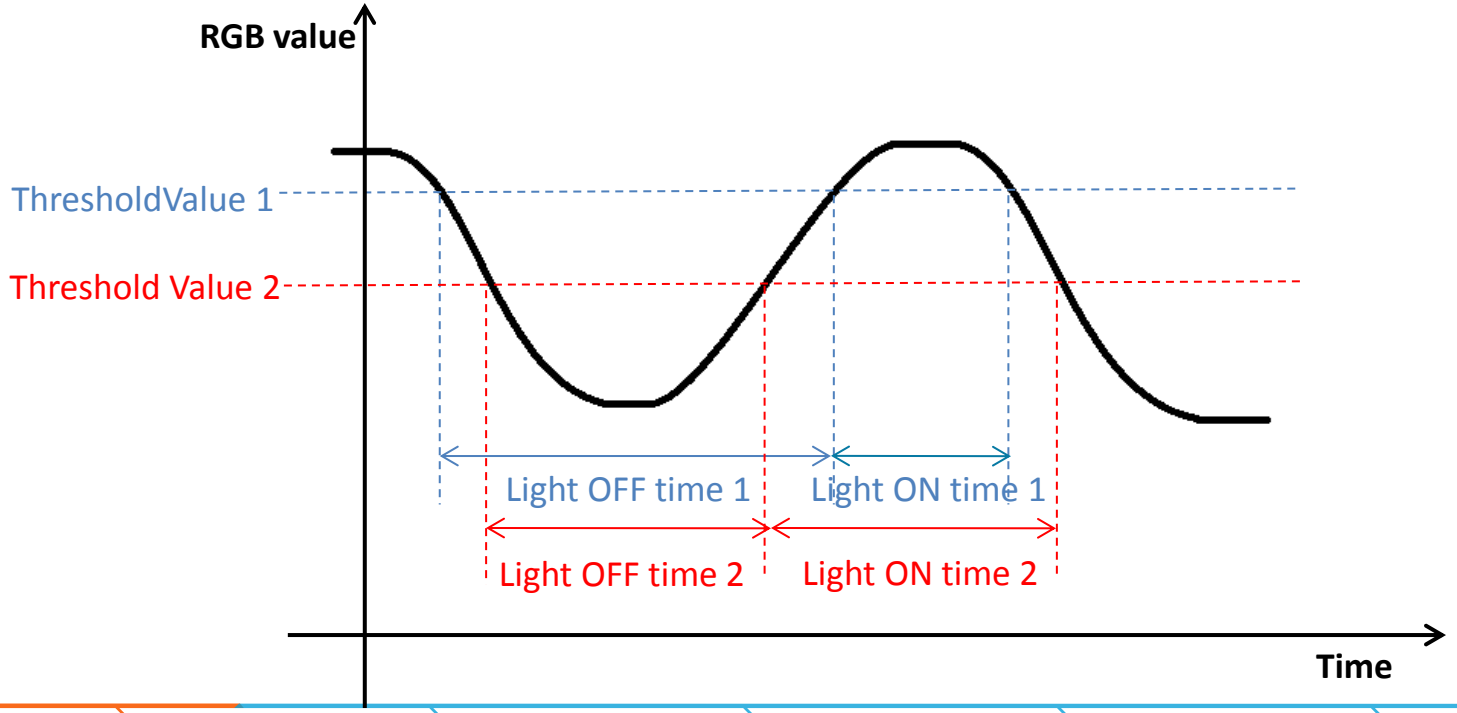
T(R) = 210 T(G) = 210 T(B) = 210

V(R) > 210 && V(G) > 210 && V(B) > 210 → Pixel is Light ON

(Light ON pixels) / (all the pixels in the rectangle) > 10% → Rectangle is Light ON

DECODING — PROCESS FRAME VALUES

- Duration of Light ON/OFF



DECODING — PROCESS FRAME VALUES

- Duration of Light ON/OFF

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

- Decode

  Make use of duration of Light ON/OFF

  Match them to the Morse code pattern

AGENDA

Introduction

Relative topics study

Design and Implementation

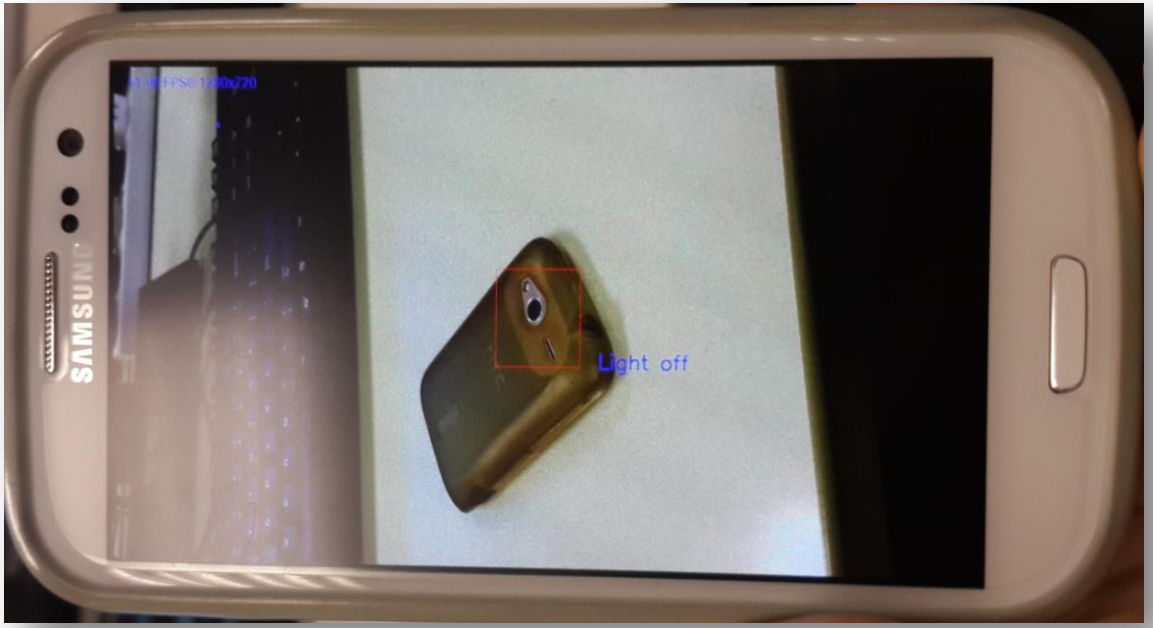
Experiments and testing

Conclusion

Project demo

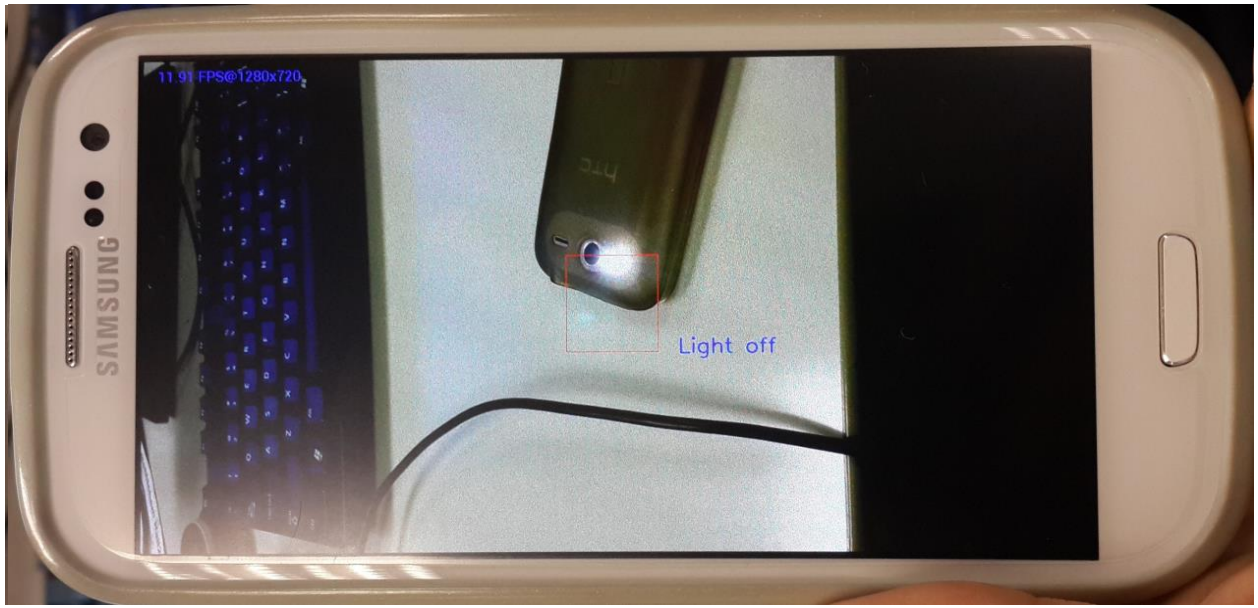
EXPERIMENTS AND TESTING — LIGHT ON/OFF

Actual Light OFF condition



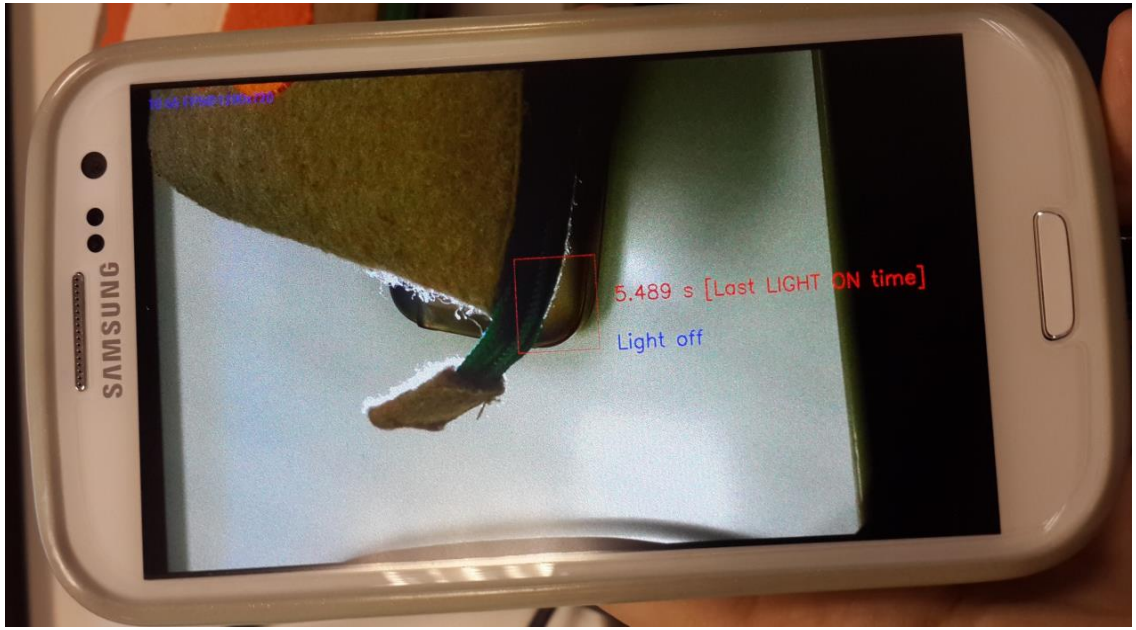
EXPERIMENTS AND TESTING — LIGHT ON/OFF

Actual Light ON condition while…



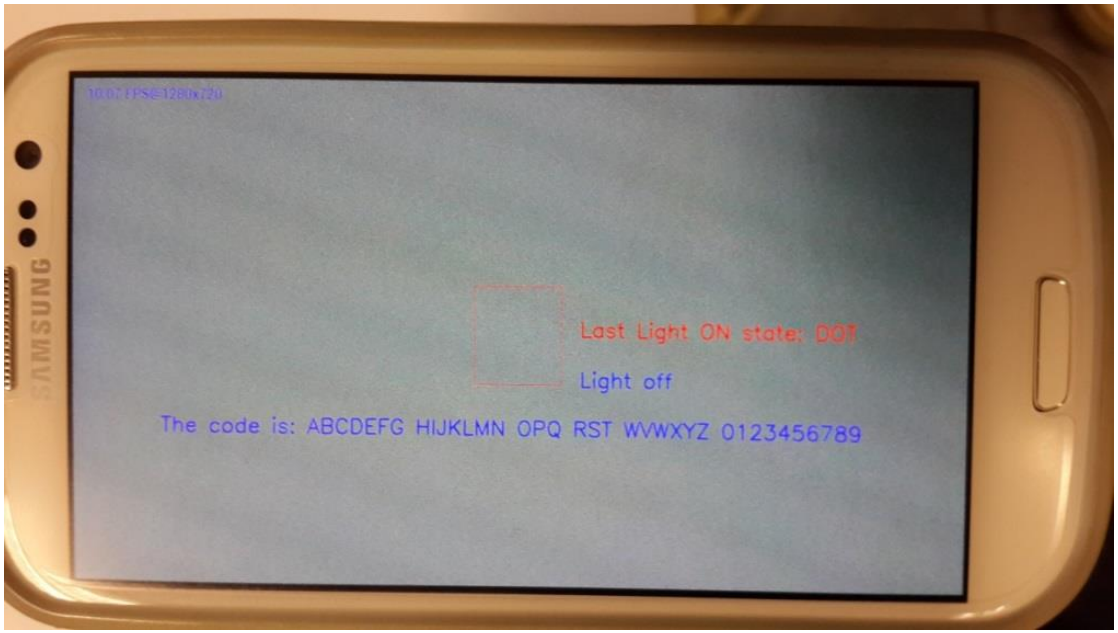
EXPERIMENTS AND TESTING

Light ON duration testing



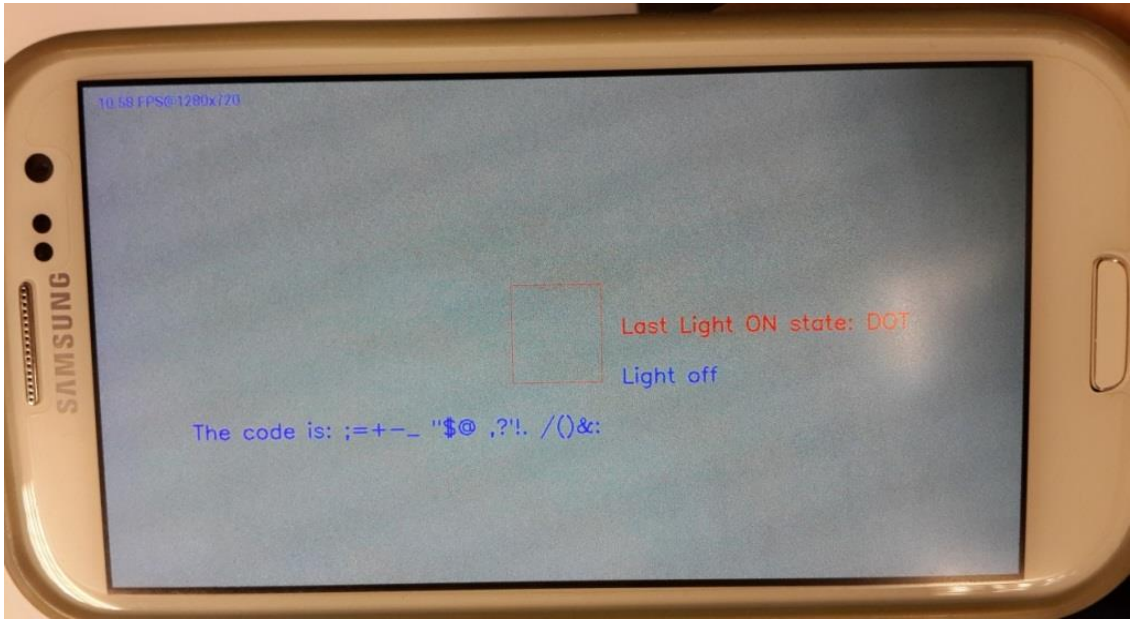
EXPERIMENTS AND TESTING

Symbol testing: 26 letters and numbers



EXPERIMENTS AND TESTING

Symbol testing: Punctuations



EXPERIMENTS AND TESTING

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 → max(emitting rate) = 8 / 2 Hz

min(dot duration) = 2 / 8 s = 0.25s

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

Department of Computer Science and Engineering
The Chinese University of Hong Kong

THANK YOU!

FINAL YEAR PROJECT FALL2013 LYU1305

Supervisor: Prof. LYU Rung Tsong Michael

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