F Y P

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

REAL-TIME MORSE CODE COMMUNICATION APP

FINAL YEAR PROJECT SPRING2014 LYU1305

Supervisor: Prof. LYU Rung Tsong Michael

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

AGENDA

Fall 2013 Review

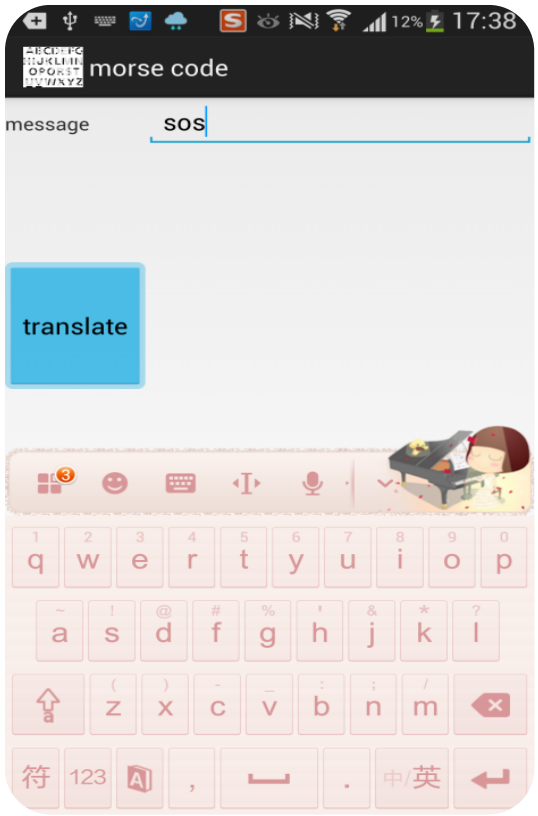
Spring 2014 Overview

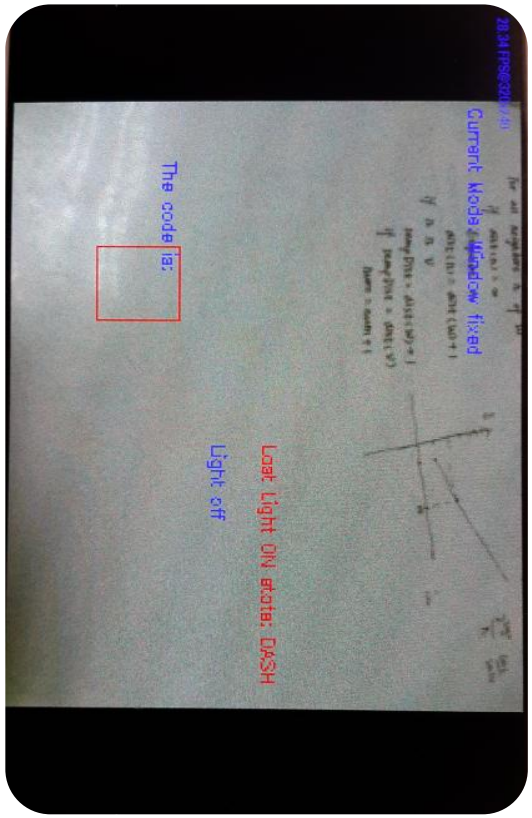
Design and Implementation

Project demo

Conclusion

FALL 2013 REVIEW

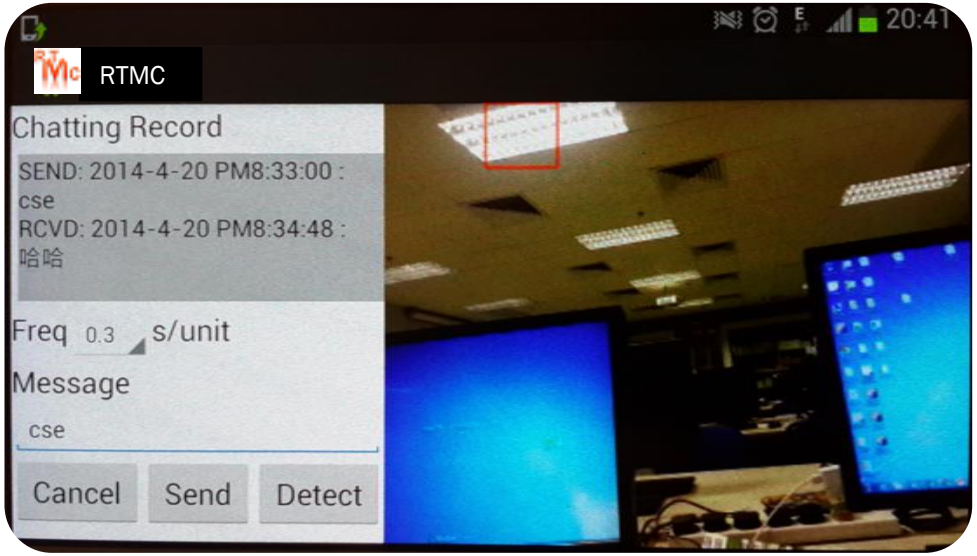




FALL 2013 REVIEW

- Separated encoding and decoding part
- Decoding with OpenCV
- Fixed Morse code frequency
- Fixed detection area

SPRING 2014 OVERVIEW



A complete application

Auto light source locating and tracking

Auto code frequency detection

Chinese supporting

IMPLEMENTATION

Apps Combination

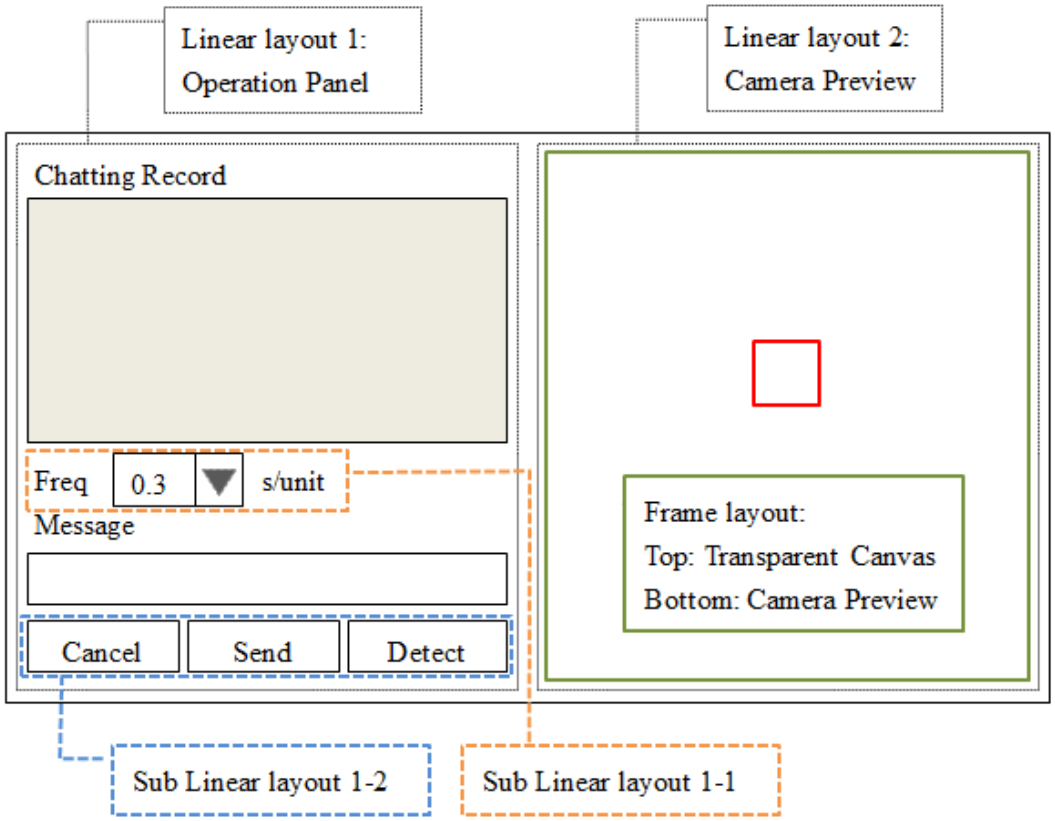
Camera Preview & Frame Buffer

Light Source Locating & Tracking

Auto Detection

Unicode Encoding & Decoding

APPS COMBINATION



APPS COMBINATION

Encoding

flash light

requires

Breakdown

Decoding

requires

APPS COMBINATION

Android Camera control

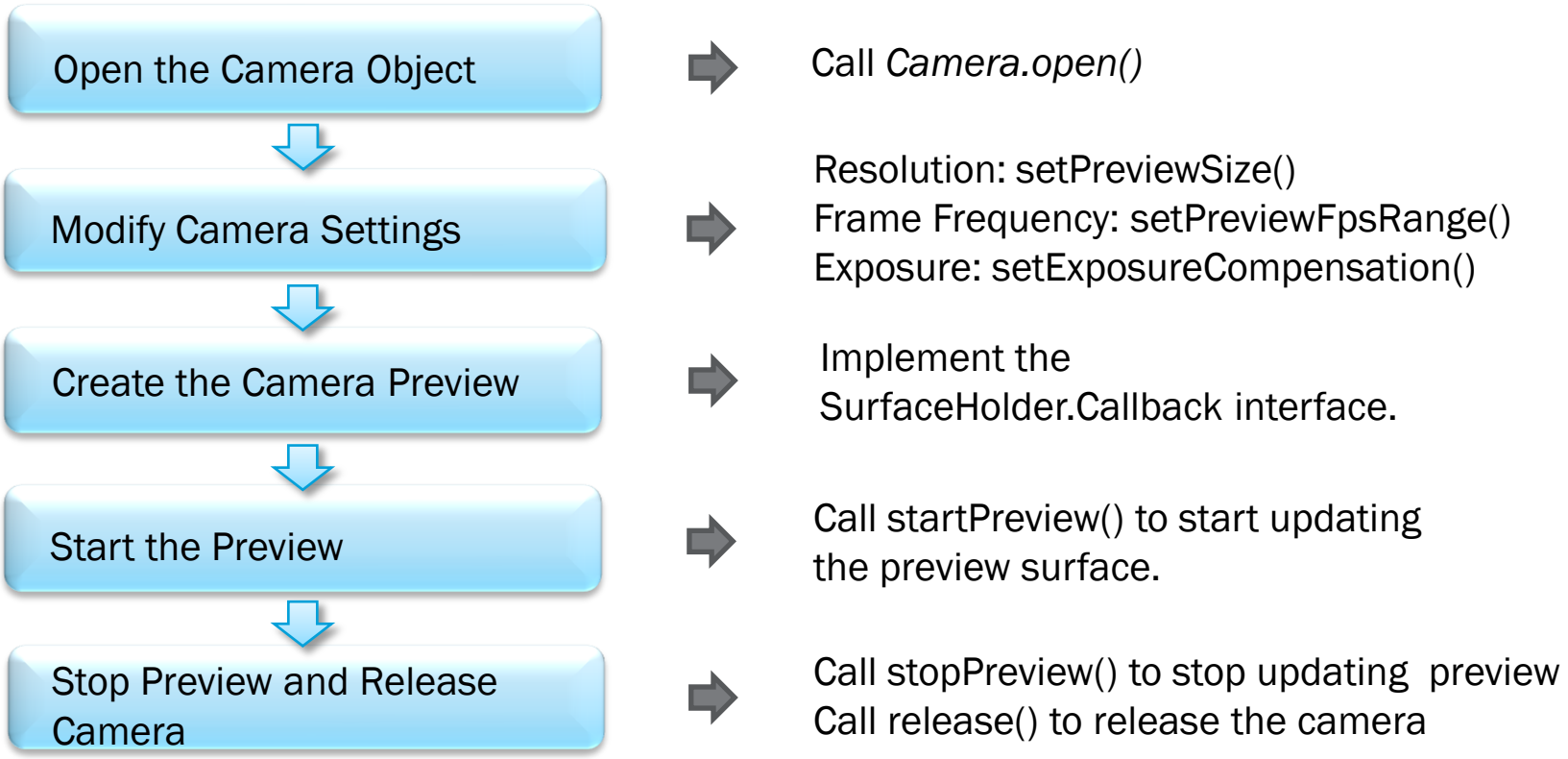
flash light

Encoding

Camera preview

Decoding

CAMERA PREVIEW



CAMERA FRAME BUFFER

YUV420sp format

Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
Y9	Y10	Y11	Y12	Y13	Y14	Y15	Y16
Y17	Y18	Y19	Y20	Y21	Y22	Y23	Y24
Y25	Y26	Y27	Y28	Y29	Y30	Y31	Y32
U1	V1	U2	V2	U3	V3	U4	V4
U5	V5	U6	V6	U7	V7	U8	V8

CAMERA FRAME BUFFER

Callback Function: onPreviewFrame(byte[] data, Camera camera)
Trigger: setOneShotPreviewCallback(MainActivity.this)

YUV420sp -> RGB:

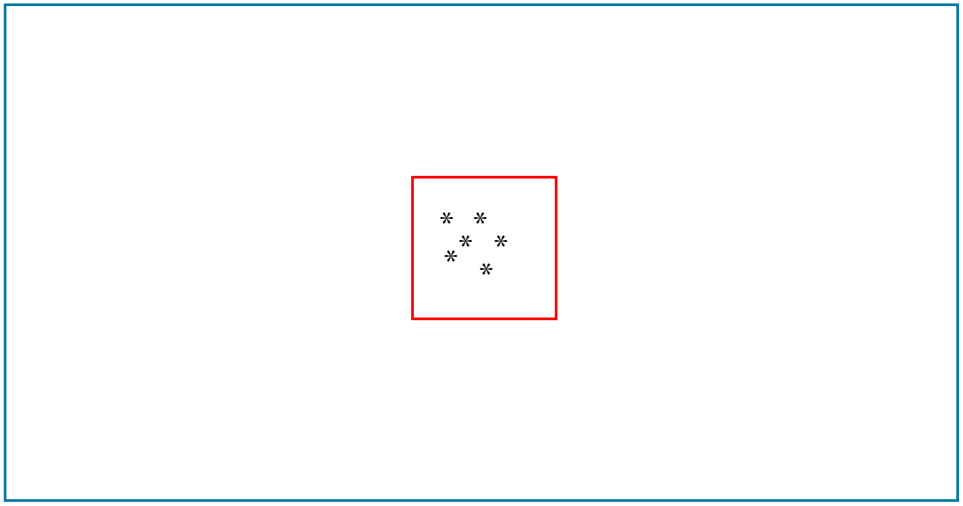
```
r = (1192 * y + 1634 * v);
g = (1192 * y - 400 * u - 833 * v );
b = (1192 * y + 2066 * u);

mRgb[i][j][0] = (int)(r >> 10);
mRgb[i][j][1] = (int)(g >> 10);
mRgb[i][j][2] = (int)(b >> 10);
```

LIGHT SOURCE LOCATING

Original version

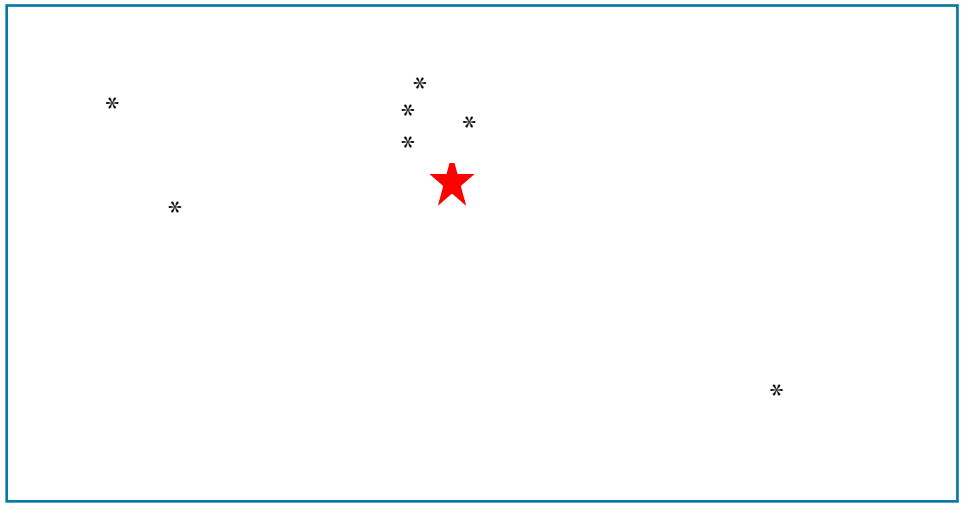
-----Depends on percentage of light ON pixels



LIGHT SOURCE LOCATING

2nd version

-----Finding light center and “cutting” the screen



LIGHT SOURCE LOCATING

2nd version

-----Finding light center and “cutting” the screen

Position of ★ is calculated by:

totalRGB = sum of all the pixels’ RGB

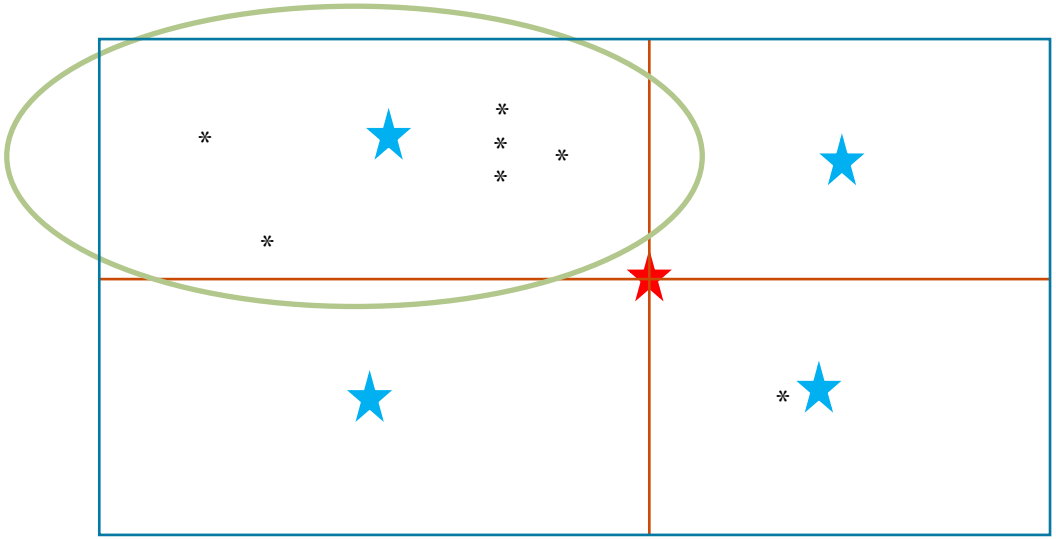
$$centerRow = \left(\sum_{r=0}^{totalRow} \sum_{c=0}^{totalCol} r * RGB(r,c)\right) / totalRGB$$

$$centerCol = \left(\sum_{r=0}^{totalRow} \sum_{c=0}^{totalCol} c * RGB(r,c)\right) / totalRGB$$

LIGHT SOURCE LOCATING

2nd version

-----Finding light center and “cutting” the screen

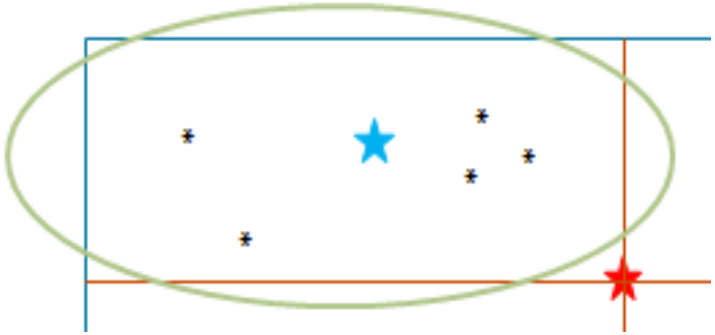


- Get the grid with the largest light center RGB value
- Repeat the previous process in this grid

LIGHT SOURCE LOCATING

2[illegible]sion

-----Finding light center and “cutting” the screen

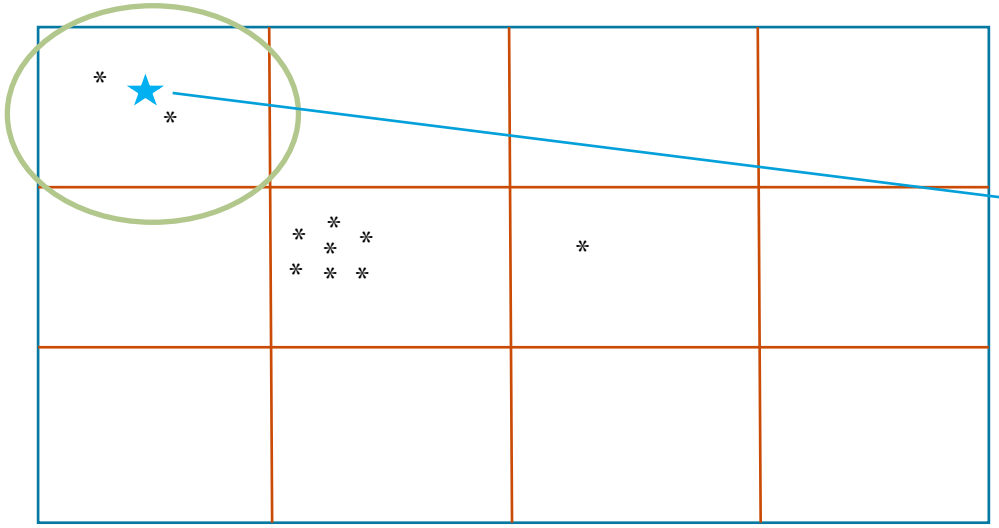


- Very time consuming
- The preview frame is not continuous

LIGHT SOURCE LOCATING

3rd version

-----Comparing Grids' light center's RGB value



(localCenterRow, localCenterCol)

LIGHT SOURCE LOCATING

3rd version

-----Comparing Grids' light center's RGB value

- Comparing each grid's light center's RGB value:

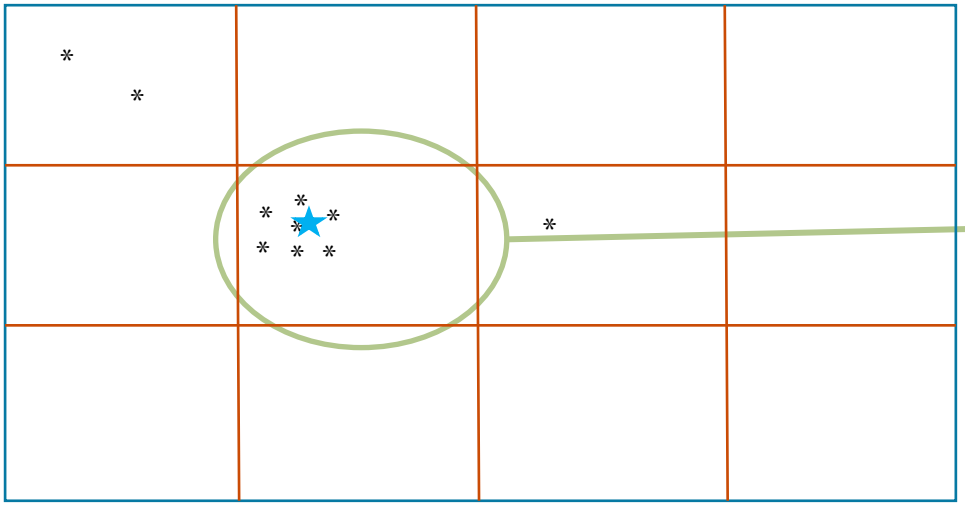
  RGB(localCenterRow, localCenterCol)

- Finding the grid with the largest light center RGB value

LIGHT SOURCE LOCATING

3rd version

-----Comparing Grids' light center's RGB value



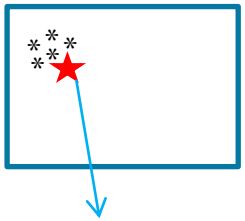
Set this grid to be the initial detection window

LIGHT SOURCE LOCATING

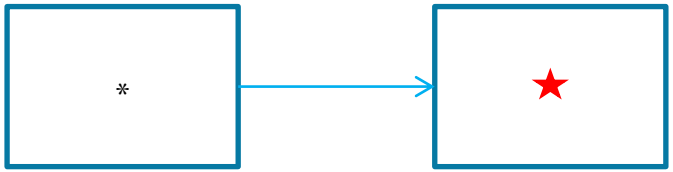
3[illegible]ion

-----Comparing Grids' light center's RGB value

However…If we have two grids like this:



Light center offset from the cluster of the light pixels

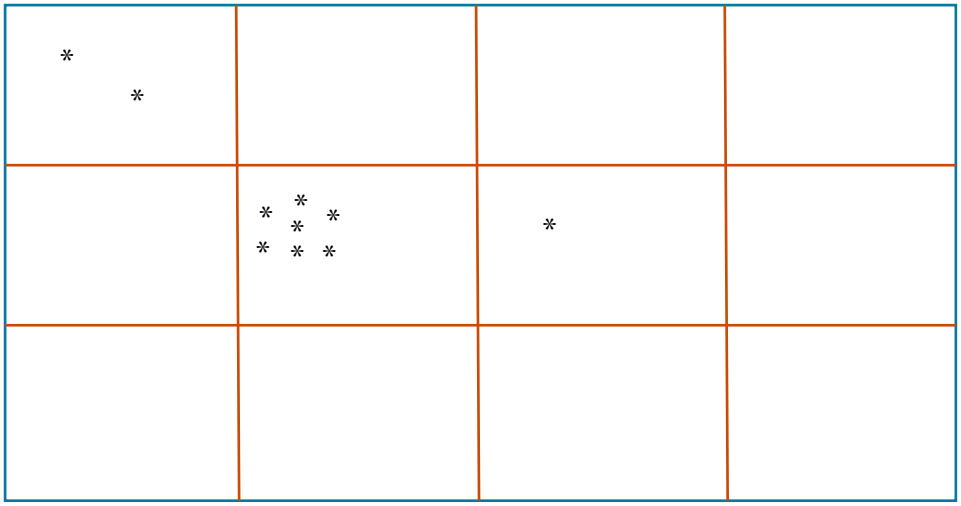


Light center accidently to be the light pixel

LIGHT SOURCE LOCATING

Final version

-----Finding local maximum RGB pixels and counting



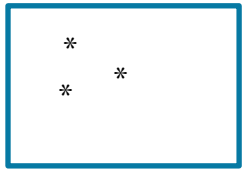
- Intra-Grid
- Inter-Grid

LIGHT SOURCE LOCATING

Final version

-----Finding local maximum RGB pixels, counting and comparing

- Intra-Grid



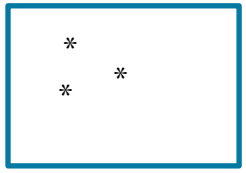
- Compare RGB value pixel by pixel
- Determine the local max RGB value
- Count the number of pixels with local max RGB value

LIGHT SOURCE LOCATING

Final version

-----Finding local maximum RGB pixels, counting and comparing

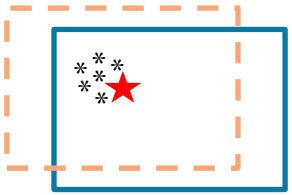
- Inter-Grid



- Compare two grid's local max RGB value
- If same, comparing number of those pixels
- Record as temp global max RGB value and grid
- If the last one's R+G+B == 765 & num >= 10, set it to be the initial detection window

LIGHT SOURCE TRACKING

After getting the initial detection window, we need to follow it in case that the camera shook accidently



- Consider the light center as the center to draw next tracking window
- Calculate the light center and draw the tracking window recursively

Problem: Cannot relocate the light source during decoding

Reason: Re-locating costs too much time ==> Preview frame is not continuous

AUTO DETECTION

Method: Made use of start signal

Sending part:

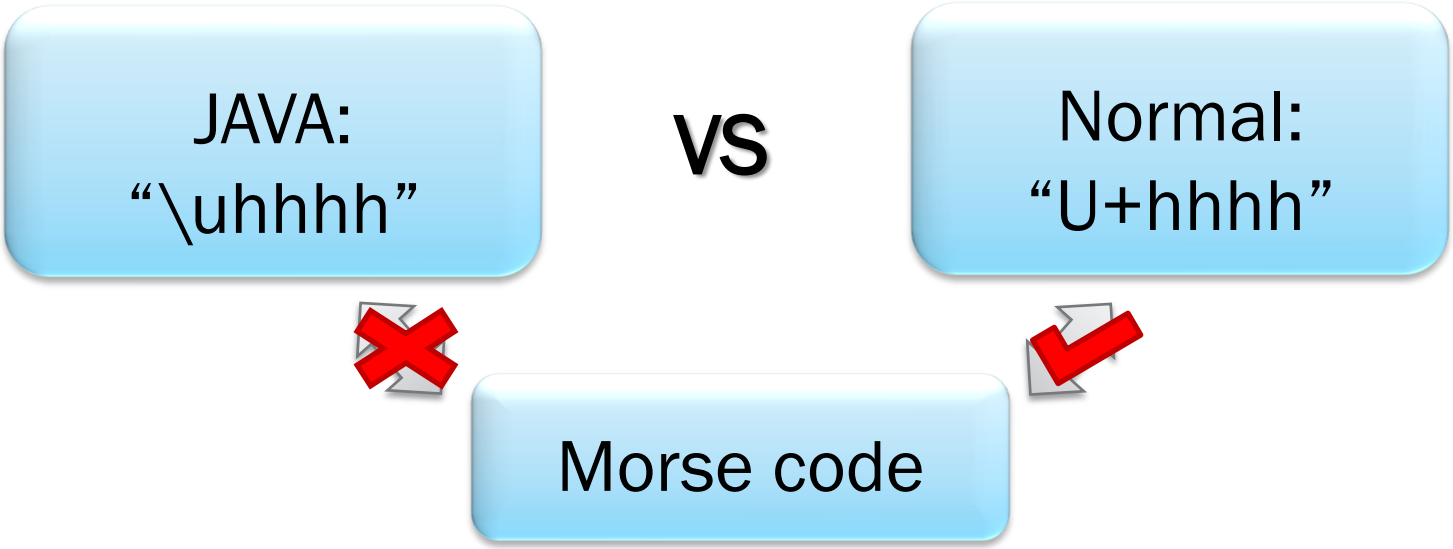
- Set start signal to be 10 times of the DOT duration

Receiving part

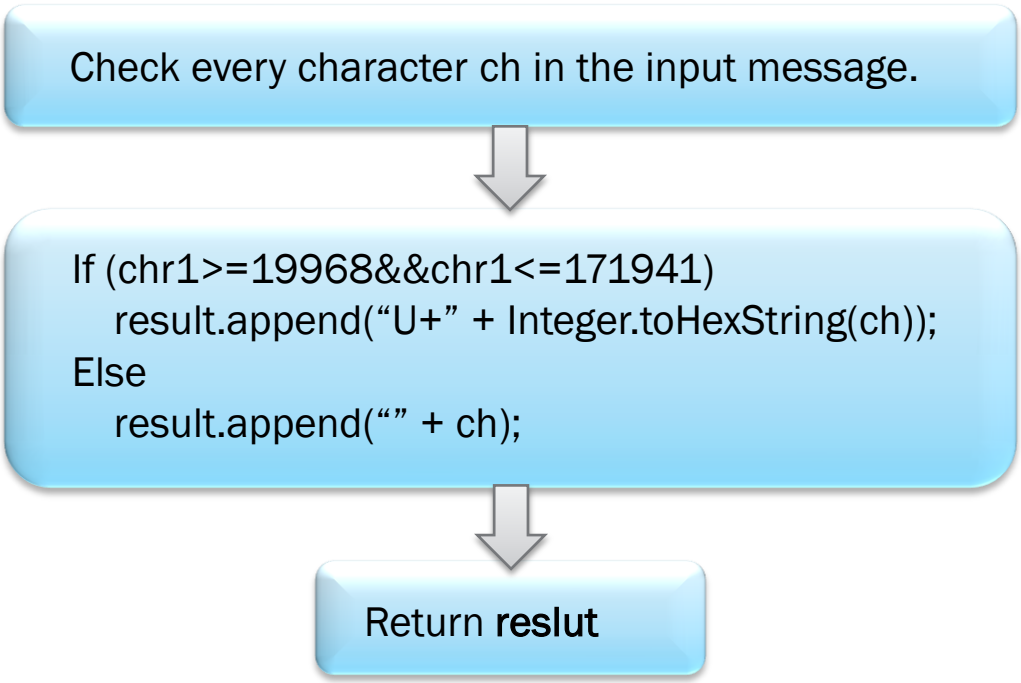
- Estimate the DOT duration according to the start signal length
- Decode the pattern according to the DOT duration

UNICODE ENCODING & DECODING

Unicode Representation:
Chinese Unicode ranges from U+4E00 to U+9FA5 (19968 – 17194).



UNICODE ENCODING



I love 中大

I love U+4e2dU+5927

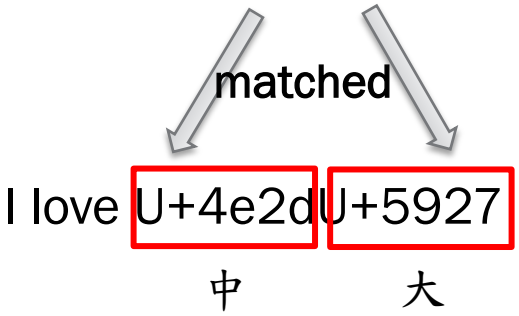
UNICODE DECODING

Define patter “U+hhhh” by “(U\\+(\\p{XDigit}{4}))”

While find the matched pattern in input string str
convert “hhhh” to Chinese character
replace “U+hhhh” to the Chinese character

Return str

Pattern: “U+hhhh”



I love 中大

DEMO

DEMO

Light Source Locating & Tracking

Transmission canceled & Invalid signal detection

Chinese Supporting

Bi-directional Communication

CONCLUSION

Improvement in Spring 2014

Limitations in Spring 2014

Summary in the whole year

IMPROVEMENT IN SPRING 2014

Limitations in Fall 2013:

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

LIMITATIONS IN SPRING 2014

- Disturbance of environmental light.
- Low accuracy under transmission frequency < 0.3s/unit.
- Cannot relocate the light source during decoding.
- Cannot determine whether the pattern “U+hhhh” is a Chinese character or not.

SUMMARY IN THE WHOLE YEAR

Real-time Morse code communication:

- Bi-directional communication
- Auto light source locating and tracking
- Auto code frequency detection
- Chinese supporting

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)