

ภาคผนวก

มหาวิทยาลัยราชภัฏสุราษฎร์ธานี

ภาคผนวก ก

รายละเอียดโปรแกรม

มหาวิทยาลัยราชภัฏสุราษฎร์ธานี

### 1) code main program

```
int main(int argc, char* argv[]){

    int red1;
    int green1;
    int blue1;
    int red2;
    int green2;
    int blue2;

    printf("Enter red1: ");
    // red1 = getch();
    //scanf("%d", red1);
    scanf("%d", &red1);

    printf("Enter green1: ");
    scanf("%d", &green1);
    printf("Enter blue1: ");
    scanf("%d", &blue1);
    printf("Enter red2: ");
    scanf("%d", &red2);
    printf("Enter green2: ");
    scanf("%d", &green2);
    printf("Enter blue2: ");
    scanf("%d", &blue2);
```

```
CvSize size640x480 = cvSize(640,480);
CvCapture* p_webcam ;
IplImage* p_origin ;
IplImage* p_imgnew;

CvMemStorage* P_storage;
CvSeq* P_seqcircles;
int j;
float * P_xyz ;

double*Arrayx;
int iSizex =10;
int i;
char EscKey ;

p_webcam =cvCaptureFromCAM(0);

if(p_webcam == NULL) {
    printf(" check the wbcam connection \n");
    getchar();
    return(-1);
}

cvNamedWindow ("origin",CV_WINDOW_AUTOSIZE);
cvNamedWindow ("process",CV_WINDOW_AUTOSIZE);

p_imgnew = cvCreateImage(size640x480,IPL_DEPTH_8U,1);
```

```
while(1){

    p_origin = cvQueryFrame(p_webcam);

    if( p_origin== NULL){
        std::cout<<"error:frame is not take \n";
        getchar();
        break;
    }

    cvInRangeS(
p_origin,CV_RGB(red1,green1,blue1),CV_RGB(red2,green2,blue2),p_imgnew);

    // cvInRangeS(
p_origin,CV_RGB(200,150,50),CV_RGB(243,182,75),p_imgnew);

    P_storage = cvCreateMemStorage(0);

    cvSmooth(p_imgnew,p_imgnew,CV_GAUSSIAN,9,9);

    P_seqcircles = cvHoughCircles(p_imgnew,
P_storage,CV_HOUGH_GRADIENT,2,p_imgnew->height/4,100,50,10,400);

    for(i=0;i<P_seqcircles->total;i++)
    {

        P_xyz=(float*)cvGetSeqElem(P_seqcircles,i);
```

```

printf("picture test x=%f ,Y=%f, Redien =%f\n",ceil(
P_xyz[0]/10),ceil(P_xyz[1]/10), P_xyz[2]);

cvCircle ( p_origin,
cvPoint(cvRound(P_xyz[0]),cvRound(P_xyz[1])),3,CV_RGB(0,255,0),CV_FILLED);
cvCircle ( p_origin,
cvPoint(cvRound(P_xyz[0]),cvRound(P_xyz[1]),cvRound(P_xyz[2]),CV_RGB(0,255,0),3);

char data;
//printf("enter char");
// data = getchar();
mkotitkenleng.startDevice("COM3",9600);
// printf("begin start ");
mkotitkenleng.send_data('+');
//mkotitkenleng.send_data(data);
// printf("sent");
mkotitkenleng.stopDevice();
// getchar();

}

cvShowImage("origin", p_origin);
cvShowImage("process",p_imgnew);
cvReleaseMemStorage(&P_storage);
EscKey=cvWaitKey(10);
}

```

```

    cvReleaseCapture(&p_webcam);
    cvDestroyWindow("original");
    cvDestroyWindow("process");
    return(0);
}

```

## 2 ) ตัวอย่าง code การดึงเอา ภาพ จากกล้อง webcam ข้างอิง จาก opencv

```

#include <cv.h>
#include <highgui.h>

int main()
{
    IplImage *im;
    CvCapture *cam = cvCaptureFromCAM(0);
    while(1)
    {
        im = cvQueryFrame(cam);
        cvShowImage(im);
        if(cvWaitKey(10) == 27)
            break;
    }
    cvReleaseImage(&im);
    cvReleaseCapture(&cam);
    cvDestroyAllWindows();
}

```

### 3) โปรแกรม ควบคุม มอเตอร์ 2 ตัว จาก Aduino Uno

```

/*
 * Brushed_H_Bridge_simple2 sketch
 * commands from serial port control motor direction
 * + or - set the direction, any other key stops the motors
 */

const int in1Pin = 5; // H-Bridge input pins
const int in2Pin = 4;
const int in3Pin = 3; // H-Bridge pins for second motor
const int in4Pin = 2;

void setup()
{
  Serial.begin(9600);
  pinMode(in1Pin, OUTPUT);
  pinMode(in2Pin, OUTPUT);
  pinMode(in3Pin, OUTPUT);
  pinMode(in4Pin, OUTPUT);
  Serial.println("+ - sets direction of motors, any other key stops motors");
}

void loop()
{
  if ( Serial.available() ) {
    char ch = Serial.read();
    if (ch == '+')
    {
      Serial.println("CW");
      // first motor
      digitalWrite(in1Pin,LOW);
      digitalWrite(in2Pin,HIGH);
    }
  }
}

```



```
//second motor
digitalWrite(in3Pin,LOW);
digitalWrite(in4Pin,HIGH);

}
else if (ch == '-')
{
  Serial.println("CCW");
  digitalWrite(in1Pin,HIGH);
  digitalWrite(in2Pin,LOW);
  digitalWrite(in3Pin,HIGH);
  digitalWrite(in4Pin,LOW);
}
else
{
  Serial.print("Stop motors");
  digitalWrite(in1Pin,LOW);
  digitalWrite(in2Pin,LOW);
  digitalWrite(in3Pin,LOW);
  digitalWrite(in4Pin,LOW);
}
}
}
```

#### 4) ตัวอย่าง code ที่ใช้ วิดีโอ เปลี่ยนเป็น เมทริก อ้างอิง จาก opencv

```

#include <vector>
#include <stdio.h>
#include <opencv2/opencv.hpp>

using namespace cv;
using namespace std ;

void createAlphaMat(Mat &mat)
{
    CV_Assert(mat.channels() == 4);
    for (int i = 0; i < mat.rows; ++i) {
        for (int j = 0; j < mat.cols; ++j) {
            Vec4b& bgra = mat.at<Vec4b>(i, j);
            bgra[0] = UCHAR_MAX; // Blue
            bgra[1] = saturate_cast<uchar>((float (mat.cols - j)) / ((float)mat.cols) *
UCHAR_MAX); // Green
            bgra[2] = saturate_cast<uchar>((float (mat.rows - i)) / ((float)mat.rows) *
UCHAR_MAX); // Red
            bgra[3] = saturate_cast<uchar>(0.5 * (bgra[1] + bgra[2])); // Alpha
        }
    }
}

int main(int argv, char **argc)
{
    // Create mat with alpha channel
    Mat mat(480, 640, CV_8UC4);
    createAlphaMat(mat);
}

```

```

vector<int> compression_params;
compression_params.push_back(CV_IMWRITE_PNG_COMPRESSION);
compression_params.push_back(9);

try {
    imwrite("alpha.png", mat, compression_params);
}
catch (runtime_error& ex) {
    fprintf(stderr, "Exception converting image to PNG format: %s\n", ex.what());
    return 1;
}

fprintf(stdout, "Saved PNG file with alpha data.\n");
return 0;
}

```

### 5 ) Code โปรแกรม inrang ซีพียูโปรแกรมแมชชีน

```

//-----inRange_gpu.cu-----start-----
#include "inRange_gpu.h"

__global__ void inRange_kernel(const cv::cuda::PtrStepSz<uchar3> src,
cv::cuda::PtrStepSz<b> dst,
int lbc0, int ubc0, int lbc1, int ubc1, int lbc2, int ubc2)
{
    int x = blockIdx.x * blockDim.x + threadIdx.x;
    int y = blockIdx.y * blockDim.y + threadIdx.y;

    if (x >= src.cols || y >= src.rows) return;

```

```

uchar3 v = src(y, x);
if (v.x >= lbc0 && v.x <= ubc0 && v.y >= lbc1 && v.y <= ubc1 && v.z >= lbc2 &&
v.z <= ubc2)
    dst(y, x) = 255;
else
    dst(y, x) = 0;
}

```

```

void inRange_gpu(cv::cuda::GpuMat &src, cv::Scalar &lowerb, cv::Scalar &upperb,
                cv::cuda::GpuMat &dst) {
    const int m = 32;
    int numRows = src.rows, numCols = src.cols;
    if (numRows == 0 || numCols == 0) return;
    // Attention! Cols Vs. Rows are reversed
    const dim3 gridSize(ceil((float)numCols / m), ceil((float)numRows / m), 1);
    const dim3 blockSize(m, m, 1);

    inRange_kernel<<<gridSize, blockSize>>>(src, dst, lowerb[0], upperb[0], lowerb[1],
upperb[1],
                lowerb[2], upperb[2]);
}

```

```
//-----inRange_gpu.cu-----end-----
```

```
////////////////////////////////////
```

```
//-----inRange_gpu.h-----
```

```
#ifndef __TEST__
```

```
#define __TEST__
```

```
#ifndef SKIP_INCLUDES
#include <vector>
#include <memory>
#include <iosfwd>
#include <iostream>
#endif

#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/core/core.hpp"
#include "opencv2/core/cuda.hpp"
#include "opencv2/videoio.hpp"
#include "opencv2/cudafilters.hpp"
#include "opencv2/cudaimgproc.hpp"
#include "opencv2/cudaarithm.hpp"

void inRange_gpu(cv::cuda::GpuMat &src, cv::Scalar &lowerb, cv::Scalar &upperb,
                cv::cuda::GpuMat &dst);

#endif /* __TEST__ */

//-----inRange_gpu.h-----end-----
```