Thermal Expert Android DEV Kit

Android Test Program

Source code Example

i3system, Inc.

2018.05.30

1. ThermalExpert Class

The class to run the camera, Thermal Expert.

Public Member Functions

1: ThermalExpert(ThermalExpertListener listener)

Constructor

2: void Initial(Context context)

Initialize the member variables in the class

3: void StartReceive()

Start to receive images

4: void StopReceive()

Stop receiving images

5: Bitmap GetImage()

Get a bitmap of a corrected image

6: short[] GetData()

Get a raw data

7: void SetColorMap(int mode)

Set color map(Gray, Iron, Rainbow, PurpleYellow)

8: void CalibrationImage()

Do Calibration

9: int[] GetARGBDate()

Get a ARGB 384 * 288 Data

10: byte[] GetYUV420()

Get a YUV420 384 * 288 Data

11: byte[] GetPointTemperature(int x, int y)

Get the temperature of the point with input(x, y);

12: byte[] GetPointTemperature(Point pt)

Get the temperature of the point

13: double[] GetRectTemperature(Rect rect)

Get the average, minimum, maximum temperature in the rectangle

14: void SetUserLevelSpan(Boolean IsLevelSpan)

Set the value to determine if level span function for user turns on

15. void SetLevelSpanMin(double min)

Set the minimum value for level span

16. void SetLevelSpanMax(double max)
    Set the maximum value for level span
17. double GetAGCLowTemp()
    Get the Minimum temperature in the image
18. double GetAGCHighTemp()
    Get the Maximum temperature in the image
19. byte[] GetGrayData()
    Get gray data in the image
20. void setUserDeadUpdate()
    Set dead upate in the image

1. ThermalExpert Class

1.1. ThermalExpert (ThermalExpertListener listener)

Construction

Parameters

listener : The variable of the class, ThermalExpertListener

Returns

Remarks

This function is constructor of the class, ThermalExeprt.

Example

```
ThermalExpert thermalExpert = new ThermalExert(thermalExpertListener);
```

1. ThermalExpert Class

1.2. void Initial (Context Context)

Initialize the class of ThermalExpert.

Returns

Remarks

This function should be called after the variable of ThermalExpert class is made

Example

```
thermalExpert.Initial(getApplicationContext());
```

1. ThermalExpert Class

1.3. void StartReceive()

Start to receive data via usb communication.

Returns

Remarks

This function must be called after the function, onFlashReadFinished(), in ThermalExpertListener, is called. This function is implemented with AsyncTask, continuously receives and corrects data.

After each frame data is received and corrected, the function, onOneFrameFinished(), in the ThermalExpertListener class is called automatically.

This function run until the function, StopReceive(), is executed.

Example

```
thermalExpert.StartReceive();
```

1. ThermalExpert Class

1.4. void StopReceive()

Stop receiving data

Returns

Remarks

This function is executed to stop receiving data.

Example

```
thermalExpert.StopReceive();
```

1. ThermalExpert Class

1.5. Bitmap GetImage()

Get a Bitmap of a corrected image.

Returns

Bitmap

Remarks

Example

```
thermalExpert.GetImage();
```

1. ThermalExpert Class

1.6. short[] GetData()

Get a raw data.

Returns

short[] : array of pixel raw data

Remarks

The size of array is 384 * 288. The size of data is 16bit.

Example

```
thermalExpert.GetImage();
```

1. ThermalExpert Class

1.7. void SetColorMap(int mode)

Set the mode of color map.

Parameters

0 : Gray

1 : Iron

2 : Rainbow

3 : PurpleYellow

Returns

Remarks

The color mode of the bitmap gotten by the function, GetBitmap(), is determined by this function, SetColorMap(int mode). The number of color map mode is 4(Gray, Iron, Rainbow, PurpleYellow).

Example

```
thermalExpert.GetImage();
```

1. ThermalExpert Class

1.8. void CalibrationImage()

Do calibration

Returns

Remarks

When Thermal Expert is directed to the flat surface, this function do calibration. It takes a few seconds(about 2 seconds) to do calibration. If calibration is finished, the function, onCalibrationFinished(), in the class ThermalExpertListener is called automatically.

Example

```
thermalExpert.Calibration();
```

1. ThermalExpert Class

1.9. int[] GetARGBData()

Get a ARGB 384 * 288 Data

Returns

Remarks

This function is used to get a ARGB Data. The size of the return data is 384 * 288. The data format is A(8bit)R(8bit)G(8bit)B(8bit).

Example

```
thermalExpert.GetARGBData();
```

1. ThermalExpert Class

1.10. byte[] GetYUV420()

Get a YUV420 Data

Returns

Remarks

This function is used to get a YUV420 Data. The size of the return data is 384 * 288 * 3/2 byte.

Example

```
thermalExpert.GetYUV420();
```

1. ThermalExpert Class

1.11. double GetPointTemperature(int x, int y)

Get the temperature of the point(x, y)

Returns

The temperature of the point(x, y)

Remarks

The intput of this function should be used with the smartphone display scale. And thermal images should be displayed with full screen. If it is not, the temperature might be not correct.

Refer below example..

Example

```
public boolean onTouchEvent(MotionEvent event) {
  if(event.getAction() == MotionEvent.ACTION_UP ) {
    int x = (int)event.getX();
    int y = (int)event.getY();

    double temperature = thermalExpert.GetPointTemperature(x, y);
  }
}
```

1. ThermalExpert Class

1.12. double GetPointTemperature(Point pt)

Get the temperature of the point

Returns

The temperature of the point

Remarks

The intput of this function should be used with the smartphone display scale. And thermal images should be displayed with full screen. If it is not, the temperature might be not correct.

Refer below example..

Example

```
public boolean onTouchEvent(MotionEvent event) {
  if(event.getAction() == MotionEvent.ACTION_UP ) {
    int x = (int)event.getX();
    int y = (int)event.getY();

    Point point = new Point(x, y);

    double temperature = thermalExpert.GetPointTemperature(point);
  }
}
```

1. ThermalExpert Class

1.13. double[] GetRectTemperature(Rect rect)

Get the temperature of the rectangle

Returns

The return value is array. The first value is the average temperature within the rectangle. The second value is the maximum value within the rectangle. The third value is the minimum value within the rectangle.

Remarks

The input of this function should be used with the smartphone display scale. And thermal images should be displayed with full screen. If it is not, the temperature might be not correct. Refer below example...

Example

```
Rectangle rect;
public boolean onTouchEvent(MotionEvent event) {
    int x = (int)event.getX();
    int y = (int)event.getY();
    If( event.getMotion() == MotionEvent.ACTION_DOWN) {
      rect.set(x, y, x, y);
    } else if( event.getMotion() == MotionEvent.ACTION_UP) {
      rect.set(rect.left, rect.top, x, y);
      double temperature = thermalExpert.GetRectTemperature(rect);
  }
 }
```

1. ThermalExpert Class

1.14. void SetUserLevelSpan(boolean isLevelSpan)

Set the value to determine if level span function for user turns on

Returns

Remarks

If you turn on user level span function, then you can control the maximum and minimum of the

view range.

Example

```
thermalExpert.SetUserLevelSpan(true);
```

1. ThermalExpert Class

1.15. void SetLevelSpanMin(double min)

Set the minimum value for level span

Returns

Remarks

You can control the minimum of the view range with this function.

Example

```
thermalExpert.SetLevelSpanMin(20.0);
```

1. ThermalExpert Class

1.16. void SetLevelSpanMax(double max)

Set the maximum value for level span

Returns

Remarks

You can control the maximum of the view range with this function.

Example

```
thermalExpert.SetLevelSpanMax(35.0);
```

1. ThermalExpert Class

1.17. double GetAGCLowTemp()

Get the minimum temperature in the image

Returns

The minimum temperature in the image.

Remarks

You can the minimum temperature in the image when User Level span is off.

Example

```
double agcLowTemperature = thermalExpert.GetAGCLowTemp();
```

1. ThermalExpert Class

1.18. double GetAGCHighTemp()

Get the maximum temperature in the image

Returns

The maximum temperature in the image.

Remarks

You can the minimum temperature in the image when User Level span is off.

Example

```
double agcHighTemperature = thermalExpert.GetAGCHighTemp();
```

1. ThermalExpert Class

1.19. byte[] GetGrayData()

Get a Gray Data

Returns

Remarks

This function is used to get a gray Data. The size of the return data is 384 * 288 byte.

Example

```
thermalExpert.GetGrayData();
```

1. ThermalExpert Class

1.20. void setUserDeadUpdate()

Set user dead update and dead correction.

Returns

Remarks

This function is used to get dead correction.

Example

```
thermalExpert.setUserDeadUpdate();
```

2. ThermalExpertListener Class

The class to listen the events. This class should be implemented in your project. This member functions are called in the library.

Public Member Functions

1: void onUsbConnected()

When thermal expert is connected with usb, this function is called

2: void onUsbDisconnected()

When thermal expert is disconnected with usb, this function is called

3: void onFlashReadFinished()

When reading flash data is finished, this function is called

4: void onOneFrameFinished()

Whenever a frame data is received, this function is called

5: void onCalibrationFinished(int mode)

When the image calibration is finished, this function is called

```
Public ThermalExpertListener thermalExpertListener = new ThermalExpertListener() {
    public void onUsbConnected() {
    }

    public void onUsbDisconnected() {
    }

    public void onFlashReadFinished() {
    }

    public void onOneFrameFinished() {
    }

    public void onCalibrationFinished() {
    }
};
```

2. ThermalExpertListener Class

2.1. void onUsbConnected ( )

When usb is connected, this function is called.

Parameters

Returns

Remarks

This function is called when the usb connection is succeeded. After this function is called, you can use the functions related to the usb communication.

2. ThermalExpertListener Class

2.2. void onUsbDisconnected ( )

When usb is disconnected, this function is called.

Parameters

Returns

Remarks

This function is called when the usb connection is removed. For example of the usage of this function, if this function is called, you can know not to use the usb communication or terminate your application.

2. ThermalExpertListener Class

2.3. void onFlashReadFinished ( )

When reading flash data is finished, this function is called.

Parameters

Returns

Remarks

There is the data for image correction in the flash memory in Thermal Expert. So the application should read this data before the function, StartReceive(), is executed. After reading flash data is finished, this function, onFlashReadFinished(), is called. The purpose of this function is that you can determine when StartReceive() function can be executed.

2. ThermalExpertListener Class

2.4. void onOneFrameFinished ( )

While receiving data via usb communication, when receiving a frame data is finished, this function is called.

Parameters

Returns

Remarks

If the function, StartReceve(), is executed, the application receives and corrects data continuously. This data is received for each frame. After one frame data is received, this function is called.

2. ThermalExpertListener Class

2.5. void onCalibrationFinished ( )

When calibration is finished, this function is called.

Parameters

Returns

Remarks

If the function, CalibrationImage(), is executed, the application start to calibrate image and this process takes a few seconds(about 2 seconds). When calibration is finished, this function is called.

Coding Example – Android

```
Public class MainActivity extends Activity {
    ThermalExpert te;

    ImageView       mImageView;
    Button          mCalibrationButton;

    int             mColorMode = 0;
    boolean         mIsPlay = true;

    protected void onCreated(Bundle savedInstanceState) {
        mImageView = (ImageView) findViewById(R.id.image_preview);
        mCalibrationButton = (Button) findViewById(R.id.calibration_button);

        te = new ThermalExpert(thermalExpertListener);
        te.Initial(getApplicationContext());
    }

    Button.OnClickLinstener mClickListener = new View.OnClickListener() {
        public void onClick(View v) {
            switch(v.getId()) {
            case R.id.calibration_button :
                mCalibrationButton.setEnabled(false);
                te.CalibrationImage();
                break;
            case R.id.color_button :
                te.SetColorMap((++mColorMode)%4);
                break;
            case R.id.play_button :
                if(mIsPlay) {
                    te.StopReceive();
                    mIsPlay = false;
                } else {
                    te.StartReceive();
                    mIsPlay = true;
                }
                break;
             }
         }
    }

    public boolean onTouchEvent(MotionEvent event) {
        te.GetPointTemperature((int)event.getX(), (int)event.getY());
        return super.onTouchEvent(event);
    }

    private ThermalExpertListener thermalExpertListener = new ThermalExpertListener() {
        public void onUsbConnected() {

        }

        public void onUsbDisconnected() {

        }

        public void onFlashReadFinished() {
            mImageView.setScaleType(scaleType.CENTER);
            te.StartReceive();
```

```
        }

        public void onOneFrameFinished() {
            mImageView.setBackground(new BitmapDrawable(te.GetImage()));
            //te.GetData();
        }

        public void onCalibrationFinised() {
            mCalibrationButton.setEnable(true);
        }
    }
}
```