

```

1  package edu.asu.msrs.artcelerationlibrary;
2
3  import android.graphics.Bitmap;
4  import android.graphics.Color;
5  import android.util.Log;
6
7  import java.nio.Buffer;
8  import java.nio.ByteBuffer;
9
10 /**
11  * Created by yitaochan on 12/4/16.
12  * The Motion Blur find the arithmetic average of the nearby pixels. For different"
    radius",
13  * different number of terms are needed to process. So first is to find out the sum of the area
    of interest
14  * and then divided by the number of terms, which is 2*radius+ 1.
15  */
16
17 public class MotionBlur {
18
19     String TAG = "MotionBlur";
20     Bitmap bmp_orig;
21
22
23
24
25     //Function: main method of the class. It calls all the other methods to implement the
    transform
26     //Input: byte array, img_width, img_height, input int args
27     //Output: byte array after motion blur transform
28
29     public byte[] motionBlur(byte[] bytes, int w, int h, int[] args){
30
31         Log.d(TAG, "Start");
32         int r = args[1];
33         int redValue;
34         int blueValue;
35         int greenValue;
36         bmp_orig = byteToBmp(bytes, w, h);
37
38         switch (args[0]){
39
40             case 0:
41                 for(int x = r; x < w - r; x++) {
42                     for (int y = r; y < h - r; y++) {
43                         redValue = Color.red(bmp_orig.getPixel(x, y));
44                         blueValue = Color.blue(bmp_orig.getPixel(x, y));
45                         greenValue = Color.green(bmp_orig.getPixel(x, y));
46                         for (int k = 1; k <= r; k++) {

```

```

47         redValue += ( Color.red(bmp_orig.getPixel((x - k), y)) + Color.red(
    bmp_orig.getPixel((x + k), y)));
48         blueValue += ( Color.blue(bmp_orig.getPixel((x - k), y)) + Color.blue
    (bmp_orig.getPixel((x + k), y)));
49         greenValue += ( Color.green(bmp_orig.getPixel((x - k), y)) + Color.
    green(bmp_orig.getPixel((x + k), y)));
50     }
51     redValue = redValue / (2 * r + 1);
52     blueValue = blueValue / (2 * r + 1);
53     greenValue = greenValue / (2 * r + 1);
54     bmp_orig.setPixel(x, y, Color.argb(255, redValue, greenValue, blueValue
    ));
55
56     }
57 }
58
59     break;
60 case 1:
61     for(int x = r; x < w - r; x++) {
62         for (int y = r; y < h - r; y++) {
63             redValue = Color.red(bmp_orig.getPixel(x, y));
64             blueValue = Color.blue(bmp_orig.getPixel(x, y));
65             greenValue = Color.green(bmp_orig.getPixel(x, y));
66             for (int k = 1; k <= r; k++) {
67                 redValue += ( Color.red(bmp_orig.getPixel(x, (y - k))) + Color.red(
    bmp_orig.getPixel(x, (y + k))));
68                 blueValue += ( Color.blue(bmp_orig.getPixel(x, (y - k))) + Color.blue
    (bmp_orig.getPixel(x, (y + k))));
69                 greenValue += ( Color.green(bmp_orig.getPixel(x, (y - k))) + Color.
    green(bmp_orig.getPixel(x, (y + k))));
70             }
71             redValue = redValue / (2 * r + 1);
72             blueValue = blueValue / (2 * r + 1);
73             greenValue = greenValue / (2 * r + 1);
74             bmp_orig.setPixel(x, y, Color.argb(255, redValue, greenValue, blueValue
    ));
75
76         }
77     }
78     break;
79 default:
80     break;
81
82 }
83
84
85     Log.d(TAG, "END");
86
87     return bmpToByte(bmp_orig);

```

```
88
89     }
90
91
92
93     //Function: convert bitmap object into byte array
94     //Input: bitmap object
95     //Output: byte array
96
97     public byte[] bmpToByte(Bitmap bitmap){
98
99         ByteBuffer buffer = ByteBuffer.allocateDirect(bitmap.getByteCount());
100         bitmap.copyPixelsToBuffer(buffer);
101
102         byte[] bytes = buffer.array();
103
104         return bytes;
105     }
106
107
108     //Function: convert byte array into bitmap object
109     //Input: byte array
110     //Output: bitmap object
111
112     public Bitmap byteToBmp (byte[] b, int w, int h){
113
114
115         Buffer buf = null;
116         buf = ByteBuffer.wrap(b);
117         Bitmap.Config conf = Bitmap.Config.ARGB_8888;
118         Bitmap bmp = Bitmap.createBitmap(w, h, conf);
119
120         bmp.copyPixelsFromBuffer(buf);
121
122         return bmp;
123     }
124 }
125 }
126
```