Gamma гамма-коррекция

О том, что такое **Гамма-коррекция**, читайте в [Википедии](http://ru.wikipedia.org/wiki/%D0%93%D0%B0%D0%BC%D0%BC%D0%B0-%D0%BA%D0%BE%D1%80%D1%80%D0%B5%D0%BA%D1%86%D0%B8%D1%8F). Сам я в этом не разбираюсь.

Реализуем гамма-коррекцию на Android. За основу возьмём [предыдущий пример](http://developer.alexanderklimov.ru/android/catshop/matrix.php#prescale).

Перейдём к коду. Поиграйте с разными значениями.

butDemo.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

BitmapDrawable mydrawable = (BitmapDrawable) imageView.getDrawable();

Bitmap b = mydrawable.getBitmap();

b = doGamma(b, 1.6, 1.6, 1.6);

imageView.setImageBitmap(b);

}

});

public static Bitmap doGamma(Bitmap src, double red, double green, double blue) {

// create output image

Bitmap bmOut = Bitmap.createBitmap(src.getWidth(), src.getHeight(), src.getConfig());

// получим размеры изображения

int width = src.getWidth();

int height = src.getHeight();

// информация о цвете

int A, R, G, B;

int pixel;

// constant value curve

final int MAX\_SIZE = 256;

final double MAX\_VALUE\_DBL = 255.0;

final int MAX\_VALUE\_INT = 255;

final double REVERSE = 1.0;

// gamma arrays

int[] gammaR = new int[MAX\_SIZE];

int[] gammaG = new int[MAX\_SIZE];

int[] gammaB = new int[MAX\_SIZE];

// setting values for every gamma channels 1098

for(int i = 0; i < MAX\_SIZE; ++i) {

gammaR[i] = (int)Math.min(MAX\_VALUE\_INT,

(int)((MAX\_VALUE\_DBL \* Math.pow(i / MAX\_VALUE\_DBL, REVERSE / red)) + 0.5));

gammaG[i] = (int)Math.min(MAX\_VALUE\_INT,

(int)((MAX\_VALUE\_DBL \* Math.pow(i / MAX\_VALUE\_DBL, REVERSE / green)) + 0.5));

gammaB[i] = (int)Math.min(MAX\_VALUE\_INT,

(int)((MAX\_VALUE\_DBL \* Math.pow(i / MAX\_VALUE\_DBL, REVERSE / blue)) + 0.5));

}

// apply gamma table

for(int x = 0; x < width; ++x) {

for(int y = 0; y < height; ++y) {

// get pixel color

pixel = src.getPixel(x, y);

A = Color.alpha(pixel);

// look up gamma

R = gammaR[Color.red(pixel)];

G = gammaG[Color.green(pixel)];

B = gammaB[Color.blue(pixel)];

// set new color to output bitmap

bmOut.setPixel(x, y, Color.argb(A, R, G, B));

}

}

// возвращаем финальное изображение

return bmOut;

}

  

Р