

# **ZPHOTO ENGINE**

**--VERSION 3.8**

**DATE:** 20170807

**UPDATE:** 20180820

**CopyRight:** Trent

## 目录

前言.....	4
基础接口说明.....	5
1.ZPHOTO_Saturation.....	5
2.ZPHOTO_NaturalSaturation.....	5
3.ZPHOTO_HueAndSaturation.....	5
4.ZPHOTO_Lightness.....	6
5.ZPHOTO_LinearBrightContrast.....	6
6.ZPHOTO_NLinearBrightContrast.....	6
7.ZPHOTO_AutoContrast.....	7
8.ZPHOTO_AutoContrastAjdustWithParameters.....	7
9.ZPHOTO_AutoColorGradation.....	8
10.ZPHOTO_AutoColorGradationAjdustWithParameters.....	8
11.ZPHOTO_Curve.....	8
12.ZPHOTO_Posterize.....	9
13.ZPHOTO_OverExposure.....	9
14.ZPHOTO_Invert.....	9
15.ZPHOTO_HistogramEqualize.....	10
16.ZPHOTO_Desaturate.....	10
17.ZPHOTO_Blackwhite.....	10
18.ZPHOTO_Threshold.....	11
19.ZPHOTO_FastGaussFilter.....	11
20.ZPHOTO_HighPass.....	12
21.ZPHOTO_USM.....	12
22.ZPHOTO_FindEdges.....	12
23.ZPHOTO_ChannelMixProcess.....	13
24.ZPHOTO_ColorTemperature.....	13
25.ZPHOTO_HighlightShadowPrecise.....	14
26.ZPHOTO_Exposure.....	14
27.ZPHOTO_FastMeanFilter.....	14
28.ZPHOTO_FastestGaussFilter.....	15
29.ZPHOTO_SobelFilter.....	15
30.ZPHOTO_ImageTransformation.....	15
31.ZPHOTO_Fragment.....	16
32.ZPHOTO_MotionBlur.....	16
33.ZPHOTO_SurfaceBlur.....	17
34.ZPHOTO_RadialBlur.....	17
35.ZPHOTO_ZoomBlur.....	18
36.ZPHOTO_Relief.....	18
37.ZPHOTO_Mean.....	18
38.ZPHOTO_Mosaic.....	19
39.ZPHOTO_ColorBalance.....	19
40.ZPHOTO_Diffusion.....	19

41.ZPHOTO_LSNBlur.....	20
42.ZPHOTO_ColorLevel.....	20
43.ZPHOTO_MedianFilter.....	21
44.ZPHOTO_MaxFilter.....	21
45.ZPHOTO_MinFilter.....	21
46.ZPHOTO_GlowingEdges.....	22
47.ZPHOTO_ImageWarpWave.....	22
48.ZPHOTO_SmartBlurFilter.....	23
49.ZPHOTO_AnisotropicFilter.....	23
50.ZPHOTO_DisplacementFilter.....	23
51.ZPHOTO_NoiseEffect.....	24
52.ZPHOTO_RGBA2BGRA.....	24
53.ZPHOTO_BGRA2RGBA.....	25
图层混合模式接口说明.....	26
1.ZPHOTO_ImageBlendEffect.....	26
2.ZPHOTO_ModeDarken.....	27
3.ZPHOTO_ModeMultiply.....	27
4.ZPHOTO_ModeColorBurn.....	27
5.ZPHOTO_ModeLinearBurn.....	27
6.ZPHOTO_ModeDarkness.....	28
7.ZPHOTO_ModeLighten.....	28
8.ZPHOTO_ModeScreen.....	28
9.ZPHOTO_ModeColorDodge.....	29
10.ZPHOTO_ModeColorLinearDodge.....	29
11.ZPHOTO_ModeLightColor.....	29
12.ZPHOTO_ModeOverlay.....	29
13.ZPHOTO_ModeSoftLight.....	30
14.ZPHOTO_ModeHardLight.....	30
15.ZPHOTO_ModeVividLight.....	30
16.ZPHOTO_ModeLinearLight.....	30
17.ZPHOTO_ModePinLight.....	31
18.ZPHOTO_ModeSolidColorMixing.....	31
19.ZPHOTO_ModeDifference.....	31
20.ZPHOTO_ModeExclusion.....	32
21.ZPHOTO_ModeSubtraction.....	32
22.ZPHOTO_ModeDivide.....	32
23.ZPHOTO_ModeDesaturate.....	32
24.ZPHOTO_ModeColorInvert.....	33
颜色空间转换接口说明.....	34
1.ZPHOTO_RGBToYCbCr.....	34
2.ZPHOTO_YCbCrToRGB.....	34
3.ZPHOTO_RGBToLAB.....	34
4.ZPHOTO_LABToRGB.....	35
滤镜引擎接口说明.....	36

1.ZPHOTO_LUTFilter.....	36
错误异常返回.....	37
Android API 接口说明.....	38
图像读写.....	39
API 说明: .....	39
1.ReadImage.....	39
2.SaveImage.....	39
3.图像格式定义.....	39
4.像素格式定义.....	39
5.图像 SFBITMAP.....	40
6.IO 返回错误码.....	40
7.SampleCode.....	40
版本更新记录.....	42

## 前言

ZPhotoEngine 是一个包含众多常用图像处理算法的引擎库，该引擎使用 C 语言开发，针对 32 位 BGRA 格式图像，实现 PhotoShop 中常用的图像处理算法功能，具有较好的跨平台移植特性，主要适用于 PC/Android/iOS 平台图像应用的快速开发。

ZPhotoEngine 说明文档分为四部分：基础接口部分，图层混合接口部分，颜色空间转换部分，滤镜引擎部分。

第一部分基础接口将详细介绍基础算法接口的使用，此部分包含了 PS 基础算法及部分滤镜特效接口以及目前 app 修图应用中主流图像调节算法(高光阴影、色温、曝光等)接口等。

第二部分图层混合接口，介绍各种 PS 图层混合模式的使用。

第三部分颜色空间转换接口，主要提供了常用颜色空间和 RGB 颜色空间的相互转换接口，方便开发者在不同空间进行处理。该引擎提供相应的开发 DEMO，供使用者查阅。

第四部分滤镜引擎，引擎提供了 LUT 滤镜接口，可以自定义设计模板和图层混合模式，方便开发者开发各种效果滤镜。

第五部分图像读写，提供 BMP/PNG/JPG 图像读写库，不依赖 opencv 等任何第三方插件，方便大家在不同平台下使用。

注意，引擎算法采用多线程优化，PC 端需配置多线程运行环境，Android 版无需配置。

最后，本人将不断对该引擎提供算法更新，以满足不同开发者的需求。

Github 地址: <https://github.com/XiuSdk> CSDN 博客: <http://blog.csdn.net/trent1985>

# 基础接口说明

## 1.ZPHOTO\_Saturation

名称:

```
int ZPHOTO_Saturation(unsigned char* srcData, int width, int height, int stride, int saturation);
```

接口描述: 饱和度调节

参数:

srcData--原始图像 Buffer;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
saturation--饱和度值, 范围为[-100, 100];

返回值:

0-OK, 其他失败;

## 2.ZPHOTO\_NaturalSaturation

名称:

```
int ZPHOTO_NaturalSaturation(unsigned char* srcData, int width, int height, int stride, int saturation);
```

接口描述: 自然饱和度调节

参数:

srcData--原始图像 Buffer;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
saturation--饱和度值, 范围为[-100, 100];

返回值:

0-OK, 其他失败;

## 3.ZPHOTO\_HueAndSaturation

名称:

```
int ZPHOTO_HueAndSaturation(unsigned char* srcData, int width, int height, int stride, int hue, int saturation);
```

接口描述: 色相和饱和度调节

参数:

srcData--原始图像 Buffer;  
width--图像宽度;

height--图像高度;  
stride--图像 Stride;  
hue--色相值, 范围为[-180, 180];  
saturation--饱和度值, 范围为[-100, 100]

**返回值:**

0-OK, 其他失败;

## 4.ZPHOTO\_Lightness

**名称:**

`int ZPHOTO_Lightness(unsigned char* srcData, int width, int height, int stride, int lightness);`

**接口描述:** 明度调节

**参数:**

srcData--原始图像 Buffer;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
lightness--明度值, 范围为[-100, 100];

**返回值:**

0-OK, 其他失败;

## 5.ZPHOTO\_LinearBrightContrast

**名称:**

`int ZPHOTO_LinearBrightContrast(unsigned char* srcData, int width, int height, int stride, int brightness, int contrast, int threshold);`

**接口描述:** 线性对亮度对比度调节

**参数:**

srcData--原始图像 Buffer;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
brightness--亮度值, 范围为[-255, 255];  
contrast--对比度值, 范围为[-100, 100];  
threshold--调节阈值, 范围为[0, 255], 默认值 128;

**返回值:**

0-OK, 其他失败;

## 6.ZPHOTO\_NLinearBrightContrast

**名称:**

```
int ZPHOTO_NLinearBrightContrast(unsigned char* srcData, int width, int height, int stride, int bright, int contrast, int threshold);
```

**接口描述：**非线性亮度对比度调节

**参数：**

srcData--原始图像 Buffer，32 位 BGRA；  
width--图像宽度；  
height--图像高度；  
stride--图像 Stride；  
brightness--亮度值，范围为[-100, 100]；  
contrast--对比度值，范围为[-100, 100]；  
threshold--调节阈值，范围为[0, 255]，默认值 128；

**返回值：**

0-OK，其他失败；

## 7.ZPHOTO\_AutoContrast

**名称：**

```
int ZPHOTO_AutoContrast(unsigned char *srcData, int width, int height, int stride);
```

**接口描述：**自动对比度调节

**参数：**

srcData--原始图像 Buffer，32 位 BGRA；  
width--图像宽度；  
height--图像高度；  
stride--图像 Stride；

**返回值：**

0-OK，其他失败；

## 8.ZPHOTO\_AutoContrastAjdustWithParameters

**名称：**

```
int ZPHOTO_AutoContrastAdjustWithParameters(unsigned char *srcData, int width, int height, int stride, float shadowCorrectRatio, float highlightCorrectRatio);
```

**接口描述：**参数限制的自动对比度调节

**参数：**

srcData--原始图像 Buffer，32 位 BGRA；  
width--图像宽度；  
height--图像高度；  
stride--图像 Stride；  
shadowCorrectRatio--阴影修剪比例，范围[0.00, 9.99]  
highlightCorrectRatio--高光修剪比例，范围[0.00, 9.99]

**返回值：**

0-OK，其他失败；

## 9.ZPHOTO\_AutoColorGradation

名称:

`int ZPHOTO_AutoColorGradation(unsigned char *srcData, int width, int height, int stride);`

接口描述: 自动色阶调节

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;

返回值:

0-OK, 其他失败;

## 10.ZPHOTO\_AutoColorGradationAjdustWithParameter

### S

名称:

`int ZPHOTO_AutoColorGradationAdjustWithParameters(unsigned char *srcData, int width, int height, int stride, float shadowCorrectRatio, float highlightCorrectRatio);`

接口描述: 参数限制的自动色阶调节

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
shadowCorrectRatio--阴影修剪比例, 范围[0.00, 9.99]  
highlightCorrectRatio--高光修剪比例, 范围[0.00, 9.99]

返回值:

0-OK, 其他失败;

## 11.ZPHOTO\_Curve

名称:

`int ZPHOTO_Curve(unsigned char* srcData, int width, int height, int stride, int dstChannel, int knotsPos[6]);`

接口描述: 曲线调节

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;



destChannel--通道选择(Gray-B-G-R:0-1-2-3);

knotsPos--三个点 XY 坐标数组(起始点+中间点+末尾点)，三个点控制一条 spline 亮度曲线;

**返回值:**

0-OK, 其他失败;

## 12.ZPHOTO\_Posterize

**名称:**

`int ZPHOTO_Posterize(unsigned char *srcData, int width, int height, int stride, int clusterNum);`

**接口描述:** 色调分离

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

clusterNum--色调数目, 范围为[2, 255];

**返回值:**

0-OK, 其他失败;

## 13.ZPHOTO\_OverExposure

**名称:**

`int ZPHOTO_OverExposure(unsigned char *srcData, int width, int height, int stride);`

**接口描述:** 过度曝光

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

**返回值:**

0-OK, 其他失败;

## 14.ZPHOTO\_Invert

**名称:**

`int ZPHOTO_Invert(unsigned char *srcData, int width, int height, int stride);`

**接口描述:** 反相

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

**返回值:**

0-OK, 其他失败;

## 15.ZPHOTO\_HistogramEqualize

**名称:**

`int ZPHOTO_HistogramEqualize(unsigned char* srcData, int width, int height, int stride);`

**接口描述:** 色调均化

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

**返回值:**

0-OK, 其他失败;

## 16.ZPHOTO\_Desaturate

**名称:**

`int ZPHOTO_Desaturate(unsigned char *srcData, int width, int height, int stride, int ratio);`

**接口描述:** 去色

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

ratio--去色程度, 范围[0, 100]

**返回值:**

0-OK, 其他失败;

## 17.ZPHOTO\_Blackwhite

**名称:**

`int ZPHOTO_Blackwhite(unsigned char *srcData, int width, int height, int stride, int kRed, int kGreen, int kBlue, int kYellow, int kCyan, int kMagenta);`

**接口描述:** 黑白

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;

width--图像宽度;

height--图像高度;

stride--图像 Stride;  
kRed-红色比例, 范围[-200, 300]  
kGreen-绿色比例, 范围[-200, 300]  
kBlue-蓝色比例, 范围[-200, 300]  
kYellow-黄色比例, 范围[-200, 300]  
kCyan-青色比例, 范围[-200, 300]  
kMagenta-洋红比例, 范围[-200, 300]

**返回值:**

0-OK, 其他失败;

## 18.ZPHOTO\_Threshold

**名称:**

```
int ZPHOTO_Threshold(unsigned char *srcData, int width, int height, int stride, int threshold);
```

**接口描述:** 阈值

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;

**返回值:**

0-OK, 其他失败;

## 19.ZPHOTO\_FastGaussFilter

**名称:**

```
int ZPHOTO_FastGaussFilter(unsigned char* srcData, int width, int height, int stride, unsigned char* dstData, float radius);
```

**接口描述:** 高斯模糊

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
dstData--目标图像 Buffer, 必须与 srcData 大小相同;  
radius--高斯半径, 范围为[0, 1000];

**返回值:**

0-OK, 其他失败;

## 20.ZPHOTO\_HighPass

名称:

```
int ZPHOTO_HighPass(unsigned char* srcData, int width, int height, int stride, float mRadius);
```

接口描述: 高反差保留

参数:

srcData--原始图像 Buffer, 32 位 BGRA;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

mRadius--高斯半径, 范围为[0,1000];

返回值:

0-OK, 其他失败;

## 21.ZPHOTO\_USM

名称:

```
int ZPHOTO_USM(unsigned char* srcData, int width, int height, int stride, float radius, int amount, int threshold);
```

接口描述: USM 锐化

参数:

srcData--原始图像 Buffer, 32 位 BGRA;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

radius--高斯半径, 范围为[0,1000];

amount--锐化程度, 范围为[0,500];

threshold--锐化阈值, 范围为[0,255];

返回值:

0-OK, 其他失败;

## 22.ZPHOTO\_FindEdges

名称:

```
int ZPHOTO_FindEdges(unsigned char *srcData, int width, int height, int stride);
```

接口描述: 查找边缘

参数:

srcData--原始图像 Buffer, 32 位 BGRA;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

返回值:

0-OK, 其他失败;

## 23.ZPHOTO\_ChannelMixProcess

名称:

```
int ZPHOTO_ChannelMixProcess(unsigned char* srcData, int width, int height, int stride, int channel, int kr, int kg, int kb, int N, bool singleColor, bool constAdjust);
```

接口描述: 通道混合

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
channel--Red-0, Green-1, Blue-2, Gray-3;  
kr--Red 通道比例, 范围[-200, 200];  
kg--Green 通道比例, 范围[-200, 200];  
kb--Blue 通道比例, 范围[-200, 200];  
N--常数比例, 范围[-200, 200];  
singleColor--是否单色调整, 单色-true, 彩色-false;  
constAdjust--是否执行常数调整, 执行-true, 不执行-false;

返回值:

0-OK, 其他失败;

## 24.ZPHOTO\_ColorTemperature

名称:

```
int ZPHOTO_ColorTemperature(unsigned char* srcData, int width, int height, int stride, int intensity);
```

接口描述: 色温调节

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
intensity--色温强度, 取值范围为[-50, 50]:  
    intensity < 0 时, 冷色;  
    intensity = 0 时, 保持原图;  
    intensity > 0 时, 暖色;

返回值:

0-OK, 其他失败;

## 25.ZPHOTO\_HighlightShadowPrecise

名称:

```
int ZPHOTO_HighlightShadowPrecise(unsigned char* srcData, int width, int height, int stride, float highlight, float shadow);
```

接口描述: 高光阴影调节

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
highlight--高光强度值, 取值范围为[-200, 100];  
shadow--阴影强度值, 取值范围为[-200, 100];

返回值:

0-OK, 其他失败;

## 26.ZPHOTO\_Exposure

名称:

```
int ZPHOTO_Exposure(unsigned char* srcData, int width, int height, int stride, int intensity);
```

接口描述: 曝光调节

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
intensity--曝光强度值, 取值范围为[0, 100];

返回值:

0-OK, 其他失败;

## 27.ZPHOTO\_FastMeanFilter

名称:

```
int ZPHOTO_FastMeanFilter(unsigned char* srcData, int width, int height, int stride, int radius);
```

接口描述: 高光调节

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;

radius--均值滤波半径，取值范围为 $[0, \text{width} / 2]$ ；

**返回值：**

0-OK，其他失败；

## 28.ZPHOTO\_FastestGaussFilter

**名称：**

```
int ZPHOTO_FastestGaussFilter(unsigned char* srcData, int width, int height, int stride, float radius);
```

**接口描述：**快速高斯滤波

**参数：**

srcData--原始图像 Buffer，32 位 BGRA；

width--图像宽度；

height--图像高度；

stride--图像 Stride；

radius--高斯模糊半径，取值范围为 $[0, 128]$ ；

**返回值：**

0-OK，其他失败；

## 29.ZPHOTO\_SobelFilter

**名称：**

```
int ZPHOTO_SobelFilter(unsigned char *srcData, int width, int height, int stride);
```

**接口描述：**Sobel 边缘滤波

**参数：**

srcData--原始图像 Buffer，32 位 BGRA；

width--图像宽度；

height--图像高度；

stride--图像 Stride；

**返回值：**

0-OK，其他失败；

## 30.ZPHOTO\_ImageTransformation

**名称：**

```
①int ZPHOTO_ImageTransformation(unsigned char *srcData, int srcImgSize[2], unsigned char *dstData, int dstImgSize[2], float H[], int Interpolation_method, int Transform_method);
```

```
②int ZPHOTO_CalcWH( int inputImgSize[2], float angle, float scale, int transform_method, int outputImgSize[2], float H[]);
```

**接口描述：**图像仿射变换，包括图像旋转，缩放，平移，镜像操作；要求先调用接口②，得到目标图像大小和 H 矩阵，然后调用接口①实现相应功能。

**参数：**

srcData--原始图像 Buffer, 32 位 BGRA;  
 srcImgSize--原始图像宽高信息数组;  
 dstData--目标图像 Buffer, 大小由接口②获得;  
 dstImgSize--目标图像宽高信息数组;  
 H--变换矩阵数组, 长度为 6;  
 Interpolation\_method--插值方法选择: interpolation\_bilinear, interpolation\_nearest;  
 Transform\_method--变换方法:  
     transform\_scale 缩放变换, 取值为 0;  
     transform\_rotation 旋转变换, 取值为 1;  
     transform\_rotation\_scale 缩放旋转变换, 取值为 2;  
     transform\_affine 仿射变换, 取值为 3;  
     transform\_mirror\_h 水平镜像变换, 取值为 4;  
     transform\_mirror\_v 垂直镜像变换, 取值为 5;  
     transform\_offset 平移变换, 取值为 6;  
 inputImgSize--输入图像宽高信息;  
 angle--旋转角度值, 取值范围为[-360-360];  
 scale--缩放变换值, 取值大于 0;  
 outputImgSize--输出图像宽高信息;

**返回值:**

0-OK, 其他失败;

## 31.ZPHOTO\_Fragment

**名称:**

`int ZPHOTO_Fragment(unsigned char *srcData, int width, int height, int stride);`

**接口描述:** 碎片功能**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;  
 width--图像宽度;  
 height--图像高度;  
 stride--图像 Stride;

**返回值:**

0-OK, 其他失败;

## 32.ZPHOTO\_MotionBlur

**名称:**

`int ZPHOTO_MotionBlur(unsigned char* srcData, int width, int height, int stride, int angle, int distance);`

**接口描述:** 运动模糊**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;  
 width--图像宽度;



height--图像高度;  
stride--图像 Stride;  
angle--运动模糊角度值, 取值范围为[0, 360];  
distance--运动模糊距离值, 取值范围为[0, 200];

**返回值:**

0-OK, 其他失败;

## 33.ZPHOTO\_SurfaceBlur

**名称:**

```
int ZPHOTO_SurfaceBlur(unsigned char *srcData, int width, int height, int stride, int threshold, int radius);
```

**接口描述:** 表面模糊

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
threshold--运动模糊角度值, 取值范围为[0, 255];  
radius--运动模糊距离值, 取值范围为[0, 10];

**返回值:**

0-OK, 其他失败;

## 34.ZPHOTO\_RadialBlur

**名称:**

```
int ZPHOTO_RadialBlur(unsigned char* srcData, int width, int height, int stride, int cenX, int cenY, int amount);
```

**接口描述:** 旋转模糊

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
cenX--旋转模糊中心 X 坐标;  
cenY--旋转模糊中心 Y 坐标;  
amount--旋转模糊程度, 范围为[1-100];

**返回值:**

0-OK, 其他失败;

## 35.ZPHOTO\_ZoomBlur

名称:

```
int ZPHOTO_ZoomBlur(unsigned char* srcData, int width, int height ,int stride, int cenX,  
int cenY, int sampleRadius, int amount);
```

接口描述: 缩放模糊

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
cenX--旋转模糊中心 X 坐标;  
cenY--旋转模糊中心 Y 坐标;  
sampleRadius--缩放模糊半径, 范围为[0-255];  
amount--缩放模糊程度, 范围为[1-100];

返回值:

0-OK, 其他失败;

## 36.ZPHOTO\_Relief

名称:

```
int ZPHOTO_Relief(unsigned char *srcData, int width, int height, int stride, int angle, int  
amount);
```

接口描述: 浮雕

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
angle--浮雕角度, 范围为[0-360];  
amount--缩放模糊程度, 范围为[1-500];

返回值:

0-OK, 其他失败;

## 37.ZPHOTO\_Mean

名称:

```
int ZPHOTO_Mean(unsigned char *srcData, int width, int height, int stride);
```

接口描述: 平均

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;

height--图像高度;  
stride--图像 Stride;

**返回值:**

0-OK, 其他失败;

## 38.ZPHOTO\_Mosaic

**名称:**

```
int ZPHOTO_Mosaic(unsigned char* srcData, int width, int height, int stride, int size);
```

**接口描述:** 马赛克

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
size--马赛克半径, 范围为[0-200];

**返回值:**

0-OK, 其他失败;

## 39.ZPHOTO\_ColorBalance

**名称:**

```
int ZPHOTO_ColorBalance(unsigned char* srcData, int width, int height, int stride, int cyan,  
int magenta, int yellow, int channel, bool preserveLuminosity);
```

**接口描述:** 色彩平衡

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
cyan--青色调调整, 范围为[-100-100];  
magenta--洋红调整, 范围为[-100-100];  
yellow--黄色调整, 范围为[-100-100];  
channel--通道选择, RGB 通道 0, R 通道 1, G 通道 2, B 通道 3;  
preserveLuminosity--是否保留明度(true-保留, false-不保留);

**返回值:**

0-OK, 其他失败;

## 40.ZPHOTO\_Diffusion

**名称:**

```
int ZPHOTO_Diffusion(unsigned char* srcData, int width, int height, int stride, int intensity);
```

**接口描述:** 扩散

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
intensity--扩散程度, 范围为[0-100];

**返回值:**

0-OK, 其他失败;

## 41.ZPHOTO\_LSNBlur

**名称:**

```
int ZPHOTO_LSNBlur(unsigned char* srcData, int width, int height, int stride, int radius, int delta);
```

**接口描述:** LSNBlur 模糊

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
radius--模糊半径, 范围为[0-200];  
delta--模糊方差, 范围为[0-500];

**返回值:**

0-OK, 其他失败;

## 42.ZPHOTO\_ColorLevel

**名称:**

```
int ZPHOTO_ColorLevel(unsigned char * srcData , int width, int height ,int stride , int destChannel, unsigned char inputLeftLimit, float inputMiddle, unsigned char inputRightLimit, unsigned char outputLeftLimit , unsigned char outputRightLimit);
```

**接口描述:** 色阶调节

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
destChannel--通道选择(Gray-B-G-R:0-1-2-3);  
inputLeftLimit--输入最小值, 范围为[0, 255];  
inputMiddle--输入中间值, 范围为[0, 9.99];

inputRightLimit--输入最大值，范围为[0, 255]；

outputLeftLimit --输出最小值，范围为[0, 255]；

outputRightLimit--输出最大值，范围为[0, 255]；

**返回值：**

0-OK，其他失败；

## 43.ZPHOTO\_MedianFilter

**名称：**

```
int ZPHOTO_MedianFilter(unsigned char *srcData, int width, int height, int stride, int radius);
```

**接口描述：**中值滤波(中间色)

**参数：**

srcData--原始图像 Buffer，32 位 BGRA；

width--图像宽度；

height--图像高度；

stride--图像 Stride；

radius--模糊半径，范围为[0, 100]；

**返回值：**

0-OK，其他失败；

## 44.ZPHOTO\_MaxFilter

**名称：**

```
int ZPHOTO_MaxFilter(unsigned char *srcData, int width, int height, int stride, int radius);
```

**接口描述：**最大值滤波(最大值)

**参数：**

srcData--原始图像 Buffer，32 位 BGRA；

width--图像宽度；

height--图像高度；

stride--图像 Stride；

radius--模糊半径，范围为[0, 100]；

**返回值：**

0-OK，其他失败；

## 45.ZPHOTO\_MinFilter

**名称：**

```
int ZPHOTO_MinFilter(unsigned char *srcData, int width, int height, int stride, int radius);
```

**接口描述：**最小值滤波(最小值)

**参数：**

srcData--原始图像 Buffer, 32 位 BGRA;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

radius--模糊半径, 范围为[0,100];

**返回值:**

0--OK, 其他失败;

## 46.ZPHOTO\_GlowingEdges

**名称:**

```
int ZPHOTO_GlowingEdges(unsigned char* srcData, int width, int height, int stride, int edgeSize, int edgeLightness, int edgeSmoothness);
```

**接口描述:** 照亮边缘

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

edgeSize--边缘宽度值, 取值范围为[1,14];

edgeLightness--边缘亮度值, 取值范围为[0,20];

edgeSmoothness--平滑度, 取值范围为[1,15];

**返回值:**

0--OK, 其他失败;

## 47.ZPHOTO\_ImageWarpWave

**名称:**

```
int ZPHOTO_ImageWarpWave(unsigned char *srcData, int width, int height, int stride, int intensity);
```

**接口描述:** 波浪变形

**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

intensity--变形强度, 取值范围为[0,100];

**返回值:**

0--OK, 其他失败;

## 48.ZPHOTO\_SmartBlurFilter

名称:

```
int ZPHOTO_SmartBlurFilter(unsigned char* srcData, int width, int height, int stride, int size, int threshold);
```

接口描述: Smart Blur

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
size--模糊半径, 取值范围[0, 100]  
threshold--阈值调节, 取值范围为[0, 255];

返回值:

0--OK, 其他失败;

## 49.ZPHOTO\_AnisotropicFilter

名称:

```
int ZPHOTO_AnisotropicFilter(unsigned char* srcData, int width, int height, int stride, int iter, float k, float lambda = 0.25, int offset = 3);
```

接口描述: 各向异性扩散滤波

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
iter--迭代次数, 范围[0, 100]  
k--扩散系数, 范围[0, 0.35], default=0.25  
offset--像素采样间距, 范围[0, 10], default=3

返回值:

0--OK, 其他失败;

## 50.ZPHOTO\_DisplacementFilter

名称:

```
int ZPHOTO_DisplacementFilter(unsigned char* srcData, int width, int height, int stride, unsigned char* maskData, int mWidth, int mHeight, int mStride, int hRatio, int vRatio);
```

接口描述: 置换滤镜

参数:

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;

height--图像高度;  
stride--图像 Stride;  
maskData--置换图 32 位 BGRA 格式  
mWidth--置换图宽度  
mHeight--置换图高度  
mStride--置换图 Stride  
hRatio--水平比例, 范围[-100, 100]  
vRatio--垂直比例, 范围[-100, 100]

**返回值:**

0-OK, 其他失败;

## 51.ZPHOTO\_NoiseEffect

**名称:**

```
int ZPHOTO_NoiseEffect(unsigned char *srcData, int width, int height, int stride, int ratio, float sigma, float phase);
```

**接口描述:** 噪声效果**参数:**

srcData--原始图像 Buffer, 32 位 BGRA;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;  
ratio--噪声图混合比例, 范围[0, 100]  
sigma--噪声方差, 范围[0, ]  
phase--噪声幅度, 范围[0, ]

**返回值:**

0-OK, 其他失败;

## 52.ZPHOTO\_RGBA2BGRA

**名称:**

```
int ZPHOTO_RGBA2BGRA(unsigned char* srcData, int width, int height, int stride);
```

**接口描述:** RGBA 转 BGRA 格式**参数:**

srcData--原始 RGBA 格式图像 Buffer, 执行后转为 BGRA 格式;  
width--图像宽度;  
height--图像高度;  
stride--图像 Stride;

**返回值:**

0-OK, 其他失败;



## 53.ZPHOTO\_BGRA2RGBA

**名称:**

```
int ZPHOTO_BGRA2RGBA(unsigned char* srcData, int width, int height, int stride);
```

**接口描述:** BGRA 转 RGBA 格式

**参数:**

srcData--原始 BGRA 格式图像 Buffer，执行后转为 RGBA 格式；

width--图像宽度；

height--图像高度；

stride--图像 Stride；

**返回值:**

0-OK，其他失败；

# 图层混合模式接口说明

## 1.ZPHOTO\_ImageBlendEffect

名称:

```
int ZPHOTO_ImageBlendEffect(unsigned char* baseData, int width, int height, int stride,
unsigned char* mixData, int blendMode);
```

接口描述: 图像图层混合

参数:

baseData--目标图像 Buffer, 作为结果图像输出;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

mixData--混合图像 Buffer, 必须与 baseData 大小相同;

blendMode--图层混合方法:

BLEND_MODE_DARKEN	变暗模式
BLEND_MODE_MULTIPLY	正片叠底模式
BLEND_MODE_COLORBURN	颜色加深模式
BLEND_MODE_LINEARBURN	线性渐变模式
BLEND_MODE_DARKNESS	深色模式
BLEND_MODE_LIGHTEN	变亮模式
BLEND_MODE_SCREEN	滤色模式
BLEND_MODE_COLORDODGE	颜色减淡模式
BLEND_MODE_COLORLINEARDODGE	颜色 线性减淡模式
BLEND_MODE_LIGHTCOLOR	浅色模式
BLEND_MODE_OVERLAY	叠加模式
BLEND_MODE_SOFTLIGHT	柔光
BLEND_MODE_HARDLIGHT	强光模式
BLEND_MODE_VIVIDLIGHT	亮光模式
BLEND_MODE_LINEARLIGHT	线性光模式
BLEND_MODE_PINLIGHT	点光模式
BLEND_MODE_SOLIDCOLORMIXING	实色混合模式
BLEND_MODE_DIFFERENCE	差值模式
BLEND_MODE_EXCLUSION	排除模式
BLEND_MODE_SUBTRACTION	减去模式
BLEND_MODE_DIVIDE	划分模式

返回值:

0-OK, 其他失败;

## 2.ZPHOTO\_ModeDarken

名称:

```
int ZPHOTO_ModeDarken(int basePixel, int mixPixel);
```

接口描述: 变暗图层混合模式

参数:

basePixel--底层像素值;

mixPixel--混合像素值;

返回值:

混合结果值, 范围为[0, 255];

## 3.ZPHOTO\_ModeMultiply

名称:

```
int ZPHOTO_ModeMultiply(int basePixel, int mixPixel);
```

接口描述: 正片叠底图层混合模式

参数:

basePixel--底层像素值;

mixPixel--混合像素值;

返回值:

混合结果值, 范围为[0, 255];

## 4.ZPHOTO\_ModeColorBurn

名称:

```
int ZPHOTO_ModeColorBurn(int basePixel, int mixPixel);
```

接口描述: 颜色加深图层混合模式

参数:

basePixel--底层像素值;

mixPixel--混合像素值;

返回值:

混合结果值, 范围为[0, 255];

## 5.ZPHOTO\_ModeLinearBurn

名称:

```
int ZPHOTO_ModeLinearBurn(int basePixel, int mixPixel);
```

接口描述: 线性渐变图层混合模式

参数:

basePixel--底层像素值;

mixPixel--混合像素值;

**返回值:**

混合结果值，范围为[0, 255];

## 6.ZPHOTO\_ModeDarkness

**名称:**

```
int ZPHOTO_ModeDarkness(int *baseRed, int *baseGreen, int *baseBlue, int mixRed, int mixGreen, int mixBlue);
```

**接口描述:** 深色图层混合模式

**参数:**

baseRed--底层像素 R 分量值索引，执行后修改为混合结果值;

baseGreen--底层像素 G 分量值索引，执行后修改为混合结果值;

baseBlue--底层像素 B 分量值索引，执行后修改为混合结果值;

mixRed--混合像素 R 分量值;

mixGreen--混合像素 G 分量值;

mixBlue--混合像素 B 分量值;

**返回值:**

0-OK，其他失败;

## 7.ZPHOTO\_ModeLighten

**名称:**

```
int ZPHOTO_ModeLighten(int basePixel, int mixPixel);
```

**接口描述:** 变亮图层混合模式

**参数:**

basePixel--底层像素值;

mixPixel--混合像素值;

**返回值:**

混合结果值，范围为[0, 255];

## 8.ZPHOTO\_ModeScreen

**名称:**

```
int ZPHOTO_ModeScreen(int basePixel, int mixPixel);
```

**接口描述:** 滤色图层混合模式

**参数:**

basePixel--底层像素值;

mixPixel--混合像素值;

**返回值:**

混合结果值，范围为[0, 255];

## 9.ZPHOTO\_ModeColorDodge

名称:

```
int ZPHOTO_ModeColorDodge(int basePixel, int mixPixel);
```

接口描述: 颜色减淡图层混合模式

参数:

basePixel--底层像素值;

mixPixel--混合像素值;

返回值:

混合结果值, 范围为[0, 255];

## 10.ZPHOTO\_ModeColorLinearDodge

名称:

```
int ZPHOTO_ModeColorLinearDodge(int basePixel, int mixPixel);
```

接口描述: 颜色线性减淡图层混合模式

参数:

basePixel--底层像素值;

mixPixel--混合像素值;

返回值:

混合结果值, 范围为[0, 255];

## 11.ZPHOTO\_ModeLightColor

名称:

```
int ZPHOTO_ModeLightColor(int basePixel, int mixPixel);
```

接口描述: 浅色图层混合模式

参数:

baseRed--底层像素 R 分量值索引, 执行后修改为混合结果值;

baseGreen--底层像素 G 分量值索引, 执行后修改为混合结果值;

baseBlue--底层像素 B 分量值索引, 执行后修改为混合结果值;

mixRed--混合像素 R 分量值;

mixGreen--混合像素 G 分量值;

mixBlue--混合像素 B 分量值;

返回值:

混合结果值, 范围为[0, 255];

## 12.ZPHOTO\_ModeOverlay

名称:

```
int ZPHOTO_ModeOverlay(int basePixel, int mixPixel);
```

接口描述: 叠加图层混合模式

**参数:**

basePixel--底层像素值;

mixPixel--混合像素值;

**返回值:**

混合结果值, 范围为[0, 255];

## 13.ZPHOTO\_ModeSoftLight

**名称:**

```
int ZPHOTO_ModeSoftLight(int basePixel, int mixPixel);
```

**接口描述:** 柔光图层混合模式

**参数:**

basePixel--底层像素值;

mixPixel--混合像素值;

**返回值:**

混合结果值, 范围为[0, 255];

## 14.ZPHOTO\_ModeHardLight

**名称:**

```
int ZPHOTO_ModeHardLight(int basePixel, int mixPixel);
```

**接口描述:** 强光图层混合模式

**参数:**

basePixel--底层像素值;

mixPixel--混合像素值;

**返回值:**

混合结果值, 范围为[0, 255];

## 15.ZPHOTO\_ModeVividLight

**名称:**

```
int ZPHOTO_ModeVividLight(int basePixel, int mixPixel);
```

**接口描述:** 亮光图层混合模式

**参数:**

basePixel--底层像素值;

mixPixel--混合像素值;

**返回值:**

混合结果值, 范围为[0, 255];

## 16.ZPHOTO\_ModeLinearLight

**名称:**

```
int ZPHOTO_ModeLinearLight(int basePixel, int mixPixel);
```

**接口描述：**线性光图层混合模式

**参数：**

basePixel--底层像素值；

mixPixel--混合像素值；

**返回值：**

混合结果值，范围为[0, 255]；

## 17.ZPHOTO\_ModePinLight

**名称：**

```
int ZPHOTO_ModePinLight(int basePixel, int mixPixel);
```

**接口描述：**点光图层混合模式

**参数：**

basePixel--底层像素值；

mixPixel--混合像素值；

**返回值：**

混合结果值，范围为[0, 255]；

## 18.ZPHOTO\_ModeSolidColorMixing

**名称：**

```
int ZPHOTO_ModeSolidColorMixing(int *baseRed, int *baseGreen, int *baseBlue, int mixRed, int mixGreen, int mixBlue);
```

**接口描述：**实色图层混合模式

**参数：**

baseRed--底层像素 R 分量值索引，执行后修改为混合结果值；

baseGreen--底层像素 G 分量值索引，执行后修改为混合结果值；

baseBlue--底层像素 B 分量值索引，执行后修改为混合结果值；

mixRed--混合像素 R 分量值；

mixGreen--混合像素 G 分量值；

mixBlue--混合像素 B 分量值；

**返回值：**

0-OK，其他失败；

## 19.ZPHOTO\_ModeDifference

**名称：**

```
int ZPHOTO_ModeDifference(int basePixel, int mixPixel);
```

**接口描述：**差值图层混合模式

**参数：**

basePixel--底层像素值；

mixPixel--混合像素值;

**返回值:**

混合结果值, 范围为[0, 255];

## 20.ZPHOTO\_ModeExclusion

**名称:**

`int ZPHOTO_ModeExclusion(int basePixel, int mixPixel);`

**接口描述:** 排除图层混合模式

**参数:**

basePixel--底层像素值;

mixPixel--混合像素值;

**返回值:**

混合结果值, 范围为[0, 255];

## 21.ZPHOTO\_ModeSubtraction

**名称:**

`int ZPHOTO_ModeSubtraction(int basePixel, int mixPixel);`

**接口描述:** 减去图层混合模式

**参数:**

basePixel--底层像素值;

mixPixel--混合像素值;

**返回值:**

混合结果值, 范围为[0, 255];

## 22.ZPHOTO\_ModeDivide

**名称:**

`int ZPHOTO_ModeDivide(int basePixel, int mixPixel);`

**接口描述:** 划分图层混合模式

**参数:**

basePixel--底层像素值;

mixPixel--混合像素值;

**返回值:**

混合结果值, 范围为[0, 255];

## 23.ZPHOTO\_ModeDesaturate

**名称:**

`int ZPHOTO_ModeDesaturate(int red, int green, int blue);`

**接口描述:** 去色模式



**参数:**

red--像素 R 分量值, 范围为[0, 255];  
green--像素 G 分量值, 范围为[0, 255];  
blue--像素 B 分量值, 范围为[0, 255];

**返回值:**

去色结果值, 范围为[0, 255];

## 24.ZPHOTO\_ModeColorInvert

**名称:**

```
int ZPHOTO_ModeColorInvert(int *red, int *green, int *blue);
```

**接口描述:** 反相模式

**参数:**

red--像素 R 分量值索引, 执行后为反相结果像素 R 分量值;  
green--像素 G 分量值索引, 执行后为反相结果像素 G 分量值;  
blue--像素 B 分量值索引, 执行后为反相结果像素 B 分量值;

**返回值:**

0-OK, 其他失败;

# 颜色空间转换接口说明

## 1.ZPHOTO\_RGBToYCbCr

名称:

```
void ZPHOTO_RGBToYCbCr(int R, int G, int B, int*Y, int*Cb, int* Cr);
```

接口描述: RGB 转 YCbCr 颜色空间

参数:

Red--像素 R 分量值, 范围为[0-255];

Green--像素 G 分量值, 范围为[0-255];

Blue--像素 B 分量值, 范围为[0-255];

Y--像素 Y 分量值;

Cb--像素 Cb 分量值;

Cr--像素 Cr 分量值;

注: 本接口中 YCbCr 为偏移之后的值, 已非浮点数。

返回值:

0-OK, 其他失败;

## 2.ZPHOTO\_YCbCrToRGB

名称:

```
void ZPHOTO_YCbCrToRGB(int Y, int Cb, int Cr, int*Red, int*Green, int* Blue);
```

接口描述: YCbCr 转 RGB 颜色空间

参数:

Y--像素 Y 分量值;

Cb--像素 Cb 分量值;

Cr--像素 Cr 分量值;

Red--像素 R 分量值, 范围为[0-255];

Green--像素 G 分量值, 范围为[0-255];

Blue--像素 B 分量值, 范围为[0-255];

注: 本接口中 YCbCr 为偏移之后的值, 已非浮点数。

返回值:

0-OK, 其他失败;

## 3.ZPHOTO\_RGBToLAB

名称:

```
void ZPHOTO_RGBToLab(int Red, int Green, int Blue, int* L, int *A, int *B);
```

接口描述: RGB 转 LAB 颜色空间

参数:

Red--像素 R 分量值, 范围为[0-255];

Green--像素 G 分量值, 范围为[0-255];

Blue--像素 B 分量值, 范围为[0-255];

L--像素 L 分量值, 范围为[0-255];

A--像素 A 分量值, 范围为[0-255];

B--像素 B 分量值, 范围为 0-255];

**返回值:**

0-OK, 其他失败;

注: 调用此接口前, 先调用接口 `void ZPHOTO_LABRGBInitial()`, 进行初始化, 该接口只需调用一次即可, 使用完毕后调用接口 `void ZPHOTO_LABRGBUnitial()` 进行销毁。

## 4.ZPHOTO\_LABToRGB

**名称:**

`void ZPHOTO_LabToRGB(int L, int A, int B, int* Red, int* Green, int* Blue);`

**接口描述:** LAB 转 RGB 颜色空间

**参数:**

L--像素 L 分量值, 范围为[0-255];

A--像素 A 分量值, 范围为[0-255];

B--像素 B 分量值, 范围为 0-255];

Red--像素 R 分量值, 范围为[0-255];

Green--像素 G 分量值, 范围为[0-255];

Blue--像素 B 分量值, 范围为[0-255];

**返回值:**

0-OK, 其他失败;

注: 调用此接口前, 先调用接口 `void ZPHOTO_LABRGBInitial()`, 进行初始化, 该接口只需调用一次即可, 使用完毕后调用接口 `void ZPHOTO_LABRGBUnitial()` 进行销毁。

# 滤镜引擎接口说明

## 1.ZPHOTO\_LUTFilter

**名称:**

```
int ZPHOTO_LUTFilter(unsigned char* srcData, int width, int height, int stride, unsigned char*Map, int ratio);
```

**接口描述:** LUT 滤镜

**参数:**

srcData--原始图像 Buffer;

width--图像宽度;

height--图像高度;

stride--图像 Stride;

Map--512X512LookupTable;

ratio---滤镜程度, 范围[0,100];

**返回值:**

0--OK, 其他失败;

## 错误异常返回

RET_OK	= 0	函数调用成功
RET_ERROR_MEMMORY	= - 100	内存申请错误
RET_ERROR_FILE	=-200	文件操作错误
RET_ERROR_PARAMETER	=-300	参数设置错误
RET_ERROR_THREAD	=-400	线程错误
RET_ERROR_UNKNOWN		未知错误类型

# Android API 接口说明

同 PC 版 API 说明，不同之处在于 Android API 接口使用 `native` 修饰；

使用说明：

- ①将 `libs` 库分别添加到对应的工程中；
- ②将 ZPHOTO 包放到 `android` 工程的 `src` 目录下；
- ③按照 API 说明进行调用即可；

使用说明详见 `ReadMe.txt`

# 图像读写

ZPhotoEngineIO 库为 ZPHOTOENGIEN 的附加库，负责图像读写，不依赖于任何第三方库，目前支持 BMP、JPEG(baseline)、PNG 三种图像格式。

类名：SF\_ImageReadWrite

## API 说明：

### 1.ReadImage

名称：

```
int ReadImage(char* fileName);
```

接口描述：读取图像

参数：

fileName--图像路径

返回值：

0-OK，其他失败；

### 2.SaveImage

名称：

```
int SaveImage(char* fileName, int imageFormat);
```

接口描述：保存图像

参数：

fileName--图像保存路径

imageFormat-图像格式

返回值：

0-OK，其他失败；

### 3.图像格式定义

```
typedef enum _sf_imageformat {SF_IMAGE_PNG, SF_IMAGE_BMP, SF_IMAGE_JPG, SF_IMAGE_GIF,  
SF_IMAGE_UNKNOWN} SFImageFormat;
```

### 4.像素格式定义

```
typedef enum  
_sf_image_pixelformat {SF_IMAGE_PIXELFORMAT_YUV420, SF_IMAGE_PIXELFORMAT_BGR24, SF_IMAGE_P  
IXELFORMAT_BGRA32, SF_IMAGE_PIXELFORMAT_RGB24, SF_IMAGE_PIXELFORMAT_RGBA32
```

```
}SFPixelFormat;
```

## 5.图像 SFBITMAP

```
typedef struct _tag_SFBITMAP{
    int pixelFormat;
    int lWidth;
    int lHeight;
    int lBitCounts;
    long lPitch[SF_IMAGE_MAX_PLANES];
    unsigned char* pPlaneData[SF_IMAGE_MAX_PLANES];

}SFBITMAP, *LPSFBITMAP;
```

## 6.IO 返回错误码

```
typedef enum _sf_ret {
    SF_RET_OK,
    SF_RET_ERR_PARA,
    SF_RET_ERR_MEM,
    SF_RET_ERR_NOT_SUPPORT    = 0x04,
    SF_RET_ERR_OPEN_FILE      = 0x08,
    SF_RET_ERR_DATA_UNINITED   = 0x10,
    SF_RET_ERR_UNKNOWN         = 0x20,
    SF_RET_ERR_END_OF_FILE      = 0x40,
    SF_RET_ERR_DATA_UNREADABLE = 0x101,
    SF_RET_ERR_DATA_NOWRITEABLE = 0x102,

}SF_RET;
```

## 7.SampleCode

```
int _tmain(int argc, _TCHAR* argv[])
{
    char* jpgPath = "C:\\Users\\Administrator\\Desktop\\1.jpg";
    char* outJpgPath = "C:\\Users\\Administrator\\Desktop\\Res.jpg";
    long start = clock();
    SF_ImageReadWrite* imgRW = new SF_ImageReadWrite();
    int ret = imgRW->ReadImage(jpgPath);
    if(ret == SF_RET_OK)
    {
        unsigned char* pSrc = imgRW->sfBitmap->pPlaneData[0];
```



## ZPHOTO ENGINE

```
int w = imgRW->sfBitmap->lWidth;
int h = imgRW->sfBitmap->lHeight;
unsigned char* tempData = (unsigned char*)malloc(sizeof(unsigned char) * w * h *
4);

memcpy(tempData, pSrc, sizeof(unsigned char) * w * h * 4);
//f_Test(pSrc, w, h, w * 4);
f_TFastGaussFilter(tempData, w, h, w * 4, pSrc, 5.0);
imgRW->SaveImage(outjpgPath, SF_IMAGE_JPG);
}
long end = clock();
imgRW->Destroy();
printf("JPG Time cost: %d ms.", (end - start));
return 0;
}
```

# 版本更新记录

20170807 V3.6

完成 PC/Android 版本递交

20180816V3.7

①ZPHOTO\_Saturation API

算法更新，新算法与 PS 参数范围一致，效果一致；

②新增 ZPHOTO\_SmartBlurFilter API 与 PS2018 效果接近

③新增 ZPHOTO\_AnisotropicFilter API

20180820V3.8

①新增 ZPHOTO\_DisplacementFilter API

置换滤镜，与 PS 效果接近

②新增 ZPHOTO\_NoiseEffect API

噪声效果

③ZPhotoEngineIO 库

提供 ZPhotoEngine 引擎的 BMP/JPG(Baseline)/PNG 图像读写操作类库，不依赖于 Opencv 等第三方库；