

IMAGE ENGINE

--VERSION 3.1

DATE: 20190101

CopyRight: www.xiusdk.cn

目录

前言.....	5
基础接口说明.....	6
1.ZPHOTO_SaturationAdjust.....	6
2.ZPHOTO_HueAndSaturationAdjust.....	6
3.ZPHOTO_LightnessAdjust.....	6
4.ZPHOTO_LinearBrightContrastAdjust.....	7
5.ZPHOTO_NLinearBrightContrastAdjust.....	7
6.ZPHOTO_AutoContrastAdjust.....	8
7.ZPHOTO_AutoColorGradationAdjust.....	8
8.ZPHOTO_CurveAdjust.....	8
9.ZPHOTO_Posterize.....	9
10.ZPHOTO_OverExposure.....	9
11.ZPHOTO_Invert.....	9
12.ZPHOTO_HistogramEqualize.....	10
13.ZPHOTO_Desaturate.....	10
14.ZPHOTO_Blackwhite.....	10
15.ZPHOTO_Threshold.....	11
16.ZPHOTO_FastGaussFilter.....	11
17.ZPHOTO_HighPass.....	11
18.ZPHOTO_USM.....	12
19.ZPHOTO_FindEdges.....	12
20.ZPHOTO_ChannelMixProcess.....	13
21.ZPHOTO_ColorTemperatureAdjust.....	13
22.ZPHOTO_ShadowAdjust.....	14
23.ZPHOTO_HighlightAdjust.....	14
24.ZPHOTO_ExposureAdjust.....	14
25.ZPHOTO_FastMeanFilter.....	15
26.ZPHOTO_SobelFilter.....	15
27.ZPHOTO_ImageTransformation.....	15
28.ZPHOTO_Fragment.....	16
29.ZPHOTO_MotionBlur.....	16
30.ZPHOTO_SurfaceBlur.....	17
31.ZPHOTO_RadialBlur.....	17
32.ZPHOTO_ZoomBlur.....	18
33.ZPHOTO_Relief.....	18
34.ZPHOTO_Mean.....	18
35.ZPHOTO_Mosaic.....	19
36.ZPHOTO_ColorBalance.....	19
37.ZPHOTO_Diffusion.....	19
38.ZPHOTO_LSNBlur.....	20
39.ZPHOTO_ColorLevelAdjust.....	20
40.ZPHOTO_MedianFilter.....	21
41.ZPHOTO_MaxFilter.....	21

42.ZPHOTO_MinFilter.....	21
43.ZPHOTO_VirtualFilter.....	22
44.ZPHOTO_GlowingEdges.....	22
45.ZPHOTO_ImageWarpPinch.....	23
46.ZPHOTO_ImageWarpWave.....	23
47.ZPHOTO_ImageWarpMagicMirror.....	23
48.ZPHOTO_ImageWarpZoom.....	24
49.ZPHOTO_RGBA2BGRA.....	24
50.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_RGBToYUV.....	34
2.ZPHOTO_YUVToRGB.....	34
3.ZPHOTO_RGBToYCbCr.....	34
4.ZPHOTO_YCbCrToRGB.....	35
5.ZPHOTO_RGBToXYZ.....	35
6.ZPHOTO_XYZToRGB.....	36
7.ZPHOTO_RGBToHSL.....	36
8.ZPHOTO_HSLToRGB.....	36
9.ZPHOTO_RGBToHSV.....	37

10.ZPHOTO_HSVToRGB.....	37
11.ZPHOTO_RGBToCMYK.....	37
12.ZPHOTO_CMYKToRGB.....	38
13.ZPHOTO_RGBToYDbDr.....	38
14.ZPHOTO_YDbDrToRGB.....	38
15.ZPHOTO_RGBToYIQ.....	39
16.ZPHOTO_YIQToRGB.....	39
17.ZPHOTO_RGBToLAB.....	40
18.ZPHOTO_LABToRGB.....	40
滤镜引擎接口说明.....	41
1.ZPHOTO_Filter.....	41
2.ZPHOTO_UniversalFilter.....	41
3.ZPHOTO_FilterId.....	42
错误异常返回.....	45

www.xiusdk.cn

前言



ImageEngine 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。同时, ImageEngine 还提供专业的滤镜特效, 万能滤镜设计接口, 方便不同开发者快速进行各种滤镜特效的开发。BeautyEngine 还将包含人脸化妆算法接口, 以满足美颜相机等类似化妆应用的开发使用。

ImageEngine 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, 主要提供了各种常用颜色空间和 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。该部分和第一、二部分构成 ImageEngine 引擎, 可单独使用。该引擎提供相应的开发 DEMO, 供使用者查阅。

第四部分滤镜特效, 模拟实现 Instagram/美图秀秀/Camera360 等大多数经典滤镜, 并包含其他常用滤镜(冲印, 怀旧, 阿宝色等)功能。另外滤镜引擎提供了万能滤镜接口, 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。

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

最后, 我们将不断对该引擎提供算法更新, 以满足不同开发者的需求。

我们的网址: www.xiusdk.cn

基础接口说明

1.ZPHOTO_SaturationAdjust

名称:

`int ZPHOTO_SaturationAdjust(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[0, 255];

返回值:

0-OK, 其他失败;

2.ZPHOTO_HueAndSaturationAdjust

名称:

`int ZPHOTO_HueAndSaturationAdjust(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;

3.ZPHOTO_LightnessAdjust

名称:

`int ZPHOTO_LightnessAdjust(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;

4.ZPHOTO_LinearBrightContrastAdjust

名称:

```
int ZPHOTO_LinearBrightContrastAdjust(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[-255, 255];
threshold--调节阈值, 范围为[0, 255], 默认值 128;

返回值:

0-OK, 其他失败;

5.ZPHOTO_NLinearBrightContrastAdjust

名称:

```
int ZPHOTO_NLinearBrightContrastAdjust(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;
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[-255, 255];
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_AutoContrastAdjust

名称:

```
int ZPHOTO_AutoContrastAdjust(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;
width--图像宽度;
height--图像高度;
stride--图像 Stride;

返回值:

0-OK, 其他失败;

7.ZPHOTO_AutoColorGradationAdjust

名称:

```
int ZPHOTO_AutoColorGradationAdjust(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;
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_CurveAdjust

名称:

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

接口描述: (曲线)色阶调节

参数:

srcData--原始图像 Buffer;
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[0,255], 默认 128;

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;

9.ZPHOTO_Posterialize

名称:

`int ZPHOTO_Posterialize(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;
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;

10.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;
width--图像宽度;
height--图像高度;
stride--图像 Stride;

返回值:

0-OK, 其他失败;

11.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;
width--图像宽度;

height--图像高度;
stride--图像 Stride;

返回值:

0-OK, 其他失败;

12.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;
width--图像宽度;
height--图像高度;
stride--图像 Stride;

返回值:

0-OK, 其他失败;

13.ZPHOTO_Desaturate

名称:

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

接口描述: 去色

参数:

srcData--原始图像 Buffer;
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_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;
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;

15.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;
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_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;
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;

17.ZPHOTO_HighPass

名称:

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

接口描述: 高反差保留

参数:

srcData--原始图像 Buffer;
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;
mRadius--高斯半径, 范围为[0,1000];

返回值:

0-OK, 其他失败;

18.ZPHOTO_USM

名称:

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

接口描述: USM 锐化

参数:

srcData--原始图像 Buffer;
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];
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;

19.ZPHOTO_FindEdges

名称:

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

接口描述: 查找边缘

参数:

srcData--原始图像 Buffer;
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;

返回值:

0-OK, 其他失败;

20.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;

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;

21.ZPHOTO_ColorTemperatureAdjust

名称:

```
int ZPHOTO_ColorTemperatureAdjust(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;

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;

22.ZPHOTO_ShadowAdjust

名称:

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

接口描述: 阴影调节

参数:

srcData--原始图像 Buffer;
width--图像宽度;
height--图像高度;
stride--图像 Stride;
intensity--阴影强度值, 取值范围为[0, 100];

返回值:

0-OK, 其他失败;

23.ZPHOTO_HighlightAdjust

名称:

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

接口描述: 高光调节

参数:

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

返回值:

0-OK, 其他失败;

24.ZPHOTO_ExposureAdjust

名称:

```
int ZPHOTO_ExposureAdjust(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;
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;

25.ZPHOTO_FastMeanFilter

名称:

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

接口描述: 高光调节

参数:

srcData--原始图像 Buffer;
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--均值滤波半径, 取值范围为[0, width / 2];

返回值:

0-OK, 其他失败;

26.ZPHOTO_SobelFilter

名称:

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

接口描述: Sobel 边缘滤波

参数:

srcData--原始图像 Buffer;
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;

返回值:

0-OK, 其他失败;

27.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;
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;

28.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 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;

29.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 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;
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;

30.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 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;
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;

31.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 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;
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;

32.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 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;
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;

33.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 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;
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;

34.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 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;

35.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 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;
size--马赛克半径, 范围为[0-200];

返回值:

0-OK, 其他失败;

36.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 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;
cyan--青色调整, 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;

37.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 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;
intensity--扩散程度, 范围为[0-100];

返回值:

0-OK, 其他失败;

38.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 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;
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;

39.ZPHOTO_ColorLevelAdjust

名称:

```
int ZPHOTO_ColorLevelAdjust(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;
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;

40.ZPHOTO_MedianFilter

名称:

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

接口描述: 中值滤波(中间色)

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;
radius--模糊半径, 范围为[0, 100];

返回值:

0-OK, 其他失败;

41.ZPHOTO_MaxFilter

名称:

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

接口描述: 最大值滤波(最大值)

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;
radius--模糊半径, 范围为[0, 100];

返回值:

0-OK, 其他失败;

42.ZPHOTO_MinFilter

名称:

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

接口描述: 最小值滤波(最小值)

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;
radius--模糊半径, 范围为[0, 100];

返回值:

0-OK, 其他失败;

43.ZPHOTO_VirtualFilter

名称:

`int ZPHOTO_VirtualFilter(unsigned char* srcData, int width, int height, int stride, int x, int y, int blurIntensity, int radius);`

接口描述: 虚化滤镜

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;
x--虚化中心 x 坐标
y--虚化中心 y 坐标
blurIntensity--模糊程度, 范围[1, 100];
radius--虚化半径, 范围为[0, +];

返回值:

0-OK, 其他失败;

44.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 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;
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;

45.ZPHOTO_ImageWarpPinch

名称:

```
int ZPHOTO_ImageWarpPinch(unsigned char *srcData, int width, int height, int stride, int cenX, int cenY, int intensity);
```

接口描述: 挤压变形

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;

cenX--变形中心点 Y 坐标值;

cenY--变形中心点 X 值;

intensity--变形强度, 取值范围为[10, 20];

返回值:

0-OK, 其他失败;

46.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 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;

intensity--变形强度, 取值范围为[0, 100];

返回值:

0-OK, 其他失败;

47.ZPHOTO_ImageWarpMagicMirror

名称:

```
int ZPHOTO_ImageWarpMagicMirror(unsigned char* srcData, int width, int height, int stride, int cenX, int cenY, int radius);
```

接口描述: 哈哈镜变形(鱼眼变形)

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;
cenX--变形中心点 Y 坐标值;
cenY--变形中心点 X 值;
radius--变形强度, 取值范围为[0, 500];

返回值:

0-OK, 其他失败;

48.ZPHOTO_ImageWarpZoom

名称:

`int ZPHOTO_ImageWarpZoom(unsigned char* srcData, int width, int height, int stride, int cenX, int cenY, int radius, float unit);`

接口描述: 哈哈镜变形(鱼眼变形)

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;
cenX--变形中心点 Y 坐标值;
cenY--变形中心点 X 值;
radius--变形强度, 取值范围为[0, 500];
unit--缩放因子, 范围[0,10], 小于 1 为缩小, 大于 1 为放大效果, 等于 1 为鱼眼;

返回值:

0-OK, 其他失败;

49.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;

50.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;

www.xiusdk.cn

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_RGBToYUV

名称:

`void ZPHOTO_RGBToYUV(int Red, int Green, int Blue, int* Y, int* U, int* V);`

接口描述: RGB 转 YUV 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;

U--像素 U 分量值;

V--像素 V 分量值;

注: 本接口中 YUV 为偏移之后的值, 已非浮点数。

返回值:

0-OK, 其他失败;

2.ZPHOTO_YUVToRGB

名称:

`void ZPHOTO_YUVToRGB(int Y, int U, int V, int* Red, int* Green, int* Blue);`

接口描述: YUV 转 RGB 颜色空间

参数:

Y--像素 Y 分量值;

U--像素 U 分量值;

V--像素 V 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 YUV 为偏移之后的值, 已非浮点数。

返回值:

0-OK, 其他失败;

3.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;

4.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;

5.ZPHOTO_RGBToXYZ

名称:

```
void ZPHOTO_RGBToXYZ(int Red, int Green, int Blue, int* X, int* Y, int* Z);
```

接口描述: RGB 转 XYZ 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];

X--像素 X 分量值;

Y--像素 Y 分量值;

Z--像素 Z 分量值;

注: 本接口中 XYZ 为偏移之后的值, 已非浮点数。

返回值:

0-OK, 其他失败;

6.ZPHOTO_XYZToRGB

名称:

```
void ZPHOTO_XYZToRGB(int X, int Y, int Z, int* Red, int* Green, int* Blue);
```

接口描述: XYZ 转 RGB 颜色空间

参数:

X—像素 X 分量值;

Y—像素 Y 分量值;

Z—像素 Z 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 XYZ 为偏移之后的值, 已非浮点数。

返回值:

0—OK, 其他失败;

7.ZPHOTO_RGBToHSL

名称:

```
void ZPHOTO_RGBToHSL (int Red, int Green, int Blue, int* h, int* s, int* l);
```

接口描述: RGB 转 HSL 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];

h—像素 h 分量值, 范围为[0-360];

s—像素 s 分量值, 范围为[0-1.0];

l—像素 l 分量值, 范围为[0-1.0];

返回值:

0—OK, 其他失败;

8.ZPHOTO_HSLToRGB

名称:

```
void ZPHOTO_HSLToRGB (int h, int s, int l, int* Red, int* Green, int* Blue);
```

接口描述: HSL 转 RGB 颜色空间

参数:

h—像素 h 分量值, 范围为[0-360];

s—像素 s 分量值, 范围为[0-1.0];

l—像素 l 分量值, 范围为[0-1.0];

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;

9.ZPHOTO_RGBToHSV

名称:

`void ZPHOTO_RGBToHSV (int Red, int Green, int Blue, double* h, double* s, double* v);`

接口描述: RGB 转 HSV 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];

h--像素 h 分量值, 范围为[0-360];

s--像素 s 分量值, 范围为[0-1.0];

v--像素 v 分量值, 范围为[0-1.0];

返回值:

0-OK, 其他失败;

10.ZPHOTO_HSVToRGB

名称:

`void ZPHOTO_HSVToRGB (double h, double s, double v, int* Red, int* Green, int* Blue);`

接口描述: HSV 转 RGB 颜色空间

参数:

h--像素 h 分量值, 范围为[0-360];

s--像素 s 分量值, 范围为[0-1.0];

v--像素 v 分量值, 范围为[0-1.0];

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;

11.ZPHOTO_RGBToCMYK

名称:

`void ZPHOTO_RGBToCMYK (int Red, int Green, int Blue, int* C, int* M, int* Y, int * K);`

接口描述: RGB 转 CMYK 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];

C--像素 C 分量值, 范围为[0-512];

M--像素 M 分量值, 范围为[0-512];

Y--像素 Y 分量值, 范围为[0-512];

K--像素 K 分量值, 范围为[0-512];

返回值:

0-OK, 其他失败;

12.ZPHOTO_CMYKToRGB

名称:

`void ZPHOTO_CMYKToRGB (int C, int M, int Y, int K, int* Red, int* Green, int* Blue);`

接口描述: CMYK 转 RGB 颜色空间

参数:

C--像素 C 分量值, 范围为[0-512];

M--像素 M 分量值, 范围为[0-512];

Y--像素 Y 分量值, 范围为[0-512];

K--像素 K 分量值, 范围为[0-512];

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;

13.ZPHOTO_RGBToYDbDr

名称:

`void ZPHOTO_RGBToYDbDr (int Red, int Green, int Blue, int* Y, int* Db, int* Dr);`

接口描述: RGB 转 YDbDr 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, 范围为[0-255];

Db--像素 Db 分量值, 范围为[0-255];

Dr--像素 Dr 分量值, 范围为[0-255];

返回值:

0-OK, 其他失败;

14.ZPHOTO_YDbDrToRGB

名称:

```
void ZPHOTO_YDbDrToRGB (int Y, int Db, int Dr, int* Red, int* Green, int* Blue);
```

接口描述: YDbDr 转 RGB 颜色空间

参数:

- Y—像素 Y 分量值, 范围为[0-255];
- Db—像素 Db 分量值, 范围为[0-255];
- Dr—像素 Dr 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;

15.ZPHOTO_RGBToYIQ

名称:

```
void ZPHOTO_RGBToYIQ (int Red, int Green, int Blue, double* Y, double* I, double* Q);
```

接口描述: RGB 转 YIQ 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, 范围为[0-255];
- I—像素 I 分量值, 范围为[-0.5957-0.5957];
- Q—像素 Q 分量值, 范围为[-0.5226-0.5226];

返回值:

- 0—OK, 其他失败;

16.ZPHOTO_YIQToRGB

名称:

```
void ZPHOTO_YIQToRGB (double Y, double I, double Q, int* Red, int* Green, int* Blue);
```

接口描述: YIQ 转 RGB 颜色空间

参数:

- Y—像素 Y 分量值, 范围为[0-255];
- I—像素 I 分量值, 范围为[-0.5957-0.5957];
- Q—像素 Q 分量值, 范围为[-0.5226-0.5226];
- 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;

17.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。

18.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_Filter

名称:

```
int ZPHOTO_Filter(unsigned char* srcData, int width, int height, int stride, int filterId)
```

接口描述: 滤镜

参数:

srcData--原始图像 Buffer;
width--图像宽度;
height--图像高度;
stride--图像 Stride;
filterId--滤镜效果 ID, 详情请看附录;

返回值:

0-OK, 其他失败;

2.ZPHOTO_UniversalFilter

名称:

```
int ZPHOTO_UniversalFilter(unsigned char* srcData, int width, int height, int stride, unsigned char* maskData, int mWidth, int mHeight, int mStride, int mergeMode, int ratio);
```

接口描述: 万能滤镜

参数:

srcData--原始图像 Buffer;
width--图像宽度;
height--图像高度;
stride--图像 Stride;
maskData--模板图像 Buffer;
mWidth--模板图像宽度;
mHeight--模板图像高度;
mStride--模板图像 Stride;
mergeMode--图层混合模式:

```
const int BLEND_MODE_DARKEN           = 1;  
const int BLEND_MODE_MULTIPLY         = 2;  
const int BLEND_MODE_COLORBURN       = 3;  
const int BLEND_MODE_LINEARBURN      = 4;  
const int BLEND_MODE_DARKNESS        = 5;  
const int BLEND_MODE_LIGHTEN         = 6;  
const int BLEND_MODE_SCREEN           = 7;  
const int BLEND_MODE_COLORDODGE      = 8;  
const int BLEND_MODE_COLORLINEARDODGE = 9;
```

```
const int BLEND_MODE_LIGHTCOLOR           = 10;
const int BLEND_MODE_OVERLAY              = 11;
const int BLEND_MODE_SOFTLIGHT           = 12;
const int BLEND_MODE_HARDLIGHT           = 13;
const int BLEND_MODE_VIVIDLIGHT          = 14;
const int BLEND_MODE_LINEARLIGHT         = 15;
const int BLEND_MODE_PINLIGHT            = 16;
const int BLEND_MODE_SOLIDCOLORMIXING    = 17;
const int BLEND_MODE_DIFFERENCE          = 18;
const int BLEND_MODE_EXCLUSION           = 19;
const int BLEND_MODE_SUBTRACTION         = 20;
const int BLEND_MODE_DIVIDE              = 21;
```

ratio--滤镜效果程度, 范围为[0-100];

返回值:

0-OK, 其他失败;

3.ZPHOTO_FilterId

名称: (附录)滤镜效果 Id 列表

////////////////////////////////////模拟实现 Instagram 滤镜效果////////////////////////////////////

```
FILTER_IDA_NONE
FILTER_IDA_1977
FILTER_IDA_INKWELL
FILTER_IDA_KELVIN
FILTER_IDA_NASHVILLE
FILTER_IDA_VALENCIA
FILTER_IDA_XPROII
FILTER_IDA_BRANNAN
FILTER_IDA_WALDEN
FILTER_IDA_ADEN
FILTER_IDA_ASHBY
FILTER_IDA_BROOKLYN
FILTER_IDA_CHARMES
FILTER_IDA_CLARENDON
FILTER_IDA_CREMA
FILTER_IDA_DOGPACH
FILTER_IDA_GINGHAM
FILTER_IDA_GINZA
FILTER_IDA_HEFE
FILTER_IDA_HELENA
FILTER_IDA_JUNO
FILTER_IDA_LARK
FILTER_IDA_LUDWIG
FILTER_IDA_MAVEN
```


FILTER_IDC_MAPLELEAF	//LOMO_枫叶
FILTER_IDC_COOLFLAME	//LOMO_冷焰
FILTER_IDC_WARMAUTUMN	//LOMO_暖秋
FILTER_IDC_CYAN	//LOMO_青色
FILTER_IDC_ZEAL	//LOMO_热情
FILTER_IDC_FASHION	//LOMO_时尚
FILTER_IDC_EKTAR	//弗莱胶片 -- Ektar
FILTER_IDC_GOLD	//弗莱胶片 -- Gold
FILTER_IDC_VISTA	//弗莱胶片 -- Vista
FILTER_IDC_XTAR	//弗莱胶片 -- Xtra
FILTER_IDC_RUDDY	//魔法美肤 -- 红润
FILTER_IDC_SUNSHINE	//魔法美肤 -- 暖暖阳光
FILTER_IDC_FRESH	//魔法美肤 -- 清新丽人
FILTER_IDC_SWEET	//魔法美肤 -- 甜美可人
FILTER_IDC_BLACKWHITE	//魔法美肤 -- 艺术黑白
FILTER_IDC_WHITENING	//魔法美肤 -- 自然美白
FILTER_IDC_JPELEGANT	//日系 -- 淡雅
FILTER_IDC_JPJELLY	//日系 -- 果冻
FILTER_IDC_JPFRESH	//日系 -- 清新
FILTER_IDC_JPSWEET	//日系 -- 甜美
FILTER_IDC_JPAESTHETICISM	//日系 -- 唯美
FILTER_IDC_JPWARM	//日系 -- 温暖
////////////////////其他滤镜效果////////////////////	
FILTER_IDD_CARTOON	//卡通
FILTER_IDD_DARK	//暗调
FILTER_IDD_GLOW	//Glow
FILTER_IDD_LOMO	//LOMO
FILTER_IDD_NEON	//霓虹
FILTER_IDD_OILPAINT	//油画
FILTER_IDD_PUNCH	//冲印
FILTER_IDD_REMINISCENT	//怀旧
FILTER_IDD_SKETCH	//素描
FILTER_IDD_GRAPHIC	//连环画
FILTER_IDD_ABAOSE	//阿宝色
FILTER_IDD_LEGE	//乐高像素拼图
FILTER_IDD_LANGMAN	//浪漫
FILTER_IDD_LUOZHUANG	//裸妆
FILTER_IDD_NENHONG	//嫩红
FILTER_IDD_QINGTOU	//清透
FILTER_IDD_ZHENBAI	//臻白
FILTER_IDD_ZIRAN	//自然

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

www.xiusdk.cn