

BEAUTY ENGINE

--VERSION 3.1

DATE: 20190101

CopyRight: www.xiusdk.cn

目录

美颜引擎接口说明.....	3
Beauty_InitBeautyEngine.....	3
Beauty_UninitBeautyEngine.....	3
Beauty_ProcessBuffer.....	3
Beauty_ProcessNV21.....	4
Beauty_ClearParams.....	4
Beauty_SetFacePoints.....	4
Beauty_SetSoftenRatio.....	4
Beauty_SetWhitenRatio.....	5
Beauty_SetFilterID.....	5
Beauty_SetFilterRatio.....	5
Beauty_SetDefreckleAutoEnable.....	6
Beauty_SetDefreckleManualRadiusAndPosition.....	6
Beauty_SetEyeBagRemoveRatio.....	6
Beauty_SetLightEyeRatio.....	6
Beauty_SetEyeWarpRatio.....	7
Beauty_SetFaceLiftRatio.....	7
Beauty_SetLipsColorRatio.....	7
Beauty_SetLipsColor.....	7
Beauty_SetHighnoseRatio.....	8
(附录)滤镜效果 Id 列表.....	8
异常列表文档.....	10
1. 返回值类型列表.....	10

美颜引擎接口说明

Beauty_InitBeautyEngine

名称:

ZBEAUTY_API BeautyHandle Beauty_InitBeautyEngine(const char* filePath);

接口描述: 创建美颜引擎

参数:

filePath--资源文件的路径

返回值:

BeautyHandle--美颜引擎实例句柄

Beauty_UninitBeautyEngine

名称:

ZBEAUTY_API void Beauty_UninitBeautyEngine(BeautyHandle handle);

接口描述: 销毁美颜引擎

参数:

handle--实例句柄

返回值:

void

Beauty_ProcessBuffer

名称:

ZBEAUTY_API int Beauty_ProcessBuffer(BeautyHandle handle, unsigned char *srcData, int width, int height, int stride);

接口描述: 美颜处理接口

参数:

handle--实例句柄

srcData--原始图像 Buffer, 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

Beauty_ProcessNV21

名称:

ZBEAUTY_API int Beauty_ProcessNV21(BeautyHandle handle, unsigned char *yuv, int width, int height);

接口描述: 美颜处理接口

参数:

handle--实例句柄
yuv--原始图像 Buffer, nv21 格式
width--图像宽度
height--图像高度

返回值:

0--OK, 其他失败

Beauty_ClearParams

名称:

ZBEAUTY_API void Beauty_ClearParams(BeautyHandle handle);

接口描述: 美颜相关参数清零

参数:

handle--实例句柄

返回值:

void

Beauty_SetFacePoints

名称:

ZBEAUTY_API void Beauty_SetFacePoints(BeautyHandle handle, int faceCount, int points[]);

接口描述: 设置人脸关键点, 磨皮/美白/滤镜的功能不依赖于人脸关键点, 其他祛斑/大眼/瘦脸等功能要用到人脸关键点信息, 如果不设置人脸关键点, 这些功能将不起作用。

人脸关键点生成可以调用我们的 facedetect engine 生成 101 个人脸关键点

参数:

handle--实例句柄
faceCount--人脸个数
points--人脸关键点坐标数组

返回值:

void

Beauty_SetSoftenRatio

名称:

ZBEAUTY_API void Beauty_SetSoftenRatio(BeautyHandle handle, int softenRatio);

接口描述: 磨皮功能设置

参数:

handle--实例句柄

softenRatio--磨皮强度, 取值 0-100, 0 关闭该功能, 参考值 60

返回值:

void

Beauty_SetWhitenRatio

名称:

ZBEAUTY_API void Beauty_SetWhitenRatio(BeautyHandle handle, int whitenRatio);

接口描述: 美白功能设置

参数:

handle--实例句柄

whitenRatio--美白强度, 取值 0-100, 0 关闭该功能, 参考值 30

返回值:

void

Beauty_SetFilterID

名称:

ZBEAUTY_API void Beauty_SetFilterID(BeautyHandle handle, int filterID);

接口描述: 滤镜效果功能设置

参数:

handle--实例句柄

filterID--滤镜功能 id, 取值 0-36, 参考附录 id 效果列表

返回值:

void

Beauty_SetFilterRatio

名称:

ZBEAUTY_API void Beauty_SetFilterRatio(BeautyHandle handle, int filterRatio);

接口描述: 滤镜强度功能设置

参数:

handle--实例句柄

filterRatio--滤镜强度, 取值 0-100, 0 关闭该功能, 参考值 80

返回值:

void

Beauty_SetDefreckleAutoEnable

名称:

ZBEAUTY_API void Beauty_SetDefreckleAutoEnable(BeautyHandle handle, int defreckleAutoEnable);

接口描述: 自动祛斑功能设置

参数:

handle--实例句柄

defreckleAutoEnable--祛斑功能开启 0 或 关闭 1

返回值:

void

Beauty_SetDefreckleManualRadiusAndPosition

名称:

ZBEAUTY_API void Beauty_SetDefreckleManualRadiusAndPosition(BeautyHandle handle, int defreckleRadius, int defrecklePosition[]);

接口描述: 手动祛斑功能设置

参数:

handle--实例句柄

defreckleRadius--祛除斑点半径, 0 关闭该功能, 参考值 5

defrecklePosition--祛除斑点位置, 两个值, x 和 y 坐标

返回值:

void

Beauty_SetEyeBagRemoveRatio

名称:

ZBEAUTY_API void Beauty_SetEyeBagRemoveRatio(BeautyHandle handle, int eyeBagRemoveRatio);

接口描述: 去眼袋功能设置

参数:

handle--实例句柄

eyeBagRemoveRatio--去眼袋强度, 取值 0-100, 0 关闭该功能, 参考值 60

返回值:

void

Beauty_SetLightEyeRatio

名称:

ZBEAUTY_API void Beauty_SetLightEyeRatio(BeautyHandle handle, int lightEyeRatio);

接口描述: 亮眼功能设置

参数:

handle--实例句柄

lightEyeRatio--亮眼强度, 取值 0-100, 0 关闭该功能, 参考值 60

返回值:

void

Beauty_SetEyeWarpRatio

名称:

ZBEAUTY_API void Beauty_SetEyeWarpRatio(BeautyHandle handle, int eyeWarpRatio);

接口描述: 大眼功能设置

参数:

handle--实例句柄

eyeWarpRatio--大眼强度, 取值 0-100, 0 关闭该功能, 参考值 60

返回值:

void

Beauty_SetFaceLiftRatio

名称:

ZBEAUTY_API void Beauty_SetFaceLiftRatio(BeautyHandle handle, int faceLiftRatio);

接口描述: 瘦脸功能设置

参数:

handle--实例句柄

faceLiftRatio--瘦脸强度, 取值 0-100, 0 关闭该功能, 参考值 60

返回值:

void

Beauty_SetLipsColorRatio

名称:

ZBEAUTY_API void Beauty_SetLipsColorRatio(BeautyHandle handle, int lipsColorRatio);

接口描述: 唇彩功能设置

参数:

handle--实例句柄

lipsColorRatio--唇彩强度, 取值 0-100, 0 关闭该功能, 参考值 60

返回值:

void

Beauty_SetLipsColor

名称:

ZBEAUTY_API void Beauty_SetLipsColor(BeautyHandle handle, int lipsColor[3]);

接口描述: 唇彩颜色设置

参数:

handle--实例句柄

lipsColor--唇彩颜色, rgb 数组值

返回值:

void

Beauty_SetHighnoseRatio

名称:

ZBEAUTY_API void Beauty_SetHighnoseRatio(BeautyHandle handle, int highnoseRatio);

接口描述: 挺鼻梁功能设置

参数:

handle--实例句柄

highnoseRatio--挺鼻梁强度, 取值 0-100, 0 关闭该功能, 参考值 60

返回值:

void

(附录)滤镜效果 Id 列表

```
//// filterId
//const int RT_FILTER_ID_BEAUTY = 0;
//const int RT_FILTER_BEAUTY_CLEAR = RT_FILTER_ID_BEAUTY + 1;
//const int RT_FILTER_BEAUTY_WHITESKINNED = RT_FILTER_ID_BEAUTY + 2;
//const int RT_FILTER_BEAUTY_COOL = RT_FILTER_ID_BEAUTY + 3;
//const int RT_FILTER_BEAUTY_ICESPIRIT = RT_FILTER_ID_BEAUTY + 4;
//const int RT_FILTER_BEAUTY_REFINED = RT_FILTER_ID_BEAUTY + 5;
```



```
//const int RT_FILTER_BEAUTY_BLUESTYLE = RT_FILTER_ID_BEAUTY + 6;
//const int RT_FILTER_BEAUTY_LOLITA = RT_FILTER_ID_BEAUTY + 7;
//const int RT_FILTER_BEAUTY_LKK = RT_FILTER_ID_BEAUTY + 8;
//const int RT_FILTER_BEAUTY_NUANHUANG = RT_FILTER_ID_BEAUTY + 9;
//const int RT_FILTER_BEAUTY_RCOOL = RT_FILTER_ID_BEAUTY + 10;
//const int RT_FILTER_BEAUTY_JSTYLE = RT_FILTER_ID_BEAUTY + 11;
//const int RT_FILTER_BEAUTY_SOFTLIGHT = RT_FILTER_ID_BEAUTY + 12;
//const int RT_FILTER_BEAUTY_TIANMEI = RT_FILTER_ID_BEAUTY + 13;
//const int RT_FILTER_BEAUTY_WEIMEI = RT_FILTER_ID_BEAUTY + 14;
//const int RT_FILTER_BEAUTY_FRESH = RT_FILTER_ID_BEAUTY + 15;
//const int RT_FILTER_BEAUTY_JPJELLY = RT_FILTER_ID_BEAUTY + 16;
//const int RT_FILTER_BEAUTY_HUAYAN = RT_FILTER_ID_BEAUTY + 17;
//const int RT_FILTER_BEAUTY_LUOZHUANG = RT_FILTER_ID_BEAUTY + 18;
//const int RT_FILTER_BEAUTY_NENHONG = RT_FILTER_ID_BEAUTY + 19;
//const int RT_FILTER_BEAUTY_BLACKWHITE = RT_FILTER_ID_BEAUTY + 20;
//const int RT_FILTER_BEAUTY_WHITENING = RT_FILTER_ID_BEAUTY + 21;
//const int RT_FILTER_BEAUTY_RUDDY = RT_FILTER_ID_BEAUTY + 22;
//const int RT_FILTER_BEAUTY_JPAESTHETICISM = RT_FILTER_ID_BEAUTY + 23;
//const int RT_FILTER_BEAUTY_PURPLEDREAM = RT_FILTER_ID_BEAUTY + 24;
//const int RT_FILTER_BEAUTY_JPELEGANT = RT_FILTER_ID_BEAUTY + 25;
//const int RT_FILTER_BEAUTY_JPFRESH = RT_FILTER_ID_BEAUTY + 26;
//const int RT_FILTER_BEAUTY_JPSWEET = RT_FILTER_ID_BEAUTY + 27;
//const int RT_FILTER_BEAUTY_JPWARM = RT_FILTER_ID_BEAUTY + 28;
//const int RT_FILTER_BEAUTY_SUNSHINE = RT_FILTER_ID_BEAUTY + 29;
//const int RT_FILTER_BEAUTY_SWEET = RT_FILTER_ID_BEAUTY + 30;
//const int RT_FILTER_BEAUTY_ABAOSE = RT_FILTER_ID_BEAUTY + 31;
//const int RT_FILTER_BEAUTY_LANGMAN = RT_FILTER_ID_BEAUTY + 32;
//const int RT_FILTER_BEAUTY_QINGTOU = RT_FILTER_ID_BEAUTY + 33;
//const int RT_FILTER_BEAUTY_ZHENBAI = RT_FILTER_ID_BEAUTY + 34;
//const int RT_FILTER_BEAUTY_ZIRAN = RT_FILTER_ID_BEAUTY + 35;
//const int RT_FILTER_BEAUTY_WARMER = RT_FILTER_ID_BEAUTY + 36;
```

异常列表文档

1. 返回值类型列表

1. BEAUTY_OK = 0 成功返回
2. BEAUTY_E_MEMMORY = -1 内存错误异常
3. BEAUTY_E_FILE = -2 文件读写异常
4. BEAUTY_E_PARAMETER = -3 参数错误异常
5. BEAUTY_E_IMAGEFORMAT = -4 线程错误异常
6. BEAUTY_E_UNKNOWN 其他异常