ArcSoft Age Estimation

开发指导文档

ArcSoft Corporation
46601 Fremont Blvd.
Fremont, CA 94538
http://www.arcsoft.com

Trademark or Service Mark Information

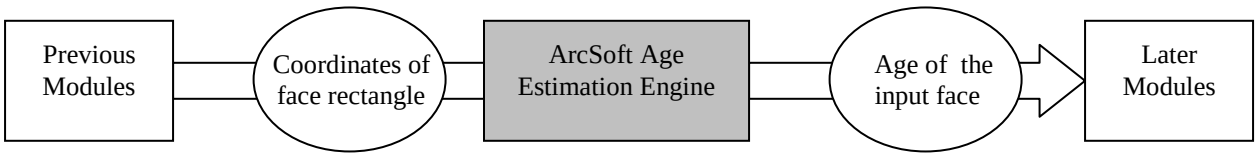
ArcSoft Inc. and ArcWare are registered trademarks of ArcSoft Inc.

Other product and company names mentioned herein may be trademarks and/or service marks of their respective owners. The absence of a trademark or service mark from this list does not constitute a waiver of ArcSoft Inc.'s trademark or other intellectual property rights concerning that trademark or service mark.

The information contained in this document is for discussion purposes only. None of the information herein shall be interpreted as an offer or promise to any of the substance herein nor as an agreement to contract or license, or as an implication of a transfer of rights. Any and all terms herein are subject to change at the discretion of ArcSoft. Copying, distributing, transferring or any other reproduction of these documents or the information contained herein is expressly prohibited, unless such activity is expressly permitted by an authorized representative of ArcSoft, Inc.

Chapter 1: 概述

虹软年龄检测引擎工作流程图.



1.1. 运行环境

- Windows

1.2. 系统要求

- 32 位系统，Windows7 以上

1.3. 依赖库

- None

Chapter 2: 结构与常量

2.1. 基本类型

所有基本类型在平台库中有定义。 定义规则是在 ANSIC 中的基本类型前加上字母“M”同时将类型的第一个字母改成大写。例如“long” 被定义成“MLong”

2.2. 数据结构

2.2.1. ASAE_FSDK_Version

描述

SDK 版本信息

定义

```
typedef struct{
	MInt32 lCodebase;
	MInt32 lMajor;
	MInt32 lMinor;
	MInt32 lBuild;
	MPChar Version;
	MPChar BuildDate;
	MPChar CopyRight;
} ASAE_FSDK_Version;
```

成员变量

lCodebase	代码库版本号
lMajor	主版本号
lMinor	次版本号
lBuild	编译版本号，递增
Version	字符串形式的版本号
BuildDate	编译时间
CopyRight	版权信息

2.2.2. ASAE_FSDK_AGERESULT

描述

定义年龄检测结果信息

定义

```
typedef struct{
        MInt32          * pAgeResultArray;
        MInt32          lFaceNumber;
} ASAE_FSDK_AGERESULT, *LPASAE_FSDK_AGERESULT;
```

成员变量

pAgeResultArray	检测出的年龄结果数组
lFaceNumber	检测出的年龄结果个数

2.2.3. ASAE _FSDK_AGEFACEINPUT

描述

定义脸部信息

定义

```
Typedef struct{
        MRECT           *pFaceRectArray;
        MInt32          *pFaceOrientArray;
        MInt32          lFaceNumber;
} ASAE_FSDK_AGEFACEINPUT, *LPASAE_FSDK_AGEFACEINPUT;
```

成员变量

pFaceRectArray	人脸框信息数组
pFaceOrientArray	输入的人脸角度数组。
lFaceNumber	人脸个数

2.3. 枚举

2.3.1. ASAE_FSDK_AgeOrientCode

描述

定义基于逆时针方向的人脸角度

定义

```
enum ASAE_FSDK_AgeOrientCode {
                ASAE_FSDK_FOC_Age_0         = 0x1,
                ASAE_FSDK_FOC_Age_90        = 0x2,
```

```
                ASAE_FSDK_FOC_Age_270      = 0x3,
                ASAE_FSDK_FOC_Age_180      = 0x4,
                ASAE_FSDK_FOC_Age_30       = 0x5,
                ASAE_FSDK_FOC_Age_60       = 0x6,
                ASAE_FSDK_FOC_Age_120      = 0x7,
                ASAE_FSDK_FOC_Age_150      = 0x8,
                ASAE_FSDK_FOC_Age_210      = 0x9,
                ASAE_FSDK_FOC_Age_240      = 0xa,
                ASAE_FSDK_FOC_Age_300      = 0xb,
                ASAE_FSDK_FOC_Age_330      = 0xc
};
```

成员变量

ASAE_FSDK_FOC_Age_0	0 度
ASAE_FSDK_FOC_Age_90	90 度
ASAE_FSDK_FOC_Age_270	270 度
ASAE_FSDK_FOC_Age_180	180 度
ASAE_FSDK_FOC_Age_30	30 度
ASAE_FSDK_FOC_Age_60	60 度
ASAE_FSDK_FOC_Age_120	120 度
ASAE_FSDK_FOC_Age_150	150 度
ASAE_FSDK_FOC_Age_210	210 度
ASAE_FSDK_FOC_Age_240	240 度
ASAE_FSDK_FOC_Age_300	300 度
ASAE_FSDK_FOC_Age_330	330 度

2.3.2. 支持的颜色格式

描述

颜色格式及其对齐规则

定义

ASVL_PAF_I420	8-bit Y 层，之后是 8-bit 的 2x2 采样的 U 层和 V 层
ASVL_PAF_YUYV	Y0, U0, Y1, V0
ASVL_PAF_RGB24_B8G8R8	BGR24, B8G8R8

Chapter 3: API 说明

3.1. ASAE_FSDK_InitAgeEngine

原型

```
MRESULT ASAE_FSDK_InitAgeEngine (
	MPChar			AppId,
	MPChar			SDKKey,
	MByte			*pMem,
	MInt32			lMemSize
	MHandle			*phEngine
);
```

描述

初始化年龄检测引擎

参数

AppId	[in]	用户申请 SDK 时获取的 App Id
SDKKey	[in]	用户申请 SDK 时获取的 SDK Key
pMem	[in]	分配给引擎使用的内存地址
lMemSize	[in]	分配给引擎使用的内存大小
phEngine	[out]	引擎 handle

返回值

成功返回 MOK，否则返回失败 code。失败 codes 如下所列:

MERR_INVALID_PARAM	参数输入非法
MERR_NO_MEMORY	内存不足

3.2. ASAE_FSDK_AgeEstimation_StaticImage

原型

```
MRESULT ASAE_FSDK_AgeEstimation_StaticImage (
	MHandle				hEngine,
	LPASVLOFFSCREEN			pImginfo,
	LPASAE_FSDK_AGEFACEINPUT	pFaceRes,
	LPASAE_FSDK_AGERESULT		pAgeRes,
);
```

描述

检测静态图片中人物的年龄

参数

hEngine	[in]	引擎 handle
pImginfo	[in]	输入的图像数据
pFaceRes	[in]	已检测到的脸部信息
pAgeRes	[out]	年龄检测结果

返回值

成功返回 MOK，否则返回失败 code。失败 codes 如下所列:

MERR_INVALID_PARAM	参数输入非法
MERR_NO_MEMORY	内存不足

3.3. ASAE_FSDK_AgeEstimation_Preview

原型

```
MRESULT ASAE_FSDK_AgeEstimation_Preview (
      MHandle                           hEngine,
      LPASVLOFFSCREEN                   pImginfo,
      LPASAE_FSDK_AGEFACEINPUT          pFaceRes,
      LPASAE_FSDK_AGERESULT             pAgeRes,
);
```

描述

检测动态视频中人物的年龄

参数

hEngine	[in]	引擎 handle
pImginfo	[in]	输入图像信息
pFaceRes	[in]	输入的图像中人脸信息， 需要事先用人脸引擎检测出
pAgeRes	[out]	年龄检测结果

返回值

成功返回 MOK，否则返回失败 code。失败 codes 如下所列:

MERR_INVALID_PARAM	参数输入非法
MERR_NO_MEMORY	内存不足

3.4. ASAE_FSDK_UninitAgeEngine

原型

```
MRESULT ASAE_FSDK_UninitAgeEngine (
      MHandle      hEngine
);
```

描述

销毁引擎，释放相应资源

参数

hEngine	[in]	引擎 handle

返回值

成功返回 MOK，否则返回失败 code。失败 codes 如下所列:

MERR_INVALID_PARAM　　参数输入非法

3.5. ASAE_FSDK_GetVersion

原型

```
const ASAE_FSDK_Version* ASAE_FSDK_GetVersion(
      MHandle      hEngine
);
```

描述

获取 SDK 版本信息

参数

hEngine	[in]	引擎 handle

Chapter 4: Sample Codes

注意,使用时请替换申请的 **APPID SDKKEY**

```
#include <stdio.h>
#include "arcsoft_fsdk_age_estimation.h"
#include "merror.h"

#pragma comment(lib,"libarcsoft_fsdk_age_estimation.lib")

#define MAXIMUM_FACE_NUMBER 10
#define WORKBUF_SIZE        (30*1024*1024)
#define APPID       ""                  //APPID
#define ASAE_SDKKey  ""                 //SDKKey

#define AGE_ESTIMATION_STATICIMAGE
#define AGE_ESTIMATION_PREVIEW

/* define global variables for age estimation */
MHandle                         AgeEngine = nullptr;
ASVLOFFSCREEN                   AgeImageInfo = { 0 };
ASAE_FSDK_AGEFACEINPUT          AgeFaceInput;
ASAE_FSDK_AGERESULT             AgeResult;
MByte *                         pWorkMem = nullptr;

/* initialize the engine and other variables */
MInt32 InitAgeEstimate()
{
	MInt32 res = MOK;
	AgeFaceInput.lFaceNumber = 0;
	AgeFaceInput.pFaceRectArray = new MRECT[MAXIMUM_FACE_NUMBER];
	if (0 == AgeFaceInput.pFaceRectArray)
		return MERR_NO_MEMORY;
	AgeFaceInput.pFaceOrientArray = new MInt32[MAXIMUM_FACE_NUMBER];
	if (0 == AgeFaceInput.pFaceOrientArray)
		return MERR_NO_MEMORY;

	pWorkMem = new MByte[WORKBUF_SIZE];
	res = ASAE_FSDK_InitAgeEngine(APPID, ASAE_SDKKey, pWorkMem, WORKBUF_SIZE,
&AgeEngine);
	return res;
}
/* release the engine and other memory handles */
MInt32 UnInitAgeEstimate()
{
	MInt32 res = MOK;
	res = ASAE_FSDK_UninitAgeEngine(AgeEngine);
	if (AgeFaceInput.pFaceRectArray != nullptr)
	{
		delete[] AgeFaceInput.pFaceRectArray;
		AgeFaceInput.pFaceRectArray = nullptr;
	}
	if (AgeFaceInput.pFaceOrientArray != nullptr)
	{
```

```
			delete[] AgeFaceInput.pFaceOrientArray;
			AgeFaceInput.pFaceOrientArray = nullptr;
		}
		if (pWorkMem != nullptr)
		{
			delete[] pWorkMem;
			pWorkMem = nullptr;
		}
		return res;
}
/* print sdk version */
void PrintVersionInfo()
{
		const ASAE_FSDK_Version * pVersionInfo = nullptr;
		pVersionInfo = ASAE_FSDK_GetVersion(AgeEngine);
		printf("%d %d %d %d\n", pVersionInfo->lCodebase, pVersionInfo->lMajor,
pVersionInfo->lMinor, pVersionInfo->lBuild);
		printf("%s\n", pVersionInfo->Version);
		printf("%s\n", pVersionInfo->BuildDate);
		printf("%s\n", pVersionInfo->CopyRight);
}

#ifdef AGE_ESTIMATION_STATICIMAGE
/* load static image and save it to "pImageInfo", which is a "ASVLOFFSCREEN"
struct. return "MOK" if succeed. */
MInt32 GetImageData(ASVLOFFSCREEN * pImageInfo)
{
		MInt32 res = MOK;

		/* load image. add your code here */
		/* ... ... */

		return res;
}

/* using ArcSoft face detection sdk to detect faces in the input image and save
the face results to "pFaceInput".
return "MOK" if succeed. */
MInt32 FaceDetect(ASVLOFFSCREEN * pImageInfo, ASAE_FSDK_AGEFACEINPUT * pFaceInput)
{
		MInt32 res = MOK;

		/* add your code here */
		/* ... ... */

		return res;
}
#endif


#ifdef AGE_ESTIMATION_PREVIEW
/* get each preview frame and save it to "pImageInfo", which is a "ASVLOFFSCREEN"
struct. return "MOK" if succeed. */
MInt32 GetPreviewData(ASVLOFFSCREEN * pImageInfo)
{
		MInt32 res = MOK;
```

```
	/* get frame data. add your code here */
	/* ... ... */

	return res;
}

/* using ArcSoft face tracking library to detect faces in the input image and save
the face results to "pFaceInput".
return "MOK" if succeed. */
MInt32 FaceTrack(ASVLOFFSCREEN * pImageInfo, ASAE_FSDK_AGEFACEINPUT * pFaceInput)
{
	MInt32 res = MOK;

	/* add your code here */
	/* ... ... */

	return res;
}
#endif

/* estimate age  */
MInt32 AgeEstimate()
{
	MInt32 res = MOK;
#ifdef AGE_ESTIMATION_STATICIMAGE
	res = ASAE_FSDK_AgeEstimation_StaticImage(AgeEngine, &AgeImageInfo,
&AgeFaceInput, &AgeResult);
#endif

#ifdef AGE_ESTIMATION_PREVIEW
	res = ASAE_FSDK_AgeEstimation_Preview(AgeEngine, &AgeImageInfo,
&AgeFaceInput, &AgeResult);
#endif

	return res;
}

int main()
{
	/* initialize the engine and other variables */
	MRESULT nRet = MERR_UNKNOWN;
	nRet = InitAgeEstimate();
	if (nRet != MOK)
	{
		printf("InitAgeEngine failed , errorcode is %d \n", nRet);
		return -1;
	}

	/* print version info */
	PrintVersionInfo();

#ifdef AGE_ESTIMATION_STATICIMAGE
	/* image data acquisition */
	nRet = GetImageData(&AgeImageInfo);
	if (nRet != MOK)
	{
		printf("GetImageData: nRet=%d\n", nRet);
```

```
            return nRet;
        }

        /* face detection */
        nRet = FaceDetect(&AgeImageInfo, &AgeFaceInput);
        if (nRet != MOK)
        {
            printf("FaceDetect: nRet=%d\n", nRet);
            return nRet;
        }

        /* do age estimation when the face number is bigger than 0. */
        if (AgeFaceInput.lFaceNumber > 0)
        {
            /* age estimation */
            nRet = AgeEstimate();
            if (nRet != MOK)
            {
                printf("AgeEstimate: nRet=%d\n", nRet);
                return nRet;
            }

            /* print age estimation result */
            for (int i = 0; i < AgeFaceInput.lFaceNumber; i++)
            {
                printf("Age: %d, ", AgeResult.pAgeResultArray[i]);
            }
            printf("\n");
        }
#endif

#ifdef AGE_ESTIMATION_PREVIEW
        /* preview data acquisition */
        while (MOK == GetPreviewData(&AgeImageInfo))
        {
            /* face tracking */
            nRet = FaceTrack(&AgeImageInfo, &AgeFaceInput);
            if (nRet != MOK)
            {
                printf("FaceDetect: nRet=%d\n", nRet);
                return nRet;
            }

            /* age estimation */
            /* no matter "AgeFaceInput->lFaceNumber" is greater than or equal to
0, next sentence must be excuted. */
            nRet = AgeEstimate();
            if (nRet != MOK)
            {
                printf("AgeEstimate: nRet=%d\n", nRet);
                return nRet;
            }

            /* print age estimation result */
            if (AgeFaceInput.lFaceNumber > 0)
            {
                for (int i = 0; i < AgeFaceInput.lFaceNumber; i++)
```

```
                    {
                        printf("Age: %d, ", AgeResult.pAgeResultArray[i]);
                    }
                    printf("\n");
                }
        }
#endif

        /* release the engine and other memory handles */
        nRet = UnInitAgeEstimate();
        return nRet;
}
```