



采集库接口函数说明

1.1 文件格式

1.1.1 文件生成

接口函数采用 Windows 动态链接库(DLL)形式发布。支持的操作系统应包括 Windows XP 或兼容系统。
接口函数应在动态链接库中输出，支持多线程。程序编译为 32 位版本。

1.1.2 文件名称

动态链接库文件名称为：ID_FprCap.dll。

1.2 接口函数

接口函数见表 1.1。

表 1.1 接口函数

编号	名称	说明
1	LIVESCAN_Init	初始化采集器
2	LIVESCAN_Close	释放采集器
3	LIVESCAN_GetChannelCount	获得采集器通道数量
4	LIVESCAN_SetBright	设置采集器当前的亮度
5	LIVESCAN_SetContrast	设置采集器当前对比度
6	LIVESCAN_GetBright	获得采集器当前的亮度
7	LIVESCAN_GetContrast	获得采集器当前对比度
8	LIVESCAN_GetMaxImageSize	获得采集器采集图像的宽度、高度的最大值
9	LIVESCAN_GetCaptWindow	获得当前图像的采集位置、宽度和高度
10	LIVESCAN_SetCaptWindow	设置当前图像的采集位置、宽度和高度
11	LIVESCAN_Setup	调用采集器的属性设置对话框
12	LIVESCAN_BeginCapture	准备采集一帧图像
13	LIVESCAN_GetFPRawData	采集一帧图像
14	LIVESCAN_GetFPBmpData	采集一帧BMP格式图像数据
15	LIVESCAN_EndCapture	结束采集一帧图像
16	LIVESCAN_IsSupportSetup	采集器是否支持设置对话框
17	LIVESCAN_GetVersion	取得接口规范的版本
18	LIVESCAN_GetDesc	获得接口规范的说明
19	LIVESCAN_GetErrorInfo	获得采集接口错误信息
20	LIVESCAN_SetBufferEmpty	设置存放采集数据的内存块为空

1.3 函数说明

1.3.1 初始化采集器

函数原型：int __stdcall LIVESCAN_Init()。

参数：无。

返回值：调用成功返回 1。

否则返回错误代码，调用 LIVESCAN_GetErrInfo 函数获取错误信息。错误代码值符合 1.4 的要求。

说明：初始化采集器，分配相应的资源，检查授权等。在所有接口函数中，通常首先调用此函数。

只需要调用一次，允许重复调用，其结果与一次调用相同。



1.3.2 释放采集器

函数原型: `int __stdcall LIVESCAN_Close()`。

参数: 无。

返回值: 调用成功返回 1。

否则返回错误代码, 调用 `LIVESCAN_GetErrInfo` 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 关闭指纹采集器, 释放分配的资源, 调用此函数后, 在没有调用 `LIVESCAN_Init` 函数之前,

仅可调用 `LIVESCAN_GetVersion`、`LIVESCAN_GetDesc` 和 `LIVESCAN_GetErrorInfo` 函数。调用此函数后, 应重新调用 `LIVESCAN_Init` 来初始化采集器。

1.3.3 获得采集器通道数量

函数原型: `int __stdcall LIVESCAN_GetChannelCount()`。

参数: 无。

返回值: 调用成功返回通道数量 (>0)。

否则返回错误代码, 调用 `LIVESCAN_GetErrInfo` 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 获得采集器可以使用的通道数量及通道号。

1.3.4 设置采集器当前的亮度

函数原型: `int __stdcall LIVESCAN_SetBright(int nChannel,int nBright)`。

参数: `int nChannel` 通道号。输入参数。

`int nBright` 亮度, 范围为 0~255, 输入参数。

返回值: 调用成功返回 1。

否则返回错误代码, 调用 `LIVESCAN_GetErrInfo` 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 设置采集器当前亮度。

1.3.5 设置采集器当前对比度

函数原型: `int __stdcall LIVESCAN_SetContrast(int nChannel,int nContrast)`。

参数: `int nChannel` 通道号。输入参数。

`int nContrast` 对比度, 范围 0~255。输入参数。

返回值: 调用成功返回 1。

否则返回错误代码, 调用 `LIVESCAN_GetErrInfo` 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 设置采集器当前对比度。

1.3.6 获得采集器当前的亮度

函数原型: `int __stdcall LIVESCAN_GetBright(int nChannel,int *pnBright)`。

参数: `int nChannel` 通道号。输入参数。

`int *pnBright` 存放当前亮度的整形指针。输出参数。

返回值: 调用成功返回 1。

否则返回错误代码, 调用 `LIVESCAN_GetErrInfo` 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 获得采集器当前亮度。

1.3.7 获得采集器当前对比度

函数原型: `int __stdcall LIVESCAN_GetContrast(int nChannel,int *pnContrast)`。

参数: `int nChannel` 通道号。输入参数。

`int *pnContrast` 存放当前对比度的整形指针。输出参数。

返回值: 调用成功返回 1。



否则返回错误代码,调用 LIVESCAN_GetErrInfo 函数获取错误信息。错误代码值符合 1.4 的要求。
说明: 获得采集器当前对比度。

1.3.8 获得采集器可采集图像的宽度、高度的最大值

函数原型: int __stdcall LIVESCAN_GetMaxImageSize (int nChannel,int *pnWidth,int *pnHeight)。

参数: int nChannel 通道号。输入参数。

int *pnWidth 存放图像宽度的整形指针。输出参数。

int *pnHeight 存放图像高度的整形指针。输出参数。

返回值: 调用成功返回 1。

否则返回错误代码,调用 LIVESCAN_GetErrInfo 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 获得采集器可采集图像的宽度、高度的最大值。

1.3.9 获得采集器当前图像的采集位置、宽度和高度

函数原型: int __stdcall LIVESCAN_GetCaptWindow(int nChannel,int *pnOriginX,int *pnOriginY,int *pnWidth,int *pnHeight)。

参数: int nChannel 通道号。输入参数。

int *pnOriginX 存放图像采集窗口的采集原点坐标 X 值的整型指针。输出参数。

int *pnOriginY 存放图像采集窗口的采集原点坐标 Y 值的整型指针。输出参数。

int *pnWidth 存放采集图像宽度的整形指针。输出参数。

int *pnHeight 存放采集图像高度的整形指针。输出参数。

返回值: 调用成功返回 1。

否则返回错误代码,调用 LIVESCAN_GetErrInfo 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 获得采集器当前图像的采集位置、宽度和高度。当前图像宽度初始值为 256, 高度初始值为 360。

1.3.10 设置采集器当前图像的采集位置、宽度和高度

函数原型: int __stdcall LIVESCAN_SetCaptWindow(int nChannel,int nOriginX,int nOriginY,int nWidth, int nHeight)。

参数: int nChannel 通道号。输入参数。

int nOriginX 图像采集窗口的采集原点坐标 X 值。输入参数。

int nOriginY 图像采集窗口的采集原点坐标 Y 值。输入参数。

int nWidth 采集图像的宽度。对于居民身份证用单指指纹采集, 应大于等于 256。否则应返回参数错误代码。输入参数。

int nHeight 采集图像的高度。对于居民身份证用单指指纹采集, 应大于等于 360。否则应返回参数错误代码。输入参数。

返回值: 调用成功返回 1。

否则返回错误代码, 调用 LIVESCAN_GetErrInfo 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 设置采集器当前图像的采集位置、宽度和高度。

1.3.11 调用采集器的属性设置对话框

函数原型: int __stdcall LIVESCAN_Setup()。

参数: 无。

返回值: 调用成功返回 1。

否则返回错误代码, 调用 LIVESCAN_GetErrInfo 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 此函数弹出一个模式对话框, 用户可以设置除去对比度、亮度、采集窗口参数外的其它参数, 如 GAMMA 值等, 使得设置适合采集器本身的特点。

1.3.12 准备采集一帧图像



函数原型: int __stdcall LIVESCAN_BeginCapture(int nChannel)。

参数: int nChannel 通道号。输入参数。

返回值: 调用成功返回 1。

否则返回错误代码, 调用 LIVESCAN_GetErrInfo 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 采集图像的一个前缀函数, 使得采集器有机会进行另外一个采集之前的初始化工作。

1.3.13 采集一帧图像

函数原型: int __stdcall LIVESCAN_GetFPRawData(int nChannel, unsigned char *pRawData)。

参数: int nChannel 通道号。输入参数。

unsigned char *pRawData 指向存放采集数据的内存块, 调用者分配。

返回图像数据, 大小应为: 当前图像采集宽度×当前图像采集高度。输出参数。

返回值: 调用成功返回 1。

否则返回错误代码, 调用 LIVESCAN_GetErrInfo 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 采集一帧图像。

1.3.14 采集一帧 BMP 格式图像

函数原型: int __stdcall LIVESCAN_GetFPBmpData(int nChannel, unsigned char *pBmpData)。

参数: int nChannel 通道号。输入参数。

unsigned char* pBmpData 指向存放 8 位灰度 BMP 格式采集数据的内存块, 调用者分配。

返回 8 位灰度 BMP 格式图像数据。大小应为: 当前图像采集宽度×当前图像采集高度+1078。

输出参数。

返回值: 调用成功返回 1。

否则返回错误代码, 调用 LIVESCAN_GetErrInfo 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 采集一帧 8 位灰度 BMP 格式图像。

1.3.15 结束采集一帧图像

函数原型: int __stdcall LIVESCAN_EndCapture(int nChannel)。

参数: int nChannel 通道号。输入参数。

返回值: 调用成功返回 1。

否则返回错误代码, 调用 LIVESCAN_GetErrInfo 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 结束采集一帧图像或预览图像。

1.3.16 采集器是否支持设置对话框

函数原型: int __stdcall LIVESCAN_IsSupportSetup()。

参数: 无。

返回值: 若采集接口支持 LIVESCAN_Setup, 则返回 1, 否则返回 0。

否则返回错误代码, 调用 LIVESCAN_GetErrInfo 函数获取错误信息。错误代码值符合 1.4 的要求。

说明: 此函数用来确认是否支持设置对话框。

1.3.17 取得接口规范的版本

函数原型: int __stdcall LIVESCAN_GetVersion()。

参数: 无。

返回值: 获得接口规范的版本。当前版本为 1.00, 返回值为 100。若以后扩展接口, 则需要修改此版本号。

否则返回错误代码, 调用 LIVESCAN_GetErrorInfo 函数获取错误信息。错误代码值应符合 1.4 的要求。



说明：获得接口版本号。

1.3.18 获得接口规范的说明

函数原型：int __stdcall LIVESCAN_GetDesc(char pszDesc[1024])。
参数：char pszDesc[1024] 存放接口说明，其中 pszDesc[0]用于存储采集器代码，缺省为 FFH。
输出参数。
返回值：调用成功返回 1。
否则返回错误代码，调用 LIVESCAN_GetErrorInfo 函数获取错误信息。错误代码值应符合 1.4 的要求
说明：获得接口说明，不需要初始化就可以调用。pszDesc 以数值 0 结尾的字符串，采用 GB 13000 中规定的字符。

1.3.19 获得采集接口错误信息

函数原型：int __stdcall LIVESCAN_GetErrorInfo(int nErrorNo,char pszErrorInfo[256])。
参数：char pszErrorInfo[256] 用来存放错误信息的内存块，错误信息的长度不能超过 256 个字节。输出参数。
int nErrorNo 错误代码(<0)。输入参数。
返回值：若为合法的错误代码返回 1，pszErrorInfo 中为错误信息。若 nErrorNo 为非法的错误代码，则返回-6，同时设置 pszErrorInfo 为“非法错误号”错误。
说明：pszErrorInfo 采用 GB 13000 中规定的字符。

1.3.20 设置存放采集数据的内存块为空

函数原型：int __stdcall LIVESCAN_SetBufferEmpty(unsigned char *pImageData,long imageLength) 。
参数：unsigned char *pImageData 指向存放采集数据的内存块。输入参数。
Long imageLength 存放采集数据的内存块长度。输入参数。
返回值：调用成功返回 1。
否则返回错误代码,调用 LIVESCAN_GetErrInfo 函数获取错误信息。错误代码值符合 1.4 的要求。
说明：将存放采集数据的内存块中的每一个字节的值置为 0x00。

1.4 错误代码

错误代码见表 1.2。

表 1.2 错误代码表

错误代码	说明
-1	参数错误
-2	内存分配失败，没有分配到足够的内存
-3	功能未实现
-4	设备不存在
-5	设备未初始化
-6	非法错误号
-7	其它错误

指纹数据结构

2.1 指纹特征数据

2.1.1 指纹中心点

指纹中心点标注方法应符合 GA 774.3—2008 的要求。

指纹中心点数据由两部分构成：

- a) 横坐标：x，有效值为 0 ~ 255；
- b) 纵坐标：y，有效值为 0 ~ 359。

每个指纹中心点数据用 3 个字节表示，见图 2.1。第 1 个字节，存放指纹中心点横坐标（x）的全部 8 位值；第 2 个字节，存放指纹中心点纵坐标（y）的低 8 位值；第 3 个字节，第 1 位存放指纹中心点纵坐标（y）的最高位，其余位备用，备用位设置为 1。

指纹中心点数据以 3 个字节为单位顺次排列。

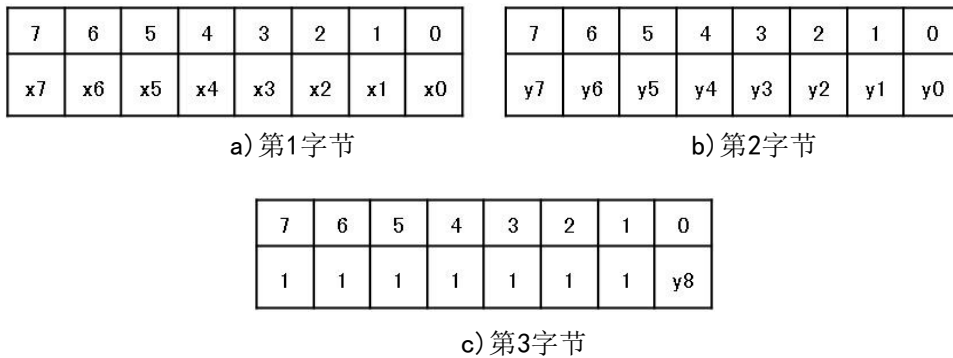


图 2.1 指纹中心点数据表示方法

2.1.2 指纹细节特征点

指纹细节特征点标注方法应符合 GA 774.5—2008 的要求。

指纹细节特征点数据由三部分构成：

- a) 横坐标：x，有效值为 0 ~ 255；
- b) 纵坐标：y，有效值为 0 ~ 359；
- c) 方 向：d，有效值为 0 ~ 359。

每个指纹细节特征点用 4 个字节表示，见图 2.2。第 1 个字节，存放指纹细节特征点横坐标（x）的全部 8 位值；第 2 个字节，存放指纹细节特征点纵坐标（y）的低 8 位值；第 3 个字节，存放指纹细节特征点方向（d）的低 8 位值；第 4 个字节，第 1 位存放指纹细节特征点纵坐标（y）的最高位，第 2 位存放指纹细节特征点方向（d）的最高位，其余位备用，备用位设置为 1。

指纹细节特征点数据以 4 个字节为单位顺次排列。

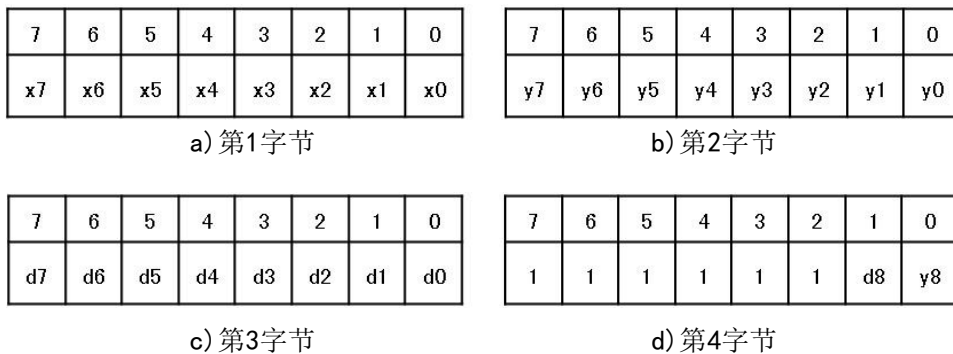


图 2.2 指纹细节特征点数据表示方法

2.2 指纹特征坐标系

指纹特征坐标系参照 GA 775—2008，见图 2.3。以指纹图像的左上角为坐标原点（0，0），X 轴坐标值从左到右增加，Y 轴坐标值从上至下增加，X、Y 值以像素为单位，X 大于等于 0 且小于 256，Y 大于等于 0 且小于 360。矢量角度以度为单位，用非负整数表示，与 X 轴平行并随 X 增大的方向为 0°，逆时针转动时角度增加，角度范围为 0°~359°。

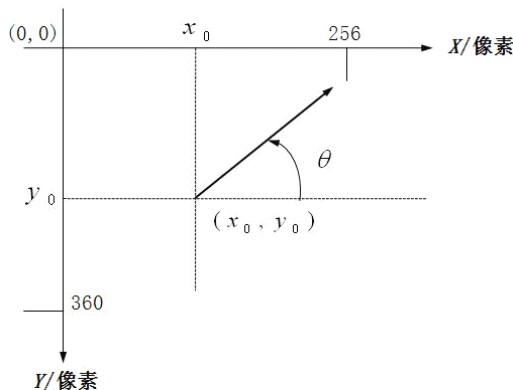


图 2.3 指纹特征坐标系

2.3 指位代码

指位代码参照 GA 777.1—2010，采用两位数字字符，转换为 1 个字节的十六进制数值后存放，见表 2.1。

表 2.1 指位代码表

代码	指位名称	转换后的数值
11	右手拇指	0BH
12	右手食指	0CH
13	右手中指	0DH
14	右手环指	0EH
15	右手小指	0FH
16	左手拇指	10H
17	左手食指	11H
18	左手中指	12H
19	左手环指	13H
20	左手小指	14H
97	右手不确定指位	61H
98	左手不确定指位	62H
99	其他不确定指位	63H

2.4 指纹特征数据文件结构

指纹特征数据文件结构见表 2.2，示例见表 2.3。

表 2.2 指纹特征数据文件结构

序号	字段	相对位置	长度（字节）	说明
1	字段 1	1	1	文件头类型，记录字符“C”（43H）
2	字段 2	2	1	指纹算法版本号
3	字段 3	3	1	指纹采集器代码
4	字段 4	4	1	指纹算法开发者代码
5	字段 5	5	1	注册结果代码，01H 表示“注册成功”，02H 表示“注册失败”，03H 表示“未注册”，09H 表示“未知”
6	字段 6	6	1	指位代码
7	字段 7	7	1	指纹质量值，以00H~64H来表示，值01H表示最低质量，值64H表示最高质量，值00H表示“未知”
8	字段 8	8~19	12	保留项
9	字段 9	20	1	指纹细节特征点数量。“注册失败”和“未注册”指纹设置为 00H
10	字段 10	21~22	2	指纹特征数据长度，以字节为单位
11	字段 11	23~31	9	指纹中心点数据
12	字段 12	32~511	480	指纹细节特征点数据
13	字段 13	512	1	校验位，验位值采用 CRC-8 生成

表 2.3 指纹特征数据文件结构示例

序号	字段	相对位置	值（十六进制）	说明
1	字段 1	1	13	“C”表示指纹特征数据
2	字段 2	2	03	指纹算法版本号为 3
3	字段 3	3	11	指纹采集器代码为 17
4	字段 4	4	0A	指纹算法开发者代码为 10
5	字段 5	5	01	“注册成功”
6	字段 6	6	10	左手拇指
7	字段 7	7	50	指纹质量值为 80
8	字段 8	8~19	FF FF FF FF FF FF FF FF FF FF FF FF	保留项
9	字段 9	20	33	指纹细节特征点数量为 51
10	字段 10	21~22	00 D5	指纹特征数据长度为 213 个字节
11	字段 11	23~31	--	指纹中心点数据
12	字段 12	32~511	--	指纹细节特征点数据
13	字段 13	512	6D	CRC 校验位

2.5 指纹压缩图像数据文件结构

指纹压缩图像数据文件结构见表 2.4，示例见表 2.5。

表 2.4 指纹压缩图像数据文件结构

序号	字段	相对位置	长度（字节）	说明
1	字段 1	1	1	文件头类型，记录字符“Y”（59H）

2	字段 2	2	1	指纹算法版本号
3	字段 3	3	1	指纹采集器代码
4	字段 4	4	1	指纹算法开发者代码
5	字段 5	5	1	注册结果代码，01H 表示“注册成功”，02H 表示“注册失败”，03H 表示“未注册”，09H 表示“未知”
6	字段 6	6	1	指位代码
7	字段 7	7~20	14	保留项
8	字段 8	21~22	2	指纹压缩图像数据长度，以字节为单位
9	字段 9	23~	X	指纹压缩图像数据
10	字段 10	Y	1	校验位，验位值采用 CRC-8 生成。Y=23+X

表 2.5 指纹压缩图像数据文件结构示例

序号	字段	相对位置	值（十六进制）	说明
1	字段 1	1	59	“Y” 表示指纹压缩图像数据
2	字段 2	2	03	指纹算法版本号为3
3	字段 3	3	11	指纹采集器代码为17
4	字段 4	4	0A	指纹算法开发者代码为10
5	字段 5	5	01	“注册成功”
6	字段 6	6	10	左手拇指
7	字段 7	7~20	FF FF FF FF FF FF FF FF FF FF FF FF FF FF	保留项
8	字段 8	21~22	4F 01	压缩图像数据长度为 20225 个字节
9	字段 9	23~20247	--	指纹压缩图像数据
10	字段 10	20248	D6	CRC 校验位



算法库接口函数说明

3.1 文件格式

3.1.1 文件生成

接口函数采用 Windows 动态链接库(DLL)形式发布。支持的操作系统应包括 Windows XP 或兼容系统。接口函数应在动态链接库中输出，支持多线程。程序编译为 32 位版本。

3.1.2 文件名称

动态链接库文件名称为：ID_Fpr.dll。

3.2 接口函数

接口函数见表 3.1。

接口函数说明中指纹图像数据不包括指纹压缩图像数据。

表 3.1 接口函数

编号	名称	说明
1	FP_GetVersion	版本信息获取
2	FP_Begin	初始化操作
3	FP_FeatureExtract	指纹图像特征提取
4	FP_FeatureMatch	指纹特征数据比对
5	FP_ImageMatch	指纹图像数据与指纹特征数据比对
6	FP_Compress	指纹图像数据压缩
7	FP-Decompress	指纹图像数据复现
8	FP_GetQualityScore	指纹图像质量值获取
9	FPGenFeatureFromEmpty1	生成“注册失败”指纹特征数据
10	FPGenFeatureFromEmpty2	生成“未注册”指纹特征数据
11	FP_End	结束操作

3.3 函数说明

3.3.1 版本信息获取

函数原型：int __stdcall FP_GetVersion(unsigned char code[4])。

参数：unsigned char code[4] 版本信息格式为 XXYY，XX 为开发者代码，YY 为版本号。例如“1201”，则 code 的填写方式为：code[0]= ‘1’，code[1]= ‘2’，code[3]= ‘0’，code[4]= ‘1’。输出参数。



返回值：调用成功，返回 1；否则返回错误代码，错误代码值应符合 3.4 的要求。
说明：获取版本信息。

3.3.2 初始化操作

函数原型：int __stdcall FP_Begin()。

参数：无。

返回值：调用成功，返回 1；否则返回错误代码，错误代码值应符合 3.4 的要求。

说明：初始化操作。

3.3.3 指纹图像特征提取

函数原型：int __stdcall FP_FeatureExtract(unsigned char cScannerType,unsigned char cFingerCode,unsigned char * pFingerImgBuf,unsigned char * pFeatureData)。

参数：unsigned char cScannerType 指纹采集器代码。输入参数。

unsigned char cFingerCode 指位代码。输入参数。

unsigned char * pFingerImgBuf 指纹图像数据指针，指纹图像为 RAW 格式。输入参数。

unsigned char * pFeatureData 指纹特征数据指针，存储生成的指纹特征数据，由调

用者分配内存空间，指纹特征数据文件结构应符合附录 A 要求。输出参数。

返回值：调用成功，返回 1；否则返回错误代码，错误代码值应符合 3.4 的要求。

说明：对指纹图像数据进行特征提取，生成指纹特征数据。

3.3.4 指纹特征数据比对

函数原型：int __stdcall FP_FeatureMatch(unsigned char * pFeatureData1,unsigned char* pFeatureData2,float * pfSimilarity)。

参数：unsigned char * pFeatureData1 指纹特征数据指针 1。输入参数。

unsigned char * pFeatureData2 指纹特征数据指针 2。输入参数。

float * pfSimilarity 相似度，取值范围为 0.00 ~ 1.00，值 0.00 表示不匹配，值 1.00 表示完全匹配。输出参数。

返回值：调用成功，返回 1；否则返回错误代码，错误代码值应符合 3.4 的要求。

说明：对两个指纹特征数据进行比对，得到相似度值。

3.3.5 指纹图像数据与指纹特征数据比对

函数原型：int __stdcall FP_ImageMatch(unsigned char * pFingerImgBuf,unsigned char * pFeatureData,float * pfSimilarity)。

参数：unsigned char * pFingerImgBuf 指纹图像数据指针,指纹图像为 RAW 格式。输入参数。

unsigned char * pFeatureData 指纹特征数据指针。输入参数。

float * pfSimilarity 相似度，取值范围为 0.00 ~ 1.00，值 0.00 表示不匹配，值 1.00 表示完全匹配。输出参数。

返回值：调用成功，返回 1；否则返回错误代码，错误代码值应符合 3.4 的要求。

说明：对指纹图像数据与指纹特征数据进行比对，得到相似度值。

3.3.6 指纹图像数据压缩

函数原型：int __stdcall FP_Compress(unsigned char cScannerType, unsigned char cEnrolResult,unsigned char cFingerCode,unsigned char * pFingerImgBuf,int nCompressRatio, unsigned char * pCompressedImgBuf, unsigned charstrBuf[256])。

参数：unsigned char cScannerType 指纹采集器代码。输入参数。

unsigned char cEnrolResult 注册结果代码。输入参数。



unsigned char cFingerCode 指位代码。输入参数。

unsigned char * pFingerImgBuf 指纹图像数据指针，指纹图像为 RAW 格式。输入参数。

int nCompressRatio 指纹图像数据压缩倍数。输入参数。

unsigned char * pCompressedImgBuf 指纹压缩图像数据指针，调用者在调用此函数前，应当分配不小于 20 480 字节的内存，指纹压缩图像数据文件结构应符合附录 A 要求。

输出参数。

unsigned char strBuf[256] 错误信息，如果压缩图像发生错误，并且返回值为-9 的情况下，strBuf 填写错误信息。错误信息为以数值 0 结尾的字符串，采用 GB 13000 中规定的字符。输出参数。

返回值：调用成功，返回 1；否则返回错误代码，错误代码值应符合 3.4 的要求。

说明：对指纹图像数据进行压缩。

3.3.7 指纹图像数据复现

函数原型：int __stdcall FP_Decompress(unsigned char * pCompressedImgBuf,unsigned char * pFingerImgBuf,unsigned char strBuf[256])。

参数：unsigned char * pCompressedImgBuf 指纹压缩图像数据指针，压缩图像数据长度不大于 20 480 字节。输入参数。

unsigned char * pFingerImgBuf 指纹复现图像数据指针，指纹图像为 RAW 格式，调用者在调用此函数前，应当分配 92 160 字节的内存。输出参数。

unsigned char strBuf[256] 错误信息，如果压缩图像发生错误，并且返回值为-9 的情况下，strBuf 填写错误信息。错误信息为以数值 0 结尾的字符串，采用 GB 13000 中规定的字符。输出参数。

返回值：调用成功，返回 1；否则返回错误代码，错误代码值应符合 3.4 的要求。

说明：对指纹原始图像数据进行复现。

3.3.8 指纹图像质量值获取

函数原型：int __stdcall FP_GetQualityScore(unsigned char * pFingerImgBuf,unsigned char * pnScore)。

参数：unsigned char * pFingerImgBuf 指纹图像数据指针，指纹图像为 RAW 格式。输入参数。

unsigned char * pnScore 指纹图像质量值指针，指纹图像质量值取值范围为 00H ~64H，值 01H 表示最低质量，值 64H 表示最高质量，值 00H 表示未知。

返回值：调用成功，返回 1；否则返回错误代码，错误代码值应符合 3.4 的要求。

说明：获取指纹图像的质量值。

3.3.9 生成“注册失败”指纹特征数据

函数原型：int __stdcall FP_GenFeatureFromEmpty1(unsigned char cScannerType,unsigned char cFingerCode,unsigned char * pFeatureData)。

参数：unsigned char cScannerType 指纹采集器代码。输入参数。

unsigned char cFingerCode 指位代码。输入参数。

unsigned char * pFeatureData 指纹特征数据指针，存储生成的指纹特征数据，由调用者分配内存空间，指纹特征数据文件结构应符合附录 A 要求。输出参数。

返回值：调用成功，返回 1；否则返回错误代码，错误代码值应符合 3.4 的要求。

说明：本函数针对“注册失败”手指的情况，生成“注册失败”指纹特征数据。其中，字段 5 设置为 02H；字段 9、字段 10 均设置为 00H。

3.3.10 生成“未注册”指纹特征数据

函数原型：int __stdcall FP_GenFeatureFromEmpty2(unsigned char cFingerCode,unsigned char * pFeatureData)。

参数：unsigned char cFingerCode 指位代码。输入参数。

unsigned char * pFeatureData 指纹特征数据指针，存储生成的指纹特征数据，由调用者分配内存空间，指纹特征数据文件结构应符合附录 A 要求。输出参数。



用者分配内存空间，指纹特征数据文件结构应符合附录 A 要求。输出参数。
返回值：调用成功，返回 1；否则返回错误代码，错误代码值应符合 3.4 的要求。
说明：本函数针对“未注册”手指的情况，生成“未注册”指纹特征数据。其中，字段 5 设置为 03H；字段 6 设置为 61H、62H 或 63H；字段 7、字段 9、字段 10 均设置为 00H。

3.3.11 结束操作

函数原型：int __stdcall FP_End()。
参数：无。
返回值：调用成功，返回 1；否则返回错误代码，错误代码值应符合 3.4 的要求。
说明：结束操作。

3.4 错误代码

错误代码见表 3.2。

表 3.2 错误代码表

错误代码	说明
-1	参数错误
-2	内存分配失败，没有分配到足够的内存
-3	功能未实现
-9	其他错误

附录 A

指纹采集函数调用流程，如图 1 所示。

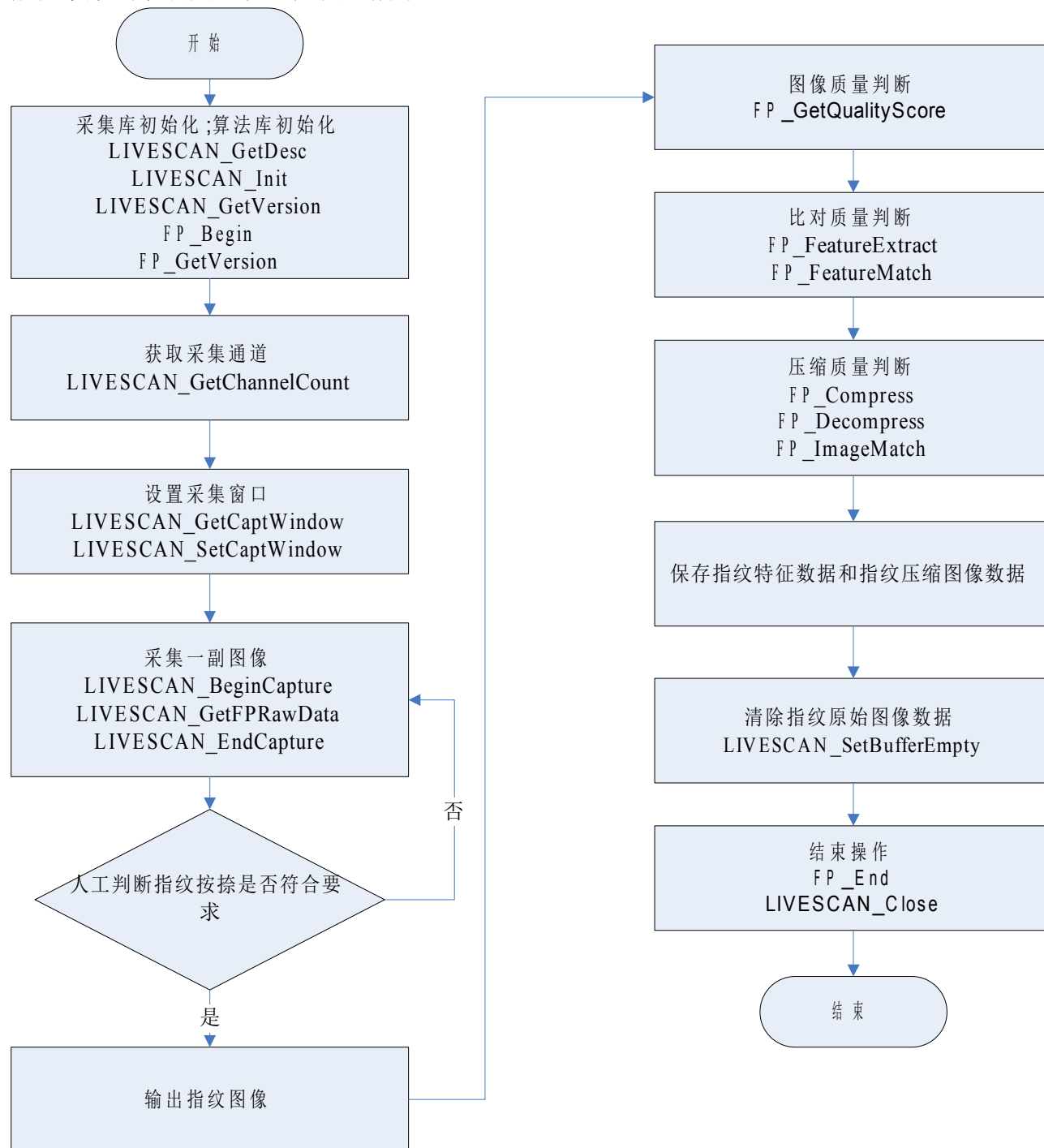


图 1 指纹采集流程图