软件设计说明

1. 软件需求说明
2. 软件监视ftp服务器指定目录。前端设备（摄像头）将上传抓拍图片到该目录中。软件需要判断上传的图片是否完整并对每张图片文件的信息进行读取并按要求将信息储存到数据库中。
3. 图片文件信息包括两部分。一部分是图片文件本身的文件属性，如创建时间，最后修改时间，另一部分图片信息在图片头中，详细说明见《抓拍系统数据包格式.docx》中第一部分“前端抓拍图片内嵌信息”的说明。
4. 获取后的信息要按照数据库表格要求储存，数据表的说明见《CDMTCP字典表结构-V1.1.doc》。数据表中的一条信息代表的是一组抓拍图片。
5. 图片的文件命名规则，如下：

2013041710510982-d137-0001-3.jpg

日期：2013-04-07

时间：10:51:10.82

MAC：d137

组号：0001

组内编号：3

5. 不同组的图片通过组号来区分。每组图片中有多张图片（现在一组图片是3张），通过组内编号来区分（编号从1开始，分别是1，2，3）。

6. 图片上传的格式为JPEG，关于JPEG文件储存格式见《JPEG文件存储格式.docx》。对于如何判断上传的图片文件是否完整，根据JPEG文件的储存格式。

7. 根据成都方面新增的需求文档《20130602成都MTCP.doc》的第一部分“图片命名规则”，图片名在储存到数据库时要符合该命名规则。由于摄像头更新软件较麻烦（需要重新烧程序到每个摄像头），所以这部分工作在文件上传到服务器上后进行处理。

8. 根据合肥方面的需求，添加按组对图片进行合成的功能。图片合成效果见《图片合成效果图.doc》

1. 软件架构设计

整体架构分为4个部分。分别是

文件夹监视，图片信息处理，数据库储存，图片合成。

1. “文件夹监视”模块监视指定的文件夹。当有新的文件上传到该文件夹时，将文件名给“图片信息处理”模块。

2. “图片信息处理”模块首先根据图片文件的格式（JPEG）判断文件的完整性，再获取图片文件的属性（创建时间，最后修改时间等），之后按照“图片内嵌信息”的规则获取图片中的信息数据，然后按照“图片命名规则”修改图片文件名。处理完一张图片后将图片文件信息给“数据库储存”模块储存到数据库中。

3. “数据库储存”模块按照数据表要求将信息储存到数据库。

4. “图片合成”对一组图片进行图片合成，并存放到指定的文件夹内。

1. 详细设计
2. 文件夹监视

输入：需要监视的文件夹路径

输出：文件夹中新上传文件的文件名

1. 图片信息处理

输入：文件夹中新上传文件的文件名

输出：一张图片信息

下面是输出格式的详细说明

一张图片的内嵌信息包括：

|  |  |  |
| --- | --- | --- |
| 项目名 | 例子 | 说明 |
| MAC | 0800280245C5 | 摄像头的MAC |
| GRUP\_TIME (RTC) | 2013041012125900 | 图片抓拍时间 |
| gps\_x | 3118.69556 | X 坐标 |
| gps\_y | 12037.05619 | Y坐标 |
| gps 车速 | 123.4 | 123.4km/h |
| gps 方向 | NE |  |
| 车牌 | 苏A12345 |  |
| 车牌颜色 | 红 |  |
| 车距 | 12 | 12米 |
| 抓拍流水号 | 0050 | 当天第50组 |
| 抓拍序号 | 1 | 组内编号 |

(次部分已不用)

一组图片的信息，即图片信息组：

|  |  |  |
| --- | --- | --- |
| 内容 | 例子 | 说明 |
| 图片文件名 | 2013041710510982-d137-0001 | 不包含组内编号 |
| 本组图片信息数量 | 3 |  |
| 图片内嵌信息组 | 见前表 | 根据图片信息数量的个数 |

本模块每次根据图片文件名打开对应的文件，首先判断文件的完整性，判断规则是图片文件是否存在“JPEG文件存储格式”的JPEG文件结束符。若图片文件不完整，则处理结束。若图片完整则继续下面的处理。按照“抓拍图片内嵌信息格式”获取图片文件的内嵌信息。根据新需求“图片命名规则”，在图片信息储存到数据库前对图片进行重命名。最后将信息给“数据库模块”储存到数据库。

关于图片重命名。

摄像头图片命名规则：年月日（8位）+时分秒（6位）+设备编号MAC后4位（4位）+组号（4位）+组内编号（1位）

例子：20130503170514-db98-0002-1.jpg

新图片命名规则：设备点编号（12位数） +年月日（6位）+时分秒（6位数） + 车道（1位数） + 违法地点（12位数） + 实际速度（5位数）+限定速度（3位数）+连续数（1位数）+本序数（1位数）

详细说明：

设备点编号：12位；510100A50\*\*\*

650台前三期设备\*\*\*为原交管编码

300台四期设备\*\*\*暂定为401—499，40A—40Z，41A—41Z，42A—42Z，43A—43Z，44A—44Z，45A—45Z，46A—46Z，47A—47Z，48A—48J(字母O和I除外)。

违法时间 ： 12位 精确到秒（YYMMDDHHMMSS），每组图片的违法时间统一为第一张图片时间；图片叠加水印时间仍精确至毫秒。

车 道 ： 1位 取值为0。

违法地点 ： 12位 取值为000000000000。

实 速 值 ： 5位 取值为00000。

限 速 值 ： 3位 取值为000。

连 续 数 ： 1位 每个证据照片张数；（至少为2,成都统一为3）

本 序 数 ： 1位 本照片顺序数。（暂同组内编号）

例子：510100A50401-130503170514-0-000000000000-00000-000-3-1.jpg

通过两个图片命名规则的对比可以看出需要改变的主要有下面几个部分：

1.设备MAC编号和设备交管编码之间一一对应（该表暂没有）

2.年月日年取后2位

3.连续数为一组图片的数量，成都MTCP一组图片数量是3

打开文件

获取文件内嵌信息

不完整

保存信息

结束

判断文件完整性

图片重命名

1. 数据库储存

输入：一张图片的信息。

输出：数据库表要求的形式储存到数据库中。下表是要求存入一条记录的列和列的含义。其他列不需要存储。

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 含义 |
| id | Int | 标识符，自增，不需要存入 |
| camera\_id | varchar(50) | 摄像头的MAC |
| picture\_name | varchar(100) | 图片名称，包含绝对路径，多个图片名用‘，’号分隔 |
| direction | varchar(50) | 方向 |
| car\_id | varchar(50) | 车牌号 |
| capture\_serial\_num | varchar(50) | 抓拍流水号 |
| license\_color | varchar(50) | 车牌颜色 |
| receive\_picture\_nums | varchar(50) | 这组图片数量 |
| collect\_date1 | datetime | 图片1抓拍时间 |
| collect\_date2 | datetime | 图片2抓拍时间 |
| collect\_date3 | datetime | 图片3抓拍时间 |
| … | … | 图片n抓拍时间（n最大为8） |
| recieve\_begin\_time1 | datetime | 图片1开始接收时间 |
| recieve\_begin\_time2 | datetime | 图片2开始接收时间 |
| recieve\_begin\_time3 | datetime | 图片3开始接收时间 |
| … | … | 图片n开始接收时间（n最大为8） |
| receive\_time1 | datetime | 图片1完全接收时间 |
| receive\_time2 | datetime | 图片2完全接收时间 |
| receive\_time3 | datetime | 图片3完全接收时间 |
| … | … | 图片n完全接收时间（n最大为8） |
| gps\_x1 | varchar(50) | 图片1,X坐标 |
| gps\_x2 | varchar(50) | 图片2,X坐标 |
| gps\_x3 | varchar(50) | 图片3,X坐标 |
| … | … | 图片n,X坐标（n最大为8） |
| gps\_y1 | varchar(50) | 图片1,Y坐标 |
| gps\_y2 | varchar(50) | 图片2,Y坐标 |
| gps\_y3 | varchar(50) | 图片3,Y坐标 |
| … | … | 图片n,Y坐标（n最大为8） |
| car\_distance1 | varchar(50) | 图片1, 车距 |
| car\_distance2 | varchar(50) | 图片2, 车距 |
| car\_distance3 | varchar(50) | 图片3, 车距 |
| … | … | 图片n, 车距（n最大为8） |
| speed1 | varchar(50) | 图片1, 车速 |
| speed2 | varchar(50) | 图片2, 车速 |
| speed3 | varchar(50) | 图片3, 车速 |
| … | … | 图片n, 车速（n最大为8） |
| create\_time | datetime | 本记录创建时间或最近数据更新时间 |
| backup1 | varchar(50) | 本组抓拍图片的日期（Ymd字符串） |

数据插入方法。

若某张图片是任意一天任意一台设备的任意一组图片的第一张，则插入数据库一条新的记录。本组后续图片信息则更新该条记录。

第一张图片信息插入数据库时

对应的插入SQL语句如下：

INSERT INTO LS\_pictures (

camera\_id,

picture\_name,

direction,

car\_id,

license\_color,

capture\_serial\_num,

receive\_picture\_nums,

collect\_dateN,

recieve\_begin\_timeN,

receive\_timeN,

gps\_xN,

gps\_yN,

car\_distanceN,

speedN,

create\_time,

backup1

)

VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)

其中的N是第图片的组内编号（一般为1）。

对应的更新SQL语句如下：

UPDATE LS\_pictures

SET picture\_name = picture\_name + ?,

receive\_picture\_nums = recieve\_picture\_nums + 1,

collect\_dateN = ?,

recieve\_begin\_timeN = ?,

receive\_timeN = ?,

gps\_xN = ?,

gps\_yN = ?,

car\_distanceN = ?,

speedN = ?,

create\_time = ?

WHERE (camera\_id = ?,

backup1 = ?,

capture\_serial\_num = ?)

1. 测试