Video Surviellance Mobile App

Team Members: Sunil Srinivas Gummadam
Email Address : fb8563@wayne.edu.

Introduction

There is a constant Increase in the need for an intelligent video surviellance system in any given Domain. The proposed solution deals with the real time monitoring of objects in an environment in any domain.Firstly, the basic principle of moving object detecting is given. Limited by the memory consuming and computing capacity of a mobile phone, a background subtraction algorithm is presented for adaptation. Then, a self-adaptive background model that can update automatically and timely to adapt to the slow and slight changes of natural environment is detailed. When the subtraction of the current captured image and the background reaches a certain threshold, a moving object is considered to be in the current view, and the mobile phone will automatically notify the central control unit or the user through phone call, SMS (Short Message System) or other means.

A low-cost intelligent wireless security and monitoring solution using moving object recognition technology is presented . The system has good mobility, which makes it a useful supplement of traditional monitoring system. It can also perform independent surveillance mission and can be extended to a distributed surveillance system. Limited by the memory consuming and computing capacity in a mobile phone, background subtraction algorithm is presented to be adopted in mobile phones. In order to be adapted to the slow and slight changes of the natural environment,

Even though there exist a myriad of background subtraction algorithms in the literature, most of them follow a simple flow diagram shown in Figure 1. The four major steps in a background subtraction algorithm are preprocessing, background modeling, foreground detection, and data validation. Preprocessing consists of a collection of simple image processing tasks that change the raw input video into a format that can be processed by subsequent steps. Background modeling uses the new video frame to calculate and update a background model. This background model provides a statistical description of the entire background scene. Foreground detection then identifies pixels in the video frame that cannot be adequately explained by the background model, and outputs them as a binary candidate foreground mask. Finally, data validation examines the candidate mask, eliminates those pixels that do not correspond to actual moving objects, and outputs the final foreground mask. Domain knowledge and computationally-intensive vision algorithms are often used in data validation. Real-time processing is still feasible as these sophisticated algorithms are applied

only on the small number of candidate foreground pixels. Many different approaches have been proposed for each of the four processing steps. We review some of the representative ones in the following subsections.

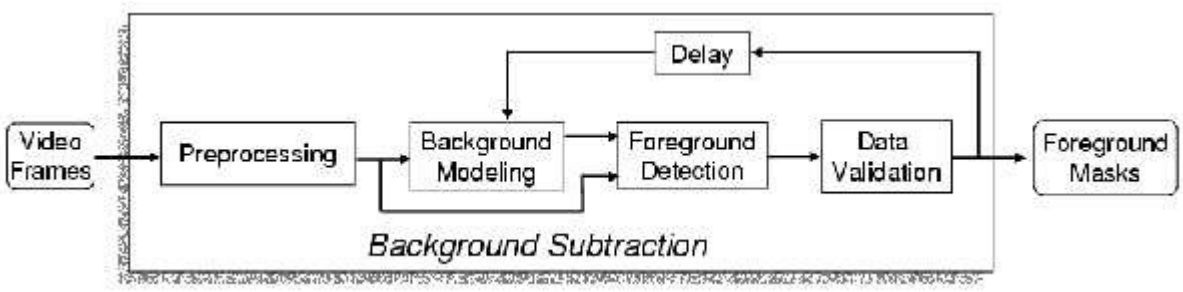

Fig 3.1 Flow diagram of a generic background subtraction algorithm.

Preprocessing

In most computer vision systems, simple temporal and/or spatial smoothing is used in the early stage of processing to reduce camera noise. Smoothing can also be used to remove transient environmental noise such as rain and snow captured in outdoor camera. For real-time systems, frame-size and frame-rate reduction are commonly used to reduce the data processing rate. If the camera is moving or multiple cameras are used at different locations, image registration between successive frames or among different cameras is needed before background modeling .Another key issue in preprocessing is the data format used by the particular background subtraction algorithm. Most of the algorithms handle luminance intensity, which is one scalar value per each pixel. However, color image, in either RGB or HSV color space, is becoming more popular in the background subtraction literature. These papers argue that color is better than luminance at identifying objects in low-contrast areas and suppressing shadow cast by moving objects. In addition to color, pixel-based image features such as spatial and temporal derivatives are sometimes used to incorporate edges and motion information. For example, intensity values and spatial derivatives can be combined to form a single state space for background tracking with the Kalman filter. Pless et al. combine both spatial and temporal derivatives to form a constant velocity background model for detecting speeding vehicles. The main drawback of adding color or derived features in background modeling is the extra complexity for model parameter estimation. The increase in complexity is often significant as most background modeling techniques maintain an independent model

Background Subtraction Technology

Background subtraction is a commonly used class of techniques for segmenting out moving objects of interest in a scene for applications such as surveillance. It involves comparing an observed image with an estimate of the image if it contained no objects of interest. The areas of the image plane where there is a significant difference between the observed and estimated images indicate the location of the objects of interest. The term “background subtraction" comes from the simple technique of subtracting the timely updated background template from the observed image and then thresholding the result to generate the objects of interest

Before the moving objects can be identified, a background template must be built. Generally, background and foreground (moving objects) are mixed together such as waving leaves in the garden and running automobiles on high way. The foreground cannot be removed so the ideal background image cannot be retrieved. But the moving objects do not exist in the same location in each frame of a real-time video sequence. An “average” frame of the video sequence can be retrieved to approach the ideal background image. The gray values of pixels which have the same location in each frame of the video sequence are averaged to represent the gray value of the pixel which located in the same place in the approximate background. An average value of pixels in the same location of each frame in a video sequence is calculated. To simplify, the approximate background is also called “background template”, “background” or “template” in the following contents.

$$Background_{first} = \frac{Frame_1 + Frame_2 + ... + Frame_i}{i} \quad (1)$$

In our prototype, the first 10 frames are captured to calculate the background template (i=10). Moving objects can not be identified in these frames. If the moving objects move too slowly, i should be increased to reduce the tolerance.

Moving Object Recognition

After the background template has been constructed, the background image can be subtracted from the observed image. The result is foreground (moving objects). Actually, the background is timely updated. The update algorithm is detailed in the next section.

Foreground j =| Frame j − Background j | (j>i) (2)

In case of some random disturbances, each pixel will fluctuate in a small range even there is no expected moving objects in the scene. So there must be a strategy to judge it. A threshold is defined in the system. If the difference of one pixel between real time frame and template is more than 10, then add 1 to the threshold. When differences of all pixels in the frame are all calculated, moving objects is thought to appear if the threshold is more then 3 percent of the total number of pixels in the frame.

MODULES

1. Capturing the Video.
2. Comparing Each Frame
3. Alerting System

Capturing The Video:

Once the camera video is shown on the device, capturing an image is easy. All you need to do is call Video Control's get Snapshot() method. The get Snapshot() method returns an array of bytes, which is the image data in the format you requested. The default image format is PNG (Portable Network Graphic). The Mobile Media API (MMAPI) extends the functionality of the J2ME platform by providing audio, video and other time-based multimedia support to resource-constrained devices.

Getting a Video Capture Player:

The first step in taking pictures (officially called video capture) in a MIDlet is obtaining a Player from the Manager.

Player mPlayer = Manager.createPlayer("capture://video");

The Player needs to be realized to obtain the resources that are needed to take pictures.

mPlayer.realize();

Showing the Camera Video :

The video coming from the camera can be displayed on the screen either as an Item in a Form or as part of a Canvas. A Video Control makes this possible. To get a VideoControl, just ask the Player for it:

VideoControlmVideoControl= (VideoControl)mPlayer.getControl("VideoControl");

Capturing an Image :

Once the camera video is shown on the device, capturing an image is easy. All you need to do is call VideoControl's getSnapshot() method. The getSnapshot() method returns an array of bytes, which is the image data in the format you requested. The default image format is PNG (PortableNetwork Graphic).

```
byte[] raw = mVideoControl.getSnapshot(null);
Image image = Image.createImage(raw, 0,
raw.length);
```

Comparing each Frames:

Background subtraction is a commonly used class of techniques for segmenting out moving objects of interest in a scene for applications such as surveillance. It involves comparing an observed image with an estimate of the image if it contained no objects of interest. The areas of the image plane where there is a
significant difference between the observed and estimated images indicate the location of the objects of interest. The term “background subtraction" comes
from the simple technique of subtracting the timelyupdated background template from the observed image and then thresholding the result to generate the objects
of interest.

After the background template has been constructed, the background image can be subtracted from the observed image. The result is foreground (moving objects). Actually, the background is timely updated.
The update algorithm is detailed in the next section.

Foreground j =| Frame j − Background j | (j>i)

In case of some random disturbances, each pixel will fluctuate in a small range even there is no expected moving objects in the scene. So there must be a strategy to judge it. A threshold is defined in the system. If the difference of one pixel between real time frame and template is more than 10, then add 1 to the threshold. When differences of all pixels in the frame are all calculated, moving objects is thought to appear if the threshold is more then 3 percent of the total number of pixels in the frame.

Alerting System:

When the subtraction of the current captured image and the background reaches a certain threshold, a moving object is considered to be in the current view,
and the mobile phone will automatically notify the central control unit or the user through phone call, MMS. The J2ME Wireless Toolkit supports the Wireless Messaging API (WMA) with a sophisticated simulation environment. WMA 1.1 (JSR 120) enables MIDlets to send and receive Short Message Service (SMS) or Cell Broadcast Service (CBS) messages. WMA 2.0 (JSR 205) includes support for MMS messages as well.

Java Code: (Distributed Programming Concepts : Using Threads)

P.java : Initializing the Video Player

SMSSend.java : Checks the validity of the phone number and connects to the port to send SMS.

SMSSENDER.java :Prompts for text and sends it via an SMS MessageConnection.Send the message. Called on a separate thread so we don't have contention for the display.

VideoPlayer.java: Play Video/Capture in a Form using MMAPI.Called by MIDlet after instantiating this object. Creates the player. Performs image processing and then appends to this form.Deallocate the player and the display thread.Some VM's may stop players and threads on their own, but for consistent user experience, it's a good idea to explicitly stop and start resources such as player and threads.

VIDEOTEST.java: All the methods for starting running and stopping the thread methods are shown in this code.

Known Issues:

1. The mobile app developed has challenges when trying to be installed to an Android Phone.
2. It runs well on phones that have java already installed on them.