Improved Morphological Method in Motion Detection

Mandeep Singh

Design Engineer, Centre for Development of Advanced Computing, Mohali, Punjab, India
(A Scientific Society of Ministry of Communication and Information Technology, Govt. of India)

ABSTRACT

This paper describes an algorithm for detecting moving objects in color image sequences acquired with a camera. A continuous video stream of traffic scenes recorded by a stationary camera is processed at various levels by comparing the current video frame with the previous frame. Vehicles and pedestrian are modeled as rectangular patches with certain dynamic behavior. The proposed method is based on the establishment of correspondences between regions and objects, as the objects move through the image sequence. The system successfully extracts moving edges from dynamic images sequences.

Keywords

Video frames, pixels, blobs

1. INTRODUCTION

MOTION detection and tracking are becoming increasingly more important capabilities in any vision system designed to operate in an uncontrolled environment. Extracting moving objects from image sequences is one of the major interests in numerous applications like: target tracking, video surveillance, vehicle navigation, traffic control systems etc [4, 11, 1]. In most of the motion detection methods, a background is subtracted from the captured images and is called background subtraction. Since this method of difference image from background subtraction affect the accuracy of motion detection, moreover it does not cop up with noise and other intensity factors.

In this paper, we use two successive images frames as the basis of motion detection. The video is divided into number of frames for example the video of up to 10 seconds is break down into 80 to 85 frames. After dividing into frames, two successive frames are subtracted and the resultant image is further passed through a number of filters. The resultant image will indicate the moving objects in that frame by drawing a box around it.

2. ALGORITHM

The System Working involves division of image frames, grayscale images, and resultant image after subtraction, display object. They are described in detail as follows.

2.1 Divide Video into Image Frames

Video technology is used for electronically capturing, recording, processing, storing, transmitting, and reconstructing a sequence of still images representing scenes in motion. The number of still pictures per unit of time of video ranges from six or eight frames per second (*frame/s*) for old mechanical cameras to 120 or more frames per second for new professional cameras. PAL (Europe, Asia, Australia, etc.) and SECAM (France, Russia, parts of Africa etc.) standards specify 25 frames, while NTSC (USA, Canada, Japan, etc.) specifies 29.97 frames. The video which is used in our study is taken from a web cam having frame rate up to 30 frames per second and it captures video images at QVGA (640 x 480 pixels, software enhanced) resolution is shown in Fig. 1 (a), 1 (b) and 1 (c).

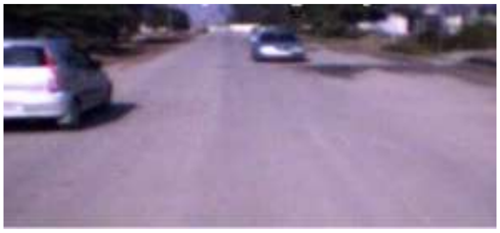

Figure 1 (a) Frame 1

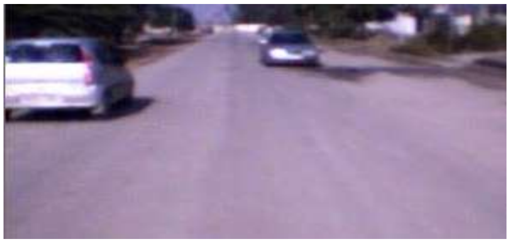

Figure 1 (b) Frame 2

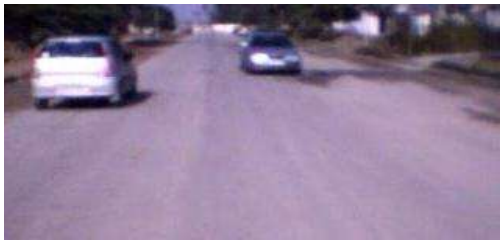

Figure 1 (c) Frame 3

There is number of Active-X controls and Dll's available in every Programming Language (like in VB6.0 AviProcessor 1.1) using, which we can divide this video into image frame. These frames are saved in the memory and processed accordingly

2.2 Converting Images into Grey Scale

As we know that an image is comprised of millions of pixels and each pixel consists of 3 colors Red, Green and Blue. Any color can be obtained by just adjusting these 3 colors. A hex triplet is a six-digit, three- byte hexadecimal number used in HTML, CSS, SVG, and other computing applications, to represent colors. The bytes represent the red, green and blue components of the colors. One byte represents a number in the range 00 to FF (in hexadecimal notation), or 0 to 255 in decimal notation. This represents the least (0) to the most (255) intensity of each of the color components. For example hex code for Red is FF 00 00 and decimal code is 255 0 0. The 2 successive Colored image frames are converted in grey images; this is done so that the resultant obtained in the end will be more reliable and accurate. Some of the color codes for grey scale are given in the table 1 below:

Table 1. Colour Codes for Grey Scale

The 2 successive colored image frames can be converted into grey scale images by using following formula.

Grey color	Hex Code			Decimal Code		
Gainsboro	DC	DC	DC	220	220	220
LightGrey	D3	D3	D3	211	211	211
Silver	C0	C0	C0	192	192	92
DarkGray	A9	A9	A9	169	169	169
Gray	80	80	80	128	128	128
DimGray	69	69	69	105	105	105
Light Slate Gray	77	88	99	119	136	153
SlateGray	70	80	90	112	128	144

Red = Pixel value of red color in decimal
Green = Pixel value of green color in decimal.
Blue = Pixel value of blue color in decimal

$$\text{Grey value} = \frac{(\text{Red} + \text{Green} + \text{Blue})}{3} \qquad (1)$$

ImagePixels (Blue, x, y) = Grey value
ImagePixels (Green, x, y) = Grey value
ImagePixels (Red, x, y) = Grey value

Where, ImagePixels (color, x-coordinate, y-coordinate) is an array used to store the RGB values of each pixel separately. ‘x’ and ‘y’ are the image coordinates , the maximum value of ‘x’ can goes up to the width of the image and maximum value of ‘y’ goes up to height of the image.

For image of the size (340 x 220), $0 \leq x \leq 340$ and $0 \leq y \leq 220$. The above given formula is applied to both image frames separately and the corresponding grey scale values can be stored into arrays. The Decimal RGB colors equivalent for grey is 128 128 128. The color values represent different shades of grey color as shown in Table 1 and tonal distribution of the images can be graphically represented by using histogram. The histogram for the Grey scale image frame up to 20791 pixels is shown in Fig. 2. This histogram plots the number of pixels in the image (horizontal axis) with a particular brightness value (vertical axis).

2.3 Enlarging Pixels of Image Frames

After converting both the images into grey scale, enlarge each pixel of both the images. This step is done so that the image which we get after subtraction contains minimum amount of noise and the objects which are detected should have sharp and well defined boundaries

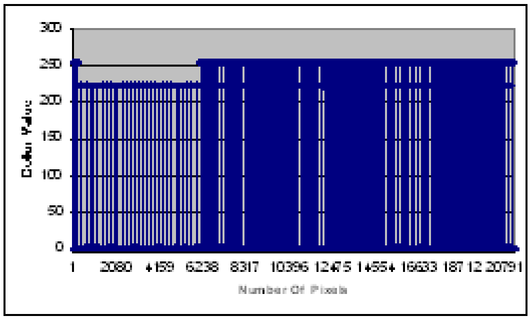

Figure 2. Histogram for Grey Image Frame

We replace an image with large pseudo-pixels so that each pixel represents the average color of the area they represent. We loop over the image in pseudo-pixel blocks of a specified size. For each block, it takes the average of the colors in the block and sets all pixels in the block to that average color. The block size may vary according to the requirements. We used a block size of 10 and the total number of pixels in this block is 100 (i.e. 10 x 10). The pseudo code for this is given below:

```
Set total_red = 0, total_green = 0, total_blue = 0
For i = 1 to 10
    For j= 1 to 10
        total_red = total_red + ImagePixels (Red, i, j)
        total_green =     total_green + ImagePixels (Green, i, j)
        total_blue = total_blue + ImagePixels (Blue, i, j)
    Next j
Next i
total_pixels = (block_size) * (block_size)
total_red= total_red / total_pixels
total_green = total_gen / total_pixels
total_blue = total_blue / total_pixels
```

where total_red, total_blue and total_green is sum of all the RGB values of the pixels in that block (i.e. 10 x 10 block)total_pixels is the total number of pixels in that block ,100 in this case.The grey image and image after enlarging image is show in Fig.3 and Fig.4.

2.4 Subtraction and Thrasholding of Image Frames

In this step RGB values of each pixel of first image are subtracted from RGB values of second image [7, 8].

Red = ImagePixels1 (Red,i,j) – ImagePixels2 (Red,i,j)
Green =I magePixels1(Green,i,j) – ImagePixels2 (Green,i,j)
Blue= ImagePixels1 (Blue,i,j) – ImagePixels2 (Blue,i,j)

Where Red, Green and Blue are the resultant colors after subtracting 2 images. ImagePixels1( ) and ImagePixels2( ) are arrays of image frame 1 and image Frame 2 respectively, which are used to store the RGB values of each pixel separately.

In the resultant image (which is obtained after subtracting the image frames) objects which are in motion are highlighted in grey shades and rest all is black out.

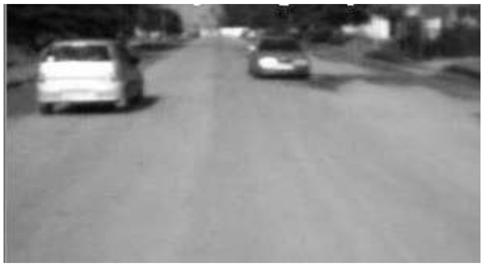

Figure 3. Grey Image

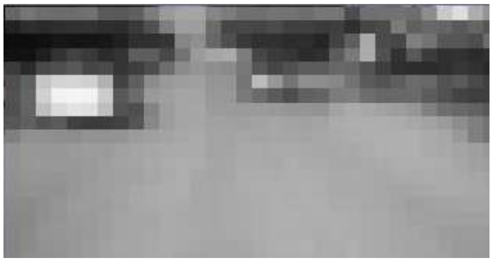

Figure 4. After Pixellat

2.4.1Thrasholding or Binarisation

The main purpose of Binarisation is to make the resultant image in 2 color black and white so that it is easier to calculate blobs in the image. The minimum and maximum values of the RGB components are set [9].
The pseudo-code looks like this.

```
If Red < 4 And Green < 4 And Blue < 4 Then
      Red = 0
      Green = 0
      Blue = 0
    Else
      Red = 255
      Green = 255
      Blue = 255
    End If
```

In the above code RGB (0 0 0) represents black colors and RGB (255 255 255) represents white colors.The resultant images are shown below from Fig.5 to Fig.8.

2.5 Blob Detection

Blobs are compact objects of approximately the same intensity. As we can see in Fig.8 the image contains number of white spots which are known as blobs. It is the most important step in motion detection. The white spots represent the moving bodies in the image frames where each white pixel is counted in the image and the pixels which are connected to each other are considered as a part of one object [5, 10]. This is useful to differentiate between various objects.
Whole image is scanned pixel by pixel and a unique label is assigned to each object (a group of connected components i.e. to each white pixel encountered). These labels are the unique values used to distinguish different objects. In an image different objects are classified by identifying groups of similar pixels. Fig.9 and Fig.10 shows the pixel positions and there corresponding color values respectively. For pixel at position (0, 0) the corresponding color value is 0 (Black), and pixel at position (0, 1) the corresponding color value is 255 (White) and so on. Fig.11 shows the label values for the pixels, clearly the groups of same color pixels are assigned same labels.

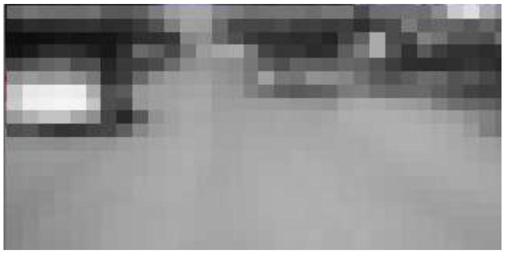

Figure5. Image Frame 1

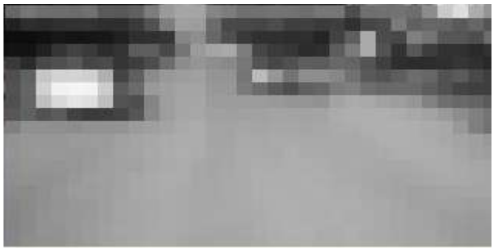

Figure 6. Image Frame 2

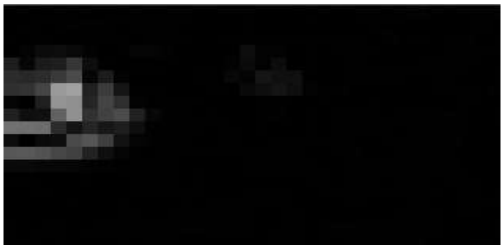

Figure7. After Subtracting Images

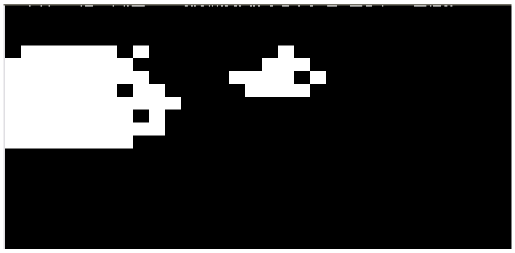

Figure8. After Thrasholding

Each pixel of the image is visited and a label is assigned to each object pixel, which is either a label of its neighbors or a new distinct label, if its neighbors are all background pixels. Let X denotes the 2D array for an image.

Then Let X [i, j] = 0 denote a background pixel and Let X [i, j] = 1, 2, 3 ….. And so on denote an object pixel.

2.6 Combining Blobs

After assigning all the labels to every pixel of the image we count all the labels other than background labels (i.e. other than 0) and store its corresponding x, y –Coordinates. The pixels having same label is considered as a single object and a box is drawn around it using the maximum and minimum x and y-coordinates.

Fig.11 show how different labels are assigned and their corresponding pixel position are shown in Fig.9. So, using this information we draw a box around it. The resultant image is as shown in Fig. 12.

(0,0)	(1,0)	(2,0)	(3,0)
(0,1)	(1,1)	(2,1)	(3,1)
(0,2)	(1,2)	(2,2)	(3,2)
(0,3)	(1,3)	(2,3)	(3,3)
(0,4)	(1,4)	(2,4)	(3,4)

Figure9. Pixel Positions

0	0	0	255
255	255	0	0
0	255	0	0
0	0	0	0
0	255	255	0

Figure10. Color Values

0	0	0	3
1	1	0	0
0	1	0	0
0	0	0	0
0	2	2	0

Figure11. Labels assigned

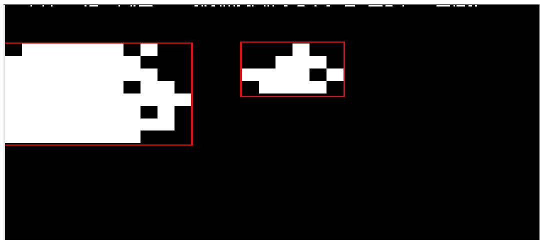

Figure12. Resultant Image

3. CONCLUSION

In this study, we had proposed the method of object detection using the symmetry of the objects and by subtracting pixel values of 2 successive image frames. The results of our experiments show that our proposed algorithm enables stable vehicle and object motion tracking in video frames. If we integrate some other algorithms to our method, the position and speed of vehicles can be easily obtained. For example, our method can be used in the traffic-actuated signal that operates by tracking the three-dimensional coordinates of vehicles using the stereo system or by using Optical flow.

4. FUTURE WORKS

The experiments yielded results under the limited conditions (like speed, environmental conditions etc.). In the future, experiments under more complicated conditions, like disappearance of vehicle, lane change, and time conditions (daytime or night) can be performed. These cases should be considered for practical applications and algorithms can be developed that can be used in any actual situation.

REFERENCES

[1] A. Tsalatsanis, K. Valavanis, A.Yalcin, 2007. Vision Based Target Tracking and Collision Avoidance for Mobile Robots. Journal of Intelligent and Robotic Systems, Volume 48, Issue 2, 285 – 304.

[2] Eiji Morimoto, Masahiko Suguri and Mikio Umeda, 2002. Obstacle Avoidance System for Autonomous Transportation Vehicle Based on Image Processing. The CIGR journal of Scientific Research and Development, Manuscript PM 01 009 Vol. IV.

[3] Hiroshi Unno, Kouki Ojima, Keikichi Hayashible and Hitoshi Saji, 2007. Vehicle Motion Tracking Using Symmetry of Vehicle and Background Subtraction. Proceedings of IEEE intelligent Vehicles Symposium Istanbul, Turkey. 1127–1131.

[4] Jose Melo, Andrew Naftel, Alexandre Bernardino and Jose Santos-Victor, 2006. Detection and Classification of Highway Lanes Using Vehicle Motion Trajectories. IEEE Transactions on Intelligent Transportation System, vol.7, No.2, 188-200.

[5] Kesheng Wu, Ekow Otoo, Kenji Suzuki, 2008. Two Strategies to Speed up Connected Component Labeling Algorithms. LBNL Tech Report, LBNL –59102.

[6] Michael B. Dillencourt, Hannan Samet and Markku Tamminen, 1992. A General Approach to Connected-Component Labeling for Arbitrary Image Representations. Journal of the ACM (JACM), Volume 39, Issue 2. 253 – 280.

[7] Nir Friedman, Stuart Russell, 2008. Image Segmentation In video Sequences: A probabilistic Approach. In Proc. of Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI).

[8] Ronan Fablet, Patrick Bouthemy, Marc Gelgon 1999. Moving Object Detection In Color Image Sequences Using Region-Level Graph Labeling. Proceeding of 6th IEEE International Conference on Image Processing.

[9] Surendra Gupte, Osama Masoud, Robert F. K. Martin and Nikolaos P. Papanikolopoulos, 2002. Detection and Classification of Vehicles. IEEE Transactions of Intelligent Transportation System, Vol. 3, No. 1, 37-47.

[10] Yijie Han and Robert A. Wagner, 1990. An Efficient and Fast Parallel-Connected component algorithm. Journal of the ACM (JACM), Volume 37, Issue 3, 626 – 642.

[11] Zu Kim, 2006. Real-time Lane Tracking of Curve Local Road, IEEE Intelligent Transportation System Conference Toronto, Canada, 1149-1155.