EE 5359 MULTIMEDIA PROCESSING

Implementation of Moving object detection in

H.264 Compressed Domain

Under the guidance of Dr. K. R. Rao

Submitted by: Vigneshwaran Sivaravindiran

UTA ID: 1000723956
Contents:

1. Acknowledgements........................................................................................................................................... 3
2. List of acronyms..................................................................................................................................................... 4
3. List of figures.......................................................................................................................................................... 5
4. Introduction.......................................................................................................................................................... 6
5. Spatio-temporal object detection.......................................................................................................................... 6
   6.1 Parametric and non-parametric object detection............................................................................................ 7
6. Compressed domain object detection.................................................................................................................... 9
   6.1 Initial region extraction ...................................................................................................................................... 10
      6.1.1 Motion vector calculation .......................................................................................................................... 11
      6.1.2 Algorithm.................................................................................................................................................. 12
   6.2 Moving region detection...................................................................................................................................... 13
   6.3 Unmoving region creation and updation.......................................................................................................... 14
7. Experimental results – spatio-temporal moving object detection................................................................. 14
8. Experimental results – moving object detection using motion vectors......................................................... 20
9. Conclusion.............................................................................................................................................................. 21
10. Future work.......................................................................................................................................................... 21
11. References.......................................................................................................................................................... 22
1. Acknowledgement

I would sincerely like to thank Dr. K.R.Rao for his constant guidance, support and motivation which led to the successful completion of this project.

I would also like to thank all my friends for their support and encouragement.
2. **List of acronyms:**

- DCT – Discrete Cosine Transform
- MATLAB – Matrix Laboratory
- MBR – Minimum Bounding Rectangles
- MPEG – Moving Picture Expert Group
- PDF – Probability density function
3. List of figures:

Fig 5.1: Block diagram of a spatio-temporal object detection
Fig 6.1: Initial region extraction
Fig 6.2: MPEG group of pictures
Fig 6.3: Conversion from bit stream order to display order
Fig 6.4: Flow chart explaining storage of motion vectors in respective arrays
Fig 7.1: Frame Differencing
Fig 7.2: Green Color Distribution
Fig 7.3: Dilated Output
Fig 7.4: Output Image obtained by Green score times Frame Differencing
Fig 7.5: Frame # 70 with two detection boxes
Fig 7.6: Frame Differencing
Fig 7.7: Green Color Distribution
Fig 7.8: Dilated output
Fig 7.9.1: Frame # 120 with two detection boxes
Fig 8.1: Threshold motion vector values for frame #70
Fig 8.2: Threshold motion vector values for frame #120
4. **Introduction:**

Moving object detection has been the integral part in many applications such as video surveillance, biomedical imaging and computer vision applications. The basic idea behind this project work is to develop an end to end system which detects moving objects both in spatial and compressed domains. The results produced are then evaluated for its accuracy of detection.

5. **Spatio-Temporal Object Detection:**

Spatio-temporal object detection is an integral part of many computer vision applications. A common approach is to perform background subtraction, which identifies moving objects from the portion of a video frame that differs significantly from the background. There are four main steps in a background subtraction algorithm [5], they are preprocessing, background modeling, foreground detection and data validation as shown in fig 6.1. Preprocessing step 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. Steps like intensity adjustments, smoothening are handled in preprocessing stage, in real-time systems frame size reduction is also done to speed up the process. The block diagram of a background detection algorithm in spatial domain is shown in fig 5.1. Background modeling is at the heart of any background subtraction algorithm, background modeling is a process to obtain static image regions from a sequence of video. Background modeling is broadly classified into recursive and non-recursive techniques.

Fig 5.1: Block diagram of a spatio-temporal object detection.[2]

Non-Recursive technique uses a sliding window method and it stores a buffer of the previous L video frames, and estimates the background image based on the temporal variation of each
Recursive technique is a non-adaptive technique and does not use past input frame information. Foreground detection just compares the input frame with the background model and uses a threshold value as in binary classification. Data validation is the process of improving the candidate foreground mask based on information obtained from outside the background model. All the background models have three main limitations, first, they ignore any correlation between neighboring pixels; second, the rate of adaption may not match the moving speed of the foreground objects; and third, non-stationary pixels from moving leaves or shadow cast are easily mistaken as true foreground objects. These limitations of background model are reduced in foreground detection and the possibility of false detection is also reduced.

5.1 Parametric and non-parametric object detection:

Background modeling stage is at the heart of any background subtraction algorithm, having said that the recursive technique based background modeling uses a parametric and non-parametric model for object detection. These two models can be intuitively understood from their names which mean a parametric model requires some parametric estimates in order to perform object detection whereas a non-parametric model can perform object detection without requiring any parameters to estimate them. An example of parametric object detection is a simple Gaussian model which requires the mean and standard distribution estimate of the object being detected. Consider that we need to estimate a particular color object from an image. First we need to get the sub window that contains samples of particular color that has to be detected, find the mean and standard deviation from the sub window pixel data.

Steps to be performed for the parametric model of object detection,

1. Estimate the mean and standard deviation of a particular color object to be detected from the sub window block which contains only particular color pixel values.
2. Find, $P(\text{RGB/color})$, which forms the training set,
3. Assume that colors are mutually independent, then,
\[ P(\text{RGB/color}) = P(\text{R/color}) \times P(\text{G/color}) \times P(\text{B/color}) \]  \hspace{1cm} (5-1-1)

4. Each \( P(\text{R/skin}), P(\text{G/skin}) \) and \( P(\text{B/skin}) \) are estimated through Gaussian probabilities.

5. Gaussian PDF is given as,

\[ F(x) = \frac{\exp\left(-\frac{(x-m)^2}{2\cdot\text{sigma}^2}\right)}{\text{sigma} \cdot 2\cdot\pi} \]  \hspace{1cm} (5-1-2)

where,
- \( F(x) \rightarrow \) Functional Gaussian probability estimate of data \( x \).
- \( x \rightarrow \) The sample data whose Gaussian probability has to be found.
- \( m \rightarrow \) The mean value of the object to be detected.
- \( \text{Sigma} \rightarrow \) Standard deviation.

Maximum likelihood estimate is then applied to the obtained Gaussian probability model to obtain the color object region from the video frame.

A non-parametric model of object detection uses minimum or no parametric estimate to perform object detection. An example of non-parametric model is histogram based object detection. For example suppose we want to detect green object, we know that each color image pixel is made up of overlapping red, green and blue intensity values, cancel out the green pixel intensity with that of the red pixel and also with the blue pixel intensity values, the formulation becomes,

\[ F(i,j) = 2 \cdot I_g(i,j,2) - I_r(i,j,1) - I_b(i,j,3), \]  \hspace{1cm} \text{where} \hspace{1cm} 1 \leq i \leq N, \hspace{0.2cm} 1 \leq j \leq M \hspace{1cm} (5-1-3)

- \( N \) and \( M \) are row and column lengths of the image \( I(N,M) \).
- \( I_g(i,j,2) \) is the index of the green pixel intensity and similarly for,
- \( I_b(i,j,3) \) is the index of the blue pixel intensity
- \( I_r(i,j,1) \) is the index of the red pixel intensity
- \( F(i,j) \) is the final functional value of the green color density distribution.
After the estimate of background modeling has been performed a correlation filter is used to remove salt pepper regions and to obtain smooth surfaced object detection from the entire background image. The steps illustrated above detects object in a single frame, In order to perform moving object detection frame differencing has to performed and the obtained results has to be multiplied with the single frame object detection. Frame differencing is performed as follows,

Let frame(n) be current frame, frame(n-1) be previous frame and frame(n+1) be next frame, then

\[
\text{Frame}_\text{diff} = \min((\text{frame}(n-1) - \text{frame}(n)), (\text{frame}(n+1) - \text{frame}(n))).
\]  

(5-1-4)

The experimental results are shown in section 9.

**6. Compressed Domain Object detection:**

The compressed domain approach exploits the encoded information of the video frame like motion vectors, discrete cosine transform (DCT) coefficients, and macroblock types which are generated as a compressed bitstream[3]. Compressed domain object detection can greatly reduce the computational complexity and make real-time or fast processing possible although it’s precision is not better than the spatial domain approach. The conventional compressed domain object detection algorithm uses motion vectors or DCT coefficients as resources in order to perform object detection and tracking[6]. These encoded data are not enough credible or insufficient to detect and track moving objects, but recent work uses a extra feature called Vector-featurated images that record moving regions and accumulate unmoving regions in which the moving objects are expected to exist after the current frame. This method uses only motion vector estimate without any other information from the encoded bit stream. In the vector featured images there are five types of block regions [2]. We consider the basic unit of a region in the vector-featured image is the motion vector value of the MPEG macro blocks. The five different blocks are reference block Br, current block Bc, background block Bb, moving block Bm and unmoving block Bu. The algorithm consists of four different steps initial region extraction, moving region detection, unmoving region creation and updation, moving object
tracking [2]. The results obtained are comparable to real time spatial object detection algorithm.

6.1 Initial region extraction:
Motion vector are present only in the regions where there is a B or P frame prediction. To represent an intra frame prediction or an I-frame, initialize an array of zeros to the frame size (in macro blocks). A vector featured image is generated by comparing the reference frame with the current frame and motion vector regions are marked as moving block, current block region labels. Figure shown below shows how such a setup work, the rectangular grids are the vector features images where each slices represents a frame and the units within each slice represents the block label obtained from macro block motion vector values. Only small portion of the vector featured image is shown in the fig 6.1, the region where object was spotted is only shown rather than showing the entire vector featured image.

Fig 6.1: Initial region extraction – each grid in the $f_n$ picture represents the motion vector of a macro block.[3]
6.1.1 Motion vector calculation:
Using the previous and the current frame as reference, the encoder finds the motion vectors for the forward and backward prediction frame. Each video sequence is divided into one or more group of pictures. The encoder outputs the motion vectors in the bit stream order. Only the motion vectors that are received in the decoder side are processed to find the moving object region.

![Diagram](image)

Figure 6.2: MPEG group of pictures – Display order [11].
Frames do not come into the decoder in the same order as they are displayed. The output frames from the encoder will be of the form I P B B P B I B B P B B. Where I is the intra coded frame, P is the predictor frame and B is the bi-directionally predicted frame. First, reorder the incoming frames or slice from bit stream order to display order. To reorder the frames to the display order the following procedure is followed,

- If an I or P frame comes in put it in a temporary storage called future.
- I or P is left in the future until another I or P frame comes in, on the arrival of a new I or P frame, already present I or P frame is taken out from the temporary storage called future and is put in the display order and the newly arrived I or P is put into the temporary storage called future.
- All B frames are immediately put in the display order.
- At the end whatever frame is left in the temporary variable called future is put in the display order.
Fig 6.3: Conversion from bit stream order to display order [11].

6.1.2 Algorithm

For ease of handling, the motion vectors are stored in a two dimensional arrays and the size of the array corresponds to the frame size in macro block (in this case 8x8 macro block was chosen). The forward prediction vectors and backward prediction vectors are stored in separate arrays. Each prediction vector in turn contains two more arrays to store the horizontal and vertical movement of vectors. To find the motion from one frame to another, a record of motion vectors of the previous frame has to be kept. Fig 6.4 shows the explanation of the algorithm in a flow chart. The process of inputting the motion vectors into correct arrays and reordering the frames into the display order were incorporated in the decoder.
Fig 6.4: Flow chart explaining storage of motion vectors in respective arrays. [11].

The final output of this algorithm stores the motion vectors of successive frames in an array. For finding the motion from frame to frame, the present and previous frame motion vectors are subtracted. If an ‘I’ frame is encountered all the values in the array are set to zero. If a B frame is encountered, forward prediction and backward prediction vectors are subtracted separately and an average total motion is found in both the horizontal and vertical directions.

The obtained motion vectors for each frame are written into a separate file each for horizontal and vertical motions.

6.2 Moving Region Detection:

A moving object generates non-zero motion vectors that appear continuously over multiple frames. The continuous appearances of motion vectors cause previous block and current block motion vectors to overlap, these regions are detected as moving regions Bm. When a current frame pointed to is an I-frame, then no motion vectors information will be there In the current frame and hence the contents of motion vector information from previous vector-featured image is copied to present frame, moving block is not created in these frames.
6.3 Unmoving region creation and updating:

The difficulty in detecting moving objects by using only motion vector information is that the probability of getting false positive (i.e) detecting a background as a moving object is high. To overcome this, methods such as connected component analysis and threshold value for motion vectors are implemented. A block which was having a motion vector in the previous frame may tend to a zero motion vector in the current frame. These regions are marked as unmoving regions as the moving object is expected to exist after current frame. When a moving object stops, zero motion vectors are generated, however just before the object stops, moving block regions will be created along the movement of the object whose current block has a zero motion vector. These regions whose previous motion vector value had a moving block related to the current zero motion vector is marked as unmoving block. The effect of the unmoving block is also crucial, there are two criterions that should be considered, one is that the unmoving regions should be considered correctly and a object which has moved in position after few frames should also be noted. For these issue unmoving block regions are monitored for some k frames, if a zero motion vector persists its brightness value is decreased gradually, if not it is marked as a moving block region.

7. Experimental results - spatio-temporal moving object detection:

The spatio-temporal object detection algorithm along with its performance metrics was implemented in MATLAB. A video sequences was considered during the testing and training phase. The Frame 70 and 120 have been worked upon to give Frame differencing, Green Color Distribution, Dilated output.
Fig 7.1: Frame Differencing

Fig 7.2: Green Color Distribution
Fig 7.3: Dilated Output

Fig 7.4: Output Image obtained by Green score times Frame Differencing
Fig 7.5: Frame # 70 with two detection boxes

Fig 7.6: Frame Differencing
Fig 7.7: Green Color Distribution

Fig 7.8: Dilated output
Fig 7.9: Output Image obtained by Green score times Frame Differencing

Fig 7.9.1: Frame # 120 with two detection boxes
8. Experimental results - moving object detection using motion vectors:

In order to represent the moving object detection using motion vectors, the direction vector plot of each frame is plotted as shown in fig 8.1.

The constraints that to be considered in compressed domain object detection using motion vector estimate is that, if the test video has moving background objects other than object to be detected then the accuracy of detection will be less.

Fig 8.1: Threshold motion vector values for frame #70
Using connected component analysis and setting up a threshold value, the regions of maximum displacement is obtained. The threshold value must be selected such a way that the important information from the motion vector should not be erased.

9. Conclusions:
The performance of moving object detection is evaluated using two different techniques. The moving object detection using motion vectors for Frame #120 gives better results than for frame #70. In frame #120 both the moving objects have been represented by the direction vectors. In frame #70 only motion of one object is represented clearly by direction vectors.
10. Future Work:

The accuracy of detection for the compressed domain case using motion vector values can be further improved by considering more parameters like the DCT coefficients associated with each motion vectors and introducing pre-processing and post processing steps that can be applied to increase the accuracy of detection.

References:


