WHITEBOARD IMAGE RECONSTRUCTION USING MATLAB

Chandika Dana Siva Sankara Rao[1], Kanuri Venkata Chaitanya[2], Kuppili Satish Kumar[3], Gedela Ramesh[4]

[1]*Final year Automotive Electronics VIT University Vellore, India*
[2]*Final year Automotive Electronics VIT University Vellore, India*
[3]*Final year Automotive Electronics VIT University Vellore, India*
[4]*Final year Automotive Electronics VIT University Vellore, India*

Abstract

Online video lectures are becoming commonplace in higher education. One problem is that the instructor might block what he has written on the board. In our project, we reconstruct the online lecture video so that the information on the whiteboard is always available to the viewers. The background subtraction, object tracking, and object replacement technique have been used to reconstruct the video. The complete access of the whiteboard information is obtained by replacing the lecturer/object with whiteboard information from the future frames or the past. The object is detected by using background subtraction method and is tracked by object tracking technique. The algorithm can be implemented for different board conditions and video resolutions.

Keywords- *background subtraction, object tracking, whiteboard reconstruction, object replacement, online lectures, and video transformation*

----------***----------

1. INTRODUCTION

The means of online communication has given us a great benefit in our lives. One useful application is the online lecture videos. More sophisticated lecture recordings can greatly enhance the learning efficiency of students, and the technology to provide effective lecture videos is being developed. One implementation, that has been developed, is a camera incorporated with an object detection technique so that it always follows the lecturer. Another approach can be made by implementing image processing techniques to the pre-recorded video files to make the video more efficient for the students. One implementation example might be recognizing the written texts on the board, translating them and saving as a document file.

There are many opportunities to improve other aspects of the video using image processing techniques. We recognized that the lecturer stands in front of the board for a considerable amount of time during the lecture, trying to explain a concept or having discussion with students. In this situation, instructor blocks the writings on the board. This makes online viewers difficult to see the whiteboard information behind the lecturer. A tedious way to deal with this situation is to just go back to the point where the board was available, pause for a while, and then resume.

This is a time consuming, discontinuous, and an inefficient way. However, having access to the video frames provides the possibility of removing the lecturer and making the whiteboard information behind visible. We propose a simple solution to provide full access to the whiteboard information even though the instructor is blocking it. For each frame, the object has been detected using a background subtraction method, and it is then replaced with whiteboard information from the future frames or the past.

2. PRIOR AND RELATED WORK

2.1 Background Subtraction and Object Detection

Subtracting a background template from a particular frame would only leave foreground objects that aren't present in the background template. Repeatedly doing this for a series of frames would essentially track objects as they move in the background template. In the context of this application, the background would be the board and surroundings, and the object would be the presenter. A big assumption that goes into this simple procedure is that the background template is to be a stable representation. That means that for frame j, the only thing that won't be part of the template is the object of interest. It also means that the subtraction procedure won't introduce art-factual objects into the detection.

Different methods have been proposed for background subtraction in the past. Some of the methods that we considered are frame to frame subtraction, mean/median background template, mixture of Gaussians and high-pass filtering. Each background subtraction implementation is discussed below.

2.1.1 Frame to Frame Subtraction

A very easy to implement algorithm where the background template used is just the previous frame. This method yielded accurate outlines of moving objects from frame to frame. However, the resulting images were very noisy and pixels that didn't change in brightness weren't detected. Also, when the object doesn't move the resulting image has no detected

object. One variations of this method include, adding multiple difference of frames by using frames further in the past as the background template. This does improve detection performance except when the object is moving faster than the frame rate, which yields undesired artifacts.

2.1.2 Mean/Median Background Subtraction

This method consists of taking a series of frames and taking the mean or median frame as the background template. The popular implementation of this method is using the previous n frames for calculation. This allows for an adaptive background template robust to slow light changes and outlier images. However, in the classroom setting where light conditions are stable and the camera doesn't move, calculating the mean/median of the whole sequence (or a long enough portion of it) is enough to have a stable background template representation. The median calculation is usually preferred since it isn't as susceptible to outlier frames, and this is the one that was implemented in this project.

2.1.3 Mixture of Gaussians Modeling

This model was a strong candidate for the background subtraction because it is suitable for multimodal background distribution. In order to take this model, it needs to have several modes pre-defined arbitrarily and to initialize the Gaussians updated over time. However, in this manner motion detection cannot be easily achieved because of the ambiguity of the Gaussian parameters.

2.1.4 High-Pass Filtering

In this method, multiple 2D high-pass filters are used to detect the edge information between background and foreground at each frame. High-pass filtered images provide many clues that can be utilized for differentiating background and foreground. However, this method is not only susceptible on the change of illumination, but also leaves limitations due to the unnecessary information distracting the object from the background.

More advanced methods were explored, like expanding the background representation into a higher dimensional space in a neural network framework, and kernel density estimations.

However, for this application very precise object detection wasn't needed and hence the simpler methods provided a sufficient level of detection.

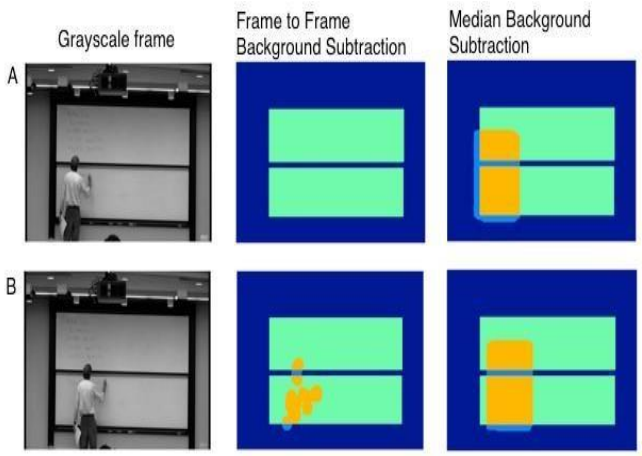

Fig-1: Comparison of Frame to Frame Subtraction and Median Background Subtraction in two different frames.

2.2 Dilation & Erosion

Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as dilation or erosion.

Dilation and erosion are often used in combination to implement image processing operations. For example, the definition of a morphological opening of an image is erosion followed by dilation, using the same structuring element for both operations. The related operation, morphological closing of an image, is the reverse: it consists of dilation followed by erosion with the same structuring element.

2.3 Object Tracking

The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. Object tracker may also provide the complete region in the image that is occupied by the object at every time instant. The tasks of detecting the object and establishing correspondence between the object instances across frames can either be performed separately or jointly. In the first case, possible object regions in every frame are obtained by means of an object detection algorithm, and then the tracker corresponds objects across frames. In the latter case, the object region and correspondence is jointly estimated by iteratively updating object.

Location and region information obtained from previous frames. In either tracking approach, the objects are represented using the shape and/or appearance models. The model selected to represent object shape limits the type of motion or deformation it can undergo. For example, if an object is represented as a point, then only a translational model can be used. In the case where a geometric shape

representation like an ellipse is used for the object, parametric motion models like affine or projective transformations are appropriate. These representations can approximate the motion of rigid objects in the scene. For a non-rigid object, silhouette or contour is the most descriptive representation and both parametric and nonparametric models can be used to specify their motion.

3. DESCRIPTION OF THE ALGORITHMS

3.1 Block Diagram

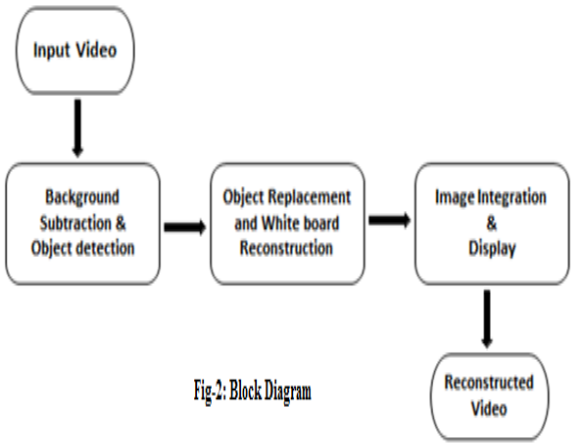

Fig-2: Block Diagram

Higher resolution lecture videos are usually sampled 40~50 frames per second. For computationally speed, simplicity and control purpose, the videos were compressed. The program MPEG Stream clip version 1.9.2 was used to reduce the original videos to 15 or 8 frames per second and the resolution of each frame is reduced to 480x720 or 240x320. For our processing stream each frame is further converted to grayscale.

3.2 Median Background Subtraction and Object Detection

The median/average method uses the average or the median of the previous n frames as the background image, B. It is quick but very memory consuming. The required memory is n*size (frame).

The running average can be calculated as follows:

$$Bi=aIi-1(x;y)+(1-a)Bi-1(x;y); \quad (1)$$

Where 'a' is the adapt or learning rate (usually chosen as 0.05).

The running average can be expanded to accommodate selectivity, as follows:

$$Bi(x;y) =Ii-1(x;y)+(1-a)Bi-1(x;y); \text{ if } Ii-1(x;y) \text{ is}$$

Background, (2)

$$Bi(x;y) =Bi-1(x;y); \text{ if } Ii-1(x;y) \text{ is foreground} \quad (3)$$

Selectivity just means that if the pixel was classified as foreground, it is ignored in the new background model. Using this we prevent the background model to contain a pixel believed to be foreground.

The background model at each pixel location is based on the pixel's recent history. The history can be the average of the previous n frames or the weighted average, where recent frames have higher weight. The weighted average is computed as the chronological average from the pixel's history and no spatial correlation is used between different (neighboring) pixel locations.

At each new frame, each pixel is classified as either foreground or background. The selective background model ignores any pixels which are classified as foreground by the classifier.

The final implementation made use of the median background template approach, in which we use the whole video sequence to create the template. After the background template subtraction, the resulting grayscale image is binarized to a threshold that allows do objecting detection and reducing noise. A binary board mask is created offline from the background template and intersected ('and-ed') with the detected object frame in order to only have objects that overlap with the board(s). Small objects are eroded, and the remaining object is dilated with a box shape to compensate for missed object parts. This step is crucial since object detection wasn't perfect from frame to frame. This box dilation will have board content surrounding the detected object, however since the intent is to replace the object with the board's content it won't degrade the quality. A downside of this over dilation is that the replacement algorithm will need to do more searching. After the object has been dilated, a four level image is constructed for each frame, in which the dilated object is added to 2 times the binary board mask. The four levels are as follows:

level 0 - background, ***level 1*** - object off whiteboard, ***level 2*** - empty whiteboard, ***level 3*** - object on whiteboard. The object detection is illustrated in Figure 2 on the left and examples of four-level- images can be seen in the four right panels.

The video is converted into frames and each frame is further converted to gray scale and these are stored. Median frame for these gray frames is calculated, now the median frame is subtracted from the stored gray scale frames. The subtracted frames are binarized. A board binary mask is created and intersected ('and-ed') to subtracted frames. Small objects are eroded, and the remaining object is dilated with a box shape to compensate for missed object parts. This step is crucial since object detection wasn't perfect from frame to frame. This box dilation will have board content surrounding the detected object. The binary mask is multiplied with two and added to output of box dilation this gives four level frames.

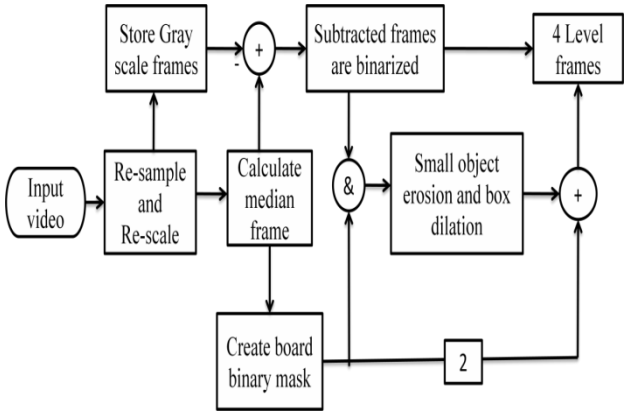

Fig- 3: Image processing algorithm diagram of object detection.

3.3 Object Replacement Algorithm

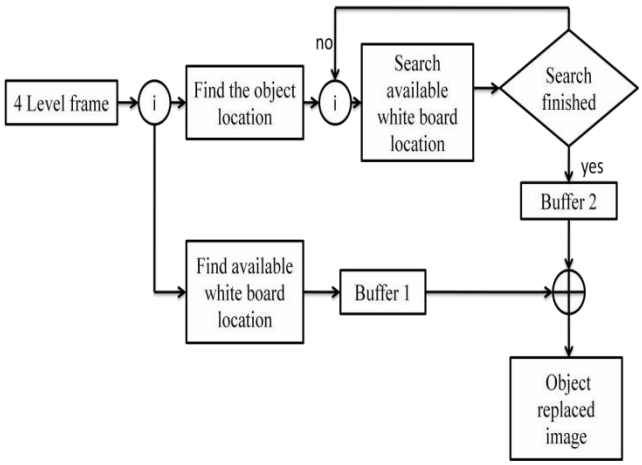

Fig- 4: Image processing algorithm diagram of object replacement

The reconstruction algorithm uses the four-level image frames to transform the original movie frames into object-replaced-image frames. Pixels of the whiteboard blocked by the object are indicated as level 3, and pixels of the whiteboard available to the viewers are marked as level 2 in the four-level-image. The level 3 part of a movie frame is replaced with the level 2 pixels from the nearest movie frames.

The reconstruction process is done in a time sequence. For the current i^{th} frame, the algorithm first finds level 2 pixels, and stores them into a buffer. Then, it starts searching for level 2 pixels in the other frames that corresponded to the level 3 pixel indices in the current i^{th} frame. When it finds those pixels, the buffer is updated, and checks if all the level 3 pixels are replaced. If all level 3 pixels are replaced, it moves on to the next $(i+1)^{th}$ frame. Otherwise, it keeps looking for the rest of pixels.

One of the major questions to be answered in reconstruction process is how do we decide which frames to look for in order to replace the currently blocked whiteboard pixels. We set the initial search direction to the future frames, and limited the number of frames to search for to be less than a maximum search range. If the algorithm failed to replace the object within the range, it uses previously reconstructed $(i-1)^{th}$ object-replaced-image frame. Thus, the algorithm replaces the object with the most relevant whiteboard information to the current i^{th}. Another important issue is the speed of the reconstruction algorithm. Most online lecture video files are high resolution and are sampled at high frame rate. Thus, reconstructing the whole video is a computationally heavy process. One way to resolve this problem is to set an update rate of 1~5 seconds for the reconstruction. Naturally, the object won't make a quick, significant movement to another position within several milliseconds, so that it is not necessary to run the reconstruction algorithm for every frame. In our simulation, we took samples every 3 second to run the algorithm, which made considerable improvement in the speed.

The sampling process may lead to some quality problems. The junction of the replaced and original pixels created artifacts, such as discontinuity. We handled this issue by detecting the edge contour of the replaced pixels, dilating them with a square filter, and interpolating them. The square filter was set to be as small as possible in order to avoid smoothing effects on the writings. A more notable enhancement was made at the image integration step, where we mixed the original movie frame and the reconstructed image frame, making the object translucent. This original image addition process made the junction more continuous. Other problems such as remainder parts of the object were dealt simply using larger box dilation.

3.4 Image Integration and Display

Once we successfully replace the object area with the relevant whiteboard information, we can build the integrated image frame, fi(n) by combining the original movie frame, fo(n)and the object-replaced image frame, fr(n.)

for i = first frame : update rate : last frame create a buffer for object-replaced-image of frame **i** find available-whiteboard location (level 2) store to the buffer find object-blocked-whiteboard location (level 3) **until** object-blocked-whiteboard pixels are all replaced **do if** search range <= maximum search range

$$Fr(n) = w*fo(n) + (1-w)*fr(n)$$

Where *w* is a weighting factor that determines the strength between the object and the replaced information. If we choose *w*=0, the movie displays only the object-replaced image whereas *w*=1 shows the original movie frames.

4. PROJECT IMPLEMENTATION

4.1 Algorithm Pseudo Code

- input : *four-level-image*, original image
- output : object-replaced-image
for i = first frame : update rate : last frame
create a buffer for object-replaced-image of frame **i**
find available-whiteboard location (level 2)
store to the buffer
find object-blocked-whiteboard location (level 3)

until object-blocked-whiteboard pixels are all replaced **do if**
search range <= maximum search range
go to the future frame
elseif search range > maximum search range
go to the past *object-replaced-image* frame
end if
search *available-whiteboard* location
store to the buffer
end until
end for

5. RESULTS

Fig- 5: Result obtained. (a) Original Movie Image (b) Integrated Image (c) Object-Replaced Image

For the input lecture video the content on the board is integrated in the place of the object by searching the previous & future frame. The algorithms are based on the classroom environment where camera is located in a fixed position and covers a fixed range of angle.

6. CONCLUSIONS AND FUTURE SCOPE

In the project the algorithms for background subtraction, object detection and replacement, and image integration technique can successfully disclose whiteboard in online video lectures. These algorithms are robust and automatically processed in MATLAB. In the future, these algorithms can be applied in many online lectures or telecommunication conferences in real-time.

REFERENCES

[1] He L., Zhang Z. Real Time Whiteboard Capture and Processing Using a Video Camera for Teleconferencing. ICASSP2005.

[2] Wienecke M., Fink G.A., Sagerer G. Towards Automatic Video-based Whiteboard Reading. ICDAR 2003.

[3] Maddalena, L., Petrosino. A. A Self-Orginizing Approach to Background Subtraction for Visual Surveillance Applications. IEEE TIP 2008.

[4] Piccardi, M., 2004, Background Subtraction Techniques: A Review Systems, Man and Cybernetics, 2004 IEEE International Conference.

[5] PRACTICAL IMAGE AND VIDEO PROCESSING USING MATLAB® : Oge Marques , ieee – wiley press, 2011.

BIOGRAPHIES

Chandika Dana Siva Sankara Rao
Pursuing M.Tech Automotive Electronics in VIT University.
Completed UG in Electronics and Communication Engineering.
Currently working towards master degree in automotive electronics with a focus on areas of Safety, Body Electronics and In-Vehicle networking, Infotainment.

Kanuri VenkataChaitanya
Pursuing M.Tech Automotive Electronics in VIT University.
Completed UG in Electrical and Electronics Engineering.
Currently working towards master degree in automotive electronics with a focus on areas of Safety, Body Electronics and In-Vehicle networking, Infotainment

Gedela Ramesh
Pursuing M.Tech Automotive Electronics in VIT University.
Completed UG in Electronics and Communication Engineering.
Currently working towards master degree in automotive electronics with a focus on areas of Safety, Body Electronics and In-Vehicle networking, Infotainment

Kuppili Satish Kumar
Pursuing M.Tech Automotive Electronics in VIT University.
Completed UG in Electrical and Electronics Engineering.
Currently working towards master degree in automotive electronics with a focus on areas of Safety, Body Electronics and In-Vehicle networking, Infotainment