Whiteboard Disclosure using Background Subtraction and Object Tracking

Alex Gonzalez, Bongsoo Suh, Eun Soo Choi
Department of Electrical Engineering, Stanford University

***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 movie. The algorithm is successfully implemented for different board conditions and video resolutions. Our implementation provides a useful feature to students who take online courses.**

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

I. 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 [5]. 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. In this report we go into the details of the different steps that allow this procedure to work.

II. Prior and Related Work: 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 artifactual objects into the detection.

Different methods have been proposed for background subtraction in the past [2]. 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:

- 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.
- Mean/Median Background Subtraction: This method consists of taking a series of frames and taking the mean or median frame as the background template [2]. The popular implementation of this method is

This manuscript was prepared for the final project in EE 368: Digital Image Processing at Stanford University and was submitted on June 6, 2012. Alex Gonzalez (email: alexg8@stanford.edu), Bongsoo Suh (email: bssuh@stanford.edu), and Eun Soo Choi(email: eschoi@stanford.edu) are a graduate student in Electrical Engineering at Stanford University.

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. A comparison of the Frame to Frame subtraction and Median subtraction is shown on Figure 1.

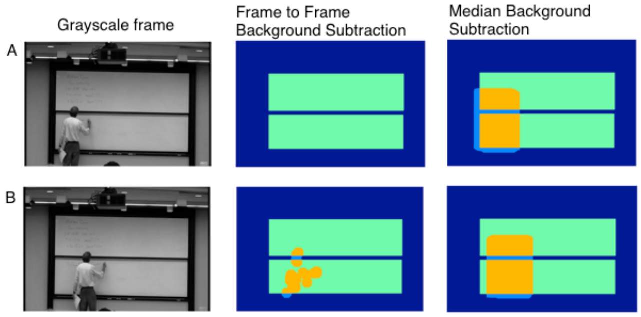
Grayscale frame
Frame to Frame Background Subtraction
Median Background Subtraction
A
B

Figure 1. Comparison of Frame to Frame Subtraction and Median Background Subtraction in two different frames. The difference is dilated in order to cover the whole object. Median background gives as consistent result. In A, Frame to Frame isn't able to detect the object.

- **Mixture of Gaussians Modeling:** This model was a strong candidate for the background subtraction because it is suitable for multimodal background distribution [2]. 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.
- **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 [3], and kernel density estimations[2]. However, for this application a very precise object detection wasn't needed and hence the simpler methods provided a sufficient level of detection.

III. Description of the Algorithms

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 Streamclip* 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. Four test video sequences were used to test our script taken from EE261 lectures, provided by Derek Pang. A breakdown of each test video sequence follows:

1) Video1: 15 frames/sec, 480x720 resolution, 1 minute long, high contrast between object and board (EE261 2010)
2) Video2: 8 frames/sec, 240x320 resolution, 7 minute long, high contrast between object and board (EE261 2010)
3) Video3: 15 frames/sec, 480x720 resolution, 1 minute long, low contrast between object and board (EE261 2011)
4) Video4: 8 frames/sec, 240x320 resolution, 7 minute long, low contrast between object and board (EE261 2011)

A. *Median Background Subtraction and Object Detection*

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 to do object detection and reduce 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 (section 4.2) 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 of Figure 1.

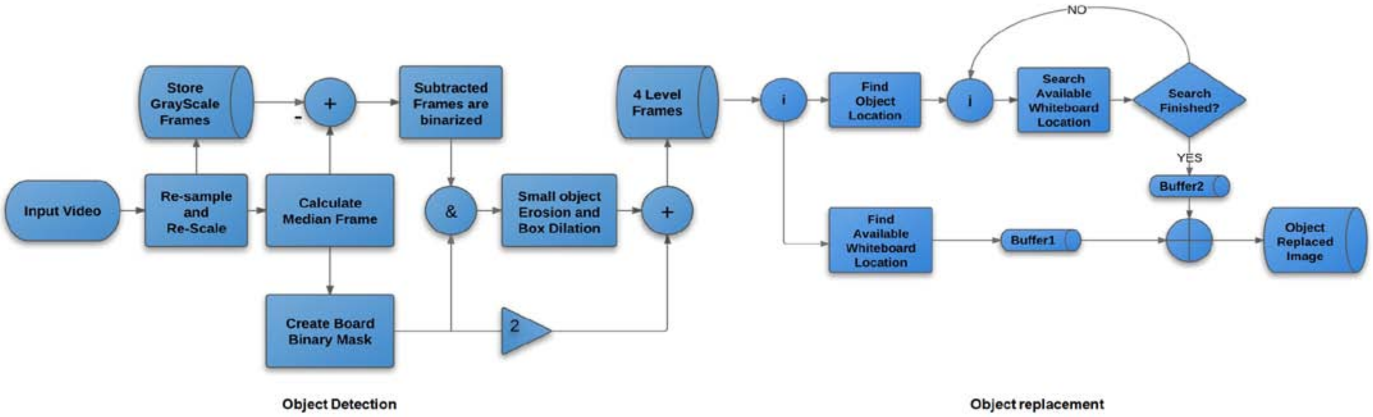

Figure 2. Image processing algorithm diagram of object detection and object replacement. It takes an original input image and convert it into object-replaced image

B. Object Replacement and Whiteboard Reconstruction

Our 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 ith 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 ith 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+1th 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-1th object-replaced-image frame. Thus, the algorithm replaces the object with the most relevant whiteboard information to the current *i*th frame, and makes the reconstructed movie more natural.

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 was dealt simply using a larger box dilation.

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
```

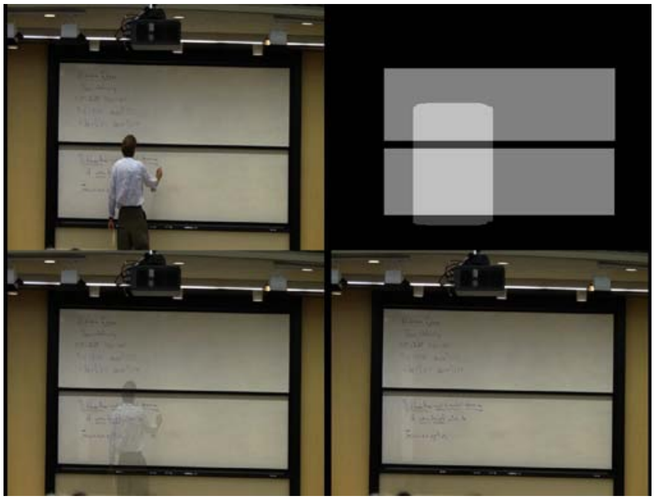

Figure 3. Classroom 1 Whiteboard with low contrast between the board and the object. (a) Original Movie Image () Four-Level-Image (c) Integrated Image (d) Object-Replaced Image

C. *Image Integration and Display*

Once we successfully replace the object area with the relevant whiteboard information, we can build the integrated image frame, $f_i(n)$ by combining the original movie frame, $f_o(n)$and the object-replaced image frame, $f_r(n)$

$$f_i(n) = w \times f_o(n) + (1-w) \times f_r(n)$$
$$0 \le w \le 1$$

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.

IV. Results

We tested our algorithm for two different classroom environments as shown towards the four different video settings with low resolution (240x320) and high resolution (480x720) and with low contrast and high contrast between object and board. The algorithms are based on the classroom environment where camera is located in a fixed position and covers a fixed range of angle.

In some situations, we observed that object detection was not successfully performed for a short time. The main problem which we faced was when the professor stretched his hand too far from his body. This problem could be resolved by enabling the dilation box size to be adaptable and sufficient to cover the object size in video lecture. However, there were some situations we have not taken into account. For example, when two boards are switched up and down, the error occurs for one to two seconds and becomes stable immediately after the shift finishes. Also, if there exists interference, such as a student passing by a camera, the reconstructed image frame gets suffered temporarily. Except for those cases, the algorithms are proven as being robust and successful in general.

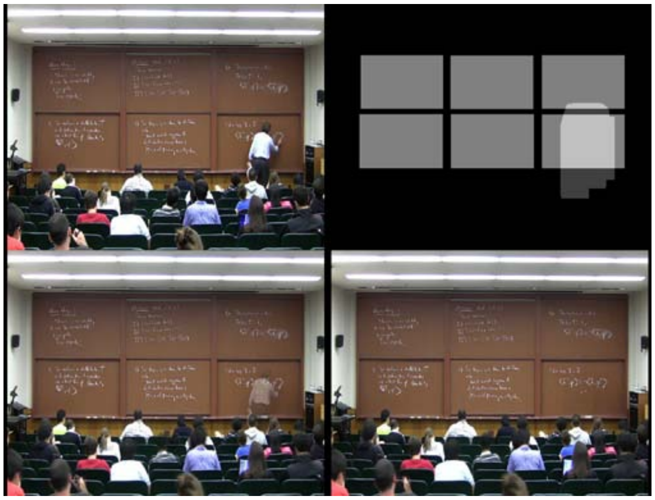

Figure 4. Classroom 2 Brown-board with high contrast between the board and the object. (a) Original Movie Image (b) Four-Level-Image (c) Integrated Image (d) Object-Replaced Image

The captured images, Figure x and Figure x, show four different steps of images under the two different classroom environments. In each figure, (b) represents the four-level-image after the original image, (a), is processed by background subtraction and the object detection. (d) describes the object-replaced image and (c) is the integrated image of (a) and (d) with the weighting factor w=0.3 here. In Figure x, the classroom is subjected to the low contrast as the colors of the whiteboard and the object wearing the white shirt give low contrast whereas Figure x has high contrast between the brown-board and the object. The algorithms perform in both cases of low and high contrast environments because the median background subtraction method is invariant on image contrast.

V. Conclusions

In this paper, we have shown that 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. We also have demonstrated these techniques using video lectures currently opened in Electrical Engineering at Stanford University which are subjected in various classroom environments with different video qualities. In the future, these algorithms can be applied in many online lectures or telecommunication conferences in real-time when they are compiled outside of MATLAB with sufficient memory capacity.

ACKNOWLEDGMENT

We thank professor Bernd Girod for his wonderful lectures, leading us to explore the amazing field of digital image processing. We also thank the head TA David Chen and our

APPENDIX

Alex contributed on background subtraction and object detection. Bongsoo worked on object replacement. Eun Soo worked on background subtraction and image integration. All of us worked on proposal, poster, and report.

REFERENCES

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

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

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

[4] Lecture notes of the class EE368 Digital Image Process

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