Content-based lecture video indexing

Martin Røst Halvorsen

HØGSKOLEN I GJØVIK

Master's Thesis
Master of Science in Media Technology
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2007

3 Implemented methods

This chapter will focus on details in the implementation of this thesis work.

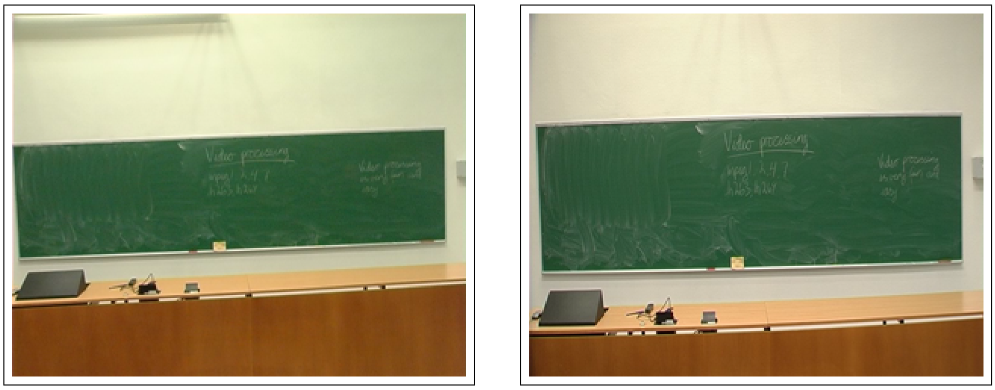
3.1 Experimental setup

In this chapter we will define the setup environment and the equipment used.

3.1.1 4:3 vs. 16:9 lenses

A blackboard is usually very wide but not that high. Using a video camera with 4:3 standard lens will capture lot of uninteresting areas above and beneath the blackboard when the whole blackboard is visible. Zooming in so that these uninteresting areas are in minor will capture little of the blackboard in width. A 16:9 lens will therefore be a little better in this case because of the wide angle.

The following images show the difference between a 4:3 (figure 16(a)) and 16:9 anamorphous (figure 16(b)) lens. Since the video camera used is a PAL 4:3, an anamorphous lens has to be used to get a 16:9 view. When capturing with the 16:9 anamorphous lens the camera can be placed closer to the blackboard since more information is compressed into the same view.



(a) 4:3 format lens

(b) 16:9 format anamorphous lens

Figure 16: Camera lenses

Using an anamorphous lens is not as good as using a 16:9 video camera because objects in the frame is curving, and the text might appear blurred. To get best quality of the video a 16:9 camera should be used to capture most of the blackboard. We only have a 4:3 camera available and will use that one, without the anamorphous lens.

3.1.2 Environment setup

The experimental setup in this thesis will be in an auditorium where spotlights are places high under the roof. This is due to get good lightning conditions on the blackboard. Lectures will be recorded with a Panasonic DV camera with a resolution of 720px width and 576px height, and a 4:3 PAL lens. The setup should be as simple as possible so that everyone can re-create it. Therefore the camera is places in a fixed and static position on a tripod. There should not be any need to pan or tilt the camera. Only half of the blackboard is being used so that the content on the blackboard is recorded best possible. This is done in order to be able to extract and read the content on the blackboard. Figure 17 shows how a frame will look like. This will coerce a limited area of movement for the teacher. Another approach might be to use two cameras, each recording each half of the blackboard, but then the videos needs to be stitched together and synchronized [34].

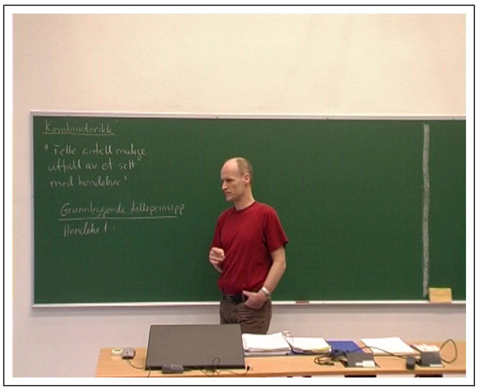


Figure 17: Lecture recorded using half of the blackboard

Focus patterns are used to focus the camera on the blackboard properly. These patterns are available for download in different formats on the Internet and are easy to use [35]. It is important to work with videos that are in focus because of small and important details that are going to be tracked. Text on the blackboard might not be as good to read initially so it will not help recording with an out-of-focus camera. Figure 18 shows a focus pattern.

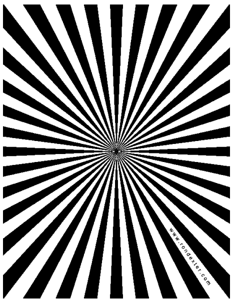


Figure 18: Focus pattern

The camera settings are set to 25p, 25 frames per seconds with progressive scanning. Using interlaced scanning is not good because of the interlaced lines which can cause flicker and distortion. A progressive frame contains one whole current frame.

3.1.3 Development environment

The development and test environment will be in MATLAB®[36]. MATLAB®has a software tool called Simulink which allows you to create a model-based design and a simulation environment to test the model. Simulink has different kinds of blocksets. The Video and Image Processing Blockset and Signal Processing Blockset have most of the blocks that is needed in this research. The videos used are transferred from a DV camera through FireWire to Adobe Premier and saved as a DV AVI file. These files can be processed in Simulink, but will not work with the function aviread in MATLAB when the file size goes above 1,5-2 GB. The video files are after processing in SimuLink saved as AVI video files.

3.1.4 Overview of proposed system

Before getting into details of each part of the system, a superior block diagram is presented in figure 19. The system consists of three main subsystems. The first one is used to separate the foreground and background. After this process the teacher should have been removed. The next subsystem extracts content from the blackboard, such as text and drawings. In the last subsystem key frames are extracted. These key frames may be used in an index for a lecture video.

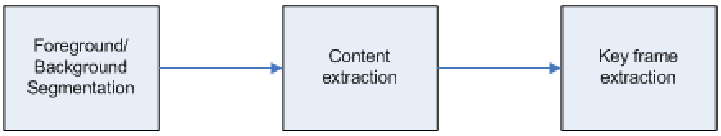


Figure 19: Superior block diagram of the system

3.2 Foreground segmentation

There are two mainly used methods to track foreground objects like for instance the teacher. One is to model the background and subtract this background model from a frame, and the second one is to use motion estimation. In a lecture the teacher is often moving around in front of the blackboard to write, explain and emphasize something. Using motion estimation is not that time consuming compared to background modelling which often need updating and statistical calculations. Also motion estimation is not dependent on a clear background for initialization. Motion from the teacher could be enough to detect and afterwards remove the teacher.

3.2.1 Motion Estimation

Motion estimators are used to track motion. They are used in several video surveillance systems to track objects like people or cars. A downside of motion estimators is that objects become part of the background when it doesn't move anymore. In general a teacher is moving when they write on the blackboard, or remove words. When they don't do this, they might stand still. Indexes in this video are going to be extracted for instance right before the blackboard is being erased, and the teacher should not be visible at this time. This is discussed further in chapter 3.4. We are going to look at two different motion estimators; Sum of Absolute Difference and Optical Flow.

Sum of Absolute Difference

The Sum of Absolute Difference (SAD) is often used in motion estimation to find where there is motion in a frame. This algorithm is based on the assumption that illumination is static and the camera is stationary. A high level block diagram of the SAD subsystem is shown in figure 20.

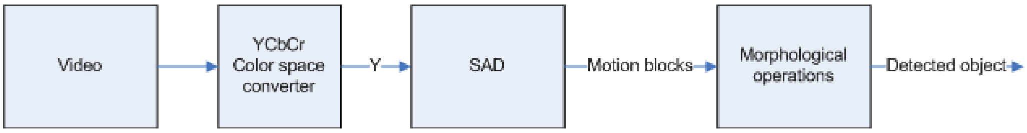
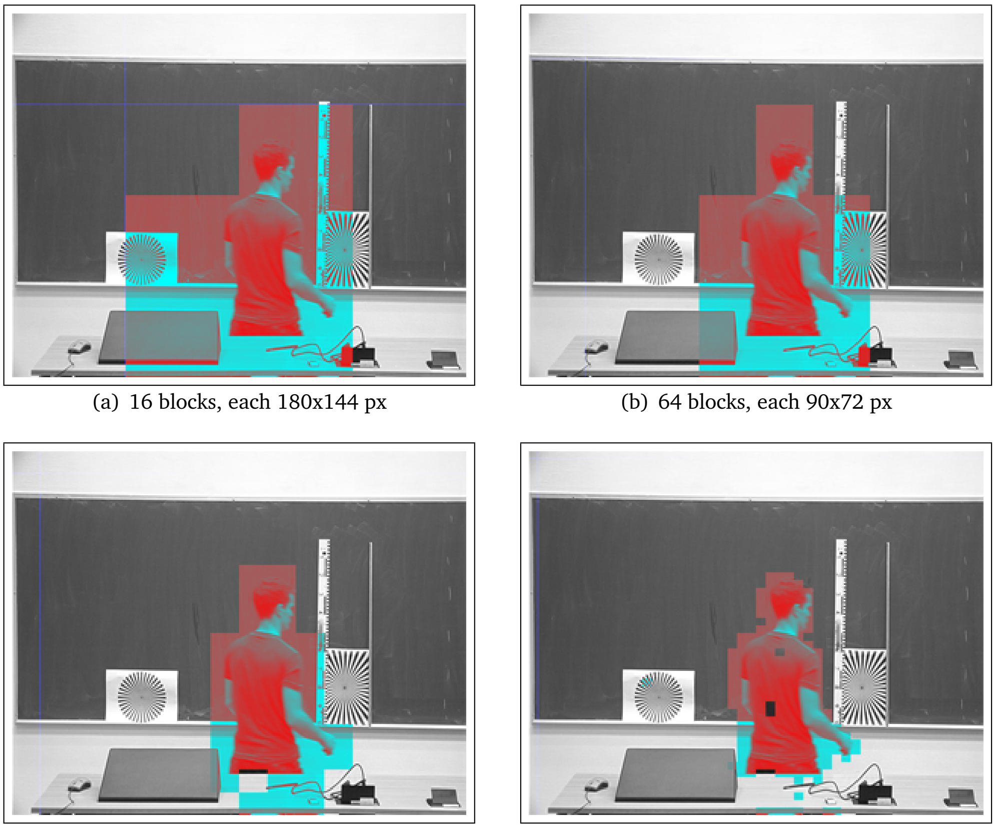


Figure 20: hig level block diagram of SAD subsystem

SAD calculates the difference between consecutive frames to find similarity, or differences. The greater the similarity is, the smaller the SAD values are. If we for instance have frame1 and frame2, where frame1 is a background frame, and frame2 has the same background and in addition a person in it, then the SAD value will vary in different regions. This can be illustrated by looking at figure 9 where there are smaller patches in moving areas. If we calculate SAD values in an area where the person is, the value would be high. Outside the area of the person the values would be very low or zero. To be able to detect objects that are moving the SAD algorithm has to be performed on different areas in a frame. This could be done by dividing each frame into equal block sizes, where each block gets its SAD-value. Based on a threshold value it is determined whether there is a significant change in the current block from the previous one. When a significant change occurs in one block, that block can be treated as a block with moving objects in it. The size of the blocks depends on what kind of objects to detect and how accurate the detection should be. An implementation that uses SAD is done earlier by Vanne et al. [23], and a similar approach of the SAD algorithm is implemented by Guo et al. [15].

The images in figure 21 shows motion detection with SAD using different block sizes. Using 16 blocks, as in figure 21(a), gives a very roughly detection of motion. This is due to very big blocks and will result in regions of the background scene being is detected as moving objects. Increasing the number of blocks to 64, as in figure 21(b), gives a somewhat, but still too much background information is detected. A good result of the detection can be seen in figure 21(c) where a more accurate detection is done using 256 blocks. Still the blocks are a little big, which can cause moving objects to be visible if it is only covering a little area in one block. Figure 21(d) shows a more accurate detecting of the moving teacher, using 2304 blocks total. One thing to take into consideration is when the teacher is writing on the blackboard, the text should be visible short after being added. A better segmentation can be achieved by using small blocks in order to keep as little of the background as possible away from being detected as a moving object. As discussed earlier objects might get part of the background when moving slowly, and cause holes in the segmentation. These holes should be closed and will be discussed in the following. Further testing with the SAD algorithm is performed by dividing the frames into 2304 blocks, each 15 by 12 pixels.



(a) 16 blocks, each 180x144 px

(b) 64 blocks, each 90x72 px

(c) 256 blocks, each 45x36 px

(d) 2304 blocks, each 15x12 px

Figure 21: SAD with different block sizes

Close holes of detected objects

When using motion detection there is a possibility that there might appear some gaps. For instance a teacher that have a one colour t-shirt and moves slowly to one side, some blocks that represents parts of the t-shirt may not register any motion. Moving from one spot on the t-shirt to another might not cause any changes in one or more blocks, and therefore no motion is detected in these blocks, as seen in figure 21(d).

In order to segment out any moving objects in a frame, these holes or gaps need to be closed. One global way to do this is to create bigger changes between frames. This can be done by for instance only considering every 12th frame by down sampling the video. The changes in each frame will then be bigger, also where there are small movements. Figure 22 shows the detected blocks with motion after down sampling. Since the SAD algorithm takes the difference between two consecutive frames, and with a down sample of 12, there will be detected changes from where the teacher was in the previous frame and where the teacher is in the current frame. Therefore there are ‘two’persons detected in figure 22. This will not be any problem since the teacher is not moving that much while adding text. The down sampling of the movie will output a video with only 2 frames per seconds, but there are not happening much important things in a lecture in a half a second.

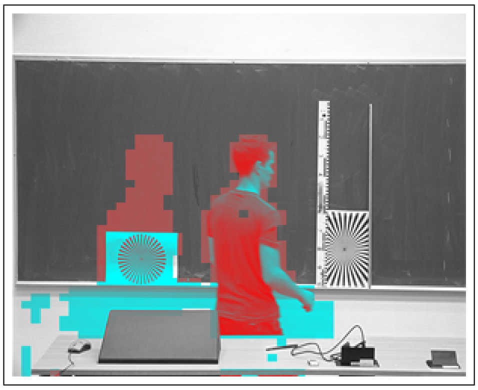


Figure 22: Motion estimation on every 12th frame

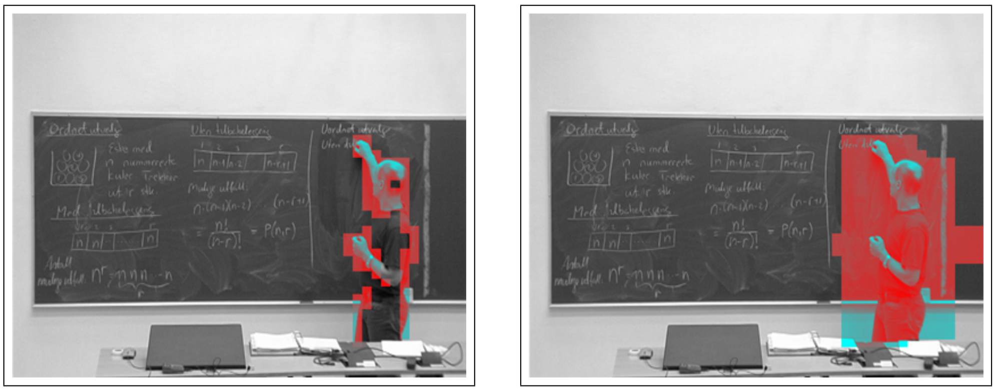
A more local way to close holes is to create a vector of 2304 indexes, one for each block. This is a boolean vector which holds the value 1 for the blocks with motion and 0 for those without motion. These holes can be cloes by going through the vector and say that: if you find a block with motion, go maximum n-blocks forward. If you find another block with motion before reaching the maximum n-blocks, set every block between these blocks to 1 (motion). N-blocks can for instance be 12 blocks, or more or less depending on the block sizes. Doing this in the whole frame will close gaps, in a column-wise way, in situations where there is small motion over a period of time. This method may also detect some parts of objects that initially may not be detected due to low representation in a block. Usually this is contour or edges of objects.

Another approach is to represent the motion vector as a binary frame and use morphological operations to close holes. Morphological dilation can be used to grow the white pixels so that they cover their surrounding undetected blocks. Following the dilation by morphological closing may give better results in closing holes in the detection. Closing may be used to connect unconnected parts in the binary frame.

Figure 23 shows the detected teacher before and after implementing the three above mentioned methods.

Motion history

A downside of using motion estimation for segmentation is that objects get visible when they don’t move. In a lecture the teacher does not move at all times, and will generate



(a) Before closing holes

(b) After closing holes

Figure 23: Morphological closing to close holes in the detection of objects

noise as parts of the teacher might get visible. Figure 24 shows a frame where parts of the teacher do not move for a period of time and are not detected, and these parts will then be visible.

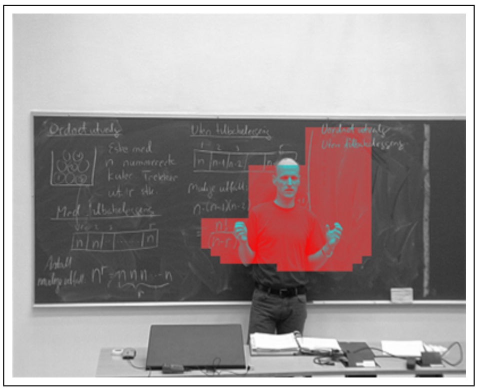


Figure 24: SAD blocks when teacher generates small motion

A solution to this problem can be to keep track of motion in time. By using a motion history vector that increments or decrements a block depending on there is motion or not in that block. If one block has had motion for 5 frames then that block will have the value 5. If the same block after these 5 frames has no motion for 3 frames then the value for that block will be 2. Figure 25 shows a motion history vector graph. The history graph will descend for each frame in a block when there is no motion and when motion is applied to a block the history graph will ascend. Since there are 2304 blocks many lines are on top of each other in the graph.

This history vector might help in the segmentation process when the teacher stands still for a period of time. By storing the block when it has its lowest value in the history vector then the segmentation will do a better job. Imagine that the block at point 1 in Figure 26 contains the background. The teacher generates motion in this block so that it increases to 1 at point 2. The teacher is still moving and creates motion in that block so it increases to 2 at point 3. Now there is not detected any motion so the value decreases to

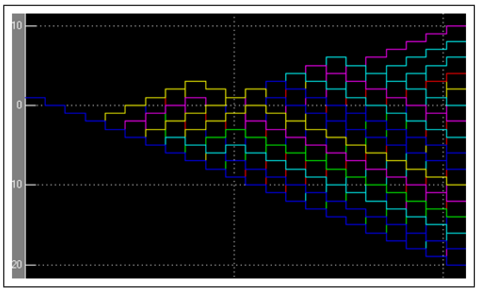


Figure 25: Motion history graph

1 at point 4. Since point 4 is not equal to or lower than point 1, that block is considered as still in motion. There is still no motion in that block and the value decreases to 0 at point 5. At point 5 the value is equal to the value in point 1, so finally, that block can be considered as having no motion in it.

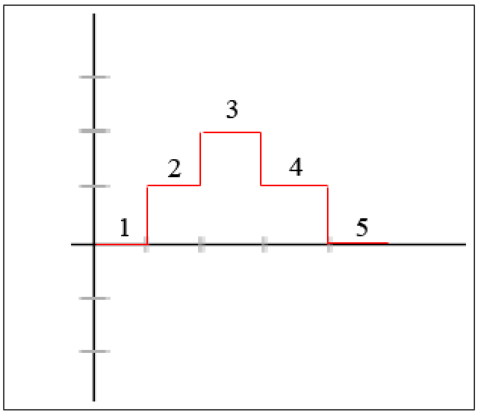


Figure 26: Motion history illustration

There is one thing to take into consideration using this motion history. If for instance block x has motion in it for 5 frames, the history vector for this block would have ascended 5 steps. In the output video we would have stored the pixel values for the first frame, since the first frame has the lowest motion history. If then the teacher moves away from this block it would take 5 frames for the motion history to be equal to the previous lowest value. In other words, it takes 5 frames before the background is updated. If the teacher has written something in this block then the text would not appear until 5 frames have passed. This would create a delay before important background appears.

One way to fix the delay is to saturate the history values. This means that a minimum and a maximum value is defined for which values the motion history vector can have. Instead of increasing the history value for one block to +30 when the teacher moves in this block for 30 frames, we could say that the value should not increase above 2, and not decrease below -2. This also means that the highest amount of frames that can pass before the background appears is 4.

Saturating the graph will also affect the segmentation. If for instance the teacher's arm do not move for 4 frames, then the arm becomes part of the background. Using a higher saturating value would hide the arm for the 4 frames when it was not moving. Choosing the right value for the saturation is important to maintain a good segmentation, but also to get the current background of the blackboard visible as soon as the teacher moves away.

Optical flow

Another way to estimate motion is to use optical flow. Optical flow is different to SAD in that it is based on vectors instead of only changes in grey or colour pixel values. It also calculates speed and directions of objects. Optical flow is tested to see if it can perform better than SAD in a indoor lecture environment. A superior block diagram of this subsystem is shown in figure 27.

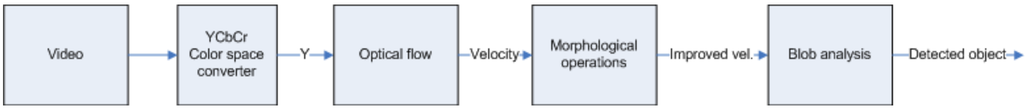
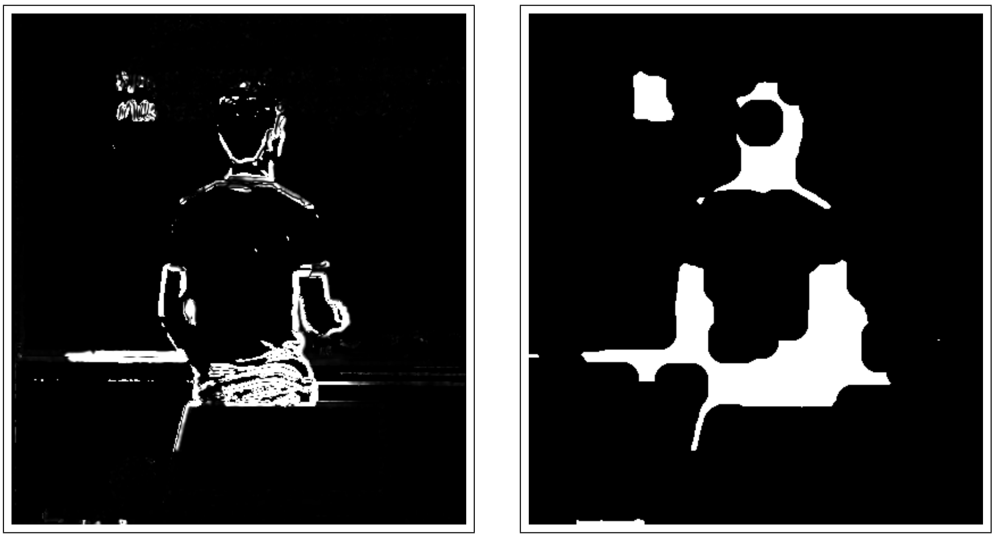


Figure 27: Superior block diagram of optical flow subsystem

An optical flow estimator where implemented which calculates optical velocity in the whole frame. The output of this process is a binary frame with blobs that represents moving objects, or changes between consecutive frames. A blob is a connected area of white pixels in a binary frame. For instance, one circle of white pixels is one blob, but a circle cut in two pieces are two blobs. These blobs can be tracked in order to find objects of interest, for instance the teacher, by performing blob analysis. The blob analysis process looks for connected white pixels, like a blob. The optical flow estimator has the same dependency of motion as the SAD algorithm.

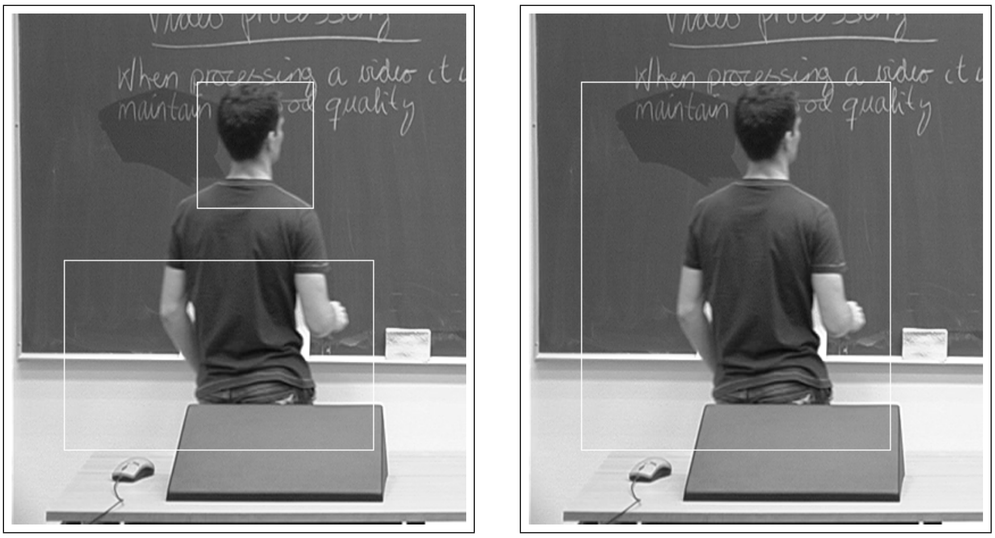
Figure 28(a) shows the optical velocity which is calculated by the optical flow estimator. To better track objects there is calculated a median velocity threshold value for the frame. In order to avoid detecting small noise as parts of moving objects, only blobs with a minimum size of 2000 pixels are considered, and to reduce noise a 3 by 3 median filter is used. Morphological closing is performed to better mark the objects. Figure 28(b) shows the frame after these steps. As seen in the images parts of the teacher is not detected, there are some gaps in the detection. Blob analysis looks for connected regions, and sometimes the teacher might be divided into several blobs like in figure 29(a). To be able to fill the gaps between the blobs we can merge blobs that are close to each other like figure 29(b) illustrates. This merging causes that more of the background is treated as a moving object. Maybe a better way would be to just stretch one of the rectangles defining the blob area close to the other one. However, this would in some cases not mark whole objects. As seen by figure 29(a), streching the blob that detected the head will not cover the teacher's shoulder. If the blob that detected the teacher's body is streched up to the other blob, it will also not cover the shoulder.



(a) Before closing

(b) After closing

Figure 28: Morphological closing on optical velocity

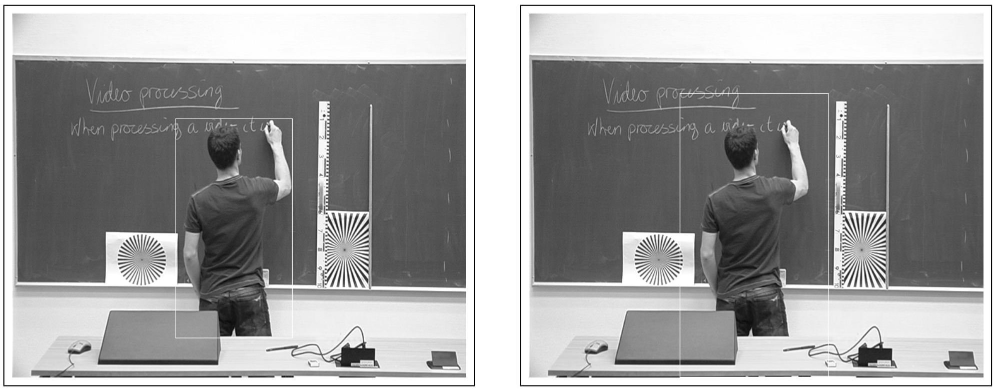


(a) Two bigger blobs found

(b) Merged blobs

Figure 29: Found objects based on blob analysis

Sometimes parts of the teacher get visible due to low optical velocity. Some parts of the teacher's outline are not detected and will then be part of the background. In several occasions the top part of the teacher's head was not detected. Sometimes this also happened with the teacher's elbow. One way to solve this might be to extend the detected area with a fixed offset value. This however is not a good solution as background information will be detected as moving objects, and it will take a longer time for text to appear when being written. The difference can be seen in figure 30 where the white boundary is the coordinates around the detected object's extremities.


(a) Without extended offsets

(b) With extended offsets

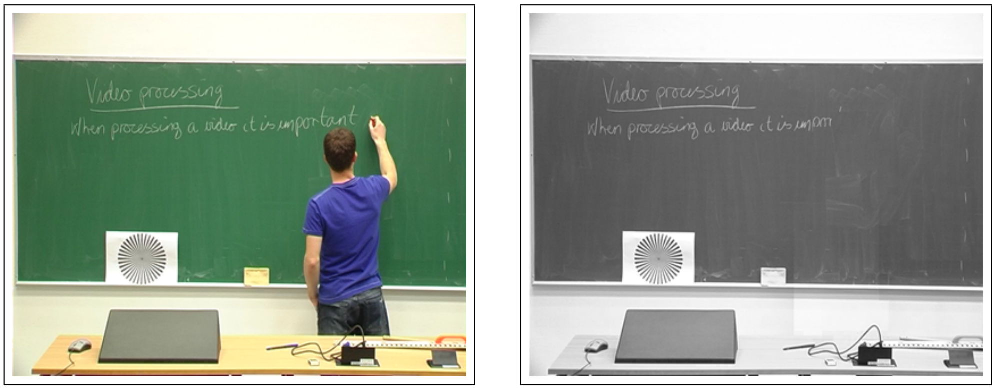
Figure 30: Optical flow detection with and without extended offsets

3.2.2 Remove detected objects

To be able to remove moving objects a replacement of regions which have detected moving objects in it has to be performed. In the case of SAD, based on the motion history vector, we know which regions or blocks to replace and when.

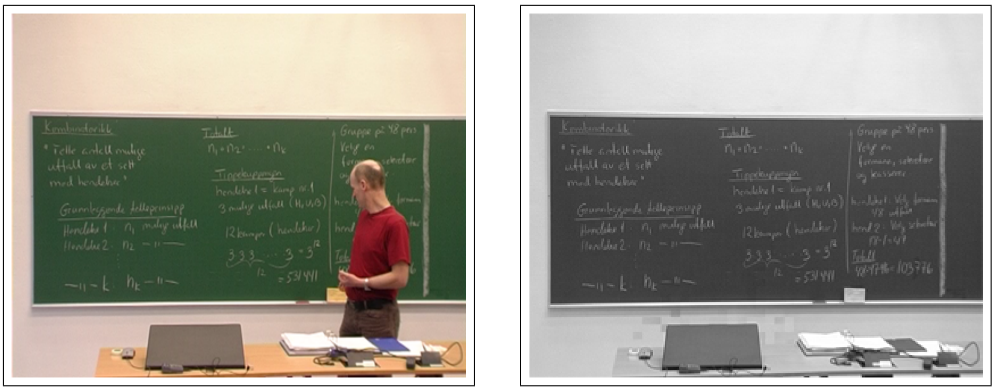
If a region contains motion, we replace that region in the current frame with the same region from the previous frame. The regions which are not covered with moving objects are updated only. The output from this segmentation is a new frame that should not contain any moving objects, such as the teacher. This new segmented frame is used as input for the segmentation process as the ’previous frame’. In order to keep the teacher segmented out in the following frames the newly segmented frame has to be used to segment out the teacher in the next frame.

Figure 31(a) and 32(a) show frames from original lecture videos. Figure 31(b) and 32(b) are the corresponding frame from the SAD segmented video. Figure 33 shows the original frame and the segmented frame using optical flow.


(a) Original frame

(b) Segmented frame

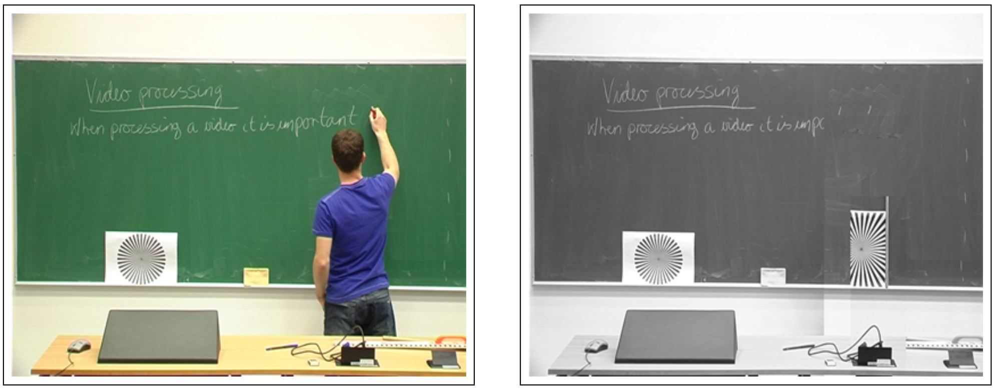
Figure 31: Segmented frame using SAD



(a) Original frame

(b) Segmented frame

Figure 32: Segmented frame using SAD



(a) Original frame

(b) Segmented frame

Figure 33: Segmented frame using optical flow and blob analysis

3.3 Meta-data extraction

Now that moving objects in front of the blackboard is removed, the next task is to detect content on the blackboard and register the teacher's actions. When content is detected it should be possible to track when it is being added or removed. This is useful information to know in order to extract key frames from the video to be used in an index.

3.3.1 Extracting content using frame difference

When any moving object is removed from the lecture video, it is only content on the blackboard left that is creating difference between frames. Over a period of time content is normally added or removed from the blackboard. Frame difference between consecutive frames can be performed to be able to track these contents. Figure 34 shows a high level subsystem of the content extraction method.

Since content is beeing added or removed slowly in time the input video can be

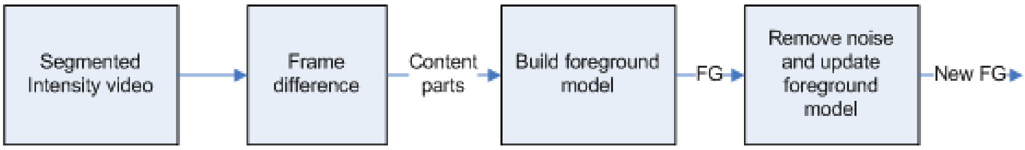

Segmented Intensity video
Frame difference
Content parts
Build foreground model
FG
Remove noise and update foreground model
New FG

Figure 34: High level blockdiagram of the content extraction subsystem

downsampled to create bigger differences. The input video is already down sampled by 12 in the segmentation process in section 3.2.2. Still there is not much content being added in a 3/4 of a second. By down sampling the video by 3 will output a total down sampling of 36, which means that only every 36th frame is considered. The difference between consecutive frames will then be bigger and this will also speed up the processing.

The frame difference is performed by taking the absolute difference between consecutive frames. The images in figure 35 show the difference between consecutive frames from one test lecture. A fixed threshold value is used to differenciate the content from the background.

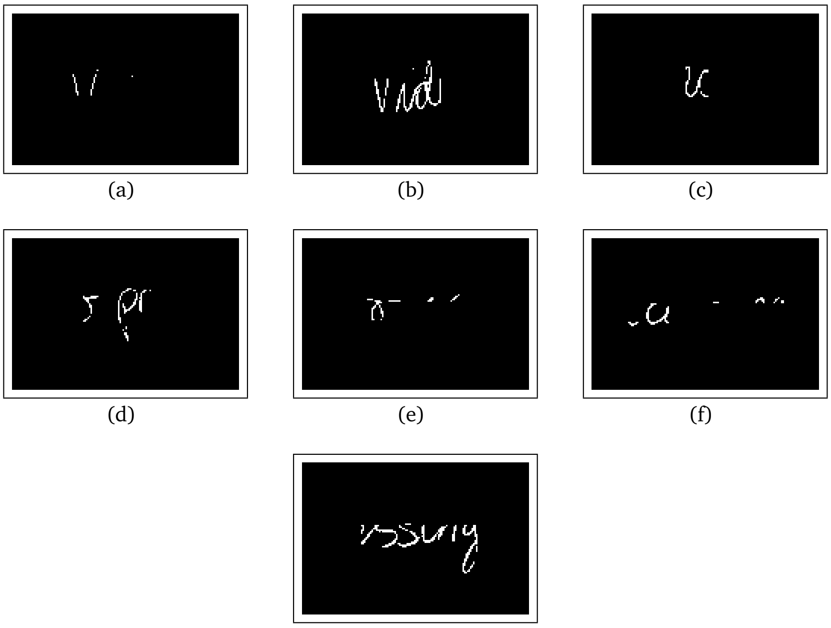

(a) (b) (c) (d) (e) (f) (g)

Figure 35: 7 text parts in consecutive frames using frame difference

3.3.2 Build and update the content foreground

To be able to keep track of for instance erasing of blackboard content, a foreground model must be built. This foreground model should consist only of the content that has been added over time. It then might be possible to count the amount of content and keep track on the amount of adding and erasing.

As discussed in section 3.3.1, frame difference can be used to track when the teacher is adding content. Also when the teacher erases content it appears as a difference between two consecutive frames, but we don't know if the word was added or removed. We have to know if the content appeared from the frame difference process already existed at the same place.

As seen in the images in figure 35 the difference between two consecutive frames are only parts of words or drawings. A foreground model can be built by adding all the parts together. Figure 36 shows the stitched version of the images in figure 35.

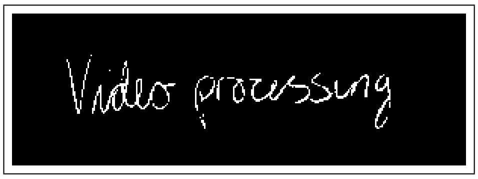


Figure 36: Textparts stitched together

Figure 37 also illustrates a foreground model after some time in an ordinary lecture. The noise at the lower part of the frame is from the segmentation and movements of other objects.

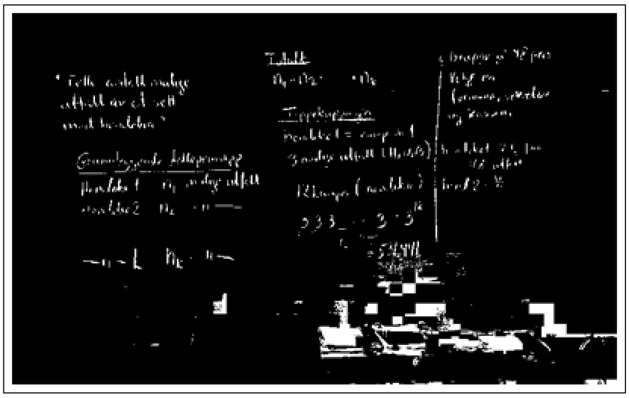


Figure 37: Foreground model in a lecture

To be able to update the foreground model correctly a method on how to find removed content has to be implemented. One way of doing this is to compare the current content from the frame difference process and see if it's already existing in the foreground model before adding it to the model. The part which is being removed can be found using the AND operator between the foreground model and the part of content that come from the frame difference process. The AND operator returns 1 when both a pixel in the foreground model and the part of content are 1, or else it returns 0 (table 1). Since chalk has the value 1 in the binary frame, 1 in these two frames at the same pixel indicates that the teacher is removing. We assume that there is little probability that the teacher writes on top of a previous written word.

FG model	Content part	Output
0	0	0
0	1	0
1	0	0
1	1	1

Table 1: AND operator

Since the foreground model is beeing updated with the content part which is also used as the second input to the AND operator, the result of the operator will always say that the content is being removed, even when the teacher is only adding content. This is because the content is added to the foreground model and checked if it exists in the model. Therefore, it is necessary to delay the updating of the foreground model by one sample. This will cause that the input ports on the AND operator are 0 and 1 which gives 0 at the output, and the content is not detected as removed.

Figure 38 shows a blackboard that is being erased, and is taken from the intensity video after teacher segmentation. From the frame in this figure we start to record erasing using the AND operator. The three consecutive frames in figure 39 show some content that is being erased. Since the video is down sampled by 3 in this process, the AND operator will output one frame that holds erased content from within three frames. Figure 40 shows the output from the operator, which is the content that is being removed by the teacher.

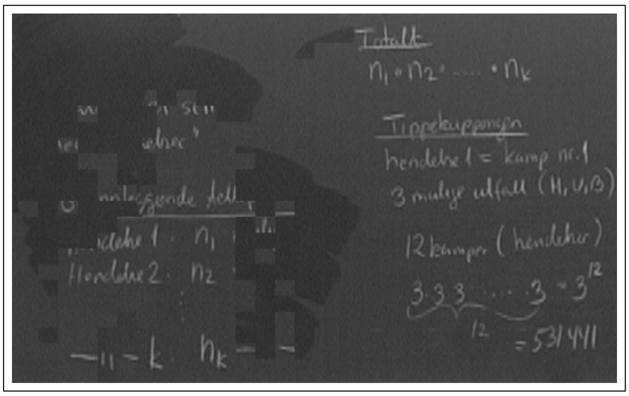


Figure 38: Segmented frame showing erasing of content

A way to automatically build and update the foreground model when the teacher adds and removes content is to use the XOR operator. The operator returns 1 only for odd values of 1 and 0 (table 2). If the first input pixel is 1 (text being added) and the second input pixel is 0 (foreground model) then the XOR operator will return 1, the content. The output of this operator is always used as the second input port for the next step. Doing this will build the foreground. If the first input pixel is 1 (text being removed) and the

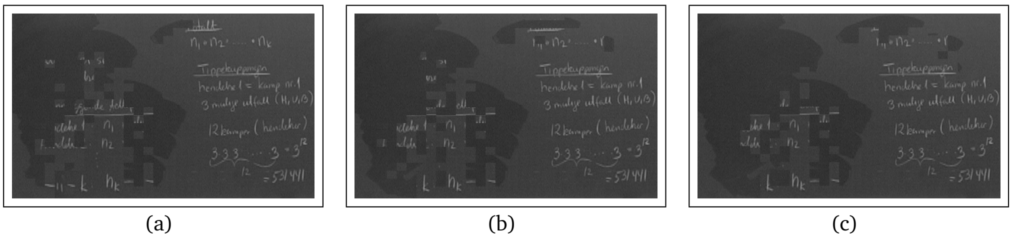


Figure 39: Content that is being erased by teacher

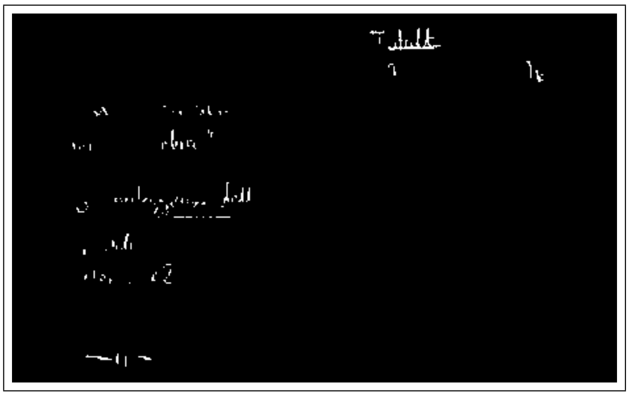


Figure 40: Using AND to detect erased text in figure 39

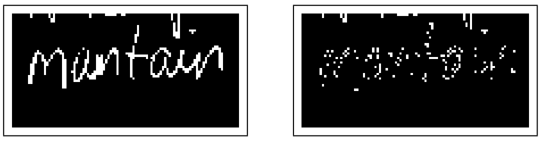
second pixel is 1 (already exist in the model) then the output is 0, and the part of content is erased.

Figure 41 shows the word *mantain*, which is spelled wrong, before and after the word is removed by the teacher. The XOR operator is used to remove the word from the foreground.

As can be seen in figure 41 there are still parts left from the word. If white pixels are exactly on top of each other in two frames then using the XOR operator would have removed these pixels without leaving parts of it behind. So obviously there is something happening. The following images in figure 42 shows the word *mantain* while it is being written, when it is being removed with a wet sponge and the difference between these two images. The written and removed content are not alike using a fixed threshold, and is a problem when updating the foreground model.

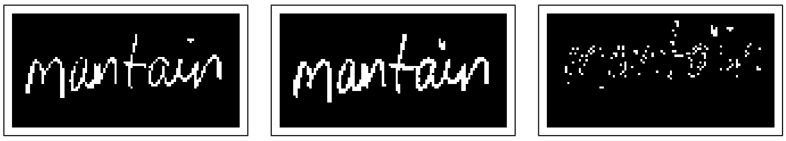
FG model	Content part	Output
0	0	0
0	1	1
1	0	1
1	1	0

Table 2: XOR operator



(a) Before being removed (b) After being removed

Figure 41: Update foreground model using XOR

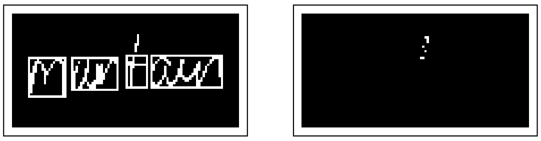


(a) Written (b) Removed (c) Difference

Figure 42: Difference between a written and removed word

Figure 42(a) has less contrast to the background then figure 42(b). More chalk of the word is represented and therefore the white pixels in these two images does not have all pixels in common. This is due to the fixed threshold value which does not take into consideration that the wet sponge gives more contrast between the blackboard and the white chalk then if the blackboard had chalk dust in that same area. The difference between the images is represented in figure 42(c).

One method that might help to solve the updating problem can be to find the coordinates for the removed word using blob analysis and clear everything inside that blob. By using the AND operator between the modeled foreground and the content part that comes from the difference betweeen two consecutive frames returns the removed part. When performing blob analysis we look for connected white pixels where the min and max size may be defined. When the correct blobs are found it is possible to extract the coordinates of a blob. The coordinates can be presented as a rectangle around the blob as drawn in figure 43(a). Figure 43(b) shows the cleared area within the coordinates of the detected content. This will only work if the content being removed is detected as a blob.



(a) Detected word by blob analysis (b) Blob area cleared

Figure 43: Improved clearing method

3.3.3 Noise reduction

This subsection will seek for methods on how to reduce noise when extracting content.

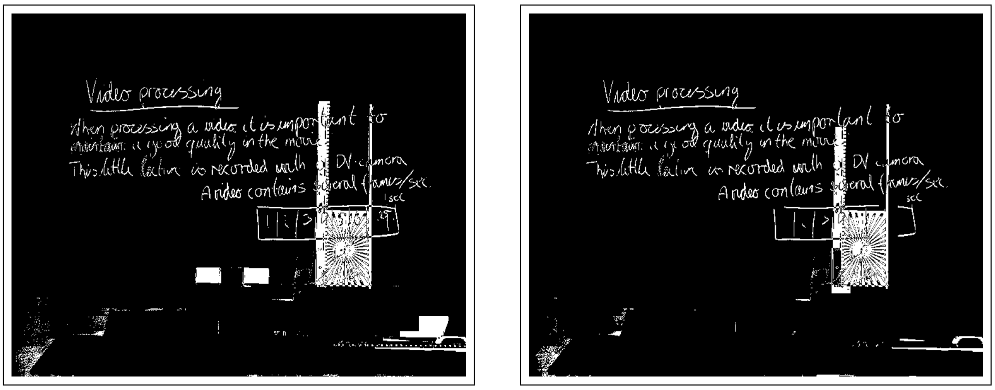
Foreground model

To be able to track the amount of text being added to the blackboard, some statistics of the white pixels in the foreground has to be calculated. This can for instance be done by summing up all the white pixels in the foreground model. Due to noise in the foreground model some white pixels are not content added by the teacher, and should be removed. For instance when the teacher removes or replaces a sponge it will be part of the foreground because of the frame difference process, found in subsection 3.3.1.

Threshold value based on blobs coverage

As earlier described, blob analysis is possible to use to find a blob's extremities. A sponge has a rectangular form so the blob area will be around the sponge's outline. For a text part the blob area will cover the text part's extremities as seen in figure 43(a). In general, this means that content being added by a teacher in a short period of time will assumedly not fill the whole rectangle. If a sponge is moved then the whole sponge will be detected by calculating the frame difference between two frames, and will appear as a white rectangle. This sponge will then fill between 80% and 100% of the extracted rectangular coordinates around the blob. With this in mind, it is possible to remove some noise by disregarding the blobs which cover the rectangle by more then a certain percent.

Figure 44(a) shows a foreground model where every blob is added, and figure 44(b) shows a frame where only blobs that have 70% or less of white pixels inside the blob rectangle area are allowed. As seen in figure 44(b) there is less noise where the sponge is and has been. But also some parts of the text have been removed. This is due to small blobs of the text being extracted, for instance one stroke of a letter which fill its rectangle area by more than 70%. If more of the text had appeared at the same time a bigger blob would have been found and it might have been a less coverage of the blob rectangle. By down sampling the video further the content added between consecutive frames will be increased. Images in figure 44 are down sampled by 4. The image in Figure 45 are down sampled by 6, and blobs with coverages above 70% are removed. As can be seen from these figures both text and noise can increase and decrease, and there is not that much difference by having a higher down sampling rate.



(a) Allow all blobs

(b) Remove blobs above 70% coverage

Figure 44: Noise reduction based on blob coverage.

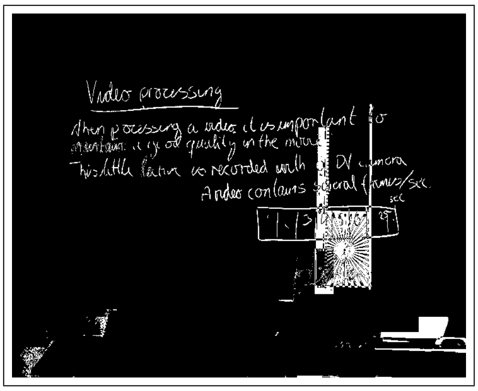
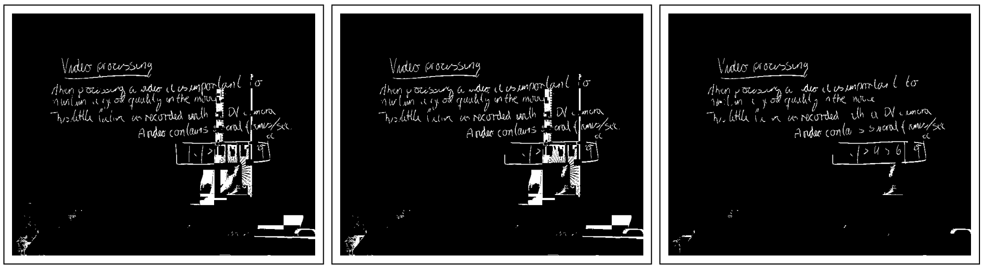


Figure 45: Blobs above 70% coverage are removed with a downsample of 6.

It is also possible to consider a lower percent coverage of white pixels inside a blob rectangle. Figure 46 shows a percent of coverage from 60 to 40%. It is visible that the level of noise is decreasing, specially from 50 to 40%. But the content on the blackboard is also decreasing. In figure 46(c) some whole words are almost missing.



(a) Discard blobs above 60% coverage

(b) Discard blobs above 50% coverage

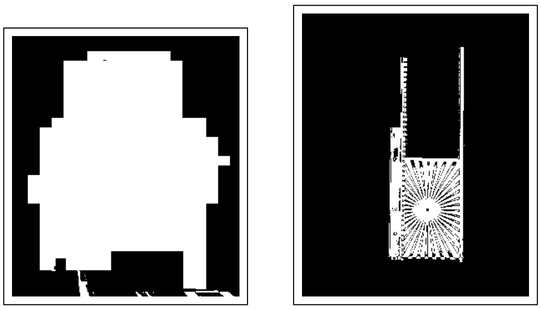
(c) Discard blobs above 40% coverage

Figure 46: Thresholding by different coverage values.

Threshold value based on amount of white pixels

A second method to get rid of noise is to look at the amount of chalk being added between two consecutive frames. The amount of content a teacher may be able to add between two frames (1-2 seconds) is not much. Some noise has a tendency to appear as a big object. Figure 47 shows two types of objects that appears in the foreground model as noise in two different videos.

By tracking the amount of white pixels, which is the difference between two consecutive frames after thresholding, it is possible to find when such big noise is appearing. Figure 48 shows a graph of the amount of white pixels through a 15 minute lecture video with no noise reduction. As seen in the graph there is a huge amount of white pixels in the start of the video. The reason for this is that the teacher was standing in the middle of the scene at the start. The segmentation part can remove the teacher when he/she has moved from the area he/she initially was standing in. Until that happens the teacher generates motion or changes in consecutive frames and is shown in the foreground model



(a) Moved object noise (b) Segmentation noise

Figure 47: Different kinds of noise.

as a big white object, the noise is shown in figure 47(a). To get rid of these huge noise objects which we don't want to appear in the foreground model, a threshold value can be calculated to only consider an amount of white pixels between two frames lower than the threshold.

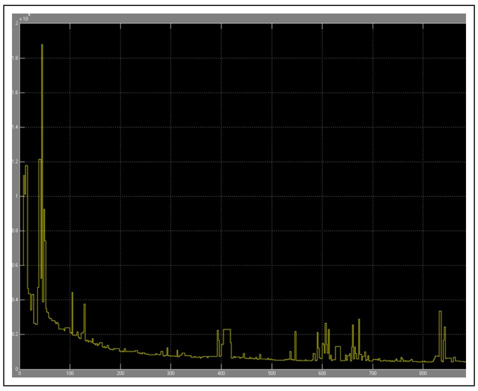
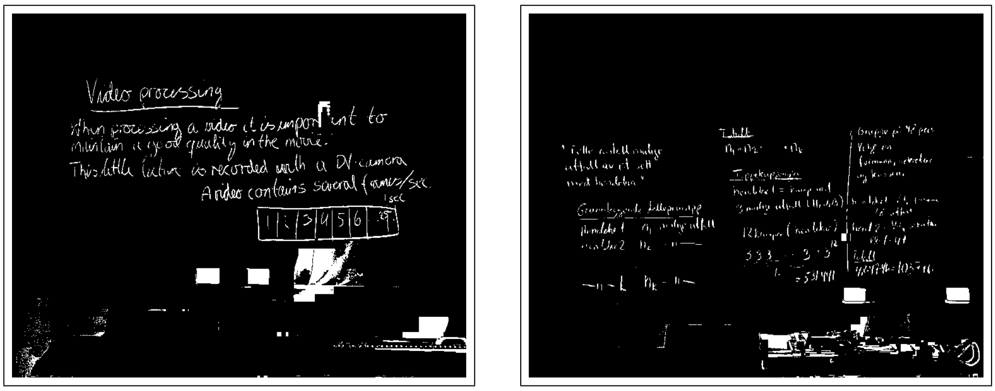


Figure 48: Amount of white pixels.

The images in figure 49 have been created by a threshold value of 3500 white pixels while letting all blobs to pass independent on the percent of coverage. The threshold value was set to 3500 in order to only remove bigger objects, and not content. All frames that had a value above the threshold value were cleared. Setting the threshold value to 2500, or even 2000, some content in both videos were removed, the noise however didn't really reduce. Figure 49(b) shows a foreground model where the teacher created a big blob in the beginning of the video, but is now removed. The big noise that was in the foreground model of both videos is almost gone. Figure 49(a) can be compared with the models in figure 46. It is visible that using an amount threshold value instead of having a low percent of coverage threshold value removes big noise better and simultaneously preserve the written content on the blackboard.



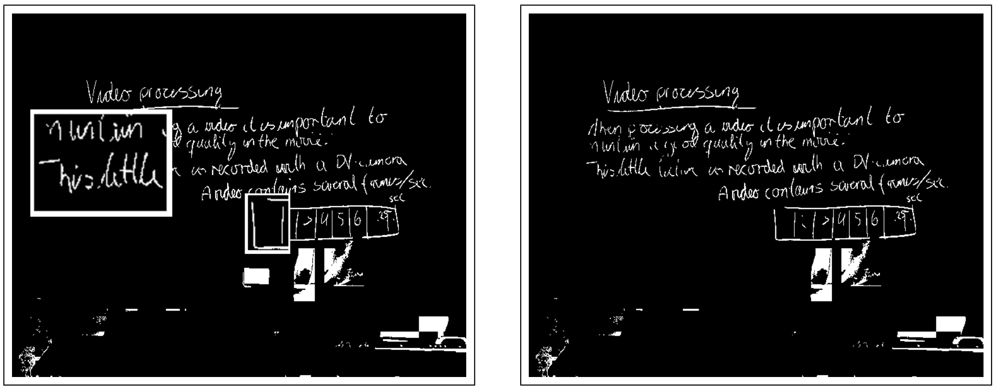
(a) From a test lecture video

(b) From a real lecture video

Figure 49: Thresholding by an amount of 3500 pixels.

Combining threshold values for coverage and amount of white pixels

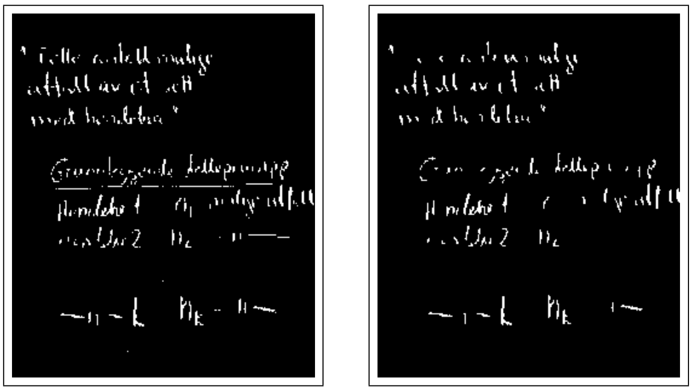
Still, the models in figure 49 have some noise from for instance the sponge. The model in figure 44(b) has no sponge in it, because of the 70% coverage thresholding. By combining a threshold value of blob coverage in percent and the amount of white pixels allowed will both remove some noise. Figure 50 shows foreground models in a test lecture video using a threshold value of 3500 white pixels and a threshold value on 80% and 75% of coverage. The sponge disappears somewhere between 75% and 80% of coverage in the test lecture video, and between 80% and 90% in the real lecture video. In both images in figure 50 some parts of the letters are missing. It is easier to see that parts of the content is missing in figure 51, which is a segment taken from a real lecture. The result is not good. By reducing the coverage threshold value the text is missing parts very quickly. Some other methods have to be implemented in order to keep the text as clear as possible.



(a) Discard blobs above 80% coverage. Removed content parts are highligthed

(b) Discard blobs above 75% coverage

Figure 50: Combining white pixel amount and coverage thresholding.



(a) Keep all blobs (b) Discard blobs above 75% coverage

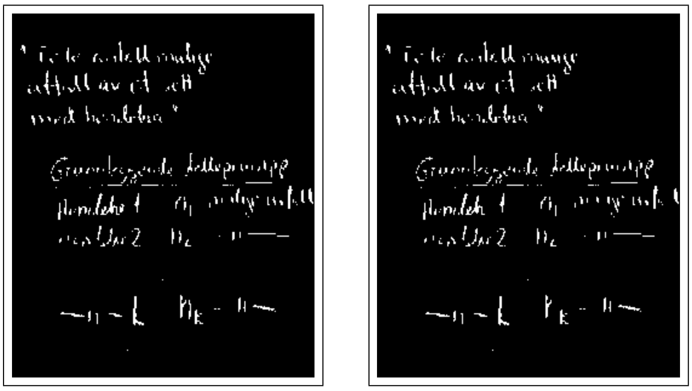
Figure 51: Reducing noise based on coverage removes content.

Content improvements

To be able to keep more of the text it could be possible to maintain every blob rectangle which is below a certain area threshold. Text is often small, and the parts of the text which is often removed due to a high percent coverage inside a blob rectangle are single rectangular strokes. The image in figure 50(a) show that the vertical stroke of the T in the word *This* is removed, and the number 1 in the first box of the drawing is missing (highlighted). To avoid removing these parts certain blobs which has a rectangle area of a certain value can be kept. Since the text parts missing are rather small a threshold value should be low so that not bigger parts of noise is also kept. The image in figure 52(a) is a segment just like in figure 51 but the difference is that blobs with smaller area than 25 pixels are not removed. The value 25 is chosen based on statistics of the blobs area, and there are many blobs with an area lower than that value, as seen in figure 53. All horizontal coloured lines in the graph represent an area value for one blob. It is possible to see that more strokes of certain characters are visible in figure 52(a) compared to figure 51(b). The improvement is quite good. This allows to decrease the threshold value on blob coverage. Figure 52(b) shows a segment og blacboard content with a coverage threshold set to 60%. It is possible to see that the content is still quite good, but there aren't really big differences. Some minor strokes are missing, and the noise in the frame didn't reduce significantly.

Short discussion of the results

A lower threshold percent of coverage could have been used, since not that much of the content is being removed. But this will however affect the updating of the model when the teacher is erasing content. The following threshold values of the above mentioned methods will be used in later processes; frames with 3500 white pixels or above between consecutive frames are removed, blobs that have above 75% of coverage are removed and blobs with an area of 25 pixels or lower are not removed. These steps provide a reasonable noise reduction method, and still keep the content on the blackboard clear.



(a) Discard above 75% coverage

(b) Discard above 60% coverage

Figure 52: Discard blobs based on coverage and keep those with area 0-25px.

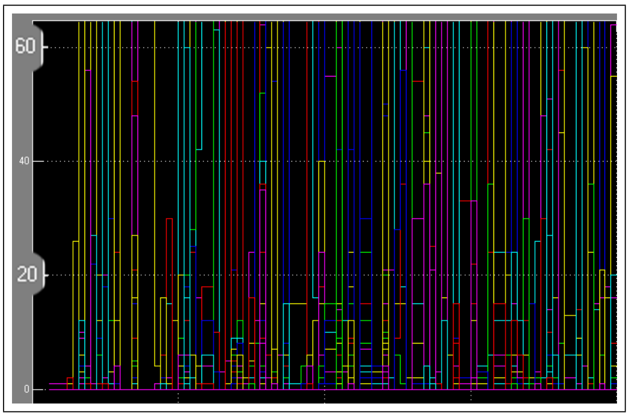


Figure 53: Graph showing area sizes of blobs.

Removed content

Noise can also be a problem when tracking removed content, but using some of the methods as described in paragraph *foreground model* could help to remove noise. In contrast to the foreground model, erased content tend to have some small noise in most of the frames. This happens because of changes that comes and goes in the same place, like for instance intensity changes. One way to remove this is to track noise in a binary frame with blob analysis and remove blobs which are smaller in size than a certain threshold value. This value should be small enough to remove only small parts so that part of the removed content is not treated as noise. The image in figure 54 shows some noise that can appear in a frame.

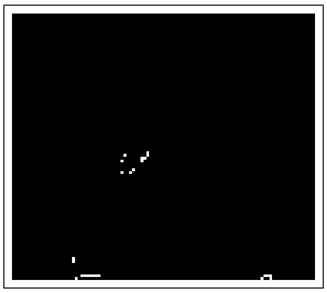


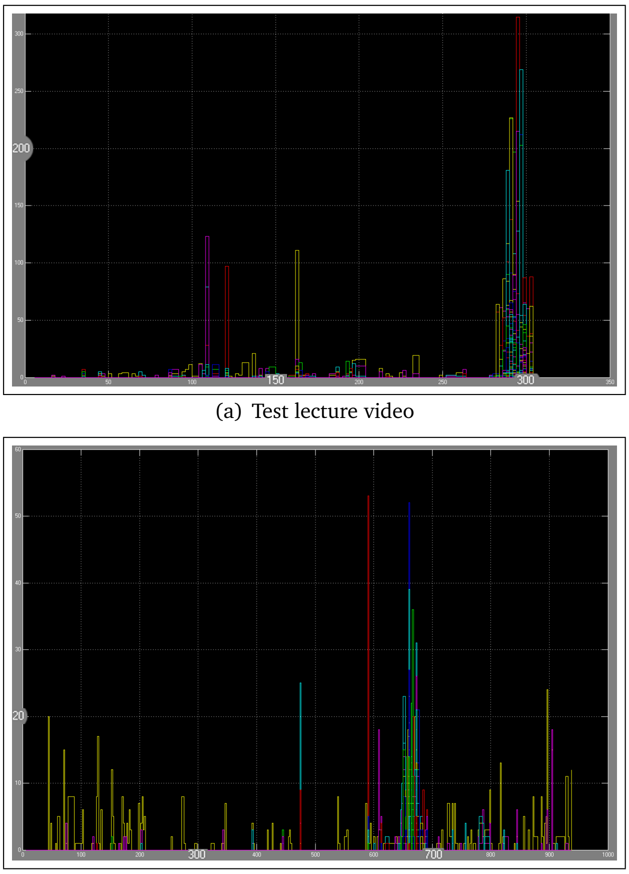
Figure 54: Small noise detected as removed content.

The size of a blob can be calculated as the sum of all white pixels, and by removing those with a low amount, the small noise can be handled. The graph in figure 55(a) shows the amount of white pixels inside each found blob for a test lecture video. Just before 300 seconds in the graph, it is possible to see that there is a lot of content being removed. In the video this is the only place where the teacher is erasing the whole blackboard. Still, there is also detected a lot of noise which has a low chalk amount. The graph in figure 55(b) shows the same content but in a real lecture video. As seen there is also a big number of blobs found that are only noise. From about 650 seconds the teacher is removing a lot of content on the blackboard, and this can be seen by the graph. Still, some of the noise should be removed.

Small noise also tends to have a high blob percent coverage. The graphs in figure 56 shows the detected removed content in two lecture videos and the blobs percent of coverage. Many of the detected blobs have a very high blob coverage. By looking at the blobs coverage where there is erasing in the videos, it is possible to see that the coverage is low. This can be seen just before 300 seconds in the test lecture video and around 650 seconds in the real lecture video.

Noise reduction of removed content

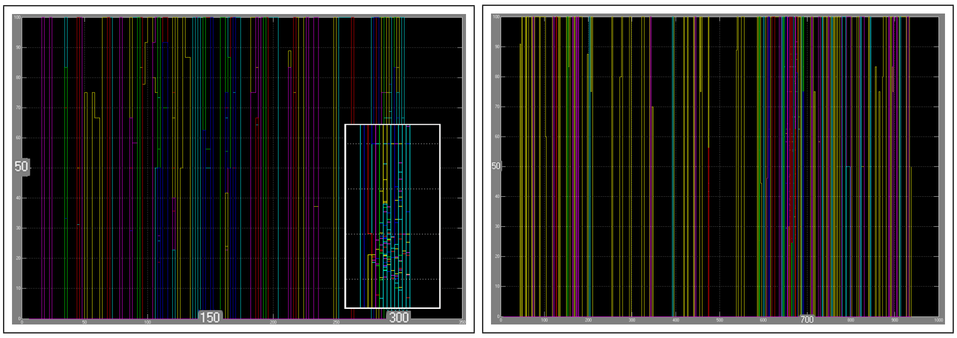
When the teacher removes something from the blackboard it is important to know when a large amount of content is being removed, not just only a word or a few letters. In order to have high level indexes of the blackboard's content, bigger amount of erasing is



(a) Test lecture video

(b) Real lecture video

Figure 55: Graphs show the blobs amount of white pixels.



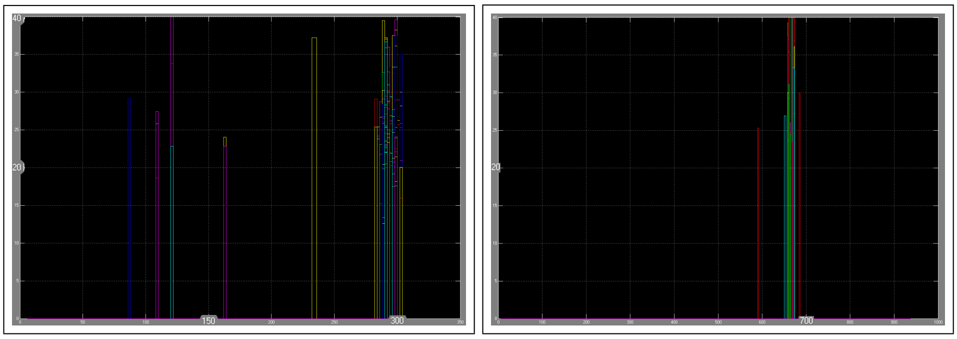
(a) Test lecture video

(b) Real lecture video

Figure 56: Graph shows the blobs coverage in percent.

interesting to detect. In figure 55 most of the blobs that are of a small size are beneath 10 pixels. But in order to keep some of the content that we want to be detected as removed content, a threshold value of 5 pixels can be used. Also alot of noise have a high blob coverage as seen in the graphs in figure 56. Blobs above 50% of coverage seems to be mostly noise, and can be used as a threshold to remove those blobs which have a bigger blob coverage.

Figure 57 shows the same information as in figure 56, but the difference is that figure 57 shows the statistics after noise reduction. It is much easier to detect where in the timeline there is erasing of the blackboard, and this will help in the indexing process.



(a) Test lecture video

(b) Real lecture video

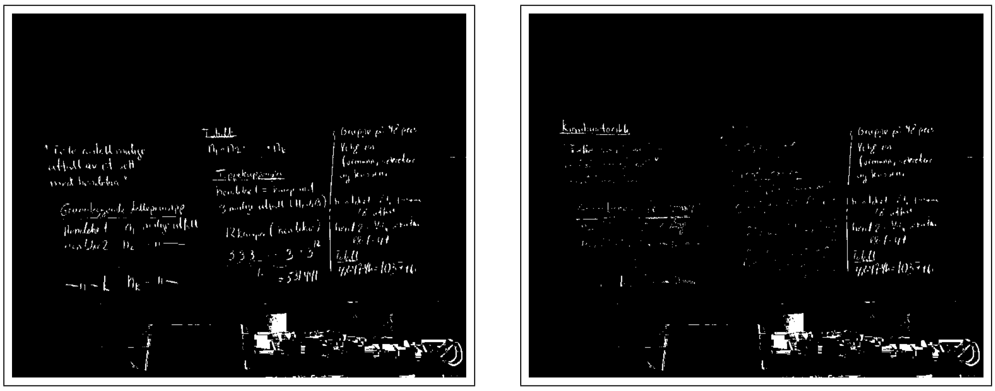
Figure 57: Graphs show the blobs amount of white pixels after noise reduction.

Updating foreground model after noise reduction

The noise reduction method performed on the foreground model and on removed content might influence the updating of the foreground model when the teacher erases content. Figure 58(a) shows a frame with content before it is being erased, and figure 58(b) the updated model after content is being erased. Only the two first columns are removed and some parts of the content are still on the blackboard, as seen in figure 58(b). The reason for this is mainly a threshold value problem, but the noise reduction also affect the updating. The more of the content being removed by the noise reduction process, the more it will affect the updating process.

3.3.4 Adaptive thresholding

When extracting text from the blackboard, chalk dust might be a problem. A clean blackboard has no chalk dust, but after the teacher adds and removes chalk for a period of time, the blackboard pixels get higher intensity values. Also if the teacher remove some words with a wet sponge then the blackboard will have a darker colour in that area. Using a fixed threshold might help to segment the text before chalk dust is added, but not so good when there is a lot of chalk dust. An adaptive threshold would probably be preferred, and this has also been discussed earlier [29].



(a) Before erasing

(b) After erasing

Figure 58: Updating of the foreground model after noise reduction.

The threshold value might be calculated as the median pluss 2 times the standard deviation of the blackboard's pixel values. Chalk that is added on a blackboard will have higher intensity, and will be found by using this threshold. For whiteboards the content have lower intensity than the background. A solution will then be to invert the signal when the background intensity value is higher than for instance 127 (middle intensity value).

Figure 59 shows an automatically calculated area of the blackboard where there is little chalk dust. There is one word in this image that is not that good to read compared to the other words. In figure 60 there is a new calculated threshold based on a wet area on the blackboard. This threshold is good for the word *maintain* but bad for the other words.

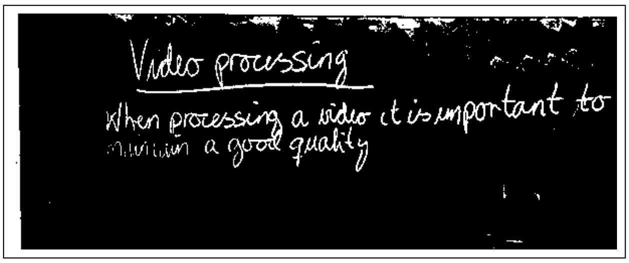


Figure 59: Threshold value calculated on an area with little chalk dust

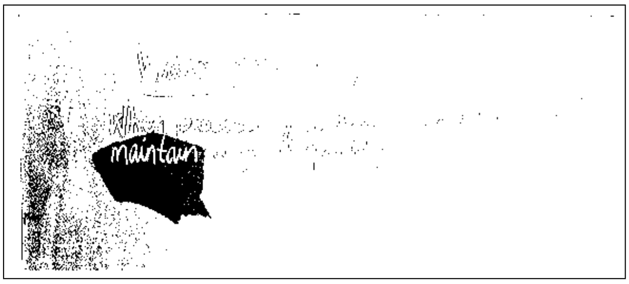


Figure 60: Threshold value calculated on a wet area

The automatic calculation of a threshold value can be implemented by using two frames, where the first frame is the current frame, and the second is the previous frame. The current frame is a binary frame where we detect content being added based on a global threshold value. We now have the coordinates for the content and can calculate the median and standard deviation for the pixels inside that area in the previous frame. The previous frame does not have the part of the content that was added in the current frame, so then the threshold is calculated on the background. This threshold can then be used to extract next content part.

To implement a generic threshold calculation might be good to overcome problems with chalk dust and wet blackboard areas, but then we have to make sure that it is chalk we have detected and not noise. If parts of the teacher appears and we calculate the median of that area, the threshold value might not show any text that appears at the same time.

3.4 Video indexing

In this section we will gather statistics from the meta-data extraction. This will be helpful to find valuable index points. Before going into details a high level block diagram is shown in figure 61.

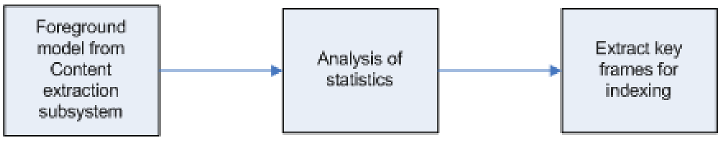

Foreground model from Content extraction subsystem
Analysis of statistics
Extract key frames for indexing

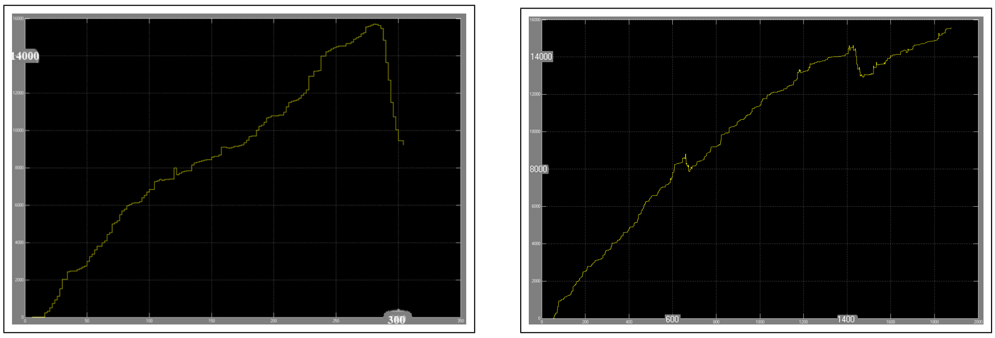
Figure 61: High level block diagram of the key frame extraction subsystem

3.4.1 Statistics of added chalk

The graphs in figure 62 show the amount of chalk in the foreground model of a test lecture video and part of a real lecture video. Using these graphs we can find out when the curve is increasing, which means that the teacher is adding content. Still, some noise will appear and be detected as adding content to the blackboard. When the graph is decreasing it means that the teacher is removing content, like in the end of figure 62(a) and two places in figure 62(b).

3.4.2 Statistics of removed chalk

Tracking the amount of erased chalk can be done in the same way as tracking added chalk. As described in section 3.3.2 the AND operator can be used to track erased chalk. By studying statistics gathered from the extraction of chalk it might be possible to tell when the teacher erases a lot of content, which is important to track in order to create indexes of the blackboard's content. The image in figure 63(a) shows the amount of white pixels in a frame of removed content between two consecutive frames, while figure 63(b)

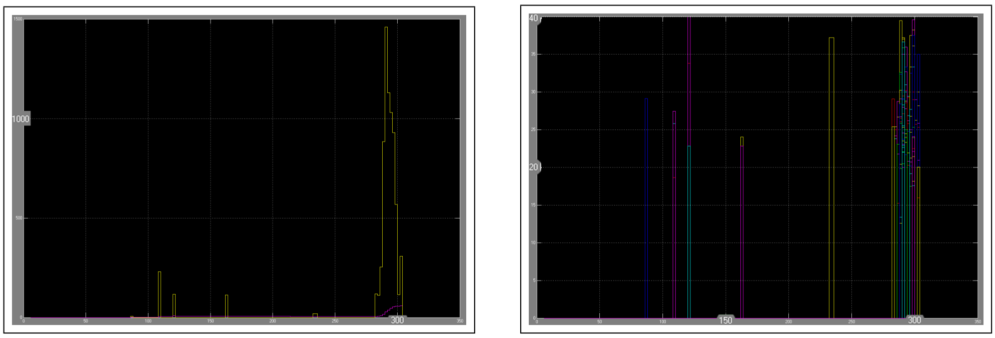


(a) Test lecture

(b) Real lecture

Figure 62: Amount of white pixels in the foreground model

shows the percent of coverage a blob has inside its rectangle. Both of them are gathered from a test lecture, and after applying noise reduction. It also shows how many blobs are found. Figure 64 shows the same as in figure 63, but in a real recorded lecture. In figure 63 the blackboard is cleaned out once, while in figure 64 the board is cleaned out 3 times; 2 of them 75% and one 25% of content is removed. It is easy to see where the teacher is removing the content because of the high consentration of detected removed content.

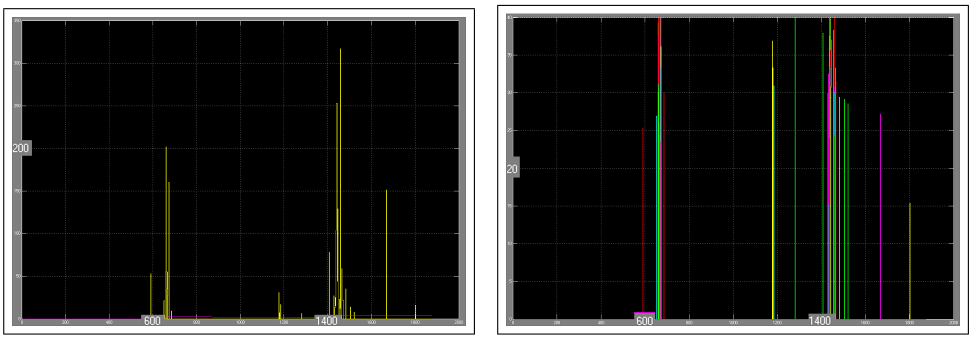


(a) Amount of white pixels

(b) Percent of blob coverage

Figure 63: Statistics of removed content in a test lecture

To be able to extract key frames for an index the erasing statistics has to be analysed. It can be seen in the extracted statistics when the teacher erases a lot of information. In the areas where there is a lot of activity of erasing, a key frame should be extracted, and only one. In figure 64(a) it is possible to see that a lot of content is removed just below 700 seconds. A key frame extracted when the first removed content in this compilation is detected can be seen in figure 65. Some of the content is already removed so in order to use an extracted key frame in an index it has to be right before the erasing. One way to do this is to always have a history buffer of the two previous frames, or even two and three frames back. When erasing is first detected, the history buffer shouldn't be updated. If there is enough erasing in a period of time, one of the frames in the buffer should be used as a key frame. Since there is a possibility that there are content erased already in



(a) Amount of white pixels

(b) Blob coverage in percent

Figure 64: Detected removed content with noise reduction

some of the frames in the buffer, a frame difference can be calculated to see if there is a big change between them. If there is, use the first frame. This may overcome the problem by extracting a key frame with erased content.

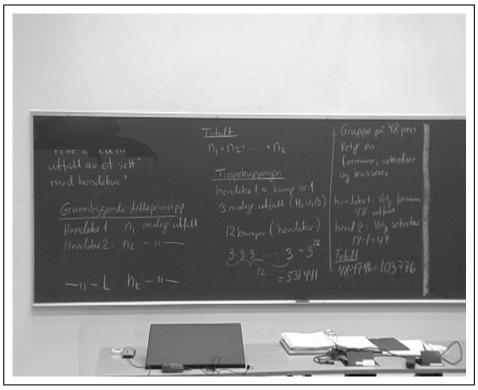


Figure 65: Extracted key frame

The next huge erasing of content is just above 1400 seconds. Figure 66 shows the extracted frame after detecting the first removal in that compilation. Also here a lot of the content is removed in the extracted key frame. A reason for the huge amount of content being erased before detecting the removal of content might be because of the noise reduction. By removing some types of noise might also remove some parts of the removed content, so that the detector doesn't react before more, or bigger, content is removed between consecutive frames.

One easy way to extract a key frame just before erasing can be done by delaying the current video frame for the extraction process. This means that if the delay is set to, for instance 3 frames, then the extracted key frame will be 3 frames back from the current frame position. Figure 67(a) and 67(b) shows the two extracted key frames, but the difference has a delay of 7 frames before getting a full blackboard without any erasing.

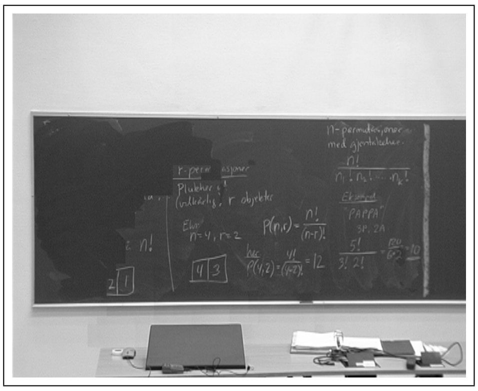
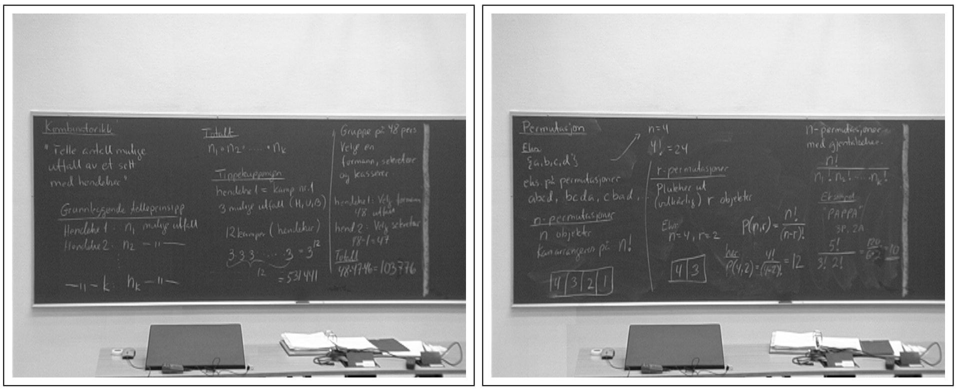


Figure 66: Extracted key frame



(a)

(b)

Figure 67: Extracted key frames with delay

3.4.3 Different types of indexes

In this thesis visual indexes are extracted, but also other kinds of indexes might be possible to track, for instance when the teacher is talking or discussing with the students. If the teacher is not adding or removing content for a period of time then the period of time this session lasted can be registered as a talking index. Long talking session may often be a discussion or important and eloborative explanations from the teacher. These kinds of indexes that indicates long talking sessions might be helpful for students in navigating through a lecture video.

Searching for information in a lecture video might not be easy, but to be able to separate different indexes from each other can be helpful when navigating. For instance to search for visual indexes or talking session longer or shorter then a time threshold spesified by user might be possible. If figures could be extracted from the blackboard and recognized as a figure of some kind, then it would be possible to search for figure indexes. Also headlines can be used in the same way as for figures. This however, needs a more work in analysing the different kinds of content that a teacher might add during a lecture.

An automatically indexed lecture video should be able to jump into the video where students requests. The request should be performed by choosing an index, and the video should start playing from where the index was extracted. For visual indexes, like key frames in this work, the video should start playing from where the content in the key frame was starting to be added on the blackboard. For figures or headlines it would probably be good to jump into the video just before the figure or headline was added.

5 Conclusion and Further Work

Overall the objectives of this thesis have been met. The teacher has been segmented out from the foreground, content has been extracted, and based on analysis of the content visual indexes has been extracted.

Q1: How can foreground/background segmentation of lecture videos work for different writing-boards?

In order to separate the foreground and the background without knowing in advance what colour objects have in the scene, motion detection has proven to be able to remove moving objects in a classroom environment. Two different motion detectors, SAD and optical flow, was tested on two recordered lecture videos. Results show that SAD performs better. By the use of video down sampling, morphological dilation and closing, a column-wise closing method and an implementation of a motion history vector it is shown that it is possible to overcome the limitations of the motion detectors

By using motion detection it is possible to detect and remove moving objects in front of a static scene, even if the background consist of a green, black or whiteboard.

Q2: How to automatically extract meta-data from lecture videos?

Meta-data, or actions performed by the teacher in a lecture, are tracked based on the content on the blackboard. In order to extract content from any coloured writing-board, frame difference between two consecutive frames extracts parts of content that are added on the writing-board. A foreground model is built by adding each of the extracted content parts. A logical XOR operator is implemented to build and update the model with the help of simple updating method to help removing content. The updating method is implemented due to problems with chalk dust and other variations on the blackboard that makes the content being added appearing slightly different as when the same content is removed. To be able to track changes in the content on the blackboard in a best possible way, the foreground model needs to go through a noise reduction process. Three implemented methods remove most of the noise.

Adding of content is tracked based on the amount of white pixels in the binary foreground model. Removal of content is tracked by a logical AND operator between each text part and the foreground model. These method gives good indication of when the teacher is adding and removing content on the blackboard.

Q3: How to use such meta-data for indexing and searching of lecture videos?

Visual indexes of the blackboard is extracted as key frames in this work. The key frames contains a blackboard full of information added by the teacher, and are extracted just before the content is being removed. Visual indexes may be used to navigate through a lecture video by looking at the key frames for content they for instance are looking for. Both students that attended the lecture and those who didn't may recognize topics and figures from the key frames, and when they choose one key frame the original video should start playing from where the content in the key frame was starting to be added.

Figures and headlines would be good indexes, and could be used as visual and textual indexes. Textual indexes can be used for searching in a lecture video, for instance after headlines og talking sessions.

Bibliography

[1] Grythe, E. Object segmentation and text detection in lecture video. Master's thesis, Institutt for Informatikk og Medieteknikk, Høgskolen i Gjøvik, 2005.

[2] Javed, O., Shafigue, K., & Shah, M. 2002. A hierarchical approach to robust background subtraction using color and gradient information. *Motion and Video Computing, 2002. Proceedings. Workshop on*, 22–27.

[3] Mittal, A., Gupta, S., Jain, S., & Jain, A. 2006. Content-based adaptive compression of educational videos using phase correlation techniques. *Multimedia Systems*, 11(3), 249–259.

[4] Friedland, G. *Adaptive Audio and Video Processing for Electronic Chalkboard Lectures*. PhD thesis, Fachbereich Mathematik u. Informatik, Freie Universität Berlin, 2006. http://www.diss.fu-berlin.de/2006/514/indexe.html (Last visited Mar. 2007).

[5] Wei, Y. & Badawy, W. 2003. A new moving object contour detection approach. *Computer Architectures for Machine Perception, 2003 IEEE International Workshop on*.

[6] Onishi, M., Izumi, M., & Fukunaga, K. 2000. Blackboard segmentation using video image of lecture and its applications. *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, 4, 615–618.

[7] Liu, T. & Kender, J. 2002. Rule-based semantic summarization of instructional videos. *Image Processing. 2002. Proceedings. 2002 International Conference on*, 1, 601–604.

[8] Liu, T. & Kender, J. 2004. Lecture videos for e-learning: Current research and challenges. In *Multimedia Software Engineering, 2004. Proceedings. IEEE Sixth International Symposium on*, 574–578.

[9] Ekinci, M. & Gedikli, E. *Background Estimation Based People Detection and Tracking for Video Surveillance*, volume 2869/2003, 421–429. Springer Berlin / Heidelberg, 2003.

[10] Niu, W., Jiao, L., Han, D., & Wang, Y. 2003. Real-time multi-person tracking in video surveillance. In *Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, 1144–1148.

[11] Elgammal, A., Duraiswami, R., Harwood, D., & Davis, L. 2002. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(4), 1151–1163.

[12] Ramanan, D. & Forsyth, D. 2003. Finding and tracking people from the bottom up. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2, 467–474.

[13] Gasserm, G., Bird, N., Masoud, O., & Papanikolopoulos, N. 2004. Human activities monitoring at bus stops. *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, 1, 90–95.

[14] Besada, J., Garcia, J., Portillo, J., Molina, J., Varona, A., & Gonzalez, G. 2005. Airport surface surveillance based on video images. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(3), 1075–1082.

[15] Guo, J., Kim, J., & Kuo, C.-C. J. 1998. Fast video object segmentation using affine motion and cradient-based colour clustering. *Multimedia Signal Processing, 1998 IEEE Second Workshop on*, 486–491.

[16] Criminisi, A., Cross, G., Blake, A., & Kolmogorov, V. 2006. Bilayer segmentation of live video. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 53–60.

[17] Repp, S. K. Forelesningsvideo: Hvordan forbedre gjenbruksverdien? Master's thesis, Institutt for Informatikk og Medieteknikk, Høgskolen i Gjøvik, 2006.

[18] IP, H.-S. & Chan, S.-L. 1997. Hypertext-assisted video indexing and content-based retrieval. *Proceedings of the eighth ACM conference on Hypertext*, 232–233.

[19] Lin, M., Nunamaker, J., Chau, M., & Chen, H. 2004. Segmentation of lecture videos based on text: a method combining multiple linguistic features. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*.

[20] Yamamoto, N., Jun, O., , & Ariki, Y. 2003. Topic segmentation and retrieval system for lecture videos based on spontaneous speech recognition. *Eurospeech 2003*, 4, 961–964.

[21] Tang, L. & Kender, J. 2005. Educational video understanding: mapping handwritten text to textbook chapters. *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, 919–923.

[22] Bhat, K., Saptharishi, M., & Khosla, P. 2000. Motion detection and segmentation using image mosaics. *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, 3, 1577–1580.

[23] Vanne, J., Aho, E., Hamalainen, T. D., & Kuusilinna, K. 2006. A high-performance sum of absolute difference implementation for motion estimation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 876–883.

[24] Lucas, B. & Kanade, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging understanding workshop*, 121–130.

[25] Lim, S. & Gamal, A. E. 2001. Optical flow estimation using high frame rate sequences. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, 925–928.

[26] Horn, B. & Schunck, B. 1981. Determining optical flow. *Artificial Intelligence*, 17, 185–203.

[27] Meier, T. & Ngan, K. 1999. Video segmentation for content-based coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 1190–1203.

[28] Heng, W. & Tan, Q. 2002. Content enhancement for e-learning lecture video using foreground/background separation. *Multimedia Signal Processing, 2002 IEEE Workshop on*, 436–439.

[29] Liu, T. & Kender, J. *Spatial-Temporal Semantic Grouping of Instructional Video Content*, volume 2728/2003 of *Lecture Notes in Computer Science*, 355–360. Springer Berlin / Heidelberg, 2003.

[30] Hasan, Y. & Karam, L. 2000. Morphological text extraction from images. *Image Processing, IEEE Transactions on*, 9, 1978–1983.

[31] Wang, F., Ngo, C.-W., & Pong, T.-C. 2007. Lecture video enhancement and editing by integrating posture, gesture and text.

[32] Smith, M. & Kanade, T. 1997. Video skimming and characterization through the combination of image and language understanding techniques. *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 775–781.

[33] Zhuang, Y., Rui, Y., Huang, T., & Mehrotra, S. 1998. Adaptive key frame extraction using unsupervised clustering. *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, 866–870.

[34] Ouglov, A. & Hjelsvold, R. 2005. Panoramic video in video mediated education.

[35] Dexter, R. 2004. Reading and downloads from the web. http://www.rondexter.com/intermediate/equipment/focus_pattern.htm. (Visited May 2007).

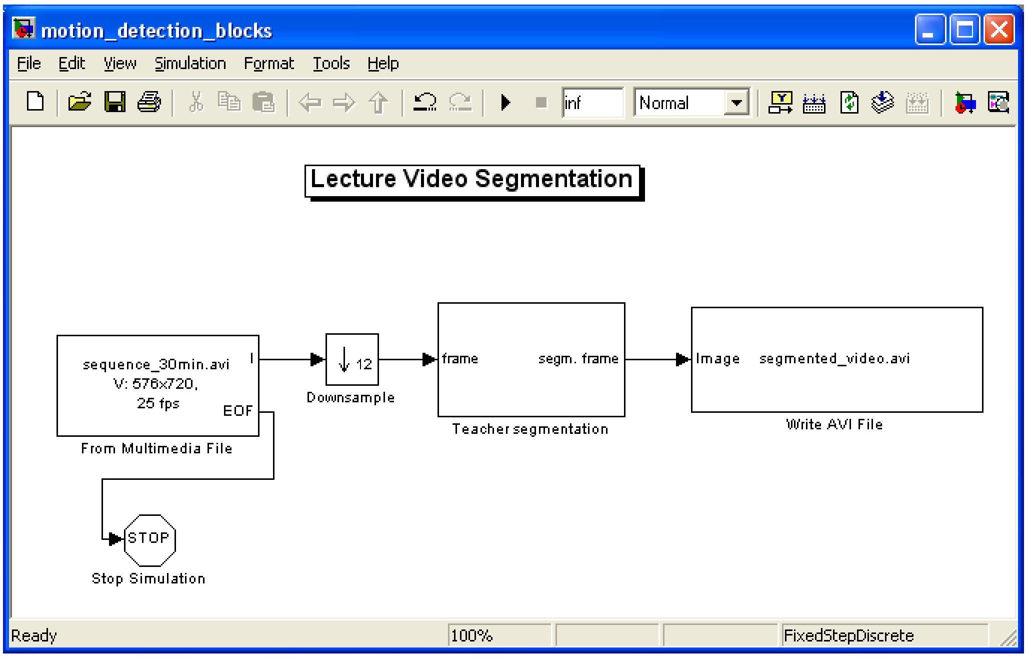
[36] The MathWorks, Inc. *MATLAB®*.

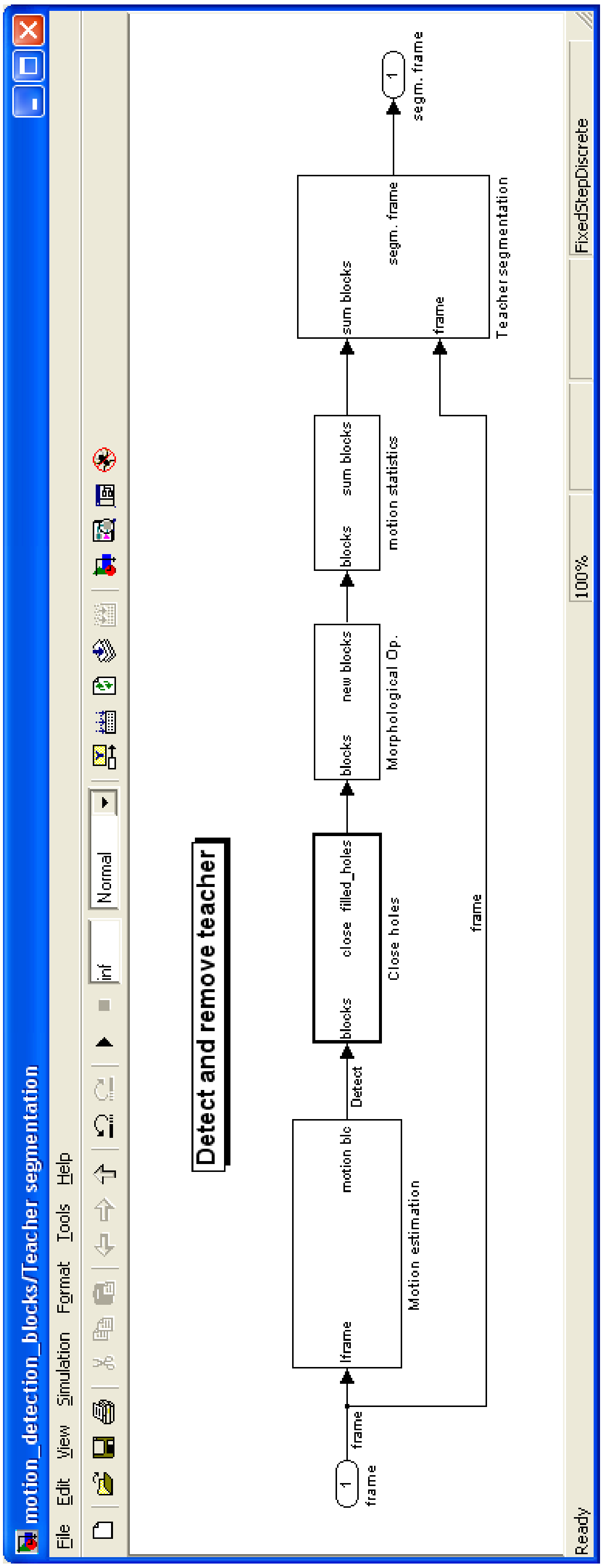
A Simulink models

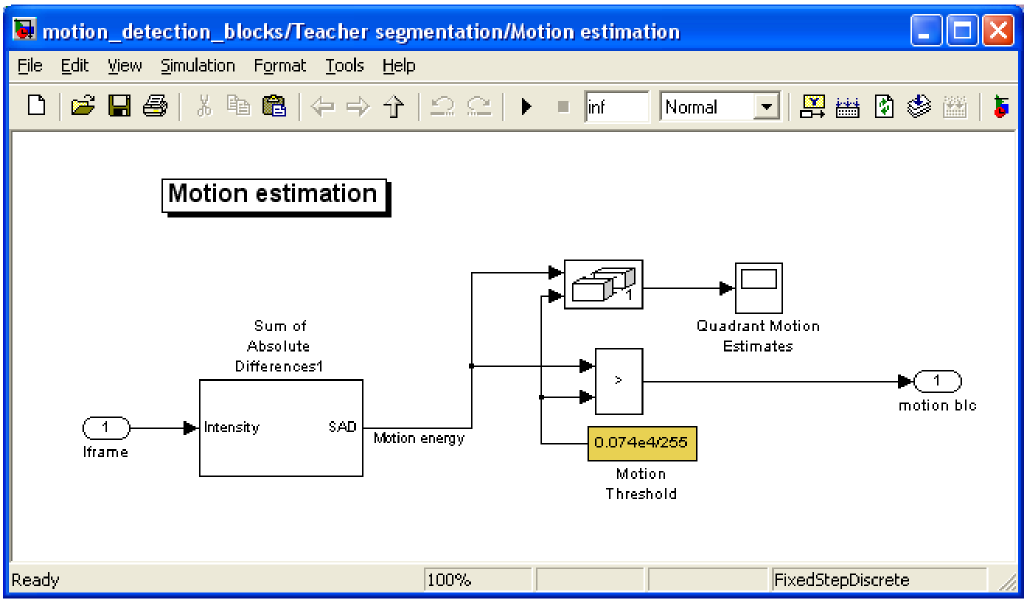
All models created in Simulink will be put shown here. The first model contains the segmentation process, where the teacher is removed. The second model contains both content extraction and key frame extraction. The reason for this is that key frame extraction is based on statistics from the content extraction process.

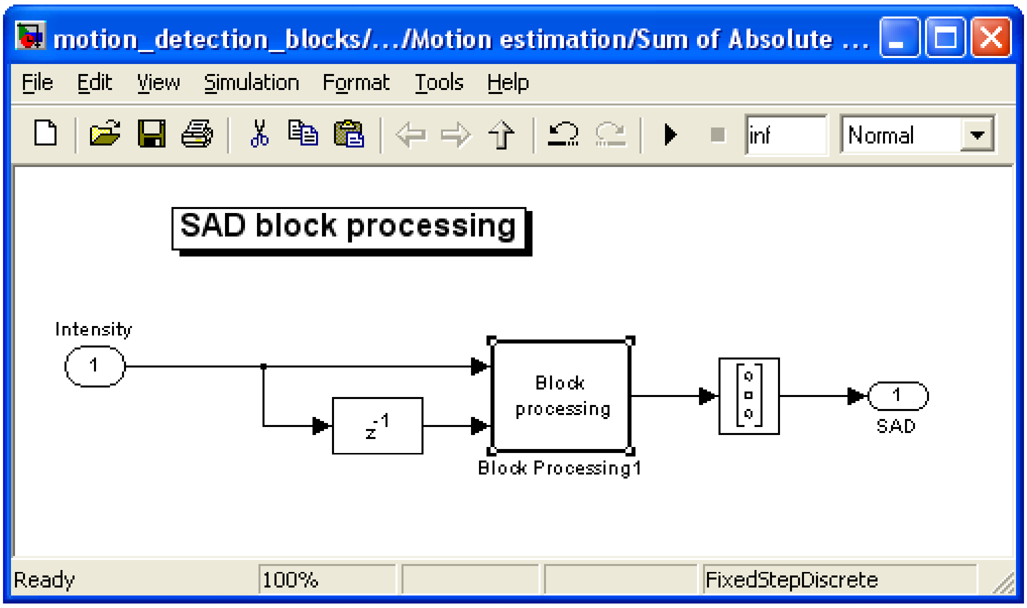
The models contain among other things subsystems and embedded MATLAB code. The embedded code is not shown here, but follows the DVD that comes with this thesis. All subsystems are shown in the order from left to right. If two subsystems cannot be distinguished who is the leftmost, then the subsystem at the top is shown first. The different Simulink blocks' properties is not shown. The Simulink models should be easy to understand in order to get a more detailed overview of the resulting system.

A.1 Segmentation model

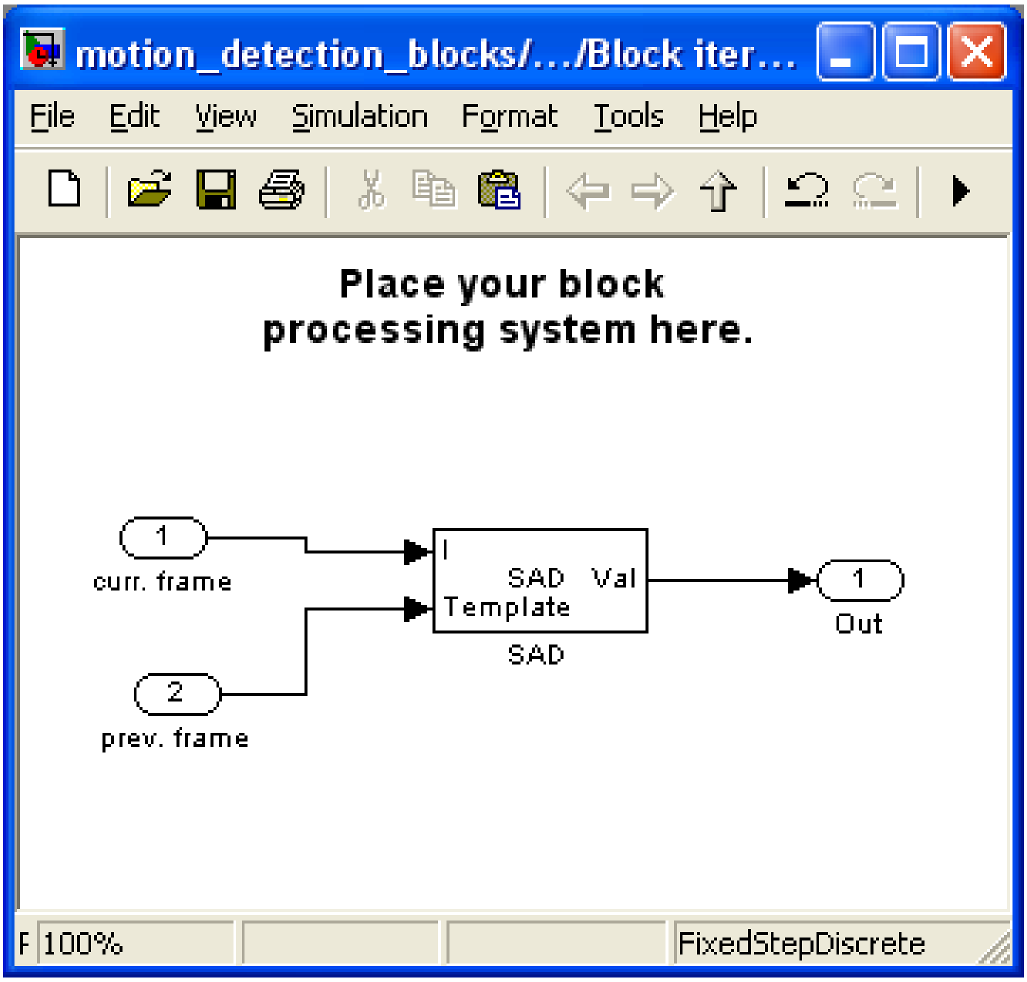



motion_detection_blocks/Teacher segmentation
File
Edit
View
Simulation
Format
Tools
Help
inf
Normal
Detect and remove teacher
1
frame
frame
Iframe
motion blc
Motion estimation
Detect
blocks
close
filled_holes
Close holes
blocks
new blocks
Morphological Op.
blocks
sum blocks
motion statistics
frame
sum blocks
frame
segm. frame
Teacher segmentation
1
segm. frame
Ready
100%
FixedStepDiscrete

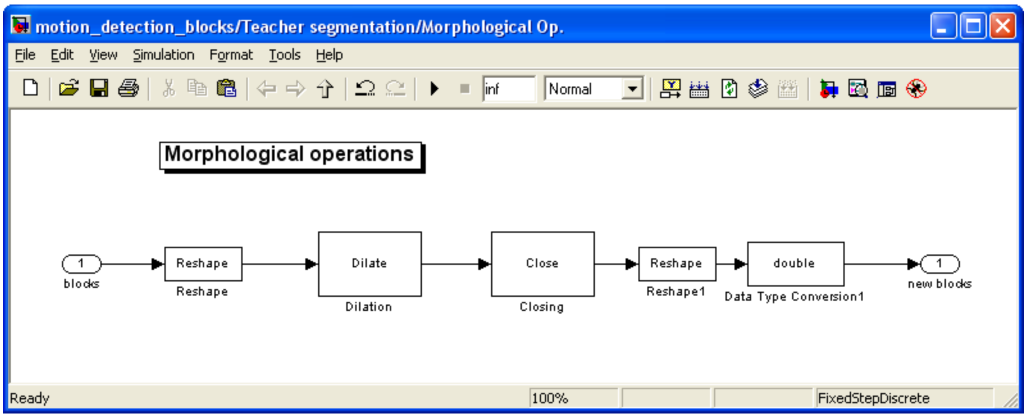

motion_detection_blocks/Teacher segmentation/Motion estimation
File Edit View Simulation Format Tools Help
inf
Normal
Motion estimation
Sum of
Absolute
Differences1
1
Iframe
Intensity
SAD
Motion energy
1
Quadrant Motion
Estimates
>
1
motion blc
0.074e4/255
Motion
Threshold
Ready
100%
FixedStepDiscrete

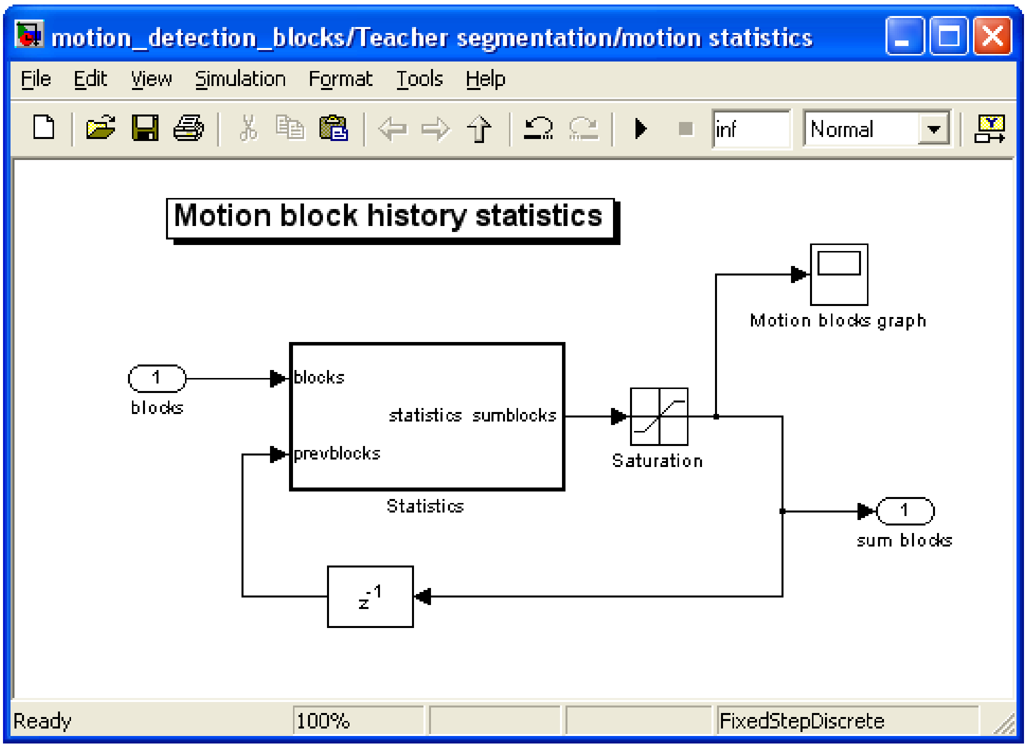


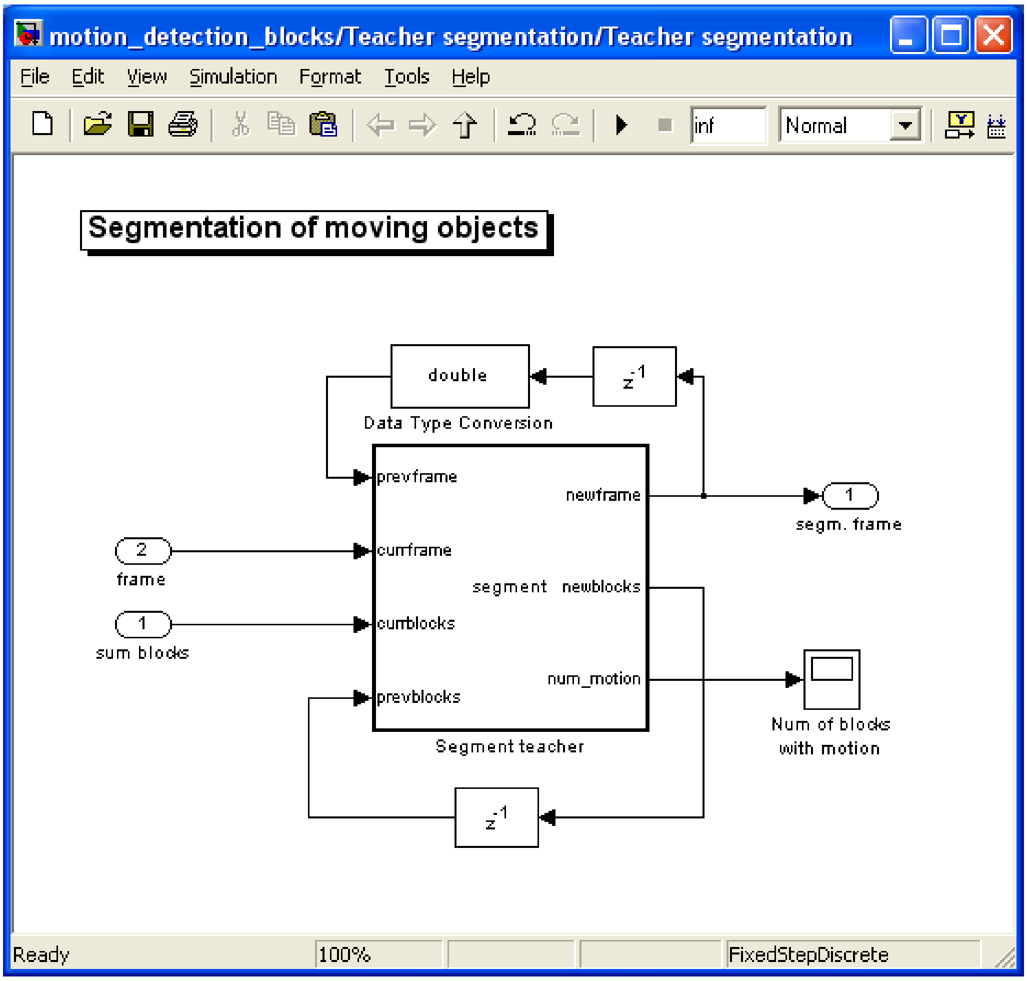
motion_detection_blocks/.../Motion estimation/Sum of Absolute ...
File Edit View Simulation Format Tools Help
inf
Normal
SAD block processing
Intensity
1
z^{-1}
Block processing
Block Processing1
1
SAD
Ready
100%
FixedStepDiscrete



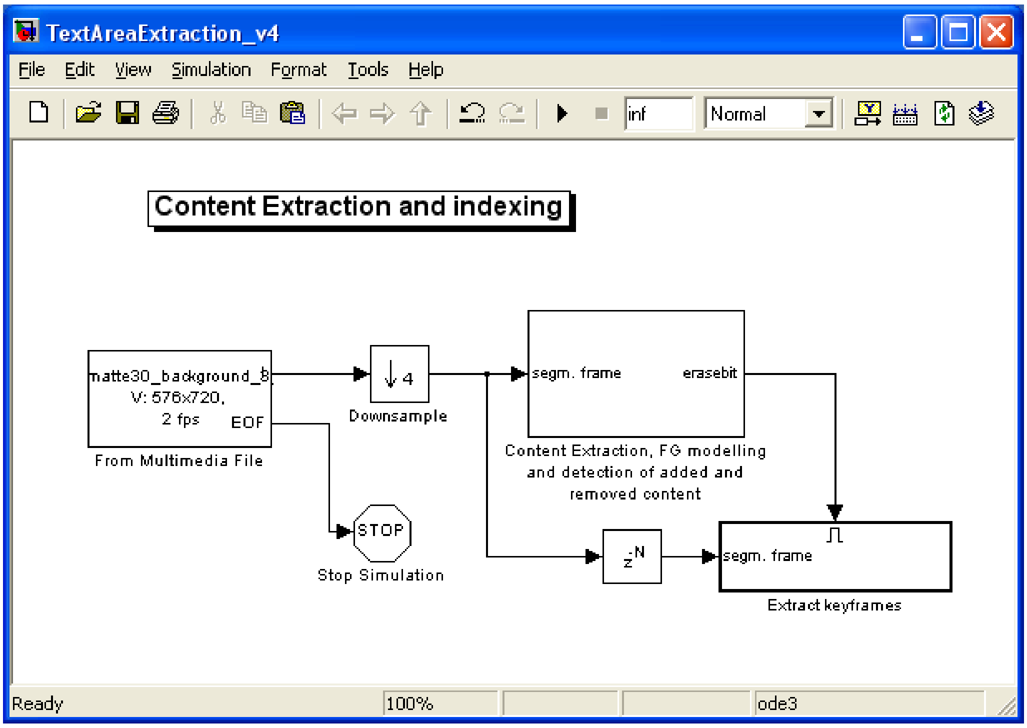
Place your block processing system here.

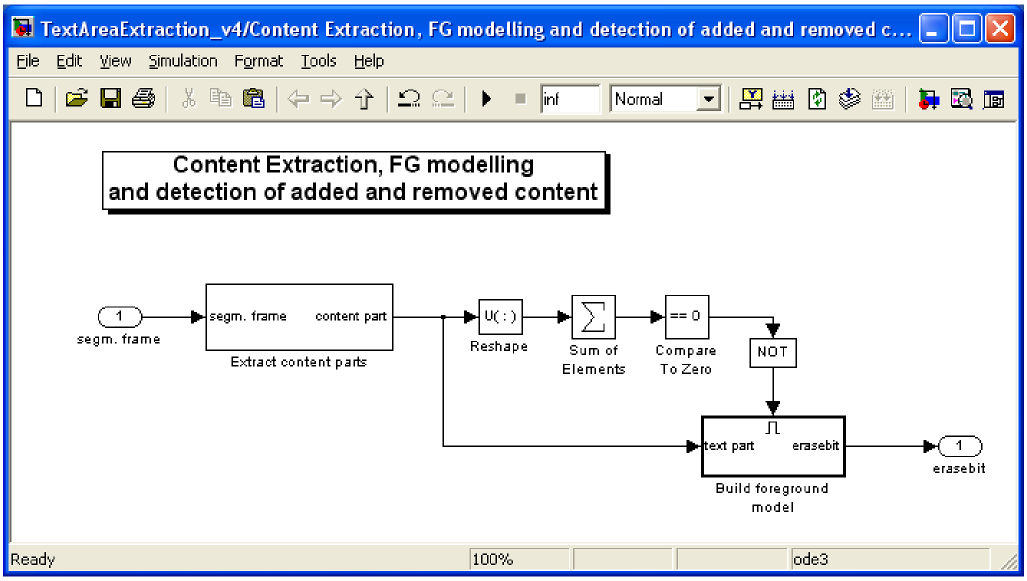

motion_detection_blocks/Teacher segmentation/Morphological Op.
File Edit View Simulation Format Tools Help
inf
Normal
Morphological operations
1
blocks
Reshape
Reshape
Dilate
Dilation
Close
Closing
Reshape
Reshape1
double
Data Type Conversion1
1
new blocks
Ready
100%
FixedStepDiscrete


motion_detection_blocks/Teacher segmentation/motion statistics
File
Edit
View
Simulation
Format
Tools
Help
inf
Normal
Motion block history statistics
Motion blocks graph
1
blocks
blocks
statistics sumblocks
prevblocks
Statistics
Saturation
1
sum blocks
z
-1
Ready
100%
FixedStepDiscrete

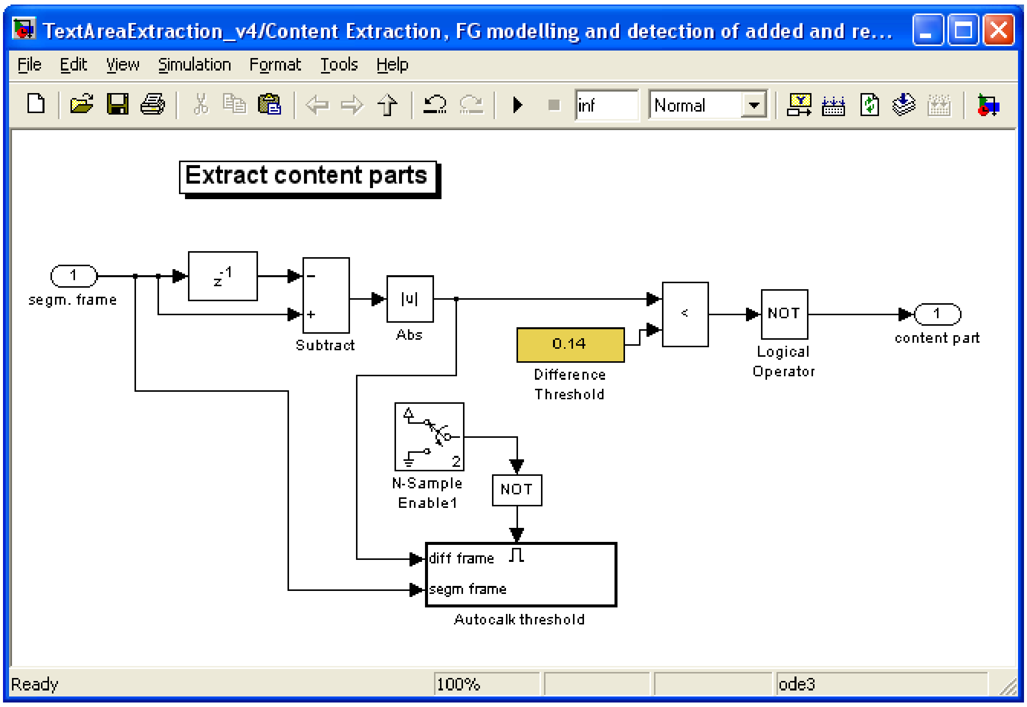

motion_detection_blocks/Teacher segmentation/Teacher segmentation
File Edit View Simulation Format Tools Help
inf
Normal
Segmentation of moving objects
double
Data Type Conversion
z-1
prevframe
newframe
1
segm. frame
2
frame
currframe
segment
newblocks
1
sum blocks
currblocks
num_motion
prevblocks
Num of blocks with motion
Segment teacher
z-1
Ready
100%
FixedStepDiscrete

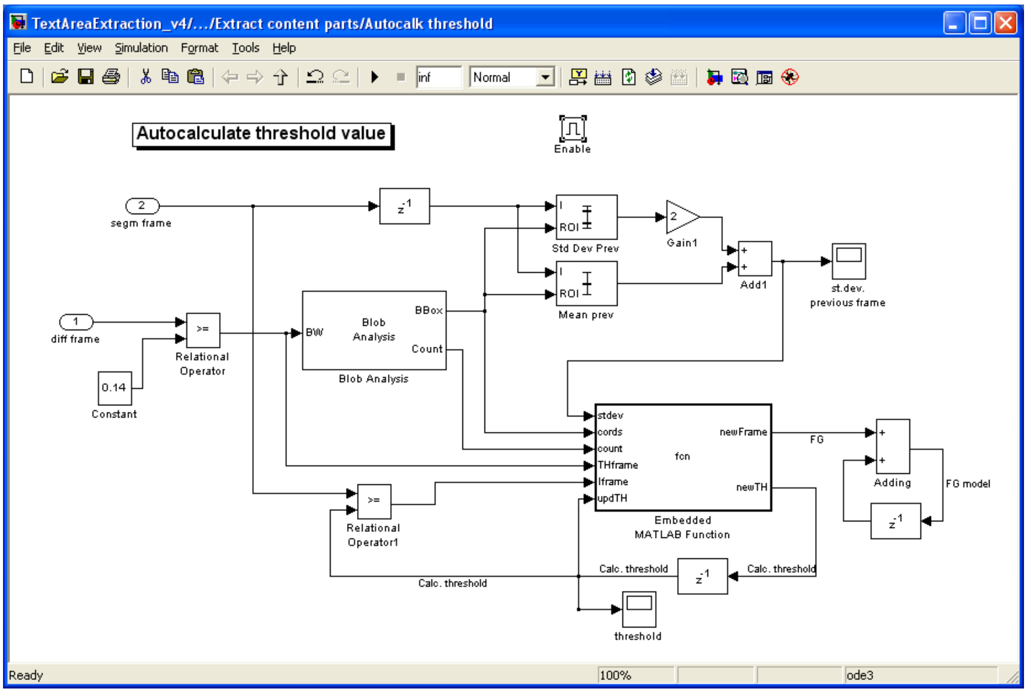
A.2 Content and key frame extraction

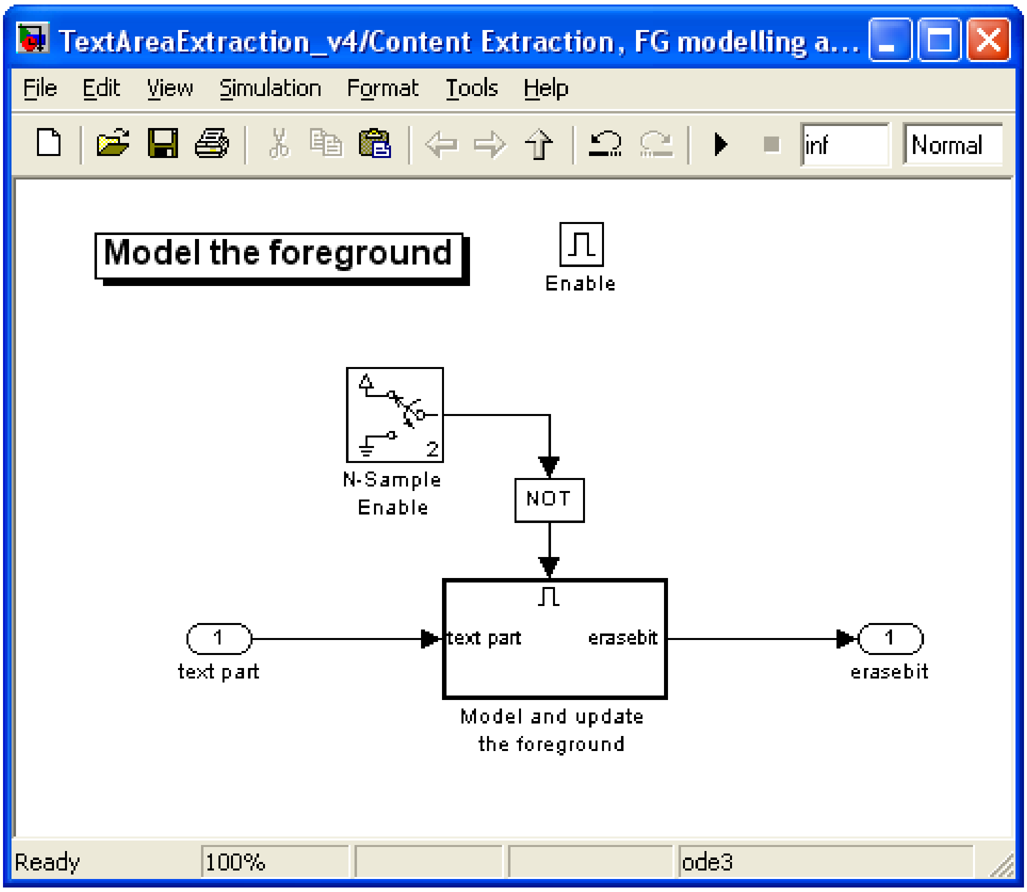


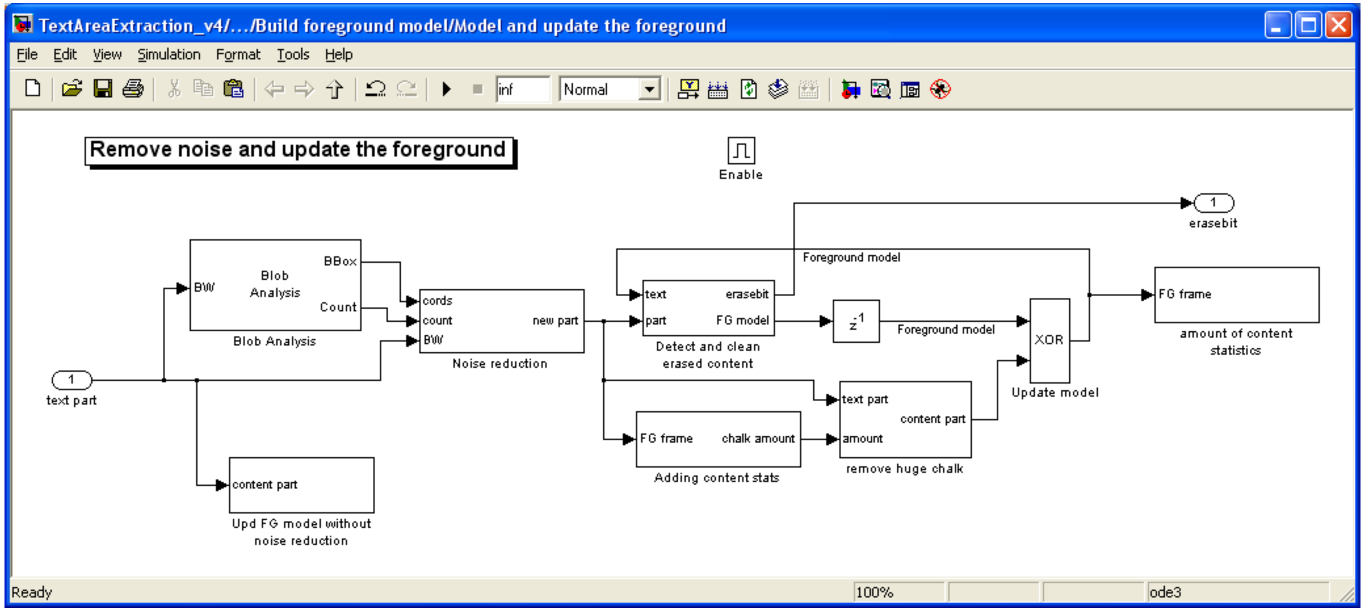


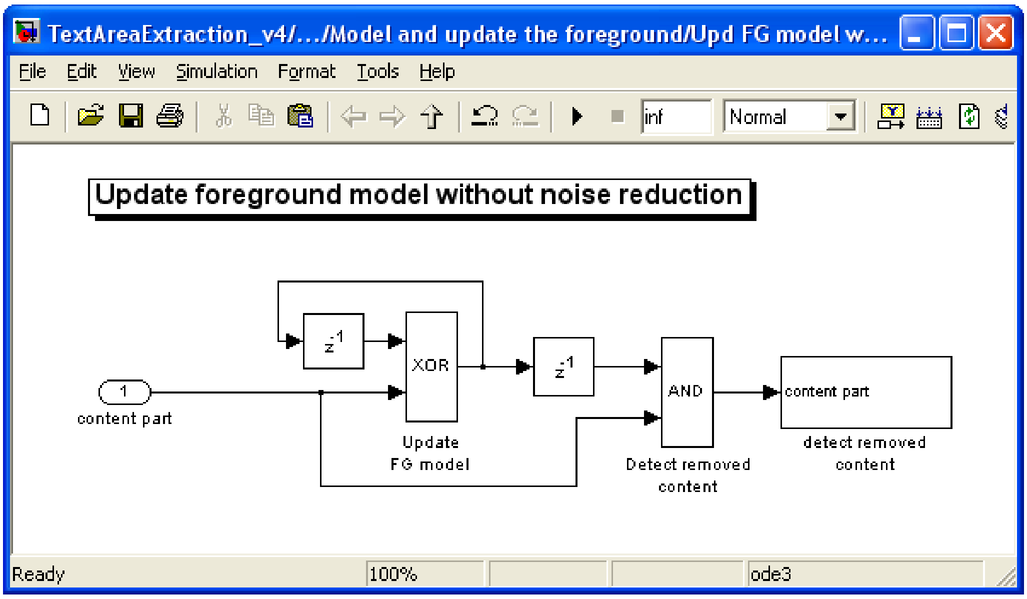
TextAreaExtraction_v4/Content Extraction, FG modelling and detection of added and removed c...
File Edit View Simulation Format Tools Help
inf
Normal
Content Extraction, FG modelling and detection of added and removed content
1
segm. frame
segm. frame
content part
Extract content parts
U(:)
Reshape
Sum of Elements
== 0
Compare To Zero
NOT
text part
erasebit
Build foreground model
1
erasebit
Ready
100%
ode3


TextAreaExtraction_v4/Content Extraction, FG modelling and detection of added and re...
File
Edit
View
Simulation
Format
Tools
Help
inf
Normal
Extract content parts
1
segm. frame
z^{-1}
−
+
Subtract
|u|
Abs
0.14
Difference Threshold
<
NOT
Logical Operator
1
content part
N-Sample Enable1
2
NOT
diff frame
segm frame
Autocalk threshold
Ready
100%
ode3

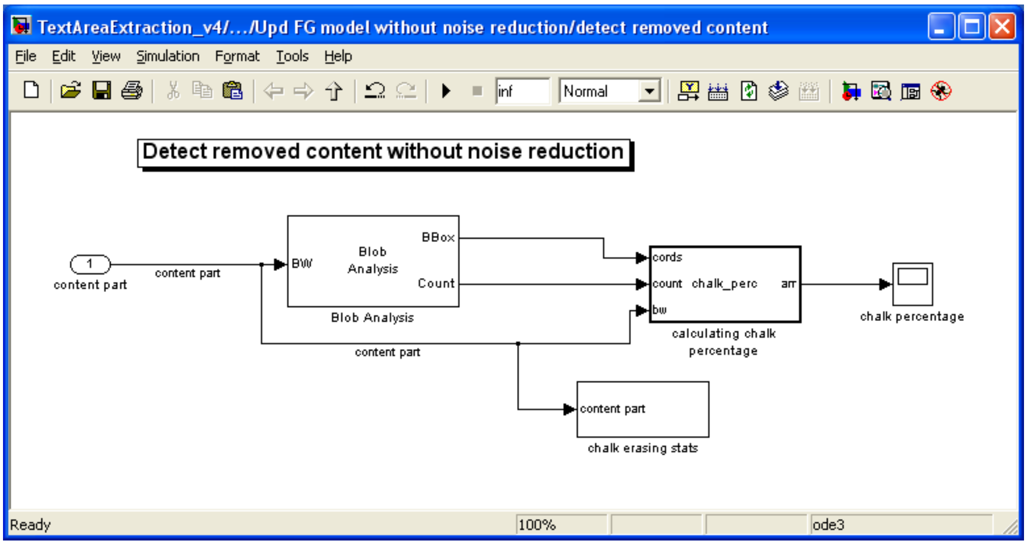

TextAreaExtraction_v4/.../Extract content parts/Autocalk threshold
File Edit View Simulation Format Tools Help
inf
Normal
Autocalculate threshold value
Enable
2
segm frame
z-1
I
ROI
Std Dev Prev
2
Gain1
+
+
Add1
I
ROI
Mean prev
st.dev.
previous frame
1
diff frame
>=
Relational
Operator
0.14
Constant
BW
Blob
Analysis
BBox
Count
Blob Analysis
stdev
cords
count
THframe
Iframe
updTH
fcn
newFrame
newTH
Embedded
MATLAB Function
FG
+
+
Adding
FG model
z-1
>=
Relational
Operator1
Calc. threshold
Calc. threshold
z-1
Calc. threshold
threshold
Ready
100%
ode3

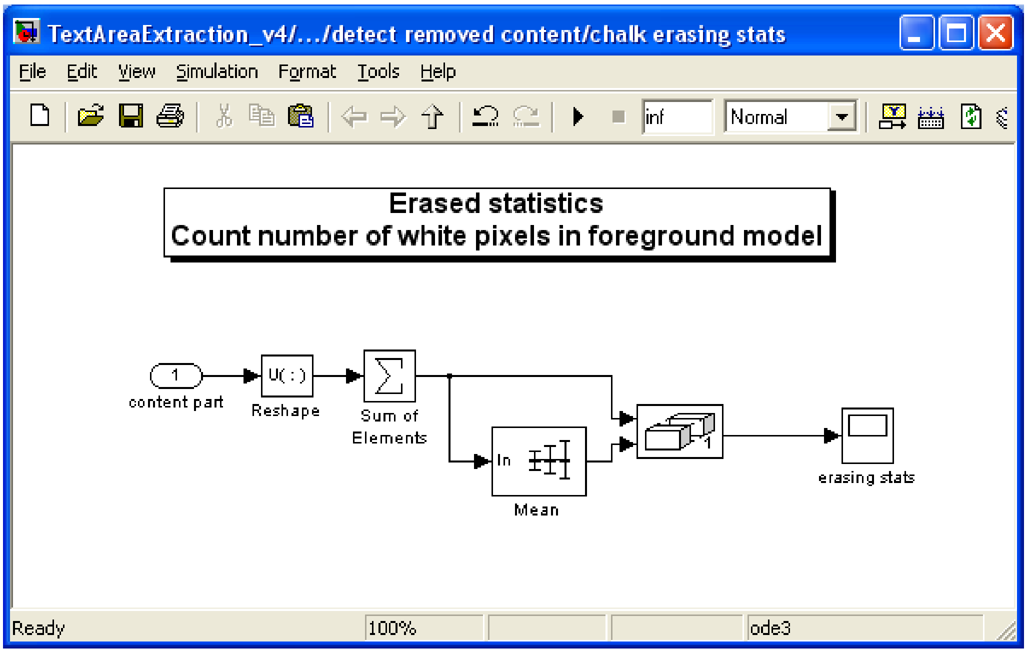

TextAreaExtraction_v4/Content Extraction, FG modelling a...
File Edit View Simulation Format Tools Help
inf
Normal
Model the foreground
Enable
2
N-Sample Enable
NOT
text part
erasebit
1
text part
1
erasebit
Model and update the foreground
Ready
100%
ode3


TextAreaExtraction_v4/.../Build foreground model/Model and update the foreground
File Edit View Simulation Format Tools Help
inf
Normal
Remove noise and update the foreground
Enable
1
erasebit
1
text part
BW
Blob
Analysis
BBox
Count
Blob Analysis
cords
count
BW
new part
Noise reduction
text
part
erasebit
FG model
Detect and clean
erased content
Foreground model
z^-1
Foreground model
XOR
Update model
FG frame
amount of content
statistics
FG frame
chalk amount
Adding content stats
text part
amount
content part
remove huge chalk
content part
Upd FG model without
noise reduction
Ready
100%
ode3



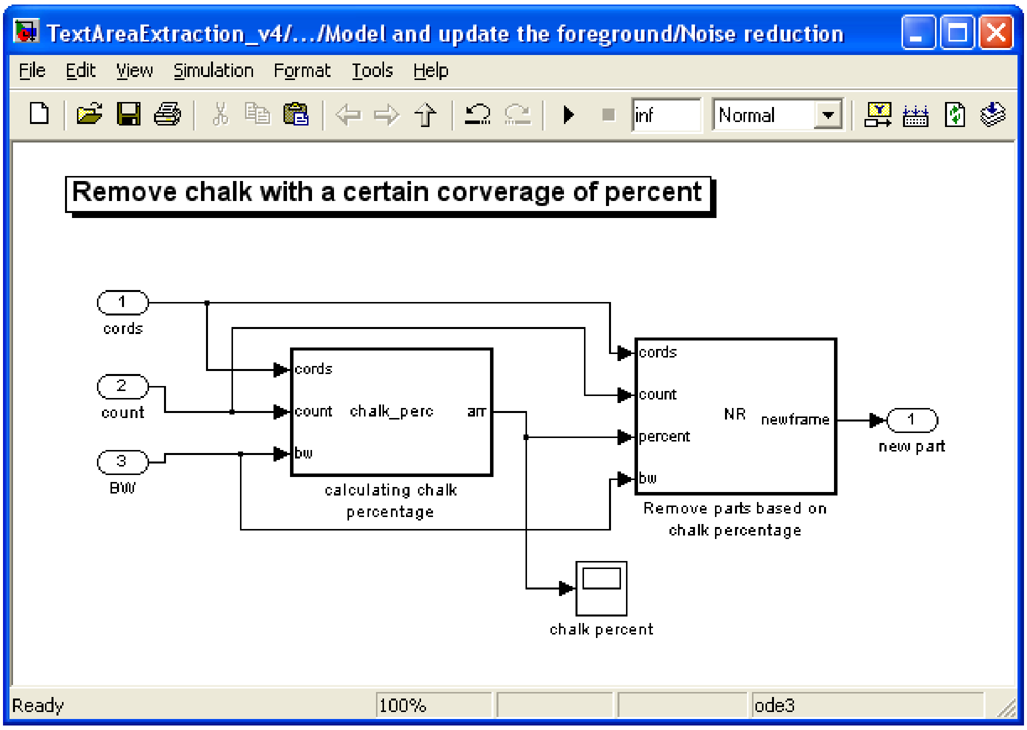
TextAreaExtraction_v4/.../Model and update the foreground/Upd FG model w...
File Edit View Simulation Format Tools Help
inf
Normal
Update foreground model without noise reduction
1
content part
z^{-1}
XOR
Update
FG model
z^{-1}
AND
Detect removed
content
content part
detect removed
content
Ready
100%
ode3

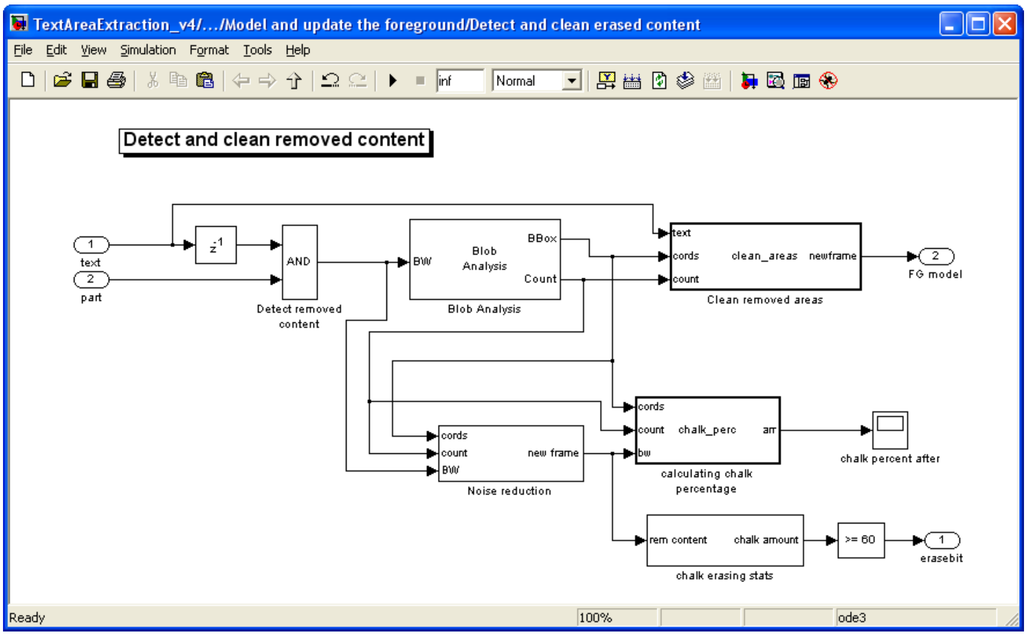

TextAreaExtraction_v4/.../Upd FG model without noise reduction/detect removed content
File Edit View Simulation Format Tools Help
inf
Normal
Detect removed content without noise reduction
1
content part
content part
BW
Blob
Analysis
BBox
Count
Blob Analysis
content part
cords
count chalk_perc arr
bw
calculating chalk
percentage
chalk percentage
content part
chalk erasing stats
Ready
100%
ode3

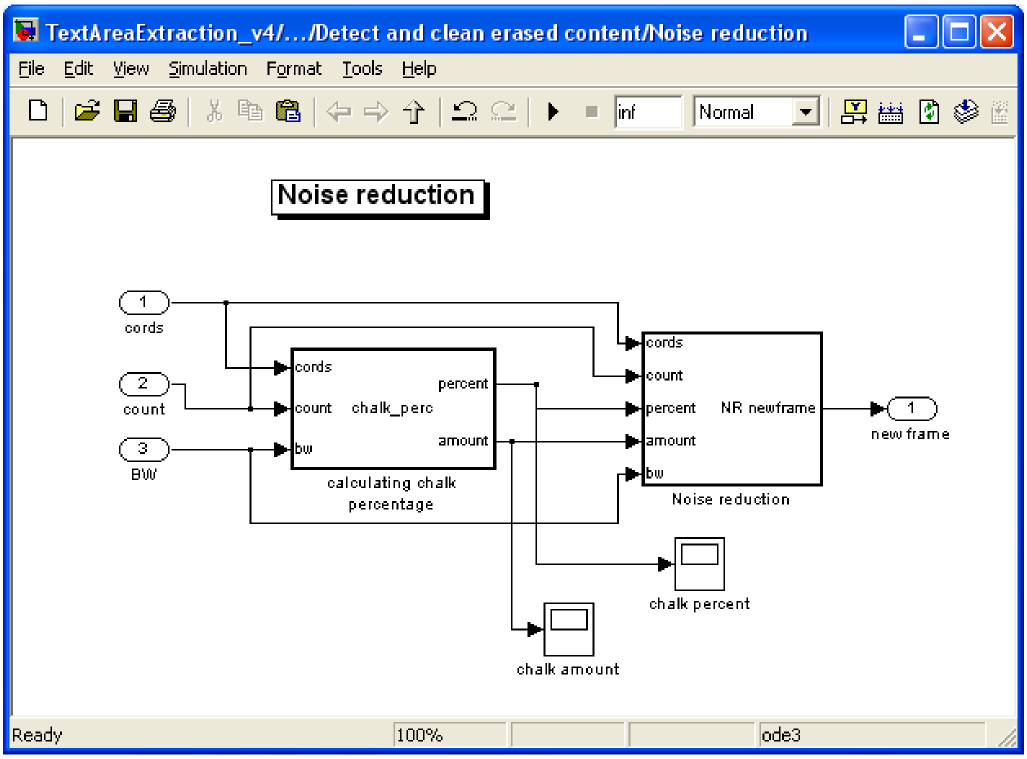


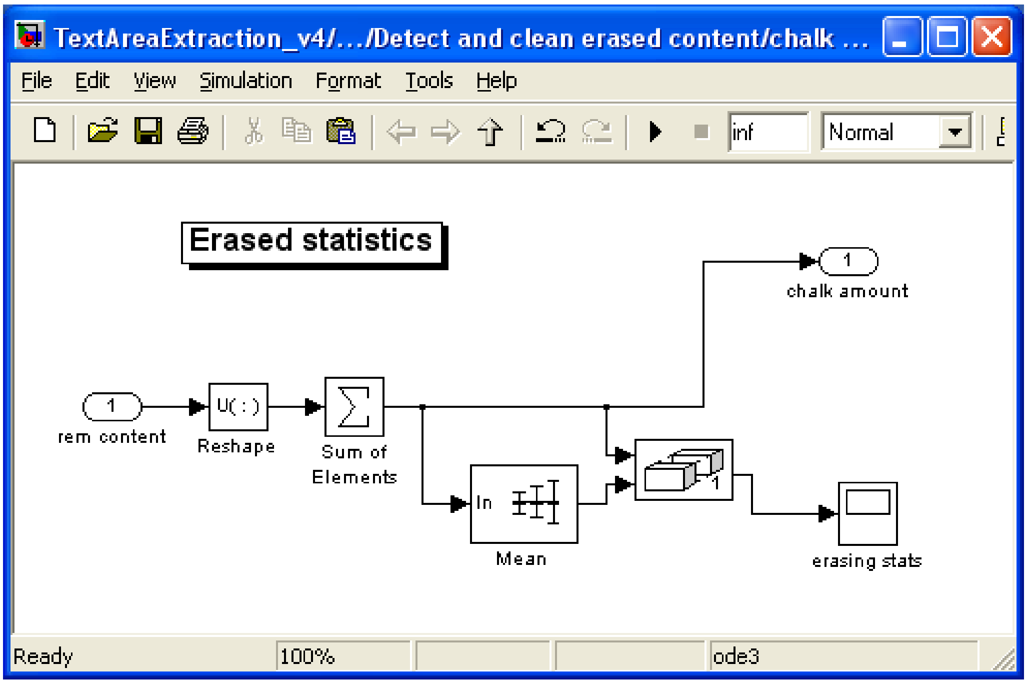
TextAreaExtraction_v4/.../detect removed content/chalk erasing stats

Erased statistics
Count number of white pixels in foreground model

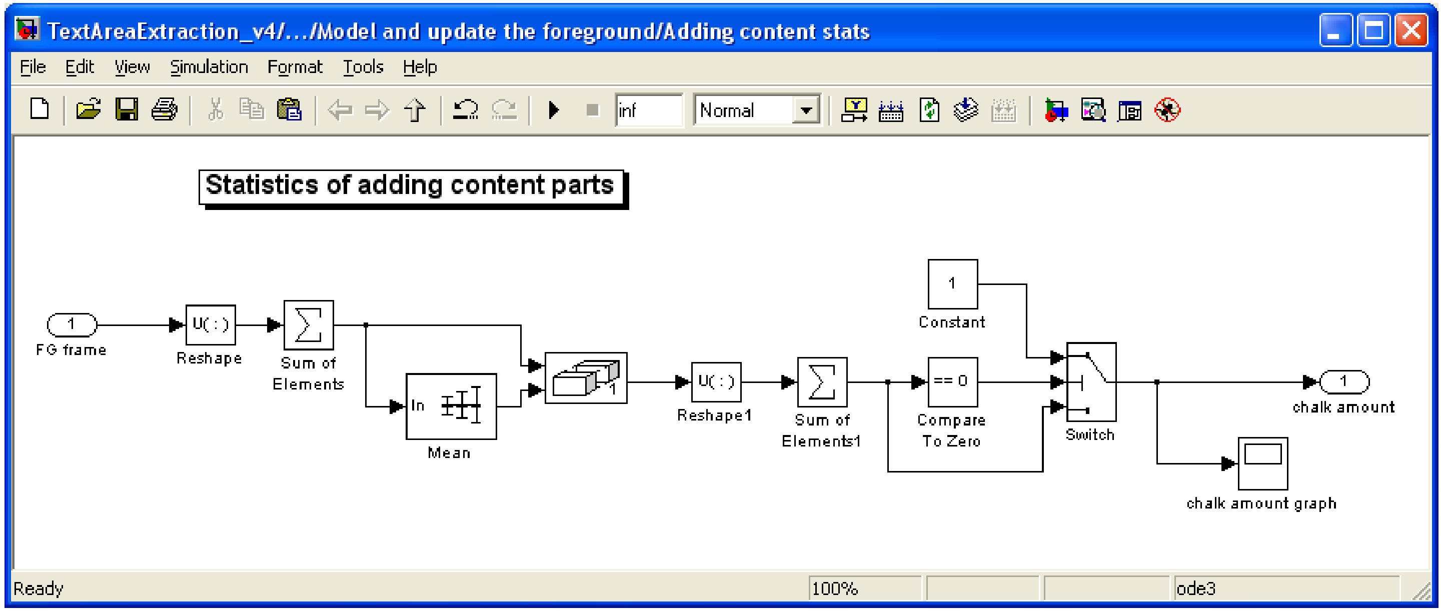

TextAreaExtraction_v4/.../Model and update the foreground/Noise reduction
File Edit View Simulation Format Tools Help
inf
Normal
Remove chalk with a certain corverage of percent
1
cords
2
count
3
BW
cords
count
bw
chalk_perc
arr
calculating chalk percentage
cords
count
percent
bw
NR
newframe
Remove parts based on chalk percentage
1
new part
chalk percent
Ready
100%
ode3


TextAreaExtraction_v4/.../Model and update the foreground/Detect and clean erased content
File Edit View Simulation Format Tools Help
inf
Normal
Detect and clean removed content
1
text
2
part
z-1
AND
Detect removed content
BW
Blob Analysis
BBox
Count
Blob Analysis
text
cords
count
clean_areas
newframe
Clean removed areas
2
FG model
cords
count
bw
chalk_perc
arr
calculating chalk percentage
chalk percent after
cords
count
BW
new frame
Noise reduction
rem content
chalk amount
chalk erasing stats
>= 60
1
erasebit
Ready
100%
ode3

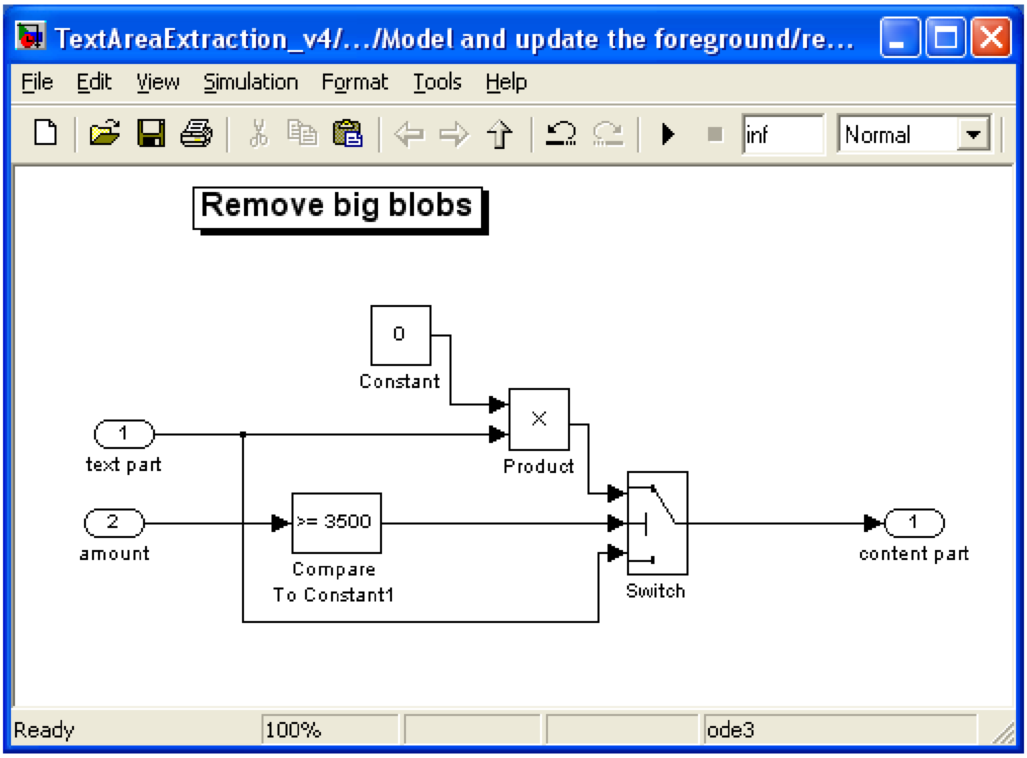

TextAreaExtraction_v4/.../Detect and clean erased content/Noise reduction
File
Edit
View
Simulation
Format
Tools
Help
inf
Normal
Noise reduction
1
cords
2
count
3
BW
cords
count
bw
percent
chalk_perc
amount
calculating chalk percentage
cords
count
percent
NR newframe
amount
bw
Noise reduction
1
new frame
chalk percent
chalk amount
Ready
100%
ode3

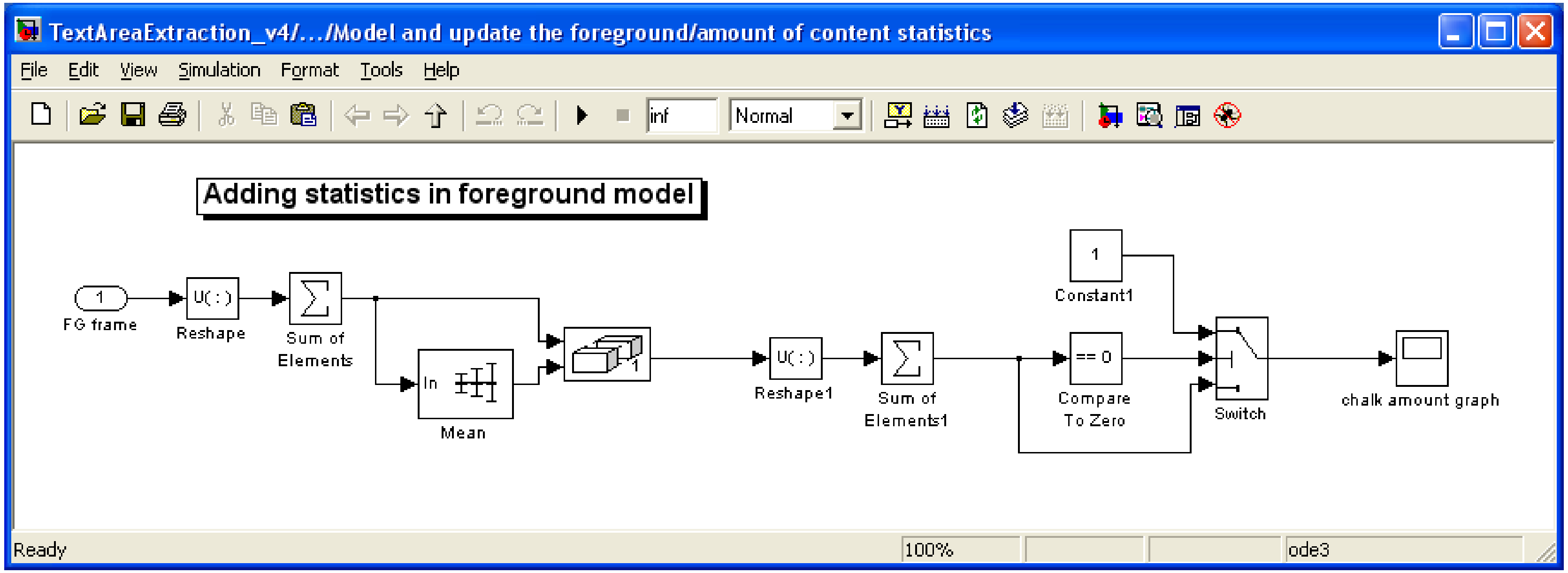


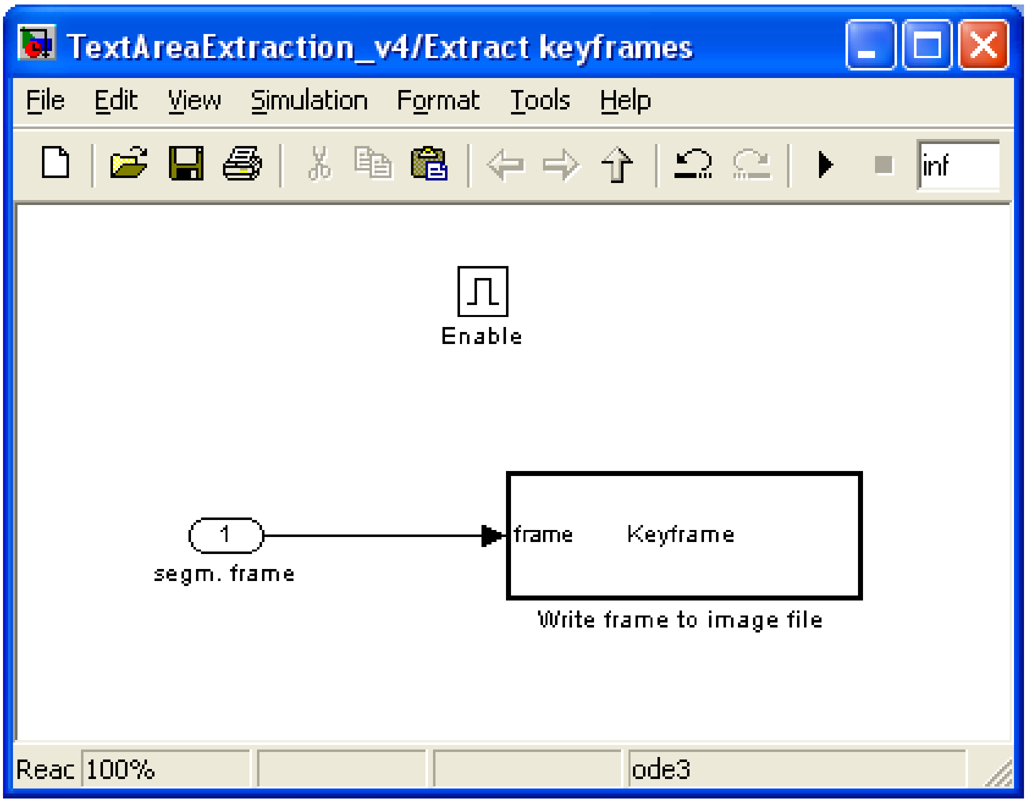
TextAreaExtraction_v4/.../Detect and clean erased content/chalk ...
File Edit View Simulation Format Tools Help
inf
Normal
Erased statistics
1
chalk amount
1
rem content
U(:)
Reshape
Sum of
Elements
In
Mean
1
erasing stats
Ready
100%
ode3



TextAreaExtraction_v4/.../Model and update the foreground/Adding content stats
File Edit View Simulation Format Tools Help
inf
Normal
Statistics of adding content parts
1
FG frame
U(:)
Reshape
Sum of Elements
In
Mean
1
U(:)
Reshape1
Sum of Elements1
1
Constant
== 0
Compare To Zero
Switch
1
chalk amount
chalk amount graph
Ready
100%
ode3


TextAreaExtraction_v4/.../Model and update the foreground/re...
File Edit View Simulation Format Tools Help
inf
Normal
Remove big blobs
0
Constant
Product
1
text part
2
amount
>= 3500
Compare
To Constant1
Switch
1
content part
Ready
100%
ode3


TextAreaExtraction_v4/.../Model and update the foreground/amount of content statistics
File Edit View Simulation Format Tools Help
inf
Normal
Adding statistics in foreground model
1
FG frame
U(:)
Reshape
Sum of
Elements
In
Mean
1
U(:)
Reshape1
Sum of
Elements1
1
Constant1
== 0
Compare
To Zero
Switch
chalk amount graph
Ready
100%
ode3


TextAreaExtraction_v4/Extract keyframes
File
Edit
View
Simulation
Format
Tools
Help
inf
Enable
1
segm. frame
frame
Keyframe
Write frame to image file
Reac
100%
ode3

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Faculty of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Abstract

E-learning technologies are aimed at augmenting the students' learning experiences - for in-stance by producing lecture videos that can be viewed anywhere, at any time. Lecture videos contain most of the instructional content, but their effective use remains challenging. One reason is that professional editing is not an option for most educational institutions due to costs. Therefore, most lecture videos are unedited. Unedited videos are rarely captivating enough to keep the students' attention for long. Neither does it facilitate easy navigation. Hence, there is a need for automatic systems for lecture video indexing and summarization.

In this thesis we propose an automatic system for content-based indexing of blackboard lecture videos. We first use low-level video processing algorithms to segment a scene into foreground moving objects and a slowly changing background consisting of the blackboard and the text written on it. Next, the blackboard contents are enhanced and extracted. The extracted contents are analysed in order to define key frames; such as when the teacher writes some text or erases it. Additionally, since it is likely that some of the blackboard content may be hidden at any point in time due to occlusion by the instructor, images to summarize the board contents are created. Extracted key frames and images may be used to efficiently navigate through lecture videos.

Sammendrag

Teknologier innen e-læring er rettet mot å bedre studentenes læringsopplevelse. Dette kan for eksempel være ved hjelp av forelesningsvideoer som kan sees fra hvor som helst, og når som helst. Forelesningsvideor inneholder det meste av det faglige innholdet, men å benytte en forelesningsvideo til læring på best mulig måte er en utfordring. Dette kommer av at profesjonell redigering er ikke et valg for mange utdanningsinstitusjoner på grunn av kostbarheten, og derfor er veldig mange forelesningsvideor uredigerte. Det er vanskelig å beholde en students oppmerksomhet gjennom en uredigert video, og det er heller ikke enkelt å navigere seg igjennom. Derfor er det bruk for automatiske systemer for å indeksere forelesningsvideoer.

I denne oppgaven foreslås det et automatisk system for innholdsbasert indeksering av tavle forelesningsvideoer. Først bruker vi enkel video prosesserings algoritme for å segmentere en scene til bevegelige objekter i forgrunnen og en langsomt endrende bakgrunn bestående av tavlen og teksten som er skrevet på den. Deretter definerer vi nøkkel rammer; som for eksempel når læreren skriver eller tar bort tekst. I tillegg, siden det er sannsynlig at noe av innholdet på tavlen til tider vil være skjult av feks læreren, vil rammer benyttes til å bygge opp en oppsummering av innholdet. Nøkkel rammer kan brukes til å effektivt navigere i en forelesningsvideo.

Preface

This master thesis has been carried out at the Department of Computer Science and Media Technology at Gjøvik University College. The purpose has been to investigate methods to create automatically generated indexes for lecture videos, based on their content. An index for a lecture video is valuable for for students who can not attend ordinary lectures or for revision purposes. This thesis work shows that high level indexes are possible to extract from lecture videos.

I would like to thank my head supervisor Dr.Tech. Faouzi Alaya Cheikh for very good technical help, advices, motivation, and for following me up during the thesis period. I would also like to thank Dr.Ing. Rune Hjelsvold for helpful comments and ideas on my work and report. Also, special thanks to both of them for encouraging and helping me getting an abstract based on this thesis approved for DIVERSE2007 conference at Lillehammer, and exhibiting a poster. I will also thank Siv.ing. Anders Oulie for letting me record one of his lectures and use it for my thesis. Finally I want to thank my sister, Cand.Scient. Therese H. Røst, for reading corrections on my report.

Martin Røst Halvorsen, 2007/05/30

List of Figures

List of Tables

Contents

1 Introduction

1.1 Purpose of this thesis

The purpose of this thesis is to propose an automatic system for indexing lecture videos, based on their content. The index is to be created automatically from a high quality recording of a lecture. Having an automatically created index the process of publishing lecture videos would become much easier, and without manual editing.

Keywords: video segmentation, semantic structure, lecture video, automatic index, text segmentation, meta-data, content

1.2 Problem description

All around the world there are students who live in other geographical areas than the educational establishment where their lectures are held. Both students that have a long travel route and distance-learning students could benefit from other ways to be educated. E-learning systems are an approach to educate these students. By using multimedia tools and contents students are able to study anywhere and anytime.

Many educational establishments make lecture notes and other related syllabus available on the Internet. In most cases PowerPoint slides and other notes make up the basis for the educational material. However, the information given during a lecture, both spoken and written information, may be lot more in-depth and elaborative than the information published. Surveys show that students consider traditional blackboard presentation as essential and indispensable [8].

Multimedia content makes the material accessible at all times for students that for instance need a revision or for those who missed a lecture. Videos of lectures could be an important source to educational information, especially in distance teaching. Even though both the expertise and equipment to create a summary video of a lecture is available, it might be time consuming and not prioritised by some teachers. Without an index, the video may be hard to navigate in. For instance in the process of revision, one may have to watch the whole video or manually navigate through it in order to find the specific information wanted. This is not a user-friendly option and for some it will be easy to give up. However, a video that is easy to navigate through may need a lot of pre-processing before it can be published.

To create an automated system for generating an index for a lecture video, some problems need to be addressed. In *Lecture Videos for E-learning: Current research and challenges* [8] several proposed works on lecture indexing are discussed. The conclusion is that more work needs to be done in this field. Indexing and recognition of teaching topics are the most crucial challenges in lecture videos.

A good index might consist of images from the blackboard, text outlining, descriptive text, drawings and so on. All of this is dependent on a clear blackboard and visible text, high resolution of the text, low noise etc. This means that object segmentation and content extraction are useful methods to overcome these challenges.

Video segmentation has been done for a long time, but in different ways. Some are used in video surveillance [9] [10] [11], object tracking and recognition [2] [12] [13] [14] and in television news broadcast [15] or weather map background situations [16]. Making good lecture videos manually is often time consuming and difficult. Although work has been done on some of the problems regarding segmentation and automatic creation of lecture summaries [1] [4] [3], still there is no complete system that can handle this task in a good way. For instance, a generic solution to the segmentation problem has not been addressed earlier. The segmentation would work for green blackboards, but not for whiteboards. Some of the segmentation algorithms used in other contexts are possible to use to segment the teacher in lecture videos. But the main difference between lecture video segmentation and for instance video surveillance is that there is mostly one person to segment, the person is close to camera and there is valuable background information. Also many segmentation algorithms use background subtraction to remove the background and keep the foreground. In lecture video it's opposite; the teacher has to be removed so that the content on the blackboard is clear and visible.

Extraction of meta-data from a lecture video has partly been done before [17] [6] [18]. Meta-data can be useful as indexes in a lecture video if they could be extracted automatically. A meta-data should therefore be a descriptive object, which would tell a person for instance something about the topic for the lecture is at a certain point. In a lecture video several things can be included as a meta-data; when the teacher is writing, cleaning out the board or talking to the students. Other meta-data may be figures and headlines. There are other methods that make use of the audio channel to segment lecture videos into topics [19] [20], but this will not be covered in this thesis. Tang et al. [21] has worked on text detection and recognition in lecture videos, using a table-of-contents from a course-book to compare with text in the videos. Text recognition and extraction of information from other sources than the lecture video will also not be covered in this work.

Different types of meta-data can be used for information retrieval in different ways. If it is possible to recognize text, text search can be used to find content in a video, but text recognition is not covered here. Figures can often be associated with content and could therefore be useful as an index. The structure of a lecture video index can also be influenced by the teacher's movements and actions. For instance if the teacher is not writing for a long period of time, this could be indexed as a talking scene, with a screenshot of the blackboard. This kind of meta-data extraction and the use of them has not been done before. However, Onishi et al. propose a method where they extract blocks of text from the blackboard and rearrange them for web publishing [6], and Repp proposes a simulated indexing algorithm based on the amount of increased chalk on the blackboard over time [17].

1.3 Justification, motivation and benefits

A large number of educational establishments don't have the required equipment or the expertise to create a user-friendly lecture video. Many videos that are created have no index and are therefore hard to navigate in. Some requirements have to be fulfilled in order to get students to use the lecture videos and have utility value from them. The video has to have good light conditions, good sound and to be navigable with for instance an index. By the use of technology, it is possible to solve the indexing problem. However, technology may not improve individual skills concerning other aspects of movie making.

There is a reason to believe that educational establishments that offer lecture videos may benefit from attracting more students, and different types of students. For instance, this may be highly relevant for distance learning students and those who for different reasons don't attend ordinary lectures. On a general basis, all students may benefit from having all lectures accessible at all times for self-studying or revision purposes.

Therefore an automatic process for creating an index for a lecture video should be of interest to a wide number of educational establishments. If the automatic system can create an index that is valuable for students in finding or revising information in a lecture video, more educational establishments can take advantages of this. It would then be possible to create good lecture videos without expensive professional applications, or skills in video editing. My contribution in the field of lecture video indexing may be used by others as building blocks to create better and more complete systems in the future.

1.4 Research questions

The research questions for this thesis are the following:

Q1: How can foreground/background segmentation of lecture videos work for different writing-boards?

When taking different writing-boards into account, the boards' possible colour or dimensions cannot be used as parameters in the algorithms. Additionally, there could be other people entering the scene. The main use for this segmentation is to segment out the teacher who stands in front of a writing-board. There should be as few static variables as possible to make the algorithm generic for lecture videos with textual content.

Q2: How to automatically extract meta-data from lecture videos?

Meta-data is defined as actions performed by the teacher. A meta-data should therefore be descriptive, and tell a person something of what it is about. In a lecture video several things can be included as a meta-data; when the teacher is writing (increase of ink/chalk), cleaning the board (wipe out blackboard, decrease of chalk, turn page on a flip-over) and when the teacher is talking (no increase or decrease of ink/chalk over a time period). Other meta-data may be figures and headlines. Meta-data can be useful as indexes in a lecture video.

Q3: How to use such meta-data for indexing and searching of lecture videos?

A good navigation tool in a lecture video is important to be able to find the information one looks for. By using meta-data for indexing purposes it would be possible to easily find information and navigate through a video of a lecture. Different approaches will be discussed on how to index, or maybe create textual meta-data or description of some of the meta-data extracted. The textual meta-data can furthermore be used in text-based searching.

2 Related work

In this chapter a review of the state of the art related to the areas covered in this thesis will be presented.

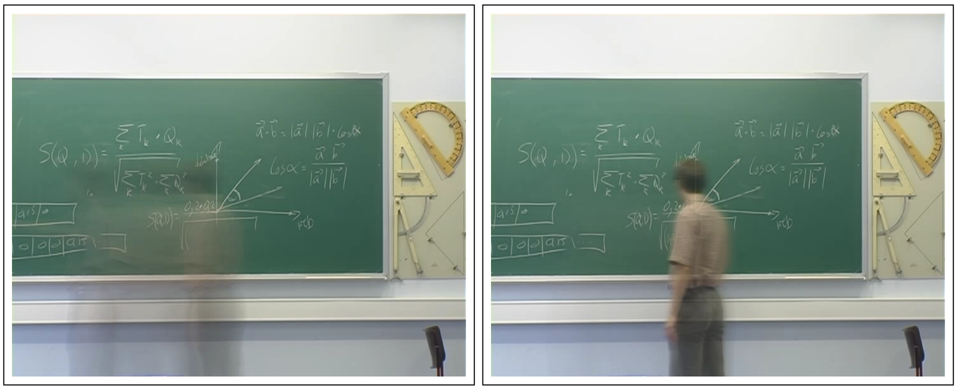
2.1 Foreground and background segmentation

In video segmentation there are two basic segmentation approaches; foreground and background subtraction. When using background subtraction the aim is often to track changes in the foreground. This can for instance be in video surveillance where the important part is to track objects such as people or cars. Another example might be in television news broadcasts where the news reader is the important part. Also in weather forecasts the anchor person has to be tracked in order to change the weather maps in the background. In foreground subtraction the purpose is often to track changes in the background, for instance in lecture videos where the blackboard and its content is of importance. Another application is in video surveillance where baggage is tracked to see if baggage is being left behind.

In this thesis the segmentation part shall remove any moving objects, mainly the teacher, from the video. There are two reasons for this. Firstly, the blackboard and its content are going to be used as indexes. It is easier to recognize parts from a lecture, and read the blackboard's content when there are no objects occluding it. Secondly, the segmented video is going to be further processed to extract meta-data content. Further discussion in this section will seek for techniques which can be used in segmentation of a teacher in a lecture video.

2.1.1 Background modelling

Background modelling is often used in background subtraction algorithms. By modelling the background, moving objects can efficiently be detected in a video sequence with the use of pixel differences from the modeled background and the consecutive frames. There are different ways to build a model. It is important that the model does not have any foreground objects in it. In many cases it is not possible to get a good background model using a single frame. Therefore, a background model must be built using a few frames from the video to remove moving objects in time, for instance at high traffic highways, crowded places and so on. One way to do this is by using temporal median filtering [1]. This technique can for instance be used for the first 1 or 2 minutes to build the background. The problem with this method is that it assumes that objects are moving most of the time. Therefore, if the teacher stands at the same place for a long period, the teacher becomes part of the background. Figure 1 is taken from Grythe's work on temporal median filter. It shows it is possible to partly hide the teacher while he is moving, but the teacher is appearing when he moves slowly or stands still. Median filtering is also computationally expensive because of the sorting operation. In a lecture video it is not easy to get hold of a clean and empty background scene, so more advanced methods have to be implemented for background modelling.

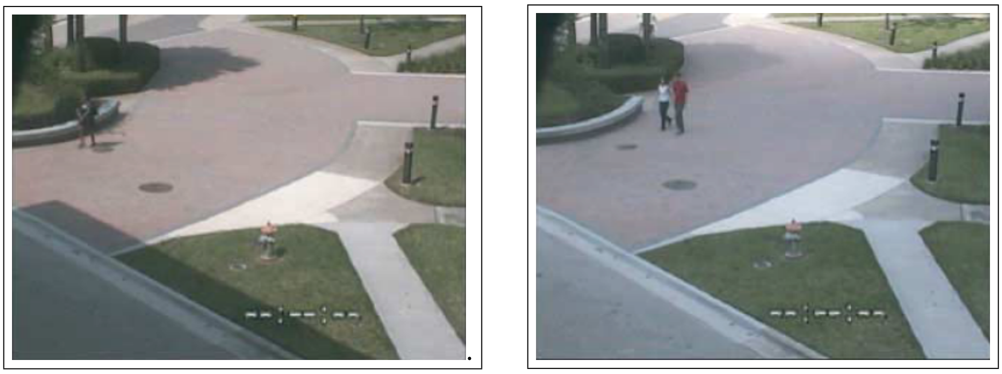


(a) The teacher moves

(b) The teacher is moving slowly

Figure 1: Temporal median filter have problems with slowly moving objects [1].

There are some situations where background modelling and differencing methods perform poorly. For instance when there is quick illumination changes, relocation of background objects, initialization with moving objects and shadows in the scene [2]. Quick illumination changes can occur for instance when the lights are turned on or off, when sunlight comes through a window etc. This will have an effect on the segmentation, and this is why it is important to update the background model so that the whole frame isn't considered as a moving object when for instance the lights are turned off. Figure 2 illustrates changes in the scene after som period of time. Clouds may be moving and hide the sunlight from the sun causing the background model to change illumunation values. These changes can cause the background to be detected as people objects, like figure 2(a) illustrates. Adaptive background models build and maintain the background model, for instance by using statistical modelling.

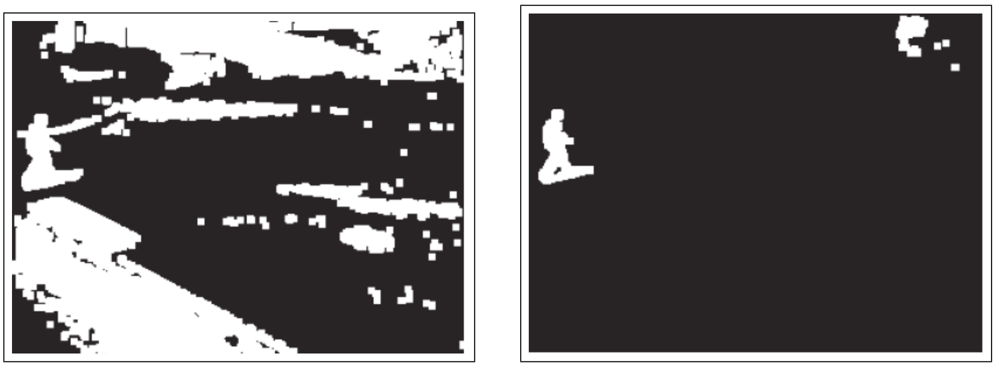


(a) Little shadow

(b) Much shadow

Figure 2: Shadow appearing in time causes problems [2].

Javed et.al [2] discuss a three level algorithm using a statistical method. The levels are divided into pixel, region and frame levels. Gradients of images are less sensitive to illumination changes than colour based background systems. A technique combining colour and gradient information will deal better with illumination changes in the scene. The colour based subtraction is done using a mixture of K Gaussians distributions to model each pixel colour. These statistical models are separately used to classify each pixel as belonging to background or foreground. At the frame level global illumination is detected, and uses only the gradient based subtraction if more than 50% of the colour based background subtracted becomes a part of the foreground. The result before and after applying the three level algorithm can be seen in figure 3.



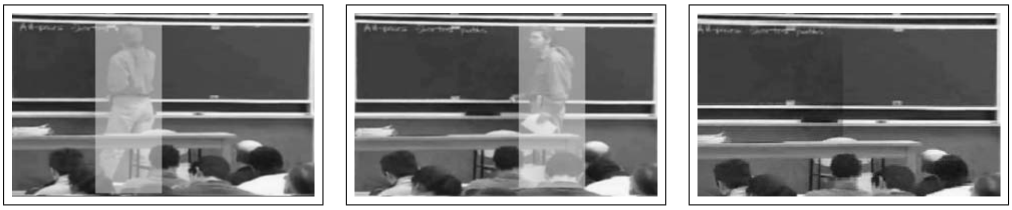
(a) Shadow causes problems

(b) Shadow problems overcome

Figure 3: Shadow problematics fixed [2].

Another approach is to calculate the redundancy ratios of the intensity values for each pixel over several seconds to distinguish pixels with moving objects from pixels with stationary objects [9]. Each of these intensity values for one pixel over a period of time are compared to each other. The intensity variations and redundancy intensity values for that pixel are stored. The pixel with the biggest redundancy ratio (BRR) represents the background in the background model. This method is based on the median filter, but differs in that the BRR doesn't assume that the background is visible more than 50% of the time, and the BRR method is rather faster since it doesn't need to sort the values, like the median filter does. Two independent methods are proposed to update this model. The first method uses a statistical model of the background to adapt to slow changes using temporal filtering. The second approach is a pixel-based method which updates the model periodically, using the BRR method, which can adapt to illumination and relocation of objects in the background scene. The latter is a periodical re-initialization of the background model.

Another technique to model the background is based on calculation of the ratio of intensity values [3]. The idea is to model the background and later the teacher can be tracked in the subsequent frames. The proposed modelling technique is done by taking two frames widely separated in time and compute the ratio of the intensity values for each pixel. The pixels are then clustered based on the intensity ratio and a pre-specified threshold, and then enclosed into rectangular boxes around the teacher. Figure 4 illustrates the method. If the boxes in these two frames are not overlapping, there is enough information to segment the teacher. If the boxes are overlapping then another frame is chosen until the boxes are not overlapping. This method will not work if there are several people in the scene, because there is too little information about the background. Also if one of the frames contains chalk then the chalk will be modeled as background. This method also updates chalk changes from time to time.



(a) Teacher in one area (b) Teacher in a different area (c) Segmented frame

Figure 4: Teacher segmentation method by Mittal et.al [3].

Bhat et.al [22] proposed to use some filters to represent each pixel, in order to model the background. $I_{n,x,y}$ represent a pixel at time n and at position x,y. $B_{(n-1),x,y}$ represents the predicted background value for this pixel. A moving object is detected if there is a significant difference between the $I_{n,x,y}$ and $B_{(n-1),x,y}$. An adaptive threshold value for each pixel is used to classify it as foreground or background, based on the difference value. This approach is also used by Niu et.al [10] for person tracking. This method suffers when objects move slowly, because the objects go from the foreground to the background scene. Bhat et.al [22] implemented a conditionally lagged background model so that slowly moving objects don't become part of the background.

Gaussian distribution model is a statistical modelling technique that is used in many background subtraction algorithms. For instance the BRR method described above could be modeled with a Gaussian distribution. In outdoor environments where there are almost no static scene because of waving trees and bushes, a single Gaussian will therefore not hold. A mixture of Gaussians will be better for such outdoor environments. Another approach to model variations in intensity is to represent the variations as modes in the environment using Hidden Markov models (HMMs). These models can for instance handle lights on/off in a scene [11].

2.1.2 Foreground modelling

These statistical models described in the previous section may require a lot of processing for a lecture video. However, indoor scenes do not have the problem with constant small motion found in outdoor environments. The main focus in a lecture video when creating indexes is the blackboard and it's content. If we can remove objects that are moving in front of it, like the teacher, it should be a good start. The discussions above have focused on temporal filters, background models and adaptation to changes in the model, which are typical approaches in background subtraction. We are now going to focus on different approaches for foreground subtraction where it is typical to track and remove foreground objects, while preserving the background.

An earlier master thesis reports on work with segmentation and text detection in lecture videos [1]. The segmentation is based on foreground subtraction to remove the teacher so that the text on the blackboard could be detected and recognized. The proposed approach was to find the teacher's outline, and make it as accurate as possible. First a mask was created to separate the teacher and the blackboard. The mask was based on the green colour of the blackboard and was chosen manually. A threshold value was manually computed and used to create binary frames so that the blackboard was detected. Figure 5 show the binary frame after applying a threshold value. There is a discussion in the thesis about the difference using RGB or YCbCr colour space. Is was concluded that using YCbCr is the best due to the separation of luma and chroma channels.

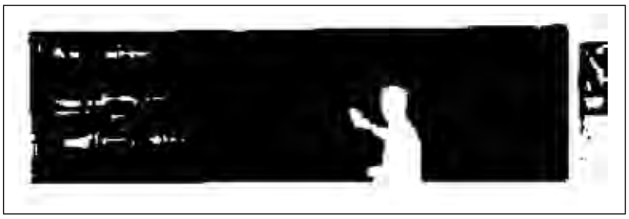


Figure 5: Blackboard separation using YCbCr colour space [1].

Morphological operations were applied to remove holes and improve the segmentation of the teacher. Prewitt edge detection was performed on the binary frame to get the outline of the teacher as seen in figure 6. Only the area within the blackboard was used. The segmentation was tested on frames in RGB and YCbCr colour space. The results in Grythe's [1] thesis shows that working with YCbCr colour space is preferred, because it's easier to work with than RGB and the segmentation can be done with fewer processing blocks. Figure 7 shows a segmented videoframe using YCbCr colour space. The results also shows that using the objects' outline and silhouette accuracy in teacher segmentation makes some parts of the object move out of the object boundary, like clothes and hair. These parts then get part of the background. Instead of using the teacher's outline, it might be better to segment a bigger surface around the teacher. The surface can later be updated when the teacher moves. The segmentation approach used is also only for green blackboards. If a lecture video has a black or white board the segmentation would not work because of the fixed colours in the algorithms. A generic solution will therefore be a better choice for lecture videos.

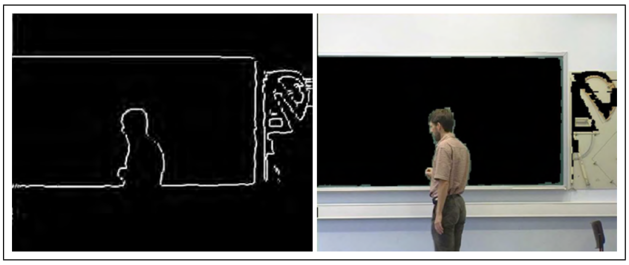


Figure 6: Contour detection using Prewitt edge detection [1].

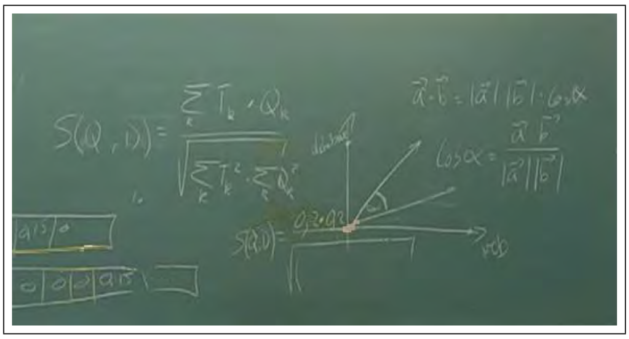


Figure 7: Segmented frame using YCbCr colour space [1].

Recently a dissertation on electronic boards (E-boards) and segmentation [4] was published. The E-boards are different from the ordinary writing boards (e.g. blackboard, whiteboard, hand slides) in that the E-board sends content to a computer which stores the information in a vector-based format. The content is then easily available for processing. Another difference is the strokes on the board. Chalk is used on ordinary blackboards, while the E-board register marks by suing a special pen. The E-board can also display images, diagrams and content from the Internet. If the teacher has filled the board it is possible to scroll the content out of the board, getting a new empty board-screen. The strokes will, in different environments, generate illumination changes. Displaying content from the computer on the board and scrolling the content will generate motion. As discussed earlier reflections might also be a problem in the segmentation part. A lecture video based on E-boards is created by segmenting the teacher with motion estimation and colour distribution, and makes the area of the segmented teacher a little transparent. Afterwards the vector-based strokes are placed on top of the teacher as shown in figure 8. This way a student can watch the blackboard content and the teacher's facial expressions and gestures. In the above mentioned dissertation there is a brief discussion about motion statistics as part of the segmentation. One of the problems by using motion estimation is when the teacher doesn't move. The teacher then becomes part of the background. A combined approach of motion statistics and colour distribution classifier was discussed to improve the segmentation. The colour classifier is using colour histogram of quantised 8x8 blocks in YUV colour space. The block histogram is then classified as foreground or background by comparing the blocks against foreground and background buffer histograms. It is concluded in the dissertation that this approach is computationally expensive, but the combination would get more accurate information of moving objects.

As described in the previous section, motion detection can be used to detect moving objects and then separate the foreground from the background. To segment out the teacher who is moving in front of the blackboard, we need to estimate the teacher's motion, and later replace pixels with high motion with pixels from earlier frames with no motion. Even if the object is in the scene from the start we can remove it after some frames. We get this missing background information as soon as the object moves from the initial area it was covering. When using motion detection there is no need to know what colours the objects have, therefore it is possible to detect objects in front of every

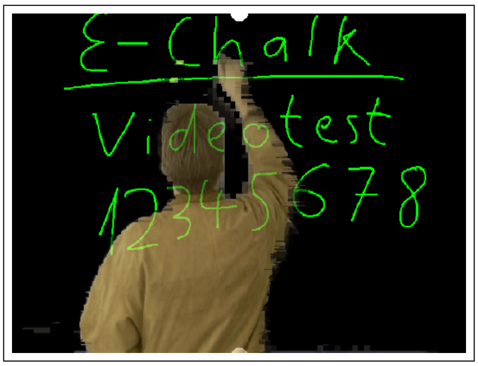


Figure 8: E-board content on top of the teacher [4].

writing board. An approach to solve the problem using motion detection when objects move slowly and get part og the background, could be by using foreground/background separation by effective combination of motion and colour segmentation modules to segment the foreground [15]. Motion is computed by computing the difference between consecutive frames in different regions, and if 85% of the pixels in a region are moving, then they conclude that there is a moving object in that region.

Frame difference is also used in the object contour detection approach by Wei et al. [5]. Each frame is divided into a mesh with square patches. Motion is estimated through frame difference and smaller patches are created in areas with more motion components. The smallest patches can be assumed to represent the moving object, as in figure 9. Since the patches can be polygonal or square, the moving objects boundary will not be accurately detected. But if the sizes of the patches can be changed then it might be possible to get a rough estimation of the teacher's body. Using motion detection might not cover the whole moving object though. If the teacher for instance has a t-shirt with a similar colour to the blackboard then it might be difficult to track motion in that region because there might be no or little changes in colours or intensity in the pixels for that region. Subsequently a threshold value is used on the results from the motion estimations, which creates a binary frame. Holes in this binary frame might be closed using morphological operations like dilation and erotion [1] [5].

Techniques used for motion estimation can be Sum of Absolute Difference (SAD) [23] and optical flow based on Lucas Kanade [24] [25] or Horn Schunck [26] [27] methods etc.

2.2 Meta-data extraction

There are different meta-data to extract from for instance a lecture video, and therefore different techniques has to be implemented. In this thesis we are going to look closer at meta-data which can be found on the blackboard, and actions that are done on the blackboard. To be able to do this it is better, and computational faster, to only consider the blackboard. Then the noise appearing outside the area of the blackboard will be removed. Noise might also appear on the blackboard, and this is also a problem that we

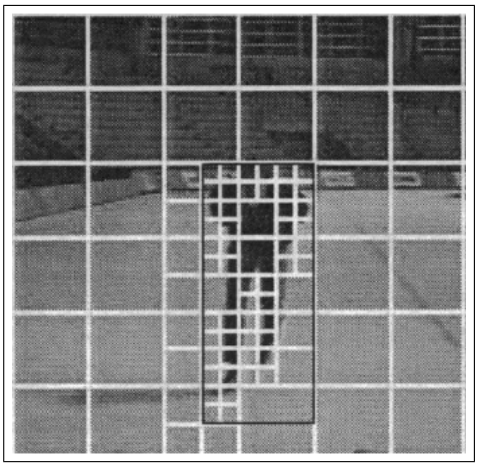
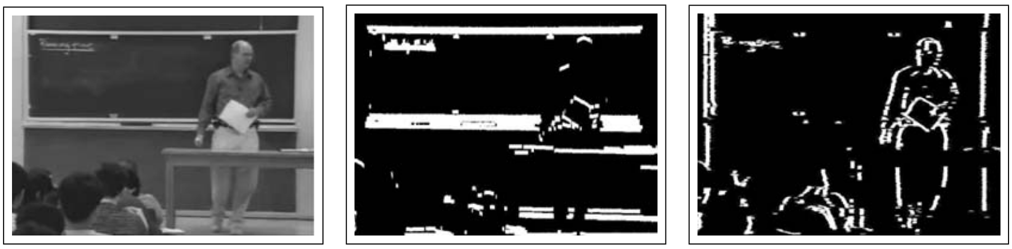


Figure 9: Small patches show where moving objects are [5].

have to overcome. In this thesis noise is defined as everything beside the content on the blackboard, and can come from the segmentation of the teacher, intensity changes, chalk dust and when moving objects, like the sponge. The text on the blackboard can also be enhanced. Further in this section we will seek to find methods to extract the blackboard and its content.

2.2.1 Board and content extraction

Mittal et al. proposed to use vertical and horizontal Canny edges to find the blackboard [3]. The longest vertical and horizontal edges are used to represent the extremities of the blackboard, figure 10 shows the detected edges. A problem with this approach is that there might be other edges, for instance from a lectern or a wall, that could be mixed up as the blackboard. Mittal et al. takes this into consideration and calculates the intensity value inside several rectangular regions for the found blackboard to verify that it has the value for the blackboard. If the intensity value does not match then the horizontal edge is discarded and the next largest edge is chosen. Another approach to detect the blackboard is to create a mask of the blackboards colour [1] [28]. By setting a low threshold value it is possible to overcome problems like videos having different lightning conditions, and also that there might be some changes in the intensity on different regions on the blackboard. Using a low threshold will also include more noise [1].



(a) Original (b) Horizontal edges (c) Vertical edges

Figure 10: Horizontal and vertical edge detection of the blackboard [3].

Another blackboard detection method is based on colour clustering and a refinement process [28]. The colour clustering is done by estimating the approximate mean colour of the blackboard. The refinement process refines the segmentation by compensating for high luminance variation caused by the chalk dust. Text is extracted using Canny edge detection and pixel luminance values. The mean luminance value of the edges is calculated and used as a threshold value to find text inside the blackboard's area.

Onishi et al. [6] proposed a method to extract blackboard content using edges from a spatiotemporal image, from which they segment two types of edges; dynamic (moving) and static (stationary) edges. The dynamic edges would be moving objects in front of the blackboard, while static edges would be text on the blackboard. Onishi et al. used the Sobel operator for horizontal and vertical edge extraction to get hold of the edges which holds the content on the blackboard, the static edges

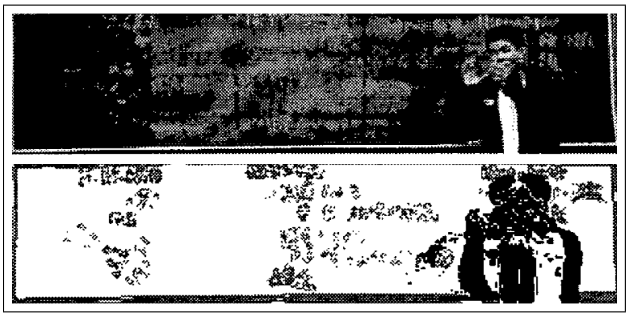


Figure 11: Dynamic and static edges [6].

Intensity variations caused by for instance the chalk dust might be a problem when extracting blacboard content. Liu et al. [29] proposed a method to overcome this. A colour distribution of the blackboard is modeled and used to extract content from the board. Each pixel is modeled as {r, g, b} = {r_1+n, g_1+n, b_1+n}, where {r_1, g_1, b_1} represents the colour of the blackboard, and n represents the colour of the chalk. n is modeled as a binomial distribution based on empirical data, and estimates the real blackboard colour. Since the colour of the blackboard is dependent on the lighting conditions a re-estimate of the blackboards colour has to be done for instance when turning off/on the lights. The average colour of blocks with less edge points than a predefines threshold are clustered. The blocks with less standard deviation of colour are less affected by chalk dust, and the block with least colour variance is selected to represent the background colour of the blackboard. Based on the model a block is categorized into one of three categories: blackboard background, content or irrelevant.

2.2.2 Content enhancement

When a lecture is recorded on video, the content on the blackboard may be difficult to read, and should therefore be enhanced. There are some techniques proposed in earlier works that will be covered in this subsection.

Eirik Grythe [1] has worked on how to enhance text from the blackboard. Contrast stretching is one method to make text differ more from the background as seen in figure 12. Also edge operators can improve text, but to eliminate noise a theshold value has to be set. If this value is set manually then it might be difficult to read parts of the text when other parts are good. Grythe also describes an approach to increase the readability of the text by accumulating frames in time. This might help since different frames contain slighty different information, and by accumulating these frames more of the text might be represented. Another method is to perform morphological operations on the text [1] [5] [30]. This could be using dilation and erosion with structure elements to better connect and visualize letters. An example that shows a more complete and connected object can be seen in figure 13.

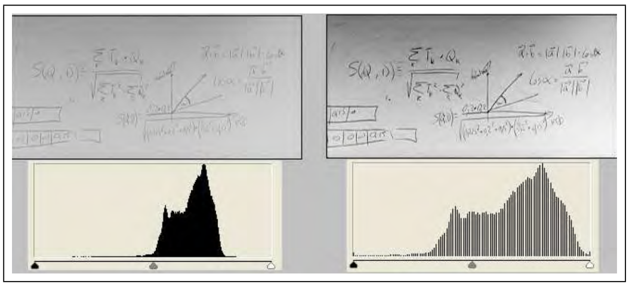
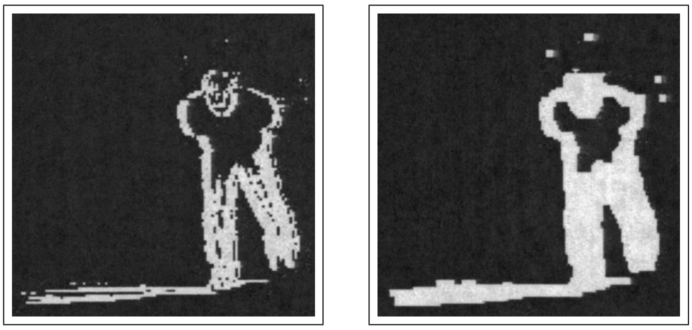


Figure 12: Contrast stretching to enhance blackboard content [1].



(a) Threshold output

(b) Morphological output

Figure 13: Morphological operation [5].

Also providing an empty background image will get better results [31]. Taking a frame with content and subtract it from the empty background image will leave us with the content. This method will not be affected by blackboards having different intensity values in different regions. But it is not always possible to get a clean background with no objects in the scene. Zooming closer to regions of interest can make it easier to see content, but that depends on the resolution of the frames. Zoomed frames might also get blurred and too much zooming is not good for the visibility [31].

2.3 Video indexing

There has been done some work in different aspects of lecture video indexing. Some work is related to creating an index based on text from PowerPoint slides of a lecture. Work has also been done in segmentation of teacher and blackboard content, as well as text recognition. In "Lecture Videos for E-learning: Current research and challenges" [8] the state of the art, as of 2004, is reviewed. We will discuss these and other proposed works in this section.

Creating an index for a video is a challenging task. One has to know what kind of indexes that are good to navigate through a video. Also finding where to extract index-points can be challenging. It would be good to find when the teacher adds text, erases an amount of text, is not writing or draws figures. These meta-data can be very helpful to use when creating an index of a lecture video.

Onishi et al. [6] proposed a method where they create boundaries around static edges, which is, as described earlier, the content. Each boundary, or written block, is extracted and stored in terms of the time being extracted, the position and number of static edges. With this information it is possible to arrange the content in a time hierarchy, and to link a spesific content to a time in the videofile, as shown in figure 14.

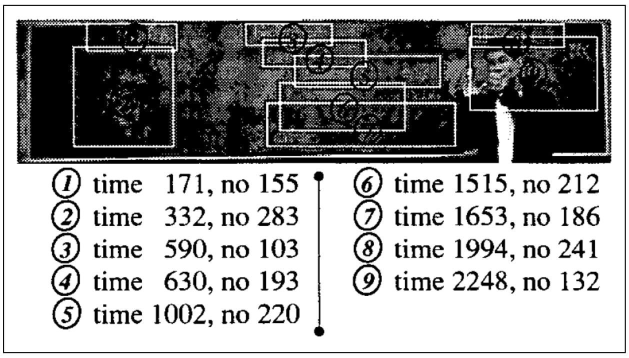
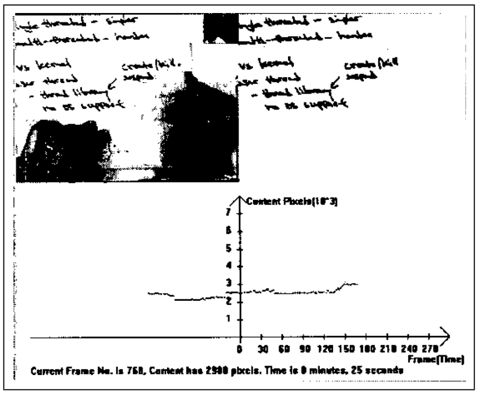


Figure 14: Content arranged in a time hierarchy [6].

Another approach using slides is to select a representation frame of each shot [18]. Between shots there are T seconds of a blank slide. Comparing pixel difference histogram makes it easy to detect such blank slides. In a lecture video using blackboard it is difficult to find shots. One shot might be found when taking the difference of a full blackboard and an empty blackboard. This could be implemented as finding changes that are bigger than a certain threshold using image differences between frames separated in time [32].

Key frame extraction can be used to extract frames that satisfy some rules [29] [7]. Such rules can be that a frame must be a clean frame, its content is not in previous key frames, and its content is clearly different from previous key frames [7]. A clean frame is a frame not occluded by moving objects. The number of ink pixels is used as a heuristic measure of the semantic content, as seen in figure 15. A key frame might also be selected based on the local content maxima [29]. Other meta-data extractions to use as indexes might be finding when the teacher is talking or writing. This could be done searching for the teacher's skin tone, and see if he is facing the audience or blackboard [29]. This might be problematic since people can have quite different skin tones.



Current Frame No. is 768, Content has 2900 pixels. Time is 0 minutes, 25 seconds

Figure 15: Segmented frame with number of inc pixels [7].

Another key frame extraction method is done by clustering of visual content [33]. Visual content is a colour histogram of a frame, and the frames are clustered based on similarity of their histograms. After clustering only the clusters that are big enough are considered as key clusters. For each key cluster, the frame which is closest to the cluster centroid is selected as the key frame.