# Visual Lecture Summary Using Intensity Correlation Coefficient

Solomon E. Garber, Luka Milekic, Nick Moran, Aaditya Prakash, Antonella Di Lillo and James A. Storer

*Department of Computer Science, Brandeis University, Waltham, Massachusetts*

**Abstract**

We present an automatic technique for creating a video summary of chalkboard and whiteboard lectures with the speaker removed, and for generating a set of slides containing all of the written content but without the lecturer present. Our system works by continuously subtracting the lecturer from the video feed using a region based correlation feature. The final presentation slides are extracted from this new version of the video that no longer contains the speaker.

**Keywords:** Lecture Recording, Video Summarization, Video Indexing, Foreground Subtraction

## 1 Introduction

Online resources are increasingly used by students and educators. Pre-recorded videos of lectures containing handwritten mathematical content are a popular educational tool, and traditional whiteboards are still the preferred format for displaying this content in the classroom [Vemulapalli and Hayes, 2014]. However, these videos can be cumbersome to navigate. Thumbnails, video titles, and high speed video scrubbing fail to offer a useful summary, so locating specific content can be time consuming. There is a need for tools which aid in the organization of this content. Specialized hardware such as electronic whiteboards are expensive and require custom software, and do not address the navigability of the many videos already available online. We present an automatic technique for creating a video summary of a chalkboard (or whiteboard) lecture with the foreground removed (e.g., remove the professor who is lecturing using the chalkboard), and for generating candidate key frames for a slideshow lecture summary.

Some attempts have been made to make educational videos easier to navigate. In [Kannan and Andres, 2010], a system creates a database of tags to search screen captured videos, but does not provide for a more easily searchable visual representation. [Yang et al., 2012a] presents a method for extracting slides from such videos, and use OCR to allow text based search on the slides. [Yang et al., 2011], [Yang et al., 2012b] and [Tuna et al., 2015] use OCR to make video lectures searchable by content, while [Vemulapalli and Hayes, 2014] and [Yang et al., 2014] combine OCR with speech recognition for the same purpose. [Yadid and Yahav, 2016] uses OCR to extract code from programming instruction videos. [Pratusevich, 2015] and [Liao et al., 2015] present methods of localizing written content and lecturer within lecture videos. [Wang et al., 2003] and [Eberts et al., 2015] align powerpoint slides with lecture videos. [Shin et al., 2015] and [Monserrat et al., 2013] generate lecture notes from computer screen captures; they refer to such screen captures as "blackboard style videos". In contrast here we address full length videos of a speaker lecturing in front of an actual blackboard, where there is noise in the video capture process, changes in ambient lighting, and occlusion of the blackboard by the lecturer. [Lin et al., 2004] presents a method for segmenting lectures by topic, given a transcript. In [Prabhu et al., 2008] and [He and Zhang, 2007], a method for removing a lecturer from a whiteboard lecture video is presented, but there is no attempt to generate a set of slides from the output. In [He et al., 2003], slides from whiteboard videos are generated, but without quantitative evaluation. [Choudary and Liu, 2007] produces bitmask slides from a chalkboard lecture video; these bitmasks do not contain color information, and evaluation is done by hand.

 ISBN 978-0-9934207-2-6

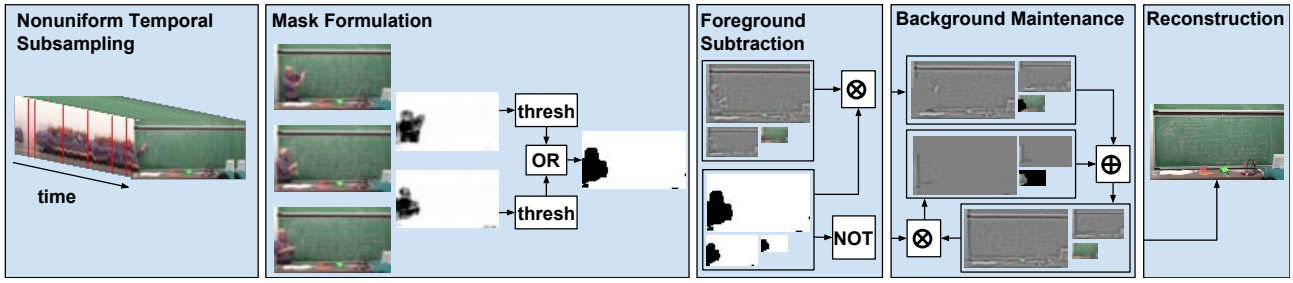


Figure 1: Our system.

Some attempts have been made to create continuous video summaries. In [Bennett and McMillan, 2007] a method is proposed of generating time-lapse videos from continuous feeds which summarize the changes between frames, as well as a method using non-uniform sampling in time to capture all important changes. Projects like Microsoft hyper-lapse [Kopf et al., 2014] warp videos in time to create a smoothly accelerated version of the video. Some attempts such as [Ejaz et al., 2012] and [Mundur et al., 2006] select key frames from the video to create a static video summary. In [Gianluigi and Raimondo, 2006] frame differences in a video feed are processed to select key frames for video summary. In [Mehmood et al., 2014] and [Ngo et al., 2003] attention models are used to predict the salient parts. These various summarization methods seek to capture the changes that occur in a video. Background is generally assumed to be uninteresting in these techniques, as it accounts for a very small portion of the changes in the videos. The majority of changes (e.g. speaker movement) from frame to frame in our videos have nothing to do with the content of the blackboard. We identify key frames in a video that we have modified to remove the speaker.

Much work has also been done on the problem of detecting and deleting the background from footage for applications such as video surveillance and object tracking. However, little has been written on the subject of removing the foreground and summarizing the background. In [Rubinstein et al., 2011] time-lapse videos are denoised by removing high speed changes and small jittery motions, under the assumption that the information in such videos is concentrated in low frequency changes. Much like time-lapse videos, most of the interesting content of chalkboard lecture videos is contained in gradual changes to the background image. Here we present an automatic technique for creating a video summary of the chalkboard lecture with the foreground (the speaker) removed, and for generating key frames for a slide show lecture summary.

## 2 Proposed Method

We process the input video in multiple stages, shown in Fig. 1. In the sampling stage described in Sec. 3, we take a nonuniform sampling from the input feed based on absolute frame difference. Next, in the foreground subtraction stage described in Sec. 4 and 5, we calculate the localized correlation coefficient computed over a sliding window, and threshold the results to obtain a foreground mask. We use the mask to update coefficients of a background image pyramid as explained in [Burt and Adelson, 1983] and described in Sec. 6. This pyramid is then reconstructed and added to the output stream. We then use a combination of edge detection and localized correlation to detect when a portion of the board has been erased or written over in the key frame detection stage described in Sec. 7. We describe our evaluation metrics and results in Sec. 8.

## 3 Nonuniform Temporal Subsampling

The changes that we wish to preserve are persistent, and failure to detect all foreground regions can lead to artifacts in the reconstructed background image. For this reason, a temporally subsampled version of the original feed is used. This sampling is done non-uniformly in time as a pre-processing step to prevent correlations created by temporarily static foreground regions. In order to avoid costly comparisons between every pair of

 ISBN 978-0-9934207-2-6

frames, we use a greedy heuristic approach to sampling in time. We perform two passes over the video. In the first pass we find the mean absolute difference between consecutive frames in a subsampled feed (for experiments reported here, approximately 1 frame for every 3 seconds of video). In the second pass, we sample from the original video by taking the first frame, and then only sample subsequent frames if the sum of the absolute differences between the current frame and the previously sampled frame is greater than the mean difference from the first pass. Because the majority of intensity and color changes are due to foreground motion, this non-uniform feed ensures that adjacent samples are uncorrelated in regions of motion.

## 4 Foreground Subtraction

Background subtraction is a common first step in many object tracking and security applications. Typically foreground masks are generated based on some pixelwise distance from a background model, or simply the pixelwise distance between consecutive frames in the input video.

Background subtraction approaches assume that the foreground is the interesting part of the video. However, as in [Rubinstein et al., 2011], we consider the foreground as noise and the background as signal, and seek a method which can preserve medium term changes in the background while deleting all foreground objects, such as the lecturer or students temporarily occluding the blackboard. Unlike [Rubinstein et al., 2011], however, we assume that the relevant background regions will not move between frames because both the camera and blackboard are assumed to be stationary. This allows us to avoid the costly message passing scheme used in [Rubinstein et al., 2011] to compute spatiotemporal displacement fields in favor of temporal displacements which can be computed using foreground masks. We therefore model the first stage of our process as foreground subtraction. Similar to background subtraction applications, the output of the foreground subtraction stage is a compact and easily scanned representation of the input video.

## 5 Mask Formulation

We model the background as regions where the shape of the intensity surface doesn't change from frame to frame in the subsampled feed. We detect such changes in shape using the regional cross correlation between input frames. We process the input video sequentially, obtaining a mask for every pair of adjacent frames. We treat each $x, y, t$ pixel in the input video $I$ as a sample from a population. At each time $t$, at each pixel $p = I(x, y, t)$, we compute the correlation coefficient for the intensity values over the region

$$\mathbf{N}(x, y) = \{(x_0, y_0) | x-\delta \leq x_0 \leq x+\delta, y-\delta \leq y_0 \leq y+\delta\} \quad (1)$$

between the frame at $t$ and $t-1$, and $t$, $t+1$, and threshold those correlation coefficients. Let $\mu_{x,y,t}$ denote the average intensity of the frame at time $t$ over the region $\mathbf{N}(x, y)$, and $Var(x, y, t)$ denote the variance of the intensity over the same region. We compute the local intensity correlation at each point $I_{x,y,t}$ as:

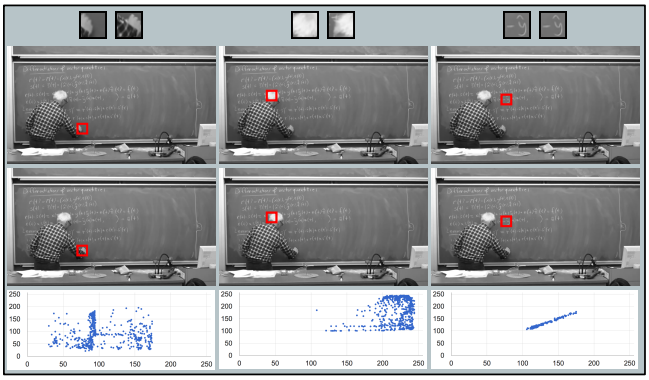

Figure 2: Three square sections taken from consecutive images in the temporally subsampled lecture. The first two blocks contain motion and therefore have weak and even negative correlations. The third does not move and thus the two blocks are strongly correlated.

$$\frac{\frac{1}{(2\delta)^2}\sum_{\mathbf{N}(x,y)}(I_{x_0,y_0,t} * I_{x_0,y_0,t+1}) - \mu_{x,y,t} * \mu_{x,y,t+1}}{\sqrt{Var(x, y, t)} * \sqrt{Var(x, y, t+1)}} \quad (2)$$

$$Var(x, y, t) = \frac{1}{(2\delta)^2} \sum_{\mathbf{N}(x,y,t)} I^2_{x_0,y_0,t} - \mu^2_{x,y,t} \quad (3)$$

 ISBN 978-0-9934207-2-6

This blockwise comparison is shown in Fig. 2 and an example of a surface obtained in this manner is depicted in Fig. 1. This metric (similar to the spatially modulated normalized vector distance used in [Matsuyama, 2000]) is sensitive to the size of the sampling window. However, as can be seen from equations 2 and 3, the computations can be done with a uniform blur kernel, which is separable and can therefore be computed in time proportional to the width of the window size $\delta$ rather than the square of the width as would be required in the naive implementation. This coupled with the fact that computations are entirely localized and can thus be massively parallelized, allows us to do the background maintenance quickly even with large spatial support for the correlation computation (we used a window size of $\delta = 12$ pixels). As a post-processing step, we apply a morphological erosion of the mask to close up holes in flat foreground regions and remove shadows.

## 6 Background Reconstruction

In most background subtraction applications, a statistical model of local features is kept for each pixel to allow for change detection. This model must be constantly maintained to accommodate dynamic backgrounds. In this sense our problem can be seen as continuous background maintenance and reconstruction. Simple low pass filters such as FIR (finite impulse response) and IIR (infinite impulse response) filters, commonly used for the task of background maintenance, tend to create ghosts in the background given any reasonable filter support. If the support is too large, important background information can take too long to appear in the smooth video. Also, since the speaker is almost always somewhere in the shot and often stays in a similar place for long amounts of time, even a median filter will output entire regions of misclassified pixels. For this reason we classify each frame into still regions and motion regions and use masks to block out any the locations in the frame where motion is detected. We update the background only in regions without motion, under the assumption that the foreground is never stationary. If the foreground does not move then large parts of the foreground will be incorporated into the estimated background image. To address this issue we process a subsampled version of the input video as described in Sec. 3. To initialize the background, we run this process without writing to the output stream until the entire frame has been updated and then start over from the beginning. In order to prevent edge artifacts from appearing at mask boundaries we store the background as a Laplacian pyramid, a tensor containing edges at different resolutions which can be used to reconstruct a full resolution image. We create a Gaussian pyramid from each mask, and use the mask pyramid to determine which coefficients to update in the background pyramid. This suppresses edge artifacts at the boundaries of the mask, especially due to aliasing introduced in the temporal subsampling process caused by light and shadow changes. In the last phase of processing, we use the edges in each frame to locate the key frames, so mask induced edge artifacts would be both distracting for the end user and detrimental to our process.

## 7 Slide Selection and Erasure Detection

Once the background video has been obtained we can create a useful and compact summary of the lecture by finding the frames in the background video that contain a blackboard full of writing, prior to some major erasure event. Our goal is to save all the slides directly before part of the board is erased. We use the intensity correlation described in Sec. 5 between consecutive frames in the background video to detect when the board is erased. When corresponding regions in consecutive frames fail to correlate the possible causes are information added, erased, or modified. We apply a Sobel filter to the first image in the pair and find the sum of edges in the changed regions. When the unmatched edges surpass a low threshold, we determine that something was erased. We save the first image in the pair, and sum the edges in the unmatched regions of the second image in the pair to start a running tally. Detected erasures will not trigger another slide to be saved unless the sum of the edges added since the last saved slide exceeds a lower limit, indicating that the lecturer has resumed writing on the board. Although this method successfully saves the slides we want, it can include extra slides containing no unique information. An example of such a slide is shown in Fig. 4 which was chosen by our algorithm because of the way we initialize the background. We do a final pass over the set of slides, automatically removing slides

 ISBN 978-0-9934207-2-6

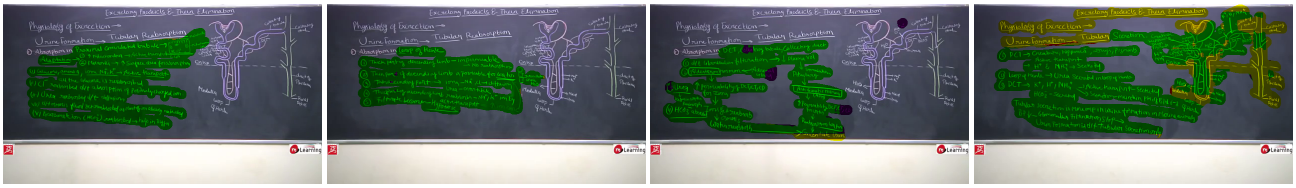

Figure 3: Ground truth slides from Fig. 4 colored based on the redundancy in our output. Best viewed in color.

whose edges can all be matched elsewhere in the set. The extra slide in Fig. 4 is not currently removed by this final pass because part of the diagram on the right hand side was written over and failed to match the similar regions in subsequent slides, a problem which we are addressing in current research.

## 8 Evaluation of Results

In this section we present a ground truth against which results may measured, together with a baseline method to which we compare our improved results. In order to evaluate our results, we selected recorded lecture videos posted to YouTube from a variety of instructors and institutions where handwritten material was presented on a whiteboard or blackboard, and where both the camera and the board remained in a fixed position over the course of the lecture. Ground truth slides were generated by hand (where a human watched each video in the test set), and every time the lecturer erased something in order to make room on the board, several frames from the video prior to the erasure were sampled and the regions of these frames where the background is visible were blended together (using the pyramid blending described in Sec. 6). The left column of Fig. 4 is an example of a ground truth slide set. Ground truth masks were constructed by marking the important regions of each ground truth slide (anything written on the board is behind just one of the masks). Fig. 3 shows how these masks can be used to evaluate a proposed slide set, where each slide in the ground truth slide set has been colored according to the number of times it appeared in our proposed slideshow in comparison with the ground truth, where purple regions were not matched to any slide, green regions occur at least once but no more than the ground truth, yellow regions one or two more times than the ground truth, and red regions more than two times more than the ground truth. The baseline takes a median filter over 50 frames sampled at 2 frames every 3 seconds, where a slide is taken when the sum of the edges from a Canny edge detector is a local maximum. Generated slideshows by both the baseline and our improved method are judged based on two metrics, $\mathscr{R}$ = the ratio of the size of a proposed slide set to the ground truth, and $\mathscr{P}$ = the percent of masked pixels with no match in the proposed

Table 1: Our results compared with the baseline described in Section 8. GT: Ground Truth. $\mathscr{R}$ indicates the number of slides generated by each method divided by the number of slides in our ground truth minimal slide set. $\mathscr{P}$ indicates the percent of masked pixels from the ground truth not matched to any slide in a generated slide set. # Slides indicates the number of slides chosen in our ground truth slide set. The baseline method against which we compare our results is described in Section 8, as well as the choice of videos.

| | **Lecture** | $\mathscr{R}$ | | $\mathscr{P}$ | | **# Slides** |
|---|---|---|---|---|---|---|
| | id | ours | baseline | ours | baseline | GT |
| Whiteboard | 1 | 1.15 | 0.42 | 4.6 | 34.6 | 26 |
| | 2 | 1.09 | 1.00 | 2.7 | 5.6 | 11 |
| | 10 | 3.50 | 2.70 | 2.5 | 4.6 | 10 |
| | 12 | 2.89 | 2.00 | 3.6 | 6.7 | 9 |
| | 15 | 1.46 | 0.62 | 2.4 | 25.4 | 13 |
| Blackboard | 3 | 1.25 | 1.50 | 1.0 | 4.4 | 4 |
| | 4 | 1.60 | 1.40 | 0.0 | 1.6 | 5 |
| | 5 | 2.25 | 1.50 | 1.2 | 5.6 | 4 |
| | 6 | 1.00 | 0.77 | 7.9 | 24.4 | 9 |
| | 7 | 1.33 | 0.72 | 5.1 | 27.4 | 18 |
| | 8 | 1.60 | 1.20 | 0.2 | 9.4 | 15 |
| | 9 | 1.22 | 0.77 | 2.5 | 27.3 | 18 |
| | 11 | 1.33 | 1.67 | 1.0 | 4.2 | 3 |
| | 13 | 1.4 | 1.00 | 3.4 | 25.3 | 5 |
| | 14 | 1.33 | 1.00 | 0.5 | 12.9 | 3 |
| | 16 | 1.17 | 0.83 | 1.0 | 24.4 | 6 |

 ISBN 978-0-9934207-2-6

Figure 4: A sample input frame (top left), ground truth slide set (left) and corresponding slides produced by our system (right). Our system produced an extraneous slide for this video, which did not correspond to any slide from the ground truth (top right). Best viewed in color.

 ISBN 978-0-9934207-2-6

slide set. Our results are reported in Table 1. In all videos tested, our method selected a set of slides containing over 90%, and in almost all cases over 95% of the pixels flagged in the ground truth as important. Our algorithm prioritizes completeness of the selected slide set over brevity, and yet the generated slide sets tended to stay within 50% of the minimal number of slides, rarely as much as double and in only one case did our slide set exceed triple the minimal number of slides from the human generated ground truth.

## References

[Bennett and McMillan, 2007] Bennett, E. P. and McMillan, L. (2007). Computational time-lapse video. In *ACM Transactions on Graphics (TOG)*, volume 26, page 102.

[Burt and Adelson, 1983] Burt, P. J. and Adelson, E. H. (1983). A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)*, 2(4):217–236.

[Choudary and Liu, 2007] Choudary, C. and Liu, T. (2007). Summarization of visual content in instructional videos. *IEEE Transactions on Multimedia*, 9(7):1443–1455.

[Eberts et al., 2015] Eberts, M., Ulges, A., and Schwanecke, U. (2015). Amigo-automatic indexing of lecture footage. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1206–1210. IEEE.

[Ejaz et al., 2012] Ejaz, N., Tariq, T. B., and Baik, S. W. (2012). Adaptive key frame extraction for video summarization using an aggregation mechanism. *Journal of Visual Communication and Image Representation*, 23(7):1031–1040.

[Gianluigi and Raimondo, 2006] Gianluigi, C. and Raimondo, S. (2006). An innovative algorithm for key frame extraction in video summarization. *Journal of Real-Time Image Processing*, 1(1):69–88.

[He et al., 2003] He, L.-w., Liu, Z., and Zhang, Z. (2003). Why take notes? Use the whiteboard capture system. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 5, pages V–776. IEEE.

[He and Zhang, 2007] He, L.-W. and Zhang, Z. (2007). Real-time whiteboard capture and processing using a video camera for remote collaboration. *IEEE Transactions on Multimedia*, 9(1):198–206.

[Kannan and Andres, 2010] Kannan, R. and Andres, F. (2010). Towards automated lecture capture, navigation and delivery system for web lecture on demand. *International Journal of Innovation in Education*, 1(2):204–212.

[Kopf et al., 2014] Kopf, J., Cohen, M. F., and Szeliski, R. (2014). First-person hyper-lapse videos. *ACM Transactions on Graphics (TOG)*, 33(4):78.

[Liao et al., 2015] Liao, H.-C., Pan, M.-H., Chang, M.-C., Lin, K.-W., et al. (2015). An automatic lecture recording system using pan-tilt-zoom camera to track lecturer and handwritten data. *International Journal of Applied Science and Engineering (IJASE) 13 (1)*, pages 1–18.

[Lin et al., 2004] Lin, M., Nunamaker, J. F., Chau, M., and 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*, pages 9–pp. IEEE.

[Matsuyama, 2000] Matsuyama, T. (2000). Background subtraction for non-stationary scenes. In *Proc. 4th Asian Conference on Computer Vision, 2000*, pages 662–667.

[Mehmood et al., 2014] Mehmood, I., Sajjad, M., and Baik, S. W. (2014). Visual attention based extraction of semantic keyframes. *Advances in Information Science and Applications*, 1.

[Monserrat et al., 2013] Monserrat, T.-J. K. P., Zhao, S., McGee, K., and Pandey, A. V. (2013). Notevideo: facilitating navigation of blackboard-style lecture videos. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1139–1148. ACM.

[Mundur et al., 2006] Mundur, P., Rao, Y., and Yesha, Y. (2006). Keyframe-based video summarization using delaunay clustering. *International Journal on Digital Libraries*, 6(2):219–232.

[Ngo et al., 2003] Ngo, C., Ma, Y., and Zhang, H. (2003). Automatic video summarization by graph modeling. In *Computer Vision, 2003. Proc. 9th IEEE International Conference on*, pages 104–109.

[Prabhu et al., 2008] Prabhu, N., Kumar, R. P., Punitha, T., and Srinivasan, R. (2008). Whiteboard documentation through foreground object detection and stroke classification. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 336–340. IEEE.

[Pratusevich, 2015] Pratusevich, M. (2015). *Edvidparse: Detecting people and content in educational videos*. PhD thesis, Massachusetts Institute of Technology.

[Rubinstein et al., 2011] Rubinstein, M., Liu, C., Sand, P., Durand, F., and Freeman, W. T. (2011). Motion denoising with application to time-lapse photography. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 313–320. IEEE.

[Shin et al., 2015] Shin, H. V., Berthouzoz, F., Li, W., and Durand, F. (2015). Visual transcripts: lecture notes from blackboard-style lecture videos. *ACM Transactions on Graphics (TOG)*, 34(6):240.

[Tuna et al., 2015] Tuna, T., Joshi, M., Varghese, V., Deshpande, R., Subhlok, J., and Verma, R. (2015). Topic based segmentation of classroom videos. In *Frontiers in Education Conference (FIE), 2015. 32614 2015. IEEE*, pages 1–9. IEEE.

[Vemulapalli and Hayes, 2014] Vemulapalli, S. and Hayes, M. (2014). Audio-video based character recognition for handwritten mathematical content in classroom videos. *Integrated Computer-Aided Engineering*, 21(3):219–234.

[Wang et al., 2003] Wang, F., Ngo, C.-W., and Pong, T.-C. (2003). Synchronization of lecture videos and electronic slides by video text analysis. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 315–318. ACM.

[Yadid and Yahav, 2016] Yadid, S. and Yahav, E. (2016). Extracting code from programming tutorial videos. In *Proceedings of the 2016 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, Onward! 2016, pages 98–111.

[Yang et al., 2012a] Yang, H., Gruenewald, F., and Meinel, C. (2012a). Automated extraction of lecture outlines from lecture videos-a hybrid solution for lecture video indexing. In *CSEDU (1)*.

[Yang et al., 2012b] Yang, H., Oehlke, C., and Meinel, C. (2012b). An automated analysis and indexing framework for lecture video portal. In *International Conference on Web-Based Learning*, pages 285–294. Springer.

[Yang et al., 2014] Yang, H., Quehl, B., and Sack, H. (2014). A framework for improved video text detection and recognition. *Multimedia Tools and Applications*, 69(1):217–245.

[Yang et al., 2011] Yang, H., Siebert, M., Luhne, P., Sack, H., and Meinel, C. (2011). Lecture video indexing and analysis using video ocr technology. In *Signal-Image Technology and Internet-Based Systems (SITIS), 2011 Seventh International Conference on*, pages 54–61. IEEE.