Whiteboard Video Summarization via Spatio-Temporal Conflict Minimization

Kenny Davila
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623
Email: kxd7282@rit.edu

Richard Zanibbi
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623
Email: rlaz@cs.rit.edu

***Abstract*—Lecture videos are a valuable resource for students, and thanks to online sources they have become widely available. The ability to find videos based on their content could make them even more useful. Methods for automatic extraction of this content reduce the amount of manual effort required to make indexing and retrieval of such videos possible. We present a method that generates static image summaries of handwritten whiteboard content from lecture videos recorded with still cameras. We generate a spatio-temporal index for the handwritten content in the video, and we use it for temporal segmentation by detecting and removing conflicts between content regions. Resulting segments are used to produce key-frame based summaries. Our method has been tested on a video collection showing promising results for automatic lecture video summarization with good compression ratios and 96.28% recall of connected components in test videos.**

I. Introduction

Many lecture recordings exist online, providing a useful resource for many students. Consider the case of a linear algebra student who wants to find a particular lecture portion where the professor explains identity matrices. This portion might be just 5 minutes long within a one hour long video. In this case, it might be faster to manually find the portion in a short summary than in the raw video.

Lecture video summaries can be useful tools for video navigation, and also for automated indexing and retrieval of lectures. In multiple fields of study, like mathematics, where explanations are usually given using handwritten content on the whiteboard or chalkboard, producing these summaries manually is time consuming and requires detailed handwritten content annotations. However, many lecture videos in these fields are only annotated using high level tags describing the topics covered. Sometimes, audio transcripts are available but these do not always describe the entire whiteboard/blackboard contents. If specialized hardware for trace capture (e.g. an interactive whiteboard) is available at recording time, then the traces could be recorded and online symbol recognition methods can be use to recognize them. However, in the absence of human annotations and specialized hardware, particularly for existing collections of lecture recordings, we require robust automated methods for extraction and summarization of whiteboard contents from the video itself.

In this paper, we focus on providing a method for extraction and summarization of handwritten whiteboard content. Existing methods for lecture video summarization typically focus on local detection of changes in whiteboard/blackboard/slide content [1], [2]. Such changes are associated with slide transitions or whiteboard/blackboard erasing events. These approaches rely on detecting sharp content changes, and if the change is not sharp enough, a transition/erasing event will not be detected, resulting in under-segmentation. In videos using whiteboard or blackboard these changes can happen gradually and can be missed if small detection windows are used.

We propose a divide-and-conquer segmentation method that starts by analyzing conflicting regions of content at a global scale, and recursively splits the video into units that contain few or no content conflicts. Two connected components are in conflict if there is an overlap in the space they occupy on the whiteboard but they exist during different time intervals. This happens when the whiteboard gets erased and something new is written on the same region. The proposed segmentation allows the summarization of the video using a small set of key-frames. A single key-frame might combine portions of content that never co-existed on the whiteboard, but that occupied different spaces and can be displayed on a single image for better compression rates, producing shorter summaries. For example, a 45 minute-long video might be summarized in just 10 frames.

In this paper we explore the following research questions: **Q1.** How can we reliably extract whiteboard contents from lecture videos? **Q2.** How can we produce a static summary for a lecture video using a minimal number of frames that contains all handwritten content of the lecture?

Contributions. We propose a novel method for lecture video summarization based on minimization of conflicts between content regions. Second, a spatio-temporal index that can be used to navigate lecture videos based on the handwritten content on the whiteboard. Finally, given the lack of labeled data for this particular domain, we provide a small dataset of labeled lecture videos that can be used for training and testing of newer approaches for this and related problems. This dataset and the tools for ground truth generation are publicly available[1].

[1] https://cs.rit.edu/~dprl/Software.html

II. Related Work

Video summaries can provide quick overviews facilitating user navigation, indexing and retrieval of videos. The survey by Hu et al. [3] provides details about approaches for indexing and retrieval of videos. Detecting handwritten content in video is a special case of text detection in imagery covered in Ye and Doermann [4]. Approaches for text detection, tracking and recognition in videos are covered by Yin et al. [5].

Our work is video summarization within the domain of videos containing handwritten whiteboard content. These videos typically represent a single scene with one shot, but some might contain multiple shots when the focus shifts from the whiteboard to another object or person in the classroom.

Key-frame extraction. A simple summarization approach is to compute video key-frames and choose the most representative of the entire video. A good key-frame set has little redundancy and good content coverage [3]. Traditional key-frame extraction techniques are based on the analysis of different types of features like: color histograms, edges, shapes, optical flow, and others [3]. Some approaches for key-frame selection use global comparison between frames to distribute the key-frames by minimizing an objective function that can be application dependent [3]. For example, the work by Li et al. [6] uses minimization of frame reconstruction distortion to select key-frames for video summaries. Our proposed approach falls within this particular type of approach. A survey on key-frame extraction methods can be found in Sujatha and Mudenagudi [7].

Video summarization. There are two types of summaries [3]: static video abstracts and dynamic video skims. The first type are small key-frame sets that can be used to create tables of contents, storyboards and pictorial summaries [3]. Generated visualizations can summarize dynamic events like Nguyen et al. [8], where 3D views summarize video segments. Key-frames help to navigate the video in a non-linear way, but most of the dynamics of the content and the audio are lost. Video skims use short video segments to create summaries that may be more appealing to users, and can keep relevant audio portions. These two summary types can be combined to create hierarchical summaries [3], where high-level key-frames can be associated with low-level short video segments. Additional video summarization techniques can be found in Truong and Venkatesh [9] and Money and Agius [10].

Evaluation. Ideal video summaries can be subjective and often quality measurements are application dependent [3]. Typical metrics include recall of application-dependent targets extracted in the summary, and video compression ratio where we want to use the smallest possible set of key-frames. Higher recall usually means lower compression ratio by requiring more frames to extract all targets. Choudary et al. [1] uses a recall-based error metric for evaluation of lecture video summaries by identifying missing content by counting connected components for words and lines for graphics. In Chang et al. [11], the authors proposed a key-frame fidelity measure that is based on a semi-Hausdorff distance.

In some cases, video summarization approaches require video segmentation, and measure the quality of chosen split points compared to ideal video segments. This is common in slide-based lecture videos. In the work by Li et al. [2], the evaluation is made in terms of the precision, recall and F-score of the slide transitions detected. In the AMIGO system [12], the Jaccard index is used to evaluate the correctly detected slide transitions within a ± 3 second error margin.

A. Lecture Video Summarization

We focus on three elements of lecture video summarization methods and similar applications: binarization/segmentation, content extraction and summarization. Content extraction refers to techniques used to analyze and separate high level units of whiteboard/blackboard/slide content. We also consider two type of lecture video content: handwritten (from paper, whiteboard, etc), and typeset (slides). It is important to note that many approaches are multi-modal and also consider audio and supplementary lecture materials. Here, we concentrate on the image-based lecture video summarization approaches.

Binarization/segmentation. For videos or sets of images of handwritten content from whiteboards/chalkboards, traditional binarization methods like Otsu's [13] and Niblack's [14] have been employed in cases where illumination changes do not represent a major issue like in the whiteboard reading works by Wienecke et al. [15], Plötz et al. [16] and Vajda et al. [17]. Other thresholding-based techniques have also been employed in the works by Liu et al. [18], [19]. Some approaches use text detection and background removal techniques to isolate first the handwritten content regions before thresholding [20], [21], [22].

When the background is not very uniform and simple threshold-based methods fail, more sophisticated segmentation approaches are employed. Color space $L^*a^*b^*$ has been employed for mean-shift segmentation in the work by Comaniciu et al. [23], and k-means segmentation in the works by Liu and Choudary [24], [25], [1] and Lee et al. [26]. In the work by Davila et al., edge detection and morphological operations have been used for whiteboard image binarization [27].

Content Extraction. A common idea is to divide the input image using a grid [18], [19], [15], [20], [17], [27], and then classifying each cell as handwritten content, background or noise using statistical methods. Some approaches then group together handwritten content cells into blocks or text lines that can be used for summarization using OCR methods [15], [16], [17]. Temporal information and heuristic rules can be used to create these blocks by grouping changes as they happen [28]. Other methods employ local features to classify image patches as text or background. In the works by Tang and Kender [21], [22], SVMs are trained using features extracted from the edge image. In the work by Banerjee et al. [29], SIFT features [30] extracted from a dense grid are employed for patch-based classification of pixels as text/background using MLPs.

A common factor affecting the quality of the extracted content is low resolution. Some approaches like the work by Tang and Kender [22], [21] have employed super resolution

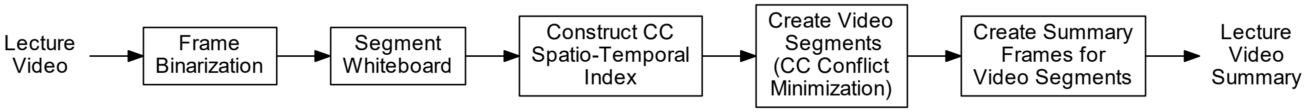

Lecture Video
Frame Binarization
Segment Whiteboard
Construct CC Spatio-Temporal Index
Create Video Segments (CC Conflict Minimization)
Create Summary Frames for Video Segments
Lecture Video Summary

Fig. 1. System Architecture.

techniques taking advantage of the temporal information available in the video domain. Another common factor affecting readability is poor contrast. Some methods employ contrast enhancing techniques to deal with this [16], [20], [17].

Most methods have taken advantage of the particularities of the lecture video domain. For example, explicit speaker modeling improves precision by avoiding extraction of occluded content [28], [1], [20], [31], [27], [32], [2], [26]. Erasing event detection in whiteboard/chalkboard videos is useful for video segmentation, and most whiteboard content will be on the image right before these erasing events happen. [1], [31], [27]. Some methods only extract content from frames with little motion assuming that this means no particular events like erasing or writing are taking place and that the speaker is probably not currently in the view of the camera [21], [22], [27], [33]. Also, not all methods need to detect the whiteboard, chalkboard or slide area in the image explicitly before doing content extraction, but some existing methods do it for different reasons including increased precision of content extraction [27] and the ability to correct camera view distortions for cleaner content extraction [2].

For slide-based videos, detecting transitions between slides is analogous to detecting erasing events in whiteboard/chalkboard based videos. OCR techniques along with supplementary materials [32], [12] can be used to accurately extract typeset text from the slide images. Sometimes videos can include shots where the slides are not visible, and detecting and removing these shots is helpful for clean content extraction. In the work by Adcock et al. [33], a SVM is trained for classification of slide/non-slide key-frames on videos. A full-optimization framework based on local feature tracking is used in the work by Li et al. [2] for detection of slide location and transition, and sharp changes in these features are used to detect and exclude segments with no visible slides.

Summarization. For whiteboard reading approaches, the final summary is given by text lines extracted and recognized text from the video [21], [22], [15], [16], [17], [31]. Region-based content extraction methods represent the video regions [28], [27]. Images coming from different camera views of the content can be mosaicked into single panoramic images [1], [26] or into large virtual slides [19].

Full key-frames can be used to summarize the video as well. Some methods try real-time selection of key-frames for lecture video summarization typically using rules based on motion and/or content change detection [18], [21], [22], [25], [1], [20], [26] where frames with low motion and large content changes from the previously selected key-frame are preferred. In particular, some methods try to identify peaks in a function that sums all chalk/ink pixels per frame for key-frame selection [1], [26]. For slide-based videos, images of the detected slides are also typically used as video summaries [2], [12]. In the work by Eberts et al. [12], local features are used for indexing of the graphic content embedded in the slides. Yadav et al. [34] uses C-NN to detect and index anchor elements (figures, tables, equations, proofs, etc) in images for indexing and summarization of slides.

III. Methodology

We propose a method (see Figure 1) that given a lecture video is able to produce a small set of frames containing the handwritten content from the whiteboard. As illustrated in Figure 2, our method is designed for videos recorded using a single still camera in a classroom. We assume that the whiteboard will be the largest object in the image surrounded by some background objects. Some moving elements like the instructor will be present on the video as well.

Frame Sampling and Binarization. We sample frames from the video at a sampling rate FPS (frames per second) to obtain a subset F. We have determined empirically that $FPS = 1$ is enough to capture relevant changes. Each selected frame is then preprocessed for background estimation and

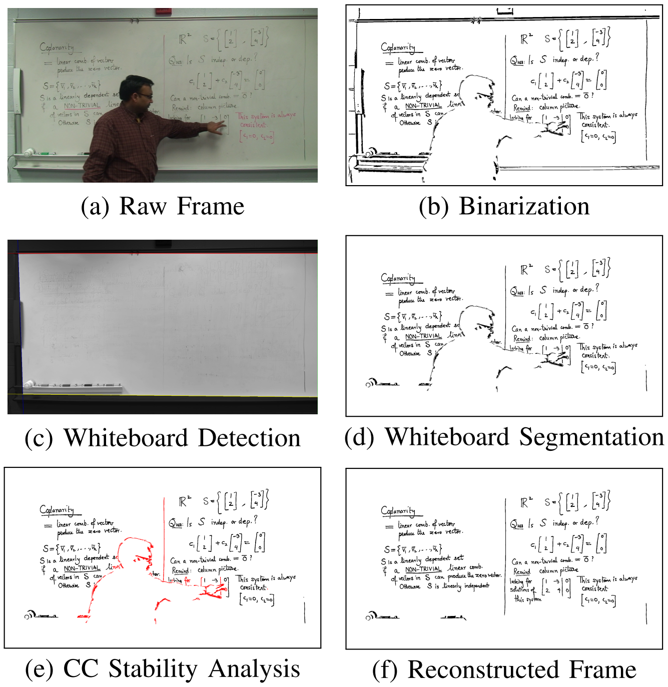
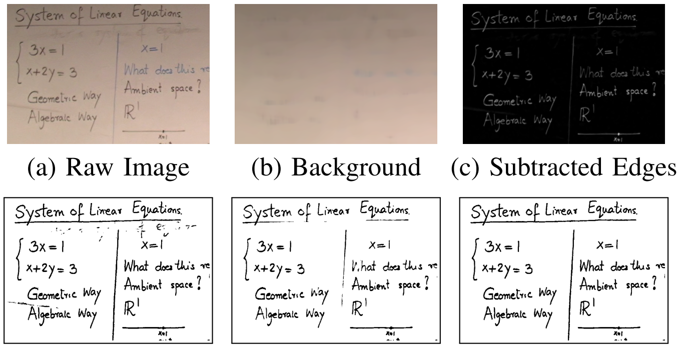

(a) Raw Frame
(b) Binarization
(c) Whiteboard Detection
(d) Whiteboard Segmentation
(e) CC Stability Analysis
(f) Reconstructed Frame

Fig. 2. Overview of our video summarization approach.



(a) Raw Image (b) Background (c) Subtracted Edges

(d) RF Binary (e) Otsu Binary (f) Hysteresis

Fig. 3. Overview of our binarization approach.

removal, and then binarized using a combination of two methods: a high-recall, low-precision machine learning binarizer, and low-recall, high-precision Otsu's [13] binarization. An example of this procedure is shown in Figure 3.

Global thresholding using methods like Otsu's [13] do not work well in gray scale images for this application. A single global threshold is unable to separate the whiteboard content from the background pixels due to non-uniform illumination on the board. Adaptive threshold methods do not work well because of many false positives in large empty regions of the whiteboard. In addition, dirty whiteboards and old markers produce traces that are hard to distinguish from background even for humans. To deal with these issues, we use a background subtraction method to generate edge frame images F^E as shown in Figure 3c.

For a given frame f, we first apply a bilateral filter [35] with Sigma color $B_{sc} = 13.5$ and sigma space $B_{sp} = 4.0$ for smoothing the whiteboard background while preserving the handwriting edges. Then, we estimate the background using a median blur filter with aperture size $B_{blur} = 33$. We then subtract estimated background from the smoothed image to obtain the difference on each RGB channel. The raw difference includes positive and negative values depending on which pixels are lighter or darker than the estimated background. Since we work with whiteboards, we only need the pixels darker than the background (negative values). We make all positive values equal to zero and then we change the sign of the negative values. We finally combine the differences on the three channels into a single edge image f^e by keeping the maximum value of all channels per pixel.

Note that for chalkboards/blackboards, we would simply need to make the negative values in the raw difference image equal to zero, and the rest of the binarization procedure would remain mostly unchanged.

We use a Random Forest classifier [36] for window-based pixel-level binarization because they produce reasonably accurate classification results in faster times than other machine learning techniques. We train it using patches of size $T_w \times T_w$ ($T_W = 7$) randomly sampled from fully labeled binary keyframes in training data. The goal is to learn the class of the pixel at the center of each window given contextual information in the edge space. We bias the patch sample by forcing a proportion of $T_{fg} = 50\%$ patches to be taken from foreground elements, where T_{fg} is typically higher than the actual proportion of foreground pixels on the whiteboard. Using object labels provided in the ground truth, we only sample patches from whiteboard pixels as we assume that general background and the speaker will be removed from binary images using other processes later. For the whiteboard background pixels sampled, we bias their distribution by assigning each pixel a probability proportional to their intensity in the edge frame space, adding 1 to all intensities to smooth probabilities for zero intensity pixels. In this way, the harder classification cases of the strong edges that are not handwriting or that are close to its boundaries will be well represented in our random sample. After sampling the training patches, we train our Random Forest classifier [36] using $T_{trees} = 16$ trees, maximum tree depth of $T_{depth} = 12$ and maximum number of features to consider at each split of $T_{feat} = 32$.

We finally binarize each frame edge image f^e using hysteresis, similar to Canny edge detection, by combining a weak (high recall, low precision) with a strong (low recall, high precision) binary image. The weak image is obtained using the window-based pixel-level Random Forest classifier described before (See Figure 3d). The strong image is obtained using Otsu's [13] binarization (See Figure 3e). The final binary image f^b is generated by overlapping the two images and keeping all connected components (CC) from the weak image that have at least one pixel overlap in the strong image.

We use a combined approach because the Random Forest generally produces easy to read binary traces and can even recover trace pixels that are hard to separate from the background in the original image (See Figure 3d), but it also produces false positives. On the other hand, Otsu's binarization [13] produces few false positives but many broken traces. The final binary keeps most handwriting CCs and will have most noisy CCs removed (See Figure 3f).

Whiteboard Segmentation This procedure estimates the whiteboard region and removes from the binary frames any CCs that are not fully contained in this region. (See Figure 2c and 2d). Similar to the optimization approach used by Li et al. [2], we estimate the whiteboard region using two high confidence estimates, one for handwriting pixels and the other for general background pixels, to choose the best quadrilateral from a set of candidates. We use a coarse sample of frames (one every 10 seconds) to estimate handwriting and general background locations based on two pixel-wise temporal statistics: median and standard deviation.

The pixel-wise temporal median image captures the temporal background of the video by keeping the most stable objects (usually background) while removing unstable elements like the speaker and most handwriting. These properties make this image ideal for estimation of the boundaries of the whiteboard region and locations of background pixels. We apply Canny edge detection (low threshold = 30, high threshold = 50) to this image to obtain a high confidence estimation of background pixels. Next, we use the Hough transform (radius resolution = 1 pixel, angular resolution = 1 degree, minimum line length = 100, maximum line gap = 10, minimum line intersections = 50) on the edge image to obtain candidate edges for the whiteboard region. We classify these edges based their angle and relative location as: top, bottom, left or right. We ignore diagonal lines that are more than 15 degrees away from the closest axis. To the list of candidate edges we add the boundaries of the image in case that no edge is detected on a particular direction.

The pixel-wise temporal standard deviation image captures what pixels change the most making it good for estimation of handwriting locations. We transform all sampled images to the same edge space used for binarization. We apply a temporal median blur filter using a window of size $C_{blur} = 11$ frames. This produces a set of edge frames where fast moving elements like the speaker are removed. From this set, we compute the final pixel-wise temporal standard deviation image. In this image, handwriting pixels tend to have the highest intensities. We threshold by setting a minimum intensity $C^{\perp}_{fg} = 5$, assuming that all pixels with intensity greater than or equal to $C^{\perp}_{fg}$ are very likely to be handwriting pixels.

Then we choose the quadrilateral whiteboard region using Hough transform line candidate edges and the estimates of handwriting and background pixels. We exhaustively evaluate all possible combinations of top, bottom, left and right edges to find a region that maximizes the C_{f1} criterion:

$$C_{f1} = \frac{2C^{rec}_{fg}C^{rec}_{bg}}{C^{rec}_{fg} + C^{rec}_{bg}} \times W_{area} \tag{1}$$

Where C_{f1} represent the harmonic mean of the high confidence foreground pixel recall C^{rec}_{fg} and high confidence background pixel recall C^{rec}_{bg} scaled by the area of the candidate W_{area}. The goal is to maximize both the proportion of assumed to be foreground pixels inside of the region and the proportion of assumed to be background outside of the region. We multiply by W_{area} to prefer larger candidate regions when many of them have very similar harmonic means. In some cases, this criteria might prefer unnecessary larger but safer regions that preserve recall with some loss in precision.

The final background removal step on a given frame f^b simply removes any CC that is not fully contained in the final whiteboard region. Applying this to every frame in F^B we obtain the frame set F^C where the background has been mostly removed and most CCs belong to either handwriting or foreground elements like the speaker.

CC Stability Analysis. We identify and group CCs that are stable for some interval of time in the video. The output of this analysis is a spatio-temporal index of groups of stable CCs. For each group, the index stores a timeline describing the life span of the group split into time intervals based on the original addition/deletion times of each stable CC in the group. For each time interval, the spatio-temporal index stores a refined image of the group of CCs and their location in the frame. We can then use this index to reconstruct the binary frames without moving objects like the speaker. Also, we can use it to detect conflicts between regions of content for temporal video segmentation.

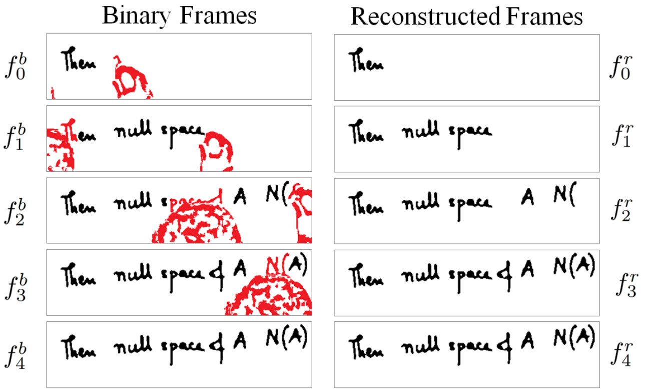


Fig. 4. CC Stability Analysis. Stable CCs are shown in black while unstable CCs are shown in red.

Figure 4 shows a comparison between raw binary frames containing both stable and unstable elements and corresponding reconstructed binary frames containing only stable elements. Static elements in the background and handwritten content in the whiteboard are expected to produce CCs that can be matched between neighboring frames. Moving elements like the speaker are expected to produced CCs that are hard to match across frames becoming quite unique and easy to identify. Also, a moving speaker will block portions of the whiteboard causing stable CCs to change in shape, merge and/or split for some frames. However, stable CCs remain with roughly the same shape on frames where they are not being occluded by the speaker.

The proposed method of analyzing stable CCs can be divided into two main steps: Stable CC identification, and Stable CC grouping.

Step 1: Stable CC identification. This step detects instances of unique CCs that appear in different frames within certain intervals of time. The input is the set of binary frames after background removal F^C. The output is a set of unique stable CCs S_{cc}. Given two binary frames f^c_i and f^c_j, we want to detect whether the CCs u and v, $u \in f^c_i$ and $v \in f^c_j$, might represent the same unique object in the video. We use temporal and spatial criteria for this task.

The spatial criterion is defined by computing the pixel-wise overlap between u and v. For this task, we use:

$$Space\,(u, v) = \frac{|p_u \cap p_v|}{|p_u|} \geq S^{\perp}_{space} \wedge \frac{|p_u \cap p_v|}{|p_v|} \geq S^{\perp}_{space} \tag{2}$$

where p_u and p_v represent sets of pixel locations of u and v respectively. We want the spatial overlap $|p_u \cap p_v|$ to represent at least $S^{\perp}_{space} = 92.5\%$ of both u and v pixels.

The temporal criterion is defined by the time difference between f^c_i and f^c_j. If t_i and t_j are the timestamps of f^c_i and f^c_j respectively, we define the temporal criterion as:

$$Temporal\,(t_i, t_j) = |t_i - t_j| \leq S_{time}^{\top} \quad (3)$$

Where the threshold $S_{time}^{\top} = 85$ seconds is the maximum time gap between f_i^c and f_j^c in order to consider u and v the same stable CC. These rules are applied between all pairs of frames on a moving window of length $S_{time}^{\top}$, and while for u and v the criteria might not hold directly, there might be another intermediate CC w, $w \in f_k^c$ with timestamp t_k such that $Space\,(u, w) \wedge Space\,(w, v) \wedge Temporal\,(t_i, t_k) \wedge Temporal\,(t_k, t_j)$ and thus we would conclude that u, v and w are 3 instances of the same CC.

The next process is to take the set of unique CCs U_{cc} and identify the subset of stable S_{cc} believed to represent handwriting assuming that all background has been correctly removed previously. For this task, we use a threshold $S_{count} = 3$ representing the minimum number of frames in which a given unique CC must appear in order to be considered stable: $S_{cc} = \{u \in U_{cc} | Count(u) \geq S_{count}\}$. By applying this filter, we manage to remove CCs belonging to moving objects like the speaker. However, this filter might also eliminate handwritten content appearing for shorts periods of time. In a sense, S_{count} is related to the minimum time that a given element must remain on the whiteboard to be considered relevant. Unstable elements are sometimes noise or even mistakes made by the writer who might quickly erased them from the whiteboard. Speakers might not be completely removed from the video if they do not move for many seconds.

Step 2: Stable CC Grouping. This procedure takes as input the set of stable CCs, S_{cc}, groups them by overlaps in space and time and finally produces a spatio-temporal index of these groups. First, we identify the sets O_s and O_t as the sets of CCs that overlap in space and time respectively. If we define $[t_0^u, t_n^u]$ and $[t_0^v, t_m^v]$ as the time intervals where u and v appear respectively, then we can define $Overlap(u, v)$ as:

$$Overlap\,(u, v) = t_0^u - S_w < t_m^v \wedge t_0^v < t_n^u + S_w \quad (4)$$

$$O_s = \{(u, v) \in S_{cc} \times S_{cc} |\, |p_u \cap p_v| > 0 \wedge u \neq v\} \quad (5)$$

$$O_t = \{(u, v) \in S_{cc} \times S_{cc} | Overlap(u, v) \wedge u \neq v\} \quad (6)$$

Where $S_w = 5$ seconds is a small window allowing stable CCs separated by a small time gap to be considered as overlapping in time. We finally define the set $O_{ts} = O_t \cap O_s$ as the set of all stable CCs overlapping both in space and time.

Now, the sets S_{cc} and O_{ts} are used to compute G_{cc}, a partition of S_{cc} that groups all stable CCs having spatial and temporal overlaps. This is an attempt to group together different CCs that might represent a unique set of objects in the video. For example, a typical CC representing handwriting might suffer multiple gradual changes as it gets written, overwritten or partially erased. Grouping CCs also accounts for some minor binarization errors that consistently split/merge CCs of a set of content objects. Here all these related CCs are merged and treated as single units that change shape over time.

We start with $G_{cc} = \{\{u\} | u \in S_{cc}\}$. Then, we use every pair in O_{ts} to agglomerate the subsets in G_{cc}. We then compute the relevant time stamps for every group of CCs in G_{cc} and create our spatio-temporal index to represent the extracted content from the whiteboard in terms of groups of CC and their corresponding timelines.

Frame Reconstruction. We use the spatio-temporal index to regenerate the binary frames of the video. First, we create refined images of each group of CCs in G_{cc} for each time interval in its timeline. These images are pixel-wise averages of foreground pixels of CCs from the group that co-existed in that particular interval of time. We then use these refined images to create the reconstructed binary frames F^R.

Using this procedure we also fill the gaps in the video where stable CCs were not visible due to occlusion by foreground elements like the speaker. Refined images of groups will be placed in all frames within its timeline, filing many holes left behind by unstable CCs removed in the previous steps. Figure 4 shows an example of this procedure. In the original frames, the head and hand of the speaker block some content that is now visible in the reconstructed frames.

Segmentation Through Conflict Minimization This process creates a temporal segmentation P^R of the video by identifying and minimizing conflicts. Here, we define two regions of contents to be in conflict with each other if they occupy the same space in the whiteboard, but exist at different time intervals. Content that gets erased will be in conflict with anything that gets written on the same space. The goal of the proposed algorithm is to find suitable split points that will automatically separate such conflicting content into different video segments using a minimum number of splits. This also means that content that never co-existed in the whiteboard might be located on the same video segment as long as they are written on different whiteboard regions. Two main steps are required for this process: CC conflict detection and CC conflict minimization.

Step 1: CC Conflict Detection During the CC stability analysis, two sets of pairs of stable CCs with spatial overlap, O_s and O_{ts}, were identified. We define the set of conflicting pairs of CCs O_{conf} using $O_{conf} = O_s - O_{ts}$. In other words, O_{conf} is the set of pairs of CCs that overlap in space but do not overlap in time.

Step 2: CC Conflict Minimization We start with a single segment corresponding to the entire video. Then, for each pair of conflicting CCs (u, v) in O_{conf} that co-exist in the current segment, we identify an interval of time such that if we split the video at any frame within that interval, u and v will go to different partitions thus resolving the conflict. The next step is to count the number of conflicts that can be resolved by splitting the video at each frame in the current segment. We greedily choose the frame where the maximum number of conflicts will be resolved as the next split point, and then we apply this procedure recursively on each partition until a stop condition is achieved. Based on assumptions about minimum time for relevant whiteboard content changes and to avoid adding to the summaries the mistakes made by the speaker that are quickly erased from the whiteboard, we do not split a segment if it contains less than $p^{\perp}_{conf} = 3$ conflicts or if splitting the current segment would create a segments shorter than $p^{\perp}_{time} = 25$ seconds.

We observed that this procedure tends to select intervals closely related to erasing events. More specifically, they are typically chosen at points where the speaker starts writing again after erasing content from the whiteboard.

Video Summarization. Given the set of video segments P^R found by conflict minimization, we use our spatio-temporal index to generate one key-frame for each segment to form the set of summary frames F^S. Instead of simply selecting frames from the segment, we compute each summary key-frame f_i^s by rendering the images of all CC groups that existed within that video segment. We use the timeline of each CC group to identify and render the latest version of its image during that video segment. In cases where the video segment still contains conflicting CC groups, we only render those that existed on the whiteboard closer to the end of the segment and have no conflicts among themselves.

IV. Experiments

In our experiments, we compensate for the lack of standard datasets for this application by creating our own labeled dataset from an existing collection of linear algebra videos. These HD videos (1920x1080 pixels) were recorded using a still camera (no zooming or panning), and the whiteboard always represents the largest element in the image. A total of 12 videos were manually labeled by 4 graduate students. Labeling each video required between 12 to 15 hours of work to define: ideal segments, best key-frames per segment, background objects per key-frame, unique content elements, and the pixel level ideal binarization for each handwritten content region. A total of 5 videos were used for training and the remaining 7 were reserved for evaluation. The average length of each testing video is about 49 minutes, and the entire dataset represents about 10 hours of fully labeled lecture footage. We used the training videos to adjust each of the parameters of our proposed pipeline, and whenever it was possible we used automated procedures to learn the values that would maximize our evaluation metrics on the training set. The newly labeled videos along with the ground truthing tools will be made publicly available.

To evaluate the effectiveness of our method at different stages, we use 4 baselines: binary, whiteboard segmentation, ground-truth-based whiteboard segmentation, and reconstructed. For each baseline, we uniformly sample 1 frame every 10 seconds of video. In addition, a fifth baseline, Max Sum, is added to compare an alternative strategy for key-frame selection after frame reconstruction using a sliding window to find key-frames with the maximum number of ink pixels within a 25 second window [1]. For each summary, we match CCs between ground truth and summary key-frames with overlapping time intervals. For every pair of frames, we find a translation of the summary key-frame that maximizes pixel-wise recall, and then evaluate pixel-wise recall and precision of overlapping CCs. We accept inexact matches between groups of CCs if their combined images match with pixel recall and precision above 50%. We chose this threshold to compensate for variations in thickness for good readable matches. For global recall metrics, we computed our metrics based on unique CCs in the ideal summaries, while global precision is measured in terms of all CCs in summary frames.

In Table I we present average results for different summarization methods. We consider number of frames, and the global and average per-frame recall/precision. As expected, the binary frames obtained the highest recall at 98.96% with the lowest precision because of all the non-content CCs.

Whiteboard Segmentation increases global precision from 64.01% to 70.32% with a small drop of 0.03% in global recall. Ground-truth based whiteboard segmentation suggests it would be possible to achieve precision of 73.27% with an ideal whiteboard segmentation method. Then, after analyzing stability and removing unstable CCs that belong to the speaker, the reconstructed binaries achieve a global precision of 94.28% which represents almost a 24% increase in precision with a drop in recall of just 2%.

Our proposed method using conflict minimization obtains a better compression rate (50% fewer key-frames on average) than selecting key-frames using the max sum method [1]. It also keeps global recall and precision almost at the same level. One of the reasons why our generated frames get slightly weaker global precision is because they render all non-conflicting CCs that exist on a given interval of time. This means that all noisy CCs that might exist on a given segment will be included in the corresponding summary frame. The same does not happen with sampled key-frames. In their original work, the max sum method produced an average of 45 frames per video with 95.08% recall on a different collection of lecture videos.

One issue with the local sliding window is that sometimes it can generate redundant key-frames if the window is too small. Our proposed method avoids this issue by optimizing key-frames globally instead of locally. Our approach can further compress the whiteboard content by placing non-overlapping regions of contents on a single frame. However, recall can drop if the video is under-segmented and conflicts still exist in a segment because our method will only display the most recent elements.

Our method uses multiple parameters in its pipeline. Many

TABLE I
Results for diff. summarization methods in 7 test videos.

Method	AVG Frames	AVG Global Rec.	AVG Global Prec.	AVG Per Frame Rec.	AVG Per Frame Prec.
Binary	295.71	**98.96**	64.01	**98.69**	63.30
W. Segm.	295.71	98.93	70.32	98.43	69.87
Gt. W. Segm.	295.71	98.94	73.27	98.49	73.29
Reconstructed	295.71	96.95	94.28	96.49	90.51
Min. Conflicts	**17.29**	96.28	93.56	95.73	**92.21**
Max Sum. [1]	34.42	96.49	**94.51**	96.13	91.95

of them would need to be fine-tuned on new collections. However, some depend on summarization goals and are input independent. More sophisticated methods will be required for automatic fine tuning of these parameters in the future.

V. Conclusions

We have proposed a novel method for lecture video summarization based on minimization of conflicts between regions of content. Our proposed CC stability analysis for reconstruction of binary frames and background removal procedures are very effective in increasing the precision of content extracted from lecture videos after binarization. As future work, we would like to test the effectiveness of this method using blackboard/chalkboard videos. We would also like to further extend the proposed method to work on harder cases where there is camera zooming/panning and lectures recorded using multiple cameras.

Acknowledgments. This material is based upon work supported by the National Science Foundation (USA) under Grant No. HCC-1218801.

References

[1] C. Choudary and T. Liu, "Summarization of visual content in instructional videos," *IEEE Transactions on Multimedia*, vol. 9, no. 7, pp. 1443–1455, 2007.

[2] K. Li, J. Wang, H. Wang, and Q. Dai, "Structuring lecture videos by automatic projection screen localization and analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1233–1246, 2015.

[3] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 797–819, 2011.

[4] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1480–1500, 2015.

[5] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2752–2773, 2016.

[6] Z. Li, G. M. Schuster, and A. K. Katsaggelos, "Minmax optimal video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1245–1256, 2005.

[7] C. Sujatha and U. Mudenagudi, "A study on keyframe extraction methods for video summary," in *International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, 2011, pp. 73–77.

[8] C. Nguyen, Y. Niu, and F. Liu, "Video summagator: an interface for video summarization and navigation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 647–650.

[9] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 3, no. 1, p. 3, 2007.

[10] A. G. Money and H. Agius, "Video summarisation: A conceptual framework and survey of the state of the art," *Journal of Visual Communication and Image Representation*, vol. 19, no. 2, pp. 121–143, 2008.

[11] H. S. Chang, S. Sull, and S. U. Lee, "Efficient video indexing scheme for content-based retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1269–1279, Dec 1999.

[12] M. Eberts, A. Ulges, and U. Schwanecke, "Amigo-automatic indexing of lecture footage," in *International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 1206–1210.

[13] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[14] W. Niblack, *An introduction to digital image processing*. Strandberg Publishing Company, 1985.

[15] M. Wienecke, G. A. Fink, and G. Sagerer, "Toward automatic video-based whiteboard reading," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 7, no. 2-3, pp. 188–200, 2005.

[16] T. Plötz, C. Thurau, and G. A. Fink, "Camera-based whiteboard reading: New approaches to a challenging task," in *International Conference on Frontiers in Handwriting Recognition*, 2008, pp. 385–390.

[17] S. Vajda, L. Rothacker, and G. A. Fink, "A method for camera-based interactive whiteboard reading," in *International Workshop on Camera-Based Document Analysis and Recognition*. Springer, 2011, pp. 112–125.

[18] T. Liu and J. R. Kender, "Rule-based semantic summarization of instructional videos," in *International Conference on Image Processing*, vol. 1. IEEE, 2002, pp. I–601.

[19] ——, "Semantic mosaic for indexing and compressing instructional videos," in *International Conference on Image Processing*, vol. 1. IEEE, 2003, pp. I–921.

[20] P. E. Dickson, W. R. Adrion, and A. R. Hanson, "Whiteboard content extraction and analysis for the classroom environment," in *IEEE International Symposium on Multimedia*. IEEE, 2008, pp. 702–707.

[21] L. Tang and J. R. Kender, "Educational video understanding: mapping handwritten text to textbook chapters," in *International Conference on Document Analysis and Recognition*. IEEE, 2005, pp. 919–923.

[22] ——, "A unified text extraction method for instructional videos," in *IEEE International Conference on Image Processing 2005*, vol. 3. IEEE, 2005, pp. III–1216.

[23] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[24] T. Liu and C. Choudary, "Content extraction and summarization of instructional videos," in *2006 International Conference on Image Processing*. IEEE, 2006, pp. 149–152.

[25] C. Choudary and T. Liu, "Extracting content from instructional videos by statistical modelling and classification," *Pattern analysis and applications*, vol. 10, no. 2, pp. 69–81, 2007.

[26] G. C. Lee, F.-H. Yeh, Y.-J. Chen, and T.-K. Chang, "Robust handwriting extraction and lecture video summarization," *Multimedia Tools and Applications*, pp. 1–19.

[27] K. Davila, A. Agarwal, R. Gaborski, R. Zanibbi, and S. Ludi, "Accessmath: Indexing and retrieving video segments containing math expressions based on visual similarity," in *Image Processing Workshop (WNYIPW), 2013 IEEE Western New York*. IEEE, 2013, pp. 14–17.

[28] M. Onishi, M. Izumi, and K. Fukunaga, "Blackboard segmentation using video image of lecture and its applications," in *International Conference on Pattern Recognition*, vol. 4. IEEE, 2000, pp. 615–618.

[29] P. Banerjee, U. Bhattacharya, and B. B. Chaudhuri, "Automatic detection of handwritten texts from video frames of lectures," in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2014, pp. 627–632.

[30] D. G. Lowe, "Object recognition from local scale-invariant features," in *IEEE International conference on Computer Vision*, vol. 2. IEEE, 1999, pp. 1150–1157.

[31] A. S. Imran, S. Chanda, F. A. Cheikh, K. Franke, and U. Pal, "Cursive handwritten segmentation and recognition for instructional videos," in *International Conference on Signal Image Technology and Internet Based Systems (SITIS)*. IEEE, 2012, pp. 155–160.

[32] R. R. Shah, Y. Yu, A. D. Shaikh, S. Tang, and R. Zimmermann, "Atlas: automatic temporal segmentation and annotation of lecture videos based on modelling transition time," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 209–212.

[33] J. Adcock, M. Cooper, L. Denoue, H. Pirsiavash, and L. A. Rowe, "Talkminer: a lecture webcast search engine," in *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 2010, pp. 241–250.

[34] K. Yadav, A. Gandhi, A. Biswas, K. Shrivastava, S. Srivastava, and O. Deshmukh, "Vizig: Anchor points based non-linear navigation and summarization in educational videos," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*. ACM, 2016, pp. 407–418.

[35] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *International Conference on Computer Vision*. IEEE, 1998, pp. 839–846.

[36] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.