Automatic Hierarchical Table of Contents Generation for Educational Videos

Debabrata Mahapatra
Videoken
Bangalore, India
debabrata.mahapatra@videoken.com

Ragunathan Mariappan
School of Computing
National University of Singapore
mragunathan@nus.edu.sg

Vaibhav Rajan
School of Computing
National University of Singapore
vaibhav.rajan@nus.edu.sg

ABSTRACT

The number of freely available online educational videos from universities and other organizations is growing rapidly. Accurate indexing and summarization are essential for efficient search, recommendation and effective consumption of videos. In this paper, we describe a new method of automatically creating a hierarchical table of contents for a video. It provides a summary of the video content along with a textbook–like facility for nonlinear navigation and search through the video. Our multimodal approach combines new methods for shot level video segmentation and for hierarchical summarization. Empirical results demonstrate the efficacy of our approach on many educational videos.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**; **Video summarization**; **Video segmentation**; • **Applied computing** → *Education*;

KEYWORDS

Table of Contents; Shot segmentation; Text Summarization; Tree knapsack

ACM Reference Format:
Debabrata Mahapatra, Ragunathan Mariappan, and Vaibhav Rajan. 2018. Automatic Hierarchical Table of Contents Generation for Educational Videos. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3184558.3186336

1 INTRODUCTION

Massive Open Online Courses (MOOCs) and other online learning resources have added many good quality educational videos on the Internet. The number, already in tens of thousands, is increasing by the day. Tools for search and recommendation are therefore indispensable for finding relevant content online. Search tools, in turn, rely on effective indexing and summarization based on metadata that describe the video content. Such metadata is usually manually created, for example through tags and titles of videos, which cannot scale and is often inaccurate.

Lecture recordings are often long videos with duration of upto several hours. Learners may be interested in only certain topics in the video or may need to review specific sections within the video. Both these tasks can be facilitated through a **Table of Contents (ToC)** that can lead the learner directly to the relevant section in the video. A well constructed ToC can itself provide both a summary as well as metadata for effective search in video databases. Despite the trend towards short videos in MOOCs, there still exists a large number of previously recorded long lectures. Moreover some presentations (e.g. in classrooms) or certain topics (e.g. long proofs) may continue to remain long videos for which a ToC would be a useful summarization and indexing tool.

Robust methods for Optical Character Recognition (OCR) are available [32] to extract visual words from frames in a video. Tools for Automatic Speech Recognition (ASR) are also rapidly improving [28]. The recognition accuracy of both these systems depends on video characteristics like image resolution, spoken language and accent used etc. but even with reasonably good outputs, there remains the problem of organizing the extracted information to form a coherent summary of the video. In this paper we develop a method of automatically creating a hierarchical table of contents for an educational video, focusing on two key aspects: video segmentation and hierarchical ToC creation.

Video segmentation is relatively easier when shot boundaries have abrupt transitions. The problem is harder for educational videos that typically have gradual transitions between shots. We design a segmentation method that uses a new content representation to capture visual shape information and a new signal construction that can detect both gradual and abrupt transitions. Moreover, in contrast to previous methods that use various similarity metrics between frames and determine the shot boundaries when the similarity is below a fixed threshold, our method does not use a fixed threshold: it adaptively determines a threshold based on signal characteristics of the input video.

Using the identified shots as the basic topical units, we develop a method that summarizes each shot and aggregates the summaries in a hierarchical manner to create a final hierarchical ToC for the video. Unlike previous extractive summarization methods that use text sources as inputs, our method infers dependency relationships from multi-modal, temporally dependent textual information extracted from videos. We develop a Tree Knapsack problem based formulation for generating the final ToC.

To summarize our contributions in this paper are:

- We develop the first method, called HMMToC, to automatically create a multi-modal hierarchical Table of Contents (ToC) for educational videos. The ToC provides a summary of the video and a textbook–like facility for nonlinear navigation and search through the video.

WWW '18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3186336

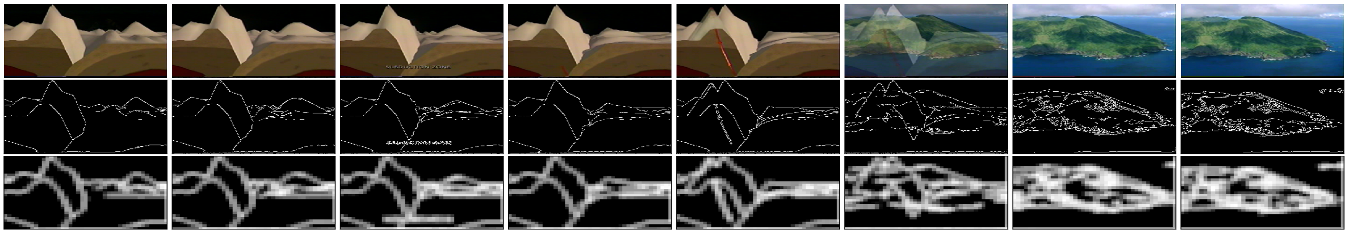


Figure 1: First row shows color image frames (size 240×352) extracted from the video UGS01.mpg obtained from TRECVID 2002 dataset [21]; Second row images are corresponding Canny edge map E_c; Third row shows the patch-wise entropy S (size 29×43); Note that the noise in edge map from binarization and subtle changes in the shape of consecutive frames are overlooked in corresponding patch-wise entropy, while showing a significant change in the representation at the point of shot transition.

- We develop a new method for shot–level segmentation of videos, that is used for ToC generation. Our threshold–free method has a new content representation and signal construction that can detect both gradual and abrupt shot transitions.
- We empirically demonstrate that HMMToC is more accurate than previous (non-hierarchical) methods in terms of both obtaining the section title and the timing of the title in the generated Table of Contents.

2 OVERALL DESIGN

A video is a multidimensional time-series signal $V : \mathbb{T} \to \mathbb{N}^{h\times w\times c}$, where $\mathbb{T} = \{1 \cdots T\}$ is the set of discrete time points, with h, w and c being the height, width and number of color channels of each frame. In general, a video is comprised of many shots. A **shot** is a group of temporally adjacent frames, which are generated from an uninterrupted camera capture. A **shot boundary** is a break in the continuum of the video signal.

Our ToC generation algorithm consists of three key steps.

1. *Segmentation and Key-frame Extraction.* The first step is the identification of the shot boundaries and segmentation of the input video into multiple shots. From multiple frames within each shot, we select representative **key–frames**.
2. *Information Extraction.* Second, we obtain textual information and associated metadata from the selected key–frames as well as from the audio transcript of the video.
3. *Table of Contents Generation.* In the final step, a summary for each shot is constructed using previously extracted information and all the summaries are aggregated to create a hierarchical ToC.

In the following sections, these three steps are described in detail.

3 SEGMENTATION AND KEY-FRAME EXTRACTION

In this section we describe the first key step of our ToC generation method: identification of shot boundaries and selection of representative key–frames.

3.1 Shot Boundary Detection (SBD)

Based on the type of transition between two adjacent shots, shot boundaries (SB) can broadly be classified into two categories: (1) *Abrupt Transition*, where the change is sudden, and (2) *Gradual Transition*, where the change occurs smoothly. The latter has subcategories, such as *desolve, fade in out, wipe* etc. [7].

We describe our method using the framework of Cotsaces *et al.* for shot boundary detection [7], comprising of three steps:

1. Visual Content Representation
2. Continuous Signal Construction, and
3. Classification of Shot Boundaries.

We design a novel content representation that captures information about shape of the objects in an image. We then, on a video level apply a novel method to construct multiple continuous signals at several time-resolutions to detect both types of shot transitions. Finally, for classification, where the SBs are chosen from the initial $T - 1$ frame boundaries, we design an adaptive method for finding a global threshold.

3.1.1 Visual Content Representation: Major variations in the video content are due to color and shape changes of objects that could be either in the foreground or background. Our novelty lies in a shape representation that is sensitive to major structural changes among neighboring frames, while maintaining invariability to subtle variations occurring by factors like motion, deformations, lighting conditions etc.

Shape information can be captured by the edge image of the frame. In general, for faster computation, Canny edge detection algorithm is used to obtain the binary edge map E_c. However, E_c is sensitive to small variations in the object. To gain invariance, we compute the entropy of E_c in a patch-wise manner. Denoting $\mathcal{P}$ as the set of patches extracted from E_c,

$$S(i) = -\left(p_0^i \log(p_0^i) + p_1^i \log(p_1^i)\right) \quad 1 \le i \le |\mathcal{P}| \tag{1}$$

is the entropy of i^{th} patch in $\mathcal{P}$, with p_0^i and p_1^i being the proportions of 0s and 1s in that patch. To further make the patch-wise entropy feature S more robust, we use overlapping patches. Figure 1 illustrates the invariance of S to subtle variations in E_c, while being sensitive to significant changes in the shape. This process can be interpreted as a pooling operation performed on the edge map. Edge based approaches have been used previously in the literature [15, 29], but not in the context of shape representation.

We capture variations in color by representing it with the 2–dimensional (2d) histogram H of Hue and Saturation values, as used in [3].

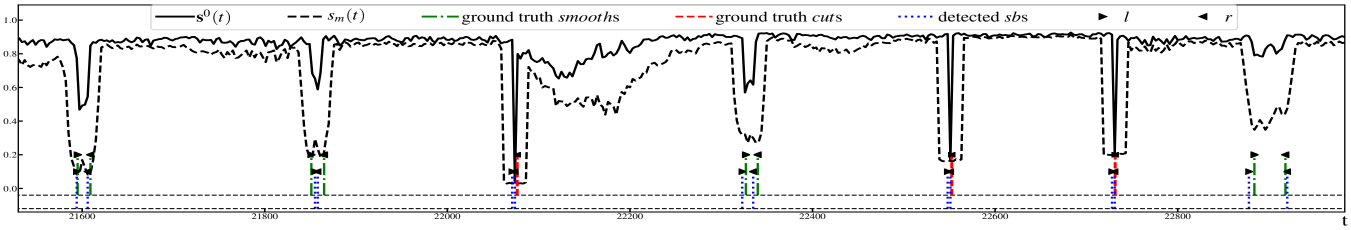


Figure 2: A portion of continuous signals for the video UGS01.mpg; ground truth labels are obtained from TRECVID dataset. The difference between $\mathbf{s}^0(t)$ and $s_m(t)$ is a clear indicator of when, and for how long, the gradual transition occurs. This is validated by juxtaposing the ground truth smooth boundaries (green) with the predicted ones (blue).

3.1.2 Continuous Signal Construction: We denote the boundary between a pair of frame numbers (l, r) as b. To start with, there are $T - 1$ boundaries. Two adjacent boundaries $b_1 = (l_1, r_1)$ and $b_2 = (l_2, r_2)$, such that $l_2 = r_1$, can be merged to form another boundary $\mathbf{b} = b_1 \cup b_2 = (l_1, r_2)$. To detect whether a particular b represents a shot transition, we compute a measure that is designed to capture the resemblance between its l and r frames as

$$\mu(b) = \sigma_c(H_l, H_r) \times \sigma_s(S_l, S_r), \tag{2}$$

where σ_c and σ_s are similarity metrics for color and shape features respectively. We use cosine similarity for σ_s. For σ_c, we use the metric suggested in [3], i.e. cross correlation

$$\sigma_c(H_l, H_r) = \frac{\sum_{i=1}^{N} \left(H_l(i) - \bar{H}_l\right)\left(H_r(i) - \bar{H}_r\right)}{\sqrt{\sum_{i=1}^{N} \left(H_l(i) - \bar{H}_l\right)^2 \sum_{i=1}^{N} \left(H_r(i) - \bar{H}_r\right)^2}},$$

where N is the total number of bins in the 2d histogram and $\bar{H}$ is the mean.

We introduce a multi-resolution approach to construct signals for a video, which will be useful for detecting both sudden and gradual transitions. At every boundary point $b_t, 1 \le t \le T - 1$, we compute the measure for several merged boundaries $\mathbf{b}_t^d, 0 \le d \le w$, such that

$$\mathbf{b}_t^d = \bigcup_{i=-d}^{d} b_{t+i} = (l_{t-d}, r_{t+d}), \tag{3}$$

where w is the window size. Consolidating all the $\mu(\mathbf{b}_t^d)$, we denote the $w + 1$ dimensional signal as $\mathbf{s}(t)$. It has been empirically shown in the literature, for example [13], that by appropriately thresholding $\mathbf{s}^0(t) = \mu(b_t)$, which is an uni-resolution approach, one can detect *cuts*, the sudden transitions, with high precision and recall. We leverage the extra information available in $\mathbf{s}(t)$ and construct another signal

$$s_m(t) = \min_{0 \le d \le w} \mathbf{s}^d(t) \tag{4}$$

that is indicative of the gradual transitions in a similar way as $\mathbf{s}^0(t)$ is for *cuts*. This is illustrated in Figure 2.

3.1.3 Classification of Shot Boundaries: The range of values that $\mathbf{s}(t)$ can take differs for different videos. So, the threshold for classifying whether a boundary is SB or not has to be adaptively decided. In our method, we use a clustering based technique to choose the threshold.

For detecting the *cuts*, first we quantize the values of $\mathbf{s}^0(t)$ into K clusters, $c_1 < c_2 \cdots < c_K$, by using *K Nearest Neighbor* algorithm. Then we pick one particular c_k as threshold, such that the membership count from c_k to c_{k+1} increases significantly. After thresholding $\mathbf{s}^0(t)$ with c_k and obtaining a binary signal, to precisely locate the *cut* shot boundaries, we perform *run length encoding* on it. Start and end points of the runs of 1s are classified as the corresponding l's and r's of the shot boundaries.

To detect gradual transitions, we perform the same steps on

$$d(t) = 1 - (\mathbf{s}^0(t) - s_m(t)) \tag{5}$$

as was done with $\mathbf{s}^0(t)$. After consolidating the boundaries obtained from $\mathbf{s}^0(t)$ and $d(t)$, we denote the set of shot boundaries as $\mathcal{B} = \{sb_1, \cdots, sb_{n_s}\}$, where n_s is the total number of shot boundaries detected, and $sb = (l, r)$ with l and r being the start and end of the corresponding runs of 1s.

A particular example of detection based on this method ($w =$ fps $= 30$ and $K = 5$) is shown in Figure 2, which clearly illustrates that the large value of $\mathbf{s}^0(t) - s_m(t)$ is indeed informative for detecting *smooth* transitions.

3.2 Key-frame Extraction in a Shot

To avoid processing all the frames in a shot, which can be time consuming and unnecessary, we identify representative *key-frames* within each identified shot. While removing redundant information, the selection must not remove useful information. In the context of educational videos, where an instructor intends to convey information visually, the portion in a shot that involves least amount of *distraction* can be assumed to contain our key-frame. In general, these distractions arise from the motion of either objects in the scene or camera. We design a novel method to find the stationary portions in a shot that contain least distractions. We first construct a time-series signal $i(t)$ that is the entropy of edge map E_c for frame $V(t)$. This is like equation 1, where the whole E_c is considered as one patch. Construction of $i(t)$ is inspired from [25]. For demarcation of the stationary regions, we apply two levels of approximations that converts $i(t)$ into a piece-wise constant signal.

Mean Shift clustering algorithm [6] is used on the values of $i(t)$ to get the cluster centers $C = \{c_1, \cdots, c_k\}$. We then approximate $i(t)$ to

$$\tilde{i}(t) = \arg \min_{c_i \in C} |c_i - i(t)|. \tag{6}$$

Although $\tilde{i}(t)$ is a piece-wise constant signal, due to the dynamism involved, even within a shot, it turns out to be highly jittery, as can

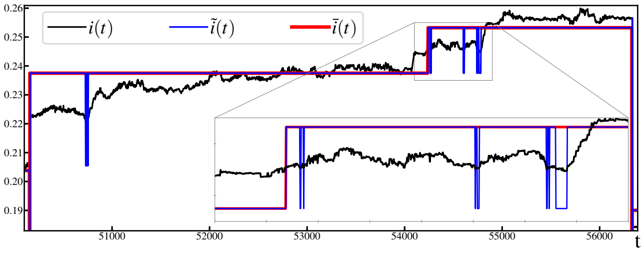


Figure 3: Signals for the entire span of a shot that is extracted from an educational video. Non-smoothness of $i(t)$ indicates motion within the shot. The zoomed portion shows jitter removal from $\tilde{i}(t)$.

be seen in figure 3. These variations are primarily due to the lack of monotonicity in $i(t)$.

We develop a signal smoothing operation, and apply it on $\tilde{i}(t)$ for the second level of approximation, to get rid of the false jitters. The noisy piece-wise constant signal $\tilde{i}(t)$ can be expressed as

$$\tilde{i}(t) = \sum_{i=1}^{n_p} \alpha_i \mathbb{I}_i(t) \tag{7}$$

$$\text{where, } \mathbb{I}_i(t) = \begin{cases} 1, & t \in \mathcal{I}_i, \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where, $\mathcal{I}_i$ is the interval in $\tilde{i}(t)$ with value α_i, $\mathbb{I}_i$ is the indicator for that interval, and n_p is the total number of piece-wise constant intervals in $\tilde{i}(t)$. In particular, α_i is one of the cluster centers, such that, $i(t)$ is closest to it as compared to other cluster centers. Jitters in $\tilde{i}(t)$ can be assumed as a group of temporally adjacent intervals having smaller lengths as compared to the stable intervals. We denote the desired noise-free piece-wise constant signal as

$$\bar{i}(t) = \sum_{i=1}^{n_p} \beta_i \mathbb{I}_i(t), \tag{9}$$

where, β_is are the parameters to be found by solving the following optimization problem:

$$\beta_1^* \cdots \beta_{n_p}^* = \underset{\beta_1 \cdots \beta_{n_p}}{\arg\min} \sum_{i=1}^{n_p} \frac{1}{2} (\alpha_i - \beta_i)^2 |\mathcal{I}_i| + \lambda \sum_{i=1}^{n_p - 1} \frac{(\beta_{i+1} - \beta_i)^2}{\min(|\mathcal{I}_i|, |\mathcal{I}_{i+1}|)}, \tag{10}$$

where $|\mathcal{I}_i|$ is the length of interval $\mathcal{I}_i$. The second term in equation 10 penalizes jitters in $\bar{i}(t)$ with appropriate smoothness factor λ, and the first term restricts β_is to original values for stable intervals. The solution for equation 10 can be found in closed form. Finally, by changing the β_i^*s to the closest value among $\{\alpha_{i-1}, \alpha_i, \alpha_{i+1}\}$ the jittery intervals get merged with the stable ones. As a result we obtain the desired $\bar{i}(t)$. This is illustrated in figure 3.

The final signal $\bar{i}(t)$ will have significantly less number of intervals than $\tilde{i}(t)$, and from each of these intervals one frame can be treated as a key–frame . The duration of a key–frame is same as that of the interval it is picked from.

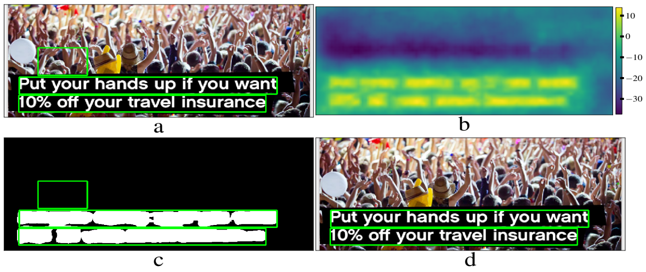

Put your hands up if you want
10% off your travel insurance
a
10
0
−10
−20
−30
b
c
Put your hands up if you want
10% off your travel insurance
d

Figure 4: a) Output of OCR (green boxes) displayed on the input image; b) Output of FCN on input image as heat map; c) Mask from thresholding overlaid with OCR output; d) Filtered OCR output.

4 INFORMATION EXTRACTION

In the second step of our ToC generation method, we extract textual information from all the selected key–frames and from the speech signal of the video. We use existing tools for Optical Character Recognition (OCR) to extract visual texts but apply a post–processing step to improve the precision. We assume that the audio transcript is available or can be obtained from an Automatic Speech Recognition (ASR) system. Saliency information for both visual and audio text are obtained.

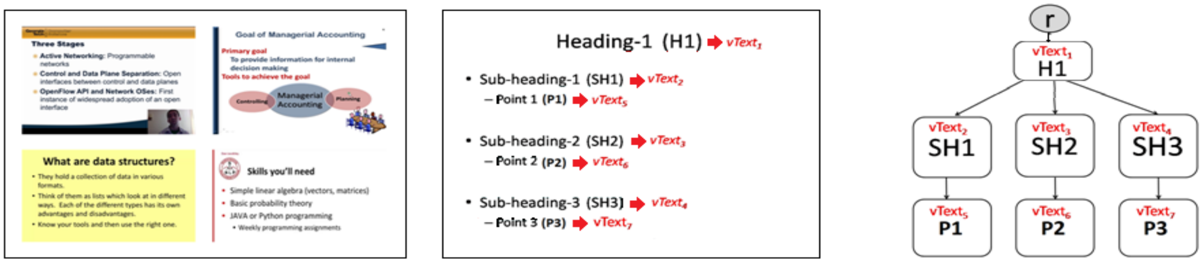
4.1 Visual-Text Extraction

We refer to a visual text entity as *vText* that comprises of the text string (in an image frame) along with its visual saliency.

4.1.1 Text Detection and Recognition: In this work, we use a commercially available OCR engine, offered by the *Cognitive Services* of Microsoft Azure cloud computing. Given an image, this service returns a set of texts and there corresponding bounding boxes. As in the case of any automatic OCR, for low-resolution or content-rich images, it misreads some of the non-text objects as text. So, to further improve its precision we have used another filter on the OCR output. This is achieved by training a *Fully Convolutional Network* (FCN) [17] to generate a heat map (HM) of the text regions in an input image, such that if a pixel in the HM is positive then it belongs to the text region. The HM is then binarized by thresholding with 0 in order to produce the mask of the text regions, see figure 4. The idea of obtaining a text map for an image is introduced in [32]. This network is trained with *COCO-Text* dataset [31]. If a bounding box obtained from the OCR engine has significant overlap (85%) with positive regions in the mask, then the corresponding text is used for further processing; this is also illustrated in figure 4.

4.1.2 Metadata and Salient Features: The timing and duration of a *vText* is inherited from the key–frame from which it is extracted. The salient features of a *vText* are *font size* (height of the bounding box), *boldness*, and *vertical location*. *Boldness* value is obtained by computing the mean stroke width of each character in the *vText* by using *Stroke Width Transform* (SWT) [10].

A key frame *kf* can be represented as a set of *vText*s. Within a shot, it may so happen that the $kf_i \cap kf_{i+1} \neq \phi$, i.e. some text



Three Stages
- Active Networking: Programmable networks
- Control and Data Plane Separation: Open interfaces between control and data planes
- OpenFlow API and Network OSes: First instance of widespread adoption of an open interface

Goal of Managerial Accounting
Primary goal
To provide information for internal decision making
Tools to achieve the goal

What are data structures?
- They hold a collection of data in various formats.
- Think of them as lists which look at in different ways. Each of the different types has its own advantages and disadvantages.
- Know your tools and then use the right one.

Skills you'll need
- Simple linear algebra (vectors, matrices)
- Basic probability theory
- JAVA or Python programming
  - Weekly programming assignments

Heading-1 (H1) ➡ $vText_1$
- Sub-heading-1 (SH1) ➡ $vText_2$
  - Point 1 (P1) ➡ $vText_5$
- Sub-heading-2 (SH2) ➡ $vText_3$
  - Point 2 (P2) ➡ $vText_6$
- Sub-heading-3 (SH3) ➡ $vText_4$
  - Point 3 (P3) ➡ $vText_7$

Figure 5: Left–to–Right (a) Example key-frames from educational videos, with a title text and body texts providing information about the title in a hierarchical fashion. (b) Textual units in key-frames ($vTexts$) represented as a template with Heading, Sub-headings and Points. (c) $vTexts$ have an inherent hierarchical/dependency relationship represented by a Tree as shown and is represented by a directed Tree with $vTexts$ as nodes and directed edges indicating the dependency relationship. We construct such *shot–level Trees* T_i for each shot in the video.

may reappear in the next key-frame. For example, in a typical slide show, bulleted points come one after another. To avoid processing repeated texts in the pipeline, we agglomerate all the kfs in a shot as

$$\mathcal{S} = \bigcup_{i=1}^{n_k} kf_i, \tag{11}$$

where n_k is the number of KFs in that shot, such that each $vText$ in the final set $\mathcal{S}$ uniquely represents a visual text. While agglomerating, two $vTexts$ are deemed the same if their salient features are similar and the texts match. If two $vTexts$ are found to be same then one of them is picked and the metadata is modified to account for both. Finally, in terms of textual information, a video can be represented as

$$\mathcal{V} = \{\mathcal{S}_i; 1 \leq i \leq n_s\} \tag{12}$$

4.2 Audio Text Extraction

The spoken text in audio is augmented with prosodic feature and metadata, together denoted by $aText$. For a spoken word, *pitch* of the speaker is used as the prosodic feature, which captures the intonations and stresses in speech [11]. We use an openly available tool [12] for pitch extraction.

5 TABLE OF CONTENTS GENERATION

In this section we describe the final step of how the text and associated metadata extracted from the selected key–frames as well as the identified shots are together used to create a Table of Contents (ToC) for a video.

ToC generation can be viewed as an extractive summarization problem. Extractive summarization problems have been formulated as combinatorial optimization problems like Maximum Marginal Relevance [5], Knapsack problem or Maximum Coverage problem [30]. Such formulations result in summaries that lack logical coherence because dependency relationships between the textual units (words or phrases) being summarized are ignored.

To generate coherent summaries [16] proposed a method for single document summarization that considers the discourse relationships i.e. logical connections between the textual units in the document. They constructed a Rhetorical Structure theory based discourse tree, inferred dependency relationships and took a tree trimming approach to summarization by formulating it as a Tree Knapsack problem (TKP). Our input text is extracted from video instead of text sources, and our method, inspired by their method, is novel in the techniques of inferring dependency relationships from multi-modal, temporally dependent information and in the choice of an appropriate cost function formulation for TKP. Our method comprises of three steps:

(1) **Shot level Tree construction.** Creation of a hierarchical representation of information in a shot.
(2) **Agglomeration of Shots.** Similar shots are agglomerated to create a hierarchical representation of the entire video as a single Tree.
(3) **Summary/ToC generation.** Selection of a best subtree representing the summary of this video formulated as a Tree Knapsack problem.

In the following, we describe each of the three steps.

5.1 Shot level Tree construction

The shots of a typical slide based educational video contains key-frames whose $vTexts$ inherently share a dependency relationship. For instance, in Figure 5 the $vTexts$ 2,3,4 can be considered to be dependent on $vText$ 1 i.e. $vText$ 1 is a parent to $vTexts$ 2,3,4. Similarly $vTexts$ 2,3,4 are parents to $vTexts$ 5,6,7 respectively. Thus a shot can be represented as a Tree which captures these dependency relationships derived from the meta-data of $vTexts$. Figure 5 illustrates an example shot and the corresponding Tree representation. Following are the steps involved in construction of a shot level tree representation:

(1) A Graph G is built with $vTexts$ as vertices and the edges representing the strength of the dependency relationship between $vTexts$. The strength of dependency relationship d is computed using the corresponding saliency scores *vertical location* (vl) and *boldness* (b) which serve as a proxy to the dependency relationships between the $vTexts$ in a key-frame (equation 13). The difference in the vl and b of the $vTexts$ within a key-frame signifies the distance between them in the hierarchy as illustrated in figure 5 for typical slide based educational videos. The λ in equation 13 is a

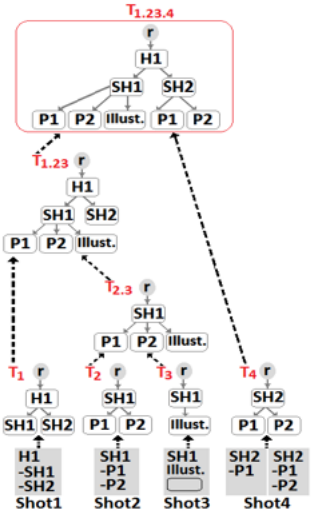


Figure 6: A single Tree T_a representing the whole video is generated by bottom up agglomeration of the shot level Trees T_i. This illustration contains 4 shots in total (shot 4 has two key–frames, others have 1 each). $T_1 - T_4$ are the shot level Trees. First T_2 and T_3 are agglomerated to $T_{2.3}$ as shots 3 and 4 are about the same topic. Among T_1, $T_{2.3}$, T_4 the Trees T_1 and $T_{2.3}$ are agglomerated to $T_{1.23}$ since $T_{2.3}$ is an elaboration of a topic in T_1. Finally $T_{1.23}$ and T_4 are agglomerated to a single Tree $T_{1.23.4}$ representing all the hierarchical relationships between $vText$s in the video.

fraction that governs the importance (weight) to be given to the differences in vl and b.

$$d(vText_i, vText_j) = \lambda * [vText_i^{vl} - vText_j^{vl}] + (1-\lambda) * [vText_i^b - vText_j^b] \tag{13}$$

(2) A Minimum Spanning Tree T' is constructed from the Graph G after negating the edge weights. The resultant T' will be a Tree with maximum weights or in other words with maximum strength of dependency relationships.

(3) The rooted shot level directed dependency Tree T is constructed by Breadth First Traversal (BFT) of T' with the $vText$ that has highest *utility* as the root. The **utility** u of a $vText$ in the hierarchy is computed using the corresponding saliency scores as in equation 14. Following similar intuition as that of equation 13, we consider a $vText$ with higher *vertical-location* (vl) and *boldness* (b) to be more useful in summarizing the content of the key-frame. Additionally we consider a $vText$ to be more useful when it gets repeated in key-frames of a shot, captured by the $frequency$ (f).

$$u(vText) = [\lambda * vText^b + (1-\lambda) * vText^{vl}] *(1 + log(vText^f)) \tag{14}$$

5.2 Agglomeration of Shots

The shots of a typical education video unfold temporally. The content of a new shot detected may be closely related to the previous shot (e.g. elaboration of the previous topic) or may not be (e.g. beginning of a new topic). Thus a group of adjacent shots could be agglomerated to a single hierarchical representation i.e. a Tree by merging the constituent shot level trees in a manner that preserves the dependency relationship between the nodes/$vText$s. These agglomerated shot level trees could be further agglomerated in a bottom up fashion to generate a single Tree representing the whole video. Figure 6 illustrates the process.

The method is essentially hierarchical clustering with an additional constraint that only temporally adjacent shots can be merged and additional logic for merging the shot level trees. We use the Earth Mover's Distance [27] between the corresponding visual information $V(\mathcal{S}_i)$ and $V(\mathcal{S}_j)$ as the distance metric (ds) between shots. Thus the following two steps are repeated until a single agglomerated Tree T_a is obtained:

(1) Construct a min-heap of distances $ds(\mathcal{S}_i, \mathcal{S}_j)$ between every two *adjacent* shots $\mathcal{S}_i$,$\mathcal{S}_j$.

(2) Pop the adjacent shots with minimum $ds(\mathcal{S}_i, \mathcal{S}_j)$ and merge the corresponding shot level Trees T_i and T_j to obtain a merged shot $\mathcal{S}_{ij}$ and the associated merged shot level Tree T_{ij}. The shot level trees T_i and T_j such that $i > j$, are merged as follows: If nodes at level 1 of T_i and T_j overlap then merge them by adding $subtrees(T_j)$ with minimum $ds(T_i, subtrees(T_j))$ as a child of the overlapping node of T_i. Otherwise merge them by making T_i,T_j siblings and children of a dummy root node r.

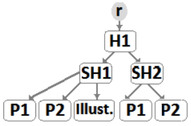
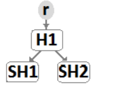
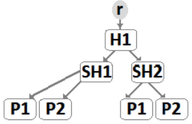
Final Agglomerated Tree (T_a)	ToC Budget(L)	Optimum Summary Subtree (T*)	ToC
	3		❖ H1 ○ SH1 ○ SH2
	7		❖ H1 ○ SH1 ■ P1 ■ P2 ○ SH2 ■ P1 ■ P2

Figure 7: The step of selecting the best subtree T^* from T_a representing the summary of the video is formulated as a Tree Knapsack problem where an optimal subtree T^* is selected to maximize the utility and constrained to the desired maximum length of the ToC L. The table illustrates how a ToC with optimum hierarchy information is generated with different values of L.

5.3 Summary/ToC generation

The problem of generating summary of desired length L (i.e. number of $vTexts$) can be formulated as a Tree Knapsack combinatorial optimization problem [16] - where an optimal rooted sub-tree T^* is selected from T_a to maximize a summary utility function $F(T)$. The summary utility for a given sub-tree is a function of various aspects of the $vTexts$ captured by the corresponding saliency scores. Further the utility of a $vText$ is re-weighted based on its *pitch* from the audio metadata and the depth of the corresponding node in the

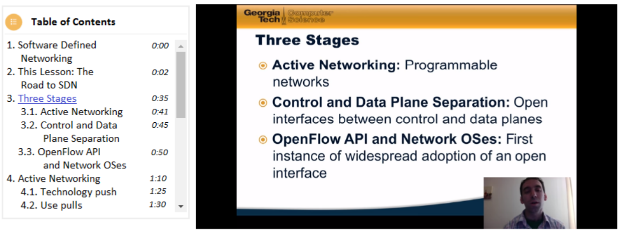


Table of Contents

1. Software Defined Networking 0:00
2. This Lesson: The Road to SDN 0:02
3. Three Stages 0:35
   3.1. Active Networking 0:41
   3.2. Control and Data Plane Separation 0:45
   3.3. OpenFlow API and Network OSes 0:50
4. Active Networking 1:10
   4.1. Technology push 1:25
   4.2. Use pulls 1:30

Three Stages

- **Active Networking:** Programmable networks
- **Control and Data Plane Separation:** Open interfaces between control and data planes
- **OpenFlow API and Network OSes:** First instance of widespread adoption of an open interface

Figure 8: The Hierarchical Table of Contents can be presented to the user facilitating non–linear navigation through the video. The user can click on a ToC entry to reach the corresponding section in the video.

hierarchy. As shown in equation 15 a $vText$ with higher *pitch* (p) and at a lesser depth has a higher utility value.

$$F(T) = \sum_{i=1}^{N} \frac{u(vText_i) * p(vText_i)}{depth(vText_i)} x_i \ , \tag{15}$$

$$x_i = \begin{cases} 1 & \text{if } vText_i \in T \\ 0 & \text{otherwise} \end{cases}$$

and N is the number of $vTexts$ or nodes in the Tree T. The $vTexts$ corresponding to the nodes of the optimal sub-tree T^* selected forms the summary. The selection can be performed by solving the ILP problem 16. Given the summary budget L the ILP problem solution, that can be found in polynomial-time [20], will be an optimum set of $vTexts$ forming a rooted subtree resulting in an hierarchical ToC. Figure 7 illustrates how hierarchical ToCs can be generated by the ILP (equation 16) from the agglomerated Tree $T_{1.23.4}$ shown in Figure 6 for varying summary lengths L.

$$\begin{aligned} T^* &= \max_{T} F(T) \\ s.t. \ &\sum_{i=1}^{N} x_i < L \quad \text{and} \quad \forall i \colon x_{parent(i)} \geq x_i . \end{aligned} \tag{16}$$

6 RELATED WORK

Segmentation of visual, speech and text data into coherent topics have been studied extensively. For example, TextTiling [14] segments text documents by detecting topic changes through vocabulary comparison. TextTiling has been extended to TopicTiling with the use of topic models [26]. A hierarchical Bayesian topic segmentation model for text documents is proposed in [8]. Topic segmentation models to structure lecture resources into cohesive segments, making them suitable for MOOCs content browsing is studied in [2].

Spoken lecture segmentation is studied in [18] where the problem is modeled through a normalized cut formulation. A Bayesian approach to unsupervised topic segmentation for both text and speech is studied in [9]. Discriminative topic segmentation of combined speech and text have also been investigated [19].

Detailed reviews on shot level segmentation of videos can be found in [7, 29, 33]. Phung *et al.* develop a method of topic segmentation in instructional videos using variation in content density [23] that is further extended with the features derived from the speech signal [24]. Phung *et al.* also develop topic transition detection methods in videos using Markov and Semi-Markov models [22]. These methods are not specifically designed for slide–based educational videos but for instructional films and are based on cinematic expressive functions. A related system, Talkminer, is a lecture webcase search engine that creates a searchable text index for lecture videos but does not segment the video or create a ToC [1]. Generation of ToC for educational videos was recently studied in [4] where their algorithm, MMToC, that uses features from all three – text, speech and visual – modalities was shown to outperform previous topic model–based methods.

Method	Precision		Recall		F-score	
	P_m	P_s	R_m	R_s	F_m	F_s
MMToC	0.56	0.3	0.13	0.08	0.21	0.12
HMMToC	**0.74**	0.23	**0.83**	0.17	**0.76**	0.20

Table 1: Experimental results on titles of ToC: Mean ($_m$) and standard deviation ($_s$) of Precision (P), Recall (R) and F-score (F) over all videos in the dataset.

Method	Precision		Recall		F-score	
	P_m	P_s	R_m	R_s	F_m	F_s
MMToC	0.83	0.22	0.21	0.14	0.31	0.14
HMMToC	**0.86**	0.23	**0.92**	0.181	**0.85**	0.21

Table 2: Experimental results on timings of ToC: Mean ($_m$) and standard deviation ($_s$) of Precision (P), Recall (R) and F-score (F) over all videos in the dataset.

7 EXPERIMENTS

We test the performance of our ToC generation method (HMMToC) with respect to two aspects of ToC, viz. the timing and title, comparing with the previous best method of ToC generation, MMToC.

7.1 Data and Evaluation

For evaluating the quality of ToC entries, we create a labeled dataset using 46 freely available educational videos from Youtube. Each video is either a part of an online course or a webinar related to computer science. Five human users, who were familiar with the video content, were asked to summarize the videos in the form of a ToC. Based on this labeled dataset, ToCs generated from competing algorithms are compared on the following two aspects.

(1) **Timing:** The chosen videos are of various lengths (5 to 35 minutes) with varying number of labeled ToC entries. Following the evaluation scheme of [4] if the ToC entry output by an algorithm falls within an interval of 10 seconds of the labeled entry, it is considered as a hit; otherwise a mis–hit.
(2) **Title:** A ToC entry obtained from an algorithm is considered a hit, if at least one of its words matches the ground truth ToC and if there is a hit with respect to timing as well.

7.2 Results

Tables 2 and 1 show the performance results of HMMToC and MMToC on our dataset. All three metrics, Precision, Recall and F-Score, for both the criteria, timing and title, are found to be better in HMMToC than in MMToC.

A qualitative comparison also clearly shows the superiority of HMMToC over MMToC. JSON files containing the ToC generated by both HMMToC and MMToC for all 46 videos can be downloaded for viewing[1].

7.3 Illustration

Figure 8 illustrates how the generated ToC can be presented to the user beside the video with hyperlinks to time points in the video thereby facilitating non–linear navigation through the video. The figure also shows how the hierarchy within the ToC is generated through the headings and sub–headings in slides present in the video.

8 CONCLUSION

We design HMMToC, the first method to automatically create a hierarchical multi-modal Table of Contents (ToC) for educational videos. Our method consists of three key steps. The first step is the identification of shot boundaries by segmentation of the input video, for which we design a new threshold–free method. The second step is the extraction of textual information and associated metadata from the selected key–frames in shots as well as from the audio transcript of the video. In the final step, a hierarchical ToC is generated using multimodal information from each shot that is formulated as a Tree Knapsack problem.

A hierarchical ToC provides a valuable summary of the video and a textbook–like facility for nonlinear navigation and search through the video. We empirically demonstrate that HMMToC is more accurate than previous (non-hierarchical) methods in terms of both obtaining the section title and the timing of the title. Our method relies on frames containing presentation slides within the video to generate the ToC. Many, but not all, educational videos usually contain such frames. Generation of ToC for videos that do not contain any slides remains a challenging problem that is not yet solved well.

REFERENCES

[1] John Adcock, Matthew Cooper, Laurent Denoue, Hamed Pirsiavash, and Lawrence A Rowe. 2010. Talkminer: a lecture webcast search engine. In *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 241–250.

[2] Ghada Alharbi and Thomas Hain. 2015. Using Topic Segmentation Models for the Automatic Organisation of MOOCs resources.. In *EDM*. 524–527.

[3] Evlampios Apostolidis and Vasileios Mezaris. 2014. Fast shot segmentation combining global and local visual descriptors. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 6583–6587.

[4] Arijit Biswas, Ankit Gandhi, and Om Deshmukh. 2015. Mmtoc: A multimodal method for table of content creation in educational videos. In *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 621–630.

[5] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 335–336.

[6] Dorin Comaniciu and Peter Meer. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence* 24, 5 (2002), 603–619.

[7] Costas Cotsaces, Nikos Nikolaidis, and Ioannis Pitas. 2006. Video shot detection and condensed representation. a review. *IEEE signal processing magazine* 23, 2 (2006), 28–37.

[8] L. Du, W. L. Buntine, and M. Johnson. 2013. Topic segmentation with a structured topic model. In *HLT-NAACL*. 190–200.

[9] Jacob Eisenstein and Regina Barzilay. 2008. Bayesian unsupervised topic segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 334–343.

[10] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. 2010. Detecting text in natural scenes with stroke width transform. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2963–2970.

[11] Hiroya Fujisaki and Keikichi Hirose. 1984. Analysis of voice fundamental frequency contours for declarative sentences of Japanese. *Journal of the Acoustical Society of Japan (E)* 5, 4 (1984), 233–242.

[12] Theodoros Giannakopoulos. 2015. pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. *PloS one* 10, 12 (2015).

[13] Alan Hanjalic. 2002. Shot-boundary detection: Unraveled and resolved? *IEEE transactions on circuits and systems for video technology* 12, 2 (2002), 90–105.

[14] Marti A Hearst. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics* 23, 1 (1997), 33–64.

[15] Wei Jyh Heng and King N Ngan. 2001. An object-based shot boundary detection using edge tracing and tracking. *Journal of Visual Communication and Image Representation* 12, 3 (2001), 217–239.

[16] Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-Document Summarization as a Tree Knapsack Problem.. In *EMNLP*, Vol. 13. 1515–1520.

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully Convolutional Networks for Semantic Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[18] Igor Malioutov and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 25–32.

[19] Mehryar Mohri, Pedro Moreno, and Eugene Weinstein. 2010. Discriminative topic segmentation of text and speech. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 533–540.

[20] Masaaki Nishino, Norihito Yasuda, Tsutomu Hirao, Shin-ichi Minato, and Masaaki Nagata. 2015. A Dynamic Programming Algorithm for Tree Trimming-based Text Summarization.. In *HLT-NAACL*. 462–471.

[21] NIST. Online. Homepage of TRECVID Evaluation. (Online). http://www-nlpir.nist.gov/projects/trecvid/

[22] Dinh Q Phung, Thi V Duong, Svetha Venkatesh, and Hung H Bui. 2005. Topic transition detection using hierarchical hidden Markov and semi-Markov models. In *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 11–20.

[23] Dinh Q Phung, Svetha Venkatesh, and Chitra Dorai. 2002. High level segmentation of instructional videos based on content density. In *Proceedings of the tenth ACM international conference on Multimedia*. ACM, 295–298.

[24] Dinh Q Phung, Svetha Venkatesh, and Chitra Dorai. 2003. Hierarchical topical segmentation in instructional films based on cinematic expressive functions. In *Proceedings of the eleventh ACM international conference on Multimedia*. ACM, 287–290.

[25] Liping Ren, Zhiyi Qu, Weiqin Niu, Chaoxin Niu, and Yanqiu Cao. 2010. Key frame extraction based on information entropy and edge matching rate. In *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, Vol. 3. IEEE, V3–91.

[26] Martin Riedl and Chris Biemann. 2012. TopicTiling: a text segmentation algorithm based on LDA. In *Proceedings of ACL 2012 Student Research Workshop*. Association for Computational Linguistics, 37–42.

[27] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision* 40, 2 (2000), 99–121.

[28] Frank Seide and Amit Agarwal. 2016. CNTK: Microsoft's Open-Source Deep-Learning Toolkit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2135–2135.

[29] Ananya SenGupta, Dalton Meitei Thounaojam, Kh Manglem Singh, and Sudipta Roy. 2015. Video shot boundary detection: A review. In *Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference on*. IEEE, 1–6.

[30] Hiroya Takamura and Manabu Okumura. 2009. Text summarization model based on maximum coverage problem and its variant. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 781–789.

[31] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge J. Belongie. 2016. COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images. *CoRR* abs/1601.07140 (2016).

[32] Qixiang Ye and David Doermann. 2015. Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence* 37, 7 (2015), 1480–1500.

[33] Jinhui Yuan, Huiyi Wang, Lan Xiao, Wujie Zheng, Jianmin Li, Fuzong Lin, and Bo Zhang. 2007. A formal study of shot boundary detection. *IEEE transactions on circuits and systems for video technology* 17, 2 (2007), 168–186.

[1] https://www.dropbox.com/sh/k1mksog9yncuu2f/AABliPpsjMuYLI9Ezh7OgwWfa?dl=0