Learning to Segment a Video to Clips Based on Scene and Camera Motion

Adarsh Kowdle and Tsuhan Chen

Cornell University, Ithaca, NY, USA
apk64@cornell.edu, tsuhan@ece.cornell.edu

Abstract. In this paper, we present a novel learning-based algorithm for temporal segmentation of a video into clips based on both camera and scene motion, in particular, based on combinations of static vs. dynamic camera and static vs. dynamic scene. Given a video, we first perform shot boundary detection to segment the video to shots. We enforce temporal continuity by constructing a Markov Random Field (MRF) over the frames of each video shot with edges between consecutive frames and cast the segmentation problem as a frame level discrete labeling problem. Using manually labeled data we learn classifiers exploiting cues from optical flow to provide evidence for the different labels, and infer the best labeling over the frames. We show the effectiveness of the approach using user videos and full-length movies. Using sixty full-length movies spanning 50 years, we show that the proposed algorithm of grouping frames purely based on motion cues can aid computational applications such as recovering depth from a video and also reveal interesting trends in movies, which finds itself interesting novel applications in video analysis (time-stamping archive movies) and film studies.

Key words: video temporal segmentation, film study

1 Introduction

Video analysis such as video summarization, activity recognition, depth from video, etc. are active research areas. Each of these communities focus on specific modalities of video; for example, estimating depth from video has been well explored for videos of a static scene captured from a dynamic camera. In addition to computational applications, studies in cognitive science [8] and film studies [3, 11, 25] require analyzing the visual activity in movies, where currently exhaustive manual effort is used [1]. In this work, we automatically segment an input video into clips based on the scene and camera motion allowing for more focused algorithms for video editing and analysis.

The task of video segmentation has been studied for various applications. Shot boundary detection [22, 32–34], performs a coarse segmentation of frames into the basic elements of a video, shots[1]. Several works build on top of this, grouping shots into scenes to aid video summarization [15, 24, 31, 35] or performing spatio-temporal segmentation [7, 9, 14, 18, 29]. Other video segmentation works include camera motion segmentation [10, 21, 23, 28, 30, 35] classifying camera motion such as tilt and zoom, albeit, without focusing on the motion of the objects within the scene. In this work, we define a novel task of labeling frames into one of four categories based on combinations

[1] A *shot* is everything between turning the camera on and turning it off, consisting of many *clips*. A *scene* consists of all the shots at a location.

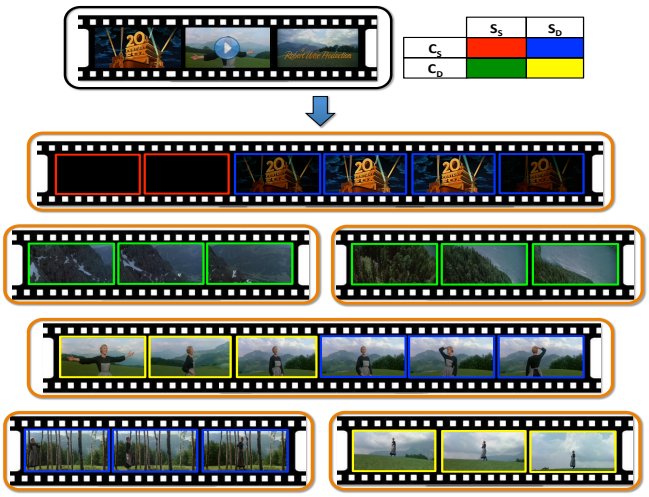


Fig. 1: The proposed algorithm takes a video as input (black box) and first segments the video to shots, where each shot is the group of frames shown in an orange box. The algorithm segments the frames of each shot to clips based on scene and camera motion, expressed as combinations of static camera (C_S) vs. dynamic camera (C_D) and static scene (S_S) vs. dynamic scene (S_D), shown with red, green, blue and yellow borders around the frame (color code on first row). The above are some results of our algorithm on the movie *Sound of Music*.

of *both* camera and scene motion i.e. static vs. dynamic camera and static vs. dynamic scene. While prior work in temporal segmentation were rule-based approaches, we propose a novel learning-based approach. Some results are shown in Fig 1.

An overview of our algorithm is shown in Fig 2. We formulate the task of video segmentation as a discrete labeling problem. We first perform shot boundary detection to segment the video into shots, and assume that the characteristics of motion between shots are independent of each other. In order to capture the heavy temporal dependence between frames within a shot, we construct an MRF over the frames of the shot with edges between consecutive frames. We extract intuitive features from the optical flow between consecutive frames and using labeled video data learn classifiers to distinguish between the different classes. Using the classifier scores as evidence for the different discrete categories we setup an energy function over the graph of frames enforcing smoothness between adjacent frames and infer the minimum energy labeling over the frames. We then use the resulting labeling to extract contiguous sequence of frames that are assigned the same label thus segmenting the video to clips.

Contributions. The main contributions of this paper are:

- We propose a novel learning-based approach to segmenting a video into clips based on *both* camera and scene motion.
- We show that the joint inference over all the frames of the shot as opposed to independent decisions on each frame leads to significantly better performance.
- Motivated by film study, we apply our approach to a collection of sixty full-length movies to reveal interesting trends of scene and camera motion over the years.
- We successfully use the proposed approach for the novel application of time-stamping archive movies and for recovering depth from video.

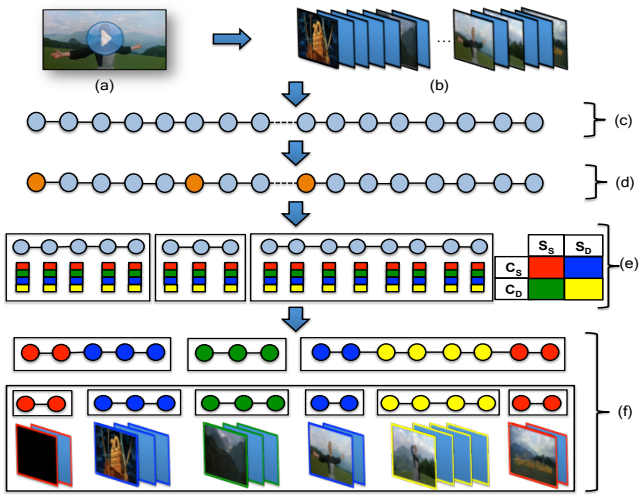


Fig. 2: Algorithm overview: (a) We start with the input video and (b) extract the frames of the video. (c) We represent each frame as a node in our graph and (d) perform shot boundary detection to identify shot boundary frames, shown in orange. (e) We now construct an MRF over the frames within each shot and formulate a discrete labeling problem where each frame has to be assigned one of four discrete labels, the combinations of static camera (C_S) vs. dynamic camera (C_D) and static scene (S_S) vs. dynamic scene (S_D), in red, green, blue and yellow. (f) The minimum energy labeling obtained via graph cuts allows us to group consecutive frames with the same label into clips. Note that in addition to segmenting the video to shots, shots 1 and 3 have been further segmented into clips based on camera and scene motion.

2 Related Work

With the number of video capture devices and the movie industry rapidly growing, video analysis is gaining a lot of popularity. A number of researchers are studying different aspects of video analytics, from shot detection to video summarization. We give an overview of related works with a focus on video segmentation and bring our work into perspective amongst these works.

Shot boundary detection. One of the key steps in most video analysis applications, is given a video extracting out the basic components *shots*, that allows for further processing. Shot boundary detection is also a form of video segmentation albeit a coarse one. Automatic shot boundary detection has been very well studied in the past with approaches varying from using the changes in image color statistics across adjacent frames to using edge change ratio [22, 33, 34]. There are a number of detailed surveys that compare the various techniques [4, 13, 19]. We refer the reader to a recent formal study of shot boundary detection by Yuan et. al. [32]. In our work, we perform shot boundary detection as our first stage of coarse segmentation, but our goal is to go beyond and obtain a finer segmentation of the shots to clips.

Spatio-temporal segmentation. With the shots from a video extracted using the shot boundary detection algorithms, an active line of research is spatio-temporal segmentation of the video shot frames. The goal of these works is to obtain a spatial grouping

of pixels enforcing temporally consistency [7, 9, 14, 18, 29]. In our work, we focus on purely temporal segmentation where we segment the video shot into clips by giving the entire frame a discrete label. However, we note that our proposed approach can allow for better, more focused spatio-temporal segmentation.

Scene segmentation. A line of active research motivated by the application of video summarization, indexing and retrieval, is segmenting a video into scenes [35]. A scene can consist of multiple shots, each with different camera and scene dynamics. Scene segmentation leverages the use of semantics to group shots to scenes and extract key frames to summarize the video. Scene Transition Graph by Yeung et. al. [31] is one such approach, where a graph is constructed with each node representing a shot and edges representing the transitions between shots. Hierarchically clustering the graph splits it into several sub graphs resulting in the scenes. A similar approach of clustering was proposed by Hanjalic et. al. [15] to find logical story units in the MPEG compressed domain. More recently Rasheed et. al. proposed a similar approach for scene detection in Hollywood movies and TV shows [24]. In our work, as opposed to clustering shots to scenes, we break the shot into clips based on scene and camera motion.

Camera motion characterization. Camera motion characterization is related to our work. Motion characterization is known to aid video compression and allow for a more compact video representation for content-based video indexing. Dorai et. al. [10], Tan et. al. [23] and Zhu et. al. [35], analyze motion vectors encoded in the P and B-frames in the compressed MPEG stream to characterize the camera motion. However, the focus here is to characterize camera motion such as tilt and zoom. Ngo et. al. proposed an approach to extract camera motion via temporal slice analysis [21]. Srinivasan et. al. [28] and Xiong et. al. [30] introduced motion extraction methods analyzing the spatial distribution of optical flow. However, these rule-based approaches assume that the Field of Expansion (FOE) [16] or Focus of Contraction (FOC) is at the center of the image, which is not always satisfied. However, unlike our work, these works are *rule-based* approaches considering only camera motion and are agnostic to the *scene motion*. Cifuentes et. al. [12] perform supervised classification of pre-segmented clips to camera motion classes to improve interest-point tracking that is complementary to our work.

In our work, we propose a *learning-based* algorithm to perform segmentation of the video into clips by analyzing *both* scene and camera motion. We apply the proposed algorithm to movies from different decades and show that such a grouping purely based on motion cues can aid computational applications such as recovering depth from a video and also reveal interesting trends in movies, which finds itself interesting novel applications in video analysis (time-stamping archive movies) and film studies.

3 Algorithm

In this section, we describe the proposed algorithm in detail. We first describe the problem formulation followed by the description of the proposed approach in detail.

3.1 Problem formulation

Given a video, we wish to split the video into shots and further split each shot into one of four categories based on camera and scene motion. We treat this as a frame level

discrete labeling problem. We first consider a binary labeling problem of shot boundary detection, labeling the frame where the shot changes as a Shot-Boundary (SB). Using these shots as structural elements of the movie, we cast the problem of splitting the shots to clips as a 4-label discrete labeling problem where each frame is to be assigned one of four discrete labels. We use the following notation throughout the paper,

- Static Camera, Static Scene (C_S, S_S)
- Dynamic Camera, Static Scene (C_D, S_S)
- Static Camera, Dynamic Scene (C_S, S_D)
- Dynamic Camera, Dynamic Scene (C_D, S_D)

We will finally use the labeling of frames to extract contiguous sequence of frames which are assigned the same label, thus segmenting the video to clips.

3.2 Proposed approach

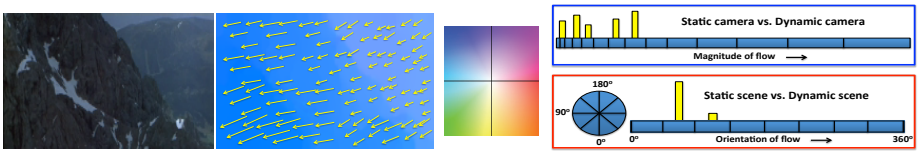
We now describe the proposed approach in detail. We leverage the use of optical flow all through our algorithm. Given a video we first compute the optical flow for each frame.

Shot boundary detection A shot is a sequence of frames captured between turning the camera on to turning it off (at one-shot). The first stage of our algorithm is identifying shot boundaries in the video. Shot boundary detection has been fairly well studied using features such as change in color statistics, edge change ratio (ECR), etc. In our experiments with movies, we found that using the color statistics is not reliable especially in case of shots boundaries between scenes of natural environments. ECR [33] using image edges is much more reliable however, suffers in the presence of image noise and requires manual tuning to work across movies.

An observation with optical flow is that, while the flow for frames on either sides of the shot boundary have similar magnitude, the flow drastically changes at the shot boundary frame. Identifying the shot boundary using a fixed threshold on the flow between only two consecutive frames is however not too reliable since, the flow is often very noisy (especially old movies). We therefore use more temporal support and dynamically change the threshold to detect a shot boundary. We use a sliding window *W*, of ten frames centered on the test frame, and compute the median optical flow magnitude for each frame in this window. Given this vector of W flow magnitudes denoted by $|\mathbf{O}|_W$, our goal is to identify the outlier. We compute the median[2] and standard deviation of this vector. Any frame with flow magnitude more than two standard deviations away from the median is identified as a shot boundary (SB).

Segmenting video to clips The shot boundaries from the previous stage are treated as the last frame of a shot, thus segmenting the video to shots. Our goal is to segment each shot into clips based on camera and scene motion. We cast this segmentation task as a frame-level discrete labeling problem. We formulate the multilabel problem as an energy minimization problem over a graph of the frames in the shot. This is analogous to constructing a graph over frames of the whole video and breaking the links when

[2] Median is more reliable in identifying the outlier than using the mean.



(a) (b)

Fig. 3: Illustration of the features used in the unary term. (a) shows a sample frame with the corresponding color coded flow. Flow vectors are illustrated with arrows. (b) Row 1 shows the log-bin histogram used to classify static vs. dynamic camera, and Row 2 shows the orientation histogram used to classify static vs. dynamic scene. Refer to Section 3.2 for details.

there is a shot boundary assuming independent motion between shots.

Notation. Let the frames in the i^{th} shot of the video be denoted as F_i. The objective is to obtain a labeling L_i over the shot frames such that each frame f is given a label $l_f \in \{(C_S, S_S), (C_D, S_S), (C_S, S_D), (C_D, S_D)\}$. We build a graph over the frames $f \in F_i$, with edges between adjacent frames denoted by $\{N\}$. We define the energy function to minimize over the entire shot as:

$$E(L_i) = \sum_{f \in F_i} E_f(l_f) + \lambda \sum_{(f,g) \in N} E_{fg}(l_f, l_g) \tag{1}$$

The unary term $E_f(l_f)$ measures the cost of assigning a frame f to the frame label l_f. The pairwise term $E_{fg}(l_f, l_g)$ measure the penalty of assigning frames f and g to labels l_f and l_g respectively, and λ is a regularization parameter.

Unary term (E_f) We extract intuitive features from the optical flow and using labeled data learn classifiers to classify between *static vs. dynamic camera* categories and *static vs. dynamic scene* categories. We now describe the unary term in detail.

Static camera vs. dynamic camera. Our goal is to learn a classifier that, given the features for the current frame returns a score indicating the likelihood that the current frame was captured with a static camera vs. a dynamic camera. In the most trivial scenario where a static camera is looking at a static scene, the optical flow between the frames should have zero magnitude. In this case, a histogram of the optical flow magnitude would peak at the zero bin indicating that it is a static camera. However, in a typical video the scene can be composed of dynamic objects as well.

We handle this by computing a histogram of the optical flow magnitude, using log-binning with larger bins as we move away from zero magnitude. We obtain a 15-dim vector of the normalized histogram represented as $\Psi(f)$ to describe each frame f as illustrated in Fig 3. Using manually labeled data, we use these vectors as the feature vectors and learn a logistic classifier to classify the frame between static and dynamic camera. Let θ_S represent the logit coefficients we learn during training to classify static camera frames against dynamic camera frames. During inference, given a frame x, we compute the likelihood of the frame being captured by a static camera represented as $L(C_S)$ and that by a dynamic camera as $L(C_D)$ given by:

$$\begin{aligned} L(C_S) &= P(C_S|\Psi(x); \theta_S) \\ L(C_D) &= 1 - P(C_S|\Psi(x); \theta_S) \end{aligned} \tag{2}$$

Static scene vs. dynamic scene. Consider a scene captured from a dynamic camera, we use a simple cue from the optical flow to separate static vs dynamic scene. Given the flow for each frame of the shot, we compute the orientation of the flow field at each pixel and bin them into an orientation histogram with eight bins evenly spaced between 0° to 360° as illustrated in Fig 3. In addition, we use a *no-flow* bin that helps characterize a static camera.

Intuitively, the peak in this histogram indicates the dominant direction of camera motion. We remove the component of the histogram corresponding of this dominant motion, and normalize the residual histogram. We use the entropy of this residual distribution $(H(f))$ as a feature to identify static scene vs. dynamic scene. We assume a gaussian distribution over the entropy values and using the labeled data fit a parametric model to the residual entropies for frames labeled as static frames and frames labeled as dynamic frames which results in two gaussians $\mathcal{N}(\mu_S, \sigma_S)$ and $\mathcal{N}(\mu_D, \sigma_D)$ respectively. At inference, given a new frame x, we compute the residual entropy $H(x)$ and compute the likelihood for static scene, $L(S_S)$ and dynamic scene, $L(S_D)$ given by:

$$\begin{aligned} L(S_S) &= P(S_S|H(x); \mu_S, \sigma_S) \\ L(S_D) &= P(S_D|H(x); \mu_D, \sigma_D) \end{aligned} \tag{3}$$

Since the dominant camera motion has been extracted out in the computation of the scene motion, we assume that the scene and camera motion are independent. We show in Section 4.2, that this assumption performs respectably on both movies and user videos, however, causes errors in the ambiguous case of a dynamic camera looking at a static object vs. a static camera looking at a dynamic object up-close where the moving object has a large spatial extent in the frame. This is a scenario ambiguous even to humans, without contextual or semantic reasoning about the actual spatial extent of the dynamic object and the surrounding scene. While we do not model this in our work, we see that such a model can be incorporated into Equation 3.

We compute the likelihoods for the four discrete frame labels to define the unary term in our MRF as follows:

$$\begin{aligned} L(C_S, S_S) &= L(C_S) * L(S_S) \\ L(C_D, S_S) &= L(C_D) * L(S_S) \\ L(C_S, S_D) &= L(C_S) * L(S_D) \\ L(C_D, S_D) &= L(C_D) * L(S_D) \end{aligned} \tag{4}$$

We use the negative log-likelihood as the unary term (E_f) for the four discrete labels.

Pairwise term (E_{fg}) We model the pairwise term using a contrast sensitive Potts model.

$$E_{fg}\left(l_f, l_g\right) = \mathrm{I}\left(l_f \neq l_g\right) \, \exp(-\beta d_{fg}) \tag{5}$$

where $\mathrm{I}\,(\cdot)$ is an indicator function that is 1(0) if the input argument is true (false), d_{fg} is the contrast between frames f and g and β is a fixed parameter. The pairwise term tries to penalize label discontinuities among neighboring frames modulated by a contrast-sensitive term. Given the optical flow between adjacent frames f and g, we warp the image from frame g to frame f. The mean error between the original frame f and the flow induced warped image is used as the contrast (d_{fg}), between the two

frames in the pairwise term. In practice, this enforces temporal continuity since the contrast is low if the flow can describe the previous frame very well.

With the energy function setup, we use graph-cuts with α-expansion to compute the MAP labels for each shot, using the implementation by Bagon [2] and Boykov et. al. [5, 6, 17]. We show some of our results on the full-length movies and user videos in Fig 1 and 5. We note that we can extend the proposed algorithm using prior work (Section 2), to obtain *true* camera motion characterization of frames into zoom, pan, tilt, etc without scene motion clutter, by either post-process the frames labeled by our algorithm as dynamic camera frames or add additional discrete labels.

4 Experiments and Results

In this section, we describe and discuss our results. We first describe our dataset followed by quantitative and qualitative analysis of the proposed approach. We then discuss interesting applications and analysis applying our approach on full-length movies.

4.1 Dataset

Our dataset consists 60 full-length movies and five user videos. The movies are a subset of the dataset used in [8] that spans from 1960-2010, with 12 full-length movies in each decade. We divide the dataset into two parts.

The first subset of the dataset *(Set-A)* consists of five user videos and one full-length movie that we use for the quantitative analysis. We sample the videos at a frame rate of 10fps. We manually label the frames with one of four discrete labels we define in this work. We use these labeled 110,000 frames (made publicly available on our website[3]) to perform our quantitative analysis. We also label shot boundaries on a subset of the movie to evaluate shot boundary detection.

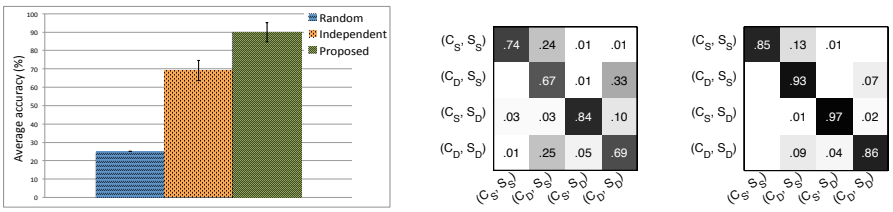
The second subset of the dataset *(Set-B)* consists of the 60 full-length movies. We sample the movies at 10fps resulting in an average of 50,000 frames for each movie. We use the results of the proposed algorithm on this exhaustive dataset to infer trends in scene and camera motion in movies over the years. We later use the dataset to show a novel application of time-stamping archive movies.

We extract the optical flow between every two consecutive frames for each of the videos in the dataset using the implementation by Liu et. al. [20].

4.2 Quantitative analysis

We use the *Set-A* to perform the quantitative analysis. Given the optical flow for each of the videos, we first perform the shot boundary detection as described in Section 3.2. We first evaluate the performance of the **shot boundary detection** in comparison to the edge change ratio (ECR) approach [33], which is known to perform better than using color histograms [32]. We use the manually labeled movie data to compute the F-score of detecting the shot boundary. The ECR performed respectably with an F-score of **0.92** (cross-validation to set threshold), well above random. The proposed flow-based

[3] `http://chenlab.ece.cornell.edu/projects/Video2Clips`



(a) (b) (c)

Fig. 4: Quantitative analysis: (a) Average accuracy of the frame level labeling using the learnt models. The proposed approach of joint inference via the graph over frames achieves significant improvement in performance over taking an independent frame level decision. Confusion matrix: (b) Independent decisions, (c) Joint inference via the temporal graph over frames results in significant improvement as evident in the diagonal elements.

approach performed better with an F-score of **0.96** without requiring to set the threshold. While both these approaches had errors due to missed detections in case of slow fades and dissolves, we observed that the lower performance of ECR is due to false positives in case of long shots, or when text is instantaneously overlaid on the movie frame. Other sophisticated approaches (Sec 2) can easily be plugged in here as shot boundary detection serves as a pre-processing step in our approach.

In order to evaluate the performance of the algorithm in segmenting the video to the four classes, we treat the problem as a multi-class labeling problem and use the frame-level labeling accuracy as our metric. We perform leave-one-out cross validation. Using all the videos in the dataset except the test video, we learn the model parameters described in Section 3.2. Given these parameters we evaluate the performance on the test video. We note that while the task we tackle has not been addressed before, prior work on camera motion characterization use a rule-based approach using features extracted from the motion vectors albeit without reasoning about the scene dynamics and making a decision at a per-frame level. We compare the performance of the proposed approach of using a multilabel MRF over the frames against taking an independent decision on each frame using the learnt models. We report the performance in Fig 4a. We see that using the learnt models on each frame independently performs much better than taking a random decision, the temporal consistency enforced by the proposed approach gives a significant additional boost of more than 20%. We compare the confusion matrices in Fig 4. We observe via the diagonal elements that the proposed algorithm performs better across all the categories.

Observations. The key observation from the confusion matrices is that the model is able to learn to distinguish between the different classes as evident from the dominant diagonal, even in the independent case (Fig 4b). We investigate the errors by considering the non-zero off-diagonal entries in Fig 4c.

Consider the case of static camera, static scene (C_S, S_S) i.e. row 1 of the confusion matrix. We first note that the ratio of frames in a movie that belongs to this category were fairly low as we see in Sec 4.4. The confusion here is due to the instability of the camera that results in dynamic camera, static scene (C_D, S_S) being a more appropriate label. We see even in Fig 4b that there is a large confusion between the classes of dynamic camera, static scene (C_D, S_S) and dynamic camera, dynamic scene (C_D, S_D) i.e rows 2 and 4 of the confusion matrix. The proposed approach (Fig 4c) performs significantly

better than the independent case, however the confusion is due to the dynamic content in the scene having varied spatial extents in the frames, as noted in Sec 3.2.

4.3 Qualitative analysis

We show some qualitative results of our proposed algorithm in Fig 1 and 5. Please see the website[3] for sample result videos on the movie Sound of Music, and the user videos. The proposed algorithm accurately segments the video to shots and shots to clips. We note in particular the result on the 1990 movie, *Goodfellas* (Fig 5b), where the famous 3 minute long Copacabana shot is accurately segmented out as a single shot. Other approaches using color statistics and edge change ratio failed to do so due to the drastically changing image content. In addition, our algorithm accurately segmented the shot into a dynamic camera, dynamic scene clip. We also note the result on a user video (Fig 5d), where the algorithm accurately segments the shot into clips captured using a dynamic camera, and switches between a static scene to a dynamic scene (when the person walks across the scene) and back. Also note the negative results in Fig 5e.

4.4 Trends in movies over the years

We use *Set-B* with movies from 1960-2010 to analyze trends in movies. We run our algorithm on each movie to obtain a frame-level labeling. We then analyze the distribution of the various labels for movies from each decade to observe trends across time.

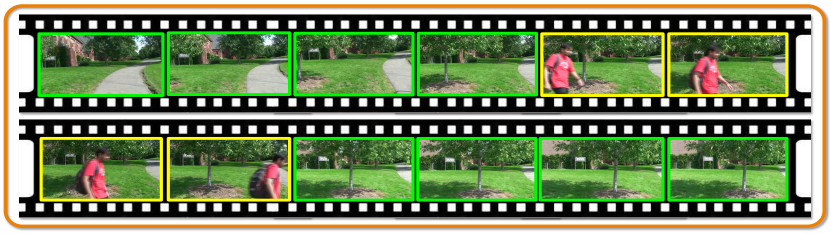
We compute the percentage of frames in the movie for each label and plot it across years in Fig 6. Our first observation, is that the movies from each decade follow a particular distribution of the four labels. We observe that, the category static camera, static scene (C_S, S_S) is very sparse in movies irrespective of the decade the movie belongs to. We also note that the category of dynamic camera, dynamic scene (C_D, S_D) dominates over all other categories. We observe from that the categories (C_S, S_D) and (C_D, S_S) follow distinct trends. The dynamic camera categories (C_D, S_D) and (C_D, S_S) seem to follow an increasing trend while the category of static camera, dynamic scene (C_S, S_D) follows a downwards trend.

We explore this further by first grouping the static and dynamic scene categories to compare the categories of static camera vs. dynamic camera in Fig 6b. We note that our earlier observation of the increase in the use of a dynamic camera clearly stands out. We discovered from literature that the movies have evolved as a result of advancement of capture technology such as cameras on helicopters, zip-lines, stable camera harnesses, etc for moving cameras, resulting in an increase in the dynamic camera shots. Our algorithm has interestingly picked up this trend, which researchers in film-studies manually observed [3,11,25]. We then group the static and dynamic camera categories to compare the static scene vs. dynamic scene categories in Fig 6c. We see that there is a decreasing trend of dynamic scene category and an increasing trend of the static scene category. One explanation for this is that, with more sophisticated and stable dynamic cameras, more static scenes are being captured now. Another explanation from recent research in cognitive science is that recent movies could have reduced dynamics due to rapid shot changes or shortened shots [8], which could explain the trends we see.

(a) *Lady Killers* - 1960

(b) *Goodfellas* - 1990

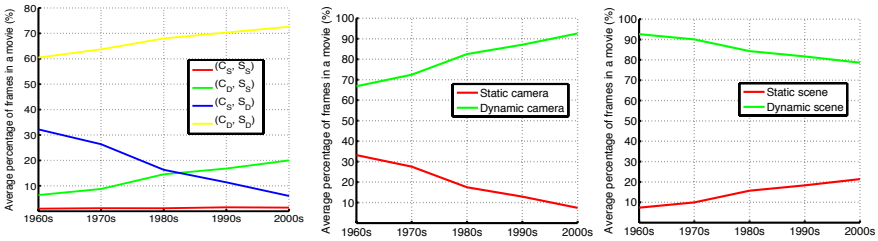
(c) *Charlies Angels* - 2000



(d) User video

(e) Negative results

Fig. 5: Results from the proposed approach (Color code for the frame borders on the last row): (a) A shot from the movie *Lady Killers* where it accurately switches between a static camera to a dynamic camera; (b) The famous 3 min long Copacabana shot from the movie *Goodfellas* is not only identified accurately as a single shot but is also accurately identified as dynamic camera, dynamic scene; (c) A shot from the more recent movie *Charlies Angels* that accurately switches from static scene to dynamic scene; (d) A shot from user video captured using a dynamic camera accurately switches from static scene to dynamic scene and back; (e) Two clips show negative results from our algorithm. While the clips belong to the (C_S, S_D) category, the clips have been incorrectly labeled (C_D, S_S) and (C_D, S_D). The error arises since the dynamic scene has a large spatial extent and occludes the static background providing no evidence of the static camera.



(a) (b) (c)

Fig. 6: Film study: (a) Trends over the years: Plot shows the trends for the four different categories as average percentage of frames of the movie, from 1960s to 2000s. (b-c) Camera and scene motion trends: Plot shows the trends for (b) Static vs. dynamic camera, (c) Static vs. dynamic scene as average percentage of frames of the movie, from 1960s to 2000s.

Time-stamping archive movies. Time-stamping archive data is an exciting new research area. While there has been some prior work in time-stamping archive photographs [26], we consider a novel application of time-stamping archive movies. We use the trends we observe (Fig 6) and learn a regression model using the 4-dim vector to predict the decade of an unknown movie. Using 80% of the movies for training and 20% of the movies for testing, we report the performance after 5-fold cross-validation in Fig 7. We perform significantly better than a random decision and in addition we are almost always $(> 96\%)$ within a decade from the actual time stamp of the movie as seen from the second set of bar graphs. In addition, we perform a human-study with 12 subjects. We provided two movies from each decade to define the task and asked them to label the decade the other movies belong to. We report the performance in Fig 7. Some of the features the users reported they used were age of famous actors, hairstyles, models of cars used, the scene setting, etc. We note that using just motion cues our algorithm performs better than humans who used a number of high-level semantic features.

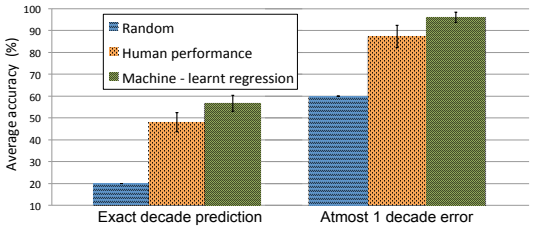


Fig. 7: Time-stamping movies: Learning a regression to estimate the year (decade) using the output of the proposed algorithm significantly outperforms random guessing and human performance (12 subjects). The first set of bars indicate the performance on the exact decade prediction, while the next set indicates the performance with an error tolerance of atmost one decade.

Depth from video. An application of the proposed approach is in aiding depth from video. One of the key constraints for geometric approaches to structure from motion (SFM) and depth from video, is a dynamic camera. Our approach can segment a long video (or movie) into clips based on the camera motion thus segmenting out frames where one can leverage an SFM algorithm. We illustrate depth from video on one of the dynamic camera, static scene clips extracted by the algorithm in the movie *Sound of Music* in Fig 8. We first recover the camera parameters for the frames using structure-

from-motion [27] followed by implementing a simple fronto-parallel plane sweep algorithm. It is worth noting here that without first separating out these clips, one cannot estimate the camera parameters for the whole video. In addition, while most prior works consider the scenario of a dynamic camera looking at a static scene, our proposed algorithm also segments frames within the shot where a dynamic camera is looking at a dynamic scene, which can lead to interesting future extensions to this work.

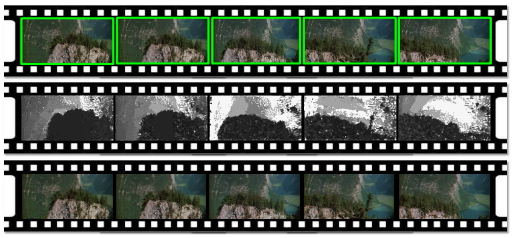


Fig. 8: 2D to 3D video: Row 1 shows a clip from the movie *Sound of music* labeled as (C_D, S_S); Row 2 is the depth map recovered using plane sweep stereo; Row 3 are anaglyph pairs generated using the depthmap (to be viewed using red-cyan glasses).

5 Conclusions and future work

In this paper, we have proposed a learning based video segmentation algorithm for the task of segmenting a video into clips based on scene and camera motion. We showed the effectiveness of the algorithm via quantitative analysis and using an exhaustive collection of movies spanning 50 years, we demonstrated applications such as inferring camera and scene motion trends, archive movie time-stamping and depth from video. We believe that proposed video temporal segmentation finds itself a number of exciting future extensions. For example, our algorithm can tease apart frames where a dynamic object was captured from a dynamic camera, which can be leveraged for applications such as, 3D modeling of a dynamic object, dynamic object co-segmentation, etc.

Acknowledgements. The authors thank Jordan DeLong [8] for collecting and sharing the dataset of movies across decades.

References

1. Cinemetrics: http://www.cinemetrics.lv.
2. S. Bagon. Matlab wrapper for graph cut, December 2006.
3. D. Bordwell. *The Way Hollywood Tells It : Story and Style in Modern Movies*. A Hodder Arnold Publication, 2006.
4. J. S. Boreczky and L. A. Rowe. Comparison of video shot boundary detection techniques. In *SPIE*, 1996.
5. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004.
6. Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *PAMI*, 20(12):1222–1239, 2001.
7. A. B. Chan and N. Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *PAMI*, 30(5):909–926, 2008.
8. J. E. Cutting, J. E. DeLong, and K. L. Brunick. Visual activity in hollywood film: 1935 to 2005 and beyond. *PACA*, 5(2), 2010.

9. D. Dementhon. Spatio-temporal segmentation of video by hierarchical mean shift analysis. In *SMVP*, 2002.
10. C. Dorai and V. Kobla. Extracting motion annotations from mpeg-2 compressed video for hdtv content management applications. In *ICMCS*, 1999.
11. T. Elsaesser and W. Buckland. *Studying Contemporary American Film: A Guide to Movie Analysis*. University of California Press, 2002.
12. C. García Cifuentes, M. Sturzel, F. Jurie, and G. J. Brostow. Motion models that only work sometimes. In *BMVC*, 2012.
13. U. Gargi, R. Kasturi, and S. Antani. Performance characterization and comparison of video indexing algorithms. In *CVPR*, 1998.
14. M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010.
15. A. Hanjalic, R. L. Lagendijk, S. Member, and J. Biemond. Automated high-level movie segmentation for advanced video-retrieval systems. *CSVT*, 1999.
16. R. Jain. Direct computation of the focus of expansion. *PAMI*, 5(1):58–64, 1983.
17. V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.
18. Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. In *ACM SIGGRAPH*, 2005.
19. R. Lienhart. Reliable transition detection in videos: A survey and practitioners guide. *International Journal of Image and Graphics*, 1:469–486, 2001.
20. C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, MIT, 2009.
21. C.-W. Ngo, T.-C. Pong, H.-J. Zhang, and R. T. Chin. Motion characterization by temporal slices analysis. *CVPR*, 2000.
22. K. Otsuji and Y. Tonomura. Projection detecting filter for video cut detection. In *ACM Multimedia*, 1993.
23. Y. peng Tan, D. D. Saur, S. R. Kulkarni, S. Member, and P. J. Ramadge. Rapid estimation of camera motion from compressed video with application to video annotation. *CSVT*, 10:133–146, 2000.
24. Z. Rasheed and M. Shah. Scene detection in hollywood movies and tv shows. In *CVPR*, 2003.
25. B. Salt. Statistical style analysis of motion pictures. *Film Quarterly*, 28(1), 1974.
26. G. Schindler and F. Dellaert. Probabilistic temporal inference on reconstructed 3d scenes. In *CVPR*, 2010.
27. N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH*, 2006.
28. M. V. Srinivasan, S. Venkatesh, and R. Hosie. Qualitative estimation of camera motion parameters from video sequences. *Pattern Recognition*, 30(4):593–606, 1997.
29. J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In *ECCV*, 2004.
30. W. Xiong and J. C.-M. Lee. Efficient scene change detection and camera motion annotation for video classification. *CVIU*, 71(2):166–181, 1998.
31. M. Yeung, B.-L. Yeo, and B. Liu. Segmentation of video by clustering and graph analysis. *CVIU*, 71:94–109, 1998.
32. J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang. A formal study of shot boundary detection. In *TCSVT*, 2007.
33. R. Zabih, J. Miller, and K. Mai. A feature-based algorithm for detecting and classifying scene breaks. In *ACM Multimedia*, 1995.
34. H. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Syst.*, 1(1):10–28, 1993.
35. X. Zhu, A. K. Elmagarmid, X. Xue, L. Wu, and A. C. Catlin. Insightvideo: toward hierarchical video content organization for efficient browsing, summarization and retrieval. *IEEE Transactions on Multimedia*, 7(4):648–666, 2005.