A TWO-LAYER GRAPHICAL MODEL FOR COMBINED VIDEO SHOT AND SCENE BOUNDARY DETECTION

Marc Al-Hames, Stefan Zettl, Frank Wallhoff, Stephan Reiter, Björn Schuller, and Gerhard Rigoll

Technische Universität München
Institute for Human-Machine Communication
Arcisstrasse 16, 80333 München, Germany
{alh, zet, waf, res, sch, rigoll}@mmk.ei.tum.de

ABSTRACT

In this work we present a novel two-layer hybrid Graphical model for combined shot and scene boundary detection in videos. In the first layer of the model, low-level features are used to detect shot boundaries. The shot layer is connected to a higher layer that detects scene or chapter boundaries from semantic features. With this structure, the model optimises the alignment for both layers at the same time and the detection results are interconnected. Experimental results on real video data show, that both layers highly benefit from this sharing of information. Compared to a baseline threshold method with the same features, the F-measure result for the shot detection has been improved by 12.6% absolute. For the scene boundary detection, the result has been improved by more than 11% absolute.

1. INTRODUCTION

Nowadays both terabytes of storage capacity and broadband Internet connections are affordable for the mass market. Todays problem is no longer the storage and sharing of video archives, but the retrieval of the right piece of video. As yet this search is mainly performed through the name of the program, e.g. "Episode No. 1734 of a series". The retrieval could be simplified, if systems would also enable intuitive queries, like "the episode, where J. Roberts played a guest role". However this requires various content information about the program, as persons or the story line. It can either be provided as metadata in the archive, or has to be extracted automatically from the audio-visual stream. The latter is of course preferable, as it avoids cost-intensive and error-prone manual work. Yet, it involves various challenging research topics, like automatic indexing [1, 2, 3], person identification [4, 5, 6], speech recognition [7], understanding [8], and summarisation [9]. In this work we address the first step towards the automatic analysis: finding shot and scene boundaries in videos.

1.1. Video layers and their boundary detection

As shown in Fig. 1, a video can be divided vertically into different layers. The lowest layer is the sequence of frames (usually 25 per second). A sequence of frames continously captured from the same recording source is grouped into shots [1] (several hundreds per hour). Subsequent shots can be connected either through a hard cut or a gradual change (e.g. fade, wipe, or dissolve). In the next layer shots belonging to the same scene are summarised [10] (up to 100 per hour). A scene can be a group of shots at the same place, with the same persons, or with the same topic. Finally a sequence of scenes forms a program, like "Episode 1734 of a series". Depending on the desired degree of granularity, a further chapter layer can be inserted between scenes and program (not shown in Fig. 1), this layer groups blocks of the story, and is usually limited to a few chapters per hour. Each layer can be represented in different ways. Fig. 1 shows some possibilities. Furthermore each layer requires a different analysis level: shots can be detected from low-level features, but program analysis requires semantic knowledge. In general higher layers require more semantic knowledge.

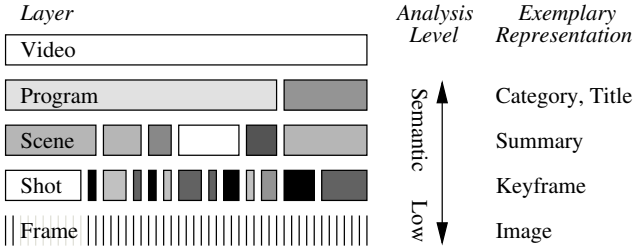

Fig. 1. Video layers with analysis level and possible representations.

In the last years, various works investigated the different layers. Especially shot segmentation has been deeply researched: An introduction and a comparison between different threshold methods gives [2, 11]. Novel methods are e.g. based on SVMs [3]. A standardised evaluation of the different shot boundary detection methods is TRECVID [1]. Shots are clearly defined, and can be detected from the raw visual stream, but they don't form the best retrieval unit: while in a news program the topic can change without a shot boundary, a movie consists of several hundred shots. On the other hand shot boundaries can be used as input for the boundary detection in higher layers. Scenes and chapters form larger blocks of the story, they are much better suited for retrieval tasks. Scene boundary detection and analysis is therefore an increasing research topic. A segmentation of news into scenes shows [12]. Events in sport programs are searched in [13]. The content is structured in [9].

As the different layers require different analysis methods, shots and scenes have mostly been analysed separately. Previous works process the layers sequently and don't include the interaction among them. Recently [10] performed a first approach towards multi-layer techniques with promising results. In this work we therefore bridge the gap between low-level and semantic features: We combine both into a two-layer Graphical model (GM). This GM is then used for combined shot- and scene-boundary detection. Thus it automatically learns and regards the relation between the two layers. Furthermore the model is designed flexible, it can be extended to further layers.

2. LOW-LEVEL FEATURES

Low-level features similar to those described in previous works [2, 11] have been derived. First the intensity $I_t(x, y)$ of each pixel in each frame t has been calculated from the RGB values. Then the average intensity difference for subsequent frames has been derived:

$$I_t(x, y) = 0.3 \cdot R_t(x, y) + 0.59 \cdot G_t(x, y) + 0.11 \cdot B_t(x, y) \quad (1)$$

$$L_t^1 = \frac{\sum_{x,y} |I_t(x, y) - I_{t-1}(x, y)|}{XY} \quad (2)$$

Furthermore the first 15 coefficient $u + v \leq 4$ of the discrete cosine transform (DCT) have been calculated for each frame and then the average frequency intensity difference for subsequent frames:

$$\mathrm{DCT}_t^{u,v} = \sum_{x,y} I_t(x, y) \cos\left(\frac{(2x+1)u\pi}{2M}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right) \quad (3)$$

$$L_t^2 = \frac{\sum_{u,v} |\mathrm{DCT}_t^{u,v} - \mathrm{DCT}_{t-1}^{u,v}|}{UV} \quad (4)$$

We also used colour difference images and colour histogram differences (both described in [2]) as further low-level features. Furthermore we experimented with audio features like MFCCs or frame energy, but found them not helpful, thus we omitted them.

3. SEMANTIC FEATURES

For the scene layer semantic features have been extracted in a semi-automatic fashion. Spoken text subtitles from the data are recognised with a commercial tool. This text has then been mapped to freely available scripts from the Internet. From the scripts the {*the current speaker, the current place, and all persons in the scene*} have been derived automatically and then summarised into a coded semantic feature vector for each frame. As spoken text in the broadcasted series is not perfectly aligned to the scripts, the correct person- and sentence alignment for the data was in average 75%, and therefore comparable to automatic recognisers for speech [7], as well as face [4, 5] and speaker [6] identification.

This approach can be used to derive semantic features for a large set of data with relatively high confidence. It is therefore helpful as a first step towards understanding the meaning. On the other hand, the drawback of this approach is its dependency on scripts, it can't e.g. be used for live broadcasts. However, the module can later easily be exchanged with automatic recognition modules.

Fig. 2 shows the output of the feature extraction and the GM. TV images are superposed with information. In the top the semantic features are displayed. In the bottom, the results of the shot and scene detection (see Sec. 5) are shown with some timing information.

Fig. 2. Frames from two TV series with superposed information.

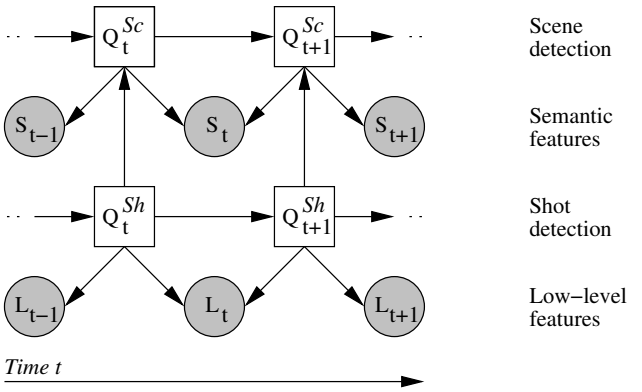

Fig. 3. Two-layer hybrid low-level and semantic feature Graphical model for combined video shot and scene detection.

4. TWO-LAYER HYBRID GRAPHICAL MODEL

A Graphical model (GM) [14] describes statistical dependencies between a set of variables. The variables are marked as nodes and the dependencies between them with edges. Popular statistical models, like the Hidden Markov Model (HMM) or linear dynamical systems (LDS) can be described and combined within this framework.

Fig. 3 shows the novel two-layer hybrid GM for combined shot and scene boundary detection. Hidden variables are printed white, observed variables shadowed. Squares mark discrete probability distributions and circles denote continuous Gaussian nodes $\mathcal{N}(\vec{x}, \vec{\mu}, \Sigma)$. If a variable depends on another, an arrow points towards the conditioned node. A column with the observed feature nodes $\{S_t, L_t\}$ and the hidden discrete state nodes $\{Q_t^{Sh}, Q_t^{Sc}\}$ represents one time slice of the model. Vertical and angular arrows represent dependencies between the variables within one time slice, i.e. the relations between scenes, shots, and the observed low-level and semantic features. A special characteristic of this model is, that the hidden nodes in one time slice are not only connected to the features for this particular frame, but also to the features of the previous frame. Finally, horizontal arrows represent the dependencies between subsequent time slices, i.e. the statistical dependencies of subsequent frames.

Let us first consider the shot detection layer of the GM: It is modelled as a first order Markov chain $\{Q_0^{Sh}, \ldots, Q_t^{Sh}, Q_{t+1}^{Sh}, \ldots\}$. Each node Q_t^{Sh} represents a discrete state and is connected to the low-level features L_{t-1} and L_t of the previous and the current frame. This structure is similar to an HMM. However the connection of each state to two feature nodes is different and especially adapted to the shot detection. Mathematically, the probability of this shot layer, up to the current time step τ can then be expressed as:

$$P_\tau^{Sh} = P(Q_0^{Sh}) \prod_{t=0}^{\tau} P(L_t|Q_t^{Sh}) \prod_{t=1}^{\tau} [P(Q_t^{Sh}|Q_{t-1}^{Sh}) P(L_{t-1}|Q_t^{Sh})], \quad (5)$$

where $P(Q_0^{Sh})$ represents the probability of a shot in the first frame (if not embedded into a further system a sequence always starts with a shot). The low-level inputs, given the current state are modelled as $P(L_{(\cdot)}|Q_{(\cdot)}^{Sh})$. The state transition, i.e. the probability of subsequent shots, is represented by $P(Q_t^{Sh}|Q_{t-1}^{Sh})$. This shot detection layer can be used independently of the remaining model. The parameters of the input nodes and the state transitions can be trained with the EM algorithm [15] and then applied to shot detection without scenes. However here it is only used in combination with the scene layer.

The structure of the scene detection layer is similar to the shot detection: The discrete states Q_t^{Sc} for the scene layer are connected to the semantic features for the current and the previous frame S_{t-1} and S_t. Yet, the Markov chain $\{Q_0^{Sc}, \ldots, Q_t^{Sc}, Q_{t+1}^{Sc}, \ldots\}$ is now not only conditioned on the previous state, but also conditioned on the current state Q_t^{Sh} of the shot layer. Thus the results in the shot detection interact with the scene layer. The scene layer up to the current time step τ can then be expressed as:

$$P_\tau^{Sc} = P(Q_0^{Sc}|Q_0^{Sh}) \prod_{t=0}^{\tau} P(S_t|Q_t^{Sc}) \prod_{t=1}^{\tau} [P(Q_t^{Sc}|Q_{t-1}^{Sc}, Q_t^{Sh}) P(S_{t-1}|Q_t^{Sc})], \quad (6)$$

where $P(Q_0^{Sc}|Q_0^{Sh})$ is the probability of a new scene in the first frame, given a new shot (again, if not embedded into a higher system a sequence will of course always start with a new scene). The semantic inputs, given the current state of the scene Markov chain are modelled as $P(S_{(\cdot)}|Q_{(\cdot)}^{Sc})$. Finally, the state transition of the scenes is now also conditioned on the current state of the shot layer, thus it is represented by $P(Q_t^{Sc}|Q_{t-1}^{Sc}, Q_t^{Sh})$.

One of the main advantages of GMs are standardised algorithms. Given the developed, novel structure, the GM can be implemented with a toolbox, like the Bayes Net Toolbox [16]. We also experimented to include the low-level features directly to the scene layer (graphically this basically means adding an extra edge between Q_t^{Sc} and L_t), but didn't found it helpful, as the information about the low-level input is already implicitly modelled in the shot nodes. The presented GM structure is loosely related to two-layer HMMs. However the feature connection is significantly different. Furthermore both decoding and training are performed differently.

4.1. Decoding

With the scene and the shot layer, the probability for the complete model at the current time step τ becomes $P(\tau) = P_\tau^{Sh} \cdot P_\tau^{Sc}$. However, for efficency reasons the probabilities of the different state sequences in the model are not calculated directly with Eq. (5) and (6), but marginalisation and inference is performed with a junction tree algorithm for Graphical models [17]. The decoding itself is then performed frame by frame, by marginalisation over the low-level and semantic input for each frame, and then maximisation in the hidden nodes: $\mathrm{argmax}\{Q_t^{Sh}, Q_t^{Sc}\}$. Thereby the result for each frame can take four different values: a shot and a scene boundary; a shot without a scene boundary; a scene without a shot boundary; or neither a shot nor a scene boundary. An advantage of this strategy is, that it can be performed online, with a delay of only one frame. Furthermore the decoding jointly optimises the alignment for both layers.

4.2. Training

In principle, two types of training from a training data set are possible: First the shot layer can be trained independently in a supervised fashion with the low-level features and known shot boundaries. Then the scene layer is trained with the semantic features, known scene boundaries, and the results of the shot layer. This is done with the EM algorithm [15]. As a second training strategy both layers could be trained at the same time with unknown shot boundaries. This however leads to an unsupervised feature decomposition of the shot layer. Then the shot layer has no direct relation to shots anymore. It couldn't be used for shot detection, and only the scene output would be optimised. We therefore only applied the first strategy.

5. EXPERIMENTS

To evaluate the novel GM, we compared it in both layers to a standard thresholding method, as e.g. introduced in [2]. As data we used six episodes of different, broadcasted series from different genres. Together the data has a length of approximately 4 hours (360'000 frames) and contains nearly 2000 shots and 90 manually annotated scenes. Both the baseline thresholding method and the GM used the same set of features, as described in Sec. 2 and 3.

To compare the approaches, we used three information retrieval measures [18]: Recall, is the proportion of correct retrievals compared to all possible correct retrievals. Precision, is the proportion of correct retrievals among all retrieval results. The F_1-Measure summarises both into one number:

$$r = \frac{\text{correct}}{\text{correct} + \text{missed}}, \quad p = \frac{\text{correct}}{\text{correct} + \text{false}}, \quad F_1(r,p) = \frac{2rp}{r+p}.$$

A shot boundary can be determined frame exact, but this is not practical for scene or chapter boundaries. If a scene change is not accompanied by a shot, the scene can shift by a couple of seconds. It is not useful to evaluate scene and chapter boundaries on a frame basis. For these results we introduced an offset in seconds. If a boundary is detected within this offset, it is considered a correct retrieval.

Four experiments were performed: shot boundary detection with two different types of training, scene boundary recognition, and in a further experiment we replaced the scene layer by a chapter layer.

5.1. Shot boundary detection

For the evaluation of the shot boundary layer, we performed experiments with two different training setups. In the first setup (*similar*) we used the first part of each episode for training and the remaining unknown parts of the episodes for tests. In the second setup we used only one complete episode for training, and the remaining unknown series for tests. This is a much more *realistic* scenario, as the test data can then be completely different compared to the training data.

	Graphical Model (in %)			*Threshold (in %)*		
Training	*r*	*p*	F_1	*r*	*p*	F_1
Similar	98.5	91.8	95.0	96.7	87.6	92.0
Realistic	94.8	88.0	91.3	84.8	73.5	78.7

Table 1. Shot boundary recall (r), precision (p), and F-measure (F_1) results for the Graphical model and for a baseline threshold method.

The results are shown in Tab. 1. As expected, the similar scenario has better rates compared to the realistic scenario for both methods. The GM significantly outperforms the baseline method for both scenarios in recall, precision, and F_1-measure results. It shall be mentioned, that state of the art approaches for shot detection reach better results [1]. However, these approaches use much more advanced low-level features. Here, the GM and the baseline method use the same set of features, this shows that the shot layer of the GM benefits highly from the joint optimisation of scenes and shots.

5.2. Scene boundary detection

The results for the scene boundary detection for the baseline method and the GM are shown in Tab 2. The GM highly outperforms the threshold approach continously. With increasing allowed offset all results improve. For an offset of +/- 20 seconds, the threshold approach reaches an F_1-measure of 53.8%, where the GM reaches

	Graphical Model (in %)			*Threshold (in %)*		
Offset	*r*	*p*	F_1	*r*	*p*	F_1
+/- 2s	44.3	63.1	52.0	7.7	10.5	8.8
+/- 5s	44.3	63.1	52.0	10.9	14.1	12.3
+/- 10s	49.1	70.2	57.8	23.2	29.2	25.9
+/- 20s	55.8	77.8	65.0	47.8	61.5	53.8

Table 2. Scene boundary recall (r), precision (p), and F_1-measure.

65.0%. However an offset of more than 10 seconds doesn't seem reasonable for scenes. For a more reasonable offset of 10 seconds, the GM outperforms the threshold by 31.9% absolute. Yet, the F_1 result of 57.8% for an offset of 10 seconds is still too low for a real application. For a better result, the model requires more advanced semantic features, like speech understanding. However, compared to the standard threshold method, the GM is highly preferable. The information sharing between low-level and semantic features through the hidden nodes contributes to significant better recognition rates.

5.3. Chapter boundary detection

In the last experiment we used the same GM with the same set of features, but replaced the scene by a chapter layer, both for training and decoding. Chapters form a very large group in a video stream. In an episode, there are significant less chapters than scenes. They are much harder to find and usually require to understand the meaning.

	Graphical Model (in %)			*Threshold (in %)*		
Offset	*r*	*p*	F_1	*r*	*p*	F_1
+/- 2s	56.3	27.4	36.8	12.5	5.0	7.2
+/- 5s	62.5	30.2	40.7	12.5	5.0	7.2
+/- 10s	62.5	30.2	40.7	29.2	11.3	16.2
+/- 20s	75.0	37.3	49.8	54.2	21.3	30.6

Table 3. Chapter boundary recall (r), precision (p), and F_1-measure.

The results for the chapter detection are shown in Tab. 3. As expected the F_1 results are continously worse compared to the scenes. However again the GM outperforms the threshold approach clearly. Furthermore, given the limited set of semantic features, the F_1 result of 49.8% for chapters within a 20 second offset is very promising.

6. CONCLUSION

In this work we proposed a novel Graphical model for combined shot and scene boundary detection in videos. The model integrates both low-level and semantic features into one model and optimises the alignment for shot and scene boundaries jointly. In an experimental section we compared the model to a thresholding method. The model outperforms the standard single thresholding methods: For the shot detection, the F-measure has been improved by 12.6% absolute. For the scenes, the result has been improved by more than 11% absolute. Both layers benefit from the joint optimisation.

The model is designed flexible, we plan to extend it to further video layers (e.g. combined shot/scene/chapter recognition). A current draw-back are the relative simplistic input features. In the future we plan to extend both low-level and semantic features. Especially in the semantic domain we like to exchange the current feature extraction by an automatic person- and speaker recognition and furthermore add speech recognition and interpretation parts to further improve the recognition rates in the scene and chapter layer.

7. ACKNOWLEDGEMENT

This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811).

8. REFERENCES

[1] A.F. Smeaton, P. Over, and W. Kraaij, "TRECVID: evaluating the effectiveness of information retrieval tasks on digital video," in *Proc. ACM Multimedia*, 2004.

[2] J.S. Boreczky and L.A. Rowe, "Comparison of video shot boundary detection techniques," in *Proc. Storage and Retrieval for Image and Video Databases*, 1996.

[3] K. Hoashi, M. Sugano, M. Naito, K. Matsumoto, and F. Sugaya, "Video story segmentation and its application to personal video recorders," in *Proc. CIVR*, 2005.

[4] G. Aggarwal, A. Roy-Chowdhury, and R. Chellappa, "A system identification approach for video-based face recognition," in *Proc. IEEE ICPR*, 2004.

[5] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.

[6] J.P. Campbell, "Speaker recognition: a tutorial," *Proc. of the IEEE*, vol. 85, no. 9, 1997.

[7] G. Evermann, H.Y. Chan, M.J.F. Gales, B. Jia, D. Mrva, P.C. Woodland, and K. Yu, "Training LVCSR systems on thousands of hours of data," in *Proc. IEEE ICASSP*, 2005.

[8] Y.-Y. Wang, L. Deng, and A. Acero, "Spoken language understanding," *IEEE Signal Processing Magazine*, vol. 22, no. 5, 2005.

[9] A. Salway, A. Vassiliou, and K. Ahmad, "What happens in films," in *Proc. IEEE ICME*, 2005.

[10] J. Adcock, M. Cooper, A. Girgensohn, and L. Wilcox, "Interactive video search using multilevel indexing," in *Proc. CIVR*, 2005.

[11] J. Nesvadba, F. Ernst, J. Perhavc, J. Benois-Pineau, and L. Primaux, "Comparison of shot boundary detectors," in *Proc. IEEE ICME*, 2005.

[12] L. Chaisorn, T.-S. Chua, and C.-H. Lee, "The segmentation of news video into story units," in *Proc. IEEE ICME*, 2002.

[13] M. Bernard and J.-M. Odobez, "Sports event recognition using layered HMMs," in *Proc. IEEE ICME*, 2005.

[14] M.I. Jordan, Ed., *Learning in Graphical Models*, MIT Press, 1998.

[15] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.

[16] K. Murphy, "The Bayes Net Toolbox for Matlab," *Computing Science and Statistics*, vol. 33, 2001.

[17] R. Cowell, "Introduction to inference for Bayesian networks," in *Learning in Graphical Models*, M.I. Jordan, Ed. 1998, MIT Press.

[18] C.J. van Rijsbergen, *Information Retrieval*, Butterworths, London, UK, 1979.