# Label-Based Automatic Alignment of Video with Narrative Sentences

Pelin Dogan[1], Markus Gross[1], and Jean-Charles Bazin[2(✉)]

[1] Department of Computer Science, ETHZ, Zurich, Switzerland
{pelin.dogan,grossm}@inf.ethz.ch
[2] Disney Research, Zurich, Switzerland
jean-charles.bazin@disneyresearch.com

**Abstract.** In this paper we consider videos (e.g. Hollywood movies) and their accompanying natural language descriptions in the form of narrative sentences (e.g. movie scripts without timestamps). We propose a method for temporally aligning the video frames with the sentences using both visual and textual information, which provides automatic timestamps for each narrative sentence. We compute the similarity between both types of information using vectorial descriptors and propose to cast this alignment task as a matching problem that we solve via dynamic programming. Our approach is simple to implement, highly efficient and does not require the presence of frequent dialogues, subtitles, and character face recognition. Experiments on various movies demonstrate that our method can successfully align the movie script sentences with the video frames of movies.

## 1 Introduction

Audio description consists of an audio narration track where the narrator describes what is happening in the video. It allows visually impaired people to follow movies or other types of videos. However the number of movies that provide it is considerably low, and its preparation is particularly time consuming. On the other hand, scripts of numerous movies are available online although they generally are plain text sentences. Our goal is to temporally align the script sentences to the corresponding shots in the video, i.e. obtain the timing information of each sentence. These sentences can then be converted to audio description by an automatic speech synthesizer or can be read by a human describer. This would provide a wider range of movies to visually impaired people.

Several additional applications could benefit from the alignment of video with text. For example, the resulting correspondences of video frames and sentences can be used to improve image/video understanding and automatic caption generation by forming a learning corpus. Video-text alignment also enables text-based video retrieval since searching for a part of the video could be achieved via a simple text search.

In this paper, we address temporal alignment of video frames with their descriptive sentences to obtain precise timestamps of the sentences with minimal manual intervention. A representative result is shown in Fig. 1. The videos


G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part I, LNCS 9913, pp. 605–620, 2016.
DOI: 10.1007/978-3-319-46604-0_43

are typically movies or some parts of movies with duration of 10 to 20 min. We do not assume any presegmentation or shot threading of the video. We start by obtaining the high-level labels of the video frames (e.g. "car", "walking", "street") with deep learning techniques [12] and use these labels to group the video frames into semantic shots. In this way, each shot contains relatively different semantics knowing that the information given by the different sentences is relatively different. Then we formulate the problem of text-video alignment as sentence-shot alignment by finding similarity between the high-level labels in the shots and the words of the sentences. This similarity is computed using the vectorial features of words and word-to-word distances. Our final alignment is formulated in a graph based approach computing the minimum distance path from the first to the last sentence-shot pair. The main contributions of our paper are:

- We align human-written sentences (such as scripts and audio description texts) with the complete set of shots that constitutes the video. Our approach does not require dialogues, subtitles, and face recognition. Our approach directly works on the raw video, i.e. no presegmentation or cut is needed.
- We automatically segment the input video into shots by using frame based high-level semantics so that a semantic change in a continuous camera shot can be detected. We refer this semantic change as *semantic cut* through the paper. We also introduce a refinement process to optimize the semantic cuts so that they tend to correspond to one sentence each.
- We introduce a novel dataset of script sentence alignments of various video sequences which are publicly available on the project page.

## 2 Related Work

Our goal is to temporally align a video with its script, which will provide a timestamp for each sentence in the script. In the following, we describe the related work on steps that will lead us towards this goal.

**Image/video description.** In the last years there has been an increasing interest in object/scene recognition, object labeling, and automatic caption generation, in part thanks to the availability of new datasets [4,29,30]. With the recent developments in deep convolutional architectures, large-scale visual recognition by convolutional neural networks [12] and recurrent neural networks [6] has achieved state-of-the-art results by surpassing the conventional methods. Moreover, impressive results have been obtained for describing images and videos with natural language [6,8,9,13,18,22,24]. These approaches rely on a corpus with strong annotations. The main reason why the current video description performance is falling behind the image description is the lack of large annotated video corpus that would provide better learning and understanding. Since the manual annotation of videos with text is very time consuming, automatic alignment methods of video-text pairs would increase the size of the corpus and the variety of the content.

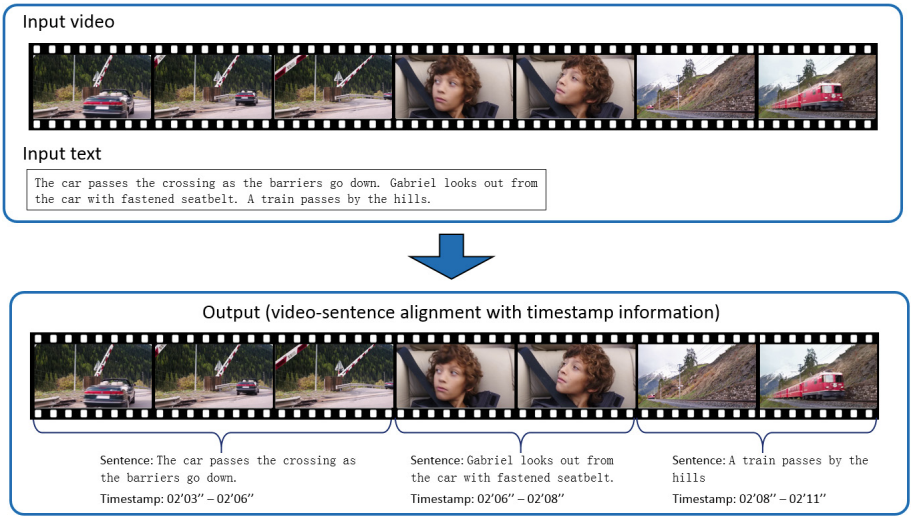


**Fig. 1.** Given an input movie and associated narrative sentences (e.g. from the movie script), our approach temporally aligns the video frames with the sentences and provides the timestamp information of the sentences. This figure illustrates a representative result for a 10-minutes long continuous video from the movie *Lucid Dreams of Gabriel*. For a better visibility of the figure, only a 8-seconds segment is shown here.

**Text similarity.** The recent developments of the deep neural networks improved the natural language processing tasks, with applications in information retrieval and artificial intelligence. The computation of the similarity and relation between words is an important step towards video-text alignment, for example to compute the similarity between the high-level labels of the video frames (e.g. "car", "walking") and the words of the script sentences(e.g. "She opens the car door"). Some approaches use either using a thesaurus or statistics from a large text corpus, or use both to compute word similarity [1,11,16,19,20]. In our work, we will use the approach of Pennington et al. [20]: they propose an unsupervised learning algorithm to obtain vector representations for words based on word-word co-occurrence statistics from a text corpus. Their vector representation demonstrates superior results in finding the semantic distance between words.

**Shot segmentation.** Aligning sentences with the corresponding video parts requires shot detection and shot segmentation. For this, many of the automated shot-change detection methods use color histograms [7,10,15,17] or visual descriptors [2,14,21]. These are mostly successful for shots that are bounded by camera cuts or abrupt visual transitions. In the context of video-text alignment, distinguishing a semantic change through a single camera shot is valuable because a semantic change in the video is usually associated to a new description sentence within the script. Therefore we are using semantic features, namely

high-level labels, to segment the full video into "semantic shots" and in turn match the sentences with them.

**Alignment.** Tapaswi et al. [28] align book chapters with video scenes using dialogues and character identities as cues with a graph based algorithm. Their alignment requires face recognition in the video frames and presence of numerous dialogues which may fail in case the movie does not have these densely. Sankar et al. [26] align the scripts to TV videos/movies using location, face and speech recognition. However this success of the method is mostly limited to TV series, since it needs pre-training of the frequent locations within the video to divide the scenes. Bojanowski et al. [3] propose a method for aligning video clips to sentences given the vectorial features for both video and text. They require the segmentation of the video unlike us and solve the problem using a conditional gradient algorithm with the strong assumption that every video clip is corresponding to exactly one sentence. Instead, we segment the video jointly while performing alignment so that we do not require such strong assumption. Tapaswi et al. [27] present an approach to align plot synopses with the corresponding shots with the guidance of subtitles and character identities using dynamic programming. They require the extraction of character information both in textual and visual forms by using face detector and tracker. Rohrbach et al. [23] provide a dataset that contains transcribed descriptive video service, which is temporally aligned to full length movies. Their dataset provides video sentences with timestamps that match the video snippets. In contrast to our automatic method, they perform the fine alignment of the sentences manually which is significantly time consuming. Zhu et al. [31] aim to align books to their movie releases by using visual correspondences as well as the correspondences between the dialogues and the subtitles, which may fail for videos with very limited dialogues. Moreover their alignment is at a coarser level: they aim to match book chapters with TV episodes. In contrast, we aim for precise (frame-level) timestamps for sentences and shots.

## 3 Proposed Approach

In this section, we present our approach for aligning a video with its narrative sentences, which results in a timestamp for each sentence. To have an accurate alignment, the text input should provide at least one sentence for each shot in the movie. By the term *shot* we refer to a series of frames that runs for an uninterrupted period of time with the same semantics, not necessarily defined by camera cuts. An example of text input for our algorithm can be a movie script (dialogues not required). Another example would be a transcribed audio description of the movie containing rich descriptions for visually impaired people. We assume that the sentences are in the same temporal order as the movie, like movie scripts and audio descriptions. Our approach is designed for videos having a dynamic plot with different scenes and actions as in the typical Hollywood movies. A counter-example is a biographical documentary film, such as

an interview, where a person speaks to the camera during the whole duration of the video, i.e. without any changes of scene or action.

### 3.1 Overview

We first obtain the high-level labels for all the video frames in the form of words, as well as their confidence scores, using deep learning techniques [12]. Then we smooth these through the time domain to obtain temporal coherency. The temporally coherent results are used to detect the semantic changes in the video, which corresponds to the beginnings and ends of the shots. Then the labels and their confidence scores of the frames of each detected shot are grouped together to represent the shots. We then calculate a similarity score for each shot-sentence pair using the labels from the shot and the sentence words. This provides a cost matrix and we then compute the minimum distance path assuming the matching of the first and last sentence-shot pairs are given. The nodes of the calculated path provides the matching of the sentence-shot pairs. This results in the annotation of each input sentence with the timestamp of the matched shot.

### 3.2 High-Level Features and Temporal Coherency

We start by obtaining the high-level features (labels) of each frame of the input video. Each video frame is processed independently and thus can be processed in parallel. These high-level labels are in the form of text words and typically refer to object (e.g. "car"), scene (e.g. "street") or action (e.g. "walking") visible in the video frame. We automatically obtain these labels, as well as their confidence score, by the deep learning based cloud service Clarifai[1] or Caffe framework [12] with pretrained models for its CNN architecture. As a result, for each video frame $i$ we obtain a feature vector $w_i$ whose number of entries is the total number of labels (around 1000) and the entry values are the confidence scores for the label corresponding to that entry index. A representative result vector for a frame from the movie *Lucid Dreams of Gabriel* is shown in Fig. 2.

By concatenating these column vectors $w_i$ over time, we obtain a matrix $\mathbf{W}$ containing the confidence scores of the labels through time. A representative example is shown in Fig. 3-top. Each row of this matrix represents the scores of the label corresponding to that row index (e.g. "car") through time. If the entries of this row are all zero or very small, it means the corresponding label is not seen in the frames, e.g. no "car" object is visible in the entire video. The values in the matrix rows are noisy due to motion blur, occlusions, lighting change, and all the effects that decrease the performance of the automatic object/scene recognition tools. Therefore the obtained matrix requires smoothing in the temporal domain (x-axis) to provide temporal coherency between the neighboring frames. We aim to find the labels that have high confidence scores while eliminating the labels that are not temporally consistent. We find the labels by a graph based shortest

[1] https://www.clarifai.com/.


**Fig. 2.** Representative example of high-level labels and their confidence scores given an input video frame. Top: the input frame $i$ from the movie *Lucid Dreams of Gabriel*. Bottom: the top 10 labels (in terms of confidence, out of 1000) and their confidence scores. The full confidence score vector (over all the labels) at frame $i$ is written $w_i$.

path approach. We empirically set $N$ to 10 and observed that higher values did not significantly change the final alignment results. We refer to the set of labels, one per frame through time, as a "path" $q$ through the cost matrix, and our aim is to find the $N$ shortest paths which will give us the $N$ most dominant and temporally coherent labels for each frame. For this, we apply a shortest path algorithm $N$ times in the following way. To find the first shortest path $q_1$, we consider the matrix $\mathbf{W}$ as a directed graph where the nodes are each $<$*frame,label*$>$ pair and the edges are defined using the entries of the matrix $\mathbf{W}$ (see Fig. 3). The weight of the edge from node $(i, l)$ to node $(i', l')$ is defined as

$$\phi\left((i,l),(i',l')\right) = \begin{cases} \lambda(1 - w_{i'}(l')) + \varphi(l,l') & \text{if } i' = i+1 \\ \infty & \text{else} \end{cases} \tag{1}$$

where $\varphi(l, l')$ returns 1 when $l \neq l'$ and 0 otherwise, and where $w_i(l)$ is the score of the label indexed by $l$ at frame $i$, i.e. node $(i, l)$. The scaling factor $\lambda$ sets the desired smoothness by penalizing the change of the label through the path and we set it to $\lambda = \frac{framerate}{10}$, where $framerate$ is the frame rate of the input video (usually 24 fps). We apply Dijkstra's algorithm [5] to obtain the minimum distance path solution. After finding the first path, we remove the edges pointing to the nodes of the calculated path so that those nodes cannot be selected for the future paths. We repeat this procedure to find the $N$ shortest paths, that is to say the $N$ most dominant labels. After the calculation of paths $q_1, ..., q_N$, the scores of the labels on the paths are smoothed with weighted moving average filter. A resulting temporally coherent matrix can be seen in Fig. 3-bottom. For writing simplicity, we still name this processed matrix as $\mathbf{W}$.

### 3.3 Shot Segmentation

So far, we explained how to obtain the temporally coherent labels and scores per frame stored in $\mathbf{W}$. We now aim to segment the whole input video into

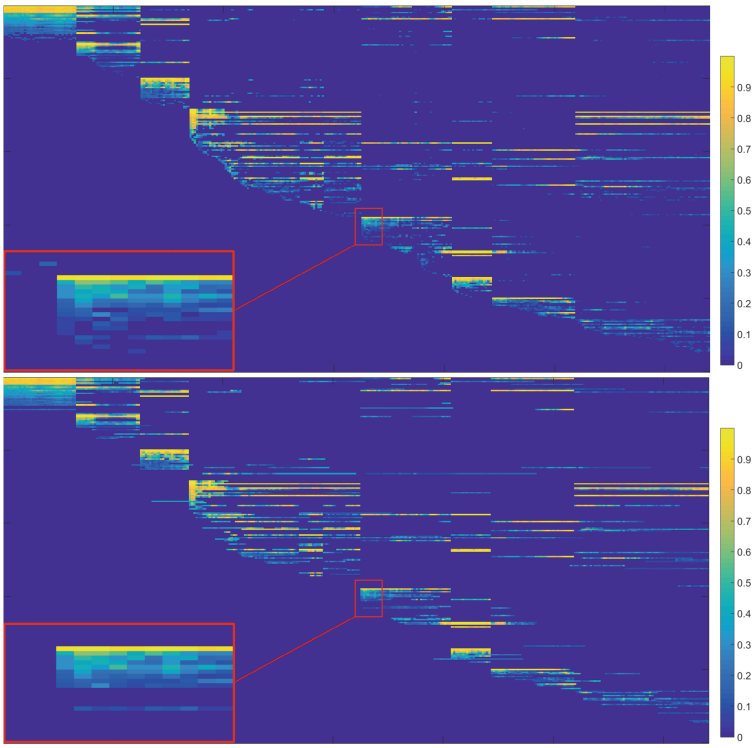


**Fig. 3.** Top: concatenated label vectors $w_i$. The height of this matrix depends on the number of unique labels detected through the whole video. Bottom: Temporally coherent result after path calculations where noisy labels are removed or smoothed.

shots by processing the matrix $\mathbf{W}$. For a frame to be the beginning of a new shot, it has to be different than the past neighboring frame and similar to the future neighboring frame. Since we already have applied temporal filtering, the scores in $\mathbf{W}$ carry temporal information from neighborhood, not just from the surrounding frames. We calculate a score $S_i$ that represents the score of frame $i$ to be the beginning of a new shot:

$$S_i = |D_C(w_i, w_{i-1})|(1 - |D_C(w_i, w_{i+1})|) \tag{2}$$

where $w_i$ is the vector of label scores of the frame $i$ and $D_C$ computes the cosine distance of the input vectors.

Then we find the top $K$ local maxima among all $S_i$, where $K$ is the number of sentences in the input text. The frames corresponding to these maxima are our initial shot beginnings. It is important to note that we do not define *shots* by camera cuts. As discussed earlier, we refer to "shot" as a sequence of consecutive frames that have a similar semantic. Other than camera cuts, semantic cuts are considered as "shots" as well. For example, a continuous panning shot might have two different semantics with a soft border around the middle of the pan.

This panning shot needs to be segmented into two shots since there might be two different sentences describing it due to semantic change. Therefore our aim is not finding the camera cuts, but optimizing (and thus detecting) the semantic cuts -including camera cuts- that will match the sentences in the best way.

### 3.4 Alignment

**Cost matrix.** In the previous sections, we have automatically segmented the input video into shots according to their semantic contents and their smoothed features. As the basis of our method, we need a robust estimate of the alignment quality for all the shot-sentence pairs. We observe that a shot and a sentence are more likely to be alignable together if the words in this sentence and the labels of this shot are semantically similar. Using this concept, we compute a similarity value $v_{ij}$ between each shot $i$ and sentence $j$. Subsequently, we transform these values into a cost matrix $\mathbf{C} \in \mathbb{R}^{K \times K}$, in which each entry $c_{ij}$ specifies a cost for aligning a pair of shot and sentence.

We represent the shot labels and the sentence words using GloVe word vector descriptors [20] of dimension $b = 300$. For each detected shot, we consider the set of all the $N$ labels and scores found in all the frames of the shot. We denote the $l$-th label of the $i$-th shot by its confidence score $f_i(l)$ and its GloVe vector descriptor $d_i(l) \in \mathbb{R}^b$ where $l \in [1...N]$. Similarly, we denote the $m$-th word of the $j$-th sentence with its GloVe descriptor $d_j(m) \in \mathbb{R}^b$. The similarity between the label $l$ and the word with index $(j, m)$ is calculated as

$$z_{ij}(l, m) = |d_i(l) - d_j(m)| \tag{3}$$

which is modified by Lorentzian stopping function as

$$y_{ij}(l, m) = \left(1 + \left|\frac{z_{ij}(l, m)}{\sigma}\right|^{\alpha}\right)^{-1} \tag{4}$$

where $\alpha = 3$ and $\sigma = 0.5$ for all the experiments shown in this paper.

Finally the similarity values $y_{ij}(l, m)$ are used to compute the cost matrix $\mathbf{C}$ in which low values indicate shot-sentence pairs that are likely to be a good match. The entries of the cost matrix $\mathbf{C}'$ are computed as

$$c'_{ij} = 1 - \frac{1}{M} \sum_{m=1}^{M} f_i(l) \max_{l \in N} y_{ij}(l, m) \tag{5}$$

Lastly, we obtain the values of the cost matrix $\mathbf{C}$ by scaling the values of $\mathbf{C}'$ with an oriented 2D Gaussian factor which penalizes the elements in the upper right and lower left corner. In this way we incorporate the global likelihood of being at any node in the graph to our cost matrix considering passing through the nodes at the top-right or bottom-left corners are very unlikely.

$$c_{ij} = c'_{ij} \exp\left(-\frac{(i - j)^2}{2K^2}\right) \tag{6}$$

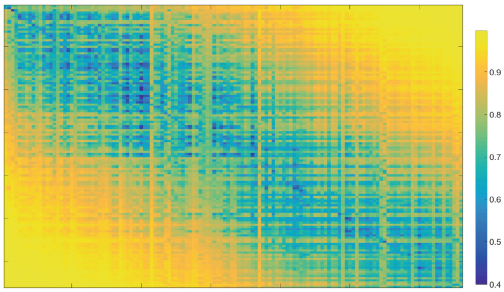


**Fig. 4.** Cost matrix whose elements $c_{ij}$ are computed using similarity score between shot $i$ (y-axis) and sentence $j$ (x-axis).

An example of cost matrix for each pair of sentences and computed shots is available in Fig. 4.

**Path calculation.** So far we have described mappings between the shots and sentences. We now explain how to find a discrete mapping $p : \mathbb{R} \rightarrow \mathbb{R}^2$ in our cost matrix: for a time $t$, $p(t) = (i, j)$ means that the shot $i$ corresponds to the sentence $j$. We refer to the discrete representation of a mapping $p$ as a path through the cost matrix $\mathbf{C}$, and consider a graph based solution to find the minimum distance path. This path will provide the optimum shot-sentence pairings. We compute the cost of a path $p$ as the average of all the entries in the cost matrix that the path goes through:

$$\psi(p) = \frac{1}{T} \sum_{t=1}^{T} \mathbf{C}(p(t)) \tag{7}$$

where $T$ denotes the number of steps in the path.

To find the path with minimum cost, we consider the cost matrix as a directed graph where a path is defined as the set of connected nodes. We identify a node by its position $(i, j)$ and edge as an ordered pair of nodes. Since we assume the input text sentences are in the same temporal order as the video, we only allow forward motion. In other words each node $(i, j)$ is connected to its three neighbors $(i, j + 1)$, $(i + 1, j + 1)$, and $(i + 1, j)$. The weight of each edge is the value of the cost matrix at the node that the edge points to. An example graph of the possible connections is shown in Fig. 5.

We use dynamic programming to find the minimum distance path [25]. Computing the shortest path from the first node $(1, 1)$ to the last node $(K, K)$ provides us the initial result for the shot-sentence pairings. An alignment result is shown in Fig. 6. The pink plot on the graph represents the ground truth alignment. The black plot shows the regions where our result is different than the ground truth. It is important to note that the y-axis represents the frames, not the shots. This is why paths have discrete vertical parts which corresponds to the set of frames corresponding to a shot.

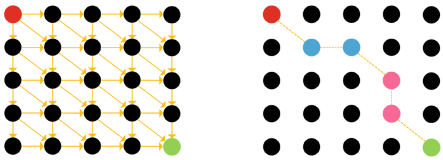

**Fig. 5.** Left: Possible oriented connections (in orange) between the nodes where the red node is considered the source and the green node is the sink. Right: An example path result from the source to the sink. (See text for details.) (Color figure online)

**Refinement.** As mentioned earlier, the sentences in the input text description do not have to correspond to the camera cuts. In addition, the result of the shot segmentation does not have to give the perfect shots for the sentences. This may cause the matching of a shot with more than one sentence (horizontal parts in the path) or matching of a sentence with more than one shot (vertical parts in the path). Therefore, the alignment that is obtained by the current cost matrix may not be the optimum.

We compute the optimum alignment by modifying the cost matrix in an iterative refinement procedure. Starting with the current optimum path, we combine the shots that are matched to the same sentence into a single shot. Conversely we segment the shot that is assigned to more than one sentence for another round. The segmentation of this shot is conducted in a way similar to Sect. 3.3. We find $r-1$ local maxima among $S_i$ in Eq. 2 in the corresponding region of frames during this shot, where $r$ is the number of resulting sentences matched with it. In this way we obtain $r$ shots that can be assigned to these $r$ different sentences.

For example, the shots corresponding to the pink nodes (same column) on the path in Fig. 5 will be combined together, while the shot corresponding to the blue nodes (same row) will be split into two shots. After this refinement, we repeat all the steps starting from Sect. 3.4 to find the new optimal path. In our experiments, we observed that the result converges in less than 4 iterations. The effect of this refinement step is shown in the cost matrices of Fig. 6.

## 4 Applications

In this section we demonstrate different applications and the results obtained by our algorithm. Please refer to our project webpage for video results.

**Video-sentence alignment.** Aligning descriptive sentences to video in an automatic way can provide rich datasets for modeling video contents. The resulting video-sentence alignments can be used as training data to learn models for the task of automatic generation of video descriptions. An example of video-sentence alignment obtained by our algorithm is available in Fig. 7. It shows two consecutive shots separated by a sharp camera cut and the automatic alignment of the corresponding sentences. The sentences are marked automatically by the

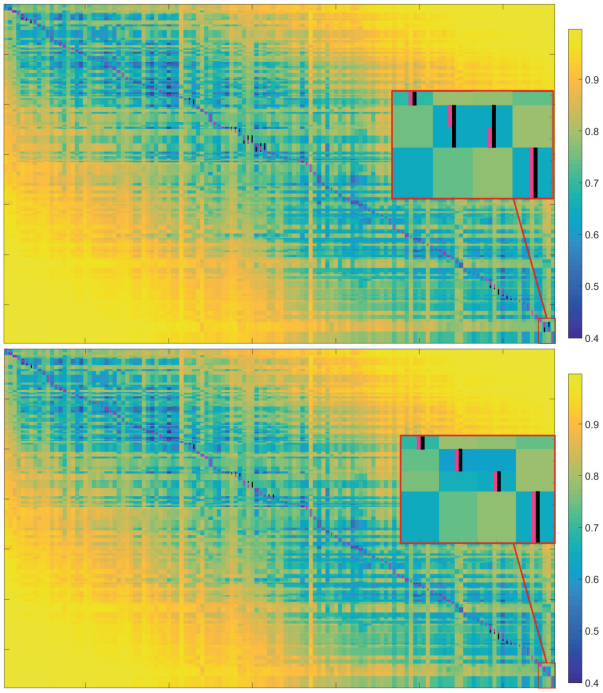


**Fig. 6.** The alignment result automatically obtained by our approach for the full movie (11 min) *Lucid Dreams of Gabriel* with its audio description sentences. Top: The initial alignment result using the initial shot segmentation results. The alignment of a shot to two consecutive sentences is seen in the close-up view (red box). Bottom: Our final alignment result after the refinement process. The close-up view shows that our result exactly matches with the ground truth alignment. (Color figure online)

timestamps that correspond to the very first frame of the shots by our algorithm since the beginning of these shots are captured perfectly.

**Shot segmentation.** Shot segmentation is used to split up a video into basic temporal units called shots. These units have consecutive frames taken contiguously by a single camera, representing a continuous action in time and space. Shot segmentation is an important step for various tasks such as automated indexing, content-based video retrieval and video summarization. While detecting sharp camera cuts is not a difficult task (as shown in Fig. 7 for a sharp camera cut), detecting only the camera cuts may not be sufficient for video-text alignment or other video retrieval tasks. The target material can have different types of separation. For example two sentences can take place in the same scene with a continuous camera shot while representing two different semantic information. A representative example of such a case is shown in Fig. 8 where the shot starts by a woman getting into the car and ends with a child having a chocolate bar. Although this scene is shot continuously by a panning camera


**Fig. 7.** Two consecutive shots (one frame of each shot is shown here) separated by a sharp camera cut and their aligned sentences, i.e. the nodes for these shot-sentence pairs are on the minimum distance path of the cost matrix.

**Fig. 8.** A continuous camera shot with two *semantic shots* aligned with the sentences from its audio description. Top row: *She opens the car door and gets in.* Bottom row: *She gives the chocolate to Gabriel that is in the car.* (From Lucid Dreams of Gabriel)

(i.e. not camera cut), it represents two different semantics which are expressed by two sentences in the audio description. Our joint segmentation approach is able to successfully detect the semantic cuts indicated by different sentences in the text input.

## 5 Evaluation and Discussion

We evaluated the proposed alignment method on a dataset of 12 videos with the sentences from their scripts or audio descriptions, including continuous long sections from the movies *Lucid Dreams of Gabriel*, *The Ninth Gate*, *The Wolf of Wall Street* and *Yes Man*. The duration of the videos in the dataset ranges from 10 to 20 min with 9.51 sentences per minute on average. The dataset is available

on the project webpage and provides our results of shot segmentation, sentence alignment including timestamps, as well as ground truth data.

We now present the evaluation of our proposed alignment approach with respect to the manually obtained ground truth data. We measure the alignment accuracy by computing the temporal error between the ground truth timestamps of the sentences and the timestamps obtained by our approach. Figure 9 shows the distribution of the temporal error. It shows that 88.64 % of the sentences have a temporal error of 0 s, i.e. our timestamps exactly correspond to the ground truth timestamps. This demonstrates the accuracy of our alignment approach.

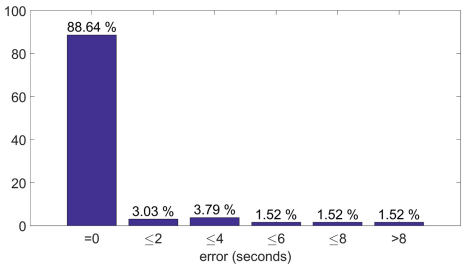


**Fig. 9.** Distribution of the absolute error in seconds on the timestamps obtained by our algorithm with respect to the ground truth timestamps. 88.64 % of the sentences are matched perfectly to the first frame of the corresponding shot.

We now present the evaluation of our proposed shot segmentation approach with respect to the manually obtained ground truth shot segmentation. We consider two metrics again. Firstly we measure the number of shots detected by our approach over the total number of ground truth shots in the movie. Secondly, we measure the number of correctly detected shots by our approach over all detected shots, which includes false positives. The evaluation is shown in Fig. 10.

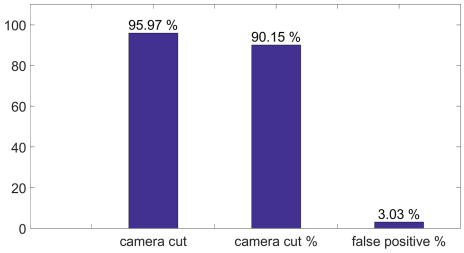


**Fig. 10.** Evaluation of our shot segmentation method. 95.97 % of the ground truth camera shots are detected by our method. 90.15 % of the shots detected by our algorithm correspond to the ground truth camera cuts. Meanwhile, only 3.03 % of the shots detected by our approach are false positives.

Our method has some limitations. First, in order to correctly align the frames with the corresponding sentences, the image labeling tools (e.g. object/scene recognition) should provide sufficiently accurate labels and scores. The accuracy of our method can directly benefit from the next advances of the image labeling tools.

Another limitation is that our method is not designed for videos that mostly consist of close-up shots (e.g. interview videos) rather than scenes, actions and motion. Such video frames would not result in sufficient object/scene labels due to the lack of action and scene changes. We focused on more general movies because we believe they are more common. However, our method is suitable for a simple integration of dialogue-caption alignment approaches used in [28,31] that could be included as another variable in our global cost matrix. In future work, this integration could improve the results in videos that lack narrative sentences during dialogues.

A future application of our approach can be segmentation and structuring of videos that will allow important post-applications in content-based media analysis. Clustering of video units like shots and scenes allows unsupervised or semi-supervised content organization and has direct applications in browsing in massive data sources. Given the framewise high-level labels and timestamps of shot intervals of a video obtained by our algorithm, we can easily cluster these shots. Treating the rows of the cost matrix as the features of the segmented shots, one can simply apply a clustering method to obtain shot clusters.

In future work, it would be interesting to extend the proposed approach to cope with different types of media materials by bringing them into a common representation. For example a storyboard with drawing and sketches could be aligned with the corresponding shots in the movie using the high-level labels and their vector descriptors in an analogous way.

## 6 Conclusion

In this paper, we proposed an automatic method for temporally aligning the video frames of a movie with narrative sentences, for example issued from the movie script. Our approach segments the video into semantic shots and aligns them with the sentences in an iterative way by exploiting vector descriptors for text representation. Experiments on various movies successfully demonstrated the validity of our approach.

## References

1. Agirre, E., Diab, M., Cer, D., Gonzalez-Agirre, A.: Semeval-2012 task 6: a pilot on semantic textual similarity. In: Joint Conference on Lexical and Computational Semantics, pp. 385–393 (2012)
2. Apostolidis, E., Mezaris, V.: Fast shot segmentation combining global and local visual descriptors. In: ICASSP (2014)
3. Bojanowski, P., Lajugie, R., Grave, E., Bach, F., Laptev, I., Ponce, J., Schmid, C.: Weakly-supervised alignment of video with text. In: ICCV (2015)

4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR (2009)
5. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269 (1959)
6. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR (2015)
7. Drew, M.S., Wei, J., Li, Z.N.: Illumination-invariant image retrieval and video segmentation. Pattern Recogn. **32**, 1369 (1999)
8. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D.: Every picture tells a story: generating sentences from images. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 15–29. Springer, Heidelberg (2010)
9. Gupta, A., Srinivasan, P., Shi, J., Davis, L.S.: Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In: CVPR (2009)
10. Hampapur, A., Jain, R., Weymouth, T.E.: Production model based digital video segmentation. Multimedia Tools Appl. **1**, 9 (1995)
11. Han, L., Kashyap, A., Finin, T., Mayfield, J., Weese, J.: UMBC ebiquity-core: semantic textual similarity systems. In: Proceedings of the Second Joint Conference on Lexical and Computational Semantics (2013)
12. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: ACM International Conference on Multimedia (2014)
13. Kulkarni, G., Premraj, V., Ordonez, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.: BabyTalk: understanding and generating simple image descriptions. PAMI **35**, 2891 (2013)
14. Lankinen, J., Kämäräinen, J.K.: Video shot boundary detection using visual bag-of-words. In: VISAPP (2013)
15. Lee, J.C.M., Ip, D.M.C.: A robust approach for camera break detection in color video sequence. In: MVA (1995)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013). arXiv preprint arXiv:1301.3781
17. Nagasaka, A., Tanaka, Y.: Automatic video indexing and full-video search for object appearances (1992)
18. Ordonez, V., Kulkarni, G., Berg, T.L.: Im2Text: describing images using 1 million captioned photographs. In: NIPS (2011)
19. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet::Similarity - Measuring the Relatedness of Concepts. In: Proceedings of Demonstration Papers at HLT-NAACL (2004)
20. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP) (2014)
21. Qu, Z., Liu, Y., Ren, L., Chen, Y., Zheng, R.: A method of shot detection based on color and edge features. In: Proceedings of IEEE Symposium on Web Society (SWS) (2009)
22. Rohrbach, A., Rohrbach, M., Qiu, W., Friedrich, A., Pinkal, M., Schiele, B.: Coherent multi-sentence video description with variable level of detail. In: Jiang, X., Hornegger, J., Koch, R. (eds.) GCPR 2014. LNCS, vol. 8753, pp. 184–195. Springer, Heidelberg (2014). doi:10.1007/978-3-319-11752-2_15

23. Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: CVPR (2015)
24. Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., Schiele, B.: Translating video content to natural language descriptions. In: ICCV (2013)
25. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Sig. Process. **26**, 43 (1978)
26. Sankar, P., Jawahar, C., Zisserman, A.: Subtitle-free movie to script alignment. In: BMVC (2009)
27. Tapaswi, M., Bäuml, M., Stiefelhagen, R.: Story-based video retrieval in TV series using plot synopses. In: Proceedings of International Conference on Multimedia Retrieval (2014)
28. Tapaswi, M., Bauml, M., Stiefelhagen, R.: Book2Movie: aligning video scenes with book chapters. In: CVPR (2015)
29. Xiao, J., Ehinger, K.A., Hays, J., Torralba, A., Oliva, A.: SUN database: exploring a large collection of scene categories. IJCV **119**, 3 (2014)
30. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: NIPS (2014)
31. Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., Fidler, S.: Aligning books and movies: towards story-like visual explanations by watching movies and reading books. In: CVPR (2015)