The video sequencing problem

Hatice Çalık* · **Aäron Hallaert** · **Farzaneh Karami** · **Greet Vanden Berghe**

Abstract We introduce a video sequencing problem that requires selecting video clips and assigning them to the ordered segments of a final video to be formed. Each segment has a set of criteria, which must be satisfied by the assigned clips and each clip-segment assignment has a nonnegative score. The objective is to maximize the total score of the ordered selection of clips. The motivation behind studying this problem is to support the automatization of the video editing process, which might be extremely time consuming otherwise. For example, a news broadcast requires cutting and combining a huge number of video clips in a very short amount of time, which is difficult to achieve via a manual process. We provide an integer programming formulation for this problem and discuss the similarities and differences of the proposed video sequencing problem with those found in the literature. We also develop a local search-based heuristic and conduct a computational study on the two methods using data generated by a recognition software package.

1 Introduction

Recent advances in technology have made it very easy to collect and store a tremendous number of video clips. Managing this large quantity of raw footage and converting it into finished videos is a cumbersome challenge because selecting and editing video material is very time-consuming. Some of these clips are used to compose larger videos and such a process requires a lot of work, which includes viewing and annotating clips

Hatice Çalık
*Corresponding author.
KU Leuven, Department of Computer Science, CODeS, Belgium
E-mail: hatice.calik@kuleuven.be

Aäron Hallaert
KU Leuven, Belgium

Farzaneh Karami
KU Leuven, Department of Computer Science, CODeS, Belgium

Greet Vanden Berghe
KU Leuven, Department of Computer Science, CODeS, Belgium
E-mail: greet.vandenberghe@cs.kuleuven.be

manually as well as cutting and fitting them into the final video. Such a task can make good use of automatization.

This study is conducted in cooperation with a company which offers support in managing video data through an online platform. This platform enables users to manually add *metadata* to videos. ‘Metadata’ is the collective term for certain annotations (labels) which can be related to video content (for example an interview), faces, objects, speech transcriptions or audio. It registers and quantifies the human perception of a video and makes it quicker and easier to search a video collection.

In the traditional video editing sector, the video archive is managed manually by appointing a person to watch every video and add metadata to the file. Manually completing this task is usually inefficient and often leads to inaccuracies. Consider the process of adding a face annotation to the metadata of a video. Information such as when this recognition occurs and how long it is visible is usually not specified since adding these timestamps would delay the entire process. The face-time percentage of an annotated person in a long video may be very low. To actually retrieve the fragment where the desired person is in frame, a video editor still needs to watch the whole video, searching for this specific person. This is a very time consuming process.

Artificial intelligence can help with automating this annotation task. Several globally-known software companies already have software packages to recognize objects and faces. These software packages can complete the annotation task much more precisely and faster than a human. Such software is essential to automatically gather metadata of individual video clips.

The focus of this paper is to introduce the video sequencing problem (VSP). The VSP consists of selecting and ordering a subset of given video clips with metadata by satisfying user-specified requirements related to the content of the resulting video. User specifications, which we also refer to as the criteria, may be face or speech related. Users can specify when they want a certain face to occur in the resulting video, if there have to be other faces in the same frame or if someone has to talk over a certain part of the video. Therefore, we consider clips with metadata containing *face and speech recognitions*. Every *recognition* has a time frame in which it occurs, a type and a description (for example, the name of the person related to this recognition). Recognitions are used to search video fragments for the criteria specified by the user. For each criterion, all the clips that meet the requirements are collected as potential clips for relevant segments via a preprocessing procedure. The objective is to maximize the total score of the selected clips, which is considered as the duration of relevant recognitions based on the requirements of the user. The output is a sequence of selected video clips where each clip belongs to at most one segment, all criteria are met for each segment and the total duration is within predefined boundaries. We provide an integer programming formulation as well as a heuristic approach for the VSP.

The remainder of this paper is organized as follows: Section 2 summarizes related work in the metadata collection and video editing literature. Section 3 formally describes the VSP with an integer programming (IP) formulation and discusses its similarities and differences with several relevant problems from the operations research literature. Section 4 initially details the preprocessing procedure to collect potential clips to be used by the IP model as well as by the local search-based approach, which is also detailed in this section. Section 5 presents the results of the computational study comparing the IP formulation and the heuristic method for several problem instances. Finally, Section 6 concludes with some suggestions concerning future research directions.

2 Related work on metadata collection and video editing

Video editing requires several inputs, including the original video files and metadata, which play a crucial role in the automatization of the process. Among the methods available for metadata collection, Ahanger and Little (1998) add metadata to a video by manually placing every input clip in the right segment. For example, video clips where two people greet each other can be a potential clip for the start segment, conversations for the middle segment and farewells for the end. This approach still requires a person to watch all the input clips and manually group them for the correct segment.

Metadata can also be collected based on technical information instead of content. This technical information may consist of timestamps indicating when the video was filmed. These timestamps can be used to specify the order in which the input clips will occur in the resulting video (Ahanger and Little, 1998). Where a shot change is detected or where certain audio clips start and stop can also be useful when editing videos (Casares et al., 2002).

A more advanced way to gather metadata is introduced by Arev et al. (2014). Synchronized video clips of the same event are used to make a 3D-reconstruction which is useful to find the event's point-of-interest. In this manner it is possible to rate each input clip based on the camera direction.

In the montage phase, the structure of a desired video is often defined by certain limitations and assumptions. One key restricting component is the video context, examples of which include: a news broadcast, a reality-show, a documentary or a thrilling action scene. While an action requires a large number of fast consecutive clips, a nature documentary needs long and slow fragments. Some video editing algorithms focus on a script (Ahanger and Little, 1998), others do not and therefore it might be necessary to make assumptions on the structure of the resulting video or to let the algorithm decide (Arev et al., 2014). Chen et al. (2012) select and montage video material based on visual similarity. They try to make a smooth visual summary, while Ahanger and Little (1998) focus on providing a smooth transition related to the content.

In addition to content-based video editing, technical video editing can also be automated. Casares et al. (2002) discuss a technical montage in which video clips are cut on the basis of a localization of audio clips and shot change detections.

A crucial part of the video editing process is collecting potential clips. One should decide which clips are too long, which are better than the others, and whether or not they satisfy user specifications. Ahanger and Little (1998) suggest two ways of selecting the potential clips: 'boolean metrics' and 'ranking metrics'. Boolean metrics select all the clips that meet the demands of the user. Ranking metrics rank all the input clips based on how well they meet the requirements.

Once the suitable clips have been collected, the final video can be obtained by filling the available time budget. This procedure is usually not completely random and takes the user-specified constraints, assumptions, context or content similarity into account. In certain cases, cutting the input clips may increase the possibility of obtaining higher quality outputs in this assembly process. Casares et al. (2002) discuss different techniques to cut clips. Audio-based cuts make sure the whole required audio fragment is present with cuts taking place at shot change. Video-based cuts ensure the required video fragment is in the cut and the audio is not abruptly interrupted.

All aforementioned papers provide interesting technology to support video editing. We will focus on a process step which has not been automatized, yet offers a huge potential for efficiency improvement.

3 Mathematical formulation

Given a set of video clips $V = \{1, \ldots, k\}$ with t_v being the length of clip $v \in V$, the VSP requires selecting a subset $\bar{V}$ of V to obtain a video whose length $T = \sum_{v \in \bar{V}} t_v$ is between L and U time units ($L < U$, $T \in [L, U]$). This final video should consist of a set of segments $S = \{1, \ldots, n\}$ where each segment $s \in S$ has a set of criteria C_s which must be satisfied by the selected clips in $\bar{V}$. Let $V_c \subset V$ be the set of clips satisfying criterion $c \in C_s$ and $p_{vc} \geq 0$ represent the satisfaction value (score) of clip $v \in V$ for criterion c, the objective is then to maximize the total score of the clips in $\bar{V}$.

Define x_{vs} as a binary decision variable that takes value 1 if clip $v \in V$ is selected for segment $s \in S$, and 0 otherwise. The following is an IP model for the VSP.

$$\max \quad \sum_{s \in S} \sum_{c \in C_s} \sum_{v \in V_c} p_{vc} x_{vs} \tag{1}$$

$$\text{s.t.} \quad \sum_{v \in V_c} x_{vs} \geq 1 \qquad \forall s \in S, c \in C_s \tag{2}$$

$$\sum_{v \in V} \sum_{s \in S} t_v x_{vs} \geq L \tag{3}$$

$$\sum_{v \in V} \sum_{s \in S} t_v x_{vs} \leq U \tag{4}$$

$$\sum_{s \in S} x_{vs} \leq 1 \qquad \forall v \in V \tag{5}$$

$$x_{vs} \in \{0, 1\} \qquad \forall v \in V, s \in S \tag{6}$$

The objective function (1) maximizes the total score of the selected clips. Constraints (2) select at least one clip for each criterion of each segment. Constraints (3) and (4) ensure that the total duration of the final video is within the predefined boundaries. Constraints (5) make sure that the same video clip does not appear in multiple segments. Finally, Constraints (6) are binary restrictions.

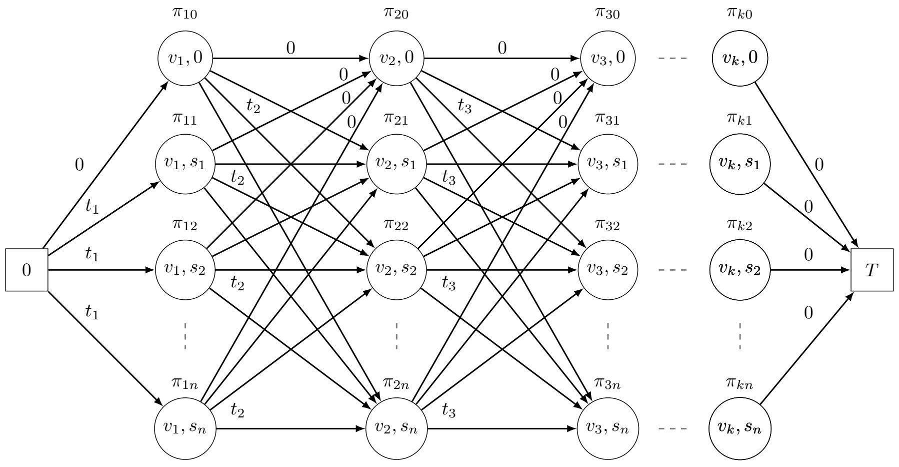

Fig. 1: A network representation with clip-segment assignments.

The VSP can be represented on a directed network such as that depicted in Figure 1. In this network, node (v_i, s_j) denotes the assignment of clip v_i to segment s_j and $(v_i, 0)$ denotes the case where clip v_i is not assigned to any segment and thus not used at all. The length of an arc represents the amount of time used. The length of arcs entering node $(v_i, 0)$ is equal to zero whereas it is equal to t_{v_i} for arcs entering node (v_i, s_j), where $v_i \in V$, $s_j \in S$. The score π_{ij} of visiting node (v_i, s_j) is equal to $\sum_{c \in C_{s_j} : v_i \in V_c} p_{vc}$ for $s_j \in S$ and $\pi_{i0} = 0$ for $(v_i, 0)$, $v_i \in V$.

Figure 2 provides a more informative network that also indicates criteria satisfaction. In this figure, we consider a sample problem instance with four video clips $\{v_1, \ldots, v_4\}$ and three segments each of which has three criteria to be satisfied. More specifically, $C_1 = \{c_1, c_2, c_3\}$, $C_2 = \{c_4, c_5, c_6\}$ and $C_3 = \{c_7, c_8, c_9\}$. If clip v_i satisfies criterion c_j, we add a node (v_i, c_j) with score p_{ij} to the network. The VSP then requires to find a maximum score path from node 0 to node T that visits at least one node for each criterion and respects the total duration limits (3) and (4).

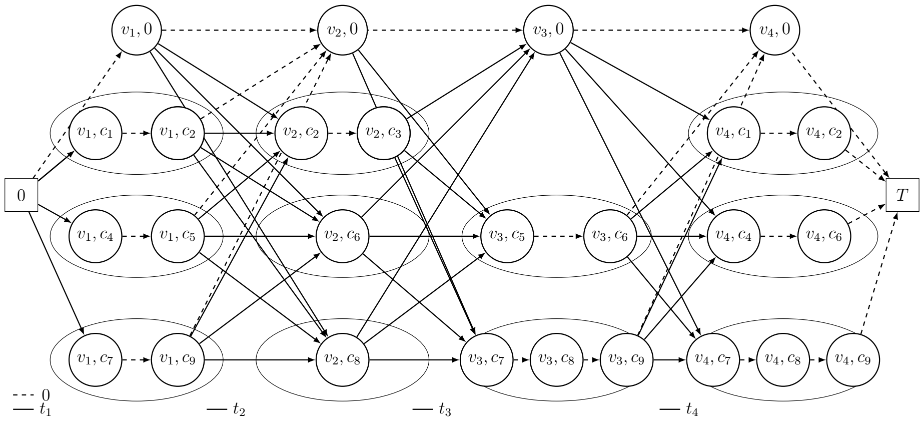

Fig. 2: A network representation with clip-criterion assignments for an example of four video clips and three segments, each with three criteria. The score of node (v_i, c_j) is equal to p_{ij}. The length is equal to zero for the dashed arcs and equal to the duration of the corresponding video clip for entering solid arcs.

The VSP contains many components similar to those present in several well-studied problems from the literature. When omitting Constraints (2) and (3), one can represent this relaxed problem as an *orienteering problem* (OP) from node 0 to node T on either of the (acyclic) networks visualized in Figures 1 or 2. The OP aims to find a path on a given network from a source node to a destination node such that the total length of the path is less than a certain threshold and the total score collected from visited nodes is maximized. For a formal description of the OP, its complexity and reviews of its variants, we refer to Vansteenwegen et al. (2011). An extended survey containing relatively recent OP variants, solution methods and applications is provided by Gunawan et al. (2016).

The most relevant OP variant is probably the split delivery capacitated team orienteering problem with minimum delivery amounts (SDCTOP-MDA) (Vansteenwegen and Gunawan, 2019). In the context of tourism applications where the OP aims to provide a visit plan, there may be lower and upper limits on the number of certain types of attractions to visit, for example at least one church and at most two museums. The VSP can be considered as a variant of the capacitated OP with minimum delivery amounts (COP-MDA) considering the criteria satisfaction constraints (2) and no-repetition constraints (5), which have a slightly different structure than the capacity constraints in the SDCTOP-MDA.

Video selection in the VSP is also similar to selection in the well-known knapsack problems. Note that the relation between the node selection in the OP and knapsack problems is already noted by Vansteenwegen and Gunawan (2019). If one considers a knapsack with multiple segments of flexible size and items yielding different values in different segments, then finding a maximum value assignment of items to segments becomes a relaxation of the VSP without Constraints (2). To the best of our knowledge, no such knapsack variant has been studied in the literature.

Although there is an upper limit on the total length of the final video in the VSP, there is no upper limit on the length of the individual segments. This property resembles the flexible compartment variant of the multi-compartment vehicle routing problem introduced by Henke et al. (2015). Even though the two problems share some properties, they differ in many aspects. The most fundamental difference is that the VSP has more of a maximal covering structure which selects the most profitable subset of items fitting in the time budget while the problem by Henke et al. (2015) aims to meet all the demand at minimum cost.

The criteria satisfaction constraints (2) of the VSP can also be associated with the well-known set covering problem. Considering this covering aspect and a routing structure similar to the OP, the VSP also shares certain features with the covering tour problem by Gendreau et al. (1997). Recalling Figure 1, the covering tour problem requires visiting a subset of nodes such that all the criteria are covered (i.e. satisfied by at least one clip) and the total length of the tour is minimized. Unlike the covering tour problem, the VSP aims to maximize the total score of the selected nodes while ensuring the total length of the tour (final video) is within certain limits.

The following section describes a heuristic which is combined with the preprocessing procedure that enables searching for video clips satisfying multiple user criteria such as face-face, face-speech, speech-speech recognitions as well as cutting clips into smaller pieces. Note that this preprocessing procedure is utilized also for obtaining the inputs of the IP model.

4 Heuristic approach

4.1 Preprocessing for multiple criteria search on clips

When searching for clips which meet certain criteria where simultaneous face or speech segments are needed, the set of recognitions belonging to a clip must be iterated too many times. Consider the example where person A needs to be in the frame while at the same time person B is speaking. If the recognition of Face A in a clip is found, it has to be iterated over all other recognitions, searching for person B who is speaking at the same time as the recognition of person A. This process can be accelerated using a binary interval tree of recognitions in a clip. If person A is recognized in the clip, all overlapping recognitions are searched in logarithmic time in relation to the number of recognitions in the tree.

Figures 3 and 4 illustrate this tree search and how the nodes of the tree are pruned in logarithmic time. In this example, we have a clip of Face A whose time interval is $[25.0, 36.0]$. The seven nodes of the tree in Figure 4 represent the time frames of the available recognitions. At the first iteration of the logarithmic search, a branch of the tree is cut since the maximum end value of its children is greater than the beginning time of Face A clip. Therefore, only four nodes are explored and three of them (in bold) represent the overlapping clips.

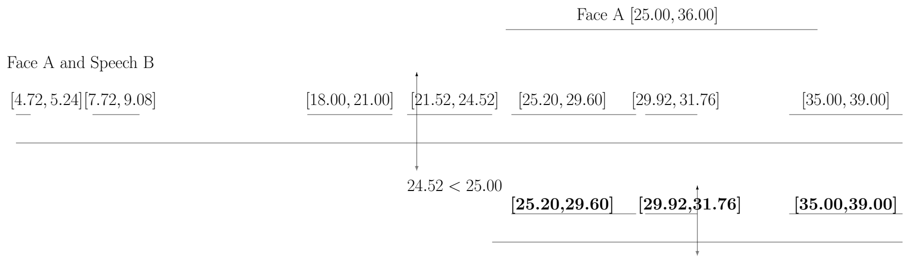

Fig. 3: Logarithmic pruning of time intervals.

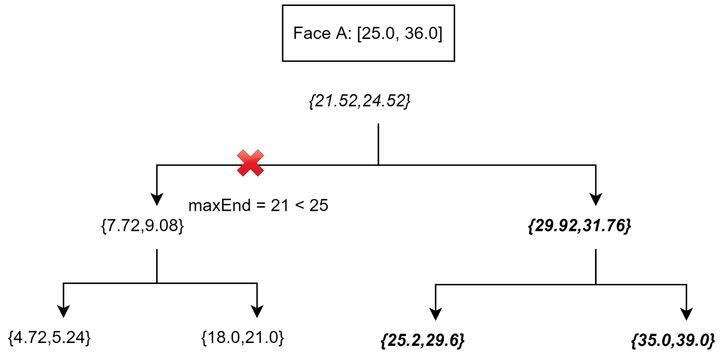

Fig. 4: Using binary interval trees for searching overlapping recognitions with respect to the recognition of Face A.

After collecting all overlapping clips, we search in this set for a recognition where person B is speaking. If this recognition is in the set of overlapping clips, the requirement is met and the clip is selected as a potential clip for the segment of this criterion.

When the input clips can be cut, all the recognitions are searched in a similar manner and cuts are made based on these recognitions. The first recognition in time that has a role in fulfilling the requirement determines the start of the cut and the last recognition in time determines its end. In doing so, the irrelevant frames can be avoided in the final video. Once a cut is made, it is not possible to cut this sub-clip into smaller pieces to satisfy another requirement. In a similar fashion these sub-clips are chosen as potential clips for the corresponding segment.

4.2 The local search algorithm

The initial solution for the local search is constructed by selecting a random candidate clip from V_c for every criterion $c \in C$. The initial solution $\bar{V}$ does not have to be feasible.

The local search strategy explores the solution space to improve this initial solution by adding, deleting or both adding and deleting the potential video clips. The local search method contains a very simple step-by-step procedure consisting of two parts that realize these modifications. At each iteration, a random video clip v from the candidate videos is added to $\bar{V}$. If the resulting video is longer than the duration limit U after adding v, a number of random clips are deleted from $\bar{V}$. The number of clips to be removed is a parameter which can be specified by the user.

The acceptance criteria are described in Algorithm 1. The current solution is accepted when it is better than the best solution, but it can also be accepted if both the best and current solution are infeasible. If the best solution is feasible and the current is not better than the best, there is still a chance that the current solution will be accepted for the next iteration. Although the thresholds of this chance can be specified a priori, preliminary experiments showed that the thresholds employed in Algorithm 1 performed the best.

```
Data: Current solution and next, best and initial solution
if current.cost > best.cost & current.isFeasible then
    next= current;
    best= next;
else if !best.isFeasible then
    next=current;
else
    generate $r_1 = rand(0,1)$;
    if $r_1 > \frac{1}{9}$ then
        next= initialSolution;
    else
        generate $r_2 = rand(0,1)$;
        if $r_2 > \frac{1}{3}$ then
            next = current;
    end
end
```

Algorithm 1: Acceptance function.

5 Computational study

The IP model and the local search algorithm were implemented in Java and the experiments were run on an Intel® Core$^{\text{TM}}$ i7-4720HQ CPU, 2.60 GHz computer with 12 GB of RAM. In order to solve the IP model, ILOG IBM CPLEX 12.9 was used. The methods were applied to video material from 'De Wereld Rond met 80-jarigen' (a production of VTM) and 'WTFock' (a production of Sputnik Media). The raw video material was annotated using a recognition software provided by Valossa. The recognition specifications of these VSP instances are denoted by the following notation:

- F: the end result must contain a fragment where a single face must be present.

– S: the end result must contain a fragment where a single speaker must be present.
– FF: the end result must contain a fragment where multiple faces must be simultaneously present.
– FS: the end result must contain a fragment where a face and speaker must be simultaneously present.
– C: the input clips can be cut to be part of the resulting video.

Further specifications of the instances will be shared during the conference presentation. These instances are available upon request.

Table 1 details the results obtained when applying the IP model and the heuristic to problem instances which only consider single face recognitions. For each of these instances, the heuristic was only run once. Although the heuristic can find optimal solutions for only two of the five instances, its solutions for the others are not considerably inferior compared to the optimal solutions. The largest gap between the optimal and the heuristic solution value is 5.5%.

Table 1: IP and heuristic results for the problems with only 'F' criterion.

	IP Model		**Heuristic**	
Specifications	**Optimal Score**	**Duration (T)**	**Best Score**	**Duration (T)**
L=8,U=12 seconds	11.44	11.92	11.44	11.92
L=24,U=36 seconds	33.32	35.84	31.68	36
L=20,U=30 seconds	21.24	23.96	21.24	23.96
L=24,U=36 seconds	35.84	35.76	33.96	34.84
L=48,U=72 seconds	96.52	71.96	94.72	71.8

Table 2 provides the results for the problem instances with multiple criteria as well as the possibility of clip cuts. The heuristic is again able to find optimal solutions for two of these instances. Among them, for the instance at the bottom row of the table, the IP solution has a longer duration than the solution of the heuristic. This is because the heuristic uses dummy clips (the clips with $p_{vc} = 0$ for all criteria) only when they are needed, that is, for satisfying the minimum length requirement. On the other hand, the IP model might provide alternative solutions including these dummy clips even when they are not needed. If dummy clips are undesirable for the user, one can easily remove these clips from the set of potential clips during the preprocessing procedure.

Both methods' computational times remain under 20 seconds when solving any instance in these tables. We observe that the size of these instances are too small to challenge the mathematical model and more extensive experiments on large scale instances need to be conducted to see the limits of the IP model.

6 Conclusion and future work

This paper introduced a video sequencing problem (VSP) and provided optimization-based methods to support the video editing process. The aim is not to take over the complete task but to support its most important parts, such as the selection and placement of video clips. Recognitions, obtained by external software, make it possible

Table 2: IP and heuristic results for the problems with multiple criteria and cuts

	IP Model		**Heuristic**	
Specifications	**Optimal Score**	**Duration**	**Best Score**	**Duration**
L=48, U=72 seconds FF (Vidal - Rene), FS (F: Aureel - S: Sieg), F (Amanda), C	56.64	71.92	40.12	49.6
L=72, U=108 seconds FF (Sieg -Olga), F (Amanda), F (Willy), C	34.64	94.44	34.16	82.04
L=32, U=48 seconds FF (Vital - Rene), FF (Sieg - Olga), F (Mariette), F (Aureel)	44.32	47.76	44.32	47.76
L=64, U=96 seconds FS (S: Sieg - F: Aureel), FF (Olga - Sieg), F (Amanda), S (Willy), C	66.28	88.36	56.48	66.96
L=24, U=36 seconds F (Amanda), F (Olga), F (Willy), C	32	35.32	32	32

to search for and identify the video material that satisfies desired specifications aided by the binary interval tree. When possible they are cut out of the input clips in order to meet the user's requirements.

Although the main focus of this paper is to introduce a new sequencing problem with an integer programming formulation, we also include a local search-based heuristic which is rather easy to combine with the preprocessing of input materials as it does not require external mathematical solvers.

An obvious extension is to add other recognitions such as object and audio recognitions. These can be implemented very easily since they will behave analogous to the current face and speech recognitions.

Introducing greater flexibility for choosing requirements can be realized by specifying the duration of individual segments or letting the user choose between new features. The more possibilities of specifying the end result, the more scenarios will benefit from the presented approach to automate the video editing process.

It would also be interesting to consider smooth transition based on content to make a more pleasing video. In order to do this, one could cluster clips based on their visual similarity and/or audio similarity, that is, those clips in which the same keywords are spoken. This would, however, require more detailed annotations than we employed for the VSP.

Acknowledgements The video sequencing problem was introduced to KU Leuven by Limecraft. Research partially supported by Data-driven logistics (FWO-S007318N). Editorial consultation provided by Luke Connolly (KU Leuven).

References

Ahanger, G. and Little, T. D. (1998). Automatic composition techniques for video production. *IEEE Transactions on Knowledge and Data Engineering*, 10(6):967–987.

Arev, I., Park, H. S., Sheikh, Y., Hodgins, J., and Shamir, A. (2014). Automatic editing of footage from multiple social cameras. *ACM Transactions on Graphics (TOG)*, 33(4):81.

Casares, J., Long, A. C., Myers, B. A., Bhatnagar, R., Stevens, S. M., Dabbish, L., Yocum, D., and Corbett, A. (2002). Simplifying video editing using metadata. In *Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 157–166. ACM.

Chen, Q., Wang, M., Huang, Z., Hua, Y., Song, Z., and Yan, S. (2012). Videopuzzle: Descriptive one-shot video composition. *IEEE Transactions on Multimedia*, 15(3):521–534.

Gendreau, M., Laporte, G., and Semet, F. (1997). The covering tour problem. *Operations Research*, 45(4):568–576.

Gunawan, A., Lau, H. C., and Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332.

Henke, T., Speranza, M. G., and Wäscher, G. (2015). The multi-compartment vehicle routing problem with flexible compartment sizes. *European Journal of Operational Research*, 246(3):730–743.

Vansteenwegen, P. and Gunawan, A. (2019). *Orienteering Problems Models and Algorithms for Vehicle Routing Problems with Profits*. Springer, Switzerland.

Vansteenwegen, P., Souffriau, W., and Van Oudheusden, D. (2011). The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10.