

Short Paper

A Domain-Independent and Personalized Video Abstraction Algorithm*

JINGUK JEONG, JONGHO NANG AND HOJUNG CHA*

Department of Computer Science

Sogang University

Seoul 121-742, Korea

E-mail: jhnang@ccs.sogang.ac.kr

**Department of Computer Science*

Yeonsei University

Seoul 120-749, Korea

A video server on the Internet usually provides a short version of a video clip, a video abstraction, in order to provide highlights or the overall story to users as quickly as possible. However, since it is statically generated by content providers using domain-dependent heuristics, it cannot satisfy all users simultaneously. This paper proposes a domain-independent video abstraction algorithm that generates various video abstractions dynamically according to the users' requirements. It first identifies some low-level visual and temporal constraints that a good video abstraction should satisfy domain-independently, such as that it should be *well-distributed*, *highly-active*, and *non-duplicated* (or *concise*), that are used to partially represent the user's requirements. These constraints are formalized as objective functions, and a simulated annealing algorithm is used to find a set of shots that maximizes the weighted sum of these objective functions as much as possible. It is a personalized abstraction algorithm since each user can generate and view various video abstractions by dynamically adjusting the weights of the constraints. From the results of several experiments with a Korean movie and other well-known movie videos, we found that the proposed algorithm can produce various useful video abstractions very quickly.

Keywords: video abstraction, video analysis, video-on-demand, simulated-annealing, video trailer, multimedia system

1. INTRODUCTION

Recently, many video servers have begun to provide a variety of video clips to users with different purposes through Intranets or the Internet. Examples of such video-based multimedia systems are news archive systems, VoD (Video on Demand) systems, and many Web server systems containing various video clips. However, although network

Received December 12, 2002; revised July 2, 2003; accepted September 15, 2003.

Communicated by Liang-Gee Chen.

* This work was supported by the Basic Research Program of the Korea Science and Engineering Foundation (grant No. R01-2002-000-00141-0).

bandwidth is increasing rapidly and robust multimedia communication protocols are being developed, it is still not possible to deliver video clips (even in compressed form) to users as naturally as VCRs or recently developed DVD players. This has stimulated several researches to develop efficient video abstraction (or summarization) schemes that enable users to quickly browse video clips in different levels of detail, without the need to receive and view the entire video clips. Although many researches have focused on abstracting a long video clip to create a shorter version automatically, they have usually been based on domain specific heuristics, such as “*selecting violence shots with high motion energy since they are usually highlights of action movies,*” so their use is limited only to specific domains. Furthermore, since video abstractions are usually generated without considering the user’s requirements for viewing the abstraction (*i.e.*, it is generated statically by a content provider or a Web manager), the usefulness of the generated abstraction is also limited. This problem can be resolved if the user’s requirements are represented as a combination of low-level constraints that the shots in the video abstraction should satisfy, and if the user is allowed to assign a relative weight to each constraint based on his/her own objectives for viewing the video abstraction interactively.

This paper proposes a simulated annealing-based abstraction algorithm that automatically and dynamically generates various video abstractions based on the user’s requirements. It first identifies a set of domain-independent conditions (or constraints) that the selected shots for video abstraction should satisfy, and formalizes them as objective functions. These constraints are: a) the selected shots should be uniformly distributed over all of the video clips (be *well-distributed*), b) their total run-time should match the target run-time of abstraction (*good fit*), c) they should contain shots with some specific events, such as actions (be *highly-active*), d) they should be different visually from each other (be *non-duplicated* or *concise*), and so on. Then, this paper formalizes the video abstraction process as a combinatorial optimization problem that selects k shots from the original video clip consisting of n shots while satisfying the above constraints as much as possible. Since this problem is NP-complete ($O(2^n)$), this paper proposes a shot selection algorithm based on simulated annealing [1] in order to generate a video abstraction, which can satisfy the user’s objectives that are represented by the weights of the objective functions as much as possible in polynomial time. We have tested the proposed video abstraction algorithm on various genres of video clips, such as Korean sitcoms and well-known action video clips, and found that the proposed algorithm could generate various useful video abstractions quickly. The proposed video abstraction algorithm can be used to build a video-based multimedia server system that dynamically generates various video abstractions interactively based on the user’s requirements.

This paper is organized as follows. In section 2, the constraints that a good video abstraction should satisfy are identified and formalized, and an abstraction algorithm based on the simulated annealing is presented. Some experiments with a Korean sitcom and an action video clip are presented in section 3, while a comparison with related works is presented in section 4. Finally, our concluding remarks are presented in section 5.

2. VIDEO ABSTRACTION ALGORITHM

In this section, we will formalize the video abstraction process as a combinatorial

optimization problem with several constraints and present a simulated annealing-based searching algorithm to generate various video abstractions.

2.1 Problem Definition

The main process in making a video abstraction is selecting some important shots from the video clip after it is segmented into a set of shots as shown in Fig. 1. Let $V = \{v_i \mid 1 \leq i \leq n\}$ be the video clip consisting of n shots, and let $X = \{x_i \mid x_i \in V, 1 \leq i \leq k, 1 \leq k \leq n\}$ be its abstraction consisting of k shots. Then, the video abstraction process can be formalized as selecting k shots from n shots, and the number of different video abstractions consisting of k shots would be ${}_n C_k$. Since the range of k would be $1 \leq k \leq n$ according to the target run-time of abstraction, the total number of different video abstractions would be ${}_n C_0 + {}_n C_1 + {}_n C_2 + \dots + {}_n C_n = 2^n$. It is the same as the total number of different subsets of V . Among these abstractions, the video abstraction algorithm should find the one that satisfies the user's requirements as much as possible.

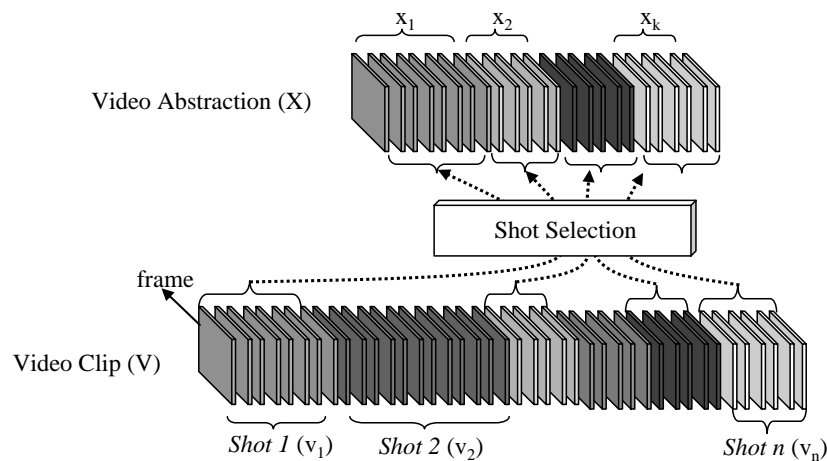


Fig. 1. Video abstraction process.

2.2 Formalizing the Constraints

The low-level visual constraints that a good video abstraction should satisfy can be classified into two classes; intra-shot constraints and inter-shot constraints. The former constraints, such as *high-activities* are determined by the visual characteristics of the shot itself, while the latter constraints, such as *well-distributed*, are determined by the relationship between the other shots in the abstraction. Let us formalize these constraints one by one. In this formalization, the start frame number, the end frame number, and the length of shot x_i are represented as S_i , E_i , and L_i , respectively. Furthermore, we assume that the target run-time of the abstraction that is dynamically specified by the user is denoted as T . These notations are shown in Fig. 2 graphically.

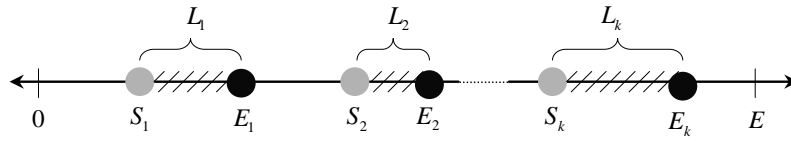


Fig. 2. The notations.

◆ Well-Distributed (\$O_1\$)

This constraint means that the set of shots selected for the abstraction should be uniformly distributed over the whole video in order to provide an impression of the entire video content. It is especially important for summary-style abstractions. If the intervals between selected shots in \$X\$ are similar to each other, we can say it is a well-distributed abstraction. The average interval between the shots in \$X\$, \$\mu\$ and the deviation of intervals, \$\Delta\$, can be computed as follows:

$$\mu = \frac{(S_1 - 0) + (S_2 - E_1) + \dots + (E - E_k)}{k + 1} = \frac{E - \sum_{i=1}^k L_i}{k + 1}$$

$$\Delta = |S_1 - \mu| + |(E - E_k) - \mu| + \sum_{i=2}^k |(S_i - E_{i-1}) - \mu|$$

Since less deviation implies a more well-distributed abstraction, its inverse function is defined as an objective function for the Well-Distributed constraint, \$O_1\$, as follows:

$$O_1(X) = \frac{1}{\Delta} = \frac{1}{|S_1 - \mu| + |(E - E_k) - \mu| + \sum_{i=2}^k |(S_i - E_{i-1}) - \mu|}$$

◆ Good Fit (\$O_2\$)

If the difference between the target run-time of the abstraction (\$T\$) and the sum of the run-time of the shots in \$X\$ (\$L = \sum_{i=1}^k L_i\$) is small (i.e., \$T \approx L\$), it will be a good abstraction. This constraint can be formalized as follows using a hyperbolic function, \$sech(x)\$¹, where \$C_1 = \frac{2}{T} \ln(2 + \sqrt{3})\$.

$$O_2(X) = \frac{2}{e^{C_1(L-T)} + e^{-C_1(L-T)}}$$

¹ Among several equations satisfying this constraint, we choose a hyperbolic function, \$sech(x)\$, in the proposed formalization since it is easy to transform its shape while keeping it symmetric along y axis. Of course, another equation could be used in this formalization as long as it imposes a penalty in proportion to the absolute difference between the target and actual run-times.

◆ Not-too-Short (O_3)

The minimum run-time of a continuous shot should be at least 3.5 seconds for the brain to be able to process it [2]. This means that a shot whose run-time is less than 3.5 seconds will be excluded from the abstraction because it can not be recognized by users. On the other hand, if the shot run-time is greater than 3.5 seconds, the shot has an equal opportunity to be selected for the abstraction. To express this constraint, $f(L_i) = \frac{L_i + C_2 - |L_i - C_2|}{2 \cdot C_2}$ is used to denote the suitability of shot x_i for the abstraction, where C_2 is fixed at 3.5. It is a monotonically increasing function when $0 < L_i < C_2$, and it is a constant function always returning 1 when $C_2 \leq L_i$. The average value of $f(L_i)$ for all shots in X is defined as an objective function for the Not-too-Short constraint as follows:

$$O_3(X) = \frac{1}{k} \sum_{i=1}^k \frac{L_i + C_2 - |L_i - C_2|}{2 \cdot C_2}$$

◆ Highly-Active (O_4)

If there is a lot of object motion in the shot, it is usually regarded as an important one so that it should be included in the abstraction. For example, if there is an explosion or gun-fire event, which usually leads a lot of object motion, it will be a highlight of the video clip, so shots with these events should be included in the abstraction. This is a commonly used heuristic in video abstraction researches, such as [2-4]. To express this constraint, a simplified version of *the motion intensity index* of the shot [5] for x_i , $g(x_i)$, is used to represent the level of visual activity of the shot. Let B_j^i be the binary image of the j -th frame of the i -th shot, and let $B_j^i(m, n) = 1$ if $|F_j^i(m, n) - F_{j-1}^i(m, n)| > \alpha$; otherwise, $B_j^i(m, n) = 0$, where $F_j^i(m, n)$ represents the pixel value at coordinate (m, n) of the j -th frame of the i -th shot, and α is a threshold. Then, the average motion intensity index of shot x_i in X is defined as an objective function for the Highly-Active constraint as follows:

$$O_4(X) = \frac{1}{k} \sum_{i=1}^k g(x_i)$$

where $g(x_i) = \frac{1}{(L_i - 1)} \sum_{j=2}^{L_i} \sum_{m,n} B_j^i(m, n)$ and L_i is the number of frames in shot x_i .

◆ Concise or Non-Redundant (O_5)

In order to include more information in a shortened video clip, similar shots should not be selected repeatedly in the abstraction process. This heuristic has been adopted in a couple of video abstraction researches [2, 6, 7], in which the video frames (or the key frames of the shots) were grouped into several clusters based on their visual characteristics (usually, their color histograms) and only a limited number of shots or their key frames were selected from each cluster in order to remove visual redundancy from the video abstraction. We also adopt this heuristic in the proposed scheme, and the degree of visual difference between the shots in X is used to denote the suitability of X for the video abstraction. In order to select a set of shots that are different from each other, the

key frames of all the shots are compared each other, and their visual similarity values are computed. Let \vec{f}_z^i and \vec{f}_z^j represent the color histograms of the key frames of the i -th shot and j -th shot, respectively. Actually, they are vectors in m -dimensional space, where m is the number of bins in the color histogram. If the color histograms of two key frames (\vec{f}_z^i and \vec{f}_z^j) are different, the angle between these two vectors (θ) is large, and its cosine value ($\cos(\theta) = \frac{\vec{f}_z^i \cdot \vec{f}_z^j}{|\vec{f}_z^i| \cdot |\vec{f}_z^j|}$) is also large. Since the range of the cosine function is $[-1, 1]$, we have transformed it to obtain the range $[1, 0]$ as shown in the objective function $O_5(X)$. The degree of visual difference between all the shots in X can be computed using the following equation:

$$O_5(X) = \frac{1}{2} - \frac{1}{2} \cdot \frac{1}{k} \sum_{i=0}^{k-1} \sum_{j=i+1}^k \frac{\vec{f}_z^i \cdot \vec{f}_z^j}{|\vec{f}_z^i| \cdot |\vec{f}_z^j|}$$

◆ Shot-Exclusion (O_6)

If the video abstraction is used as a video trailer, the last part of the video clip should be concealed. In this case, only the shots in the first 80% of the video clips can be candidates for the video abstraction [3]. In order to express this constraint, $h(x_i) = \left(\frac{E - E_i - |E_i - C_3|}{2 \cdot (E - C_3)} + \frac{1}{2} \right)$, is used to denote the suitability of x_i in X for the video abstraction, and their average value is defined as an objective function for the Shot-Exclusion constraint as follows:

$$O_6(X) = \frac{1}{k} \cdot \sum_{i=1}^k h(x_i) = \frac{1}{k} \cdot \sum_{i=1}^k \left(\frac{E - E_i - |E_i - C_3|}{2 \cdot (E - C_3)} + \frac{1}{2} \right)$$

where C_3 is the start frame number of the last 20% of video clip.

◆ Non-Bias (O_7)

If the run-time of shot x_i is too long, only a relatively small number of shots can be included in the video abstraction. To avoid this problem, a shot with a run-time that is too long will be excluded from the video abstraction. This means that if X includes a shot whose run-time is too long compared to the other shots in X , then it will not be a good abstraction. To express this constraint, the difference between the average shot length of the shots in X ($\alpha = \frac{1}{k} \cdot \sum_{i=1}^k L_i$) and the longest run length of the shots in X ($\max(L_i)$) is used to denote the suitability of X for the abstraction. Since a smaller difference implies greater suitability, its inverse function is defined as an objective function for the Non-Bias constraint as follows:

$$O_7(X) = \frac{1}{|\alpha - \max(L_i)|}, \forall i \in \{1, \dots, k\}$$

Since the ranges of the return values of these objective functions will be different from each other (for example, the ranges will be $[0 : \infty]$ for O_1 and O_7 , while others will

not), they should be normalized to the same range in order to evaluate their suitability precisely. In the proposed scheme, a normalizing function, $O'_n(X) = \frac{O_n(X)}{O_n(X)+1}$, is used to normalize the objective function values to $[0 : 1]$.

A good video abstraction will be a set of shots that simultaneously satisfies the above constraints as much as possible. However, since the relative importance of the constraints represented by objective functions are dependent on the aims of the abstraction, the objective function values should be weighted when computing the overall suitability of X for the abstraction. Thus, the process of making a good abstraction can be formalized by finding a set of shots X that maximizes the weighted sum of the objective functions, $G(X)$, using the following equation:

$$G(X) = \sum_{p=1}^7 W_p \cdot O'_p(X)$$

where W_p is the weight of the objective function $O'_p(X)$.

2.3 Abstraction Algorithm using Simulated Annealing

Since the number of possible video abstractions for a video clip consisting of n shots is 2^n , it would be very hard to generate a good video abstraction in polynomial time as mentioned before. It is a sort of combinatorial optimization problem to find X among 2^n candidates that maximizes the overall objective function $G(X)$. Several search algorithms have been proposed that can find the near-optimal solutions of combinatorial optimization problems. The simulated annealing algorithm [1] is one such search algorithm that can find a sub-optimal solution in polynomial time. It is a stochastic search algorithm derived from statistical mechanics for finding near globally-minimum-cost solutions to large optimization problems. We have used it to find a set of shots X that maximizes the overall objective function $G(X)$ among 2^n candidates.

In order to apply the simulated annealing algorithm to the video abstraction problem, initially, a set of shots, X_1 , is randomly selected from among 2^n candidates, and its overall objective function value ($G(X_1)$) is computed. Then, another set of shots, X_2 , is selected, and its overall objective function value ($G(X_2)$) is also computed. If $G(X_2) > G(X_1)$, then X_2 is accepted as a candidate for good abstraction. Otherwise, X_2 is accepted as a candidate for a candidate for a good abstraction with probability $e^{\frac{-(G(X_2)-G(X_1))}{T}}$, where T is an initial temperature which controls the annealing process. Let the accepted abstraction be the candidate for a good abstraction, and repeat the above process while decreasing the temperature T until it is less than a predefined temperature ϵ . In this annealing process, when T is high enough, the probability of accepting an abstraction that is worse than current one is also high. However, as the annealing process progresses (i.e., T is decreased), the probability of accepting a worse abstraction as a candidate for a good abstraction also decreases. This stochastic annealing process helps to avoid locally optimal abstraction, and to eventually find a globally near optimal abstraction in a reasonable amount of time. This video abstraction algorithm using simulated annealing is presented in the Fig. 3.

```

Procedure AbstractionBySA( $\rho, T, \varepsilon, W[]$ )
  Real  $\rho, T$ ;           //  $T$ : Initial Temperature,  $\rho$ : Cooling Rate
  Real  $\varepsilon$ ;         // Termination Condition
  Real  $W[]$ ;          // Weights of the Objective Functions

  Real  $G_1, G_2$ ;
  Integer  $X_1[], X_2[]$ ; // Randomly Selected Set of Shots
  Real  $p$ ;           // Temporal variable to store the probability of accepting  $G_2$ 

  begin
     $X_1 \leftarrow \text{RandomlyChooseShotSet}()$ ;
     $G_1 \leftarrow \text{ObjectFunction}(X_1, W)$ ;
    while ( $T > \varepsilon$ ) do
       $X_2 \leftarrow \text{RandomlyChooseShotSet}()$ ;
       $G_2 \leftarrow \text{ObjectFunction}(X_2, W)$ ;

      if ( $G_2 \geq G_1$ )
        then  $X_1 \leftarrow X_2$ ;  $G_1 \leftarrow G_2$ ;
      else
         $p \leftarrow e^{-\frac{(G_2 - G_1)}{T}}$ 
        if ( $(p \geq \text{Random}(0, 1))$ ) then  $X_1 \leftarrow X_2$ ;  $G_1 \leftarrow G_2$ ;
      end_if
    end_while
     $T \leftarrow \rho * T$ ;
  return  $X_1$ 
end.

```

Fig. 3. Video abstraction algorithm using simulated annealing.

3. EXPERIMENTAL RESULTS AND ANALYSES

In order to show the effectiveness of the proposed abstraction scheme, we tested the proposed abstraction algorithm on various kinds of the video clips, such as Korean movies, Korean sitcoms, and well-known action movie clips. However, as mentioned before, there is still no common criterion that can be used to judge the quality of a video abstraction quantitatively. Furthermore, the quality of an abstraction usually depends on the aims of the abstraction, so it is very hard to show the effectiveness of the proposed abstraction scheme precisely. Because of this difficulty, we will simply show the key frames (the 1st frame) of shots selected for the abstraction in order to roughly determine out the quality of the video abstraction.

We will first show the experimental results for a Korean sitcom video clip (30 minutes long) consisting of 52,873 frames that were grouped into 239 shots. The target run-time of the abstraction was fixed at 2 minutes (3,600 frames) in this experiment. Note that the total number of possible video abstractions was, theoretically, 2^{239} in this experiment. We ran our abstraction algorithm four times while varying the cooling rate (ρ) of simulated annealing. The four generated video abstractions are shown in Fig. 4, in



Fig. 4. An example of video abstractions generated with different cooling rates.

which the key frames (or first frames) of the shots in each abstraction are shown along with their shot numbers and motion energies. Since the weights of the objective functions used to compute $G(X)$ were adjusted so as to be the same in this experiment, the selected shots equally satisfy the proposed seven constraints as much as possible. For example, the selected shots are uniformly distributed over the whole video, the number of frames in the abstraction is close to 3,600, visually similar shots are seldom selected together, and finally, shots with high motion energy are selected as shown in Fig. 4. We can find from this experiment that if a higher cooling rate is used in the abstraction process, an abstraction with higher $G(X)$ can be obtained as shown in Fig. 4. This is a characteristic of search algorithms that are based on simulated annealing, and it was also true in this experiment. Another finding from this experiment is that visually similar shots (for example, the 4th and 5th shots in Fig. 4 (b), and the 5th and 6th shots in Fig. 4 (c)) were not selected as a higher cooling rate was used as shown in Fig. 4 (d). This was due to the fact that the probability of selecting visually similar shots decreases when a higher cooling

rate is used (the slow annealing process) because of the objective function O_5 . Note that the execution time of the proposed algorithm is dependent on the cooling parameter (ρ) and the termination condition (ϵ), and that actually the average execution time for generating the video abstraction presented in Fig. 4 was less than 0.2 second on a Pentium PC.

The proposed abstraction algorithm can generate various video abstractions through adjustment of the weights of the constraints (or objective functions). However, casual users usually do not know the meaning of the low-level constraints in detail, so it may be very hard for them to adjust the weights of the constraints directly. Instead, they might have a higher-level requirement, such as “show me a video abstraction focusing on the main actor” or “show me a video abstraction focusing on the car race”. In order to map these high-level requirements to the weights of low-level constraints, some sophisticated artificial intelligence technologies that can totally recognize and understand the objects and events in the video clips are required. Since these technologies will not be available in the near future, a simple mapping rule can be used in the video archive system. For example, if the user wants a summary-style abstraction, the *well-distributed* and *good fit* constraints can be used to generate the abstraction. On the other hand, if the user wants to a highlight-style abstraction, the *highly-active* and *good fit* constraints can be mainly used. Although these mapping rules are simple and somewhat ad hoc, they can be used to generate various video abstractions with variable run-times without the use of sophisticated artificial intelligence technologies. Fig. 5 shows the results of experiments in which various video abstractions were generated by adjusting the weights of the constraints, with the video clip that is first 30 minutes of a famous movie clip titled “Terminator-2”. It consisted of 52,755 frames, and they were grouped into 525 shots. We abstracted this video clip into video clips consisting of 1,200 frames (40 seconds) while changing the weights of the objective functions. Fig. 5 (a) shows a summary-style abstraction, which shows the overall story of the video clip where W_1 and W_2 were set to one, and others are all zero, (b) is a highlight-style abstraction which shows the shots with high actions by setting W_2 and W_4 to one, and others are all zero, and finally (c) is a trailer-style abstraction which shows the highlight of video clip without showing the final part by setting W_2 , W_4 , and W_6 to one and the other constraints were set to zero. Although these abstractions were generated without *understanding* the contents of the video clip, they could be used as summary, highlight, and trailer abstractions of the long video clip. From this experiment, we can argue that the constraints formalized in this paper are general enough to generate various kinds of video abstractions by changing the weights of the objective functions.

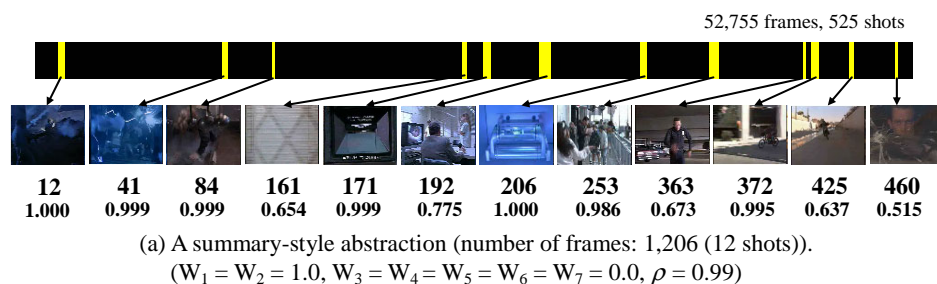


Fig. 5. Examples of video abstraction for “Terminator-2” (52,755 frames, 525 shots).

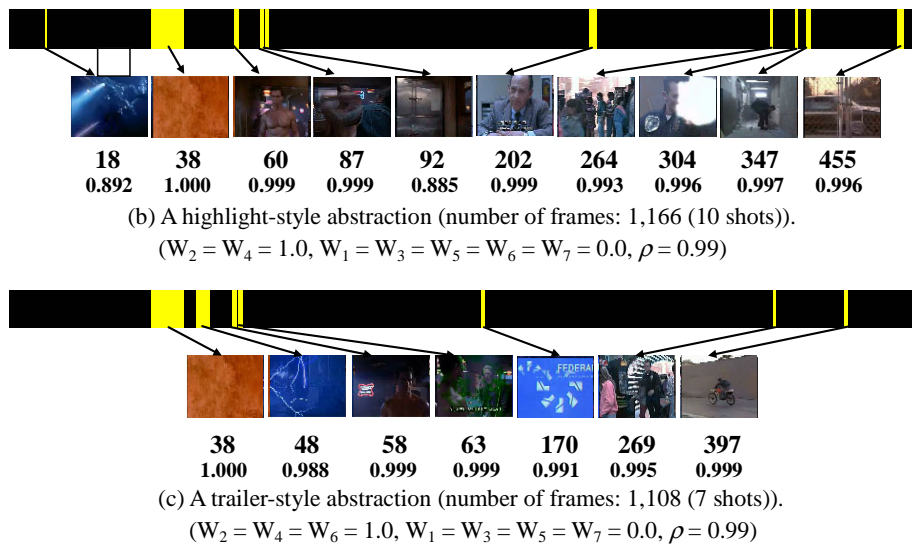


Fig. 5. (Cont'd) Examples of video abstraction for "Terminator-2" (52,755 frames, 525 shots).

4. COMPARISON WITH RELATED WORKS

Many other researches have attempted to devise a domain independent video abstraction algorithms, for example, VAbstract [3], Nam's work [5], the basic algorithm used in Video Manga [8], the CMU work [4], the desirable attributes of the video summary proposed in [9], and Saarela's work [2]. The scene/shot selection rules or heuristics used in these researches are summarized in Table 1. As shown in this table, the heuristics rules or constraints adopted in these researches and in ours are similar to each other. However, the ability to dynamically adjust the weights of the constraints to generate various kinds of video abstractions, and to find a set of shots satisfying these weighted constraints as much as possible using simulated annealing are the main features of our algorithm that differentiate it from the others. Of course, Saarela *et al.* [2] also argued that different domains require different types of summaries, and proposed a general framework in which summary creation is viewed as optimization with constraints. They explained that, by varying the constraints and cost function, one could apply their framework to any situation. However, they did not explain how to formalize these constraints or how to solve this optimization problem. This paper has formalized constraints and shown how a set of shots can be found that simultaneously satisfies the constraints as much as possible. There is also a system that generates a video abstraction based on the user's requirements [10], in which the user chooses a few important scenes according to the application and lets the abstraction algorithm automatically select the remaining important scenes in the video. However, since this algorithm tries to find scenes that are visually similar to the submitted scenes, it is not general enough to be used to abstract various genres of video abstractions although their experimental results on news video finding the anchor shots are noticeable.

Table 1. Comparison with related works.

	Scene/Shot Selection Rules or Heuristics	References
VAbstract (Moca)	<ul style="list-style-type: none"> ◆ High Contrast Scene ◆ High Motion Energy Scene ◆ Basic Color Composition ◆ Dialogue Scene 	[3]
Nam's Work	<ul style="list-style-type: none"> ◆ An Adaptive Non-Linear Sampling of a Video based on visual activities 	[5]
Video Manga	<ul style="list-style-type: none"> ◆ A Segment is important if it is both long and rare 	[8]
CMU Work	<ul style="list-style-type: none"> ◆ Introduction Scenes ◆ Similar Scenes ◆ Short Sequences ◆ Object Motion ◆ Bounded Camera Motion ◆ Human Faces and Captions ◆ Significant Audio ◆ Grayscale Video 	[4]
Microsoft Work	<ul style="list-style-type: none"> ◆ 4C (Conciseness, Coverage, Context, Coherence) 	[9]
Saarela Work	<ul style="list-style-type: none"> ◆ Natural Constraints <ul style="list-style-type: none"> → Minimum Shot Duration (at least 3.5 seconds) → Synchronization between the Video and Corresponding Audio Track → Avoiding Continuous Video ◆ Redundancy Constraints <ul style="list-style-type: none"> → Avoiding Redundant Scenes → Avoiding Multiple Appearance Scene 	[2]
Ours	<ul style="list-style-type: none"> ◆ Selecting a set of Shots that Satisfy the Following Constraints <ul style="list-style-type: none"> → Well-Distributed, Good Fit, Not-too-Short, Highly Active, Non-Redundant, Shot-Exclusion 	

5. CONCLUDING REMARKS

As digital video clips are being used in a wide range of applications on the Internet or Intranets, the ability to preview the highlights or a summary of a long video clip without viewing the whole clip has become an essential feature that a video-based server should provide. However, to automatically abstract (or summarize) a long video clip to obtain a shorter one requires the use of sophisticated artificial intelligence technology to map the low-level visual/aural features to high-level semantics. Since this technology will not be available in the near future, this paper has proposed a framework that lets the user dynamically express his/her requirements by adjusting the weights low-level constraints, and the abstraction algorithm finds a set of shots that satisfies these weighted

constraints as much as possible using a searching algorithm based on simulated annealing. Of course, the constraints proposed in this paper would not be the best ones for generating a good video abstraction, and the formalization for these constraints could be also modified. However, even if some constraints are modified or formalized using other equations, the proposed abstraction framework can still be used to dynamically generate video abstractions for various genres of video clips. We argue that although the low-level constraints formalized in this paper may not represent a high-level user's requirements completely, this approach is a good compromise between those used to obtain abstraction schemes based on just pattern-matching of pre-defined low-level visual/aural features and abstraction schemes based on fully understanding the high-level video contents. The proposed scheme could be used to build a video-based server that can quickly generate a video abstraction dynamically based on the user's requirements.

REFERENCES

1. R. Otten and L. van Ginneken, *The Annealing Algorithm*, Kluwer Academic, Boston, MA, 1989.
2. J. Saarela and B. Merialdo, "Using content models to build audio-video summaries," in *Proceedings of the Electronic Imaging Conference SPIE '99*, 1999, pp. 338-347.
3. S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg, "Abstracting digital movies automatically," *Journal of Visual Communication and Image*, Vol. 7, 1996, pp. 345-353.
4. M. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding techniques," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 775-781.
5. J. Nam and A. H. Tewfik, "Event-driven video abstraction and visualization," *Multimedia Tools and Applications*, Vol. 16, 2002, pp. 55-77.
6. A. Hanjalic and H. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 9, 1999, pp. 1280-1289.
7. H. S. Chang, S. Sull, and S. U. Lee, "Efficient video indexing scheme for content-based retrieval," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 9, 1999, pp. 1269-1279.
8. S. Uchihashi, J. Foote, A. Girgensohn, and J. S. Boreczky, "Video Manga: generating semantically meaningful video summaries," in *Proceedings of ACM Multimedia Conference*, 1999, pp. 383-392.
9. L. He, E. Sanocki, A. Gupta, and J. Grudin, "Auto-summarization of audio-video presentations," in *Proceedings of ACM Multimedia Conference*, 1999, pp. 489-498.
10. J. H. Oh and K. A. Hua, "An efficient technique for summarizing videos using visual contents," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2000, pp. 1167-1170.

Jinguk Jeong (鄭鎮國) is a Ph.D. candidate in the Department of Computer Science at Sogang University, Seoul, Korea. His research interests include multimedia computing system, content-based multimedia indexing and retrieval algorithm, and MPEG-7. He received his B.S. and M.S. in Computer Science from Sogang University in 1998 and 2000, respectively.

Jongho Nang (浪鍾鎬) is a Professor in the Department of Computer Science at Sogang University. His research interests are in the fields of multimedia systems, digital video library, and Internet technologies. He received his B.S. degree from Sogang University, Korea, in 1986 and M.S. and Ph.D. degree from KAIST, in 1988 and in 1992, respectively.

Hojung Cha (車浩晶) is currently a Professor in Computer Science at Yonsei University, Seoul, Korea. His research interests include multimedia computing system, multimedia communication networks, wireless and mobile communication systems, and embedded system software. He received his B.S. and M.S. in Computer Engineering from Seoul National University, Korea, in 1985 and 1987, respectively. He received his Ph.D. in Computer Science from University of Manchester, England, in 1991.