An Authoring Tool Generating Various Video Abstractions Semi-automatically

Jongho Nang[1], Jinguk Jeong[1],
Myung-hwan Ha [2], Byunghee Jung [2], and Kyeongsoo Kim [2]

[1] Dept. of Computer Science, Sogang University, 1 Shinsoo-Dong, Mapo-Ku
Seoul 121-742, Korea
jhnang@ccs.sogang.ac.kr
[2] KBS Technical Research Institute, 18 Yoido-dong, Youngdungpo-gu, Seoul 150-790, Korea

Abstract. Video abstraction is a short version of the original video, and it is used to deliver the summary *or* the highlight of video contents as quickly as possible. According to the objectives of the video abstraction, the set of shots constituting the abstraction should be different. This paper presents an authoring tool that could automatically generate various kinds of video abstractions according to the objectives of the abstraction, and allows the author to easily edit the resulting abstraction manually. In the proposed automatic video abstraction algorithm, a simulated annealing algorithm is used to select the set of shots that simultaneously satisfies several constraints, such as *well-distributed*, *well-fitting*, *high-activities*, and *non-duplicated* (or *concise*), as much as possible. This set of shots could be used as a final video abstraction, or a candidate of the video abstraction that the author could replace with just a drag-and-drop of the key frame of the selected shot on the time-line of target video abstraction.

1 Introduction

Although there have been many researches [2,3,4,7,8,9] to abstract a long video clip to a shorter version automatically, they are usually based on the domain specific heuristics so that their usages would be limited only to the specific domains. For example, the heuristics used to abstract the action movie clips could not be used to abstract the documentary video clips. This problem could be resolved if the abstraction process has reflected the user's requirements for the video abstraction as much as possible, and let the author additionally edit the resulting video abstraction manually.

This paper proposes a subjective video abstraction algorithm that generates various video abstractions according to the user's requirements, and an authoring tool that helps the author to easily edit the generated video abstraction manually. It first analyzes a set of conditions (or constraints) that a good video abstraction should satisfy, and formalizes them as the objective functions. The proposed main constraints that the set of selected shots constituting the abstraction should satisfy are

Y.-C. Chen, L.-W. Chang, and C.-T. Hsu (Eds.): PCM 2002, LNCS 2532, pp. 311–318, 2002.

well-distributed, *well-fitting*, *high-activities*, *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 so called an NP-complete (O(n^2)), this paper proposes a shot selection algorithm based on the simulated annealing [6] in order to generate a video abstraction, which could satisfy the user's requirements represented by the weights of the objective functions as much as possible, in a polynomial time. After the set of shots are selected, their key frames are placed on the time-line of the video abstraction in the authoring tool and could be replaced by author by just a drag-and-drop operation. We have implemented proposed authoring tool on MS-WINDOWS platform, and this paper explains its main functions with easy-to-use GUI, after presenting the basic idea of proposed video abstraction algorithm together with some experimental results.

2 A Video Abstraction Algorithm

2.1 Problem Definition

The main process to make a video abstraction is to select some important shots from the video clip after it is segmented into a set of shots. Let $V = \{v_i \mid 1 \le i \le n\}$ be the video clip consisting of n shots, and $X = \{x_i \mid x_i \in V, 1 \le i \le k, 1 \le k \le n\}$ be its abstraction consisting of k shots. Then, the video abstraction process is to select the k shots from n shots, and the number of different video abstractions consisting of k shots would be ${}_nC_k$. Since the range of k would be $1 \le k \le n$ accroding to the target run-time of abstraction, the total number of different video abstractions would be ${}_nC_0 + {}_nC_1 + {}_nC_2 + \ldots + {}_nC_n = 2^n$. A good video abstraction would be the one among these 2^n candidates that satisfies the desirable conditions as much as possible. In the proposed abstraction scheme, some low-level visual constraints that a good abstraction should satisfy are idnetified, and the users express their requirements by adjusting the importantces of these constraints. Although this approach could not meet the high level user requirements exactly, the abstraction process could be performed almost automatically while meeting the user's requirements as much as possible. Let us formalize these constriants in the following section.

2.2 Formalizing the Constraints

In the proposed 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. Let us also assume that the target run-time of the abstraction that is required by the user dynamically is denoted as T.

- *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 on the entire video content. It is especially important for the summary-style abstraction. If the intervals between selected shots in X are similar to each other, we can say it is well-distributed abstraction. The average interval between the shots in X, $\mu,$ could be computed as follows;

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

Since a lower variance implies a higher well-distributed abstraction, its inverse function is defined as an objective function for Well-Distributed constraint, O_1, as follows;

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

- *Well-Fitting (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 could be a good abstraction. This constraint could be formalized as follows using a transcendental function, *sech(x)*, where $C_1 = \frac{2}{T}\ln(2+\sqrt{3})$.

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

- *Not-too-Short (O_3)*

The minimum run-time of a continuous shot should be at least 3.5 seconds to be processed completely by brain [8]. On the other hand, if the shot run-time is greater than 3.5 seconds, the shot has an equal opportunity to be selected as a member of the abstraction. To express this constraint, $f(L_i) = \frac{L_i + C_2 - \left|L_i - C_2\right|}{2 \cdot C_2}$ is used to denote the suitability of the shot x_i for the abstraction, where C_2 is fixed as 3.5. The average value of $f(L_i)$ for all shots in X is defined as an objective function for Not-too-Short constraint as follows;

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

- *Highly-Active (O_4)*

If there are a lot of object motions in the shot, it is usually regarded as an important one so that it should be included in the abstraction. It is a commonly used heuristics in the video abstraction researches as in [7, 8, 9]. To express this constraint, *the motion intensity index* of the shot [4] for x_i, $g(x_i)$, is used to represent the degree of activity of the shot. The average motion intensity index of the shot x_i in X is defined as an objective function for Highly-Active constraint as follows;

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

where $g(x_i)=\frac{1}{L}\sum_{j=b}^{e}\sum_{m,n}\left|m_j(m,n)\right|$ and $m_j(m,n)$ is the j-th frame of motion sequence within the i-th shot unit and L is the length of the analysis window beginning at b-th frame and ending at e-th frame.

- *Concise or Non-Redundancy (O_5)*

In order to include more information in the video abstraction, the similar shots should not be selected repeatedly in the abstraction process. This heuristics has been adopted in a couple of video abstraction researches [1, 2, 8]. We also adopt this heuristics in the proposed scheme, and the degree of the visual differences between the shots in X is used to denote the suitability of X for the video abstraction. Actually, the historgram differences between the key frames of shots are used to compute the visual differences of the shots in X. The overall visual differences between all shots in X could be computed using following equaltion ;

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

where $\vec{f}_z^{\,i}$ is the color histogram of key frame of i-th shot.

- *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 could be the candidates for the video abstraction [7]. In order to express this constraint, the function, $h(x_i)=\left(\frac{E-E_i-\left|E_i-C_3\right|}{2\cdot(E-C_3)}+\frac{1}{2}\right)$, 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 Shot-Exclusion constraint as follow ;

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

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

- *Non-Bias (O_7)*

If the run-time of the shot x_i is too long, a relatively small number shots could be included in the video abstraction. To avoid this problem, the shot with too long run-time would be excluded in the video 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 shot in X ($\max(L_i)$) is used to denote the suitability of the X for the abstraction. Since less difference implies more suitability, its inverse function is defined as an objective function for Non-Bias constraint as follow;

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

Since the range of the return values of these objective functions would be different from each other (for example, the ranges would be [0:∞] for O_1 and O_7, while others are not), they should be normalized to the same range in order to evaluate their suitability precisely. In the proposed scheme, a normalizing function, $f(x)=\frac{x}{x+1}$, is used to normalize the objective function values to [0:1]. The normalized objective functions could be computed using following equation, and they are used to compute the overall suitability of the *X* for the abstraction.

$$O'_n(X)=\frac{O_n(X)}{O_n(X)+1}$$

A good video abstraction would be a set of shots that simultaneously satisfies above constraints as much as possible. However, since the relative importance of the constraints represented by objective functions are dependent on the aims of abstraction, the objective function values should be weighted in computing the overall suitability of X for the abstraction. Thus, the process to make a good abstraction could 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

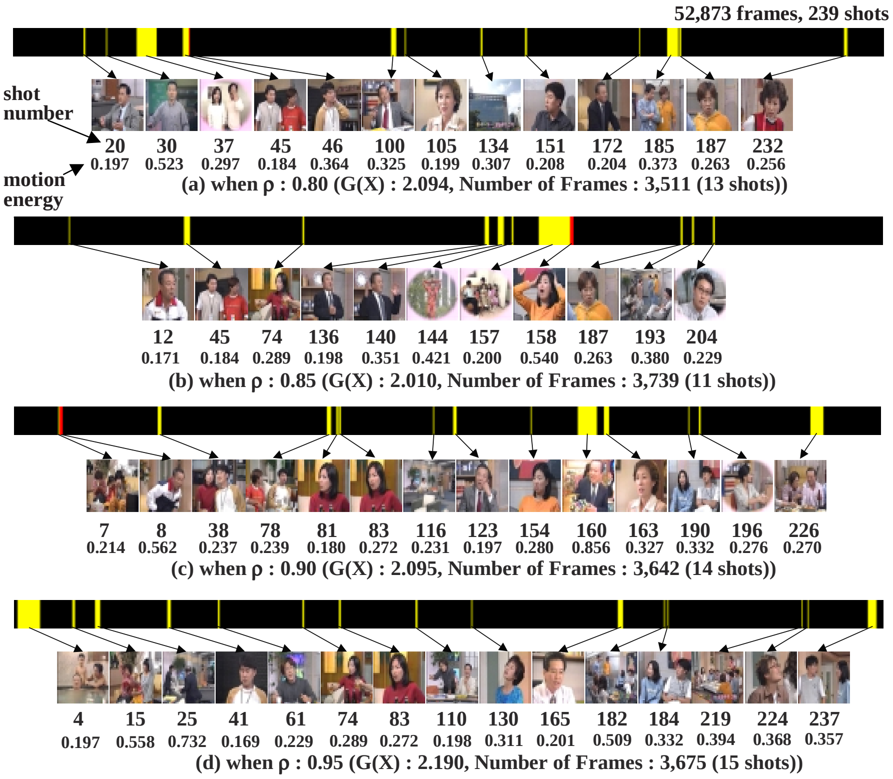
Since the number of possible video abstractions for the video clip consisting of *n* shots is *2^n*, it would be very hard to generate a good video abstraction in a polynomial time as mentioned before. It is a sort of combinatorial optimization problem to find *X* among *2^n* candidates that maximizes the overall objective function *G(X)*. There have been several search algorithms that could find the near-optimal solution of combinatorial optimization problems. Simulated annealing algorithm [6] is one of such search algorithms that could find a sub-optimal solution in a polynomial time. 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 among *2^n* candidates, and its overall objective function value (*G(* X_1)) is computed. Then, another set of shots, X_2, is selected again 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 of the good abstraction. Otherwise, X_2 is accepted as the candidate of the good abstraction with the probability of $e^{\frac{-(G(X_2)-G(X_1))}{T}}$, where *T* is an initial temperature which controls the annealing process. Let the accepted abstraction as the candiate of the good abstraction, and repeat the above process while decreasing the temperature *T* until it is less than predefined temperature *ε*. In this annealing process, when *T* is high enough, the probability of accepting the worse abstraction than current one is also high.

However, as the annealing process is being progressed (i.e., *T* is being decreased), the probabilty to accept the worse abstraction as the candidate of good abstraction is also decreased. This stochastical annealing process helps to avoid the locally optimal abstraction, and to eventually find a globally near optimal abstraction in a reasonable time.

2.4 Experimental Results and Analyses

Let us show an experimental result on the Korean drama video clip (30 minutes long) consisting of 52,873 frames that are grouped into 239 shots. The target run-time of the abstraction is fixed as 2 minutes (3,600 frames) in this experiment. Note that the total number of possible video abstractions is theoretically 2^{239} in this experiment.



(a) when ρ : 0.80 (G(X) : 2.094, Number of Frames : 3,511 (13 shots))

(b) when ρ : 0.85 (G(X) : 2.010, Number of Frames : 3,739 (11 shots))

(c) when ρ : 0.90 (G(X) : 2.095, Number of Frames : 3,642 (14 shots))

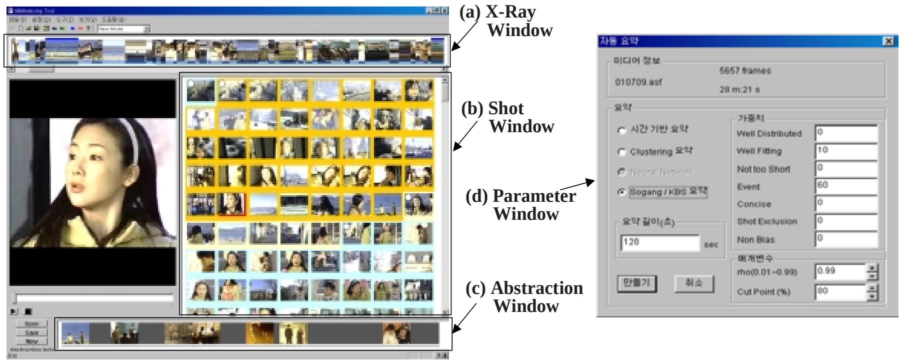
(d) when ρ : 0.95 (G(X) : 2.190, Number of Frames : 3,675 (15 shots))

Fig. 1. An example of video abstractions generated with different cooling rates

We have experimented our abstraction algorithm four times while varying the cooling rate (*ρ)* of simulated annealing. The generated four video abstractions are shown in Figure 1, in which the key frames (or first frames) of the shots in each abstraction are shown with their shot numbers and motion energies. Since the weights of the objective functions to compute *G(X)* are adjusted 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 similar to *3,600*, the visually similar shots are seldom selected together, and finally the shots with high motion energies are selected as shown in Figure 1. We can find from this experiment that the visually similar shots (for example, 4th and 5th shot in Figure 1-(b), 5th and 6th shots in Figure 1-(c)) are disappeared as the cooling rate is raised as shown in Figure 1-(d). It is due to the fact that the probability of selecting the visually similar shots are lowered as the cooling rate is raised (slow annealing process) because of the objective function O_5.

3 An Authoring Tool

Although the video abstraction generated by the proposed algorithm could reflect the user's requirements as much as possible by adjusting the weight of the constraints, the resulting abstraction would not be satisfiable because it was generated without a full understanding of the video contents. This problem could be resolved if the selected shots are used as the candidates for the abstraction and replaced with more suitable shots manually by the author, as in the authoring tool presented in this section. In the proposed authoring tool, the video clip is first segmented into a set of shots automatically, the candidate shots for the video abstraction among them are selected using the proposed algorithm, and finally they are edited by the author manually to produce a final video abstraction using the interface as shown in Figure 2. It also provides an user interface to modified the shot boundaries manually, since the automatic indexing algorithm could not find all shot boundaries completely. After the vide clip is segmented into a set of shots, the thumbnail images of the 1st frame of the shots (key frames) are enumerated as shown in Figure 2.



미디어 정보
010709.asf
5657 frames
28 m:21 s

요약
○ 시간 기반 요약
○ Clustering 요약
○ Neural Network
● Sogang / KBS 요약

요약 길이(초)
120 sec

가중치
Well Distributed 0
Well Fitting 10
Not too Short 0
Event 60
Concise 0
Shot Exclusion 0
Non Bias 0

매개변수
rho(0.01~0.99) 0.99
Cut Point (%) 80

Fig. 2. The User Interface of Authoring Tool

4 Concluding Remarks

As the digital video clips are being used in wide range of applications on Internet or Intranet, the capability to preview the highlight or summary of the long video without viewing the whole video clips becomes an essential feature that a video-based server should provide. However, to automatically abstract (or summarize) the long video clip to a shorter one requires a sophiscated artificial intelligence technology to map the low-level visual/aural features to hight-level semantics. Since this technology would not be available in the near future, this paper proposes other framework that let the user/author express the his/her requirements by the weights of the constraints that a good video abstraction should satisfy, and the abstraction algorithm find a set of shots that satisfies these weighted constraints as much as possible using the 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, although some constraints are modified or formalized with other equations, the proposed abstraction framework could still be used to dynamically generate a video abstraction for various genre of video clips with respect to them. We argue that although the video abstraction generated with the proposed abstraction algorithm could not meet the user's requirements directly, this approach would be a good compromise between the abstraction schemes based on just pattern-matching of pre-defined low-level visual/aural features and the abstraction schemes based on fully understanding of high-level video contents.

References

[1] H. Chang, S. Sull and S. Lee, "Efficient Video Indexing Scheme for Content-Based Retrieval," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol.9, No. 8, pp. 1269-1279, 1999.

[2] 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, No.8, pp.1280-1289, 1999.

[3] L. He, E. Sanocki , A. Gupta , J. Grudin, "Auto-Summarization of Audio-Video Presentations," *Proceedings of ACM Multimedia Conference*, pp. 489 - 498, 1999.

[4] J. Nam and A. H. Tewfik, "Video Abstract of Video," *Proceedings of the 3rd IEEE International Workshop on Multimedia Signal Processing (MMSP '99)*, pp. 117-122, Sep. 1999.

[5] J. Nang, S. Hong and Y. Ihm "An Effective Video Segmentation Scheme for MPEG Video Stream using Macroblock Information," *Proceedings of the ACM Multimedia Conference 1999*, ACM Press, 1999, pp.23-26.

[6] R. Otten, and L. van Ginneken, *The Annealing Algorithm*. Kluwer Academic, Boston, MA, 1989.

[7] S. Pfeiffer, R. Lienhart, S. Fischer and W. Effelsberg, "Abstracting Digital Movies Automatically," *Journal of Visual Communication and Image*, Vol. 7, No. 4, pp.345-353, 1996.

[8] J. Saarela and B. Merialdo, "Using Content Models to Builds Audio-Video Summaries," *Proceedings of the Electronic Imaging Conference SPIE'99*, 1999.

[9] M. Smith and T. Kanade, "Video Skimming and Characterization Through the Combination of Image and Language Understanding Techniques," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp.775-781, 1997.