Computational Video Editing Model based on Optimization with Constraint-Satisfaction

Ryo OGATA, Yuichi NAKAMURA, Yuichi OHTA
IEMS, University of Tsukuba, Tsukuba, 305-8573 JAPAN
{ogata, yuichi}@image.esys.tsukuba.ac.jp

Abstract

This paper presents a novel video editing model. In this model, a video is considered as a sequence of small boxes each of which has a length of 0.5 or 1 second, and editing is defined as the problem of filling each box with an appropriate shot. Each editing rule is implemented independently of the others, as an evaluation function or a constraint for choosing shots and arranging shots. This formalism enables easy and systematic editing investigation by including or excluding editing rules. We implemented this model in an actual video editing system and verified its usefulness for multi-angled videos.

1 Introduction

Movies and TV programs are carefully edited by professional editors or directors. This process takes a long time, even longer than filming. Editing time and the cost incurred thereby seriously limit the applications of video-based media.

To cope with this problem, our research aims to introduce a computational editing model for videos taken by multiple cameras, *i.e.,* multi-angle videos, and to enable non-professionals to produce videos of good quality.In this model, a video is considered as a sequence of small boxes each of which has a length of 0.5 or 1 second, and editing is defined as the problem filling each box with an appropriate shot. Based on this model, our objective is to support the generation of a variety of editing patterns or a simulation of typical editing patterns in movies and TV programs.

We implemented the model on our video capturing and editing system[4, 5], that is composed of multiple PCs with hardware MPEG encoders and decoders, a video switcher controlled by a host PC, etc. Generated editing patterns are shown on a PC monitor, and a selected pattern can be shown by TV monitor. Through trial experiments, we verified that our editing model generally works for multi-angle videos, and that our set of rules produce good editing patterns.

2 Editing and Camera Switching

As the first step toward automated editing, we investigated the problem of "camera switching" for a scene taken by multiple cameras. Although this problem is only a portion of editing, we believe our model shows good potential for extension into more complicated editing takes such as story editing.

For the camera switching problem, so far, event-driven or reflective algorithms that switch views by selecting the most relevant camera have been proposed[1, 2]. For example, a system selects " a bust shot of person A" when "person A speaks". However, such these event-driven editing does not have the potential either for obtaining good editing or for investigating the rules of editing. Suppose two events, say events A and B, occur almost simultaneously. Editing result may change if the occurrence time of event A is slightly earlier or later than the other. Systematic evaluation of such effects is difficult. Adjusting to conflicting requests is also difficult in this type of editing model. A number of editing techniques that have been accumulated so far have many different purposes, for example, giving accurate information, enhancing the entertainment value , directing attentions to a particular point, etc. Integrating these often conflicting objectives into a event-driven algorithm is almost impossible.

In contrast to such approaches, we are investigating a novel method based on optimization with constraint-satisfaction. This model searches for the best editing patterns out of all possible shot combinations that satisfy given constraints. As a result, this model has greater flexibility in integrating many editing rules simultaneously. Various objectives and preferences can be introduced into this model as evaluation functions or constraints. Although a thorough search can potentially cause combinatorial explosion, the number of candidates can be suppressed and made tractable through the of use of effective constraints.

3 Computational Editing Model

3.1 Model Definition

First, we consider a video as a sequence of short video segments, each of which has a length of 0.5 or 1 second. This partition of a video is based on the following observation:

- Old movies rarely have shots whose length is shorter than 1 second. This proves that a sequence of shots, each of which is equal or longer than 1 second is powerful enough for composing movies or a TV programs.
- Frequent shot changes induce perceptual strain in the audience, and should not be used without clear intention to produce special effects. In this sense, we can consider a combination of two or more consecutive shots, each of which is less than 1 second as a compound shot that gives a special effect[1]. Consequently, we can assume that each shot is equal or longer than 1 second.

Next, we define video editing as the problem of assigning an appropriate shot to each video segment. The possible patterns of assignments for an entire video are limited by constraints, and the quality of the resulting assignments is scored by evaluation functions.

The model is formally composed of five elements:

$$\text{Editing} = \{S, E, C, O, V\} \tag{1}$$

The explanation of the above terms is given below:

Shots (S): S is a set of shots *i.e.,* $S = \{s_0, \ldots, s_n\}$, where s_i is a shot, *e.g.,* "a bust shot of person A", "a long shot of person B", etc.

Video (V): V is a sequence of video segment units, *i.e.*, $V = \{v_0, \ldots, v_{tmax}\}$, each of which has a length of 0.5 or 1 second. An appropriate shot (s_i) is assigned to each video segment (v_j).

Events (E): E is a collection of events (e_i), each of which occurs in the scene. An event is something important to be watched or to be a trigger for view switching, for example, "person A spoke", "person B laughed", "person A and B shook hands", and so on. If e_i occurs at time t with the certainty of 0.9, we denote it as $e_i(t) = 0.9$.

Evaluation (O): O is a set of evaluation (Objective) functions, each of which determines the appropriateness of an assignment of a shot (or shots) to a video segment (or segments). The criterion may be comprehensibility, entertainment quality, or many other factors.

Constraints (C): C is a set of constraints. Since combinatorial explosion will occur if we allow for all the possible editing patterns, the number of candidates must be limited before they are thoroughly evaluated. For this purpose, we utilize a constraints satisfaction mechanism, and implement some of the editing rules as constraints, *e.g.* "do not use shots (each of which is given a score below g_1) longer than t_n seconds". In this sense, the difference between an evaluation function and a constraint is only in their computational roles, and there are no clear semantic differences.

The objective of this model is optimization of G in the following formula.

$$G = \sum_{t=0}^{t_{max}} \sum_{i=0}^{N_o} o_i(t) \tag{2}$$

In other words, the objective is to find the best assignment of shots to video segments that maximizes evaluation value G based on O and satisfying the constraints C.

3.2 Flow of Computation

Figure 1 shows the flow of computation. The flow is mainly composed of three steps: *pre-scoring*, *candidate searching*, and *post-scoring and selection.*

In the pre-scoring step, the relevance of each shot is evaluated for each video segment. Each shot at each time is scored based on the events occurring around that time by evaluation functions. This evaluation does not concern the editing quality of a combination or a sequence of shots.

In the candidate-searching step, based on the given scores and constraints, the possible editing candidates are discovered. With constraints strong enough to suppress most of unusable patterns, all of the candidates can be numbered. As shown in the experiments, we often obtain more than one million candidates, which can be, however, handled with an ordinary PC.

Then, in the post-scoring step, each candidate, or shot sequence, is scored by evaluation functions. This evaluation takes account of the editing quality as a sequence of shots. Combinations of consecutive shots can be evaluated only in this step, since all the shots in a sequence are instantiated at the previous step. Finally, the editing pattern(s), *i.e.*, a shot sequence or sequences that received the highest score, is/are chosen.

[1] At the current stage, we did not create a category for such shots. This is left for future work.

4 Events, Evaluation Functions, and Constraints Definition

4.1 Events

We need to consider a variety of events that might have relevance to video editing, such as speeches, various kinds of motion, facial expressions, object movements, and so on. Table 1 shows the type of events we are currently considering. They are events that are often focused on in videos, or they are often used for triggers for view switching.

A collection of those events is given for each scene. Since we wanted to concentrate on the editing model, we made the above list manually, some of these events can be automatically detected with good reliability. Other types of events were left for future works.

4.2 Variety of Constraints and Evaluation Functions

We need to consider a variety of factors for making good videos. We are currently concentrating on the following three aspects:

Focus: The focus of attention should allow the audience to receive important information. This is the most common request for video editing, and we can think of natural preferences, such as "show the person who is speaking", "show the object pointed by someone", etc.

Not-Misleading: A shot or a combination of shots often suggests something that has not really occurred in the immediate scene. The "montage" is famous for explaining this effect. Although this is powerful and fundamental function of video editing, we must be careful to avoid misleading composition. As a typical example, the 180 degree system (imaginary line) is a good guideline for providing spatially consistent views. If it is violated, the audience may be confused with spatial arrangements or motion direction.[2] Table 2 shows some examples that would cause misunderstanding.

Table 1: Events used in our experiments

speech: The fact of speaking, starting speech, or ending speech, is one of the most important clues. The spoken words are also good clues that specifying where the focus is.

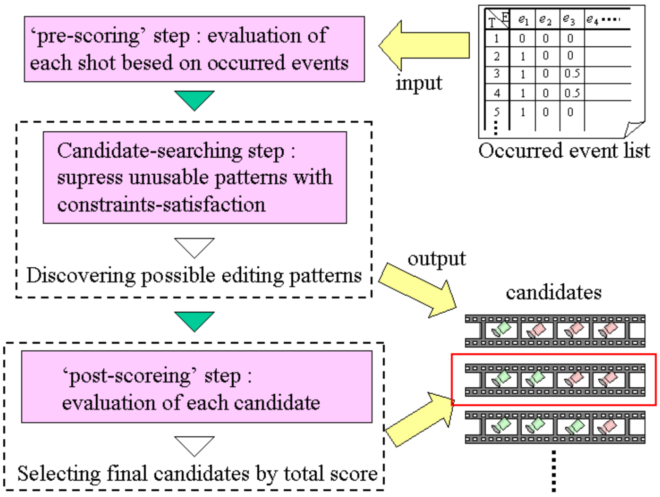
gestures: Movements, especially deictic movements and illustrations[a], strongly suggest where the focus of attention is.

hand manipulations: The current target scene of our system is conversation or meeting around a table. In such a situation, hand manipulations of objects are good clues for focus detection.

facial expression and head motion: Facial expressions and head motions, such as nodding and turning the head, express listeners' attitudes, and therefore strongly suggest where to look.

touching between persons: Body touching, such as tapping, touching, hand shaking, etc. are also key events to be looked at.

[a] These categories are given in [3]


'pre-scoring' step : evaluation of each shot besed on occurred events
input
Candidate-searching step : supress unusable patterns with constraints-satisfaction
Discovering possible editing patterns
output
candidates
'post-scoreing' step : evaluation of each candidate
Selecting final candidates by total score
Occurred event list

T \ E	e_1	e_2	e_3	e_4 ⋯
1	0	0	0	
2	1	0	0	
3	1	0	0.5	
4	1	0	0.5	
5	1	0	0	
⋮				

Figure 1: Flow of computation

Perceptual load: A good video gives an appropriate number of stimuli to produce a reasonable perceptual load. Too simple a composition is boring, while too many stimuli cause considerable perceptual strain. A good illustration of perceptual load is the issue of shot length. A succession of short-length shots, *e.g.,* less than 1 second, strains audience perception, while a long-length shot, *e.g.,* 30 seconds, can easily become boring. Table 3 shows examples that causes inadequate perceptual load.

4.3 Implementation of Constraints and Evaluation Functions

As mentioned above, there is no clear distinction between an evaluation function and a constraint. The computa-

[2] Although the 180 degree system is a good practical guideline, it is not an absolute rule. In Yasujiro Ozu's movies, we can find editing examples that violate the 180 degree system, and they are not confusing since the shots are well organized.

Table 2: Examples of negative effects caused by inadequate editing (misunderstanding)

Violation of the 180 degree system: This leads to misunderstanding of spatial arrangements or motion directions.

Inappropriate shot changes: Shot changes suggest focus-of-attention changes. An inappropriate shot change directs the audience's attention to a wrong portion.

Connecting shots with similar angles: Jump cuts between similar angles create the illusion of apparent motions.

Table 3: Examples of negative effects caused by inadequate editing (perceptual load)

Too few shot changes: This often makes a boring video.

Too many shot changes: This requires too much perceptual strain.

Too many close shots: This requires too much perceptual strain. perception.

Inappropriate shot changes: This disturbs the natural understanding of a story and causes considerable strain.

tional feasibility of our editing model, however, owes to the constraints' role in reducing the number of possible editing patterns.

For convenience, evaluation functions are used in two ways. One is pre-scoring , which evaluates every shot at each time. The other is post-scoring , which evaluates a combination of shots.

The upper part of Table 4 shows examples of pre-scoring evaluation with the setting of shots and events as shown in Table 5. For example, o_1 gives scores to shot s_3 ("bust shot of person A") at time t, if person A speaks at time t. This type of evaluation can be done independently of preceding or succeeding shots.

The lower part of Table 4 shows examples of post-scoring evaluation, where o_6 gives the preference for avoiding negative effects shown as the third example in Table 2. Each evaluation function can use the information from the preceding and succeeding video segments, since the value of each video segment is already instantiated after the candidate searching step.

Table 4: Examples of pre-scoring functions (upper) and post-scoring functions (lower): A vector in each figure shows a collection of the values for each shot at each time.

$o_{1A}(t)$	$[0, 0, 10e_1(t), 0, 0, 0]$
$o_{1B}(t)$	$[10e_1(t), 0, 10e_1(t), 0, 0, 0]$
$o_{2A}(t)$	$[0, 0, 0, 0, 10e_2(t), 0]$
$o_{2B}(t)$	$[10e_2(t), 0, 0, 0, 10e_2(t), 0]$
$o_3(t)$	$[0, 0, 15e_1(t+1), 0, 0, 0]$
$o_4(t)$	$[0, 0, 0, 0, 15e_2(t+1), 0]$
$o_{5A}(t)$	$[0, 30e_3(t), 0, 0, 0, 0]$
$o_{5B}(t)$	$[0, 50e_3(t), 0, 0, 0, 0]$
$o_6(t)$	$[0, 0, 0, 50(e_2(t) \wedge e_4(t)), 0, 0]$
$o_7(t)$	$[0, 0, 0, 0, 0, 50(e_1(t) \wedge e_5(t))]$

o_8	decrease the score by 10 points, if s_3 and s_5, or s_4 and s_6 are connected
o_9	increase the score by 15 points, if bust shots of two persons have the same or similar length

Table 5: Shot and event setting for experiments: This table shows shots, events, constraints, and evaluation functions. For the evaluation functions, .

shots (S)	
s_1	long shot of two people
s_2	close shot of table-top
s_3	bust shot of person A
s_4	bust shot of person B
s_5	over-the-shoulder shot of person A
s_5	over-the-shoulder shot of person B

events (E)	
e_1	person-A is speaking
e_2	person-B is speaking
e_3	a keyword is spoken that refers an object or a work on a table
e_4	person-A is nodding
e_5	person-B is nodding

We used a constraint-satisfaction library called "iZ-C" [6] in the candidate searching step, and implemented editing rules as constraints among integers and high-level constraints by their combinations, such as "equal, "greater, "occurs n times", etc. Table 6 shows examples of constraints. For example, c_1 prohibits any video segment equal or shorter than the threshold. The threshold was set to 2 seconds in our experiments.

4.4 Parameter Setting

There exist many parameters in evaluation functions and constraints as shown in the previous section. Table 4 and 6 show parameter examples in our experiments. In the current configuration, the scores given by each evaluation function range between 0 and 100, and scores over 50 points are used in cases of strong preference. The parameter values in the evaluate functions and constraints are empirically determined through experiments. As the next step, learning by showing good examples is a possible way of acquiring

Table 6: Examples of constraints

c_1	prohibits shots equal to or shorter than 2 seconds
c_2	prohibits shots that have scores equal to or less than 0 continue more than 3 seconds
c_3	stipulates that the establishing shot (s_1) must be contained in the first 10 seconds

these parameters. This problem is left for future work.

5 Experiments

We filmed a conversation scene between two people for about 2 minutes using multiple cameras they are corresponding to the first 20 seconds of the video. They talked about materials for data storage, with sometimes manipulating a removable medium. We generating various editing patterns by including/excluding evaluation functions or constraints, and thereby verified that our computational model really works to produce a variety of editing results from the filming of an actual scene.

Table 7 shows six different conditions for editing. The evaluation functions (o_i) and constraints (c_j) are already shown in Tables 4 and 6. The right column in Table 7 gives the number of candidates remaining after constraint-satisfaction step.

Figure 2 shows editing patterns, each of which obtained the best score under its respective condition. The difference between edit1 and edit2 is the parameter value of evaluation function o_5. By making the score for the table shot (s_2) greater, edit2 employs the shot from 8 seconds through 11 seconds. The difference between edit2 and edit3 is constraint c_2. By adding c_2, the number of editing candidates was greatly reduced from 1,002,156 to 596, while the same editing pattern was selected for both. This example shows the power of good constraints.

In edit4, evaluation functions o_4 and o_5 were added in order to show, by over-the-shoulder shots, listener's attitudes as well as the speaker's face. Shot s_5 is used from 13 seconds through 16 seconds. In practical editing, connecting two shots that have similar angles such as s_3 and s_5, or s_4 and s_6, is not preferable. Such shot changes create the illusion that objects on screen have quickly moved. To prevent such transitions, evaluation function o_6 is added in edit5. As a result, shot s_4 from 9 seconds through 13 seconds is replaced by shot s_3.

In edit6, we added constraint c_3 to present an "establishing shot". As a result, shot s_1 is inserted from 3 seconds through 6 seconds.

In edit7, the system gives a good score for the bust shot of a person before he/she begins to speak. A shot change preceding the beginning of a speech is conventional technique that delineates who talks for what purposes, and it helps the audiences understanding of the story. For this purpose, evaluation functions o_3 and o_4 are used.

Table 7: Conditions for editing and the number of candidates

	applied C,O	number of candidates
edit1	o_{1A},o_{2A},o_{5A},c_1	1,002,156
edit2	o_{1A},o_{2A},o_{5B},c_1	1,002,156
edit3	o_{1A},o_{2A},o_{5A},c_1,c_2	596
edit4	o_{1A},o_{2A},o_{5B},o_6,o_7,c_1,c_2	1,752
edit5	o_{1A},o_{2A},o_{5B},o_6,o_7,c_8,c_1,c_2	1,752
edit6	o_{1A},o_{2A},o_{5B},c_1,c_3	459,816
edit7	o_3,o_4,o_{5B},c_1,c_2	610
edit8	o_{1B},o_{2B},o_{5B},c_1,c_3	459,816
edit9	o_{1B},o_{2B},o_{5B}	about3 $\times 10^{15}$

In edit8, instead of o_{1A} and o_{2A}, the system used o_{1B} and o_{2B} that give medium score to long shot s_1 as well as bust shots s_3 and s_4. This setup led that long shot s_1 was often selected when both of two persons were simultaneously speaking.

Edit9 shows the importance of constraints. Edit9 is the editing result with the same evaluation functions of edit8, but any constraints aren't used. This setting causes frequent shot changes at almost all seconds, since the shot that obtains the best score changes frequently. Consequently, the editing result is choppy and we cannot keep watching the movie clip from edit9.

As shown in the experiments, the model has good ability to generate a variety of editing patterns by including/excluding each editing rule. While this model is a novel and good model for editing, there are still problems to be solved.

- Without adequate constraints, the number of candidate editing patterns will explode. In some cases, better algorithms for suppressing useless editing patterns are needed.
- Our current editing model is designed for offline processing only. For some purposes, more flexible editing will be required allowing online processes in which new video segments or information will be fed into the editing system in succession.

As well as tackling these problems, we will gradually move to the next step, which involves systematic generation of editing patterns and comparison with editing patterns in movies and TV programs.

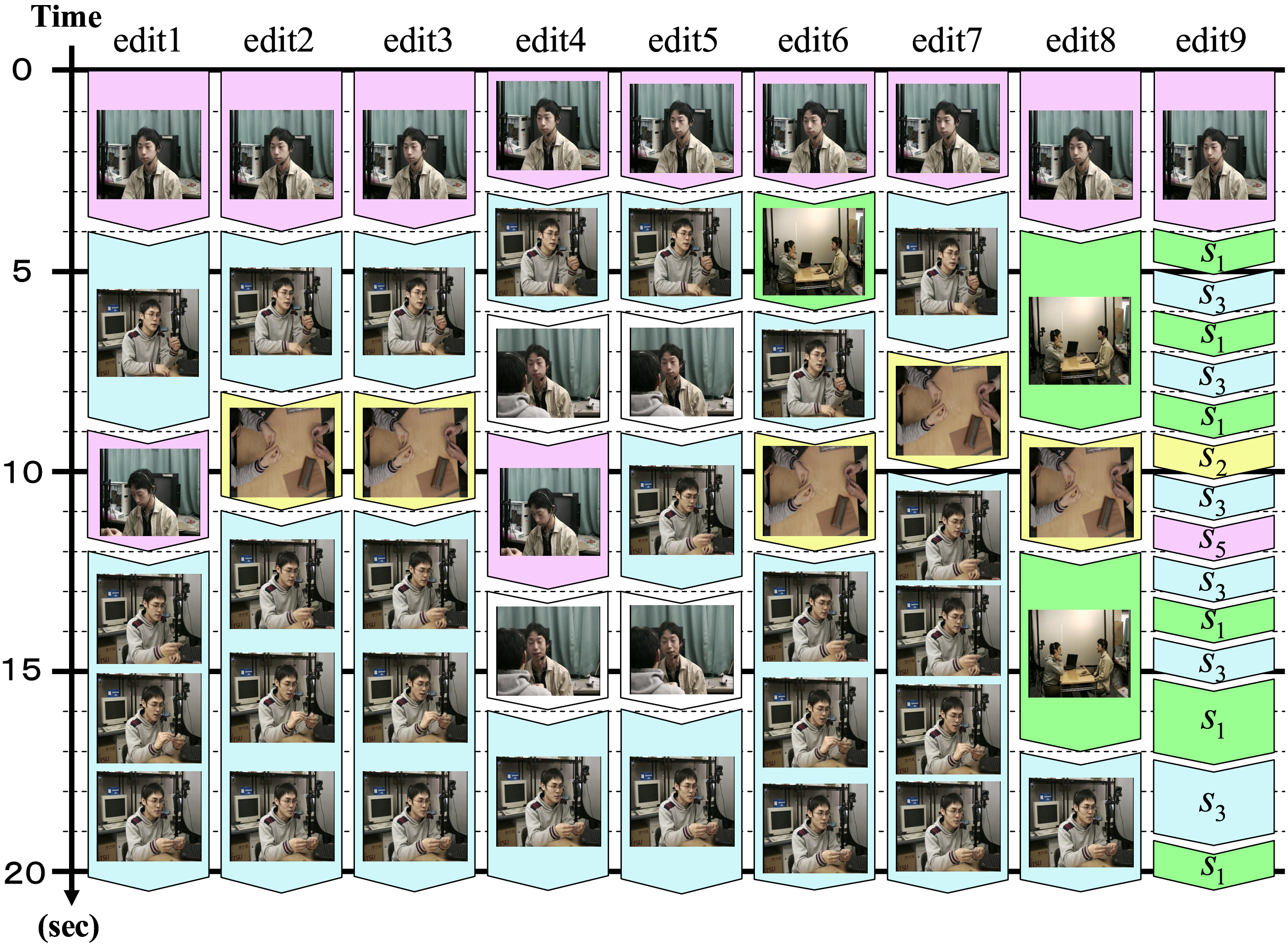


Figure 2: Editing results with six different conditions

6 Conclusion

This paper introduced a novel video editing model based on optimization with constraint satisfaction. We can generate a variety of editing patterns by including/excluding editing rules, as was verified through our experiments. We believe this editing model provides a good basis for video editing, although there is still room for further improvement. For a future work, we will systematically collate and evaluate editing rules and patterns that are used in movies or TV programs.

References

[1] Yasutaka ATARASHI, et al: Controlling a Camera with Minimized Camera Motion Changes under the Constraint of a Planned Camera-work Proceedings of International Workshop on Pattern Recognition and Understanding for Visual Information Media 2002, pp.9-14, 2002.

[2] M.Onishi, T.Kagebayashi, K.Fukunaga Production of Video Images by Computer Controlled Cameras and Its Application to TV Conference System Proc. of IEEE Conference on Computer Vision and Pattern Recognition,2, II-131-II-137 (2001).

[3] P. Ekman, W. Friesen The Repertoire of Nonverbal Behavior : Categories, Origins,Usage,and Coding: Semiotica, pp.49–98, vol.1, 1969.

[4] M.Ozeki, Y.Nakamura, Y.Ohta: Camerawork for Intelligent Video Production —-Capturing Desktop Manipulations, Proc. Int. Conf. on Multimedia and Expo, pp.41–44, CD–ROM TA1.5, 2001

[5] M. Ozeki, Y. Nakamura, and Y. Ohta. “Human behavior recognition for an intelligent video production system,” IEEE Proc. Pacific-Rim Conference on Multimedia, pp.1153–1160, 2002.

[6] ISAC, Inc. http://www.isac.co.jp