

# Object Tracking and Task Recognition for Producing Interactive Video Content — Semi-automatic indexing for QUEVICO

Motoyuki Ozeki, Masatsugu Itoh, Hidekatsu Izuno,  
Yuichi Nakamura, and Yuichi Ohta

IEMS, University of Tsukuba, 305-8573, 1-1-1 Ten'noudai Tsukuba, Japan  
ozeki@image.esys.tsukuba.ac.jp

**Abstract.** This paper presents a semi-automatic indexing method for “QUEVICO” — A QA model for video-based interactive media. We first provide an overview of QUEVICO, and then discuss which indices are acquired from the scenario and which must be acquired by manual or automated processing. To obtain these indices, we implemented a prototype system whose processes include human behavior recognition, object tracking, and speech recognition. Through some experiments applying the prototype system to actual indexing of QUEVICO video data, the strong potential of our framework are demonstrated.

## 1 Introduction

The aim of our research is to create video-based interactive media that can provide comprehensible answers to users’ questions. For this purpose, we have proposed a *QUEVICO* framework that realizes intelligent video-based teaching materials[3]. In this framework, video indexing and editing are designed from the viewpoint of “question and answer”, and multi-view video, image, audio, and scenarios can be effectively used as answers.

To enable this function, sufficient indexing of video data is required. Manual indexing, however, is often costly, especially for precise annotation of events or situations in a video, for recording the time that a speech, behavior, or event has occurred, or for recording an object or a human position. To cope this problem, we are developing a video production system that synchronizes a video that has been recorded with its scenario. By finding correspondence between the recorded video and its scenario, we can obtain semantic information from the scenario as well as other types of information such as the position or time from the scene or the video. Both types of information are combined to provide sufficient indices.

In this paper, we first provide an overview of QUEVICO and discuss which indices are acquired from the scenario and which need to be acquired by manual or automated processing. We then describe our behavior-detection and object-tracking method for automatic indexing, and demonstrate its strong potential by preliminary experiments of applying our framework to actual desktop assembly work.

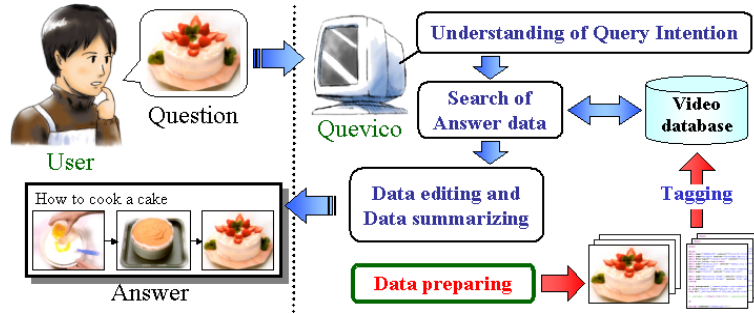


Fig. 1. Outline of QUEVICO.

## 2 Indexing of Assembly Work

### 2.1 Outline of QUEVICO

The goal of QUEVICO is to create interactive teaching manuals for assembly work such as cooking manuals, operation manuals for appliances, teaching manuals for science laboratories, etc. Because such teaching materials include demonstrations that are difficult to explain using only a natural language, the QUEVICO framework designed so that visual or multimodal data plays an essential part. The QUEVICO QA (Question-Answering) model provides a mechanism that answers users' questions by using a video segment, an image, a sentence, or an audio (speech). The QA model deals with approximately 30 kinds of questions, for example, questions regarding method, degree, input, output, quantity, etc. For example, when a user asks QUEVICO "how much sugar do I need to add?" the system answers with a video segment in which a chef explains the quantity of sugar.

Figure 1 provides an outline of the question-answering process that is composed of the following steps: (1) analyzing a question, (2) searching data for potential answers in the database, and (3) displaying summarized/edited data as an answer. For more detail on QUEVICO, please refer to [3]<sup>1</sup>. To enable an effective search to be carried out for potential answers, the QUEVICO model requires that video data are stored with appropriate indices.

In the following sections, we propose a semi-automatic indexing method for QUEVICO data that uses our intelligent video-production system. The "semi-" means that we need preceding tagging to the scenario, and that this process is manually performed at the current stage. Automatic tagging to a scenario will be investigated in the near future.

<sup>1</sup> This is available also at [http://www.image.esys.tsukuba.ac.jp/research/publications/izuno\\_PRICAL2002Aug.pdf](http://www.image.esys.tsukuba.ac.jp/research/publications/izuno_PRICAL2002Aug.pdf).

## 2.2 Semi-automatic indexing by video-scenario alignment

A video for educational or instructional purpose often accompanied by its own scenario. Usually, such a scenario contains a variety of important information: who will do what for what purpose, the name of each object, the name of each manipulation, and so on. In this sense, a scenario specifies the existence of something that should be indexed, and also gives partial information about it, which may be the “task’s name”, “task’s meaning”, “object’s name”, “object’s role”, “material”, etc. It is, however, difficult to obtain from a scenario other types of information such as object positions in an image or event-occurrence times in a video, etc. Because manual tagging of such information would be extremely labor-intensive, we need to extract this type of information by automatic or semi-automatic processing.

One promising approach to coping with this problem is matching between a pre-indexed scenario and a video by image and speech recognition. In our framework, a scenario is manually indexed using QUEVICO tags, a process that will be semi-automated in the near future by using natural language processing. The tagged scenario is then aligned to the video by speech recognition and human behavior recognition.

We basically expect that word matching between the speech and the scenario will provide alignment between the video and the scenario. A speaker, however, does not exactly speak his/her part lines in a scenario, and speech recognition often has serious errors. To cope with this problem, we use ambiguous matching based on phoneme<sup>2</sup>. This concept will be introduced in section 3.

Behavior recognition of a speaker and object detection also provide good clues regarding alignment. For this purpose, we use behavior-recognition results together with speech-recognition results, since human movements tend to be continuous and segmentation of these movements is often difficult. Once given the above correspondence, object positions, human motions, and events occurrence times detected by our system, they become useful indices or attribute values.

## 2.3 Video production system

We consider here the presentation of assembly work on a desktop:

- One person explains the assembly work; hereafter we refer to this person as the *speaker*, and zero to several assistants appear in the scene.
- We assume that an audience is sitting or standing in front of the speaker, and the speaker explains the assembly work by manipulating parts in front of him/her.
- All manipulations are performed in a fixed space, *e.g.* above a desk.

---

<sup>2</sup> More precisely, we use KANA in Japanese as the primitives for matching. KANAs can be regarded as an alternative to phonemes because they can not be further divided.

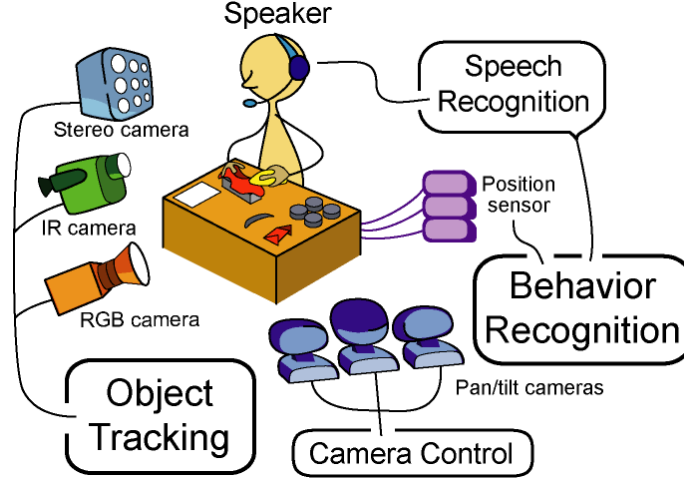


Fig. 2. Intelligent video production system.

We have developed an *intelligent video production system*, as shown in Fig. 2, which automates video capture, editing, and indexing. Multi-view videos are taken by multiple cameras that track the head, hands, and other important places with the cooperation of a magnetic position sensor. Speech data are obtained by speech-recognition software<sup>3</sup>. The motion data, which includes the positions of both hands and the waist, are recorded, and typical behaviors are detected by using the motion and speech data. Any object held by the speaker's hand is tracked, and its position is recorded. All the video content and data obtained by our system are synchronously stored with the index of UNIX time.

### 3 Scenario-Speech Matching

The alignment between a scenario and a video is roughly obtained by sentence-matching between the speech lines in a scenario and the sentences from a speech-recognition result. The problem of finding pattern  $R$  from string  $T$ .  $R$  and  $T$  is denoted as follows:

$$T = a_1 a_2 a_3 \cdots a_I, \quad R = b_1 b_2 b_3 \cdots b_J$$

where  $a_i$  and  $b_j$  represent KANA characters that compose  $T$  and  $R$ , respectively and KANA is a phoneme or a pair of phonemes that composes Japanese words.  $i$  and  $j$  represent the character position from the beginning of  $T$  and  $R$ , respectively.

In actual experiments, we consider  $T$  to be the whole of the transcript that is obtained by speech recognition, and  $R$  as a sentence in the scenario. The values

<sup>3</sup> IBM ViaVoice.

```

1.  $g(0,0) = 0, B(0,0) = 0$ 
2.  $for(i = 1, 2, \dots, I)$ 
    $g(i,0) = 0, B(i,0) = i$ 
3.  $for(j = 1, 2, \dots, J)$ 
    $g(0,j) = g(0,j-1) + 2, B(0,j) = 0$ 
4.  $for(i = 1, 2, \dots, I)$ 
    $for(j = 1, 2, \dots, J)$ 
      $g(i,j) = \min \begin{cases} g(i-1,j) + 1 & (a) \\ g(i-1,j-1) + 2 & (b) \\ g(i,j-1) + 2 & (c) \end{cases}$ 
      $\begin{cases} \text{In the case of (a)} & B(i,j) = B(i-1,j) \\ \text{In the case of (b)} & B(i,j) = B(i-1,j-1) \\ \text{In the case of (c)} & B(i,j) = B(i,j-1) \end{cases}$ 
5. Search the value  $i_{match}$  which minimalizes  $g(i,J)$ 

```

**Fig. 3.** The algorithm for string DP matching.

of array  $g(i,j)$  mean the distance between  $a_{B(i,j)} \dots a_i$ , which is a substring of  $T$ , and pattern  $R$ . The actual matching process is given in Fig. 3. Array  $g(i,j)$  and  $B(i,j)$  are initialized in steps 1 through 3, and are evaluated by the loop in step 4.  $g(i,j)$  represents the similarity defined above, and  $B(i,j)$  represents a matching starting position of  $R$  when  $a_1 \dots a_i$  most likely to matches  $b_{B(i,j)} \dots b_j$ . The algorithm searches the value of  $i_{match}$ , which minimizes the value of  $g(i,J)$  in step 5. Thus,  $i_{match}$  represents the position of the match, and substring  $a_{B(i_{match},J)} \dots a_{i_{match}}$  represents the portion of string  $T$  that matches with string  $R$ .

## 4 Behavior Recognition

Our system detects the following behaviors:

**Pointing/Holding-out:** A speaker's hand(s) is stretched away from his/her body, and he/she speaks one of keywords for pointing/holding-out behaviors such as "this is".

**Manipulation/Illustration:** A speaker's hands are on/above a desk, and he/she speaks one of the keywords for manipulation/illustration behaviors such as "in this way".

We can expect these behaviors to be well described in a scenario, since they play an important role in giving demonstrations. By making a correspondence between these descriptions and behavior-detection results, we can obtain good indices, each of which has semantic attributes such as a task name or purpose, as well as its occurrence time and position.

Figure 4 shows the outline of the motion detection. By using both motion and speech clues, the behaviors are detected from the speaker's continuous motions. A motion clue of pointing/holding-out behavior is detected when a speaker's

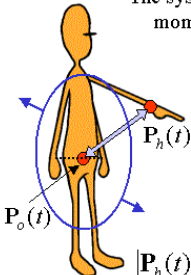
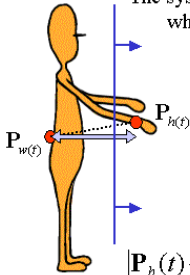
Pointing/Holding-out		Manipulation/Illustration	
Motion clues			
 <p>The system simply detects moments when the arm stretches beyond the threshold</p> $ P_h(t) - P_o(t)  > Th_{P/H}$		 <p>The system simply detects whether the hands are on/above the desk</p> $ P_h(t) - P_w(t)  > Th_{M/I}$	
Speech clues			
English	Japanese	English	Japanese
this is	[KOREGA]	in this way	[KONNOYOUNI]
this (+ object)	[KONO (+object)]	like this	[KONNAFUUNI]

Fig. 4. Outline of behavior detection.

arm is stretched beyond the threshold. When both hands are stretched and are close to each other, the system regards the movement as a motion clue for pointing/holding-out behavior with both hands. If both hands are held apart from each other, the system regards the movement as a motion clue for pointing/holding-out behavior with a single hand whose position is higher than the other's.

Because a manipulation/illustration movement is originally a simulation or a demonstration of movements or shapes, it has no fixed pattern. To deal with the motion clue of this behavior, we are currently using hand position, identifying whether the hands are on/above a desk.

If the system detects both speech clues and motion clues within a certain period, the system accepts them as corresponding behaviors. For more detail regarding the behavior-detection method, please refer to [4].

## 5 Object Tracking

Information regarding object position is important and can be used for video segmentation, for preparation for a clickable icon, or for image cropping to emphasize an important portion. Because we cannot expect ideal automatic processing that detects the position of every object in a scene at any time, we have developed a system that detects and tracks an object while it is held by a speaker. Because an important objects are often held, moved, or manipulated, object detection provides important information even with the above limitation.

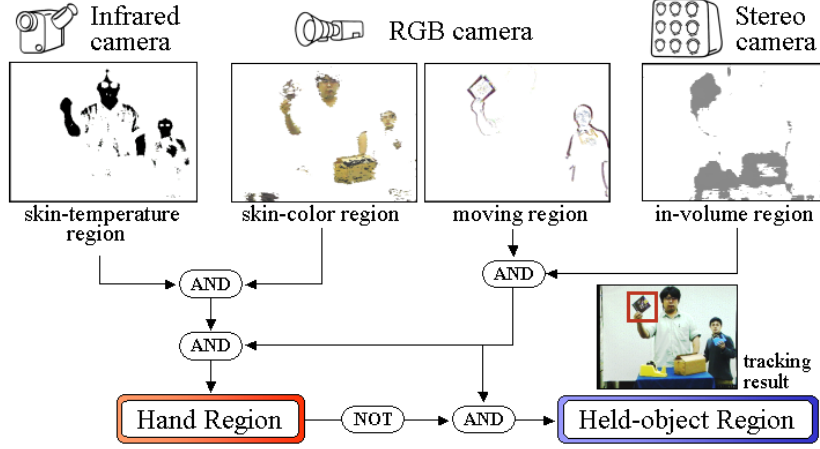


Fig. 5. Outline of object tracking.

Figure 5 shows an outline of the detection and tracking process. Our system uses three image sensors, an ordinary RGB camera, an IR camera, and a stereo camera, and it tracks a hand-held object at video-rate. First, the system detects *skin-color regions*, *moving regions*, *skin-temperature regions*, and *in-volume regions*<sup>4</sup> based on information provided by the three sensors. Then, by integrating the above regions, the *hand regions* and *held-object regions* are detected based on the following principle:

$$\begin{aligned}
 \text{hand region} &= \text{in-volume region} \wedge \text{moving region} \\
 &\quad \wedge \text{skin-temperature region} \wedge \text{skin-color region} \\
 \text{held-object region} &= \text{in-volume region} \wedge \text{moving region} \wedge \neg \text{hand region}
 \end{aligned}$$

By the above method, our system can track an object without prior knowledge of the object, even if there are other moving people or objects with skin-color tones in the background. The number and position of hand-held objects appearing in the scene and its change are recorded while the object(s) are held by a speaker. For more detail regarding the object-tracking method, please refer to [5].

## 6 Information Integration to Indices

We here show how the information from a scenario and information from the recognition system are integrated, and how the video data are indexed. Figure 6 shows a tagged scenario, whose contents explain the assembly of a toy car. QUEVICO has “*task*” tag and an “*object*” tag to provide annotation regarding

<sup>4</sup> The in-volume region is the volume in which hands and related objects appear.

```

<quevico>
<scene>
  <task id="t1" name="assemble" output="#o1" method="#v7"
    instrument="#o2">
    <object id="o1" name="toy car"/>
    <object id="o2" name="power tool" description="#a3"
      state="#v8 #v13"/>
    <task id="t2" name="attach" patient="#o3" instrument="#o2"
      input="#o4" method="#v2">
      <object id="o3" name="chassis" state="#v9 #v10"/>
      <object id="o4" name="body cover" state="#v10"/>
    </task>
  </scene>
</stream>
<stream name="video(action)" src=" ">
  <vsegment id="v1" begin=" " end=" "/>
  <point id="p1" time=" " x=" " y=" " object=" "/>
  <point id="p2" time=" " x=" " y=" " object=" "/>
  </vsegment>
  <vsegment id="v2" begin=" " end=" "/>
</stream>
<stream name="video(patient)" src=" ">
  </stream>
<stream name="audio(speech)" src=" ">
  <asegment id="a1" begin=" " end=" ">
    I will now start to explain how to assemble a toy car.
  </asegment>
  <asegment id="a2" begin=" " end=" ">
    For a start, we use this power tool to assemble a toy car.
  </asegment>
</stream>
</quevico>

```

**Fig. 6.** An example of a tagged scenario.

tasks and objects. The attributes of those tags include a video segment and an audio segment as well as their id and names. The position information is given as attribute values of a *position* tag, *e.g.*, its time, x-coordinate value, and y-coordinate value.

The attribute values of each *vsegment* tag, *asegment* tag, and *position* tag are not filled with real values, while attribute values of each *task* tag are given such as “attaching” and also are the values for each *object* tag. The blanks (null strings) as attribute values for each *asegment* tag are given by using the DP matching mentioned in section 3. If the system detects a speaker’s pointing/holding-out behavior or a manipulation/illustration behavior, the “begin” and “end” attribute values are given to a *vsegment* tag. If the system detects an object, the “time” attribute value, the “x” and “y” attribute values are given, which indicate x-coordinate and y-coordinate values for the object’s position, respectively.

Figure 7 shows an outline of information integration for semi-automatic tagging regarding a *vsegment* tag and a *point* tag. These types of integration are based on the time stamps that are shared by multi-view videos, audio (speech), behavior-detection results, and object-tracking results.



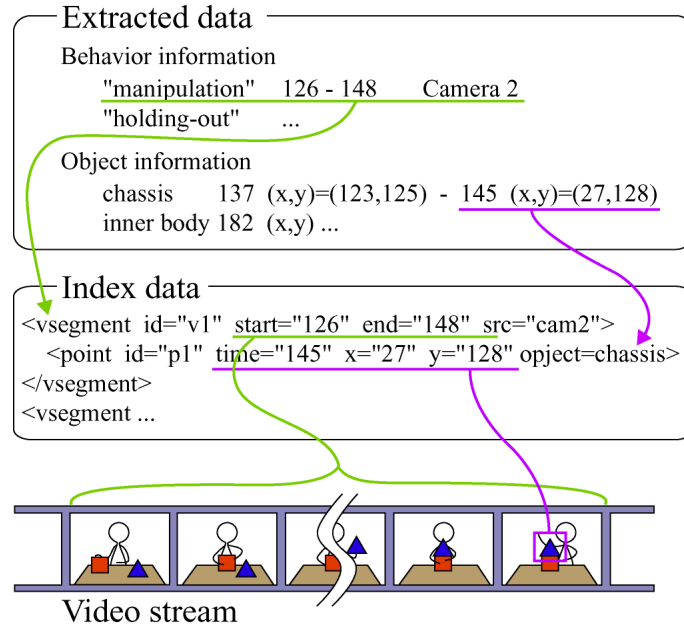


Fig. 7. Example of automatic indexing.

## 7 Example of Semi-Automatic Indexing

We applied our semi-automatic indexing method to the actual assembly work mentioned in the previous section. The work “How to make a toy car” takes 6 minutes.

In this case, the system detected all behaviors of the above types without any mis-detections. In approximately 90% of cases, the system successfully detected an object when a speaker picked the object up or put it on a desk. The speech data obtained from the speech recognition had an approximately 10% errors rate, and included some words that were not in the scenario such as “well ...”.

We gave the video and its indices to our QA system and carried out a preliminary experiment for Question-Answering, as shown in Fig. 8. In the followings, we show some examples in which our QA system gave appropriate answers:

**What kind of tools do I need to assemble the toy car?** The system displays a video frame of the patient view, which contains the “power tool”.

**What is the power tool?** The system displays the sentence in the scenario that explains how to use the power tool.

**How can I attach tires to a chassis?** The system displays the diagram as shown on the left in Fig. 8, which explains how to attach front and rear tires to the chassis.

**How can I attach the inner body to the chassis?** The system displays the video segment of the action view as shown on the right in Fig. 8.



Fig. 8. Example of Question Answering by QUEVICO.

## 8 Conclusion

This paper has introduced the overview of the semi-automatic indexing method for QUEVICO, which is the framework for composing interactive video content. The method is based on manual tagging of a scenario, scenario-video alignment by speech recognition and behavior detection, and indexing of object position. We have provided some examples by taking a video of actual assembly work and indexing the video. Throughout the experiment, the strongly potential of our framework was demonstrated.

Although the preliminary results are satisfactory, our system is still under development. In this sense, we have many things to tackle, for example, improving in the accuracy of object tracking, semi-automated scenario indexing, and so on.

## References

1. Wactlar, H., Kanade, T., Smith, M., Stevens, S.: Intelligent Access to Digital Video: The Informedia Project IEEE Computer (1996) Vol.29 No.5
2. Jiang, H., Elmagarmid, A.: WVTDB - A Semantic Content-Based Video Database System on the World Wide Web IEEE Trans. on KDE (1998) vol.10 NO.6
3. Izuno, H., Nakamura, Y., Ohta, Y.: Quevico: A framework for video-based interactive media. Proc. Int'l Workshop on Intelligent Media Technology for Communicative Reality (2002) 6–11
4. Ozeki, M., Nakamura, Y., Ohta, Y.: Human behavior recognition for an intelligent video production system. Proc. PCM (2002) 1153–1160
5. Ozeki, M., Itoh, M., Nakamura, Y., Ohta, Y.: Tracking hands and objects for an intelligent video production system. Proc. ICPR (2002) 1011–1014