# Acquiring hand-action models by attention point analysis

Koichi Ogawara Soshi Iba [†] Tomikazu Tanuki [††] Hiroshi Kimura [†††] Katsushi Ikeuchi

Institute of Industrial Science, Univ. of Tokyo, Tokyo, 106-8558, JAPAN
[†)] The Robotics Institute, Carnegie Mellon University, Pittsburgh PA, USA
[††)] Research Division, Komatsu Ltd. Kanagawa, 254-8567, JAPAN
[†††)] Univ. of Electro-Communications, Tokyo, 182-8585, JAPAN

{ogawara, ki}@iis.u-tokyo.ac.jp, iba+@cmu.edu
tomikazu_tanuki@komatsu.co.jp, hiroshi@kimura.is.uec.ac.jp

## Abstract

*This paper describes our current research on learning task level representations by a robot through observation of human demonstrations. We focus on human hand actions and represent such hand actions in symbolic task models. We propose a framework of such models by efficiently integrating multiple observations based on attention points; we then evaluate the produced model by using a human-form robot.*

*We propose a two-step observation mechanism. At the first step, the system roughly observes the entire sequence of the human demonstration, builds a rough task model and also extracts attention points (APs). The attention points indicate the time and the position in the observation sequence that requires further detailed analysis. At the second step, the system closely examines the sequence around the APs, and obtains attribute values for the task model, such as what to grasp, which hand to be used, or what is the precise trajectory of the manipulated object.*

*We have implemented this system on a human form robot and demonstrated its effectiveness.*

## 1 Introduction

One of the most important issues in robotics is how to program robot behaviors. Several methodologies for programming robots have been proposed. We can classify them into the following three categories: static textual programming, manipulation by a human through a control device, and automatic programming. The former two methods require human intervention throughout the entire task. In contrast, automatic programming is intended to reduce human aid and to generate an entire robot program automatically. Given the necessary initial knowledge, robots try to acquire their behavior automatically from observation, simulation or learning.

Our research goal is automatic acquisition of robot behavior, in particular, hand-actions, from observation based on the automatic programming approach. We divide the acquisition process of human tasks into two levels: task level, e.g, what-to-do and behavior level, e.g., how-to-do it. This paper covers the former one, task level acquisition, while the latter one is presented in [4]. In Chapter 2, we discuss the necessity of integration of multiple observations. In Chapter 3, we introduce the concept of attention points and present a method for constructing a task model by two kinds of attention point (AP) analyses. In Chapters 4 and 5, we describe implementation details for each attention point analysis. In Chapter 6 we present experimental results. Chapter 7 contains our conclusions and remarks on future work.

## 2 Acquisition of human task

Ikeuchi, Suehiro and Kuniyoshi et al. studied vision based task acquisition [1, 2]. In their research, the acquisition system observed a human performing an assembly task and constructed high-level task models. Then, using those constructed models, a robot performed the same task. Kimura et al. proposed task models which could be used to realize cooperation between a human and a robot [3]. In this scheme, the robot first observes sequential human operations, referred to as events, by vision and analyzes mutual event dependencies (pair of pre-conditions and results) in the tasks. The robot is able to change its assistant behavior according to the current event observed and the knowledge of what is to be done next, derived from the task model, and to generate a large number of cooperative patterns from a single task model. However, these models depend on one (typically a single camera) or a few sensors and are constructed through one-time observation, therefore they are not suitable for close analysis.

Our approach utilizes multiple observations which vary in sensor variety and granularity for efficient analysis. By analyzing each observation sequentially or repeatedly, we can determine the necessary part in the human demonstration where the level of detail in the subsequent analysis should be changed and can then accumulate each result to build the task model efficiently. Integration of observations enables us to build heterogeneous task models in which accuracy is enhanced locally. We introduce the concept of attention point (AP) as a key of integration and propose a two-step analysis based on APs as a method of constructing a human task model.

# 3 Attention point

## 3.1 Two-step analysis

Integration of multiple observations is accomplished by two-step analysis. At the first step, the system roughly analyzes the input modalities and recognizes the outline of the entire human demonstration (rough task model). At the same time, the system also extracts APs. APs, which require close observation to learn a particular behavior, are defined around specific time and position along a sequence of a human demonstration.

At the second step, the system closely examines the demonstration around each AP to enhance the task model. This sequence can be the same observation data or another one. In the latter case, the system synchronizes two observation data which are derived from different demonstrations of the same task.

We employ a type of task models similar to Ikeuchi's. We decomposed a hand-action task as a sequence of discrete hand-actions, during which a human performs some action by manipulating objects, and we symbolized possible hand-actions as "Action Symbols", which indicate what-to-do information. The task model also includes several attributes for "Action Symbol", detailed information to achieve that "action," such as which hand to use or which object to grasp (Table 1). In the proposed two-step approach, this "Action Symbol" is obtained from the rough analysis at the first step. Then, from the detailed analysis around the APs previously determined, those attributes are obtained at the second step.

Table 1: Task Model

| Attributes | Priority | Value |
|---|---|---|
| Action Symbol | 3(high) | Power Grasp, Precision Grasp Release, Pour, Hand Over |
| Object Model | 3 | Shape and Color histogram |
| Hand | 2 | Right, Left, Both |
| Position | 1 | Absolute Position in 3D space |
| Time Stamp | 1(low) | Absolute Time (start and stop time) |

We propose two different kinds of AP analyses in the following sections.

## 3.2 Integration of sensors separated in space

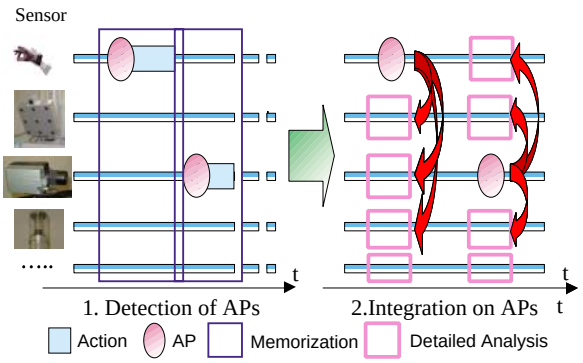


Fig. 1: Two Steps Analysis using Attention Point

When several input sensors are available simultaneously, it is generally ineffective to precisely analyze all the data along the entire human demonstration. So the system temporally records all the raw data available and employs a two-step analysis of a human task (Fig.1).

To realize the two-step analysis, we utilized the short-term memorization method. At one observation sequence, the system first analyzes the input data given by the set of modalities that require the cheapest computation. It extracts "action symbol," and APs as the boundaries of each segmented action while recording all the data around each AP on storage devices.

After an observation sequence completed, i.e., after one demonstration was finished, the system acquires the recorded data corresponding to each AP from the storage devices and applies a detailed analysis on them off-line. This process obtains the remaining attributes in the task model.

## 3.3 Integration of sensors separated in time

The method described above requires temporal sets of recorded input data; as the number of sensors and work time increases, the amount of unused data expands. And also, for some sensors, it is not advisable to adopt a specific sensor configuration at all times because of range, speed, precision trade-off.

So we propose another two-step analysis in which the system requires quantitative evaluations of a number of demonstrations for the same task. The system roughly analyzes the demonstration and extracts APs at the first observation. Then the system changes the sensor configuration if necessary and examines the second demonstration around the APs to enhance the task model. For the synchronization issues, the system can predict the hand motion from

the first observation and, by watching for the appearance of the predicted motion at each AP in the second observation, the multiple observations can be synchronized.

## 4 AP analysis for sensors separated in space

Our system employs a pair of data gloves and a 9-eye real-time stereo vision system. We can acquire depth and color images from the stereo vision system and can acquire hand motion (finger shape, absolute position and orientation) from the data gloves. The image processing is much more time-consuming as opposed to the processing of the data gloves; thus we adopted the AP analysis described in Section 3.2. We utilized the data gloves to extract APs and "action symbols;." then, to determine attributes of the task model, the system analyzes depth and color images around those Aps.

Fig.2 and Fig.3 show the flow of the AP based two-step analysis. The subsequent sections describe the outline of the analysis. Please refer [5] for details.

### 4.1 Rough analysis by gesture spotting

We set up a task domain for a specific hand-action task and built a finite set of "action symbols," which represents all the possible hand motions that appeared in that task domain. "Action symbols" are combinations of finger actions and local hand motions. For now, we classify possible finger actions into three actions: "Power Grasp", "Precision Grasp"[6] and "Release," and described human hand actions as a finite set of "Action Symbols" which are combinations of above finger actions and local hand motion. By excluding hand actions composed of independent finger motion, we can segment the entire hand-action task into meaningful "Action Symbols".

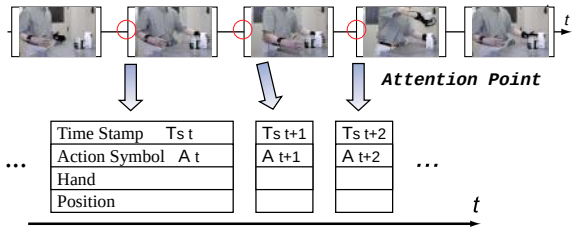


Fig. 2: Rough Analysis by Gesture Spotting

To obtain "Action Symbols" in the task model, we aim to spot human gestures from hand-actions performed by a human demonstrator as shown in Fig.2. In this experiment, we chose "transferring content of container" as a task domain and selected five gestures as possible hand actions (Table2). APs are defined as the starting point of each gesture. To extract these "Action Symbols," we employ datagloves and a gesture spotting technique based on Hidden Markov Models (HMMs).

Table 2: Gesture definitions

| Gesture | Primitives | Action |
|---|---|---|
| Grip | cls+sp | Power-grasp from open position |
| Pick | prc+sp | Precision-grasp from open position |
| Pour | cls+roll+sp | Power-grasp, and roll the wrist |
| Hand-over | prc+forw+sp | Precision-grasp, move forward, and back |
| Release | opn+sp | Open a grasp hand |
| Garbage | gb | A filler model for spotting |
| Start,End | sil | Silence at the start and end |

We utilize a pair of data gloves (CyberGlove 18-DOF each), and 6-DOF position sensors (Polhemus) as input devices for the HMM-based gesture spotting module. So, 24 dimensions and their differentials are the input to the HMM module for each hand. The second column of Table2 indicates the defined HMM primitives for each gesture. Each primitive is defined as 5-state left-right HMMs.*sil* is a silent state used at the time of training, *sp* is a short pause which tends to occur at the end of the gesture corresponding to an action symbol, and *gb* is a garbage collector trained on arbitrary non-gesture movement. By sharing primitives, each action symbol requires a small number of training data with better efficiency.

Left and right single-hand gestures are spotted separately in a parallel manner, while two-handed gestures are spotted by combining results from the analysis of both hands. Our system can sample the data from a pair of data gloves in 30Hz and can spot gestures corresponding to action symbols in parallel without delay.

### 4.2 Attention point analysis by vision

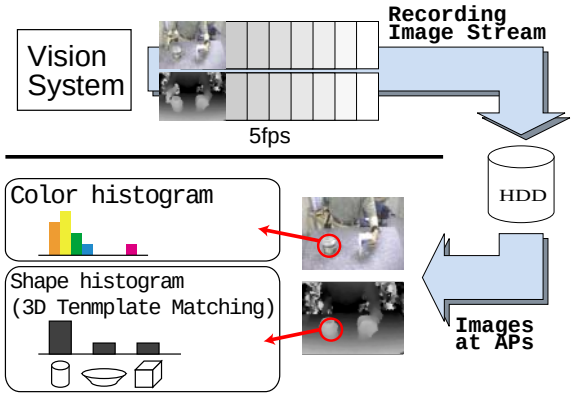


Fig. 3: AP analysis by vision

Computation time for image processing is rather time-consuming. So we first record all the raw data from the vision system around APs. These recorded data are syn-

chronized with the data-glove analysis and the correspondence between them is easily made. After the first analysis is finished and extracts APs, the system fetches the corresponding images and extracts the information about the manipulated objects (Fig. 3).

By analyzing just before each AP, we can obtain the images in which the target object is not occluded by the hand. The object is modeled by calculating shape and color histograms.

We assume that a human task is demonstrated on a table whose geometric information is known. By extracting depth regions corresponding to each object on the table, we calculate shape histogram as a list of goodness of matching between an object extracted in the depth image and each object model in the database. This goodness of matching is obtained by using the 3D Template Matching(3DTM)[7] technique.

3D Template Matching, a technique for localization, finds the precise position and orientation of the target object in depth data. This process is calculated by projecting the corresponding 3D model into the 3D space generated from a depth image and calculates goodness of matching between the 3D model and the 3D data by summing up weighted distance between each center point of the meshes in the template model and the closest 3D point. 3DTM adopts M robust estimator to eliminate the effect of outliers.

Color histogram is calculated as a normalized hue histogram which counts pixels with large saturation value among the area of the object on the color image. These depth and color images are produced at 5 fps (up to 30 fps) synchronously by the 9-eye multi-baseline stereo vision system.

The system registers this histogram information in the attribute slot of the task model.

# 5 AP analysis for sensors separated in time

In the previous chapter, we described the hand-action model in terms of classified gestures. This model gives a good notion of hand motion and the type of the manipulated objects, but tells nothing about the manipulated object's motion.

In oder to model the delicate motion of the manipulated object or to judge the success/failure of the task performed by the robot automatically, a task model must contain some information about precise position and orientation of the manipulated object at particular parts in the entire task.

We developed an efficient method to acquire the precise trajectory based on repeated observations and APs. In this chapter, we present the method, which uses the zoom stereo vision system.

## 5.1 Repeated observation

To acquire the precise trajectory of the object, we combined two kinds of two-step analyses as shown in Table3.

Table 3: Process of repeated observation

| | Zoom | Model | Time | description |
|---|---|---|---|---|
| 1 | x1 | Coarse | 0.6s | extraction of APs |
| 2 | x2 | Coarse | 1.4s | tracking the object in real-time |
| 3 | x3 | Fine | 2.0s | tracking the object in off-line |

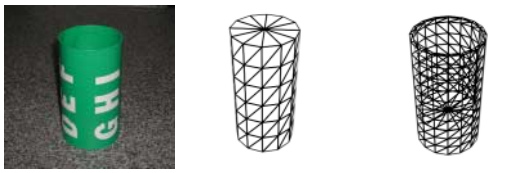


Fig. 4: 3D Model

Process 1→ 2 adopts the method described in Section 3.2 (repeated observation), while Process 2 → 3 adopts the method described in Section 3.1.

Fig.4 shows the object and its CAD model used in this experiment.

### 5.1.1 Extraction of APs

At first, the zoom configuration is set to x1(default) and the system roughly tracks the object for each hand action using the 3DTM method. To estimate the initial position and time of the object to be tracked, we utilized data-gloves used in the previous chapter; the gloves were enhanced by tactile sensors to classify the grasping. The system can detect the grasping motion directory from tactile sensors, so it roughly estimates the initial object position (from the polhemus sensor) at the time of grasping.

We used the coarse object model during tracking, because precise position and orientation is not important. The system gets the rough trajectory and also gets the APs as the initial position and time of the tracking.

Fig.5 shows the tracking result (intensity images overlaid with the wire-frame model).

### 5.1.2 Tracking in repeated observation

At this stage, the system demands the repeated demonstration of the same task. In this stage, the doubly zoomed cameras cannot put the entire action in sight in a fixed orientation; therefore, driving of the pan/tilt moving mechanism synchronized with image processing in real-time is necessary to track the target object to be kept in the center of the view.

All the depth and intensity images are recorded during tracking. These images are used in the third process below.

Fig. 5: Tracking at the first stage

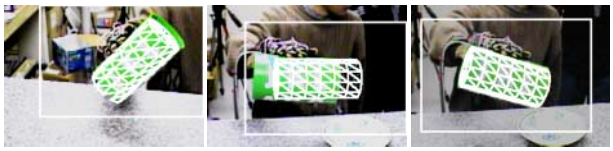

Fig. 6: Tracking at the second stage

This tracking is also processed by 3DTM with the coarse object model, because precise localization is not important.

Fig.6 shows the tracking result (intensity images overlaid with the wire-frame model).

#### 5.1.3 Estimation of the precise trajectory

At the third stage, the system fetches the recorded images and localizes the object in each scene to estimate the precise trajectory with the fine model. This image fetching is the same technique as that described in chapter 4.

This process is executed off-line. To localize the object precisely, we developed a method to combine the 3DTM and 2DTM. 3DTM is a method for localizing the 3D model in the 3D points obtained from the depth data[7]. 2DTM is the edge-based localization method between the 3D model and the estimated 3D edges of the contour of the object, which are derived from the intensity image[7].

2DTM is sensitive to the edges in the image background and does not offer a good guess about z position (parallel to the viewing direction) of the model because of the approximation of z position of the 3D edge. But, at the final stage of the localization, 2DTM offers a good guess about the position and orientation perpendicular to the viewing direction.

So, we first adopt the 3DTM only to localize the object to the approximate position and then we adopt 2DTM & 3DTM combined method to localize the object to the exact position as shown in Table4.

2DTM and 3DTM are calculated in the same 3D space by M-estimator (Lorentzian) with different weight. Sigma is the parameter to reduce the effect of the outliers.

Fig.7 shows the tracking result. The upper row shows the intensity images and the lower row shows the disparity images. The contour of the object's model is overlaid in each image.

Fig. 7: Tracking at the third stage

Table 4: 2DTM & 3DTM combined localization

| Method | Sigma[mm] |
|---|---|
| 3DTM | 10.0 |
| 3DTM | 4.0 |
| 3DTM | 2.0 |
| 3DTM & 2DTM | 2.0 |
| 2DTM | 1.0 |

### 5.2 Experimental result

Our stereo vision consists of zoom lens cameras. This is actually digital zooming but, when capturing images, the stereo system re-samples each pixel at the ratio of one-quad, so we can expect that to doubling the power of zooming value will not reduce the quality of the image captured by our stereo system.

Stereo processing is done on a hardware chip and the system can acquire a depth image and the corresponding intensity image ($280 \times 200$) in 15fps at most.

The average tracking rate of the first stage is 0.6 [sec/frame] on our Pentium3 500MHz PC. Similarly, the tracking rates of the second stage and third stage are 1.4 [sec/frame] and 2.0 [sec/frame], respectively.

The difference in the rate between the first and the second stages is mainly due to the construction time of the Kd-tree used in localization. We restricted the search area of 3DTM to be very close to the object, the inside of the rectangle shown in Fig.5 and Fig.6, so the search area of the first stage is relatively smaller and the construction time is short.

The difference in frame-rate between the second and the third stages is due to the difference in the number of iterations in the localization process and the level of detail of the model.

## 6 Performance by robot

We have developed a human-form robot as an experimental platform for learning and performing human hand-

action tasks[9]. The robot has similar capabilities and body parts to those of humans, including vision, dual arms and upper torso.

When the robot is to perform the same task after constructing a task model, it searches for objects on the table and, for each object, it calculates mean square distance between the shape and color histogram of the object on the table and those in the model database. The smallest value determines the best matching objects. In this way, the robot recognizes the object. Once the recognition of the current environment is done, the robot sequentially executes the action corresponding to each "Action Symbol" in the task model adapting to the current environment condition.

Fig.8 shows the experimental result in which the robot performed the same task successfully.


Fig. 8: Experiment

## 7 Conclusion

We proposed a novel method of constructing a human task model by attention point (AP) analysis. Attention points relate and integrate multiple observations and construct a locally enhanced task model of human demonstration. AP analysis consists of two steps. In the first step, action segment and APs are extracted. Then, at the second step, by closely examining human demonstration only around APs, the system extracts the attribute values and improves the model.

By reducing unnecessary analysis, the system can construct the task model efficiently. Efficiency is important when we consider human-robot cooperation tasks in which the robot must respond to the action taken by both a human and the robot itself in relatively short time.

We presented two kinds of AP analyses, one for integration of sensors available simultaneously and the other for integration of sensors derived from different observations of the same task by repeated demonstration.

To realize the first AP analysis, we proposed a short-term memorization method, which records all the raw input data around each AP to be processed at the second step. And also, we proposed a localization method which combines 2DTM and 3DTM to track and localize a moving object robustly.

The future work is to solve the problem of integrating the trajectory information into the current task model for training the robot itself automatically. We are also planning to combine this task level acquisition with the behavior level acquisition method [4]. First, the task level acquisition constructs task models to perform the entire task to be adapted to the environment. It also extracts special APs that require behavior level acquisition. Second, the behavior level acquisition analyzes those APs closely and obtains a suitable motion sequence (sub-skill). This two-layer approach should extend the capabilities of the learning robot that can acquire a human task through observation.

## Acknowledgment

This work is supported, in part, by Japan Society for the Promotion of Science (JSPS) under the grant RFTF 96P00501, and, in part, by Japan Science and Technology Corporation (JST) under Ikeuchi CREST project.

## References

[1] K. Ikeuchi and T. Suehiro: "Toward an Assembly Plan from Observation Part I: Task Recognition With Polyhedral Objects," *IEEE Trans. Robotics and Automation*, 10(3):368–384, 1994.

[2] Y. Kuniyoshi, M. Inaba, and H. Inoue: "Learning by watching," *IEEE Trans. Robotics and Automation*, 10(6):799–822, 1994.

[3] H. Kimura, T. Horiuchi and K. Ikeuchi: "Task-Model Based Human Robot Cooperation Using Vision," *IROS '99*, 2:701–706, 1999.

[4] J. Takamatsu, H. Tominaga, K. Ogawara, H. Kimura and K. Ikeuchi: "Symbolic Representation of Trajectories for Skill Generation," *IEEE ICRA*, 4:4077-4082, 2000.

[5] K. Ogawara, S. Iba, T. Tanuki, H. Kimura, K. Ikeuchi: "Recognition of Human Task by Attention Point Analysis," *IEEE/RSJ IROS*, 3:2121-2126, 2000.

[6] M. R. Cutkosky, "On Grasp Choice, Grasp Models, and the Design of Hands for Manufacturing Tasks," *IEEE Trans. on Robotics and Automation*, 5(3):269–279, 1989.

[7] M. D. Wheeler: "Automatic Modeling and Localization for Object Recognition", *Ph.D Thesis*, CMU, 1996.

[8] K. M. Knill and S. J. Young: "Speaker Dependent Keyword Spotting for Accessing Stored Speech," *Cambridge University Engineering Dept., Tech. Report*, No. CUED/F-INFENT/TR 193, 1994.

[9] K. Ogawara and J. Takamatsu and S. Iba and T. Tanuki and H. Kimura and K. Ikeuchi: "Acquiring hand-action models in task and behavior levels by a learning robot through observing human demonstrations," *IEEE Conf. on Humaniod Robots*, 2000.