

# Maintenance and Drift of Attention in Human-Robot Communication

Jun Mukai<sup>1</sup> and Michita Imai<sup>1</sup>

Department of Science and Technology, Keio University  
3-14-1 Hiyoshi Yokohama, Kanagawa, Japan 223-8522

**Abstract.** We describe an attention based human-robot communication system called ACS. Here, an attention refers to an arbitrary policy for selecting behaviors. Attention is usually defined a priori by the designers of the robots, which prevents communication between humans and robots. The reason is that the reactions of such robots are fixed for specific situations so that humans are easy to predict the robots reactions. We therefore developed ACS to enable robots to generate their own attentions without predefined settings. We propose Feature Drift, which enables the system to dynamically maintains its attention based according to environmental objects. In particular, Feature Drift can change the attention spontaneously in over time, which solves the problem of fixed reaction. We implemented ACS in a communication robot, Robovie, and evaluated it. The results showed that the robot could maintain its own attention and react to human utterances according to this attention.

## 1 Introduction

In order for robots to become part of everyday life, they must be able to communicate so that they can develop “relationships” with humans [1]. However, there are some problems in human-robot communication. One of these problems is that each behavior of a robot is fixed according to specific human utterances during conversations. This means that the robot behaves in the same way in a given situation. The predictability of the robot’s reactions prevents the development of a relationship between humans and robots.

There have been numerous studies on developing robots which communicate with humans [2, 3]. The typical approach used is as follows. First, basic communication behaviors such as “greeting” and rules for these behaviors are designed. Robots then execute these behaviors according to the rules. Other robots such as Kismet [4] or QRIO[5] are based on behavior-based robotics [6, 7]. Behavior-based communication robots behaves according to their own model. These models are typically based on emotional and/or physical states.

However, these types of robots are also affected by the problem of fixed reactions. The reason for rule-based robots is that each reaction is completely defined for each situation according to the appropriate rule. The rationale for behavior-based robots is that the reaction is determined by its model. In this case, each reaction is also fixed to each situation and inner state.

Our approach is to enable the robots to spontaneously generate their own policies for reactions without predefined status. We call this policy *attention* of robot. Although some robots such like Kismet has attention systems, such attention systems and ours differ because the attention of Kismet is determined from current state of interaction. In our approach, robots need to be able to: 1) acquire their attentions spontaneously, not defined a priori; 2) modify their attentions dynamically.

The first requirement means that robots acquire their own attention with their own reason, not predefined rule. Kismet cannot achieve this requirement. This requirement also means that their attention should be acquired from their own, not from human. The second requirement that the acquired attention should change over time. Our purpose is not only variety of each robot's reaction but also variety over time. The reaction should change over time.

In this paper, we describe the development of a system called ACS (Attention-based Communication System). ACS can generate attention spontaneously, which is achieved by Feature Drift mechanism. Here, "feature" refers to an aspect of an object such as color or size. In Feature Drift, attention is locally consistent, but drifts over time. Then robot can behave unpredictable behaviors.

This paper is organized as follows. In section 2, we describe the robot which we implemented ACS on, and discuss the formalism of robot's behaviors and effect of attention. In section 3, the mechanism of Feature Drift is described in detail. In section 4, we explain the design of ACS. Section 5 describes the examples of conversation between a human and a robot using ACS. In section 6, we discuss the experimental results, and then we conclude with a brief summary in section 7.

## 2 Robots and Attentions

### 2.1 Robovie: Communication Robot

In this paper, Robovie[2] was used as the subject. Robovie is a humanoid robot developed at ATR Intelligent Robotics and Communication Laboratories<sup>1</sup>. It is designed to have the ability to communicate with humans.

Robovie has two arms(each arm has 4 DOF). The neck has 3 DOF and the head has two pan-tilt cameras, a microphone, and a speaker. Robovie also has an omni-directional camera on top of the shoulder pole, 24 ultrasonic ranging sensors, and touch sensors on its arms, chest, and head.

Robovie has fundamental communication behaviors such as "pointing" and "looking at a human's face". Robovie also has been used in some experiments of cognitive psychology.

### 2.2 Definition Robot's Attention

In order to solve the problem of fixed reaction, as I mentioned, robot must have an arbitrary policy for its behaviors, called attention.

<sup>1</sup> <http://www.irc.atr.co.jp/>

First of all, a robot behaves according to its own rules. The reaction of a robot is calculated from stimuli, and its own inner-state such like emotion. This relationship is written as follows:

$$\mathbf{r} = \beta(\mathbf{s}, \sigma). \quad (1)$$

where  $\mathbf{r}$ ,  $\beta$ ,  $\mathbf{s}$ ,  $\sigma$  denotes reaction of robot, a behavior rule of robot, stimuli for robot, and inner-state of robot. Because a robot has many behaviors, robot must have cooperation function,  $\mathbf{C}$ , to select one of the behaviors or to blend multiple behaviors. Then, the actual reaction,  $\rho$ , is defined as follows:

$$\rho = \mathbf{C} \left( \begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_n \end{bmatrix} \right) = \mathbf{C} \left( \begin{bmatrix} \beta_1(\mathbf{s}, \sigma) \\ \vdots \\ \beta_n(\mathbf{s}, \sigma) \end{bmatrix} \right). \quad (2)$$

In this case, the reaction of robot is fixed for a situation. When robot cannot determine its reaction rationally in above situation, the reaction must be determined by its designer a priori. Then we introduce attention of robot to this model.

Attention is an arbitrary policy to select the target of a behavior and to determine the behavior of a robot at the time. So, the equation (1) is rewritten as follows:

$$\mathbf{r}_i = \beta_i(\mathbf{s}, \sigma, \alpha) \quad (3)$$

and the equation (2) is rewritten as follows:

$$\rho = \mathbf{C}(\alpha) \left( \begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_n \end{bmatrix} \right) = \mathbf{C}(\alpha) \left( \begin{bmatrix} \beta_1(\mathbf{s}, \sigma, \alpha) \\ \vdots \\ \beta_n(\mathbf{s}, \sigma, \alpha) \end{bmatrix} \right) \quad (4)$$

where  $\alpha$  is attention of the robot.

The target of a robot's attention should change over time. If attention cannot change over time, reaction of robot is fixed for each situation. In fact, the predictability of robot's reaction strongly depends on the update rule of attention. If robot's attention drifts too frequently, its reaction becomes chaos. If robot's attention does not drift rarely, however, its reaction becomes fixed. Therefore, attention of robot should be stable in short term to prevent chaos reaction, and various in long term to prevent fixed reaction. To solve these requirements, we propose Feature Drift.

### 3 Feature Drift

Feature Drift is a method of maintenance and update of robot's attention. In Feature Drift, robot has a "context" for attention. A context of a robot is a set of feature which the robot paid attention to. Because a robot's attention is generated from its context, the attention has tendency to be similar to its context. Therefore, attention becomes stable. But, a robot's attention becomes inconsistent to its context sometime, which causes the drift of attention. We describe this below in detail.

First, we define the “feature” of an object. Let us assume that there is an object  $obj_i$  in environment. The  $obj_i$  has various features, such as “color”, “size” and so on. The  $obj_i$  has a value for a feature each other. Robots recognize an object as a set of pairs of feature and value. This is shown as,

$$obj_i = ((f_0, v_0), (f_1, v_1), \dots, (f_n, v_n)) \quad (5)$$

This may be written as follows: “ $v_i$  is a value of  $f_i$  in  $obj_i$ ”, or  $v_i = \phi_{obj_i}(f_i)$ .

Then, context in this paper is defined a set of pairs of feature and its value. This is written as:

$$c^t = \{(f_{c_0}, v_{c_0}), (f_{c_1}, v_{c_1}), \dots, (f_{c_l}, v_{c_l})\}. \quad (6)$$

A pair in context means that the robot paid attention to the pair at that time. Here,  $l$  is called the length of context.

In Feature Drift, an attention is generated from context. First, a number of pairs in the context is selected randomly. This number  $n$  is called as selecting number for attention. Next, the selected pairs is grouped by its feature and a range of value is constructed for each feature. If there are more than one values for the feature, the maximum and minimum value are the upper bound and lower bound for the feature. If there are only one value for a feature, the range is constructed as neighbor to the value.

Now robot can determine whether or not robot pays attention to an object at the time. If an object is included in the attention at the time, then robot pays attention to the object. This means that an object  $obj_i = (f_i, v_i)$  is included in an attention  $\alpha$  iff:

$$\forall f_j \in obj_i, \exists (r_j, f_j) \in \alpha, \phi_{obj_i}(f_j) \in r_j, \text{ or } f_j \notin \alpha \quad (7)$$

If an object  $obj_i$  is included in the attention at the time, the context is updated from a pair of  $obj_i$ . A pair  $(f_j, v_j) \in obj_i$  is selected and added into the context. This selection is based on the  $\alpha$ . the number of values is counted for each feature, and the most frequently appeared feature are selected.

Because the length of context  $l$  is fixed, one of pairs in the context is removed when a pair is added into context. The pair that is removed is selected randomly.

Iterating this update, the context tends to contain same features and similar values. Therefore, robot tends to pay attention to a same object over time, which achieves stability of attention.

However, if a context is filled by one feature, the context will not change any longer. This leads to fixed reaction of robot. Then, other feature should be added into context in this case. For this case, when  $\alpha$  has only one feature, it is thought that the context has converged into one feature and a pair of  $obj_i$  is not selected from frequency of feature in attention but selected randomly. In this case, other feature may be added into the context. This random selection of feature prevents convergence of context.

## 4 ACS: Attention-based Communication System

In this section, we explain the Attention-based Communication System (ACS). An overview of ACS is shown in Figure 1. ACS consists of five modules: Cooperation, Speech-Recognition, Context, Behavior-List, and Feature Drift.

The central component of ACS is Context Module. Context Module holds the context of robot and generate an attention from the context. Context Module does not receive sensor data directly, instead Feature Drift module receives sensor data and update the context. The detail of updating context is already described above.

Behavior of other modules are described below.

#### 4.1 Cooperation of Behaviors

Cooperation Module selects a behavior and executes it. This module corresponds to C in equation (4). The selection of Cooperation Module is affected by the attention. For example, when there seems to be no objects included in attention at the time, robot looks around for such objects.

Each behavior has the form of equation (3), then it requires attention. For example, `point_to` behavior requires the target of pointing. So, Cooperation Module selects behavior and applies attention into the behavior.

Speech-Recognition Module affects the selection rule of Cooperation Module. It receives human utterances, recognizes them, and puts the result of the recognition into Cooperation Module. Although Speech-Recognition Module also receives commands of human, the target of a behavior is attention of robot. So, the context should be able to be rewritten by human commands. The relationship between command and attention is described below.

#### 4.2 Top-Down Updating of Context

In ACS, robot's attention is determined by context which is maintained regardless of state of their conversation. However, in some situations, the target of a behavior should be determined by rules. For example, there may be a rule that if a human says "look at this", then the robot looks at the object. In these situations, it is desired that the output action update the context. As a result of updating the context, the robot begins to pay attention to the object. This process is called Top-Down Updating.

It is considered that the details of Top-Down Updating differs according to the situations or command, but the process is typically as follows. First, the target object is detected. Then, all of the contents in the context is then rewritten as a pair of target object. The pair selection process is same as the updating by Feature Drift.

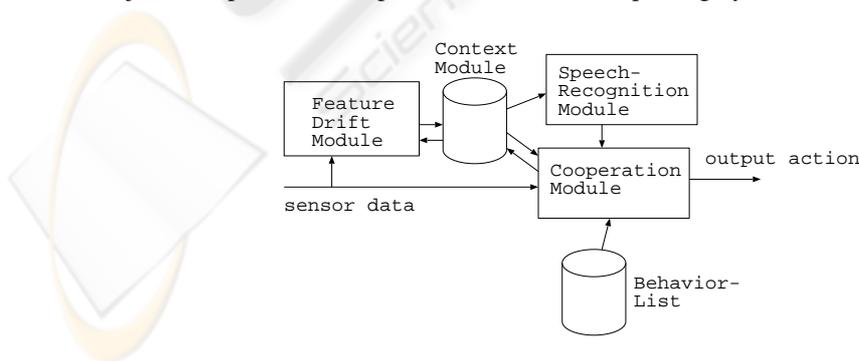


Fig. 1. Overview of ACS



**Fig. 2.** Example of Human-Robot Communication

However, the context is also modified by the normal process of Feature Drift. When a rule means that this drift of attention is not acceptable, Cooperation Module may suppress the normal process of Feature Drift to prevent spontaneous drift of attention.

## 5 Example of Interaction

In this section, we explain how ACS operates using examples of human-robot interaction. Assume a situation such as that shown in Figure 2. This scene includes one human, one Robovie, and two blocks. Both blocks are same size, but different color: One block is red and the other is blue. ACS has two behaviors in this example: `point` and `look_at`. When the human ask that “which one do you like?”, Cooperation Module selects `point` and Robovie points the block it pays attention to. When the human ask that “look at this”, Robovie selects an object and update context in top-down.

Table 1 shows the result of the interaction. H denotes human and R denotes Robovie.

All of the examples were executed in sequence: first was example 1, second was example 2, then example 3, example 4, and finally example 5 was done.

In example 1, the human asked Robovie’s attention. In this case, Robovie generate its attention from context and determined which the block included in the attention. In this example, Robovie had an attention for the red block at this time, and point it.

In example 2, the human ordered Robovie to look at the blue block. As a result of example 1, Robovie already has attention for the blue block. Then, Top-Down Updating of the context occurs so that Robovie updates the context for the red block. Robovie then faces the red block, according to the new attention. After that, the attention remained unchanged in example 3. Robovie still paid attention to the red block and then point it.

In example 4, the human ordered Robovie to look at the blue block again. the behavior was same to example 2. But in the following interaction, example 5, Robovie’s reaction was different. In this case, the drift of attention occurred before human’s question because the `look_at` does not suppress dynamic update of context by Feature

**Table 1.** Example of Conversation between human and Robovie

<p>Example 1</p> <p>H: Which one do you like?</p> <p>R: I like this one.(points to blue block)</p>
<p>Example 2</p> <p>H: Look at this(meaning the red block)</p> <p>R: OK.(turns towards the red block)</p>
<p>Example 3</p> <p>H: Which one do you like?</p> <p>R: I like this one.(points to the red block)</p>
<p>Example 4</p> <p>H: Look at this(meaning the red block)</p> <p>R: OK.(turns towards the red block)</p>
<p>Example 5</p> <p>H: Which one do you like?</p> <p>R: I like this one.(points to the blue block)</p>

Drift. Therefore, the robot paid attention to blue block at this time. The drift of attention in example 6 solves the problem of fixed reaction of robot.

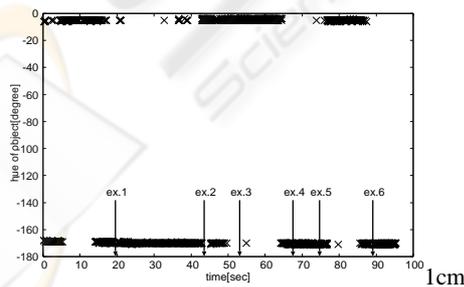
## 6 Evaluation

In this section, we discuss the results of using Robovie to demonstrate the system described in section 5. The length of context was 8 and the selecting number was 3.

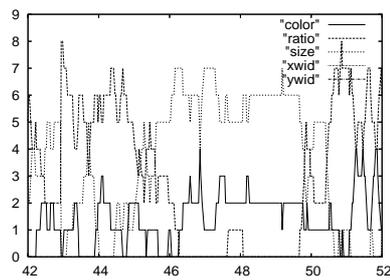
Figure 3 shows a transition of attention. The horizontal axis denotes time in seconds and the vertical axis denotes the color in hue. In addition, the markers on the horizontal axis denote the timing of the human utterance for each example.

As shown in Figure 3, Robovie at first paid attention to the red block. Then, its attention changed after about 15[sec] because of Feature Drift. Then, the human questioned Robovie (example 1). In this case, the response was “the blue one”.

The human ordered Robovie to look at the blue block at 43[sec]. Then the attention changed to red (example 2). This is the result of Top-Down Updating. Then, when human asked Robovie at 52[sec](example 3), it pointed red block.



**Fig. 3.** Transition of attention shown by Robovie



**Fig. 4.** Transition of context of Robovie

The human ordered again (example 4), Robovie became to look at the red block. But in this case, the attention drifted and the response was “the blue one”(example 5).

Then, the context were examined at each step when drift of attention occurs. Figure 4 denotes contents of the context at 42–52[sec]. In Figure 4, the horizontal also axis also denotes time in seconds, and the vertical axis denotes the number of pairs in the context for each feature.

In this figure, the most characteristic transition is at 43[sec], which is caused by Top-Down Update. Then, all of the contents in context is filled with a pair of red block. Therefore, attention drifts to red block, but this “ratio” pairs are removed as time passes. Then the context had many “size” pair, but the two blocks has similar size, then Robovie paid attention to both blocks at this time. Then the context became to contain many “ratio” again, the attention became stable.

In summary, ACS achieves dynamic maintenance of attention using Feature Drift and rule-based update of attention via Top-Down Updating.

## 7 Conclusion

In this paper, we described the problem of fixed reaction of robot and propose a system called ACS. In ACS, we introduce Feature Drift, then robot can maintain its attention and drift dynamically. Then, robot has various reactions for each situation, the problem of fixed reaction is solved. In case that various attention of robot may prevent some human commands, we also introduce Top-Down Update of context. With Top-Down Update, the attention is forced to drift to specific target. Therefore robot obeys human command. Then, the communication between human and robot is achieved by robot’s attention.

In future, we plan to And we will also add multi-robot interchange of context. For example, two robots pay attention to the same object, or a robot becomes to pay attention to what a human pays attention to. Because ACS has a few behaviors now, we will test the validity of ACS after implementing other behaviors.

## References

1. Ono, T., Imai, M.: Reading a robot’s mind: A model of utterance understanding based on the theory of mind mechanism. In: Proc. of AAAI-2000. (2000) 142–148
2. Ishiguro, H., Ono, T., Imai, M., Maeda, T., Kanda, T., Nakatsu, R.: Robovie: an interactive humanoid robot. *Intl. J. of Industrial Robot* **28** (2001) 498–503
3. NEC: (PaPeRo) NEC Personal Robot Center, <http://www.incx.nec.co.jp/robot/>.
4. Breazeal, C., Scasselatti, B.: How to build robots that make friends and influence people. In: Proc. of IROS99. (1999)
5. Sawada, T., Takagi, T., Fujita, M.: Behavior selection and motion modulation in emotionally grounded architecture for QRIO SDR-4X II. In: Proc. of IROS 2004. (2004) 2514–2519
6. Brooks, R.A.: Intelligence without reason. In: Proceedings of IJCAI ’91. (1991) 561–595
7. Arkin, R.C.: Behavior-Based Robotics. MIT Press (1998)