Teaching a Tele-robot using Natural Language Commands

Chandimal Jayawardena, Keigo Watanabe and Kiyotaka Izumi

Department of Advanced Systems Control Engineering, Graduate School of Science and Engineering, Saga University, Japan.
chandimal@ieee.org, {watanabe, izumi}@me.saga-u.ac.jp

ABSTRACT

For Internet-based tele operation systems, user-friendly natural interfaces are advantageous because those systems are intended to be used by non-experts. In developing user-friendly interfaces, natural language communication is mandatory. This paper presents a system in which a sub-set of natural language is used to command a tele-robot manipulator doing an object sorting task. The paper discusses about referring to objects with natural language commands such as “pick the small red cube”. This is achieved by learning individual lexical symbols that refer to colors, shapes, and sizes independently, and then inferring the meaning of a combination of them.

1 INTRODUCTION

During the last decade a new class of robots has been emerged. These Internet or Online Robots, provide users with unprecedented experiences such as visiting museums, tending gardens, drawing pictures, etc. over the Internet. However, such systems pose a number of technical challenges too. Unlike early telerobotic systems, these are used by non-specialists. Therefore, intuitive user interfaces and human-friendly control methods are necessary.

Using Internet as the medium of communication for tele-robots has become increasingly popular due to advantages such as, low cost, easy accessibility, easy maintenance, possibility of allowing public participation, etc.

Since Internet based tele-robots are meant to be used by non-experts and the expected user base is very high and diverse, the human-robot interaction should be as natural and user-friendly as possible. In most existing Internet based robotic systems [1][2][3][4][5][6], user interaction with robot is via filling forms, selecting commands, clicking with mouse, etc. In contrast, a more user-friendly framework for tele-operation using natural language commands would be more appropriate.

<table>
<thead>
<tr>
<th>Table 1: Grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>pick</td>
</tr>
<tr>
<td>grab</td>
</tr>
<tr>
<td>take</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Sanz et al. [7] proposed for the first time, the importance of using natural words to refer to objects in tele-operation systems. However, there, the authors’ concentration was on object recognition aspects. In contrast, this paper proposes a model for complete voice command understanding via lexical symbol learning.

2 SYSTEM DESCRIPTION

Overview of the system is shown in the Fig. 1. On the object table, objects of different colors, shapes and sizes are placed.

User can see the objects and can ask the robot to pick any object using a sub-set of the natural language. For example, user may say “pick the small red cube”. Valid grammar is shown in the Table 1. Any combination of action, size, color, and shape would form a valid command. The article is optional.

Ignoring the article, this type of a command can be analyzed as follows:

\[
\text{command} = \langle \text{action} \rangle + \langle \text{size.adjective} \rangle + \\
\langle \text{color.adjective} \rangle + \langle \text{shape} \rangle \\
= \langle \text{action} \rangle + \langle \text{size.adjective} \rangle + \\
\langle \text{colored.object} \rangle \\
= \langle \text{action} \rangle + \langle \text{object} \rangle
\]

In order to perform the \(\langle\text{action}\rangle\) on the \(\langle\text{object}\rangle\), the robot should be able to understand the action user intended.
and the object user referred. In this implementation, only object identification is considered. All action commands, pick, grab, and take are treated as the same action of picking the referred object.

To identify the correct object, all individual lexical symbols, <size_adjective>, <color_adjective>, and <object> should be known. The algorithm shown in the Fig. 2 is used to learn lexical symbols and to identify the object.

Object identification is performed in the shape, color, size order. This is in coincidence with the correct adjective order of English language.

This poses two questions. First, how objects are perceived and differentiated by the robot. Second, how the lexical representations of objects in human language is attached to the objects perceived by the robot.

2.1 Object Perception by the Robot

Object perception by any robot is only via sensors. In this case camera images. It is possible to identify different objects in the scene and to extract their features without referring to any lexicon. Let the number of objects be n. Object i can be represented with a vector \( r_i \).

\[
r_i = \{s_i, c_i, h_i\}
\]

\( s_i, c_i, \) and \( h_i \) are size, color, and shape feature vectors respectively.

\[ R = \{r_1, r_2, ..., r_n\} \]

is the set of all objects and it is stored in the low level knowledge base (Fig. 1). It is low level in the sense that it contains only organized raw sensory data and does not contain lexical symbols of human language.

Since these objects \( 1, ..., n \) belong to a finite number of colors and shapes, it should be possible to identify color and shape clusters within sensory data given by vectors \( s_i, c_i, \) and \( h_i \). If these clusters are correctly identified, the number of color clusters (v) should be equal to the number of object colors and the number of shape clusters (w) should be equal to the number of object shapes.

Objects clustered according to color are given by:

\[
a_p \subset R
\]

where \( p = 1, ..., v \) and \( a_p \cap a_q = 0 \) where \( p_1 \neq p_2 \). For \( \forall r \in a_p \), \( \|c_r - \mu_{a_p}\| < \theta_c \), where \( c_r \) is the color feature component of \( r \), \( \mu_{a_p} \) is the mean of the color feature components of all elements in \( a_p \), and \( \theta_c \) is a threshold value which sets the color deviation within the cluster. If the set of all \( a_p \) is \( V \),

\[
V = \{a_1, a_2, ..., a_v\}
\]

Objects clustered according to shape are given by:

\[
b_q \subset R
\]

where \( q = 1, ..., w \) and \( b_q \cap b_q = 0 \) where \( q_1 \neq q_2 \). For \( \forall r \in b_q \), \( \|h_r - \mu_{b_q}\| < \theta_h \), where \( h_r \) is the shape feature component of \( r \), \( \mu_{b_q} \) is the mean of the shape feature components of all elements in \( b_q \), and \( \theta_h \) is a threshold value which sets the shape deviation within the cluster. If the set of all \( b_q \) is \( W \),

\[
W = \{b_1, b_2, ..., b_w\}
\]
01. Get the user command.
02. Extract <action>,
   <size.adjective>,
   <colour.adjective>,
   <shape>
03. IF <shape> is known
04. IF <colour.adjective> is known
05. IF <size.adjective> is known
06. • Localize the object
07. • Perform <action> on object.
08. • Stop.
09. ELSE
10. • Consult the user.
11. • User points to an example.
12. • Learn <size.adjective>.
13. • Go to step 6.
14. ELSE
15. • Consult the user.
16. • User points to an example.
17. • Learn <colour.adjective>.
18. • Go to step 5.
19. ELSE
20. • Consult the user.
21. • User points to an example.
22. • Learn <shape.adjective>.
23. • Go to step 4.

Figure 2: Object identification.

Therefore, sets objects which belong to the same color and same shape are given by:

$$t_{pq} = a_p \cap b_q$$  \hspace{1cm} (7)

where \(p = 1, \ldots, v\) and \(q = 1, \ldots, w\).

Elements of \(t_{pq}\) are of color \(p\) and shape \(q\). If \(m\) is the number of objects in \(t_{pq}\),

$$t_{pq} = \{r_1^{pq}, r_2^{pq}, \ldots, r_m^{pq}\}$$  \hspace{1cm} (8)

Note that \(t_{pq}\) is a well ordered set, if the size representing numerical value obtained from \(s_i\) of \(r_i^{pq}\) is taken to be the ordering relation.

2.2 Lexical Representations

Although the robot perception is limited to sensory data, a human user may refer to objects with lexical symbols. "red cube", "blue cylinder", or "big yellow sphere" are some examples. In order to execute user commands which consist of such references, there should be a method to map these lexical symbols to raw sensory data.

There have been many important work related to this problem [8][9][10]. However, those work considered the problem as a fundamental cognitive problem. In this paper, learning the mapping between lexical symbols and their physical representations with the help of a human user is studied. On the other hand, it is not limited to acquiring knowledge of some symbols; rather it uses independently learned lexical symbols to understand the meaning of a composite lexical item: i.e. a complete command. For example, after the meaning of the lexical symbol "red" is learned, it is meaningful for any red object; "red cube", "red cylinder", etc.

Suppose, in the user lexicon, the set of color lexical symbols is \(\ell_c\) and the set of shape lexical symbols is \(\ell_s\). Color and shape lexical symbol learning is described by the bijective functions \(f_c\) and \(f_s\) such that

$$f_c : \ell_c \rightarrow V$$  \hspace{1cm} (9)

$$f_s : \ell_s \rightarrow W$$  \hspace{1cm} (10)

Let the set of lexical symbols that represent relative sizes of objects within any \(t_{pq}\) is \(\ell_s\). Then, learning of size lexical symbol is described by the bijective function \(f_s\) such that

$$f_s : \ell_s \rightarrow t$$  \hspace{1cm} (11)

where \(t\) represents any \(t_{pq}\).

High level knowledge base contains these mapping information.

3 IMPLEMENTATION

3.1 Experimental Setup

Server side of the experimental setup consisted of a PA-10 industrial manipulator and controller, object table, three USB cameras, and a PC running WindowsXP. The three cameras were placed over, in front of, and on left of the object table. For the image acquisition DirectX technology was used.

The client side consisted of a PC running WindowsXP and a microphone. Voice recognition was performed using IBM ViaVoice SDK.

Both the server and the client were connected to an Ethernet network and the communication was performed using TCP/IP sockets.

3.2 Image Acquisition

The camera placed right above the table provides a calibrated image and it is further processed in order to extract object features. Low resolution versions of all three images are sent to the communication module and they are sent to the user in order to provide 3 dimensional details of the workspace.
3.3 Low Level Knowledge Base

Object feature extraction module in Fig. 1 extracts shape, color and size representations of each object.

3.3.1 Shape representation

Shape representation of an object should be invariant to change in size, to change in location and to rotation. Although there are various descriptors such as thinness ratio, shape elongation, spreading, compactness, etc. Hu descriptors has the particularity of being invariant to scale, translation and rotation [11].

For a 2 dimensional function \( f(x, y) \), the moment of order \((p+q)\) is defined as:

\[
m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) dxdy
\]

(12)

for \( p, q = 0, 1, 2, \ldots \)

A uniqueness theorem [12] states that if \( f(x, y) \) is piecewise continuous and has nonzero values only in a finite part of the \( xy \)-plane, moments of all orders exist, and the moment sequence \( \{ m_{pq} \} \) is uniquely determined by \( f(x, y) \). Conversely, \( m_{pq} \) uniquely determines \( f(x, y) \).

The central moments are defined as:

\[
\mu_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dxdy
\]

(13)

where \( \bar{x} = m_{10}/m_{00} \) and \( \bar{y} = m_{01}/m_{00} \).

If \( f(x, y) \) is a digital image, the Eq. 13 becomes:

\[
\mu_{pq} = \sum_{x} \sum_{y} (x - \bar{x})^p (y - \bar{y})^q f(x, y)
\]

(14)

The normalized central moments are defined as:

\[
\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}
\]

(15)

where \( \gamma = \frac{p+q}{2} + 1. \)

From the normalized moments of order up to three, it is possible to derive seven invariant moments or Hu descriptors. In this work, only first Hu descriptor, \( \phi_1 \) was used as the shape representation,

\[
\phi_1 = \eta_{20} + \eta_{02}
\]

(16)

because it is sufficient to distinguish among the shapes used in this experiment.

If more complicated and diverse shapes are used, all seven Hu descriptors can be used to increase the representing accuracy.

3.3.2 Size representation

Number of pixels of an object was used as the size representation.

3.3.3 Color representation

Normalized red \( r \), green \( g \), and blue \( b \) components were used as the color representations.

\[
r = \frac{R}{R+G+B}
\]

(17)

\[
g = \frac{G}{R+G+B}
\]

(18)

\[
b = \frac{B}{R+G+B}
\]

(19)

where \( R, G, \) and \( B \) are the components of an RGB color pixel.

These three kind of representations are the elements of \( r_i \) in Eq. 1. All \( r_i \)'s (or \( R \)) are stored in the low level knowledge base.

Since these objects belong to finite number of colors and shapes, it should be possible to identify color and shape clusters within sensory data shown in the Table 2. If these clusters are correctly identified, the number of color clusters should be equal to the number of object colors and the number of shape clusters should be equal to the number of object shapes.

3.3.4 Object clustering

We need to cluster objects according to their shapes and colors only. Size is a relative term which is interpreted once a <colored object> is identified (Section 2). For example, once two "red cubes" are identified, then only we can find the "big red cube" and the "small red cube".

The number of clusters is not a priori known for both shape and color. Therefore, we have used a leader-follower algorithm to find clusters because it need not know the number of clusters in advance [13]. The algorithm is as follows.

Let,
Table 2: Object representations.

<table>
<thead>
<tr>
<th>Object No.</th>
<th>Pixels</th>
<th>Hu1</th>
<th>Color (r, g, b)</th>
<th>Center (x, y) pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2081</td>
<td>496</td>
<td>0.1067, 0.3528, 0.5652</td>
<td>255.765</td>
</tr>
<tr>
<td>2</td>
<td>5744</td>
<td>206</td>
<td>0.1181, 0.3465, 0.5354</td>
<td>262.461</td>
</tr>
<tr>
<td>3</td>
<td>5296</td>
<td>498</td>
<td>0.1897, 0.4545, 0.3557</td>
<td>288.606</td>
</tr>
<tr>
<td>4</td>
<td>5067</td>
<td>494</td>
<td>0.6102, 0.2165, 0.1732</td>
<td>366.301</td>
</tr>
<tr>
<td>5</td>
<td>2458</td>
<td>207</td>
<td>0.6024, 0.2362, 0.1614</td>
<td>375.759</td>
</tr>
<tr>
<td>6</td>
<td>5467</td>
<td>207</td>
<td>0.2047, 0.4685, 0.3268</td>
<td>437.555</td>
</tr>
<tr>
<td>7</td>
<td>540</td>
<td>209</td>
<td>0.1344, 0.3360, 0.5296</td>
<td>473.415</td>
</tr>
<tr>
<td>8</td>
<td>2445</td>
<td>207</td>
<td>0.1600, 0.3241, 0.5099</td>
<td>540.811</td>
</tr>
<tr>
<td>9</td>
<td>1698</td>
<td>508</td>
<td>0.1850, 0.4488, 0.3661</td>
<td>562.386</td>
</tr>
<tr>
<td>10</td>
<td>2951</td>
<td>498</td>
<td>0.1732, 0.4331, 0.3937</td>
<td>559.270</td>
</tr>
<tr>
<td>11</td>
<td>4917</td>
<td>505</td>
<td>0.1462, 0.3557, 0.4980</td>
<td>622.611</td>
</tr>
<tr>
<td>12</td>
<td>5171</td>
<td>206</td>
<td>0.6181, 0.2362, 0.1457</td>
<td>675.438</td>
</tr>
<tr>
<td>13</td>
<td>507</td>
<td>207</td>
<td>0.3700, 0.4348, 0.3953</td>
<td>713.642</td>
</tr>
<tr>
<td>14</td>
<td>2910</td>
<td>517</td>
<td>0.1024, 0.3386, 0.5591</td>
<td>781.284</td>
</tr>
<tr>
<td>15</td>
<td>2333</td>
<td>206</td>
<td>0.2087, 0.4409, 0.3504</td>
<td>774.748</td>
</tr>
<tr>
<td>16</td>
<td>2990</td>
<td>503</td>
<td>0.2612, 0.2292, 0.1581</td>
<td>847.568</td>
</tr>
<tr>
<td>17</td>
<td>418</td>
<td>209</td>
<td>0.4900, 0.2451, 0.3269</td>
<td>822.451</td>
</tr>
<tr>
<td>18</td>
<td>1912</td>
<td>499</td>
<td>0.6220, 0.2165, 0.1614</td>
<td>854.805</td>
</tr>
</tbody>
</table>

\( w_i \) = current center for cluster \( i \),
\( \theta \) = threshold,
\( x \) = a sample.

\[ \text{begin initialize } \nu, \theta \]
\[ w_0 \leftarrow x \]
\[ \text{do accept new } x \]
\[ j \leftarrow \arg \min_j \| x - w_j \| \text{ (find nearest cluster)} \]
\[ \text{if } \| x - w_j \| < \theta \]
\[ \text{then } w_j \leftarrow 0.5(w_j + x) \]
\[ \text{else add new } w \leftarrow x \]
\[ \text{until no more patterns} \]
\[ \text{return } w_1, w_2, ... \]
\[ \text{end} \]

For shape clustering, \( \theta \) is taken to be 100. \( x \) are the Hu moments given in the third column of the Table 2. For color clustering, \( \theta \) is taken to be 0.1. \( x \) are the normalized \( r, g, b \) vectors given by the fourth column.

3.4 High Level Knowledge Base

High level knowledge base contains the mappings between lexical symbols and corresponding classes. Initially, high level knowledge base is empty. It is filled according to the algorithm in the Fig 2. Once learned, it contains the mapping between high level lexical symbols and the clusters based on the sensory data.

### Table 3: Clustered objects.

<table>
<thead>
<tr>
<th>Shape Cluster</th>
<th>Object No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1, 3, 4, 9, 10, 11, 14, 16, 18</td>
</tr>
<tr>
<td>2</td>
<td>2, 5, 6, 7, 8, 12, 13, 15, 17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Color Cluster</th>
<th>Object No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1, 2, 7, 8, 11, 14</td>
</tr>
<tr>
<td>2</td>
<td>3, 6, 9, 10, 13, 15</td>
</tr>
<tr>
<td>3</td>
<td>4, 5, 12, 16, 17, 18</td>
</tr>
</tbody>
</table>

### Table 4: Lexical symbols to cluster mapping

<table>
<thead>
<tr>
<th>Lexical symbol</th>
<th>Shape cluster</th>
<th>Color cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>cube</td>
<td>1</td>
<td>~</td>
</tr>
<tr>
<td>cylinder</td>
<td>2</td>
<td>~</td>
</tr>
<tr>
<td>red</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>green</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>blue</td>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table 5: Objects of same color and shape

<table>
<thead>
<tr>
<th>Colored Object</th>
<th>Object Nos.</th>
</tr>
</thead>
<tbody>
<tr>
<td>red cube</td>
<td>4, 16, 18</td>
</tr>
<tr>
<td>red cylinder</td>
<td>5, 12, 17</td>
</tr>
<tr>
<td>green cube</td>
<td>3, 9, 10</td>
</tr>
<tr>
<td>green cylinder</td>
<td>6, 13, 15</td>
</tr>
<tr>
<td>blue cube</td>
<td>1, 11, 14</td>
</tr>
<tr>
<td>blue cylinder</td>
<td>2, 7, 8</td>
</tr>
</tbody>
</table>

#### 3.4.1 Interpretation of the size symbol

In this implementation we have considered only three sizes. They correspond to the lexical symbols small, medium, and big. Out of the three objects that belong to the same color and the same shape, the object having the smallest number of pixels is mapped to small, next one to medium and the one with the highest number of pixels to big.

### 4 RESULTS AND CONCLUSION

An image of the object table taken from the top camera is shown in the Fig. 3. The Table 2 shows how the object 1 to 18 are represented in the low level knowledge base. Table 3 shows objects clustered according to shape and color.

Mapping between individual lexical symbols and clusters is shown in the Table 4. By comparing Tables 3 and 4, we can
Table 6: Final object identification.

<table>
<thead>
<tr>
<th>Object No.</th>
<th>Lexical Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>small blue cube</td>
</tr>
<tr>
<td>2</td>
<td>big blue cylinder</td>
</tr>
<tr>
<td>3</td>
<td>big green cube</td>
</tr>
<tr>
<td>4</td>
<td>big red cube</td>
</tr>
<tr>
<td>5</td>
<td>medium red cylinder</td>
</tr>
<tr>
<td>6</td>
<td>big green cylinder</td>
</tr>
<tr>
<td>7</td>
<td>small blue cylinder</td>
</tr>
<tr>
<td>8</td>
<td>medium blue cylinder</td>
</tr>
<tr>
<td>9</td>
<td>small green cube</td>
</tr>
<tr>
<td>10</td>
<td>medium green cube</td>
</tr>
<tr>
<td>11</td>
<td>big blue cube</td>
</tr>
<tr>
<td>12</td>
<td>big red cylinder</td>
</tr>
<tr>
<td>13</td>
<td>small green cylinder</td>
</tr>
<tr>
<td>14</td>
<td>medium blue cylinder</td>
</tr>
<tr>
<td>15</td>
<td>medium green cylinder</td>
</tr>
<tr>
<td>16</td>
<td>small red cylinder</td>
</tr>
<tr>
<td>17</td>
<td>small red cylinder</td>
</tr>
<tr>
<td>18</td>
<td>medium red cube</td>
</tr>
</tbody>
</table>

derive the table 5.

We can see that there are three objects of same color and shape. They should be identified with their sizes. As stated in Section 3.4.1, this is performed based on the pixel size. Final object identification result is shown in the Table 6.

In this paper, we have discussed the possibility of introducing a natural language interface for a tele-robotic system.

The proposed concept was demonstrated with an object sorting experiment using a PA-10 redundant manipulator. Users could command the robot with natural language commands to pick any of the objects placed on a table. To identify the referred objects, complete user grammar understanding system based on individual lexical symbol learning was presented.

In this implementation, only three size related lexical symbols were considered. They are small, medium and big. Incorporating more such lexical symbols and study about their interpretation is a future work. On the other hand, in this paper we have not considered the learning of actions. That too is a possible improvement that can be included in a future work.

REFERENCES


