ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200371

A Fuzzy Inference System for a Visually Grounded Robot State of Mind

Avinash Kumar Singh and **Neha Baranwal** and **Kai-Florian Richter**[1]

Abstract. In order for robots to interact with humans on real-world scenarios or objects, these robots need to construct a representation ('state of mind') of these scenarios that a) are grounded in the robots' perception and b) ideally should match human understanding and concepts. Using table-top settings as scenario, we propose a framework that generates a robot's ' 'state of mind' by extracting the objects on the table along with their properties (color, shape and texture) and spatial relations to each other. The scene as perceived by the robot is represented in a dynamic graph in which object attributes are encoded as fuzzy linguistic variables that match human spatial concepts. In particular, this paper details the construction of such graph representations by combining low-level neural network-based feature recognition and a high-level fuzzy inference system. Using fuzzy representations allows for easily adapting the robot's original scene representation to deviations in properties or relations that emerge in language descriptions given by humans viewing the same scene. The framework is implemented on a Pepper humanoid robot and has been evaluated using a data set collected in-house.

1 INTRODUCTION

Creating a 'robot state of mind', i.e., an internal representation about the perceived outer world, is an important step in capturing a robot's understanding of this outer world [18]. A robot state of mind can be based on different sensory inputs, such as audio, infrared/laser, or vision. Vision is one of the primary sensors for most humanoid robots. It is useful in SLAM, navigation, object recognition, or obstacle avoidance, among others. Vision is also the primary sense for us humans; thus, it seems advantageous to ground the robot's state of mind in visual perception for human-robot interaction. Ideally, this state of mind should match human concepts to facilitate interaction.

Our work is set in the context of human-robot interaction. It is the robot's task to identify objects in its visual perception that have been mentioned by a human user in natural language requests. Specifically in this paper, we propose a framework that captures object properties, including color, shape, and texture, and their spatial relationships in a dynamic graph, in which edges are labeled using fuzzy linguistic variables that match human concepts of these relationships. These properties are helpful in identifying an object in the presence of ambiguity or false positives produced by the robot's recognition system. False positives occur where a robot mistakes an object for one of a different (incorrect) type. For example, if the robot recognizes an apple for a sports ball, then in interaction with the robot additionally specifying the shape, color or texture of the apple may help the robot to still identify the correct (intended) object despite having recognized the wrong type. Furthermore, spatial relationships between objects, or of an object to the overall scene, may help resolving such recognition issues, for example, talking about 'the apple next to the cup' or 'the apple in the top-right corner of the table.'. However, often people do not agree on the applicability of specific values for the properties or spatial relations. For example, while some may call an apple 'red', others may name it 'orange,' or some may describe the apple to be next to the cup, while others may say 'the apple to the right of the cup.' Thus, there is ambiguity and impreciseness in these properties and relations, and in their linguistic labels (cf. [14]). That is why we propose a fuzzy inference system to deal with these issues.

We use artificial neural networks (ANN) to extract object properties from the perceived scene [12, 16]. The probabilities at the networks' output layers are then translated into fuzzy membership values. For example, the color property is represented by the fuzzy set $object\_color$; all color categories are assigned to this set with their membership value derived from the network's output layer. This allows coping with discrepancies between a color perceived by the robot—say a 'yellow' banana—and what the human seems to see—e.g., a 'green' banana. Similar fuzzy membership functions are constructed for shape and texture as well as for spatial relations.

In constructing the knowledge graph that represents the robot's 'state of mind', for every attribute (the different object properties and spatial relations) we select the highest membership value of the corresponding fuzzy sets; these are taken to be the most likely, or most applicable, attributes to hold for each object in the scene. Figure 1 provides an overview of our framework[2]. The graph is grounded in a robot's visual perception of tabletop scenes, which we use as setting here. The graph is produced dynamically and changes as the robot shifts visual attention. The nodes of the graph represent the different objects of the scene while their attributes are captured by the edges.

The proposed system not only allows for identifying objects under ambiguity and imprecision, but can also be used to provide feedback to human users if the robot cannot resolve mismatches between its perception and the human request. Mismatches may occur because of some failed visual recognition or because the request is ambiguous, among others. Examples of mismatches are shown in Figure 2. Here, the robot generates natural language requests for clarification using the properties and relationships it has previously extracted.

An empirical evaluation estimates the differences between the system-generated visual grounding and human-given language grounding for several different scenes. Differences are measured in terms of graph mismatch. The main contributions of the paper are:

1. A combination of low-level neural network-based feature extraction from a robot's camera image and a high-level fuzzy inference system that allows handling of ambiguity in and mismatches be-

[1] Ume University, Sweden, email: {avinash,neha,kaifr}@cs.umu.se

[2] A larger image of the scene can be seen in Figure 3.

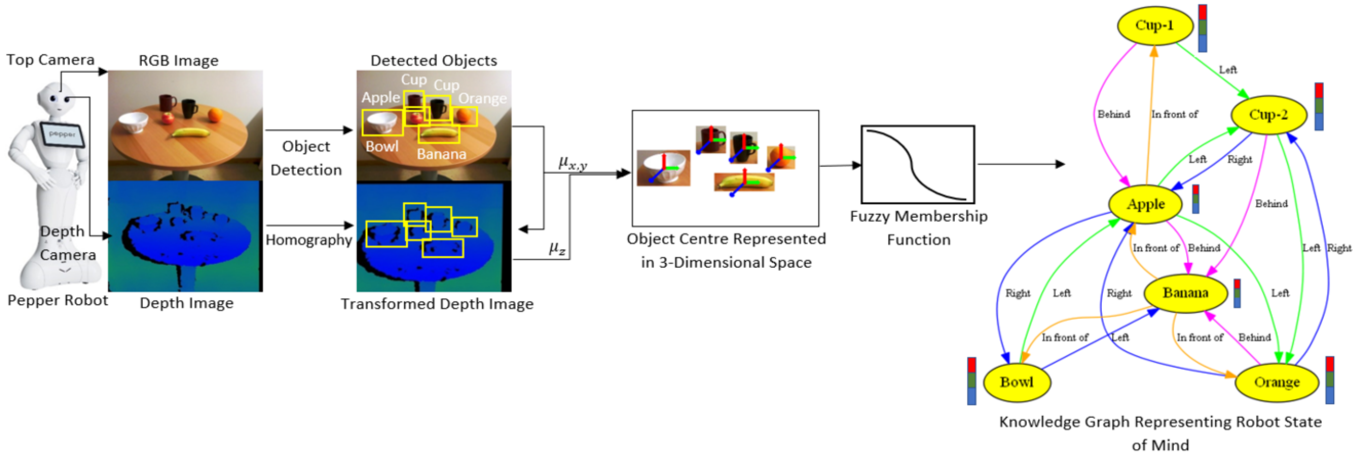

Figure 1. Overview of the visual grounding / fuzzy inference framework proposed in this paper. The edges of the graph (right) represent the most likely spatial relations. For readability reasons we symbolize object properties as an RGB vector indicating color, shape, and texture properties instead of showing actual values and edges for these attributes.

tween human natural language requests and the robot's visually grounded 'state of mind.'

2. the generation of targeted natural language clarification requests that allow a robot to resolve said ambiguities in interaction with a human user.

The paper is structured as follows. Section 2 discusses related work. Section 3 provides details on visually grounding object properties; Section 4 introduces the fuzzy inference system and grounding of spatial relations. Section 5 explains how natural language interaction between human and robot is realized in our system, while Section 6 presents an evaluation of the combined feature extraction / fuzzy inference system. Finally, Section 7 concludes the paper.

2 RELATED WORK

Visual grounding of a scene is fundamental in being able to interacting on it. It provides a representation of the different objects present, their properties, and location with respect to each other. Most research (e.g., [8, 18, 21]) has used convolutional neural network or Long-Short Term Memory (LSTM)-based deep neural network architectures for visual grounding, even if there is some research using statistical modelling (e.g., [3, 11]).

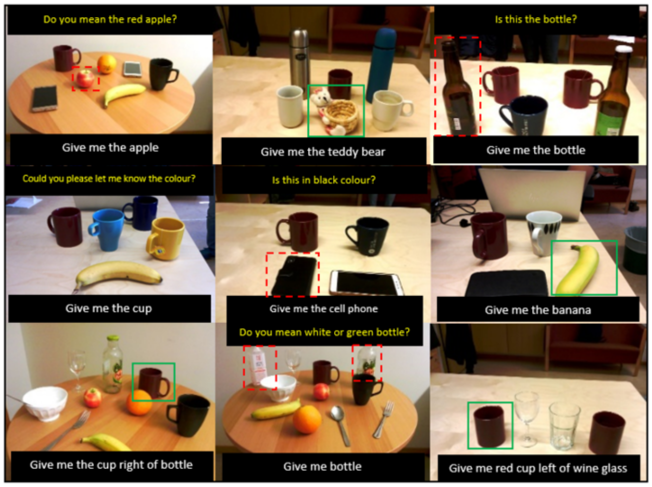
Do you mean the red apple?
Give me the apple
Give me the teddy bear
Is this the bottle?
Give me the bottle
Could you please let me know the colour?
Give me the cup
Is this in black colour?
Give me the cell phone
Give me the banana
Do you mean white or green bottle?
Give me the cup right of bottle
Give me bottle
Give me red cup left of wine glass

Figure 2. Examples of mismatches and the robot's clarification requests (bottom: human request; top: robot answer). Red boxes mark the most likely object shown as feedback response; green boxes the identified object.

Vavrecka et al. [21] presented an unsupervised learning algorithm for spatial grounding, which extracts shape, color and spatial relations of objects in a scene. Self-organizing maps and a neural gas algorithm are used for grounding, using scene descriptions and object features as input. Kittler et al. [11] introduced unsupervised physical symbol grounding using a visual bootstrapping method. They employ a recursive clustering approach where each domain, such as color or shape, are clustered and mapped to achieve the final goal. A data driven approach is proposed by Grollman et al. [5] where an infrared sensor is used as an additional sensor to capture depth information. Bayesian clustering and ISOmap as a dimension reduction method are used to classify categories. The authors tested their system in indoor and outdoor environments.

Golland et al. [3] proposed a game theoretic model for grounding that aims at identifying spatial relations between objects. Guadarrama et al. [6] mitigated the constraints setup in [3] by using a probabilistic approach. The main object of a visual scene and its spatial relationships with other objects are extracted. This information is combined with semantic parsing of sentences using template matching and a probabilistic approach. The authors used explainable AI concepts for visual grounding [8]. The features obtained from each module are fed into a LSTM network to obtain the final score of grounding. Some researchers (e.g., [9, 10, 20]) have used fuzzy systems to ground visual understanding of objects. However, they have not fully exploited object properties, such as color, shape, or texture.

Mast et al. [14] developed a system that can handle vagueness in understanding and producing object descriptions in visual scenes. Vagueness, or graded category membership, is computed using a exponentially decaying similarity function (following [2]). Similar to our work, they used color, shape, size, and spatial relations as features in the object descriptions. Their work differs in how these features are identified and represented, and only uses configurations of simple 2D geometric shapes in the visual scenes.

3 VISUALLY GROUNDING OBJECT ATTRIBUTES

We extract the relevant properties of all objects from an image taken by the robot's cameras. We use shape, texture, and color as they define an object visually (to a large extent). Determining spatial relations between objects is further detailed in the next section. Taken together, these attributes provide a visual grounding of a scene's objects and their relationship to each other.

3.1 Object recognition

In order to extract their properties we need to first detect the objects in the image. We use Mask-RCNN [7] for object recognition (i.e., identifying the type of object) and segmentation. Mask-RCNN is pre-trained on the COCO dataset [13], which has 12 main categories, such as 'sports', 'food', 'electronic', 'kitchen', 'furniture', or 'indoor.' These categories cover 80 different objects of which we use the following 21 in our experiments: 'dining table', 'laptop', 'mouse', 'keyboard', 'cell phone', 'banana', 'apple', 'orange', 'broccoli', 'carrot', 'cake', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'book', 'scissors', 'teddy bear.' An example object recognition and segmentation is shown in Figure 3.

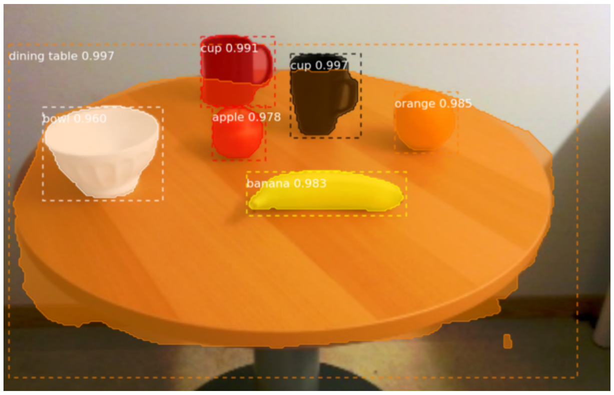

Figure 3. An example scene from our experiments showing the bounding boxes of the recognized objects, as well as the labels and probabilities for the specific object types.

3.2 Estimating object shape

Mask-RCNN [7] provides bounding boxes for each region containing the pixels of each object. In order to extract the actual shape of an object, we convert these bounding boxes into dummy images where all points (pixels) belonging to the object are 'black' and all others 'white.' From this black and white image we easily get the boundary points of the object as shown in Figure 4; these define the shape of the object as perceived by the robot. We distinguish five categories of shapes, namely 'triangle', 'rectangle', 'square', 'circle', and 'cylinder.' Due to occlusions shapes may be distorted, thus, we use the convex hull rather than the actual extracted boundary. We represent the shape of an object using a 5-dimensional vector ($O_i \in R^5$), representing the shape features listed below. We train a multi layer perceptron [16] with 2 hidden layers and 10 neurons in each layer with a softmax activation function to classify these shapes, using labeled data of table-top scenes (see Section 6 for details on the data).

1. **Aspect Ratio:** states the ratio between the width and height of an object's boundary. This feature discriminates triangle and circle from rectangle and square.

$$AspectRatio = \frac{width}{height} \tag{1}$$

2. **Extent:** is the ratio between the object area and area of its bounding box.

$$Extent = \frac{ObjectArea}{h_{BB} * w_{BB}} \tag{2}$$

3. **Solidity:** is the ratio between the object area and the area of its convex hull.

$$Solidity = \frac{ContourArea}{ConvexHullArea} \tag{3}$$

4. **Equivalent Diameter:** is the diameter of the circle whose area is the same as the contour area. This helps in discriminating circular shapes from other shapes.

$$EquivalentDiameter = \frac{4 * ContourArea}{\pi} \tag{4}$$

5. **Degree of Polynomial:** helps to predict a shape, e.g., a degree of 3 likely indicates a triangle. We first estimate the convex hull of an object and then approximate this polygon with another polygon of lesser degree.

3.3 Determining object texture

We use Local Binary Pattern (LBP) [15] to estimate the texture of an object. LBP codes each texture pixel based on its neighbors (Equation 5). If the intensity (i_c) of the center pixel (x_c, y_c) is less than the intensity (i_p) of its neighbor (x_p, y_p), then the pixel value of (x_p, y_p) is set to 1 else to 0. We use a radius R of 3 and 24 neighbors P.

$$\begin{aligned} \text{LBP}_{P,R} &= \sum_{i=0}^{P-1} 2^i * S(i_p - i_c) \\ \text{where } S(x) &= \begin{cases} 0, & \text{if } x >= 0 \\ 1, & \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

Further, we use the intensity histogram of an object as a feature, represented as a vector R^{256}. Examples of some objects, their texture, and histogram are depicted in Figure 5. We use the three texture classes 'shiny', 'smooth' and 'rough' and train another multi-layer perceptron (2 hidden layers with 100 neurons) to classify the texture of objects.

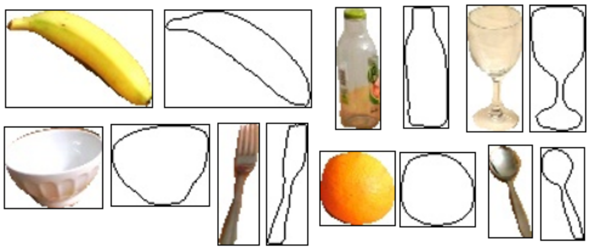

Figure 4. Examples of segmented objects and their extracted shape.

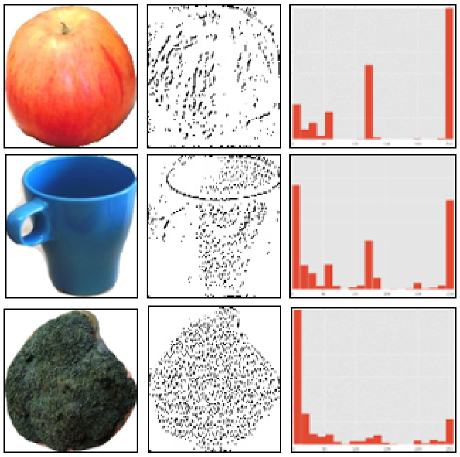

Figure 5. Examples of objects, their LBP image and intensity histogram (x-axis: intensities in the range [0-255]; y-axis: frequency). The first row shows a ‘shiny’, the second a ‘smooth’, and the third a ‘rough’ texture.

3.4 Determining object color

We use a convolutional neural network (CNN) for classifying objects according to nine predefined color categories: ‘black’, ‘blue’, ‘orange’, ‘purple’, ‘red’, ‘white’, ‘yellow’, ‘green’, ‘pink.’ An AlexNet [12] with five convolution layers, three max pooling layers, and two fully connected layers with dropout is trained on 128x128 pixel images downloaded from Google image searches on specific color names. Sample images are shown in Figure 6. 500 images per color category are used, with a 60:40 ratio between training and validation set. Clearly, human users may use other colors as well in their verbal requests, such as ‘gray’ or ‘maroon’, but these can be covered under the nine color categories (e.g., ‘maroon’ under ‘red’) as explained in the next section.

Figure 6. Sample images of the 9 color categories used in training the color classification network.

4 A FUZZY INFERENCE SYSTEM

This section details the high-level fuzzy inference system used to match a robot’s state of mind to the object(s) mentioned in a human request. Both an object’s shape, texture, and color properties, and spatial relations between objects are represented using fuzzy sets.

4.1 Fuzzy object features

As discussed in Section 3 three different neural networks are trained to recognize object properties. At the output layer of these networks we use a softmax classifier [4]. Softmax is an exponential averaging function (Equation 6) with values between 0 and 1. Here i represents one class and j represents all other classes.

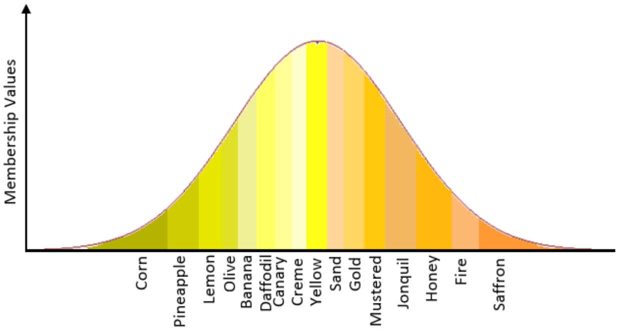

Figure 7. Fuzzy set representation of the ‘yellow’ color category.

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{6}$$

Values of the softmax function can be directly mapped to fuzzy membership values, which makes it straightforward to define fuzzy membership functions for the different object features (Equation 7).

$$\text{U}(\mu_x) = \text{Softmax}(x_i) \tag{7}$$

Here, U is the fuzzy membership function for ‘color’, ‘shape’ and ‘texture’, respectively, while μ_x denotes the membership value of the property category x belonging to the fuzzy set U for a given object.

For each color category (e.g., ‘yellow’ or ‘blue’) we specifically define aliases that may be used in verbal requests instead of the category names, for example ‘gold’ instead of ‘yellow.’ These aliases are mapped to the respective color categories again using fuzzy membership functions; Figure 7 illustrates this for ‘yellow.’

4.2 Fuzzy spatial relations

In the object segmentation step (Section 3.1), we determine the coordinate (x, y, z) of each object’s center point [19]. With these coordinates we can compute spatial relations between the different objects. However, the applicability of their corresponding linguistic labels are inherently fuzzy, i.e., the degree to which a certain relational term is appropriate may vary. And different people may use relations differently. For example, all other objects depicted in Figure 3 are to the right of the bowl, but for some using the relation ‘right of’ seems more appropriate than for others. This varying degree of applicability can be easily represented using fuzzy membership functions.

We distinguish three kinds of spatial relations: 1) topological relations, 2) directional relations, 3) distance relations. Topological relations are defined with respect to the image plane rather than to other objects. Directional relations are further divided into 2D and 3D relations and rely on the angle formed between two objects. Distance relations are based on the distance between objects in the image plane.

1. **Grounding topological relations:** Topological relations are image-centric. We divide the image in 9 equal-sized regions ‘top left’, ‘top right’, ‘top center’, ‘middle left”, ‘middle right”, ‘middle center’, ‘bottom left’, ‘bottom right’, ‘bottom center.’ These regions are crisp for now; there is no fuzzy membership defined for them. Each object is assigned to one of these regions based on largest overlap with its bounding box.

2. **Grounding directional relations:** 2D directional relations do not rely on depth information (e.g., 'left', 'next to') whereas 3D relations use depth information (e.g', 'in front of', 'on').

   (a) **Grounding 2D relations:** we consider 'left', 'right' and 'next to' as 2D relations. We use Equation 8 to estimate them; 'next to' is taken to be a generalization (or unspecified version) of the relations 'left' and 'right.' There is no specific membership function for it. The two fuzzy sets 'left' and 'right' are defined based on the difference in x coordinate between the referring object and the reference object. If the difference is negative, then we assign the referring object to the set 'right', else 'left.'

$$\mu_{(U)}(y) = \begin{cases} 0, & \text{if } y <= a \\ \frac{y-a}{m-a}, & a < y <= m \\ \frac{b-y}{b-m}, & m < y < b \\ 0, & y >= b \end{cases} \qquad (8)$$

   (b) **Grounding 3D relations:** We use 'in front', 'behind', 'in', and 'on' as 3D directional relations. These relations are dependent on depth perception; accordingly we use the robot's depth sensor to estimate these relations.

   Grounding 'in front' and 'behind' relations: These two relations express opposites similar to 'left' and 'right.' We use the distance between objects to estimate whether an object is 'in front' or 'behind' the other one. As objects consist of a set of points, just using one coordinate would not be adequate. We select 50 random points of each object and calculate distances between them. The average distance is then considered the actual distance between the two objects. If the distance is positive then the referring object is 'behind' the reference object else 'in front' (e.g., the banana is 'in front' the cup in Figure 3). In other words, computing these relations is similar to computing 'left' and 'right', just using the y axis instead.

   Grounding 'in' and 'on' relation For 'in' and 'on' to hold, the reference object should (partly) cover the referring object. This coverage can be expressed by a fuzzy membership value stated in Equation 9. Here O_j represents the referring object and O_i the reference object. This relation holds if O_j's area is less than O_i's otherwise we treat O_j as reference object and O_i as referring (similarly for object height). We use a triangular membership function to represent membership values.

$$\mu_{ratio} = \frac{O_j(area)}{O_i(area)} * \frac{O_j(height)}{O_i(height)} \qquad (9)$$

3. **Grounding distance relations:** We use 'near', 'close' and 'between' as distance relations. We take 'close' to express shorter distances between two objects than 'near', while both express nearness of course. The relation 'between' is used to cover cases where an object's location is described using two (or more) reference objects. Figure 8 depicts the membership distribution of 'close' and 'near.' The inner dark-blue ring represents 'close'; the outer blue ring 'near.' To determine membership values we map the image to a 1:1 coordinate system and then use thresholds to define 'close' (distance < 0.2) and 'near' (distance between 0.2 and 0.5). We again use a triangular membership function.

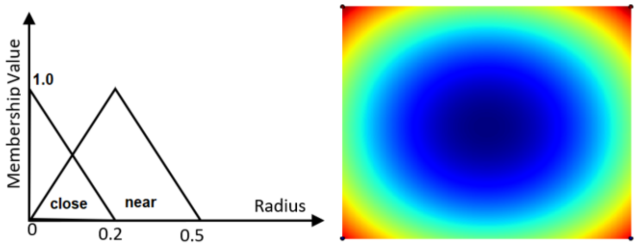

Figure 8. Visualization of the 'close' and 'near' membership functions relative to the reference object's center.

Having determined fuzzy membership functions of all object attributes, we have defined a fuzzy inference system for the robot's state of mind. For example, the fuzzy set representations for the two cups 'cup-1' and 'cup-2' of Figure 3 are as follows:
Cup-1{**cylinder, red, smooth**}
$Right = \{0.4/cup\}$
$Behind = \{0.8/orange\}$
Cup-2{**cylinder, black, smooth**}
$Left = \{0.2/orange, 0.6/cup\}$
$Right = \{0.6/orange\}$
$Behind = \{0.9/banana\}$
Given a verbal request "Give me the cup behind the banana," we select all instances of the category cup and their fuzzy sets. We then check whether the reference object (the banana) is present in these sets. If it appears in the sets of multiple objects, preference is given to the object associated with the highest membership value. Here, the inference system would select 'cup-2' (see Section 5).

4.3 Constructing the knowledge graph

Based on the fuzzy membership functions for all object attributes (shape, color, texture, spatial relations), we construct a knowledge graph that reflects the robot's 'state of mind' of the perceived scene. This graph is dynamic as it needs to capture changes induced by changed perception (e.g., due to robot movement). The vertices of the graph represent the detected objects. Object features are represented by edges pointing to the object itself, while spatial relations are represented by edges between two objects. In case of multiple occurrences of objects of the same type, we name these objects object-1, object-2, ', object-n (e.g., cup-1, cup-2, cup-3), which results in a unique identifier for each object (vertex). Since we use a fuzzy representation, multiple values may hold for object attributes to varying degrees. Only the value with highest membership is represented directly in the graph, i.e., as an edge. All other values are stored in a secondary look-up table, which may be queried if some attribute in a human request does not match with the graph representation.

Representing just the 'best fit' in the graph as a direct edge is beneficial in terms of computational complexity. We use graph matching to map a human verbal statement to the robot's state of mind (see Section 5). Reducing the number of edges in the graph drastically reduces the effort in graph matching. A complete graph for a given scene has complexity O(n(n-1)), where n is the number of objects in the scene. In our proposed system graph complexity is as follows:

$$Complexity = \begin{cases} O(n(n-1), & 1 < n < 5 \\ O(4n), & 4 < n \end{cases} \qquad (10)$$

In other words, with more than five objects present in a scene complexity is linear. Additionally, we believe that representing only the attributes with the highest applicability reflects those statements humans are most likely to make about the objects in the scene.

5 NATURAL LANGUAGE INTERACTION

As stated in the introduction, our work is set in the context of human-robot interaction on table-top settings. A Pepper robot is to receive a user's natural language request for an object on the table, and to identify this object. The robot can also generate natural language clarification requests. We use Google's speech engine[3] for converting spoken language into text. We then apply language parsing to extract the referring and reference objects, and any of their properties and spatial relations included in the verbal description; see [1] for more details on language parsing. Figure 9 shows an example parse tree of the statement "Give me the black cup behind the banana." Here, 'cup' is the referring object and 'banana' the reference object. A color property ('black') and a spatial relation ('behind') are mentioned. While there is explicit handling of color name aliases (Section 3.4), 'aliases' for commands (or intents) of the user are hand-coded for now; there is no intent recognition in the system at this point.

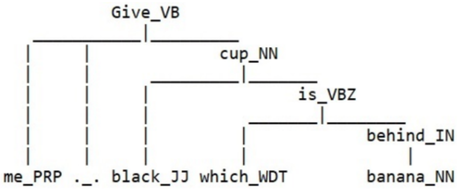

Figure 9. Parse tree of the request "Give me the cup behind the banana."

From this, we construct a scene representation of the human request that is matched to the previously constructed knowledge graph. The representation corresponding to above statement is as follows:

```
Language Parsing
{
"rel_obj": "cup",
"ref_obj": "banana",
"rel_pro": {
  "colour": "black",
  "shape": "nil",
  "texture": "nil"
 },
"ref_pro": {
  "colour": "nil",
  "shape": "nil",
  "texture": "nil"
 }
}
```

Parts of the robot's 'state of mind' for the scene in Figure 3 is as follows (this is just for illustration purposes and, hence, incomplete):

```
Visual Grounding
{
"apple-1": [
  {
    "colour": "red",
    "shape": "circle",
    "texture": "smooth",
    "left": [’bowl’],
    "right": [’cup-2’,’orange’],
    "infront": [’cup-1’],
    "behind": [’banana’],
  },
],
"cup-2": [
  {
    "colour": "black",
    "shape": "cylinder",
    "texture": "smooth",
    "left": [’apple’,’cup-1’],
    "right": [’orange’],
    "behind": [’banana’],
  },
]
...
```

To match natural language requests to the visual grounding, we search the referring object—here 'cup'—in the knowledge graph, using object type and its properties. If the object type leads to an unambiguous match, e.g., there is only one cup, or if the object type in combination with its mentioned properties can be unambiguously identified in the visual scene, matching has been successful and any further spatial relations are ignored. Otherwise (e.g., if there is a second black cup in the setting), these relations are used for further disambiguation. The interaction scenario is further illustrated in Figure 10.[4]

The robot displays the identified objects on the tablet attached to its chest, which provides some feedback about what the robot actually understands. Users can also ask to receive a natural language scene description. Further, in case the robot is unable to match a human request to its knowledge graph, it can generate a clarification request based on object features and spatial relations. To that end we use a simple template-based approach (see below). It uses a set of question prefixes followed by those attributes most relevant for object disambiguation. All tags in the <> expressions are optional except for the question tag. For example, the robot may request the color of the intended object ("Could you tell me the color?"), or suggest the object it considers most likely ("Do you mean the cup behind the apple?").

Feedback question template
<question tag><color|shape|texture><rel_object>
<relation><color|shape|texture><ref_object>
<question tag>={'Do you mean the','Is this the', 'Could you tell me the', 'Is it'}
Example feedback statements
1. Do you mean the red cup ?
2. Do you mean the cup behind the apple ?
3. Is this the cup ?
4. Could you tell me the color ?
5. Could you tell me the relation to the orange ?
6. Is it circular ?
7. Is it left of the white bowl ?
8. Is it center left ?

[3] https://cloud.google.com/speech-to-text/

[4] see also the video at https://rb.gy/0ziwjm. This video is for illustrative purposes, it is not part of the evaluation.

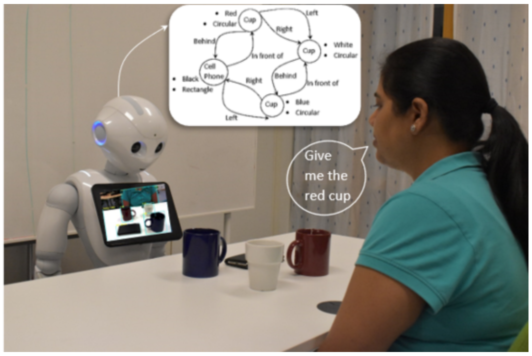

Figure 10. An example scene of interacting with a Pepper robot on a table-top setting, including a visualization of the robot's 'state of mind.'

6 EVALUATION

We have created a data set of table-top scenes, i.e., objects placed in some arrangement on a table. The data set consists of 128 images using 21 different object types. We manually labeled the 'shape', 'color', 'texture' and 'spatial relations' of these objects in order to have 'ground truth' for the training of the different classifiers and to be able to evaluate system performance. The distribution of these attributes are shown in Figure 11. We can see that the 'simple' directional relations ('left', 'right', 'in front of', 'behind') and distance relations ('close to', 'near') are used much more frequently than the topological (image-centric) relations. And while the different shape and texture categories are relatively evenly distributed, there is a large variation in color frequency in the data.

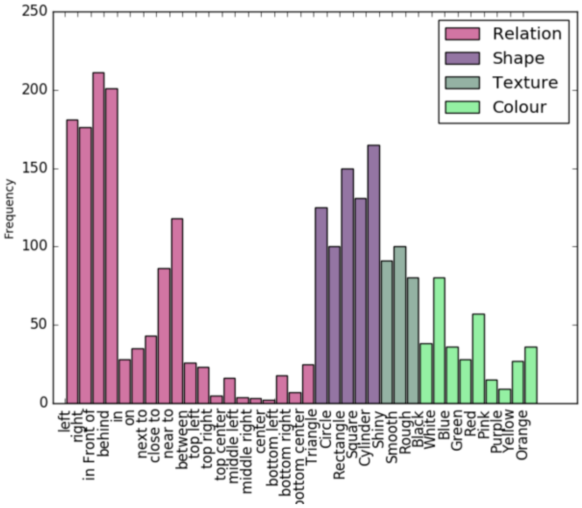

Figure 11. Object features, spatial relations and their frequency as they appear in the evaluation dataset.

This data set was used to train and test the ANNs for classifying the 'shape' and 'texture' properties; the network for the 'color' property was trained and tested on the Google images (Section 3.4). We evaluated different combinations of hidden layers and number of neurons for shape and texture networks. For shape, a configuration of 2 hidden layers with 10 neurons each results in a misclassification of only 4.8%. As the texture property vectors have a much higher dimensionality (R^{256}), more neurons (100) are needed to handle non-linearity. We also explored different parameter settings for LBP [15]. With a radius of 3 and 24 neighbors, and the adapted ANN we achieve a validation error of 5.6%. Training the 'color' network, we divide the data into training and validation using a 60:40 ratio and use a batch size of 256. This results in a classification error of 6.2%.

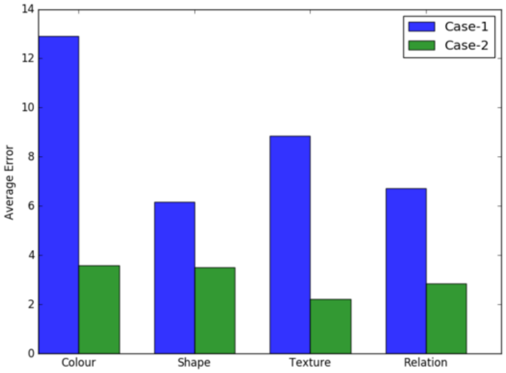

Figure 12. Average case-1 (blue) and case-2 (green) mismatches ('errors') between human description and the robot's visual grounding for attributes color, shape, texture, and spatial relation, over all 128 scenes (in percent).

Classifying color is more challenging as there is a high inter- and intra-class variance in the images. Non-linearity in the inherent color distribution further contributes to these challenges. Occlusions between objects provide challenges to the shape classifier. However, these challenges are (to a large part) captured by the fuzzy set representations. Neural networks use maximum likelihoods, which we take to be the best linguistic description for a given attribute. But in case this does not match with human understanding of this attribute, the fuzzy membership functions give access to other likely possibilities, which allows for a natural way of adapting these likelihoods.

As explained in Section 4, the knowledge graph only represents the attributes with highest membership value. For example, for the scene in Figure 3 the apple may be seen to be 'right' or 'behind' the bowl. However, 'right' has a higher membership value and, thus, gets represented in the graph. Still, all other attributes are stored in a secondary lookup table. That is, if the apple is described to be 'behind' the bowl by a user, we can still retrieve this relation. We treat the graph as case-1 matching and the lookup table as case-2 matching.

We evaluated the whole system from object recognition to the knowledge graph and the accompanying fuzzy inference system, using the labeled ground-truth data set to that end. Four different people (including one of the authors) generated example questions for the 128 scenes (see [1]). These questions were fed into the system and then compared how well this linguistic input matches with the knowledge graph, and look-up table, constructed from visual input. Figure 12 summarizes these results, showing average classification error over all categories of a given attribute. Case-1 matches record direct matches, for example, if the system extracted a 'square' shape and the object is also described as 'square' by the human labeler. If the human described the shape as, e.g., 'rectangular', we get a case-1 mismatch, but may still achieve a case-2 match, i.e., find 'rectangle' in the object's fuzzy shape function. As we can see, classification errors are generally low, with the highest being 12.9% case-1 color mismatches over the 128 scenes. Using fuzzy membership functions (case-2 matches) further reduces these low numbers of mismatches significantly. Hence, we conclude that our proposed combination of low-level neural network-based feature extraction and high-level fuzzy inference system captures human understanding of the table-top scenes, but also human imprecision and variation, very well.

7 CONCLUSIONS

The second order theory of mind suggests that if a human can predict a robot's state of mind, and if the robot in turn can reflect the human's state of mind in its representation of the world, and if this second order modeling is correct (concepts and relations match), then human and robot will understand each other correctly [17]. In this paper we aim to construct such a robot's state of mind for restricted scenarios—tabletop scenes of typical household objects. We presented a framework that combines low-level neural network-based visual processing (feature extraction) with a high-level fuzzy inference system that captures possible ambiguity and extraction errors to construct a knowledge graph that reflects the robot's visually grounded scene understanding. We detailed how this graph and the fuzzy inference system can be used in natural language interaction between human and robot. Our evaluation on a large variety of table-top scenes shows that the system is robust, i.e., very well captures human understanding of these scenes, but also human imprecision and ambiguity. Future work will extend evaluation to dynamic scenarios of human and robot actually interacting on table-top settings.

ACKNOWLEDGEMENTS

We thank Suna Bensch, Thomas Hellström and Ola Ringdahl for their help in labeling the data and general advice. We gratefully acknowledge funding by the Kempe Foundations.

REFERENCES

[1] Neha Baranwal, Avinash Kumar Singh, and Suna Bench, 'Extracting primary objects and spatial relations from sentences', in *11th International Conference on Agents and Artificial Intelligence, Prague, Czech Republic, 19-21 February 2019.*, (2019).

[2] P. Gärdenfors, *Conceptual Spaces: The Geometry of Thought*, MIT Press, Cambridge, MA, 2004.

[3] Dave Golland, Percy Liang, and Dan Klein, 'A game-theoretic approach to generating spatial descriptions', in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 410–419. Association for Computational Linguistics, (2010).

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016. http://www.deeplearningbook.org.

[5] Daniel H Grollman, Odest Chadwicke Jenkins, and Frank Wood, 'Discovering natural kinds of robot sensory experiences in unstructured environments', *Journal of field robotics*, **23**(11-12), 1077–1089, (2006).

[6] Sergio Guadarrama, Lorenzo Riano, Dave Golland, Daniel Go, Yangqing Jia, Dan Klein, Pieter Abbeel, Trevor Darrell, et al., 'Grounding spatial relations for human-robot interaction', in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1640–1647. IEEE, (2013).

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, 'Mask R-CNN', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, (2017).

[8] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata, 'Grounding visual explanations', in *European Conference on Computer Vision*, pp. 269–286. Springer, (2018).

[9] Céline Hudelot, Jamal Atif, and Isabelle Bloch, 'Fuzzy spatial relation ontology for image interpretation', *Fuzzy Sets and Systems*, **159**(15), 1929–1951, (2008).

[10] James M Keller and Xiaomei Wang, 'A fuzzy rule-based approach to scene description involving spatial relationships', *Computer Vision and Image Understanding*, **80**(1), 21–41, (2000).

[11] Josef Kittler, Mikhail Shevchenko, and David Windridge, 'Visual bootstrapping for unsupervised symbol grounding', in *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 1037–1046. Springer, (2006).

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, 'Imagenet classification with deep convolutional neural networks', in *Advances in Neural Information Processing Systems*, pp. 1097–1105, (2012).

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, 'Microsoft CoCo: Common objects in context', in *European Conference on Computer Vision*, pp. 740–755. Springer, (2014).

[14] Vivien Mast, Zoe Falomir, and Diedrich Wolter, 'Probabilistic reference and grounding with PRAGR for dialogues with robots', *Journal of Experimental & Theoretical Artificial Intelligence*, **28**(5), 889–911, (2016).

[15] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää, 'Multiresolution gray-scale and rotation invariant texture classification with local binary patterns', *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7), 971–987, (2002).

[16] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al., 'Learning representations by back-propagating errors', *Cognitive Modeling*, **5**(3), 1, (1988).

[17] Brian Scassellati, 'Theory of mind for a humanoid robot', *Autonomous Robots*, **12**(1), 13–24, (2002).

[18] Mohit Shridhar and David Hsu, 'Interactive visual grounding of referring expressions for human-robot interaction', in *Proceedings of Robotics: Science and Systems*, (2018).

[19] Avinash Kumar Singh, Neha Baranwal, and Kai-Florian Richter, 'An empirical review of calibration techniques for the Pepper humanoid robots RGB and depth camera', in *Proceedings of SAI Intelligent Systems Conference*, pp. 1026–1038. Springer, (2019).

[20] Jiacheng Tan, Zhaojie Ju, and Honghai Liu, 'Grounding spatial relations in natural language by fuzzy representation for human-robot interaction', in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1743–1750. IEEE, (2014).

[21] Michal Vavrečka, Igor Farkaš, and Lenka Lhotská, 'Bio-inspired model of spatial cognition', in *International Conference on Neural Information Processing*, pp. 443–450. Springer, (2011).