Chapter 5
A Perceptual System for Language Game Experiments

Michael Spranger[1,2], Martin Loetzsch[3] and Luc Steels[1,4]

This paper is the authors' draft and has been officially published as: M. Spranger, M. Loetzsch and L. Steels (2012). A Perceptual System for Language Game Experiments. In Luc Steels and Manfred Hild (Eds.), *Language Grounding in Robot*, 89-110. New York: Springer.

Abstract This chapter describes key aspects of a visual perception system as a key component for language game experiments on physical robots. The vision system is responsible for segmenting the continuous flow of incoming visual stimuli into segments and computing a variety of features for each segment. This happens by a combination of bottom-up way processing that work on the incoming signal and top-down processing based on expectations about what was seen before or objects stored in memory. This chapter consists of two parts. The first one is concerned with extracting and maintaining world models about spatial scenes, without any prior knowledge of the possible objects involved. The second part deals with the recognition of gestures and actions which establish the joint attention and pragmatic feedback that is an important aspect of language games. experiments.

Key words: visual perception, humanoid robots, world models

5.1 Introduction

The field of autonomous robots has made huge progress the past decade, so that we now have robots, even with humanoid shapes, that perform stable locomotion, navigation, or object manipulation using rich body sensing, real-time vision and sophisticated behavior control (e.g. Fujita et al, 2003; Ishiguro, 2006; Kanda et al, 2007). Progress is due partly to great advances in hardware technologies (batteries, motors, sensors, processors, memory), partly to many new algorithms that perform specific subtasks much better, and partly to the development of architectures for combining computation from many sources (vision, proprioception, object model-

[1]Sony Computer Science Laboratory Paris, e-mail: spranger@csl.sony.fr
[2]Systems Technology Laboratory, Sony Corporation, Tokyo
[3]AI Lab, Vrije Universiteit Brussel, Brussels
[4]ICREA Institute for Evolutionary Biology (UPF-CSIC), Barcelona

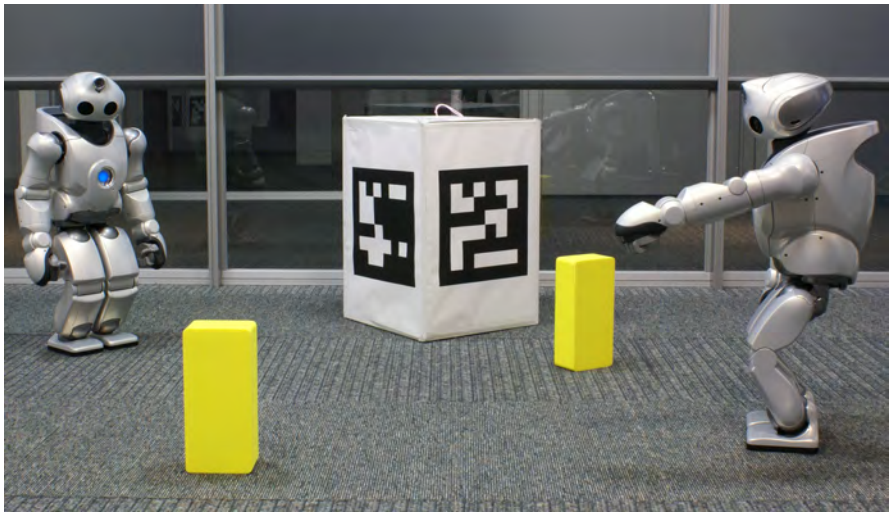

Fig. 5.1 Humanoid robots play language games about physical objects in their shared environment. The big box is enhanced with visual markers to give it a front and back. Robots need to segment the objects in this environment and collect features about them that can be the basis for conceptualization and language.

ing, planning) into effective real-time behavior (e.g. Brooks and Arkin, 2007; Cruse et al, 2007; Pfeifer et al, 2007).

On the other hand, cognition and intelligence has been progressing less rapidly. Autonomous robots today cannot yet be said to have the kind of rich conceptualizations of the world that we find in human cognition. A prerequisite for such conceptualizations is that the robot builds a rich world model using all its sensory and motor capabilities. The world model consists of segmentations of sensory-motor streams and features for these segments. The features are still in the continuous domain, i.e. segments are not categorized in terms of types of objects or events. Thus an object is not yet categorized as red or green but as having a specific range of values along the three color dimensions (yellow-blue, red-green, brightness). This world model is then used by the conceptualization processes described in a later chapter by Spranger et al (2012) to come up with the semantic structure that is expressed in language.

The present chapter describes a perceptual system that mediates between raw sensory-motor data and conceptualization processes. It has been developed in such a way that it is portable between different platforms and can be extended in a modular fashion. The system is operational on several robotic platforms, including the MYON robot, and has been used in language game experiments with MYON robots discussed later. However in order to focus the presentation, we use a particular em-

bodiment, namely the Sony non-commercial humanoid robots shown in Fig. 5.1 (Fujita et al, 2003), for which the system was originally developed.

The Sony non-commercial humanoid robots (Fujita et al, 2003, see Fig. 5.1) used in this experiment are about 60 cm high, weigh approximately 7 kg and have 38 degrees of freedom (4 in the head, 2 in the body, 5×2 in the arms, 6×2 in the legs and 5×2 in the fingers). The main sensors are three CCD cameras in the head, of which we used here only one. The camera delivers up to 30 images per second, has an opening angle of about 120° and a resolution of 176×144 pixels. It uses the *YCrCb* color space (*Y*: luma or brightness, *Cr*: chroma red and *Cb*: chroma blue) with 8 bits per channel. Furthermore, the robots have three accelerometers and gyro sensors in the trunk and one accelerometer in each foot. The feet are equipped with force feedback sensors to detect ground contact. The batteries have enough capacity for about an hour of autonomous operation.

The remainder of the paper is divided in two sections. Section 5.2 below focuses on the vision system itself which is described in much more detail in Spranger (2008). Section 5.3 introduces additional perceptual skills needed for language games, in particular for establishing joint attention and recognizing pointing gestures.

5.2 Visual Object Recognition and Tracking

The environment of the robots consists of a variety of physical objects such as toys, cones, barrels and cuboids that are initially unknown to the robots. Objects are frequently added to the scene and removed again. In addition, objects are moved within a scene and their appearance may alter. For example the red block in Fig. 5.2a) is standing up in the beginning and then put down, changing the perception of the object from being high and thin to low and broad. In addition, perceiving objects is made difficult by partial occlusions and other interfering factors such as human experimenters manipulating the objects in front of the robots.

A prerequisite for building the internal structures needed for communicating about objects is that the robots have mechanisms for constructing perceptual representations of the objects in their immediate surroundings from the raw sensations streaming from the robots' sensors. Constructing such representations involves three sub-systems:

- First, low-level vision routines process raw camera images to yield basic *percepts* – connected regions that differ from the background of the environment. Fig. 5.2b) gives an example and the mechanisms involved are explained in Section 5.2.1 below.
- Second, these foreground regions are tracked in subsequent camera images. In order to do so, the vision system needs to establish a correspondence between an internal *object model* and the image regions that refer to the same physical object, a process known in robotics as *anchoring* (Coradeschi and Saffiotti, 2003). For example as illustrated in Fig. 5.2d), the changing raw sensations for the red block

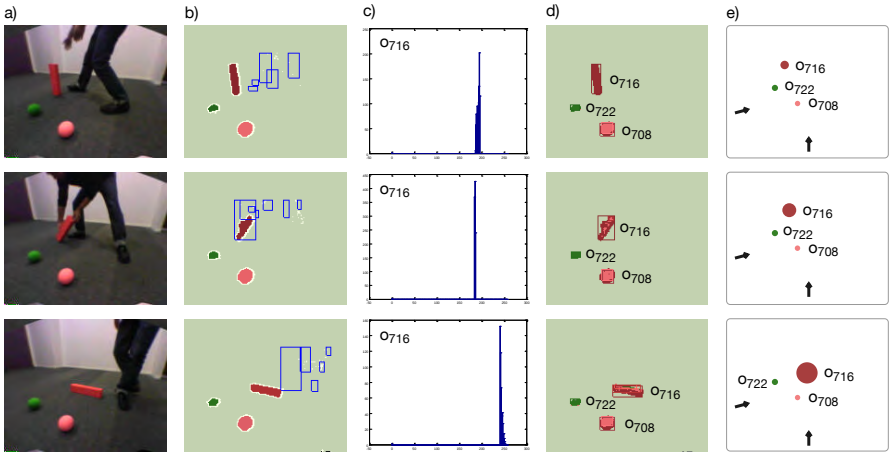

Fig. 5.2 Image processing steps for three subsequent points in time. a) Source images provided by the camera of the robot. b) Foreground/ background classification and motion detection (blue rectangles). Foreground regions are then associated to existing object models or become seeds for new object representations. c) The changing histogram of the green-red channel for object o_{716} is d) used to track o_{716} in space and time and thus to create a persistent model of the object. e) Knowing the offset and orientation of the camera relative to the body, the robots are able to estimate the position and size of objects in the world. Black arrows denote the positions of the two robots perceiving the scene (see Section 5.3.3).

in Fig. 5.2a) are continously connected to the same *anchor* o_{716}. We used *Kalman Filters* for maintaining such persistent object models (Section 5.2.2).

- Third, when needed in communicative interactions, the vision system encodes a set of visual properties about each object model. In this particular setup these properties are the object's position in a robot egocentric reference system, an estimated width and height, and color information, as shown in Fig. 5.2e). This process is discussed further in Section 5.2.3.

5.2.1 Detecting Foreground Regions in Images

The robots do not know in advance what kind of objects to expect in their environment. Thus, the assumption is made that everything that was not in the environment before is considered to be a potential object. The system, therefore, gathers statistical information about the environment's background in a calibration phase and those image regions that sufficiently differ from the background are treated as candidates for object models. For generating a statistical model of the scene background, the robots observe the experiment space without objects for some time and perceive

a series of calibration images such as in Fig. 5.3a). For all three color channels $c \in \{Y, Cr, Cb\}$ the mean $\mu_{c,\mathbf{p}}$ and variance $\sigma^2_{c,\mathbf{p}}$ of the image intensities at every image pixel $\mathbf{p}$ are computed over all calibration images.

After the calibration phase the robots are presented with objects, resulting in raw camera images such as in Fig. 5.3b). The generated background statistics are used to classify all image pixels as being foreground or background. A pixel is considered foreground when the difference between the image intensity $i_c(\mathbf{p})$ and the mean of that pixel is bigger than the pixel's standard deviation ($\mid i_c(\mathbf{p}) - \mu_{c,\mathbf{p}} \mid > \sigma_{c,\mathbf{p}}$) for one of the color channels $c \in \{Y, Cr, Cb\}$. As a result, a binary image as shown in Fig. 5.3c) is generated with all foreground pixels having the value of 1 and all others 0.

This binary image is further noise-reduced using standard image operators (dilatation, erosion, see for example Soille (2003)) as illustrated in Fig. 5.3d). First, noise is removed through applying a 3×3 erosion operator. Second, the change in size of regions caused by the erosion operator is compensated by applying a 3×3 dilation operator. Then a segmentation algorithm scans the filtered image and computes for all connected foreground pixels a surrounding polygon, the bounding box, and color histograms of the pixels contained in the region (for each color channel, from the original image).

Color histograms M^c represent frequencies of image intensities on the color channel c, computed either over complete images or parts of them in the case of foreground regions. The whole range of intensities is divided into m bins $k \in \{1, \ldots, m\}$ of equal size. The number of pixels that have intensities falling into each bin $M^c(k)$ is counted using a function $h(i_c(\mathbf{p}))$ that assigns the intensity i_c of a pixel $\mathbf{p}$ to a bin k. Normalized histograms $\hat{M}^c(k)$ are computed from such histograms by dividing each frequency $M^c(k)$ by the number of pixels sampled, resulting in a representation where the sum of all $\hat{M}^c(k)$ for $k \in \{1, \ldots, m\}$ is equal to 1, allowing to interpret $\hat{M}(h(i_c(\mathbf{p})))$ as the probability of an image intensity to occur in an image (or a sub-region). Fig. 5.3e) shows the estimated bounding boxes and average colors extracted from the regions.

Objects frequently occlude each other, due to particular spatial placement, but also when moved around in the scene. For example the green cube is partly overlapping the blue cuboid in the right bottom of Fig. 5.3b) and thus the segmentation algorithm creates only one foreground region for both objects. Provided that there is an established object model (see next Section 5.2.2) for at least one of the objects, it is possible to further divide such regions. Each pixel in a foreground region is assigned to the most similar color model of previously perceived objects as shown in Fig. 5.3f). Given the normalized color histograms M^c_I of all pixels in the current image I and $M^c_1, \ldots, M^c_n$ of the n previously established object models, the likelihood p_j of a pixel $\mathbf{p}$ in a foreground region to belong to a color model j can be calculated:

$$p_j(\mathbf{p}) = M^Y_j(h(i_Y(\mathbf{p}))) \cdot M^{Cr}_j(h(i_{Cr}(\mathbf{p}))) \cdot M^{Cb}_j(h(i_{Cb}(\mathbf{p})))$$

Based on this probability, each pixel is either classified to belong to the model j with the highest likelihood $\text{class}(\mathbf{p}) = \arg\max_{j=1..n}(p_i(\mathbf{p}))$ or, when the highest p_j is smaller than a threshold t or when no previous model exists, to a "no model"

class. Classified pixels are again segmented into connected regions. As shown in Fig. 5.3g) and h), the initially connected foreground region for the blue and green objects in the right bottom of the image could be divided into separate regions due to the use of previous color models.

The resulting subdivided foreground regions are called *percepts*. They represent the result of the low-level image processing mechanisms acting separately on each image without incorporating past knowledge (except for the color information of previous objects). A percept P is defined as $P := \langle x_P, y_P, w_P, h_P, M_P^Y, M_P^{Cr}, M_P^{Cb}, n_P \rangle$ with x_P, y_P describing the center of the percepts bounding rectangle in image coordinates, w_P and h_P the width and height of the bounding rectangle in pixels, M_P^Y, M_P^{Cr} and M_P^{Cb} the normalized histograms for the three color channels and n_P the number of pixels contained in the region.

In order to improve the tracking algorithm described in the next Section, we also implemented a component for identifying regions in the image where motion has occured. Image intensities $i_{c,t}(\mathbf{p})$ at time t are compared to those of images taken at time $t-1$. A pixel $\mathbf{p}$ is classified as subject of motion when the difference is bigger than the standard deviation $\sigma_{c,\mathbf{p}}$ of this pixel's intensities calculated during the calibration phase ($\mid i_{c,t}(\mathbf{p}) - i_{c,t-1}(\mathbf{p}) \mid > \sigma_{c,\mathbf{p}}$) for one of the color channels $c \in \{Y, Cr, Cb\}$. The resulting classification image is noise-reduced and segmented into regions of motion as shown in Fig. 5.2b). This information is used to loosen the parameters for the association of percepts to object models. If there is motion in a particular region of the image, then object models are allowed to move and change color more drastically than if there is no motion.

5.2.2 Maintaining Persistent Object Models

For maintaining a set of stable and persistent models of the objects in their environment, the robots have to associate the percepts extracted from each raw image to existing object models. Furthermore, they have to create new models when new objects enter the scene and eventually delete some models when objects disappear. This task is difficult because objects can move and the detection of regions through foreground/background separation is noisy and unreliable. Extracted properties such as size or position may highly vary from image to image and it can happen that objects are only detected in some of the images streaming from the camera.

The internal object model O_t of an object at time step t (whenever a new camera image is processed) is defined as $O_t := \langle id_O, s_{O,t}, \Sigma_{O,t}, M_{O,t}^Y, M_{O,t}^{Cr}, M_{O,t}^{Cb} \rangle$, with id_O being an unique id serving as an anchor for the object, $s_{O,t}$ a state vector capturing spatial properties, $\Sigma_{O,t}$ the 8×8 state covariance matrix and $M_{O,t}^Y$, $M_{O,t}^{Cr}$ and $M_{O,t}^{Cb}$ normalized color histograms. A state vector s is defined as $s_{O,t} := \left(x_{O,t}\ y_{O,t}\ w_{O,t}\ h_{O,t}\ \dot{x}_{O,t}\ \dot{y}_{O,t}\ \dot{w}_{O,t}\ \dot{h}_{O,t}\right)^T$, with $x_{O,t}, y_{O,t}$ describing the center of the

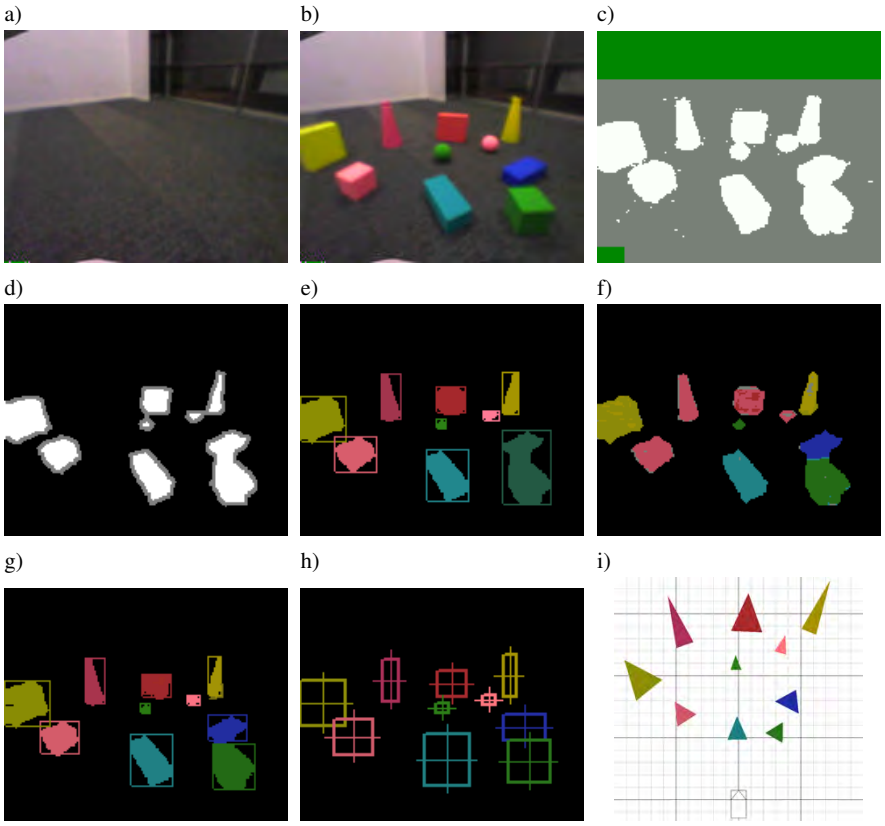

Fig. 5.3 From foreground regions to object models. a) A raw camera image taken during the calibration phase. b) A camera image of a scene containing objects. c) The result of foreground/background classification. White pixels are foreground, green pixels were not classified. d) The noise-reduced classification image. e) The segmented foreground regions drawn in their average color and with bounding boxes. Note that the partially overlapping blue and green blocks in the right bottom of the original image are segmented into the same foreground region. f) Classification of foreground pixels using existing color models. Pixels are drawn in the average color of the most similar object model. g) Bounding boxes and average colors of the segmented classification image. Note that the use of previous color models helped to generate separate percepts for the blue and green blocks at the right bottom of the image. h) Kalman filtered object models. The state bounding boxes are drawn in the average color of the model. i) Computation of position and size in a robot-egocentric reference system. The width and height of objects is indicated by the width and height of the triangles.

object in the image, $w_{O,t}$ and $h_{O,t}$ the object's width and the height in pixels and $\dot{x}_{O,t}$, $\dot{y}_{O,t}$, $\dot{w}_{O,t}$ and $\dot{h}_{O,t}$ the change variables (speed of change in position and size).

We use Kalman Filters (Kalman, 1960) to model the spatial component $s_{O,t}$ of object models. In every time step t all Kalman Filter states $s_{O,t-1}$ and $\Sigma_{O,t-1}$ of the last time step $t-1$ are used to *predict* a new a priori state $\bar{s}_{O,t}$ and a state covariance matrix $\overline{\Sigma}_{O,t}$ given the 8×8 state transition matrix A and the process noise covariance matrix Q:

$$\begin{aligned} \bar{s}_{O,t} &:= A s_{O,t-1} \\ \overline{\Sigma}_{O,t} &:= A \Sigma_{O,t-1} A^T + Q \end{aligned}$$

We found it sufficient to use a constant state transition matrix A, which predicts every dimension via its change variable and a constant noise covariance matrix $Q = 1^{-5} \cdot I_8$.

Next attempts are made to associate percepts to existing models. Since the position, dimension and color of objects change over time, no a priori known invariant properties of objects allow to decide which percept belongs to which model. Instead, a similarity score $\hat{s}$ based on position and color is used. The score reflects a set of assumptions and heuristics, which are based on intuitive notions of how objects behave, so that experimenters can change the scene, without having to adjust to particular properties of the vision system. First it is assumed that an object can not randomly jump in the image or disappear at one point in space and appear at another. Consequently, a spatial similarity $\hat{s}_{euclid}$ can be defined using the Euclidean distance between the center of a percept P and the predicted position $\bar{x}_{O,t}, \bar{y}_{O,t}$ of a model O

$$\hat{s}_{euclid}(P,O) := 1 - \frac{\sqrt{(x_P - \bar{x}_{O,t})^2 + (y_P - \bar{y}_{O,t})^2}}{l}$$

with l being the length of the image diagonal in pixels. The result of $\hat{s}_{euclid}$ is 1 when the two points are identical and 0 when they are in opposite corners of the image. Since objects are assumed to move in a predictable fashion, a threshold t_{space} restricts the radius around a model in which percepts are associated – the spatial association score $\hat{s}_{space}$ equals to $\hat{s}_{euclid}$ when it is bigger than t_{space} and 0 otherwise. Second, it is assumed that objects do not change their color in a random fashion. An object's color histogram that has a very high value in a certain bin will not have a zero value in that bin in the next image. Percepts and object models can thus be compared using a color similarity $\hat{s}_{color}$. It is based on the Bhattacharyya coefficient *BC* (Bhattacharyya, 1943; Aherne et al, 1998) that is used as a similarity measure between two normalized histograms M and M':

$$BC(M,M') := \sum_{k=1}^{m} \sqrt{M(k) \cdot M'(k)}$$

Using the color histograms M_P^c of a percept P and the histograms $M_{O,t-1}^c$ of a previous model O, a similarity measure combining all three color channels is defined as:

$$\hat{s}_{Bhatt}(P,O) := \prod_{c\in\{Y,Cr,Cb\}} BC(M_P^c, M_{O,t-1}^c)$$

The association score $\hat{s}_{color}(P,O)$ then yields the result from the above measure when it is bigger than a threshold t_{color} or 0 otherwise. In order to allow more rapid changes in space and color when objects move, the two association thresholds t_{space} and t_{color} are loosened when motion has been detected within the area spawned by a state.

The overall similarity score between a particular percept and an existing object model is then defined as:

$$\hat{s}(P,O) = \hat{s}_{space}(P,O) \cdot \hat{s}_{color}(P,O)$$

Each percept is associated with the internal state that has the highest association non-zero score $\hat{s}$ with respect to that percept. If no such state exists (when either the spatial or color similarity is below the threshold), then the percept is stored in a list of unassociated percepts.

The Kalman Filter states are *updated* given the associated percepts, which are beforehand combined into a single percept. Percepts are combined by computing a bounding polygon and a histogram representing the color frequency in the combined region. Using the predicted a priori state vector $\bar{s}_{O,t}$ and state covariance $\overline{\Sigma}_{O,t}$ as well as the spatial components p of the combined percept $p := \begin{pmatrix} x_P & y_P & w_P & h_P \end{pmatrix}^T$, the a posteriori state s_t and the a posteriori state covariance matrix $\Sigma_{O,t}$ are computed

$$\begin{aligned} K_{O,t} &= \overline{\Sigma}_{O,t} H^T H \overline{\Sigma}_{O,t} H^T + R \\ s_{O,t} &= \bar{s}_{O,t} + K_{O,t}(p - H\bar{s}_{O,t}) \\ \Sigma_{O,t} &= (I - K_{O,t}H)\overline{\Sigma}_t \end{aligned}$$

with R as the constant 4×4 measurement covariance matrix (with $R = 1^{-1} \cdot I_4$) and H a constant 8×4 matrix relating the measurement space and the state space (with $h_{i,j} = 1$ for all $i = j$ and 0 for all others). In principle H and R are allowed to change over time, but the above estimates resulted in sufficient tracking performance. Additionally, the color histograms of a state S are updated using

$$M_{O,t}^c(k) := (1-\alpha) M_{O,t-1}^c(k) + \alpha M_P^c(k)$$

for all color channels $c \in \{Y,Cr,Cb\}$, all histogram bins $k \in \{1,\ldots,m\}$ and with $\alpha \in [0,1]$ being the influence of the combined percept.

New object models are created from unassociated percepts. All unassociated percepts lying in the same foreground region are combined and used as a seed for a new model which is assigned a new unique ID. In order to avoid creating models from percepts generated for body parts of the experimenter, new models are only created when no motion was detected. Models that have not been associated with percepts for some time are deleted. This mainly happens when objects disappear from the scene and consequently no percepts are associated with them.

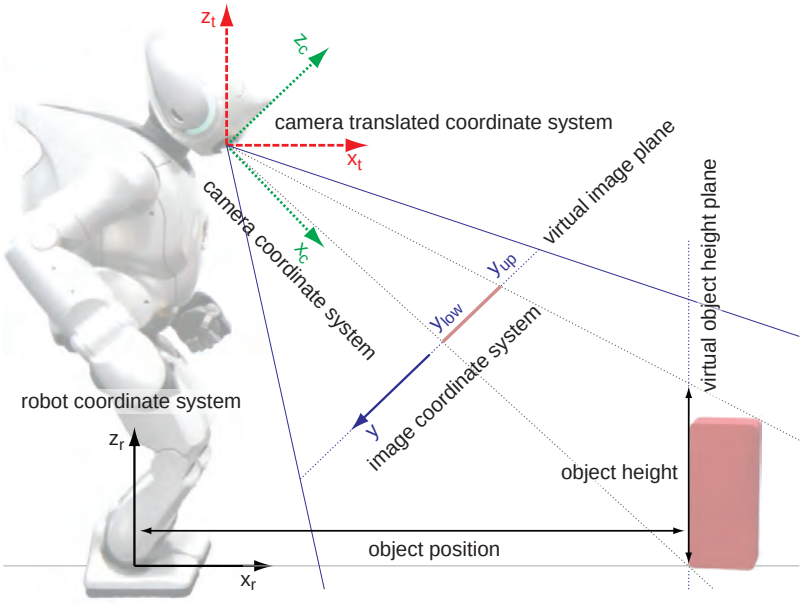

Fig. 5.4 Computation of object positions on the ground plane, size estimation and the involved coordinate systems. Note that all systems except the image coordinate system are three dimensional.

5.2.3 Computing Object Features

From each object model, a set of seven features such as color, position and size are extracted. We call these feature vectors *sensory experiences*.

The two robots can perceive the environment from arbitrary angles, which makes the position and size of objects in the camera image bad features for communicating about objects. For example the width of an object in the image depends on how far the object is away from the robot and is thus not at all shared by the robots. In order to be independent from how objects are projected onto camera images, spatial features are computed in an egocentric coordinate system relative to the robot. However, without the use of stereo vision or a priori known object sizes, positions can not be determined solely from camera images. But given the reasonable assumption that objects are located on the ground, they can be calculated by geometrically projecting image pixels onto the ground plane using the offset and rotation of the camera relative to the robot as shown in Fig. 5.4. The egocentric robot coordinate system originates between the two feet of the robot, the z axis is perpendicular to the ground and the x axis runs along the sagittal and the y axis along the coronal plane. First, a virtual image projection plane orthogonal to the optical axis of the camera is used to relate image pixels in the two-dimensional image coordinate system to

the three-dimensional camera coordinate system (which has its origin in the optical center of the camera, with the x axis running along the optical axis and the y and z axis being parallel to the virtual image plane). Given the camera resolution height and width r_w and r_h (in pixels) as well as the horizontal and vertical camera opening angle ϕ_v and ϕ_h, the x_i and y_i coordinates of an image pixel can be transformed into a vector $\mathbf{v}_c$ in the camera coordinate system

$$\mathbf{v}_c = \begin{pmatrix} 1 \\ -\frac{x_i}{r_h} \cdot \tan\frac{\phi_h}{2} \\ \frac{y_i}{r_v} \cdot \tan\frac{\phi_v}{2} \end{pmatrix}$$

that "points" to the pixel on the virtual projection plane. Given the orientation of the camera relative to the robot represented by the 3×3 rotation matrix R_c, a vector $\mathbf{v}_c$ can be rotated into a vector $\mathbf{v}_t$ in the camera translated coordinate system (which originates in the center of the camera, with the axes being parallel to the robot coordinate system) with $\mathbf{v}_t = R_c \cdot \mathbf{v}_c$. Furthermore, given the offset from the origin of the robot coordinate system to the center of the camera $\mathbf{t}_c$, the position of a pixel projected onto the ground plane $\mathbf{v}_r$ in the egocentric robot coordinate system can be computed by intersecting the ray $\mathbf{v}_t$ with the ground plane using simple geometric triangulation: The equation

$$\mathbf{v}_r = a \cdot \mathbf{v}_t + \mathbf{t}_c$$

with the unknown scalar a has exactly one solution for x_r and y_r when the pixel designated by $\mathbf{v}_t$ lies below the horizon. The operating system of the Sony humanoid readily provides estimates for R_c and $\mathbf{t}_c$ that are computed from joint sensor values.

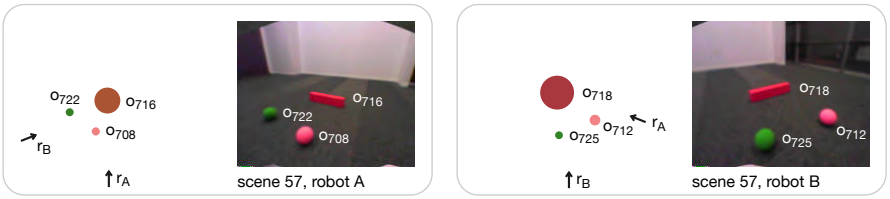

Fig. 5.5 Snapshots of the sensory experiences of both robots at the end of the image sequence in Fig. 5.2. Top: The camera images at that point in time are overlaid with the object anchors maintained by the tracking system. Left of them, the positions of objects and other robots in the egocentric reference system of each robot are shown. Each object is drawn as a circle in its average color, with the radius representing the object's width. The positions of the two robots (see Section 5.3.3 below) are indicated using black arrows. Bottom: The actual feature values are shown in each first column and feature values scaled to the interval $[0,1]$ in each second column. On the right side of the table, the third columns give for each scaled feature the difference between the perception of robot A and B.

Using these transformations, the position features `x` and `y` (in mm) are extracted from an object model by projecting the pixel at the center of the lower edge of the object's bounding box onto the ground plane. For estimating a `width` feature, the

lower left and right corner of a the bounding box are transformed into positions relative to the robot and the distance between them is calculated. For the computation of `height`, the ray of the pixel on the middle of the upper bounding box edge is intersected with a virtual plane perpendicular to the ground and through the position of the object as shown in Fig. 5.4. The extraction of color features from object models is also straightforward. The feature `luminance` is computed as the mean of an internal state's color histogram M_t^Y, `green-red` as the mean of M_t^{Cr} and `yellow-blue` from M_t^{Cb}.

The values of the `x` and `y` features are ususally in the range of meters, `width` and `height` can range from a few centimeters up to half a meter and values on color channels are within the interval $[0,255]$. In order to be able to handle all features independently from the dimensions of their domains, feature values are scaled to be within the interval $[0,1]$ using the statistical distributions of feature values. In theory the robots could gradually build up such distributions by seeing many different objects over the course of time, in practice the distributions are sampled from objects of recorded data sets. Given the mean μ and standard deviation σ of the distribution of a feature over a (large) number of objects, a scaled value is computed by mapping values in the interval $[\mu-2\sigma,\mu+2\sigma]$ onto $[0,1]$ and clipping all others. Fig. 5.5 gives an example of the sensory experiences of the two robots. For each object, both the unscaled and scaled feature values are given.

5.2.4 Related Work

The psychological and neurobiological literature on vision contains a lot of evidence for correlates of these three sub-systems in the human brain. First, there are dedicated neural assemblies along the visual stream from the retina to the primary visual cortex that detect basic visual features on a number of separable dimensions such as color, orientation, spatial frequency, brightness and direction of movement. These *early vision* processes operate independently from attention to objects and features "are registered early, automatically, and in parallel across the visual field" (Treisman and Gelade, 1980, p. 98). From there on, two separate visual pathways (also known as the "what" and "where" systems) are responsible for identifying objects and encoding properties about them (see Mishkin et al, 1983 for an early review):

- A dorsal stream (the "where" system) connecting the primary visual cortex and the posterior parietal cortex is responsible for the primitive individuation of visual objects, mainly based on spatial features. "Infants divide perceptual arrays into units that move as connected wholes, that move separately from one another, and that tend to maintain their size and shape over motion" (Spelke, 1990, p. 29). These "units" can be understood as "pointers" to sensory data about physical objects that enable the brain for example to count or grasp objects without having to encode their properties. They can be compared to the *anchors* mentioned above and are subject of a large number of studies: Marr (1982) calls them *place tokens*, Pylyshyn (2001, 1989) *visual indexes*, Ballard et al (1997) *deictic codes*

and Hurford (2003) discusses them from an artificial intelligence and linguistics perspective as *deictic variables*.

- There is a second, so called ventral, stream (the "what" system). It runs to the infero-temporal cortex. Properties of objects are *encoded* and temporarily stored in the *working memory* (Baddeley, 1983) for the use in other cognitive processes. What these properties are depends on top-down attentional processes – for example different aspects of objects have to be encoded when a subject is asked to count the number of "big objects" vs. the number of "chairs".

In addition to findings from neuroscience, there is also a variety of previous work in robotics to rely on. The most widely known setups for grounding symbolic representations in visual data for the purpose of communication is probably the Talking Heads experiment (Steels, 1998). The vision system of that experiment is discussed in Belpaeme et al (1998). Static scenes consisting of geometric shapes on a blackboard are perceived by robotic pan-tilt cameras and the vision system is able to extract features such as color, size and position from these shapes. Siskind (1995) describes a computer program for creating hierarchical symbolic representations for simple motion events from simulated video input and in Siskind (2001) from real video sequences. Similar systems have been proposed by Baillie and Ganascia (2000); Steels and Baillie (2003); Dominey and Boucher (2005) and Chella et al (2003), which is inspired by *conceptual spaces* Gärdenfors (2000).

Furthermore, there is a vast literature on object detection and tracking algorithms for other purposes than symbol grounding (see Yilmaz et al, 2006, for an extensive review). And the vision system introduced here does not reinvent the wheel but makes use of well-established techniques such as color histograms and Kalman filters. It differs, however, from many other approaches in the notion of what is considered to be an object. The types of objects that are expected to occur in the world are often explicitly represented in the vision system, for example by using pre-specified color ranges for identifying different object classes in images (e.g. Pérez et al, 2002), by matching (sometimes learnt) object templates with images (e.g. Hager and Belhumeur, 1998) or by engineering dedicated algorithms tailored for recognizing specific classes of objects (e.g. Jüngel et al, 2004).

In contrast, our robots have no preconceptions of what to expect in their environment and thus can detect and track any type of object, using only two assumptions: First, everything appearing in the environment that sufficiently distinguishes itself from the background and that was not there before is considered to be an object. Second, objects have to be on the ground for being able to make reliable position and size estimates.

5.3 Joint Attention and Social Learning in Robots

Robots learning a language are not only grounded in the physical world through their sensorimotor apparatus but also socially grounded in interactions with others. In addition to perceptual capabilities for detecting and tracking objects in their envi-

ronment they need a set of social skills for engaging in communicative interactions with each other. This includes mechanisms for joint attention and pointing as well as behavioral scripts for structured conversations. Joint attentional scenes (Tomasello, 1995) "are social interactions in which the child and the adult are jointly attending to some third thing, and to one another's attention to that third thing, for some reasonably extended length of time" (Tomasello, 1999, p. 97). Establishing joint attention means in our robotic experiments that two robots taking part in a language game must (1) share a physical environment, (2) attend to a set of objects in their surrounding, (3) track whether the respective other robot is able to attend to the same set of objects and (4) be able to manipulate attention by pointing to distal objects and perceiving these pointing gestures (see Fig. 5.6).

5.3.1 Social Robotics

How social mechanisms can be implemented in robots is a research area in its own. Scientist in this field are mainly interested in how social skills can improve communication and collaboration between humans and robots (Breazeal, 2002). Additionally, by trying to endow robots with social behaviors that appear "natural" to human observers, they want to understand what social cues humans are responding to. For reviews, refer to Dautenhahn et al (2002) who developed taxonomies for different degrees of robots' embodiment and "social embeddedness", Fong et al (2002) who give a general survey of socially interactive robots, and Vinciarelli et al (2009) who review the field of "social signal processing", i.e. the detection of social cues in human behavior. For an overview of skills that are prerequisites for joint attention and the state of the art in robotic experiments trying to implement these skills, refer to Kaplan and Hafner (2006). Some examples of work relevant for the experiments in this paper are listed below.

Scassellati (1999) endowed the "Cog" robot with capabilities for finding human faces, extracting the location of the eye within the face, and determining if the eye is looking at the robot for maintaining eye contact (or mutual gaze). Marjanovic et al (1996) showed how the same robot could learn to control his arm for pointing at distal objects in the surrounding space, guided by the camera of the robot. Gaze recognition was investigated among many others by Kozima and Yano (2001). They demonstrated how the "Infanoid" robot is able to track gaze direction in human faces and use this information to identify objects that humans are looking at. Joint attention is established by alternatingly looking at distal objects and the faces. Nagai et al (2003) modeled the transitions between different developmental stages that infants are going through in the process of learning to engage in joint attentional scenes, resulting in the robot being able to determine which object a human caregiver is looking at.

For recognizing pointing gestures performed by humans, Kortenkamp et al (1996) developed a robot that can detect and track the 3D positions of arm and shoulder joints of humans in dynamic scenes, without requiring the humans to wear

Fig. 5.6 Demonstration of a Sony humanoid robot drawing the attention of the other robot to an object in the shared environment by pointing at it. The images at the right show the scene as seen through the camera of the pointer (top) and the robot observing the pointing (bottom). However, please note that the robots are not able to detect pointing gestures using their built-in cameras. Instead, they directly transmit x, y coordinates of the object pointed at.

special markers. By searching along the vector defined by the detected arm joints, the robot can determine which object the experimenter was pointing at. Similarly, Martin et al (2009) used pointing gestures to instruct a mobile robot where to navigate to. Colombo et al (2003) used multiple cameras for tracking humans pointing at areas on walls in a room. Nickel and Stiefelhagen (2007) equipped a robot with stereo cameras and use color and disparity information and Hidden Markov Models to track both the direction of gaze and the position where a human is pointing at. Haasch et al (2005) apply the ability to recognize pointing gestures for teaching words for objects in a domestic environment and Imai et al (2004) showed how the robot "Robovie" could combine mechanisms for establishing mutual gaze and pointing at objects to draw the attention of humans to a poster in the environment of the robot. Finally, Hafner and Kaplan (2005) demonstrated how recognition of pointing gestures could be learned in Aibo robots. One robot performs a hard-wired pointing gesture and the other one has to detect whether it was to the left or to the right.

Additionally there is considerable research into implementing and learning the necessary behaviors for engaging in structured conversations. Breazeal (2003) investigated turn taking with the kismet robot, focussing on the factors regulating the exchange of speaking turns so that the communication seems natural to human interlocutors. Cassell et al (1999) discussed how nonverbal gestures and gaze can support turn taking behaviors in multimodal dialogs with the embodied conversational agent (ECA) “Gandalf”, trying to replicate findings from psychologic data. Recent work on communication with ECAs is reviewed by Kröger et al (2009) for the co-

ordination of communicative bodily actions across different modalities and by Kopp (2010) for the alignment of communicative behaviors between interlocutors.

5.3.2 Implementing Language Games in Robots

Language games are coordinated by behavioral scripts. Every agent in the population knows the language game script and individually reacts to changes in the environment and actions of the other robot. For example the speaker triggers the action of pointing to the intended topic when the hearer signals that he did not understand the utterance. The scripts are implemented in the form of finite-state machines: actions are performed depending on the current state in the game flow, the perception of the environment and the history of the interaction.

Joint attention is monitored by an external computer program, that has access to the world models of both interacting robots. This system initiates the interaction between two agents as soon as both agents observe the same set of objects. It is the task of the human experimenter to find spatial setups in which joint attention is possible, the program only monitors whether robots are seeing the same set of objects. But in the literature there are also other proposals for establishing joint attention in embodied language game experiments. For example Steels and Vogt (1997) programmed sophisticated signaling protocols into LEGO robots. A robot that decides to become a speaker emits an infrared signal and the other robot then aligns its position so that it faces the speaker. The robots "point" to objects by orienting themselves toward them. In the Talking Heads experiment (Steels, 1998), the speaker directly controls the view direction of the hearer's camera in order to make sure that their cameras perceive the same objects on the whiteboard. An agent points to an object by letting the other agent's camera zoom in on it. In contrast, establishing joint attention in social language learning scenarios between humans and robots is usually easier because the human experimenter (as a well-trained social being) is good at monitoring the attention of the robot and can for example (as in Dominey and Boucher, 2005) point to an object by moving it.

For playing a language game robots need non-linguistic means of conveying information, such as pointing to an object or conveying notions of success, failure and agreement in communication. For demonstration purposes robots were equipped with pointing gestures but in the communicative interactions underlying the results presented in this paper, robots use a different mechanism in order to avoid further difficulties stemming from uncertainties in pointing (see Steels and Kaplan, 1998, for a discussion of the impact of such uncertainties on the performance in language games).

When a robot wants to point to an object in the environment, he directly transmits the x_o, y_o coordinates of the intended object o to the interlocutor. Since robots model object positions in their own (egocentric) coordinate systems, additional steps have to be taken to interpret these coordinates. Most importantly the robot has to know the position x_r, y_r and orientation θ_r of the robot that is pointing r (see next Section

5.3.3 for details on how robots estimate these values). With this information robots transform the coordinates into their own coordinate system:

$$\mathbf{v} = \begin{pmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \end{pmatrix} \begin{pmatrix} x_o \\ y_o \end{pmatrix} + \begin{pmatrix} x_r \\ y_r \end{pmatrix}$$

The robot interpreting the pointing is determining the intended object by choosing the object in his world model that is closest to **v**. Similarly, robots directly exchange other non-linguistic feedback, for instance agreement and disagreement in communication by exchanging signals whose meaning is shared. Moreover, linguistic utterances are directly passed between interlocutors.

The mechanisms presented in this section provide simple solutions to required capacities for social language learning that are not meant to be in themselves proposals as to how these skills could be implemented. Nevertheless, we claim that the realism of this study does not suffer from this simplicity: humans rely on extremely powerful mechanisms for perceiving and sharing intentions within interactive situations Tomasello et al (2005) and similarly our solutions provide us with the technical prerequisites for letting our robots learn from communicative interactions.

5.3.3 Robot Pose Estimation

In order to point to a distal object, robots directly signal the coordinates of the object in their coordinate system to interlocutors. To interpret the transmitted coordinates, robots need to estimate the position and orientation of the other robot. To that end robots localize themselves with respect to landmark objects in the environment and transmit their position with respect to these landmarks to the other robot. This way both agents establish mutual knowledge about their position.

We use carton boxes enhanced with visual markers (see Fig. 5.7) as landmark objects. The unique, black and white, barcode-like, 2D-patterns attached to carton boxes are tracked using the ARToolKitPlus library (Wagner and Schmalstieg, 2007). Raw camera images from the camera of the robot are preprocessed before being passed to the toolkit. From each camera image, a histogram of the pixel luminance is computed. This histogram is then used to derive a threshold for creating a binary image as shown in the top right of Fig. 5.7. The binary image is passed to the tracking library, which searches it for marker patterns and determines the four vertices of the polygon surrounding the marker in the image (see bottom left of Fig. 5.7). Provided with the camera resolution width and height (in pixels), the width and height camera opening angle (in deg) and the widths of the markers used on the carton boxes (in mm), the tracking library is able to make an orientation and position estimate from the edges of the detected patterns, which is then iteratively enhanced by matrix fitting. As a result, the system returns for each detected marker pattern a unique ID and a matrix describing the position and orientation of the marker relative

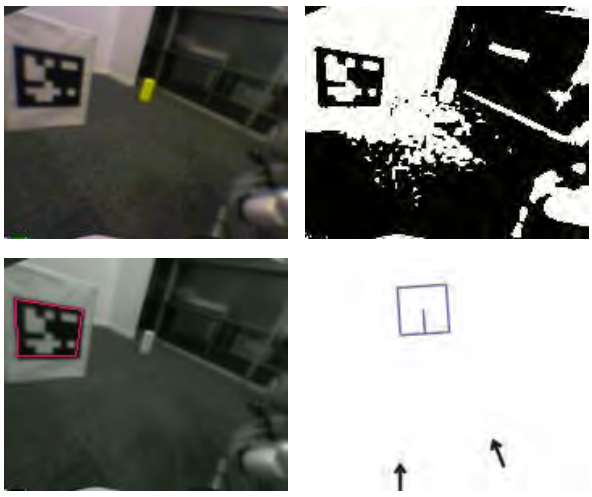

Fig. 5.7 Using objects enhanced with visual markers for estimating the position and orientation of the other robot. Top left: A 2D pattern attached to a carton box as seen through the camera of a Sony humanoid robot. Top right: Binary image generated from the original image. Bottom left: The marker as detected by the ARToolKit tracking system. Bottom right: Both robots send the position and orientation of the carton box (blue) to each other and are thus able to deduce the position and orientation of the respective other robot.

to the camera of the robot (for details of the pose estimation algorithm see Kato and Billinghurst, 1999).

To transform the camera relative marker position and orientation into robot egocentric coordinates, they are transformed using the offset and orientation of the camera relative to the ground point of the robot (see Section 5.2.3). Finally, for each marker attached to a carton box, the offset and orientation relative to the center of the box, which is a priori known, is used to determine the position and orientation of the box in egocentric coordinates. To filter out noise and recognition errors, the resulting box poses are averaged over the last n images. Also, when two markers of the same box are detected in the same image, their resulting box poses are averaged. The output of the landmark modeling system is a list of objects consisting of an ID (an ID of the box, not to confuse with the ID of the marker patterns) and a pose $\mathbf{b} := \begin{pmatrix} x_b & y_b & \theta_b \end{pmatrix}$ of the carton box in robot egocentric coordinates.

In order to determine the position x_r, y_r and orientation θ_r of the respective other robot, the robots use the carton boxes as global landmarks (see bottom right of Fig. 5.7). About five times per second they exchange the poses of the boxes they have seen over a wireless network connection. Given that both robots see the same box (all robots use the same box IDs for the same visual markers), they can compute the pose of the other robot from the box pose $\mathbf{b}$ as perceived by the robot (in egocentric

coordinates) and the $\mathbf{b}'$ as sent by the other robot (in the coordinate system of the other robot):

$$\begin{pmatrix} x_r \\ y_r \\ \theta_r \end{pmatrix} := \begin{pmatrix} x_b - \cos(\theta_b - \theta'_b) \cdot x'_b + \sin(\theta_b - \theta'_b) \cdot y'_b \\ y_b - \cos(\theta_b - \theta'_b) \cdot x'_b + \sin(\theta_b - \theta'_b) \cdot x'_b \\ \theta_b - \theta'_b \end{pmatrix}$$

When both robots see multiple boxes the results of the above transformation are averaged.

5.4 Discussion and Conclusion

Both the visual perception for constructing a world model and the recognition of bodily gestures to achieve feedback in language games have been tested extensively in various language game experiments reported in this book and companion volumes. The software is parameterized from the viewpoint of the precise shape of the robot (particularly for robot pose estimation), and consequently it is possible to port the present software relatively easily from one platform to another one. Indeed such porting activities have been performed to accommodate the new MYON platform developed in ALEAR.

Acknowledgements

The research reported here was carried out at the Sony Computer Science Laboratories in Paris and Tokyo. We are greatly indebted to Masahiro Fujita, Hideki Shimomura, and their team for creating the Sony humanoid robots and for making them available for the experiments reported here. This research was funded by the Sony Computer Science Laboratory in Paris with additional funding from the ECAGENTS and ALEAR projects funded by the EU FP6 and FP7 frameworks.

References

Aherne F, Thacker NA, Rockett PI (1998) The Bhattacharyya metric as an absolute similarity measure for frequency coded data. Kybernetika 34(4):363–368

Baddeley AD (1983) Working memory. Philosophical Transactions of the Royal Society of London Series B, Biological Sciences (1934-1990) 302(1110):311–324

Baillie JC, Ganascia JG (2000) Action categorization from video sequences. In: Horn W (ed) Proceedings ECAI, IOS Press, pp 643–647

Ballard DH, Hayhoe MM, Pook PK, Rao RPN (1997) Deictic codes for the embodiment of cognition. Behavioural and Brain Sciences 20(4):723–742

Belpaeme T, Steels L, Van Looveren J (1998) The construction and acquisition of visual categories. In: Proceedings EWLR-6, Springer, LNCS, vol 1545, pp 1–12

Bhattacharyya A (1943) On a measure of divergence between two statistical populations defined by their probability distributions. Bulletin Calcutta Mathematical Society 35:99–110

Breazeal C (2002) Designing Sociable Robots. MIT Press

Breazeal C (2003) Toward sociable robots. Robotics and Autonomous Systems 42(3-4):167–175

Brooks A, Arkin R (2007) Behavioral overlays for non-verbal communication expression on a humanoid robot. Autonomous Robots 22(1):55–74

Cassell J, Torres OE, Prevost S (1999) Turn taking vs. discourse structure: how best to model multimodal conversation. Machine Conversations pp 143–154

Chella A, Frixione M, Gaglio S (2003) Anchoring symbols to conceptual spaces: the case of dynamic scenarios. Robotics and Autonomous Systems 43(2-3):175–188

Colombo C, Del Bimbo A, Valli A (2003) Visual capture and understanding of hand pointing actions in a 3-D environment. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 3(4):677–686

Coradeschi S, Saffiotti A (2003) An introduction to the anchoring problem. Robotics and Autonomous Systems 43(2-3):85–96

Cruse H, Dürr V, Schmitz J (2007) Insect walking is based on a decentralized architecture revealing a simple and robust controller. Phil Trans R Soc A 365:221–250

Dautenhahn K, Odgen B, Quick T (2002) From embodied to socially embedded agents–implications for interaction-aware robots. Cognitive Systems Research 3(3):397–428

Dominey PF, Boucher JD (2005) Learning to talk about events from narrated video in a construction grammar framework. Artificial Intelligence 167(1-2):31–61

Fong T, Nourbakhsh I, Dautenhahn K (2002) A survey of socially interactive robots. Robotics and Autonomous Systems 42(3-4):143–166

Fujita M, Kuroki Y, Ishida T, Doi TT (2003) Autonomous behavior control architecture of entertainment humanoid robot sdr-4x. In: Proceedings IROS '03, pp 960–967, vol. 1

Gärdenfors P (2000) Conceptual Spaces: The Geometry of Thought. MIT Press

Haasch A, Hofemann N, Fritsch J, Sagerer G (2005) A multi-modal object attention system for a mobile robot. In: Proceedings IROS '05, pp 2712–2717

Hafner V, Kaplan F (2005) Learning to interpret pointing gestures: experiments with four-legged autonomous robots. In: Biomimetic Neural Learning for Intelligent Robots, LNCS, vol 3575, Springer, pp 225–234

Hager GD, Belhumeur PN (1998) Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(10):1025–1039

Hurford JR (2003) The neural basis of predicate-argument structure. Behavioral and Brain Sciences 26(3):261–316

Imai M, Ono T, Ishiguro H (2004) Physical relation and expression: joint attention for human-robot interaction. IEEE Transactions on Industrial Electronics 50(4):636–643

Ishiguro H (2006) Android science: conscious and subconscious recognition. Connection Science 18(4):319–332

Jüngel M, Hoffmann J, Lötzsch M (2004) A real-time auto-adjusting vision system for robotic soccer. In: Polani D, Browning B, Bonarini A (eds) RoboCup 2003: Robot Soccer World Cup VII, Springer, LNCS, vol 3020, pp 214–225

Kalman RE (1960) A new approach to linear filtering and prediction problems. Transactions of the ASME-Journal of Basic Engineering 82(1):35–45

Kanda T, Kamasima M, Imai M, Ono T, Sakamoto D, Ishiguro H, Anzai Y (2007) A humanoid robot that pretends to listen to route guidance from a human. Autonomous Robots 22(1):87–100

Kaplan F, Hafner V (2006) The challenges of joint attention. Interaction Studies 7(2):129–134

Kato H, Billinghurst M (1999) Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: Proceedings ISAR '99, pp 85–94

Kopp S (2010) Social resonance and embodied coordination in face-to-face conversation with artificial interlocutors. Speech Communication 52(6):587–597

Kortenkamp D, Huber E, Bonasso RP (1996) Recognizing and interpreting gestures on a mobile robot. In: Proceedings AAAI-96, pp 915–921

Kozima H, Yano H (2001) A robot that learns to communicate with human caregivers. In: Proceedings EPIROB '01

Kröger B, Kopp S, Lowit A (2009) A model for production, perception, and acquisition of actions in face-to-face communication. Cognitive Processing

Marjanovic M, Scassellati B, Williamson M (1996) Self-taught visually-guided pointing for a humanoid robot. In: Proceedings SAB '96, The MIT Press, pp 35–44

Marr D (1982) Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. W. H. Freeman, San Francisco, CA

Martin C, Steege FF, Gross HM (2009) Estimation of pointing poses for visually instructing mobile robots under real world conditions. Robotics and Autonomous Systems 58(2):174–185

Mishkin M, Ungerleider LG, Macko KA (1983) Object vision and spatial vision: two cortical pathways. Trends in Neurosciences 6:414–417

Nagai Y, Hosada K, Morita A, Asada M (2003) A constructive model for the development of joint attention. Connection Science 15(4):211–229

Nickel K, Stiefelhagen R (2007) Visual recognition of pointing gestures for human-robot interaction. Image and Vision Computing 25(12):1875–1884

Pérez P, Hue C, Vermaak J, Gangnet M (2002) Color-based probabilistic tracking. In: Proceedings ECCV '02, Springer, LNCS, vol 2350, pp 661–675

Pfeifer R, Lungarella M, Iida F (2007) Self-organization, embodiment, and biologically inspired robotics. Science 318:1088–1093

Pylyshyn ZW (1989) The role of location indexes in spatial perception: A sketch of the FINST spatial-index model. Cognition 32(1):65–97

Pylyshyn ZW (2001) Visual indexes, preconceptual objects, and situated vision. Cognition 80(1):127–158

Scassellati B (1999) Imitation and mechanisms of joint attention: A developmental structure for building social skills on a humanoid robot. In: Nehaniv CL (ed) Computation for Metaphors, Analogy, and Agents, LNCS, vol 1562, Springer, pp 176–195

Siskind JM (1995) Grounding language in perception. Artificial Intelligence Review 8(5-6):371–391

Siskind JM (2001) Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. Journal of Artificial Intelligence Research 15:31–90

Soille P (2003) Morphological Image Analysis: Principles and Applications. Springer

Spelke ES (1990) Principles of object perception. Cognitive Science 14(1):29–56

Spranger M (2008) World models for grounded language games

Spranger M, Pauw S, Loetzsch M, Steels L (2012) Open-ended procedural semantics. In: Steels L, Hild M (eds) Grounding Language in Robots, Springer Verlag, Berlin

Steels L (1998) The origins of syntax in visually grounded robotic agents. Artificial Intelligence 103(1-2):133–156

Steels L, Baillie JC (2003) Shared grounding of event descriptions by autonomous robots. Robotics and Autonomous Systems 43(2-3):163–173

Steels L, Kaplan F (1998) Stochasticity as a source of innovation in language games. In: Proceedings ALIFE '98, MIT Press, pp 368–376

Steels L, Vogt P (1997) Grounding adaptive language games in robotic agents. In: Proceedings ECAL '97, The MIT Press, pp 473–484

Tomasello M (1995) Joint attention as social cognition. In: Moore C, Dunham PJ (eds) Joint Attention: Its Origins and Role in Development, Lawrence Erlbaum Associates, Hillsdale, NJ

Tomasello M (1999) The Cultural Origins of Human Cognition. Harvard University Press, Harvard

Tomasello M, Carpenter M, Call J, Behne T, Moll H (2005) Understanding and sharing intentions: The origins of cultural cognition. Behavioral and Brain Sciences 28:675–691

Treisman AM, Gelade G (1980) A feature-integration theory of attention. Cognitive Psychology 12(1):97–136

Vinciarelli A, Pantic M, Bourlard H (2009) Social signal processing: Survey of an emerging domain. Image and Vision Computing 27(12):1743–1759

Wagner D, Schmalstieg D (2007) ARToolKitPlus for pose tracking on mobile devices. In: Proceedings CVWW '07

Yilmaz A, Javed O, Shah M (2006) Object tracking: A survey. ACM Computing Surveys 38(13):1–45