Learning Visual Object Definitions by Observing Human Activities

Manuela Veloso, Felix von Hundelshausen, and Paul E. Rybski
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, 15213, USA

Abstract*— Humanoid robots, while moving in our everyday environments, necessarily need to recognize objects. Providing robust object definitions for every single object in our environments is challenging and impossible in practice. In this work, we build upon the fact that objects have different uses and humanoid robots, while co-existing with humans, should have the ability of observing humans using the different objects and learn the corresponding object definitions. We present an object recognition algorithm, FOCUS, for** *Finding Object Classifications through Use and Structure. FOCUS learns** *structural properties* **(visual features) of objects by knowing first the object's** *affordance properties* **and observing humans interacting with that object with known activities. FOCUS combines an activity recognizer, flexible and robust to any environment, which captures how an object is used with a low-level visual feature processor. The relevant features are then associated with an object definition which is then used for object recognition. The strength of the method relies on the fact that we can define multiple aspects of an object model, i.e., structure and use, that are individually robust but insufficient to define the object, but can do so jointly. We present the FOCUS approach in detail, which we have demonstrated in a variety of activities, objects, and environments. We show illustrating empirical evidence of the efficacy of the method.**

***Index Terms*— Affordance, Cognition**

I. Introduction

Learning by observation is a powerful technique by which a humanoid robot can obtain knowledge about the physical world by watching humans interact with objects within it. Specifically, such observations provide a powerful method for learning affordance properties [1] of those objects. Affordance properties, or how an object can be interfaced with, capture the essence of an object's utility. For instance, a chair can commonly be used as a place to sit, but it can also be used as something to stand on in order to reach something on a high shelf. Likewise, a ladder can be used for obtaining something on a high shelf, but can also be used as something to sit on. By observing different activities performed on and with specific objects, the functional descriptions of these objects can be further enriched.

These techniques are particularly relevant for humanoid robots. By definition, a humanoid robot's physical characteristics and appendages mimic that of a human's. As a result, humanoids can be expected to be able to perform sets of activities similar to what humans are capable of performing. Our long-term research goal is to have humanoid robots observe and interact with humans over long periods of time. During this period of time, the robots learn to identify objects by observing humans use them and learn to associate sets of activities that can be performed with those objects.

As with any sensor-based system, the importance of good prior knowledge cannot be emphasized enough. Object models provided to an intelligent sensor (as carried by a robot) dictate how it can take signal data from a sensor stream, such as images from a camera, and distill meaningful semantic information. Our efforts mainly focus on robots using vision-based sensor systems to understand the world around them. Several other efforts have focused on enriching such object models with a variety of knowledge, such as context [2] or activity [3]. In contrast to these approaches, which make use of known visual object models, we relax the assumption that specific visual models of objects are provided to the robot ahead of time. For example, the robot knows that chairs exist but initially has no idea what they look like. Instead, we assume that the robot is aware of humans and can recognize activities that these humans perform. By observing activities performed with specific objects, the robot can learn the appropriate affordance properties for these objects and properly classify them.

In this paper, we present an algorithm called FOCUS (*Finding Object Classification through Use and Structure*) which models inanimate objects in the environments by structural and functional definitions. The structural part of the model aims at capturing a simple and generalized visual definition of an object through robust feature detectors. The functional part of the model captures the affordance properties of that object: *one sits down on a chair*. Objects in the environment are recognized by associating an observed action with a particular environmental feature.

The classification of the object is dependent upon the specific activity for which it is used by the person. As an example, if a robot equipped with the FOCUS algorithm observes a human walk through a room and sit in a chair, then the visual features nearest to where the human sat would fall under the classification of "chair." In this case, a "chair" is anything that a human will sit upon. This classification of "chair" could very well be given to a small table, a couch, or even a heat register if the human chose to sit upon it. The interesting aspect of this functional view is that it can be rather robust to the specific environment conditions of the signal capture. By connecting "sitting" with non-ambitious definition of a "chair,"

the problem is converted mainly into motion recognition and the robustness to the environment is achieved. By finding one object in the image, we can then generalize and find multiple similar objects.

While FOCUS is a general sensor interpretation algorithm capable of operating on any video stream, it is particularly useful for humanoid robots because it relies on the ability to recognize humans and to model their activity. Possession of this prior knowledge is very appropriate for humanoids because they are assumed to be capable of performing a set of pre-defined physical human activities such as sitting, walking, moving through portals, etc...)

The primary contributions of this paper are as follows:

- Unlike other object recognition algorithms which require exemplar object definitions, FOCUS does not require specific visual models of the objects ahead of time. The robot's initial concept of an object has no visual features associated with it until they are learned by observing a human interacting with that object in a specific fashion.
- FOCUS provides a generalization of the object descriptor which abstracts away from specific visual feature modalities. Any feature detection algorithm may be used in so far as a *FeatureDescriptor* function and data structure can be defined for it. This descriptor abstraction allows any feature detector to identify regions of interest in the image. Similarly, for each descriptor, a corresponding *SimilarityMeasure* function must be defined which provides a metric by which individual features of the same type can be compared.

The paper is organized as follows. Section II discusses related work. The FOCUS algorithm is described in Section III. Examples of how the algorithm performs on video data sequences are presented in Section IV. Finally, section V concludes the paper.

II. Related Work

In the history of computer vision, many different approaches have been proposed for how to incorporate prior knowledge for recognition. Some examples of this range from active contour models [4], [5], to autonomous vehicle guidance on a highway [6], and tracking a leaf in a cluttered background [7]. These methods worked well because of their use of appropriate prior knowledge. Other work has been devoted to the problem of learning specific features through methods such as PCA [8]. In [2], context is used to help disambiguate objects observed in scenes. Additionally, activity was used to remarkably recognize a variety of known objects [3], in specific environments. These approaches have relied upon specific *a priori* visual object models. Our work assumes that no such visual information is available and that the robot must learn it by observing humans.

In the vision literature, the term functional object recognition has traditionally been used to describe the visual analysis and recognition of the parts of an object [9]. For instance, by recognizing the handle and striking surface of an object, that object can be recognized as a hammer [10]. Other systems reason about the causal structure of the visually-observable physics behind a scene [11]. In contrast, our work relies on recognizing how an object is used, i.e., its function, in terms of the activities that a person performs on or near them rather than a detailed analysis of the visual features of the objects themselves.

In the humanoids literature, object recognition and learning object models from multiple sources and modalities of information has been explored by [12]. Another method, described in [13], discusses how a robot can use active perception to explore its environment to learn structural properties of objects through manipulation. The paradigm of learning from observation has also been used very successfully to train humanoids on specific tasks [14] as well as action generation [15]. Our algorithm combines these two paradigms by making use of multiple sources of information gleaned by observing human activities to learn the visual object definitions.

Representing a concept in multiple fashions has been proposed by [16]. In our representation, we consider both the function (how can the person interact with the object) as well as the structural definition (what does the object look like). We merge these two models to create a richer description of the object.

III. FOCUS : Finding Object Classification through Use and Structure

FOCUS combines activity recognition with low-level visual feature detection to learn visual models for object classes. Figure 1 illustrates how the FOCUS algorithm combines low-level visual features and activities identified from the video stream to learn visual models for objects. Because this algorithm depends on humans to teach the robot about its environment, a significant emphasis has been placed on detecting the presence of humans and understanding their motions. People are detected in the video stream with a face detection and tracking algorithm described in [17]. Once detected, the people are tracked and their motions are fed into an HMM-based activity recognition algorithm, described in [18], which returns course body-specific activities such as standing, sitting, and walking. FOCUS possesses a library of pre-defined object classes which are indexed by the classified activity. The low-level features that are closest to the location in the image where the activity took place are selected and become part of that object's visual definition. The updated object class definition is then stored in the library once again. Table I illustrates the steps that the FOCUS algorithm takes when determining which visual features belong to the object with which the human is interacting. Definitions of object classes, low-level feature abstractions, and algorithms for extracting features and generalizing to new objects are described in more detail in the following sections.

A. Object Classes

Object classes in FOCUS consist of several different components, some of which must be pre-defined. The pre-defined components represent the prior knowledge about humans and

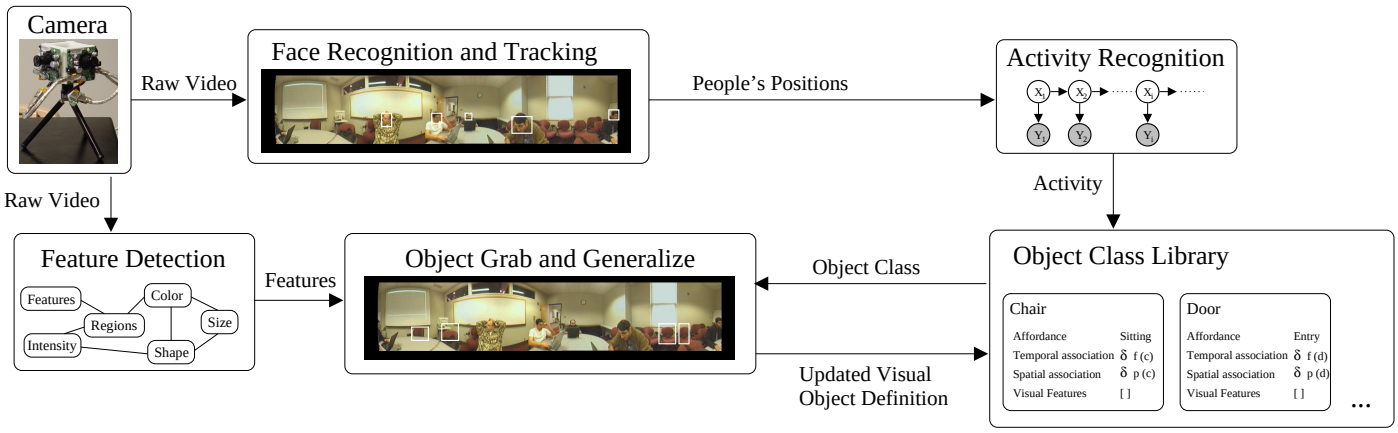


Fig. 1. Learning visual features from identifying activities. Raw video taken in the environment is captured by a camera and passed both to a low-level feature extractor as well as a face recognizer and tracker. The motions of tracked faces are passed to an activity recognizer which returns the activity of that person. If an activity is defined such that it requires the human to be interacting with a particular object, then an element in the object class library is selected and updated with the visual features that are closest in proximity to the location of the activity. This updated object definition is passed back to the class library for later use in identifying that class of objects.

FOCUS

For each image F_j

- Search images for low-level features R_i, $i = 1, ..., n$,
- Run activity recognition based on face tracking
- If a known activity is recognized with face at pixel P_s, do:
  - **Temporal Association:** index the image $F_{j-\Delta f}$, where Δf is the temporal association of object.
  - **Spatial Association:**
    - Predict the expected pixel location $P_{s-\Delta p}$, where Δp is the spatial association of object.
    - Select the feature R_k with center of gravity closest to $P_{s+\Delta p}$.
    - Classify region R_k as object
  - **Object Generalization:** Search for regions with similar color to R_k, to classify other objects of the same class.

TABLE I

FUNCTIONAL OBJECT RECOGNITION

activities that must be known before the visual features of any object can be learned. These pre-defined components include the specific affordance properties of the object, the expected spatial association with respect to the human's activity, and a temporal association which is used to avoid occlusions. The unknown component of an object is the visual feature definition which, unlike the previous components, can be initially unknown. All of the *a priori* object class properties are stored in an *ContextDescriptor* which is used by the FOCUS algorithm to identify the low-level visual features to store.

1) Affordance Property: Every object class to be detected must have a specific activity that can be recognized when the human uses it. Two examples of such object categories are chairs and doors. For FOCUS, a chair is any object that a human can sit down upon, and a door is any object that a human can walk through. As long as the object recognizer can identify those activities when they occur, the location of the person is used to specify low-level features that are part of the object.

2) Spatial Association: Some *a priori* assumptions must be made about objects and how a person will interact with them. For instance, if an object is expected to be at head height or higher, candidate feature definitions that are below that location can be excluded. Likewise, additional heuristics can be employed to identify typical background regions that are walls and ceiling. Note that these assumptions only put very weak constraints on the kinds of features that can make up these objects and cannot be used to solely identify the object in question.

3) Temporal Association: In many cases, when a particular activity is identified which would indicate the presence of a certain object, the person will most likely be occluding some or all of the object from the camera's point of view. In order to obtain an unobstructed view of the object, FOCUS keeps a history of the captured frames of video. This history is used to effectively "rewind" the visual record to a time well before the human performed the specific activity. This is intended to allow FOCUS an unobstructed view of the object.

4) Visual Features: This component represents the learned definition of the object that consists of a list of low-level feature abstractions (defined in the next section). As the FOCUS algorithm observes more examples of a particular object class, the definition of that class is enriched by additional visual feature examples.

B. Low-Level Visual Feature Abstractions

FOCUS makes use of a visual feature abstraction which allows any low-level feature type (or combinations of feature types) to be used as the foundations for learning visual object models. This is one of the primary strengths of the FOCUS

algorithm since it does not rely on any specific feature modality but can easily use any that are made available to it. A FOCUS low-level visual feature consists of the following components: a *FeatureDescriptor*, at least one *SimilarityMeasure*, and a *CandidateFeature*.

Grab *for region features*

- Given:
  1) A *ContextDescriptor*, in particular a point $\mathbf{p}$ from activity recognition where a region feature might be and a time shift Δt that specifies image $I_{t-\Delta t}$ when the hypothetical feature is likely to have been visible.
  2) An occupancy grid O of the same size than the image
  3) An connectivity grid C of the same size than the image
  4) A color similarity measure s_c.
  5) A color similarity threshold t_c.
  6) A queue Q in which initially only p is stored.
- Initialize:
  1) Clear the occupancy grid, only mark the corresponding cell of p.
  2) Clear the connectivity grid
  3) Set the mean color c_m to zero.
- Iterate while Q is not empty
  - extract p from Q
    - for all neighbors q_i of p
      1) if the occupancy grid is free at q_i
         - Update the mean color c_m by the color at q_i
         - Calculate the color difference Δc between pixel p and q_i using similarity measure s_c.
         - If $\Delta c > t_c$ add q_i to Q and mark the occupancy grid at q_i
      2) else mark the corresponding edge in the connectivity grid C
- Extract all boundary contours of the region from C, including "islands" inside regions, determine the region's outer boundary as a sequence of points.
- **Results:** A region feature descriptor (c_m, B) where
  - c_m is the mean color of the region
  - and B its outer boundary contour.

TABLE II
REGION GROWING WITH BOUNDARY EXTRACTION

- *FeatureDescriptor* : Specifies the definition of a particular feature and stores the necessary data fields. For example, a contiguous region feature is represented by a mean color and a boundary contour, while a SIFT [19] feature is represented by scale, orientation, a local image patch, and a keypoint vector.
- *SimilarityMeasure* : Compares two *FeatureDescriptors* of the same type. For a single *FeatureDescriptor* there might exist many different *SimilarityMeasures*.
- *CandidateFeature* : Describes two featire-specific functions **Grab** and **Generalize** which are the core of the FOCUS algorithm. The **Grab** function takes an object class *ContextDescriptor* and returns a *FeatureDescriptor* from the image if one can be found at the location specified by the *ContextDescriptor*. The **Generalize** function takes as input the previously-defined *FeatureDescriptor* and a *SimilarityMeasure*, searches the image for all features of that type, and returns the ones that match the similarity criteria.

Generalize *for region features*

- Given:
  1) A region feature descriptor F from a grabbed region
  2) An image I where to find similar features
  3) A *SimilarityMeasure* S that compares to region *FeatureDescriptors*.
  4) An initially empty list L of *FeatureDescriptors*.
- Determine all region *FeatureDescriptors* E_i in the image by tesselating the image by subsequent region growing with boundary extraction.
- For each of these *FeatureDescriptors* E_i evaluate the given *SimilarityMeasure* S with respect to the learned *FeatureDescriptor* F. If the *SimilarityMeasure* is met, include E_i in the list of features L.
- **Results:** The list L of features.

TABLE III
GENERALIZE FOR REGION FEATURES

Region Similarity Measure Based on Euclidean Color Difference

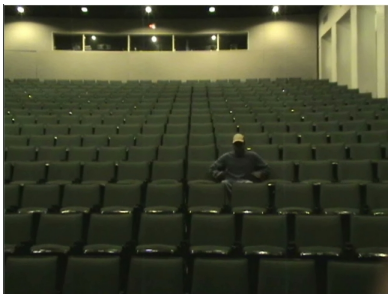
- Given:
  1) Two region feature descriptors F_1 and F_2
  2) A threshold t_c
- Calculate the Euclidean color distance c_e in RGB space of the mean color of region F_1 and F_2
  $c_e := \sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2}$
- **Results:** True if $c_e < c_t$, false otherwise.

TABLE IV
REGION SIMILARITY MEASURE BASED ON EUCLIDEAN COLOR DIFFERENCE

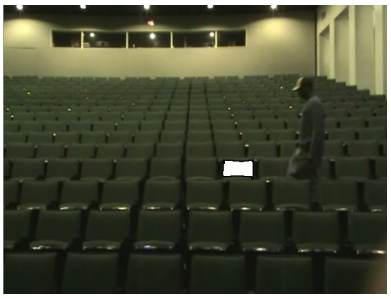
Currently, two different types of low-level features are implemented in FOCUS. The first is a contiguous region tracker, described in [20], and the second is based on the PCA-SIFT [21] algorithm.

C. Candidate Feature Extraction and Object Generalization

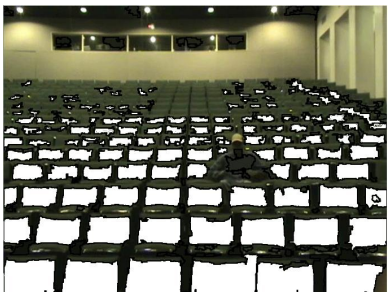
Because each low-level feature abstraction must define its own specific *CandidateFeature* and *SimilarityMeasure* components, we present example algorithms for the contiguous region feature. The algorithm for grabbing a contiguous region feature is defined in Table II. Similarly, the algorithm for generalizing that feature type is defined in Table III. Two different similarity measure for this feature type are defined for color, as shown in Table IV, and for shape, as shown in Table V.



(a) Example 1: A person is tracked in a theater as they walk in from the left and sit down in a chair. This sitting down action is detected by the activity recognizer which triggers the search for "chair" objects.



(b) The **Grab** function uses the history of saved frames to go back to a point where the person was not occluding the chair and obtains a contiguous image at that location (shown in white)



(c) The **Generalize** function takes the parameters learned from the **Grab** function and highlights all of the features that match those values (shown in white)

(d) Example 2: A person has emerged from their office. Their face is detected at the entryway to their door and the contiguous region represented by their door is obtained with the **Grab** function. The door is highlighted in white.

(e) The **Generalize** function uses the parameters returned by the contiguous region capture routine to find another door in the foreground of the image.

(f) On a different floor of the building, the newly generalized door parameters are used to find additional doors (in white).

Fig. 2. Examples of FOCUS for learning to recognize chairs as well as doorways.

Region Similarity Measure Based on Shape Difference

- Given:
  1) Two region feature descriptors F_1 and F_2
  2) A threshold t_s
- Calculate the bounding boxes R_1 and R_2 of the boundary contour of F_1 and F_2, respectively
- Calculate $c_s := |\log \frac{w_1}{h_1} - \log \frac{w_2}{h_2}|$
- **Results:** True if $c_s < t_s$, false otherwise.

TABLE V
REGION SIMILARITY MEASURE BASED ON SHAPE

IV. AN ILLUSTRATIVE EXAMPLE

Figure 2 illustrates two different examples of the FOCUS algorithm running on video streams. In the first example, shown in Figures 2(a)-(c), a person is tracked as they walk down the rows of a theater. The camera is placed in a stationary position near the bottom of rows of chairs. The theater is filled with chairs but FOCUS has not seen any examples of anyone sitting in these chairs before, so it does not know to recognize them. In Figure 2(a), the person sits down in a chair. The activity recognition system recognizes this event and FOCUS uses this information to identify the visual features for a chair. In Figure 2(b), the FOCUS algorithm backtracks a few seconds in the visual history until the tracked person's face is several body widths away from where the sit down activity occurred. This is to make sure that the person is not occluding the object on which they sit. In this example, the contiguous region feature detector was used to segment the image. The closest region to that sit down activity was selected and is highlighted in white. Finally, by using parameters learned from the example, FOCUS is able to generalize from that first region and identify a large number of other chairs in the image, as shown as white blobs in Figure 2(c). From this example, we can see that FOCUS has identified an association between a particular set of low-level vision features and the activity of sitting down. Note that not many of the chairs in the very back of the room are identified. This is merely a limitation of the resolution of the camera. With a higher-resolution camera or

a zoom lens to bring those chairs into view, they could also be identified.

In the second example, shown in Figures 2(d)-(f), FOCUS identifies that a person has appeared suddenly in the image. In this example, the camera is placed on a moving platform and travels at a fixed rate down the hallway. The sudden appearance of the person is an indication that they have emerged through a doorway of some sort. In Figure 2(d) FOCUS uses the contiguous region feature extractor to identify the region that was closest to the location where the person's face appeared. The region that is highlighted in white is the door. In Figure 2(e), the parameters of the region are used by the generalize function to locate other regions in the image which could correspond to doors (also shown in white). In this figure, a second doorway is identified. Finally, to illustrate the robustness of this algorithm, these new parameters were used on a different floor of the building, shown in Figure 2(f), to identify two additional doorways based on those learned parameters.

V. Conclusions

Object recognition is a well-known and difficult problem particularly from images where different views of objects and taken in different environments. In this paper, we introduce an approach well suited to use with humanoid robots because of the need for the system to understand human activity models *a priori*. Object usage is tracked in terms of motion detection which is quite robust with respect to the environment. FOCUS tracks the use of an object through recognizing activities of people in interacting with the environment. The main contributions of FOCUS are: (i) the use of object class definitions which describe how the object can be detected by its affordance properties rather than its specific visual characteristics; (ii) a framework for representation of low-level visual feature abstractions which allow the visual field to be segmented into potential object hypotheses.

Our definition of the object classes allows for any sort of generalized visual feature algorithm to be used. As FOCUS observes different instances of the same kind of object class, or even from the same direction, the object models can be expanded to be made more robust. Following our current examples, as FOCUS observes humans sitting in chairs in multiple different environments, each of these different visual features would be added to the list of features that identify a chair and greatly increase the performance and expressibility of the object models.

The FOCUS algorithm is general purpose to any visual data stream. The experiments in this paper were performed from a stationary viewpoint and from straight-line motion down a hallway. We are currently extending FOCUS to operate on a mobile robot platform which will perform active exploration to seek out multiple viewpoints from which to observe the object.

Acknowledgments*

This research was supported by the National Business Center (NBC) of the Department of the Interior (DOI) under a subcontract from SRI International. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, by the NBC, DOI, SRI or the US Government.

References

[1] J. Gibson, *The Ecological Approach to Visual Perception*. New Jersey, USA: Lawrence Erlbaum Associates, 1979.

[2] K. Murphy, A. Torralba, and W. Freeman, "Using the forest to see the trees: A graphical model relating features, objects and scenes," in *NIPS'03, Neural Information Processing Systems*, 2003.

[3] Darnell J. Moore and Irfan A. Essa and Monson H. Hayes III, "Exploiting human actions and object context for recognition tasks," in *Proceedings of the International Conference on Computer Vision, V1*, 1999, pp. 80–86.

[4] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Proc. of IEEE Conference on Computer Vision*, pp. 259–268, 8-11 1987.

[5] A. Blake and M. Isard, *Active Contours*. Springer-Verlag, 1998.

[6] E. D. Dickmanns and B. D. Mysliwetz, "Recursive 3-d road and relative ego-state recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 199–213, February 1992.

[7] M. Isard and A. Blake, "Condensation: conditional density propagation for visual tracking," *International journal of Computer Vision*, 1998.

[8] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object detection," in *International Conference on Computer Vision (ICCV'95)*, Cambridge, USA, June 1995, pp. 786–793. [Online]. Available: citeseer.ist.psu.edu/moghaddam95probabilistic.html

[9] L. Stark and K. Bowyer, "Generic recognition through qualitative reasoning about 3-d shape and object function," in *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, Maui, HI, 1991, pp. 251–256.

[10] E. Rivlin, S. J. Dickinson, and A. Rosenfeld, "Recognition by functional parts," *Computer Vision and Image Understanding: CVIU*, vol. 62, no. 2, pp. 164–176, 1995. [Online]. Available: citeseer.ist.psu.edu/article/rivlin95recognition.html

[11] M. Brand, "Physics-based visual understanding," *Computer Vision and Image Understanding: CVIU*, vol. 65, no. 2, pp. 192–205, 1997. [Online]. Available: citeseer.ist.psu.edu/brand96physicsbased.html

[12] A. M. Arsenio, "Object recognition from multiple percepts," in *IEEE-RAS/RSJ International Conference on Humanoid Robots*, Los Angeles, USA, November 2004.

[13] P. Fitzpatrick, "Object lesson: discovering and learning to recognize objects," in *IEEE International Conference on Humanoid Robots*, Karlsruhe and Munich, Germany, September/October 2003.

[14] M. Ehrenmass, R. Zöllner, O. Rogalla, S. Vacek, and R. Dillmann, "Observation in programming by demonstration: Training and execution environment," in *IEEE International Conference on Humanoid Robots*, Karlsruhe and Munich, Germany, September/October 2003.

[15] D. Bentivegna, C. Atkeson, and G. Cheng, "Learning from observation and practice at the action generation level," in *IEEE International Conference on Humanoid Robots*, Karlsruhe and Munich, Germany, September/October 2003.

[16] M. Minsky, *Society of Mind*. Simon & Schuster, March 1988.

[17] F. D. la Torre Frade, C. Vallespi, P. Rybski, M. Veloso, and T. Kanade, "Learning to track multiple people in omnidirectional video," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, May 2005.

[18] P. E. Rybski and M. M. Veloso, "Using sparse visual data to model human activities in meetings," in *Workshop on Modeling Other Agents from Observations, International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2004.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, 2004.

[20] F. von Hundelshausen and R. Rojas, "Tracking regions and edges by shrinking and growing," in *Proceedings of the RoboCup 2003 International Symposium, Padova, Italy*, 2003.

[21] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors." in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, vol. 2, 2004, pp. 506–513.