“Look at this!” Learning to Guide Visual Saliency in Human-Robot Interaction

Boris Schauerte¹ and Rainer Stiefelhagen¹

Abstract— We learn to direct computational visual attention in multimodal (i.e., pointing gestures and spoken references) human-robot interaction. For this purpose, we train a conditional random field to integrate features that reflect low-level visual saliency, the likelihood of salient objects, the probability that a given pixel is pointed at, and — if available — spoken information about the target object’s visual appearance. As such, this work integrates several of our ideas and approaches, ranging from multi-scale spectral saliency detection, spatially debiased salient object detection, computational attention in human-robot interaction to learning robust color term models. We demonstrate that this machine learning driven integration outperforms the previously reported results on two datasets, one dataset without and one with spoken object references. In summary, for automatically detected pointing gestures and automatically extracted object references, our approach improves the rate at which the correct object is included in the initial focus of attention by 10.37% in the absence and 25.21% in the presence of spoken target object information.

I. INTRODUCTION

Verbal and non-verbal signals that guide our attention are an essential aspect of natural interaction and helps to establish a joint focus of attention (e.g., [1–6]). The ability to generate and respond to such signals allows to establish a common point of reference or conversational domain with an interaction partner, which is fundamental for “learning, language, and sophisticated social competencies” [7].

When talking about the focus of attention (FoA), we have to distinguish between the focus of attention within the conversation domain (i.e., what people are talking about), and the perceptual focus of attention (e.g., where people are looking at). In many situations, the conversational FoA and the perceptual FoA are distinct. However, when persons are referring to specific objects within a shared spatial environment, multimodal – here, non-verbal and verbal – references are an important part of natural communication to direct the perceptual FoA toward the “referent”, i.e. the referred-to object, and achieve a shared conversational FoA. Accordingly, we have to distinguish between the saliency of objects in the context of the conversation domain at some point during the interaction and the inherent, perceptual saliency of objects present in the scene (see [8]). Although the conversational domain is most important when identifying the referent – especially when considering object relations –, the perceptual saliency can influence the generation and interpretation of multimodal referring acts to such extent that in some situations “listeners […] identify objects on the basis of ambiguous references by choosing the object that was perceptually most salient” [8, 9].

We focus on situations in which an interacting person uses non-verbal (i.e., pointing gestures) and verbal (i.e., spoken object descriptions) signals to direct the attention of an interaction partner toward a target object. Consequently, our task is to highlight and hopefully correctly identify the intended target object. One of the challenges in such a situation is that we may have none or just very limited knowledge about the target object’s appearance. In fact, it might be the actual goal of the multimodal reference to teach something about the referent; for example, imagine a pointing gesture that is accompanied by “Look at this! Have you ever seen Razzmatazz¹ before?” Interestingly, exactly in situations in which we know nothing about the target object’s visual appearance, we can use modern visual saliency models to determine image regions that are highly likely to render potential objects of interest. And, as we have mentioned before, the actual multimodal reference itself might even be influenced by the perceptual saliency. Furthermore, some target information (e.g., information about the target’s color) is known to subconsciously influence our visual attention system and accordingly we would like to integrate such information as well, if it is available.

Extending our previous work [10, 11], we learn to high-

¹Boris Schauerte and Rainer Stiefelhagen are with the Institute for Anthropomatics, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany (boris.schauerte, rainer.stiefelhagen}@kit.edu

¹Among other things, Razzmatazz is a crayon color name that describes a shade of rose or crimson. Look at bit.ly/1nJv5Ut
light and, in most cases, successfully determine the referent in multimodal – here, pointing gestures and spoken references – human-robot interaction. For this purpose, we propose to use CRFs to integrate different features that represent the perceptual saliency, the spatial information provided by pointing gestures, and information about the target object’s visual appearance. This way, we can calculate a saliency map that highlights potential target object regions and seamlessly integrates the available multimodal information, see Fig. 1 and Fig. 2. Compared to our previous approaches that have a stronger psychological and/or biological motivation and did not rely on machine learning, we are able to improve our ability to focus the intended target object in the initial focus of attention by 10% and 25% without and with, respectively, information about the target’s visual appearance.

II. RELATED WORK

Since our work is related to a series of different research topics (e.g., visual attention, gesture recognition, human-robot interaction), we can only provide a brief overview.

A. Joint Attention

Establishing a joint focus of attention describes the human ability to verbally and non-verbally coordinate the focus of attention with interaction partners. On one side this is achieved by directing the attention towards interesting objects, persons, or events, and on the other side by responding to these attention directing signals. Since this ability is one of the most important aspects of natural communication and social interaction, it has been addressed in various research areas, most importantly: psychology (e.g., [1, 7, 12]), computational linguistics (e.g., [6]), computer vision (e.g., [13]), and robotics (e.g., [6, 14–17]). And, consequently, initiating (e.g., [6, 16, 18]) and responding to joint attention signals (RJA; e.g., [13, 14, 16]) are crucial tasks for social robots.

In this contribution, we address a specific aspect of RJA, i.e. how verbal object descriptions and pointing gestures can influence perceptual saliency and thus visual search.

1) Pointing: Pointing gestures form a non-verbal signal that directs the attention along a spatial corridor – originating from the pointing origin, along the indicated pointing direction – towards the indicated spatial region to establish a joint focus of attention (see, e.g., [1, 3, 4, 12, 19]). Accordingly, the ability to recognize pointing gestures and infer the referred-to object (i.e., the “referent”), the coarse spatial target region, or just the indicated direction has been addressed by several researchers in the context of robotics (e.g., [10, 11, 19–24]). Unfortunately, most of these systems assume that the objects in the scene are already detected, segmented, recognized, categorized, and/or their attributes identified. In contrast, our approach relies on saliency – you can think of it as a generalized object detector (see, e.g., [25, 26]) – to direct the attention towards the referent without or with only very limited knowledge about the actual objects in the environment. Non-verbal signals such as pointing gestures circumscribe a referential domain to direct the attention towards an approximate spatial region (see [1]). Naturally, this can clearly identify the referent in simple, non-ambiguous situations (see, e.g., Fig. 2). However, as pointing gestures are inherently inaccurate in ambiguous situations (see [2, 20]), context knowledge may be necessary to clearly identify the referent (see [10, 12, 16]).

2) Language and Color Terms: Language can provide contextual knowledge about the referent such as, e.g., spatial relations and information about the object’s visual appearance (see, e.g., [10, 27]). Here, verbal and non-verbal references can be seen to form composite signals, i.e. the speaker will compensate the inaccuracy or ambiguity of one signal with the other (see [1, 3, 4, 8, 12, 16, 20]).

Verbal references can use relative and absolute features to describe the referent (see [8]). Relative features require reference entities (e.g., “the left cup”, or “the big one”), whereas absolute features do not require reference objects (e.g., “the red cup”). Possibly the most fundamental absolute properties of an object are its name, class, and color. When verbally referring-to color, color terms (e.g. “green”, “dark blue”, or “yellow-green”) are used in order to describe the perceived color (see [28]). Here, it is important to know about the cross-cultural concept of universal “basic color terms” [29], i.e. that there exists a limited set of basic color terms in each language of which all other colors are variants.

B. Saliency

To calculate our target saliency maps, we integrate visual saliency algorithms that have been developed with the goal to predict “interesting” image regions that attract human gaze (e.g., [30, 31]) and salient object detection methods that try to identify and segment the most prominent object in an image (e.g., [32–34]). Interestingly, our method is able to accurately segment the target object in most “simple” situations, see Fig. 2, whereas it can only predict salient regions of interest in complex situations, see Fig. 1.

1) Visual saliency: Following the assumption that interesting objects are visually salient (see [26]), computational attention models (see [35]) can be used to focus complex...
processes onto potentially relevant, thus salient, sensor data. Recently, spectral visual saliency models that exploit the image’s frequency spectrum [30, 31, 36, 37] have attracted an increasing interest, especially in applied fields such as robotics (e.g., [11, 38]), due to their outstanding computational efficiency. These models exploit that magnitude suppression highlights sparse salient regions [30, 39].

Inspired by Treisman et al.’s work [40], most attention models calculate saliency maps to predict bottom-up gaze patterns and/or visual search behavior. Here, visual search refers to the task to search for specific stimuli in scenes. Interestingly, knowledge about specific aspects of the search target can influence the perceptual saliency to highlight the target, thus speeding-up visual search (see [41, 42]). However, not every piece of knowledge can influence the perceptual saliency. Instead, only specific information that refers to preattentive features allows such guidance [42]. For example, knowing the specific object or at least its color reduces the search slope, whereas categorical information typically does not provide top-down guidance (see [42]). In recent years, various computational saliency models have been developed that try to integrate knowledge to guide the perceptual saliency (e.g., [27, 43–47]). Most related to this paper, Navalpakkam and Itti [45] modulate feature map weights in such way that the expected signal-to-noise, i.e., target-to-distractor, ratio of the saliency combination across and within all feature dimensions is maximized. Schauerte and Fink [10] adapted Navalpakkam’s model to use spectral saliency as contrast operator on each individual feature map.

2) Salient object detection: Without going into any detail, salient object detection refers to the task to identify the most prominent object in an image [34, 48]. However, our work is different from most work on salient object detection: First, we try to integrate top-down information such as pointing gestures and language. Second, our images are not center-biased [32]. Third, our target objects are substantially smaller compared to salient objects in web images (see [34, 48]).

III. METHOD

A. Conditional Random Field: Structure, Train, Predict

CRFs model the conditional probabilities of \( x \) (here, “does this pixel belong to a target object?”), given the observation \( y \) (i.e., features), i.e.,

\[
p(x | y) = \frac{1}{Z(y)} \prod_{c \in C} \psi(x_c, y) \prod_{i \in V} \psi(x_i, y) ,
\]

where \( C \) is the set of cliques in the CRF’s graph and \( i \) represents individual nodes. \( \psi \) indicates that the value for a particular configuration \( x_c \) depends on the input \( y \).

Naturally, we address a binary segmentation task, because the location depicted by a pixel can either belong to the target object or not, i.e., \( x_i \) can either be “target” or “background”. We use a pairwise, 4-connected grid CRF structure and linearly parametrize the CRF parameter vector \( \Theta \) in unary node \( \psi(y, i) \) and edge features \( \psi(x_i, j) \). The former feature type represents information at an image location (e.g., image intensity) while the latter relates neighbored image locations (e.g., feature value differences). Here, it is important to consider that the cliques in a 4-connected, grid-structured graph are the sets of connected nodes, which are represented by the edges. Consequently, we fit two matrices \( F \) and \( G \)

\[
\Theta(x_i) = F \psi(y, i) \quad (2)
\]

\[
\Theta(x_i, x_j) = G \psi(y, i, j) . \quad (3)
\]

Here, \( y \) is the observed image and \( \Theta(x_i) \) represents the parameter values for all values of \( x_i \). Similarly, \( \Theta(x_i, x_j) \) represents the parameter values for all \( x_i, x_j \). Then, we can calculate

\[
p(x; \Theta) = \exp \left[ \sum_i \Theta(x_i) + \sum_j \Theta(x_i, x_j) - A(\Theta) \right] \quad (4)
\]

where \( A(\Theta) \) is the log-partition function that ensures normalization.

We use tree-reweighted belief propagation (TRW) to perform approximate marginal inference [49]. TRW addresses the computational intractability of the exact log-partition function \( A(\Theta) \) and thus approximates \( A(\Theta) \) with

\[
\hat{A}(\Theta) = \max_{\mu \in \mathcal{L}} \Theta \cdot \mu + \hat{H}(\mu) , \quad (5)
\]

where \( \hat{H} \) is TRW’s entropy approximation [49]. Here, \( \mathcal{L} \) denotes the valid set of marginal vectors

\[
\mathcal{L} = \{ \mu : \sum_{x \in i} \mu(x_c) = \mu(x_i) \land \sum_x \mu(x_i) = 1 \} , \quad (6)
\]

where \( \mu \) describes a mean vector, equaling a gradient of the log-partition function. The approximate marginals \( \hat{\mu} \) are the maximizing vector

\[
\hat{\mu} = \arg \max_{\mu \in \mathcal{L}} \Theta \cdot \mu + \hat{H}(\mu) . \quad (7)
\]

This can be approached iteratively until convergence or a maximum number of updates [50].

To train the CRF, we rely on the clique loss function [49]

\[
L(\Theta, x) = -\sum_c \log \hat{\mu}(x_c; \Theta) , \quad (8)
\]

where \( \hat{\mu} \) indicates that the loss is implicitly defined with respect to TRW’s marginal predictions and not the true marginals. This loss can be seen as empirical risk minimization of the mean Kullback-Leibler divergence of the true clique marginals to the predicted ones.

B. Features

As primitive unary image-based features, we include the following information in the feature vector at each CRF grid point: First, we include each pixel’s normalized horizontal and vertical image position. Second, the image intensity, scaled to the CRF’s grid size. Furthermore:
1) Multi-scale Spectral Saliency: As a low-level, bottom-up visual saliency feature to highlight general regions of interest, we use Schauerte and Stiefelhagen’s multi-scale quaternion discrete cosine transform (QDCT) spectral image signature saliency [31] with CIE Lab as color space. For this purpose, after scaling each saliency map to the CRF grid size, we append QDCT image signature saliency maps that we calculate at three scales (96 × 64 px, 168 × 128 px, and 256 × 192 px).

2) Locally Debiased Region Contrast: Furthermore, we use Schauerte and Stiefelhagen’s locally debiased region contrast saliency (LDRC) [32] as additional low-level feature to highlight object-like areas of interest. LDRC is an adapted salient object detection algorithm that has the advantage that it does not have an intrinsic center-bias. Such a target object location center-bias is common in web images [32], where – as a consequence – the integration of a center-bias model is beneficial. However, since the object locations in our datasets are not center-biased, see Fig. 1 and 2, LDRC has proven to be advantageous for our intended application. While QDCT processes the image as a whole – highlighting sparse salient regions –, LDRC uses the sizes, spatial distances, and color distances of image segments to calculate the saliency. In our implementation, we use Felzenszwalb’s image segmentation method [51].

3) Probabilistic Pointing Cone: To model that pointing gestures define a spatial corridor of attention, Schauerte, Richarz, and Fink introduced the probabilistic pointing cone (PPC) [10, 11]. The PPC models the likelihood that an image pixel is being pointed at, given a pointing origin (i.e., typically the pointing finger or hand), the pointing direction (i.e., typically the direction defined by the eye-hand line [21, 52]), and an estimate of the pointing gestures inaccuracy or imprecision [2, 20, 52]. This inaccuracy can be caused by several factors such as, most importantly, the inherent (in-)accuracy of the performed gesture, the method to infer the pointing direction, and the automatic pointing gesture detection itself.

We calculate the resulting map as defined by Schauerte, Richarz, and Fink [10, 11]

\[ p_{PPC}(x) = p(\alpha(x, o)|d, o) \sim \mathcal{N}(0, \sigma^2_{\alpha}), \]  

with \( \alpha(x, o) \) being the angle between the vector from the pointing origin \( o \) to the image point \( x \) given the pointing direction \( d \), and the estimated inaccuracy \( \sigma^2_{\alpha} \). This equation represents the probability that a point \( x \) in the image plane was referred-to by the pointing gesture and thus defines our spatial corridor of attention. To account for the findings by Kranstedt et al. [20], we further enforce a lower bound of \( 3^\circ \), i.e., \( \sigma_{\alpha} = \max(3^\circ, \sigma_{\alpha}) \), so that 99.7 %, which corresponds to 3 \( \sigma \), of the distribution’s probability mass covers at least a corridor of 9\(^{\circ}\).

4) Target Feature Map: Language often provides the discriminating context to identify the referent amidst other potential target objects. Most importantly, it is used to specify objects (e.g., “my Ardbeg whisky package”), classes (e.g., “whisky package”), visually deducible attributes (e.g., “red”, or “big”), and/or relations (e.g., “the cup on that table”). To integrate the influence of knowledge about the target object’s visual appearance, which can be part of a verbal object reference, we build on the same target features as Schauerte and Fink [10]. Here, Schauerte and Fink focused on color information either in the form of color term models or object-specific models. This is motivated by the fact that only specific preattentive features have been shown to provide top-down attentional guidance (see [42]). In contrast to Schauerte and Fink [10], we do not use a neuron-based model with a signal-to-noise target function and instead let the CRF learn to integrate these features. This, in general, makes it very simple to specialize and extend our model with further, more specific features (e.g., faces [53]) as long as we are able to specify a target feature map.

a) Target Color: To model the influence of color terms (e.g., “red”), we use probabilistic color term models \( p(\theta|T_{\text{color}}) \) that have been learned using the Google-512 data set (see [54, 55]), which was gathered from the Internet for the 11 English basic color terms. As feature, we calculate \( p(x||T_{\text{color}}) \) for each image pixel \( x \) and include the scaled probability map.

b) Target Object: To model the influence that the reference to a specific, known object can have (e.g., the “Hobbits cookies package” or “red bull can”), we can calculate object-specific target feature models \( p(\theta||T_{\text{obj}}) \). Such a target feature model can either be a part of memory or a database, or we can calculate the target model based on available images of the target object. Again, we calculate \( p(x||T_{\text{obj}}) \) for each image pixel \( x \) and, after scaling to the CRF dimensions, include the probability map as feature.

Given the close-up object views that are part of the ReferAt dataset’s object database, we exploit that the target objects are usually well-centered in the model views and use the color spatial variance – i.e., a known salient object detection feature [34] – to perform a foreground/background separation. Additionally, the acquired segmentation mask is dilated to suppress noise and omit background pixels around the object boundaries. Then, we calculate \( p(\theta|T_{\text{obj}}) \) as the color distribution of the foreground image pixels. If we have access to multiple views of the same object, we use a uniform combination to combine the models. Furthermore, we slightly smooth the model to generalize and improve the robustness against slight illumination changes.

IV. EVALUATION

A. Evaluation Measures

The fovea is responsible for detailed, sharp central vision. As such, it is essential for all tasks that require visual details such as, most importantly, many recognition tasks. The fovea itself comprises of less than 1 % of the retinal area and perceives the central 2\(^{\circ}\) of the visual field. In our evaluation, we want to simulate attentional shifts to identify the most likely target object locations. We define that an object has been perceived, if it or a part of it has been projected onto the fovea. The assumption of an FoA area has another important benefit when we work with saliency models: Since saliency
models tend to highlight edges, the most salient point in an image is often related to object boundaries and as a consequence can be located just a bit outside of the actual object, very close to the boundary.

We derive three FoA related evaluation measures: The pixel hit rate (PHR) measures how often the most salient pixel lies within the object boundaries (see [32]). The focus of attention hit rate (FHR) measures how often the object is covered – and thus perceived – at least partially by the (initial) FoA (see [10, 11]). To compute the FoA hit rate (FHR), we calculate whether the radial FoA and the annotated object’s boundary polygon collide. Additionally, we can calculate the FHR after shifting the focus of attention to the next most salient region in the image. To this end, we inhibit the location that has already been attended, i.e. we set the saliency of all pixels within the current FoA to zero. Let $FHR_{+k}$ denote the FoA hit rate after $k$ attentional shifts, i.e. how frequently the target is perceived within the first $k$ shifts of attention. Then, we can integrate over the $FHR_{+k}$ until a given $k \leq n$ (in the following, we set $n = 10$). We refer to this measure as $\int FHR$ and it has the advantage that it also reflects the cases in which the target object has not been found after $n$ shifts$^2$.

B. Procedure and Algorithm Parameters

To train and evaluate our CRF models, we use a leave-one-person-out training procedure. Furthermore, we mirror the samples along the vertical axis to double the available image data. The CRF is trained with a grid resolution of $381 \times 284$ and a 4-connected neighborhood.

C. Baseline Algorithms

In the absence of spoken target information (Sec. IV-D, PointAt dataset), we use the heuristic model by Schauerte and Fink as a baseline [11]. Since we relabeled the PointAt dataset$^3$, we calculate and present new baseline results. For this purpose, we use a different visual saliency model: QDCT image signature saliency [31] with CIE Lab as color space.

In the presence of spoken target information (Sec. IV-E, ReferAt dataset), we use Schauerte and Fink’s neuron-based model as baseline [10]. Here, we rely on the previously reported FHR results, which are comparable to ours. Since the performance of the CRF is a substantially better than the neuron-based model (the CRF’s FHR is often higher than the neuron-based model’s FHR), we refrain from (re-)evaluating the neuron-based model to calculate PHR and $\int FHR$.

D. Evaluation: Pointing

1) Dataset: To assess the ability of systems to identify arbitrary point-at target objects, Schauerte, Richarz, and Fink collected a dataset (PointAt) that contains 220 images of 3 persons pointing at various objects [11]. The dataset was recorded in two environments with a large set of objects of different category, shape, and texture.

2) Hardware Setup: The dataset was recorded using a monocular pan-tilt-zoom camera. The camera provides images at a resolution of $762 \times 568$ px, offers an optical zoom of up to $\times 18$, and a wide horizontal opening angle of $48^\circ$. To reflect a human or humanoid point of view, the camera was mounted on eye height of an averagely tall human [11].

3) Procedure: Each person performed several pointing sequences, with varying numbers and types of objects present in the scene. Here, neither the subjects’ body postures nor the subject or object positions have been pre-defined or restricted. However, to comply with the line-of-sight, the subjects were instructed to point with their arms extruded. Accordingly, the dataset contains a wide variety of pointing references, see Fig. 2. In total, the dataset contains 220 pointing references.

For each object reference, the following aspects have been manually annotated: each target object’s boundaries$^4$, the dominant eye, the pointing finger, and the resulting pointing direction. The last three items make it possible to assess how the automatic pointing gesture recognition’s quality influences the identification of the pointed-at object.

4) Pointing Gestures: The pointing gesture detection relied on a histogram of oriented gradients (HOG) based head-shoulder detector and a hand detection, see [10, 11, 24]. Therefore, the inherent holding phase of the pointing hand was detected to detect the occurrence of a pointing gesture. For this purpose, the origin $o_t$ (i.e., hands) and direction $d_t$ (calculated from the head-shoulder area and the detected hands) hypotheses are clustered over time and large temporal clusters indicate the presence of a pointing gesture (see [11]).

5) Properties: On average the target object occupies only 0.58% of the image area. The average differences between the annotated and automatically determined pointing origin and direction are 15.30 px and 2.95°, respectively. The former is mostly caused by the fact that the system detects the center of the hand, instead of the finger. The latter is due to the fact that the eye positions are estimated given the head-shoulder detection (see [24]), and that the bias introduced by the dominant eye is unaccounted for (see [52]). In some cases, the ray that originates from the pointing origin and follows the pointed direction does not intersect the target object’s boundaries, it “misses” the object. The rate of how often the object’s annotated boundary polygon is missed by the pointing ray is 5.66% for the annotated pointing information and 19.34% for the automatic detection.

6) Results: As can be seen in Tab. I, CRFs provide a better predictive performance than the heuristic baseline method. Most interestingly, the CRF that we trained and tested with automatic gesture detections is able to outperform the heuristic method even if the latter relies on groundtruth information. This shows that the CRF model is better capable to compensate for the inaccuracy of automatically detected pointing gestures. Accordingly, if we compare the performance of both methods with detected and annotated point-
ing information, we can see that the relative performance difference of the heuristic model is much higher than the performance difference for the CRF. Furthermore, we can see that LDRC in addition to our spectral features helps us to improve the results of the overall approach.

To serve as a baseline, human observers were asked to guess the pointed-at object and they were able to estimate the correct object for about 87% of the images [11]. Accordingly, we can see that our model is able to come close to this baseline in terms of PHR. However, we can also see that the FHR is in fact higher than those 87%. How can that be? Most importantly, in ambiguous situations in which two potential target objects stand close to each other, the predicted target object location tends to be between both objects or just on the point of a border of one object that is closest to the other object. Thus, the target object might not have been selected by the most salient pixel, but at least a part of it is nevertheless visible in the assumed FoA.

E. Evaluation: Pointing and Language

To evaluate how well multimodal – here, pointing gestures and spoken language – references guide computational attention models, Schauerte and Fink collected a dataset (ReferAt) which contains 242 multimodal referring acts that were performed by 5 persons referring to a set of 28 objects in a meeting room [10], see Fig. 1. This limited set of objects defines a shared context of objects that are plausible in the scene and can be addressed. The objects were chosen from a limited set of classes with similar intra-class attributes, i.e. size and shape. Consequently, in most situations, object names and colors are the most discriminant verbal cues for referring-to the referent. The limited number of classes further forces the participant to use specifiers to address the objects, because the object class alone would almost always lead to ambiguous references.

1) Hardware Setup: The hardware setup is identical to the PointAt dataset’s hardware setup, see Sec. IV-D.2.

2) Procedure: To create a challenging dataset, participants were allowed to freely change their own position as well as select and (re-)arrange the objects in the scene, see Fig. 1. Furthermore, the task (i.e., determine the right target object) was explained to the participants and they were actively encouraged to create complex situations. However, again participants were asked to point with their arms extruded to comply with the line-of-sight [1, 24]; it is not the intended goal of the dataset to evaluate different methods to determine the pointing direction (cf. [21]). To verbally refer to an object, the participants were allowed to use arbitrary sentences. All spoken references have been transcribed to avoid the influence of speech recognition errors. Then, for each linguistic reference, the following items have been annotated: the target object, the spoken as well as each object’s attributes, and whether the specific target object can be recognized without the visual context of the complementary pointing gesture. As for the PointAt dataset, the dominant eye, the pointing finger, and the resulting pointing direction have been manually annotated.

3) Pointing Gestures: The pointing gesture detection is identical to the PointAt method, see Sec. IV-D.4.

4) Object Database: To make it possible to highlight a known target object and to evaluate the benefit of active vision on object recognition, a database was collected that contains images of all objects that have been referenced in the dataset. For each object there exists at least one close-up view, which is linked with its verbal object specification. Interestingly, this database was created interactively using the proposed system itself in a training mode (see [10, 11]).

5) Dataset Properties: On average the target object occupies 0.70% of the image area⁴ (compare Sec. IV-D.5). The average differences between the annotated and automatically determined pointing origin and direction are 12.50 px and 3.92˚, respectively. The rate of how often the object’s annotated boundary polygon is missed by the pointing ray is 7.85% for the annotated pointing information and 26.03% for the automatic detection.

6) Spoken Reference Detection: To automatically determine spoken object or object attribute references, Schauerte and Fink trained a Brill tagger that identifies different part-of-speech tags [56]. These tags were then used to determine the noun-phrases and their constituents with a shallow parser. Then, the occurrence of color terms is determined via keyword spotting. To determine references to specific, known objects, an edit distance is calculated to measure the similarity of the spoken description to each object’s specifications in the database. The focus of this matching is to avoid misdetections (i.e., false positives), because wrong target information would highlight the wrong image areas, leading to very inefficient search paths. This way, the language processing correctly detected 123 of 123 color references and 123 of 143 references to specific objects.

7) Results: We present the results for different target information conditions in separate tables. Tab. II(a) shows the results without any linguistic target information, Tab. II(c) provides the results achieved with groundtruth target information, and Tab. II(b) contains the results obtained with automatically determined target object information. Each table presents the results achieved without pointing information.

⁴We would like to note that such small target sizes stand in contrast to traditional salient object detection tasks (see, e.g., [32]), in which the salient objects cover a substantial part of the image.
close objects in which the target object is contained, which definition. But, it often highlights small clusters of spatially dense cluster of distractors, which is not surprising given its LDRC seems to be unable to highlight a single object in a and can most likely be explained by the fact that PointAt is integrated. This stands in contrast to our results on when LDRC is integrated, at least if pointing information done by Schauerte and Fink [10], then the integration of LDRC as a CRF feature clearly improves the results. However, the performance as quantified by PHR decreases of LDRC as a CRF feature clearly improves the results. Nevertheless, CRFs clearly outperform the neuron-based model, often by a more than 20% higher FHR. In fact, the performance of the CRFs with detected pointing information often even outperform the performance of the neuron-based model with groundtruth pointing information, although the use of detected pointing information leads to a substantial drop in the overall performance. The performance difference that is caused by the use groundtruth and detected pointing information can be explained by the imprecision of the detected pointing origin and pointing direction – see the dataset property discussion in Sec. IV-E.5 –, which often causes the pointing ray to miss the intended target object.

Since pointing gestures substantially limit the spatial area in which we expect target objects, it is intuitively clear that the integration of pointing gestures substantially improves the performance under all three target information conditions (i.e., no language, automatically extracted spoken target object information, and annotated target information, see Tab. II(a), II(b), and II(c), respectively; compare “no pointing” to “pointing detected” and “pointing annotated”).

The integration of language also substantially improves the performance on its own, i.e., without accompanying pointing gestures (compare Tab. II(a) to Tab. II(b) and II(c)). Again, we see that LDRC leads to different FHR and PHR trends.

Finally, the combination of both modalities leads to further improvements compared to each unimodal result, i.e., speech or pointing gestures, which confirms that the modalities complement each other.

V. CONCLUSION

We have shown that CRFs can be used successfully to guide the visual saliency and thus a robot’s attention in the presence of pointing gestures and spoken target object information. For this purpose, we use a CRF to integrate a set of relevant features – multi-scale spectral saliency, salient object detection, probabilistic pointing cone, and probabilistic target maps – and learn to highlight image regions that are highly likely to contain the intended target object. We demonstrated effectiveness of the proposed approach on two datasets, one without and one with spoken target object descriptions. We improved the performance, quantified by the FHR, up to roughly 10% on the former dataset and 25% on the latter.

Table I: Target object detection on the ReferAt dataset.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>No language</th>
<th>Detected language</th>
<th>Groundtruth language</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PHR</td>
<td>FHR</td>
<td>PHR</td>
</tr>
<tr>
<td></td>
<td>FHR</td>
<td></td>
<td>FHR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neuron-based [10] no LDRC</td>
<td>15.70%</td>
<td>0.467</td>
<td>5.47%</td>
</tr>
<tr>
<td>CRF, no LDRC</td>
<td>24.38%</td>
<td>0.487</td>
<td>5.57%</td>
</tr>
<tr>
<td>CRF, w/ LDRC</td>
<td>24.79%</td>
<td>0.476</td>
<td>5.57%</td>
</tr>
<tr>
<td>Neuron-based [10] pointing detected</td>
<td>50.00%</td>
<td>0.815</td>
<td>65.20%</td>
</tr>
<tr>
<td>CRF, no LDRC</td>
<td>55.37%</td>
<td>0.803</td>
<td>66.12%</td>
</tr>
<tr>
<td>CRF, w/ LDRC</td>
<td>52.48%</td>
<td>0.801</td>
<td>65.29%</td>
</tr>
<tr>
<td>Neuron-based [10] pointing annotated</td>
<td>65.20%</td>
<td>0.830</td>
<td>66.66%</td>
</tr>
<tr>
<td>CRF, no LDRC</td>
<td>66.12%</td>
<td>0.830</td>
<td>66.66%</td>
</tr>
<tr>
<td>CRF, w/ LDRC</td>
<td>65.29%</td>
<td>0.834</td>
<td>66.66%</td>
</tr>
<tr>
<td>(a) without spoken target object information</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neuron-based [10] no LDRC</td>
<td>16.50%</td>
<td>0.424</td>
<td>6.65%</td>
</tr>
<tr>
<td>CRF, no LDRC</td>
<td>19.83%</td>
<td>0.458</td>
<td>6.65%</td>
</tr>
<tr>
<td>CRF, w/ LDRC</td>
<td>19.42%</td>
<td>0.452</td>
<td>6.65%</td>
</tr>
<tr>
<td>Neuron-based [10] pointing detected</td>
<td>54.10%</td>
<td>0.780</td>
<td>54.10%</td>
</tr>
<tr>
<td>CRF, no LDRC</td>
<td>54.13%</td>
<td>0.780</td>
<td>54.13%</td>
</tr>
<tr>
<td>CRF, w/ LDRC</td>
<td>47.93%</td>
<td>0.794</td>
<td>54.16%</td>
</tr>
<tr>
<td>Neuron-based [10] pointing annotated</td>
<td>59.90%</td>
<td>0.837</td>
<td>63.63%</td>
</tr>
<tr>
<td>CRF, no LDRC</td>
<td>63.63%</td>
<td>0.837</td>
<td>63.63%</td>
</tr>
<tr>
<td>CRF, w/ LDRC</td>
<td>60.74%</td>
<td>0.842</td>
<td>63.63%</td>
</tr>
<tr>
<td>(b) automatically determined spoken target object information</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neuron-based [10] no LDRC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRF, no LDRC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRF, w/ LDRC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neuron-based [10] pointing detected</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRF, no LDRC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRF, w/ LDRC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neuron-based [10] pointing annotated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRF, no LDRC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRF, w/ LDRC</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

with detected pointing information, and with groundtruth pointing information.

If we use FHR as key evaluation measure – as has been done by Schauerte and Fink [10] –, then the integration of LDRC as a CRF feature clearly improves the results. However, the performance as quantified by PHR decreases when LDRC is integrated, at least if pointing information is integrated. This stands in contrast to our results on PointAt and can most likely be explained by the fact that LDRC seems to be unable to highlight a single object in a dense cluster of distractors, which is not surprising given its definition. But, it often highlights small clusters of spatially close objects in which the target object is contained, which increases FHR as it decreases PHR.

REFERENCES