Inferring door locations from a teammate's trajectory in stealth human-robot team operations

Jean Oh, Luis Navarro-Serment, Arne Suppé, Anthony Stentz and Martial Hebert[1]

Abstract— **Robot perception is generally viewed as the interpretation of data from various types of sensors such as cameras. In this paper, we study indirect perception where a robot can perceive new information by making inferences from non-visual observations of human teammates. As a proof-of-concept study, we specifically focus on a door detection problem in a stealth mission setting where a team operation must not be exposed to the visibility of the team's opponents. We use a special type of the Noisy-OR model known as BN2O model of Bayesian inference network to represent the inter-visibility and to infer the locations of the doors, *i.e.*, potential locations of the opponents. Experimental results on both synthetic data and real person tracking data achieve an F-measure of over .9 on average, suggesting further investigation on the use of non-visual perception in human-robot team operations.**

I. Introduction

The possibility of developing a robotic system that can co-operate with human teammates has been met with ever-increasing interest. One of the key challenges in working towards this goal lies in robot perception–a robot's ability to 1) recognize objects in an environment and 2) understand the semantic relationships between the team's task and those objects. For example, consider the following scenario where a team consisting of a robot and a human must "screen the back door of the building." This seemingly simple mission statement imposes several technical challenges in the perception and interpretation of contextual information, *e.g.*, a robot needs to be able to recognize regions and detect objects in its environment, and to be able to understand the language to match the symbols in the command–such as doors and buildings–with actual objects in the robot's environment [15], [2].

Robot perception is generally viewed as the interpretation of data from various types of sensors. For example, camera-based scene understanding approaches, such as hierarchical semantic labeling [11], can achieve good performance classifying regions into semantic categories such as sky, trees, road, and buildings. Vision-based approaches for object detection such as Deformable Part Models (DPM) [5] and Convolutional Neural Network (CNN) [9], [6] have performed well on benchmark data sets identifying individual objects.

In the screen-the-back-door example, however, the objects that are relevant to mission contexts can be outside the robot's current field of view, *e.g.*, a robot won't be able to see a back door until it physically faces the back of the building. Therefore, in addition to traditional perception approaches, our research also investigates how to use inputs that are generally not considered for the perception task. In the work presented in [15], for instance, a robot can hypothesize the parts of an environment that are beyond the robot's sensor range by utilizing language clues coming from its human teammate. In this paper, we investigate an approach that exploits a human teammate's actions–such as trajectories and movement speeds–to understand the environment. This approach is based on a few specific assumptions: First, we assume that humans try to make optimal decisions according to team objectives. Second, we assume that, through the use of proprioceptive sensors (*e.g.*, wearable devices), human trajectories and motions can be accurately measured from farther distances than with visual sensors, and from outside the robot's field of view.

An earlier work on the idea of using human teammates in robot perception was briefly introduced in [16]. The temporal update method used, however, was too domain-specific to represent general dependence relationships among variables, thus lacking flexibility of generalization. In this paper, we describe a principled approach for handling uncertainty in more generalized problem domains. Specifically, we represent the causal relationship between perception and action as a Bayesian inference network (Bayes Net).

As a proof-of-concept study, we focus on the door detection problem in a hostile environment, *e.g.*, in the cover-the-back-door scenario. In this context, a "door" can be interpreted as a possible egress for armed insurgents hiding inside the building. We develop a Bayes Net to represent the dependence relationship between human movements and inter-visibility to and from the doors. Specifically, we use a special type of the Noisy-OR model known as BN2O [8] to deal with the scalability of conditional probability tables of a Bayes Net.

We test our approach using both software simulation and actual person tracking using a laser line scanner. We show that the Bayesian approach significantly improves performance, achieving over .8 precision and .9 recall on synthetic problems. More importantly, our Bayes net representation is a general framework that can be applied to other types of inference-based perception problems beyond the door detection problem shown in this paper.

II. Related work

The majority of prior work on object recognition and scene understanding relies on visual perception. Vision-based techniques extract semantic information from natural images

The authors are affiliated with the [1]Robotics Institute at Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, U.S. {jeanoh,lenscmu,suppe,axs,hebert}@cs.cmu.edu

using various visual features that represent those images. Although recent work addresses recognizing nonparametric objects and scenes [22], much of vision-based prior work focuses on recognizing class-specific objects in specific task contexts, *e.g.*, face recognition [23], door detection for indoor robot navigation [12], [10], [1], [21], or recognizing door handles for manipulation [20].

Since our approach is to discover the rationale behind human teammate's actions (*e.g.*, to recognize visual cues that have caused a human teammate to move faster or slower at certain locations), it is related to the notion of inverse reinforcement learning [14] (or imitation learning) where a hidden reward function is learned from observed optimal behavior. Generally, rewards are defined as a funcion of state features that are observable. The main task is to learn the set of feature weights in a way that the observed behavior is optimally represented. By contrast, we aim to infer unobservable information about an environment by utilizing team objectives that are shared amongst team members.

Bayesian inference techniques have been used in computer vision, *e.g.*, to infer object categories by incorporating prior knowledge about a scene with new observations of visual features [4]. A generative model is also used for monitoring and analyzing human behavior in a visual surveillance task [18], assuming that there exists a model that represents causal relationships between visual features and behavior categories. Similarly with the notion of inverse reinforcement learning, this approach also aims to learn a model that can explain and predict people's behaviors. Various techniques developed for inverse reinforcement learning and behavior prediction can be applied for perception through inference, but few works have addressed the perception issue to date.

III. Problem definition

We use the following scenario to illustrate the target problem.

***Example** 1 (Cover the back door.):* Consider a team consisting of a robot and a human performing a military operation in a hostile environment. According to intelligence, armed insurgents are hiding in the vicinity of an urban street. The team is deployed to cover the buildings in the surrounding area, focusing on doors from which the insurgents may try to egress. This is a stealth operation, *i.e.*, the team makes an effort to remain undetected.

***Definition** 1 (visibility):* A position[1] (*destination*) is said to be "exposed" to another position (*source*) if a line of sight can be established from the source to the destination as illustrated in Figure 1. Here, the visibility from a source is bounded with a range, *i.e.*, a source fails to achieve visibility outside its range. If a destination's line of sight to a source is obstructed, it is said to be "shadowed" from that source.

***Definition** 2 (riskiness):* The *riskiness* of a position is a measure of its vulnerability to exposure, and is proportional to the number of sources to which it is exposed.

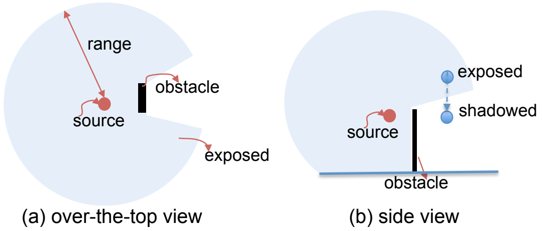


Fig. 1: An example of a source with a finite range

We assume that a human perceives the environment and subsequently chooses actions based on that information. For example, in the cover the back door scenario, a human will move at a faster speed when passing those positions that are exposed to doors than those that are shadowed. Here, both sources and destinations are defined at a constant height (*i.e.*, a human height). Based on this assumption, we formulate the door detection problem as an inference task as follows: Using the dependence relationship between the locations of doors and the riskiness of a position, the task is to infer the locations of doors by analyzing human movements observed over time.

IV. Perception through inference

This section describes the technical details of our approach.

A Bayesian Inference Network (*i.e.*, Bayes net) is a graphical model that succinctly represents causal relationships among variables [19]. When the values of some variables–referred to as evidence nodes–in the network are observed, the values of unobserved nodes can be inferred from those evidence nodes by updating conditional probabilities–*i.e.*, the probability of a variable having a certain value given the evidence.

We use a Bayes net to represent inter-visibility among positions as follows. First, we discretize the map of an area into *cells* such that a node is created per cell. We define two types of nodes: *source* and *destination*. Both source and destination nodes have binary values: the value of a source node indicates whether the cell represented by that node belongs to a door or not, whereas the value of a destination node specifies whether the corresponding cell is exposed to any sources. The default type of a node is a destination.

Next, we add a set of candidate doors as source nodes. We leverage building detection and prediction techniques developed in our earlier work [15] and identify candidate doors on the building façades. To detect, for instance, we use the technique based on hierarchical inference machines [11] to classify buildings and walls. When tested on a data set containing 500 outdoor images taken at a test facility in central Pennsylvania, the precision and recall rates for detecting buildings were 0.937 and 0.934, respectively [16]. Given that buildings and walls can be reliably detected or predicted, we localize the search for candidate doors to those cells on the buildings and walls. A source node is associated with the probability that the cell represented by that node is a door. In this paper, we use a constant prior probability for all candidate source nodes.

[1] A position is defined in terms of 3D-coordinates (x, y, z).

After creating the nodes, we add a set of edges to represent the causal relationships among nodes. For each source, using the line of sight inter-visibility calculation [7], we compute a set of destinations that are exposed to that source. These destinations are linked to the source as children through a directed edge from a parent to a child. Finally, a conditional probability table (CPT) is constructed for each node, representing the probability of a node having a certain value for every possible combination of parents' values.

In the case of destination nodes, the size of a node's conditional probability table grows exponentially in the number of parent nodes. Let m denote the average number of a node's parents in a network. When the set of candidates includes a large number of false positives, the Bayes net approach becomes infeasible. We can, however, utilize the domain specific knowledge such that the probability of a destination being exposed is conditioned only on the number of sources as opposed to all possible combinations of source values. This fact gives rise to a compact CPT representation that reduces its size from 2^m to $m + 1$.

This idea can be generalized to apply to a class of problems where the conditional probability of a child node can be defined as a function of the number of parents, known as the Noisy-OR model.

A. The Noisy-OR Belief Networks

In Bayesian networks, the number of entries in the conditional probability table is exponential in the number of parent nodes. The Noisy-OR Belief network was developed to overcome this scalability issue by exploiting the domain-specific causal structure of the nodes in the network [19]. The Noisy-OR generalizes the logical-OR, incorporating failure probabilities. The Noisy-OR model can be applied when the following condition is met: only those parents that have positive values (as opposed to negated value) have disjunctive influence on their children, modulo small errors. Then, the conditional probability of a child node d with n parent nodes, denoted by $s_1, ..., s_n$, can be represented by a Noisy-OR model as follows:

$$p(d|s_1, ..., s_k, \neg s_{k+1}, ..., \neg s_n) = \\ 1 - \prod_{i=1}^{k} p(\neg d|s_i) \prod_{i=k+1}^{n} \{1 - p(d|\neg s_i)\} \quad (1)$$

where $p(\neg d|s)$ and $p(d|\neg s)$ represent noise in the model, *e.g.*, both probabilities are 0 in the logical-OR. Thus, the size of CPT is reduced to the number of parents.

A class of Noisy-OR networks consisting of only two levels is known as BN2O. The BN2O model has been used to build Quick Medical Reference (QMR) medical knowledge base [3]. In the QMR example, parent nodes represent a set of possible causes (or diseases) whereas the child nodes specify observable symptoms. BN2O intuitively captures the causal dependence that a (child) symptom can be caused by one or more (parent) diseases.

B. Inter-visibility as BN2O

Figure 2 illustrates an example of BN2O representing inter-visibility. In this example, the destination node d_1 is exposed if at least one of its parent nodes, s_1 and s_2, is positive.

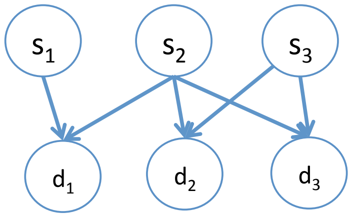


Fig. 2: Example: inter-visibility as a BN2O network where s and d denotes source and destination nodes, respectively.

In order to relax the assumption that a human teammate will always act optimally, two types of errors are included in the model. First, we use the *source error probability*, denoted by ϵ, such that a positive source would still miss a destination with probability ϵ. This corresponds to the noise probability $p(\neg d|s)$ in Equation 1. Second, we also include *leak probability*, denoted by λ, to specify the error rate representing the likelihood that a human may act as if exposed when in fact she is not; leak probability corresponds to $p(d|\neg s)$ in Equation 1. More generally, leak probability represents the probability of an event caused by something outside the model. Then, we rewrite the conditional probability of destination being exposed given its parents as the following:

$$p(d|s_1, ..., s_k, \neg s_{k+1}, ..., \neg s_n) = 1 - \epsilon^k (1 - \lambda)^{n-k}. \quad (2)$$

A prior model containing these two types of errors can be learned through tranining with the same teammate over time. For simplicity, we assume that we have learned the model by using constants for both types of errors.

Here, the source nodes are not directly observable but the values of destinations can be observed through a human teammate. For instance, the stealthiness of a human teammate on a cell, such as pose or velocity information, can indicate whether that cell is exposed or shadowed. In our experiment, we use velocity such that if a human's velocity on a cell falls below a certain threshold then negative evidence (*i.e.*, shadowed) is reported on that cell, and vice versa.

Since multiple–and possibly conflicting–observations can be made from the same cell, we update the observed value assuming that the observed samples follow a Bernoulli distribution–*i.e.*, a human moves fast with probability θ or move slowly with probability $(1 - \theta)$. We compute the expected value for each node from a sequence of observations made from the cell as follows. Let p and n denote positive and negative observations, respectively. Then the expected value of a node's observations is computed as: $\frac{p+1}{p+n+2}$ [24].

C. Likelihood-Weighted Sampling

We use a sampling-based inference method. Because the values of destination nodes depend on sources, we take the likelihood weighted forward sampling approach [8]. We first sample the values of the source nodes according to their probabilities, and then the values of destinations are selected based on the source values chosen. Here, instead of selecting a binary value as in the case of a source, we only keep the conditional probability for the destination nodes. The conditional probability is computed by using Equation 2. A resulting sample $\vec{a}$ is a vector whose i^{th} element, denoted by $\vec{a}[i]$, holds the probability of the i^{th} node being positive, where source nodes will have deterministic binary values.

The weight $w_{\vec{a}}$ of a sample $\vec{a}$ is computed as a product of all weight values in the sample, specifying the likelihood of the sampled event. When the value of an evidence node is observed its node index is sorted into positive or negative evidence sets E and $\neg E$, respectively. Given evidence sets E, $\neg E$, the weight of sample is computed as follows:

$$w_{\vec{a}} = \prod_{e \in E} \vec{a}[e] \prod_{\neg e \in \neg E} (1 - \vec{a}[e]).$$

Finally, the posterior probability of source s is computed by summing up the weights of those samples that have positive value for that source, normalized by the total sum of all weights as follows:

$$p(s|E, \neg E) = \frac{\sum_{\vec{a} \in A, \vec{a}[s]=1} w_{\vec{a}}}{\sum_{\vec{a} \in A} w_{\vec{a}}}.$$

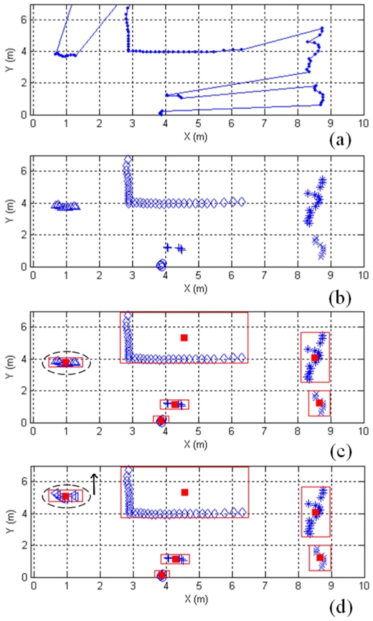
V. Experiments

This section describes how the robot collects observations from a human teammate, followed by experimental results. Here, we used a laser line scanner to track human teammates; in future work, we also plan to incorporate wearable sensors to track human teammates.

A. *Person detection and tracking*

We conducted a series of experiments where a person was tracked using a Hokuyo UTM-30LX 2D laser line scanner. This sensor produces a vector of 1,080 range measurements from consecutive bearings at $0.25\,^{\circ}$ intervals, called a *scan line*. A sample scan line is illustrated in Fig. 3-(a). Each scan line undergoes the processes of segmentation, tracking, and human detection. We use the detection and tracking algorithm described in [13].[2] The segmentation process groups the points in the scan line that are likely to belong to the same object (Fig. 3-(b)). It is assumed that two neighboring points belong to the same object if their separation is less than 0.8m. Each object is represented by the center of the bounding box enclosing the points that belong to it. Then, each object is tested for association with objects from previous scans by checking whether its bounding box overlaps with those of objects from past scans (Fig. 3-(c)). If there is a match, the motion of the center feature is used by a Kalman filter to estimate the object's position and velocity. These estimates are added to the history of the matching object, which is identified by a unique *Object ID*. If a match cannot be found, a new object history is created. Finally, each object is evaluated by computing its *Strength of Detection* (SOD), which is a measure of how confident the algorithm is that the object is actually a human. The SOD is a function of the object's size, the distance it has traveled, and the variation in its size and velocity. Details of the computation of the SOD are presented in [13].

For our experiments, we placed a set of obstacles whose average height was about 1.5m above the ground. We placed a sensor on a tripod at a fixed location 1m above the ground. This was done to generate a map of the testing area. Then, we increased the height of the sensor to 1.7m. This allowed us to track people in the area without occlusions from the obstacles.



(a)

(b)

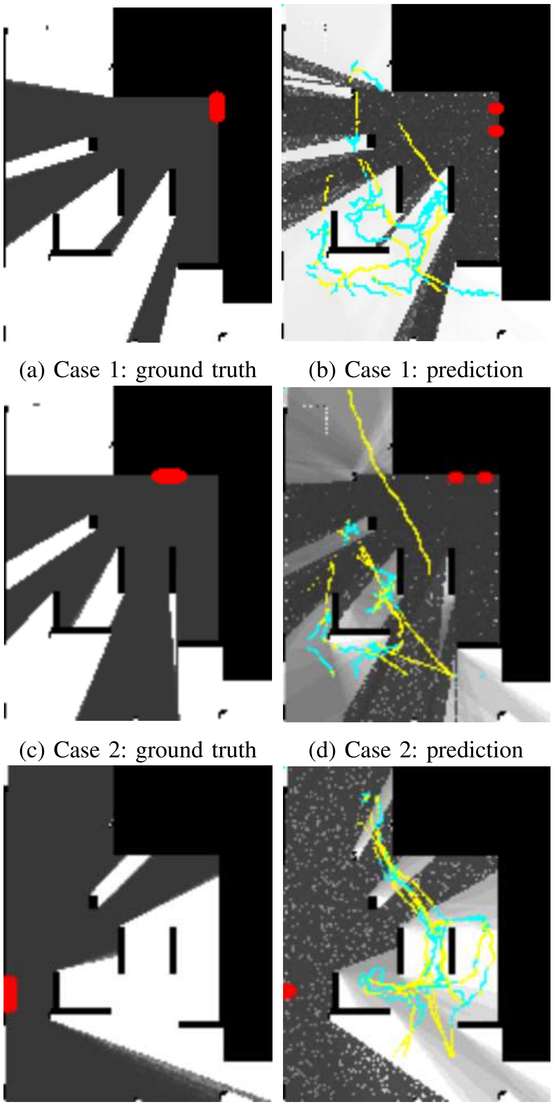
(c)

(d)

Fig. 3: A sensor placed at $(0, 0)$ produces a scan line (a). The segmentation step groups points into potential objects, as indicated by the sets of points with similar markers (b). The objects are represented by the center of their bounding boxes (c), whose motion is used for tracking (d).

The system keeps track of both static and moving objects. We filter out irrelevant observations according to the following criteria: first, objects whose SOD is smaller than a threshold θ_{sod} are rejected. Next, because the object ID of the same target can change (*e.g.*, when the tracking algorithm loses the target due to occlusion, and then assigns it a new object ID when the same object reappears), the tracking histories of the objects that passed the previous test are analyzed to determine if their histories should be merged. Therefore, the sequences of tracking data that start and end in close spatial and temporal proximity (within a distance threshold θ_d, and time threshold θ_t) to each other are joined together. All the thresholds were determined experimentally. For each object, every data point in its tracking history

[2] A lengthy description of the detection and tracking algorithm is outside of the scope of this paper. A comprehensive presentation of these steps is found in [13].



(a) Case 1: ground truth (b) Case 1: prediction

(c) Case 2: ground truth (d) Case 2: prediction

(e) Case 3: ground truth (f) Case 3: prediction

Fig. 4: Three scenarios: detecting doors by observing a human teammate. Static objects are shown in bounding boxes in black; human trajectory, in yellow and light blue; perceived doors, in a heatmap where red representing high confidence. (This paper is best viewed in color.) ($\epsilon = 0.1, \lambda = 0.1$)

is translated into an observation for the inference network. Here, if the human's moving velocity is higher than $0.5m/s$ then the cell is considered exposed; it is shadowed otherwise.

In the following two sections, we present the results using the following evaluation metrics.

Evaluation metrics. We use precision and recall to measure performance. Let tp, tn, fp, and fn denote true positive, true negative, false positive, and false negative, respectively. Precision and recall are defined as: $precision = \frac{tp}{tp+fp}$; $recall = \frac{tp}{tp+fn}$.

Instead of using a threshold to convert continuous output to a binary value, *e.g.*, a source is positive if its probability is higher than a certain threshold, we measure the performance using continuous values. For instance, the value of true positives tp is the sum of probabilities as opposed to the number of candidates whose probability values are higher than a threshold. This method, when compared to binary value counting, gives partial credit for correct predictions with low confidence (lower than the threshold used in binary value counting); at the same time, the score is penalized for correct predictions with high confidence.

B. *Results on tracking a person*

This set of experiments was carried out in a $12m \times 15m$ indoor space. A map was created by detecting static objects and walls using the laser line scanner. This map was used to compute inter-visibility when constructing a Bayes net.

For the purpose of this experiment, a human subject was instructed to navigate the area to monitor a door in a stealthy manner. The location of a ground truth door was configured differently for three cases, as shown in Figure 4. The left column shows the ground truth doors in red; the riskiness of a cell is expressed in shades (the darker, the riskier). The intial set of candidate doors included discretized points along the left wall as well as rotated L-shaped walls on the right. Thus, the algorithm initially predicted that the entire space was exposed to some sources. The right column shows the doors detected by our approach in a heat map form where red indicates strong positive. The human teammate's trajectory is also shown in the figure; yellow indicates that the human was moving fast whereas light blue corresponds to relatively slow movements. The actual locations of doors do not match precisely due to coarse resolution of discretization, but are fairly close, resulting in a riskiness map that closely resembles the ground truth.

C. *Results for tracking a software agent*

The following set of experiments were performed on synthetically generated scenarios similar to the real example shown in Figure 4. Each problem contained walls on both left and right sides and a set of random obstacles. The ground truth doors were also placed according to a configuration parameter specifying the percentage of a wall that constitutes doors. We then placed a human software agent having the ground truth information (where the doors are) to stealthily navigate through the area. The robot agent was given only information regarding where the walls were located. The robot agent initially predicted the entire wall (discretized) as candidate doors and assigned low confidence values. Using the human agent's positions and stealthiness modulo small errors, the robot agent updated the posterior probability of each source in the Bayes net.

Figure 5 shows that the robot improves its predictions over time as it collects more observations from its human teammate. The results were averaged over 100 arbitrarily generated problems. The x-axis shows the percentage of observed portion in an environment–*i.e.*, as a ratio of the number of observed nodes to that of total number of nodes; the Y-axis shows the harmonic mean of precision p and recall r known as F-measure, defined by $\frac{2pr}{p+r}$. The performance was compared to an earlier approach [16] that uses a temporal weighted-sum update method where newer observations

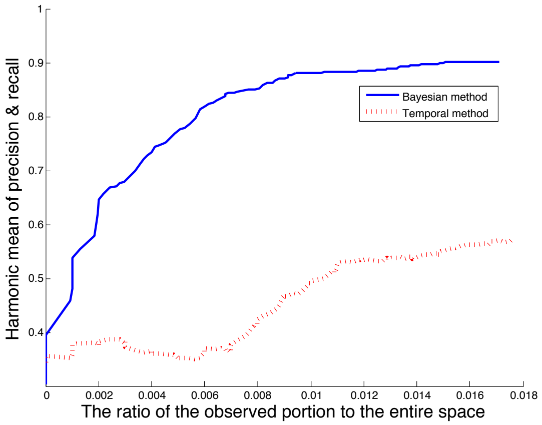


Fig. 5: F-measure of door detection by tracking a software teammate agent. ($\epsilon = 0.05, \lambda = 0.05$)

are more heavily weighted than older ones. Although the earlier approach fared well in terms of estimating riskiness, actual door detection accuracy was generally low as shown in Figure 5. Using our Bayesian method, the robot detected doors with high accuracy, achieving over .88 precision and over .96 recall rates, resulting in an F-measure of over .9, after observing fewer than 1% of nodes.

VI. Conclusion

In this paper, we investigated an alternative approach to vision-based robot perception by applying a probabilistic inference technique to interpret indirect cues. Specifically, we used a 2-layer Noisy-OR Bayes net known as BN2O to represent the dependence relationship between perceived information and human actions. We evaluated our approach in a door detection problem using both synthetic and real tracking data. The results from a small set of real tracking data were promising. On a larger set of synthetic problems similar to the real scenarios, the performance was highly reliable, achieving an F-measure over .9. The idea of inference-based perception is general and can be applied to other perception problems such as detecting other types of objects, the types of terrain, or predicting unseen portions of an environment.

Acknowledgment

This work was conducted in part through collaborative participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016, and in part by ONR under MURI grant "Reasoning in Reduced Information Spaces" (no. N00014-09-1-1052). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] O Alegre and Frank Dellaert. A probabilistic approach to the semantic interpretation of building facades. In *Int. Workshop on Vision Techniques Applied*, pages 1–12, 2004.

[2] Abdeslam Boularias, Felix Duvallet, Jean Oh, and Anthony Stentz. Learning to ground spatial relations for outdoor robot navigation. In *Proc. ICRA*, 2015.

[3] B. D'Ambrosio. Symbolic probabilistic inference in large bn2o networks. In *Proc. UAI*, pages 128–135, 1994.

[4] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, pages 580–587. IEEE, 2014.

[7] Juan Pablo Gonzalez, Bryan Nagy, and Anthony (Tony) Stentz. The geometric path planner for navigating unmanned vehicles in dynamic environments. In *Proceedings ANS 1st Joint Emergency Preparedness and Response and Robotic and Remote Systems*, February 2006.

[8] D. Koller and N. Friedman. *Probabilistic graphical models, principles and techniques*, chapter 12. The MIT Press, 2009.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, pages 1097–1105, 2012.

[10] Pascal Müller, Gang Zeng, Peter Wonka, and Luc Van Gool. Image-based procedural modeling of facades. In *Proc. SIGGRAPH*, New York, NY, USA, 2007. ACM.

[11] Daniel Munoz, J. Andrew Bagnell, and Martial Hebert. Stacked hierarchical labeling. In *Proc. ECCV*, 2010.

[12] A. C. Murillo, J. Kosecká, J. J. Guerrero, and C. Sagüés. Visual door detection integrating appearance and shape cues. *Robotics and Autonomous Systems*, 56(6):512–521, 2008.

[13] L. Navarro-Serment, C. Mertz, N. Vandapel, and M. Hebert. Ladar-based pedestrian detection and tracking. In *Proc. 1st Workshop on Human Detection from Mobile Robot Platforms, ICRA*. IEEE, May 2008.

[14] A.Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proc. ICML*, pages 663–670, 2000.

[15] J. Oh, A. Suppe, F. Duvallet, A. Boularias, J. Vinokurov, L. Navarro-Serment, O. Romero, R. Dean, C. Lebiere, M. Hebert, and A. Stentz. Toward mobile robots reasoning like humans. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[16] J. Oh, A. Suppe, A. Stentz, and M. Hebert. Enhancing robot perception using human teammates. In *Proc. AAMAS*, pages 1147–1148, 2013.

[17] N. M Oliver, B. Rosario, and A. P. Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 22(8):831–843, 2000.

[18] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[19] Radu Bogdan Rusu, Wim Meeussen, Sachin Chitta, and Michael Beetz. Laser-based perception for door and handle identification. In *Proc. ICAR*, pages 1–8. IEEE, 2009.

[20] O. Teboul, L. S. P. Koutsourakis, and N. Paragios. Segmentation of building facades using procedural shape priors. In *Proc. CVPR*, pages 3105–3112, 2010.

[21] A. Torralba, R. Fergus, and W. T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1958–1970, 2008.

[22] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, volume 1, pages 511–518. IEEE, 2001.

[23] L. Wasserman. *All of statistics: a concise course in statistical inference*, chapter 4. Springer, 2003.