Semantic-level Decentralized Multi-Robot Decision-Making using Probabilistic Macro-Observations

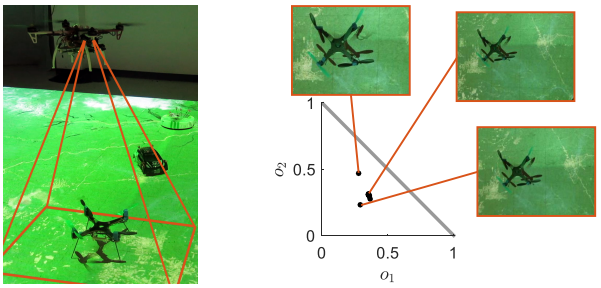
Shayegan Omidshafiei[1], Shih-Yuan Liu[1], Michael Everett[1], Brett T. Lopez[1]
Christopher Amato[2], Miao Liu[3], Jonathan P. How[1], John Vian[4]

***Abstract*— Robust environment perception is essential for decision-making on robots operating in complex domains. Intelligent task execution requires principled treatment of uncertainty sources in a robot's observation model. This is important not only for low-level observations (e.g., accelerometer data), but also for high-level observations such as semantic object labels. This paper formalizes the concept of macro-observations in Decentralized Partially Observable Semi-Markov Decision Processes (Dec-POSMDPs), allowing scalable semantic-level multi-robot decision making. A hierarchical Bayesian approach is used to model noise statistics of low-level classifier outputs, while simultaneously allowing sharing of domain noise characteristics between classes. Classification accuracy of the proposed macro-observation scheme, called Hierarchical Bayesian Noise Inference (HBNI), is shown to exceed existing methods. The macro-observation scheme is then integrated into a Dec-POSMDP planner, with hardware experiments running onboard a team of dynamic quadrotors in a challenging domain where noise-agnostic filtering fails. To the best of our knowledge, this is the first demonstration of a real-time, convolutional neural net-based classification framework running fully onboard a team of quadrotors in a multi-robot decision-making domain.**

I. Introduction

Portable vision sensors, parallelizeable perception algorithms [1], and general purpose GPU-based computational architectures make simultaneous decision-making and scene understanding in complex domains an increasingly-viable goal in robotics. Consider the problem of multi-robot perception-based decision-making in noisy environments, where observations may be low in frame-rate or where semantic labeling is a time-durative process. Each robot may observe an object, infer its underlying class, change its viewpoint, and re-label the object as a different class based on new observations (Fig. 1). Robots must infer underlying object classes based on histories of past classifications, then use this information to execute tasks in a team-based decision-making setting.

For autonomous execution of complex missions using perception-based sensors, robots need access to high-level information extending beyond the topological data typically used for navigation tasks. Use of semantic maps (qualitative environment representations) has been recently explored for intelligent task execution [2]–[4]. Yet, limited work has been conducted on semantic-level multi-robot decision-making in stochastic domains. Heuristic labeling rules [5] or rigid, hand-tuned observation models are failure-prone as they do not infer underlying environment stochasticity for robust decision-making. As real-world robot observation processes are notoriously noisy, semantic-level decision-making can benefit from principled consideration of probabilistic observations.



(a) Macro-observations received onboard moving quadrotor.

(b) Example classification probability outputs on Δ^2.

Fig. 1: Real-time onboard macro-observations in environments with varying lighting conditions, textures, and motion blur.

Cooperative multi-agent decision-making under uncertainty, in its most general form, can be posed as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [6]. Yet, infinite horizon Dec-POMDPs are undecidable and finite horizon Dec-POMDPs are NEXP-complete, severely limiting application to real-world robotics [7], [8]. Recent efforts have improved Dec-POMDP scalability by introducing macro-actions (temporally-extended actions) into the framework, resulting in Decentralized Partially Observable *Semi*-Markov Decision Processes (Dec-POSMDPs) [9]–[11]. Use of durative macro-actions significantly improves planner scalability by abstracting low-level actions from high-level tasks.

So far, research focus has been on action-space scalability—no similar work targeting observation-space scalability has been conducted. Further, the scope of the large body of work on Dec-POMDPs has primarily been within the artificial intelligence perspective, with limited focus on traditional robotics applications [6]. While the strength of Dec-POMDPs and Dec-POSMDPs comes from principled treatment of stochasticity, they have primarily been applied to benchmark domains with simple or hand-crafted observation models [6]. Derivation of data-driven, robust observation processes usable for Dec-POSMDP policy search remains a challenge. As planning complexity is exponential in the number of observations, abstraction to meaningful high-level macro-observations (appropriate for the tasks being completed) is

*This work was supported by The Boeing Company.

[1]Laboratory for Information and Decision Systems (LIDS), MIT, Cambridge, MA 02139, USA {shayegan,syliu,mfe,btlopez,jhow}@mit.edu

[2]College of Computer and Information Science (CCIS), Northeastern University, Boston, MA 02115 USA camato@ccs.neu.edu

[3]Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA miao.liu1@ibm.com

[4]Boeing Research & Technology, Seattle, WA 98108, USA john.vian@boeing.com

desired. Thus, major research gaps exist in leveraging the Dec-POSMDP's full potential for real-world robotics. This paper addresses these issues, providing a high-level abstraction of observation processes and scalability improvements in a similar manner as previous work on macro-actions.

This paper's primary contribution is a formalization of macro-observation processes within Dec-POSMDPs, with a focus on the ubiquitous perception-based decision-making problem encountered in robotics. A hierarchical Bayesian macro-observation framework is introduced, using statistical modeling of observation noise for probabilistic classification in settings where noise-agnostic methods are shown to fail. The resulting data-driven approach avoids hand-tuning of observation models and produces statistical information necessary for Dec-POSMDP solvers to compute a policy. Hardware results for real-time semantic labeling on a moving quadrotor are presented, with accurate inference in settings with high perception noise. The entire processing pipeline is executed onboard a quadrotor at approximately 20 frames per second. The macro-observation process is then integrated into a Dec-POSMDP planner, with demonstration of semantic-level decision-making executed on a quadrotor team performing a perception-based health-aware disaster relief mission.

II. Decentralized Multi-Robot Decision-Making

This section summarizes the Dec-POSMDP framework, a decentralized decision-making process targeting large-scale multi-agent problems in stochastic domains. The Dec-POSMDP addresses scalability issues of Dec-POMDPs by incorporating belief-space macro actions, or temporally-extended actions. For details on Dec-POSMDP fundamentals, we refer readers to our previous work [9]–[11].

Robots involved in Dec-POSMDPs operate in belief space, the space of probability distributions over states, as they only perceive noisy observations of the underlying state. Solving a Dec-POSMDP results in a hierarchical decision-making policy, where a macro-action (MA) $\pi^{(i)} \in \mathbb{T}^{(i)}$ is first selected by each robot $i \in \mathbb{I}$, and low-level (primitive) actions are conducted within the MA until an ϵ-neighborhood of the MA's belief milestone $\check{b}^{goal}$ is reached.[1] This neighborhood defines a *goal belief node* for the MA, denoted $B^{goal} = \{b : \|b - \check{b}^{goal}\| \leq \epsilon\}$. Each MA encapsulate a low-level POMDP involving primitive actions $u_t^{(i)}$ and observations $o_t^{(i)}$.

Definition 1: The Dec-POSMDP is defined below:

- $\mathbb{I} = \{1, 2, \ldots, n\}$ is the set of heterogeneous robots.
- $\mathbb{B}^{(1)} \times \mathbb{B}^{(2)} \times \ldots \times \mathbb{B}^{(n)} \times \mathbb{X}^e$ is the underlying belief space, where $\mathbb{B}^{(i)}$ is the set of belief milestones of the i-th robot's MAs and $\mathbb{X}^e$ is the environment state space.
- $\bar{\mathbb{T}} = \mathbb{T}^{(1)} \times \mathbb{T}^{(2)} \ldots \times \mathbb{T}^{(n)}$ is joint independent MA space, where $\mathbb{T}^{(i)}$ is the set of MAs for the i-th robot. $\bar{\pi} = \{\pi^{(1)}, \ldots, \pi^{(n)}\}$ is the team's joint MA.
- $\breve{\mathbb{O}}^e = \{\breve{\bar{o}}^e\}$ is the set of all joint MA-observations.
- $P(\bar{b}', x^{e'}, k | \bar{b}, x^e; \bar{\pi})$ is the high-level transition probability model under MAs $\bar{\pi}$ from $(\bar{b}, x^e)$ to $(\bar{b}', x^{e'})$.
- $\bar{R}^\tau(\bar{b}, x^e; \bar{\pi})$ is the generalized reward of taking a joint MA $\bar{\pi}$ at $(\bar{b}, x^e)$, where $\bar{b}$ is the joint belief.
- $P(\breve{\bar{o}}^e | \bar{b}, x^e)$ is the joint observation likelihood model, with observation $\breve{\bar{o}}^e = \{\breve{o}^{e(1)}, \breve{o}^{e(2)}, \ldots, \breve{o}^{e(n)}\}$.
- $\gamma \in [0, 1)$ is the reward discount factor.

Let $\mathbb{X}^e$ be the high-level or macro-environment state space, a finite set describing the state space extraneous to robot states (e.g., an object in the domain). An observation of the *macro-environment state* $x^e \in \mathbb{X}^e$ is denoted as the *macro-observation* $o^{e(i)}$. Upon completion of its MA, each robot makes macro-observation $o^{e(i)}$ and calculates its final belief state, $b^{f(i)}$. This macro-observation and final belief are jointly denoted as $\breve{o}^{e(i)} = (o^{e(i)}, b^{f(i)})$.

The history of executed MAs and received high-level observations is denoted as the *MA-history*,

$$\xi_k^{(i)} = \{\breve{o}_0^{e(i)}, \pi_0^{(i)}, \breve{o}_1^{e(i)}, \pi_1^{(i)}, \ldots, \breve{o}_{k-1}^{e(i)}, \pi_{k-1}^{(i)}, \breve{o}_k^{e(i)}\}. \quad (1)$$

The transition probability $P(\bar{b}', x^{e'}, k | \bar{b}, x^e; \bar{\pi})$ from $(\bar{b}, x^e)$ to $(\bar{b}', x^{e'})$ under joint MA $\bar{\pi}$ in k timesteps is [11],

$$\begin{aligned} P(\bar{b}', x^{e'}, k | \bar{b}_0, x_0^e, o_k^e; \bar{\pi}) &= P(x_k^e, \bar{b}_k | \bar{b}_0, x_0^e, o_k^e; \bar{\pi}) \\ &= \sum_{x_{k-1}^e, \bar{b}_{k-1}} \Big[P(x_k^e | x_{k-1}^e, o_k^e; \bar{\pi}(\bar{b}_{k-1})) \times \quad (2) \\ P(\bar{b}_k | x_{k-1}^e, \bar{b}_{k-1}; \bar{\pi}(\bar{b}_{k-1})) P(x_{k-1}^e, \bar{b}_{k-1} | x_0^e, \bar{b}_0; \bar{\pi}(\bar{b}_0)) \Big]. \end{aligned}$$

The generalized team reward for a discrete-time Dec-POSMDP during execution of joint MA $\bar{\pi}$ is defined [11],

$$\bar{R}^\tau(\bar{b}, x^e; \bar{\pi}) = \mathbb{E}\left[\sum_{t=0}^{\tau-1} \gamma^t \bar{R}(\bar{x}_t, x_t^e, \bar{u}_t) | P(\bar{x}_0) = \bar{b}, x_0^e = x^e; \bar{\pi} \right] \quad (3)$$

where $\tau = \min_i \min_t \{t : b_t^{(i)} \in B^{(i),goal}\}$ is the timestep at which robot i completes its current MA, $\pi^{(i)}$. Note that τ is itself a random variable, since MA completion times are also non-deterministic. Thus, the expectation in (3) is taken over MA completion durations as well. In practice, sampling-based approaches are used to estimate this expectation.

MA selection is dictated by the *joint high-level policy*, $\bar{\phi} = \{\phi^{(1)}, \ldots, \phi^{(n)}\}$. Each robot's high-level policy $\phi^{(i)}$ maps its MA-history $\xi_k^{(i)}$ to a subsequent MA $\pi^{(i)}$ to be executed. The joint value under policy $\bar{\phi}$ is,

$$\begin{aligned} \bar{V}^{\bar{\phi}}(\bar{b}, x^e) &= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^{t_k} \bar{R}^\tau(\bar{b}_{t_k}, x_{t_k}^e; \bar{\pi}_{t_k}) | \bar{b}_0, x_0^e; \bar{\phi} \right] \quad (4) \\ &= \bar{R}^\tau(\bar{b}, x^e; \bar{\pi}) + \sum_{k=1}^{\infty} \gamma^{t_k} \sum_{\bar{b}', x^{e'}, \bar{o}^{e'}} P(\bar{b}', x^{e'}, \bar{o}^{e'}, k | \bar{b}, x^e; \bar{\pi}) \bar{V}^{\bar{\phi}}(\bar{b}', x^{e'}). \end{aligned}$$

The optimal joint high-level policy is then,

$$\bar{\phi}^* = \underset{\bar{\phi}}{\operatorname{argmax}} \bar{V}^{\bar{\phi}}(\bar{b}, x^e). \quad (5)$$

To summarize, the Dec-POSMDP is a hierarchical decision-making process which involves finding a joint high-level policy $\bar{\phi}$ dictating the MA $\pi^{(i)}$ each robot $i \in \mathbb{I}$ conducts based on its history of executed MAs and received high-level observations. Within each MA, the robot executes low-level actions $u_t^{(i)}$ and perceives low-level observations $o_t^{(i)}$. Therefore, the Dec-POSMDP is an abstraction of the Dec-POMDP process which treats the problem at the high macro-action level to significantly increase planning scalability.

[1] We denote a generic parameter p of the i-th robot as $p^{(i)}$, joint parameter of the team as $\bar{p}$, and joint parameter at timestep k as $\bar{p}_k$.

III. Semantic Macro-Observations

This section formalizes Dec-POSMDP macro-observations. It also outlines the sequential-observation classification problem for macro-observation models and introduces a hierarchical Bayesian scheme for semantic-level macro-observations.

A. Macro-Observation Processes

Dec-POSMDPs naturally embed state and macro-action uncertainty into a high-level decision-making process. In a similar manner, task planning can benefit from the robot's high-level understanding of the environment state. Previous research has focused on formal definitions of MAs in terms of low-level POMDPs and on algorithms for automatically generating them [10]. Yet, no formal work on automatic macro-observation generation has been done to date. Benchmark domains used to test Dec-POSMDP search algorithms use simplistic or hand-coded high-level observation processes, which are subsequently sampled during the evaluation phase of policy search algorithms [10], [11]. In contrast, this paper provides a foundation for deriving meaningful, data-driven macro-observations. We formally define macro-observations herein by distinguishing them from low-level observations:

Definition 2: Macro-observations are durative, generative probabilistic processes within which sequences of low-level observations are filtered, resulting in a semantic-level observation of the environment.

Macro-observations allow each robot's noisy semantic perception of the world to affect its task selection. Just as MAs provide an abstraction of low-level actions to a high-level task (e.g., "Open the door"), macro-observations abstract low-level observations to a high-level meaningful understanding of the environment state (e.g., "Am I in an office?").

For uncertainty-aware planning, Dec-POSMDP policy search algorithms require sampling of the domain transition and observation model distributions discussed in Section II. Thus, the following distributions must be calculable for any robot's derived macro-observation process:

1) a semantic output distribution $P(o^e|b^{(i)}, x^e)$ of underlying macro (environment) state
2) a distribution over computation time τ

While low-level observation processes can be treated as instantaneous for simplicity, observations related to scene semantics require non-negligible computation time which must be accounted for in the planner. Dec-POSMDPs seamlessly take this computation time into account. Definition 2 provides a natural representation for real-world high-level observation processes, as they are durative (i.e., take multiple timesteps to process low-level data). Further, this computation time is non-deterministic (e.g., the amount of time needed to answer "Am I in an office?" is conditioned on scene lighting). The existing Dec-POSMDP transition dynamics in (3) take an expectation over MA completion times. As every macro-observation is perceived following an MA, the time distribution in (3) can seamlessly include macro-observation computation time.

The result is a particularly powerful semantic-level decision-making framework, as MAs targeting desired macro-observations can be embedded in the Dec-POSMDP (e.g., "Track object until its class is inferred with 95% confidence"). The next sections focus on development of an automatic process which provides Dec-POSMDP solvers with the two necessary macro-observation distributions (semantic output distribution and computation time distribution).

B. Sequential Classification Filtering Problem

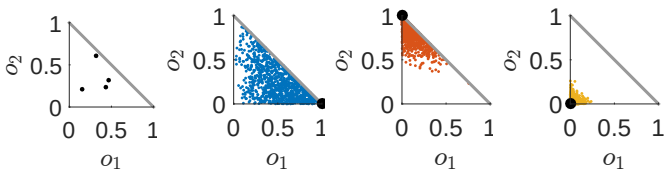
We now detail generation of macro-observations in the context of probabilistic object classification. Specifically, consider a ubiquitous decision-making scenario where a robot observes a sequence of low-level classifier outputs and must determine its surrounding environment state or class of an object in order to choose a subsequent task to execute. A unique trait of robotic platforms is locomotion, allowing observations of an object or scene from a variety of viewpoints (Fig. 1). This motivates the need for a sequential macro-observation process using the history of classification observations made by the robot throughout its mission. In contrast to naïve reliance on frame-by-frame observations, sequential filtering offers increased robustness against domain uncertainty (e.g., camera noise, lighting conditions, or occlusion).

In settings with high observation noise, or where training data is not representative of mission data, statistical analysis of low-level classifier outputs both improves accuracy of macro-observations and provides useful measures of perception uncertainty. As a motivating example, consider the 3-class scenario in Fig. 2. A low-level classifier predicts the probability of a single image belonging to each class. A sequence of images results in a corresponding sequence of observed class probabilities, as in Fig. 2a for a 4-image sequence. This makes inference of the underlying class nontrivial.

Let us formalize the problem of constructing semantic macro-observations using streaming classification outputs. Given input feature vector f_i at time i, an M-class probabilistic classifier outputs low-level probability observation $o_i = (o_{i,1}, \ldots, o_{i,m}, \ldots, o_{i,M})$, where $o_{i,m}$ is the raw probability of f_i belonging to the m-th class (e.g., Fig. 2a). Thus, o_i is a member of the $(M-1)$-simplex, Δ^{M-1}.

In object classification, f_i may be an image or feature representation thereof, and $o_{i,m}$ represents probability of the object belonging to the m-th class. This probabilistic classification is conducted over a sequence of N images $f_{1:N}$, resulting in a stream of class probability observations $o_{1:N}$. In robotics, this macro-observation process is inherently durative as multiple low-level observations of the object need to be perceived to counter domain noise. Simply labeling the object as belonging to the class with maximal probability, $\mathrm{argmax}_m(o_{i,m})$, can lead to highly sporadic outputs as the image sequence progresses. A filtering scheme using the history of classifications $o_{1:N}$ is desired, along with the two aforementioned characterizing macro-observation distributions necessary for Dec-POSMDP search algorithms.

Prior work on aggregation of *multiple* classifiers' predictions can be extended to single-classifier multi-observation filtering, where in each case, the posterior outputs would become macro-observation o^e. Fixed classifier combination rules offer simplicity in implementation at the cost of sub-optimality. One example is the max-of-mean approach [12], where the m-th class posterior probability, o'_m, is the mean



(a) 4 low-level classifier observations. (b) Classifications for $c = 1$, $\theta_{c=1} = 1$. (c) Classifications for $c = 2$, $\theta_{c=2} = 6$. (d) Classifications for $c = 3$, $\theta_{c=3} = 20$.

Fig. 2: Motivating macro-observation example with 3 classes. Each point represents a single low-level observation $o_i \in \Delta^2$.

of observed probabilities throughout the image sequence,

$$o'_m = \frac{1}{N}\sum_{i=1}^{N} o_{i,m} \quad \forall m \in \{1, \ldots, M\}. \tag{6}$$

Another strategy is voting-based consensus [13], with posterior class c chosen based on the highest number of votes from all individual prediction observations o_i,

$$c = \underset{c' \in \{1,\ldots,M\}}{\text{argmax}} \sum_i \delta(c', \underset{m \in \{1,\ldots,M\}}{\text{argmax}} \; o_{i,m}) \tag{7}$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta function.

The above approaches do not exploit the probabilistic nature of underlying classifier outputs, o_i. A Bayes filter offers a more principled treatment of the problem. For example, binary Bayes filters are a popular approach for occupancy grid filtering and object detection [14], [15], where repeated observations are filtered to determine occupancy probability or presence of an object (both are $M = 2$ class cases, with classes 'occupied/present' or 'empty/absent'). Binary Bayes filters can be extended to M-class recursive classification by applying Bayes rule and a Markovian observation assumption,

$$P(c{=}m|f_{1:N}) \propto \frac{P(c=m|f_N)}{P(c=m)} P(c=m|f_{1:N-1}), \tag{8}$$

where $P(c = m)$ is the prior class distribution and $P(c = m|f_N) = o_{N,m}$. This Bayes filter assumes a fixed underlying class, henceforth called a Static State Bayes Filter (SSBF).

Though SSBF allows probabilistic filtering of classifier outputs, it assigns equal confidence to each observation o_i in its update. It takes equal amount of evidence for a class to "cancel out" evidence against it, an issue encountered in Bayes-based occupancy mapping [16]. In settings with heterogeneous classifier performance, this approach performs poorly. One class may be particularly difficult to infer in a given domain, increasing probability of misclassifications compared to other classes. In our motivating example, Figs. 2b to 2d illustrate noisy classification samples for the 3 underlying object classes. Class $c = 1$ (Fig. 2b) is particularly difficult to classify, with a near-uniform distribution of o_i throughout the simplex, in contrast to high-accuracy classifications of $c = 3$ (Fig. 2d). In this case, given uniform observations throughout the simplex *and* knowledge of underlying classifier noise, the filter update weight on underlying class $c = 1$ should be higher than $c = 3$, since the classifier outputs are most sporadic for class $c = 1$.

The critical drawback of the above approaches is that they simply *filter*, but do not *model*, the underlying observation process. As discussed earlier in Section III-A, generative high-accuracy macro-observation models are necessary for Dec-POSMDP policy search algorithms [9], [11]. Perception-based observations are highly complex and involve images/video sequences generated from the domain, making them (currently) impossible to replicate in these offline search algorithms. While it may be tempting to use hand-coded generative distributions for the above *filter-based* macro-observation processes during policy search, such an approach fails to exploit the primary benefit of POMDP-based frameworks: the use of data-driven noise models which result in policies that are robust in the real world.

C. Hierarchical Approach for Semantic Macro-Observations

This section introduces a generative macro-observation model titled Hierarchical Bayesian Noise Inference (HBNI), which infers inherent heterogeneous classifier noise. HBNI provides a compact, accurate, generative perception-based observation model, which is subsequently used to sample the two macro-observation distributions in Dec-POSMDP solvers. The combination of Dec-POSMDPs with HBNI macro-observations allows robust, probabilistic semantic-level decision-making in settings with limited, noisy observations.

To ensure robustness against misclassifications, HBNI involves both noise modeling and classification filtering, making it a multi-level inference approach. Given a collection of image class probability observations $o_{1:N}$ (Fig. 2a), the underlying class for each image f_i is inferred while modeling classifier noise distributions.

Hierarchical Bayesian models allow multi-level abstraction of uncertainty sources [17]. This is especially beneficial in stochastic settings targeted by Dec-POSMDPs, which have layered sources of uncertainty. In semantic labeling, for instance, parameterization of the classifier confidence for the M classes can be modeled using a set of noise parameters $\theta_{1:M}$. Moreover, it is beneficial to model the relationship between noise parameters through a shared prior (Fig. 3). Consider, for instance, a robot performing object classification using a low-quality camera or in a domain with poor visibility. In this setting, observations may be noisier than expected a priori, indicating presence of a high-level, class-independent uncertainty source. This information should be shared amongst all class models, allowing more accurate modeling of domain uncertainty through the noise parameters. Layered sharing of statistical information between related parameters is a strength of hierarchical Bayesian models, and has been demonstrated to increase robustness in posterior inference compared to non-hierarchical counterparts [18].

Fig. 3 illustrates the graphical model of HBNI. A categorical prior is used for classes,

$$c_i \sim Cat(p_{1:M}), \tag{9}$$

where $p_i \in \Delta^{M-1} \; \forall i \in \{1, \ldots, M\}$. This allows integration of prior domain knowledge into HBNI. A Dirichlet observation model is used for raw classifier outputs $o_i \in \Delta^{M-1}$,

$$o_i \sim Dir(\theta_{c_i} \vec{1}_{c_i} + \vec{1}), \tag{10}$$

where $\theta_{c_i} \geq 0$ is a scalar noise parameter for the associated class, $\vec{1}_{c_i}$ is an $M \times 1$ categorical vector with the c_i-th element

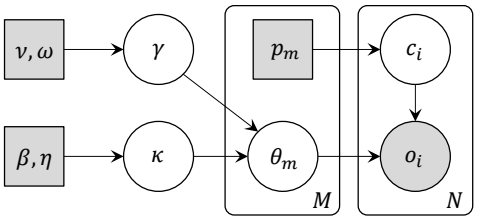


Fig. 3: The HBNI model, with per-class noise parameters θ_m and shared hyperparameters, κ, γ.

equal to 1 and remaining element equal to zero, and $\vec{1}$ is an $M \times 1$ vector of ones. Each class observation o_i has an associated class label c_i, which in turn links o_i to the appropriate noise parameter θ_{c_i} (the c_i-th element of parameter set $\{\theta_1, \ldots, \theta_M\}$). This choice of parameterization offers two advantages. First, the selection of θ_{c_i} provides a direct, intuitive measure of noise for the classifier observations. As in Figs. 2b to 2d, θ_{c_i} is the Dirichlet concentration parameter and is related to the variance of the classification distribution. Low values of θ_{c_i} imply high levels of observation noise, and vice versa. A second advantage is that it simplifies the posterior probability calculations used within Markov chain Monte Carlo (MCMC) inference, as discussed below.

A gamma prior is used for noise parameter θ_m,

$$\theta_m \sim Ga(\kappa, \gamma), \tag{11}$$

where κ and γ themselves are treated as unknown hyperparameters. The role of κ and γ is to capture high-level sources of domain uncertainty, allowing sharing of cross-class noise statistics. Gamma priors (parameterized by (β, η) and (ν, ω)) were also used for these hyperparameters in our experiments, although results showed low sensitivity to this prior choice.

Given raw class probability observations $o_{1:N}$, the posterior probability of noise parameters and associated classes is,

$$\begin{aligned} &P(\theta_{1:M}, c_{1:N}, \kappa, \gamma | o_{1:N}) \\ &\quad \propto \prod_{i=1}^{N} P(o_i|\theta_{c_i}, c_i) P(c_i) \prod_{m=1}^{M} P(\theta_m|\kappa, \gamma) P(\kappa) P(\gamma) \quad (12) \\ &\quad = \prod_{i=1}^{N} \Big[Dir(o_i; \theta_{c_i} \vec{1}_{c_i} + \vec{1}) \, Cat(c_i; p_{1:M}) \Big] \\ &\qquad \times \prod_{m=1}^{M} Ga(\theta_m; \kappa, \gamma) \, Ga(\kappa; \beta, \eta) \, Ga(\gamma; \nu, \omega). \quad (13) \end{aligned}$$

This allows inference of noise parameters $\theta_{1:M}$ and hyperparameters κ and γ using the collection of observed data $o_{1:N}$. The computational complexity of (13) can be further reduced. The log of the prior (9) is simply $\log Cat(c_i; p_{1:M}) = \log p_{c_i}$. To efficiently compute $\log Dir(o_i; \theta_{c_i} \vec{1}_{c_i} + \vec{1})$, consider a notation change. Letting $\bar{\alpha} = \{\alpha_1, \ldots, \alpha_M\} = \theta_{c_i} \vec{1}_{c_i} + \vec{1}$,

$$Dir(o_i; \bar{\alpha}) = \frac{1}{B(\bar{\alpha})} \prod_{m=1}^{M} o_{i,m}^{\alpha_m - 1}, \tag{14}$$

with $B(\cdot)$ as the Beta function. Based on the definition of $\bar{\alpha}$,

$$\alpha_m - 1 = \begin{cases} \theta_{c_i}, & m = c_i. \\ 0, & m \neq c_i. \end{cases} \tag{15}$$

Combining (15) with (14) and taking the log,

$$\begin{aligned} &\log Dir(o_i; \bar{\alpha}) \\ &\quad = -\log B(\bar{\alpha}) + \theta_{c_i} \log o_{i,c_i} \quad (16) \\ &\quad = -\sum_{m=1}^{M} \log \Gamma(\alpha_m) + \log \Gamma(\sum_{m=1}^{M} \alpha_m) + \theta_{c_i} \log o_{i,c_i}, \quad (17) \end{aligned}$$

where Γ is the gamma function. Note that as per (15),

$$\log \Gamma(\alpha_m) = \begin{cases} \log \Gamma(1 + \theta_{c_i}), & m = c_i. \\ 0, & m \neq c_i. \end{cases} \tag{18}$$

and $\sum_m \alpha_m = M + \theta_{c_i}$. Thus, the Dirichlet log-posterior is,

$$\begin{aligned} &\log Dir(o_i; \bar{\alpha}) \\ &\quad = -\log \Gamma(1 + \theta_{c_i}) + \log \Gamma(M + \theta_{c_i}) + \theta_{c_i} \log o_{i,c_i}. \quad (19) \end{aligned}$$

Finally, the log-probability of θ_m (and similarly κ, γ) is,

$$\log Ga(\theta_m; \kappa, \gamma) \propto (\kappa - 1) \log \theta_m - \theta_m \gamma^{-1}. \tag{20}$$

To summarize, the log of (13) is efficiently computed by combining (19) and (20). An MCMC approach is used to calculate the posterior distribution over noise parameters ($\theta_{1:M}$) and hyperparameters (κ, γ). This allows a history of observations $o_{1:N}$ to be filtered using the noise distributions, resulting in posterior class probabilities,

$$\begin{aligned} &P(c = m | o_{1:N}, \theta_{1:M}) \\ &\quad \propto P(o_{1:N}|\theta_m, c = m) P(c = m) \quad (21) \\ &\quad = P(o_N|\theta_m, c = m) \Big[\prod_{i=1}^{N-1} P(o_i|\theta_m, c = m) p_m \Big], \quad (22) \end{aligned}$$

where c is conditionally independent of κ and γ given $\theta_{1:M}$, allowing hyperparameter terms to be dropped. Recall $P(o_N|\theta_m, c = m) = Dir(o_N; \theta_m \vec{1}_m + \vec{1})$, the Dirichlet density at o_N. Thus, (13) provides a generative distribution for low-level observations (after noise parameter inference), and (22) provides a recursive filtering rule for macro-observations given each new observation o_N. Combined, these equations provide a macro-observation model and filtering scheme which can be used in Dec-POSMDP search algorithms.

To summarize, the proposed HBNI approach uses the collection of classification observations o_i to calculate a posterior distribution on noise parameters $\theta_{1:M}$ for each object class, and shared hyperparameters κ and γ. These noise distributions are then used for online streaming of class probability macro-observations. While HBNI noise inference is computationally efficient and can be conducted online, the complexity of Dec-POSMDPs means that existing sampling-based policy search algorithms are run offline. Thus, integration of HBNI macro-observations into Dec-POSMDPs is a three-fold process. First, domain data is collected and HBNI noise inference of parameters and hyperparameters is conducted, resulting in a generative observation distribution. This distribution is then used for domain sampling and policy search in Dec-POSMDP search algorithms. The resulting policy is then executed online, with HBNI-based filtering used to output macro-observations. The generative nature of HBNI allows usage of complex, durative macro-observation

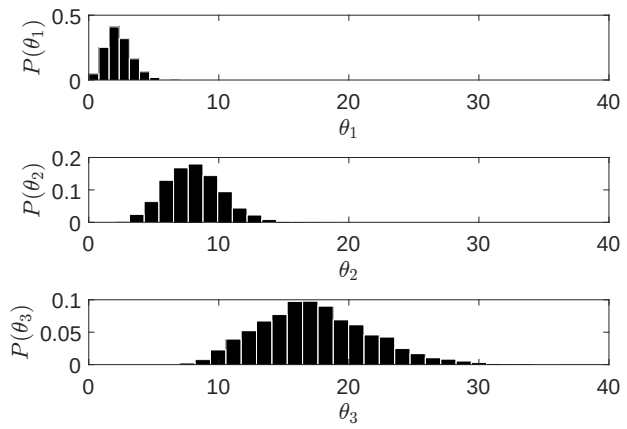
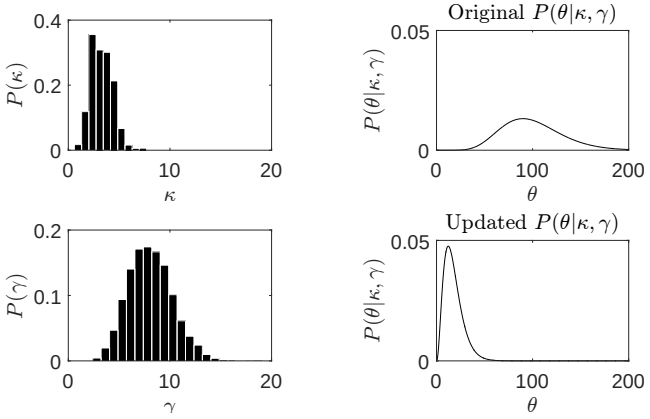


Fig. 4: Inferred noise parameters θ for the $M = 3$ classification problem illustrated in Fig. 2. True noise parameter values are $\theta_1 = 1$, $\theta_2 = 6$, $\theta_3 = 20$.



(a) Posterior distributions of κ and γ.

(b) θ prior before/after hyperparameter update.

Fig. 5: Inference of high-level noise parameters κ, γ. Median hyperparameters were used for the plots on the right.

processes, which can filter the stream and output a macro-observation *only when* a desired confidence level is reached.

IV. Simulated Experiments

This section validates HBNI's performance in comparison to noise-agnostic filtering schemes, before integration into Dec-POSMDPs. As stated earlier, an MCMC approach is used to compute the posterior over $\theta_{1:M}$, κ, and γ. Specifically, the experiments conducted use a Metropolis-Hastings (MH) [19] sampler with an asymmetric categorical proposal distribution for underlying classes c_i, with high weight on previously-proposed class and low weight on remaining classes (given uniform random initialization). Gaussian MH proposals are used for transformed variables $\log(\theta_m)$, $\log(\kappa)$, and $\log(\gamma)$.

Fig. 4 shows noise parameter ($\theta_{1:M}$) posterior distributions for the $M = 3$ problem outlined in Fig. 2. Parameter inference was conducted using only $N = 15$ classification observations o_i (5 from each class). Despite the very limited number of observations, the posterior distributions provide reasonable inferences of the true underlying noise parameters.

Hyperparameter (κ, γ) posteriors are shown in Fig. 5a. Recall these shared parameters capture trends in outputs o_i which indicate shifts in classification confidence levels (for all classes) due to domain-level uncertainty. To test sensitivity of θ_m inference to the hyperparameters, priors for κ and γ were chosen such that (on average) they indicate very high values of θ_m (Fig. 5b, top). This sets a prior expectation of near-perfect outputs from classifiers (median $\theta_m = 100$). However, given only $N = 15$ classifier observations, posteriors of κ and γ shift to indicate much lower overall classification confidence θ_m (Fig. 5b, bottom). $P(\theta_m|\kappa, \gamma)$ has now shifted to better capture the range of noise parameters expected in the domain. This sharing of high-level noise statistics improves filtering of subsequent observations (even if from an entirely new class).

HBNI classification error is evaluated against the voting, max-of-mean, and SSBF methods discussed in Section III-B. Fig. 6 shows results for varying number N of class observations o_i, with 2000 trials used to calculate error for each case. Voting performs poorly as it disregards class probabilities altogether. HBNI significantly outperforms the other methods, requiring 5-10 observations to converge to the true object class for all trials. The other methods need 4-5 times the number of observations to match HBNI's performance. One interesting result is that for $N = 1$, predictions for voting, max-of-mean, and SSBF are equivalent. However, due to noise modeling, HBNI makes an informed decision regarding underlying class, leading to lower classification error.

V. Hardware Experiments

This section evaluates HBNI on a robotics platform to ascertain the benefits of noise modeling in real-world settings. It then showcases multi-robot Dec-POSMDP decision-making in hardware using HBNI-based macro-observations.

A. Underlying (Low-Level) Classification Framework

Low-level classifier training is conducted on a dataset of 3 target vehicle classes (iRobot, Quadrotor, Racecar) in a well-lit room, using a QVGA-resolution webcam (Fig. 8). 100 snapshots of each object type are used for training, including crops and mirror images for increased translational and rotational invariance. Feature extraction is done using a Convolutional Neural Net (CNN) implemented in Caffe [20] (though the proposed HBNI approach is agnostic to underlying classifier type). Images are center-cropped with 10% padding and resized to 227×227 resolution. Features are extracted from the 8-th fully connected layer of an AlexNet [21] trained on the ILSVRC2012 dataset [1]. These features are used to train a set of Support Vector Machines (SVMs), with a one-vs-one approach for multi-class classification. As SVMs are inherently discriminative classifiers, probabilities o_i for each image f_i are calculated using Platt Scaling, by fitting a sigmoid function to SVM scores [22]. These probabilities are then processed using HBNI-based macro-observations.

B. Hardware Platform

DJI F330 quadrotors with custom autopilots are used for the majority of experiments (Fig. 7), with a Logitech C615 webcam for image capture. The macro-observation pipeline is executed on an onboard NVIDIA Jetson TX1, powered using a dedicated 3-cell 1350mAh LiPo battery. Runtime for the underlying classifier is 49±5ms per frame, and the entire pipeline (including communication and filtering) executes fully onboard at approximately 20 frames per second.

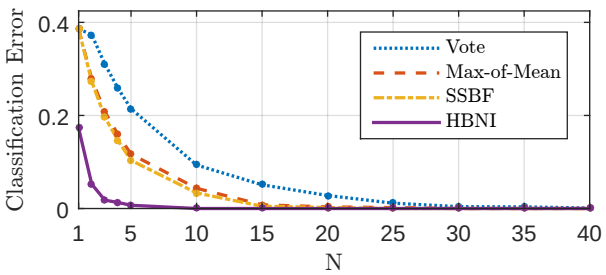


Fig. 6: Filtering error for varying observation lengths N.

C. Results: HBNI-based Macro-Observations

Classification robustness is verified using an augmented reality testbed [23] to change domain lighting conditions. In contrast to the well-lit images used to train the underlying classifier (Figs. 8a to 8c), test images have textured backgrounds and dim lighting which reduce camera shutter speed, increasing blur (Fig. 8d). Experiments are designed to simulate typical scenarios in robotics where the training dataset is not fully representative of mission test data.

Filtered classification results for the test dataset are shown in Fig. 9. In new lighting conditions, classification of the Quadrotor object class is particularly difficult, resulting in nearly equal raw probabilities o_i amongst all three classes (raw data in Fig. 9). Noise-agnostic filters such as SSBF fail to correctly classify the object as a Quadrotor, instead classifying it as an iRobot with high confidence (filtered output in Fig. 9a). Moreover, probability of the Quadrotor class asymptotically approaches zero as more observations are made. In contrast, HBNI infers underlying noise, leading to robust classification of the Quadrotor object after only 7 frames (Fig. 9b). In the $N = 70$ to $N = 75$ range, due to improved lighting, raw classifier probabilities increase for the Quadrotor class. SSBF only slightly lowers its probability of the object being an iRobot, whereas the HBNI approach significantly increases probability of the true Quadrotor class. Fig. 10 shows HBNI macro-observations on a quadrotor exploring an environment with multiple objects. The results indicate that HBNI accurately classifies objects onboard a moving robot in noisy domains. For additional HBNI results and analysis, readers can refer to our technical report [24].

D. Results: Multi-Robot Decision-Making

HBNI-based macro-observations were integrated into the Dec-POSMDP framework (as described in Section III) and evaluated on a multi-robot health-aware disaster relief domain (Fig. 11). This is an extension of the Dec-POSMDP package delivery domain [10] involving a team of quadrotors. Disaster relief objects of 6 types (ambulance, police_car, medical_copter, news_copter, food_crate, medical_crate) are randomly generated at 2 bases, each with an associated delivery destination (hospital, airport, or crate_zone). Nine MAs are available for execution by each robot: *Go to* $Base_1/Base_2/Hospital/Airport/Crate\_zone$, *Go to repair station for maintenance*, *Infer object class with 95% confidence*, *Pick up disaster relief object*, *Put down disaster relief object*. Quadrotors are outfitted with the hardware discussed in Section V-B and use HBNI to infer the underlying disaster relief object class during policy execution. The team

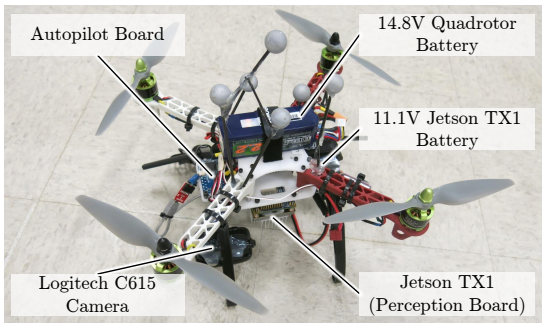
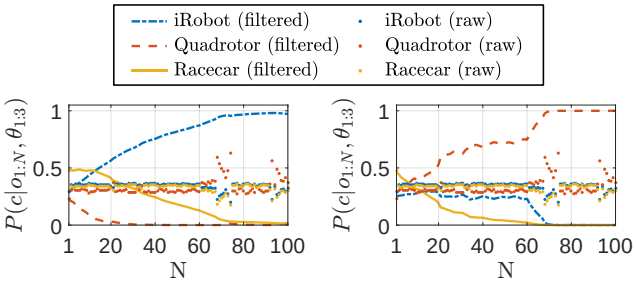


Fig. 7: Hardware overview.

(a) iRobot class example. (b) Quadrotor class example. (c) Racecar class example. (d) Test domain conditions.

Fig. 8: Comparison of training and test images in domains with varying lighting conditions.



(a) SSBF posterior over time. (b) HBNI posterior over time.

Fig. 9: Comparison of SSBF and HBNI filtering, recorded on a moving quadrotor. True object class is Quadrotor.

receives a reward for each object delivered to the correct destination. Quadrotors also receive noisy observations from onboard health sensors and maintain a belief distribution over their underlying health state (high, medium, and low health), indicated by colored rings in Fig. 11. Robots with low health take longer to complete MAs, thereby reducing overall team reward due to the discount factor in (4). Perception data is collected and used to train the HBNI-based macro-observation process, which is then used for Dec-POSMDP policy search via the Graph-based Direct Cross Entropy algorithm [11].

MAs in this domain have probabilistic success rates and completion times. An augmented reality system is used to display bases, disaster relief objects, and delivery destinations in real-time in the domain. The domain includes shadows and camera noise, but perception uncertainty is further increased by projecting a dynamic day-night cycle and moving backdrop of clouds on the domain.

Our video attachment shows this multi-robot mission executed on a team of quadrotors. HBNI inference occurs onboard, with the necessary number of low-level observations processed to achieve high confidence. Mission performance matches that of previous (simpler) results for this domain which simulated all observations [11]. To the best of our knowledge, this is the first demonstration of real-time, CNN-based classification running onboard quadrotors in a team setting. It is also the first demonstration of data-driven multi-

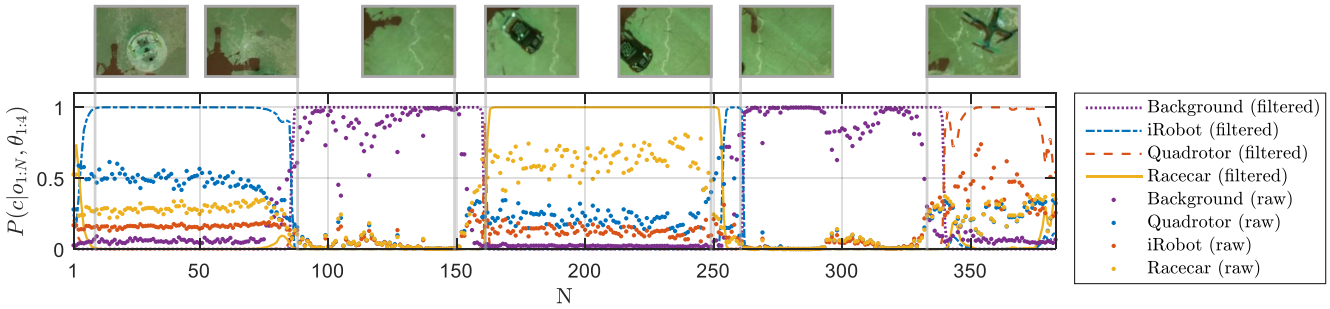


Fig. 10: HBNI-based filtered macro-observations onboard a moving quadrotor (example frames indicated).

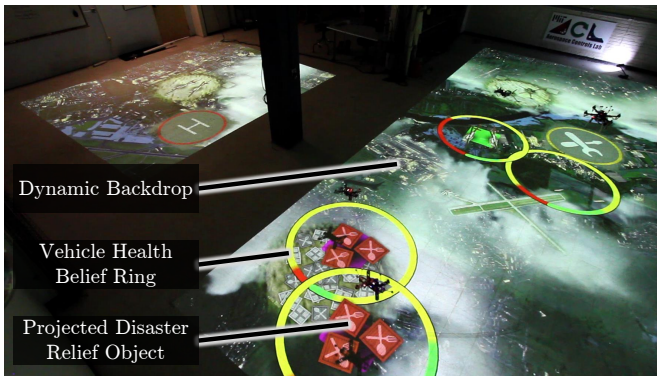


Fig. 11: Health-aware multi-quadrotor disaster relief domain, via macro-observation-based planning in dynamic environment. Video: https://youtu.be/XXYSAmdHn38.

robot semantic-level decision-making using Dec-POSMDPs.

VI. Conclusion

This paper presented a formalization of macro-observation processes used within Dec-POSMDPs, targeting scalability improvements for real-world robotics. A hierarchical Bayesian approach was used to model semantic-level macro-observations. This approach, HBNI, infers underlying noise distributions to increase classification accuracy, resulting in a generative macro-observation model. This is especially useful in robotics, where perception sensors are notoriously noisy. The approach was demonstrated in real-time on moving quadrotors, with classification and filtering performed onboard at approximately 20 frames per second. The novel macro-observation process was then integrated into a Dec-POSMDP and demonstrated in a probabilistic multi-robot health-aware disaster relief domain. Future work includes extension of existing Dec-POSMDP algorithms to online settings to leverage the computational-efficiency of HBNI.

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. Journal of Computer Vision (IJCV)*, pp. 1–42, April 2015.

[2] O. M. Mozos, P. Jensfelt, H. Zender, G.-J. Kruijff, and W. Burgard, "From labels to semantics: An integrated system for conceptual spatial representations of indoor environments for mobile robots," in *Proc. of the Workshop "Semantic Info. in Robotics" at IEEE ICRA*, April 2007.

[3] C. Galindo, J.-A. Fernández-Madrigal, J. González, and A. Saffiotti, "Robot task planning using semantic maps," *Robot. Auton. Syst.*, vol. 56, no. 11, pp. 955–966, November 2008.

[4] C. Wu, I. Lenz, and A. Saxena, "Hierarchical semantic labeling for task-relevant RGB-D perception." in *Robotics: Science and Systems*, D. Fox, L. E. Kavraki, and H. Kurniawati, Eds., 2014.

[5] C. Chanel, F. Teichteil-Knigsbuch, and C. Lesire, "Planning for perception and perceiving for decision POMDP-like online target detection and recognition for autonomous uavs," in *Proc. of the 6th Int. Scheduling and Planning Applications Workshop*, 2012.

[6] F. A. Oliehoek and C. Amato, *A Concise Introduction to Decentralized POMDPs*. Springer, 2016.

[7] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Math. of Oper. Research*, vol. 27, no. 4, pp. 819–840, 2002.

[8] D. S. Bernstein, C. Amato, E. A. Hansen, and S. Zilberstein, "Policy iteration for decentralized control of Markov decision processes," *Journal of Artificial Intelligence Research*, vol. 34, pp. 89–132, 2009.

[9] C. Amato, G. Konidaris, A. Anders, G. Cruz, J. How, and L. Kaelbling, "Policy search for multi-robot coordination under uncertainty," in *Robotics: Science and Systems XI (RSS)*, 2015.

[10] S. Omidshafiei, A.-A. Agha-Mohammadi, C. Amato, and J. P. How, "Decentralized control of partially observable markov decision processes using belief space macro-actions," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5962–5969.

[11] S. Omidshafiei, A.-A. Agha-Mohammadi, C. Amato, S.-Y. Liu, J. P. How, and J. Vian, "Graph-based cross entropy method for solving multi-robot decentralized pomdps," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5395–5402.

[12] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition." *IEEE Trans. on Systems, Man, and Cybern.*, vol. 22, no. 3, 1992.

[13] R. Florian and D. Yarowsky, "Modeling consensus: Classifier combination for word sense disambiguation," in *Proc. of the ACL-02 Conf. on Empir. Methods in Nat. Lang. Proc.*, vol. 10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 25–32.

[14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 2001.

[15] A. Coates and A. Y. Ng, "Multi-camera object detection for robotics." in *ICRA*. IEEE, 2010, pp. 412–419.

[16] M. Yguel, O. Aycard, and C. Laugier, "Update policy of dense maps: Efficient algorithms and sparse representation." in *FSR*, ser. Springer Tracts in Advanced Robotics, C. Laugier and R. Siegwart, Eds., vol. 42. Springer, 2007, pp. 23–33.

[17] I. J. Good, "Some history of the hierarchical Bayesian methodology," in *Bayesian Stat.*, J. M. Bernardo, M. H. DeGroot, D. V. Lindley, and A. F. M. Smith, Eds. Valencia University Press, 1980, pp. 489–519.

[18] J. Huggins and J. Tenenbaum, "Risk and regret of hierarchical bayesian learners." in *ICML*, ser. JMLR Proc., F. R. Bach and D. M. Blei, Eds., vol. 37, 2015, pp. 1442–1451.

[19] W. K. Hastings, "Monte Carlo methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.

[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[22] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.

[23] S. Omidshafiei, A. Agha-mohammadi, Y. F. Chen, N. K. Ure, S. Liu, B. Lopez, J. How, J. Vian, and R. Surati, "MAR-CPS: Measurable Augmented Reality for Prototyping Cyber-Physical Systems," in *IEEE CSM*, 2016.

[24] S. Omidshafiei, B. T. Lopez, J. P. How, and J. Vian, "Hierarchical bayesian noise inference for robust real-time probabilistic object classification," Tech. Rep., 2016, http://arxiv.org/abs/1605.01042.