Online Active Trajectory Classification for Motion-based Communication of Robots*

Haruki Nishimura
hnishimura@stanford.edu

June 6, 2016

1 Introduction

Designing cooperative multi-robot systems has gained much attention of researchers in the robotics community. Such multi-robot systems need a communication scheme to appropriately share the information among the team. While wireless electric communication is commonly used, it is not robust to adversarial jamming. One emerging alternative is motion-based communication: the idea that a message from the sender is encoded into its own trajectory and the receiver decodes the message by observing it. Some previous work [1][2] employed control theoretic approach to design a set of distinguishable and energy-optimal trajectories. However, it is also possible to think of a counter problem from the receiver's perspective; given a codebook of trajectories and the observations of the sender, how can we tune the receiver so that it can correctly distinguish the trajectories and decode the messages?

This problem is highly non-trivial especially when the receiver's observation model is monocular vision without depth perception because the relative attitude between two robots cannot be directly estimated from one observation. Although recent work [1][2][3][4] has explored the emerging field of motion-based communication, the use of monocular vision has yet to be addressed to the best of author's knowledge.

Thus, the present paper addresses a trajectory classification problem for motion-based communication between two robots using monocular vision only. The main controbutions of this study are two fold. First, we formulate the online classification problem in which both of the sender's message encoded in its trajectory and the receiver's relative position to it are sequentially estimated as the receiver moves around the sender. We provide a recursive Bayesian estimation algorithm to handle the multimodal distribution over the joint belief state. Gaussian approximation to the belief and model linearization lead to a Multi-hypothesis Extended Kalman Filter approach. Similar algorithms to ours can be found in [5][6][7]. Second, we employ an entropy minimization method to actively control the receiver's camera pose so that it can optimally move to minimize the expected uncertainty about the message over a one-step horizon. This type of information theoretic control for monocular vision is also studied in [8][9][10].

The rest of the paper is organized as follows. In the next section we formally state the problem and define the robot models. In Section 3 we derive the recursive Bayesian update formula and provide the Multi-hypothesis Extended Kalman Filter algorithm. We also formulate the active control policy of the receiver in Section 4. Simulation results are presented for 3-class classification in Section 5 with a comparison of the random and the active control policies. Conclusion with future direction is discussed in Section 6.

2 Preliminaries

The sender encodes a message into its pre-specified trajectory. The corrensponddence between the true trajectory and the message is known *a priori* as the trajectory codebook to both of the sender and the receiver; if the sender intends to send the message $z = i \in \{1, \ldots, N\}$, the corresponding trajectory $\tilde{\zeta}_i$ is chosen from the trajectory codebook $Z = \{\tilde{\zeta}_1, \ldots, \tilde{\zeta}_n\}$ and performed. However, the sender's trajectory is not necessarily directed to the receiver while the sender is engaed in its own task. This can bring ambiguity to the message decoding depending on the design of the trajectory codebook because the receiver's monocular vision is unable to determine the relative position between two robots from a single observation.

Hence, in order to correctly classify the trajectory and decode the message, the receiver is allowed to sequentially move around the sender and observe it repeating one particular class of the trajectories. This leads to a sequential state estimation

*The author has worked on the same project in AA 290 Spring 2016 with Prof. Mac Schwager as instructor.

and decision making problem. The Bayesian network structure of this problem is depicted in Figure 1. The receiver's position and attitude r is assumed to have the Markov property. η and u represents the observed trajectory and the control input to the receiver, respectively. In what follows, we will formulate the trajectory generation process, receiver's state transition model and the observation model.

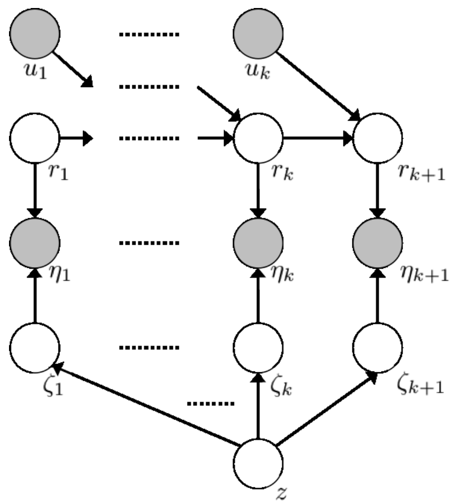


Figure 1: Bayesian network structure of the trajectory classification problem.

2.1 Trajectory generation

In this work, we assume that the smooth trajectory of the sender can be represented by a set of m points $\{\zeta^{(1)}, \ldots, \zeta^{(m)}\} \subset \mathbb{R}^3$, which are labeled by time. Therefore, the complete trajectory is given by a vector $\zeta = (\zeta^{(1)\mathrm{T}}, \ldots, \zeta^{(m)^\mathrm{T}})^\mathrm{T} \in \mathbb{R}^{3m}$. This is expressed in the sender's reference frame that is fixed in the inertial space. The value of m depends on the duration of the trajectory and the frame rate of the receiver's vision sensor, but here we restrict all the trajectories in the codebook to have the same duration so that the receiver cannot classify them based on it.

The sender chooses a message z and the corresponding trajectory $\tilde{\zeta}$ from the codebook. When it is executed, however, the sender's motion is subject to disturbance. We assume the resulting complete trajectory has a Gaussian distribution.

$$p(\zeta \mid z = i) = \mathcal{N}(\zeta; \tilde{\zeta}_i, Q) \tag{1}$$

2.2 State transition model of the receiver

The receiver's state r specifies its position and attitude with respect to the sender's reference frame, which is expressed in the receiver's camera frame.

$$r = (\omega^\mathrm{T}, t^\mathrm{T})^\mathrm{T} \in \mathbb{R}^6 \tag{2}$$

$\omega = (\omega_x, \omega_y, \omega_z)^\mathrm{T} \in \mathbb{R}^3$ gives the exponential coordinates on $SO(3)$, which specifies the axis of rotation around which the camera frame is rotated with respect to the reference frame. The norm of ω gives the magnitude of the rotation angle. Similarly, $t = (t_x, t_y, t_z)^\mathrm{T} \in \mathbb{R}^3$ represents the translation of the camera frame with respect to the reference frame.

The state transition is based on the current state and the control input u with Gaussian noise.

$$r_{k+1} = r_k + u_k + w, \quad w \sim \mathcal{N}(r; 0, R) \in \mathbb{R}^6 \tag{3}$$

Note that the receiver will move only after it observes one complete trajectory performed by the sender.

2.3 Observation model

The details of the pinhole camera model is omitted here for saving space and the reader is referred to [11][12]. The pinhole camera model projects the trajectory onto the image plane of the camera. The resulting observation $\eta = (\eta^{(1)\mathrm{T}}, \ldots, \eta^{(m)\mathrm{T}})^\mathrm{T} \in \mathbb{R}^{2m}$ is given by a nonlinear projection function $f(\cdot,\cdot) : \mathbb{R}^6 \times \mathbb{R}^{3m} \to \mathbb{R}^{2m}$ that takes r and ζ as its arguments. The correspondence between $\zeta^{(i)} \in \mathbb{R}^3$ and $\eta^{(i)} \in \mathbb{R}^2$ is assumed to be known to the receiver.

Analogous to (3), the actual observation is subject to Gaussian noise.

$$\eta = f(r, \zeta) + v, \quad v \sim \mathcal{N}(\eta; 0, S) \in \mathbb{R}^{2m} \tag{4}$$

3 Bayesian Online Learning for State Estimation

3.1 Recursive Bayesian estimation formula

We are interested in estimating the joint distribution over r and z given the history of observations and control inputs: $p(r_{k+1}, z \mid \eta_{1:k+1}, u_{1:k})$. Leveraging the Bayes's rule and the structure of the Bayesian network, it can be decoupled and simplified as follows.

$$p(r_{k+1}, z \mid \eta_{1:k+1}, u_{1:k}) \propto \int p(\eta_{k+1} \mid r_{k+1}, \zeta_{k+1}) p(\zeta_{k+1} \mid z) d\zeta_k \times \int p(r_{k+1} \mid r_k, u_k) p(r_k \mid \eta_{1:k}, u_{1:k-1}, z) dr_k \\ \times \int p(\eta_{k+1} \mid r_{k+1}, z) p(r_{k+1} \mid \eta_{1:k}, u_{1:k}, z) dr_{k+1} \times P(z \mid \eta_{1:k}, u_{1:k-1}). \tag{5}$$

(5) implies that we can separately update our belief of r conditional on z and the belief of z itself, given the state transition model and the observation model. This allows us to derive the closed-form update formula discussed in the next section.

3.2 Multi-hypothesis Extended Kalman Filter algorithm

The problem of estimating the joint distribution over r and z given past observations and control inputs can be viewed as multi-hypotheses filtering. Thus we will adapt the Multi-hypothesis Extended Kalman Filter algorithm, which is a common parametric filter to handle multimodal distributions with linear model approximation. In our problem, note that the observation function takes two arguments r and ζ. Therefore, the first order taylor expansion around the current estimate $\mathbb{E}[r_{k+1} \mid \eta_{1:k}, u_{1:k}, z = i] \triangleq \bar{\mu}^{(i)}_{k+1}$ and $\mathbb{E}[\zeta_{k+1} \mid z = i] = \tilde{\zeta}_i$ is given by,

$$f(r_{k+1}, \zeta_{k+1}) \mid z = i \approx f(\bar{\mu}^{(i)}_{k+1}, \tilde{\zeta}_i) + \frac{\partial}{\partial r_{k+1}} f(r_{k+1}, \tilde{\zeta}_i)\bigg|_{\bar{\mu}^{(i)}_{k+1}} (r_{k+1} - \bar{\mu}^{(i)}_{k+1}) + \frac{\partial}{\partial \zeta_{k+1}} f(\bar{\mu}^{(i)}_{k+1}, \zeta_{k+1})\bigg|_{\tilde{\zeta}_i} (\zeta_{k+1} - \tilde{\zeta}_i). \tag{6}$$

Denoting $\frac{\partial}{\partial r_{k+1}} f(r_{k+1}, \tilde{\zeta}_i)|_{\bar{\mu}^{(i)}_{k+1}} \triangleq F^{(i)}_{k+1} \in \mathbb{R}^{2m \times 6}$ and $\frac{\partial}{\partial \zeta_{k+1}} f(\bar{\mu}^{(i)}_{k+1}, \zeta_{k+1})|_{\tilde{\zeta}_i} \triangleq G^{(i)}_{k+1} \in \mathbb{R}^{2m \times 3m}$, we obtain the following linearized observation model.

$$f(r_{k+1}, \zeta_{k+1}) \mid z = i \approx f(\bar{\mu}^{(i)}_{k+1}, \tilde{\zeta}_i) + F^{(i)}_{k+1}(r_{k+1} - \bar{\mu}^{(i)}_{k+1}) + G^{(i)}_{k+1}(\zeta_{k+1} - \tilde{\zeta}_i) \tag{7}$$

In order to derive the Extended Kalman Filter update formula, we further assume that the prior belief of r conditional on z before taking an action u_k and observing η_{k+1} is a Gaussian distribution.

$$p(r_k \mid \eta_{1:k}, u_{1:k-1}, z = i) \triangleq \mathcal{N}(r_k; \mu^{(i)}_k, \Sigma^{(i)}_k) \tag{8}$$

Substituting (1), (3), (4), (7) and (8) into the update formula (5), we derive Algorithm 1. We see that the categorical distribution over z is modified in line 10 based on how well the hypothesis $z = i$ can explain the actual observation.

3.3 Beleif initialization

The convergence of the Extended Kalman Filter algorithm is sensitive to the accuracy of the prior $p(r_1, z \mid \eta_1)$. Fortunately, estimating the means $\mu^{(i)}_1$ and the categorical distribution parameters $\phi^{(i)}_1$ can be evaluated by the maximum likelihood parameter fitting, which is well studied in the computer vision literature. Specifically, $\mu^{(i)}_1$ can be estimated by the Direct Linear Transformation algorithm and the Levenberg-Marquardt algorithm to minimize the nonlinear least squares $||\eta_1 - f(\mu^{(i)}_1, \tilde{\zeta}_i)||^2_2$ [11]. $\phi^{(i)}_1$ can be also found by comparing their residuals, assuming the same initial covariance matrix $\Sigma^{(i)}_1$ for all $i \in \{1, \dots, N\}$.

4 Active-vision Control for Entropy Minimization

Given the current esimate of the state, our goal is to control the position and the attitude of the receiver in order to correctly classify the trajectory and decode the message. One information theoretic approach is to select u_k from the control space $\mathcal{U}$ so that the expected entropy of z taken over the next obervation η_{k+1} is minimized. Formally, the control objective is given by,

$$\mathbb{E}_{\eta_{k+1}}\left[H(z \mid \eta_{1:k+1}, u_{1:k}) \mid \eta_{1:k}, u_{1:k}\right] = \int p(\eta_{k+1} \mid \eta_{1:k}, u_{1:k}) H(z \mid \eta_{1:k+1}, u_{1:k}) d\eta_{k+1}. \tag{9}$$

input : $\mu_k^{(i)} \in \mathbb{R}^6$, $\Sigma_k^{(i)} \in \mathbb{R}^{6\times 6}$, $\phi_k^{(i)} \triangleq P(z = i \mid \eta_{1:k}, u_{1:k-1}) \in \mathbb{R} \ \ \forall i \in \{1, \ldots, N\}$, $u_k \in \mathbb{R}^6$, $\eta_{k+1} \in \mathbb{R}^{2m}, Z = \{\tilde{\zeta}_1, \ldots, \tilde{\zeta}_n\}$
output: $\mu_{k+1}^{(i)} \in \mathbb{R}^6$, $\Sigma_{k+1}^{(i)} \in \mathbb{R}^{6\times 6}$, $\phi_{k+1}^{(i)} \in \mathbb{R} \ \ \forall i \in \{1, \ldots, N\}$

for *each* $i \in \{1, \ldots, N\}$ **do**
- $\bar{\mu}_{k+1}^{(i)} \leftarrow \mu_k^{(i)} + u_k$
- $\bar{\Sigma}_{k+1}^{(i)} \leftarrow \Sigma_k^{(i)} + R$
- $F_{k+1}^{(i)} \leftarrow \frac{\partial}{\partial r_{k+1}} f(r_{k+1}, \tilde{\zeta}_i)\Big|_{\bar{\mu}_{k+1}^{(i)}}$
- $G_{k+1}^{(i)} \leftarrow \frac{\partial}{\partial \zeta_{k+1}} f(\bar{\mu}_{k+1}^{(i)}, \zeta_{k+1})\Big|_{\tilde{\zeta}_i}$
- $H_{k+1}^{(i)} \leftarrow S + G_{k+1}^{(i)} Q G_{k+1}^{(i)\mathrm{T}} + F_{k+1}^{(i)} \bar{\Sigma}_{k+1}^{(i)} F_{k+1}^{(i)\mathrm{T}}$
- $K_{k+1}^{(i)} \leftarrow \bar{\Sigma}_{k+1}^{(i)} F_{k+1}^{(i)\mathrm{T}} H_{k+1}^{(i)-1}$
- $\mu_{k+1}^{(i)} \leftarrow \bar{\mu}_{k+1}^{(i)} + K_{k+1}^{(i)} \left(\eta_{k+1} - f(\bar{\mu}_{k+1}^{(i)}, \tilde{\zeta}_i)\right)$
- $\Sigma_{k+1}^{(i)} \leftarrow (I - K_{k+1}^{(i)} F_{k+1}^{(i)}) \bar{\Sigma}_{k+1}^{(i)}$
- $\phi_{k+1} \leftarrow \mathcal{N}\left(\eta_{k+1}; f(\bar{\mu}_{k+1}^{(i)}, \tilde{\zeta}_i), H_{k+1}^{(i)}\right) \phi_k^{(i)}$

end

$c \leftarrow \sum_{i=1}^{N} \phi_{k+1}^{(i)}$

for *each* $i \in \{1, \ldots, N\}$ **do**
- $\phi_{k+1}^{(i)} \leftarrow \phi_{k+1}^{(i)} / c$

end

Algorithm 1: Multi-hypothesis Extended Kalman Filter algorithm for the trajectory classification problem.

By substituting the Kalman update formulas and extracting the terms dependent on u_k only, this yields the following formula.

$$\sum_{i=1}^{N} \phi_k^{(i)} \log |2\pi H_{k+1}^{(i)}| + \int \left(\sum_{i=1}^{N} \phi_k^{(i)} \mathcal{N}(\eta_{k+1}; f(\bar{\mu}_{k+1}^{(i)}, \tilde{\zeta}_i), H_{k+1}^{(i)}) \right) \log \left(\sum_{i=1}^{N} \phi_k^{(i)} \mathcal{N}(\eta_{k+1}; f(\bar{\mu}_{k+1}^{(i)}, \tilde{\zeta}_i), H_{k+1}^{(i)}) \right) d\eta_{k+1} \tag{10}$$

The first term in (10) follows from the entropy of Gaussian distributions. The second term is the negative entropy of a Gaussian mixture and exact solution is not available. However, Huber et al. [13] provide an approximation method based on the Taylor expansion and we will employ their first order approximation formula. The resulting control objective is presented below. Note that this function is independent of the actual observation η_{k+1}, thus it can be evaluated before taking the action u_k.

$$J(u_k \mid \eta_{1:k}, u_{1:k-1}) = \sum_{i=1}^{N} \phi_k^{(i)} \left[\log |2\pi H_{k+1}^{(i)}| + \log \left(\sum_{j=1}^{N} \phi_k^{(j)} \mathcal{N}(\eta_{k+1} = f(\bar{\mu}_{k+1}^{(i)}, \tilde{\zeta}_i); f(\bar{\mu}_{k+1}^{(j)}, \tilde{\zeta}_j), H_{k+1}^{(j)}) \right) \right] \tag{11}$$

$$u_k = \arg\min_{u \in \mathcal{U}} J(u \mid \eta_{1:k}, u_{1:k-1}) \tag{12}$$

5 Simulation Results

Figure 2 shows the trajectory codebook with one circular trajectory and two different elliptic trajectories used in the simulation. All the trajectories were assumed to be two dimensional. As can be seen in Figure 3, all of the three trajectory classes fit equally well to the observed image on the image plane of 100x100 pixels. With the initial covariance $\Sigma_1^{(i)} = 10000 I_{6\times 6} \ \forall i \in \{1, 2, 3\}$, the prior parameters were $\phi_1^{(1)} = 0.3414$, $\phi_1^{(2)} = 0.3404$ and $\phi_1^{(3)} = 0.3183$ while the true trajectory class was 1. As the simulation was run, the receiver chose the control inputs according to (12) with $\mathcal{U} = \{(-4.0, -4.0, -4.0, 0, 0, 0), (-2.0, -4.0, -4.0, 0, 0, 0), \ldots, (4.0, 4.0, 4.0, 0, 0, 0)\}$. The translational inputs were all set to 0 so that the camera axis was always pointed to the trajectories. Observed images at $k = 2$ through $k = 5$ are presented in Figure 4. In this simulation, the configuration of the receiver was converged to the perpendicular position to the sender's trajectory, as shown in Figure 4d. The posterior parameters at $k = 5$ were $\phi_5^{(1)} = 1.0$, $\phi_5^{(2)} = 0.0$, $\phi_5^{(3)} = 0.0$, resulting in the correct classification.

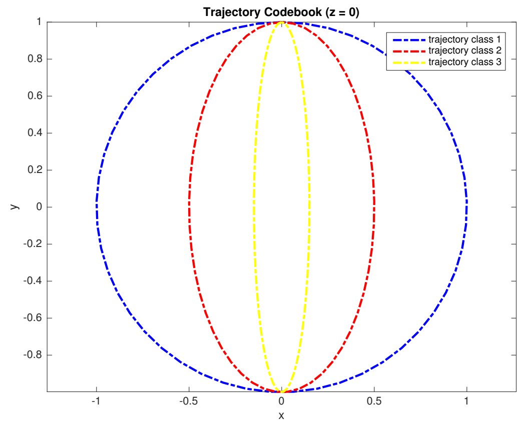


Figure 2: The trajectory codebook used in the simulation.

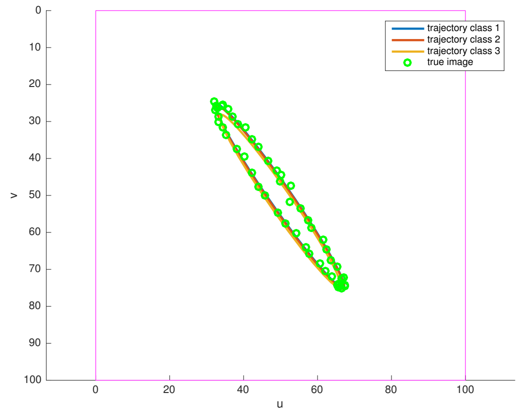
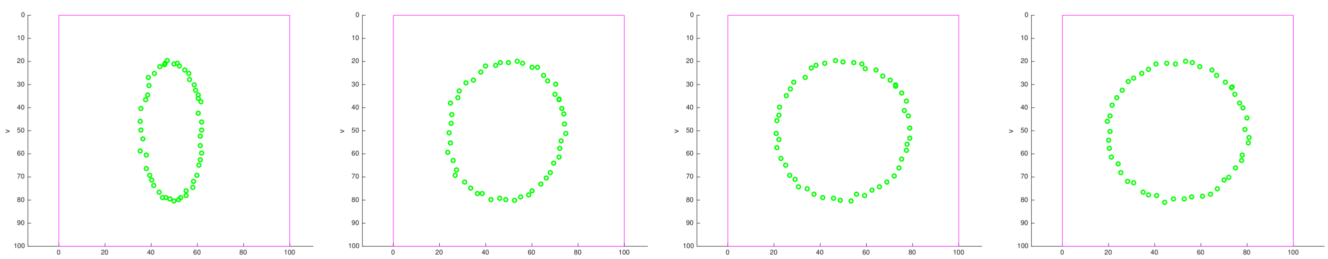
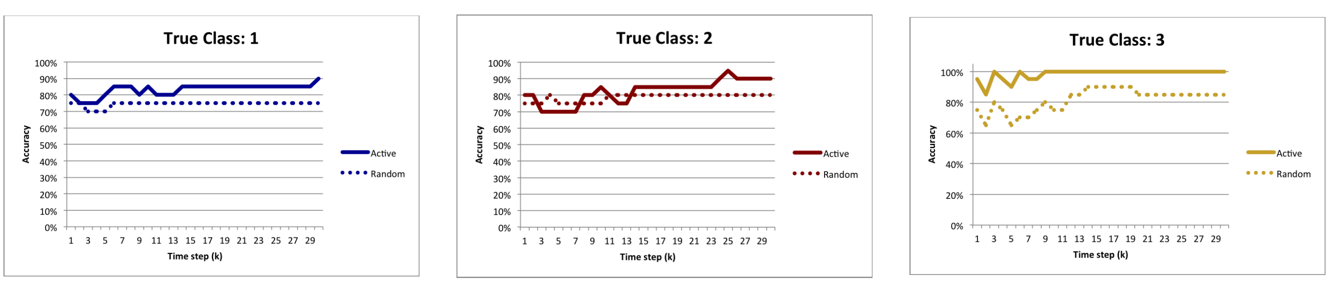


Figure 3: The initial observation and the fitted trajectories.



(a) $k = 2$ (b) $k = 3$ (c) $k = 4$ (d) $k = 5$

Figure 4: Observed images at different time steps k.



(a) Class 1 (b) Class 2 (c) Class 3

Figure 5: Classification accuracy averaged over 20 simulations.

We also compared the performance of the active control policy to the random policy based on 20 simulations. In each simulation, the true position of the receiver was randomly initialized. As Figure 5 illustrates, the active control policy always outperformed the random policy after $k = 15$ for all trajectory classes. In those simulations, we also observed that the receiver converged to the perpendicular configuration to the trajectory when the classification was correct.

6 Conclusion

In this paper we have presented an online trajectory classification algorithm for motion-based communication between two robots with monocular vision. Bayesian update formulas were derived in the form of Multi-hypothesis Extended Kalman Filter. We have also derived the active control algorithm and demonstrated in simulations that the proposed framework outperforms the random control policy.

In future research we intend to concentrate on two main aspects to extend the proposed approach. First, the use of Extended Kalman Filter might not be the best algorithm to implement the Bayesian state estimation. We will also adapt other filtering algorithms such as Unscented Kalman Filter and Particle Filter. Second, we will apply this algorithm not only to 2D trajectories but to 3D trajectories in order to evaluate the general performance of our method in 3D space.

References

[1] J. Baillieul and K. Özcimder, “The Control Theory of Motion-Based Communication: Problems in Teaching Robots to Dance,” p. 20, sep 2011.

[2] A. Jones and S. Andersson, “A motion-based communication system,” in *2013 American Control Conference*. IEEE, jun 2013, pp. 365–370.

[3] D. Raghunathan and J. Baillieul, “Motion based communication channels between mobile robots - A novel paradigm for low bandwidth information exchange,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, oct 2009, pp. 702–708.

[4] ——, “Exploiting information content in relative motion,” in *2009 American Control Conference*. IEEE, 2009, pp. 2166–2171.

[5] D. L. Alspach and H. W. Sorenson, “Nonlinear bayesian estimation using gaussian sum approximations,” *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, 1972.

[6] R. Chen and J. S. Liu, “Mixture Kalman filters,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, no. 3, pp. 493–508, aug 2000.

[7] G. Jochmann, S. Kerner, S. Tasse, and O. Urbann, “Efficient multi-hypotheses unscented kalman filtering for robust localization,” in *Robot Soccer World Cup XV*. Springer-Verlag, jun 2012, pp. 222–233.

[8] J. Denzler and C. Brown, “Optimal selection of camera parameters for state estimation of static systems: an information theoretic approach,” *Technical Report*, p. 40, 2000.

[9] M. Zobel, J. Denzler, and H. Niemann, “Entropy based camera control for visual object tracking,” in *Proceedings. International Conference on Image Processing*, vol. 1. IEEE, 2002, pp. 901–904.

[10] A. Seekircher, T. Laue, and T. Röfer, “Entropy-based active vision for a humanoid soccer robot,” in *Robot Soccer World Cup XIV*, 2011, vol. 6556 LNAI, pp. 1–12.

[11] V. Lepetit and P. Fua, “Monocular Model-Based 3D Tracking of Rigid Objects: A Survey,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, jan 2005.

[12] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[13] M. F. Huber, T. Bailey, H. Durrant-Whyte, and U. D. Hanebeck, “On entropy approximation for Gaussian mixture random vectors,” in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. IEEE, aug 2008, pp. 181–188.