Fault-tolerant Probabilistic Sensor Fusion for Multi-Agent Systems

Abdolkarim Pahliani   Matthijs T. J. Spaan   Pedro U. Lima

Abstract— In this work we focus on the problem of probabilistic sensor fusion in Multi-Robot Multi-Sensor Systems (MRMS), taking into account that some sensors might fail or produce erroneous information. We study fusion methods that can successfully cope with situations of agreement, partial agreement, and disagreement between sensors. We define a set of specifications for fusion methods appropriate for MRMS environments. In light of these specifications, we review two popular algorithms for probabilistic sensor fusion, Linear Opinion Pool (LOP) and Logarithmic Opinion Pool (LGP). To overcome difficulties of applying them to a MRMS setting, a new method is introduced, p-norm Opinion Pool (POP). Comparing to LOP and LGP, POP is more compatible with the specifications and more flexible, successfully handling situations of agreement and disagreement between sensors. Through simulation and real-world experiments, we check performance of the POP and compare it with LOP and LGP. We also implement a real-world experiment through which the performance of POP is examined.

I. INTRODUCTION

Autonomous robots depend on information provided by their sensors to interact with the environment. However, sensors in general provide noisy and uncertain information, and might even fail completely in certain situations. Sensor fusion is a strategy to reduce inherent uncertainty of information provided by sensors. In our work, the goal is to find an axiomatic fusion method to combine the measurements received from several probabilistic sensors, taking into account that some of them might fail. Fusion of such information can lead to disagreement among sensors and generates large error.

To motivate our work, the following example shows how some fusion methods, e.g., using Weighted Linear Combination of observations (WLC [1]) [1] and standard Bayesian methods are sensitive to sensor failure. In this example, we attempt to track a robot by the information provided by the robot odometry sensor and a number of ceiling mounted cameras. All information is routed to a central processing unit. Sensors noise is considered to be Gaussian. To estimate the robot position, first, the information received from the cameras is fused. Then, by using a filter, the fused

information and the odometry, the position of the robot is estimated. In Fig.1(a), the real path and two estimated paths are shown. The first estimation is obtained when WLC is used to fuse the information and standard Kalman filter is used to determine the final estimation. For the second estimation, a standard Bayesian method is used to fuse the information and particle filter is used to estimate the path. In this simulation, we assume that the sensors are unbiased. As we can see from the figure, the estimated path is a good approximation of the actual path. Now, to see the effect of sensor failure, 30% of the cameras observing the robot are biased. A biased sensor is considered as a model for sensor failure. Fig.1(b) shows the real and estimated paths. Similar examples are ran 100 times. Comparing to the previous case, if we use WLC and the Kalman filter, we get a 10 times increase in estimation error and if we use the standard Bayesian fusion method and particle filter, we have 17 times increase in error. This shows that excluding the erroneous measurements and resolving the disagreement is a crucial issue.

In this work, first, we define the basic requirements of a fusion method. By that we mean the properties that a fusion method should have in order to produce desirable results. An important issue in information fusion in such a environment is sensor disagreement and failure. The sensor fusion method should be able to recognize and filter out these faulty measurements and also resolve disagreement among the information sensors.

Linear Opinion Pool (LOP) and Logarithmic Opinion Pool (LGP) are two popular probabilistic sensor fusion methods which have been widely utilized [10], [11], [13], [14]. However, using these methods in a MRMS environment might be problematic. To overcome their disadvantages, we introduce a probabilistic sensor fusion algorithm for fusion of information which generalizes LOP and LGP. In particular, we use *p-norm* (also called *lp-norm*) to fuse the information. We call this method *p-norm opinion pool* (**POP**) and check its compliance with the defined specifications.

Throughout the paper, we assume a probabilistic model for sensors. In other words, sensor delivers its measurement in the form of a probability mass function (pmf). Moreover, all observations are assumed to refer to a common global frame. We also assume that the majority of sensors is unbiased. This is a very weak assumption comparing to assumption (all sensors are unbiased and independent) [1].

This paper is organized as follows. First, some basic concepts are explained in Section II. The fusion problem and the fusion specifications are defined in Section III. In this section, we also explain LOP and LGP methods,

[1]For N observations with estimations $\widehat{x_1}, ..., \widehat{x_N}$, the fused observation estimate is: $\widehat{x} = [\sum_{i=1}^{N} \Sigma_i^{-1}]^{-1}[\sum_{i=1}^{N} \Sigma_i^{-1}\widehat{x_i}]$. Σ_i is the covariance matrix of the ith observation.
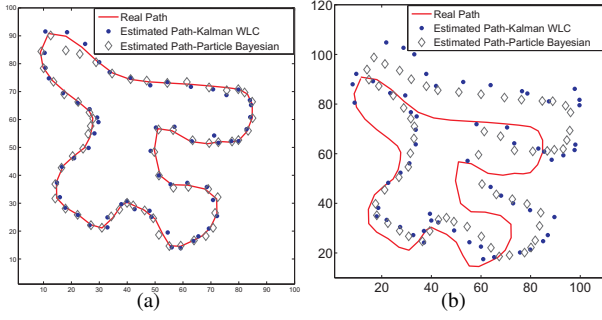
Fig. 1. Results of the motivation example. A tracking scenerio is considered in which a robot is tracked by a number of fixed cameras. The left plot is the result of tracking using two well known methods when sensors are unbiased. The right plot is the result of tracking when 30% of the sensors are biased.

evaluate the compatibility of these two fusion methods with respect to the specifications. In the next section, the **POP** method is introduced and properties of this method are examined. In this section we compare the performance of the **POP** method with the other two. In Section V, we explain the experiments and results. In Section VI we present our conclusions and suggest future work.

II. DEFINITIONS

Here and henceforth, we assume the information provided by the sensors is a pmf over the state space. We use a probability column vector to represent the pmf. For instance, sensor i provides a pmf $\mathbf{P}_i$ over the state space as its measurement. Each element of the vector is a probability assigned to a state. Therefore, $\mathbf{P}_i = [P_{i1}, ..., P_{iM}]$ denotes the measurement of ith sensor, P_{ij} is the probability assigned by i^{th} sensor to state j, $\sum_{j=1}^{M} P_{ij} = 1$, $0 \leq P_{ij} \leq 1$ and M is number of states in the state space.

A. Agreement and Disagreement

There is agreement between the observations $\mathbf{P}_1$ and $\mathbf{P}_2$ of two sensors if:

$$d(\mathbf{P}_1, \mathbf{P}_2) \leq \xi \tag{1}$$

where $d(.)$ is a distance measure between the two observations and ξ is a positive number. The distance measure can be defined as Kullback-Leibler distance [2], Bhattacharrya distance [3] or other. Otherwise we say that the two sensors disagree.

B. Measure of Observations Uncertainty

A major problem in sensor fusion is how to measure observations uncertainty. When the uncertainty has Gaussian form, the covariance matrix can represent the uncertainty. When the form of uncertainty is non-Gaussian, we use entropy [4] to measure the amount of uncertainty, e.g., for $\mathbf{P}_i$:

$$H(\mathbf{P}_i) = -\sum_{j}^{M} P_{ij} \ln P_{ij} \tag{2}$$

Entropy is a positive number. When entropy is zero we are fully certain about the outcome of the sensor. As the value of H increases we become less certain.

C. Measure of Observation Quality

We assign a weight to each sensor observation. Weights express our certainty in observations. We assign larger weights to less uncertain observations when fusing them. Entropy can be interpreted as the quantification of the uncertainty associated to an observation [5], [6], [7]. For N measurements $\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_N$, a weight w_i is assigned to observation $\mathbf{P}_i$ such that:

$$w_i = \frac{1}{H(\mathbf{P}_i)^k} \tag{3}$$

where $1 \leq i \leq N$ and $k \geq 0$. Larger k assign larger relative importance to more certain observations.

III. PROBLEM STATEMENT AND PROPOSED FUSION METHOD

A. Problem statement

Consider a scenario where a group of N distributed sensors are measuring a quantity of interest. In general, the sensor measurements are assumed to be uncertain. Sensor i provides a pmf P_i over the state space as its measurement model. The sensor observations are fused together using fusion method f:

$$f : \mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_N \longmapsto \mathbf{P} \tag{4}$$

where $\mathbf{P}$ is also a pmf. The main problem is to find an appropriate function f. Actually, there can be an unlimited number of candidates for function f. It can be a simple analytical function, e.g., arithmetic or geometric mean. To limit the number of candidates, we define a set of specifications. Based on these specifications, we evaluate the fusion method.

1) The fusion specifications: Now, we define a set of specifications that a fusion algorithm should comply with. These specifications play the role of constraints that limit the number of potential candidates for f in (4). These specifications are considered for a MRMS environment.

Based on these specifications, we evaluate the fusion method. In general, it might not be possible to find a function that fully satisfies all of them. So, we should find a fusion method that has maximum compatibility with the specifications.

1-**Uncertainty reduction**: A fusion rule should be defined in such a way that the uncertainty of fused observation is less than minimum uncertainty of the original observations if there is a general agreement among observations. For two observations $\mathbf{P}_1, \mathbf{P}_2$:

$$d(\mathbf{P_1}, \mathbf{P_2}) \leq \xi \Rightarrow \Psi(f(\mathbf{P_1}, \mathbf{P_2})) < min(\Psi(\mathbf{P_1}), \Psi(\mathbf{P_2})) \tag{5}$$

where Ψ is a measure for the uncertainty, e.g. the entropy and min stands for minimum. If there is disagreement among the measurements, depending on the number of disagreements and amount of disagreement, uncertainty of fused observation can be larger than the uncertainty of the best observation. Fusing inconsistent observations creates more hypotheses, which cause an increase in uncertainty of the fused observation.

2-**Fault tolerance**: One important issue in sensor fusion is detecting and excluding faulty measurements. Faulty measurements are defined as measurements whose distance from actual value is more than a given threshold. To come up with this difficulty, we assume the majority of the sensors are unbiased.

3-**Zero preservation**: If probability of the fused measurement for a state is zero then all sensors should assign zero probability to the state.

$$P_j = f(P_{1j}, ..., P_{Nj}) = 0 \quad \Rightarrow \quad \forall i \ \ P_{ij} = 0 \quad (6)$$

where $\mathbf{P}_{ij}$ represents the probability that its sensor assigns to the state.

4-**Non-zero preservation**: If the probability of fused observation for a state is non-zero then then at least one of the measurements should assign a non-zero probability to the state.

$$P_j = f(P_{1j}, ..., P_{Nj}) \neq 0 \quad \Rightarrow \quad \exists i \ \ P_{ij} \neq 0 \quad (7)$$

By providing an example, we show the importance of this specification. Consider N sensors are measuring a quantity of interest. Suppose all of them except one (which assigns a zero probability for a particular state) report a non-zero probability for that state. Logically, the fused observation should not assign a zero probability to the state, specially if N is large. This property is very important, especially when some sensors are faulty but we can not tell which ones. Unfortunately, some fusion approaches, e.g., Bayesian-like approaches do not comply with this specification.

5-**Sequential updating**: a fusion specification is called sequentialy updating if the result of fusing N sensor measurements is the same as fusing $N-1$ sensor measurements with an Nth sensor measurement.

$$f(\mathbf{P}_1, ..., \mathbf{P}_{N-1}, \mathbf{P}_N) = f(f(\mathbf{P}_1, ..., \mathbf{P}_{N-1}), \mathbf{P}_N) \quad (8)$$

6-**Monotonicity**: Let $\mathbf{P} = f(\mathbf{P}_1, ..., \mathbf{P}_i, ..., \mathbf{P}_N)$ and $\mathbf{P}' = f(\mathbf{P}_1, ..., \mathbf{P}'_i, ..., \mathbf{P}_N)$. f is monotone if $P_{ij} < P'_{ij}$ then $P_j < P'_j$. In other words, this property says any change in one of the measurements should directly and proportionally affect the fused observations.

B. Linear and Logarithmic Opinion Pool

1) Logarithmic Opinion Pool: If we assign a weight to each observation, measuring its uncertainty, then, for N observations, LGP [13], [14] is defined as:

$$\mathbf{P} = \frac{\mathbf{P}_1^{w_1} \circ \mathbf{P}_2^{w_2} \circ ... \circ \mathbf{P}_N^{w_N}}{(\mathbf{P}_1^{w_1} \circ \mathbf{P}_2^{w_2} \circ ... \circ \mathbf{P}_N^{w_N})\mathbf{e}} \quad (9)$$

where $\mathbf{e} = [1, 1, ..., 1]^T$ is a $1 \times m$ vector, w_is are weights as in (3), and $\circ$ is the Hadamard product[2] of two matrices. The standard Bayesian fusion is a special case of LGP when all $w'_i s$ are set to 1.

---

[2]For two matrices A and B, Hadamard product of two matrices $A = [a_{ij}]$ and $B = [b_{ij}]$ of the same size is just their element-wise product $A \circ B \equiv [a_{ij}b_{ij}]$.

2) Linear Opinion Pool: For N observations $\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_N$, LOP [10], [11] is defined as:

$$\mathbf{P} = \sum_{i=1}^{n} w_i \mathbf{P}_i \quad (10)$$

where $\mathbf{P}$ is a PV and w_is are weights $w_i \geq 0$ that add up to one, as in (3).

3) LOP and LGP compliance: LOP complies with monotonicity, sequential updating, zero preservation and non-zero preservation specifications. Despite that, the uncertainty reduction and fault tolerance are not satisfied. LGP complies with monotonicity, sequential updating, zero preservation and uncertainty reduction specifications. However, it does not satisfy non-zero preservation and fault tolerance.

IV. PROPOSED FUSION METHOD

In this section, first we propose and formalize the **POP** method. Then, we check and compare **POP** compliance with LOP and LGP. Here, we also introduce a method to automatically adjust and set the **POP** parameters. We also check the **POP** method in presence of sensor failure through the simulation.

A. P-Norm Opinion Pool

In subsection III-A.1 a set of specifications is defined to ensure the fusion function possesses certain properties and found out that LOP and LGP do not satisfy some of the specifications. Besides these, in a MRMS, the fusion specification should be able to cope with different situations. For example, in situations where there is a general disagreement among the sensors, LOP is preferred over LGP (because LGP does not comply with non-zero preservation). There are situations where there is agreement among the sensors and LGP is preferred (because LOP does not comply with uncertainty reduction). However, in some situations using only one of them may not be the solution. To overcome the difficulties of LOP and LGP and also provide a more generic and flexible probabilistic fusion method appropriate for a MRMS environment, another probabilistic fusion method is introduced.

Here, first, we formalize the fusion problem as an optimization problem. After checking the compliance with the defined specifications, the solution can be used as a fusion method.

As we mentioned before, the output of the sensors is considered as a pmf. Let $\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_N$ be a set of PV related to observations of N sensors $S_1, S_2, ..., S_N$ respectively. Let $\mathbf{P}$ represents the fused pmf. Let $d(.)$ represents the distance between two pmfs. We define the following objective function:

$$f = \sum_{i=1}^{N} w_i d(\mathbf{P}, \mathbf{P}_i) \quad (11)$$

where $d(\mathbf{P}, \mathbf{P}_i)$ is the distance between the fused measurement and i^{th} sensor measurement and $0 \leq w_i \leq 1$ and $\sum_{i=1}^{N} w_i = 1$. w_i is a weight assigned to $d(\mathbf{P}, \mathbf{P}_i)$ to represents the importance of the ith sensor measurement over

other measurements. Actually, f is a function that measures average weighted distance between the original observation and the fused one. If f is minimized, $\mathbf{P}$ will become the closest pmf to all the measurements:

$$\mathbf{P} = argmin_{\mathbf{P}}\ f = argmin_{\mathbf{P}} \sum_{i=1}^{N} w_i d(\mathbf{P}, \mathbf{P}_i) \qquad (12)$$

Now, the problem is to consider an appropriate distance function. It can be defined as:

$$d(\mathbf{P}, \mathbf{P}_i) = (\mathbf{P}^\kappa - \mathbf{P}_i^\kappa)(\mathbf{P}^\kappa - \mathbf{P}_i^\kappa)^T \qquad (13)$$

and $-\infty \leq \kappa \leq +\infty$. [3] Here κ plays an important role. It allows us to conclude the different fusion rules under a similar framework. Replacing (13) in (12) and solving the minimization problem results in:

$$f = \sum_{i=1}^{N} w_i (\mathbf{P}^\kappa - \mathbf{P}_i^\kappa)(\mathbf{P}^\kappa - \mathbf{P}_i^\kappa)^T$$

$$\mathbf{P} = \mathbf{POP}(\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_N) = \eta(\sum_{i=1}^{N}(w_i \mathbf{P}_i^\kappa))^{\kappa^{-1}} \qquad (14)$$

It is designated as P-norm Opinion Pool (**POP**). η is a normalization factor that is determined by the requirement that $\mathbf{P}$ is a PV. Once again, w_is weight the measurement uncertainty of sensor i.

LGP and LOP are particular cases of **POP**. In fact, if we make $\kappa \to 0$, we obtain LGP, while if we make $\kappa = 1$, we get LOP.

Method ╲ Specification	LOP	LGP	**POP**
$Uncertainty\ reduction$	NO	YES	YES
$Fault\ tolerance$	NO	NO	YES
$Zero\ preservation$	YES	YES	YES
$Non-zero\ preservation$	YES	NO	YES
$Sequential\ updating$	YES	YES	YES
$Monotonicity$	YES	YES	YES

TABLE I

COMPARING PROBABILISTIC FUSION METHOD PROPERTIES

B. **POP** compliance

POP satisfies monotonicity, sequential updating, zero-preservation, non-zero preservation, uncertainty reduction and fault tolerance. In Table (I), the compliance of LOP, LGP and **POP** are specified.

---

[3] A^b raise each element of A, A(i) to the power b (vector element-wise power operator).

Algorithm set $\kappa(O, \alpha, \beta)$

A: Group The Observations
 To each observation $o_k \in O$ assign an index i and let $I = \{i\}$, $1 \leq i \leq N$ be the set of indexes.
 Calculate $D = \{d(o_j, o_k)\}$, $1 \leq j, k \leq N$, $j \neq k$ where $d(.)$ is a distance (e.g., Bhattacharrya distance [3]) and $o_j, o_k \in O$.
 forever do
  Find the closest observations (the smallest member of set D, $d_{min} = argminD$) and return the observation indexes (o_{i_1}, o_{i_2}).
  if $d_{min} > \alpha$
    break
  else
    Remove observations o_{i_1} and o_{i_2} from the set O, add $o_{i_1 i_2} = \{o_{i_1}, o_{i_2}\}$ to set O.
    Remove $d(o_{i_1}, o_j)$ and $d(o_{i_2}, o_j)$ from D, add $d(o_{i_1 i_2}, o_j) = mean(d(o_{i_1}, o_j), d(o_{i_2}, o_j))$, $1 \leq j \leq N$.
    Remove indexes i_1 and i_2 from set I, add new index $i_1 i_2$ to I.
  endif
B: Remove The Outliers
 find μ, the size of the largest subset of set O.
 Remove subsets of O whose size is smaller than $\beta\mu$.
C: Fuse Information Within Subgroups
 Do for each subgroup
   Fuse subgroup observations using **POP**. Proportional to the maximum distance between subgroup members, choose a value for κ.
 end Do
D: Fuse Information Between Subgroup
 Using **POP**, fuse subgroup representatives. Take $\kappa = 1$

TABLE II

ALGORITHM FOR GROUPING OBSERVATIONS AND REMOVING FAULTY OBSERVATIONS

C. **POP** and the Motivation Example

Here, we consider the robot localization example (in Section I) and evaluate **POP** using the example. We use the same configuration (same sensor and error models, failure rates) except that we use a particle filter to track the robot. We use **POP** to fuse the observations. The result of simulation is shown in Figure 2(a). The estimate is considered to be the weighted average particle. This example shows that comparing to a standard method such as WLC which considers covariance-based average, **POP** provides a better and less erroneous estimate.
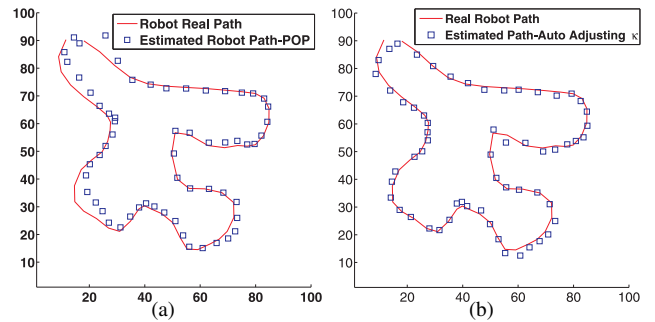

Fig. 2. The figures represent the results of the tracking example but this time we use **POP** and the particle filter. The left figure shows the result when we use a fixed κ ($\kappa = .1$). In the right figure κ is dynamically set using the auto-adjusting algorithm TableII. Compared to Fig.1(b), a noticable reduction in estimation error can be seen.

D. Computational Cost

Here, the computational cost of calculating the weights is not considered because it depends on the way we assign weight to observations. Moreover, the weight allocation cost is the same for all three methods. Among the three methods, LOP is computationally the cheapest and **POP** is the most expensive one. However, the difference is not considerable, especially if the number of states is large enough. If N and M represent the number of observations and the size of PV respectively, computational cost of LOP is $O(M(2N-1))$. Computational cost LGP and **POP** is $O(M(2N+1)-1)$ and $O(2M(N+1))$ respectively. It is clear when $N \gg 1$, the three methods impose the same cost. When N is small, there is a difference, although it is negligible.

E. Mechanism For determining κ

An important issue is how to choose an appropriate value for κ. As we explained earlier, different κ generates different results. Therefore, we need a mechanism to set κ automatically. The algorithm in TableII can be used to set κ.

V. Experiment

In order to evaluate the **POP** fusion method, we run one real world experiments. In this experiment, we try to estimate position of a mobile robot which is manually driven through our lab. There are 10 ceiling-mounted fixed camera, each of which observes part of the lab. Parts of the lab are covered by more than one camera. On the other hand, there are areas which are not covered by any of the cameras. The cameras are networked. Each camera in the network has a frame rate of 30 fps, and a resolution of 640×480 pixels. A background subtraction algorithm is used to detect the robot positions. A 2-D Gaussian model is considered for the cameras. More details on the experimental setup are provided in [16]. The robot is equipped with a Sick laser scanner and odometry sensor. A particle filter is used to combine the data. The filter estimate is considered as the ground truth for localization of the robot. We use **POP** in combination with a particle filter with 500 particles to track the robot. **POP** with a fixed κ ($\kappa = 0.1$) is used to fuse the information when more than one camera observes the target. The weighted average particle is chosen as the final estimation. To compare the results, two methods are chosen as bench marks: WLC in combination with a standard Kalman filter and a Bayesian method in combination with a particle filter. First, we fuse the observations if there are more than one camera observing the robot. Then a filter is used to determine the final estimate. We randomly biased 30% of the cameras detecting the robot (the number of the cameras should be more than 2). In Figs. 3(a) and 3(d), the results of tracking using Kalman filter and WLC are drawn (To have a better visibility, the robot path is depicted in two different figures). The solid (green) line represents the robot real path (generated by laser scanner) and markers represent the estimated path. Different markers are used to represent the number of cameras observing the robots for each time instant. For a similar senerio, the result of standard Bayesian in combination with a particle filter are

shown in Figs.3(b) and 3(e). **POP** and particle filter results are drawn in Figs. 3(c) and 3(f). Error histogram of three different methods are compared in Fig.4(a). As we can see from the figures, **POP** and Particle Filter results are much better than the other two. On the average, combination of **POP** and particle filter compared to WLC and Kalman filter reduces the estimation error by more than 20% and compared to Bayesian method and particle filter reduces the estimation error by more than 35%. As we mentioned above, a fixed κ is used to fuse the information using **POP**. In IV-E, we explain an algorithm for automatically adjusting κ. Now, we evaluate the **POP** algorithm using the auto adjusting algorithm. Fig. 4(b) compares the error histogram auto adjusting and fixed κ. From the figure, we can conclude that **POP** with auto adjusting κ generates better results than **POP** with a fixed κ. Figure 4(c) represents variation of within-group κ during the process.

VI. Conclusion and Future Work

In this paper we defined a series of specifications for evaluating a fusion method in MRMS environments. Two popular methods of sensor fusion, LOP and LGP, are overviewed and evaluated based on the specifications. A new method to overcome the difficulties of LOP and LGP is introduced. This new method complies with the defined specifications. The computational cost of **POP** is compared to the other two. Also an algorithm is provided to set the **POP** parameters. Results from the simulation and real world experiments shows that, comparing to WLC and Bayesian method, **POP** reduces the error in the presence of sensor failures. Based on the simulations and experimental results, we can conclude the **POP** is a reliable method and a good substitute for LOP and LGP.

For future works we consider to use the method in a more complex environment. For the environment with large amount of states, we will investigate solutions based on **POP** with less computational burden.

References

[1] Y. Bar-Shalom, T. Fortmann, "Tracking and Data Association.", Academic Press, 1988.

[2] S. Kullback, "Information Theory and Statistics", Dover Books, New York, 2nd ed., 1968. first edition New York:Wiley, 1959.

[3] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their population distributions", Bulletin Calcutta Mathematical Society, 35, 99-109, 1943.

[4] C. E. Shannon, "A Mathematical Theory of Communication", The Bell System Technical Journal, 27, 379-423 and 623-656, 1948

[5] A. Adjoudani and C. Benoit, "On the integration of auditory and visual parameters in a HMM-based asr", In David Stork and Marcus Hennecke, editors, NATO ASI: Speechreading by Humans and Machines. Springer- Verlag, 1996.

[6] A.C.S. Chung and H.C. Shen, "Entropy-Based Markov Chains for Multisensor Fusion", Journal of Intelligent and Robotic Systems, 29, October 2000, pp.161-189(29),

[7] H. Wang, G. Pottie, K. Yao and D. Estrin, "Entropy-based sensor selection heuristic for target localization", Information Processing and Sensor Networks 2004, Berkeley, California, April 2004, pp.36-45.

[8] Y. Zhou, H. Leung, "Minimum entropy approach for multisensor data fusion", in: IEEE Signal Processing Workshop on High-Order Statistics (SPW-HOS 97), 1997, pp. 336-339.
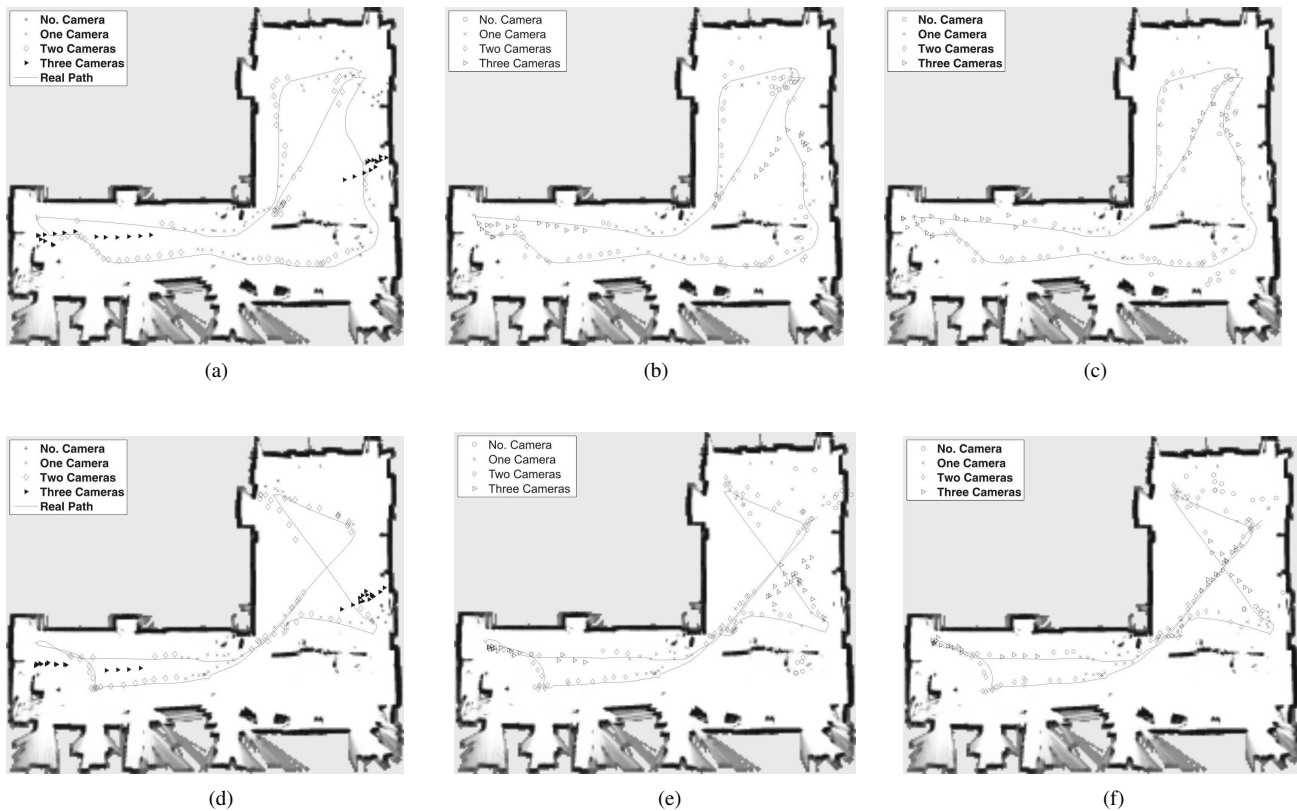
Fig. 3. The figures represents the results of the real experiment. In total, three different methods are examined: WLC for fusion and Kalman filter for the tracking (the left column), Bayesian method is used for the fusion and particle filter for the tracking (the middle column) and finally **POP** for the fusion and particle filter for the tracking (the right column). To make the results more visible, the result of each method is drawn into two different figures (top and bottom figures).
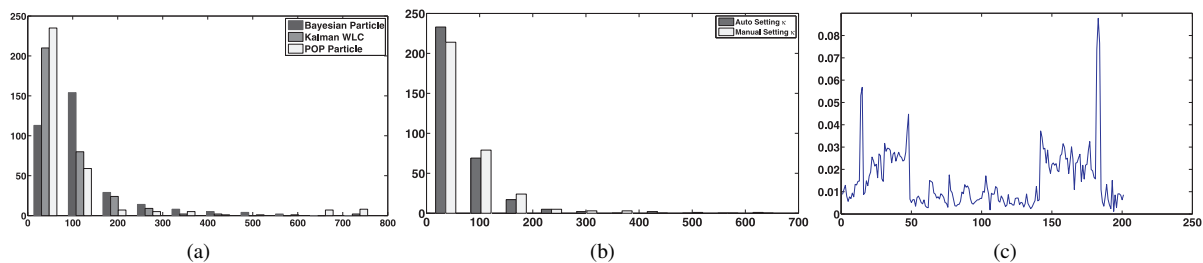

Fig. 4. The left figures compares the error histogram of the 3 different methods. The middle figure compares the error histogram **POP** with manual and auto adjustment. The right figure shows the variation of κ when the algorithm is used for automatically adjusting the κ.

[9] B. Fassinut-Mombot, and J. Choquel, "A New Probabilistic and Entropy Fusion Approach for Management of Information Sources", Information Fusion, 5, 2004, pp.35-47

[10] C.C. Sun, G.S. Arr, R.P. Ramachandran,S.G. Ritchie, "Vehicle Reidentification Using Multidetector Fusion", IEEE Transactions on Intelligent Transportation Systems, Volume 5, Issue 3, Sept. 2004 Page(s):155-164.

[11] F. K. Soong and A. E. Rosenberg, "On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition", IEEE Trans. Acoustics, Speech and Signal Proc., ASSP-36:871-879, 1988.

[12] K. F. Wallis, "Combining Density and Interval Forecasts: A Modest Proposal", Oxford Bulletin of Economics and Statistics67, 983-994, December 2005.

[13] H. F. Durrant-Whyte, "Sensor Models and Multisensor Integration", Int. J. Robot. Res., vol. 7, no. 6, pp. 97-113, 1988.

[14] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation", Computer, Vol. 22. No. 6., pp. 46-57, 1999.

[15] A. Pahliani and P. Lima, "Improving Self Localization and Object Localization by a Team of Robots Using Sensor Fusion", Proc. of CONTROLO 2006, Lisbon, Portugal, September 2006.

[16] M. Barbosa, A. Bernardino, D. Figueira, J. Gaspar, N. Gonçalves, P. Lima, P. Moreno, A. Pahliani, J. Santos-Victor, M. Spaan, J. Sequeira, "IS-Robot-Net: A Testbed for Sensor and Robot Network Systems", In Proc. of IROS 2009.

[17] M. B. Eisen, P T Spellman, P. O. Brown and D Botstein,"Cluster analysis and display of genome-wide expression patterns", Proceedings of the National Academy of Science of the United States of America, 1995, Pages 14863-14868.