Computing with Uncertainty: Intervals versus Probabilities

Mark J. L. Orr *
Advanced Robotics Research Ltd. and SD-Scicon UK Ltd.

Robert B. Fisher and John Hallam
Dept. of Artificial Intelligence, Edinburgh University

Abstract

We compare two well known methods of computing with uncertain quantities as used for geometric reasoning in robotics and computer vision. One method is based on intervals, the other on normal probability distributions. We find the latter method is better, in terms of both speed and accuracy.

1 Introduction

One of the key problems in sensor fusion and computer vision is finding out where things are. This is usually done by relating sensed geometric features to to features describing what is known *a priori* (a map of the environment or some object models). The problem is often referred to as *geometric reasoning*.

In practice geometric reasoning involves dealing with quantities whose values are uncertain because they rely on imperfect sensor measurements. Indeed, the two major approaches to geometric reasoning can be distinguished by the way in which they represent and manipulate errors. One method, originating with Brooks [4] and later refined by Fisher and Orr [7], represents uncertain numerical quantities as error intervals. The computations performed in this method are based on interval arithmetic [1] and, in Fisher and Orr's version, take place in a parallel network. In the other method, which seems to be more popular [8, 5, 3, 2, 6], uncertainty parameter vectors are represented by normal joint probability distributions and the main computation tool is the Kalman filter.

A typical example is the matched directions problem: a pair of 3-D direction vectors measured imprecisely in one reference frame coorespond to another pair

*Currently a visiting worker at the Department of Artificial Intelligence, Edinburgh University.

in some other reference frame; what is the rotation between the two frames? In this paper we compare the above two methods for speed and accuracy using this example.

2 Computing with Intervals

Solving a particular geometric problem using intervals starts with the compilation of constraints on error bounds from the equations that would describe the problem if everything was known precisely (see [7] for details). For example suppose x is related to a and b by $x = a - b$. In that case, it follows that the upper bounds (suprema) and lower bounds (infima) of the corresponding intervals are related by

$$\sup X \leq \sup A - \inf B,$$

$$\inf X \geq \inf A - \sup B.$$

For more substantial problems, such as the problem of matching two pairs of directions, the constraints are more complicated but can be compiled automatically by an off-line process. Whenever the on-line program is posed with a particular problem, it instantiates values for the known parameters in the appropriate set of compiled constraints and then uses them to tighten the bounds on the unknowns as much as possible. An iterative process gives tighter bounds because of cross-dependencies and recursion amongst the constraints.

In the experiments described below we used Fisher and Orr's network implementation [7]. Using a quaternion representation for rotations, their basic equations for the matched directions problem were

$$\mathbf{v}_i * \mathbf{q} = \mathbf{q} * \mathbf{u}_i, \quad i = 1, 2$$

where $\mathbf{u}_1$ and $\mathbf{u}_2$ are the model (unrotated) directions, $\mathbf{v}_1$ and $\mathbf{v}_2$ are the data (rotated) directions, $\mathbf{q}$ is the unknown quaternion and $*$ is the (non-commutative) quaternion multiplication operator.

3 Computing with Probabilities

In this method the analysis is centred on a measurement equation or sensor model involving an observation vector, $\mathbf{x}$, and a state vector $\mathbf{a}$ as arguments to a function $\mathbf{f}$ satisfying $\mathbf{f}(\mathbf{x}, \mathbf{a}) = \mathbf{0}$. The uncertainty associated with states and observations is represented by variance-covariance matricies and state estimates can be generated from observations using an estimation tool such as a Kalman filter. For most practical problems, including the matched directions problem, the function $\mathbf{f}$ is non-linear, and extensions to the basic linear tools are required. For our experiments we used an Iterated Extended Kalman filter [3, 2].

In the matched directions problem there are two observations, each consisting of the concatenated components of one of the pairs of matched directions: $\mathbf{x}_i = [\mathbf{u}_i^t\ \mathbf{v}_i^t]^t$, $i = 1, 2$. For the state, $\mathbf{a}$, we used the exponential rotation parameterisation, $\mathbf{r}$ [6] constructed by multiplying the rotation axis by the rotation angle. The sensor model is thus

$$\mathbf{f}(\mathbf{x}_i, \mathbf{a}) = \mathbf{f}([\mathbf{u}_i^t \mathbf{v}_i^t]^t, \mathbf{r}) = \mathbf{v}_i - e^{\mathbf{H}}\mathbf{u}_i = 0,\ i = 1, 2$$

where $\mathbf{H}$ is the anti-symmetric matrix

$$\mathbf{H} = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}.$$

4 Experiments

The basic experiment we did was to try both methods on many random examples of the matched directions problem. Each random example was generated by the following steps: 1) choose a random rotation and express it in the form of a matrix $\mathbf{R}$, 2) choose random directions $\mathbf{u}_1$ and $\mathbf{u}_2$, 3) compute rotated directions $\mathbf{v}_1 = \mathbf{R}\mathbf{u}_1$ and $\mathbf{v}_2 = \mathbf{R}\mathbf{u}_2$, 4) corrupt all four directions with noise of magnitude ϵ.

Each direction was corrupted by first adding to it a vector of independent zero-mean Gaussian noises of variance ϵ^2 and then renormalising the result to be of unit length. In the case of the interval based method, the intervals used were centred on the corrupted vector components and of half-width $\sqrt{3}\epsilon$ (ensuring a variance of ϵ^2). The output from the interval method was arranged to be in the form of a vector $\mathbf{r}^{(I)}$ of mid-range values for the components of the exponential form of the solution (converted from the quaternion) along with a vector $\mathbf{s}^{(I)}$ of interval half-widths (after dividing by $\sqrt{3}$). The output of the probability method was a vector $\mathbf{r}^{(P)}$ of mean values and a vector $\mathbf{s}^{(P)}$ of standard deviations from taking square roots of the variances in the uncertainty matrix.

To compare each method's estimate of its own accuracy we directly compared the uncertainties $\mathbf{s}^{(I)}$ and $\mathbf{s}^{(P)}$. We also computed rotation matricies $\mathbf{R}^{(I)}$ and $\mathbf{R}^{(P)}$ from the estimates $\mathbf{r}^{(I)}$ and $\mathbf{r}^{(P)}$ and with them the costs

$$c^{(I)} = \|\mathbf{v}_1 - \mathbf{R}^{(I)}\mathbf{u}_1\| + \|\mathbf{v}_2 - \mathbf{R}^{(I)}\mathbf{u}_2\|,$$

$$c^{(P)} = \|\mathbf{v}_1 - \mathbf{R}^{(P)}\mathbf{u}_1\| + \|\mathbf{v}_2 - \mathbf{R}^{(P)}\mathbf{u}_2\|.$$

5 Results

We found that, on average, the components of $\mathbf{s}^{(I)}$, the vector of interval widths, were about 100 times larger than ϵ, whereas the components of $\mathbf{s}^{(P)}$, the vector of standard deviations, were only 2 to 3 times larger. Thus the probabilistic method produces a much more constrained solution. This is the most striking difference between the two methods and is probably explained by the extra information carried by covariance terms in the uncertainty matrix.

The size of the ratio $c^{(I)}/c^{(P)}$ varied between 1 and about 100 but was usually small (less than 2 in the majority of cases). Thus we found that $\mathbf{r}^{(P)}$ was always a better solution than $\mathbf{r}^{(I)}$ but often only marginally.

The interval method is much slower to compute. The comparison depends on two thresholds: the minimum change of the error bounds which terminates the network iterations, and the minimum change in the state vector which terminates the iterations of the Iterated Extended Kalman filter. However the dependency is not great for reasonable choices of these thresholds. We used 0.001 for each threshold and in this case the interval method was about 50 times slower than the probabilistic method.

In conclusion we find that the method based on probabilities 1) produces significantly better uncertainty estimates, and 2) produces better mean estimates, 3) is significantly faster.

References

[1] G. Alefeld and J. Herzberger. *Introduction to Interval Computations.* Academic Press, UK, 1983.

[2] N. Ayache and O.D. Faugeras. Maintaining representations of the environment of a mobile robot. In *Robotics Research 4*, pages 337–350. MIT Press, 1988.

[3] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association.* Academic Press, UK, 1988.

[4] R.A. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence Journal*, 17:285–348, 1981.

[5] H.F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal of Robotics and Automation*, 4:23–31, 1988.

[6] O.D. Faugeras. A few steps towards artificial 3-d vision. In M. Brady, editor, *Robotics Science.* MIT Press, 1989.

[7] R.B. Fisher and M.J.L. Orr. Geometric reasoning in a parallel network. *International Journal of Robotics Research*, 10(2):103–122, 1991.

[8] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.