A Bayesian framework for geometric uncertainties handling

Kamel Mekhnacha, Pierre Bessiere, Emmanuel Mazer

A Bayesian framework for geometric uncertainties handling

K. Mekhnacha*, P. Bessière* and E. Mazer†

*Laboratoire LEIBNIZ, 46, avenue Félix Viallet, 38031 Grenoble, France
†Laboratoire GRAVIR, INRIA Rhône-Alpes, ZIRST 38030 Montbonnot, France

Abstract. We present a Bayesian CAD modeler for robotic applications. We describe the methodology we use to represent and handle uncertainties using probability distributions on the system parameters and sensor measurements. We address the problem of the propagation of geometric uncertainties and how to take this propagation into account when solving inverse problems. The proposed approach may be seen as a generalization of constraint-based approaches where we express a constraint as a probability distribution instead of a simple equality or inequality. We also describe appropriate numerical algorithms used to apply this methodology. Using an example, we show how to apply our approach by providing simulation results using the implemented CAD modeler.

INTRODUCTION

The use of geometric models in robotics and CAD systems necessarily requires a more or less realistic modeling of the environment. However, the validity of calculations with these models depends on the capacity of these systems to represent and take into account possible differences between the models and reality when solving a given problem.

Bayesian formalism [1, 2, 3] has been applied in diverse research fields. Numerous applications have been developed in physics, in artificial intelligence [4], as well as in mobile robotics [5, 6] and computer vision [7], and especially in parameter identification problems [8].

This paper presents a new methodology based on Bayesian formalism to represent and handle geometric uncertainties in robotics and CAD systems. For a given problem, the marginal distribution of the unknown parameters is inferred using the probability calculus. The original geometric problem is reduced to an optimization problem over the marginal distribution to find a solution with maximum probability. In the general case, this marginal probability may contain an integral on a large dimension space. The resolution method used to solve this integration/optimization problem is based on an adaptive genetic algorithm. The problem of integral estimation is approached using a stochastic Monte Carlo method. The accuracy of this estimation is controlled by the optimization process to reduce computation time.

A large category of robotic applications are instances of inverse geometric problems in presence of uncertainties, for which our method is well suited. The proposed approach has been applied to numerous robotic applications [9, 10] such as kinematics inversion for possibly redundant systems, robot and sensor calibration, parts' pose and shape calibration using sensor measurements, as well as in robotic workcell design. Experimental

results have demonstrated the effectiveness and the robustness of our approach. An example of this experimentation is presented in this paper.

This paper is organized as follows. Section 2 reports related work. In Sect. 3 we present our specification methodology and how to obtain an optimization problem from an original geometric problem. In Sect. 4 we describe our numerical resolution method. We present an example to illustrate our approach in Sect. 5 and give some conclusions in Sect. 6.

RELATED WORK

The representation and handling of geometric uncertainties is a central issue in the fields of robotics and mechanical assembly. Since the work of Taylor [11], in which geometric uncertainties were taken into account in the manipulator planning process for the first time, numerous approaches have been proposed to model these uncertainties explicitly.

Methods modeling the environment using "certainty grids" [12] and those using uncertain models of motion [13] have been extensively used, especially in mobile robotics.

Partially Observed Markov Decision Processes (POMDPs) [14] are a convenient representation for reasoning and planning in mobile robot applications, but they are inappropriate for more complex tasks such as manipulator planning.

Gaussian models to represent geometric uncertainties and to approximate their propagation have been proposed in manipulators programming [15] as well as in assembly [16]. Kalman filtering is a Bayesian recurrent implementation of these models. This technique has been used widely in robotics and vision [17], and particularly in data fusion [18]. Gaussian model-based methods have the advantage of economy in the computation they require. However, they are only applicable when a linearization of the model is possible and are unable to take into account inequality constraints.

Geometric constraint-based approaches [11, 19] using constraint solvers have been used in robotic task-level programming systems. Most of these methods do not represent uncertainties explicitly. They handle uncertainties using a least-squares criterion when the solved constraints systems are over-determined. In the cases where uncertainties are explicitly taken into account (as is the case in Taylor's system), they are described solely as inequality constraints on possible variations.

PROBABILISTIC GEOMETRIC CONSTRAINTS SPECIFICATION

In this section, we describe our methodology by giving some concepts and definitions necessary for probabilistic geometric constraints specification. We further show how to obtain an objective function to maximize from the original geometric problem.

Probabilistic kinematic graph

A geometric problem is described as a "probabilistic kinematic graph," which we define as the directed graph having a set of n frames (a frame is attached to each part and geometric entity) $S = \{S_1, \cdots, S_n\}$ as vertices and a set of m edges $A = \{A_{i_1 j_1}, \cdots, A_{i_m j_m}\}$, where $A_{i_k j_k}$ denotes an edge between the parent vertex S_{i_k} and its child S_{j_k} and represents a probabilistic constraint on the corresponding relative pose. We call these edges "probabilistic kinematic links". A given edge may describe:

- a modeling constraint (a piece of knowledge) on the relative pose between the parent frame and its child,
- a sensor measurement of the pose of a given entity,
- or a constraint we wish to satisfy to solve the problem (an objective value with a given precision, for example).

Each edge $A_{i_k j_k}$ is labeled by:

1. a probability distribution $p(Q_{i_k j_k})$ where $Q_{i_k j_k}$ is the relative pose vector (6-vector) $Q_{i_k j_k} = (t_x t_y t_z r_x r_y r_z)^T$. The first three parameters of this 6-vector represent the translation, while the remaining three represent the rotation.
2. possible equality/inequality constraints ($E_k(Q_{i_k j_k}) = 0$, $C_k(Q_{i_k j_k}) \leq 0$). These constraints represent possible geometric relationships between the two geometric entities attached to these two frames. Their shapes depend on the type of the geometric relationship. We implement several relationships between geometric entities in this work, such as points, polygonal faces, edges, spheres and cylinders. The details on equality/inequality constraints induced by these relationships can be found in [9].
3. a "status" 6-vector describing for each parameter of $Q_{i_k j_k}$, its role (nature) in the problem. A status can take one of the three following values:
   - *Unknown* (denoted by X) for parameters representing the unknown variables of the problem and whose values must be found to solve the problem.
   - *Free* (denoted by L) for parameters whose values are only known with a probability distribution.
   - *Fixed* (denoted by F) for parameters having known fixed values that cannot be changed.

In the general case, the kinematic graph may contain a set of cycles. The presence of a cycle represents the existence of more than one path between two vertices (frames) of the graph. To ensure the geometric coherence of the model, the computation of the relative pose between these two frames using all paths must give the same value. For each cycle containing k edges, we must have:

$$T_{ii} = T_{i\ i+1}^{s_i\ i+1} * T_{i+1\ i+2}^{s_{i+1}\ i+2} * \cdots * T_{k-1\ k}^{s_{k-1}\ k} * T_{k\ 1}^{s_k\ 1} * T_{1\ 2}^{s_1\ 2} * \cdots * T_{i-1\ i}^{s_{i-1}\ i} = I_4, \tag{1}$$

where T_{ij} is the 4×4 homogeneous matrix corresponding to the pose vector Q_{ij}, I_4 is the 4×4 identity matrix and $s_{ij} \in \{-1, 1\}$ is the direction in which the edge A_{ij} has been used.

We call these additional equality constraints the "cycle-closing constraints". They are global constraints involving, for each cycle, all the parameters it contains. The minimal number of cycles allowing coverage of a connected graph having n vertices and m edges is $p = m - n + 1$ [20]. Consequently, we obtain p cycle-closing constraints for a given problem.

Objective function construction

Given a probabilistic kinematic graph, we are interested in constructing a marginal distribution over the unknown parameters of the problem. Maximizing this distribution will give a solution to the problem.

To do so, we define the following sets of propositions:

- A set of p propositions $\{\mathcal{K}_i\}_{i=1}^{p}$ such as:
  $\mathcal{K}_i \equiv$ "cycle c_i is closed".
- A set of m propositions $\{\mathcal{H}_k\}_{k=1}^{m}$ such as:
  $\mathcal{H}_k \equiv$ "$C_k(Q_{i_k j_k}) \leq 0$ and $E_k(Q_{i_k j_k}) = 0$".

If we denote the unknown parameters of the problem by X, a solution to a problem is a value of X that maximizes the distribution

$$p(X|\mathcal{H}_1 \cdots \mathcal{H}_m \mathcal{K}_1 \cdots \mathcal{K}_p).$$

For each edge A_{ij}, if we denote by L_{ij} the set of parameters having the L status, and by X_{ij} the parameters having the X status, we can write, using the probability calculus and the p cycle-closing constraints (Eq. 1), the following general form:

$$p(X|\mathcal{H}_1 \cdots \mathcal{H}_m \mathcal{K}_1 \cdots \mathcal{K}_p) \propto p(X)I(X),$$

where

$$\begin{aligned} I(X) \;=\; & \int dL \\ & p(L_{i_1 j_1}) p(\mathcal{H}_1 | X_{i_1 j_1} L_{i_1 j_1}) \\ & \vdots \\ & p(L_{i_{m-p} j_{m-p}}) p(\mathcal{H}_{m-p} | X_{i_{m-p} j_{m-p}} L_{i_{m-p} j_{m-p}}) \\ & p_{O_1}(F_1(X,L)) p(\mathcal{H}_{m-p+1} | F_1(X,L)) \\ & \vdots \\ & p_{O_p}(F_p(X,L)) p(\mathcal{H}_m | F_p(X,L)). \end{aligned} \tag{2}$$

For each cycle c_i, where $i = 1, \cdots, p$, O_i denotes a pose vector pertaining to c_i and F_i is the function allowing computation of the value of this pose vector using the values of all other pose vectors pertaining to c_i (using Eq. 1). p_{O_i} denotes the distribution over O_i, while L is the concatenation of $L_{i_1 j_1}, \cdots, L_{i_{m-p} j_{m-p}}$.

RESOLUTION METHOD

We described in the previous section how to formulate an integration/optimization problem:

$$X^* = \max_X \left[p(X | \mathcal{H}_1 \cdots \mathcal{H}_m \mathcal{K}_1 \cdots \mathcal{K}_p) \right].$$

This section will present the practical numerical methods we use to solve the integration and the optimization problems.

Numerical integration method

Integral calculus is the basis of Bayesian inference. Unfortunately, analytic methods for integral evaluation seem very limited in real-world applications, where integrands may have complex shapes and integration spaces may have very high dimensionality.

Domain subdivision-based methods (such as trapezoidal or Simpson methods) are often used for numerical integration in low-dimensional spaces. However, these techniques are poorly adapted for high-dimensional cases.

Monte Carlo methods for numerical estimation

Monte Carlo methods (MC) are powerful stochastic simulation techniques that may be applied to solve optimization and numerical integration problems in large dimensional spaces. Since their introduction in the physics literature in the 1950s, Monte Carlo methods have been at the center of the recent Bayesian revolution in applied statistics and related fields, including econometrics [21] and biometrics. Their application in other fields such as image synthesis [22] and mobile robotics [23] is more recent.

Principles.
The principle of using Monte Carlo methods for numerical integration is to approximate the integral

$$I = \int p(x) g(x) \, d^d x,$$

by estimating the expectation of the function $g(x)$ under the distribution $p(x)$

$$I = \int p(x) g(x) \, d^d x = \langle g(x) \rangle.$$

Suppose we are able to get a set of samples $\{x^{(i)}\}_{i=1}^N$ (d-vectors) from the distribution $p(x)$, we can use these samples to get the estimator

$$\hat{I} = \frac{1}{N} \sum_i g(x^{(i)}).$$

Clearly, if the vectors $\{x^{(i)}\}_{i=1}^{N}$ are generated from $p(x)$, the variance of the estimator $\hat{I} = \frac{1}{N}\sum_i g(x^{(i)})$ will decrease as $\frac{\sigma^2}{N}$ where σ^2 is the variance of g:

$$\sigma^2 = \int p(x)(g(x) - \hat{g})^2 \, d^d x,$$

and $\hat{g}$ is the expectation of g.

This result is one of the important properties of Monte Carlo methods:
"The accuracy of Monte Carlo estimates is independent of the dimensionality of the integration space".

Using MC methods for our application

Using an MC method to estimate the integral (2) requires the following steps.

1. Sample a set of N points $\{L^{(i)}\}_{i=1}^{N}$ from the prior distribution $p(L)$ such that the sampled points respect local equality/inequality constraints (i.e. $\{\mathcal{H}_i\}_{i=1}^{m-p}$ have the value *true*).
2. Estimate the integral $I(X)$ using the set $\{L^{(i)}\}_{i=1}^{N}$ of points as follows.

$$\hat{I}(X) = \frac{1}{N}\sum_{i=1}^{N} \begin{array}{l} p_{O_1}(F_1(X, L^{(i)}))p(\mathcal{H}_{m-p+1}|F_1(X, L^{(i)})) \\ \vdots \\ p_{O_p}(F_p(X, L^{(i)}))p(\mathcal{H}_m|F_p(X, L^{(i)})) \end{array}$$

Points sampling.
The set of N points used to estimate the integral may be sampled in various ways. Since parameters pertaining to different kinematic links are independent, we can decompose the "state vector" L to $m-p$ components $\{L_{i_k j_k}\}_{k=1}^{m-p}$ and apply a local sampling algorithm [21]. Updating the state vector L

$$L^{(t)} = (L_{i_1 j_1}^{(t)}, L_{i_2 j_2}^{(t)}, \cdots, L_{i_k j_k}^{(t)}, \cdots, L_{i_{m-p} j_{m-p}}^{(t)})$$

only requires updating one component $L_{i_k j_k}$

$$L^{(t+1)} = (L_{i_1 j_1}^{(t)}, L_{i_2 j_2}^{(t)}, \cdots, L_{i_k j_k}^{(t+1)}, \cdots, L_{i_{m-p} j_{m-p}}^{(t)}).$$

N iterations of this procedure give us the set $\{L^{(i)}\}_{i=1}^{N}$, which will be used to estimate the integral.

To update a component $L_{i_k j_k}$ (a set of parameters pertaining to the same pose vector $Q_{i_k j_k}$), we must take into account possible dependencies between these parameters.

Consequently, we have to face the following two problems.

- **Candidate point sampling**
  A candidate $L_{i_k j_k}^{c}$ is drawn from the distribution $p(L_{i_k j_k})$. If we do not have a direct

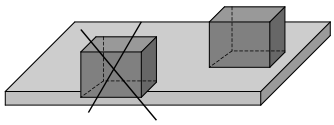


FIGURE 1. The candidate point is rejected because it does not respect the *Face-On-Face* constraint.

sampling method from this distribution at our disposal, an indirect sampling method must be used. In this work, we chose to use a Metropolis sampling algorithm [21].

- **Candidate validity checking**
  Suppose we have a geometric relationship between two geometric entities E_i and E_j. A geometrical calculus depending on the type of this relationship allows checking of the constraint $C_k(Q_{i_k j_k}) \leq 0$. If this constraint is respected (i.e. $p(\mathcal{H}_k|X_{i_k j_k} L_{i_k j_k}) = 1$), the candidate $L^c_{i_k j_k}$ is accepted, otherwise it is rejected. Figure 1 shows a *Face-On-Face* relationship example.

Optimization method

For our application, we chose a genetic-based algorithm. Since their introduction by Holland [24] in the 1970s, these stochastic techniques have been used for numerous global optimization problems, thanks to their ease of implementation and their independence of application fields [25]. However, practical problems related to the nature of our objective function have to be faced. We propose two major improvements to the genetic algorithm (abbreviated GA) we use. In the following, we will use $G(X)$ to denote the objective function $p(X|\mathcal{H}_1 \cdots \mathcal{H}_m \mathcal{K}_1 \cdots \mathcal{K}_p)$.

Narrowness of the objective function - constraints relaxation

In our applications, the objective function $G(X)$ may have a narrow support (the region where the value is not null) for very constrained problems. The initialization of the population with random individuals from the search space may give null values of the function $G(X)$ for most individuals. This will make the evolution of the algorithm very slow and its behavior will be similar to random exploration.

To deal with this problem, a concept inspired from classical simulated annealing algorithms consists of introducing a notion of "temperature". The principle is to first widen the support of the function by changing the original function to obtain non-null values even for configurations that are not permitted. To do so, we introduce an additional parameter we call T (for temperature) for the objective function $G(X)$. Our goal is to obtain another function $G^T(X)$ that is smoother and has wider support, with

$$\lim_{T \to 0} G^T(X) = G(X).$$

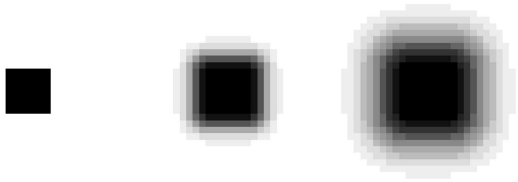


FIGURE 2. The distribution (in grey level) corresponding to inequality constraints induced by a *Point-On-Face* relationship for a square face at different values of temperature. The left figure shows the original constraints ($T = 0$), while the middle and the right ones show these constraints relaxed at ($T = 50$) and ($T = 100$) respectively.

To widen the support of $G(X)$, all elementary terms (distributions) of this later are widened, namely:

- distributions $p_{O_i}(F_i(X,L))$, where $i = 1, \cdots, p$.
- inequality constraints $p(\mathcal{H}_{m-p+j}|F_j(X,L))$, where $j = 1, \cdots, p$.

For example:

- for a Gaussian distribution:

$$\begin{aligned} f(x) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \\ f^T(x) &= \frac{1}{\sqrt{2\pi}\sigma(1+T)} e^{-\frac{1}{2}\frac{(x-\mu)^2}{[\sigma(1+T)]^2}} \end{aligned}$$

- for an inequality constraint over the interval $[a,b]$:

$$\begin{aligned} f(x) &= \begin{cases} 1 & \text{if } a \leq x \leq b \\ 0 & \text{else} \end{cases} \\ f^T(x) &= \begin{cases} 1 & \text{if } a \leq x \leq b \\ e^{-\frac{(x-a)^2}{(b-a)T}} & \text{if } x < a \\ e^{-\frac{(x-b)^2}{(b-a)T}} & \text{otherwise} \end{cases} \end{aligned}$$

In the general case, inequality constraints may be more complex. Figure 2 shows the case of a *Point-On-Face* inequality constraint for a square face.

Accuracy of the estimates - multiprecision computing

The second problem we must face is that only an approximation $\hat{G}(X)$ of $G(X)$ is available, of unknown accuracy. Using a large number of points to obtain sufficient accuracy may be very expensive in computation time, which makes the use of a large number of points in the whole optimization process inappropriate.

Since the accuracy of the estimate $\hat{G}(X)$ of the objective function depends on the number N of points used for the estimation, we introduce N as an additional parameter to define a new function $\hat{G}_N(X)$. Suppose we initialize and run for some cycles a genetic

algorithm with $\hat{G}_{N_1}(X)$ as evaluation function. The population of this GA is a good initialization for another GA having $\hat{G}_{N_2}(X)$ as evaluation function with $N_2 > N_1$.

General optimization algorithm

In the following, we label the evaluation function (the objective function) by the temperature T and the number N of points used for estimation. It will be denoted by $G_N^T(X)$. Our optimization algorithm may be described by the following three phases.

1. Initialization and initial temperature determination.
2. Reduction of temperature to recreate the original objective function.
3. Augmentation of the number of points to increase the accuracy of the estimates.

Initialization:. The population of the GA is initialized at random from the search space. To minimize computing time in this initialization phase, we use a small number N_0 of points to estimate integrals.

The choice of the initial temperature T_0 is crucial for the behavior of the algorithm. It depends strongly of the problem to solve: the more constrained the problem, the higher must be the initial temperature to allow a sufficient relaxation of constraints. We propose the following algorithm as an automatic initialization procedure for the initial temperature T_0, able to adapt to the complexity of the problem.

INITIALIZATION(GA)
BEGIN
    FOR each population[i] $\in$ GA's population DO
        REPEAT
            population[i] = random(S)
            value[i] = $G_{N_0}^T$(population[i])
            if (value[i] == 0.0)
              T = T + ΔT
        UNTIL (value[i]$>$ 0.0)
    FEND
    Re-evaluate(population)
END
where ΔT is a small increment value.

Temperature reduction:. To get the original objective function ($T = 0.0$), a possible scheduling procedure consists of multiplying the temperature, after running the GA for a given number of cycles nc_1, by a factor α ($0 < \alpha < 1$). In this work, the value of α has been experimentally fixed to 0.8. We can summarize the proposed algorithm as follows.

TEMP_REDUCTION(GA)
BEGIN
WHILE (T $>T_\varepsilon$) DO
FOR i=1 TO nc_1 DO
Run(GA)
FEND
T = T * α
WEND
T = 0.0
Re-evaluate(population)
END
where T_ε is a small threshold value.

Augmenting the number of points:. At the end of the temperature reduction phase, the population may contain several possible solutions for the problem. To decide between these solutions, we must increase the accuracy of the estimates. One approach is to multiply N, after running the GA for a given number of cycles nc_2, by a factor β ($\beta > 1$) so that the variance of the estimate is divided by β:

$$Var(G^0_{\beta * N}(X)) = \frac{1}{\beta} Var(G^0_N(X)).$$

We can describe this phase by the following algorithm.

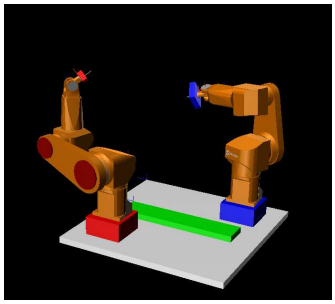
N_POINTS_AUGMENTATION(GA)
BEGIN
WHILE (N $<N_{max}$) DO
FOR i=1 TO nc_2 DO
Run(GA)
FEND
N = N * β
WEND
END
where N_{max} is the number of points that allows convergence of the estimates $\hat{G}^0_N(X)$ for all individuals of the population.

EXAMPLE

In this section, we describe how to use our CAD system for concrete problems. We present in detail a kinematics inversion problem under geometric uncertainties.

Problem description

Using two Stäubli Rx90 robot arms with six revolute joints, we are interested in placing two prismatic parts one against the other. The only constraint is that a face of the first part will be in a *Face-On-Face* relationship with a face of the second.



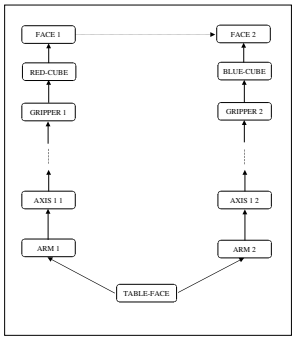


FIGURE 3. Kinematics inversion example using two Stäubli Rx90 arms and the corresponding kinematic graph.

The two arms are modeled as a set of parts attached to each other using probabilistic kinematic links. We assume that the more significant uncertainties are on zero positions. The two parts are also attached to arms' end effectors using probabilistic kinematic links. The added constraint we wish to satisfy to solve the problem is represented by a link between the two faces to place in *Face-On-Face* relationship. We use for in this link three Gaussians on the three constrained parameters t_z, r_x and r_y with zeros as mean values and 0.5mm, 0.01rad and 0.01rad respectively as standard deviations. Figure 3 shows the two arms and gives the corresponding kinematic graph.

We suppose in this example that zero positions uncertainties of the arm on the right of Fig. 3 ($Arm1$) are five times more important than the ones of the arm on the left ($Arm2$) (for each joint, we suppose a Gaussian distribution on the zero position with 0.01rad as standard deviation for $Arm1$ and with 0.05rad for $Arm2$). Our aim is to comment qualitatively on the solution obtained and to show the importance of taking uncertainties propagation into account when choosing a solution.

Results

Figure 4 shows the solution obtained by the system. This solution gives a maximal precision for the required *Face-On-Face* relationship because:

1. $Arm1$ (the less accurate) is coiled to minimize the propagation of the uncertainties on its zero positions.
2. Rotation axes are perpendicular to the common normal of the two faces.

Table 1 summarizes the problem complexity and the system performances for this problem using a PowerPC G3/400 machine.

Discussion

This example shows how the proposed method takes geometric uncertainties into account in a general and homogeneous way. No assumptions have been made, either

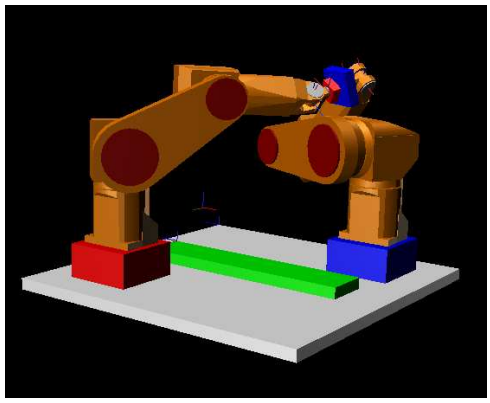


FIGURE 4. The solution obtained by the system.

TABLE 1. Some parameters summarizing the problem complexity and the system performances for this kinematics inversion problem.

Integration space dimension	50
Optimization space dimension	12
Number of cycles	1
Number of frames	28
Number of inequality constraints	16
Computation time (seconds)	13

on the uncertainties models (shapes of the used distributions), nor on the linearity of the model or the possibility of it being linearized. It also shows how possible redundancy of the system relating to the required task is used to find the most accurate solution.

CONCLUSION

We have presented a generic approach for geometric problems specification and resolution using a Bayesian framework. We have addressed the problem of the propagation of geometric uncertainties, and how to take this propagation into account when solving inverse problems. We have shown how a given geometric problem is first formulated as a probabilistic inference problem, and then reduced to an integration/optimization one. For generality, no assumptions have been made on the shapes of the distributions or on amplitudes of uncertainties.

Experimental results made on our system have demonstrated the effectiveness, the robustness and the homogeneity of representation of our approach. However, we are considering to improve the integration module by avoiding numerical integration when the integrand is a product of generalized normals (Dirac's delta functions and Gaussians) and when the model is linear or can be linearized (errors are small enough). Future work

will aim to extend our system so that it allows using high-level sensors such as vision-based ones. It will also aim to include non-geometrical parameters (inertial parameters for example) in problem specification.

REFERENCES

1. Jaynes, E. T., Probability Theory - The Logic of Science (1996), unfinished book publicly available at http://bayes.wustl.edu.
2. Mohammad-Djafari, A., and Demoment, G., *Maximum Entropy and Bayesian Methods*, Kluwer Academic Publishers, 1992.
3. Neal, R. M., Probabilistic inference using Markov Chain Monte Carlo methods, Research Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto (1993).
4. Jaakkola, T., and Jordan, M., *J. Artif. Intellig. Res. (JAIR)*, **10**, 291–322 (1999).
5. Thrun, S., *Machine Learning*, **33**, 41–76 (1998).
6. Bessière, P., Dedieu, E., Lebeltel, O., Mazer, E., and Mekhnacha, K., *Intellectica*, **26-27**, 257–311 (1998).
7. Weiss, Y., and Adelson, E. H., Slow and smooth: a Bayesian theory for the combination of local motion signals in human vision, Research Report AI Memo 1624, MIT (1998).
8. Presse, C., and Gautier, M., "Bayesian estimation of inertial parameters of robots", in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Nice, France, 1992, vol. 1, pp. 364–369.
9. Mekhnacha, K., *Méthodes probabilistes Bayesiennes pour la prise en compte des incertitudes géométriques: application à la CAO-robotique*, Thèse de doctorat, Inst. Nat. Polytechnique de Grenoble, Grenoble, France (1999).
10. Mekhnacha, K., Mazer, E., and Bessière, P., "A Bayesian CAD system for robotic applications", in *Proc. of the IASTED Int. Conf. on Modelling and Simulation*, Pittsburgh, PA, USA, 2000, pp. 527–534.
11. Taylor, R., *A synthesis of manipulator control programs from task-level specifications*, Ph.d thesis, Stanford University, Computer Science Department (1976).
12. Moravec, H. P., *AI Magazine*, **9**, 61–74 (1988).
13. Alami, R., and Simeon, T., "Planning robust motion strategies for mobile robots", in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, San Diego, California, 1994, vol. 2, pp. 1312–1318.
14. Kaelbling, L. P., Littman, M., and Cassandra, A., *Artificial Intelligence*, **101**, 99–134 (1998).
15. Puget, P., *Vérification-Correction de programme pour la prise en compte des incertitudes en programmation automatique des robots*, Thèse de doctorat, Inst. Nat. Polytechnique de Grenoble, Grenoble, France (1989).
16. Sanderson, A. C., "Assemblability based on maximum likelihood configuration of tolerances", in *Proc. of the IEEE Symposium on Assembly and Task Planning*, Marina del Rey, CA., 1997.
17. Zhang, Z., and Augeras, O., *3D Dynamic Scene Analysis: A Stereo Based Approach*, Springer, Berlin, Heidelberg, 1992.
18. Bar-Shalom, Y., and Fortmann, T. E., *Tracking and Data Association*, Academic Press, 1988.
19. Owen, J. C., *Int. J. of Computational Geometry and Applications*, **6**, 421–434 (1996).
20. Gondran, M., and Minoux, M., *Graphes et Algorithmes*, Eyrolle, Paris, 1990.
21. Geweke, J., "Monte Carlo simulation and numerical integration", in *Handbook of Computational Economics*, edited by H. Amman, D. Kendrick, and J. Rust, Elsevier North-Holland, Amsterdam, 1996, vol. 13, pp. 731–800.
22. Keller, A., "The fast calculation of form factors using low discrepancy point sequence", in *Proc. of the 12th Spring Conf. on Computer Graphics*, Bratislava, 1996, pp. 195–204.
23. Dellaert, F., Fox, D., Burgard, W., and Thrun, S., "Monte Carlo localization for mobile robots", in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Detroit, MI, 1999.
24. Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
25. Mazer, E., Ahuactzin, J., and Bessière, P., *J. Artif. Intellig. Res. (JAIR)*, **9**, 295–316 (1998).