T&T Proof 03TA2005 5 September 2003

10.1098/rsta.2003.1269

THE ROYAL SOCIETY

Modelling the world in real time: how robots engineer information

By Andrew J. Davison

Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK

Programming robots and other autonomous systems to interact with the world in real time is bringing into sharp focus general questions about representation, inference and understanding. These artificial agents use digital computation to interpret the data gleaned from sensors and produce decisions and actions to guide their future behaviour. In a physical system, however, finite computational resources unavoidably impose the need to approximate and make selective use of the information available to reach prompt deductions. Recent research has led to widespread adoption of the methodology of Bayesian inference, which provides the absolute framework to understand this process fully via *modelling* as informed, fully acknowledged approximation. The performance of modern systems has improved greatly on the heuristic methods of the early days of artificial intelligence. We discuss the general problem of real-time inference and computation, and draw on examples from specific recent research in computer vision and robotics: specifically visual tracking and simultaneous localization and mapping.

Keywords: robotics; computer vision; Bayesian inference; real-time systems

1. Introduction

The ubiquity of modern computer processors means that many artificial devices can now be charged with the task of reacting usefully and automatically to their surroundings. At the luxury end of the scale are robots as they are commonly perceived, no longer just executing sequences of car-spraying commands but aiming to move autonomously through the world and interact with it in human-like ways. However, many more mundane devices are achieving similar capabilities, including automated security cameras and lawnmowers, personal computer user interfaces, medical imaging devices, and automated diagnostics systems for cars and photocopiers. These devices receive information about the world via sensors, be they video cameras, microphones or various simpler encoders, and produce output either as direct action or an interpretation to be passed to a human operator. The task of their processing units is to *link sensing and action* via computation.

One contribution of 22 to a Triennial Issue 'Mathematics, physics and engineering'.

(*a*) *Real-time systems and sequential computation*

In this paper we are concerned with systems which operate in *real time*. In a given scenario, where a robotic or other artificial system is charged with performing a task in the dynamic world, a certain threshold of performance makes interaction possible: it can process sensor data fast enough to make decisions which usefully influence its immediate future actions. It is possible for a computer to analyse a sequence of video images of a ball in flight to calculate its trajectory, but this is of no use to a table tennis-playing robot unless the results of the calculation are available in time actually to play the next stroke.

In some applications of computer sensing technology it is perfectly acceptable to operate at less than real-time rates. Rather than connecting a sensor directly to a computer and processing data as they arrive, the sensor's output can be recorded and digitized into an array which the computer can access at its leisure. In computer vision, computation of this type has recently seen success in the area of cinema post-production (Pollefeys *et al*. 2000), where analysis of a stored film sequence permits recovery with great accuracy of the trajectory of the moving camera which took the footage, and this permits artificial graphical objects to be inserted convincingly into a scene ('augmented reality'). In such off-line implementation it is possible and indeed preferable to use algorithms which use all the data available at once in performing a global estimation. In the case of the flying table tennis ball, a smooth parametric curve (probably not an exact parabola because of the effects of spin and air resistance) could be fitted to the locations detected in all the images, providing better estimates for the ball location at each frame than the potentially noisy individual measurements obtained at each time-step.

Change OK?

In systems which must operate in real time (the table tennis robot, or augmented reality for live television), the computation must be *sequential*: the effect of each new arrival of data must be calculated promptly because the result is needed *now* and more data will be arriving soon. The typical situation is that these data are arriving at a constant rate—a video camera may produce a new image every $\frac{1}{30}$ s for instance. In this case the constraint on a real-time algorithm can be simply stated: it must operate in constant time, requiring an amount of processing bounded by a constant to take account of each new image. The value of this constant will depend on the details of the processor available and the sensing rate. Real-time applications often present severe limitations to the processing power available—the system may be embedded in a mobile or remote device. When constructing a real device, processing capability, measured in operations per unit time, is a resource, limited and expensive. But although processors are being constructed which are faster and smaller for the same price, year after year, the constant time requirement imposes fundamental restrictions on real-time algorithms. It rules out extensions of the optimization schemes favoured in off-line methods, in which a 'best fit' is repeatedly performed to all data received to date, since over time the archive of data will grow steadily, and whatever processor is available, a point in time will be reached where the optimization will take too long. One way to achieve constant time performance would be to save measurement data only within a moving window of time directly preceding the present, forgetting everything before, and then to use these data to calculate quantities of interest as needed. However, this is an unnecessarily arbitrary choice; it marks all data from before the start of the window as of zero importance.

Rather, we are led towards a time-independent *state*-based representation, in which everything of interest about the system in question is captured in a snapshot of the current instant. Rather than storing a history of measurements of the position of a ball, estimates of its current position, velocity, acceleration, perhaps spin and other higher-order terms provide the same ability to predict its future behaviour. Sequential estimation then proceeds at each time-step by updating the state estimate to include the effects of the passage of time and any new data obtained.

Constant time operation requires that this state estimate be expressed with an amount of information which is fixed over time—so that it can always be updated with a finite amount of processing. In the 1950s, Shannon showed that information is precisely quantified in bits exactly as they are familiar today from computer technology; each bit in a computer's memory is able to store the outcome of a yes/no question. For example, a floating point number requires storage bits proportional to its number of significant decimal digits (think of storing the results to a series of increasingly more precise yes/no questions about the number's value). There can be no doubt therefore that storage of what is learnt from a constantly growing archive of data using a finite number of bits will involve approximation—but what kind is appropriate? The key insight is that in order for an approximated estimate to be truly useful it must be accompanied by a measure of its veracity so that it can be weighed against new data. Rather than using all the information processing and storage available in a real-time budget simply to calculate the current state estimate with maximum precision, some effort and information should be devoted to evaluating and storing a measure of the *uncertainty* in this estimate. Sequential estimation then proceeds as a series of weighted combinations of old and new data.

Minsky (1985) said of the human visual system:

> We have the sense of actuality when every question asked of our visual systems is answered so swiftly that it seems as though those answers were already there.

Real-time processing is what the human mind is doing all the time. Experience gained from working with artificial real-time systems has led cognitive scientists to consider what has been termed the 'frame problem': the question of the context to a decision. The great complexity of processors such as human and animal brains which are able to interact with the world has become clear: their role is to simulate the world and 'produce future'. A mind contains a functioning virtual reality system (Dawkins 1998), a simulator where the effect of actions can be predicted. Any action or interpretation will depend not only on current sensor data but a large amount of prior knowledge about a device's structure, likely surroundings and short- and long-term goals. Dennett (1984) discussed a hypothetical robot, required to act quickly in order to replenish its diminishing power supply but surrounded by potential hazards; a rushed decision could lead to its downfall at the hands of some unforeseen danger, while over-contemplation could leave it paralysed by indecision. Inevitably, some eventualities and details must simply be ignored rather than thought through and accepted or rejected. It is exactly this question of acceptance and acknowledgement of the limits of one's knowledge which is of concern in real-time robotic systems.

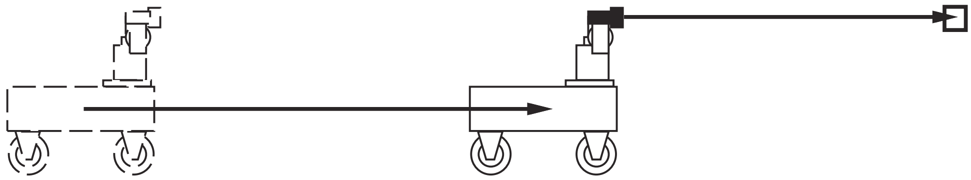


Figure 1. Uncertainty in a simple robot localization scenario.

(*b*) *Modelling and uncertainty*

In fact, managing uncertainty requires rather a large amount of processing resource. In this section we shall see how modelling in the context of Bayesian probability theory provides the absolute framework for propagation of the uncertainty inherent to all real data. The real world is complex, and sensors produce a very large amount of raw data—in the case of a camera, the intensity and colour of light arriving at each of perhaps several million sensing elements. Storing, copying or performing calculations with these data all require various computer operations. Rather than using all the information available to learn everything possible, it is essential to extract what is important via *modelling*: summarizing complex entities by simpler parametrizations. Modelling is necessarily approximation. But it is important to realize that it always involves a choice—what level of approximation is necessary? While a certain type of model may seem the obvious or only option, it is worth being clear that some degree of approximation is present unless the infinitesimal details of every atom are explicitly represented. For example, it is standard to parametrize the position of a rigid body of matter in terms of the coordinates of its centre of mass and the angles describing its angular orientation, but no macroscopic object is truly rigid and so in structural engineering it may be profitable to add parameters for the flexing of a load-bearing rod, or in astrophysics to parametrize the varying density of a planet. Via modelling, a view of the world should be reduced in complexity to the point where a reasonable decision is possible. Word added – OK?

A model's usefulness in making decisions depends not only on the accuracy of representation it provides, but also a measure of the uncertainty involved—as an approximation, a model represents a range of possible true states, and we must know how large this range is. Scientists are all taught to calculate error measures for the results of experiments, but their value becomes clear when decisions must be made on the basis of results. Consider an example from robot localization as depicted in figure 1: it is known that during a simple manoeuvre, if a certain voltage is applied to a robot's drive motor for 1 s, it will move forward 1.00 m on average, but many repetitions of this motion have revealed that depending on the exact current conditions (floor slipperiness, bumpiness, tyre condition, etc.) the actual distance covered is sometimes 0.95 m, 1.02 m, 0.97 m—overall, a Gaussian distribution with standard deviation 0.05 m. The robot also has a stereo camera rig that can measure the forward distance to a beacon but with standard deviation 0.02 m. From an initial position known to within 0.10 m, the robot moves forward for 1 s, stops and makes a visual measurement of its distance from the beacon. How should its new position be estimated? The calculation will involve combining its previous position estimate, motion model and sensor measurement model information. It is the explicit uncertainties attached to each of these data which allow us to know how much faith can be placed in each one—data with small uncertainty (in this case the camera measurement) will have the greatest effect on the result.

Early approaches to artificial intelligence and robotics used rule-based systems based on Aristotelian logic: if A and not B, then C. But this is a language of certainty, not well suited to real-world problems which unavoidably involve models and uncertainty. For this reason, recent research in robotics has turned towards Bayesian probabilistic methods. Bayes's rule itself is a simple formula, but 'Bayesian' has generally become known as a certain way of interpreting probability theory, a method of inference in which the probability associated with a certain outcome is interpreted as a subjective degree of the belief held in that hypothesis.

In Bayesian inference, even before any new data are incorporated, a prior probability must be assigned to every possible consequence within the domain specified by the model, and it is this aspect which makes some uncomfortable—particularly those used to probability and statistics as theories of the 'frequencies' of random events. From the Bayesian viewpoint, the probability distribution associated with a tumbling die (that the chance of landing on each of its faces is $\frac{1}{6}$) does not represent any intrinsic randomness, but a lack of knowledge about its precise state—after all, there are no random events in deterministic (classical) physics. Random effects or 'noise' are simply the result of unmodelled factors (the slipperiness of floor or inflation of tyre of our mobile robot example). Jaynes (2003) commented forcefully on the role of probabilities as subjective degrees of belief:

> ...apart from probability theory, no scientist ever has sure knowledge of what is really true; the only thing we can every know with certainty is what is our state of knowledge? ... the belief that probabilities are realities existing in Nature is pure Mind Projection Fallacy.

The prior probability distributions used may be simple, but the Bayesian approach is to set them up as honestly as possible and use them. It is perfectly possible to assign equal probabilities to all possible outcomes, indicating a complete lack of knowledge—but usually consideration reveals this not to be the case and suggests a better assignment. Quoting from Torr *et al.* (1999), who in turn cite Jaynes:

> Some will complain that to use Bayesian methods one must introduce arbitrary priors on the parameters. However, far from being a disadvantage, this is a tremendous advantage as it forces open acknowledgement of what assumptions were used in designing the algorithm, which all too often are hidden away beneath the veneer of equations.

Bayes's rule is rearrangement of the chain rule for probability relating discrete propositions $\boldsymbol{X}$ and $\boldsymbol{Z}$; the probability $P(\boldsymbol{X}, \boldsymbol{Z})$ of both $\boldsymbol{X}$ and $\boldsymbol{Z}$ occurring is $P(\boldsymbol{X}, \boldsymbol{Z}) = P(\boldsymbol{X} \mid \boldsymbol{Z})P(\boldsymbol{Z}) = P(\boldsymbol{Z} \mid \boldsymbol{X})P(\boldsymbol{X})$, which is rearranged to give Bayes's rule:

$$P(\boldsymbol{X} \mid \boldsymbol{Z}) = \frac{P(\boldsymbol{Z} \mid \boldsymbol{X})P(\boldsymbol{X})}{P(\boldsymbol{Z})}.$$

The notation $P(\boldsymbol{X} \mid \boldsymbol{Z})$ is read 'the probability of $\boldsymbol{X}$ given $\boldsymbol{Z}$' and describes the conditional probability of proposition $\boldsymbol{X}$ being true if $\boldsymbol{Z}$ <u>is</u> known to be true. Bayes's Change OK?
rule is always valid but interesting when propositions $\boldsymbol{X}$ and $\boldsymbol{Z}$ are correlated—learning something new about one reveals more about the other. Using the type of

example familiar from probability textbooks, imagine that in some game you and an opponent each roll a die in secret. Of interest is the sum of the two values rolled; say if the sum is 9 it may be worth betting—however, you only have knowledge of the value of your die, and this comes up with the value 5. We choose to define $\boldsymbol{X}$ as the proposition that the sum is 9; $\boldsymbol{Z}$ is the proposition that the die you can see comes up 5. Trivially, $P(\boldsymbol{Z}) = \frac{1}{6}$. $P(\boldsymbol{X})$, the *prior* probability of the sum 9 is evaluated by considering all 36 possible combinations of the two dice: $P(\boldsymbol{X}) = \frac{4}{36} = \frac{1}{9}$. $P(\boldsymbol{Z} \mid \boldsymbol{X})$ is the *likelihood* of your die coming up 5 given that the sum is 9—of the 4 out of 36 combinations summing to 9, only one includes a 5 on your die so $P(\boldsymbol{Z} \mid \boldsymbol{X}) = \frac{1}{4}$. Feeding through Bayes's rule:

$$P(\boldsymbol{X} \mid \boldsymbol{Z}) = \frac{\frac{1}{4} \times \frac{1}{9}}{\frac{1}{6}} = \frac{1}{6}.$$

To highlight here is the difference between the prior probability $P(\boldsymbol{X}) = \frac{1}{9}$ and *posterior* $P(\boldsymbol{X} \mid \boldsymbol{Z}) = \frac{1}{6}$: knowledge of the value of the one visible die has reduced uncertainty about the value of the sum (we are now 17% sure that the sum is 9 rather than 11% sure). Learning the value of the one die can be viewed as a measurement of a quantity which is correlated with the hidden state (the sum) and which therefore improves the state estimate. Clarify sentence?

Bayes's rule is valid for such discrete propositions, but also in the case of continuous probability density functions (PDFs):

$$p(\boldsymbol{X} \mid \boldsymbol{Z}) = \frac{p(\boldsymbol{Z} \mid \boldsymbol{X}) p(\boldsymbol{X})}{p(\boldsymbol{Z})}.$$

Here we consider $\boldsymbol{X}$ and $\boldsymbol{Z}$ to be vector variables which can take on a continuous range of values; $p(\boldsymbol{X})\,\mathrm{d}\boldsymbol{X}$ is the probability of $\boldsymbol{X}$ having a value in the range $\boldsymbol{X} \to \boldsymbol{X} + \mathrm{d}\boldsymbol{X}$. When Bayes's rule is applied to sequential estimation, it is interpreted in the following way: $p(\boldsymbol{X})$ is the prior, a PDF over the model parameters before a measurement is made; the goal is the calculate the new PDF $p(\boldsymbol{X} \mid \boldsymbol{Z})$ over these parameters given a measurement $\boldsymbol{Z}$. Bayes's rule is useful in practice because typically it is straightforward to determine the form of the likelihood function $p(\boldsymbol{Z} \mid \boldsymbol{X})$: the probability of making a certain measurement given a certain state. The form of this likelihood function is defined using a generative model of the measurement process, the result of consideration of the sensor type and the possible uncertainties involved.

In the rest of this paper, two closely related research areas in computer vision and robots in which real-time Bayesian inference has been employed to great effect will be discussed, the role of Bayesian techniques highlighted and recent results presented. In first discussing visual tracking we will look at the steady progress of probabilistic techniques and how they have improved greatly on early ad hoc methods. Secondly, we will introduce the domain of simultaneous localization and mapping for mobile robots, in which acknowledgement of uncertainty is the key to consistency in autonomous navigation.

2. Visual tracking

Tracking is the general question of continuously identifying the location of a known object over time, and is the problem posed by many domains including surveillance

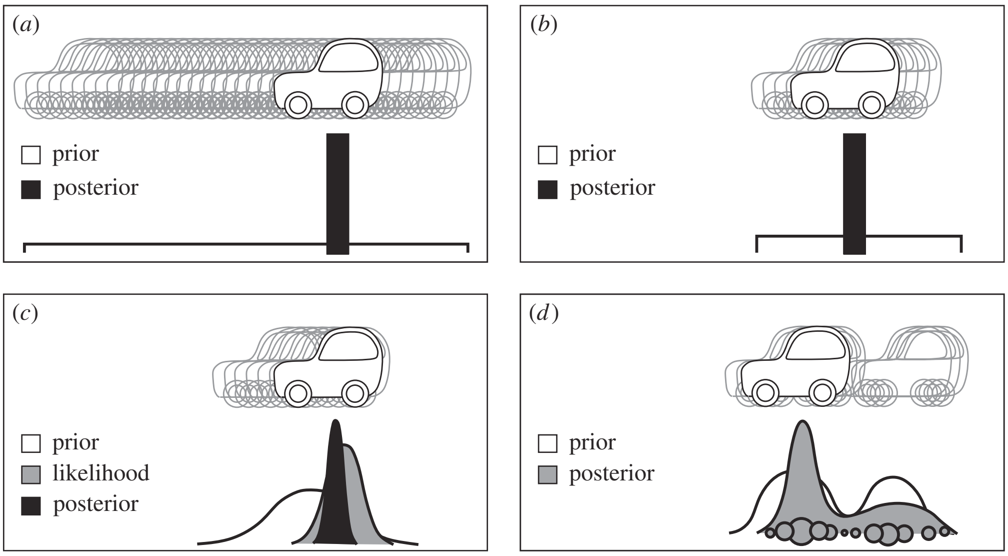


Figure 2. Increasingly effective tracking: (*a*) an exhaustive search of all possible object configurations is performed at each step; (*b*) the search region is restricted using a prediction based on earlier data; (*c*) first-order propagation of uncertainty in a Kalman filter framework permits Gaussian prior and likelihood information both to contribute to a posterior estimate; (*d*) in a particle filter the true shape of the PDF can be accurately represented, allowing multi-modal distributions and increased robustness.

or human–computer interaction (where knowledge of the positions of a user's head and hands for instance allows their intentions to be deduced without a keyboard). In this section we will look at some approaches taken to tracking using computer vision, though a similar progression has been taken in other fields using different sensing modalities. Tracking is the quintessential problem of real-time inference, and has therefore been studied in great depth in computer vision and other fields and provides good ground for exposition of the array of techniques attempted.

Posing the problem generally, a vector of model parameters $\boldsymbol{X}(t)$ specifying the location of an object is to be estimated sequentially based on a series of sensor measurements $\boldsymbol{Z}(t)$. Consider the simplified situation depicted in figure 2: a camera images a passing car, whose movement is known closely to follow a narrow road such that its position can reasonably be modelled using a single parameter. To 'measure' the car's position, the computer has some representation of its appearance, and knowledge of the mapping between the parametrized car position and the position at which it is imaged (which will be a function of the world position of the camera and the camera's internal characteristics, such as focal length).

In early model-based tracking using vision (e.g. Lowe 1992), the targets were simple objects which could closely be modelled with mathematically convenient geometrical shapes, and clear edges or other image features were available as reliable measurements. In cases like these, localizing the object at each time-step can proceed as either an exhaustive or gradient descent search in which a measure of fit of a hypothesized model configuration is repeatedly evaluated based on how well it predicts the measurements obtained, with the model degrees of freedom adjusted to find the global best fit (figure 2*a*). An unconstrained search of this type, generally initialized at each new time-step at the model configuration found in the previous frame, can get into trouble with local maxima in the search space, and will not be able to track rapid movement. It is profitable to constrain the search area using information which is available about the possible motion of the object (figure 2*b*)—given knowledge about where the object was at a sequence of earlier time-steps, it is possible to make a prediction, with associated uncertainty, about where it will be at the current time, and limit search to this part of configuration space.

When combining motion and measurement information in this way, however, we can do better than simply using motion information to initialize a search by putting both types of information into a Bayesian probabilistic framework. The extended Kalman filter (EKF) has been widely used in visual tracking to achieve this, Harris (1992) describing a classic implementation. The 'goodness-of-fit' function associated with measurements must now take the form of a Gaussian likelihood $p(\boldsymbol{Z} \mid \boldsymbol{X})$ which describes the probability of measurements $\boldsymbol{Z}$ given a state $\boldsymbol{X}$ (a generative measurement model), and the motion model of what is known in advance about the types of motion expected has the Gaussian form $p(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-\Delta t})$. Tracking now proceeds as the sequential propagation of a PDF in configuration space. The estimate of model configuration at any time-step is a weighted combination of both information from the most recent set of measurements and, via motion continuity, that from previous measurements (figure 2*c*). In more difficult tracking problems—where the models are, for Change OK? example, deformable two-dimensional templates tracking complicated objects with agile motion—EKF-based tracking was enhanced with the use of learnt motion models (Reynard *et al.* 1996): analysis of a training dataset enabled probabilistic models of motion to be built, giving much better tracking of future motions of the same type.

The EKF provides a probabilistic framework for tracking, but supports only the case where observation and motion PDFs can be approximated as multivariate Gaussians. While Gaussian uncertainty is sufficient for modelling many motion and measurement noise sources, the EKF has been shown to fail catastrophically in cases where the true probability function has a very different shape. Attempts to track objects moving against a very cluttered background, where measurement densities include the chance of detecting erroneous image features, led to the first application of particle filtering in visual tracking (Blake & Isard 1998). In particle filtering (figure 2*d*), the posterior density $p(\boldsymbol{X} \mid \mathbf{Z})$ representing current knowledge about the model state after incorporation of all measurements is represented by a finite set of *weighted particles*, vector samples $\boldsymbol{s}^{(n)}$ of the state with scalar weights $\pi^{(n)}$. The weights are normalized such that

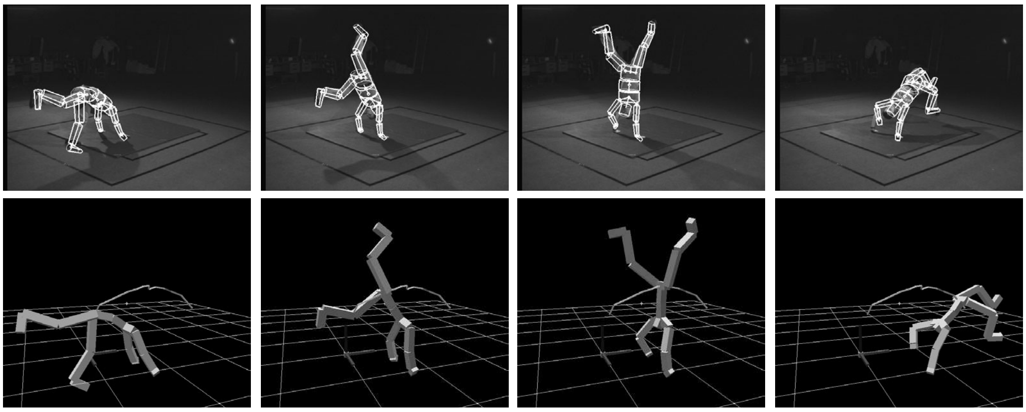
$$\sum_{n=1}^{N} \pi^{(n)} = 1.$$

The state $\boldsymbol{X}$ can be estimated by the mean

$$\boldsymbol{X} = \sum_{n=1}^{N} \pi^{(n)} \boldsymbol{s}^{(n)}$$

of the particle set, and variance and other high-order moments can also be calculated.

Essentially, a smooth PDF is approximated by a finite collection of weighted sample points— each like a Dirac delta function, a point spike—and it can be shown



0 s 0.3 s 0.6 s 1.2 s

Figure 3. Multi-camera tracking of a handstand using annealed particle filtering permits recovery of body geometry for character animation.

that as the number of points tends to infinity the behaviour of the particle set is indistinguishable from that of the smooth function. Tracking with a particle filter works by

(i) resampling, in which a weighted particle set is transformed into a set of evenly weighted particles distributed with concentration dependent on probability density;

(ii) stochastic movement and dispersion of the particle set in accordance with a motion model to represent the growth of uncertainty during movement of the tracked object;

(iii) measurement, in which the likelihood function is evaluated at each particle site, producing a new weight for each particle proportional to how well it fits image data.

The weighted particle set produced represents the new probability density after movement and measurement. Particle filtering works well for tracking in clutter because it can represent arbitrary functional shapes and propagate multiple hypotheses. Less likely model configurations will not be thrown away immediately but given a chance to prove themselves later. Blake & Isard (1998) and others have demonstrated very robust tracking of hands and other fast-moving objects against complex backgrounds.

While particle filtering represents perhaps the current state of the art of real-time Bayesian inference, serious problems arise in applications where sensible modelling leaves high-dimensional parameter spaces. A common example is detailed tracking of the whole human body, which is typically modelled in terms of 30 or so parameters representing the angles of an articulated model of rigid sections (one parameter for the elbow angle, three for the ball-joint at the shoulder, etc.). Essentially, the number of particles needed to populate such a high-dimensional space is far too high to be manageable in real time (MacCormick & Blake 2000). In this situation a modified algorithm called the annealed particle filter (see Deutscher *et al.* (2001) and figure 3) has proven to be successful and operate at near-real-time rates, though at the cost of loss of true Bayesian representation and occasional instability.

3. Simultaneous localization and mapping

Consider a mobile robot which finds itself within an environment about whose layout it has little or no prior knowledge—this could be the surface of a remote planet, underwater, or within an office or hospital. Whatever specific task it is charged with, a basic skill that is likely to be required is navigation: the robot should be able to map its environment autonomously such that places of interest, once located, can be revisited repeatedly. This problem of incremental mapping while maintaining continuous localization relative to the map as it is created is known in the robotics community as simultaneous localization and mapping (SLAM).

Humans are able effortlessly to 'map' unfamiliar areas they enter, such that specific parts of the area can be revisited and recognized at will, though it is likely that the representations a human builds are quite different from the absolute maps of streets or buildings that we are familiar with on paper (and which robot SLAM systems usually aim to create). A human would probably not be able to point reliably to a known place in another part of town, or estimate its distance accurately, though would have no trouble in walking there—human mapping seems to form a series of associations between objects and locations seen along a route (landmarks of one type or another), so that navigation can be achieved by moving from one to the other even if the overall geometry is unknown. However, perhaps a robot mapping system can do better than this and provide reliable navigation even when a human would be lost (when many landmarks look similar, for instance, in a dense forest, or when they are very sparse in a desert).

The global positioning system (GPS) has recently become available, and uses satellites as artificial landmark beacons in known positions (though they are moving relative to the Earth's surface, the trajectories of the satellites can be calculated with great accuracy, and they transmit time-stamped signals to a user's receiver). However, GPS does not work indoors, underwater, on other planets, or if the signals are Change OK? interrupted or encoded for some reason—and it does not give the kind of final accuracy often required of robots. There is therefore still great interest in robots which can makes their own maps from natural features detectable using their sensors. The interesting facet of the SLAM problem is that whether the goal of the exercise is for the robot to maintain a position estimate within an area or to map it for future reference, it must accomplish both tasks on the fly.

Autonomous map building is a process which must be carefully undertaken. A map which is made by a traveller or robot who does not have some external measure of ego-motion is fundamentally limited in its accuracy. The problem is caused by the compound errors of successive measurements. Consider, for example, a human given the task of drawing a very long, straight line on the ground, but equipped with only a 30 cm ruler, and unable to use any external references such as a compass or the bearing of the sun. The first few metres would be easy, since it would be possible to look back to the start of the line when aligning the ruler to draw a new section. Once this had gone out of view, though, only the recently drawn nearby segment would be available for reference. Any small error in the alignment of this segment would lead to a misalignment of new additions. At a large distance from the starting point, the cumulative uncertainty will be great, and it will be impossible to say with any certainty whether the parts of line currently being drawn are parallel to the original direction. Changing the measurement process

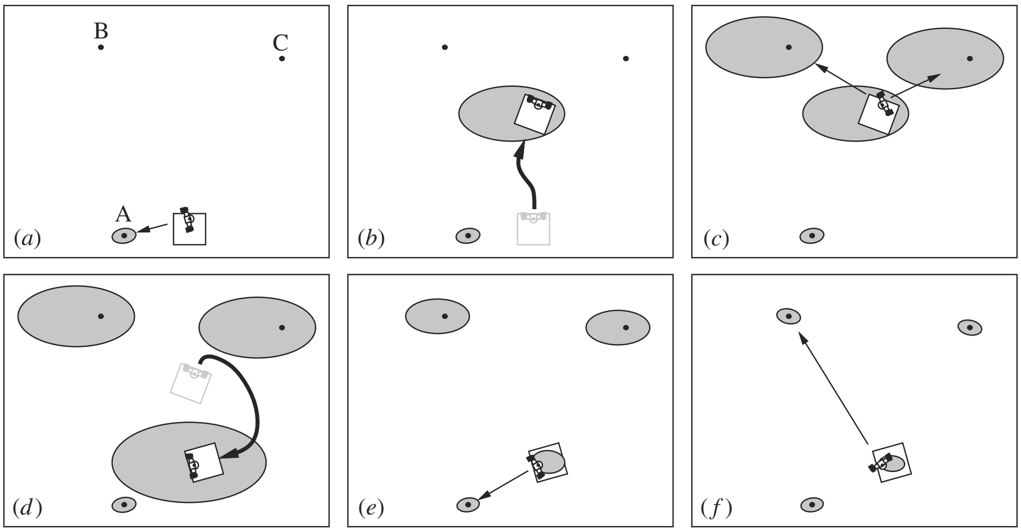


Figure 4. Six steps in an example of sequential map-building, where a robot moving in two dimensions is assumed to have a fairly accurate sensor allowing it to detect the relative location of point features, and less accurate odometry for dead-reckoning motion estimation. Black points are the true locations of environmental features, and grey areas represent uncertain estimates of the feature and robot positions. (*a*) Initial measurement of A; (*b*) move forward; (*c*) initial measurement of B and C; (*d*) move back; (*e*) refind A; (*f*) refind B.

could improve matters—if, for instance, flags could be placed at regular intervals along the line which were visible from a long way, then correct alignment could be better achieved over longer distances. However, eventually the original flags would disappear from view and errors would accumulate—just at a slower rate than before.

Something similar will happen in a robot map-building system, where at a certain time measurements can be made of only a certain set of features which are visible from the current position—probably these will in general be those that are nearby, but there are usually other criteria, such as occlusion (some features may be hidden behind other objects from some points of view) or maximum viewing angle. It will be possible to be confident about the robot's position relative to the features which can currently be seen, but decreasingly so as features which have been measured in the more distant past are considered. To give a flavour of the interdependence of estimates in sequential map-building, and emphasize that it is important to estimate robot and feature positions together, steps from a simple scenario are depicted in figure 4. The sequence of robot behaviour here is not intended to be optimal; the point is that a map-building algorithm should be able to cope with arbitrary actions and make use of all the information it obtains.

In figure 4*a*, a robot is dropped into an environment of which it has no prior knowledge. Defining a coordinate frame at this starting position, it uses a sensor to identify feature A and measure its position. The sensor is quite accurate, but there is some uncertainty in this measurement which transposes into the small grey area representing the uncertainty in the estimate of the feature's position. The robot drives forward in figure 4*b*, during this time making an estimate of its motion using dead-reckoning (for instance counting the turns of its wheels). This type of motion

Author: addition of text to caption (and removal from figure) and changes to the text OK?

estimation is notoriously inaccurate and causes motion uncertainties which grow without bound over time, and this is reflected in the large uncertainty region around the robot representing its estimate of its position. In figure 4*c*, the robot makes initial measurements of features B and C. Since the robot's own position estimate is uncertain at this time, its estimates of the locations of B and C have large uncertainty regions, equivalent to the robot position uncertainty plus the smaller sensor measurement uncertainty. However, although it cannot be represented in the diagram, the estimates in the locations of the robot, B and C are all coupled at this point. Their relative positions are well known; what is uncertain is the position of the group as a whole.

The robot turns and drives back to near its starting position in figure 4*d*. During this motion its estimate of its own position, again updated with dead-reckoning, grows even more uncertain. In figure 4*e* though, remeasuring feature A, whose absolute location is well known, allows the robot dramatically to improve its position estimate. The important thing to notice is that this measurement also improves the estimate of the locations of features B and C. Although the robot had driven farther since first measuring them, estimates of these feature positions were still partly coupled to the robot state, so improving the robot estimate also upgrades the feature estimates. The feature estimates are further improved in figure 4*f*, where the robot directly remeasures feature B. This measurement, while of course improving the estimate of B, also improves C thanks to their interdependence (the relative displacement of B and C is well known). At this stage, all estimates are good and the robot has built a useful map.

(*a*) *Map-building with first-order uncertainty propagation*

Implementation of SLAM in real robotic systems has been achieved with most success using the EKF described earlier in the discussion of tracking (the huge number of coupled parameters involved making particle filters or similar unfeasible). All parameters (the positions of the robot and mapped features) are represented with Gaussian uncertainty distributions, and these distributions are propagated through time as the robot moves and measures features. Current estimates of the state of the robot and the scene features which are known about are stored in the system state vector $\hat{\boldsymbol{x}}$, and the uncertainty of the estimates in the covariance matrix P. Both $\hat{\boldsymbol{x}}$ and P will change in size dynamically as features are added to or deleted from the map. They are partitioned as follows:

$$\hat{\boldsymbol{x}} = \begin{pmatrix} \hat{\boldsymbol{x}}_v \\ \hat{\boldsymbol{y}}_1 \\ \hat{\boldsymbol{y}}_2 \\ \vdots \end{pmatrix}, \qquad \mathsf{P} = \begin{bmatrix} \mathsf{P}_{xx} & \mathsf{P}_{xy_1} & \mathsf{P}_{xy_2} & \cdots \\ \mathsf{P}_{y_1x} & \mathsf{P}_{y_1y_1} & \mathsf{P}_{y_1y_2} & \cdots \\ \mathsf{P}_{y_2x} & \mathsf{P}_{y_2y_1} & \mathsf{P}_{y_2y_2} & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix},$$

where $\hat{\boldsymbol{x}}_v$ is the robot state estimate, and $\hat{\boldsymbol{y}}_i$ that of the ith feature. By the 'state' of the robot and features, generally we mean a vector of all the modelled parameters of interest relating to those objects. Of course this means their positions, defined by a number of parameters depending on the geometrical type of the object and dimensionality of the map; but also, there may be other parameters

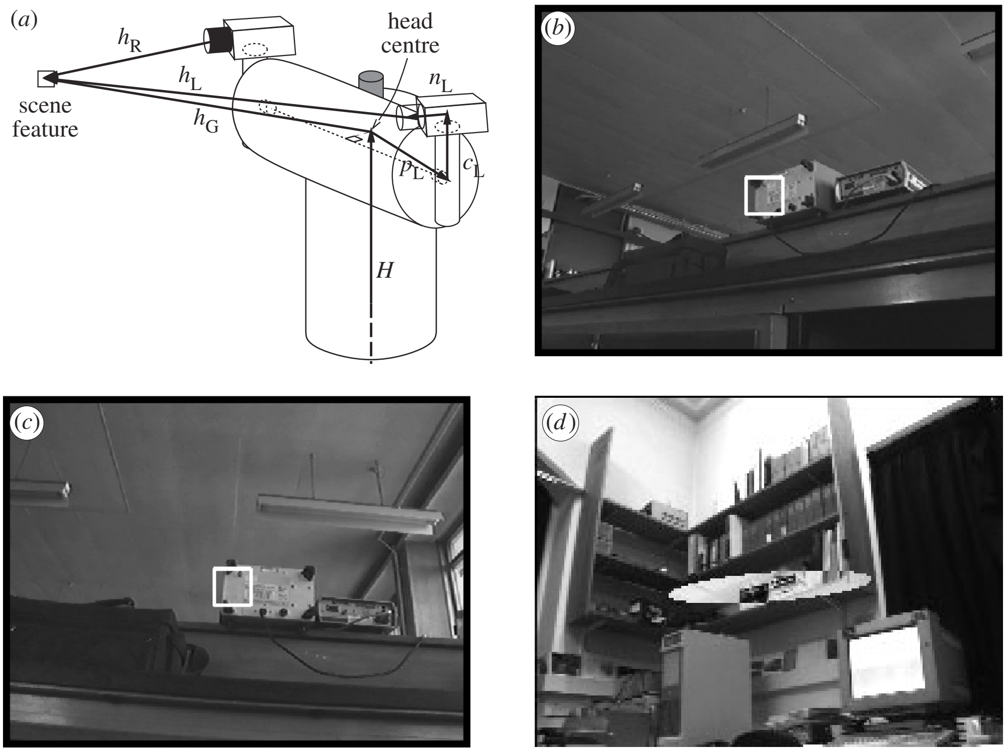


Figure 5. Feature detection and matching using active vision. (*a*) High-performance binocular camera platform ('active head') which was mounted on a mobile robot platform. (*b*), (*c*) Normalized cross-correlation allows feature templates to be repeatedly identified despite robot motion; three-dimensional measurements of the position of a feature relative to the robot are achieved by stereo fixation. (*d*) Current uncertainty in the position of a feature with respect to the robot is projected into image space to give an elliptical search region. The feature template is known to lie within this region with a high probability and thus expensive search can be minimized.

which we would like to estimate, usually because they will affect future motion or measurements.

(*b*) *SLAM using active vision*

Figures 5 and 6 depict EKF-based SLAM in action in an implementation using binocular active vision (Davison & Murray 2002). A sparse map of automatically detected natural features (the corners of boxes, windows, doors, etc.) was constructed during real-time navigation through a cluttered environment, and visual measurement combined with odometry for accurate localization. Of particular interest was the active role played by the uncertainty propagation enabled by the Bayesian framework used: automatic decisions about which features to measure at a given time were based on the projected reduction in uncertainty (gain in information) that they would provide. The robot's active motorized cameras were seen to switch attention frequently between features covering a wide field of view, recalling the way a human repeatedly glances at landmarks when navigating. More concentrated tracking behaviour when manoeuvring past obstacles was also observed.

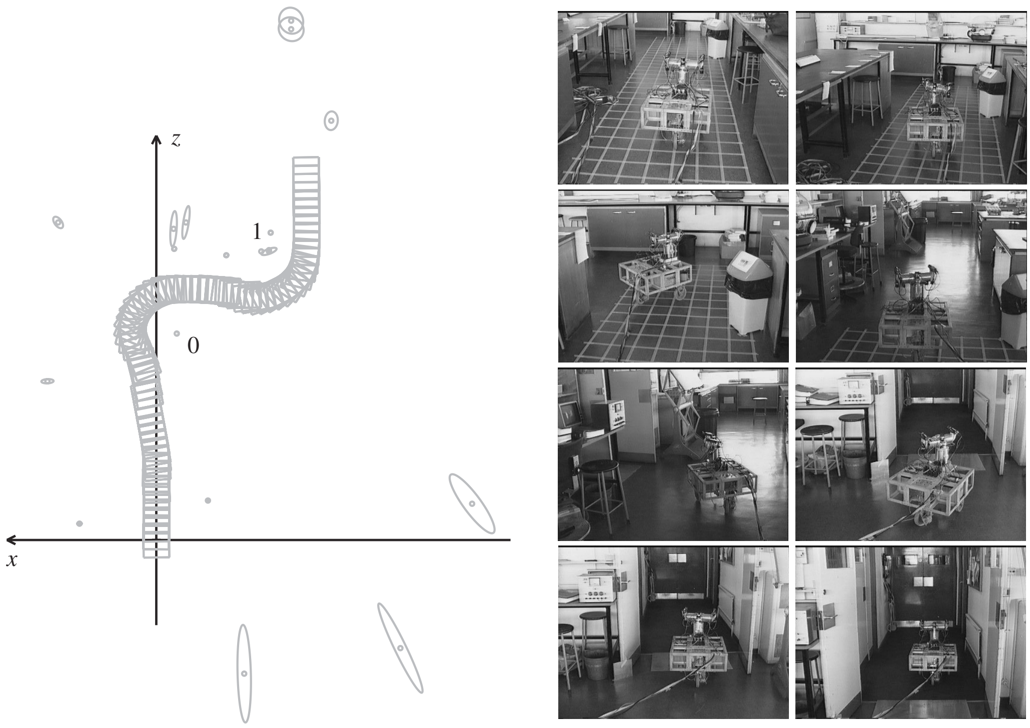


Figure 6. The estimated trajectory and frames cut from a video as a robot navigated autonomously through a cluttered laboratory and built a map of natural visual features. Final localization accuracy was of the order of a few centimetres.

4. Conclusions

As computer processors continue to get faster, real-time systems will be able to understand the natural world with increasing facility and penetrate domains which are currently too complex. The development of large-scale autonomous robots will continue to advance—in Japan especially, where there are several humanoid robot projects. More likely, however, to have a rapid impact on daily life are various increasingly intelligent embedded devices—the automated vacuum cleaners, security systems, vehicle safety systems and so on which will make decisions and take action on the basis of their sensor data.

A number of current research issues will no doubt grow in importance. The rise of Bayesian methods will continue, not only in the domain of artificial intelligence but in all areas of science as they are come to be seen as the only rigorous framework within which to solve problems of inference from incomplete information. In particular, methods for model selection, the process of automatically choosing the model which most efficiently explains a certain dataset, are being developed based on general Bayesian principles (e.g. Torr *et al*. 1999). Information theory, a natural extension of probability theory developed in the 1950s, has been conspicuously underused outside of signal-processing research. It provides a framework for comparing in absolute terms the information content and therefore value of measurements and actions in terms of achieving a certain goal and is now coming into its own when implemented on

Word added – OK?

modern computers. The real-time domain, where processing resources are limited, will benefit most from the application of this theory.

In SLAM, as research is pushed outside the simple two-dimensional office or laboratory scenarios previously considered towards outdoor or undersea environments, the issue of maintaining real-time operation as feature maps grow larger is key, and much research is developing new methods for rescheduling as well as approximating Bayesian computations. The current goal is a rigorous algorithm for constant time mapping, which would mean that arbitrarily large areas could be mapped by a mobile robot with finite processing resources.

Alongside these technological implications, the author is convinced that this research will continue to inspire cognitive scientists and others to consider its impact on understanding how human minds work. If one accepts, as many do, that the human brain is equivalent to a digital computer in the calculations it can perform, there must surely be many parallels to draw between artificial real-time embedded systems and their biological counterparts. This philosophy has already led for instance to advances in the study of the human vision system.

The author is supported by an EPSRC Advanced Research Fellowship. He is grateful to his collaborators, particularly Nobuyuki Kita, Ian Reid and David Murray, and to Walterio Mayol and Nick Molton for useful discussions.

References

Blake, A. & Isard, M. 1998 *Active contours*. Springer.

Davison, A. J. & Murray, D. W. 2002 Simultaneous localization and map-building using active vision. *IEEE Trans. Pattern Analysis Machine Intell.* **24**, 865–880.

Dawkins, R. 1998 *Unweaving the rainbow*. London: Penguin.

Dennett, D. C. 1984 Cognitive wheels: the frame problem of AI. In *Minds, machines and evolution* (ed. C. Hookway), pp. 129–151. Cambridge University Press.

Deutscher, J., Davison, A. J. & Reid, I. D. 2001 Automatic partitioning of high-dimensional search spaces associated with articulated body motion capture. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, 2001.*

Harris, C. G. 1992 Tracking with rigid models. In *Active vision* (ed. A. Blake & A. Yuille). Cambridge, MA: MIT Press.

Jaynes, E. T. 2003 *Probability theory—the logic of science.* Cambridge University Press.

Lowe, D. G. 1992 Robust model-based motion tracking through the integration of search and estimation. *Int. J. Comput. Vis.* **8**, 113–122.

MacCormick, J. & Blake, A. 2000 Partitioned sampling, articulated objects and interface-quality hand tracking. In *Proc. Eur. Conf. on Computer Vision, Dublin, 2000*, pp. 3–19.

Minsky, M. 1985 *The society of mind.* New York: Simon & Schuster.

Pollefeys, M., van Gool, L., Zisserman, A. & Fitzgibbon, A. W. (eds) 2000 *3D structure from images—SMILE 2000.* Lecture Notes in Computer Science, no. 2018. Springer.

Reynard, D., Wildenberg, A. P., Blake, A. & Marchant, J. 1996 Learning dynamics of complex motions from image sequences. In *Proc. Eur. Conf. on Computer Vision, Cambridge, April 1996*, pp. 357–368.

Torr, P. H. S., Szeliski, R. & Anandan, P. 1999 An integrated Bayesian approach to layer extraction from image sequences. In *Proc. Int. Conf. on Computer Vision, Corfu, 1999*, pp. 983–990.