Autonomous Exploration: Driven by Uncertainty

Peter Whaite and Frank P. Ferrie

TR-CIM-93-17 1993 (Modified September 1995)

Artificial Perception Laboratory
Centre for Intelligent Machines
McGill University
Montréal, Québec, Canada

Submitted to *IEEE Trans. Pattern Analysis and Machine Intelligence*, March 1994

Postal Address: 3480 University Street, Montréal, Québec, Canada H3A 2A7
Telephone: (514) 398-6319 Telex: 05 268510 FAX: (514) 398-7348
Email: cim@cim.mcgill.ca

Autonomous Exploration: Driven by Uncertainty

Peter Whaite and Frank P. Ferrie

Abstract

Passively accepting measurements of the world is not enough, as the data we obtain is always incomplete, and the inferences made from it uncertain to a degree which is often unacceptable. If we are to build machines that operate autonomously they will always be faced with this dilemma, and can only be successful if they play a much more active role. This paper presents such a machine. It deliberately seeks out those parts of the world which maximize the fidelity of its internal representations, and keeps searching until those representations are acceptable. We call this paradigm *autonomous exploration*, and the machine an autonomous explorer.

This paper has two major contributions. The first is a theory that tells us how to explore, and which confirms the intuitive ideas we have put forward previously. The second is an implementation of that theory. In our laboratory we have constructed a working autonomous explorer and here for the first time show it in action. The system is entirely bottom-up and does not depend on any a priori knowledge of the environment. To our knowledge it is the first to have successfully closed the loop between gaze planning and the inference of complex 3D models.

Résumé

Accepter passivement des mesures du monde est insuffisant étant donné que les données obtenues sont toujours incomplêtes, et que les inférences qui en découlent sont incertaines à un degré tel qu'elles sont souvent inacceptable. Si l'on construit des machines qui opèrent de façon autonome, elles feront toujours face a ce dilemme et ne réussiront que si elles jouent un rôle beaucoup plus actif. Ce papier présente une telle machine. Elle cherche délibérément ces parties du monde qui permettent d'accroître la fidélité de la représentation interne, et continue cette recherche jusqu'à ce que cette représentation soit acceptable. On appelle ce paradigme, l'*exploration autonome*; et on appelle la machine, l'explorateur autonome.

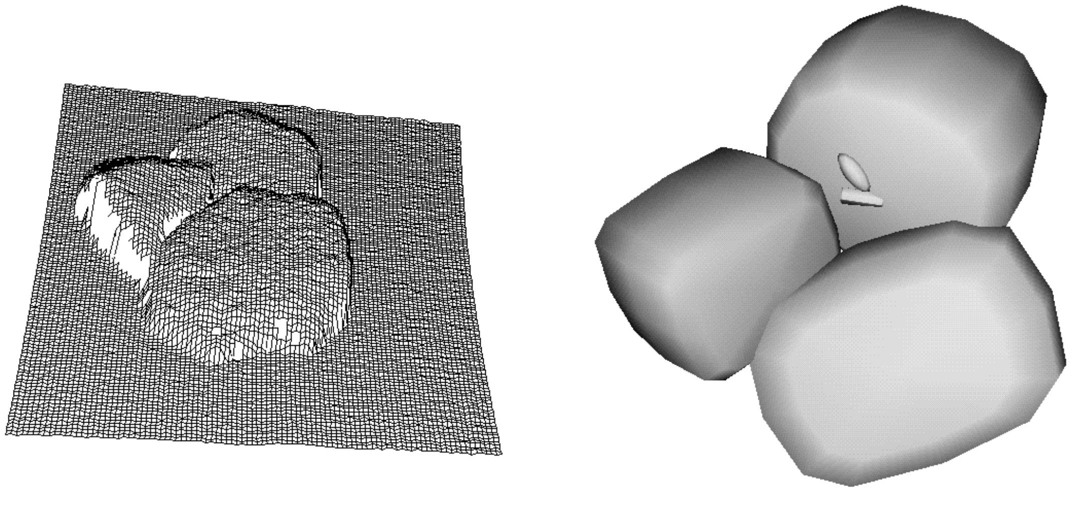
Ce papier contient deux contributions majeures. La première est une théorie qui nous dit comment explorer, et qui confirme les idées intuitives que nous avons mentionnées précédemment. La deuxième est une implémentation de cette théorie. Dans notre laboratoire, nous avons construit un explorateur autonome fonctionnel et ici pour la première fois, nous pouvons le montrer en action. Le système est entièrement guidé par les données et ne dépend aucunement de connaissance sur l'environnement acquise a priori. À notre connaissance, c'est la première fois que la boucle de la planification du regard et de l'inférence de modèles complexes tridimensionnels est complêtée avec succès.

Acknowledgements

This work was made possible by NSERC grant OGPIN 011, and by funding from the NCE IRIS network.

1. Introduction

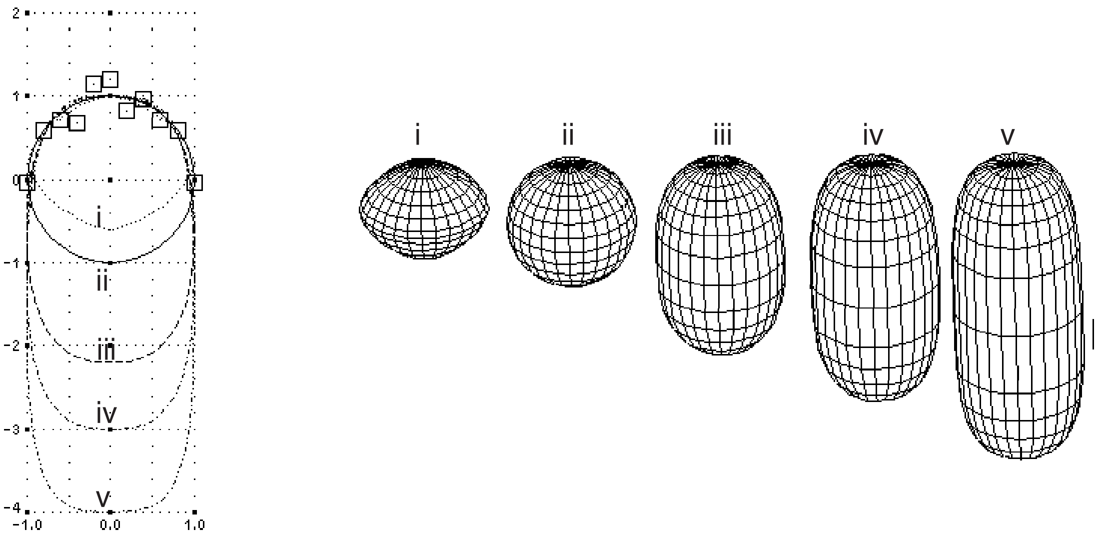
One can define active exploration as a process in which an observer can interact with its surroundings by moving about and collecting information in order to learn about its environment. This ability is essential for autonomous systems which must operate in unstructured environments where it is difficult (if not impossible) to characterise the environment beforehand. Consider, for example, a mobile robot designed to collect rock samples for planetary exploration [9]. In order to grasp and manipulate such samples, information is required about their three-dimensional shape. But given the wide range of shapes that are possible, it is not feasible to represent each and every instance. Shape descriptions must be computed from more general purpose models that can be adapted according to measurements obtained by sensors. In the context of artificial perception, the latter often takes the form of determining the parameters of some model used to reflect the salient properties of the environment [15, 26].



(a) (b)

FIGURE 1. (a) Laser rangefinder image of a rock pile. (b) Model of the rock pile using superquadrics.

Figure 1a shows a range map of a rock pile obtained with a laser range-finding system. From an analysis of the geometric structure of the acquired surfaces, the data are partitioned into patches corresponding to the component rocks. An approximation of the position, orientation, and shape of each component is then determined by fits to superquadric models (Figure 1b) [8]. While this strategy appears to work well in the example shown, it is in fact flawed. This can be seen in the example in Figure 2 which shows the same fitting process applied to points sampled from the surface of a noisy hemisphere. Each of the resulting models (Figure 2b) describes the data to within the same error of fit [30, 31], yet the models look quite different as they move away from the data. The problem is that the data acquired do not sufficiently constrain the model. This should not be surprising given that only part of the surface is visible in a given view. The example of Figure 1 worked because an additional constraint was available, namely the distance from the camera to the supporting plane.



(a) (b)

FIGURE 2. Model uncertainty. (a) Data points sampled from the surface of a noisy hemisphere. (b) Fits to the data using superellipsoid models. Each of the 5 models shown fits the data to within the same tolerance.

Without additional information in the form of such constraints there is no alternative but to collect it in the form of additional data. In this respect some data are better than others, so the system must actively seek out those places in the world that have the most useful information. In other words, and as stated by Bajcsy with the introduction of the active perception and exploration paradigms [3, 4], the system needs to *explore* its environment and it must keep doing so until there is a sufficient basis from which to make useful inferences.

The idea of moving a sensor to constrain interpretation has a long history in the computer vision literature. Many of the methods rely on global geometric arguments to overcome the limitations of the sensor in specific situations. For example, Connolly [12], then Ahuja and Veenstra [1] have considered the problem of the views needed to build an octree representation of a 3D scene; Tarabanis and Tsai [29] have worked on a theoretical analysis of the best camera viewpoint for detecting a generic feature; and more recently Maver and Bajcsy [24] developed a technique for filling in the range image shadows when sampling a scene with a light stripe range finder.

Some important general principals were established with the introduction of the active vision paradigm by Aloimonos [2]. The basis of the approach is a parametrized model of how the sensor views features in the scene. Continuously varying the view parameters causes the observed features to undergo measurable local transformations which can be used to simplify and constrain the computation of unknown scene parameters. For example, Kutulakos and Dyer [21] cleverly exploit the differential properties of smooth surfaces to model local changes in the appearance of an occluding contour due to camera movement. This knowledge shows them how to position the camera, first to extract occluding contours from an edge map, and then to use the extracted contours to sweep out the complete 3D shape.

Much of the power of the active vision paradigm comes from the simplification of only having to model local changes to local features, so it is more applicable to problems in low and intermediate vision where local features play the major role. However in a complex vision system these problems are only the first steps towards a description of the world which will enable the system to perform some specific task, so if there are any deficiencies in the low levels it becomes essential that they be taken into account. For example it is impossible to reconstruct some surface concavities using the method of occluding contours proposed by Kutulakos and Dyer. Without that knowledge decisions based on the reconstructed surface could result in a system failure. With it, other instrumentation can be called upon to make up for the method's shortcomings.

It has also been demonstrated that "high-level" control of data acquisition is very useful. For example, Birnbaum, Brand, and Cooper [6] present a system (`BUSTER`) which uses a generative causal semantics to control the visual exploration of arbitrary stacked block structures. The system works by encoding the knowledge that the scene is stable under the force of gravity in a simple set of rules. The rules are used to direct visual attention to search for blocks which make an unstable scene stable.

Thus in complex systems such as the one we are working on, the information used to select sensor viewpoints can come from many different sources and must be combined in ways that go beyond the principals established by the active vision paradigm. One can attempt to formulate general frameworks to represent and solve these problems [13, 17] but tractable solutions often require the development of specific insight into the structure of the problem. We attempt to avoid this dilemma with an active vision approach that generalizes the characterization and manipulation of uncertainty. As a consequence it can be applied wherever the interaction between a sensor and its environment is modelled parametrically.

We begin development of the theory in §2 with a description of a system where the location of a sensor is determined by a set of control parameters, and in which the interaction between the sensor and its environment is modeled by the linear combination of an arbitrary set of basis functions. We show how to find a maximum likelihood estimate of unknown model parameters from a set of noisy measurements, and illustrate that the parameter covariances represent and encapsulate the model uncertainty.

In §3 we continue with an information based measure of uncertainty, the determinant of the covariances, and consider the problem of how to reduce it incrementally by taking a single extra measurement. We find a theoretical solution to this problem which is identical to a proposition we have made in previous work [31] — that the best sensor locations are those where our ability to predict is worst.

This leads to a gaze-planning strategy, described in §4, that uses model uncertainty as a basis for selecting viewpoints. We show theoretically that the strategy ensures convergence when applied to linear models, and present experimental results for non-linear superellipsoid models which verify that the linear theory can be applied as a local approximation. We test the speed with which it can estimate superellipsoid model parameters and show that, unlike other strategies, it not only does this faster but can also adapt robustly to changes in model pose and size.

By closing the loop around bottom-up vision with this gaze-planning strategy, we design in §5 an exploration system that is capable of autonomously building a description of its environment from a sequence of exploratory probes. This leads to the implementation in §6, and a sequence of experimental results which show how the system performs on real scenes obtained with a mobile laser rangefinding system.

2. Linear Model Inference

Although the volumetric models we use to represent surfaces in a 3D scene are highly non-linear many of the basic concepts and insights are obtained from a study of the linear case. Non-linear analytic solutions are usually hard to obtain and invariably one must resort to numerical iterative techniques to get results. Once this is done however the system can be linearized around the solution, and the linear analysis applied. Provided that perturbations in the state of the system are small enough, the linear analysis is valid and not as severe a restriction as might first appear.

2.1. The linear model. Consider the scenario where we have a sensor making measurements of a physical system and where the location of the sensor is determined by specifying a vector of control parameters $\mathbf{x}$.[1] In a linear model data measurements can be predicted by a linear combination of basis functions defined over the space of control parameters. That is given known model parameters $\mathbf{m}$ the measurement obtained at location $\mathbf{x}_i$ can be written in the form

$$d_i = \mathbf{g}_i^T \mathbf{m} \tag{1}$$

where $\mathbf{g}_i^T = (g_1(\mathbf{x}_i), \ldots, g_p(\mathbf{x}_i))$ are the basis functions evaluated at $\mathbf{x}_i$.

[1] By "location" we mean the sensor's location in the space of control parameters. This could be its physical location, but could also be many other things, e.g. the direction of gaze, sampling density, beam intensity, etc.

The basis functions themselves do not have to be linear in the control parameters. For example suppose our sensor is a depth probe constrained to move in a horizontal plane. Its location is given by the cartesian coordinates $\mathbf{x}_i^T = (x_i, y_i)$, and the measurement d_i it makes there is the vertical distance to some surface. We can choose to model the surface as a general quadratic $d_i = ax_i^2 + bx_iy_i + cy_i^2 + dx_i + ey_i + f$, then the model parameters are $\mathbf{m}^T = (a, b, c, d, e, f)$, and the basis functions are $\mathbf{g}_i^T = (x_i^2, x_iy_i, y_i^2, x_i, y_i, 1)$. These are definitely non-linear in the control parameters though the model as a whole is still linear, and the linear analysis to follow applies without approximation.

2.2. The maximum likelihood solution. Of course it is usually the case that the model parameters are not known, and the whole purpose of making measurements is to solve the *inverse problem*, that is to find model parameters that explain the observed measurements. If we have made n measurements such a solution amounts to solving the linear system of equations $\{d_i = \mathbf{g}_i^T\mathbf{m} \; ; \; i = 1 \dots n\}$ which can be written in the form

$$\mathbf{d} = \mathbf{G}\mathbf{m}. \tag{2}$$

where the $\mathbf{g}_i$ form the rows of $\mathbf{G}$.

In general it is not possible to find an exact solution to (2) as the data are contaminated with noise. However when the measurement errors are randomly and independently sampled from a normal distribution with zero mean and variance σ^2 it can be shown that the *maximum likelihood estimate* $\widehat{\mathbf{m}}$ of the true model $\mathbf{m}_{T}$ is given by the pseudo inverse

$$\widehat{\mathbf{m}} = \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{d}. \tag{3}$$

Furthermore, the parameter errors $\hat{\mathbf{e}}(\mathbf{m}_{T}) = \mathbf{m}_{T} - \widehat{\mathbf{m}}$ are distributed as a zero mean $p-$variate normal distribution

$$N(\hat{\mathbf{e}}(\mathbf{m}_{T}); \mathbf{C}) = \frac{1}{\sqrt{(2\pi)^p \, |\mathbf{C}|}} \exp\left(-\frac{1}{2}\,\hat{\mathbf{e}}(\mathbf{m}_{T})^T\mathbf{C}^{-1}\hat{\mathbf{e}}(\mathbf{m}_{T})\right) \tag{4}$$

with covariances

$$\mathbf{C} = \sigma^2\left(\mathbf{G}^T\mathbf{G}\right)^{-1} = \sigma^2\,\mathbf{H} \tag{5}$$

which determine the dispersion of the probability density in the different parameter directions.

2.3. The uncertainty of the inverse solution. The presence of random errors in the inverse solution is an indication of its inherent uncertainty or non-uniqueness. To see this consider the quadratic form that appears in the exponent of (4) as $Q(\mathbf{m}_{T})$, that is

$$Q(\mathbf{m}) = \hat{\mathbf{e}}(\mathbf{m})^T \mathbf{C}^{-1} \hat{\mathbf{e}}(\mathbf{m}) = \frac{1}{\sigma^2}\, \hat{\mathbf{e}}(\mathbf{m})^T \mathbf{H}\, \hat{\mathbf{e}}(\mathbf{m}) \tag{6}$$

(7)

The true model must lie somewhere on the hyper-surface $Q(\mathbf{m}) = Q(\mathbf{m}_T)$. Because $\mathbf{C}^{-1}$ is symmetric and positive definite this surface is ellipsoidal.

Now we do not know the value of $Q(\mathbf{m}_T)$ but it is a well known result of statistical theory that the quantity is randomly sampled from a chi-square distribution with p degrees of freedom[25]. For some *confidence level* γ we can find from that distribution a number χ^2_γ for which there is a probability of γ that $Q(\mathbf{m}_T) < \chi^2_\gamma$. It follows that there is also a probability of γ that the ellipsoid

$$\sigma^2 Q(\mathbf{m}) = \hat{\mathbf{e}}(\mathbf{m})^T \mathbf{H}\, \hat{\mathbf{e}}(\mathbf{m}) = \sigma^2 \chi^2_\gamma \tag{8}$$

will enclose the true model,[2] and for this reason it is called the *ellipsoid of confidence*. The ellipsoid of confidence gives us a useful visual image of the non-uniqueness of the inverse solution as it shows us the region of model parameter space in which any of the models could be the true one. Further localization of this region can only be made by lowering the confidence level, or equivalently, by being increasingly mistaken in ones belief that the true model is still enclosed within it.

3. Reducing Uncertainty

When the parameter errors are normally distributed, the covariances can be used to communicate the uncertainty to the tasks which make use of the inverse solution. By applying classic statistical methods the solution can be tested to see if it meets standards of acceptability predetermined to keep system failure rates below bearable levels. When the uncertainty is such that it does not we must take steps to improve the uniqueness until it does.

One way is to build better sensors with lower noise figures. It can be seen from (8) that lower values of σ^2 simply scale the ellipsoid of confidence to be smaller. The eccentricity and pose of the ellipsoid are unchanged, though in the limit for the perfect instrument ($\sigma^2 = 0$) the ellipsoid shrinks to become a point at the true model, and there is complete certainty.

Another way in which we may affect the character of the uncertainty is through the choice of measurement locations. From (2) we see that $\mathbf{H} = \mathbf{G}^T\mathbf{G}$ is dependent only upon the $g_j(\mathbf{x}_i)$, that is on the form of the basis functions and the locations of the measurements.[3] As $\mathbf{H}$ defines the eccentricity, size, and pose of the ellipsoid of confidence we have a potentially powerful method for controlling uncertainty. By

[2] This statement is often misinterpreted to mean there is a probability of γ that $\mathbf{m}_T$ is inside *this particular* ellipsoid. This is not so — it means that when (8) is repeatedly computed from many independently sampled data sets, then $\mathbf{m}_T$ will fall inside the computed ellipsoids γ% of the time

[3] It is perhaps surprising that the uncertainty is totally independent of the true model and therefore of the actual measurements. This is not true when the model is non-linear however.

simply selecting appropriate $\mathbf{x}_i$ we can cause the ellipsoid of confidence to fall in regions of parameter space which meet criterion of acceptability imposed by the task at hand.

3.1. A measure of uncertainty. A central question is what constitutes an appropriate criterion of acceptability. In general, it really depends upon the use to which the model parameters are put and can only be answered in an operational context. For example we are currently investigating the object recognition problem and there we need to obtain parameter covariances which allow us to discriminate between models in a data base. The ability to discriminate depends largely on the makeup of the data base. In some cases precise knowledge of only one model parameter is necessary for positive identification, whereas in others all must be found to a high degree of precision.

However a generally useful criterion is one based on the well known Shannon entropy, and which measures the amount of information contained in the probability distribution representing the parameter errors. It can be shown that maximizing this information is equivalent to a minimization of the determinant of the parameter covariances [23]. Geometrically $|\mathbf{C}|$ is the product of the eigenvalues of $\mathbf{C}$, which can be shown from (8) to be proportional to the square of the lengths of the axes of the ellipsoid of confidence. The determinant is therefore proportional to the square of the *volume* of the ellipsoid of confidence. Small values correspond to small volumes of model parameter space, which indicate that the true parameters are well localized, and that the knowledge or information we have about them is high.

A problem with using the determinant is that it will vanish if any single eigenvalue vanishes, despite the fact that high covariances could still exist along other directions in parameter space. However, as we shall demonstrate later (Figure 7), the incremental strategy we employ to minimize $|\mathbf{C}|$ prevents this from happening.

3.2. The location of uncertainty. Finding the sensor locations that minimize $|\mathbf{C}|$ is a useful result, both practically and for the insights it gives us in general. Here we will concentrate on what we call the *incremental problem*:

> *Given covariances $\mathbf{C}_n$ computed from n measurements, what single additional sensor location $\mathbf{x}_{n+1}$ will minimize $|\mathbf{C}_{n+1}|$?*

We note that this is equivalent to maximizing $|\mathbf{H}_{n+1}|$. Each additional measurement incrementally updates $\mathbf{H}$, so after $n+1$ measurements its value is

$$\mathbf{H}_{n+1} = \mathbf{H}_n + \mathbf{g}_{n+1}\mathbf{g}_{n+1}^T, \tag{9}$$

where $\mathbf{g}_{n+1}$ are the basis functions evaluated at each new location $\mathbf{x}_{n+1}$. After factoring $\mathbf{H}_n$ out on the right we have

$$\mathbf{H}_{n+1} = (\mathbf{I} + \mathbf{g}_{n+1}\mathbf{g}_{n+1}^T\mathbf{H}_n^{-1})\mathbf{H}_n,$$

and that the determinant is

$$\left|\mathbf{H}_{n+1}\right| = \left|(\mathbf{I} + \mathbf{g}_{n+1}\mathbf{g}_{n+1}^T\mathbf{H}_n^{-1})\right| \, |\mathbf{H}_n|. \tag{10}$$

Further simplification requires us to compute the determinant of the quantity $\mathbf{I} + \mathbf{g}_{n+1}\mathbf{g}_{n+1}^T\mathbf{H}_n^{-1}$. We do so by finding its eigenvalues with the aid of two results which follow from the basic definition of an eigenvalue (e.g. [19, definition 1.1.2]).

i) *If λ is an eigenvalue of $\mathbf{A}$ then $1+\lambda$ is an eigenvalue of $\mathbf{I}+\mathbf{A}$.*

ii) *If $\mathbf{x}$ is an m–dimensional vector and $\mathbf{A}$ an $m \times m$ matrix then there is only one eigenvalue of $\mathbf{x}\mathbf{x}^T\mathbf{A}$ and its value is $\mathbf{x}^T\mathbf{A}\mathbf{x}$.*

It follows that there is only one non-unit eigenvalue of $\mathbf{I} + \mathbf{g}_{n+1}\mathbf{g}_{n+1}^T\mathbf{H}_n^{-1}$ and that its value is $1 + \mathbf{g}_{n+1}^T\mathbf{H}_n^{-1}\mathbf{g}_{n+1}$. As the determinant of a matrix is the product of the eigenvalues (10) simplifies to

$$\left|\mathbf{H}_{n+1}\right| = \left(1 + \mathbf{g}_{n+1}^T\mathbf{H}_n^{-1}\mathbf{g}_{n+1}\right) \, |\mathbf{H}_n|, \tag{11}$$

or in terms of covariances,

$$\left|\mathbf{C}_{n+1}\right| = |\mathbf{C}_n| / \left(1 + \frac{\mathbf{g}_{n+1}^T\mathbf{C}_n\mathbf{g}_{n+1}}{\sigma^2}\right). \tag{12}$$

There is an important interpretation which can be placed on the quantity $\mathbf{g}_{n+1}^T\mathbf{C}_n\mathbf{g}_{n+1}$ and which confirms some of our intuitive notions as to where the best sensor placement is. The model parameters $\widehat{\mathbf{m}}_n$ estimated from the first n measurements allow us to *predict* the measurement that will be obtained at location $\mathbf{x}_{n+1}$

$$\hat{\mathbf{d}}(\mathbf{x}_{n+1}) = \mathbf{g}_{n+1}^T\widehat{\mathbf{m}}_n. \tag{13}$$

However, there are random errors in the estimated model parameters so we would expect there to be random errors in the predicted measurement as well. A well known result from statistics shows that there is a simple mapping between the model and prediction covariances [25].

> *If a random vector $\hat{\mathbf{x}}$ is sampled from a p–variate normal distribution with zero mean and covariances $\mathbf{C}$, and if $\mathbf{A}$ is a linear mapping $\hat{\mathbf{y}} = \mathbf{A}\hat{\mathbf{x}}$ to the q–dimensional vector $\hat{\mathbf{y}}$, then $\hat{\mathbf{y}}$ will be sampled from a q–variate normal distribution with covariances $\mathbf{A}\mathbf{C}\mathbf{A}^T$.*

In our case (13) is a linear mapping between the estimated model parameters and the measurement so by the above result the variance of $\hat{\mathbf{d}}$ is

$$\sigma_D^2(\mathbf{x}_{n+1}) = \mathbf{g}_{n+1}^T\mathbf{C}_n\mathbf{g}_{n+1}. \tag{14}$$

This is called the *prediction variance* at location $\mathbf{x}_{n+1}$.

The ratio of $|\mathbf{C}|$ before and after an additional measurement is seen from (12) to be

$$\frac{\left|\mathbf{C}_{n+1}\right|}{|\mathbf{C}_n|} = \frac{1}{1 + \sigma_D^2(\mathbf{x}_{n+1})/\sigma^2}. \tag{15}$$

A similar result can also be found in another field, the theory of optimal experiments as developed by Federov [14], and as expounded upon in the more accessible work of MacKay [23]. It shows us i) that adding any data will always result in a reduction of $|\mathbf{C}_n|$, and ii) that $|\mathbf{C}_{n+1}|$ can be minimized by taking a measurement at the location when σ_D^2 is largest. As our intuition might lead us to expect, any additional data is beneficial but the best locations to gather new measurements are those where our ability to predict is worst.

4. Looking: The Gaze Planning Strategy

The theory we have presented tells us the best place to take a *single measurement* but it is rarely the case that this measurement will meet our needs. Instead we have to collect data at a sequence of locations $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$ until the estimated parameter covariances $\mathbf{C}_n$ are acceptable. Here we consider the problem of how to choose such a sequence.

The approach we have taken is largely inspired by the problem on which we are working, i.e. to build volumetric representations of objects in a scene from data collected by a laser range scanner mounted on the end effector of a robot. The scanner's operation is controlled by a number of parameters which determine not only its position in the scene, but also the direction in which it is pointed, and the sampling density with which the beam is scanned. We will therefore refer to a sensor location as the scanner's "gaze", to the sequence of sensor locations as a "gaze trajectory", and to the problem of choosing a trajectory as a "gaze planning strategy". Note however that, although the strategy is presented in the context of 3D volumetric modelling, the methodology we develop is applicable to a much wider range of problems.

There are many trajectories that will eventually result in an acceptable value of $\mathbf{C}_n$ so the trick is to find those which are optimal from the point of view of the higher level tasks using the volumetric models. This requires us to formulate and minimize some cost function, for example the elapsed time, the quantity of computer resources used, or even the amount of energy consumed[4]. As we shall see below (§4.2) it is sometimes possible to tailor solutions to specific cases but we would prefer not to do this. Instead we would like to design a module that is generally useful for a wide range of tasks, even if it is at the expense of some operational optimality.

We start by considering linear models and give, in §4.1, the theoretical guarantee that a strategy which always moves towards uncertain viewpoints has the important property that the determinant of the covariances will converge below any arbitrary value. We conclude the linear case with an example, and point out that optimal trajectories are independent of the model being measured and can therefore always be computed off-line §4.2. However it is the non-linear case which interests us more so in §4.3 we show how the linear theory can be applied, and verify it with empirical results.

[4] The amount of energy consumed could be of critical importance, e.g. when the sensor is mounted on a space craft, or on a battery operated vehicle.

It is not possible to compute non-linear gaze planning strategies off-line so in §4.4 we develop a general iterative gradient strategy based on the model estimate at each iteration. We then present in §4.5 a specific implementation of this strategy, and the kinds of trajectories obtained when exploring the surfaces of superellipsoidal objects. In §4.6 we develop ways to measure the strategy's performance and present the results of simulation experiments confirming that, not only does the use of model uncertainty result in faster convergence, but that it does so even when model pose and size are arbitrarily changed. Finally in §4.7 we consider additional problems encountered when scenes are explored in the real world and outline the way in which we can decrease their effect. Ultimately however such problems are beyond the domain of the strategy, and point to the necessity of an overseer with task specific knowledge.

4.1. The convergence of linear models. When the model is linear we see from (15) that an additional measurement always results in some reduction of $|\mathbf{C}_n|$ irrespective of the sensor's location, and therefore that any gaze trajectory must result in a monotonically decreasing sequence $|\mathbf{C}_1|, |\mathbf{C}_2|, |\mathbf{C}_3|, \ldots$. However for some trajectories the sequence can converge to a positive, non-zero value, and it may prove impossible to reduce $|\mathbf{C}_n|$ to an adequate level. Fortunately, as the following argument shows, our condition that we always measure at locations of high σ_D^2 prevents this from happening.

First we note that $\mathbf{C}_n = \sigma^2 \mathbf{H}_n^{-1}$ so $|\mathbf{C}_n| \to 0$ provided $|\mathbf{H}_n| \to \infty$ as $n \to \infty$. Now because

$$\mathbf{H}_n = \sum_{i=1}^{n} \mathbf{g}(\mathbf{x}_i)\, \mathbf{g}^T(\mathbf{x}_i),$$

then its diagonal elements must be positive and monotonically increase as more measurements are added. We will not go into the mathematical details here but it can be shown that $|\mathbf{H}_n| \to \infty$ if any of its elements do so as well. The trajectories for which this doesn't happen are the ones which will cause us problems.

The sensor trajectories for which the components of $\mathbf{H}_n$ don't diverge are those where $\mathbf{g}(\mathbf{x}_n) \to \mathbf{0}$. As an illustration consider the problem of estimating from depth probes z the slant α and tilt β of a planar surface known to pass through the origin. We can model this with the equation $z = \alpha x + \beta y$, where the sensor is located at $\mathbf{x}^T = (x, y)$, the basis functions are $\mathbf{g}(\mathbf{x}) = \mathbf{x}$, and the model parameters are $\mathbf{m}^T = (\alpha, \beta)$. If the sensor takes a trajectory that approaches the origin then the components of $\mathbf{H}_n$ could converge. For example, if the trajectory is along the straight line $\mathbf{x}_j^T = (x_0\, r^j,\ y_0\, r^j)$, $|r| < 1$ then

$$\mathbf{H}_\infty = \lim_{n\to\infty} \mathbf{H}_n = \mathbf{H}_0 + \frac{r^2}{1-r^2} \begin{pmatrix} x_0^2 & x_0 y_0 \\ x_0 y_0 & y_0^2 \end{pmatrix}. \tag{16}$$

where $\mathbf{H}_0$ is the value of $\mathbf{H}_n$ obtained before the sensor started moving towards the origin. The eigenvalues of $\mathbf{H}_\infty$ are non-zero and finite so $|\mathbf{C}_n|$ will *never* converge to zero in this particular case. Many, but not all, of the trajectories that spiral inward to $\mathbf{g}(\mathbf{x}) = 0$ will behave similarly.

Places where $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ are rather special in that they are extremely non-informative. First, because $\mathbf{g}\mathbf{g}^T = \mathbf{0}$ any measurements taken there do not change the value of $\mathbf{H}_n$. Second, the prediction variance $\sigma_D^2 = \mathbf{g}^T\mathbf{H}^{-1}\mathbf{g}$ is always zero no matter what the value of $\mathbf{H}$. That is we know a priori what the model is at these places (in the example above the surface was known to pass through the origin). There is no point in taking measurements from locations where $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ because they can not contribute anything to our knowledge of the model[5].

Furthermore, for some local neighbourhood around the place where $\sigma_D^2 = 0$, (provided of course that $\mathbf{g}(\mathbf{x})$ is continuous over that neighbourhood) σ_D^2 increases monotonically as one moves *away* from the location where $\sigma_D^2 = 0$. A gaze planning strategy that drives the sensor towards locations of maximum σ_D^2 will avoid places where $\mathbf{g} = \mathbf{0}$, and therefore ensure that $|\mathbf{C}_n|$ converges to zero.

4.2. Gaze planning strategies for linear models. For some linear models the best gaze trajectory can be found analytically. For example if we have a sensor probing the height of planar polyhedral objects then these can be modelled with the basis functions $\mathbf{g}^T = (x, y, 1)$ where x and y are the location of the probe. The prediction variance is easily shown to be parabolic in x and in y, so it grows monotonically, and quadratically, as one moves away from the data. The highest values of $\sigma_D^2(x, y)$ therefore occur on the edges of the objects, and as others have shown [7], the best place to probe for new data is on one of the vertices. The strategy that results in the greatest decrease in $|\mathbf{C}_n|$ at each step is to successively sample from the vertex where σ_D^2 is highest.

Even when analytic solutions are not available it is always possible to find, for the linear case, an optimal gaze trajectory by numeric means. This is because the basis functions $\mathbf{g}(\mathbf{x})$, and therefore the covariances $\mathbf{C}_n$, are totally independent of the model being measured and depend only on the distribution of sensor locations. As no a priori knowledge of the true model in the scene is required, the sensor parameter space can be exhaustively searched for the gaze trajectory that results in the lowest value of an arbitrary cost function (provided, of course, that the cost function doesn't require knowledge of the models in the scene). Such a search could be very costly but need only be done once off-line, and the results simply played back to direct the gaze of the sensor while model inference was taking place.

4.3. Non-linear models. Because of their descriptive power, we use non-linear superellipsoid models to describe objects in the scene. Given that we have a theory of where to take measurements in a linear system we would like to know if, and to what extent, that theory can be usefully applied to non-linear models. Note that

[5] However they would be a good place to measure the sensor noise σ^2.

although we use superellipsoid models, the methodology outlined below is applicable to non-linear models in general.

In our laboratory we sample 3D coordinates from surfaces in the scene with a laser range scanner and, after several layers of "bottom-up" processing, infer those superellipsoid models which best explain the measurements [15]. Points $\{\mathbf{s}_i,\ i = 1,\ldots,n\}$ on the surface of a superellipsoid model with parameters $\mathbf{m}$ satisfy the implicit equation

$$D(\mathbf{s}_i, \mathbf{m}) = D_i(\mathbf{m}) = 0, \tag{17}$$

which is highly non-linear in both $\mathbf{s}_i$ and $\mathbf{m}$. Specific details, e.g. the form of (17), are given in [30]. In fact D has the metric property that it is the radial distance of $\mathbf{s}_i$ from the surface. Therefore when the sensor returns noisy measurements, such that $D(\mathbf{s}_i, \mathbf{m})$ is randomly sampled from a zero-mean normal distribution with variance σ^2, a maximum likelihood estimate of the true parameters can be found by finding those model parameters $\widehat{\mathbf{m}}$ which minimize $\sum_{i=1}^{n} D_i(\mathbf{m})^2$. Because of the non-linearities iterative techniques must be employed.

Once the solution is obtained the covariances are found by linearizing the model around $\widehat{\mathbf{m}}$. It is easy to show that this gives a locally linear model of the same form as (1) but where

$$\mathbf{g}_i = \frac{\partial D}{\partial \mathbf{m}}(\mathbf{s}_i, \widehat{\mathbf{m}}), \tag{18}$$

is the Jacobian of D evaluated for $\mathbf{s}_i$ on the surface of $\widehat{\mathbf{m}}$. By following the linear analysis in §2 the prediction variance at a point on the surface of a superellipsoid model is

$$\sigma_D^2(\mathbf{s}) = \left(\frac{\partial D}{\partial \mathbf{m}}(\mathbf{s}_i, \widehat{\mathbf{m}})\right)^T \hat{\mathbf{C}} \left(\frac{\partial D}{\partial \mathbf{m}}(\mathbf{s}_i, \widehat{\mathbf{m}})\right). \tag{19}$$

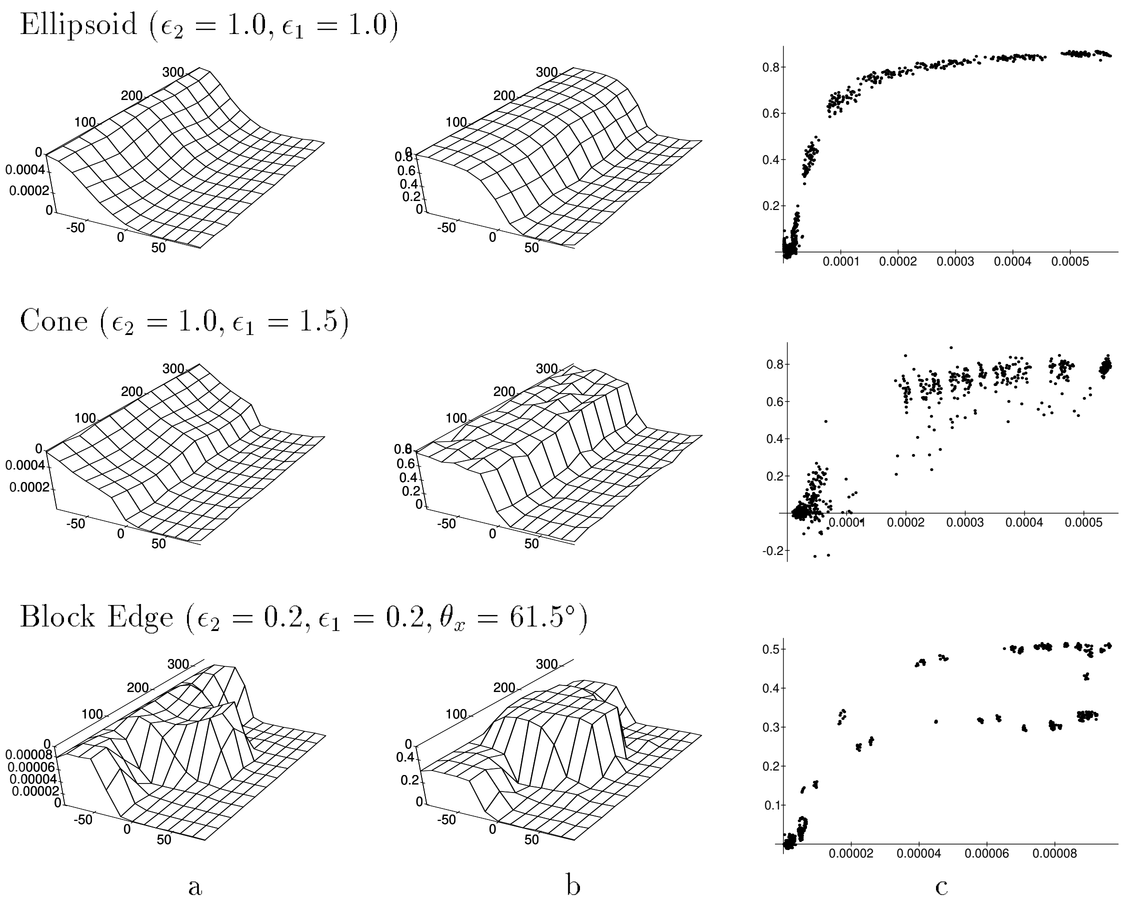
We have previously derived this quantity by other means [30] where we called it the prediction error $\delta_\gamma = \sqrt{\chi_\gamma^2 \sigma_D^2}$.

There is an implicit assumption in the non-linear solution that a posteriori parameter distribution is Gaussian. In general this is not true for two reasons: i) the errors in $D(\mathbf{s}, \mathbf{m})$ will usually not be Gaussian, and ii) the basis functions $\mathbf{g}_i$ exhibit random variation due to the noisy measurements $\mathbf{s}_i$. This variation combines multiplicitatively with the assumed Gaussian randomness in data errors, and results in a distribution that is not normal nor even symmetric. One consequence is that the maximum likelihood solution will in general be biased, so on repeated trials the expected value of $\widehat{\mathbf{m}}$ will differ significantly from the true value $\mathbf{m}_T$. The seriousness of this depends upon the sensor noise level, and on the distribution of sensor locations. For superellipsoid models the fitted surface stays within the envelope of the measurement noise, so provided the sensor noise is low, and that the data obtained constrains the model parameters, any biases due to non-linearities will be small. During the initial stages of exploration the parameters will not be well constrained so estimation bias could cause problems, for example by falsely estimating the position and size of the model. However the whole purpose of exploration is to find data that do constrain the model parameters so ultimately bias will not be a problem.

Another consequence of a non-normal parameter distribution is that the covariances do not fully account for the variation, and are generally an optimistic estimate of the expected parameter errors. This is a problematic when Kalman filter techniques are applied to non-linear models (the extended Kalman filter) because the weighting new data receives is based on how believable it is. If the current estimate of the model parameter variance is too low then the data will appear excessively noisy so it will be given a lower weight than it should be. Ultimately this can result in a situation where less and less importance is attached to new data, and more and more to an estimate which diverges increasingly from the correct value [27]. The exploration strategy we propose differs fundamentally from a Kalman filter in that state updates are based only on the data collected and not at all on any previous state estimate. Thus, although a poor estimate of the covariances might cause us to sample data from a sub-optimal location that data will receive just as much weight when used to update the model parameters, and no weight at all will be given to the previous, erroneous, state. Therefore we do not observe the kind of divergence exhibited by extended Kalman filters.

It is probably not wise to trust the parameter covariances if they are used probability computations, for example when evaluating the risk of a course of action. For these purposes a more general probability distribution should be adopted, such as sums of Gaussians [23], or even grid-based methods [17]. For exploration this could be a problem except, as we shall see later, the strategy we adopt follows the gradient of σ_D^2 (19), so it is not the magnitude of the parameter covariances that is important, but their "direction". Provided we operate within the region of parameter space where the linear approximation is valid then we can expect the gradient to be valid as well.

We know from the linear theory that the best place to take new data is where σ_D^2 is greatest, but it is not clear to what extent this is true in the non-linear case. For superellipsoid models the non-linearities have so far proven intractable, so we used Monte Carlo simulations to test the validity of the linear theory [31]. In these experiments we simulated data acquisition by a range scanner in orbit on a view sphere about a superellipsoid model. For some latitude ϑ and longitude φ on the view sphere we computed the improvement in the ability of the model estimate to predict surface position due to the addition of a single measurement, and correlated this with the amount of prediction error on the surface where the measurement was taken. These trials were repeated a large number of times for scanner positions covering the entire view sphere, and some of the results obtained are shown in Figure 3. In the first two columns the prediction error (a), and improvement (b), are plotted as a function of ϑ and φ where the added datum was collected. It is immediately evident that there is a strong correlation between the two quantities and that the additional



The height of the mesh plots in (a) shows the prediction error as a function of view sphere location. At each location 5 values were averaged to obtain the plotted results. Latitude varies along the short horizontal axis, and longitude along the long one. In (b) is shown the incremental improvement in a model's ability to predict surface position when an extra datum is taken with the sensor at different view sphere locations. Column (c) is a scatter plot that shows the relationship between prediction error (horizontal axis) and improvement for every view sphere location in (a) and (b).

In all cases except the bottom row the model size was kept constant ($a_x = 20mm$,$a_y = 25mm$,$a_z = 30mm$) and posed in an unrotated position with respect to the scene coordinate axes. In the bottom row the model was a little larger ($a_x = 40mm$,$a_y = 50mm$,$a_z = 30mm$) and the base view was chosen so as to sample only a portion of the tilted edge of a block shaped superellipsoid.

For a more detailed description of the experiment see [31].

FIGURE 3. Empirical experiments demonstrating how improvement is a function of prediction error for non-linear superellipsoid models.

data with most benefit is collected from regions on the view sphere which "look" at high prediction errors.

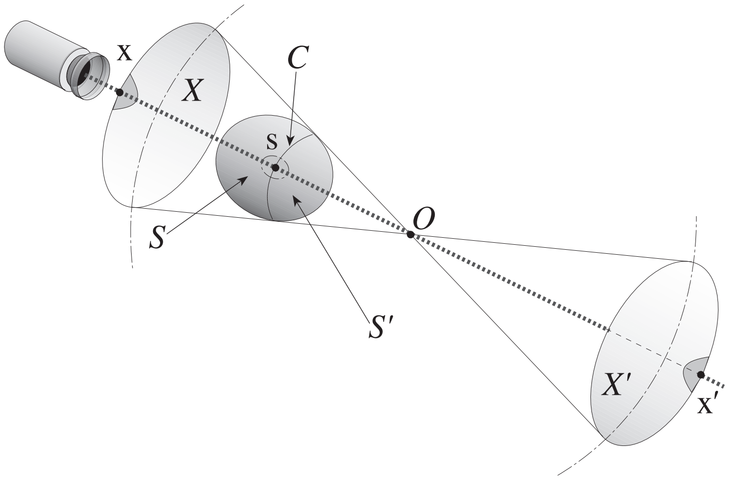
4.4. The gradient strategy – a general non-linear gaze planning strategy. Although the simulations strongly suggest we can apply the linear analysis to non-linear models there is an important and fundamental difference which must be taken into account. As we have seen optimal trajectories for a linear model depended only on the sensor locations, and could be computed without having to know anything at all about the actual model in the scene. This is not true when the model is non-linear as the local linear approximations for $\mathbf{g}_i$, and for $\mathbf{C}$ and σ_D^2, are all dependent upon $\widehat{\mathbf{m}}$. As $\widehat{\mathbf{m}}$ is estimated from the measurements taken of the model in the scene we are placed in the untenable position of having to know the unknown model before an optimal trajectory can be computed. It is not possible to find truly optimal gaze trajectories for non-linear models, so we must soften our expectations.

In view of the the dependency of σ_D^2 upon $\widehat{\mathbf{m}}$ we adopt the following general iterative strategy. At each step n of the gaze trajectory we compute $(\sigma_D^2)_n = \sigma_D^2(\mathbf{x}, \widehat{\mathbf{m}}_n)$ using the current estimate of the model $\widehat{\mathbf{m}}_n$. The next sensor trajectory location $\mathbf{x}_{n+1}$ is chosen to be that which maximizes $(\sigma_D^2)_n$, but subject to the constraint that it lie within the region of sensor locations for which the linear approximation is valid. Once $\mathbf{x}_{n+1}$ is found the sensor is moved there, and an additional measurement taken. The model estimate is then updated by re-fitting to a data set in which the new measurement is appended to all of those obtained previously. The process repeats using the updated estimate $\widehat{\mathbf{m}}_{n+1}$, and runs until the parameter covariances meet some operationally defined criteria of acceptability.

A problem with the approach is that it is difficult to determine the region of sensor locations over which the non-linear relationships of the model can be approximated sufficiently well by the linearized form (18). What we suspect, and what our observations suggest, is that the region is indicated by low values of $(\sigma_D^2)_n$, but it is not clear how to quantify the linear approximation error in σ_D^2. In fact the size of the approximation error is not important provided the monotonic relationship between σ_D^2 and the decrease in $|\mathbf{C}_n|$ is preserved. Again we suspect that this relationship remains true even when the approximation errors are large, and observations indicate that the region of allowable sensor locations can include regions of large σ_D^2 without any noticeable degradation in performance.

However because of worries about the validity of the linearized theory, we took a conservative approach in which the sensor was always moved in small steps. Our rationale was that in the region around the current sensor location the position of the surface would be well known due to the measurement already made there, so locally the difference between the true and estimated models would be small, and the linearized form would serve as an adequate approximation. Thus instead of globally maximizing (14) we do it in the local neighbourhood of the current location, and move the sensor in the direction of maximum of σ_D^2. In effect the scanner follows the gradient $\partial\sigma_D^2/\partial\mathbf{x}$ (or an approximation of it), so we will refer to this as the *gradient strategy*.

At first glance the gradient strategy might seem to be an implementation of the classical gradient ascent method but there is an important difference. This is that the form $\sigma_D^2(\mathbf{x})$ being ascended is continuously changing as new data are added. When a measurement is taken from a location where $\sigma_D^2(\mathbf{x})$ is high the prediction variance there will be reduced to the level of sensor noise. The effect of this reduction is to "push" the sensor away from the current location on the next iteration and thus to help overcome a major problem of gradient methods — that the sensor trajectory will falter on top of a local maximum.

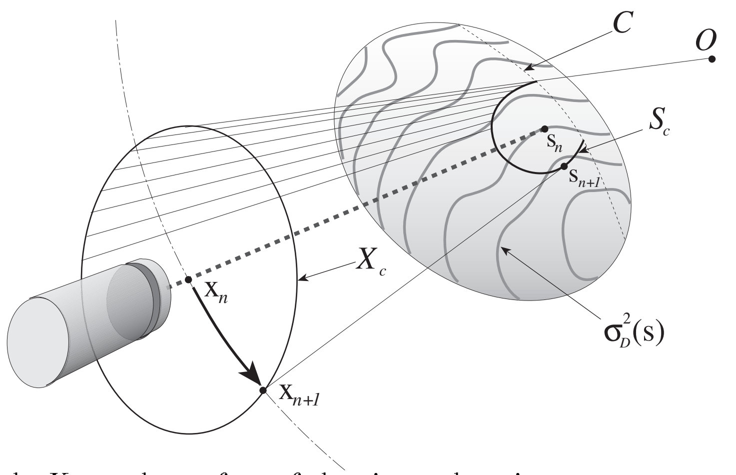


When the model does not enclose the view sphere center O there will be a closed contour C where the scanner beam is tangential to the models surface. The contour divides the model into two regions S and S' which map into the disjoint view sphere regions X and X' respectively. Thus the point $\mathbf{s}$ on the contour can be "seen" by the scanner when it is positioned at either $\mathbf{x}$ or $\mathbf{x}'$, and a small movement across the contour will require a discontinuous jump in view sphere position between $\mathbf{x}$ and $\mathbf{x}'$.

FIGURE 4. The mapping between $\mathbf{x}$ and $\mathbf{s}$ is discontinuous

4.5. The view sphere gradient strategy. At this point our gaze planning strategy is quite general in that it simply tells us to keep moving the scanner in a direction which locally maximizes the prediction variance. Here we present an implementation in which a laser range scanner is able to orbit about a single superellipsoidal object. The location of the scanner is given by its position on the surface of a view sphere, and is directed so it always points towards the sphere's center. When describing the view sphere location we will usually refer to its radius ϱ, latitude ϑ and longitude φ; but because this parametrization has singularities at the poles, a better way to represent sensor location $\mathbf{x}$ is by a 3D vector from the center of the view sphere, to the position of the sensor on the sphere's surface.

The first problem is to compute the gradient $\partial\sigma_D^2/\partial\mathbf{x}$ of the prediction variance. However given the expression we have in (19), σ_D^2 is a function of the position $\mathbf{s}$ on the surface of the estimated model, and not of the sensor location $\mathbf{x}$. The relationship between these quantities $\mathbf{s} = \mathbf{s}(\mathbf{x}, \widehat{\mathbf{m}})$ is essentially a ray tracing problem. That is, for a given scanner gaze we need to compute where the laser beam will hit the surface on the estimated model. Provided the form of $\mathbf{s}(\mathbf{x}, \widehat{\mathbf{m}})$ is known and is continuous around $\mathbf{x}$ then it is in principal an easy matter to obtain the gradient analytically. Unfortunately we fail on both counts as $\mathbf{s}(\mathbf{x}, \widehat{\mathbf{m}})$ is not always continuous (Figure 4), and a closed form solution to the ray tracing problem is unknown for superellipsoid models.



The search circle X_c on the surface of the view sphere is at a constant geodesic radius from the current scanner location $\mathbf{x}_n$, and its image S_c is formed where the cone of beams directed towards the center of the view sphere O intersect the surface of the model. Here we show a case where some of the beams emanating from X_c miss the surface. The next scanner location $\mathbf{x}_{n+1}$ is the one which projects to the surface coordinate $\mathbf{s}_{n+1}$ that maximizes $\sigma_D^2(\mathbf{s})$ on S_c. The gradient $\partial\sigma_D^2/\partial\mathbf{x}$ is approximated by the direction of the arrow from $\mathbf{x}_n$ to $\mathbf{x}_{n+1}$.

FIGURE 5. The numerical solution to $\partial\sigma_D^2/\partial\mathbf{x}$

Because there is no general analytic solution we employ the numerical technique illustrated in Figure 5. Essentially we search for the maximum σ_D^2 on the view sphere at a fixed geodesic distance from the current scanner location. The scanner is circled around what we call the *search circle* thus generating a cone of laser beams directed towards the center of the view sphere. We use Newton's method to find where each beam strikes the surface of the estimated model, and if it does we compute σ_D^2 at that spot. The direction of sensor travel is chosen to be towards the circle position which resulted in the maximum value of σ_D^2.

The conic angle θ_c which determines the radius of the search circle is set equal to the distance that the scanner travels with each iteration of the gaze trajectory. If θ_c

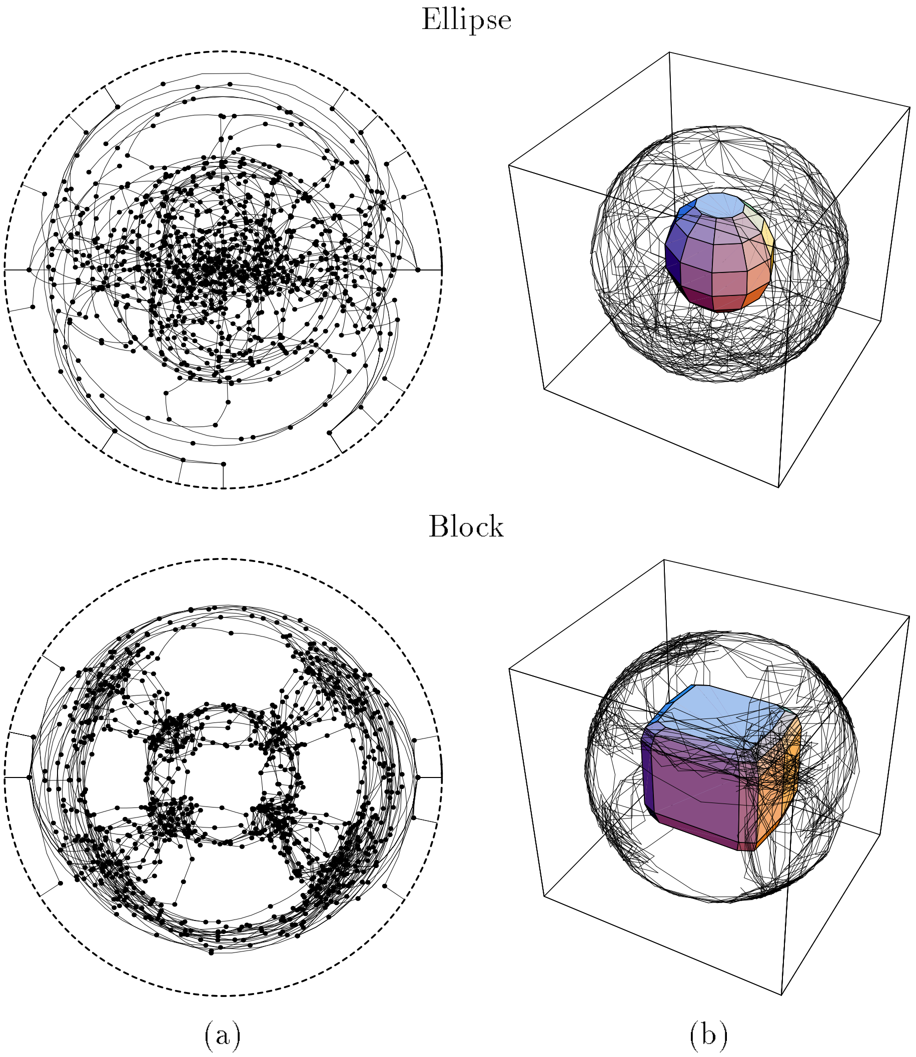
is too small convergence of $|\mathbf{C}_n|$ will be slow, but if too large it might result in the scanner missing important features on the surface. This suggests the step distance be set according to the scale of the models in the scene but such an approach is possible only if something about them is known beforehand. In the experiments that follow we do make use of what we know about the scene and set the scanner to travel along a great circle arc which subtends and angle of $20°$ at each iteration. In the more general situation this parameter will have to adapt to the unknown models as they are inferred.

We investigated the behaviour of the above strategy by running simulations on various superellipsoid models positioned in the center of a view sphere. After initially fitting a model to data scanned from above ($\vartheta = 90°, \varphi = 0°$) the gradient of σ_D^2 on the view sphere was evaluated numerically as shown in Figure 5, and the scanner location then stepped along a great circle arc of $20°$ towards the maximum. At the new view sphere location a noisy measurement was taken, added to existing data, and to this a new model was fit. The process was repeated for 100 iterations.

The view sphere trajectories from 40 separate explorations of an ellipse and a block are shown in Figure 6. Because the measurements are noisy, each individual path is subject to a certain amount of random wandering which obscures any structure in its location. By superimposing a large number the path density gives us some idea of those locations which attract the "attention" of the algorithm. In the case of the ellipse it can be seen that the path density is greatest at the pole opposite the initial data collection ($\vartheta = -90°, \varphi = 0°$). This matches the strategy most people take when asked to resolve the ambiguity of a single view, that is to look at the other side.

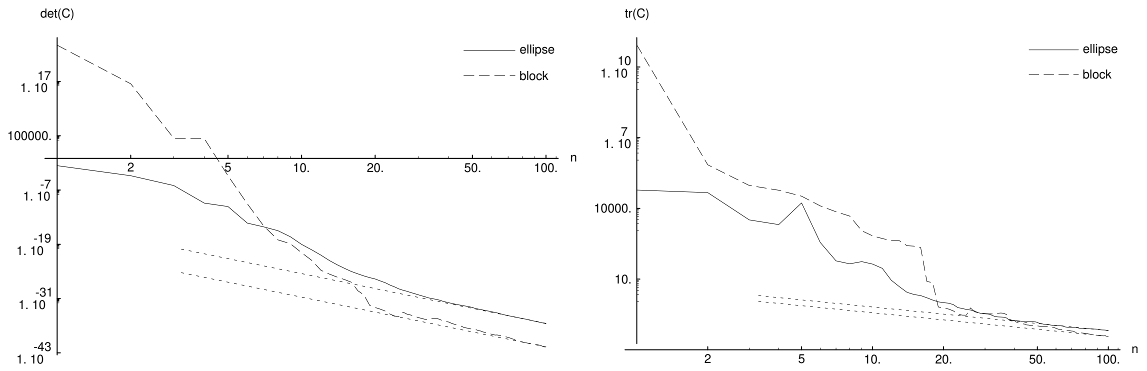
What is interesting however is the anisotropy of the path locations. There is definitely a greater density along the meridians $\varphi = 0°$ and $\varphi = 180°$ and a close examination of the corresponding 3D rendition of the paths shows that the scanner spends more time exploring the narrower, more highly curved surfaces of the ellipsoid. The attraction of the scanner to places of high curvature is demonstrated graphically by the exploration paths of the block where it is obvious that scanner spends most of its time collecting data from the edges and the corners. An explanation for this phenomenon, at least in the case of the block, can be found in the example at the beginning of §4.2. Here we pointed out that the linear theory predicted the edges of planar polyhedra objects as being the places to get the most information about the model parameters. Because the "faces" of the superellipsoidal block are approaching planarity we would expect the edges of the block to be the best places as well.

4.6. The performance of the view sphere gradient strategy. While the results above are intuitively appealing, the real test of the algorithm is how rapidly it refines the estimates of the model's parameters, and if fares better in this respect than other approaches. To evaluate these aspects of the strategy's performance, we used the same Monte-Carlo simulation techniques that generated the trajectories shown in Figure 6.



The view sphere trajectories of all the explorations are displayed in two forms.
(a) View sphere coordinates are shown in polar form, as though the view sphere were seen from the underneath. The view sphere latitude is plotted radially with $\vartheta = -90°$ in the center and $\vartheta = 90°$ at the outer dashed circle. Longitude is the angular coordinate with $\varphi = 0$ being horizontal and to the right. The initial data was scanned from $\vartheta = 90°$ so the small radial lines show the initial $20°$ step. The dots mark the places from which additional data was scanned and the lines connecting them the great circle paths taken by the scanner.
(b) The 3D positions of the paths in (a) are shown on the surface of a transparent view sphere. The model being explored is rendered in the center.

FIGURE 6. 40 explorations of an ellipse and a block.



a b

FIGURE 7. Exploration simulations showing the convergence of a) $|\mathbf{C}_n|$ and b) $\operatorname{tr}(\mathbf{C}_n)$. The dotted asymptotes indicate the rate of decrease expected due to repeated measurements.

Some typical results showing the decrease in $|\mathbf{C}_n|$ are presented in Figure 7a. The first point to note is that because the models are non-linear, the decrease isn't strictly monotonic as predicted by the linear theory. However the upward swings are minor in comparison to the general trend, so it does appear that the linear approximation holds quite well.

Although the explorer works by minimizing the determinant, the simulations indicate that it also has the effect of decreasing the trace as well (Figure 7b). There are situations where this might not be the case, for example in a linear regression, sampling data continuously at the origin will decrease the variance of the intercept yet contribute nothing to our knowledge of the slope. However we believe that in most situations the exploration algorithm will ensure that this doesn't happen. It may be that at any moment the best strategy is to decrease a subset of the covariance eigenvalues, but it can be shown that the decrease in each eigenvalue is asymptotic to $1/n$. That is, the improvement at each step becomes vanishingly small, so at some stage we would expect there to be an optimal sensor location that results in the decrease of some other eigenvalue. Ultimately we expect there will be no preference for any one subset of eigenvalues, and they will all end up vanishing.

However there will be some conditions that result in a sensor trajectory which does not minimize all of the eigenvalues. In general this can happen where the mapping from the space of sensor locations given by the parametric model of the system does not give full access to the space of model parameters. As we have already demonstrated in the linear regression example above, it can occur when there are accessibility constraints on the sensor position, though what usually happens is that sensor movement is not so restricted and the available locations do provide some information about all the parameters. As a result all the eigenvalues converge, but some converge very slowly. A case that concerns us very much is where some of the model parameters are meaningless. For example, no matter how much data is taken, the variances will remain very high for the rotational parameters of a spherical superellipsoid. We would like to detect this case and use a more appropriate parametrization, but before we can we must make sure that the high variances really are because the model is degenerate, and not because exploration is incomplete. Minimizing the determinant is useful in this regard because it will still work to reduce those eigenvalues that can be reduced, despite the presence of large eigenvalues for which no improvement is possible.

The question of when to terminate exploration can be resolved based on the observation that there are two phases to the convergence. Initially $|\mathbf{C}_n|$ decreases rapidly: approximately proportional to $n^{-2.3p}$ for the ellipse and $n^{-4.3p}$ for the block; but is soon followed by the final phase where, as shown by the dotted lines, $|\mathbf{C}_n| \propto n^{-p}$. This convergence pattern is typical of what we observe, not only in simulations, but also for the implementation operating in our laboratory.

The significance of the asymptotic approach to n^{-p} is that this is the rate of decrease expected due to a reduction in sensor noise because of repeated measurements, and it indicates that the final stage of convergence is due to the scanner repeatedly re-measuring the model's surface. The picture we have of the strategy's behaviour is that during the initial stage the scanner is attracted to unknown parts of the model's surface and as a result the new measurements reveal new structural information about the model. However once the surface has been "covered" with measurements the basic structure of the model is known, further scans repeat earlier ones, and the decrease in $|\mathbf{C}_n|$ slows to the "background" rate due to repeated measurements. If we want to evaluate how well the strategy does, it is important to see how quickly the algorithm can gather the structural information. For this we will develop the notion of coverage as a way of determining when the first stage of convergence has completed.

A good indicator of coverage is the distribution of prediction variance on the surface of the model. When a scanner measurement is made it allows us to predict surface position there to the same level of accuracy as the sensor noise. Furthermore, the ability of the model to interpolate between measurements can result in a reduction of σ_D^2 over sizeable regions, and not just in the local vicinity of the measurement. Taking more measurements from within that region will not contribute greatly to our knowledge of it, though as pointed out above, it will reduce σ_D^2 due the effect of repeated data. The coverage of the surface is indicated by the area over which $\sigma_D^2/\sigma^2 \leq 1$, and leads us to adopt the following definition.

> We say that the surface S of the model $\widehat{\mathbf{m}}$ is *covered* by measurements when $\sigma_D^2(\mathbf{s})/\sigma^2 \leq 1$ for all $\mathbf{s}$ on S. In other words a surface is covered when it can be predicted everywhere to the same accuracy that it can be measured. We shall designate this condition by the relation $\lceil_S \sigma_D^2(\mathbf{s})/\sigma^2 \leq 1$.

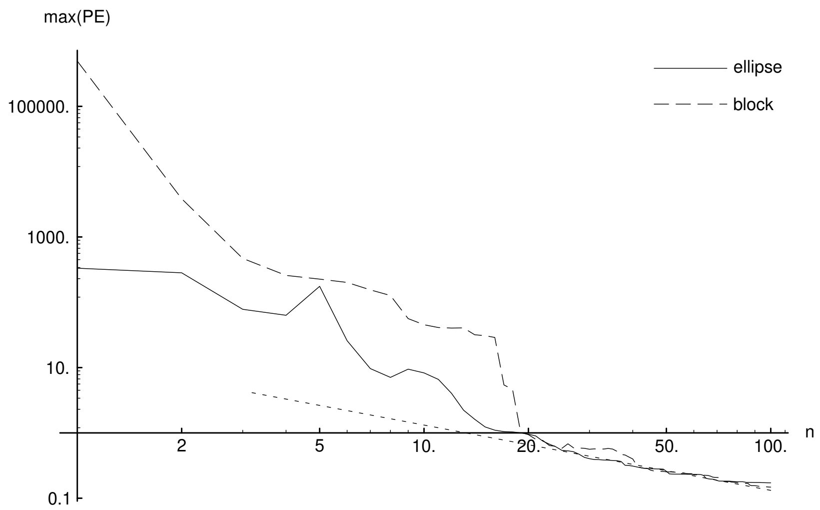


FIGURE 8. Exploration simulations showing the convergence of the maximum value of σ_D^2/σ^2 on the surface of the estimated model.

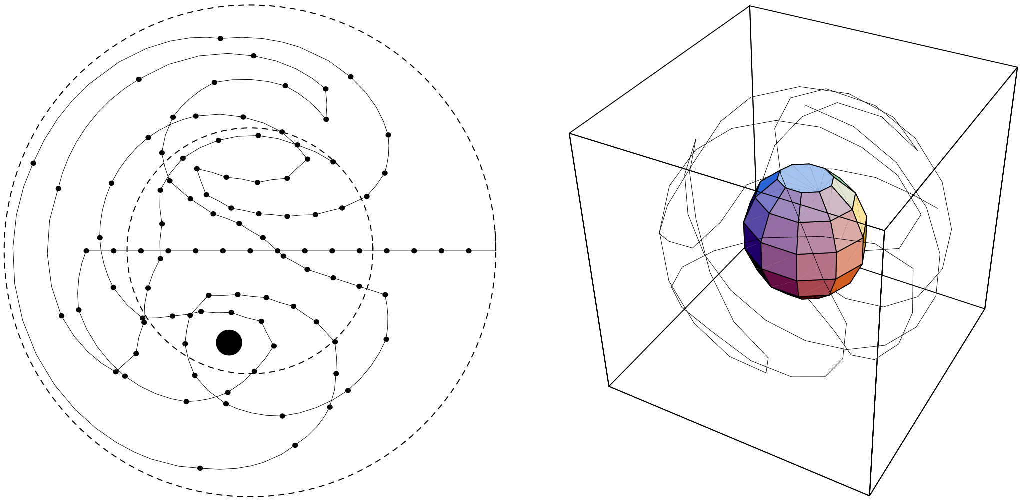
In Figure 8 we show the convergence of $\lceil_S \sigma_D^2(\mathbf{s})/\sigma^2$ for exactly the same gaze trajectories as in Figure 7. As can be seen its value falls below 1.0 at around 20 iterations for both models, and that this is also where $|\mathbf{C}_n|$ becomes proportional to n^{-p}.

The number of iterations taken to cover the surface gives us a useful and intuitive way to compare the performance of different strategies. In particular we would like to compare gradient strategy with others that make minimal use of the model estimate. One of these is that the sensor should be positioned to collect data from "the other side" of 3D objects but there are a couple of problems with it. The first is that the location of "the other side" is a function of the model's pose and, to some extent, its shape. Because these are exactly the quantities which are uncertain, it is not clear how one can find the other side when its location isn't known very well. The other problem is that for some shapes, e.g. for block-like objects, there may be several "other sides", and there is no obvious solution as to which of these it might be best to measure.

Because of these difficulties we adopt a related approach as the basis for comparison. In what we call the *avoidance strategy* the sensor always moves away from the places it has visited previously. We do this by finding the position $\mathbf{x}_k$ on the search circle which minimizes the potential

$$p_c = \sum_{i=1}^{n} \frac{1}{1 + \lambda_p \left(|\mathbf{x}_c - \mathbf{x}_i| / (\varrho\, \theta_c) \right)^2}, \tag{20}$$

where the $\mathbf{x}_i$ are the n sensor locations of the current gaze trajectory, and their distances from the search circle $|\mathbf{x}_c - \mathbf{x}_i|$ are normalized with respect to the distance that the scanner steps, $\varrho\, \theta_c$, at each iteration. The value of the free parameter λ_p



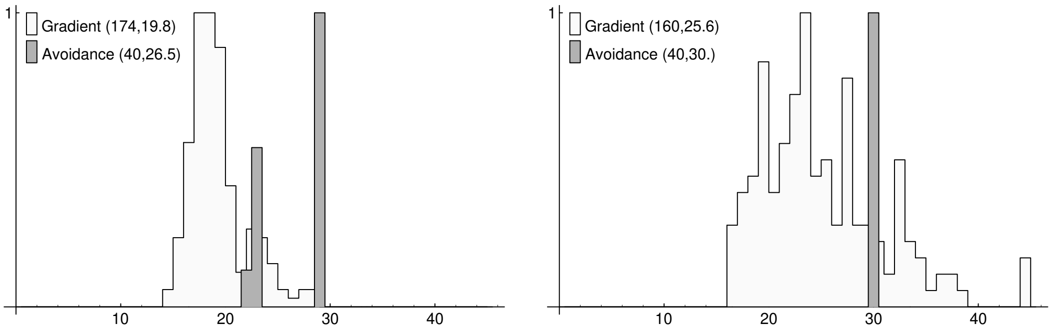
a   b

FIGURE 9. The avoidance strategy gaze trajectory ($\lambda_p = 2.0$). The dot in (a) shows the position of the largest "hole" after 100 iterations.

affects the scanner path and can be used to tune the performance of the strategy to specific cases. For example in the experiments that follow we shall use the trajectory shown in Figure 9. It was computed for $\lambda_p = 2.0$, the value which minimized the number of iterations taken to cover the ellipsoidal surface shown in the Figure 9b.

It should be noted that the specific form of (20) is not important, and in fact there are many ways to achieve essentially the same result. What is important is that the avoidance strategy requires absolutely no knowledge of the model parameters or their uncertainty, so it serves as a useful baseline. In particular, a comparison between it and the gradient strategy tells us how much better we can do if the model uncertainty is taken into account when planning sensor trajectories.

The results of Monte Carlo simulations used to compare the performance of the gradient and avoidance strategies are presented in Figure 10. From the histogram for the ellipse we see that the gradient strategy completed in an average of 19.8 iterations, which for a step size of 20° represents only 1.1 circumnavigations of the view sphere. In general the gradient strategy fares better. It takes the avoidance strategy 35% longer to cover the ellipse, and 17% longer to cover the block. However because the gradient trajectory is affected by random variations in the model estimate it sometimes fares worse. For example approximately 22% of the trials took longer than the 30 iterations required by the avoidance strategy to cover the block. The insensitivity of the avoidance strategy to such probabilistic uncertainty, and its consequent ability to complete predictably within 30 iterations, lends the strategy a certain attractiveness. However it must be remembered that the performance of the avoidance strategy was optimized for the ellipsoid model used in these experiments,

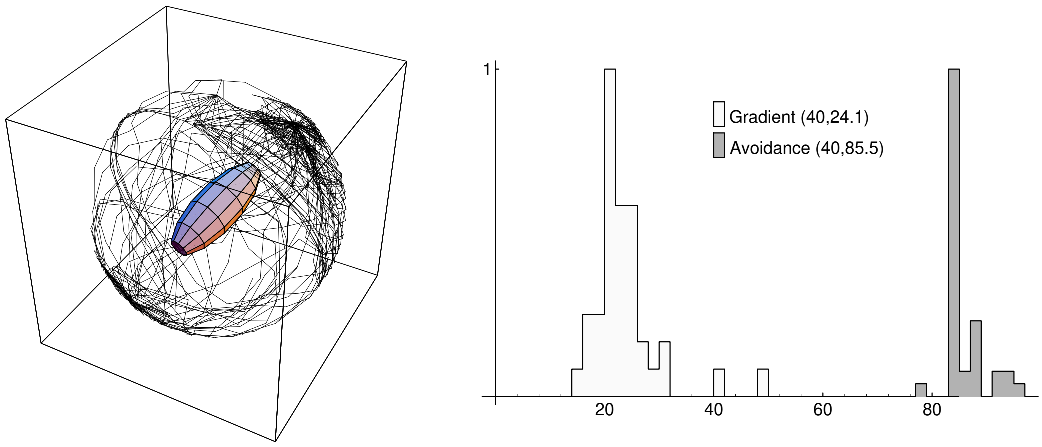


ellipse block

Comparison of gradient and avoidance strategies. The histograms show the relative frequencies of the number of iterations taken to cover the surface of the model, i.e. the iteration at which $\int_S \sigma_D^2(\mathbf{s})/\sigma^2 \leq 1.0$. In the legend labels the first number is the total number of trials used to generate the histograms, and the second is the average number of iterations taken to cover the surface.

FIGURE 10. Comparison of gradient and avoidance strategies

and the results therefore present that strategy in its best possible light. The avoidance strategy performs well on the block model too, but it is in an identical pose to the ellipse and is of similar size. As we shall see below this picture can change dramatically when size and pose are changed.



a b

FIGURE 11. The gradient strategy can adapt to model pose.

The true strength of the gradient strategy is its adaptability. Figure 11 illustrates the severe degradation in the performance of the avoidance strategy when model pose and size are changed. Here we have constructed a "cigar-shaped" ellipsoid model, oriented so its long axis points directly at the largest hole in the coverage given by the avoidance strategy (see Figure 9a). We see in Figure 11a that the gradient strategy has adapted to the change and still manages to measure the narrow ends of the object even though its pose is initially unknown. From the histogram in Figure 11b the average number of iterations required to cover the surface has increased moderately by 25% from 19.8 to 24.1. In contrast the avoidance strategy cannot adapt as it blindly pursues its predetermined path, and misses seeing the narrow end of the object until much later. Consequently the number of iterations taken to cover the surface rises from 26.5 to 85.5, an increase of over 320%.

4.7. Real-world complications. Unlike the simulations of the previous sections, there are further complications in the real world. One problem has its roots in the discontinuous nature of the mapping between sensor locations and surface points shown in Figure 4. In the simulations this was not a problem because the model's surface enclosed the center of the view sphere. However in a real scene that might not be true, in which case the scanner will travel downward until it reaches the edge of the region where its beam is tangential to the model's surface (e.g. at $\mathbf{x}$ in Figure 4). Once there some of the beams emanating from the search circle (Figure 5) will fail to intersect the surface, so the direction of maximum σ_D^2 will have to be chosen from the subset that do. As a result the scanner will be unable to leave the upper view sphere region and will only be able to collect data from the top portion of the model's surface. The convergence of $|\mathbf{C}_n|$ will be much slower (proportional to n^{-p}) than if the sensor were free to make measurements of the lower portion of the surface.

In the current implementation we solve this problem by always repositioning the view sphere so that its center is at the center of the current model estimate. However we are also investigating another approach based on the observation that the mapping $\mathbf{x} \mapsto \mathbf{s}$ is a projection of the $\sigma_D^2(\mathbf{s})$ field from the surface of the model to the surface of the view sphere. In the gradient strategy this projection happens to be discontinuous but there are many others which are not, for example if we project the field radially from the model's center. We have obtained some encouraging preliminary results with model centered projections that adapt to increase the camera step size as σ_D^2/σ^2 decreases.

Problems also arise because the next scanner position is based upon an estimate of where the model's surface should be. In the early iterations there is usually a large amount of error and it is possible to move the scanner to a location where the beam would intersect the estimated model's surface, but from which no measurement of the true surface is possible. By keeping the step size small we reduce, but do not eliminate, the chance of this happening. However a better approach is to adjust the scanner travel so it never moves into regions where σ_D^2 is very large. This is exactly the behaviour that some of the adaptive projections mentioned in the previous paragraph exhibit and we are optimistic that we can use them to reduce the frequency of this problem.

A difficult problem is that of accessibility. It is rarely the case that there is complete freedom to sample any surface in the scene. Real objects are supported by, embedded within, or occluded by other objects. Real scanners cannot move everywhere. Their physical size prevents them passing along narrow passages, and the device that moves them, e.g. a robot arm, has its own limitations. It will simply not be possible to sample the most uncertain surface in some, and probably most, cases. The successful resolution of these problems would require task specific knowledge, for example details of the dimensions of the scanner, the robot, and the workspace configuration. A basic tenant of our design is that it be modular and general, so we should not have to customize the gaze planning strategy for each application. We consider the resolution of accessibility problems to be beyond the domain of the gaze planning strategy.

What is important is that the strategy should operate in a way which makes it possible to handle external problems like lack of accessibility. In this regard our requirement that the sensor only move incrementally is a prudent course of action. If a problem occurs the sensor will still be close to its last position, and it will be easy to backtrack and recover. There are also practical reasons to prefer small, incremental, movements. When the scanner is mounted on a robot arm it is much simpler to compute a path (and to implement collision avoidance) over a short distance than it is to traverse from one side of the work space to the other. From our point of view another important requirement is the need to convert scanner centered 3D coordinates into the scene coordinates used to fit models [28] (§6). Because the position of the sensor is uncertain we can only do this by the registration of overlapping data sets, and smaller movements let us to keep the area scanned small whilst still maintaining a useful overlap.

Although we can alleviate many of the above problems there is always a chance that they will occur, and that task specific knowledge will be required to take corrective action. For this we need a higher authority. The analogy we use is that the gaze planning strategy plays the role of a *navigator*, and has the specific task of determining what the best heading is at each iteration. The navigator operates under the command of a more general module which we call the *explorer*, and it is this which has the task specific knowledge needed to the verify the operational feasibility of the heading, and to detect and correct the kinds of problems which arise when the heading is followed.

5. Exploring

Higher level tasks, i.e. those that make use of the volumetric models we infer, attempt to deal the non-uniqueness (or uncertainty) by bringing in a priori constraints appropriate to their particular area of expertise. However it is often the case that these constraints fall short, and the models are still too ambiguous to be useful. This was the motivation for the gaze planning strategy outlined in §4. Higher level tasks could invoke that strategy directly by taking control of the lower levels of processing, but we would prefer to decouple them and maintain a more modular approach.

For this role we envisage an autonomous agent which we call the explorer. The explorer takes, from any higher level task, a specification as to the allowable amount of uncertainty, then proceeds to collect data until that specification is met. Once complete the explorer reports back with a new set of inferred models.

At the heart of the explorer is a servo loop built around the gaze planning strategy. As input it takes the uncertain state of some model in the scene (parameters and covariances), and delivers at its output the direction in which the sensor should move to maximally decrease that uncertainty. The loop is closed by updating the input state with additional data scanned from the new sensor location.

The explorer is more than a simple servo loop however. It is an executive that delegates jobs, monitors progress, and makes decisions. The decision as to when to stop is largely determined by the application task requesting the models. There are complications because the specification is application dependent, for example an object recognition system will want a better knowledge of specific model parameters in order that it may disambiguate two models stored in its data base, and a vehicle might be more interested in the surface location along its proposed path. In principal however these criteria are equivalent but different mappings of the parameter covariances.

Handling accessibility constraints should also be dealt with by the explorer. The incremental approach to gaze control helps somewhat but inevitably an exploration path will be forced to terminate because of occlusion or because of the inability of the robot moving the scanner to access the correct location. In these cases the explorer must take control, override the servo loop, and reinitialize the scanner at a new location.

The gaze planning strategy is also very single minded – it only works upon a single model. When there are multiple models in the scene it is the explorer which must decide the focus of attention. The strategy employed will depend to a large extent on which models the application task is currently interested in.

A very important task for the explorer is to monitor the behaviour of the lower levels of processing. None of these can be guaranteed to operate flawlessly, for reasons ranging from excessively noisy data, to situations in the scene that break the assumptions upon which the algorithms are based. A general "catch-all" way of determining if something has gone wrong is to measure the amount of misfit between the inferred models and the data, i.e. to examine the residual errors. When there is misfit the residual errors will significantly exceed the expected sensor noise, the servo loop can be aborted, diagnostics run to isolate the source of the error, and remedial action taken to fix the problem where possible. The problem with this is that we must have good models of sensor noise that we may compare statistically with the residual errors. We have investigated this and have found that the classic statistical methods for detecting misfit can be quite sensitive to departures from the theoretical model. To this end we have devised a sequential estimator of sensor noise which operates within the servo loop. We have shown that this estimator reliably detects misfit even when the sensor noise is not that of the theory, and when the noise level varies during the course of exploration [32].

6. Implementation of an Autonomous Explorer

The concepts of the previous sections are now used to implement an autonomous explorer capable of building an articulated volumetric description of its environment through a sequence of exploratory probes. Figure 12 shows a block diagram of the resulting implementation. The left side corresponds to a classical model of bottom-up vision in which sensor data are transformed into various levels of representation through successive stages of processing [15]. In our implementation data are acquired through a laser range-finding system mounted on the end-effector of an inverted PUMA 560 robot as shown in Figure 13. The system has a field of view of approximately $1m^3$ which can be positioned anywhere in the robot workspace. Because of the relatively low positioning accuracy of the robot (on the order of $1.0cm$), the transformation parameters relating different viewpoints must be computed from the acquired data.

To facilitate estimation of these parameters and to provide the necessary structure from which to perform shape analysis, a visual reconstruction procedure is used to turn the discrete sampled data from the rangefinder into a piecewise-smooth (C^2) representation of the surfaces in the scene [16]. From there data acquired from different vantage points can be fused by determining the correspondence between features in adjacent views. A temporal extension to our reconstruction procedure [28] is used to determine the transformation parameters on a local basis. An important feature of this algorithm is its ability to deal with non-rigid motions. Such might be the case, for example, when dealing with objects that can change configuration in between changes of sensor position.

At the next higher level of abstraction, reconstructed surface information from multiple viewpoints is used to determine surface boundaries corresponding to the parts of an object. The perceptual basis of the algorithm is the Hoffman and Richards [18] principle of transversality regularity. Objects in the scene are represented as conjunctions of convex solids. Boundaries between parts of objects thus correspond to concave discontinuities and/or negative local minima in the principle curvatures of the surface [15]. These features are made explicit as a by-product of the reconstruction procedure. However, the task of interpolating such features in to part boundaries is non-trivial and the subject of much work in the literature, e.g. the work of Kimia et al. [20]. The procedure we use to solve this problem is a special case of the more general model described by Kimia and adapted for surfaces [22].

Finally, at the highest level of abstraction, surface regions defined by part boundaries are described by parametric forms (i.e. models) such as superquadrics [5]. In addition to serving as a basis for the characterization of uncertainty, these descriptors provide additional cues for maintaining correspondence at the level of parts, for describing general shape properties, and for recognition [15].

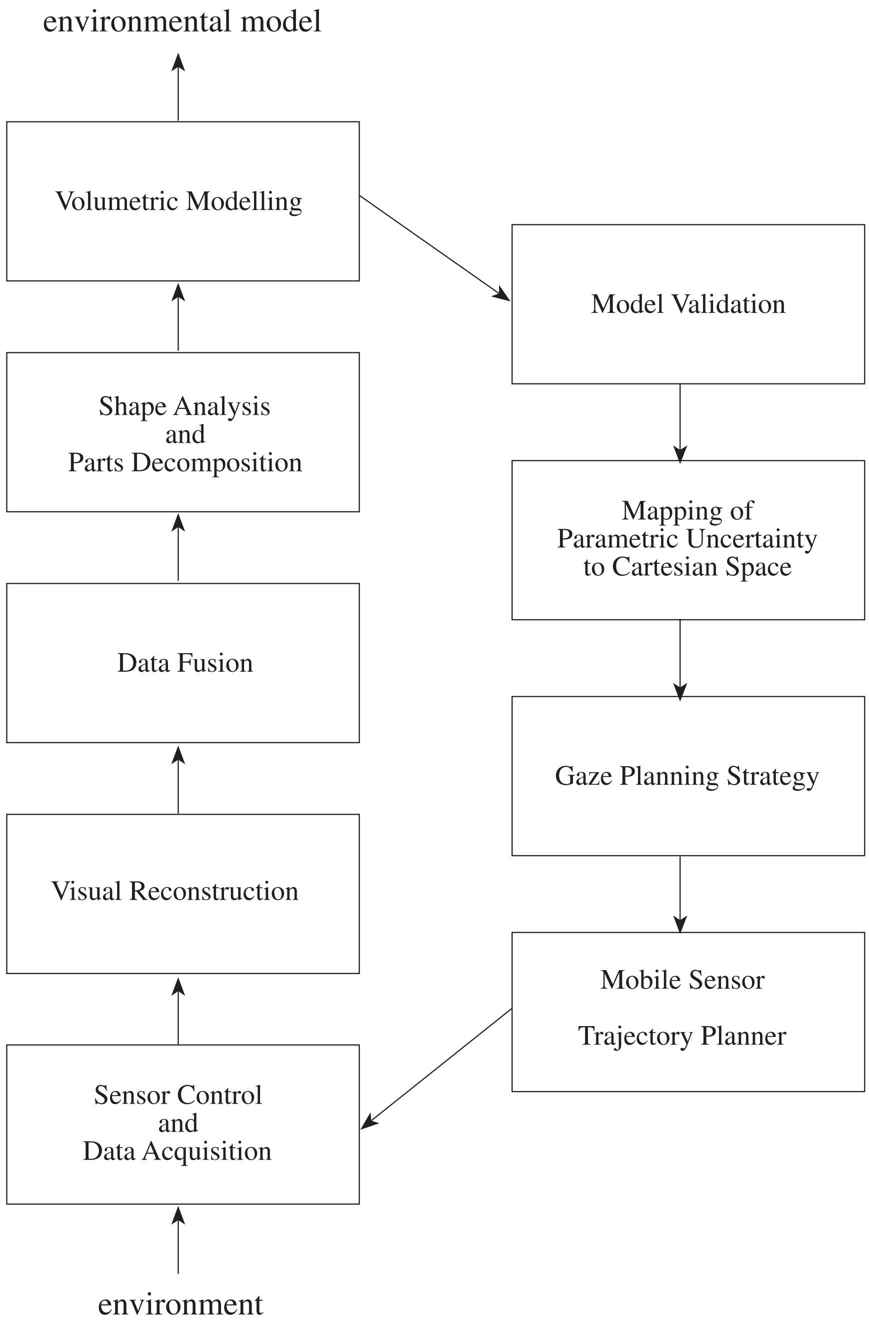


FIGURE 12. Process flow in the autonomous explorer. The left hand side of the figure corresponds to a classical bottom-up vision strategy. The right hand side corresponds to feedback derived from parametric uncertainty which is used to close the loop around bottom-up perception.

FIGURE 13. The mobile sensor consists of a laser range-finding system with a $1m^3$ field of view mounted on the end-effector of an inverted PUMA 560 robot.

6.1. Closing the Loop. Much as in the same way that feedback serves to reduce plant uncertainty in a conventional control system, the autonomous explorer uses feedback to minimize the uncertainty of the parametric models used to describe the scene. The right hand side of Figure 12 shows how this feedback is implemented. A fundamental assumption implicit in the strategy is the validity of the models used as reference, i.e., that a particular model is competent to describe its data in the first place. The sequential estimator of sensor noise discussed earlier in §5 is used for this purpose. At the present time no attempt is made to backtrack in the event that a model is deemed invalid; the exploration process for the model in question is simply re-initialized. The default strategy is simply to throw away the data corresponding to the model in question and begin again with the data in the current viewpoint. This strategy, while not optimal, works quite well in practice as failures are more often a result of segmentation errors where a single model may overlap two or more parts.

Gaze planning is based on the view sphere gradient strategy developed for a single model in §4.5. However, the problem becomes much more complicated when an object comprised of many parts (models) must be explored by a single mobile sensor. Different uncertainty surfaces give rise to different viewpoint requirements which often cannot be met within the field of view of a single sensor in any position.

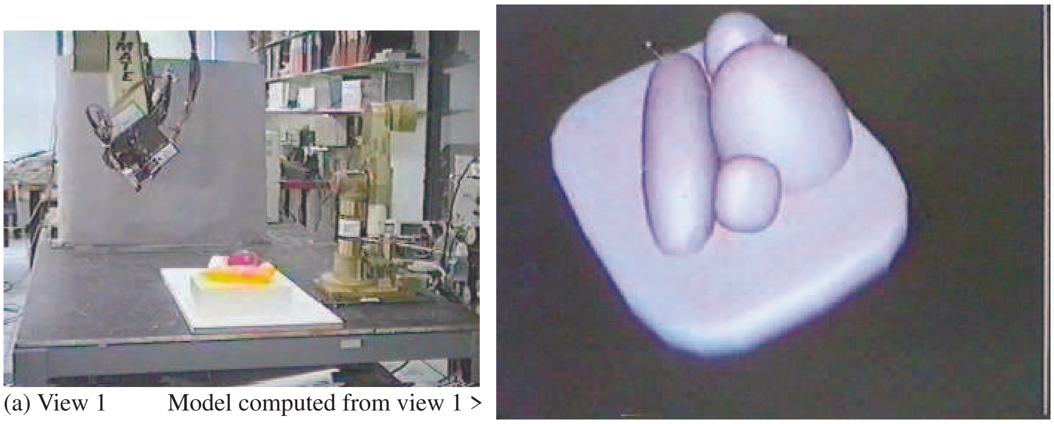
Our initial solution, the one reported in this paper, was to apply a focus of attention mechanism to break this dilemma. At each gaze planning iteration all models are examined and valid ones ranked according to size, magnitude of the uncertainty gradient and distance to the sensor. The "winner" gains control of the sensor for that iteration and a next view is chosen that corresponds to the direction on the view sphere with the largest uncertainty gradient relative to the current position. The exploration process is allowed to run until all models fall below a prescribed error of fit threshold or until a maximum number of exploratory probes have been completed.

6.2. Experimental Results. The prototype implementation of our exploration system consists of a collection of processes distributed over a network of special and general purpose computers corresponding to the flow diagram shown in Figure 12. A 128×128 laser range-finder image was collected at each iteration. The total processing time per iteration is data-dependent, but for the example shown here was approximately 30 seconds. Of this the time taken to plan the trajectory was under 1 second, so it essentially comes for free in these experiments. The majority of the time was taken by the correspondence process used in data fusion. A more accurate positioning system than the one in our laboratory would speed up the process considerably since time to convergence of the correspondence algorithm is directly proportional to the initial guess determined from the robot. For all processes except volumetric modeling, computations were distributed between a Silicon Graphics IRIS 4D/35 and an Indigo VX workstation.

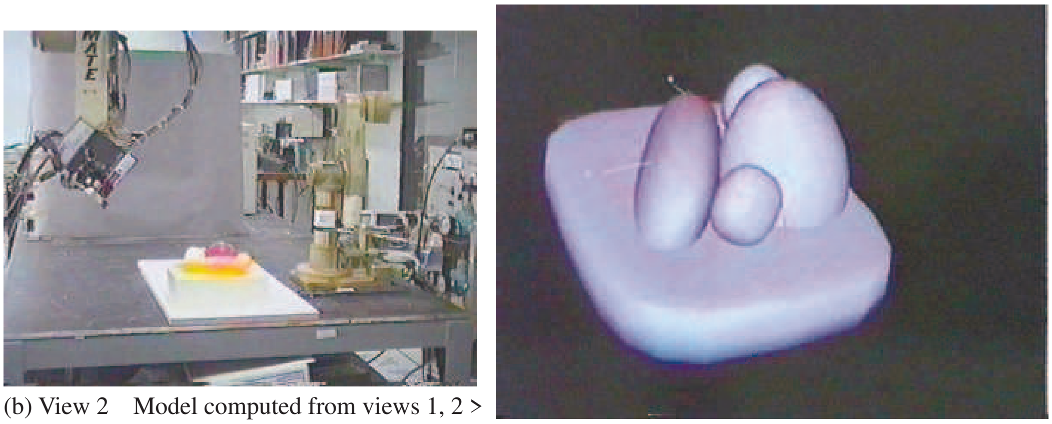
Some qualitative results are shown in Figure 14, where the explorer is discovering a scene consisting of a wooden block on top of which is an assortment of 4 fruits. The system starts off by coarsely sampling the workspace until the block and fruit come into view. It then positions the rangefinder such that the block and fruit are at the center of its field of view. This defines the initial position shown in Figure 14a. The corresponding model determined from this initial viewpoint is shown immediately to the right. All 5 parts comprising the "object" are correctly localized, but there are significant errors in the positions and shapes of each part. This is about the best that can be expected given only a single view of the scene without additional constraint information.

Next, a second viewpoint is computed based on the uncertainty surface of the part closest to the rangefinder. This new viewpoint is shown in Figure 14b along with the model computed from the previous and current views. The additional data serves to further constrain the shapes of the fruit, but provides little additional constraint on the wooden block. Using the same part as in the previous iteration, an uncertainty surface is computed and a third viewpoint determined (Figure 14c). This viewpoint brings the scanner low on the horizon from which more of the wooden block is visible. The resulting model, shown to the right, incorporates data from all three viewpoints and now correctly represents the shape of each part. We have verified that the overall process is stable, producing near identical results when the initial positions of objects in the scene are perturbed.

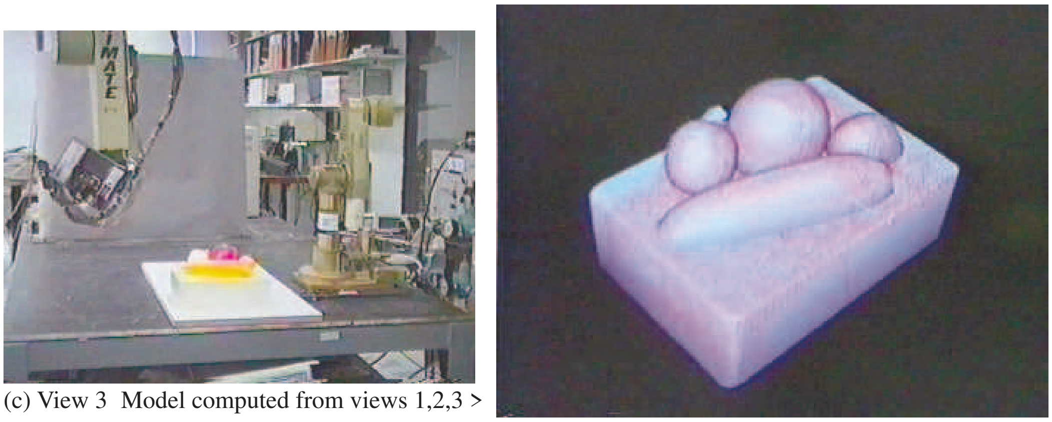
Some quantitative results from exactly the same system are given in Figure 15. In this case the object positioned in the work space was a small stone owl. Figure 15a



(a) View 1 Model computed from view 1 >



(b) View 2 Model computed from views 1, 2 >



(c) View 3 Model computed from views 1,2,3 >

FIGURE 14. Autonomous exploration sequence. (a) The initial viewpoint and corresponding scene model. (b) Second viewpoint determined from the uncertainties in the initial model. Scene model computed from fusion of data in the first and second views. (c) Final viewpoint and composite scene model computed from all 3 views.

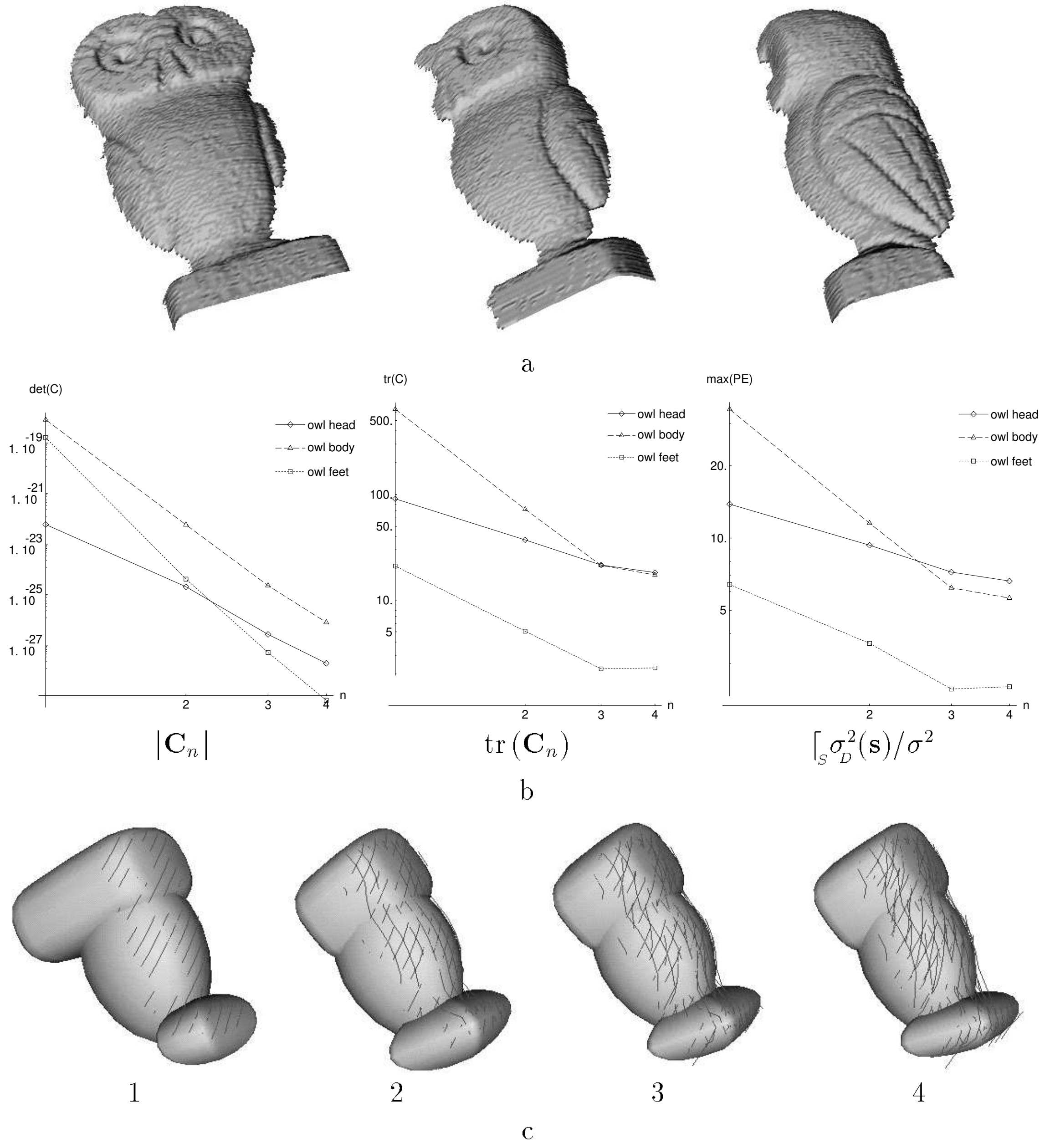


FIGURE 15. Autonomous exploration of owl statuette. a) Typical range data showing three different views of the owl's surface. b) The decrease in $|\mathbf{C}_n|$, $\mathrm{tr}\,(\mathbf{C}_n)$, and $\int_S \sigma_D^2(\mathbf{s})/\sigma^2$. c) The recovered superellipsoid part models at each iteration.

shows the kind of range data obtained from the scanner, though for this experiment the actual data used was sampled from grid four times courser, so we should not expect parts to evolve which depend on resolution of finer details (e..g the wings). A wide field of view was required so that range images from different views could be registered into the same coordinate frame [28].

Processing of the initial scan resulted in a segmentation of the range data into three separate patches. The superellipsoid part models fitted to each patch are shown in Figure 15c-1. Some of the range data scan lines can be seen on the surface, and give an idea of the amount of surface covered. Because it was closer than the other two parts, the explorer focussed its initial attention on the owl's body to select a new viewpoint. After moving the scanner to the new view another scan was taken, segmented, then registered with the first. The second set of scan lines crossing the original in Figure 15c-2 indicate the region from which new data was obtained, and also the success of registration algorithm. The part models, particularly the head, are now more constrained because the second view added additional data on the side of owl. The process continues for another two iterations, with the part models, especially the feet, becoming smaller at each step. Further improvement would require access to the back surfaces, but these are inaccessible because the owl is resting on a table.

All the uncertainty criteria for each part model decrease at approximately the same rate (Figure 15b) probably because of the wide field of view, and because each part is roughly in the same pose. Under these conditions similar sensor trajectories would have been obtained with attention focussed on any of the models. The slightly faster rate of decrease for the feet can be attributed to the model's smaller size and the larger percentage of its surface covered by the data collected at each iteration. The leveling off of $\int_S \sigma_D^2(\mathbf{s})/\sigma^2$ (and $\mathrm{tr}\,(\mathbf{C})$) after the third iteration is an indication of the inaccessibility of parts of the surface, and of the inability of the collected data to improve knowledge of the surface in those positions.

It is important to note that in both of these experiments many of the assumptions upon which the exploration algorithm is based are not valid. For example, the shape of the owl cannot be described precisely with superellipsoid models — the body is asymmetric, and the head is not even convex. Despite this exploration performs well and recovers useful models which capture the general size, pose, and shape of each part. What we observe in practice is that the distribution of prediction variance across the surface of the model still makes sense even when the model fits the data badly. It is lower where there is data, and increases rapidly where there is none. A strategy based on this distribution might be sub-optimal, but the sensor will still be driven to locations where no data was previously obtained. We have assumed that the a posteriori model parameter distribution is normal, but even gross departures from normality result in reasonable behaviour, and in a system which exhibits a high degree of robustness.

7. Conclusions

The results presented in the previous section demonstrate that feedback based on model uncertainty can effectively be used to plan gaze and reliably infer scene descriptions. In particular, we demonstrated how a description comprised of articulated volumetric models could be automatically computed from a sequence of exploratory probes obtained by a mobile laser range-finding system. The resulting models are sufficiently robust to serve as object descriptors for purposes of manipulation and recognition. Because the process is entirely data-driven, the system is well-suited as a basis for artificial perception in unstructured environments. It is also completely-autonomous. Sensor measurement, data fusion, model inference and gaze planning proceed iteratively either until a stable description of the scene is obtained or a prerequisite amount of data has been collected.

We are currently extending this research in a number of directions. More general purpose models are being investigated which can take into account the dynamic behaviour of objects. Ways of implementing backtracking when a model fails to account for newly acquired data are being incorporated into the system. The current method used to merge (fuse) information from different viewpoints is being generalized to properly account for occlusions. By proceeding in this manner we hope to eventually learn enough about the general problem to build systems that are truly capable of autonomous exploration.

References

[1] N. Ahuja and J. Veenstra. Generating octrees from object silhouettes in orthographic views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:137–149, Feb. 1989.

[2] J. Aloimonos and A. Bandyopadhyay. Active vision. In ICCV87 [10], pages 35–54.

[3] R. Bajcsy. Active perception. *The Proceedings of the IEEE*, 76(8):996–1005, 1988.

[4] R. Bajcsy and M. Campos. Active and exploratory perception. *Computer Vision, Graphics, and Image Processing*, 56(1):31–40, July 1992.

[5] R. Bajcsy and F. Solina. Three dimensional object recognition revisited. In ICCV87 [10].

[6] L. Birnbaum, M. Brand, and P. Cooper. Looking for trouble: Using causal semantics to direct focus of attention. In ICCV93 [11], pages 49–56.

[7] V. Caglioti. The optimal next exploration. In *PROC. European Conference on Robotics and Intelligent Systems*, 1991.

[8] W. Cheung, F. P. Ferrie, G. Carayannis, and J. B. Edwards. Rockpile surface decomposition: Machine vision in mining. In *PROC. Canadian Conference on Industrial Automation*, Montréal, Québec, June 1-3, 1992 1992.

[9] T. Choi, H. Delingette, M. DeLusie, Y. Hsin, M. Hebert, and K. Ikeuchi. A perception and manipulation system for collecting rock samples. In *Proceedings, 4TH Annual Space Operations, Applications, and Research Symposium: SOAR 90*, Albuquerque, NM, June 1990.

[10] Computer Society of the IEEE. *Proceedings, 1ST International Conference on Computer Vision*, London,UK, June 1987. IEEE Computer Society Press.

[11] Computer Society of the IEEE. *Proceedings, 4TH International Conference on Computer Vision*, Berlin, Germany, May11-14 1993. IEEE Computer Society Press.

[12] C. Connolly. The determination of next best views. *Proc. IEEE Int. Conference of Robotics Automation*, pages 432–435, 1985.

[13] P. Cooper and M. Brand. A knowledge framework for seeing and learning. In K. Ikeuchi and M. Veloso, editors, *Symbolic Visual Learning*, volume 2. Oxford University Press, 1995.

[14] V. Federov. *Theory of optimal experiments*. Academic Press, New York, 1972.

[15] F. P. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and super-quadrics: Geometry from the bottom up. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8):771–784, Aug. 1993.

[16] F. P. Ferrie, S. Mathur, and G. Soucy. Feature extraction for 3-D model building and object recognition. In A. Jain and P. Flynn, editors, *3D Object Recognition Systems*. Elsevier, Amsterdam, 1993.

[17] G. D. Hager. *Task-Directed Sensor Fusion and Planning: a computational approach*. Kluwer Academic Publishers, 1990.

[18] D. Hoffman and W. Richards. Parts of recognition. *Cognition*, 18:65–96, 1984.

[19] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[20] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker. Shapes, shocks, and deformations, I: The components of shape and the reaction-diffusion space. Technical Report LEMS 105, LEMS, Brown University, May 1992.

[21] K. N. Kutulakos and C. R. Dyer. Global surface reconstruction by purposive control of observer motion. In *Proceedings, Conference on Computer Vision and Pattern Recognition*, pages 331–338, Seattle, June 1994. Computer Society of the IEEE, IEEE Computer Society Press.

[22] A. Lejeune and F. Ferrie. Partitioning range images using curvature and scale. In *PROC. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 800–801, New York City, New York, June 15-17 1993.

[23] D. J. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4:590–604, 1992.

[24] J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):417–433, May 1993.

[25] A. M. Mood and F. A. Graybill. *Introduction to the Theory of Statistics*. McGraw-Hill Book Company Inc., New York, N.Y., 1963.

[26] A. Pentland. Recognition by parts. In ICCV87 [10], pages 612–620.

[27] H. Sorenson. Least-squares estimation: from Gauss to Kalman. In H. W. Sorenson, editor, *Kalman Filtering: Theory and Application*. I.E.E.E. Press, 1985.

[28] G. Soucy. View correspondence using curvature and motion consistency. Master's thesis, Dept. of E.E., McGill Univ., 1992.

[29] K. Tarabanis and R. Y. Tsai. Computing viewpoints that satisfy optical constraints. In *Proceedings, Conference on Computer Vision and Pattern Recognition*, pages 152–158, Lahaina, Maui, Hawaii, June 1993. Computer Society of the IEEE, IEEE Computer Society Press.

[30] P. Whaite and F. P. Ferrie. From uncertainty to visual exploration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1038–1049, Oct. 1991.

[31] P. Whaite and F. P. Ferrie. Uncertain views. In *PROC. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3–9, Champaign, Illinois, June 15-18 1992.

[32] P. Whaite and F. P. Ferrie. Active exploration: Knowing when we're wrong. In ICCV93 [11], pages 41–48.