Active Shape-from-Shadows with Controlled Illuminant Trajectories

James J. Clark[1] and Lei Wang[2]

[1]Centre for Intelligent Machines, McGill University
[2]Center for Imaging Science, Washington University

February 7, 2002

Abstract

We present an active vision algorithm for computing the orientation and position of a locally planar object, onto which is cast a shadow of the edge of a half-plane at an unknown location. This algorithm utilises active position control of a point light source, and employs a Kalman filter to perform temporal integration of measurements. The light source position is adjusted after each measurement so as to reduce the trace of the expected state estimate error covariance matrix for the next measurement. We demonstrate the active shape-from-shadows algorithm using a real robotic system.

1 Introduction

It has long been known that images of object self- and cast-shadow boundaries contain information as to the shape of the object. This information is not, in general, sufficient to completely determine the object shape. The additional information that is required is often provided in the form of *a priori* assumptions about the object shape [22]. This often produces poor results, as the *a priori* model overly biases the result, leading to unacceptable distortions in the computed object shape.

One way of providing additional information, and of reducing dependence on *a priori* assumptions, is to obtain shadow images of the object taken using a set of differing illuminant positions. There have been many techniques proposed that use moving shadows, such as those produced by the track of the sun as the day progresses, to provide the missing information needed to determine object shape (e.g. [10, 12, 17]).

In this paper we describe a technique that differs from the previously cited approaches in that it uses controlled motion of a nearby light source, and provides absolute surface depth information as well as surface normal information. Our technique is based on the principles of *active vision* [1, 2].

Active vision systems are characterized by control over various aspects of the imaging process. A number of active vision techniques have been proposed that use such control to improve the process of obtaining shape information about object surfaces. For example, Whaite and Ferrie [24] introduced a method for determining the location of a sensor that minimizes a measure of uncertainty about the shape of an object. They applied this to the task of building up object surface descriptions from range-finder data. Shmuel and Werman [19] presented a Kalman Filter based temporal integration scheme for depth from stereo where the cameras are positioned so as to minimize an uncertainty measure. The Kalman filter allows information from a sequence of sensor readings to be integrated, resulting in a reduction in the effect of noise on the information derived from the sensor data. Each of these techniques shares a common thread in that they have control over some aspect of the imaging process, and use this control in a way so as to optimize the extraction of information from the sensor data.

Control of imaging parameters in active vision systems is not limited to control over the position of the image sensor. One can control other parameters such as the position of the illumination source. For example, Clark [4] describes a technique for obtaining the shape and absolute position of objects from shading information that rely on the control over the position of a light source. Our shape-from-shadows technique is based on this idea, as it obtains object shape and position information from the sequences of images of shadows cast by objects, acquired as we purposely vary the position of a point light source. We apply a trajectory specification process based on those proposed by Whaite and Ferrie [24] and Shmuel and Werman [19] to determine the light source motions.

The organization of the paper is as follows: section 2 describes the geometry underlying the shadow formation process and provides the derivation of the equations that form the basis of the shape-from-shadows algorithm; section 3 describes how we make use of active control over the light source position to facilitate estimation of the object shape and position parameters; section 4 describes a set of experiments with real imagery that demonstrates the validity of the active shape-from-shadows technique and the effectiveness of the trajectory optimization process. The appendix gives the derivation of equations used to generate the light source trajectory using the Shmuel-Werman approach to optimizing the acquisition of data.

2 The Shape From Controlled Shadow Motion Algorithm

2.1 Geometry of the shape-from-shadows Algorithm

Our active shape-from-shadows technique is based on controlling the motion of a point light source that shines on an object and a background plane. The object casts a shadow onto the background, and also self-shadows a portion of its own surface. The locations of the cast-shadow and self-shadow boundaries are determined with a simple image analysis process acting on images acquired from a video camera viewing the scene.

The geometrical aspects of our algorithm are shown in figure 1. For simplicity we show the 2-D case, but the analysis given below is for the more general 3D case. We assume that the light

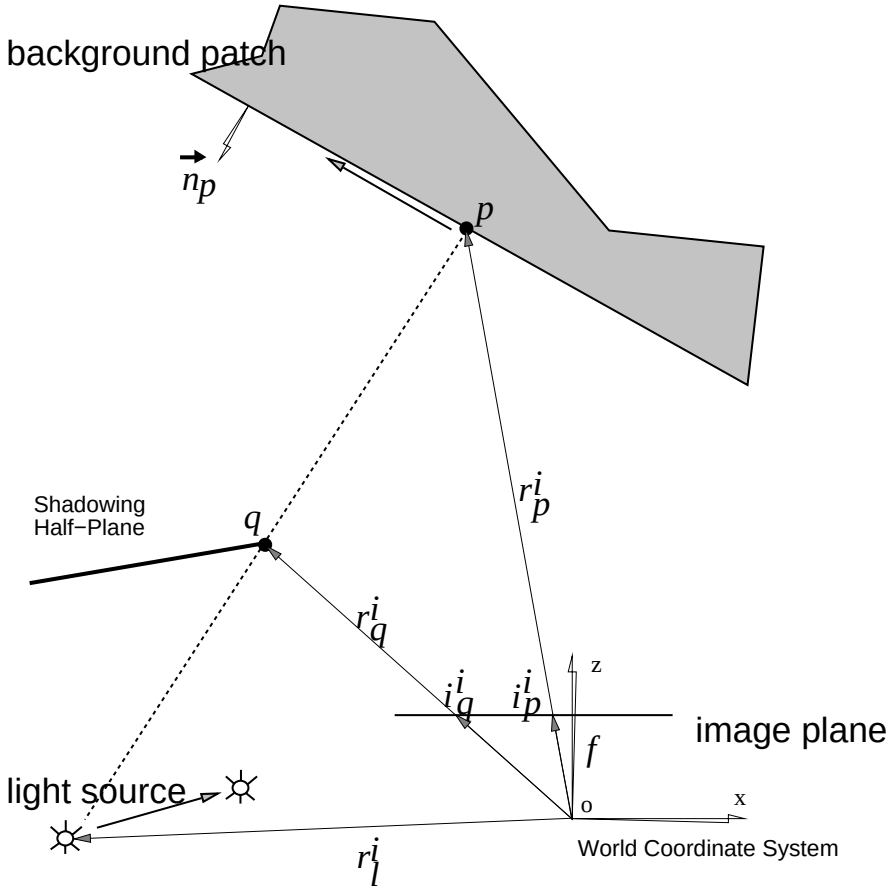


Figure 1: Geometry for the active shape-from-shadows technique.

source is a point source, that the background object (onto which the shadow is cast) is planar and that the foreground object (which casts the shadow) is a half-plane or a surface having a sharp edge. As the algorithm determines shape on a pixel by pixel basis, the planarity assumption on the background object needs only be satisfied locally. The background plane is defined by the vector, $\vec{n}$, normal to its surface, and by a vector, $\vec{r}_p$, from the origin of a world coordinate frame to any point on the plane, $\vec{P}$, via the following equation:

$$\vec{n}_p^T \vec{r}_p + 1 = 0 \tag{1}$$

The three components of $\vec{n}$ define the plane. The reciprocal of the magnitude of $\vec{n}$ can be seen to be the closest distance from the background plane to the origin of the world coordinate system. The direction of $\vec{n}$ is seen to be normal to the background plane. Thus $\vec{n}$ is a constant, independent of the shadow point p that is being viewed. The shadow boundaries at points p and q project to the camera's image plane at the points represented by the vectors $\vec{i}_p$ and $\vec{i}_q$.

With the image measurements $\vec{i}_p$ and $\vec{i}_q$, and the known light source position $\vec{r}_l$, we seek to find the least number of equations from which we can solve for the unknown object shape and position parameters: $\vec{n}_p$, $\vec{t}_q$, M_q, $\vec{r}_p$, and $\vec{r}_q$. Each light source position gives rise to a set of constraint

equations that describe the geometric relations among points on the surfaces and their images on the image plane. Assuming perspective projection and a pinhole camera model, the image formation process gives rise to the following *perspective projection equations*:

$$\vec{i}_p = \frac{f}{r_{pz}}\vec{r}_p; \quad \vec{i}_q = \frac{f}{r_{qz}}\vec{r}_q \tag{2}$$

Noticing that, for each light source position, the vectors $\vec{r}_l$, $\vec{r}_p$, and $\vec{r}_q$, are coplanar, we have the *illumination constraint equations*:

$$(\vec{r}_p - \vec{r}_l) \times (\vec{r}_q - \vec{r}_l) = 0, \tag{3}$$

The three equations provided by this constraint are linearly dependent, however. To see this, consider the case where both $\vec{r}_l$ and $\vec{r}_q$ are known. These define a line on which $\vec{r}_p$ must fall, but leave free where on the line $\vec{r}_p$ falls.

It can be seen that the illumination constraint permits us to find correspondences between the images of self-shadow points and cast-shadow points. To see this, consider the line that passes from the illumination source, through a given point on the self-shadow boundary, and thence on to the corresponding point on the cast-shadow boundary (this is the dashed line in figure 1). This line in 3-D space will project to a line in the image plane. We assume that we know the position of the light source so that its "image" on the image plane can be assumed to be known (even if it might not actually fall on the limited extent of the camera sensor). We can choose a self-shadow boundary point in the image plane. These two points then define a line in the image plane. The cast-shadow boundary point that corresponds to the chosen self-shadow boundary point is easily found as the intersection of this line with the 2-D cast-shadow boundary curve. This correspondence technique works no matter whether the shadow boundaries are straight or curved.

The assumption that the object that is casting the shadow contains a "sharp" edge at the self-shadow boundary simplifies the solution for the position and orientation parameters. At a sharp edge the self-shadow boundary does not move as the light source moves. Thus the vector $\vec{r}_q$ does not vary with the light source position. The sharp edge assumption is, of course, not universally valid, but in man-made environments objects with sharp edges abound.

There are a total of $6 + 3k$ unknowns, where k is the number of different light source positions. These are the 3 components of $\vec{r}_q$ and the 3 components of $\vec{n}_p$; the 3 components of $\vec{r}_p$ at each light source position. There are a total of 1+5k independent equations obtained from the constraint relating $\vec{n}$ and $\vec{r}_p$; the 4 image plane measurements $i_{px}, i_{py}, i_{qx}, i_{qy}$ and the single independent equation implied by equations (3). Thus, we need at least three light source positions to be able to solve for the unknowns. If we use only three equations, however, the solution process will involve finding the roots of quadratic equations. While this is possible, it results in multiple solutions. If we use four light source positions, however, we can reformulate the equations in a way which only require the solution of a linear system.

To arrive at the linear system solution, we split the solution process into two stages. In the first stage we look at the illumination constraint equation (eq. 3). This equation expands to:

$$\vec{r}_p \times \vec{r}_q + \vec{r}_q \times \vec{r}_l + \vec{r}_l \times \vec{r}_p = 0 \tag{4}$$

As noted earlier, two of the three equations in this vector equation are redundant, so we should only consider one component. Let us choose the y-component. Expanding out the cross products and choosing the y-component gives us:

$$(r_{px}r_{qz} - r_{pz}r_{qx}) + (r_{qx}r_{lz} - r_{qz}r_{lx}) + (r_{lx}r_{pz} - r_{lz}r_{px}) = 0 \tag{5}$$

If both r_{pz} and r_{qz} are non-zero (which will usually be the case) we can divide this equation by $r_{pz}r_{qz}$. We can use the perspective projection equations (2) to express r_{px}, r_{py}, r_{qx} and r_{qy} in terms of i_{px}, i_{py}, i_{qx} and i_{qy}. Doing so allows us to rewrite the above expression as:

$$(i_{px} - i_{qx}) + (i_{qx}r_{lz} - fr_{lz})/r_{pz} + (fr_{lx} - i_{px}r_{lz})/r_{qz} = 0 \tag{6}$$

Finally, we can replace the inverse depth of the background plane with:

$$1/r_{pz} = -i_p^T \vec{n}_p / f \tag{7}$$

With this substitution we get, for each light source position, an equation which is *linear* in three of the unknowns - $\vec{n}_p$ and the reciprocal of another, $1/r_{qz}$:

$$\begin{pmatrix} \frac{i_{px}}{f} & \frac{i_{py}}{f} & 1 & \frac{\gamma_x}{\omega_x} \end{pmatrix} \begin{pmatrix} n_{px} \\ n_{py} \\ n_{pz} \\ 1/r_{qz} \end{pmatrix} = -\frac{\xi_x}{\omega_x}. \tag{8}$$

where

$$\begin{aligned} \omega_x &= f\, r_{lx} - i_{qx}\, r_{lz} \\ \gamma_x &= f\, r_{lx} - i_{px}\, r_{lz} \\ \xi_x &= i_{px} - i_{qx}. \end{aligned} \tag{9}$$

A linear system constructed from four such equations, obtained from four different positions of the light source,

$$\begin{pmatrix} \frac{i_{px}^{(1)}}{f} & \frac{i_{py}^{(1)}}{f} & 1 & \frac{\gamma_x^{(1)}}{\omega_x^{(1)}} \\ \frac{i_{px}^{(2)}}{f} & \frac{i_{py}^{(2)}}{f} & 1 & \frac{\gamma_x^{(2)}}{\omega_x^{(2)}} \\ \frac{i_{px}^{(3)}}{f} & \frac{i_{py}^{(3)}}{f} & 1 & \frac{\gamma_x^{(3)}}{\omega_x^{(3)}} \\ \frac{i_{px}^{(4)}}{f} & \frac{i_{py}^{(4)}}{f} & 1 & \frac{\gamma_x^{(4)}}{\omega_x^{(4)}} \end{pmatrix} \begin{pmatrix} n_{px} \\ n_{py} \\ n_{pz} \\ 1/r_{qz} \end{pmatrix} = - \begin{pmatrix} \frac{\xi_x^{(1)}}{\omega_x^{(1)}} \\ \frac{\xi_x^{(2)}}{\omega_x^{(2)}} \\ \frac{\xi_x^{(3)}}{\omega_x^{(3)}} \\ \frac{\xi_x^{(4)}}{\omega_x^{(4)}} \end{pmatrix} \tag{10}$$

will give a unique solution for the shape variable $\vec{n}_p$, and the inverse depth of the object edge point, $1/r_{qz}$, as long as the coefficient array is invertible. From these values, the remaining parameters can be recovered, in stage 2 of the solution process, as follows:

$$\vec{r}_q = \frac{r_{qz}}{f}\vec{i}_q; \quad \vec{r}_p = \frac{-1}{i_p{}^T n_p}\vec{i}_p \tag{11}$$

Using these equations, we can solve, in closed form, for the desired quantities from image measurements obtained at four different light source positions.

It should be noted that the algorithm determines the orientation and location of the shadowed object pointwise, and so the assumption of planarity for the shadowed object need only be satisfied locally. Thus the algorithm will work for piecewise planar surfaces, such as polyhedra, except at the edges between facets. The algorithm will also function on curved surfaces, as long as the curvature is low enough that the surface is approximately flat on the scale of the movement of the shadow boundary between successive light source positions. Deviations from flatness will result in an error in the derived shape and position quantities.

2.2 Relation to Structured Light Techniques

The shape-from-shadows approach described above is similar in many ways to structured light techniques [16, 18]. In a typical structured light approach, a planar sheet of light, or "light stripe", is generated and used to illuminate an object. A camera views the object and measures the locations at which the sheet of light intersects the object surface. From the observed image coordinates of the intersection points one can compute their 3-D locations in space. One can think of the shadow boundary cast in our approach as equivalent to the light stripe in the structured light techniques. Our shadow casting method has a number of advantages over the light striping methods, however. The first advantage is that our method provides the surface normal of the shadowed surface, which light striping methods do not give directly. Secondly, our method provides the 3-D position of the edge of the shadowing object. In structured light techniques, this edge is fixed in space and is either known *a priori* or determined with a calibration process. In our approach the object that casts the shadow can be in an unknown location, and does not have to be part of the imaging apparatus. For example, a robot waving a flashlight around could use the sharp edge of an object in its vicinity to cast shadows on walls, and use our algorithm to compute the distance to the wall and its orientation. Finally, observation of the self-shadow boundary permits frame-to-frame correspondence of points on the cast-shadow boundary as the light source moves. Determining the correspondence between points as the light stripe moves is often one of the most difficult aspects of structured light techniques.

The active shape-from-shadows algorithm can be simplified somewhat by fixing the shadowing object in space relative to the camera. For example, the shadowing edge could be made part of the measurement apparatus along with the controllable light source and the camera. In this case equation (10) reduces to a system of 3 equations, so we need to make only 3 measurements. This implementation would be closer to that of standard structured light techniques, except that we still retain the ability to compute the local surface normal of the shadowed object.

3 Temporal Integration and the Control of the Light Source Position

3.1 Temporal Integration

In the previous section we showed that the desired shape and position parameters of the shadowing and shadowed surfaces can be obtained from measurements of the images of the self-shadow and cast-shadow boundaries made from four different positions of the light source. The results of doing this straightforward solution are, in general, unsuitable in practice due to noise in the localization of the images of the shadow boundaries and to uncertainties in the location of the light source. To alleviate this sensitivity to noise, we can apply a recursive filtering scheme, such as Least-Squares filtering, Wiener filtering, or Kalman filtering [8], to provide estimates of the orientation and position parameters, integrating information obtained over many light source positions.

In applying a recursive estimator to our problem we have two approaches that we can use. The first is to temporally integrate the solutions to the linear system given in equations (7). The primary drawback to this approach is that the noise in these solutions is, in general, non-Gaussian. Even if we assume that our sensor noise (i.e. shadow edge image localization error) is Gaussian, the nonlinear dependence of the solution with the sensor measurements results in non-Gaussian noise in the computed solutions. While some work has been done on estimation with non-Gaussian models (e.g. [11, 25]), the techniques are applicable only in certain cases and are computationally expensive. The paper by Wu and Kundu [25] provides a good overview of the difficulties involved in estimation with non-Gaussian noise models.

Another approach is to use the recursive filter to *regularize* the ill-conditioned solution process directly, rather than simply filtering the solution. As shown in [5], a recursive filter can be thought of as a Bayesian regularization, in which an ill-conditioned or ill-posed solution process is replaced by the well-conditioned process of finding the mean of the *a posteriori* conditional probability density of the state variables given the measurements. This probability density is the product of a prior model of the state estimate and a model of the relationship between the measurement and the state. In this approach, the state vector is as before, but the measurements are now the raw sensor values rather than the outputs of the solution process. In this case the measurement noise can often be well approximated as Gaussian, but the dependence of the measurement on the state variables is usually nonlinear. Unlike the case of non-Gaussian measurement noise, however, effective and computationally efficient techniques, such as the Iterated Extended Kalman Filter [8], exist to handle the case of nonlinear measurement equations. For these reasons it is better to apply the recursive filter to estimating the state directly from the raw sensor measurements rather than merely filtering the outputs of some, possibly ill-conditioned, solution process.

The Kalman filter is a specific type of recursive filter that has been successfully used in many vision algorithms. For example, the various vision algorithms described in [3, 6, 7, 9, 13, 14, 21] use the Kalman filter to integrate multiple measurements. Recursive estimators filters are especially suited for use in active vision algorithms, as active vision algorithms typically produce temporal sequences of image data and associated derived quantities. For example, in the work of Shmuel and Werman [19], a Kalman filter was used to integrate depth estimates obtained with a position controlled camera. At each step of the integration process, the position of the camera

was determined that would minimize the predicted covariance of the updated estimate. Similarly, Whaite and Ferrie [24] computed the sensor position that would minimize the expected variance in shape model parameters after the subsequent measurement.

It should be noted that, if we adopt the active vision approach of Shmuel and Werman and of Whaite and Ferrie, in which the imaging system is altered between measurements, then the statistics of the measurement noise may be non-stationary (but still well approximated as Gaussian). For this reason recursive filters that assume stationarity of the measurement noise, such as the Wiener filter, will not be applicable. The Kalman filter does not have this problem, however, and so is well suited to the active vision approach to temporal integration.

3.2 Application to Shape-From-Shadows

Based on the discussion above, we will apply an active Kalman filter to our shape-from-shadows task. We will use the approach of Shmuel and Werman [19] to specify the trajectory of the light source.

In this section, let us begin by outlining the basic equations involved in applying the Kalman filter to our task. Begin by defining, as our state vector, the six desired surface orientation and position parameters:

$$\vec{x} = \{r_{qx}, r_{qy}, r_{qz}, n_{px}, n_{py}, n_{pz}\} \tag{12}$$

The general form for the measurement equations are:

$$\vec{z} = h(\vec{x}, \vec{u}) + \eta(\vec{u}) \tag{13}$$

where the measurement vector $\vec{z} = (i_{px}, i_{py}, i_{qx}, i_{qy})$ consists of the x and y coordinates of the images of the cast and self-shadow boundaries of a single point on the surface. We will use a separate filter for each point along a shadow boundary. η is a vector of Gaussian random variables with mean zero and covariance matrix $R(\vec{u})$. For our application the precise form of $h(\vec{x}, \vec{u})$ can be obtained from equation (2) and the illumination constraint equation (3). The four components of the measurements can be written as:

$$i_{px} = f\frac{(e_1 \times n_p) \cdot (r_l \times r_q) + r_{lx} - r_{qx}}{(e_3 \times n_p) \cdot (r_l \times r_q) + r_{lz} - r_{qz}} \tag{14}$$

$$i_{py} = f\frac{(e_2 \times n_p) \cdot (r_l \times r_q) + r_{ly} - r_{qy}}{(e_3 \times n_p) \cdot (r_l \times r_q) + r_{lz} - r_{qz}} \tag{15}$$

$$i_{qx} = f\frac{r_{qx}}{r_{qz}} \tag{16}$$

$$i_{qy} = f\frac{r_{qy}}{r_{qz}} \tag{17}$$

where $e_1 = (1, 0, 0)$, $e_2 = (0, 1, 0)$ and $e_3 = (0, 0, 1)$.

Since the measurement equations are nonlinear, we will use the iterated extended Kalman filter. The update equations for the state variable estimates in the iterated extended Kalman filter are [8]:

$$\hat{x}_k^{i+1} = \hat{x}_{k-1} + K_k^i[z_k - h_k(\hat{x}_k^i - H_k(\hat{x}_k^i)(\hat{x}_{k-1} - x_k^i)] \tag{18}$$

with Kalman gain matrix

$$K_k^i = P_k H_k^T(\hat{x}_k^i)[H_k(\hat{x}_k^i)P_k H_k^T(\hat{x}_k^i) + R_k]^{-1} \tag{19}$$

where $i = 1, 2, ...M$ is an iteration counter, and the updated state estimate is taken to be the estimate after the last iteration, that is:

$$\hat{x}_k = \hat{x}_k^M \tag{20}$$

and

$$\hat{x}_k^1 = \hat{x}_{k-1} \tag{21}$$

$H_k(\hat{x}_k^i)$ is the linear term of the taylor's series expansion of the measurement function $h(\vec{x})$ evaluated at $\vec{x} = \hat{x}_k^i$,

$$H_k(\hat{x}_{k-1}) = \left.\frac{\partial h_k(\vec{x})}{\partial \vec{x}}\right|_{\vec{x}=\hat{x}_{k-1}} \tag{22}$$

Between measurements, the state estimate error covariance is updated using

$$P_k = [I - K_k^M H_k(\hat{x}_k)]P_{k-1} \tag{23}$$

The state estimate error covariance matrix P provides a measure for the amount of uncertainty in the estimate of the state variables. At each step of the Kalman filter estimation process the information provided by the new measurement reduces the error in the state estimates.

Note that, in our application, the scene is assumed to be static, apart from the motion of the light source. Thus we will assume that the state vector is absolutely static.

3.3 Trajectory Determination

In our active shape-from-shadows technique it is assumed that we are able to control the position of the light source. This is typical of active vision algorithms in general, which usually incorporate the control of some imaging parameter. Developers of active vision algorithms frequently neglect to provide a specification for trajectory generation, however, relying instead on predefined trajectories, or on trajectories specified by a process external to the algorithm (such as the use of the motion of the sun, or the obstacle-avoidance system of a mobile robot). In all active vision systems the controllable parameters influence, to some extent, the measurements that are made, and one should choose the parameters so that this influence is beneficial. There is a wealth of

literature on the subject of sensor planning in active vision systems that we can draw on. A good review of sensor planning approaches in computer vision can be found in Tarabanis *et al* [20].

The sensor planning problem has also been studied extensively in the field of automatic control. For example, Meier *et al* [15] describes a method for jointly optimizing feedback control of a plant and the measurement subsystem. They show that, assuming linearity in the plant, assuming that the measurement system is linear in the state and measurement noise (but not necessarily in the measurement control), and taking as the performance measure a quadratic function of the state and measurement noise with an additive measurement control cost, then the plant control policy can be decoupled from the measurement control policy, and this measurement control policy can be determined *a priori*.

In most active vision algorithms, including the one being discussed in this paper, one is not concerned with control of the state variables, but only with estimation of the state variables. Thus, the control-theoretic formulation of the measurement optimization process made by Meier *et al* is overly complex. One can simplify their approach by neglecting the plant control and considering only the optimization of the measurement subsystem control. Following Meier *et al* we can use the trace of the state estimate error covariance matrix, P, plus a control cost, as the performance measure to be optimized.

Sensor planning based on the concept of minimizing some function of the state estimate covariance matrix is common in active vision. For example, Shmuel and Werman [19] consider the case of controlling the direction of motion of a single camera in order to optimize the computation of depth from the disparities induced by the camera motion. They derive an expression for the predicted uncertainty (variance of the state variable, in this case inverse depth) as a function of the camera displacement direction. They then find the direction which minimizes this predicted uncertainty, by setting to zero the derivative of this expression with respect to the direction. In the work of Whaite and Ferrie [24], the determinant of the state estimate error covariance matrix was used as the objective function for optimization purposes.

For control of our active shape-from-shadows technique we propose to take the same approach as that of Shmuel and Werman [19]. That is, we will use an (iterated extended) Kalman filter to estimate the object position and orientation parameters, and control the light source position so that some measure of the magnitude of the state error covariance matrix is minimized at each time step. Our problem is somewhat more complicated than that of Shmuel and Werman, as we have more than one state variable, and these state variables have different units. In our application, some of the elements of the state error covariance matrix have units of meters^2, some have units of degrees^2, and some have mixed units of meters*degrees. In the Shmuel and Werman algorithm it was obvious that the covariance itself could be used as an optimization measure, as it was a scalar quantity. In our application, however, the covariance is a matrix, and so we need to decide on a suitable optimization measure which somehow reflects the magnitude of the covariance matrix. There are many appropriate candidates, including the determinant and the maximum element, but we chose to use the *trace* (the sum of the diagonal elements) of the covariance matrix. As we see in the appendix this choice of optimization measure permits us to obtain a closed form analytical expression for the gradient of this measure with respect to the light source position.

To handle the differences in the units of various elements when computing the trace of the covariance matrix, the diagonal elements are weighted. The weighting can be implemented by

pre- and post- multiplying the covariance matrix by a constant diagonal matrix before computing the trace. In terms of the notation introduced earlier, the optimization problem associated with this choice of objective function can be expressed as follows (we drop the time step subscript from H and K for clarity):

$$\vec{u} = \arg\min_{\vec{u}} tr[WP_kW] = \arg\min_{\vec{u}} tr[W(I - KH)P_{k-1}W] \tag{24}$$

where $\vec{u}$ is the light source position, which is under our control, and W is a constant diagonal weighting matrix. This problem is equivalent to

$$\vec{u} = \arg\max_{\vec{u}} \epsilon(\vec{u}) \tag{25}$$

where

$$\epsilon(\vec{u}) = tr[WK(\vec{u})H(\vec{u})P_{k-1}W] \tag{26}$$

Noting that $K = P_{k-1}H^T[HP_{k-1}H^T + R]^{-1}$, we can express ϵ in the simple form:

$$\epsilon = tr[A^TBA] \tag{27}$$

by taking

$$A = HP_{k-1}W \tag{28}$$

and

$$B = [HP_{k-1}H^T + R]^{-1} \tag{29}$$

The matrix B is symmetric. The matrices A and B can be seen to depend on the value of the control vector $\vec{u}$. This dependency arises through the dependency of the measurement matrix H and the measurement noise covariance matrix R on the control variables. The prior covariance matrix P_{k-1} does not depend on the new control variables, only on their previous values.

The quantity, $\epsilon(\vec{u})$ that is to be maximized is a nonlinear function of the light source position $\vec{u}$. As such, solving for the value of $\vec{u}$ that minimizes ϵ may be difficult. The situation is exacerbated by the fact that, in practice, there may be constraints on the values of $\vec{u}$ that can be used. For example, the light source may be mounted on the end of a robot arm which has a limited range of motion. This means that solution of a constrained nonlinear optimization problem is required. We can, however, use the constraints on the values of the control parameters to our advantage in solving the optimization problem. In many active vision tasks that involve temporal integration, measurements are taken after small, incremental, changes of the imaging parameters (see, for example, [14]). In such a case we can replace the global constraints (those due to physical limits on the actuators that move the light source) with artificially imposed limits that constrain the space of possible control parameter values much more. In what follows we will make the assumption that the range of possible light source positions is restricted to a spherical region centred on the current value of the control parameter vector, and that the radius of this region is sufficiently small so as to allow the assumption that the gradient of ϵ with respect to the components of $\vec{u}$ is constant. We further assume that this constant gradient is non-zero. These assumptions imply that the extrema will lie on the boundary of the constraint sphere and that the extremal values are equal to

$$\vec{u}^* = \vec{u}_0 \pm r\frac{\nabla_{\vec{u}}\epsilon(\vec{u})}{|\nabla_{\vec{u}}\epsilon(\vec{u})|} \tag{30}$$

where r is the radius of the constraint sphere, $\vec{u}_0$ is the current control vector, and

$$\nabla_{\vec{u}} = \left(\frac{\partial}{\partial u_1}, \frac{\partial}{\partial u_2}, ..., \frac{\partial}{\partial u_n} \right) \tag{31}$$

It is shown in the Appendix that, because of the special form of the objective function being used (equation (19)), one can obtain the following closed form expression for the gradient of the objective function with respect to the control vector:

$$\frac{\partial \epsilon}{\partial u_i} = tr \left[WK \left(2 \frac{\partial H}{\partial u_i} P_k - \frac{\partial R}{\partial u_i} K^T \right) W \right] \tag{32}$$

It should be noted that this equation for the gradient is quite general, and can be used in other active vision applications, not just the shape-from-shadows technique being described here. Any application which needs to compute the gradient of the trace of the covariance matrix with respect to a controllable parameter of the measurement equation can make use of this result.

The gradient approach that we have taken to the light source trajectory specification gives rise to many issues that need to be considered. The first issue is one of the step size that is used in the computation of the displacement of the light source from one measurement to the next. The question of the proper step size needs to be considered with regard to the assumptions that we make. We assume that the illuminant displacement is restricted to lie within a spherical region. Thus the optimization problem is a constrained optimization problem. As with all constrained optimization problems the optimum will either lie in the interior or on the boundary of the constraint region. If we assume that the optimum lies on the boundary the solution process is simplified. Furthermore, if we assume that the objective function is planar within the constraint region, then the optimum will be located on the boundary at the location corresponding to the gradient of the objective function. Thus the step size (i.e. the radius of the constraint sphere) should be chosen so as to make the objective function approximately planar. If the step size is chosen to be too large, the objective function may deviate from planarity, and the boundary point corresponding to the gradient direction may no longer attain the maximum of the objective function. Conversely, if the step size is too small, the information gained by the new measurement will not reduce the error covariance significantly as compared with the globally optimum light source displacement. The choice of light source step size in practice must, therefore, balance the need for a large displacement with the error induced by our assumption of a planar objective function. Evaluation of this balance requires a detailed model of the objective function, which is difficult to obtain. In the next section we present a limited empirical study of the effect of step size, but clearly a more in-depth study is required.

It may appear that an incremental approach of the sort we are proposing results in a gradient descent on the objective function $\epsilon(\vec{u})$. This is not true, however, since the objective function depends on the current state vector estimate, $\hat{x}_k$. Thus, the form of the objective function is constantly changing, and so the "landscape" which is being "descended" is constantly changing its topography. In such a situation the objective function will never increase, but the system is not guaranteed to converge to a minimum. This aspect of incremental active vision control systems was noted by Whaite and Ferrie [24]. They observed that the change in the objective function due to the information gained at a sensor location is of a form that decreases the value of the

objective function at the current location relative to other locations, thereby forcing the sensor away from its current location. We have observed, however, that the objective function becomes flatter and flatter as convergence is attained.

Another issue is whether one can perform optimization of the entire trajectory, rather than merely optimization of the next measurement. In the work of Meier *et al* [15], it was shown that, under certain strong assumptions, it is possible to derive optimal policies for controlling the measurement system control variables *a priori*. Thus the measurement process effectively proceeds in an open-loop fashion, following the predefined optimal measurement policy. One can formulate a similar optimization problem for the active shape-from shadows task, but its solution is still very much an open research problem.

4 Experiments

In this section we describe a series of experiments intended to demonstrate the active shape-from-shadows algorithm as well as the benefits of the light source position control technique.

The experiments were implemented using a 6 degree-of-freedom industrial robotic manipulator and a real-time image analysis system. Details of the experimental setup can be found in [23]. In figure 2 we show a snap shot of the experiment area, in which one can see the robot manipulator used to move the point light source, the light source itself, the video camera which acquired the image data, and the objects under view. Figure 2b) shows the scene with an experiment in progress, where the only scene illumination is provided by the point light source.

4.1 The Experimental Procedure

We outline the experimental procedure below. Numbered entries are in the order of execution. The calibration process is split into two parts, an off-line part which needs only to be done once, and an on-line part which is done once at the beginning of every experiment. Details of the calibration process can be found in [23].

1. Off-line calibration.

   - Camera intrinsic parameters calibration.
   - Camera rotation calibration.
   - Robot hand calibration.
   - Light source–robot hand calibration.

2. On-line calibration.

- Hand-Camera calibration.
  The relationship of the robot hand with respect to the camera defines the position of the light source in the camera's coordinate system.

3. Initialization.
   This procedure generates the initial raw surface map needed for the recursive estimation process.

   (a) *Determine the initial light source trajectory.*
   We pick the initial light source trajectory to be a helix. This is used as the complete trajectory for the unfiltered experiments and for the state variable initialization stage of the experiments using recursive filtering. See section 4.2.1 for an example of an actual trajectory used in the experiments.

   (b) *Moving the light source.*
   Commands are issued to the robot controller to move the robot hand to carry the light source to a desired point on the trajectory.

   (c) *Recording the light source position and acquiring an image.*
   When the robot hand stops moving, the forward kinematics matrix of the robot hand is recorded. This together with the light source-robot hand calibration gives the coordinates of the light source in camera's coordinate system.
   At this point an image of the scene is captured for processing.

   (d) *Shadow detection & correspondence.*
   (Semi-)Detection of self and cast shadow boundaries.

   (e) *Repeat steps 3b through 3d until the loop count is 4.*

   (f) *Computing the raw surface map (background position and orientation) from the first four images using equations (10) and (11).*

4. Execution.

   (a) *Trajectory generation.*
   Depending on the application, there are two possibilities:

   - a fixed, pre-determined, helical trajectory.
   - compute the next light source position using equations (30) and (32).

   (b) *Moving the light source.*

   (c) *Recording the light source position and acquiring an image.*

   (d) *Shadow detection & correspondence.*

   (e) *Update the estimates of the background plane position and orientation.*

   (f) *Repeat steps 4a through 4e until the estimates converge.*

4.2 Experimental Results

We performed the following three sets of experiments:

1. The implementation of the two-stage linear solution to three dimensional surface orientation and position using equations (10) and (11). Raw (unfiltered) position and orientation measurements for the background plane and position measurements for the shadowing object are obtained in this set of experiment.

2. The implementation of a recursive filtering scheme using the iterative extended Kalman filter. In this experiment the light source is moved along a predefined helical trajectory.

3. As in experiment 2, except that the light source trajectory is specified by equations (30) and (32).

We show a typical series of images acquired by the camera as the light source moves in figure 3. For the entire sequence the image of the object edge segment remain unchanged. For each point along this segment we find its corresponding cast shadow within the dark box. This process is semi-automatic; for the first image acquired we overlay by hand two rectangular regions, which limit the area over which the search for the self- and cast-shadow boundaries is performed.

In our experiments the shadow boundaries were detected with an interactive procedure. During initialization, two rectangles are overlaid by a user on the first image of the scene, in such a way as to enclose the self-shadow and cast-shadow boundaries. These rectangles can be seen in figure 3. The dark rectangle encloses the self-shadow boundary and the light rectangle encloses the cast-shadow boundary. In subsequent time steps these rectangles are moved by a simple tracking algorithm to keep the shadow edges roughly centred within them. Once the rectangles have been defined, the shadow boundaries are located by finding the maximum gradient value along horizontal scan lines within the rectangles. This method works well if there are no surface markings near to the shadow boundary (which could be confused with the shadow boundary) and if the shadow boundaries are not near horizontal (which would cause problems with the computation of the gradient). In our experiments, the contrast of the shadow boundaries was high due to the high albedo of the surfaces, and to the minimization of the penumbra through the use of a point light source. In a less-controlled environment, it may be necessary to use a more complex shadow-detection scheme, which can distinguish shadow edges from non-shadow edges, and which can handle the blurring of shadow edges caused by extended light sources.

4.2.1 Experiment 1: Unfiltered Measurements with a Predefined Light Source Trajectory

In what follows we let the term "surface map" refer to the recovered positions of the background plane edge point and its associated background normal. In this experiment, the light source is moved along a helical path whose origin is at $(560, -160, 420)$ mm. The helices range in the y direction is 100 mm, and it projects into the x-z plane as a circle of radius 50 mm; Four different light source positions are required to recover one set of variables: the background normal n_p, the location of the shadowing half-plane edge points r_q, and, derived from r_q, their corresponding cast shadow points r_p on the background plane. After the initial set of variables have been computed, we use each new image together with three previous ones to obtain a new surface map. We do so for fifty images, and show the histograms of the recovered surface parameters in figures 4 and 5.

Recall the definition of n_p from equation (1):

$$n_p{}^T r_p = -1.$$

We have assumed that the background surface is locally planar, implying that n_p is the same for all the r_p's that make up the local background patch. Instead of plotting its three components, in figure 4 we visualize the background normal by rotating it about the camera coordinate axes. Let $\angle n_{p\_yz}$ denote the amount by which n_p has rotated *about* the y-axis *referenced* to the z-axis. Similarly, define $\angle n_{p\_zx}$ to be the rotation about the z-axis referenced to the x-axis, and $\angle n_{p\_xy}$ about the x-axis referenced to the y-axis. If we imagine that we are looking at the background through the camera lens, then $\angle n_{p\_yz}$ is the amount of rotation of the background *toward* the camera, $\angle n_{p\_zx}$ is the tilt. In the terms borrowed from robotics nomenclature, $\angle n_{p\_zx}$ = "roll", $\angle n_{p\_xy}$ = "pitch", and $\angle n_{p\_yz}$ is the amount of "yaw".

Since there are many points along the shadowing half-plane edge (for this set of experiments, 225), we have as many local surface maps for each four-image sequence. We look at one of them as a representative. In figure 5 we plot the histograms of the coordinates of the first point, r_{q_1}, along the object edge. In figure 3, the camera sees r_{q_1} as the top-most point in the white box.

We can observe a few things from these histograms. We get a sense of how the background plane is oriented with respect to the camera: it stands nearly upright with a few degrees of tilt, rotated approximately 120 degrees toward the camera. We also have an estimate of where the object edge approximately is. We did not obtain accurate ground truth in our experiments, but only obtained a rough measure of how good the results are by way of measurements with a yard stick. This gives us a sense of whether the reconstruction is near the truth, and in later experiments, whether the filters are converging towards reasonable values.

The statistical measures on the local surface maps indicate large errors as indicated by the large standard deviations. This is due to the unfiltered noise in acquiring the images and in placing the light source.

From the raw surface maps we can reconstruct the scene. To visualize the background, we simply plot all the cast shadows, since by definition they are part of the background (See figure 6(a)). We can see that the unfiltered noise in the local surface maps make the "surface" unrecognizable. This illustrates the fact that the simple two-stage surface recovery (raw map) is inaccurate in the presence of noise. The shadowing half-plane edge can be represented by collecting all the surface maps for points along the self-shadow boundary at one particular step. Shown in figure 6(b) as the "object edge" is the collection of such surface maps at the last time step. The particular helical light source trajectory that was used in this experiment is shown in figure 6(c).

4.2.2 Experiment 2: Iterated Extended Kalman Filtering with a Pre-defined Trajectory

In this section we describe the experiments and results of using the iterated extended Kalman filter with the same sequence of images acquired using the same pre-defined light source trajectory as in the previous experiment.

The first four images that are acquired as the light source moves gives, using equation (10), a raw surface map which serves as the initial state estimates for the Kalman filters. Each subsequent image provides the measurements of self- and cast-shadow pixels needed for the filters. All self-shadow points and their associated cast shadow points are assumed to be independent, so we use a separate Kalman filter on each shadow boundary pixel. For each of these Kalman filters we assume that the measurement noise covariance matrix is initially:

1. $R_0 = 1.18 \times \begin{pmatrix} 10^{-3} & 0 & 0 & 0 \\ 0 & 10^{-3} & 0 & 0 \\ 0 & 0 & 10^{-3} & 0 \\ 0 & 0 & 0 & 10^{-3} \end{pmatrix} \text{mm}^2.$

   The elements of this initial error covariance R_0 are chosen to correspond to a measurement noise with a standard deviation of 2-pixels. This number was based on empirical measurements on the shadow edges detected by our algorithm. The CCD elements of the Panasonic camera have a size of 17.2×10^{-3} mm. Doubling this value and then squaring we get the value for the diagonal entries of R_0.

2. We set the initial values of the state estimate error covariance matrix to be:

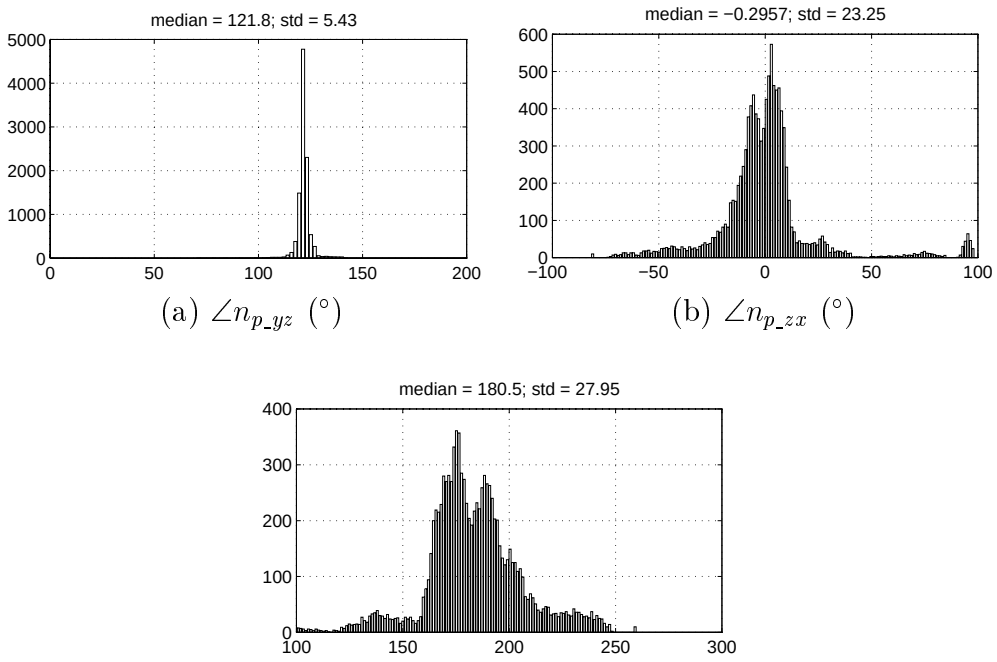
   $$P_0 = \begin{pmatrix} \hat{r}^2_{qx_0} & & \cdots & & 0 & 0 \\ & \hat{r}^2_{qy_0} & & & & 0 \\ \vdots & & \hat{r}^2_{qz_0} & \ddots & & \\ & & \ddots & \hat{n}^2_{px_0} & & \vdots \\ 0 & & & & \hat{n}^2_{py_0} & \\ 0 & 0 & & \cdots & & \hat{n}^2_{py_0} \end{pmatrix} \text{mm}^2,$$

   where $\hat{r}_{q_{(\cdot)_0}}$ is the initial estimate for the state variable $r_{q_{(\cdot)}}$, which is a component of the position of the self shadow, and $\hat{n}_{p_{(\cdot)_0}}$ is the initial estimate for the state variable $n_{p_{(\cdot)}}$, which is a component of the background normal vector. These values are based on our findings, via simulation studies, that the standard deviations in the initial state estimates can be well approximated by the magnitude of the state estimates produced by the linear solution action on the first four images.

To compare with the raw surface maps obtained in the previous experiment (section 4.2.1), we plot the distributions of the state estimates provided by the Kalman filter in figures 7 and figure 8.

Figure 2: Snap shots of experiments. Shown here are top: at the beginning of the experiments under indoor lighting; bottom: experiment in progress.

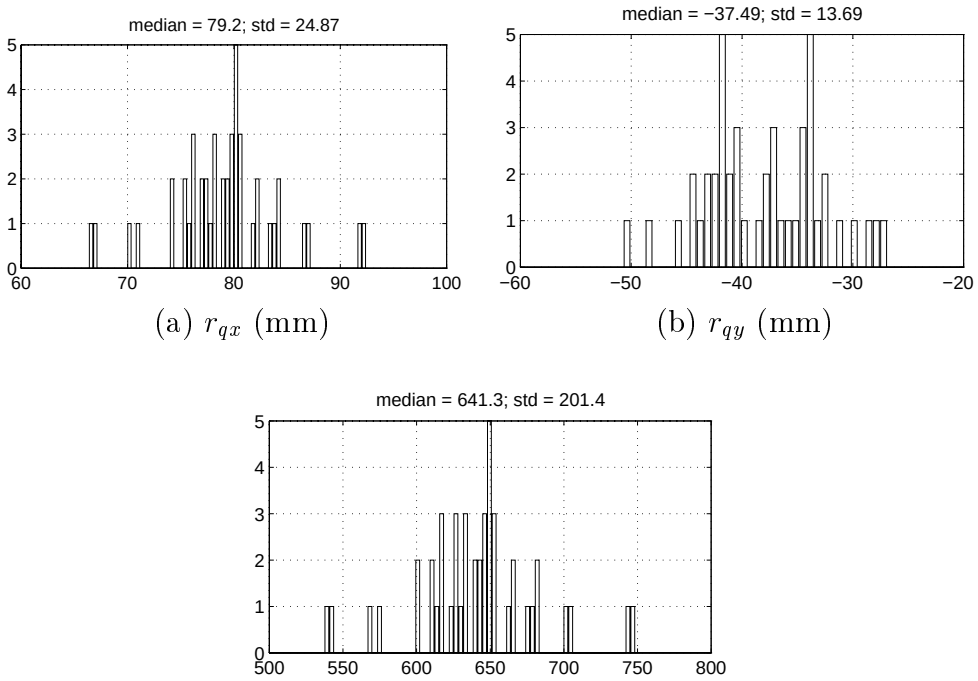
Figure 3: A series of images captured as the light source moves. The white and black rectangular outlines are overlaid by hand and serve as constraint regions for the shadow boundary detection algorithm. Enclosed in the white box is the self cast shadow, i.e., the portion of the object edge of interest. Enclosed in the dark box is the region in which we wish to find the corresponding cast shadow.



(a) $\angle n_{p\_yz}$ (°) (b) $\angle n_{p\_zx}$ (°)

(c) $\angle n_{p\_xy}$ (°)

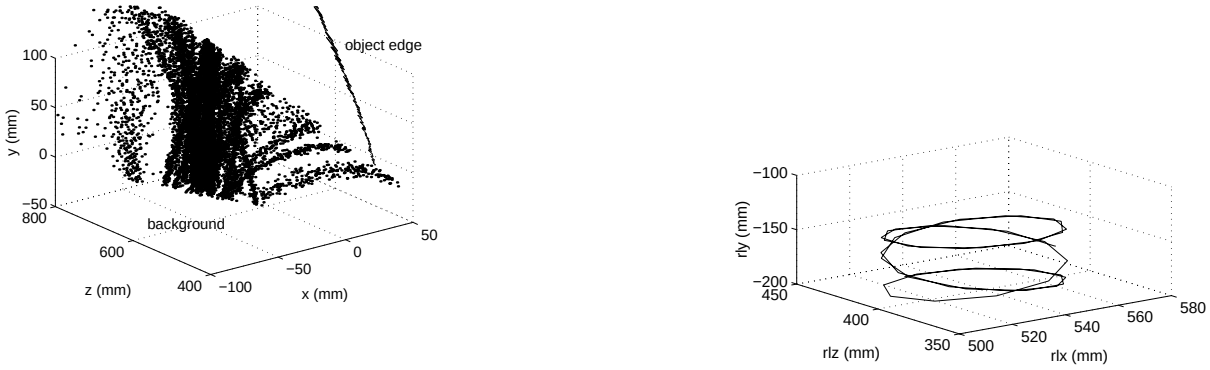
Figure 4: Histograms of background angles for the raw surface map: (a) amount of rotation of n_p about $\hat{y}$ w.r.t. $\hat{z}$—"yaw", (b) amount of rotation of n_p about $\hat{z}$ w.r.t. $\hat{x}$—"roll", (c) amount of rotation of n_p about $\hat{x}$ w.r.t. $\hat{y}$—"pitch". Their respective median values and standard deviations (in °) are marked at the top of each graph. Values along the horizontal axes are all in degrees (°).



(a) r_{qx} (mm) (b) r_{qy} (mm)

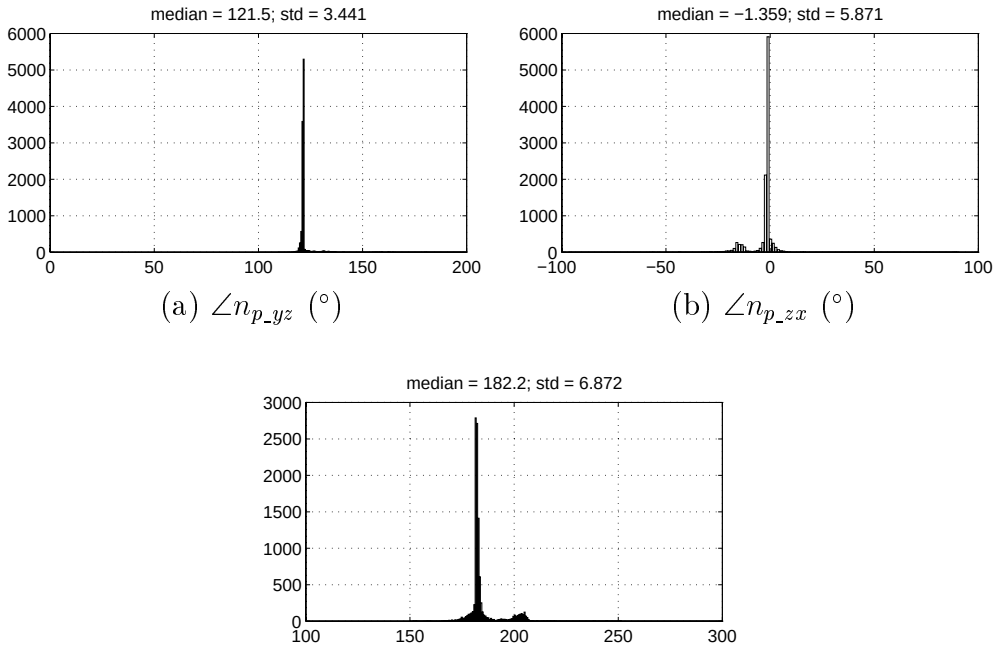
(c) r_{qz} (mm)

Figure 5: Histograms of object edge position r_{q_1} for the raw surface map: (a) the x-component r_{qx}, (b) the y-component r_{qy}, (c) the z-component r_{qz}. Their respective median values and standard deviations (in mm) are marked at the top of each graph. Values along the horizontal axes are all in mm.



(a) scene reconstruction (b) light source trajectory for the image sequence

Figure 6: 3D reconstruction for raw surface map: (a) A portion of the shadowing object edge map plus all the cast shadow maps which make up a portion of the background, (b) the helical light source trajectory for the images taken in this sequence. All values along the axes are in mm.



(a) $\angle n_{p\_yz}$ (°) (b) $\angle n_{p\_zx}$ (°)

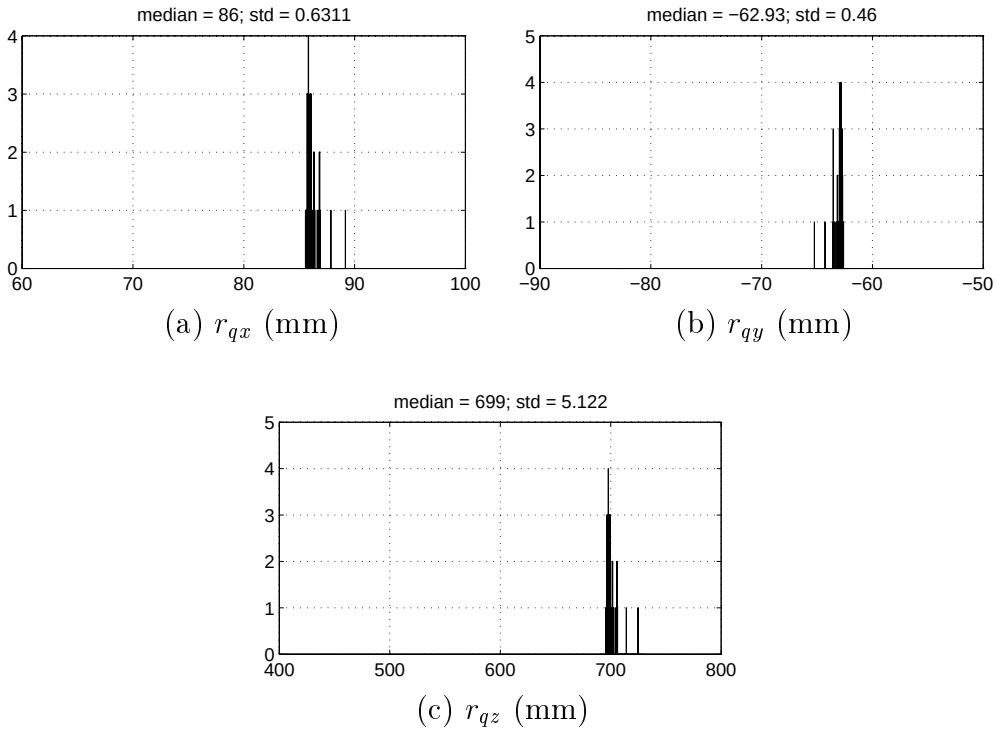
(c) $\angle n_{p\_xy}$ (°)

Figure 7: Histograms of background angles for the iterated extended Kalman filter: (a) amount of rotation of n_p about $\hat{y}$ w.r.t. $\hat{z}$—"yaw", (b) amount of rotation of n_p about $\hat{z}$ w.r.t. $\hat{x}$—"roll", (c) amount of rotation of n_p about $\hat{x}$ w.r.t. $\hat{y}$—"pitch". Their respective median values and standard deviations (in °) are marked at the top of each graph. Values along the horizontal axes are all in degrees (°).

As in figures 4 and 5 in section 4.2.1, we plot $\angle n_{p\_yz}$, $\angle n_{p\_zx}$, $\angle n_{p\_xy}$ (figure 7), and components of the first point on the object edge r_{q_1} (figure 8). We see that indeed the estimates provided by the Kalman filter exhibit much lower variances than the unfiltered raw surface maps.

How well does the Kalman filter work? To answer this question, we plot the state variables against time and examine their convergence behaviour. We pick the first point on the self shadow boundary, r_{q_1}. whose image in figure 3(b) is the top-most pixel inside the white box. In figure 9 we plot the angles $\angle n_{p\_yz}$, $\angle n_{p\_zx}$, and $\angle n_{p\_xy}$ for the background normal n_{p_1} associated with r_{q_1}, and the components of r_{q_1} in figure 10. We observe that n_{p_1} converges quickly, and it does not take more than 40 steps for r_{q_1} to converge. This is a reasonable speed, since if we could put all the components of the system together (i.e., image acquisition, image processing, the Kalman filter, robot command) on one machine and have the robot operate at real-time speed, it will take about a second to converge at video rate (30 image frames per second).

We would like to show the time evolution of the entire object edge. Since each point along the self shadow boundary carries a Kalman filter, we pick a few such points and track their state variables together to form an informative picture. This is shown in figure 11, where we plot the x, y, and z-components of a few estimated object edge points. We notice that although the graphs show convergence for all, they do not seem to all converge to a line (which they should since in this case the object edge is a line segment). For example see figure 11(c). An explanation for this can possibly come from the fact that the filters are independent and corrections to the state estimates may not occur at the same instance, and that at different shadow boundary locations



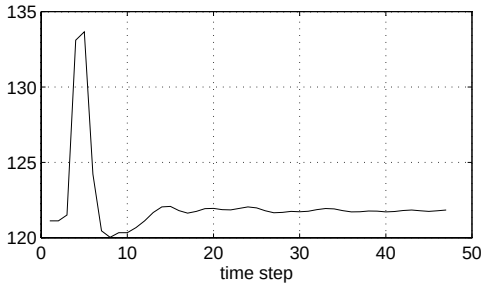
(a) r_{qx} (mm) (b) r_{qy} (mm)

(c) r_{qz} (mm)

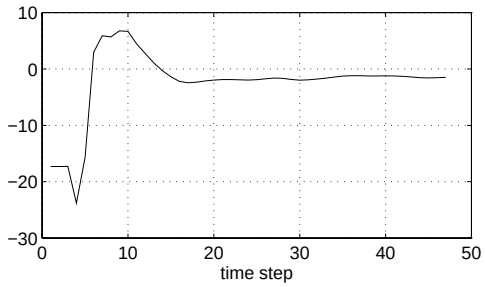
Figure 8: Histograms of object edge position r_{q_1} for the iterated extended Kalman filter: (a) the x-component r_{qx}, (b) the y-component r_{qy}, (c) the z-component r_{qz}. Their respective median values and standard deviations (in mm) are marked at the top of each graph. Values along the horizontal axes are all in mm.

the noise contributes to the image measurements equation differently. In this respect, correlating neighbouring points may help keep their spatial relationships correct, but it would not necessarily force the neighbouring points to be on a straight line or an arc or some specification. To do so, prior knowledge about the shape of the edge is needed, which we have not assumed so far. This prior knowledge about the object is also necessary when we integrate local maps to form a global structure, but this is a topic of future research. It should be noted that, in the figure, it appears that the shadow edge point that is nearest the bottom (pixel 0) is the one most inconsistent with the other shadow edge estimates. This implies some sort of systematic error, perhaps in the measurement of the shadow edge location on the image. It may also be due to a deviation in linearity of the edge of the object casting the shadow.

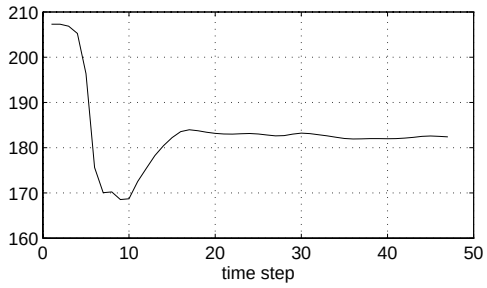
Finally, we present the three dimensional reconstruction. We collect all the estimates of the position of the shadowing half-plane edge at the last time step to form the final estimate of the shadowing half-plane edge, seen in figure 12(a) marked as "object edge". To represent the local background patch, we use all the estimates of the background normal n_p to derive the cast shadow positions r_p. Then we gather some of the cast shadows represent portions of the background patch. The reconstruction is depicted in figure 12, together with the helical light source trajectory as a reference. The reconstruction shows significant improvement over the raw surface map, although the background plane is not quite planar in shape. This is so because the cast shadow r_p depends on the state variables n_p and the reconstruction includes all r_p's in time. The larger errors are due to the errors in the early estimates. The relative position and orientation between the shadowing shalf-plane edge and the background is in accordance with what is roughly measured in the lab.



(a) convergence of $\angle n_{p\_yz}$ (°)



(b) convergence of $\angle n_{p\_zx}$ (°)



(c) convergence of $\angle n_{p\_xy}$ (°)

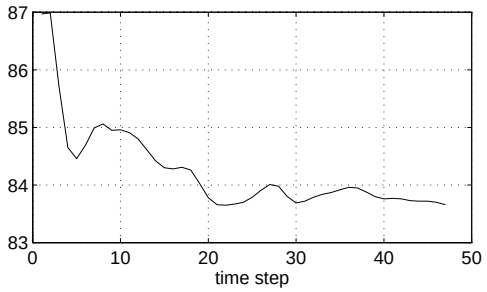
Figure 9: Convergence of the background angles for the iterated extended Kalman filter: (a) amount of rotation of n_p about $\hat{y}$ w.r.t. $\hat{z}$—"yaw", (b) amount of rotation of n_p about $\hat{z}$ w.r.t. $\hat{x}$—"roll", (c) amount of rotation of n_p about $\hat{x}$ w.r.t. $\hat{y}$—"pitch". Horizontal axes mark the time steps. Values along the vertical axes are all in degrees (°).

4.2.3 Experiment 3: Kalman Filtering with On-Line Light Source Trajectory Specification

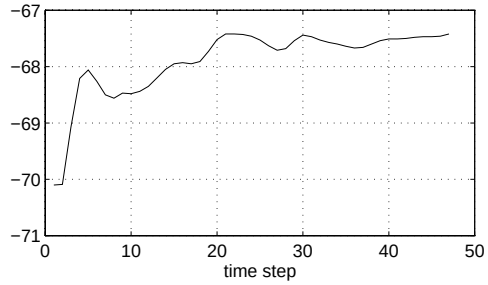
In this experiment we implement the incremental light source position specification described in section 3.3 in conjunction with the iterated extended Kalman filtering process.

The results of the experiment are summarized in the following figures. In figure 13 we show the histograms of the angles $\angle n_{p\_yz}$, $\angle n_{p\_xy}$, and $\angle n_{p\_zx}$, similarly defined as in sections 4.2.1 and 4.2.2.

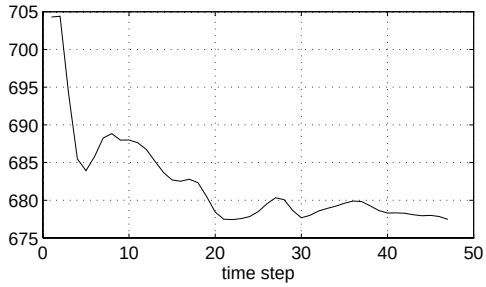
The histograms for the components of the first point on the object edge r_{q_1} are plotted in figure 14. Compare these histograms to those in the pre-defined light source trajectory case (figures 7 and 8). We see that, on average, with the on-line trajectory specification algorithm, the standard deviations of the estimated states are smaller. This implies a faster convergence rate, which is indeed the case when we examine the evolution of these state estimates in figures 15 and 16. Shown in figure 15 are the convergence rates for the angles $\angle n_{p\_yz}$, $\angle n_{p\_zx}$, and $\angle n_{p\_xy}$, and in figure 16, the convergence rates for the first point on the object edge, r_{q_1}. While in the pre-defined trajectory case it takes from 20 to 40 steps for the estimates to converge, with the on-line trajectory generation process the state estimates converge after about 6 to 7 steps. This is a significant improvement. Indeed, with the control loop which maximizes the drop in the estimate covariance we quickly come to a point where new measurements do not bring in any more useful



(a) convergence of r_{qx} (mm)



(b) convergence of r_{qy} (mm)



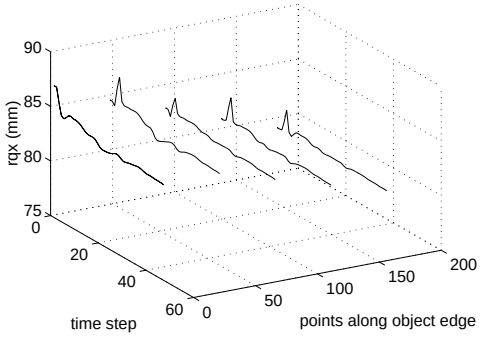
(c) convergence of r_{qz} (mm)

Figure 10: Convergence of object edge position r_{q_1} for the iterated extended Kalman filter: (a) the x-component r_{qx}, (b) the y-component r_{qy}, (c) the z-component r_{qz}. Horizontal axes mark the time steps. Values along the vertical axes are all in mm.

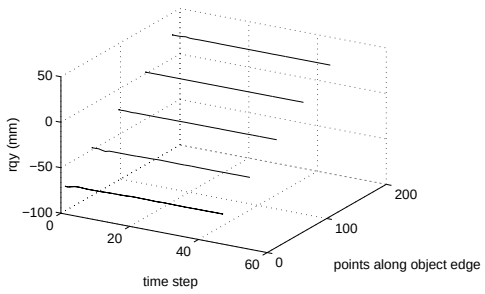
information for the estimation process to utilize. So, little change to the state estimates will be made from that point onward.

In figure 17 we render a three dimensional representation of the convergence for a few points along the object edge. While the figure shows the correct convergence for all the components, notice that different points along the object edge do converge to points along a line segment now. The three dimensional reconstruction of the partial scene is rendered in figure 18 where we again use estimates of the background normal to derive the cast shadows, and plot some of the cast shadows as portions of the background patch. Also plotted is the light source trajectory, with its starting position marked. The arrowheads mark the light source position at the end of each time step. Note that the distance between successive light source positions is fixed, at a value of 20mm. We have yet to understand the exact mechanism which produces the light source trajectory as shown in the figure, other than explaining it as the obvious result of the control scheme. Experimentally, the algorithm pays more attention to where the uncertainty in the state estimate is high, i.e., where stronger changes to the state estimate may be needed.

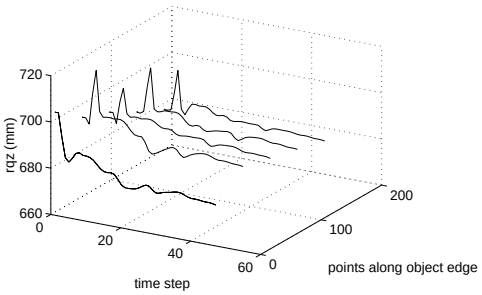
The light starts off with the same initial four locations as in the pre-defined trajectory case. When the control loop comes into effect the robot hand takes the light to regions as described in, e.g., figure 18. When comparing this with figure 12, we postulate that by going into this new region some information is gained to enable the algorithm to converge faster, and that the Kalman filtering technique may show the same behaviour if the light source in that case is allowed into the same region. This is, in fact, precisely the purpose of the control loop, to find out where the light



(a) convergence of r_{qx}



(b) convergence of r_{qy}

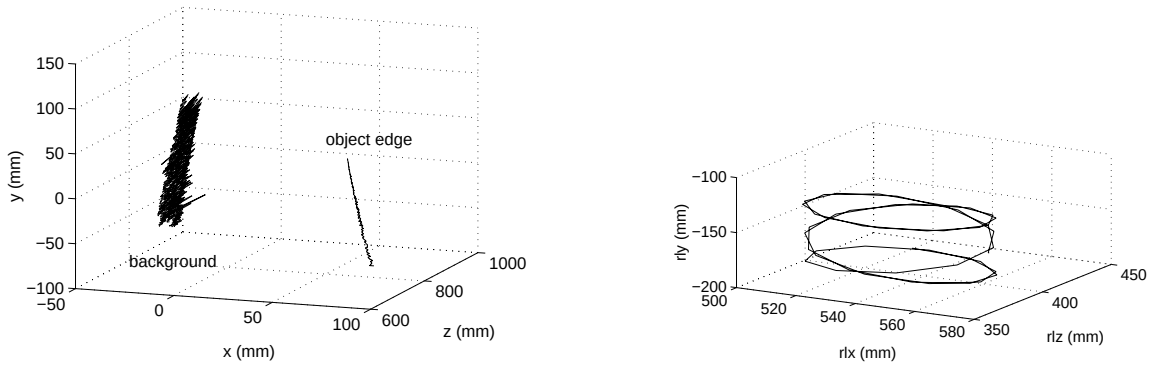


(c) convergence of r_{qz}

Figure 11: Convergence of object edge positions r_q for the iterated extended Kalman filter: (a) x-components r_{qx}, (b) y-components r_{qy}, (c) z-components r_{qz}. Values along the vertical axes are all in mm. The components in each graph converge to different values because they belong to different points along the object edge.

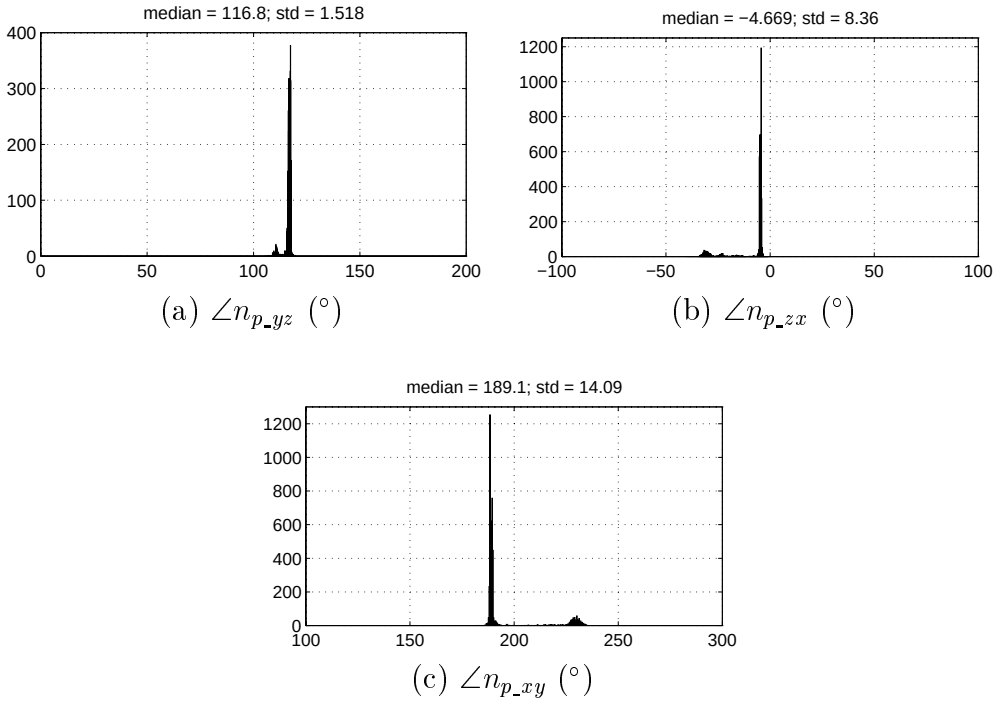
source should go for faster convergence. In looking at the light source trajectory, we see that the light moves in three distinct phases: initially the light moves towards the background plane, then moves parallel to the background plane, and then away from the background plane. In the first phase, the estimate of the gradient of the trace of the covariance matrix is poor, due to the error in the state variable estimates. Thus the initial light source motion is essentially in a random direction. In the second phase, the state variable estimates have improved enough so that an accurate estimate of the gradient can be obtained. The light moves parallel to the surface, which provides the largest displacement of the shadow edge on the image plane, thereby maximizing the signal to noise ratio. Note that, by the second stage, after about 10 time steps, the estimate has more or less converged to its final value. It is in the transition interval, between steps 5 and 10, that the influence of the light source trajectory specification is at its peak.

In figures 19 and 20 we show the histograms of the position parameters obtained for light source position step sizes of 25mm and 12.5mm respectively. It can be seen, along with figure 14, that lower step sizes produce lower error variances, implying that, at least initially, the assumption of a constant gradient in the objective function was not valid for the constraint regions implied in the larger step sizes. A much more detailed study of step-size needs to be performed, however.



(a) scene reconstruction (b) light source trajectory for the image sequence.

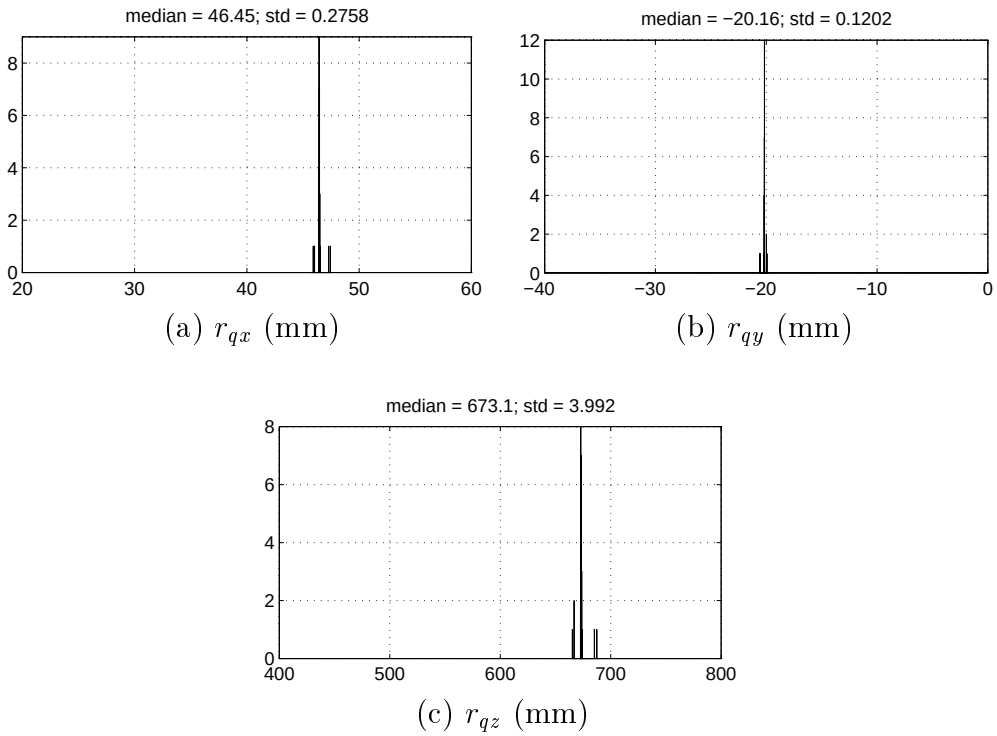
Figure 12: 3D reconstruction for the iterated extended Kalman filter: (a) one object edge map plus the cast shadow maps which make up a portion of the background, (b) the light source trajectory for the images taken in this sequence. All values along the axes are in mm.



(a) $\angle n_{p\_yz}$ (°) (b) $\angle n_{p\_zx}$ (°)

(c) $\angle n_{p\_xy}$ (°)

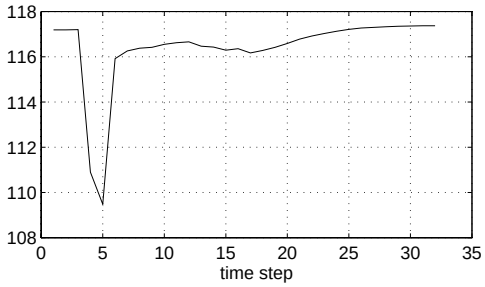
Figure 13: Histograms of background angles for on-line trajectory specification with step = 20 mm: (a) amount of rotation of n_p about $\hat{y}$ w.r.t. $\hat{z}$—"yaw", (b) amount of rotation of n_p about $\hat{z}$ w.r.t. $\hat{x}$—"roll", (c) amount of rotation of n_p about $\hat{x}$ w.r.t. $\hat{y}$—"pitch". Their respective median values and standard deviations (in °) are marked at the top of each graph. Values along the horizontal axes are all in degrees (°).



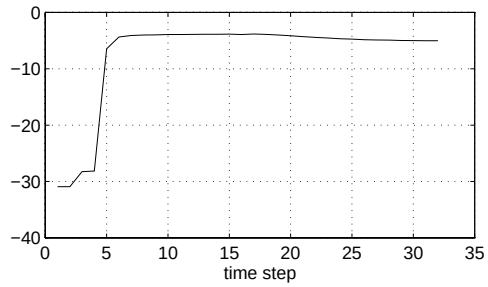
(a) r_{qx} (mm) (b) r_{qy} (mm)

(c) r_{qz} (mm)

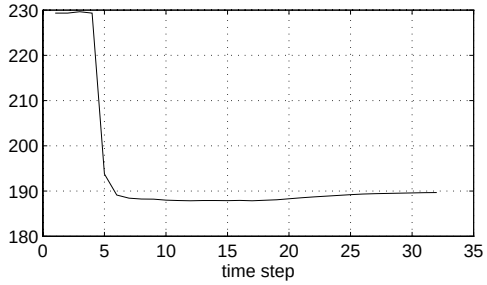
Figure 14: Histograms of object edge position r_{q_1} for on-line trajectory specification with step = 20 mm: (a) the x-component r_{qx}, (b) the y-component r_{qy}, (c) the z-component r_{qz}. Their respective median values and standard deviations (in mm) are marked at the top of each graph. Values along the horizontal axes are all in mm.



(a) convergence for $\angle n_{p\_yz}$ (°)

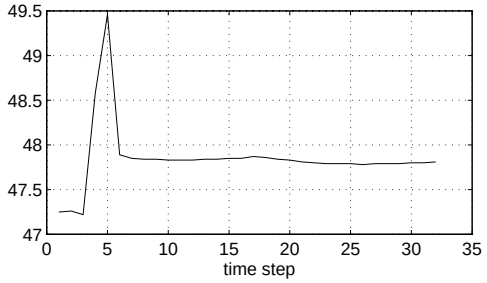


(b) convergence for $\angle n_{p\_zx}$ (°)

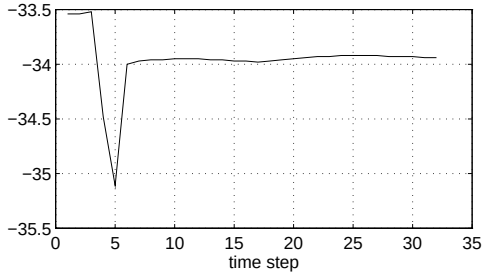


(c) convergence for $\angle n_{p\_xy}$ (°)

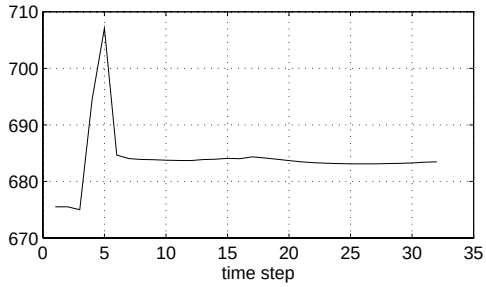
Figure 15: Convergence of the background angles for on-line trajectory specification with step = 20 mm: (a) amount of rotation of n_p about $\hat{y}$ w.r.t. $\hat{z}$—"yaw", (b) amount of rotation of n_p about $\hat{z}$ w.r.t. $\hat{x}$—"roll", (c) amount of rotation of n_p about $\hat{x}$ w.r.t. $\hat{y}$—"pitch". Horizontal axes mark the time steps. Values along the vertical axes are all in degrees (°).



(a) convergence of r_{qx} (mm)

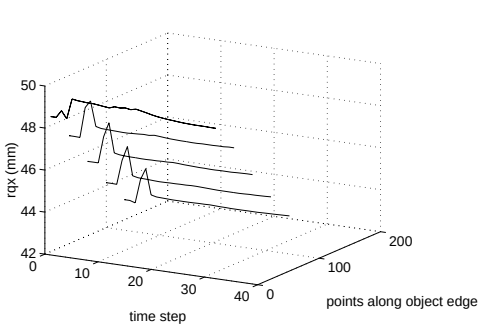


(b) convergence of r_{qy} (mm)

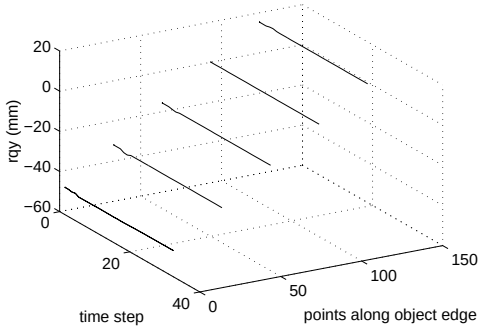


(c) convergence of r_{qz} (mm)

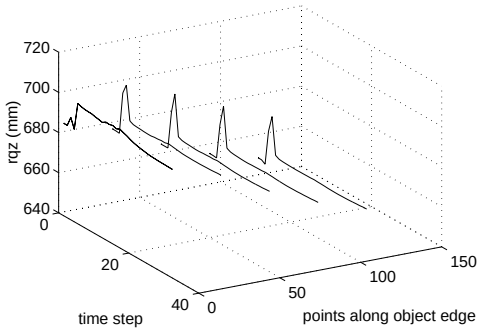
Figure 16: Convergence of object edge position r_{q_1} for on-line trajectory specification with step = 20 mm: (a) the x-component r_{qx}, (b) the y-component r_{qy}, (c) the z-component r_{qz}. Horizontal axes mark the time steps. Values along the vertical axes are all in mm.



(a) convergence of r_{qx}

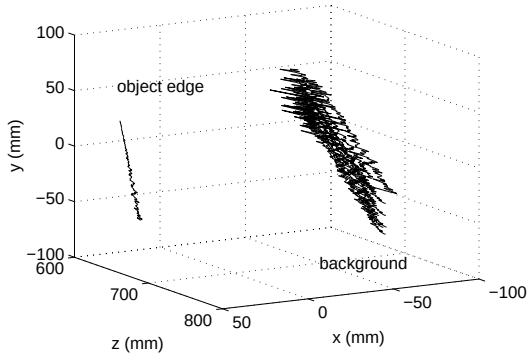


(b) convergence of r_{qy}

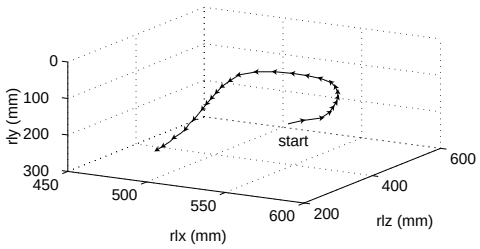


(c) convergence of r_{qz}

Figure 17: Convergence of object edge positions r_q for on-line trajectory specification with step = 20 mm: (a) x-components r_{qx}, (b) y-components r_{qy}, (c) z-components r_{qz}. Values along the vertical axes are all in mm. The components in each graph converge to different values because they belong to different points along the object edge.

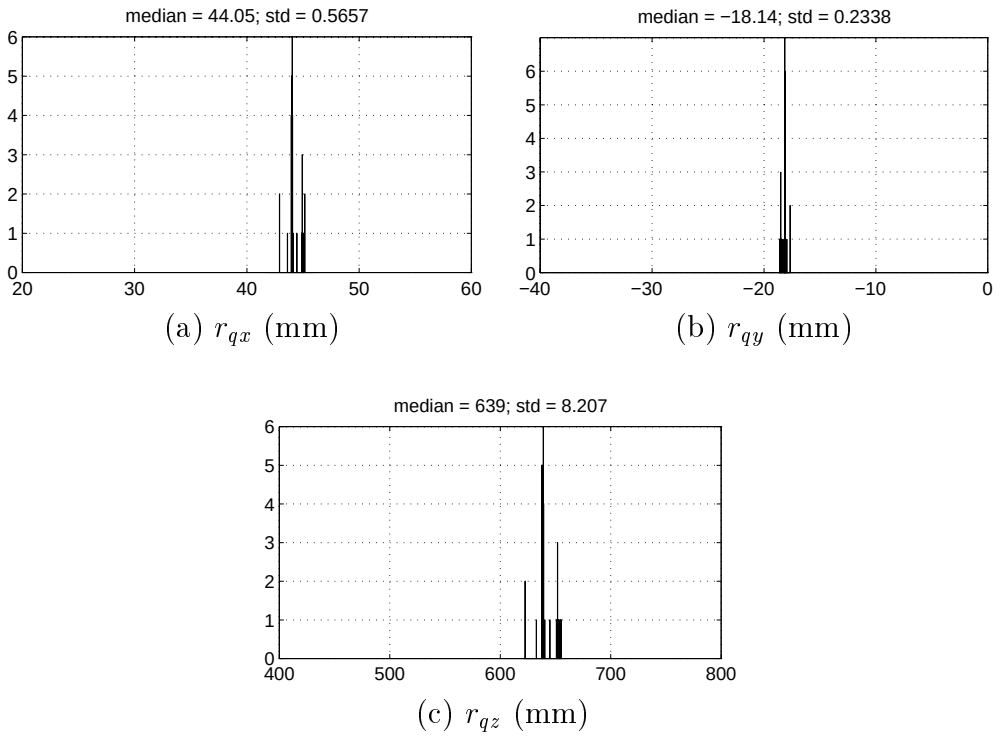


(a) scene reconstruction



(b) light source trajectory for the image sequence.

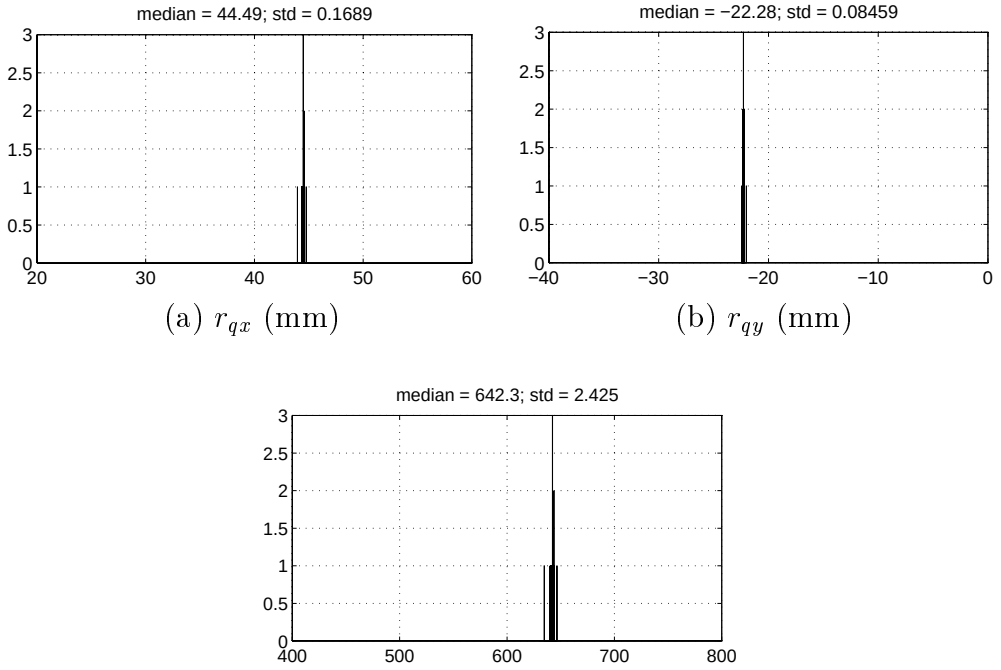
Figure 18: 3D reconstruction for on-line trajectory specification with step = 20 mm: (a) one object edge map plus the cast shadow maps which make up a portion of the background, (b) the light source trajectory for the images taken in this sequence. All values along the axes are in mm.



(a) r_{qx} (mm) (b) r_{qy} (mm)

(c) r_{qz} (mm)

Figure 19: Histograms of object edge position r_{q_1} for on-line trajectory specification with step = 25 mm: (a) the x-component r_{qx}, (b) the y-component r_{qy}, (c) the z-component r_{qz}. Their respective median values and standard deviations (in mm) are marked at the top of each graph. Values along the horizontal axes are all in mm.



(a) r_{qx} (mm) (b) r_{qy} (mm)

(c) r_{qz} (mm)

Figure 20: Histograms of object edge position r_{q_1} for on-line trajectory specification with step = 12.5 mm: (a) the x-component r_{qx}, (b) the y-component r_{qy}, (c) the z-component r_{qz}. Their respective median values and standard deviations (in mm) are marked at the top of each graph. Values along the horizontal axes are all in mm.

5 Conclusions

In this paper we have presented an active vision technique for determining the orientation and position of a planar surface onto which is cast a shadow by a half-plane, or object edge, with unknown position. This unknown position is also determined by the algorithm. The algorithm uses position control over a nearby point light source to provide the necessary information required to obtain a solution.

Using the technique proposed by Shmuel and Werman [19] we reduce the effect of measurement noise and light source position uncertainty through the use an iterated extended Kalman filter to temporally integrate measurements taken with different light source positions. Following common practice in active vision systems, we take advantage of the fact that we have control over the light source position to plan the trajectory of the light source so as to locally optimize the temporal integration process. We take an incremental approach, similar to that of Whaite and Ferrie [24] in which the next light source position is chosen in the direction of the gradient of a measure of the expected uncertainty in the solution with respect to the light source position. The difficulty in this type of approach lies in the computation of the gradient. In this paper, however, we show that, if the trace of the state estimate covariance matrix is used as a measure of uncertainty, and if a Kalman filter is used to compute the state estimate, then a closed form solution can be found for the gradient. This is a general result which can be used in other active vision applications that similarly use a Kalman filter for temporal integration and which require the computation of the gradient of uncertainty with respect to one or more control parameters.

We presented experiments using a real robotic light source position system and real-time vision processing which demonstrate the validity of the shape-from-shadows algorithm as well as the effectiveness of the on-line light source trajectory specification technique.

6 Acknowledgements

This research was supported in part by the U.S. National Science Foundation through grants IRI-92-23676 and CDR-85-00108, and by the Canadian Natural Sciences and Engineering Research Council.

References

[1] Aloimonos, Y., Weiss, I. and Bandyhopadhyay, A., "Active vision", *International Journal of Computer Vision*, No. 2, pp 333-356, 1988

[2] Bajcsy, R., "Active perception", *Proceedings of the IEEE*, Vol. 76, No. 8, pp 996-1005, 1988

[3] Chaumette, F. and Boukir, S., "Structure from motion using an active vision paradigm", *Proc. 11th Int. Conf. Pattern Recognition*, The Hague, Netherlands, 1991

[4] Clark, J.J., "Active Photometric Stereo", in *Proceedings of the 1992 IEEE Computer Vision and Pattern Recognition Conference*, Champaign, IL, pp 29-35, June 1992

[5] Clark, J.J. and Yuille, A.L., **Data Fusion for Sensory Information Processing Systems**, Kluwer Academic Publishers, Boston, MA, 1990

[6] Durrant-White, H.F., **Integration, Coordination, and Control of Multi-Sensor Robot Systems**, Kluwer Academic Publishers, Boston, MA, 1988

[7] Faugeras, O., Ayache, N., and Faverjon, B., "Building visual maps by combining noisy stereo measurements", Proceedings of the 1986 *IEEE International Conference on Robotics and Automation*, pp 1433-1438, 1986

[8] Gelb, A. (ed.), **Applied Optimal Estimation,** MIT Press, Cambridge, MA, 1974

[9] Harris, C., "Tracking with rigid models" in **Active Vision**, Blake, A. and Yuille, A., eds., MIT Press, Cambridge, MA, 1992

[10] Kender, J. and Smith, E., "Shape from darkness: Deriving surface information from dynamic shadows", *Proceedings of AAAI*, pp 539-546, 1987

[11] Kirlin, R.L., and Moghaddamjoo, "Robust adaptive Kalman filtering for systems with unknown step inputs and non-Gaussian measurement errors", *IEEE Transactions on ASSP*, Vol. 34, No. 2, pp 252-263, 1986

[12] Langer, M.S., Dudek, G., and Zucker, S.W., "Space occupancy using shadow images", Proceedings of International Robotics Symposium (IROS), 1995

[13] Martinez, J.M. and Montano, L., "A camera motion strategy to localize uncertain 3D lines", *Proc. 1993 IEEE Int. Conf. on Systems, Man and Cybernetics*, Le Touquet France, pp 517-522, 1993

[14] Matthies, L.H., Kanade, T., and Szeliski, R., "Kalman filter-based algorithms for estimating depth from image sequences", in *International Journal of Computer Vision*, Vol. 3, pp 209-236, 1989

[15] Meier, L., Peschon, J., and Dressler, R.M., "Optimal control of measurement subsystems", *IEEE Transactions on Automatic Control'*, Vol. AC-12, No. 5, pp 528-536, 1967

[16] Oshima, M. and Shirai, Y., "Object recognition using 3-D information", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5, No. 4, pp 353-361, 1983

[17] Raviv, D., Pao, Y. and Loparo, K., "Reconstruction of three-dimensional surfaces from two-dimensional binary images", *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 5, pp 701-710, 1989

[18] Shirai, Y., "Recognition of polyhedrons with a range finder", *Pattern Recognition*, Vol. 4, pp. 243-250, 1972

[19] Shmuel, A. and Werman, M., "Active vision: 3D from an image sequence", Proceedings of the 10th International Conference on Pattern Recognition, pp 48-54, June 1990

[20] Tarabanis, Konstantinos A. Allen, Peter K. Tsai, Roger Y. A survey of sensor planning in computer vision, *IEEE Transactions on Robotics and Automation*, v. 11, Feb. 1995, p. 86-104

[21] Terzopoulos, D. and Szeliski, R., "Tracking with Kalman snakes", in **Active Vision**, Blake, A. and Yuille, A., eds., MIT Press, Cambridge, MA 1992

[22] Waltz, D., "Understanding line drawings of scenes with shadows", in *The Psychology of Computer Vision*, P.H. Winston, ed., McGraw-Hill, New York, pp 19-91, 1983

[23] Wang, L., **3D Structure from Active Shadowing**, Ph.D. Thesis, Division of Applied Sciences, Harvard University, 1995

[24] Whaite, P, and Ferrie, F.P., "Autonomous exploration: driven by uncertainty", *Proceedings of 1994 IEEE Conference on Computer Vision and Pattern Recognition*, Seattle WA, pp 339-346, June 1994

[25] Wu, W.R. and Kundu, A., "Recursive filtering with non-Gaussian noises", *IEEE Transactions on Signal Processing*, Vol. 44, No. 5, June 1996, pp 1454-1469

Appendix - Derivation of the Gradient of the Trace of the State Estimate Covariance Matrix

In this appendix we derive an expression for the gradient of $\epsilon(\vec{u})$ with respect to the components of the control parameter vector $\vec{u}$ (equation (32) in section 3). This derivation was originally presented in a conference paper [4]. We repeat it here for completeness and ease of reference.

It suffices to find a general form for the individual gradient components, so we will determine the partial derivative of $\epsilon(\vec{u})$ with respect to an arbitrary component u_i. In the derivation of the expression for the gradient we will use the following matrix differentiation identities:

Let $X = X(t)$ and $Y = Y(t)$ be matrix functions of a scalar t. Then

$$\frac{\partial}{\partial t} tr[X^T Y X] = tr\left[\frac{\partial}{\partial t}(X^T Y X)\right] \tag{33}$$

Let $X(t) = Y^{-1}(t)$ be a symmetric matrix function of a scalar t. Then

$$\frac{\partial X}{\partial t} = -X\left(\frac{\partial Y}{\partial t}\right)X \tag{34}$$

From equation (19) we have that:

$$\frac{\partial \epsilon}{\partial u_i} = \frac{\partial}{\partial u_i} tr[A^T B A] \tag{35}$$

Using the first of the matrix differentiation identities this can be written as:

$$\frac{\partial \epsilon}{\partial u_i} = tr\left[\frac{\partial A^T}{\partial u_i}(BA) + A^T \frac{\partial B}{\partial u_i} A + A^T B \frac{\partial A}{\partial u_i}\right] \tag{36}$$

From the definition of the matrix A we have that:

$$\frac{\partial A}{\partial u_i} = \frac{\partial H}{\partial u_i} P_{k-1} W \tag{37}$$

(since P_{k-1} and W do not depend on the new value of the control vector $\vec{u}$). Using the second identity we can write:

$$\frac{\partial B}{\partial u_i} = -B \left(\frac{\partial H}{\partial u_i} P_{k-1} H^T + H P_{k-1} \frac{\partial H^T}{\partial u_i} + \frac{\partial R}{\partial u_i} \right) B \tag{38}$$

Expanding equation (28), using the linearity of the trace operator, we get

$$\begin{array}{l} \frac{\partial \epsilon}{\partial u_i} = tr\left[W P_{k-1} \frac{\partial H^T}{\partial u_i} B A \right] - tr\left[A^T B \frac{\partial H}{\partial u_i} P_{k-1} H^T B A \right] - \\ tr\left[A^T B H P_{k-1} \frac{\partial H^T}{\partial u_i} B A \right] + tr\left[A^T B \frac{\partial H}{\partial u_i} P_{k-1} W \right] - tr\left[A^T B \frac{\partial R}{\partial u_i} B A \right] \end{array} \tag{39}$$

Since the trace of a matrix is equal to the trace of its transpose, the above equation can be written as

$$\frac{\partial \epsilon}{\partial u_i} = 2tr\left[A^T B \frac{\partial H}{\partial u_i} P_{k-1} W - A^T B \frac{\partial H}{\partial u_i} P_{k-1} H^T B A \right] - tr\left[A^T B \frac{\partial R}{\partial u_i} B A \right] \tag{40}$$

Using $A^T B = W K$, we get

$$\frac{\partial \epsilon}{\partial u_i} = 2tr\left[W K \frac{\partial H}{\partial u_i} P_{k-1} (I - H^T K^T) W \right] - tr\left[W K \frac{\partial R}{\partial u_i} K^T W \right] \tag{41}$$

Note that $P_{k-1}(I - H^T K^T) = (I - KH) P_{k-1} = P_k$, the update to the covariance matrix. Thus we can write:

$$\frac{\partial \epsilon}{\partial u_i} = tr\left[W K \left(2 \frac{\partial H}{\partial u_i} P_k - \frac{\partial R}{\partial u_i} K^T \right) W \right] \tag{42}$$

Usually R depends on u only indirectly, through the influence of u on the measurement function h. So we use

$$\frac{\partial R}{\partial u_i} = \frac{\partial R}{\partial h} \frac{\partial h}{\partial u_i} \tag{43}$$

in the above equation. The quantity $\frac{\partial R}{\partial h}$ is obtained from a model for how the measurement noise varies with the measured value.