Real-time tracking of surfaces with structured light

Peter Lindsey and Andrew Blake,
Robotics Research group, Dept of Engineering Science,
Oxford university.

Abstract

We detail the development of deformable surface models to track non–rigid objects undergoing motion, using real time range data acquired from our sensor. The surface model for an object consists of B-spline tensor product patchs, augmented with a dynamic and measurement model. To maximise performance, careful consideration is made of where to make the next range measurements, based on the current uncertainty of the model. By calculating the reduction in uncertainty that each measurement makes, the optimal sampling position at every step can be found.

1 Introduction

A considerable amount of research has recently focused on active vision, and in particular dynamic contours. These deformable model curves are capable of tracking objects in images, using models of the behaviour of the target that can be learnt over time. This work was pioneered by Hogg (1983) with his "walker" program, and rekindled by the seminal work of Kass et al (1987). These "snakes" provided enormous potential for dynamic applications. Good dynamic behaviour is obtained by supplementing the parametric shape model and elastic attraction to data, with dynamical properties in the form of distributed mass and viscosity which interact with intrinsic elastic shape memory. This is best expressed in the context of statistical estimation theory, following the pioneering work of Hallam (1983) in uncertainty models for sonar sensing, and developed for inelastic parametric models by Harris (1990). Statistical models based on a standard framework (Gelb 1974) for elastic trackers have been developed by Curwen and Blake (1992), and Terzopoulos and Metaxas (1991).

An equivalent to the contour tracking problem in images is the tracking of non–rigid object motion in 3–D. This has previously been investigated using deformable superquadrics (Terzopoulos, 1992) and articulated structures (Rehg and Kanade, 1994). Our approach uses a partial surface model of the object, and tracks the visible parts of the surface; for this, we need to obtain measurements of the moving surface in real time. The range sensor developed at Oxford is capable of measuring moving scenes, by using only a single image to measure depth unambiguously, even in the presence of occlusion. This sensing process is tightly coupled to our model, as the calculation of range data over the whole image is costly in terms of processing

BMVC 1994 doi:10.5244/C.8.61

power. In the absence of elaborate parallel hardware, it is crucial for real-time performance to develop methods of predictively and selectively measuring position at a relatively few regions each frame. These measurements should be chosen to be maximally informative on the basis of the previous frame's estimates. The surface models we have developed are capable of tracking simple objects moving under the sensor (figure 1). The aim is to apply this to the tracking of a flexing hand.

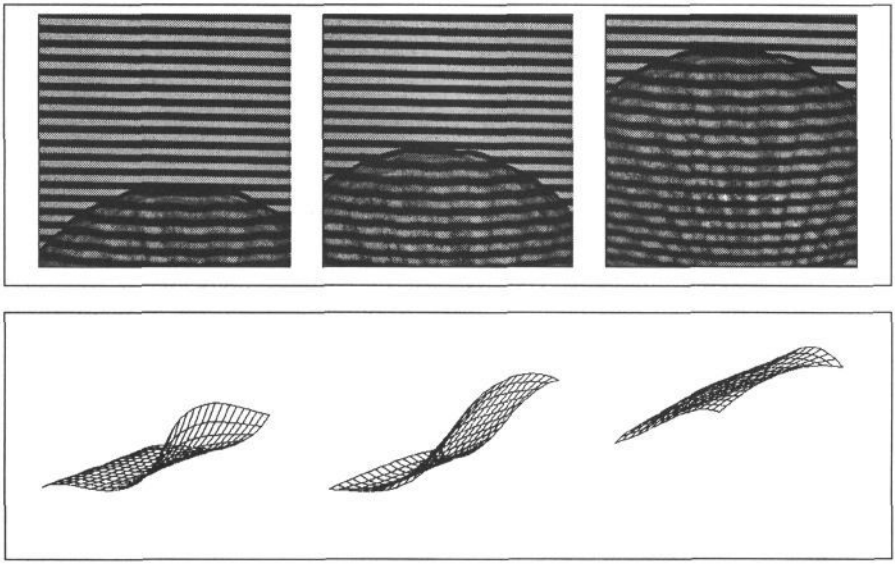


Figure 1: *The contour flows up the side of the shell as it passes across the sensor.*

2 Real time range sensing

There are a number of structured lighting principles currently used for active range finding. The most common are the single stripe systems; these scan a light stripe across the scene, processing a number of images of the stripe at different positions. This technique does not allow for objects that can move, however, since the object will be in a different position as each stripe is imaged. Sensors which use area filling patterns, such as a set of stripes, can overcome this drawback but need a system of stripe identification to deal with the correspondence problem. Several methods attempt to solve this problem by intrinsicly labeling the stripes, using colour (Boyer and Kak, 1987), space, or thickness encoding (Posdamer and Altschuler, 1982). However, these methods are susceptible to coding corruption from surface properties in the scene, which can result in mislabeling of the stripes.

Our approach is to use two sets of stripes, effectively forming a grid pattern. To eliminate ambiguity, it turns out that the orientation of this grid is critical. Rather than setting the grid at an arbitrary angle with respect to the epipolar lines (whose epipole is the projection of the camera's optical centre), it is set to be almost parallel. This is termed *near-degenerate epipolar alignment* and greatly reduces the number of candidate solutions (Blake et al, 1992). If a working volume

is imposed as a bounding box then it can be shown that ambiguity is altogether removed; only a single image is required for range sensing, and a moving object can be "frozen" by a single frame. An example of a range scan of a pipe is shown in figure 2. The accuracy of the sensor has been shown to be about one one part in 500 of its working volume e.g. 0.1 mm for a working volume height of 5cm.

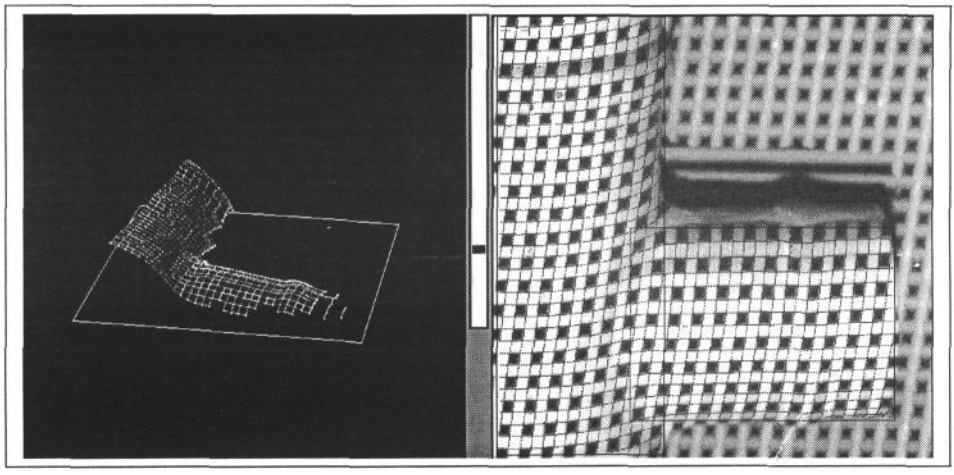


Figure 2: *The range sensor output for a piece of ceramic piping.*

In fact, by making use of a surface tracker, the correspondence problem for a single stripe set can be solved, using back–projection from the image onto the model to disambiguate the stripes. This means that more accurate edge detectors can be used, rather than the corner detectors required for the grid. The grid pattern need only be used for the initialisation of the model.

3 Surface modeling

The surface description is critical for real time tracking; it must be as compact as possible, and cheap to compute. The existing real–time contour tracking software developed in Oxford uses B–splines as the parametric representation for contours. A natural and efficient extension to this work are the B–spline tensor–product patches to represent surfaces (Bartels, Beatty, and Barsky 1987). These have the same desirable properties as the curves, namely compact representation, simple control over continuity and local effect of control points.

A B–spline surface can be described as a tensor product of two B–spline curves

$$\mathbf{s}(u,v) = \sum_i \sum_j \mathbf{c}_{i,j}\ B_i(u)\ B_j(v)$$

where $B_k(u)$ are the B-spline basis functions. This is conventionally written in matrix form as $\mathbf{s}(u,v) = M(u)\ C\ N(v)^T$.

Alternatively, a polymorphic subscript can be used, for example $\mathbf{n}$, referring to either a single index value, n, or a tuple (i_n, j_n). The expression for the surface

can then be written with the control point matrix transformed to a vector:

$$\mathbf{s}(u,v) = H_{\mathbf{n}}\, C_{\mathbf{n}} \tag{1}$$

where the repeated subscript indicates summation over the range of that subscript, and $H_{\mathbf{n}} = B_i(u)\, B_j(v)$.

This notation can be used to calculate the metric matrix, $\mathcal{H}$, which defines a norm for the B–spline surfaces, that is $\|C\|^2 = C_{\mathbf{m}}\mathcal{H}_{\mathbf{m},\mathbf{n}}C_{\mathbf{n}}$:

$$\mathcal{H} = \int_0^N \int_0^M H(u,v)^T\, H(u,v)\, du\, dv$$

The elements of this metric matrix have values

$$\begin{aligned} \mathcal{H}_{\mathbf{m},\mathbf{n}} &= \int_0^N \int_0^M H_{\mathbf{m}}.H_{\mathbf{n}}\, du\, dv \\ &= \mathcal{h}^u_{i_m,i_n}\, \mathcal{h}^v_{j_m,j_n} \end{aligned}$$

where $\mathcal{h}^u$ and $\mathcal{h}^v$ are the equivalent metric matrices for the univariate curves.

If enough points can be acquired in a time step to fully determine the model, the surface can be fitted to the data using standard least squares. If the approximation to the the surface is $\underline{\mathbf{s}}(u,v) = H(u,v)\,\underline{C}$, then the solution to $\underline{C}$ is

$$\underline{C} = \mathcal{H}^{-1} \sum_i H(u_i,v_i)^T.\mathbf{p}_i$$

where $\mathbf{p}_i$ are the range points sampling $\mathbf{s}(u_i,v_i)$.

The least squares approach is used for initialisation, and simple surface models, where the number of points sampled per frame is great enough to make the fit well–conditioned.

3.1 The dynamic model

Normally there is not enough time at each step to measure sufficient points to fully determine the surface. In this case, the surface model is augmented with dynamics.

The dynamic system can be modeled as a linear state transition equation:

$$\mathcal{X}(k+1) = \mathbf{F}(k)\mathcal{X}(k) + v(k).$$

where the system noise $v(k)$ is gaussian, mean-zero, and temporally un-correlated. For a first order filter, the state vector will comprise the position and velocity of the control points, $\mathcal{X} = \begin{bmatrix} C \\ \dot{C} \end{bmatrix}$, and have covariance P. The state transition matrix for this first order filter is then $\mathbf{F} = \begin{bmatrix} I & \Delta t\, I \\ 0 & I \end{bmatrix}$. This matrix defines the predicted state of the model from the previous time step, $\mathcal{X}(k) = \mathbf{F}(k-1)\mathcal{X}(k-1)$.

The observation model describes how each measurement is incorporated into the filter:

$$z(k) = H(k)\mathcal{X}(k) + w(k).$$

where $w(k)$ is the measurement noise with covariance R, and $H(k) = \begin{pmatrix} H(u,v) & 0 \end{pmatrix}$. The estimate of the state after an observation, $\mathcal{X}_+$ is then

$$\mathcal{X}_+(k) = \mathcal{X}(k) + K(k)[z(k) - H(k)\mathcal{X}(k)]$$

where $K(k)$ is the Kalman gain. At each time step, the state of the surface is predicted from the previous state, and a number of range measurements are taken. These are incorporated into the filter to give the current position and velocity of the surface.

4 Measuring uncertainty

A consequence of basing the filter on a model of statistical uncertainty is that there is potential for a natural, automatic mechanism for control of search scale and memory duration. Given linear dynamics, a Riccati equation (Bar–Shalom and Fortmann 1988) governs the covariance of the current state of the filter and, the "validation gate" mechanism uses this covariance to determine a natural spatial–scale for feature–search. The result is that a limited volume is defined around each estimated surface point in which it is reasonable to search for surface data. These volumes should shrink as data is accumulated, reaching a minimum size in the steady–state, or grow in the absence of data until new data is captured. Theory predicts accompanying temporal variations, the tracker's "memory" being longest when spatial scales are smallest. This mechanism can be used to select which regions in the range image to observe to maximise the gain in information, based on the current state.

4.1 The prior probability distribution

The uncertainty of the model at a given time is described by the covariance P of the state vector $\mathcal{X}$. The variance ρ^2 of a point on the surface is then

$$\rho^2(u,v) = H(u,v)\ P\ H(u,v)^T. \tag{2}$$

where $H(u,v)$ has now been augmented with zeros.

The prior probability distribution for a surface is chosen as being uniform, and can be considered as a gaussian distribution in a family of continuous surfaces. The covariance of the surface is then homogeneous and spatially independent, and $P = \alpha\mathcal{H}^{-1}$ (Curwen and Blake 92).

When the surface is described as a set of piecewise continuous surfaces, however, this ideal p.d.f. is restricted to the basis function representation, and the covariance of the surface is no longer uniform. This can be seen in figure 3 for the positional variance $\rho^2(u)$ of a five span cubic B-spline curve, with covariance as above. There is a fluctuation across the curve, where the variance is higher at the knot points (since the continuity of the curve is only C^2 at the knots). Also, the variance is much higher at the endpoints, where there are multiple knots (i.e. C^0 continuity).

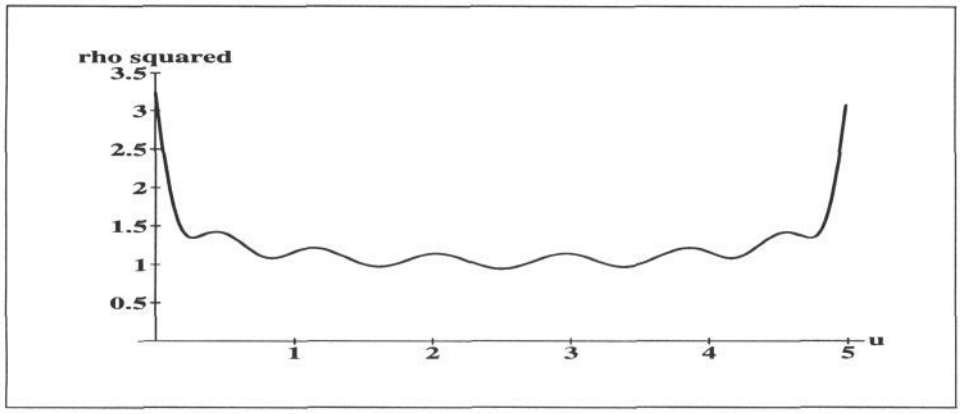


Figure 3: *The a priori positional variance $\rho^2(u)$ for a five span B-spline curve.*

4.2 The effect of a measurement

To analyse the relationship between positional variance and optimal measurements, we will first observe how the covariance of the state changes as measurements are made, using the Kalman filter update equations. The model is set up with no dynamics and the simplifying assumption that there is no state noise, so that the covariance of the state P does not change over the prediction phase of the filter. The Kalman filter then behaves as a recursive least squares estimator, and the effect of positional variance can be observed.

Given an observation of a point at $\hat{\mathbf{s}} = (\hat{u}, \hat{v})$, the equations for the covariance P_+ and the Kalman gain K after the observation are (assuming we are looking at one component of $\mathcal{X}$, and the noise process is independent, so $R = \sigma^2$):

$$\begin{aligned} K &= P\, H(\hat{\mathbf{s}})^T\, [H(\hat{\mathbf{s}})\, P\, H(\hat{\mathbf{s}})^T + R]^{-1} \\ &= \frac{1}{\rho^2(\hat{\mathbf{s}}) + \sigma^2}\, P\, H(\hat{\mathbf{s}})^T \end{aligned} \tag{3}$$

$$P_+ = [I - K\, H(\hat{\mathbf{s}})]\, P \tag{4}$$

Combining (3) and (4), the change in the covariance resulting from a measurement at $\hat{\mathbf{s}}$ is

$$\Delta P(\hat{\mathbf{s}}) = -\frac{1}{\rho^2(\hat{\mathbf{s}}) + \sigma^2}\, P\, H(\hat{\mathbf{s}})^T\, H(\hat{\mathbf{s}})\, P \tag{5}$$

The local nature of the B–spline basis functions mean that the reduction of the covariance is restricted to a neighbourhood of control points around the measurement point, i.e. the effect of the measurement is also local.

4.3 Defining an information metric

Having calculated the change in covariance from the observation of a point, we need to quantify just what we mean by the "best" improvement in information,

in order to specify the optimal place for an observation. To do this we define an information metric as the mean value of the positional variance across the surface.

$$\begin{aligned} \overline{\rho^2} &= \int_A \rho^2(u,v)\, du\, dv \\ &= tr\,(\,P\,\mathcal{H}\,) \end{aligned}$$

So, using the change in covariance calculated from (5), the change in $\overline{\rho^2}$ after a measurement of a point on the surface at $\hat{\mathbf{s}}$ gives:

$$\Delta\overline{\rho^2}(\hat{\mathbf{s}}) = -\frac{1}{(\rho^2(\hat{\mathbf{s}}) + \sigma^2)} \parallel H(\hat{\mathbf{s}})P \parallel^2 \tag{6}$$

The optimal measurement is the one that maximises the value of $\Delta\overline{\rho^2}(\hat{\mathbf{s}})$.

4.4 Finding the optimal measurement

It would be convenient to express $\Delta\overline{\rho^2}(\hat{\mathbf{s}})$ in a simpler form involving $\rho^2(\mathbf{s})$ – indeed, it seems intuitive that the best place to take a measurement would be the place of highest positional uncertainty (this volume of uncertainty approach has been used before for guiding visual exploration (Waite & Ferrie, 1991)). However, doing so reveals the rather suprising result that $\rho^2(\mathbf{s})$ is actually *not* a good indicator.

To show this, consider a state covariance consisting of a uniform component plus some perturbation, so that $P = P_s + \epsilon P'$, where P_s is the uniform distribution $P_s = \alpha\mathcal{H}^{-1}$. The positional variance of the surface, $\rho^2(\mathbf{s})$ is, from (2)

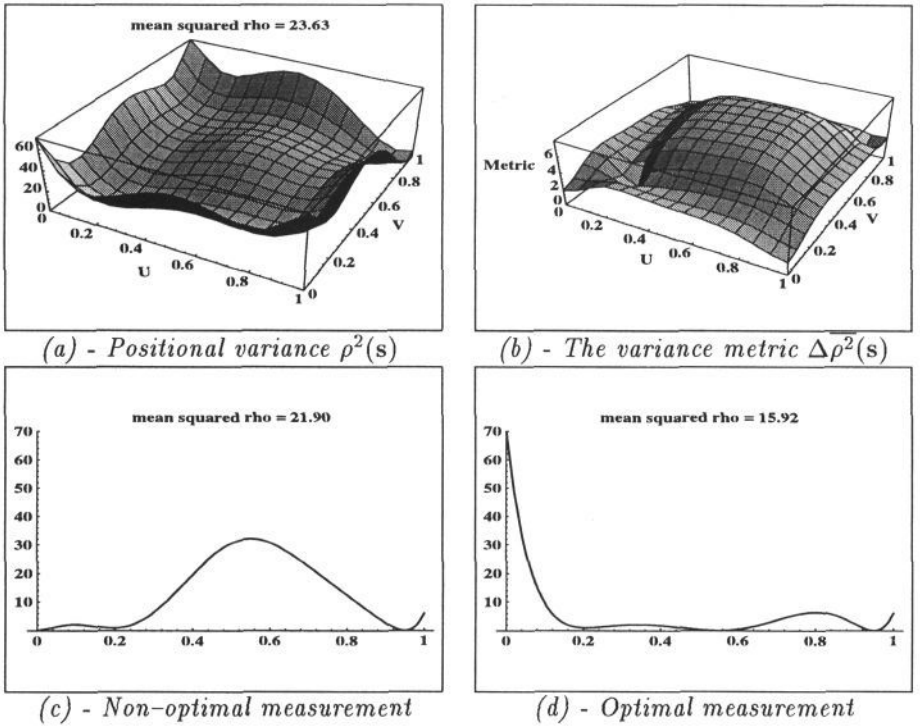
$$\begin{aligned} \rho^2(\mathbf{s}) &= H(\mathbf{s})\,(P_s + \epsilon P')\,H(\mathbf{s})^T \\ &= \rho_s^2(\mathbf{s}) + \epsilon\rho'^2(\mathbf{s}) \end{aligned} \tag{7}$$

Combining (6) and (7) for small ϵ, gives the metric,

$$\Delta\overline{\rho^2}(\hat{\mathbf{s}}) = -\frac{\alpha\,(\rho^2(\hat{\mathbf{s}}) + \epsilon\rho'^2(\hat{\mathbf{s}}))}{(\rho^2(\hat{\mathbf{s}}) + \sigma^2)}$$

It can be seen from this that the metric is only proportional to the positional variance in the case where ϵ tends to zero, i.e. where the uncertainty of the surface is uniform! This is not a particularly useful result, although it does show that when the measurement noise is much smaller than the positional variance, the metric $\Delta\overline{\rho^2}(\hat{\mathbf{s}})$ is constant i.e. it doesn't matter where you measure. Generally, however, the optimal measurement *cannot* be found just from observing the positional variance. This is illustrated in figure 4.

The problem with this information metric is that it's more expensive to compute than the positional variance. It may turn out that the overall gain from taking optimal measurements is outweighed by the reduction in the number of points measured at each time step; more, sub–optimal measurements may be better. This is currently being explored.



(a) - Positional variance $\rho^2(s)$ *(b) - The variance metric $\Delta\rho^2(s)$*

(c) - Non-optimal measurement *(d) - Optimal measurement*

Figure 4: The best place to make a measurement. *(a) shows the positional variance of a surface after several measurements have been made. Based on this, the best place to take a measurement would be at the corners, since these are the areas of highest positional variance. In fact, the variance metric, shown in (b) indicates that the optimal place is near the centre. Diagonal cross-sections of the positional variance after a measurement are shown in (c) and (d). The first shows a non-optimal measurement at the corner $u = v = 0$, the second the optimal measurement, based on the value of the metric. The decrease in $\overline{\rho^2}$ is much greater in the latter case.*

5 Results from the surface tracker

The current equipment setup consists of the range sensor connected to a CRS 1000 frame store, which in turn connects to the VME backplane of a SUN 4/110, acting as an image server. To achieve faster processing, another machine on the Ethernet is used for the image and tracking processing, normally a Silicon Graphics Indigo. This setup will run at about frame rate, with some variability due to network traffic and machine load (none of the machines involved run real time operating systems). Even with just a simple zero order (i.e. position based) model, it is capable of tracking objects with velocities of up to 10cm/sec. This will be further improved with the implementation of refined object model dynamics.

Figure 5 shows another example of real time data of a hand being tracked.

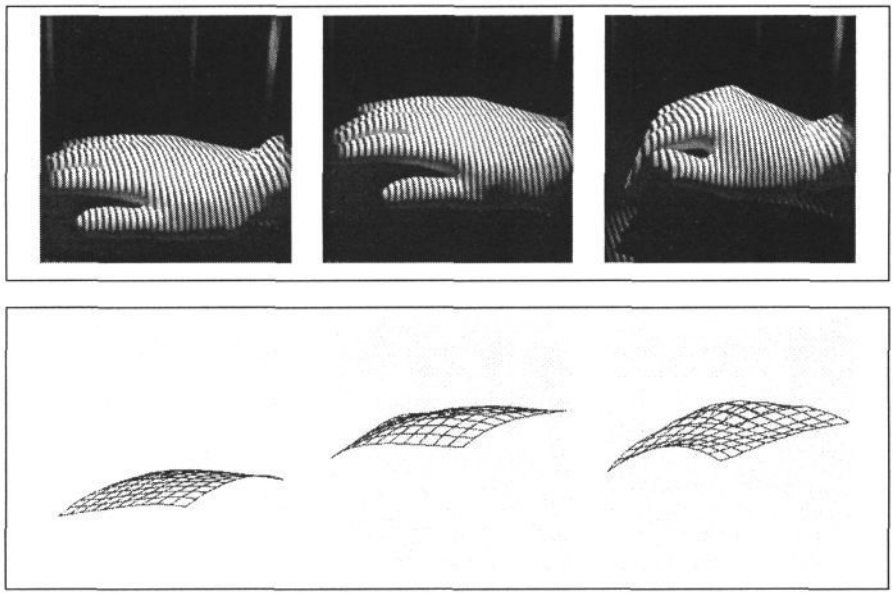


Figure 5: Hand tracking. *By treating the hand as a single continuous surface, it is possible to track the motion using just the simple surface tracker. Seen from the side, the hand moves under the sensor, and from a different viewpoint, the tracker follows the motion as it first moves up, and then flexes. To allow for individual articulations of the fingers, a more complex model would be required.*

6 Future work

For simple objects, tensor product patchs satisfy our requirements for a surface model. For more complex models such as a hand, a union of surface patchs could be used to model the fingers, but it is not clear that this will be the best representation to use for a number of reasons. Other surface descriptions are being investigated such as Bezier triangles and multivariate B–spline basis functions (Farin 1990) that will allow a variable topology for modeling "tears" in a surface.

A more ambitious aim, given the basis in statistical uncertainty, is to learn uncertainty models from motion sequences. The tracker itself would be used to do this. The process could be bootstrapped with a hand–built tracker, using "natural" default values for uncertainties. Once the tracker works roughly, following data sequences, the aim would be to estimate principal components of shape variability between frames (Grenander et al., 1991, Cootes and Taylor, 1992). This information could be incorporated into the tracker to improve its dynamics — that is, to make it faster and more selective for shape and motion.

7 Conclusion

In this paper we have presented a novel, non–rigid tracker, using range data gathered in real time. For computational efficiency, careful consideration is made of

where to take measurements of the surface,using the statistical model that the tracker is based on. By defining a metric on the positional variance of our surface model, we have shown where the next place to look should be, based on our current uncertainty. This allows us to maximise the information gained at each time step. This work provides a promising foundation for the development of more complex trackers.

References

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

[2] R.H. Bartels, J.C. Beatty, and B.A. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling.* Morgan Kaufmann, 1987.

[3] A. Blake, H.R. Lo, D.McCowen, and P.J. Lindsey. Trinocular active range sensing. *IEEE Trans. Pattern Analysis and Machine Intell.*, in press, 1992.

[4] K.L. Boyer and A.C. Kak. Color-coded structured light for rapid image ranging. *IEEE Trans. PAMI*, 1:14–28, 1987.

[5] T.F. Cootes and C.J. Taylor. Active shape models. In *Proc. BMVC*, pages 265–275, 1992.

[6] R. Curwen and A. Blake. Dynamic contours: real-time active splines. In A. Blake and A. Yuille, editors, *Active Vision*, pages 39–58. MIT, 1992.

[7] G.E. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A practical Guide.* Academic Press, 1990.

[8] Arthur Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge, MA, 1974.

[9] U. Grenander, Y. Chow, and D. M. Keenan. *HANDS. A Pattern Theoretical Study of Biological Shapes.* Springer-Verlag. New York, 1991.

[10] J. Hallam. Resolving observer motion by object tracking. In *Procs. of 8th International Joint Conference on Artificial Intelligence*, volume 2, pages 792–798, 1983.

[11] C.G. Harris and C. Stennett. Rapid – a video-rate object tracker. In *Proc. 1st British Machine Vision Conference*, pages 73–78, 1990.

[12] D. Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[13] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, pages 259–268, 1987.

[14] J.L. Posdamer and M.D. Altschuler. Surface measurement by space encoded projected beam systems. *Computer Graphics and Image Processing*, 18:1–17, 1982.

[15] D. Terzopoulos. Tracking nonrigid 3D objects. In A. Blake and A. Yuille, editors, *Active Vision*, pages 75–89. MIT, 1992.

[16] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: deformable superquadrics. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7), 1991.

[17] P Whaite and F.P. Ferrie. From uncertainty to visual exploration. *IEEE Trans. Pattern Analysis and Machine Intell.*, 13(10):1038–1049, 1991.