Hand tracking for vision-based drawing

M. Isard and J. MacCormick,
Department of Engineering Science,
University of Oxford

Abstract

This report should be regarded as an appendix to our paper [9]. It provides technical details on the hand tracking system described there, and on the vision-based drawing package developed to demonstrate its use in applications.

1 The hand tracker

This section describes the implementation of a robust and accurate hand tracker using ICondensation [5] and partitioned sampling [9].

1.1 State space

A second-order state-space is used to represent the hand, giving

$$\mathbf{X}_t = \begin{pmatrix} \mathbf{x}_t \\ \mathbf{x}_{t-1} \end{pmatrix} \quad \text{where} \quad \mathbf{x} = (x, y, \theta, r, \phi, \psi_1, \psi_2, \chi)^T.$$

Here (x, y, θ, r) is a non-linear representation of a Euclidean similarity transform applied to a template as in [5], ϕ is the offset angle of the index finger from the template position, and ψ_1, ψ_2 are offset angles for the first and second thumb joints respectively. The extra parameter χ is a discrete label which determines whether the template is left- or right-handed. For measurement and display purposes, the 7 parameters are mapped to a B-spline contour approximating the outline of the hand [1].

1.2 Dynamics

A principled choice of dynamics is essential for good tracking. Our experience is that obvious choices such as constant velocity or constant position are either too unstable or too unresponsive, but that second-order auto-regressive processes produce excellent results while being easy to implement and inexpensive to compute. Even in the absence of training data there exist established methods for choosing the parameters of such ARPs (see [1]) and this is the approach taken here. Each continuous parameter is modeled as an independent 2nd-order ARP (i.e. an oscillator). The oscillators are specified by parameters defining a damping constant β, a natural frequency f and a root-mean-square average displacement ρ. These parameters can be used to determine the ARP model in one dimension

$$x_t = a_2 x_{t-2} + a_1 x_{t-1} + b\omega_t$$

where ω_t is Gaussian noise drawn from $N(0, 1)$, a_1, a_2 and b are given by

$$a_2 = -\exp(-2\beta\tau), \quad a_1 = 2\exp(-\beta\tau)\cos(2\pi f\tau)$$

$$b = \rho\sqrt{1 - a_2^2 - a_1^2 - 2\frac{a_2 a_1^2}{1 - a_2}}$$

and τ is the time-step length in seconds (so $\tau = 1/25\,s$ for PAL frame-rate). Sensible default parameters for the oscillators are chosen by hand, and given in section 2.

Note that because the parameters are modeled as *independent* oscillators, it is trivial to decompose the dynamics in the form of

$$f(\mathbf{x}''|\mathbf{x}) = \int f_2(\mathbf{x}''|\mathbf{x}')f_1(\mathbf{x}'|\mathbf{x})d\mathbf{x}', \tag{1}$$

for any cross product partition of the parameters; the choice of which parameters to put in the first partition $\mathcal{U}$ is determined by the availability of a computationally inexpensive measurement density m_U. In this application the parameters (x, y, θ, r) describing the main hand are placed in the first partition, with two subsequent partitions: one for the index finger parameter ϕ, and another for the two thumb parameters ψ_1, ψ_2.

1.3 Importance sampling

Using well-established methods for identifying skin-coloured blobs in office scenes [6, 3], we follow the ICondensation methodology [5] of using a sum of Gaussians centred on such blobs as an importance function for Condensation. The implementation closely follows that of [5], but with a useful addition: the offset from the centre of a colour blob to the centroid of a hand template is adaptively estimated using a Kalman filter. Note that the colour segmentation also provides a convenient method for initialising and reinitialising the tracker, and this is another feature of ICondensation which has been adopted in this paper. The left–right parameter χ is held constant in the motion model, and can only change as a result of a reinitialisation.

1.4 Measurement likelihood

It remains to specify the measurement likelihood $m(\mathbf{z}|\mathbf{x})$. Recall that the parameters $\mathbf{x}$ correspond to a B-spline in the image. A one-dimensional grey-scale edge operator is applied to the normal lines to this B-spline at 28 points (8 on the main hand, 6 on each of the thumb joints and 8 on the index finger). Each of the 28 resulting "edges" (actually points which are the nearest above-threshold responses of a 1D operator) has a normal distance ν_i from the B-spline, which would be zero if the model fitted the image edges perfectly. By assuming (i) the deviations of the model from the template shape are Gaussian, (ii) that such deviations are independent on different normal lines, and (iii) there is a fixed probability of finding no edge, it is easy to see that the form of $m(\mathbf{z}|\mathbf{x})$ should be

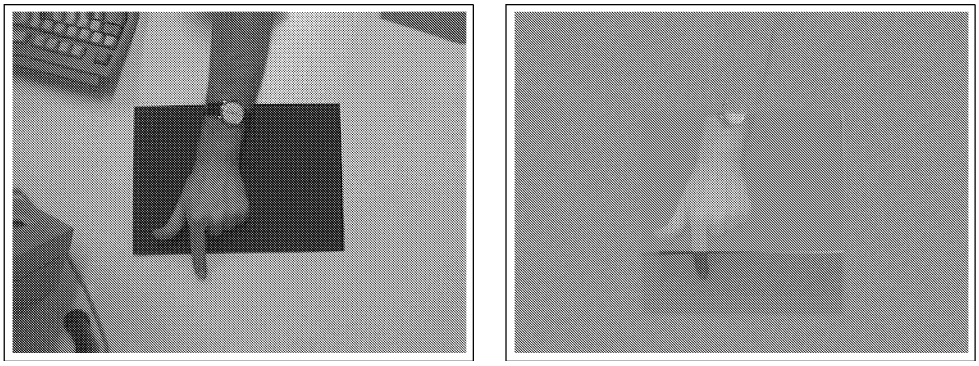
$$\log p(\mathbf{z}|\mathbf{x}) \propto \text{const} + \sum_m \nu_m, \tag{2}$$

where the constant was set by hand for this application. We can also exploit the fact that the portion of a normal line on the *interior* of the B-spline should be skin-coloured. This is reflected by adding to (2) the output of correlating the (colour) normal line pixel values with a colour template. Full details on densities of the form (2) can be found in [7, 8]. The measurement density m_U for the first partition (i.e. the fist) is of the same form (2), but the summation is over only those normal lines on the outline of the fist; similarly the measurement densities on the thumb and index finger partitions are calculated from their relevant normal lines.

1.5 Background subtraction

An adaptive background subtraction methodology in the same spirit as [4] is used to assist the tracker. The background image is estimated as the weighted sum of recent images, except in a region near the present hand position, where the background is defined to be the most recent "uncorrupted" estimate. The alpha-blending hardware on an SGI Octane permits this type

background processing with minimal computational overhead. All grey-scale edge detection is done on the background-subtracted image, whereas colour processing is done on the raw image. Our experience agrees with the somewhat notorious reputation of background subtraction: that it helps least when it is most needed (figure 1). Hence the need for a robust tracking algorithm such as Condensation.



(a) Raw image (b) Background-subtracted image

Figure 1: ***Background subtraction.*** *Note that while background subtraction removes some edges from image, it does not do so perfectly and often introduces strong new edges in the interior of the target object.*

1.6 Fingertip localisation

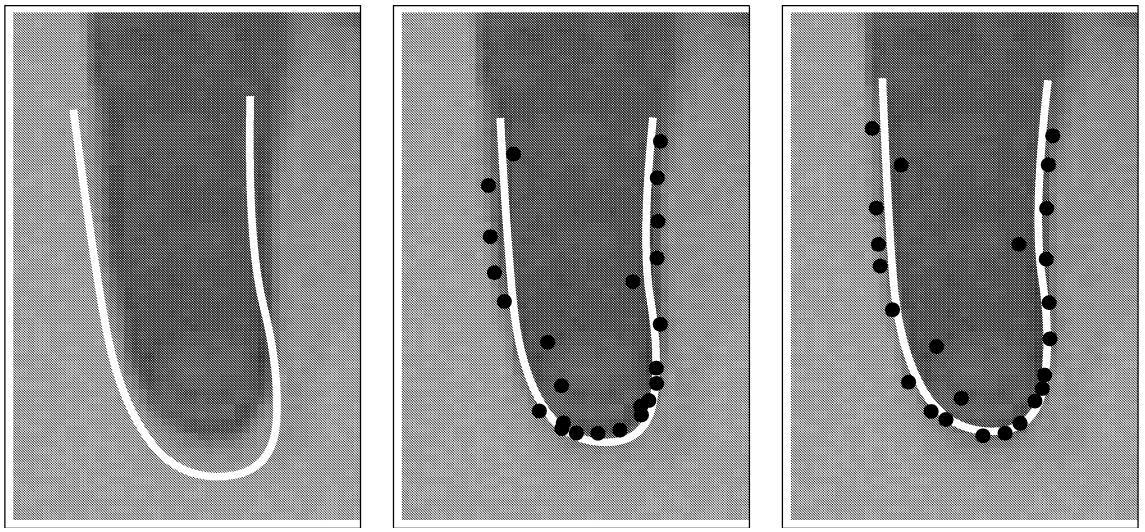
The output of the Condensation tracker is a weighted particle set, which represents a probability distribution. An interactive tool such as a mouse pointer, however, requires a single value of the current state. An obvious choice for this value would be the mean of the probability distribution, which can be estimated as the weighted sum $\overline{\mathbf{x}}$ of the particles. However, it turns out that since $\overline{\mathbf{x}}$ is estimated from only $N_3 = 90$ particles in this application, it has a variance of the order of many pixels — too high for a genuinely usable mouse pointer. Hence $\overline{\mathbf{x}}$ is used as the starting point for a deterministic fitting procedure which outputs $\overline{\mathbf{x}}^*$, the actual parameters of the pointer shown on the screen. This procedure has two steps (see figure 2):

1. A new B-spline whose template is the shape of a fingertip is initialised at the position implied by $\overline{\mathbf{x}}$. The regularised least squares routine of [1] is then applied four times in succession: twice with a search scale of 40 pixels, and twice with 14 pixels. The B-spline is permitted to deform in the 4D linear shape space of Euclidean similarities, and the edge measurements for the least squares fitting are found by applying the same 1D grey scale edge operator as for the measurement density. This may produce several different features whose response is above threshold; in this case the feature with the greatest response to the colour template is selected as the one to be used for the least squares fit.

2. The result of the fitting procedure is passed through a Kalman filter [2] to obtain $\overline{\mathbf{x}}^*$.

2 A vision-based drawing package

The hand tracker described in the previous section was implemented on an SGI Octane with a single 175MHz R10000 CPU. Using 700 samples for the hand base, 100 samples for each of the thumb joints and 90 samples for the index finger, the tracker consumes approximately 75% of the machine cycles, which allows real-time operation at 25Hz with no dropped video frames even



(a) (b) (c)

Figure 2: ***Fingertip localisation.*** *(a) A fingertip B-spline initialised at the position implied by* $\overline{\mathbf{x}}$*, the mean of the Condensation distribution. (b) Result after one application of regularised least squares. (c) Result after four iterations. This is fed into a Kalman filter to give* $\overline{\mathbf{x}}^*$*.*

while other applications are running on the machine. The parameters used for the dynamics are shown in table 1.

	β (s^{-1})	f (Hz)	ρ
x	3	0	100 pixels
y	3	0	100 pixels
θ	5	0	0.5 rad
r	5	0	0.2
ϕ	3	0	0.5 rad
ψ_1	3	0	0.5 rad
ψ_2	3	0	0.5 rad

Table 1: **Oscillator coefficients for the hand dynamics.** *Note that* $f = 0$ *corresponds to a choice of critical damping for all parameters.*

2.1 Tracking accuracy

For a hand-tracker to be seriously considered as an input device it must compare favorably with existing technologies, in particular the mouse. It must be possible to reliably position a pointer to sufficient accuracy without latencies which the user finds unacceptable. "Sufficient accuracy" is obviously application-dependent; the type of icon selection task typically performed with a touch-screen requires only fairly gross pointer localisation, for example. The application chosen here, a freehand drawing package, is rather demanding if convincing results are to be obtained; it must be possible to draw smooth lines rapidly and also locate the pointer to within a few pixels. The drawing canvas must also be large enough for comfortable use, certainly larger than the 768x288 pixel image field on which tracking is taking place. It is therefore necessary to scale the output of the hand-tracker so that a movement of one screen pixel corresponds to sub-pixel motion in the image. We choose a scaling of 2x4 which allows comfortable movement around an 800x600 pixel drawing canvas without excessive excursions of the hand.

The output of the tracker on a stationary hand is shown in figures 3 and 4. The raw output of the Condensation tracker has a jitter of up to 8 image pixels, leading to a 16 pixel screen jitter, which is unacceptably noisy. It is possible to reduce the jitter somewhat at the expense of processing time by increasing the number of samples. The random-sampling nature of the algorithm, however, combined with the problem that the contour does not perfectly fit the entire hand and is thus susceptible to jumping between nearby local minima of the measurements, means that it is impossible to obtain sufficiently smooth output using Condensation alone in real-time. Figure 3 shows the output of the least-squares fingertip fitting process described in section 1.6. This is much smoother, but the jitter is still slightly too great for comfortable use. This output is therefore smoothed using a Kalman filter. The choice of filter parameters is a tradeoff; too high and the jitter is not significantly reduced, too low and the pointer lags noticeably and feels sluggish. A second-order oscillator as described in section 1.2 is used, and the parameters were chosen as follows: $\beta = 2s^{-1}$, $f = 0$, $\rho = 100$ pixels. Figure 4 shows that after smoothing the jitter is on the order of 4 screen pixels at most, which is found to be perceptually acceptable. Moreover at 25Hz PAL video rate the tracker feels very responsive with no noticeable lag. The smoothing introduces small overshoots when the hand decelerates suddenly, and there is certainly scope for further research to fine-tune the behaviour. The stationary jitter could probably be substantially reduced by introducing a variable gain in the Kalman filter, which would permit more smoothing at low velocity. It may also be possible to explicitly remove overshoots using post-processing. Simply reducing the gain in the Kalman filter as it stands is unacceptable, since while the jitter is removed, overshoots become very noticeable and the pointer lags annoyingly.

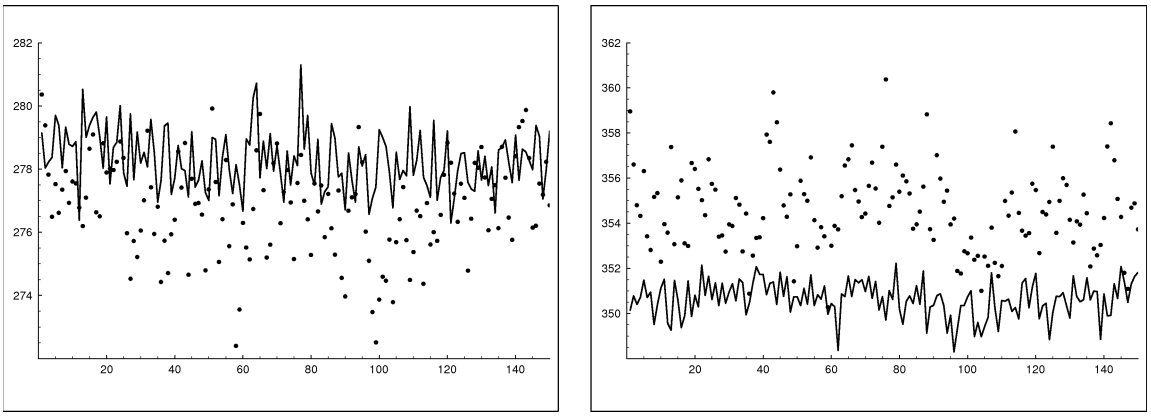


Figure 3: **Least-squares fitting** *reduces noise on the fingertip localisation. The left and right-hand figures show the estimated x and y coordinates respectively of the fingertip over a period of 150 fields when the hand was held stationary. The dots show the raw output of the Condensation tracker and the solid line is the result of performing a least-squares fit to localise the tip as described in section 1.6. The y axes are in units of pixels in the camera image. The bias evident in the right hand figure is due to the Condensation output being a fit to fist* and *finger.*

In practice, the gross tracking is extremely robust to image clutter. Figure 5 shows a desk covered in clutter edges, and the output of the Condensation tracker is more or less unaffected by this clutter, even when the papers are moved, invalidating the background subtraction. The fingertip localisation is somewhat more susceptible to clutter, however, so the jitter on the pointer location increases noticeably over heavy moving clutter. Thus for applications where high accuracy is a prerequisite it is advantageous to minimise clutter as much as possible. A few pens or pieces of paper lying on the desk do not noticeably affect tracking, however. Due to the reliance on colour information for discriminating between feature edges, the tracker can fail if large areas of orange or pink paper are placed in the image. Again, if only a small amount of

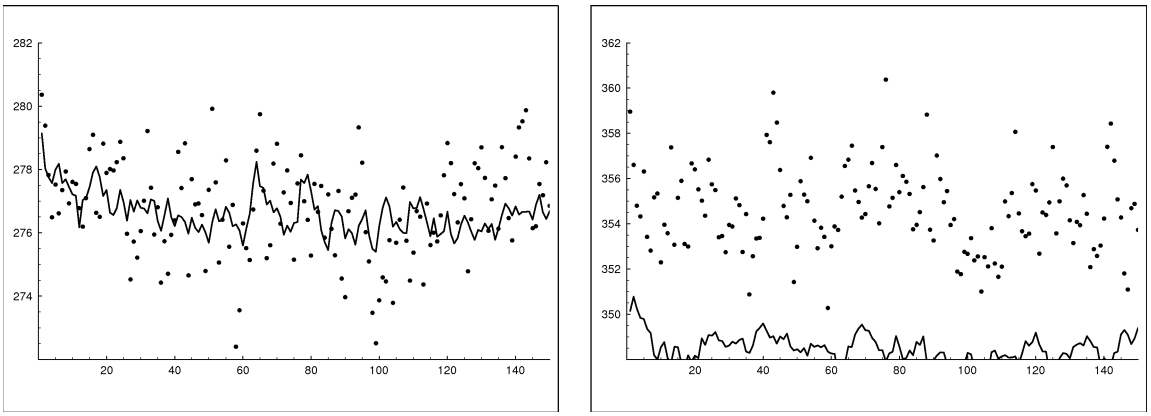


Figure 4: **Kalman filtering** *the output of the least-squares tip localisation smoothes away jitter in the estimates. The left and right-hand figures show the estimated x and y coordinates respectively of the fingertip over a period of 150 fields when the hand was held stationary. The dots show the raw output of the Condensation tracker and the solid line is the result of Kalman-filtering the least-squares fingertip fits. The y axis is in units of pixels in the camera image.*

distracting skin-coloured material is present, the tracking is unaffected. Figure 6 demonstrates successful tracking despite the presence of two hand-coloured objects in the image.

The index finger is followed successfully as it rotates about the knuckle (figure 7). This allows small pointer gestures to be made without moving the hand. The fingertip is still located if the finger is bent slightly (reducing the apparent length of the finger) and a useful improvement to the model would be to include an extension parameter for the index finger. This might also increase the accuracy of the Condensation estimates by allowing the tracker to better fit the true outline. The two thumb joints are also tracked (figure 8) but with slightly less accuracy. This is probably largely because the thumb is not well modeled by two jointed segments pivoting about the knuckle; in fact the knuckle itself moves as the thumb is opposed, so often the model cannot adequately fit the visible thumb with any choice of parameters. The tracking is nevertheless quite sufficient to allow the thumb to be used as a toggle switch for the drawing package to be describe in the next section.

2.2 Drawing with the hand

We are developing a simple drawing package to explore the utility of a vision-based hand tracker for complex and subtle user-interface tasks. We believe that the tracking achieved is sufficiently good that it can compete with a mouse for freehand drawing, though (currently) at the cost of absorbing most of the processing of a low-end workstation. It is therefore instructive to consider what additional strengths of the vision system we can exploit to provide functionality which could not be reproduced using a mouse.

The current prototype drawing package provides only one primitive, the freehand line. When the thumb is extended, the pointer draws, and when the thumb is placed against the hand the virtual pen is lifted from the page. Immediately we can exploit one of the extra degrees of freedom estimated by the tracker, and use the orientation of the index finger to control the width of the line being produced. When the finger points upwards on the image, the pen draws with a default width, and as the finger rotates the width varies from thinner (finger anti-clockwise) to thicker (finger clockwise) — see figure 9. The scarcity of variable-thickness lines in computer-generated artwork is a testament to the difficulty of producing this effect with a mouse.

Having a camera looking at the desk also allows other intriguing features not directly related to the hand-tracking. We have implemented a natural interface to translate and rotate the

Figure 5: **Heavy clutter** *does not hinder the Condensation tracker. Even moving the papers on the desk to invalidate the background subtraction does not prevent the Condensation tracker functioning. The fingertip localisation is less robust, however, and jitter increases in heavily cluttered areas.*

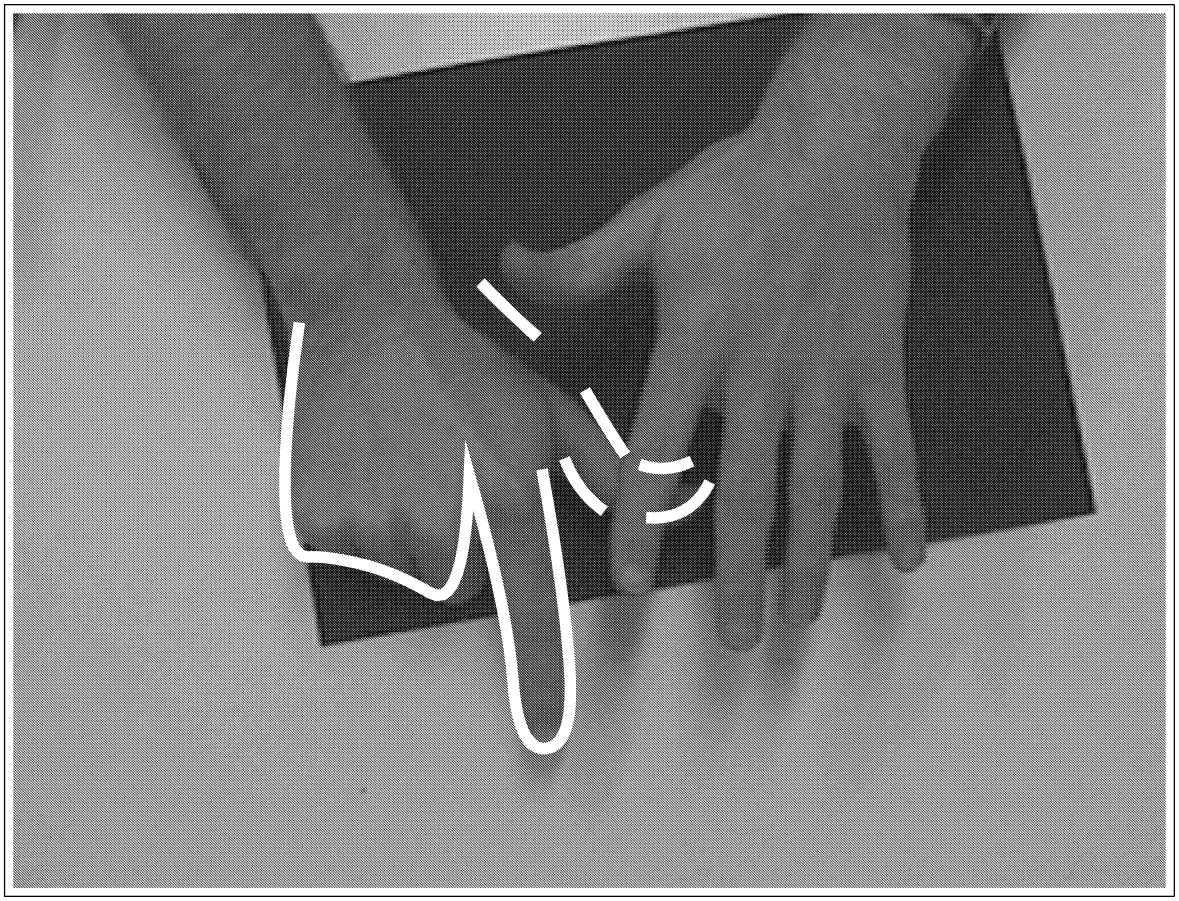


Figure 6: **Skin-coloured objects** *do not distract the tracker. Here two hands are present in the image but tracking remains fixed to the right hand. If the right hand were to leave the field of view the tracker would immediately reinitialise on the left hand.*

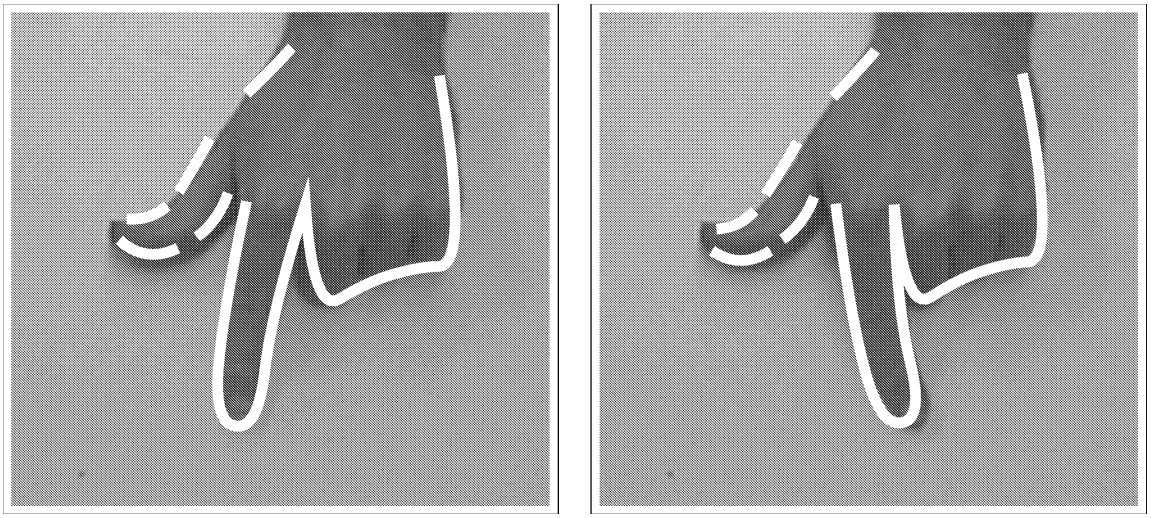


Figure 7: **The index finger** *is tracked rotating relative to the hand body. The angle of the finger is well-estimated, and agile motions of the fingertip, such as scribbling gestures, can be accurately recorded.*

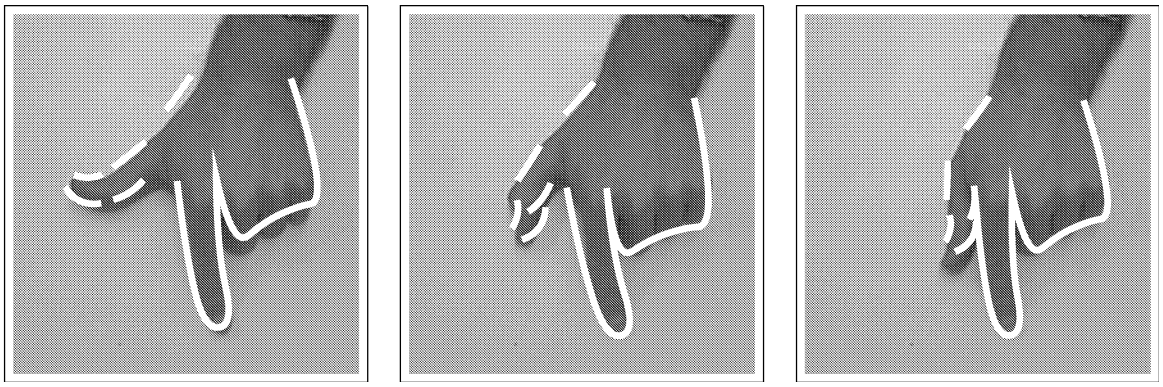


Figure 8: **The two degrees of freedom of the thumb** *are tracked. The thumb angles are not very reliably estimated. This is probably partly because the joints are short, and so offer few edges to detect, and more importantly because the shape model gives a poor approximation to the thumb when it opposes. The gross position of the thumb can be extracted consistently enough to provide a stable switch which can be used analogously to a mouse button.*

virtual workspace for the modest hardware investment of a piece of black paper (figure 10). The very strong white-to-black edges from the desk to the paper allow the paper to be very reliably tracked using a simple Kalman filter, at low computational cost. Translations and rotations of the paper are then reflected in the virtual workspace, a very satisfying interface paradigm. While one hand draws the other hand can adjust the workspace to the most comfortable position, and this is demonstrated on the accompanying video. The video begins with an establishing shot of the user at the Octane workstation, drawing in real time. The recording then cuts to an extended drawing sequence recorded directly from a workstation screen. The Octane does not support recording from screen, so the canvas is being displayed on another machine along with a direct video feed from the camera, and hence it is impossible to show the tracked contour outline. The drawing package is controlled entirely from the output of the video, except for zooming which is accomplished by pressing keys on the keyboard. In the future it should be possible to perform discrete operations such as switching between drawing tools using simple static gesture recognition on one of the hands. Tracking both hands would allow more complex selection tasks, for example continuous zooming, or colour picking.

References

[1] A. Blake and M. Isard. *Active contours.* Springer, 1998.

[2] A. Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge, MA, 1974.

[3] H.P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen, and E. Petajan. Multi-modal system for locating heads and faces. In *Proc. 2nd Int. Conf. on Automatic Face and Gesture Recognition*, pages 88–93, 1996.

[4] I. Haritaoglu, D. Harwood, and L. Davis. w^4s: A real-time system for detecting and tracking people in 2.5D. In *Proc. 5th European Conf. Computer Vision*, volume 1, pages 877–892, Freiburg, Germany, June 1998. Springer Verlag.

[5] M.A. Isard and A. Blake. ICondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. 5th European Conf. Computer Vision*, pages 893–908, 1998.

[6] R. Kjeldsen and J. Kender. Toward the use of gesture in traditional user interfaces. In *Proc. 2nd Int. Conf. on Automatic Face and Gesture Recognition*, pages 151–156, 1996.

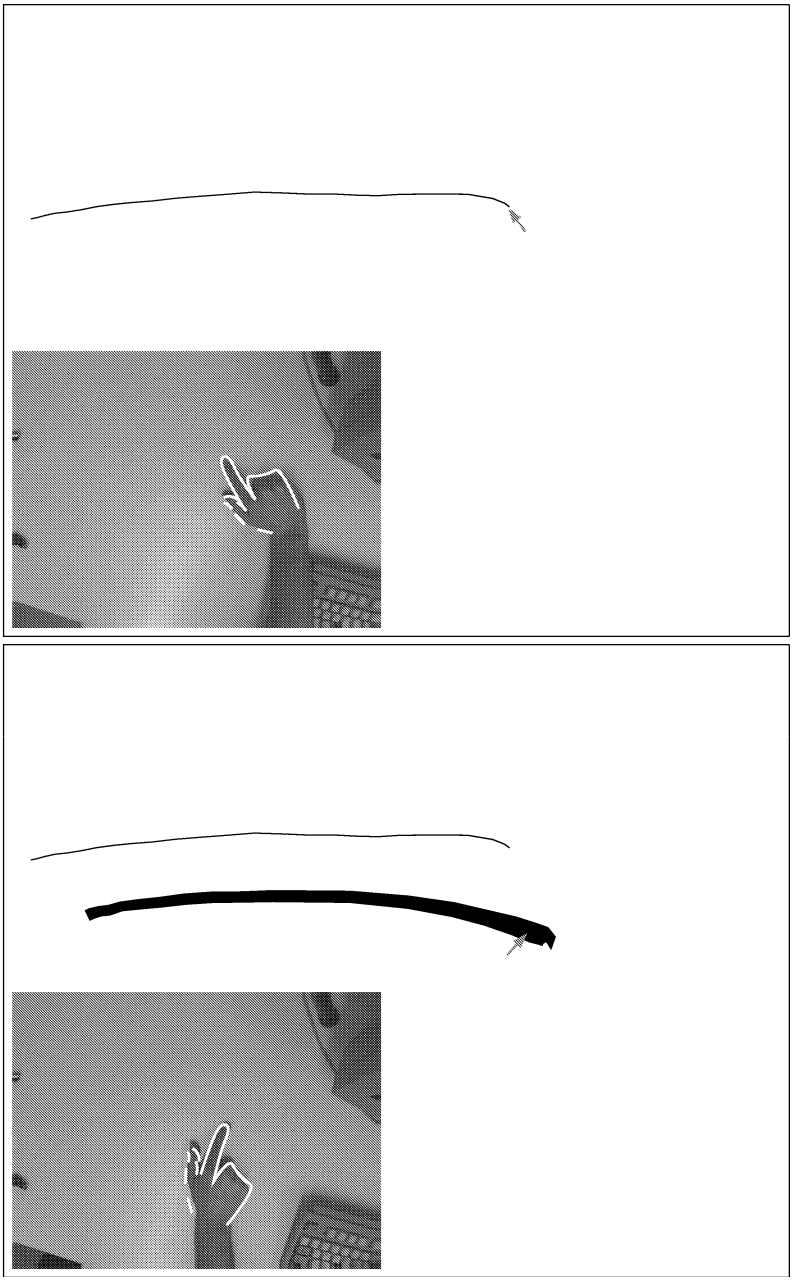


Figure 9: **Line thickness** *is controlled using the orientation of the index finger. The top image shows a line drawn with the index finger pointing to the left, producing a thin trace. In the bottom image the finger pointed to the right and the line is fatter. Of course if the finger angle varies while the line is being drawn, a continuous variation of thickness is produced.*

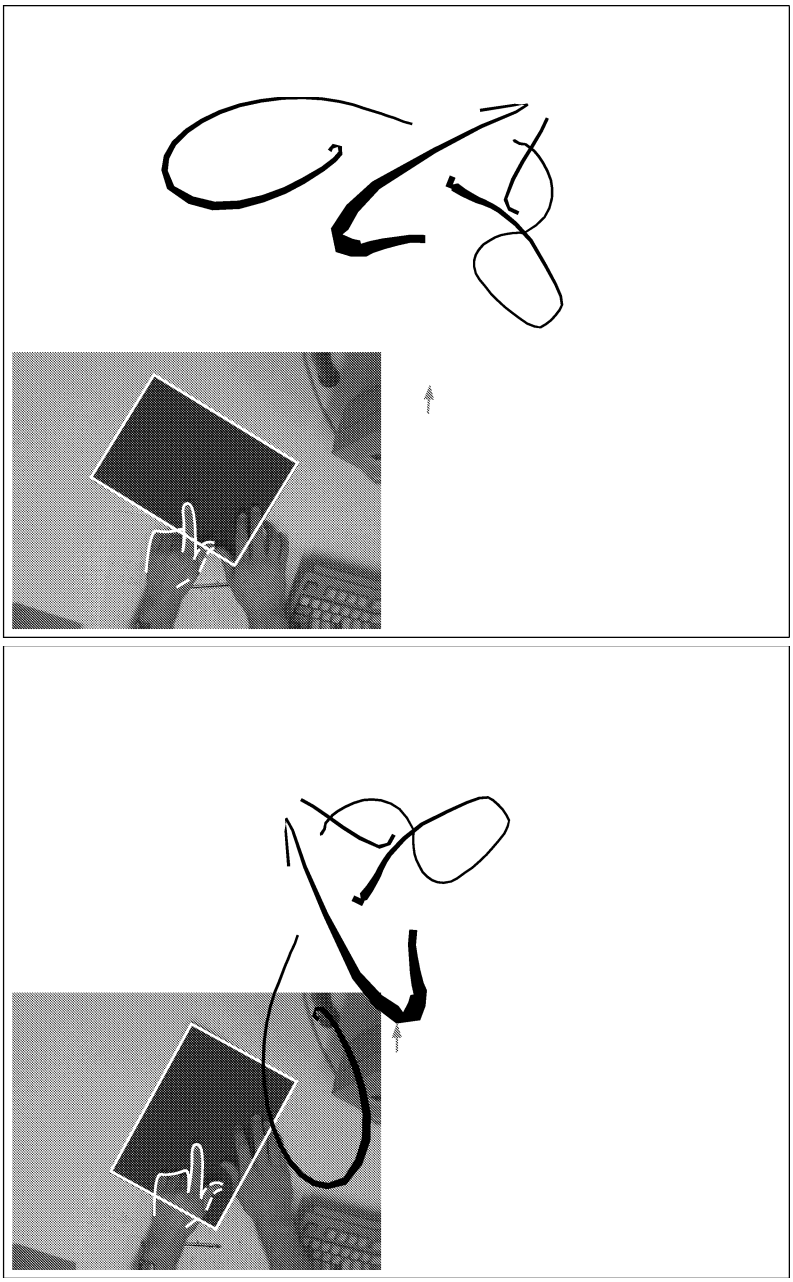


Figure 10: **Moving around the virtual workspace** *is accomplished by following the tracked outline of a physical object. The piece of black paper can be tracked with a simple Kalman filter, and the virtual drawing follows the translations and rotations of the paper. The virtual workspace has been rotated anti-clockwise between the top and bottom frames. This is a very natural interface which can be used while drawing.*

[7] J. MacCormick and A. Blake. A probabilistic contour discriminant for object localisation. In *Proc. 6th Int. Conf. on Computer Vision*, pages 390–395, 1998.

[8] J. MacCormick and A. Blake. Spatial dependence in the observation of visual contours. In *Proc. 5th European Conf. Computer Vision*, pages 765–781, 1998.

[9] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. 6th European Conf. Computer Vision*, Dublin, 2000.