FingerCloud: Uncertainty and Autonomy Handover in Capacitive Sensing

Simon Rogers, John Williamson, Craig Stewart, Roderick Murray-Smith
Department of Computing Science University of Glasgow, G12 8QQ, UK
{ simon, jhw, craig, rod } @dcs.gla.ac.uk

ABSTRACT
We describe a particle filtering approach to inferring finger movements on capacitive sensing arrays. This technique allows the efficient combination of human movement models with accurate sensing models, and gives high-fidelity results with low-resolution sensor grids and tracks finger height. Our model provides uncertainty estimates, which can be linked to the interaction to provide appropriately smoothed responses as sensing perfomance degrades; system autonomy is increased as estimates of user behaviour become less certain. We demonstrate the particle filter approach with a map browser running with a very small sensor board, where finger position uncertainty is linked to autonomy handover.

Author Keywords
Uncertainty,particle filters, probabilistic interaction, capacitive sensing, H-metaphor.

ACM Classification Keywords
H.5.2:Input devices and strategies

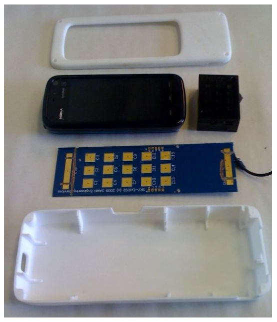
General Terms
Human Factors

INTRODUCTION
This paper presents a technique for interpretation of an array of capacitive sensors such that the information can be used for direct foreground interaction control and background control or context sensing. These techniques allow us to cope with dynamic, noisy sensor inputs, and to create systems whose level of autonomy increases as ambiguity increases – the system takes over as user input becomes less certain. This sensing ability allows the creation of interfaces for rich interaction, open to the inclusion of more sophisticated models based on context of use, or prior behaviour, to support direct human motor control.

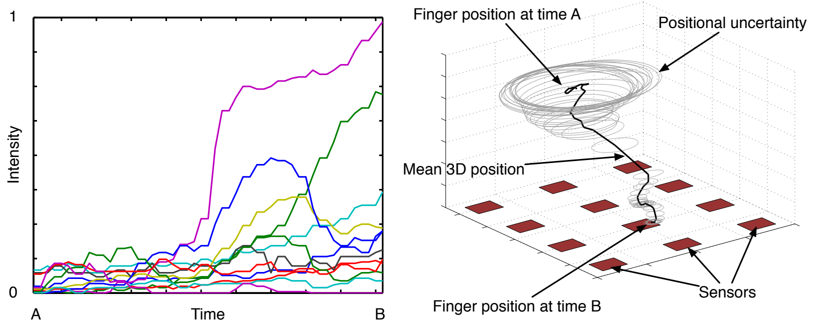
A human-computer interface interprets user actions and carries out the user's intention. The computer system estimates states of the physical world using its sensors. These estimates are the means by which a human can control the internal state of the computer and cause it to perform actions. All systems have some level of ambiguity, due to a mix of limited sensing of the environment, and poor models of the complexity of human users. [10] and [9] discuss the importance of maintaining uncertainty in user interfaces. Interaction design has, however, typically been challenged by designing solutions for inputs which are high-dimensional, dynamic and uncertain. Many rich analogue sensors such as accelerometers or capacitive inputs have been used in an essentially discrete fashion with simple thresholds triggering key-press equivalent events. Even simple tasks such as using accelerometers to re-orient the screen from portrait to landscape have undesirable consequences because of the difficulty of interpreting context appropriately.

Designing for fluid handover of autonomy
In [5], the "H-metaphor" is described, where a rider's interaction with his or her horse is used as an analogy for the handover of autonomy in computer systems as the certainty of control varies. If a rider uses certain, frequent and deliberate movements, the horse follows movements exactly; as the control becomes vaguer, the horse resorts to familiar behaviour patterns and will take over more of the control. We believe that this notion of being able to 'loosen or tighten the reins' of interaction with an intelligent device is likely to be vital in the creation of future human–computer interaction systems. Capacitive sensing systems are ideally suited for this style of communication and control, as the control metaphor is literal tightening or loosening of the grasp of the device itself.

BACKGROUND
To explicitly model uncertainty, we adopt a *Bayesian* approach. This involves a shift from considering particular values of objects of interest (for example, a contact pointon a sensor array) to distributions over possible values. Uncertainty is naturally handled by the degree of variance in the distribution. Unfortunately, for most interesting applications, such distributions are not analytically tractable and we must resort to sampling techniques. For distributions that evolve over time, the most popular family of sampling techniques are Sequential Monte-Carlo techniques (a.k.a. particle filters, PFs) [3]. Broadly speaking, PFs approximate the distribution at a particular time-point with a set of samples (particles). The set of particles at the next time instant is generated from the current set by re-sampling particles



(a)



(b)

Figure 1. (a) SK7 Sensor pack with optional 24 channel external capacitive sensing board. (b) *left* Time series, giving sensor intensity across a 4x3 sensor array (3mm^2 pads) as the finger is brought into contact. The complexity of the trajectory is apparent. (b) *right* Inferred 3D position as a finger approaches the sensors. Ellipses show inferred uncertainty in x_1 and x_2.

with probability proportional to how well they match the observed data.

Capacitive Sensing

Capacitive sensing works by measuring the change in capacitance of a virtual capacitor between sense pads and ground. When a grounded object, such as a human body, comes into proximity with these pads, the apparent capacitance increases. Figure 1(a) shows the sensing hardware used in this paper. An SK7 sensor packfrom SAMH Engineering Services was used for capacitive sensing. Data is sampled from 12 small square pads in a 4×3 configuration, measuring 26×20mm and providing 8-bit resolution at 110Hz. With a 0.012pF sensitivity, this gives a proximity range from 0-10mm, with the value decreasing exponentially with the distance. An optional 24 channel board can be used for higher resolution sensing in a larger form factor. Figure 1(b) (left) shows a typical time-series from a 12 sensor array with the finger moving in and contacting the board. The resulting inferred 3D position shown in Figure 1(b) (right).

Whole device interaction

There has been increasing recent interest in sensing contact with the whole device[2, 11, 1], where current prototypes have typically used either existing mouse pads, or combined multiple touch sensitive off-the-shelf devices. The advantages of such systems include interacting without obscuring the screen [1], dual front-and-back interaction [8, 7, 12] and sensing hand posture.

IMPLEMENTATION

We have built a robust, high-quality realtime finger tracking system which uses a particle filter to estimate finger position using relatively low-resolution sensor arrays. This system is implemented using custom hardware and is implemented in Python. This filter can track finger position at 100Hz or more, with sub-fingertip accuracy. In testing, our particle filter roughly halves the RMS error in XY position for repeated trials of touching a sensor pad compared to standard linear interpolation methods, as well as providing robust height values. The particle filter is 3D, and can reliably track the finger during approach to the sensor pad, with realistic estimates of sensor uncertainty. It supports multitouch sensing, with up to three points of contact. It also supports spatially-varying prior distributions over velocity, so that beliefs about flows can be encoded in the filter and used during estimation.

3-DIMENSIONAL SENSING

We now describe the method that we use to track the 3D location of a finger. In particular, we are interested in the *posterior* distribution $p(\mathbf{x}|\mathbf{c})$ which encapsulates the (un)certainty we have in the finger location ($\mathbf{x}$) conditioned on the values from the sensors $\mathbf{c}$. We will generate samples from this distribution using a PF, depicted in Figure 2. The filter consists

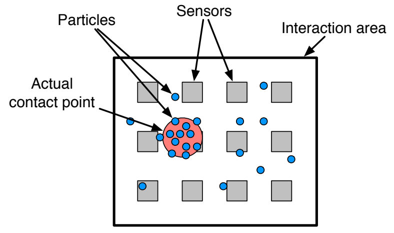


Figure 2. Particle filter to determine contact location on a sensor grid (third (height) dimension ommitted for clarity).

of a population of S particles that at any moment represent a sample from $p(\mathbf{x}|\mathbf{c})$. At each iteration a new population is created by *re-sampling* particles from the population with probability proportional to how well they agree with the observed data. In our application each particle consists of a location in 3D and we denote the sth particle by $\mathbf{x}_s$. After initialisation with random particles (sampled from a prior distribution), the filter consists of two steps – *particle weighting* and *particle re-sampling* – repeated at each sensor update.

Particle weighting In this stage, the particles are assessed to determine how well they fit to the observed data. This stage relies upon a model of sensor response for any particular contact location. We use a negative exponential function for the response of the ith sensor defined on the distance between a particle and the sensor location $\mathbf{q}_i$: $r_i(s) = \exp\{-K||\mathbf{x}_s - \mathbf{q}_i||_2\}$. The response becomes flatter as the contact is moved away from the sensor plane, which results in broadening the posterior distribution (and hence increasing posterior uncertainty). The explicit modelling of the sensor response makes it easy to support varying sensor pad shapes. The filter could, for example, be used to track fingers using hierarchical sensors, where coarse pads provide long-range depth information and low-resolution XY information, and smaller pad provide high resolution XY information as fingers approach the surface. The vector of values obtained when this is evaluated for all I sensors is then compared against the current observed values using a Gaussian likelihood. This gives the un-normalised weight for the sth particle, w_s (the higher the weight, the better the particle approximates the true position): $w_s \propto \exp\left\{-\gamma \sum_{i=1}^{I}(r_i(s) - c_i)^2\right\}$. Finally, these weights are normalised to give particle probabilities: $\hat{w}_s = \frac{w_s}{\sum_{s'=1}^{S} w_{s'}}$.

Particle re-sampling In this step, we use the current particle population to generate a new population. Firstly, we create a proportion (N_r) of the S particles randomly from some distribution $p(\mathbf{x}_s)$ - in our demonstrations we use a uniform distribution over the 3-dimensional space. This step ensures that the filter can track rapid changes in the distribution (for example, the finger being removed and then placed back again). Secondly, we generate the remaining $S - N_r$ particles by sampling *with replacement* from the distribution defined by the normalised weights $\hat{w}_2$ - i.e., we produce the tth particle in the new generation from the sth particle in the previous generation with probability $\hat{w}_s$. As we are sampling with replacement, it is possible that a single particle (with high $\hat{w}_s$) could produce several particles in the next generation. Finally, we must define a prior distribution on the evolution of $\mathbf{x}$ - $p(\mathbf{x}_t|\mathbf{x}_s)$. This distribution encapsulates our beliefs regarding how the contact will move through the 3D space. We define $p(\mathbf{x}_t|\mathbf{x}_s)$ as a spherical Gaussian - $p(\mathbf{x}_t|\mathbf{x}_s, \sigma^2) \sim \mathcal{N}(\mathbf{x}_s, \sigma^2\mathbf{I})$. This encodes the prior belief that the user moves in a smooth manner. Many movement models beyond the simple smoothness constraint are possible. As long as we can define the density $p(\mathbf{x}_t|\mathbf{x}_s)$ and sample from it, it can be incorporated into our model. For example, a minimum-jerk movement model can implemented as a simple prior on the third derivative of position. Richer movement models can be obtained by suppling each particle with a velocity that is depends on the particle's location. We have implemented models where a spatially-varying velocity distribution can be learned from previous movement patterns. At each time step, particles are moved according to this velocity field and will only survive if this movement is consistent with the user's.

Multi-touch The particle filter approach naturally handles multi-touch interaction. Particles are generalised to have multiple locations. Imagine a particle that corresponded to two locations. As long as we could compute the theoretical sensor output for contacts at both of these locations (we have found that an additive model works well), we can compute the particle weight and proceed as above. Allowing particles with differing numbers of contacts to coexist in the filter allows us to track the probability distribution over both contact location and number of contacts. The particle filter is robust to additional fingers. Tracking of the original finger(s) remains undisturbed by additional contact points.

Whole device sensing Velocity models are also useful when sensors are distributed about the body of a device – not just on flat surfaces, but curved around the entire form of a mobile device or hand controller. Our model allows to encode priors about movement flows across the surface which result from the effect of the curvature and textures of the object surface. A rich variety of forms can become useful interaction surfaces, where the model copes well with the intrinisic movement distortions caused by the shape of the object. The particle filter also makes it easy for researchers to rapidly prototype novel arrangements of sensor arrays with crude copper tape and glue construction without high accuracy engineering.

APPLICATION - A SINGLE FINGER MAP BROWSER

We now demonstrate how explicitly incorporating uncertainty into the sensing regime can lead to efficient, natural interaction. A common method for navigating around an image on a mobile device is to combine dragging movements with a multi-touch pinch-release movement on a touch-sensitive screen. This has the drawback that the user must use two fingers and cannot support the device and perform the interaction at one time. The same functionality can be accomplished using just one finger by incorporating the 3D location and explicit uncertainty modeling. Assume we have a large map of which we can only view a small display area that has fixed aspect ratio but can increase and decrease in area (zoom), shown in the first pane of Figure 3. The sensor array is assumed to be centred on the centre of the viewing area but exists in the map space so has fixed size (can be bigger or smaller than the current display). Given a current input (blue particles in Figure 3(left)), the system moves and scales the current display window in an attempt to capture as many input samples as possible. Having updated the display window, the virtual sensor array position is updated accordingly.

Zoom and display position naturally adapt to the certainty of the input. A very certain input will result in a small cloud, a decrease in display area and hence zooming in. An uncertain input will force the display area to cover more of the map resulting in zooming out. This is an extension of speed-dependent zooming [6, 4]. The analogy to the 'reins' in the 'H-metaphor' is that when the user behaves purposefully, with firm movements, they have tight control, while when they provide more vague input, the priors on different cities have more effect, leading to more autonomous control behaviour from the system. In essence, the problem is a control task, where the system attempts to maximize the density of particles in view space, subject to dynamical constraints which encode human perceptual limits. The particles from the filter are projected into view space and the viewing volume is adjusted to maximize the particles in view, i.e. we define a cost function $E(\mu, \Sigma, b)$, defined by the set of particle positions p and the viewport bounding box b and minimize this by solving a set of differential equations. The controller is a simple second order system:$\ddot{b}_x = k_1(b_x - \boldsymbol{\mu}), \ddot{b}_s = k_2\left(\sqrt{\boldsymbol{\Sigma}} + b_x - \boldsymbol{\mu}\right)^{k_3}$ where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance of the particle locations, b_x and b_s are the current viewport centre and scale factor, respectively, and k_1, k_2, k_3

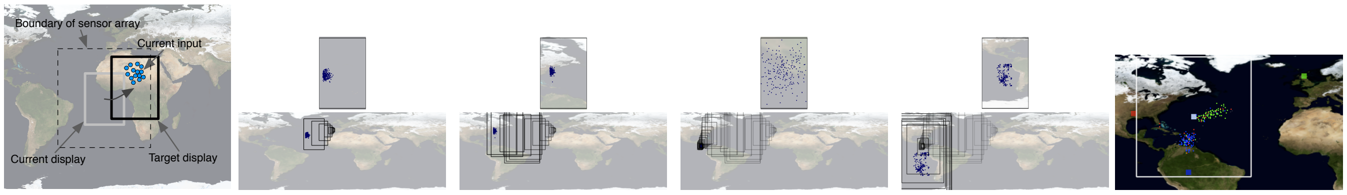


Figure 3. Left pane:Schematic of finger browser operation. Middle panes:browser in use. Final pane: browser incorporating priors over targets.

specify the responsiveness of the system. k_1, k_2, k_3 are adjusted to create a smooth and controllable interaction. We explicitly include constraints to the dynamics because we can only control what we can perceive, and while in principle we can navigate instantly in an arbitrary information space given a static interaction mechanism (e.g. clicking on a scroll bar), if we are dependent on feedback to be displayed while pursuing our goals there are upper limits on the speed at which the display can change. The middle panes of Figure 3 show the system in action. The far right pane illustrates an extension to the model where priors over targets are included. This is an example of the ease with which the sensing model can be combined with domain-specific knowledge. In this case, as uncertainty increases the priors dominate and more control is handed over to autonomous navigation to likely targets (e.g. major cities).

CONCLUSIONS

User interfaces must deal with uncertainty about intention. We have shown how Monte Carlo inference techniques can be used to improve interaction with low-resolution sensing. The negotiation of control between the system and the user is directed by the estimated uncertainty in the intention of the user; as the system's reasoning about a user's behaviour degrades, the system takes over more of the task. We have shown how a simple uncertainty-based linkage between a finger tracking model and a map browsing application leads naturally to an intuitive zooming and browsing paradigm which includes elements of speed-dependent zooming. The technique is simple to understand and implement and can be adapted to virtually any sensing hardware. The filter opens up the possibility of scheduling of different feedback modalities according to the certainty of the inputs. Writing the interaction as a controller trying to maximize the distribution of particles in view space results in a simple and elegant implementation of techniques which have otherwise been implemented *ad hoc* and automatically accounts for the changes in distribution caused by varying uncertainties. A key feature of our system is that reliable z-axis tracking opens up the potential for expressive motion sensing beyond binary finger up/down detection.

Visualising and linking the full distribution of such estimators to the interaction is an essential tool in building interactive systems that degrade gracefully. Degradation can be deliberately be used as in interaction technique; a user can hand over control by ceasing to provide evidence of intention, effectively "letting go of the reins". This leads to an elegant ebb and flow of interaction between the user and computer: dancing with the machine rather than traditional 'command-and-control'.

ACKNOWLEDGEMENTS

This work is funded by EPSRC project EP/E042740/1. Nokia donated devices and additional funding.

REFERENCES

1. P. Baudisch and G. Chu. Back-of-device interaction allows creating very small touch devices. In *CHI '09*, pages 1923–1932, New York, NY, USA, 2009. ACM.
2. P. Cheng and J. Buur. Qualities of the past telephones of the future. In *UIST 04*, pages 77–78. ACM Press, 2004.
3. A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
4. P. Eslambolchilar and R. Murray-Smith. Control centric approach in designing scrolling and zooming user interfaces. *Int. J. Hum.-Comput. Stud.*, 66(12):838–856, 2008.
5. O. Flemisch, A. Adams, S. R. Conway, K. H. Goodrich, M. T. Palmer, and P. C. Schutte. The H-Metaphor as a guideline for vehicle automation and interaction. Technical Report NASA/TM2003-212672, NASA, 2003.
6. T. Igarashi and K. Hinckley. Speed-dependent automatic zooming for browsing large documents. In *UIST '00*, pages 139–148. ACM, 2000.
7. E. E. Shen, S. D. Tsai, H. Chu, Y. J. Hsu, and C. E. Chen. Double-side multi-touch input for mobile devices. In *CHI'09*, pages 4339–4344, 2009.
8. M. Sugimoto and K. Hiroki. Hybridtouch: an intuitive manipulation technique for pdas using their front and rear surfaces. In *MobileHCI'06*, pages 137–140, 2006.
9. J. Williamson. *Continuous Uncertain Interaction*. PhD thesis, University of Glasgow, 2006.
10. J. Williamson, S. Strachan, and R. Murray-Smith. It's a long way to Monte Carlo: probabilistic display in GPS navigation. In *Proc. MobileHCI '06*, pages 89–96. ACM, 2006.
11. J. O. Wobbrock, B. A. Myers, and H. H. Aung. The performance of hand postures in front- and back-of-device interaction for mobile computing. *Int. J. Hum.-Comput. Stud.*, 66(12):857 – 875, 2008.
12. X.-D. Yang, E. Mak, P. Irani, and W. F. Bischof. Dual-surface input: Augmenting one-handed interaction with coordinated front and behind-the-screen input. In *MobileHCI'09*, pages 39–48, 2009.