Tracking Visitor's Fields of Interest in Large Scale Art Installations

Brandon J. DeHart, Rob Gorbet

Abstract—***Aurora*** **is a large-scale kinetic art installation that reacts to human presence directly, with sensors triggering outputs, and indirectly, by modifying output behaviour rules. This paper describes a novel method for estimating visitors' fields of interest, their attention to specific parts of the installation, with a future goal of using this measure as a fitness function for output behaviour modification based on genetic algorithms. Due to constraints in *Aurora*, distributed overhead distance sensors were used as the sensory inputs. A low resolution height graph of the space below the installation is created, and the active sensors are clustered into groups. The height graph and sensor groups are used to produce a probability map of possible visitor locations. Based on these, particle filters are created to estimate the visitors' state, and by extension their fields of interest. Using this overall strategy for tracking and interest prediction, an average prediction accuracy of 92% is found when compared to a set of simulated people moving within a simulated space.**

I. Introduction

This work was developed to answer a question brought forward through a series of large-scale collaborative artistic installations in concert with Philip Beesley Architect Inc. over the last several years. The goal of these collaborations is the development of interactive art as a breeding ground for new forms of design in architecture and engineering.

The first of these, the *Hylozoic Series*, has had a number of generations displayed all around the world. The most public of these installations were representing Canada at the 2010 Venice Architecture Biennale, Venice, Italy and a permanent installation at The Leonardo museum, Salt Lake City, Utah, USA. The *Hylozoic Series* is described as being “artificial responsive forests with organic movements, embedded intelligence, and ongoing chemical reactions.”

Another series of installations, which made their debut at Nuit Blanche 2010 in Toronto (see Fig. 1) is called the *Aurora Series*, and has been described as an “environment for human-aware artificial life within hanging columns of light, movement, and sound.” An updated version of *Aurora* can be found at the Simons store in the West Edmonton Mall, Edmonton, Alberta, Canada. The work in this paper is based mainly on developing needs in the *Aurora Series*.

In the *Aurora* installation, each of the 108 columns of 24 LEDs and 24 vibrating mylar feathers includes an infrared distance sensor, pointed towards the ground. The columns are grouped in sixes, with each group containing and running a single cellular automaton (CA), which is connected with those around it. The state of the cells of each CA is displayed on the 24 x 6 array of LEDs on those six columns. The creative goal in this work is to have instances of the *Aurora Series* get more “interesting” to visitors over time. One possible method for generating “interesting” cellular automata (CAs) in a specific setting is to use genetic algorithms to evolve the CAs' rules and other controllable parameters, which define a specific set of 'genes'.

Thanks to the physically distributed nature of the genes throughout the installation, the fitness function for the genetic algorithm is defined based on how interesting a specific region (and therefore subset of CAs) is to the people passing below. To find these interesting regions, it is necessary to first find what the people are looking at, or interested in, within the installation over time. Determining, with reasonable accuracy, the interesting regions of an installation is the aim of this paper.

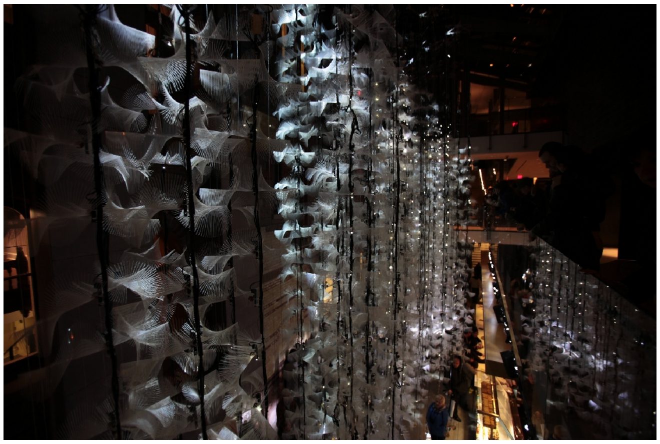

Fig. 1. A side view of the *Aurora* installation hanging in the atrium of the Royal Conservatory of Music in Toronto for Nuit Blanche 2010.

II. Background

Genetic algorithms, arguably the most common form of evolutionary computing, are a very well known optimization method for large and/or difficult search spaces, such as that of finding CA parameters with which to solve real problems ([1], [2]). The method consists of evolving and mutating a set of genes (specified system parameters) towards a global goal, defined by a fitness function. The fitness function itself is a measure of how successful each specific gene is, so finding an appropriate fitness function for a given goal is the main objective of much of the work in this field.

In order to find which regions of an installation people are actively interested in, a required part of our fitness function, there is a need to discover first where the people are within the installation. The majority of the work in tracking human movement ([3], [4], [5], [6]) consists of attempting to recognize people in video sequences. Unfortunately, due to the small budget and large scale of the installations, as well as the low-hanging, visually-diffusive elements involved, video taken from the top or the sides will not be able to see much detail other than peripheral information.

Other work ([7], [8]) focuses on tracking people using an infrared camera, in order to ignore small physical obstructions like those found in these installations. However, due to the frequent use of high-temperature shape memory alloy actuators in these series of installations, the body heat of people may be heavily masked from almost any angle.

Finally, the least prevalent possibility is to use the floor, either with capacitive sensors as explored in [9] or using temporary mats with built in pressure sensors, to track people's movements. In some installations this could work, depending heavily on location, but most of the instances of this series are placed in public spaces where there is no access or ability to cover the ground with anything.

III. Simulation Environment

Thanks to the tracking issues discussed in the background section, the only real location that any sensing solution can exist is along the top of the space that the people themselves occupy. Since the installations are fairly large, some form of distributed sensing must take place. Given the available budgets for the installations and the simplicity of the low-level electronics, distributed video is not an option.

This leaves a number of simple analog sensors as options, with the most economical and sensible option in this specific case being to use the built-in sensors in the hanging columns of the installation. This consists of a set of infrared distance sensors pointed directly at the ground from a known height, which can give a good estimate for the height of anything below. A distributed set of height measurements also has the advantage of enabling a distinction between specific people if they come together and then separate, if necessary.

Due to the length of the Nuit Blanche festival (12 hours) and large space required for these installations, a physical test bed does not currently exist, leading to the need for a functional simulation environment, discussed in Parts A and B of this Section.

A. Simulated People

In order to develop and test this work, a simulated set of height measurements was created to allow for testing of a range of possible scenarios without the need for a physical installation to test in. These measurements were based on a set of simulated people moving in a 6x6m area, viewed from overhead by a distributed set of simulated range sensors.

Each of the people was randomly initialized with a set of values based on height data from Statistics Canada [10] and anthropometric ratios from [11], shown in Fig. 2. The ranges are: standing head height, 150 - 190cm; body width, 40 - 60cm; and shoulder offset, 20 - 30cm below head height.

All people were modelled from above, represented in 2D as a circle, for the head, and an underlying ellipse, for the shoulders, with shading given by height. Each simulated person's head has a diameter of 25cm, while the shoulders have a minor axis diameter of 25cm in the heading direction (forward) and a major axis diameter equal to the body width.

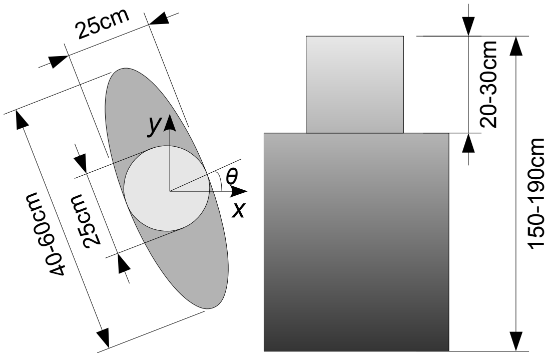

Fig. 2. Diagram showing the allowable size ranges of simulated people.

The state (***x***) of each person was governed by a simple model as shown in (1), where the position (*x, y*) is measured in centimeters and the heading (*θ*) is measured in radians. The inputs (***u***) are the speed in the direction of the heading (*s*) in centimeters per second and the angular velocity (ω) in radians per second.

$$\boldsymbol{x_t} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + s_t \cos(\theta_{t-1}) \\ y_{t-1} + s_t \sin(\theta_{t-1}) \\ \theta_{t-1} + \omega_t \end{bmatrix} + \varepsilon_t \qquad (1)$$

The disturbance to the state (ε) is modeled purely as a multivariate Gaussian distribution with a diagonal covariance matrix, with the non-zero members listed in (2).

$$\Sigma_{xx} = \Sigma_{yy} = 0.025, \Sigma_{\theta\theta} = 0.0001 \qquad (2)$$

The position and heading of each person were initialized at random from within the full set of possible values. As shown in (3), where N(*μ*,*σ*) is a normal distribution with a mean of *μ* and a standard deviation of *σ*, the input model is completely probabilistic to better simulate a human walking pattern in an open space.

$$\begin{gathered} \boldsymbol{u_t} = \begin{bmatrix} s_t \\ \omega_t \end{bmatrix} = \begin{bmatrix} s_{t-1} + \Delta s_t \\ \omega_{t-1} + \Delta \omega_t \end{bmatrix} \\ p(\Delta s_t = N(0,\ 2.5)) = 1\%, \mathit{else}\ \Delta s_t = 0 \\ p(\Delta \omega_t = N(0,\ 0.01)) = 1\%, \mathit{else}\ \Delta \omega_t = 0 \end{gathered} \qquad (3)$$

The speed was initialized to a random value from 15 to 90 cm per second, while the angular velocity was randomly chosen from -π/10 to π/10 radians per second. These ranges were determined heuristically to simulate human motion. In addition, to better simulate the observed motion of visitors to a museum or gallery, there a chance that the visitor will slow to a stop (*s=0*) for a random, non-zero length of time.

Although this model is only a very basic representation of full dynamic human motion, a more complete model such as [12] could be used in its place without significantly affecting the methods described in this paper.

B. Simulated Sensors

To simplify the simulation and boundary formulations, the installation used is a 6x6m square, with distributed hanging points located at the intersections of an overlaid rectangular grid. The grid consists of squares 25cm wide by 25cm long, in order to ensure that a person's head can never be detected on more than one sensor at a time, similar to the installation.

Each simulated sensor outputs a height measurement as given in (4), with the variance on the noise taken from experience with a number of infrared range sensors.

$$h_{x,y} = h_{x,y:true} + N(0,\ 0.025) \tag{4}$$

Once the sensors have found their measurements for the given time-step, they are sorted into four distinct classes. These classes take into account the possibility of noise in the measurements and so have slightly expanded boundaries compared to the possible ranges of heights for the related people. There are 4 classes as follows:

- Head, if h > 175cm;
- Head or Shoulder, if 145 ≤ h ≤ 175cm;
- Shoulder, if 115 ≤ h < 145cm;
- Nothing, if h < 115cm.

These classes are used in order to facilitate all forms of measurement interpretation as discussed in the next Section.

IV. Measurements

There are two main steps in interpreting the data that come from the sensor readings, both of which are discussed in this Section. First the active sensors are clustered into sensor groups, as discussed in Part A. Then, in Part B, the construction of a probability map using both the raw sensor classes and the sensor groups is outlined. In order to reduce computational time, only active sensors (those in a class other than Nothing) are considered in both sensor groups and construction of the probability map. Both the sensor groups and the probability map play important roles in the initialization, iteration, and error correction of the particle filters that are discussed in the next Section.

A. Sensor Groups

Sensor groups are clusters of one or more sensors which are likely to be seeing the same person. They are initialized by iterating through all active sensors (S) and grouping them together based on euclidean distance, as given in (5).

$$d_{ij} = \sqrt{(S_{i,x} - S_{j,x})^2 + (S_{i,y} - S_{j,y})^2} \tag{5}$$

Any two active sensors which are within the maximum person-width of each other are grouped. If two sensors are in range of each other, and one is in a group, then the non-grouped sensor must be in range of all sensors in the group to be added to the group. The initial sensor groups evolve with time: newly active sensors within one person-width of an entire sensor group are added, and group sensors entering the Nothing class are removed.

Rejection policies also exist in the groups in order to attempt to ensure that sensors detecting two different people are not in the same group. The simplest of these consist of each group only being allowed to have one sensor in the Head class, and all sensors in the group needing to be within half the maximum body width from the group centre.

Once the groups have been formed and no currently active sensors are without a group, the group centres are assumed to be very likely locations for people to be in.

B. Probability Map

At each time-step, a probability map (with a resolution of 2.5cm) is uniformly assigned a low, non-zero probability value at each point, prior to the addition of a set of sensor-based high-probability regions. The regions, whose shapes and sizes are described below, are integrated into the map additively such that the probabilities are cumulative. Once all regions have been added, the values are normalized using the highest probability point in the map.

The regions are each created based on using either sensors directly or the sensor groups found in Part A of this Section. There are high-probability regions created and centred around each of the active sensors, with different shapes and sizes of region based on the various classes the sensors are in. The probability of a given point in a region due to a particular sensor is strictly a function of its distance from that sensor as is seen in (6).

$$d_S = \sqrt{(S_x - x)^2 + (S_y - y)^2} \tag{6}$$

Each of the different active sensor classes creates a uniquely shaped region with equally high probabilities (*K%*). Since the map is normalized once constructed, the value of *K* is arbitrary. If the sensor is in the Head class, then the region is a circle with radius 12.5cm as shown in (7). This is due to the fact that if the sensor reads a Head, then the furthest the centre of the person can be is 12.5cm from the sensor, due to a head having a diameter of 25cm.

$$p(x, y \mid d_S \leq 12.5) = K\% \tag{7}$$

If the sensor is in the Shoulder class, the region is a ring with an outer radius of 30cm and an inner radius of 12.5cm as shown in (8). Since a definite shoulder reading can only happen if the sensor is outside of the head but still within half the maximum body width, the region looks like a ring.

$$p(x, y \mid 12.5 < d_S \leq 30) = K\% \tag{8}$$

Finally, if the sensor is in the Head or Shoulder class, then the region is a circle with radius 30cm as shown in (9). As the measurement could be either a head or a shoulder, the region is a union of the head and shoulder regions.

$$p(x, y \mid d_S \leq 30) = K\% \tag{9}$$

In addition to the individual sensors contributing high probability regions, the sensor groups create regions of their own that always overlap with their associated sensors. These are also strictly a function of distance, but in this case it is the distance from a given point to the group as seen in (10), where the group position (G) is given by the average of the position values over all sensors (S) in the group.

$$d_G=\sqrt{(G_x-x)^2+(G_y-y)^2} \tag{10}$$

There are only two distinct regions that are created from the sensor groups. In the case of 1 active sensor (G_{num} = 1), similar reasoning to that of a Head or Shoulder class sensor applies to create a 30cm radius circle, as seen in (11).

$$p\left(x, y \mid d_G \leq 30,\ G_{num}=1\right)=K\,\% \tag{11}$$

In the case of 2 or more active sensors ($G_{num} \geq 2$), the distance of the group centre from any person's location will be a maximum of 18cm, giving an 18cm radius circle as shown in (12).

$$p\left(x, y \mid d_G \leq 18,\ G_{num} \geq 2\right)=K\,\% \tag{12}$$

In order to clarify how the regions interact once they are integrated into the probability map, a couple of possible combinations are shown in Fig. 3. These are just two of many different possible combinations that can occur.

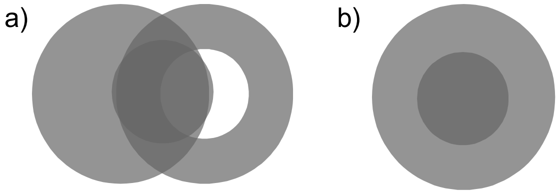

Fig. 3. Diagram showing two possible combinations of probability regions, where darker regions indicate a higher probability of being the actual position of a person: a) a Shoulder or Head sensor (large circle) and a Shoulder sensor (ring) side by side in a group (small circle) together; and b) a Head sensor (small circle) in a group of its own (large circle).

V. Particle Filters

Particle filters were selected for this work because of the non-linear nature of both the measurement model and the motion models used. Since non-linear models usually require several modifications that remove guarantees of optimality, a solution was found which uses a set of particle filters. Also, since this work will eventually be implemented on a distributed set of microprocessors, an inherently discrete solution at multiple levels provides many methods of division of labour in order to allow real-time computation.

One of the main limitations in particle filters is particle deprivation [13]. This can happen when there is little to no new information presented, and results in the particle filter focusing all of its particles in one small area. This focus can cause the filter to ignore future conflicting information. Unfortunately, in a number of different possible positions a person can be setting off none of the sensors in their region. Other times a person may stop, or be spinning in place, which will lead to a constant sensor reading. Both of these scenarios result in particle deprivation when only one particle filter is used to track all people in the installation.

In order to avoid this issue and make deprivation a useful attribute, one filter is created and assigned to each likely person, given by the centre of each sensor group. In this way, the filter will be narrowly distributed, and the probability map in the vicinity of a sensor group will encourage an even narrower distribution. Therefore, deprivation of the filters will actually lead to a better estimate of a particular person's state within the installation, as it will improve the focusing of particles in a particular filter on one specific person.

The remainder of this Section will deal with the following aspects of each particle filter: the initialization in Part A, updating and resampling in Part B, estimating the overall state in Part C, and error correction in Part D.

A. Initialization

Each particle filter is initialized based on a new sensor group being found, and the filter is associated with the group in order to allow for later error correction. The locations of particles within the filter are normally distributed around the group centre while the heading of each particle is randomly chosen from the full range as shown in (13).

$$\boldsymbol{x}_{p,0}=\begin{bmatrix} G_x+N(0,\,9) \\ G_y+N(0,\,9) \\ random(0,\,2\pi) \end{bmatrix} \tag{13}$$

The inputs to each particle are initialized in the same range as the input model for the people, as shown in (14).

$$\boldsymbol{u}_{p,0}=\begin{bmatrix} random(15,\,90) \\ random(-\pi/10,\,\pi/10) \end{bmatrix} \tag{14}$$

B. Update

The state update of the particles is done using the same kinetic model as the simulated people as given in (1,2) so it will not be repeated here. However, the change in inputs that the people will use are unknown, so the input update for the particles in a given filter will have a zero-mean, normally distributed, additive disturbance as shown in (15).

$$\boldsymbol{u}_{p,t}=\begin{bmatrix} s_{p,t-1}+N(0,\,1) \\ \omega_{p,t-1}+N(0,\,0.01) \end{bmatrix} \tag{15}$$

Once the motion model update has been completed, each particle is given a weight taken from the probability map based on their estimated location. Based on these weights, a cumulative weight density function is created from all particles in a given filter. This function is uniformly sampled and evaluated in order to build a new set of particles from the old. This process is known as resampling and will allow for the old distribution of particles to more closely match the true state as modified by the new measurements [13].

C. Estimate

The resampled particles are used to find an estimate for each filter's overall state, given in the same way as the simulated people or particles. In order to do this, the state is assumed to be made up of independent Gaussian distributions, leading to the need only to find the mean and variance of each state variable. In the case of the location variables, this is fairly simply done with a basic average and simple variance as given in (16).

$$E[a]=\frac{1}{P}\sum_{p=1}^{P} a\,,\; Var(a)=E[a^2]+E[a]^2 \qquad (16)$$

However, in the case of the heading variable the formulas in (16) do not hold because of the periodic nature of the values. In this case, a circular mean and variance are necessary. To accomplish this, the heading is assumed to be the angular component of a location on the unit circle in polar coordinates, which can easily be converted into a location in Cartesian coordinates using basic trigonometry. An average location can then be found in Cartesian coordinates which is converted back into polar coordinates. The angular component of this polar location will be the correct average heading as shown in (17).

$$\boldsymbol{\mu}=\begin{bmatrix}\mu_x\\ \mu_y\\ \mu_\theta\end{bmatrix}=\begin{bmatrix}E[x]\\ E[y]\\ atan2\left(E[\sin\theta],E[\cos\theta]\right)\end{bmatrix} \qquad (17)$$

The radial component of the polar location will be a measure of how tightly grouped the headings were with a value of 1 being all equal and a value of 0 meaning uniformly distributed. In order to convert this value into a variance, it is subtracted from 1 as shown in (18).

$$Var(\boldsymbol{x})=\begin{bmatrix}E[x^2]+\mu_x^2\\ E[y^2]+\mu_y^2\\ 1-\sqrt{E[\sin\theta]^2+E[\cos\theta]^2}\end{bmatrix} \qquad (18)$$

The maximum distance of the estimated location compared to the simulated person being tracked was found to be 12.5cm, which is almost entirely due to the method of building the probability map and sensor groups using the assumption that a head is a 12.5cm radius circle.

Even when a person stops within the installation, which is typically a problem for particle filters, the estimated location stays within 12.5cm of the actual person. The only issue that arises from a stopped person is an increased variance on the heading, as without further sensor input there is no way to know if the person has stopped completely or is spinning.

Once an estimate is found, provided the error correction does not remove the filter from use, each filter's state estimate is used as the origin and direction of a field of interest as described in the next Section.

D. Errors

In order to ensure that the particle filter estimates are valid, various forms of error detection are applied. These errors, if detected, signal that the particle filter estimate is in one of three different states that all require that the filter be removed from use and reassigned in some way.

The first state occurs if an estimate has drifted more than 30cm away from the centre of its associated sensor group as shown in (19). When a filter estimate has moved this far away from its group, it is assumed that the filter is on a divergent path from that of the person being tracked. If a filter is found in this state, it is reinitialized using the location of its current sensor group.

$$\sqrt{(\mu_x-G_x)^2+(\mu_y-G_y)^2}>30 \qquad (19)$$

The second state occurs when the estimate leaves the installation boundaries, which happens in the general course of use as tracked people leave the installation. As such, this is the most often detected error state. In order to detect when this is the case, one or more of the logical statements in (20) must be true. The variable E is a heuristically chosen value (12.5cm is used) of how close to the edge an estimate should be before it is assumed that it will be leaving the installation. The values x_{MAX} and y_{MAX} are both 600cm (6m), due to the size of the simulation environment chosen. Also note that in this environment, the origin is in the upper left corner, with positive x to the right and positive y downwards.

$$\begin{array}{lll}\mu_x<E & \& \cos(\mu_\theta+\pi) & >0\\ \mu_x>(x_{MAX}-E) & \& \cos(\mu_\theta) & >0\\ \mu_y<E & \& \cos(\mu_\theta-\pi/2) & >0\\ \mu_y>(y_{MAX}-E) & \& \cos(\mu_\theta+\pi/2) & >0\end{array} \qquad (20)$$

Finally, the third state occurs when the filter's associated group orphans it (this can happen when a person stops in a position between sensors) and the filter estimate is left in the middle of open space for a time. When this happens, the variances over the state become very large very rapidly and the total weight of all of the particles gets close to zero.

In both the second and third states, the filter in question is removed from use and placed into a list of free filters. These filters are then used when new sensor groups are created as new people enter the installation.

VI. Interest Maps

The estimated interest map is built using the particle filters' state estimates for the locations and headings of people within the installation. Each estimate is only considered valid if the variance on the heading is less than 0.1, and the particle filter is associated with a sensor group.

If a filter passes both of these checks, then it is assumed that there is a trapezoidal field of interest projected from the estimated location in the direction of motion, the extremes of which are found based on the following assumptions: the viewing angle is 60°, centred around the estimated heading; the start of the field is 30cm from the estimated location; and the depth of the field is 150cm from the estimated location.

The estimated interest map itself is made by integrating all of these predicted fields of interest over time. The fields are each treated as a medium probability region, which are then overlaid together with a slightly eroded copy of the previous map estimate in order to emphasize recent interest more than past interest. This estimated interest map will slowly change over time to show what regions of the installation the people currently find interesting and which ones they do not.

In a similar fashion, with the same assumptions, and using the simulated people's actual locations and headings, a comparison is made through direct subtraction of the actual interest map from the estimated one. The values on both maps are chosen so that errors can be distinguished into false positives (estimate shows interest when people do not) and false negatives (estimate shows no interest when people do), along with true positives and true negatives (estimate and simulated people agree on presence/lack of interest). These can be seen distinctly as shown in Fig. 4.

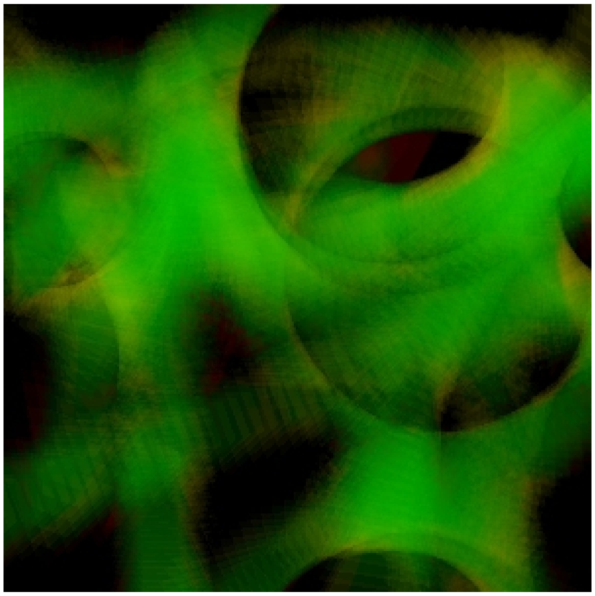

Fig. 4. Interest map comparison. In this image, the colour representations are: green for true positives, red for false positives, yellow for false negatives, and black for true negatives. One of the key causes of error in these interest map comparisons is the elimination then recreation of a sensor group when two people get very close to each other then move apart, suggesting better group rejection policies are required.

VII. Conclusion

As should be evident from Fig. 4, the performance of the system described in this paper in terms of being able to track and predict the interest of people moving through a large scale installation is very good. After more than fifty executions of the simulation with anywhere from one to ten people, the average percentage of true interest points out of the overall interest map comparison was 92%. Also, of the errors, the false negatives were an average of 6% of the comparison, while false positives were only 2%.

Some future possibilities that are planned for this work include the incorporation of better pattern recognition in the measurement interpretation task in order to: create a more useful probability map (and possibly use different values of K for different sensor classes), be able to initialize the filters with some form of subset of possible headings, and improve group rejection when people are close to one another.

Another future improvement is the possibility of extending a single particle filter to track all people in a subsection of the installation and perform hand-offs between them in order to better correct for individual filter errors. This will allow for removing the dependence on the deprivation problem.

Finally, this work could benefit from a number of iterative improvements related to the interest fields themselves and the human motion data: a more realistic model of the human motion (such as found in [12]) would improve the testing and tracking of real people; using real data from an installation along with verification of the true motion would allow a better correlation between measurements and truth; and a better idea of what the field of interest actually looks like based on a person's orientation would directly inform the fitness function.

Acknowledgement

I would like to thank Professor Steve Waslander for some essential insights and extensions on this research work.

References

[1] M. Mitchell, J. P. Crutchfield and R. Das, “Evolving Cellular Automata with Genetic Algorithms: A Review of Recent Work,” Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA '96). Moscow, Russia: Russian Academy of Science, 1996.

[2] A. Ugur, and M. Conrad, "Building Evolution Friendliness into Cellular Automaton Dynamics: The Cytomatrix Neuron Model," in Proceedings of the 1999 Congress on Evolutionary Computation (CEC99, Washington DC, USA, July 1999), Vol. 3, pp. 2071-2077, IEEE, Piscataway, NJ, 1999.

[3] T. B. Moeslund and E. Granum "A Survey of Computer Vision-Based Human Motion Capture", in Computer Vision and Image Understanding, Vol. 81, No. 3, pp.231 -268, 2001.

[4] L.Bazzani, D.Bloisi, V.Murino, “A Comparison of Multi Hypothesis Kalman Filter and Particle Filter for Multi-target Tracking,” 11th IEEE Int’l Workshop on Performance Evaluation of Tracking and Surveillance PETS 2009, Miami, FL, USA, June 2009.

[5] C. Chang, R. Ansari, A. Khokhar, "Multiple Object Tracking with Kernel Particle Filter," CVPR, Vol. 1, pp.566-573, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) – 2005.

[6] M. Perše, M. Kristan, J. Perš, G. Vučkovič, S. Kovačič, “Physics-Based Modelling of Human Motion Using Kalman Filter and Collision Avoidance Algorithm,” International Symposium on Image and Signal Processing and Analysis, ISPA05, Zagreb, Croatia, pp. 328-333, 2005.

[7] H. Nanda, L. Davis, “Probabilistic Template Based Pedestrian Detection in Infrared Videos,” Proc. IEEE Intelligent Vehicles Symposium 2002, Paris, France, 2002.

[8] E. Goubet, J. Katz and F. Porikli, "Pedestrian Tracking using Thermal Infrared Imaging", Mitsubishi Electric Research Laboratories, Technical Report, TR2005-126, 2005.

[9] M. Valtonen, J. Maentausta, and J. Vanhala, “TileTrack: Capacitive human tracking using floor tiles”, in Proc. IEEE International Conference on Pervasive Computing and Communications 2009, pp 1-10, Los Alamitos, CA, USA, 2009.

[10] Statistics Canada, “Canadian Health Measures Survey: Cycle 2 Data Tables”, August 2009 – November 2011, Catalogue No. 82-626-X.

[11] D. A. Winter, “Biomechanics and Motor Control of Human Gait: Normal, Elderly and Pathological”, Waterloo Biomechanics, 1991.

[12] M.H. Ghaeminia, A.H. Shabani, and S.B. Shokouhi, “Adaptive Motion Model for Human Tracking Using Particle Filter”, in Proc. Of 20th International Conference on Pattern Recognition (ICPR) 2010, pp. 2073-2076, Istanbul, Turkey, 2010.

[13] A. Doucet and A.M. Johansen, “A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later”, The Oxford Handbook of Non-linear Filtering, Oxford University Press, Oxford, UK, 2008.