

# Automatic Pan-Tilt-Zoom Calibration in the Presence of Hybrid Sensor Networks

Christopher R. Wren  
MERL Research  
201 Broadway  
Cambridge, MA, USA  
wren@merl.com

U. Murat Erdem  
Boston University  
111 Cummington Street  
Boston, MA, USA  
merdem@bu.edu

Ali J. Azarbayejani  
MERL Technology  
201 Broadway  
Cambridge, MA, USA  
ali@merl.com

## ABSTRACT

Wide-area context awareness is a crucial enabling technology for next generation smart buildings and surveillance systems. It is not practical to cover an entire building with cameras, however it is difficult to infer missing information when there are significant gaps in coverage. As a solution, we advocate a class of hybrid perceptual systems that builds a comprehensive model of activity in a large space, such as a building, by merging contextual information from a dense network of ultra-lightweight sensor nodes with video from a sparse network of high-capability sensors. In this paper we explore the task of automatically recovering the relative geometry between a pan-tilt-zoom camera and a network of one-bit motion detectors. We present results for the recovery of geometry alone, and also recovery of geometry jointly with simple activity models. Because we don't believe a metric calibration is necessary, or even entirely useful for this task, we formulate and pursue the novel goal we term *functional calibration*. Functional calibration is the blending of geometry estimation and simple behavioral model discovery. Accordingly, results are evaluated in terms of the ability of the system to automatically foveate targets in a large, non-convex space, not in terms of pixel reconstruction error.

## Categories and Subject Descriptors

I.2.10 [Vision and Scene Understanding]: Motion; I.2.10 [Vision and Scene Understanding]: Video analysis; I.4.8 [Scene Analysis]: Sensor fusion; I.2.9 [Robotics]: Sensors; C.3 [Special-purpose and application-based systems]: Real-time and embedded systems

## General Terms

Sensor Networks, Video Surveillance, Adaptive Systems

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

VSSN'05, November 11, 2005, Singapore.  
Copyright 2005 ACM 1-59593-242-9/05/0011 ...\$5.00.

## 1. INTRODUCTION

The next generation of smart buildings will manage security, efficiency, comfort, and safety with help of global context gleaned from a network of sensors. It is not feasible to completely cover a building with high-capability sensors, such as cameras. Nor is it desirable to incur the significant complexity and brittleness that often comes with attempts to infer the data lost by leaving large tracks of a building completely unobserved. As a result, building-wide context will be sensed by a heterogeneous network of sensors that will include high-cost, high-capability nodes such as pan-tilt-zoom (PTZ) cameras, but will be dominated by swarms of low-cost, low-capability sensors that will provide most of the coverage. The manual calibration of such a network would be prohibitively expensive even if it were necessary only once. Buildings are continually reconfigured over their lifetime to support new uses, so a one-time calibration is almost certainly insufficient, and therefore any comprehensive context awareness system should continually adapt to these changes.

This paper examines the task of automatically adapting a PTZ camera to a network of one-bit motion detectors, for the purpose of creating a system that can automatically foveate the sources of events generated by the motion-detector network. The classical solution involves a specialist performing a labor-intensive site survey[17]. Another approach is to generate a known, or very easy to detect pattern of motion: such as having a person or robot navigate an empty space while following a pre-determined path[16]. These approaches place severe constraints on the system. We advocate a system that places as few constraints on the building inhabitants as possible. By accepting unconstrained motion as input we ensure that our method will be acceptable to the widest possible set of situations; we reduce cost by eliminating the need for robots, or for a specialist to visit the site; and we allow the system to be as responsive as possible to changes in the space by utilizing, on a daily basis, the constant flow of information provided by the motion of the inhabitants.

We achieve this flexibility and generality by adopting a functional definition of calibration. We seek to recover a description of the relationship between the camera and the environment that will allow us to make the best use of the PTZ camera. The classical solution estimates a metric calibration in the form of a map, tracks targets on the map, makes predictions, and controls the PTZ to acquire the desired images. As opposed to this series of marginal solutions

(which incur the overhead of performing all the marginal operations, such as target tracking at runtime), we propose a joint solution that directly estimates the objective: a policy that allows the PTZ camera to capture high-quality video of targets.

## 2. RELATED WORK

Over the past ten years there has been some interest in the calibration of camera networks in general, and in particular those containing pan-tilt-zoom cameras[5, 15, 3, 14, 11]. This interest was enhanced by the DARPA Video Surveillance and Monitoring initiative. Most of this work has focused on classical calibration between the cameras and some fixed coordinate system.

Camera network geometry discovery has also been exploited indirectly in Khan[8], where the objective is to find the pairwise camera field of view borders such that target correspondences in different views can be found and successfully inter camera hand-off can be achieved. On a more practical side, Trivedi[18] shows a nice example of a camera network with cooperating low and high resolution cameras in a relatively difficult outdoor (highway) environment.

Some effort has been expended to make these methods more tractable by combing mostly autonomous systems with structured light[2], calibration widgets[1], mobile robots [6], or surveyed landmarks[3]. However, we believe that these methods are still impractical because they either require too much labor (in the case of calibration widgets), components that are so far costly and unreliable (in the case of mobile robots), or place too many constraints on the host environment (in the case of structured light, or globally visible and surveyed landmarks). In any case they all assume that calibration will be done at setup, and make no provision for re-calibrating during operation.

The work of Stein[15], and later Stauffer[14] addresses this shortfall by relying on tracking data to estimate transforms to a common coordinate system for their camera networks. This work has the advantage of not distinguishing between setup and operational phases: any tracking data can be used to calibrate, or re-calibrate the system. Neither of these approaches directly addressed the question of PTZ cameras. More importantly their approach places severe constraints on the sensors used in the network: the sensors must not only be able to report very detailed position data for targets, but must also be able to differentiate targets sufficiently to successfully track them. This is true because tracks, and not individual observations, are the basic unit used in the calibration algorithms. Our approach allows us to operate with networks of extremely low-capability, low-cost sensors. Of course, our approach is also applicable to networks of expensive sensors.

Another fundamental difference between our approach and the literature cited so far is the validation metric. All the methods mentioned so far have the goal of recovering a detailed geometric model of the camera network. We advocate a more functional objective criterion: the ability to usefully employ the PTZ camera in response to events from the network. Specifically the ability to foveate the causes of those events. In this way our work is closest to the work of Rahimi[10], although in contrast, our system does not require high-capability sensors for tracking and does not restrict the space to a single occupant.

## 3. METHOD

We explore several related approaches, starting with the simplest and moving to the more complex. The unifying theme is policy-learning[7]: a stream of events come into the system, and the system consults a policy to choose action that will result in the most favorable result. In our case the stream of events are the activity detections from the context network, the actions are commands to the PTZ camera, the favorable result is capturing an image of the cause of the activity detection, and the policy is a look-up table that maps events to actions. The task of the estimator is to generate the best look-up table. This lookup table represents the functional calibration of the camera to the space: encoding relationships between the context sensors and the PTZ parameters that have previously resulted in high-quality video (as defined by some externally supplied value function: the presence of people, the presence of faces, or any other characteristic that can be algorithmically specified as a function of the PTZ video stream).

We assume in the text there is a single PTZ camera in the system. The techniques scale linearly in the number of cameras. In fact, the solution for each PTZ is independent of all the others and can therefore be pursued in parallel. This is because we set aside the difficult task of scheduling in the presence of multiple targets. Our algorithm will provide an attentional mechanism: the reflexive foveation of the camera to a peripheral stimulus. While reliable, it is reactive. However, it is possible to embed this algorithm as the reliable base for a any of the high-level camera scheduling solutions in the literature, such as[9].

In the following section we cover the estimation of the pan and tilt parameters from the interaction of the PTZ camera and the context network. Then in Section 3.2 we will address the issue of computing the optimal zoom setting given the recovered pan and tilt parameters.

### 3.1 Pan-Tilt Learning

The simplest policy table  $A_s$  is a vector where each entry  $a_\gamma$  maps the individual discrete events ( $\gamma \in \Gamma$ ) to specific PT parameters ( $\pi \in \Pi$ ). This is the form of the manually specified policy that we will use as our benchmark. Despite the fact that we call this form of policy simple, it is also the most complex policy that a human could be reasonably expected to specify manually.

To estimate each entry in the table we wish to find the parameters that cause the PTZ camera to view the same event that the context sensor is observing. We must find the one choice among the discrete set of PTZ parameters that generated a signal in the training data that is most similar to the signal from a particular context sensor. The signals should be similar if they are viewing the same underlying process. In fact, if a particular context sensor  $\gamma$  often views the same underlying process as the PTZ camera in a particular setting  $\pi$ , then those two signals should be more correlated than signals deriving from independent underlying processes. Therefore the best match should be:

$$a_\gamma = \arg \max_{\pi \in \Pi} \frac{R_{pc}(p_\pi[t], c_\gamma[t])}{R_{pp}(p_\pi[t])} \quad (1)$$

where  $p_\pi[t]$  is the event sequence generated by the PTZ camera in configuration  $\pi$ ,  $c_\gamma[t]$  is the event sequence generated by context sensor  $\gamma$ ,  $R_{pc}$  is the correlation between the two sequences, and  $R_{pp}$  is the auto-correlation of the PTZ event sequence [13].

Without loss of generality, we can assume that the events from both the context network and the PTZ camera can be modeled as a binary process. In this case the equation above becomes:

$$a_\gamma = \arg \max_\pi \frac{\|p_\pi[t] \wedge c_\gamma[t]\|}{\|p_\pi[t]\|} \quad (2)$$

where the  $\|\cdot\|$  operator represents the number of true events in a process, and the  $(\cdot \wedge \cdot)$  is the boolean intersection operator. This operation estimates the policy model by examining all the static relationships in the data: how the events coincide during a given instant. We call this estimator ‘‘Static’’.

The more powerful form of the policy is a matrix that captures the dynamic relationships in the data. Here we will consider only two-dimensional policies, where an action  $a_{\gamma\lambda}$  is chosen based on a sequence of observed events: a detection from sensor  $\lambda$  followed by a detection from sensor  $\gamma$ . A fixed-lag estimator chooses a particular offset  $\Delta t$  and attempts to model the dynamic relationships between event streams skewed in time. We augment the estimator with the new constraint:

$$a_{\gamma\lambda} = \arg \max_\pi \frac{\|p_\pi[t] \wedge c_\gamma[t] \wedge c_\lambda[t - \Delta t]\|}{\|p_\pi[t]\|} \quad (3)$$

This estimator will reject any actions that do not fit the exact time signature specified by the parameter  $\Delta t$ . In particular we would not expect the embedded static policy on the diagonal ( $a_{\gamma\gamma}$ ) within this larger dynamic policy to match that found by the static estimator above. That is because we do not expect the context sensors to typically generate sets of redundant events separated by exactly  $\Delta t$ . We can see that this is true from the results (note that the policies do not agree in table 1). We call this estimator ‘‘Dynamic’’.

To make the best possible use of the data available, and also to allow for more variability in the velocity of the inhabitants, we extend eq. 3 to admit a broader set of examples:

$$a_{\gamma\lambda} = \arg \max_\pi \frac{\|p_\pi[t] \wedge c_\gamma[t] \wedge \bigcup_{\delta=0}^{\Delta t} c_\lambda[t - \delta]\|}{\|p_\pi[t]\|} \quad (4)$$

Here the  $\bigcup$  operator is the union over the bit-streams. We use it here to allow the estimator to consider any event from sensor  $\lambda$  so long as it happened within a set window preceding the second event. Since the window extends down to  $\Delta t = 0$ , it also admits simultaneous events. This allows the estimator to also do a good job of building the embedded static policy elements on the diagonal,  $a_{\gamma\gamma}$ . We call this estimation strategy ‘‘Lenient Dynamic’’ or simply ‘‘Lenient’’.

## 3.2 Focal Length Estimation

It is not necessary for the above learning machinery to explore the space of all focal length settings for the PTZ camera. This is because the relationship between a wide-angle view of a scene and a tight framing of some sub-window of that view is well understood and easy to solve for in closed-form. The internal calibration necessary for these computations can be easily measured at the factory and burned into ROM. In fact, it is the case that PTZ cameras now come with embedded controllers that can foveate and zoom selected video sub-windows[12]. In any case autonomous internal calibration in the field is also possible[17].

We are only left with the task of selecting the appropriate sub-window. We will select views to exclude regions that are unlikely to have useful information. To define this mathematically, we need a function that specifies the value of a captured image as a function of image location:

$$v_{I,x,y} = V(I, i, j)$$

Where  $I$  is the captured image, and  $(i, j)$  is a location within the image. This function is application specific, and could be any function computable from the PTZ video stream, such as number of faces detected:

$$V_{face}(I, i, j) = \begin{cases} 1 & \text{There is a face at } (i, j) \\ 0 & \text{otherwise} \end{cases}$$

In this paper, we will use occupancy: the number of times a region was marked as foreground by a classical background segmentation algorithm[4].

$$V_{motion}(I, i, j) = \begin{cases} 1 & (I(i, j) - B(i, j)) > \tau \\ 0 & \text{otherwise} \end{cases}$$

Where  $B$  is a scene model, and  $\tau$  is a threshold. Specifically we will build a map that represents the all the moving objects observed by a particular PTZ parameter setting:

$$H_{motion}(\pi, i, j) = \sum_t V_{motion}(I_\pi, i, j)$$

The middle of Figure 1 illustrates one such histogram.

At the end of the policy learning algorithm in Section 3.1, we have assigned a context event,  $\gamma$ , to a particular view  $\pi$ . The value function assigns credit to motion that is seen in the appropriate context only:

$$V_{context}(I, i, j, \gamma, B, \tau) = \begin{cases} 1 & \gamma \wedge V_{motion}(I, i, j) \\ -1 & \neg\gamma \wedge V_{motion}(I, i, j) \\ 0 & \text{otherwise} \end{cases}$$

We can now compute a context-specific histogram by considering all the motion events that occur in the correct context as positive events, and all the motion events that occur in the incorrect context as negative events.

$$H_{context}(\pi, \gamma, i, j) = \frac{\sum_t V_{context}(I_\pi, i, j, \gamma, B_t, \tau_t)}{\|p_\pi[t]\|}$$

The right frame of Figure 1 illustrates such a context-embedded histogram. Notice that, even though the two histograms from Figure 1 are from the same scene, the context-gated version has emphasized a particular mode of data that corresponds to activity near a particular context sensor. So the pan-tilt policy estimation may assign two context sensors to the same wide-angle view. However, the refinement in this step may result in those context sensors being assigned to

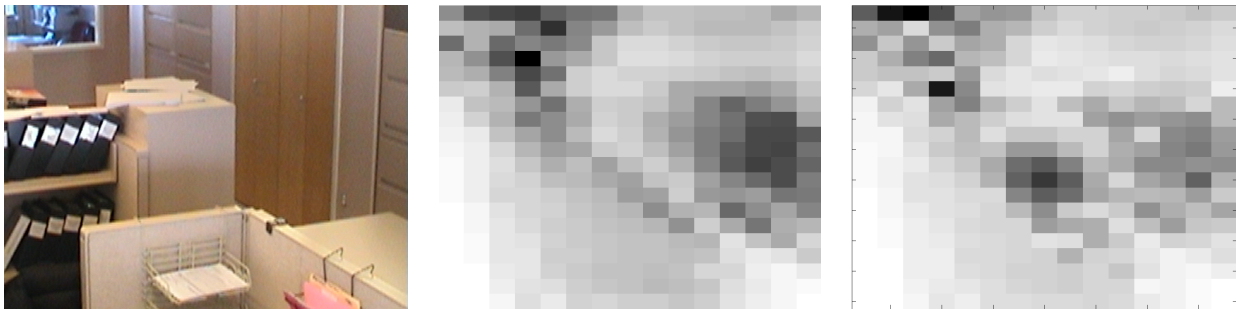


Figure 1: One of the finite camera views (left), the associated total activity histogram (middle, dark is high activity), and the activity histogram gated by a context event (right)

different pan, tilt, and zoom parameters that frame the data modes associated with the particular context sensor.

Once the histogram is determined, selecting the sub-window is a matter of trading off the probability of capturing high-value video against the risk of missing events. This application specific setting is the trade-off between, say capturing more faces with the wider field of view afforded by a short focal length, but getting fewer pixels on each face because the face images smaller than it would in a frame shot with a longer focal length.

#### 4. EXPERIMENT

The experimental facility is  $132m^2$  of inhabited office space that is home to approximately a dozen executives and administrators and their associated copiers, printers, and filing cabinets. The space also covers a high-traffic footpath that connects parts of the larger facility that lie outside the test area. The entire facility employs approximately 100 people.

The space is covered by a network of twenty motion detectors that observe the entire space. The space is also observed by a single PTZ camera. The majority of the space is occupied by low walls that allow the PTZ to view the upper bodies of standing people without obstruction. The PTZ is located so that it has unobstructed views down one entry hallway. Parts of the space is unobservable by the PTZ due to obstruction by walls. Please see the map in Figure 2 for details.

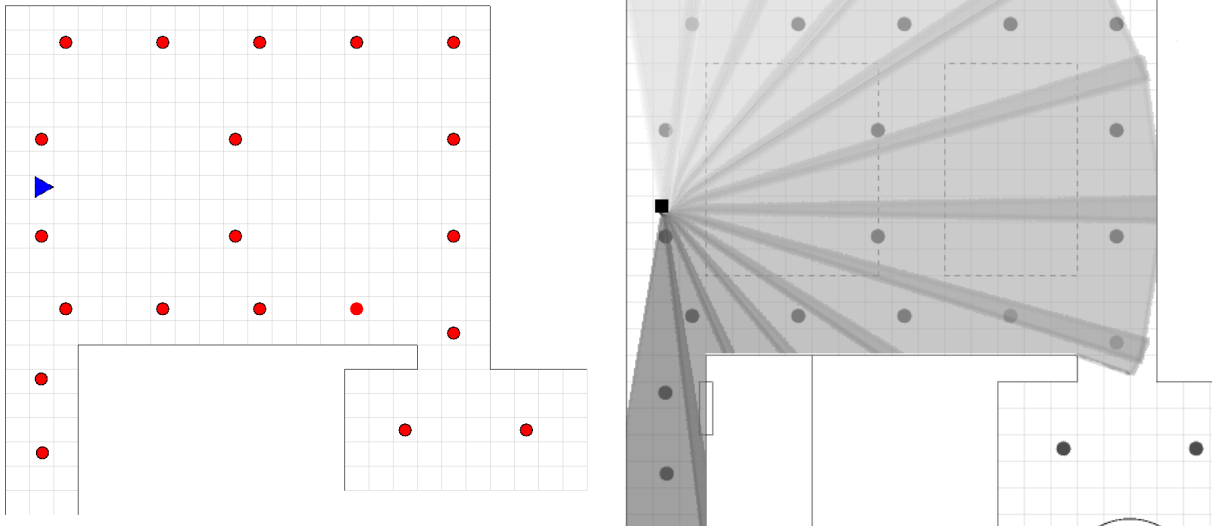
Data was collected from the motion sensors over a single 8-hour day. At the same time, the PTZ camera was set to scan sequentially through a discrete set of 12 evenly-spaced azimuth positions at a fixed altitude. At each location the PTZ recorded 100 frames of video at 30Hz. These frames were processed into motion event streams using standard background subtraction techniques. This data collection effort resulted in 240K events total from the twenty sensors in the context network and 40K events total from the 12 interleaved views from the PTZ stream.

The processing schemes described above were used to generate a variety of policies for the PTZ. In addition, a manually coded policy and a random policy were also generated. The random policy assigned one of the 12 discrete positions to each slot in the policy according to a uniform distribution. The manual policy was not restricted to the 12 discrete azimuth positions. It represents a human using their best judgment to calibrate the system.

Example policies are summarized in Table 1. The rows of the table represent the 20 individual context sensors. The

Label	Manual	Static	Dynamic	Lenient
S3	1	2	2	2
S2	1	1	1	1
E3	1	1	3	1
S1	2	2	1	2
F3	3	4	5	4
F2	4	3	6	3
F1	5	6	6	6
L2	5	6	7	6
L1	5	5	6	5
D1	5	5	5	5
F4	6	7	6	7
D2	6	5	5	5
D3	7	4	4	4
B4	7	1	1	1
D4	8	7	7	7
B1	8	2	2	2
B2	8	2	2	2
B3	9	10	10	10
E1	12	12	12	12
E2	12	12	10	12

Table 1: Some example policies. Please refer to the text for a complete explanation.



**Figure 2:** Left: The test space:  $132m^2$  office space with 20 motion detectors (circles) and one PTZ camera (triangle). Areas without detectors are personal workspaces surrounded by low walls. Right: the 12 discrete fields of view overlaid on the map.

columns represent a policy generated by one of the learning methods. Each entry shows the PTZ position index that should be chosen, according to that particular policy, in response to an event from that particular sensor.

Remember that the “Dynamic” and “Lenient” policies generate decisions only in response to pairs of events. Therefore, they may only be completely represented by a full matrix, such as shown below in Figure 4. Table 1 shows only the diagonal elements of that matrix, i.e.: the decision made by the dynamic policy in response to seeing a sequence of two events from the same context sensor. The dynamic policies are more expressive than the static policies. The diagonal of the dynamic policy matrix approximates the subset of decisions that are most closely related to the decisions being made by the static policies. We want the dynamic policies to function well for stationary targets as well as moving targets, so the diagonals of the dynamic policies are given alongside the static policies in Table 1 to help the reader judge this important subset of the overall performance. For a more complete understanding of the relative performance of the policies, please refer to the next section.

To generate validation data, an individual walked a 90 second serpentine path through the empty space that was designed to traverse every segment of walkway at least once in each direction. The path was repeated as uniformly as possible for each policy. Upon each event from the context network, the policy was evaluated, the PTZ was moved if necessary, and an image was captured. For the dynamic policies, if a single event was seen in isolation, then it was treated as a repeated pair, and the decision was therefore taken from the diagonal of the policy. The path had the potential to generate up to 44 events per execution, and therefore a maximum of 44 opportunities to correctly foveate the target.

The images were manually evaluated for partial and foveal hits. A partial hit means that the target was visible in the wide angle frame. A foveal hit means that the target was

Method	total hits	(%)	foveal hits	(%)
Manual	42	(95%)	24	(57%)
Lenient	37	(84%)	25	(68%)
Static	30	(68%)	19	(63%)
Dynamic	28	(64%)	8	(29%)
Random	8	(18%)	4	(50%)

**Table 2:** The results of the live experiment. Please refer to the text for a description of partial and foveal hits.

in the central 25% of the frame and could therefore have been captured in high detail by a more tightly zoomed shot. Figure 3 may help to clarify those definitions.

## 5. RESULTS

Table 2 summarizes the results of the validation runs. The best performer in total hits is the manually coded static policy that maps individual events to positions for the PTZ camera. The manually-coded policy captures 42 of the 44 possible opportunities (95%), but only 57% of those hits were solid enough to have allowed for tight zoom. The line marked “Static” is the attempt to learn this static mapping directly, and shows that the policy was able to capture two-thirds of the possible opportunities. For comparison, the randomized policy only managed an 18% hit rate, with half of those being marginal hits.

There are two dynamic algorithms on the table: the poorly performing “Dynamic” and the much better “Lenient”. “Dynamic” is the fixed-lag estimator from eq. 3. Its overall hit rate is similar to “Static” but it has an exceptionally high fraction of marginal hits. Very similar, poor results were obtained with a variety of values for the fixed-lag parameter  $\Delta t$ .



**Figure 3: Example images from the validation run. Left: a miss. Middle: a hit. Right: a foveal hit, because the target is in the center 25% of the image (marked by the dashed frame for illustration).**

The more admissive “Lenient” utilizes a larger pool of data by considering a large range of lags during estimation, and the resulting performance is very good. With an overall hit rate nearly as good as the manual policy, and with a much higher fraction of foveal hits (68%), it is easily the most successful automatic algorithm tested.

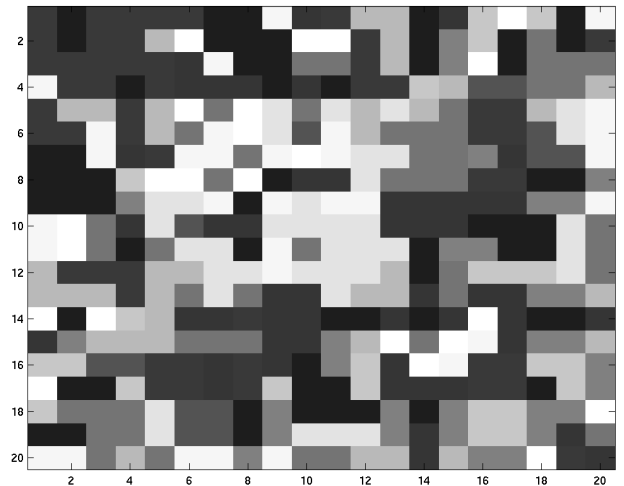
## 6. DISCUSSION

Even with a relatively small database of motion and a very limited set of PTZ positions the system was able to capture much of the human performance in a static policy. Given that the PTZ doesn’t just capture binary motion events, but reports the position of the motion within the frame, we believe that a successive refinement strategy could tune the policy to greatly improve the foveation rate, if not the overall hit rate. This could be facilitated by an estimate of the pan-tilt-zoom camera intrinsics[5], which we are not computing now.

The Lenient Dynamic algorithm is attempting to estimate a far larger set of parameters ( $N^2$  in the number of context sensors, or 400, compared to  $N$ , or 20, for the static policy) with the same data, however the performance is much higher, particularly with respect to the foveation rate. This can be understood by realizing that the static policy doesn’t model the movement of the target. Even with a very weak, pure-Markov policy, the foveation rate is significantly improved. This assertion is supported by the fact that the “Static”, “Dynamic” and “Lenient” policies largely agree about the correct action in the purely static case: they only differ in the dynamic case.

An example dynamic policy is shown in fig. 4. The diagonal of this matrix represents the embedded static policy. Those elements are used when two events arrive in sequence from the same sensor, and also when a single event is received alone, without a recent, preceding event. This could be expected to be the same as the static policy, although it is not constrained to be the same (see table 1). Everything off this diagonal is policy for the more typical case when an event from a sensor is followed by an event from a different sensor. It should be obvious that many, if not most of these entries are nonsensical, since the sensors may be nowhere near each other. In those cases policy is likely chosen solely based on noise in the training data.

We believe that further improvements in the performance of dynamic policies could be found by pre-filtering the run-time event set to ignore nonsensical pairings. These pairings will either be generated by noise in the system, or by



**Figure 4: The two-dimensional action table for the lenient-dynamics case. The row is determined by the source of the current event, the column is determined by the source of the prior event. Shades encode the PTZ positions (dark for position #1 though bright for position #12).**

coincidental events generated by multiple, independent inhabitants. In either case, executing those policies is likely to be detrimental to the performance of the system. It has been shown that one can reliably recover rough topology of these kinds of sensor networks[19, 10], so discarding event-pairs from non-adjacent sensors should be quite possible, and should result in a large improvement in performance.

It is worth noting that the nature of these systems make them very difficult to evaluate. The policy must be tested *in vivo*. It is not possible to record a dataset and then run a variety of policies, because the PTZ must be actuated in real time or the data is lost. It might be possible to work in simulation, but the model would have to be exceptionally detailed to capture all the aspects of the system that are relevant to the results.

## 7. CONCLUSION

We have shown that it is possible to create pan-tilt-zoom camera systems that can automatically calibrate to a network of simple sensors using statistical methods, and that these methods can approach human performance in their ability to create policies for the successful foveation of previously unobserved targets. We have further demonstrated that estimation of crude behavior models can be pursued jointly with geometry estimation, and that this provides a significant improvement in performance over estimation of static geometry alone. These are crucial enabling capabilities for the hybrid networks that will provide context awareness to the smart buildings of the future.

## 8. REFERENCES

- [1] Patrick Baker and Yiannis Aloimonos. Calibration of a multicamera network. In Robert Pless, Jose Santos-Victor, and Yasushi Yagi, editors, *The fourth Workshop on Omnidirectional Vision, Camera Networks and Non-classical cameras*, Madison, Wisconsin, USA, 2003.
- [2] J. Barreto and K. Daniilidis. Wide area multiple camera calibration and estimation of radial distortion. In Peter Sturm, Tomas Svoboda, and Seth Teller, editors, *The fifth Workshop on Omnidirectional Vision, Camera Networks and Non-classical cameras*, Prague, 2004.
- [3] Robert T. Collins and Yanghai Tsin. Calibration of an outdoor active camera system. In *Computer Vision and Pattern Recognition*, pages 528–534, Fort Collins, CO, USA, June 1999. IEEE.
- [4] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.
- [5] Richard I. Hartley. Self-calibration from multiple views with a rotating camera. In *The Third European Conference on Computer Vision*, pages 471–478, Stockholm, Sweden, 1994. Springer-Verlag.
- [6] Boyoon Jung and Gaurav S. Sukhatme. Cooperative tracking using mobile robots and environment-embedded, networked sensors. In *Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation*, pages 206–211, Banff, Alberta, Canada, July 2001.
- [7] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
- [8] S. Khan, O. Javed, and M. Shah. Tracking in uncalibrated cameras with overlapping field of view. In *Workshop on Performance Evaluation of Tracking and Surveillance*. IEEE, 2001.
- [9] Ser-Nam Lim, Larry S. Davis, and Ahmed Elgammal. A scalable image-based multi-camera visual surveillance system. In *IEEE AVSS*, Miami, Florida, July 2003.
- [10] Ali Rahimi, Brian Dunagan, and Trevor Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Computer Vision and Pattern Recognition*, pages 187–194. IEEE Computer Society, June 2004.
- [11] S.N. Sinha and M. Pollefeys. Towards calibrating a pan-tilt-zoom cameras network. In Peter Sturm, Tomas Svoboda, and Seth Teller, editors, *The fifth Workshop on Omnidirectional Vision, Camera Networks and Non-classical cameras*, Prague, 2004.
- [12] Sony Corporation. *SNC-RZ30 CGI command manual*, 2.0 edition, April 2003.
- [13] Henry Stark and John W. Woods. *Probability, Random Processes, and Estimation Theory for Engineers*. Prentice Hall, 2 edition, 1994.
- [14] Chris Stauffer and Kinh Tieu. Automated multi-camera planar tracking correspondence modeling. In *Computer Vision and Pattern Recognition*, pages 259–266. IEEE, July 2003.
- [15] Gideon P. Stein. Tracking from multiple view points: Self-calibration of space and time. In *Image Understanding Workshop*, Monterey, CA, USA, 1998. DARPA.
- [16] Sabastian Thrun, Dieter Fox, and WWolfram Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots (joint issue)*, 31 & 5:29–53 & 253–271, 1998.
- [17] Miroslav Trajkovic. Interactive calibration of a pan-tilt-zoom (ptz) camera for surveillance applications. In *Asian Conference on Computer Vision*, 2002.
- [18] M. M. Trivedi, A. Prati, and G. Kogut. Distributed interactive video arrays for event based analysis of incidents. In *International Conference on Intelligent Transportation Systems*, pages 950–956, Singapore, September 2002. IEEE.
- [19] Christopher R. Wren and Srinivasa G. Rao. Self-configuring, lightweight sensor networks for ubiquitous computing. In *The Fifth International Conference on Ubiquitous Computing: Adjunct Proceedings*, October 2003. also MERL Technical Report TR2003-24.