Building Adaptive Camera Models for Video Surveillance*

James W. Davis
Ohio State University
Columbus OH 43210 USA
jwdavis@cse.ohio-state.edu

Alexander M. Morison
Ohio State University
Columbus OH 43210 USA
morisona@cse.ohio-state.edu

David D. Woods
Institute for Ergonomics
Ohio State University
woods.2@osu.edu

Abstract

We address the limited automatic scanning functionality of standard PTZ camera systems. We present an adaptive, scene-specific model using standard PTZ camera hardware. The adaptive model is constructed automatically by detecting human activity in Motion History Images (MHIs) using an iterative candidacy-classification-reduction process. The target motion is quantified and employed in the construction of a global Activity Map, which in turn is used to direct or navigate the camera.

1. Introduction

In security surveillance there exists a large population of video surveillance cameras which provide pan/tilt/zoom (PTZ) functionality. As a result, in this work we focus on enhancing current functionality for single PTZ cameras. Many PTZ camera systems provide basic or naive automatic scanning technology. An example scan mechanism is Frame Scan, where the camera pans over one field-of-view, pauses, and then repeats. In complex environments, these generic algorithms typically result in sub-optimal scanning of the space due to camera position, obstructing scene structure, or non-uniform user interest across the scene.

In this work we develop a scene-specific, adaptive, focus-of-attention camera navigation model for video surveillance by automatically learning locations of high activity and directing the camera to sample these areas more frequently. The algorithm measures activity from a Motion History Image (MHI) [2] at each view across the full viewable field of a PTZ camera. This information is then used to construct a scene-specific model of interest (Activity Map). Several new scanning algorithms are presented that take advantage of this Activity Map for camera navigation.

Our work spans human detection/activity recognition, scene modeling, and scene scanning as focus-of-attention.

*Appears in Workshop on Applications of Computer Vision, Austin, TX, February 21-22, 2006.

Human Activity	Env. Noise	Cam. Noise
ped. walking	tree shaking	brick work
person biking	smoke/steam	building edges
moving vehicle	reflections	lamp posts

Table 1. Instances of motion patterns

Human activity recognition includes a broad range of approaches, which are succinctly described in [1]. Our work also spans scene modeling and analysis techniques which fall into two distinct categories: structure based and semantic-based [6]. Recent work by [8] alternatively employs a two camera system (master-slave) for focus-of-attention scene scanning.

2. Stage 1: Measuring Human Activity

In order to measure human activity from a fixed camera position (i.e., pan/tilt), we employ an iterative process to extract specific motion regions in a single MHI. In Sect. 2.1, we introduce the main categories of motion within MHIs. We then outline the human activity detection algorithm, beginning with building the MHI (Sect. 2.2). Then, the iterative extraction begins by examining individual MHI blobs for candidacy (Sect. 2.3). Next, the MHI blobs are classified as either the target motion class or noise (Sect. 2.4). Finally, our reduction step removes select noise pixels (Sect. 2.5). The iterative candidacy-classification-reduction repeats until all MHI pixels are classified as the target motion class or removed. The final segmentation image is quantified into a single activity measurement and used to construct a global Activity Map for the entire scene (Sect. 2.6).

2.1. Categories of Motion

Examination of MHIs from a commercial PTZ video surveillance camera typically shows three distinct categories of patterns. We label these general categories as hu-

man activity, environmental noise, and camera noise. Example sources of each category are provided in Table 1.

In typical surveillance videos, our target motion corresponds to pedestrians, groups of pedestrians, moving vehicles, etc. Consequently, we simply define human activity as any translating object with a minimum spatial size and temporal length. In Fig. 1 we provide examples of such human activity. Rather than using a specific activity detector (e.g., pedestrian templates [7]), we believe an MHI-based approach can be simple and effective at separating human activity motion patterns from the noise categories. We reemphasize here that we are including more than pedestrians in our definition of human activity, which is also motivated from interviews with security surveillance personnel.

2.2. Motion History Image

An MHI is a single image which represents extended temporal motion (> 2 frames). Current motion pixels are updated in the MHI using a timestamp, with higher MHI values corresponding to more recent motion

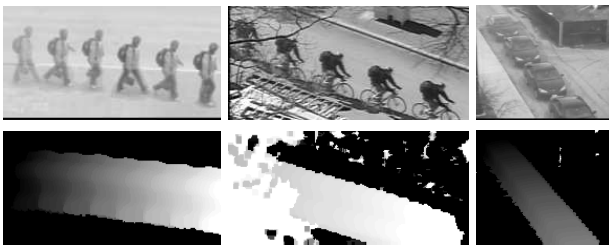
$$MHI_t(x,y) = \begin{cases} t & \text{if } |D_t(x,y)| > T_{Diff} \\ MHI_{t-1}(x,y) & \text{otherwise} \end{cases} \quad (1)$$

where D_t contains the difference images and T_{Diff} is the difference threshold. We set the temporal decay to the length of the entire (short) sequence (i.e., no pixels are removed). Details of the MHI technique are found in [2].

The long motion trails seen in the raw MHIs of Fig. 1 visually capture a basic property of translating objects. We refer to this property as "temporal consistency", in that, a translating object within an MHI will have a trail consisting of an intensity fade of extended temporal length. Furthermore, an intensity fade for a semi-rigid, constant velocity, translating object will have equal quantities of all MHI timestamps (for a particular duration). In addition, we conjecture that noise will not exhibit this strong temporal consistency, given the nature of noise sources (generally static). We therefore use temporal consistency to classify each MHI blob as a translating object or noise. In the ideal case of a blob consisting of only a translating object with minimal noise the classification process is as follows (we describe later how we manage significant noise contribution).

2.3. MHI Candidacy

First, for each MHI blob, we determine whether the blob is a potential candidate for human activity (i.e., translating motion). We define blob candidacy with the two properties of temporal consistency previously discussed, which are a minimum spatial size (i.e., $size(MHI\_blob) > T_{SpatialSize}$, e.g., 400 pixels for a 320×240 image)



(a) (b) (c)

Figure 1. Timelapsed images and raw MHIs for (a) Pedestrian, (b) cyclist, and (c) vehicle.

and a minimum temporal length (i.e., $max(MHI\_blob) - min(MHI\_blob) > T_{TemporalLength}$, e.g., 0.5 sec.). If an MHI blob fails either of these two criteria then the blob is not considered a candidate for human activity and its pixels are removed from the MHI. These thresholds can be tightened or relaxed depending on the application to identify more or less temporally/spatially significant signatures, respectively. All blobs that are selected as candidate human activity are passed to the classification stage for stricter evaluation of translating motion.

2.4. MHI Classification

In this stage, we examine the intensity fade of each MHI blob. For classification, if we assume that an MHI blob is indeed human activity (and therefore is a valid translating object), then the resulting MHI blob will have a trail of relatively equally spaced timestamps (i.e., spatially and temporally). As a result, classification of a candidate MHI blob (translating vs. noise) can be accomplished using a histogram similarity measure to quantify the degree of match between the normalized candidate MHI blob timestamp distribution and the ideal/expected timestamp distribution (uniform) for that blob, where normalization is based on blob size. A similarity value greater than a threshold, T_{class}, is considered valid translating motion.

The candidacy and classification approaches together are designed for classifying MHIs in the presence of minimal noise. In actuality, however, MHIs are more complex due to the presence and overlap of environmental camera noise (See Fig. 1(b)). These noise sources, when attached to the MHI intensity fade of a true translating object, will cause the current classification algorithm to fail (i.e., classify the blob as noise). A method is necessary to separate noise pixels from the translating object.

2.5. MHI Reduction

We remove MHI noise pixels using the gradient magnitude image of the MHI blob (motivated by [3]). Our MHI reduction algorithm is based on the gradient magnitudes of the MHI which provide a means to identify and remove pixels in order of confidence to translating objectness.

We first remove pixels with a gradient magnitude larger than a threshold value (corresponding to blob-background boundary pixels and some noise pixels). We then remove the most current timestamp pixels, since these pixels do not form part of the MHI intensity fade. Next, we fill in any zero gradient magnitude pixels (within regions of the same non-zero timestamp) using the following process. Each zero gradient magnitude pixel is assigned the average gradient value of the 8-connected neighbors having a gradient magnitude > 0. This process is performed iteratively.

Next, we recursively remove any remaining noise pixels. In order to tightly control the reduction method, a seed pixel is selected for the reduction that is the maximum gradient magnitude for the given MHI blob. From this seed pixel we recursively grow out 8-connected to all other pixels in the gradient magnitude image with magnitude greater than a growing threshold (T_{grow}). The value of T_{grow} is equal to the ith element of the blob's gradient magnitudes sorted in descending order, where i corresponds to a percentage (e.g., 10%) of the blob size. All pixels collected by the growing operation are then removed from the current MHI blob, which may result in one or more MHI blobs. Each of these blobs is returned to the candidacy stage for consideration as human activity. This iterative candidacy-classification-reduction process continues until all blobs are either classified as human activity (valid translating motion) or removed by the candidacy test.

The final binary segmentation image is converted into a single activity measurement to give a relative indication of the amount of activity for the sequence at a particular camera position. We selected a simple summation of pixels due to the difficulty of, segmenting overlapping blobs, blob fragmentation due to noise/partial occlusion, and scaling without knowledge of the ground plane.

2.6. Global Activity Map

We create an "Activity Map" for a camera's full viewable field using the local activity measures described above. In order to create this Activity Map, we divide the full field into $m \times n$, discrete pan/tilt locations (at a fixed zoom), which naturally results in a $m \times n$ rectilinear Activity Map. The rectilinear Activity Map simplifies construction (and the navigation algorithms in Sect. 3), and produces reasonable results. Building the map consists of visiting each pan/tilt location in the Activity Map in a random order, applying the activity detection algorithm at each location, and accumulating the activity measures over time (for each location). An example Activity Map is displayed in Fig. 2.

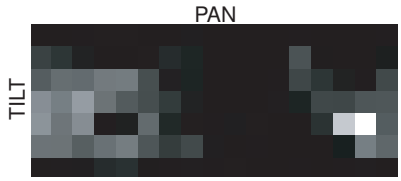


Figure 2. A 7×17 Activity Map. Brighter areas correspond to locations with more activity.

3. Stage 2: Camera Navigation

In this section, we present several activity-based navigation methods that can exploit the Activity Map to improve scanning efficiency as compared to current automatic scanning algorithms. For each of the methods we describe the Activity Map interpretation and the scanning procedure.

Probabilistic Jump
For this navigation method the Activity Map is normalized (Activity Map values sum to 1) and considered as a pseudo-probability distribution. Each subsequent location for the camera is selected using a probabilistic roulette wheel sampling of the Activity Map. We limit the scanning memory/history to only the previous location. Goal selection is accomplished by probabilistically selecting a new location until the location selected is not equal to the current location. Then, the camera moves to this new location and the algorithm repeats.

Probabilistic Walk
This navigation method is similar to the previous jump method, except that the next location is selected probabilistically based on the 8-connected neighborhood of the current location. This implicitly creates a path since the distance between a location and any neighbor is a straight line of unit distance (Activity Map resolution determines unit distance). Similar to Probabilistic Jump, the history is limited to only the previous location.

Inhibited Probabilistic Walk
The next navigation method is a variation of Probabilistic Walk and uses a suppression mechanism to control the history. The approach was motivated by the saliency/saccade modeling method of [5]. The approach maintains an implicit history of recently visited locations using a spatio-temporal inhibition mask to decrease, and then slowly re-

cover, the normalized Activity Map values. Here we suppress the 5-connected neighbors in the opposite direction of the next location chosen. The probabilities are returned to original values using an inverted exponential decay function

$$Inhibit(t) = 1 - \exp(-\alpha * (t - t_0)) \quad (2)$$

where t_0 is the time the location was initially inhibited, and α is the inhibition rate (e.g., $\alpha = 0.1$). Note, other functions can create the recovery such as linear and exponential.

Reinforcement Learning Paths

In machine learning, the goal of reinforcement learning is to determine optimal actions given a set of rewards to maximize cumulative reward. Specifically, Q-learning is an iterative method for learning a function (Q) which provides the optimal policy when rewards and actions are deterministic.

For our domain, the Activity Map is considered as the reward function (R), the Activity Map locations are the states (S), and the move to any 8-connected neighbor is the set of possible actions (A). We simply select a set of M goal locations (G) based on a probabilistic selection of locations from the Activity Map. Then, for each goal location $g \in G$ we have a separate reward function R_g, which is the Activity Map modified to give g extremely high reward (extremely high activity). Each reward function R_g is then input to the Q-learning algorithm. The Q-learning algorithm finds the optimal path from each state s to each $g \in G$ using R_g, which is then stored and used for navigation.

Navigation consists of probabilistically selecting a new goal location based on the Activity Map values and a goal location history. The path is then provided from the Q-learning results for that goal. Finally, the history is updated with all goal locations visited along the navigated path.

4. Experiments

In this section we examine the performance of our algorithms. First, we provide results for the MHI-based human activity extraction method and provide experimentally determined threshold values. Next, we examine the resulting global Activity Maps. Lastly, the navigation techniques are compared using paths generated by each algorithm.

We used three Pelco Spectra III SE series dome cameras mounted outdoors on three different university campus buildings, two of which have overlapping views. Additionally, the cameras are mounted at varying heights (two, three, and four stories). Images are captured with a Matrox Meteor frame-grabber (RGB, 320×240, $\sim$ 12fps).

4.1. Measuring Human Activity

Our training set consisted of 59 MHIs, each containing one or more MHI blobs (object and noise). These train-

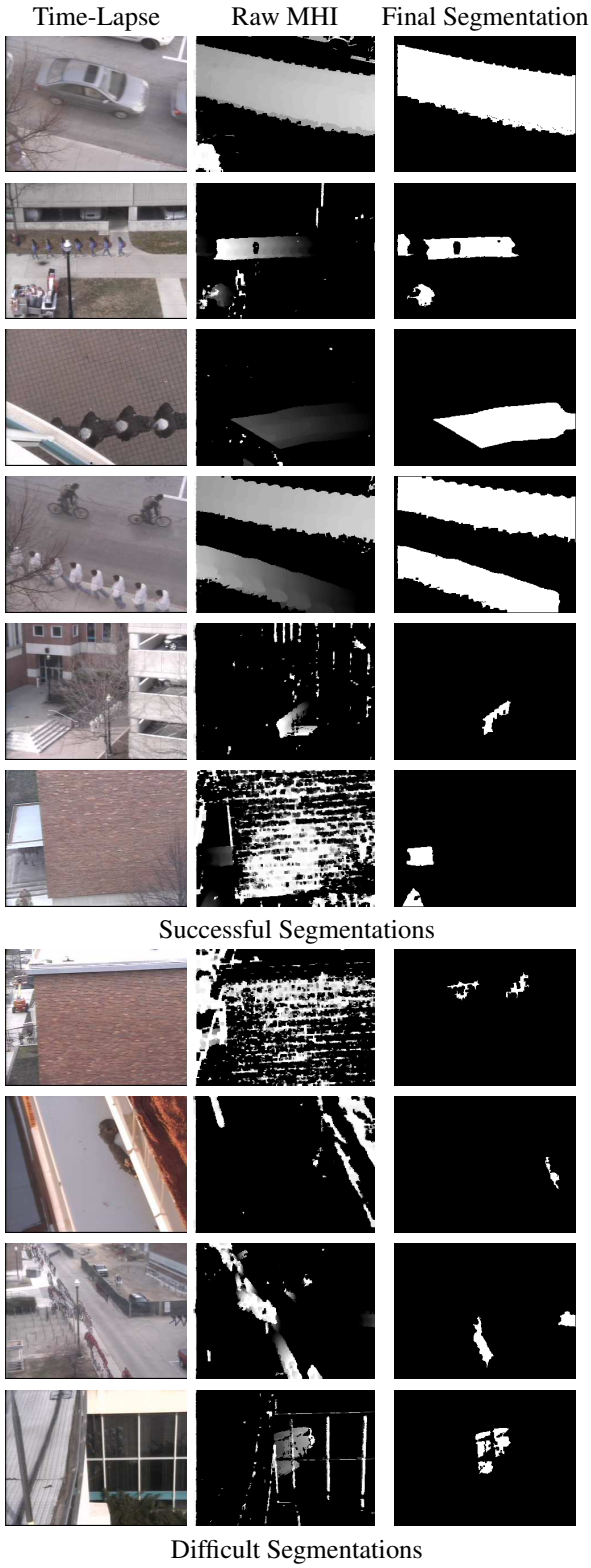
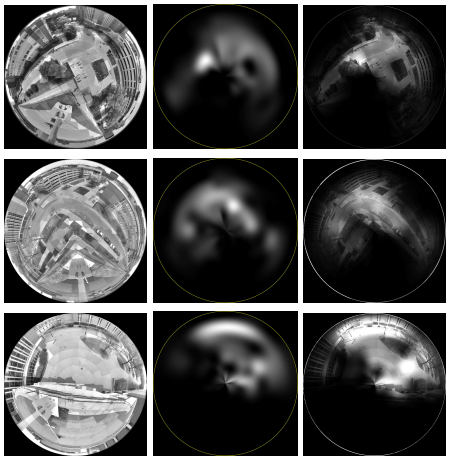


Figure 3. Examples of MHI segmentations for testing data.



(a) (b) (c)

Figure 4. Spherical images of (a) camera view, (b) Activity Map, and (c) overlays.

ing sequences were collected across the three cameras, at different times-of-day and days-of-week. The training images/locations are similar to the test data shown in Fig. 3. We manually segmented the MHI test images to learn the optimal algorithm parameters.

The timestamp similarity metric for classification was the Bhattacharya Distance, and the value for T_{class} was varied over a range of $0 - 1$. The percentage for the reduction threshold T_{grow}, defined previously, was varied over the range 4% – 18% of the MHI blob size. For evaluation, Precision/Recall data was collected for the segmentation results for each combination of T_{class} and T_{grow} compared to the manually segmented MHIs. In order to accurately compare these results, we used the F-measure (or harmonic mean) of Precision and Recall. The optimal thresholds were $T_{class} = 0.79$ and $T_{grow} = 0.10$ (based on F-measure). Four other similarity measures (Jeffrey Divergence, Minkowski-Form Distance, and Match Distance) were also tested, but none were an improvement.

To further evaluate our algorithm, we collected 10 additional passes of the full scene for each camera (10×119 MHIs) and computed the automatic segmentations (25 min. per pass). In the successful segmentation images of Fig. 3, a variety of different motion patterns are captured under varying situations. The algorithm detected a moving vehicle, a biker, and several pedestrians. Also, our approach captured small translating signatures, for example in the second row from a street vendor and a pedestrian under partial occlusion due to a tree. Finally, in the last successful segmentation two pedestrians are detected while a large amount of noise due to spatial aliasing (from brick patterns) is removed.

The previous results are extremely encouraging, however, our algorithm does have difficulty in some cases and we provide some examples in Fig. 3 (bottom set of images). In the first row, some of the spatial aliasing is included in the final segmentation. In the next segmentation a portion of the roof is included. This is due to the texture of the material and the proximity of the roof to the camera (3ft). In the third example, a large group of pedestrians is not detected as a result of the view angle and the MHI overlap. Finally, in the last example a pedestrian's reflection, as expected, is captured in a building window. From the highly varying data presented (view angle, scene complexity, occlusion, and camera noise), we believe our algorithm is a simple, yet robust, method for dynamically extracting translating motion from a scene.

For evaluation of the resulting global Activity Map, we analyzed the spatial consistency of a single camera and then multiple cameras with overlapping views. For comparison, we created spherical panoramas of each camera (Fig. 4(a)) and then warped the rectilinear Activity Maps into a spherical representation (Fig. 4(b)). Visually, the overlay images in Fig. 4(c) emphasize walkways and roadways and eliminate buildings and rooftops. Also, the first and second row overlays (Fig. 4(c)) have highly overlapping views. Even though no two passes were collected simultaneously the Activity Maps in the top two rows of Fig. 4(c) converged to similar activity emphasis across the same physical space measured from different positions (and time). Hence, the Activity Map captures an appropriate activity model for the full viewable field of a single PTZ camera and that the results are robust across space and time.

4.2. Navigation

We next compared the four activity-based navigation techniques presented. For Inhibited Probabilistic Walk, we set the inhibition rate $\alpha = 0.1$, and for Reinforcement Learning Paths the number of goal locations was set to 9. We note here, all zero-valued locations in the Activity Map are set to a value of 10% of the minimum non-zero Activity Map value (so all locations were reachable).

We provide a sample path of each navigation technique by overlaying the sample path on a spherical panorama Activity Map blend (see Fig. 5). For each algorithm we display 20 steps (camera moves) about the scene, and show the sampled Activity Map locations using disks labeled in sequential order (to designate multiple samples on a location we shift the disks and labels slightly). In addition, for the three "walking" techniques we also provide a line indicating the path between sequential locations.

The first navigation technique, Probabilistic Jump (Fig. 5(a)), illustrates how the highest probability locations are sampled multiple times while the lower activity areas are sampled less. The next navigation method, Probabilistic

Walk (Fig. 5(b)), tends to stay focused in local maxima. The next navigation technique, Inhibited Probabilistic Walk (Fig. 5(c)), eliminates the problem of becoming stuck in local maxima, as it tends to move away from recently sampled locations. Most notably, the low activity patches of grass in this panorama overlay are avoided. Finally, the Reinforcement Learning Paths (Fig. 5(d)) by design, moves along predetermined paths of high reward and avoids areas of low activity. Selecting a single optimal algorithm or combination depends on the user, application, and context. However, we have demonstrated that navigation methods based on Activity Maps are useful to more efficiently scan the scene as compared to existing methods. Additional experimental evaluations of the Activity Map and navigation techniques can be found in [4].

5. Summary & Conclusion

We presented an adaptive, scene-specific model using standard PTZ camera hardware to address the lack of scene-specific information used in current automatic camera scanning algorithms. The results show our MHI-based human activity measure captures a basic property of translating motion. This is demonstrated by our Activity Map, which accurately reflects the scene activity of a PTZ camera and is robust across both space and time. Overall, our current approach shows very promising results. In future work, we plan to examine additional features (e.g., color, texture), methods for temporal and spatial updating of the Activity Map, and incorporating multiple Activity Map scales.

References

[1] J. Aggarwal and Q. Cai. Human motion analysis: a review. In *Nonrigid and Artic. Motion Wkshp.*, pages 90–102, 1997.

[2] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 23(3):257–267, 2001.

[3] G. Bradski and J. Davis. Motion segmentation and pose recognition with motion history gradients. In *Proc. Wkshp. Appl. of Comp. Vis.*, Dec. 2000.

[4] J. Davis, A. Morison, and D. Woods. An adaptive focus-of-attention model for video surveillance and monitoring. *Machine Vision and Appl.*, to appear.

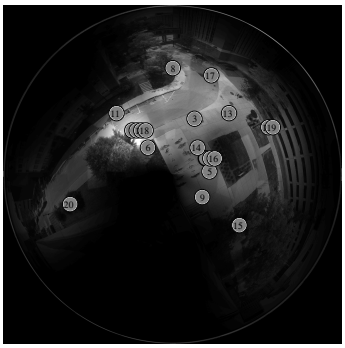
[5] L. Itti and P. Baldi. A principled approach to detecting surprising events in video. In *Proc. Comp. Vis. and Pattern Rec.*, pages 631–637, 2005.

[6] D. Makris and T. Ellis. Automatic learning of an activity-based semantic scene model. In *Advanced Video and Signal Based Surveillance*, pages 183–188, 2003.

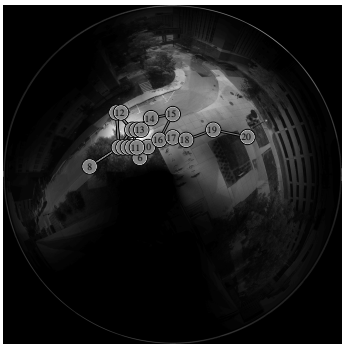
[7] M. Oren, C. Papageorgiou, P. Sinha, E. Osuma, and T. Poggio. Pedestrian detection using wavelet templates. In *Proc. Comp. Vis. and Pattern Rec.*, pages 193–199, 1997.

[8] X. Zhou, R. Collins, T Kanade, and P Metes. A master-slave system to acquire biometric imagery of humans at distance. *Int. Wkshp. on Video Surveillance*, pages 113–120, 2003.

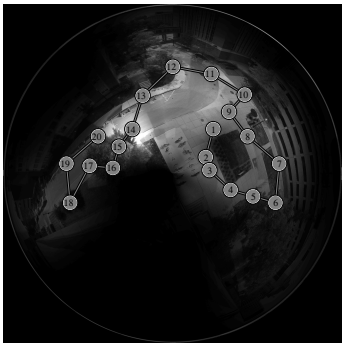
Probabilistic Jump



Probabilistic Walk



Inhibited Probabilistic Walk



Reinforcement Learning Paths
(goal locations are 1, 3, 7, 17, and 20)

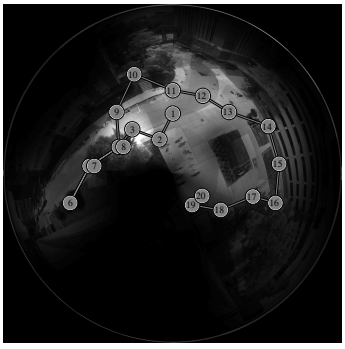


Figure 5. Paths for four navigation techniques.