LIGHTWEIGHT PEOPLE COUNTING AND LOCALIZING IN INDOOR SPACES USING CAMERA SENSOR NODES

Thiago Teixeira and Andreas Savvides

Electrical Engineering Department
Yale Univerisity
New Haven, CT 06520

{thiago.teixeira, andreas.savvides}@yale.edu

ABSTRACT

This paper presents a lightweight method for localizing and counting people in indoor spaces using motion and size criteria. A histogram designed to filter moving objects within a specified size range, can operate directly on frame difference output to localize human-sized moving entities in the field of view of each camera node. Our method targets a custom, ultra-low power imager architecture operating on address-event representation, aiming to implement the proposed algorithm on silicon. In this paper we describe the details of our design and experimentally determine suitable parameters for the proposed histogram. The resulting histogram and counting algorithm are implemented and tested on a set of iMote2 camera sensor nodes deployed in our lab.

1. INTRODUCTION

The integration of wireless sensor networks and camera technologies is rapidly pacing towards a new generation of low-cost, low-power camera sensing nodes. To enable a large number of ubiquitous applications and services, these wireless camera nodes should be able to recognize, count and track humans, preferably anonymously as they move inside buildings. The BehaviorScope project at Yale [**?**] uses such information to infer people behaviors, with assisted living as the application focus. Human locations collected from a sensor network deployed inside a house are processed together with building map information to recognize the activities of the house inhabitants. The locations of people in time and space during the course of the day provide a set of macro-gestures that are parsed by a framework of Hierarchical Probabilistic Context Free Grammars into a set of predefined activities [1].

Assisted living as well as other similar applications in security, workplace safety and entertainment share a common sensing requirement. The sensor network should cover large indoor spaces localizing and counting people as they move about the space. Furthermore, for practical purposes, this sensing has to be done without requiring people to wear badges; it should be lightweight and low-cost, and should maintain some level of privacy. To address these demands, our research pursues the development of lightweight motion discriminative sensors that provide more precise information than Passive Infrared Sensors (PIR) but more lightweight and privacy preserving than cameras. Our approach to this challenge is to explore biologically inspired Address-Event architectures that operate at the pixel level instead of frame level to provide feature information and ultra-low power operation.

Our previous work in [2] provided an initial evaluation of new address-event (AE) imager architectures and a model for emulating this architecture on wireless sensor nodes. In this paper, we present and evaluate a design for localizing and counting people in indoor spaces with a set of wide-angle camera sensor nodes mounted to the ceiling, facing down. Our design targets the architecture presented in [2] and localizes and counts people using a histogram derived from motion and size information. The resulting algorithm was implemented on a camera sensor node and is currently deployed on a home testbed for assisted living. The main contribution of the work described here is the design and evaluation of a lightweight histogram-based method for counting and localizing people using motion and size information and the fact that this can be applied on the motion outputs of an address-event image sensor. It is implemented on a real testbed currently deployed inside a house for an assisted living application.

The rest of the paper is organized as follows. Section 2 provides some background to the problem and surveys the related work. Sections 3 and 4 outline our approach and describe the details of the motion histogram. Section 5 explains how a sensor node uses the histogram to localize and count and Section 6 presents our experimental results. Video demonstrations of our experiments are available at http://www.eng.yale.edu/enalab/behaviorscope/counting.htm. Section 7 concludes the paper.

1-4244-1354-0/07/$25.00

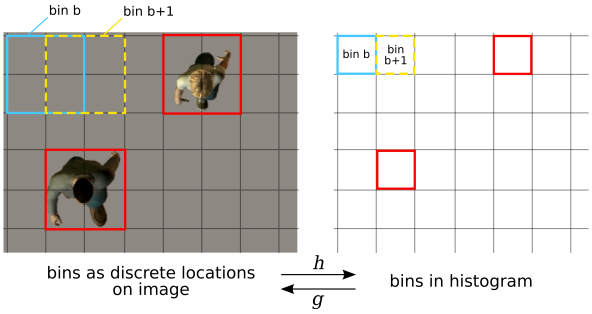


Fig. 1. Histogram structure: histograms are composed of multiple bins defined from overlapping areas in the image (left). The bin size is calculated from human dimensions and each bin can be uniquely identified its top-left corner position. Using these positions, a more traditional representation of the histogram may be composed (right).

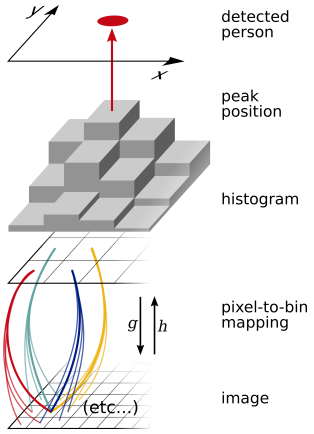


Fig. 2. Detecting positions from motion images: each pixel in the image is mapped to one or more histogram bins (as shown in Figure 1). Bin values are incremented for each foreground pixel the bin contains. Histogram peaks detect people's positions. Note that, for simplicity, this diagram shows each bin connected to 4 pixels. In reality, bins encompass many more pixels.

2. BACKGROUND AND RELATED WORK

Traditionally, human-tracking is achieved by, first, detecting the people that are visible in each frame and, afterwards, tracking them across multiple frames utilizing either extracted features (such as size, color histograms, edge orientation, etc.) [3], or motion correspondence using a Kalman filter, a particle filter, or other methods [4].

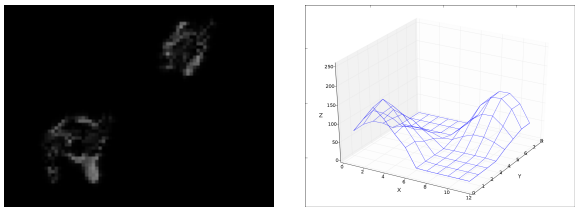
For the first part, that is, the problem of detecting of people in a video frame, the typical approach is to employ background differencing followed by a series of morphological operations in order to obtain a workable silhouette of a person to be "blobbed". Since the low-level morphological operations don't guarantee that each person translate to exactly one blob, a further pass has to be performed where blobs that are close enough are merged together. The end result is that it is common to merge blobs that do not belong together, as well as to separate blobs that compose the same object, as in [5]. This has the additional effect of adding uncertainty to the locations that are extracted from the blob. Some have attempted to obtain more precise locations from each blob by employing the distance transform [6] rather than center-of-mass or foot estimation, but that approach fails for fragmented blobs.

Moreover, there is the problem of maintaining and updating the background model. This is a necessary process due to the presence of a series of change factors in a stream of frames, among which are: (1) Natural oscillations in pixel intensity; (2) Gradual changes in lighting, such as those imposed by the movement of the sun; (3) Presence of repetitive background motion, such as waving foliage; (4) Changes in position of static objects, such as furniture.

Adaptive background-modeling approaches are computationally expensive, sometimes modeling each single pixel as a mixture of Gaussians [7] or with a separate Kalman filter [8]. Many of these approaches require the field of view to be empty at initialization — something that may not be possible in the practical settings we are interested in. Even then, most approaches either fail or recover slowly from at least one of the above scenarios, especially the last one, where an object is moved or a new object is introduced to the scene. In assisted-living and office situations, though, background changes occur very often. Take as an example the presence of office chairs, which are moved every time someone sits or stands. Other, simpler adaptative background-modeling techniques, such as continuously averaging frames, have the opposite effect of counting people that seldom move as a background objects.

Other research in counting untagged people includes the use a number of different types of sensors and techniques. [9] uses an array of PIR sensors in arranged in a line to detect the number of people through a flight of stairs. In [10], the center scanlines of an image frame is used for a similar effect. These approaches are used to detect people that pass through a confined area, and do not adapt well to open spaces. Moreover, since each person is only detected at entrances and exits, any error at detection time will propagate indefinitely.

Another approach [11] utilizes multiple cameras with largely overlapping field-of-views to get information about the 2-dimensional cross-section of a room, its objects and its occupants. The algorithm provides good location precision, but requires the use of multiple cameras to achieve that when covering even a small room. What is more, their approach demands precise calibration of intrinsic and extrinsic camera parameters. In the setting of assisted-living situations, however, the largest issue is that the computed cross-section fails



(a) **(b)**

Fig. 3. Example histogram frame: (a) moving pixels, (b) resulting histogram (video available)

to capture a person that is is sitting on a couch, lying down, or one that has fallen on the floor.

Some researchers utilize stereo cameras to assist in the image segmentation process, as is the case in [5]. In that paper, the authors describe their tracking system for assisted living. Their background model takes into consideration only the pixel intensity oscillations, and would fail in a less controlled environment. More importantly, their system does not handle rooms larger than the single stereo-pair's field-of-views.

In [12], binary edge-detected images are used in a people counter neural network. A different approach is taken in [13], where pressure sensors underneath floor tiles are the chosen sensing modality. The common problem shared by these methods is the laborious set-up process. In the first case, extensive training must take place at each new location. In the second, the installation of special floor tiles make it a cumbersome choice for existing homes.

3. OUR APPROACH

In light of the afore-mentioned problems with background differencing, we choose to shift the emphasis away from background subtraction into a different paradigm. The intuition behind this is that humans can recognize and count other humans based on shape, size and movement. The background differencing approach attempts to extract and operate on mainly the first two types of information. We choose to focus on the latter two, while at the same time simplifying them by introducing a set of constraints on the deployment and the environment: First, we assume that people inside the room are typically in motion. Even though this does not *always* hold, it is certain that it must be true for each person at some instant in time. Second, in order to cover a large area (requiring fewer sensors) and to minimize occlusions, we choose to place the wide-angle cameras on the ceiling, facing straight down. In this configuration, and given the ceiling height, it is fair to assume that human size lies within a certain pre-defined range. Using these two assumptions, we pose the following problem statement: to classify as a human each image entity that meets our movement and size criteria and extract their discrete physical location from our measurements.

In light of this, we construct a motion histogram from frame-differenced images and utilize that information to pinpoint each person's location. The histogram is designed to consider a typical human size in pixels, given the known characteristics of our camera and its position, and use it to compute the discrete human locations (histogram peaks) which best explains the moving pixels in the frame-differenced image. These locations can then be processed with higher-level algorithms to track each person and recognize their behavior [14]. The unique labeling of each human and the association problems that arise are not considered in this paper, as we focus on our lightweight sensing algorithm.

4. THE MOTION HISTOGRAM

4.1. Histogram structure

The primary goal of the motion histogram is to determine the probable location of each person given the coordinates of the moving pixels in each frame. The value of each histogram bin corresponds to the number of foreground pixels in a unique area of the image. In Figure 1, bin b is associated to the set of pixels in the blue square on the top-left side of the image. It is said that b contains those pixels. Therefore, the relation $g : H \mapsto I$ can be defined, mapping each bin in the histogram H to the set of pixels in the image I that it contains. Thus $g(b) = \{x : x \in b\}$. Conversely, $h : I \mapsto H$, $h(x) = \{b : b \ni x\}$ gives for each pixel the set of bins that contain it.

For each bin b, we define the $g(b)$ according to the size of a human and their possible physical locations. In the figure, adjacent bins overlap with each other, working as a discretized sliding window across the image in both the vertical and horizontal directions. The bin size is calculated from the expected image size of a human, so that, in optimal conditions, a person in the field of view of the overhead camera is entirely covered by a single bin, and partially covered by neighboring bins. In typical operation, though, people may span multiple bins (when they extend their arms, for example), but the algorithm described here still holds.

If the bin areas on the left side of Figure 1 are square with width w, and if the smallest distance between bin centers is δ, then g can be defined as

$$g(b) = \{x : x_x \in [b_x\delta, b_x\delta + w] \wedge x_y \in [b_y\delta, b_y\delta + w]\}$$

where b_x and b_y are the coordinates of bin b in the histogram and x_x and x_y are the coordinates of pixel x in the image. Similarly, the h for the histogram described in the figure is:

$$h(x) = \left\{ b : b_x \in \left[\frac{x_x}{\delta}, \frac{x_x - w}{\delta} \right] \wedge b_y \in \left[\frac{x_y}{\delta}, \frac{x_y - w}{\delta} \right] \right\}$$

The relations g and h need not be as trivial as these, and better results may be extracted from irregular bins as we shall describe later.

4.2. Filling the Histogram

Using this definition, the histogram is filled using motion information from frame differencing provided by the camera. For each foreground (motion) pixel we increment all bins that contain it. That is, given an above-threshold pixel x, we increment all bins in the set $h(x)$. The end result is that each histogram bin is assigned a value corresponding to the total number of foreground pixels it encompasses:

$$b = |\{x : x \in g(b) \wedge x > T\}|$$

where the vertical bars denote set cardinality, and T is the motion threshold. The location of a person on the image plane can then be computed by running a peak-finding algorithm on the histogram. Figure 2 illustrates the entire process.

The histogram filling so far produces new bin values for each new video frame without taking into consideration the histogram for the previous frame, producing noise-prone centroids. For increased robustness, a modified algorithm (Figure 4) takes care of this by incorporating the composition variable, α. Each new histogram is superimposed on the previous histograms, with transparency $0 < \alpha \leq 1$. Hence, the instance where the past histogram values are not considered is a special case of this, with $\alpha = 1$. Figure 3 shows a histogram produced by the superimposed-histogram algorithm.

```
FillHistogram

for each b ∈ histogram
    b ← b × (1 − α)

for each x ∈ image frame
    if x > T
        for each b ∈ h(x)
            b ← b + α
```

Fig. 4. Pseudocode for histogram computation algorithm.

4.3. Optimizing the Histogram: Wide-Angle Considerations

In the case where each bin maps to equal, but shifted areas in the image, the histogram can be seen as the result of the cross-correlation of the image with a human model. In the simplest case, this model is a square, as in our discussion so far. Another possibility is to utilize a more complex function as a kernel, such as a multivariate Gaussian distribution. For the other types of models considered later in this paper, the histogram-producing operation may no longer be a cross-correlation, in that the kernel shape may vary with its position.

The type of model utilized has an immense effect on the efficacy of the histogram. This is an extension of the effects that are seen in a cross-correlation: the breadth and height

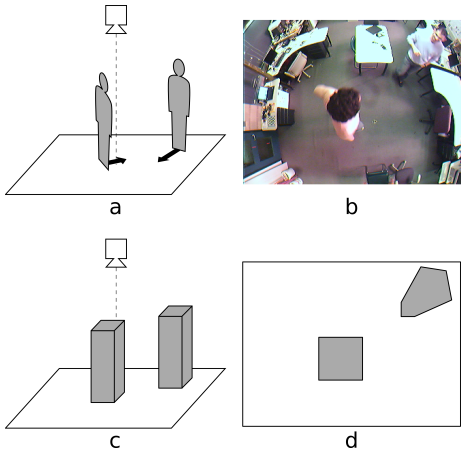


Fig. 5. Effect of perspective and lens distortion on histogram: (a) ground-truth positions; (b) image from top-view camera; (c) 3D bin model in the two locations that best match the ground-truth position; (d) bins projected using camera and deployment parameters. These are the h mappings of the two peak bins of the motion histogram.

of the correlation peaks are the best when the mask perfectly matches the image. In the case of the motion histogram, if the model is too small, multiple histogram peaks may appear for each person. If, on the other hand, it is too large, then the chance that two people incorrectly produce only one peak increases. Similar considerations must be made when picking the window-shift step size, δ: if the bins are too close, multiple bins may enclose the same person; if too far, the person may be missed entirely. These parameters are initially picked to match the average human dimensions in the described setup, then fine-tuned empirically (Section 6.1).

There are two additional effects that have not yet been accounted for, but which must be considered when building the histogram: perspective and lens distortion. Their effect is especially accentuated for wide-angle lenses and situations where the object distance is fairly small compared to its size. Since a person is relatively large compared to the typical ceiling height (Figure 5a), this must be taken into account for our setup. The top camera in the figure (Figure 5b) produces very distinct images for each of the people depending on their distance from the center axis of the camera: people near the center of the image appear as seen from the top, while those at the edges are seen diagonally from the side. Hence, the square histogram bins yield good results for subjects near the center of the image, where there is an approximate top-view, but not so much as people wander toward the image edges.

Accounting for this, a human model is derived from a 3-dimensional object that is then projected back into the image plane using the camera's intrinsic calibration parameters. We take a rectangular cuboid as the 3D model (Figure 5c), with width and height taken from average human measurements.

The model's image is calculated by applying geometric optics equations in conjunction with the Brown-Conrady distortion equations [15] to the coordinates of each of the cuboid's corners. The resulting bins provide a more accurate model as can be seen in Figure 5d. The motion histogram can, then, be constructed as follows: for each bin b in the histogram, the 3D model is shifted by an amount δ and the relation $g(b)$ is mapped to the set of pixels that are inside the projected image of the cuboid.

5. CAMERA-NODE LEVEL COUNTING

The goal of the algorithm described here is not to uniquely identify each person. Instead we reduce the tracking problem to that of uniquely identifying people within small periods of time, until an ambiguous situation occurs. After that, the algorithm should reassign IDs to each person involved in the ambiguity.

We utilize for this purpose an algorithm similar to [3] due to its speed. The major difference is that it is adapted for use with the histogram. At each time instant t we wish to find the list of detected people $P_t = \{\mathbf{p}_{ti}\}_{i=0}^{n_t}$, based on observed histogram peaks $Q_t = \{\mathbf{q}_{tj}\}_{j=0}^{m_t}$ and predicted locations $\hat{P}_t = \{\hat{\mathbf{p}}_{ti}\}_{i=0}^{n_t}$. The variable $n_t = |P_t|$ is, thus, the number of detected people at time t, while $m_t = |Q_t|$ is the number of peaks in that instant. As a *nota bene* on notation, for the remainder of the paper the ith element of a list (A) is denoted with the same letter as the list, but in lower case, and its index i appended to the subscript (a_i).

Each detected (and predicted) person is represented by a feature vector $\langle \mathbf{s}_{ti}, h_{ti}, b_{ti}, \mathbf{v}_{ti}, \mathbf{a}_{ti} \rangle$, denoting the values for the peak position, peak height, peak breadth, velocity and acceleration, respectively. Note that the height corresponds to the value of the histogram bin — *not* the height of the detected person. Position, height and breadth are acquired directly from a single histogram, while velocity and acceleration are be computed by following the changes in position over time. Note that $\mathbf{s}_{ti}$, $\mathbf{v}_{ti}$ and $\mathbf{a}_{ti}$ are 2-dimensional vectors, with x- and y-axis components. Meanwhile, peaks are represented as a vector containing only the features that can be extracted from a single histogram: $\mathbf{q_{tj}} = \langle \mathbf{s}_{ti}, h_{ti}, b_{ti}, 0, 0 \rangle$. The notation we use to refer to a component of a vector is as such: the height component of p_{ti} is simply h_{ti}.

For each person in $\hat{P}_t$, the algorithm tries to find the matching peak in Q_t by using a distance measure. For this, we define the distance d_{ij} between a predicted person $\hat{\mathbf{p}}_{ti}$ and a peak $\mathbf{q}_{tj}$ as the following weighted sum:

$$d_{ij} = \beta_s \|\hat{\mathbf{s}}_{ti} - \mathbf{s}_{tj}\| + \beta_h |\hat{h}_{ti} - h_{tj}| + \beta_b |\hat{b}_{ti} - b_{tj}|$$

where each weight β_x is a scalar weight chosen such that the distance between the most different peaks is 1. With this definition, the best match is the one with distance closest to 0. A distance matrix D is computed, recording the distance between each prediction and each peak.

People's positions are then predicted by inputting the current position, velocity and acceleration into the following kinematics equation: $\mathbf{s}_{t+1} = \mathbf{s}_t + \mathbf{v}_t t + (\mathbf{a}t^2)/2$. This assumes that between two consecutive time instants (frames) the acceleration is constant. This method was chosen due to it being the simplest that suits our goals of low-level tracking. Otherwise, a Kalman filter could have been used without any change in the rest of the algorithm. We predict only the position component of the feature vector because our experiments have shown that the height and breadth change slowly enough that the previous value is generally a good estimate.

The algorithm updates its belief P_t by finding for each prediction $\hat{\mathbf{p}}_{ti}$ the peak $\mathbf{q}_{tj}$ that minimizes the distance function:

$$\mathbf{p}_{ti} = \arg \min_{\mathbf{q}_{tj} \in \hat{Q}_t} \text{dist}(\hat{\mathbf{p}}_{ti}, \mathbf{q}_{tj})$$

At this stage, only *perfect matches* are considered. That is, the prediction $\hat{p}_{ti}$ is matched to the observed peak q_{tj} if and only if q_{tj} is the peak that is closest to $\hat{p}_{ti}$ and $\hat{p}_{ti}$ is the prediction that is closest to q_{tj}. This can be found by looking at the distance matrix D to see whether d_{ij} the minimum value of row i and column j at the same time. All peaks and predictions that are not matched at this stage are further investigated in the stages below.

False Negatives and Person Leaving — Given that the histograms are built from frame differencing data, the peaks tend to shorten and disappear when a person stops moving, causing a false negative. This does not mean that the person is outside the field-of-view, so the person should continue being accounted for. In order to resolve this scenario, at each new frame we construct the list $\hat{S}_t$ of predicted stops, containing the people whose motion is under a certain threshold. This can be measured by watching the height component of the histogram peak. In the next frame, any person that has not been matched but who can be found in $\hat{S}_t$ is believed to have stayed at the same location. Those who aren't present in $\hat{S}_t$ are discarded, as they either left the field-of-view or were an uncaught false positive. Additionally, peaks at the histogram edges were allowed to disappear even if their height was small.

False Positives and Person Entering — Although the superimposed-histogram algorithm does away with most of the noise, false positives may still arise in a few scenarios. For this purpose, peaks that thus far have not been matched by the algorithm are grouped into a list of person candidates C_{t+1} for the next time instant. In effect, the list contains all the new peaks that cannot be accounted for. These peaks will be considered in the next time instant alongside the person predictions $\hat{P}_{t+1}$, and only if they match a peak in the new frame will they be included in the detected list P_{t+1}. This does away with transient peaks while still counting the new peaks which are more stable, allowing new people to enter the covered area.

Merging and Splitting — It is common for image seg-

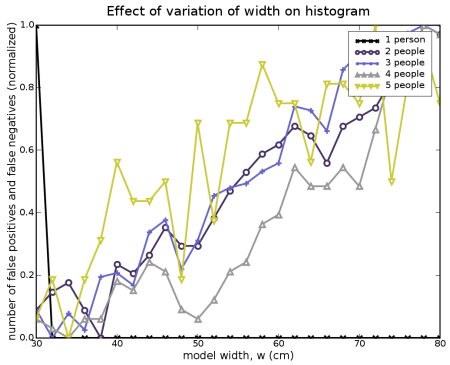


Fig. 6. Effect of varying the bin width w in the 3-dimensional bin model. The value of δ was kept at $15cm$. The y-axis shows the number of detection errors, normalized for easier comparison between experiments with different numbers of people.

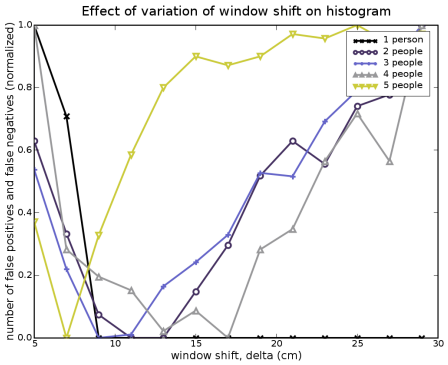


Fig. 7. Effect of varying the model shift δ in the 3-dimensional bin model. The width was kept at $50cm$. The y-axis shows the number of detection errors, normalized for easier comparison between experiments with different numbers of people.

mentation and blobbing algorithms to count two or more foreground objects as a single one. The same is true with the histogram algorithm described here. The counting algorithm handles these cases by predicting them ahead of time: a merge-candidates list $\hat{M}_{t+1}$ is generated containing all predicted positions $\hat{\mathbf{p}}_{t+1,i}$ that are close enough to cause a merge. The counting algorithm described above is suitably altered to allow merge candidates to match to a previously-matched peak. Hence, each merged person gets counted as being in the location of the merged peak, causing their position to become less accurate at that moment. While this may be problematic for tracking, it is not so for the purpose of counting.

6. EXPERIMENTAL RESULTS

6.1. Histogram Parameters

To empirically find the best w and δ sizes, we run the histogram algorithm on a single camera in a room with a known number of people walking inside. Then we vary the number of people and re-run the experiment. We increment a counter each time the number of peaks in a histogram does not match the number of people in the room. The best histogram structure is chosen as the one that provides the least amount of errors (lowest counter) for most situations. Figure 6 shows the effect of varying w for a fixed $\delta = 15cm$. Meanwhile, Figure 7 is the result of varying δ while holding w at $50cm$. Both plots were generated using the 3-dimensional bin model, with a height of $170cm$. Due to space limitations, the plots for different numbers of people have been combined, and the number of errors (y axis) was therefore normalized. There is no plot for when the room is empty because there were no false positives in that situation, so the plot is all zeros.

From the first plot (Figure 6) one finds that the number of errors plateaus around the range $w = 30cm$ to $50cm$, then sharply rises. Meanwhile, Figure 7 shows a much different shape, with a dip in the number of errors for $\delta \in [8, 15]$. For this paper we choose $w = 50cm$ and $\delta = 15cm$ due to being the largest values in these w and δ ranges. This increases the histogram's robustness to false positives: using these values, our tests with people sitting and lying down show no false positives, even though the 3-dimensional bin model is admittedly more suited for standing position.

The room where these experiments were performed has dimensions $9m \times 5m$, with a ceiling height of $3m$. The entire floor was covered by a single camera node with wide-angle lens mounted on the ceiling. On the other hand, if the *usable* field-of-view is defined as the one where a person is seen in their entirety, then the dimensions get reduced to around $3.2m \times 2.4m$. The people in the room were asked to stay within those bounds, but, in the experiments with 5 people, they often moved outside due to space constraints. Sample videos of or experiments can be found on the website cited in Section 1. Note that, since the motion histogram boils down to a type of correlation, due to space constraints we refrain from characterizing the precision of the locations given by the histogram, as correlations are common textbook knowledge.

6.2. Prototype Network

We have implemented the motion histogram and counting algorithm in a sensor network composed of multiple Intel iMote2 sensor nodes. The nodes are suited with a custom-built camera-board (Figure 8) that contains the OmniVision OV7649 imager. The nodes acquire images at 320×240 resolution, downsample them to 80×60, then run the algorithm described in Figure 4. The mappings g and h are precomputed and kept in the node's memory for fast operation. Each peak

Fig. 8. Our custom camera board with wide-angle lenses mounted onto the iMote2 sensor node.

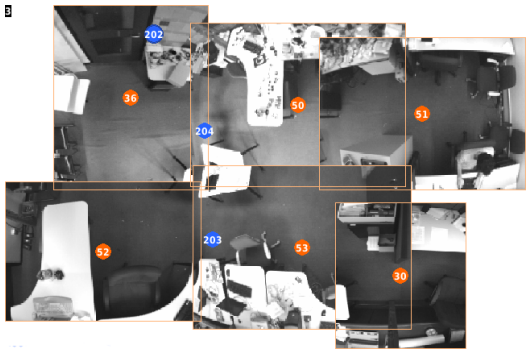


Fig. 9. Composite of images from all 6 nodes, with 3 people in the scene (blue circles). The numbers on the circles are each person's temporary ID. The correct count is shown in the top-left corner.

is recorded along with time of detection, and sent through the radio once the send buffer is full. This entire process repeats approximately every $110ms$, allowing a frame rate of just over 8 frames per second. The packets are processed by the base node, which reports the person-count to the gateway computer, along with the peak data for visualization purposes.

The nodes are placed on the testbed structure on the ceiling of our lab, where they are a single hop away from their base. Given the lower ceiling height at the lab ($240cm$) and the presence of cubicle walls, 6 nodes are required to cover the entire area. In this configuration, each node has an *active* field-of-view of approximately $3m \times 2m$. The node positions are chosen to minimize field-of-view overlaps, and the images they acquire are cropped until the overlap is virtually zero. This way, we attempt avoid most of the correspondence issues to focus on the histogram performance. However, since the node time-synchronization protocol we utilize gives us a measured discrepancy of $187ms$, and given that the nodes acquire pictures in an unsynchronized manner, the peak detection timestamp has a significant margin of error. This uncertainty in the timing between nodes makes it possible for physical inconsistencies to occur, such as a person apparently being in two places at the same time. That is, the correspondence problem reappears in a different form. The details of timing have not been entirely worked out, and are a possible direction for future work. Our preliminary results show good accuracy regardless.

We tested the histogram positional accuracy by having two people walk toward one-another and meet at the center of the image. This was captured by a single node. The histogram was able to differentiate distances of up to $15cm$ 100% of the time. This is the maximum possible resolution, given that $\delta = 15cm$. This resolution greatly suits the assisted-living scenario, where the main interest is in the *logical spacial location* (such as "on the sofa", or "by the stove"), instead of exact coordinates. The same test was performed for locations increasingly farther from the center. The result was the same for distances up to $1m$ from the image center. At that distance, although the histogram at times produced a single peak for both people, the tracking/counting algorithm was able to disambiguate them. At the farthest position where one fully covered by the camera ($1.5m$), the algorithm missed around 42% of all detections. We believe there is room for improvement in those conditions, by utilizing a better tracker. The histogram achieves its best precision at the center two-thirds of the image. This is clear when people walk closely and side-by-side near the edges, which causes an occlusion to occur. Near the center, the maximal accuracy ($15cm$) was achieved on all runs of the parallel-walking tests. Additionally, on the runs where the two people crossed paths, the tracking algorithm was able to keep the correct count and locations regardless of distance from the image center.

For the next experiment, people walked around the testbed through every node's field-of-views, at times standing still for a few seconds, and other times changing the position of office chairs. During the experimental runs where a single person was present, the network correctly counted the number of people 89.5% of the time. The majority of the errors were false positives that occurred at field-of-view overlaps, due to the timing issues already discussed. For two people, that number drops to 82.48%, and 79.8% for three. This small decrease in accuracy follows an increase in packet drop rates, which reached 6.6% when all three people were in the lab. Packet loss is expected to increase with the number of people, given that all nodes will be attemting to transmit data simultaneously. We are currently working on a more apt messaging and routing scheme. None of the errors were caused by the change in furniture placement. A close examination of the data has shown that the sensing at each node functioned correctly. The discrepancies were due to the time synchronization that resulted in double counting.

7. CONCLUSION

We have developed a lightweight, online people-counter utilizing a novel, AE-friendly motion-histogram. The histogram is robust to pixel intensity fluctuations, gradual lighting changes and furniture repositioning. Abrupt alterations in lighting may,

at times, cause false positives, but they vanish within a few frames. The algorithm described in this paper was implemented on a prototype counting network with multiple cameras nodes. While in the center area of the image the histogram has proven to be very accurate, near the edges it is subject to occlusions, due to the presence of perspective effects. This may be resolved by overlapping the edge areas of multiple calibrated cameras similarly to [11]. More importantly, the motion histogram need not be utilized as the sole visual sensing modality. It is also possible that the histogram may be improved by making use of other visual features in addition to motion. Candidate features for this are color and percentage of straight edges, as in [16], for possibly more accurate human-detection. Our plan for the immediate future is to deploy this algorithm in our assisted living deployment were 7 camera sensor nodes cover a 1,100 square foot, 2-bedroom apartment.

8. ACKNOWLEDGMENTS

This material is based upon work supported in part by the National Science Foundation under Grant No. 0622133. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

9. REFERENCES

[1] D. Lymberopoulos, A. Ogale, A. Savvides, and Y. Aloimonos, "A sensory grammar for inferring behaviors in sensor networks," in *Proceedings of Information Processing in Sensor Networks, IPSN*, April 2006.

[2] T. Teixeira, E. Culurciello, E. Park, D. Lymberopoulos, and A. Savvides, "Address-event imagers for sensor networks: Evaluation and modeling," in *Proceedings of Information Processing in Sensor Networks, IPSN*, April 2006.

[3] J. Owens, A. Hunter, and E. Fletcher, "A fast model-free morphology-based object tracking algorithm," in *Proceedings of the British Machine Vision Conference*, 2002.

[4] C. J. Veenman, M. Reinders, and E. Backer, "Resolving motion correspondence for densely moving points," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, March.

[5] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easyliving," in *VS '00: Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS'2000)*, Washington, DC, USA, 2000, IEEE Computer Society.

[6] A. Utsumi, H. Mori, J. Ohya, and M. Yachida, "Multiple-human tracking using multiple cameras," in *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, Washington, DC, USA, 1998, p. 498, IEEE Computer Society.

[7] C. Stauffer and E. L. Grimson, "Learning patterns of activity using real-time tracking," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, August 2000.

[8] C. Ridder, O. Munkelt, and H. Kirchner, "Adaptive background estimation and foreground detection using kalman filtering," 1995.

[9] K. Hashimoto, K. Morinaka, N. Yoshiike, C. Kawaguchi, and S. Matsueda, "People count system using multi-sensing application," in *IEEE International Conference on Solid-State Sensors and Actuators*, June 1997.

[10] Gary Conrad and Richard Johnsonbaugh, "A real-time people counter," in *SAC '94: Proceedings of the 1994 ACM symposium on Applied computing*, New York, NY, USA, 1994, pp. 20–24, ACM Press.

[11] D. Yang, H. Gonzalez-Banos, and L. Guibas, "Counting people in crowds with a real-time network of simple image sensors," in *Proceedings of Ninth IEEE International Conference on Computer Vision, 2003.*, October.

[12] A. J. Schofield, T. J. Stonham, and P. A. Mehta, "A ram based neural network approach to people counting," in *IEEE Conference on Image Processing and its Applications*, July 1995.

[13] T. Murakita, T. Ikeda, and H. Ishiguro, "Human tracking using floor sensors based on the markov chain monte carlo method," in *Proceedings of the 17th International Conference on Pattern Recognition*, August 2004.

[14] D. Lymberopoulos, A. Barton-Sweeney, and Andreas Savvides, "Sensor localization and camera calibration using low resolution cameras," Tech. Rep. ENALAB 080501, Yale University, 2005.

[15] D.C. Brown, "Decentering distortion of lenses," in *American Society of Photogrammetry, Annual Convention*, March 1965, vol. 32, pp. 444–462.

[16] Ali Maleki Tabar, Arezou Keshavarz, and Hamid Aghajan, "Smart home care network using sensor fusion and distributed vision-based reasoning," in *VSSN '06: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, New York, NY, USA, 2006, pp. 145–154, ACM Press.