Entity Counting and Tracking Using Microsoft Kinect Depth Maps

Jared Meade, Sheree Enlow, John Chen, Eric Chen, Aaron Crandall

School of Electrical Engineering and Computer Science, Washington State University

Introduction

The Palouse Discovery Science Center (PDSC) in Pullman has requested a head count system to keep track of the number of patrons visiting them as well as basic information such as their heights to distinguish parent/child distributions. Gathering accurate headcount data through traditional methods can be intrusive and not ideal for a family friendly facility. Our multi-part solution utilized an unsupervised learning method and entity tracking algorithms coupled with depth maps from a inconspicuous, ceiling mounted Microsoft Kinect V1 above an entry way combined with the CASAS Smart Home in a Box (SHiB) sensor system.

Hardware/Software

The two parts of the Headcount system are the Microsoft Kinect V1 infrared camera and the open source driver that allowed us to pull depth map frames in C++ on a Linux machine called libfreenect. In addition to the headcount system, CASAS SHiB sensors as well as data visualization software PyViz and Graphana are also utilized.

a. Depth image b. RGB image

c. CASAS SHiB Sensors

d. Microsoft Kinect V1

Data Visualization

Graphana is data visualization software we used in conjunction with our databases of information. We are able to interpret the data in meaningful ways such as displaying the number of adults versus children in the building, the headcount over time, or more whimsical ones like the total height of everyone in the building.

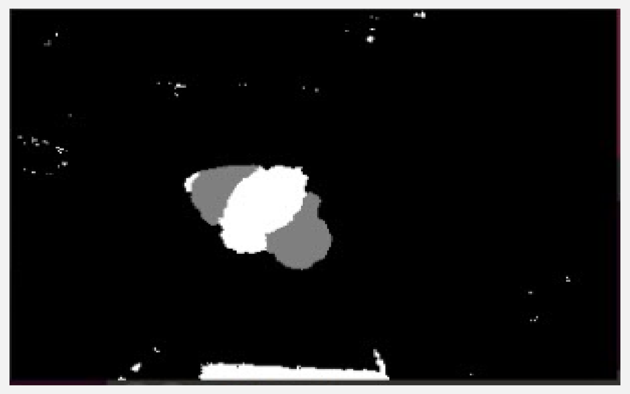
e. Graphana Dashboards displaying information

Methods and Algorithms

Head Count/WaterFill Algorithm[1]:

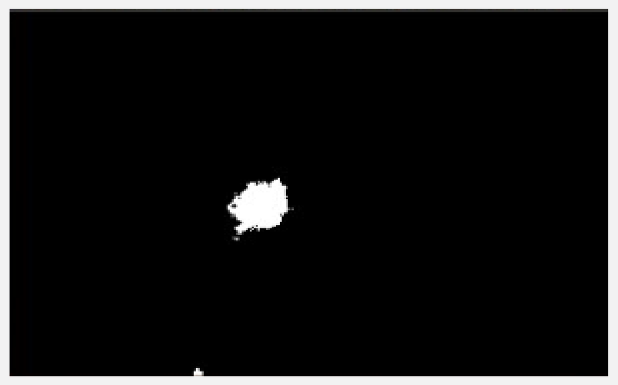
The WaterFill algorithm works by identifying people/heads as the foreground of the image by using a Gaussian mixture model background subtraction method. The statistical nature of this subtractor allows for the WaterFilll algorithm to adapt to a dynamically changing background.

e. Depth image greyscale

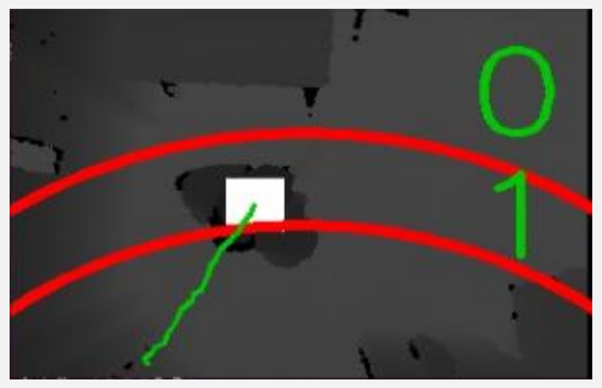


f. Foreground/subtracted background

After filtering out objects in the foreground image that have contours (the boxed outline over the person's head) with area's smaller then some threshold value, the initial depth map's pixels are all turned to one except for the pixels corresponding to white regions on the filtered foreground region. "Water drops", where the algorithm gets its name, are dropped, randomly distributed over the filtered depth map in all regions that aren't equal to one and travel in the direction of descent. All of these drops are kept track of in a separate blank image representing areas where "water" has been collected (f). We then filter out the noise and enhance the water image is then by eliminating regions that did not reach the threshold of depth. The contour boxes of the remaining water filled regions now correspond to heads and can be seen drawn on image (d) in the final image (g).

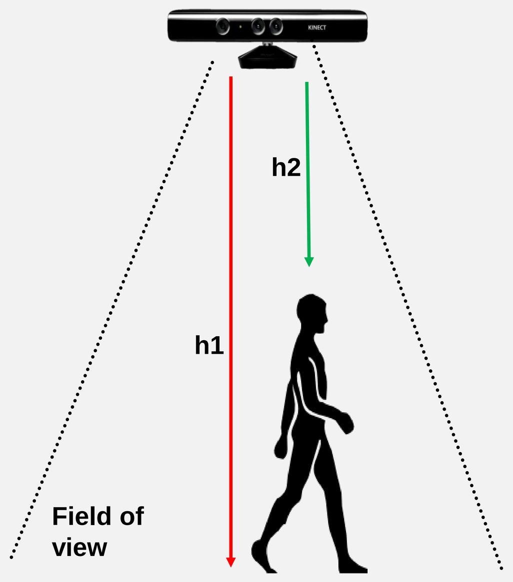


g. Greyscale representation of water filled regions



h. Contour box/head

Determining Head Height



h1-h2 = height. For a more robust solution we created a method of determining the floor distance from a sample of initial images. The first 30 depth images (1 sec worth) are taken and a 5x6 portion of the images centered in the middle is assumed to be the floor and each value is stored. Next, each pixel in 30 more images is iterated through and if the pixel is within one standard deviation from the mean of our stored floor values it is added to the list of floor values. After all 60 initialization images have been processed we find the mean of our floor list and use this value as the constant floor to Kinect distance.

Results

Test Dataset	In Count	Out Count	Total
Head Tracking Algorithm	64	55	119
Actual Count	66	63	129
Accuracy	96.97%	87.30%	92.25%

Table 1: Accuracy of Head Tracking Algorithm

Entity	1	2	3	4	5	6	7	Average
Actual Height (ft)	3.88	4.60	5.02	5.54	5.33	5.79	6.06	5.18
Measured Height (ft)	4.41	4.58	4.98	5.43	5.63	5.72	5.93	5.24
Difference	0.53	0.03	0.04	0.11	0.30	0.08	0.14	0.17

Table 2: Actual vs. Measured Height of Entities

(Test dataset is not representative of an average dataset/day at PDSC and was designed to test the robustness of our system.)

Conclusion

Our Kinect head tracking system struggled in the following edge cases and can be improved upon in these aspects:

- Small children/heads (parameters can be tuned)
- When a person was carried on another's shoulder/back
- When arms were raised above one's head
- Person walked out of left or right edge of view before crossing line

The CASAS SHiB provides reliable activity data but is most effective when used with the Kinect head tracking system.

Moving Forward

- We can determine more meaningful information and trends from the data captured and visualize them with additional Graphana dashboards or visualization methods
- Send weekly/monthly comprehensive reports to the PDSC or CASAS
- Apply our system to other buildings or venues to gather and interpret different types data
- Outlier detection may be used in tandem with existing security systems to help identify intrusions or people in the building at closing time

References

[1] Zhang, X., Yan, J., Feng, S., Lei, Z., Yi, D., and Li1, S. Z., 2012, Water Filling: Unsupervised People Counting via Vertical Kinect Sensor, *IEEE*.

Acknowledgments

We would like to thank the National Institutes of Health & the National Institute of Aging and their generous support of the GSUR program under grant 1R25AG046114.