

Demo: FORK: Fine grained Occupancy estimator using Kinect on ARM Embedded Platforms

Sirajum Munir, Le Tran,
Jonathan Francis, Charles
Shelton
Bosch Research and Technology
Center, Pittsburgh, PA
{sirajum.munir,le.tran,jon.francis,
charles.shelton}@us.bosch.com

Ripudaman Singh Arora
University of Michigan,
Ann Arbor, MI
asripu@umich.edu

Craig Hesling, Matias
Quintana, Anand Krishnan
Prakash, Anthony Rowe,
Mario Berges
Carnegie Mellon University,
Pittsburgh, PA
{jhesling,mquintan,anandkrp,agr,
mberges}@andrew.cmu.edu

ABSTRACT

Occupancy estimation is very useful for a wide range of smart building applications including energy efficiency, safety, and security. In this demonstration, we present a novel solution called FORK, which uses a Kinect depth sensor for estimating occupancy in real-time. Unlike other camera-based solutions, FORK is much less privacy invasive (even if the sensor is compromised) and it does not require a powerful machine like a Microsoft XBOX or an Intel[®] Core[™] i7 processor to process the depth data. Our system performs the entire depth data processing on a cheaper and lower-power ARM processor, in real-time. In order to do that, FORK uses a novel lightweight human model by leveraging anthropometric properties of human bodies for detecting individuals. We will show how FORK detects, tracks, and counts occupants accurately in real-time.

CCS CONCEPTS

• **Computer systems organization** → *Embedded and cyber-physical systems; Embedded systems; Real-time system architecture*; • **Hardware** → *Sensor applications and deployments*;

KEYWORDS

Occupancy Estimation, People Counting, Depth Sensing

ACM Reference format:

Sirajum Munir, Le Tran, Jonathan Francis, Charles Shelton, Ripudaman Singh Arora, and Craig Hesling, Matias Quintana, Anand Krishnan Prakash, Anthony Rowe, Mario Berges. 2017. Demo: FORK: Fine grained Occupancy estimator using Kinect on ARM Embedded Platforms. In *Proceedings of BuildSys '17, Delft, Netherlands, November 8–9, 2017*, 2 pages. <https://doi.org/10.1145/3137133.3141461>

1 INTRODUCTION

Occupancy estimation is very useful for improving energy efficiency, safety, and security of buildings. Heating, ventilation, and air conditioning (HVAC) consumes a significant amount of energy in the US. In 2006, approximately 35% of the total energy in the

US was used for HVAC [4]. Most HVAC systems operate by assuming maximum occupancy in each room, which wastes a significant amount of energy. Feeding actual occupancy count to HVAC systems can help in reducing such energy waste. Lighting control systems that rely on PIR-based motion sensors often turn off lights when people are inside but not moving, and they keep the lights on for a timeout interval even after everyone leaves a room, thus wasting energy. Both issues can be addressed if the number of occupants can be estimated accurately. In emergency situations, e.g., a building fire, knowing the number of occupants that are inside is very useful for rescue operations.

There are several proposed occupancy-estimation solutions that use break-beam sensors, ultrasonic sensors, cameras, and thermal imagers. Break-beam sensor solutions do not work when multiple people enter/exit simultaneously through a door. Ultrasonic solutions [8] require significant training. RGB cameras are too privacy-invasive to be deployed in many sensitive settings, e.g., offices, and they do not work in the dark. Low resolution thermal imagers (e.g., an 8x8-pixel Panasonic GridEYE) do not have good enough resolution to detect multiple people entering/exiting simultaneously [6] and high resolution thermal imagers are relatively expensive, e.g., a 32x32 Heimann thermopile array sensor costs over \$200. A depth sensing based solution can overcome these limitations to a great extent: our system detects multiple people entering/leaving a room simultaneously, requires almost no training, is not as privacy-invasive as RGB cameras, and works in dark environments.

2 PROPOSED SYSTEM

Our proposed system uses a time-of-flight sensor, more specifically a Kinect depth sensor for occupancy estimation; we call it FORK (Fine grained Occupancy estimator using Kinect). Our solution is different from other depth sensing based solutions in two ways. First, we place the Kinect at the ceiling (Figure 2(a)) looking downwards near to a door instead of placing it in front of people. Second, instead of using a powerful machine like an XBOX, we perform the entire processing on a cheaper and lower-power ARM processor.

2.1 Hardware

We use the depth sensor in Kinect for XBOX One (Figure 1(a)) in this work. We choose an Odroid-XU4 (Figure 1(b)) for processing, as it is one of the few embedded platforms that supports USB 3.0 that is required by this Kinect. FORK does not use the Kinect SDK. So, any depth sensor with similar hardware functionality will work.

Permission to make digital or hard copies of part or all 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. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

BuildSys '17, November 8–9, 2017, Delft, Netherlands

© 2017 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5544-5/17/11.

<https://doi.org/10.1145/3137133.3141461>



Figure 1: (a) Kinect sensor for XBOX One. (b) Odroid-XU4.

2.2 Software

FORK is implemented in C++, it uses OpenCV library, and it runs on Ubuntu 15.04 as an application on Odroid XU4. To access the depth frames from a Kinect, FORK uses the *libfreenect2* library of the OpenKinect project. The resolution of a depth frame is 512x424. Each pixel of the depth frame provides the distance from the Kinect to the nearest object, in millimeters. FORK performs occupancy estimation in several steps.

- (1) During preprocessing, it resets all noise pixels (reported as NULL from the driver) and outliers (depth too high) to the floor's depth.
- (2) FORK detects humans in three steps: multilevel scanning, head verification, and shoulder verification:
 - (a) Multilevel Scanning: The average height of an adult male is about 5'3" to 6'5" and a female is about 4'8" to 5'7.5" [3]. Relative to the floor, we scan depth data between 6' and 2' at every 6-inch intervals and find all contours at each level. For each contour, we find the minimum enclosing circle. The center (c_x, c_y) and radius (r) of that circle is considered the center and radius of the head.
 - (b) Head Verification: We verify the presence of a human head at (c_x, c_y) location. We use a hemi-ellipsoid (top half of an ellipsoid) to model a human head, which is represented by equation (1), where a , b , and c are the lengths of the semi axes and (c_x, c_y, c_z) is the center of the ellipsoid.

$$\frac{(x - c_x)^2}{a^2} + \frac{(y - c_y)^2}{b^2} + \frac{(z - c_z)^2}{c^2} = 1 \quad (1)$$
 We set $c_x = c_x$, $c_y = c_y$, and $c_z = T + 0.5 * D$, where T is smallest distance between the Kinect and the head, and D is the depth of a human head. Based on the average length of a human head [1, 2], we set $D = 220$ mm. We iterate over the x, y values of the detected contours and use equation (1) to compute a z value for each (x, y) and compare it with the corresponding z value in the depth frame. If the average difference is less than a threshold, T_{head} (which is set to 40), we report that the head is verified.
 - (c) Shoulder Verification: The end-to-end distance between the two shoulders of a person is around three times their head's diameter [1]. Hence, we choose a slightly bigger square ROI around the head. We compute a histogram of depth data at different height levels and check if there is at least one bin at the shoulder depth level that has enough depth data points to represent a shoulder.
- (3) In order to track individuals, FORK uses a lightweight greedy bipartite matching algorithm by leveraging the position, height, and head radius of people.

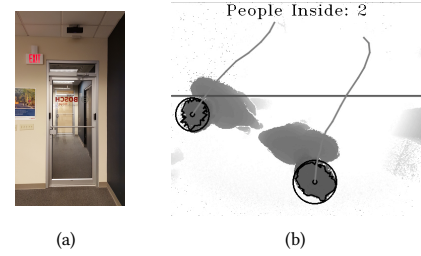


Figure 2: (a) Kinect mounted near a door way. (b) Sample depth data, where FORK is tracking individuals.

- (4) FORK tracks individuals and increases/decreases occupancy count when someone crosses door line segments.

The detailed technical description of the algorithm is available in [7]. Some findings of our nine months of occupancy data collection using FORK are described in [5].

3 DEMO DESCRIPTION

In our demonstration, we plan to mount a Kinect using a horizontal extender on a tripod. We will ask conference attendees to walk underneath the Kinect. We will run FORK on an Odroid XU4 and we will show the following activities in a monitor attached to the Odroid XU4 in real-time:

- FORK detecting and tracking head of the participant.
- FORK tracking multiple people walking simultaneously.
- FORK increasing/decreasing occupancy count when someone crosses a virtual geo-fence line (representing a door).
- Dynamic frame rate of FORK. FORK processes at around 9 FPS (Frames Per Second) when there is no one in the FOV. When there is someone, the frame rate drops to 6-7 FPS.

A sample depth image is shown in Figure 2(b). The circles around the heads show FORK detecting the precise locations of the heads. The trailing lines behind the two heads show the tracking lines of the walking path of the two subjects. The middle horizontal line represents the door. The occupancy count is shown at the top.

ACKNOWLEDGMENTS

This work was supported, in part, by DOE grant DE-EE0007682. The opinions expressed here are those of the authors and do not necessarily reflect the views of the DOE.

REFERENCES

- [1] 2017. Human figure average measurements. <http://www.fas.harvard.edu/loebinfo/loebinfo/Proportions/humanfigure.html>. (2017).
- [2] 2017. Human head. https://en.wikipedia.org/wiki/Human_head. (2017).
- [3] 2017. Human height. https://en.wikipedia.org/wiki/List_of_average_human_height_worldwide. (2017).
- [4] 2017. EIA. <http://www.eia.doe.gov/>. (2017).
- [5] K. S. Liu, S. Munir, J. Francis, C. Shelton, and S. Lin. 2017. Poster Abstract: Long term occupancy estimation in a commercial space: an empirical study. In *IPSN*.
- [6] H. Mohammadmoradi, S. Munir, O. Gnawali, and C. Shelton. 2017. Measuring People-Flow Through Doorways using Easy-to-Install IR Array Sensors. In *DCOSS*.
- [7] S. Munir, R. S. Arora, C. Hesling, J. Li, J. Francis, C. Shelton, C. Martin, A. Rowe, and M. Berges. 2017. Real-Time Fine Grained Occupancy Estimation using Depth Sensors on ARM Embedded Platforms. In *RTAS*.
- [8] O. Shih and A. Rowe. 2015. Occupancy Estimation Using Ultrasonic Chirps. In *ICCPs*.