Creating a Room Connectivity Graph of a Building from Per-Room Sensor Units

Carl Ellis
Microsoft Research, UK and
Lancaster University, UK
carl.ellis@comp.lancs.ac.uk

James Scott
Microsoft Research, UK
jws@microsoft.com

Ionut Constandache
Duke University, US
ionut@cs.duke.edu

Mike Hazas
Lancaster University, UK
hazas@comp.lancs.ac.uk

Abstract

Sensor and actuator networks are often installed in buildings for energy-related applications such as lighting and climate control. Such systems require metadata about the deployed hardware (e.g. which room each is in, what the function of each room is) in order to operate effectively. In this paper we present methods to automatically determine such metadata, in particular the room connectivity graph (i.e., which rooms share a doorway/interior window). Crucially, our method works with just one sensor unit per room, does not require special placement of any of the sensors, and can therefore work on data from existing widely-deployed applications (such as burglar alarms). We apply this method to a 30-day data set from single per-room sensor units deployed in two residential homes in the United Kingdom. Room connectivity is determined based on: spillover of artificial light between rooms; occupancy detections due to movement between rooms; and a fusion of the two. The fusion of both techniques is shown to work better than either technique alone, with a 93% true positive rate and 0.5% false positive rate (aggregate across both houses), and a convergence time of under a week.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Experimentation, Management

Keywords

Sensors, Algorithms, Auto-Calibration, Motion, Light

Buildsys'12, November 6, 2012, Toronto, ON, Canada.

1 Introduction

Room-level sensing and actuation are increasingly being used to optimise building energy consumption. An example of this is a house which heats individual rooms, in advance of that room's occupancy. To deploy such systems, one must not only physically install embedded devices, but also determine metadata such as which room each device is in and the layout of those rooms in the home (e.g. a floorplan or a room connectivity graph).

Such metadata can be provided manually, but this will add time and complexity to the installation process, as has been previously articulated [1, p. 128]. Sketching a floorplan (for example via touchscreen or stylus) with deployed sensor IDs requires some expertise on the part of the user, in addition to the challenges of the sketching tool interface design, and correct interpretation of the sketched rooms, doorways and sensors. Providing even basic information about the number of rooms, floors, and their connections also suffers from ambiguities. Since installation should ideally be achievable by untrained home owners/occupants rather than professional installers, automatic methods of inferring this data without manual input are valuable. Furthermore, these methods may be more robust to changes within the house or to the devices installed within it — e.g. replacement of a failing device.

We propose a set of algorithms to calculate room connectivity (rooms which share a doorway or other opening such as an interior window) of a home, relying on only single sensor units deployed in each room, specifically using light and motion sensing. We chose these sensor types, and chose to use one per room, to reflect existing applications for smart homes such as "dusk-till-dawn" lighting and burglar alarm systems. Thus, the sensors may already be in place in the target environment, or their installation planned as part of the home automation effort. Our algorithms leverage the data gathered by these sensors to reduce the need for manual input of metadata, but without requiring additional hardware.

The room connectivity graph that these algorithms output can be useful for a number of applications. Here we give four examples: First, most research in predictive heating control has focused on temporal priors for occupancy prediction (i.e. a room's future occupancy is predicted using its past occupancy). Information from connectivity graph could be used to determine spatial priors (i.e. a room's future occupancy is predicted by the occupancy of neighbouring rooms). Second, lighting and media devices (e.g. radios, televisions) might be switched off when there is no occupant in the current or connected rooms. Third, smart burglar alarm systems could look for movement in rooms where no connected room was first occupied. This would give warning even when the owner is at home, but in another part of the building. Fourth, smart heating systems could use the connectivity graph to assist in training advanced heat flow models for more efficient HVAC control solutions.

The contributions of this paper are: algorithms for room connectivity inference based on *light data* (looking at light levels in neighboring rooms and noticing when room lighting is turned on and off) and *motion data* (looking at motion data and noticing when people walk from room to room); We evaluate these algorithms using data previously gathered using sensors deployed for the purposes of predictive heating control [2]. We show that a fusion of both techniques works better than either alone, with a 93% true positive rate and 0.5% false positive rate (aggregate across both houses).

2 Related Work

Many home infrastructure deployments have relied upon manual surveys to provide metadata about the home (room connections, sensor positions, orientations, window properties, etc) [3] [4] [5]. Manual surveys can add time and complexity to the installation process [1], may be error prone, and may fall out of date as sensors are repaired or moved.

Automatically creating maps of a building or surroundings has been investigated by the robotics community [6] but the methods used normally require specialized sensors or custom robots which are currently impractical to apply at scale. In the same vein, researchers have used the building's inhabitants to create blueprints whether by using wearable sensors [7] or centimetre-level location readings [8]. Furukawa et al. [9] used images from a stereo camera to create a blueprint and a 3D model of the interior of a building. However, these systems go far beyond what off-the-shelf sensors can provide, instead requiring specialist equipment or a high-performance location system.

In home heating control applications, the literature has focused mostly on a whole house occupancy metric [10] [11]. However, with research looking towards per-room control [2] it is becoming more important not to just know which room an occupant is in currently, but which rooms they are likely to go to *next*. These systems only need a simple building map, which shows how the rooms are linked by doors and passages.

We feel that once sensors have been installed in a building, calibration should be an automatic affair. Brumitt et al. [12] call for a geometric model to be created which would allow for sensors and other devices to be added to infrastructure in a "Plug and Play" manner.

Lu and Whitehouse [1] describe a method for automatically generating representative floorplans for a house; they show this works in three of their four house deployments. Their method clusters sensors into rooms and assigns connectivity based on the concurrent firings of window– and door jamb–placed sensors. The walls of these rooms are assigned doorways (using magnetometer readings) and windows (based on sunlight intensity readings over the course of the day). The possible floorplan topologies are minimised using heuristic filters, narrowing it down to a small number of possible candidates.

Lu and Whitehouse's method has the potential to expose aspects such as floorplan orientation with respect to magnetic north, and identification of which internal/external walls have doorways/windows. By contrast, our method exposes only *path-based connectivity* between rooms, i.e. those with connecting doorways or passages that people and light pass through. This data is nonetheless useful for applications (described in Section 1) such as predictive heating control and adaptive burglar alarm systems.

Compared to that of Lu and Whitehouse, our method exposes this path-based connectivity by operating on a **lower sensor density** (one per room, compared to three or more); utilising **simpler sensor nodes** with just one sensor of each type rather than dual sensors (PIR-PIR and PIR-light) with opposite-facing elements; and **does not require special placement** of any of these sensors. Thus, our method can work with data from existing sensors such as those used in burglar alarms or lighting control, in which one sensor is typically deployed in each room, and it does not have to be mounted in a specific place, e.g. on the door of the room.

3 Algorithms

We use data from sensors originally deployed to perform automatic per-room heat control based on predicted occupancy, with the aim of reducing energy consumption for heating [2]. This system required a single sensor node to be deployed in each room (Figure 1) which measured: light intensity, temperature and humidity at 5 second granularity, and motion data (through a passive infrared sensor) at 1 second granularity. This data was transmitted in real time to a PC in the home using an 802.15.4 network. The system was installed in two family homes in the United Kingdom, and was used to conduct real time control of the heating system in those homes. House A had 4 occupants and 15 rooms, while House B had 3 occupants and 13 rooms. For this study, we used 30 days of data from April 2011.

No special activity by the occupants was required by the heating study or was undertaken in order to facilitate the work presented in this paper.

While we have not evaluated different motion and light sensor placements in the two homes, neither was the sensor placement carefully optimized: the aims were decent motion sensor coverage for each room (as would be common in a burglar alarm installation), and practical and aesthetic appeal (a concern for any real home). Moreover, we would like to stress that the reported characteristics of this algorithm used motion and light data arising from the natural behaviour of occupants in two homes over the course of four weeks.

To get from light and motion data to room connectivity graphs, the overall operation of our algorithms (and the ordering of this section) is as follows.

First, we determine a "transition matrix" from motion and light data, e.g. we see how often the data suggests connec-

Figure 1: Deployed sensor node.

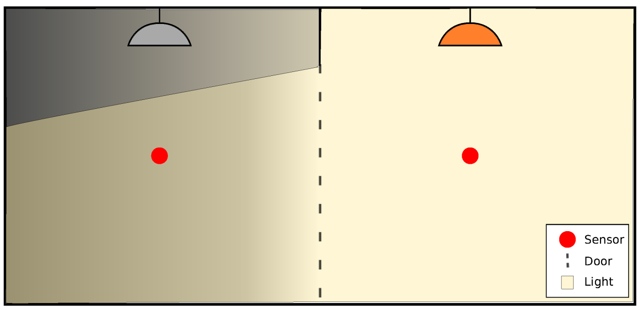


Figure 2: Light levels in the right-hand room affect the sensor readings of both rooms.

tivity between any two rooms. Then, we transform the transition matrix into a list of connections, using the same technique for both light and motion data. Finally, we discuss how we fuse light and motion data if both are available.

3.1 Determining a Transition Matrix with Light Intensity

We use the fact that lighting in a given room spills over into connected rooms in order to detect connectivity (Figure 2). Light level changes which happen at the same time are likely to be related, so observing instances where rooms incur simultaneous light level changes suggests that those rooms are connected. However, co-temporal events can occur through coincidence as well as through connectivity so we use an analysis of the frequency of such events rather than relying on single events. To eliminate the effect of daylight on the light sensors, we used known sunrise and sunset times (given the location of the houses) to restrict the data to night-times only.

When a light in a room is turned on or off there is a sudden change in light intensity which is large and easily identifiable. However, it also creates a smaller change in connected rooms, as shown in Figure 3.

To detect both large and small light level changes, we used a dynamic thresholding function based upon the current light levels and ambient fluctuations. The dynamic threshold is determined by using sensitivity analysis to sample the

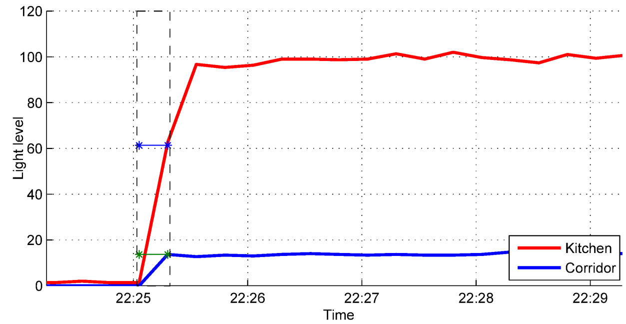


Figure 3: Light intensity values recorded in two connected rooms resulting in a spillover event detected. NB the algorithm copes with this low energy light bulb which grows brighter over 30s.

light data (into 1 minute bins) and measure how much the data varies in each sample. By sampling in this way, the large changes in light intensity essentially become outliers are exposed. The threshold function works on changes of light intensity by looking if the change was larger than the expected variance plus 2 standard deviations.

We then look for light change events that happen at the same time and are either all getting brighter or all getting darker. Due to the 5 second sampling interval (and allowing for variability in latency of the 802.15.4 network), we consider light level changes occurring within 6 seconds of each other to be simultaneous.

We then sort the rooms in each group by the size of the change in intensity level, to determine which room the light level changed in, and determine that one event has occurred linking that room to each of the other rooms in the group. For example, if three rooms A, B, and C were the subject of a group, and room B had the largest change in light level, then we determine an indication of connectivity for AB and BC. We do not use the directionality of these events further, so a light turning on in A and being seen in B is the same as the other way around, or similar light-off events.

3.2 Determining a Transition Matrix with Motion Data

Using passive infrared (PIR) motion sensor data we can exploit occupants' everyday movements between rooms to determine room connectivity. As an occupant moves from one room to another, the sensor in the first room will cease to detect movement and the sensor in the second room will start to detect it.

This method has the advantage of working all hours when the house is occupied, not just at night as for the light-based classifier. However, this classification method does have its own challenges, as the PIR data is binary. This means that if multiple people are in the house many transitions will not be detected and erroneous ones will be detected. For example, if a person transitions from an empty room into one which is already occupied, there will be no new location detected.

PIR sensors only detect motion within their volume of sensitivity. Thus, unlike the light sensors which can detect

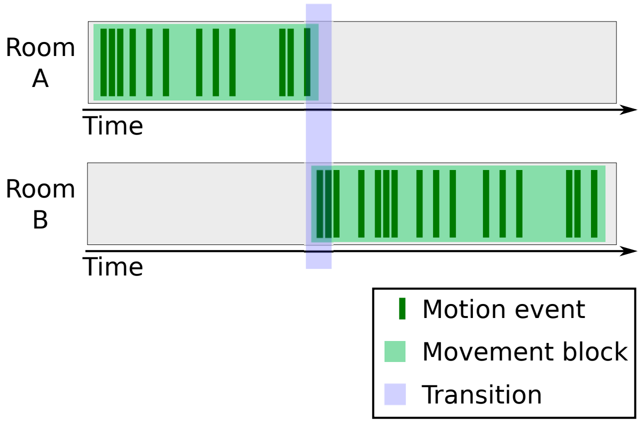


Figure 4: Motion events are grouped into movement blocks. When a block ends in one room and starts in another, a transition event is detected.

spillover ambient light from obstructed lights (e.g. in other rooms), the PIR sensors must be directed at movement to detect it. This means that motion events may be missed due to temporary obstructions, or black spots caused by sensor placement/orientation, or stillness even in an occupied room. Also, unlike with light spillover, it takes time for people to walk between the coverage zone of two sensors, which may be quite far away if they are at opposite ends of the respective rooms. So, we need to use a larger time window to detect movement, and this can introduce more opportunities for false movements to be detected.

The algorithm works as follows. We group motion events in a room that happen within 30 seconds of each other into movement blocks (Figure 4). A candidate room connectivity is created when the following conditions are both true:

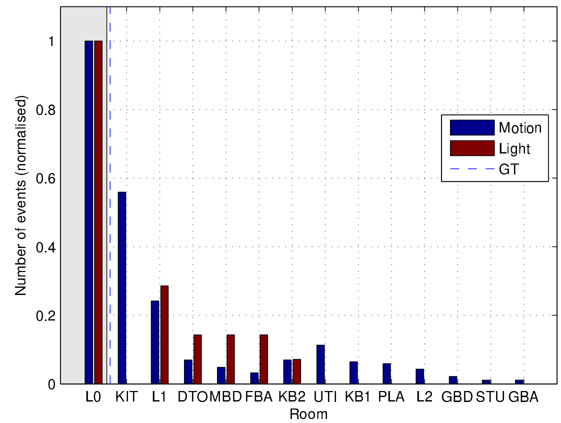
1. A movement block from one room comes to an end, while another motion blocks starts for a different room.
2. The time of one block stopping and another starting are within 30 seconds of one another.

If more than one room has motion starting, we infer a candidate connectivity only for the motion with the closest time-stamp (so that if someone walks from room A through B to C, we only infer AB not AC — and we can also infer BC since B's motion ended just before C's began).

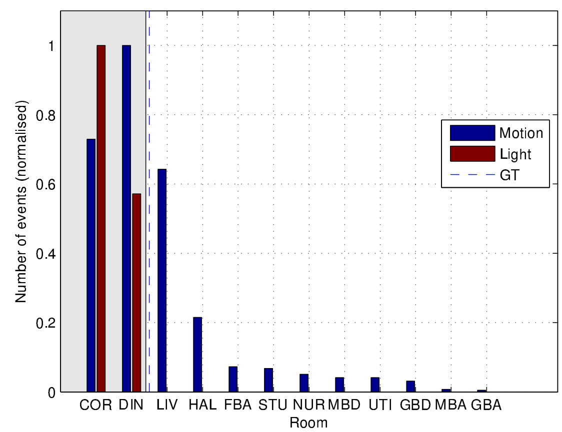
3.3 From Transition Matrix to Room Connectivity Graph

To distill a transition matrix into a list of connections, we first tried simply thresholding the list in various ways, however, this always included some false positives. This is because some rooms are used very rarely and so have few motion and light events associated, while other often-used rooms have much more motion and have the lights turned on and off with a much higher frequency, so that the occasional false positive involving that room outweighs the true positives associated with the rarely-used rooms.

To address this, for each room (A), and each sensor type, we normalize the number of events between that room and each other room ($B_{1,2,3,...}$) by dividing each by the highest



(a) Living Room, House A



(b) Kitchen, House B

Figure 5: Normalized frequencies from transition matrices, which are thresholded by the κ value (e.g., 1.0) to determine initial connections. The ground truth transitions are in the highlighted blocks on the left hand side.

number of events, so that the most frequent candidate for connectivity B_{max} always had a score of 1.0, and the others in the range 0.0-1.0. This normalization means rooms that have little use are compared on level terms with rooms that are often used. Figure 5 show graphs of such normalized data.

We can then apply a threshold, κ, in the range 0.0-1.0, which we define as the minimum score that a room B has to reach in order to be considered connected to room A. Clearly, at least one room B is connected to room A and this is reflected in the fact that the room B_{max} is always chosen (since its score is 1.0).

While the value κ=1 (i.e., for each room, only choosing the other room with the most transition occurrences as connected) works well, we will explore the effects of changing the parameter κ in the evaluation.

Once each room is considered as room A in turn, we have

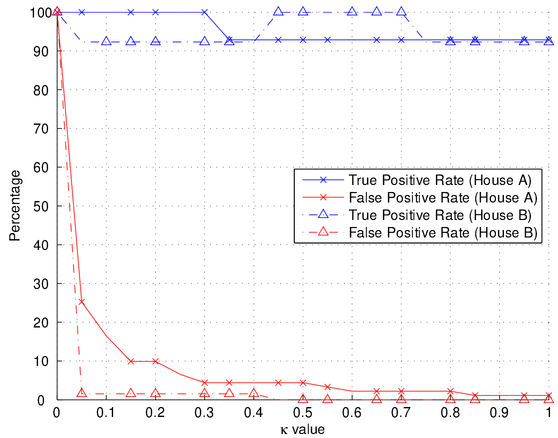


Figure 6: True positive and false positive rates for the fused light/motion classifier in each house, for various κ values.

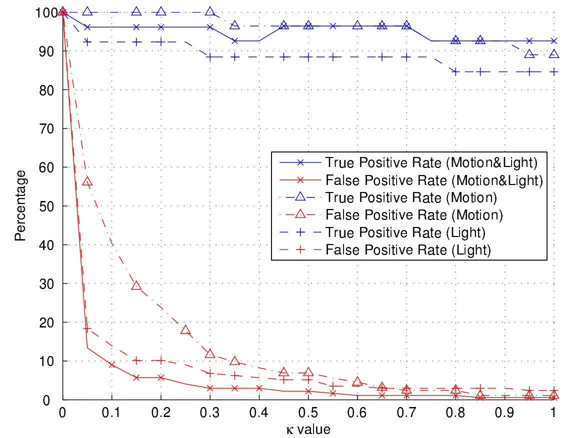


Figure 7: True positive and false positive rates for the light-based, motion-based and fused classifiers, aggregate across both houses, for various κ values.

lists of connections that arise from the different sensing techniques. The union of these lists is taken as the initial set of connections in the room graph. In other words, if room C is found to have room D as an connectivity but room D does not find room C connected, then the CD connectivity is nonetheless kept. This again promotes discovery of connections for less-travelled rooms since the DC traffic might be "drowned out" if D is heavily used as is its other neighbor(s).

On examining these initial connectivity sets, we found them to often comprise multiple disjoint groups of rooms. An event that often occurred was that all the rooms on one floor are joined but the floor is not joined to neighboring floors. We therefore applied a final a subgraph-stitching algorithm to ensure the graph is fully connected (i.e., you can get to any room from any other). To stitch the subgraphs, we take at the smallest subgraph and look at all the transition events from each room in the subgraph to each room outside the subgraph. The edge with the highest normalized frequency is chosen as an connectivity. This process is repeated until all of the subgraphs have been stitched together.

Note that the procedure above applies separately to light and motion data, and separate room connectivity graphs are produced for each.

3.4 Fusion

As we will see in the evaluation, the motion and light based results both exhibit significant errors. So, we also explored a joint motion/light classifier to improve accuracy. To avoid carrying through errors from the motion and light based results, we combine the motion and light initial connectivity set data by using the intersection of the two, i.e. where *both* classifiers agree an connectivity is present. We then apply the subgraph stitching algorithm to connect rooms together, during which we use the sum of normalized frequencies of motion and light events (i.e. range 0.0-2.0) to determine which connections to add to connect the subgraphs, also multiplying the κ parameter by 2. This makes sure that motion and light data are weighted equally at the subgraph stitching stage.

A comparison of separate and fused performance is presented in Section 4.2.

4 Evaluation

In this section, we compare the accuracy of light-based, motion-based, and fused classifiers, and we look at the effect of the κ parameter. We look at the false positive and negative connections in the context of the house layouts. We then look at the graph accuracy over time and how much data the algorithms need to converge.

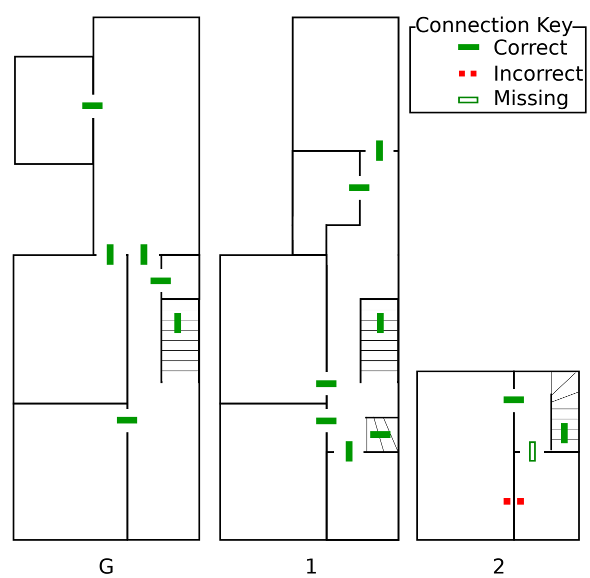


Figure 8: House A: Connections inferred from fused light/motion data with κ=1.0.

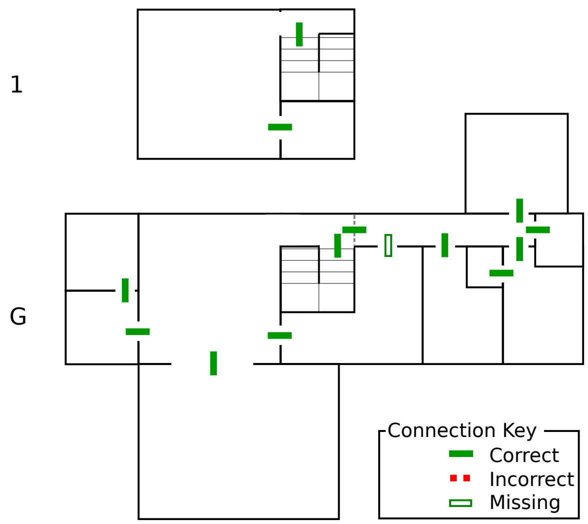


Figure 9: House B: Connections inferred from fused light/motion data with κ=1.0.

4.1 Overall accuracy and κ parameter

Figure 6 shows the true positive and false positive rates for each house, for the combined light and motion classifier. With a "default" κ value of 1.0 (i.e., each room nominates a single other room for the initial connectivity graph), in both houses all but one connectivity is found, leading to an average true positive rate (TPR) of 93%. With just one false positive in House A and none in House B, the false positive rate (FPR) is 0.5%.

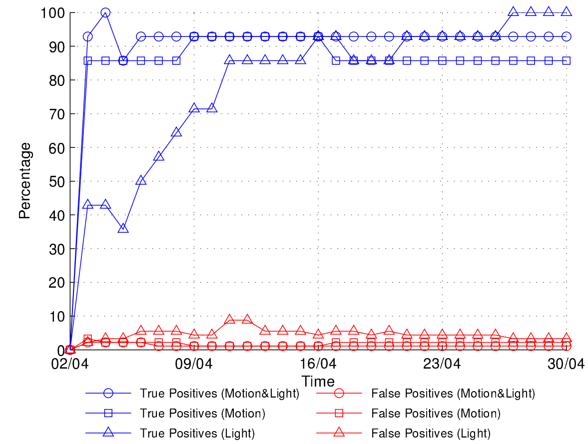
By varying the κ parameter, we can choose other operating points for the algorithm - e.g. with a κ of 0.7, the combined TPR is 96% while the FPR is 1%, and in fact House B's graph is totally accurate! However, with just two houses in the study, it is not reasonable to extrapolate that κ=0.7 is a good value in general, so we stick with κ=1.0 as a default value until further work can be conducted to explore this parameter in more houses.

4.2 Light, Motion and Both

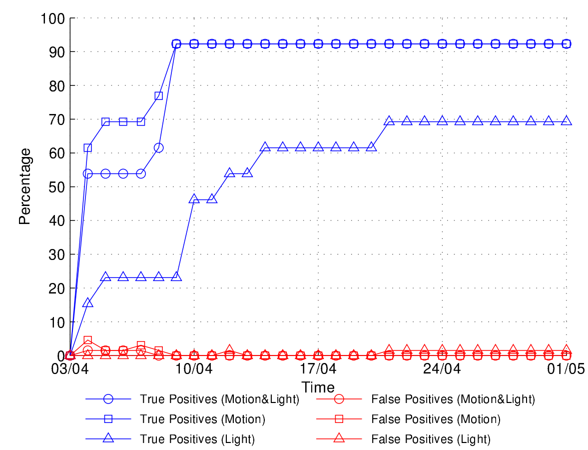
Figure 7 shows the effect of using light data and motion data alone (averaged between both houses). At κ=1, light data alone has 84% TPR and 2% FPR, while motion alone has 89% TPR and 1% FPR. By fusing the two data sources, we achieve a TPR of 93% and FPR of 0.5%, i.e. more correct connections are found while also reducing the number of incorrect connections. Thus, we have shown that when both sensors are available it is worthwhile fusing data from both sources to infer the most accurate room connectivity graph.

4.3 Understanding the errors

Figure 8 shows the connectivity graph overlaid onto the floor plan of House A. All but one of the connections were correctly classified, but an extra incorrect connectivity was also present. This was due to the space between the doors on the second floor landing (to the guest bedroom and guest bathroom) being close together - so that motion essentially happened simultaneously in both spaces, and light spillover happened between the bedroom and bathroom. While this



(a) House A



(b) House B

Figure 10: Connectivity graph accuracy over time with different classifiers.

is an error, with regards to actual applications of inferred connectivity graphs e.g. for predictive heating or for burglar alarms, the existence of this false link may not significantly harm the applications since the spaces are essentially connected.

Figure 9 shows the calculated connectivity graph overlaid onto the floor plan of House B. All but one of the connections were correctly classified with no incorrect connections due to ambiguities. Note that this floorplan has a loop, so there are more doors per room on average than for non-loop layouts such as House A. This would suggest that a lower κ value (allowing more connections to be inferred per room) works better - and indeed a κ value of 0.7 does give a totally accurate graph (adding the missing connectivity).

4.4 Convergence Time

An important aspect of an auto-calibration system such as this is how much data is required in order to get an accurate result. We therefore studied the accuracy with fewer than 30 days of data, as shown in Figures 10a and 10b. House A and B respectively took just 6 days and 4 days to converge on the final results when using both motion and light data. With motion or light data alone, convergence took 1-4 weeks and was unstable. This provides another argument for using both light and motion data where available.

5 Concluding Remarks

This paper has described new algorithms to construct a geometric model of room connectivity using single per-room sensor units as might be deployed for a burglar alarm system or smart lighting system. We describe how to use light sensors to detect artificial light spill-over from one room to the next, and motion sensors to detect movement between rooms, and a further algorithm fusing both light and motion data. Using two houses in an exploratory data set, we showed that the fusion of both types of data performed better than either alone, and achieved 93% true positive rate and 0.5% false positive rate, with a convergence time of under a week.

While this exploratory study has given promising results for two UK houses with different floorplans, there must be further investigation into the generality of our solution. Primarily, this means evaluating the algorithms against sensor deployments in a wider variety of floorplans; as with previous methods, a particular challenge may prove to be "open plan" designs. Where we have applied heuristics, we have tried to describe these clearly, in terms of the attributes of the sensor data (e.g. sampling rates) so that others can reproduce the algorithm. It remains to be seen how well our algorithms can be applied to sensor data sets with different attributes.

Acknowledgement. Lancaster University's involvement in this research was supported by the EPSRC (grants EP/I00033X/1 and EP/G008523/1).

6 References

[1] Jaikang Lu and Kamin Whitehouse. Smart blueprints: Automatically generated maps of homes and the devices within them. In *Proceedings of Pervasive Computing*, 2012.

[2] James Scott, A J Bernheim Brush, John Krumm, Brian Meyers, Mike Hazas, Steve Hodges, and Nicolas Villar. Preheat : Controlling home heating using occupancy prediction. *Proceedings of UbiComp 2011*, pages 281–290, 2011.

[3] Michael C. Mozer. The Neural Network House: An Environment that Adapts to its Inhabitants. In *Proceedings of AAAI Spring Symposium on Intelligent Environments, March 23-25, 1998, Stanford University, Palo Alto, CA, USA*, pages 110–114, 1998.

[4] Henry C. Spindler and Leslie K. Norford. Naturally ventilated and mixed-mode buildings–part ii: Optimal control. *Building and Environment*, 44(4):750 – 761, 2009.

[5] Cory D. Kidd, Robert Orr, Gregory D. Abowd, Christopher G. Atkeson, Irfan A. Essa, Blair Macintyre, Elizabeth D. Mynatt, Thad E. Starner, and Wendy Newstetter. *The Aware Home: A Living Laboratory for Ubiquitous Computing Research*. 1999.

[6] Hartmut Surmann, Andreas Nchter, and Joachim Hertzberg. An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45:181–198, 2003.

[7] Grant Schindler, Christian Metzger, and Thad Starner. *A Wearable Interface for Topological Mapping and Localization in Indoor Environments*. 2006.

[8] Robert K. Harle and Andy Hopper. Using Personnel Movements for Indoor Autonomous Environment Discovery. In *IEEE International Conference on Pervasive Computing and Communications*, pages 125–132, 2003.

[9] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Reconstructing building interiors from images. In *International Conference on Computer Vision*, pages 80–87, 2009.

[10] MC Mozer and L Vidmar. The neurothermostat: Predictive optimal control of residential heating systems. *Neural Information Processing Systems*, May 1997.

[11] Manu Gupta and Stephen Intille. Adding gps-control to traditional thermostats: An exploration of potential energy savings and design challenges. *Pervasive Computing*, 5538:95–114, 2009.

[12] B. Brumitt, J. Krumm, B. Meyers, and S. Shafer. Ubiquitous computing and the role of geometry. *Personal Communications, IEEE*, 7(5):41 –43, oct 2000.