Sensorium: Commissioning Abundant Sensors with Augmented Reality and QR Codes

Eric A. Bier[1], Saman Mostafavi[1], Lester D. Nelson[1], Alejandro E. Brito[1], Raman Goyal[1], Shamus Li[2], Ritesh Sharma[3*], Mahabir S. Bhandari[4], Niraj Kunwar[4]
[1]Palo Alto Research Center, Inc., part of SRI International, Palo Alto, CA, USA
[2]University of California at Berkeley, Berkeley, CA, USA
[3]University of California at Merced, Merced, CA, USA
[4]Oak Ridge National Laboratory, Oak Ridge, TN, USA

Abstract

In the future, it will be possible to build high-quality models of building interiors based on data from a dense fleet of sensors reporting on air volumes much smaller than a room or zone. To enable such models, we are creating technologies that commission a fleet of sensors quickly at low cost. Our sensor commissioning process builds a three-dimensional model of each building interior that includes sensor positions and sensor networking information such as sensor Media Access Control (MAC) addresses. It employs Augmented Reality, Light Detection and Ranging (LiDAR), Quick Response (QR) codes, and computer vision. Sensors can be commissioned at more than ten times the speed and at less than one tenth the cost of traditional approaches.

Highlights

- Sensor positions are tracked and a data streaming system collects their data, in a single session
- More than 10x the speed and less than 1/10th the cost
- Can support dozens of sensors per room
- Automatically produces a 3D model, floor plans, and a database of sensor data values

Introduction

Sensors are used in buildings to support many applications that enhance performance, including energy efficiency, occupant comfort, and equipment monitoring. More recently, wireless sensor networks have been integrated into heating and cooling systems (Pang, 2019), heat flow modelling (Shan, 2019), air quality assessment (Salman, 2019), and lighting control (Magno, 2014). However, the implementation of sensor networks in buildings faces challenges related to cost and complexity.

The existing literature highlights the need for pervasive sensor networks in buildings, to collect performance data effectively (Chiou, 2007). Nevertheless, the process of adding sensors to a building has traditionally been expensive and time-consuming, involving updates to building management systems, installation of new signal collection hardware, and revisions to blueprints, floor plans, and building models (Pang, 2019). The cost of commissioning wireless sensors can be high, discouraging building owners and managers from optimizing sensor configurations for occupant comfort and energy efficiency.

Recent advancements in augmented reality (AR) technologies and computer vision show the potential to address these challenges. Zhen (2020) demonstrates the use of computer vision to determine the position, orientation, and data content of QR codes. Furthermore, O'Donnell (2019) discusses the benefits of automatic mapping for building modelling, highlighting the importance of detailed sensor data for improved building performance.

Contribution.

We introduce Sensorium, a novel commissioning system that overcomes the challenges associated with deploying sensors in buildings. Our system leverages AR technologies and computer vision to achieve several goals concurrently, including capturing the 3D geometry of building interiors, determining sensor positions with centimetre-level precision, facilitating wireless BLE sensor data collection and streaming, generating building floor plans, and providing interactive tools for visualizing and modifying floor plans, 3D models, and sensor readings in their geometric context. Other aspects of sensor commissioning, such as data validation are not performed by Sensorium, but can be layered on top of it.

Sensorium significantly improves the sensor deployment process by enabling the rapid commissioning of up to 25 BLE sensors per room and hence hundreds of sensors per building. The system is compatible with low-cost wireless sensors and gateways, making it feasible to commission large numbers of sensors across various applications.

Our contribution to the field is twofold:

1. We present a comprehensive commissioning system that integrates AR and computer vision to streamline sensor deployment, making it more accessible and affordable for building owners and managers.
2. We demonstrate the versatility of Sensorium, showing its potential in multiple applications: providing input to sub-zone level control systems, facilitating personal comfort device control, and supporting predictive maintenance in factories and mechanical areas.

By addressing the challenges associated with sensor deployment in buildings, Sensorium has the potential to revolutionize the way sensors are utilized, ultimately enhancing occupant comfort and energy efficiency.

[*] Worked on this research during his internship at the Palo Alto Research Center

https://doi.org/10.26868/25222708.2023.1424

Methods

This section describes the Sensorium workflow as it commissions sensors. First, the user places wireless sensors in a container that is easy to carry, along with gateway devices for streaming the sensor data. The user also brings an augmented reality device, such as a Microsoft HoloLens 2 headset or an iOS device with LiDAR, such as an iPhone 12-14 or iPad Pro; this device will be loaded with the Sensorium AR application and used to collect information about the building's walls, floors, ceilings, doors, windows, vents, thermostats, and sensors.

Digitizing walls, floors, ceilings, and furniture

Next, the user walks around inside the building of interest, looking in different directions, allowing the AR device to map the building interior from different directions and collect information about its geometric features.

Figure 1. As the user walks through a living room, Sensorium AR builds a triangle mesh model of the building interior.

If the HoloLens is used as the AR device, it generates a triangle mesh to represent the geometry of the building. The user watches the mesh forming, superimposed over the actual building geometry as viewed through the HoloLens. Figure 1 shows an example, where triangles are starting to appear on top of visible building features in a residential living room. The user can see which parts of the room are well-covered by the mesh and which require more triangles. The user can walk to the parts that need more triangles, until the entire room is well-covered.

If the iOS AR device is used to capture the building geometry, the Apple RoomPlan libraries generate white lines to show those places where a wall meets the floor, the ceiling, or another wall. The white lines also indicate positions of doors and windows, and form bounding boxes around pieces of furniture and other objects.

Figure 2. Feedback from the Apple RoomPlan libraries is overlaid on top of the live image of the room.

Figure 2 is an example of the feedback provided by the RoomPlan libraries. The user can see where lines are incomplete or in the wrong place and spend more time in those areas until the model is accurate. A tiny room model appears during the room scanning process (in the lower middle of the figure) to show the model as it forms.

Marking doors and windows

In the HoloLens case, the user marks doors and windows using eye gaze and voice commands. For example, the user can look at three of the four corners of a window, giving the command "Mark" at each corner. Once the three corners are marked (with white spheres), the user gives a final voice command "Fill" and the system generates a rectangle that touches the marked locations, as shown in Figure 3. In the iOS case, the RoomPlan code estimates door and window positions automatically.

Marking sensors and discovering MAC addresses

During the AR session, the system builds a model of the position of each sensor within the environment. In the HoloLens case, the user can direct his/her eye gaze at a sensor and give a voice command to place a visual annotation at the location of the sensor. Figure 4 shows the view through the HoloLens as Sensorium adds a semi-transparent block around the sensor to mark its position and then displays some metadata above the block. Later, this block will become part of the 3D model of the scene, showing where the sensor was placed. Sensors can be placed on walls, tables, pieces of mechanical equipment, etc. The user can mark both newly added sensors and pre-existing sensors.

In addition to finding the position and orientation of the BLE sensor, we need to discover its MAC address to correlate the sensor data being collected by our sensor data streaming (SDS) system with the position and orientation of that sensor in a room.

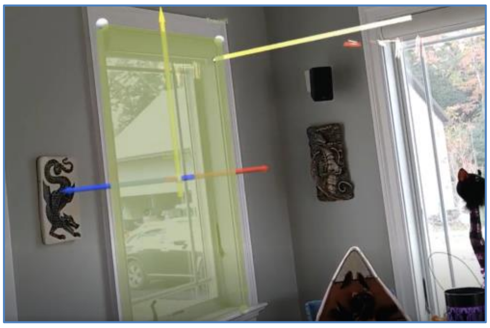

Figure 3. A user adds a window object by placing three "trace" objects (white spheres) and then giving a voice command. The window object is displayed as a yellow-green rectangle.

In the HoloLens case, we find the MAC address by taking advantage of the HoloLens' built-in Bluetooth transceiver. Using this transceiver, we receive BLE advertisements from nearby BLE devices. After filtering out devices of no interest, we take the device with the highest signal strength. To increase accuracy, the user may pick up the sensor and place it near the HoloLens during this process. After a few BLE sensor advertisements, the user presses a button in the augmented reality user interface to record the MAC address of the nearest device. This MAC address is then associated with the sensor object to be added by the HoloLens application.

https://doi.org/10.26868/25222708.2023.1424

Using iOS, we attach a QR code to each BLE sensor, which encodes the MAC address of that sensor. Once the QR code is affixed to the sensors, they can be placed at their desired locations. Next, the user walks through a room in two passes. In the first pass, the iOS device records the positions of walls, floors, ceilings, and pieces of furniture. Next, the user starts up a second stage of our iOS application, built to find QR codes. The user makes a second pass through the room, this time pointing the iOS device's camera at each of the sensors. Computer vision algorithms detect the QR codes. Each QR code is processed in two ways. First, the QR is decoded to determine the MAC address of the sensor. Second, the position and orientation of the QR code in 3D space is computed using the same coordinate system that was used to record the walls, floors, ceilings, and furniture. These QR code positions are used as the positions of the sensors and are added to the 3D model that Sensorium constructs.

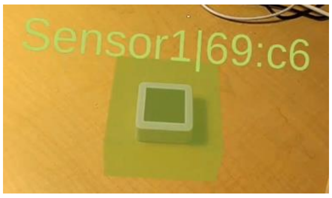

Figure 4. The user marks a sensor. Sensorium places a colored block around the sensor and displays metadata near the block.

The QR code method significantly reduces the amount of time needed to record the position and MAC address of each sensor in a room. However, it requires affixing a QR code to each sensor in advance. To automate this process, Sensorium includes an application that generates the printable QR codes. A user starts the QR generation application and holds in the Bluetooth pairing button on each sensor. The QR generation application detects that a sensor is in pairing mode, uses the BLE protocol to get the MAC address of that sensor, and then generates a QR code image that encodes the MAC address. The user sends the QR to a printer, scaling it so that it fits on the sensor and pastes the QR code to the sensor. Note that QR codes can be affixed to any sensor, whether it is newly added or pre-existing, so this method can also be used to model legacy sensors.

The application adds additional geometry to the augmented reality view that the user sees. In Sensorium, a white box, similar in size to the sensor, is added to the scene at a location just in front of the QR code. The QR code's decoded data (a MAC address) is displayed on the front face of the white box.

As the computer vision algorithm detects each sensor, the position, orientation, and MAC address of each sensor is determined in a few seconds per sensor, allowing the addition of more than 12 sensors per minute to our database.

Geometric processing

Once the user has completed a session with the HoloLens or iOS device, Sensorium processes the collected data to build a three-dimensional model and several kinds of floor plan of the building. While the details of the geometric modelling will be described in a separate paper, we summarize the processing steps here.

With the HoloLens, our app exports a triangle mesh as an OBJ file and the positions of sensors, doors, windows, and other metadata in a comma-separated values file. Our geometric processing identifies the most likely direction of gravity and orients the model so the floors are below and levelled, and the ceilings are above and levelled. Using a spherical coordinates k-means algorithm (like the one used to cluster point cloud normals in Straub (2015)), it finds the dominant facing directions of walls and orients the model so that the main walls line up with the primary coordinate axes. The ceiling is removed by finding triangles that face down or nearly down.

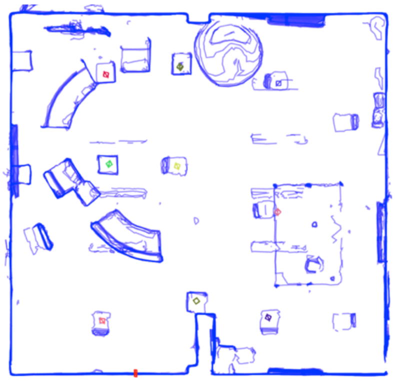

Figure 5. A pen-and-ink style floor plan of a large room, from HoloLens data. The colored shapes are sensor positions.

During the above steps, the application records the coordinate transformations that are applied to the mesh and applies the same transformations to the sensors, walls, doors, etc. to produce a consistent 3D model. This model is sliced along planes parallel to the floor at one or more altitudes, projecting the resulting slices down to a single plane to produce a pen-and-ink floor plan (e.g., Figure 5).

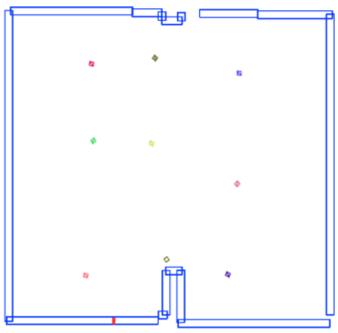

Figure 6. A drafting-style floor plan of the conference room from the previous figure, computed from HoloLens data. The nine colored dots in the interior are sensor positions.

For some applications, we also require simplified drafting-style floor plans. Sensorium generates these by fitting rectangular blocks to regions of mesh triangles that face in the same direction and are largely co-planar. Blocks that are too small or too far from the floor or ceiling are discarded. The remaining blocks are sliced at an altitude

https://doi.org/10.26868/25222708.2023.1424

between the ceiling and floor to produce the floor plan (e.g., Figure 6).

Using iOS, only a drafting-style floor plan is generated. The RoomPlan library detects some types of furniture. Each furniture piece is replaced with a block. Figure 7 shows a 3D model made from iOS data using the RoomPlan libraries in the same conference room as Figure 5 and Figure 6. Twenty sensors with QR codes were placed on the table in the middle right corner (shown as red dots).

Sensorium produces 3D models and floor plans of building interiors with sensor positions and can detect the network identifier of each sensor. As in Okorn (2010), we generate floor plans automatically from cluttered environments. However, in our case, we begin with triangle mesh data instead of a point cloud, integrate sensor positions, and produce both 3D models and floor plans.

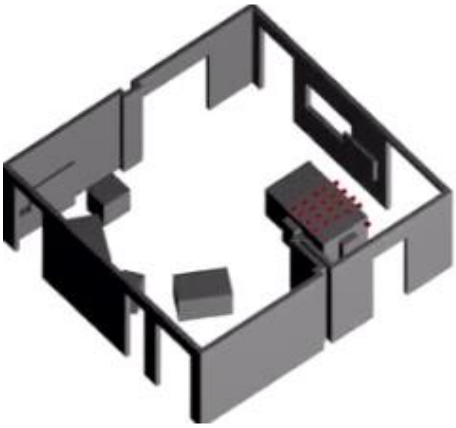

Figure 7. A 3D model with 20 sensors made on the iOS platform.

Sensor Data Streaming (SDS) System

In parallel, as the user is placing the sensors and digitizing a room, the SDS system starts the data collection process. Figure 8 shows a block diagram of our SDS architecture. This notional architecture can run on cloud hosting services or on servers available at the building site. As in Hunkeler (2008), our SDS system uses message queues to deliver sensor data.

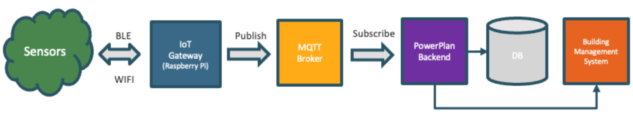

Figure 8. A diagram of our sensor data streaming architecture. Sensor readings flow from the sensors to the gateways and on to a message queueing protocol broker, which streams the values to a database and any other applications that need the values.

The Sensorium SDS system consists of sensor gateways, a Message Queuing Telemetry Transport (MQTT) broker, a database (DB), and a backend server implementing services to stream the sensor data to the DB and provide access to the DB data via RESTful APIs.

We use Raspberry Pi computers to implement the sensor gateways. For the use case of BLE temperature and humidity sensors, a sensor data acquisition service is installed on a Raspberry Pi. This service handles the sensor network protocol, acquires the sensor payload, decodes the sensor data values, and publishes them to the MQTT broker. When a user is commissioning sensors, the gateways can be placed in different parts of the building to maximize the sensor placement coverage and to ensure that every sensor can communicate with a gateway. A single gateway can handle 20 sensors or more if they are within Bluetooth range of the gateway.

Once the sensor data is acquired and published to the MQTT broker, a backend service subscribes to the MQTT broker and persists the streamed sensor data to a database. The Sensorium SDS system supports Postgres and SQL DBs in site or on the cloud. Additionally, a backend service is available to persist the building model data collected by the Sensorium AR application, which includes 3D room models, floorplans, sensor positions and orientations, and other metadata generated during the sensor commissioning process.

There is also a backend service implementing various RESTful APIs for client applications to request data from the sensors and the building models, or to compute analytics using the sensor data. One example of the later is describe in the next section.

The PowerPlan Web UI

Finally, Sensorium includes a Web-based user interface, called PowerPlan (see Figure 9) that allows the user to see the floor plan and 3D model side by side, to add annotations to the floor plan, and to view the sensor data in the context of its associated floor plan location.

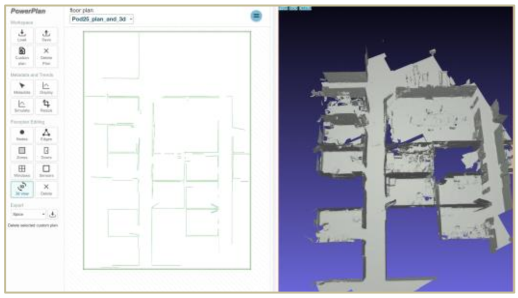

Figure 9. The PowerPlan web user interface.

Sensor Data Application

Using the sensor location information, the sensor network identifiers, and the database of sensor readings, we can generate a model of how the environment differs from place to place and moment to moment. For example, heat maps like the ones shown in Figure 11 can be generated to show temperature or humidity differences over time.

A history of sensor readings can be analysed to observe how a building reacts to changing external influences, such as the changing temperature of the outside air over the course of a day, or the heating and cooling actions of the building's HVAC system.

Results

In this section, we describe our results. We describe the kinds of buildings we have been able to digitize, the accuracy of the models, the usability of the system, the size of the sensor fleets deployed, and the estimated reduction in cost to deploy a large fleet of sensors.

https://doi.org/10.26868/25222708.2023.1424

Diverse Types of Buildings

Using the HoloLens version of Sensorium, we have digitized more than two dozen building models including models of a single-story home; several multi-story homes; sections of an office building; a large conference room with chairs, sofas, and tables; two floors of a small office building; part of the mechanical area of a commercial building, including pipes, pressure vessels, and gauges; and a hotel room. Some of these models are shown in Figure 10, Figure 12, and Figure 13.

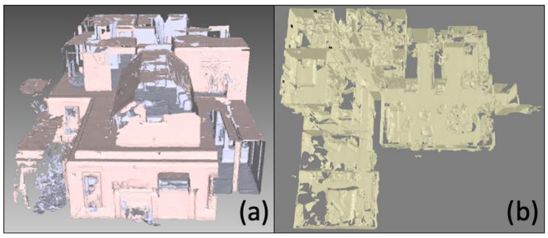

Figure 10. Residential models captured with the HoloLens: (a) a house with a small second story, (b) part of a one-story home.

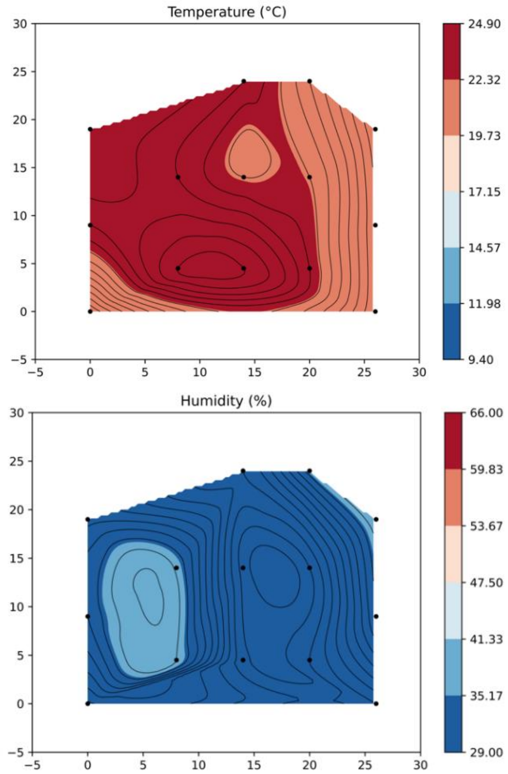

Figure 11. Heat map showing temperature and humidity interpolation across a single zone in a building, derived from the readings of an array of sensors (shown as black dots).

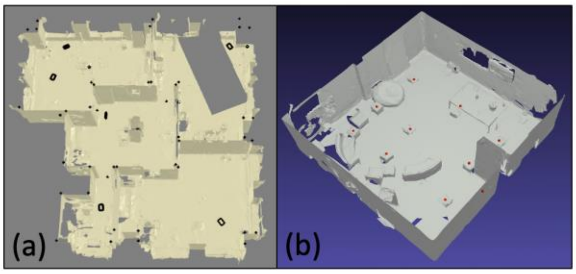

Figure 12. Commercial models captured with the HoloLens: (a) one floor of an office tower, (b) a large conference room.

The iOS version of Sensorium is newer, and we have had less time to test it. Nonetheless, the tests so far are quite promising. Figure 14 shows part of an office building, as digitized by Sensorium using an iPad Pro.

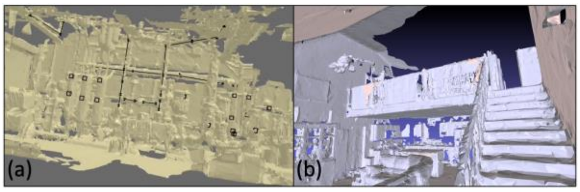

Figure 13. Other HoloLens models: (a) an HVAC equipment area with pipes and pressure vessels, (b) a home with a loft.

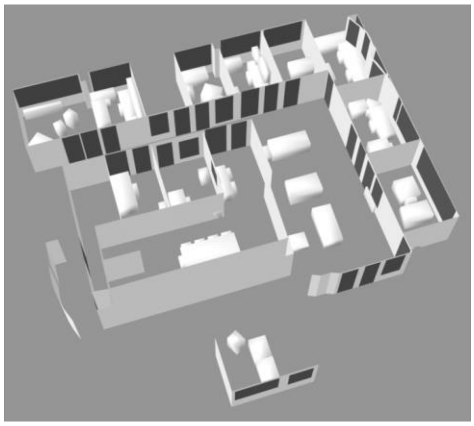

Figure 14. Digitized on an iPad: part of a commercial building with over ten offices, a conference room, and a lounge.

Building Model Accuracy

In Weinmann (2021) a team from KIT evaluated the Microsoft HoloLens 1 for its ability to produce accurate 3D models of building interiors. They compared its performance to the data from a Leica HDS6000 TLS system and found that the HoloLens was able to digitize both room-scale and building-scale environments with an accuracy in the range of a few centimetres.

As we are using a different generation (the HoloLens 2) with newer software, we decided to test the accuracy as well. We manually measured one floor of an office tower (see Figure 12(a)) and scanned it twice using Sensorium. We compared the floor areas of the scanned rooms as computed by Sensorium to the hand-measured areas and computed the error as shown in equation (1):

$$error = |(A_s - A_m)| / A_m \qquad (1)$$

https://doi.org/10.26868/25222708.2023.1424

where A_s is the area as measured by Sensorium and A_m is the hand-measured area. For four out of the five rooms, the error was below 1.7%. For the small room in the upper right corner, which is near a stairwell, the results were less consistent, with errors of 12.8% and 30.8% respectively. Our conclusion is that the accuracy is acceptable for our applications when scanning traditional office rooms. We have not yet evaluated the accuracy of the models from the iOS devices, but given that they are based on LiDAR data, we expect similar or better results.

System Usability

As a usability test, we made our HoloLens Sensorium AR available to a team at another institution (name to be provided after blind review). They installed the technology on their own HoloLens 2, learned the eye gaze and voice commands provided and succeeded in making several models of their own building interiors, including the one shown in Figure 12(a). Thanks to their feedback and our own internal trials, we improved the user interface.

Sensor Fleet Size

We have verified that both HoloLens and iOS versions of Sensorium can be used to record the locations of large numbers of sensors. For example, the HVAC model of Figure 13(a) includes 13 sensors and gauges, all of which were added in a single HoloLens session. The office building model of Figure 12(a) includes 59 annotations including five sensors and 39 objects to mark the upper corners of each room; these corner objects are added using voice commands like those used to place sensors. This session took 30.5 minutes.

In the case of the iOS-based AR, we were able to achieve a larger fleet of sensors together with their MAC addresses. For example, the large conference room of Figure 7 has an array of 20 sensors on a single table. The room, the table, and all 20 sensors were digitized in 3 minutes 51 seconds, including the time to expose those sensors that were hidden behind a personal heater.

Table 1: Time to add sensor locations in several trials

Method	# Sensors	Example	Duration
HoloLens	13	HVAC model of Figure 13(a)	14.4 minutes
HoloLens	5 sensors + 54 other objects	Office building model of Figure 12(a)	30.5 minutes
iOS/QR	20	Large conference room of Figure 8	3.9 minutes
iOS/QR	8	Demo with 8 sensors on a table	24 seconds

In a demonstration where eight sensors were placed near each other on a table, the QR-code based sensor tracking was able to find and decode all eight sensors in 22 seconds or less than three seconds per sensor. In addition, when one of the sensors was picked up and moved, QR-code based tracking found its new location in under four seconds. Because the QR code encodes the MAC address of each sensor, this approach also assembles a complete set of sensor MAC addresses.

Reduction in Cost

In support of a technoeconomic analysis for a prior project, our building controls provider estimated installation of a wired sensor to take about 4.5 hours of time from an electrician, charging about $175 per hour, totalling $83,000 in labor to install 100 sensors. We also learned that installing a wireless sensor would take about 1 hour and require the skills of an HVAC technician at $135 per hour; 100 sensors would cost over $13,000 in labor.

Sensorium provides a large reduction in commissioning cost. To measure the time per sensor for the HoloLens method, we recorded several sessions on video and measured sensor addition time from the video. As the technology improved, the time in seconds to capture the sensor position and MAC address went from 72 to 57 to 40.

At 40 seconds per sensor, it should take 67 minutes to deploy 100 sensors. They can be placed by an installer with a few hours of training paid $50 per hour, for a total of $56 labor cost. With wireless sensors costing $10 each, 100 sensors cost $1,000. Six gateways add $300. The total cost is $1,356. That is 9.6x less than the labor alone for traditional wireless sensors. In addition, the installation time drops **from 100 hours to 1 hour**.

Table 1 shows the total elapsed time we measured from some of our experiments; the iOS/QR times include finding MAC addresses. Table 2 shows the estimated difference in installation time and cost for Sensorium compared to traditional sensor installation.

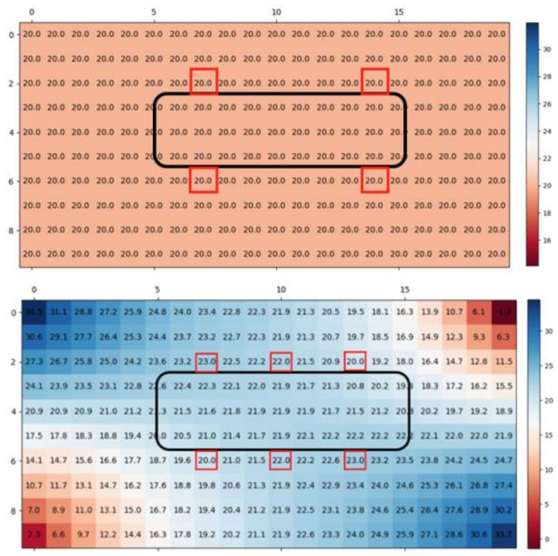

Figure 15. PDE-based heat control ensuring sub-zone level control (indicated by red boxes) based on occupant preferences. Sensorium collected the data for calibrating the PDE models.

Using the iOS/QR method, we reduce the time further. We measured 3 seconds per sensor in our first trial (on 8 sensors) and 3.6 seconds per sensor in a second trial (72 seconds for 20 sensors).

https://doi.org/10.26868/25222708.2023.1424

Table 2: Cost reduction of the two Sensorium methods

Variable	Traditional Wired Sensors	Traditional Wireless Sensors	HoloLens Method	QR Code Method
Labor cost per sensor	$830	$130	$0.56	$0.38
Time per sensor	4.5 hours	1 hour	40 seconds	27 seconds
Total labor cost (100 sensors)	$83,000	$13,000	$56	$38
Total installation time (100 sensors)	100 hours	100 hours	1.1 hours	0.75 hours
Cost reduction	N/A	N/A	9.6x	9.7x

In the second trial, it also took 159 seconds to digitize the room. So, the whole process took 231 seconds or about 12 seconds per sensor plus the time to enter the room and place the sensors. Adding on 5 minutes for those activities, we compute 27 seconds per sensor. At that speed, 100 sensors can be commissioned in 0.75 hours for a labor cost of less than $38. The installation time drops from 100 hours to 0.75 hours, or a **133x time reduction**.

If we run Sensorium on an existing sensor fleet, i.e., to update the model when some sensors have moved, there are no hardware costs and the building geometry has already been digitized; sensor re-commissioning can be done in **6 minutes for 100 sensors,** making it nearly free.

Discussion

In this section, we describe applications that will be enabled by the ability to commission sensors at high speed and low cost. Next, we compare the HoloLens approach to the iOS approach and suggest applications that are best suited for each. Finally, we discuss additional work needed to allow such technologies to be adopted widely.

Applications Enabled by a Fleet of Sensors

Having abundant cheap sensors enables a variety of applications including 1) building simulation and modelling, 2) optimization of personal comfort devices, 3) optimization of sensor and actuator placement, and 4) predictive maintenance. We discuss these next.

(1) Simulating energy performance. As described in Bottaccioli (2017) and Mostafavi (2017), building energy simulations and models can be used to evaluate building energy performance and to evaluate possible building improvements. Some of the key components of such a system are sensors for monitoring temperature and humidity and IoT gateways for collecting sensor data. By providing rapid low-cost commissioning of sensors, including data collection, and geometric information about building interiors, Sensorium can support these systems. In addition, it produces a 3D model of as-built building geometry, which can support Building Energy Modelling like that described in Garwood (2018).

(2) Personal comfort devices and sub-zone temperature control. In open plan office spaces like buildings with cubicles, temperature can vary significantly within a single zone. Sensorium data assists in providing initial and boundary condition data to model spatiotemporal temperature profiles within these spaces. This information can then be utilized for nonlinear control of sub-zones within a larger area, with the support of personal comfort devices. Figure 15 demonstrates an example of PDE control, using Sensorium for model calibration.

(3) Optimal sensor and actuator placement. Zhang (2019) highlights problems that can emerge due to improper sensor placement. When a permanent installation of a dense sensor fleet may not be practical, Sensorium can temporarily collect enough data for modelling. This model can guide the design of a more permanent sensor collection, such as wired sensors. Figure 16 illustrates a scenario in which data gathered by Sensorium is used to recommend the addition of sensors to specific wall locations, enabling the control of subzones around a table in an office space. (The authors have submitted papers discussing optimal placement and control algorithms to the same conference.)

(4) Predictive maintenance. Sensors for sound, vibration, heat, light, etc. can be used as input to algorithms that detect when equipment in factories or mechanical areas will soon need repair or replacement. Sensorium allows such sensors to be added with fewer resources. As noted in Geoffroy (2019) such improved maintenance can also increase building efficiency.

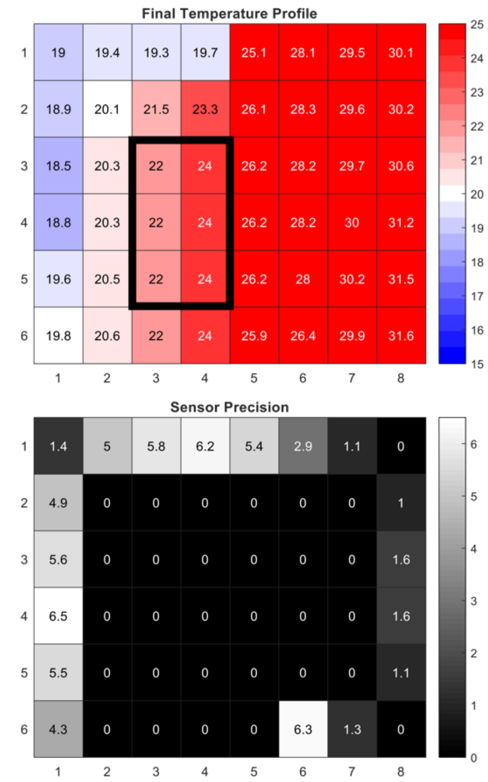

Figure 16. Using optimal sensor placement to achieve desired temperatures in the bounded box: The higher

https://doi.org/10.26868/25222708.2023.1424

precision sensor values indicate a greater need for sensors. The final temperature profile in the top figure is obtained by selecting the top three sensors from the bottom figure.

HoloLens vs. iOS methods

Having tried both HoloLens and iOS methods for digitizing building geometry and sensor locations, we have seen the trade-offs between them. Geometry from the HoloLens includes more details of the environment, such as furniture, pipes, ducts, and equipment. We prefer the HoloLens when these features are important. The pen-and-ink style floor plans produced by this method preserve these features.

In contrast, the iOS/QR code method reduces the cost to add many sensors, particularly if the same sensors are moved to a new zone or are moved to different locations in the same zone and the model needs to be updated. In these situations, we prefer the iOS/QR method.

Enabling broad adoption

Given that Microsoft support for the HoloLens will be reduced in the future, we recommend that an alternative technology be used for those cases where details of the environment are important. It may be that the iOS/QR code platform can be modified to provide these details while still providing tidy 3D models and floor plans. We expect that the iOS platform will have additional capabilities in newer versions. For example, LiDAR was introduced for the iPhone 12, but with each passing year, a larger percentage of active iPhones have that capability.

Conclusion

We have presented the Sensorium system, which can be used to reduce the cost of commissioning sensors by more than 10x, while increasing the speed of commissioning sensors by more than 10x. Sensorium does this by combining augmented reality technologies, computer-vision-based QR code recognition, wireless gateways, and data streaming technologies. It has been tested on both commercial and residential buildings.

Sensorium can support a variety of applications, including building energy efficiency, occupant comfort, optimization of sensors and personal comfort devices, and predictive maintenance.

Acknowledgments

We thank the Xerox Corporation for supporting this research and the U.S. Department of Energy (under contract No. NFE-21-08865) for providing resources to help test this technology.

References

Bottaccioli, Lorenzo, Alessandro Aliberti, Francesca Ugliotti, Edoardo Patti, Anna Osello, Enrico Macii, and Andrea Acquaviva. "Building energy modelling and monitoring by integration of IoT devices and building information models." In 2017 IEEE 41st annual computer software and applications conference (COMPSAC), vol. 1, pp. 914-922. IEEE, 2017.

Garwood, T. L. et al. (2018) 'A framework for producing gbXML building geometry from Point Clouds for accurate and efficient Building Energy Modelling', Applied Energy, 224(May), pp. 527–537. doi: 10.1016/j.apenergy.2018.04.046.

Geoffroy, Hugo, Julien Berger, Benoît Colange, Sylvain Lespinats, Denys Dutykh, Gérard Sauce, and Catherine Buhé. "Use of multidimensional scaling for fault detection or monitoring support in a continuous commissioning." In Proceedings of the 16th IBPSA Conference, pp. 877-884. IBPSA, 2019.

Hunkeler, Urs, Hong Linh Truong, and Andy Stanford-Clark. "MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks." In 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08), pp. 791-798. IEEE, 2008.

Magno, M., Polonelli, T., Benini, L., & Popovici, E. (2014). A low cost, highly scalable wireless sensor network solution to achieve smart LED light control for green buildings. *IEEE Sensors Journal*, *15*(5), 2963-2973.

Mostafavi et. al. "A calibration process focused on predicting both energy performance and indoor thermal conditions." In IBPSA Building Simulation 2017, San Francisco, 7-9 August 2017. IBPSA, 2017.

O'Donnell, J. et al. (2019) 'LiDAR point-cloud mapping of building façades for building energy performance simulation', Automation in Construction, 107(July), p. 102905. doi: 10.1016/j.autcon.2019.102905.

Okorn, Brian, Xuehan Xiong, Burcu Akinci, and Daniel Huber. "Toward automated modeling of floor plans." In Proceedings of the symposium on 3D data processing, visualization and transmission, vol. 2. 2010.

Pang, Z. et al. (2019) 'Location aided commissioning of building automation devices enabled by high accuracy indoor positioning', Journal of Industrial Information Integration, 15(January 2018), pp. 258–263. doi: 10.1016/j.jii.2017.12.002.

Salman, N., Kemp, A. H., Khan, A., & Noakes, C. J. (2019). Real time wireless sensor network (WSN) based indoor air quality monitoring system. *IFAC-PapersOnLine*, *52*(24), 324-327.

Shan, X., Xu, W., Lee, Y. K., & Lu, W. Z. (2019). Evaluation of thermal environment by coupling CFD analysis and wireless-sensor measurements of a full-scale room with cooling system. *Sustainable cities and society*, *45*, 395-405.

Straub, Julian, Jason Chang, Oren Freifeld, and John Fisher III. "A Dirichlet process mixture model for spherical data." In *Artificial Intelligence and Statistics*, pp. 930-938. PMLR, 2015.

Weinmann, Martin, Sven Wursthorn, Michael Weinmann, and Patrick Hübner. "Efficient 3D mapping and

https://doi.org/10.26868/25222708.2023.1424

modelling of indoor scenes with the Microsoft HoloLens: a survey." PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science 89, no. 4 (2021): 319-333.

Yun-Shang et. al. "Sensor-based information modeling for life-cycle commissioning of residential buildings." In Tenth International IBPSA Conference, edited by Yi J, Zhu YX, Yang XD and Li XT, pp. 1572-1579. 2007.

Zhang et. al. "Commissioning of HVAC systems in a campus building with regard to the indoor environment and energy performance." In Proceedings of Building Simulation 2019: 16th Conference of IBPSA, pp. 806-813. IBPSA, 2019.

Zhen, Shi. "Web-based indoor positioning system using QR-codes as markers." (2020)

https://doi.org/10.26868/25222708.2023.1424