How to Measure the Future

Jason Griffey

There have been a lot of words written about the Measure the Future project, and even more presentations given about it over the last two years. What I hope to accomplish here in this issue of *Library Technology Reports* is not so much to revisit what's been said about the project. I also don't want to write a pitch or marketing message for the project. What I really want to do is to tell the story of Measure the Future, from the initial ideas that started it on its way to the current state and where we hope to be in the future. Part of what I want Measure the Future to do is help libraries tell better stories about themselves, to have data to back up the stories that they tell, and hopefully to have stories to tell that they didn't even know about. The stories that libraries tell about themselves have changed over the years, and I think they will continue to change and evolve even further over the next several years. If I want Measure the Future to be a part of telling those stories, maybe I need to tell its story first.

Measure the Future
http://measurethefuture.net

Setting the Stage

It was 2011 or maybe 2012. I was gathering statistics as part of my role as head of information technology at the library at the University of Tennessee at Chattanooga, and I was getting more and more fed up with numbers by the minute. The sorts of things that we tracked, according to our requirements for accreditation and for ACRL, were useful only to compare us to other institutions in ways that didn't seem particularly meaningful in the modern age. These numbers didn't tell me much about how to make our services and spaces better for our patrons. And even numbers that might guide us in better acquisitions practices were incredibly difficult to pull from the morass of database vendors.

I started thinking, trying to consider carefully what might be important for libraries to know, what might give us insights into how patrons were using libraries. The more I thought about what was likely to be important and what was also a huge gap in our knowledge, the more I was convinced that we needed lots more data about our building use. The library building is, in most cases, the library's most valuable fixed asset. The library building is a huge aspect of the library's net worth, and yet we don't focus our attention on how it's used in the same way we look at our materials. So how could we start to better understand how our buildings were being used?

Moreover, as collections shift from physical to digital, communicating the importance of the physical building to those who oversee the funding for libraries is a key for future growth.

That question was the central one around which I began to brainstorm ideas. It was immediately obvious to me that since we didn't have really any statistics about use of the building (other than gate counts of number of people who walked in), some thought about what to gather and how it might be gathered was the first order of business. I realized that what I wanted was a system that would tell me how people were using the space. Were they coming into the library, collapsing into a chair, and not moving for hours? Were they just using the building as a pass-through to another location? Were patrons using the stacks to browse for items at all? Did patrons use our big tables in groups, or was it just one person camping all day? All of these questions seemed valuable to try and sort through.

The next step was to try and see if there was a way to capture that data. Luckily, I had some experience with small electronics platforms, like the Arduino and the Raspberry Pi, and so I knew there were at least a dozen ways one could approach the problem. Measuring occupancy per room was part of the challenge, but we also needed to be able to parse what the people were doing in the space. Not individually, perhaps, but as a collective, what sorts of actions are people doing while they are in the library space? This makes door sensors and other point-of-contact sensors difficult from a logistical point of view. There are just too many points of contact to wire all of them in any given space, much less across different libraries.

A number of commercial entities at the time were using cellphone signals as a stand-in for "individual" movement. By using Wi-Fi or Bluetooth signals or both from mobile phones moving around a space, you can track where people are and what they are doing with a high degree of accuracy. Big box retailers and the like have used this technology for years, and it was starting to trickle into the cost range where libraries were beginning to play with Bluetooth beacons and other types of tracking technology. After looking at the options, I abandoned this idea for my project for several reasons. The largest is that it simply cannot assure privacy in a way that I was comfortable with implementing inside of a library. There are mechanisms for "anonymizing" data from mobile connectivity, but (especially at the time) I didn't feel that they were enough for me to be confident in protecting the identity of patrons in the library.

The privacy issue compounded with the somewhat obvious problem—if we track people with cell phones, we have information only about people with cell phones. This ignores many of the patrons of public libraries, like children, the homeless, recent immigrants, and more. Putting together a system for making decisions about library services and then ignoring swaths of the community that would likely benefit the most from library services did not seem like the wisest course of action. Between privacy issues and this selection bias issue, I took Wi-Fi and Bluetooth tracking as a method for our new tool off the table.

My thoughts turned to imaging. What was the possibility of using some kind of image sensor to capture the whole space at once and analyze how patrons were moving? If we could do this without actually taking pictures, just by capturing the location of people in the space without any identification, then it would pass the security test (more on that later). Images would also gather everyone equally, without bias, across the types of technology the patrons used. One possibility was using an infrared sensor that looked for body heat, but after putting together a quick demo using a standard webcam as a data source, I realized that it was possible to use computer vision to solve this problem without the added expense of the infrared camera.

Decisions and Solutions

Once there was a demo in place, the project applied to the 2015 Knight Foundation's News Challenge for Libraries, a grant round for funding ideas that would benefit libraries around the US.[1] The newly named Measure the Future project was one of eight winners of the News Challenge, which gave us funding for initial development of the project and supported the development through the fall of 2016.

Our first set of decisions revolved around which hardware to settle on. We needed a microcomputer, something that was capable of running some amount of computer vision locally on the sensor itself. This was due to an early decision that was made to start by having any processing of the location data done locally, on board the device itself. This was partially because it was slightly easier to build and could be realized faster. It was also done because we realized very early that in order to make libraries comfortable with installing cameras in their spaces, there had to be a good security story to tell. The best security story is that the data is collected locally, processed locally, never leaves your building, and doesn't include any information about your patrons—and so that's the tool we set out to build even though in some ways it was more difficult than other possible solutions.

The obvious answer for which microcomputer platform to use was the Raspberry Pi, the most popular small computer in the world. The only problem was that at the time, the current Raspberry Pi model (Model 2) didn't include wireless networking by default. In order to get Wi-Fi, you had to buy a separate USB Wi-Fi adapter and then hope that it was stable and ran well on the operating system—neither of which was an assumption I was willing to make. USB Wi-Fi dongles are notorious for their flakiness, and for a device that I was hoping to install in libraries around the country, I needed something far more reliable. We looked for a board that would run the software needed, that had Wi-Fi on board, and that was low-power enough to not need any sort of special attention paid to it over time. We found that in the Intel Edison and began development of the alpha units in 2015.

The other aspect of the project that is worth calling attention to is that it is being built using open source code, and all of the code that we have developed is also being released via an open source license on our

Github repository. We are using standard tools in the development and are sticking with standard web technologies for the user interface. Raspian, OpenCV, Go, Python, React.js, and the other tools used to build this project are well understood and openly supported, with no proprietary or controlled code that can cause issues with some vendor software. The data is in a standard JSON format, and libraries that implement Measure the Future have direct access to the sensors and software. They also own the data they collect; it isn't collected by Measure the Future without permission and request. We are dedicated to making these tools as widely available as possible in order to enable libraries everywhere to be able to test and implement them. Using open and easily available hardware, 3-D printable cases that are made available for reproduction, and open source code that is licensed for sharing and reuse is the best way to do this.

Measure the Future Github Repository
https://github.com/MeasureTheFuture

Design and Development

The next goal of the project was to develop software that would use a standard webcam attached to the Edison to act as a sensor and gather data points as people move through a space. Capturing the location and duration of movement for each recording individual as they move through a space and recording those data points to a database was the first order of business. Clinton Freeman, a developer located in Cairns, Australia, was recommended to me as someone with the technical background to be able to pull this off. Clinton had worked with both health care and libraries in the past and had a great grounding in the sort of privacy issues that arise from using cameras in public and how libraries, librarians, and patrons might react to them. Clinton understood from the beginning the sort of issues we needed to avoid and quickly became the primary developer of the project.

Measure the Future gathered information from two initial partners in the design stage of the project, the State University of New York at Potsdam library, directed by Jenica Rogers, and the Meridian Public Library in Idaho, directed by Gretchen Caserotti. Both were involved in early discussions that set the path for the project development and initial goals. Among other librarians who helped in the initial design phases, particularly in some of the key early thinking, Andromeda Yelton was invaluable. She helped in thinking hard about the privacy model we should follow and in the development of the early UI and UX models for the project.

Security

Several security principles arose from these early discussions. The first was that the alpha units would concentrate on gathering the information and acting as a distribution point for the gathered statistics with no central server architecture. The sensors wouldn't yet talk to a central server due to complexity and implementation difficulties in local libraries. Instead they would act as individual "islands" of data gathering, and libraries could query the individual sensors to see a current heat map of the space or to download the data for analysis. It was clear even in these early stages that the end game for the project needed to be a central visualization and data analysis server that would gather multiple sensors in multiple branches together in one interface. That complexity, however, was well beyond the minimum viable product stage, and we wanted to prove worth before we embarked on that much more involved and difficult process.

The second principle was linked to the privacy issues inherent in gathering data about patrons in a library. We decided that a standing goal would be to never gather any information that could be used to personally identify individuals. This approach complicates many aspects of the project, not the least of which is that as a result of this decision, we are forced into a corner with the way we interpret and can present data about patrons in the space. If the system can't tell Person 1 from Person 2, it has no way of determining if Person 1 enters and exits the area being measured. It simply says "oh look, another person," and counts Person 1 as another unique patron. This means that "patron counts" using Measure the Future are necessarily fuzzy, but the other options for dealing with the issue all led to the potential for patron identification, especially if multiple types of data for a given time period existed. So we made the conscious choice to make our data slightly less precise in service of being extra cautious about patron privacy. I think that's the correct call to make, although it is an incredibly common request from libraries I have spoken with about the project.

The way I describe our approach to security is that we are attempting to measure the space, not individual library users. We're dealing with aggregate movement data and anonymous individuals with no visual information stored for later analysis. We're not even saving the "blob size" information because that could theoretically be used to de-anonymize someone in specific circumstances. Instead, we store only the center location of the identified blob, reducing the ability to identify individuals. We store data in fifteen-minute "buckets" of data as well, in order to prevent identification attacks that rely on precise timing of individuals in spaces. This doesn't reduce the value of the aggregate data, nor even of the movement data; it just

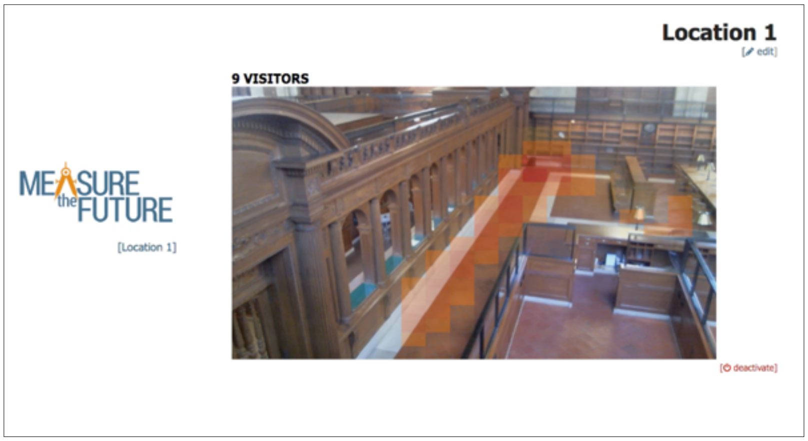

Figure 2.1
Early Measure the Future Interface

prevents precise identification of individual patrons.

Technically, we also ensure that the connections between the sensor and the device used by librarians to view and download data are secured via WEP2 and strong passwords, as well as strong passwords at the system level. It isn't exaggerating to say that we spent nearly as much time discussing and modeling our security plan as we did designing the rest of the system. Moreover, as we move into our more connected Beta development round, we will maintain this focus on security, even as we move to a more cloud-based data visualization and aggregation service.

How Measure the Future Works

Measure the Future works by using a webcam as a sensor for a computer vision system running on a microcomputer. The webcam is placed in a position such that it can have a vantage point to "watch" the space, which normally means as vertical and overhead as we can get. Most installations have been high and at an angle, not truly overhead, although more is better than less for the camera to be able to capture accurate data. The system is calibrated by taking a single reference image, preferably when the space is clear of people. Once calibrated, the sensor is switched into Measurement mode, where it is actively capturing data about movement through the space (see Figure 2.1).

Once per second, the system checks the image sensor in the camera and compares it to the calibration image. Areas that are different are analyzed for size, and if it fits within the settings boundaries, then the different area is identified as a computer vision "blob." Believe it or not, a blob is actually a technical term in computer vision work and designates a contiguous area of pixels that the system should keep track of, identify, or watch. The size of a blob is variable and can be adjusted in the settings panel in order to prevent either false positives (huge shadow moves across the room due to a window) or false negatives (missing people because the sensor is far away and they appear too small).

A blob is identified as soon as it enters the frame, and while it is in the frame, every second another data point is created that notes the location of the blob in X,Y coordinates that are mapped to the calibration image. Each data point is also timestamped with a duration of time. With the calibration image, coordinates, and timestamps, each blob can be tracked through the space in question. You can see how patrons move through the space, where they stop and linger, where they congregate, and where they never go. Over time, you can see what areas in your space are popular and what areas aren't used by patrons. You can query the data to tell you how many people stopped by the new book display and how long on average they spent there.

In the current release, the default display for librarians using the system is a cumulative heat map of the space with controls for calibration and for downloading the sensor data locally. The data is stored on the sensor in a relational database, but the download link on the interface provides easy-to-use JSON formatted files and the calibration image in a zip file. This gives the library all it would need to do whatever sort of data analysis it would like, from advanced heatmaps (see Figure 2.2) to patrons counts to specific location queries.

Sensor units can be installed in fixed locations, for gathering data over time about a specific space, or they can be moved in a more tactical process of measuring specific locations or programs for limited times. Measuring the usage of the library reading room is a great use case, but so is gathering data on a new book display to see how patrons are interacting with it. As the system develops, I hope to see libraries using it in ways that we never expected. That is, for me, the measure of an interesting technology project. As William Gibson famously wrote, "the street finds its own uses" for technology.[2]

Alpha Testing

For our alpha testing of the system, the project had the opportunity to be a part of the reopening of the Rose Reading Room in the New York Public Library in the fall of 2016. We really could have found no bigger stage, nor larger room, in which to try the first installation of the sensors. Six of our alpha sensors based on the Edison were installed in the fall of 2016 and were left to run over the course of the fall and winter.

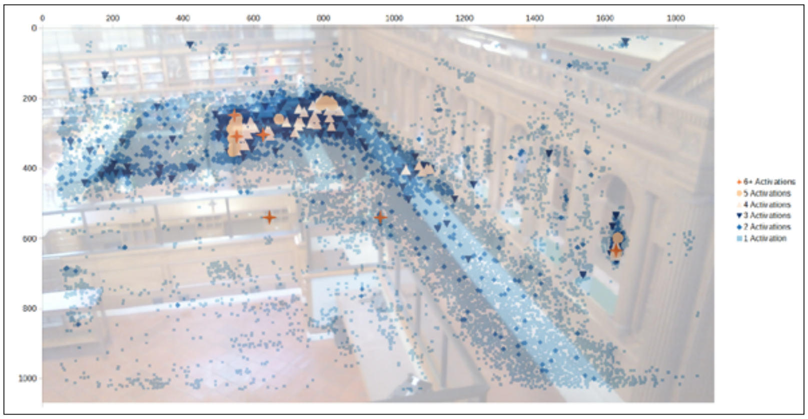

Figure 2.2
Detailed Heatmap of Measure the Future Data

The installation was a bit overkill for the rooms, in that we could have covered the same space with fewer sensors, but we were being careful and ensuring we'd have some fallback if we found issues with hardware or software—you can never be too careful with alpha systems. Two sensors were placed in the Bill Blass catalog room, and two each in Rose Reading Room North and Rose Reading Room South (see Figure 2.3).

It was apparent quickly that there were issues with the Edison platform. Initial testing had been done in very limited traffic areas, and when the Edison attempted to keep up with the traffic in one of the busiest library rooms in the country, and during the single busiest period, the sheer volume of computation needed swamped the microcomputer and caused every type of computing issue possible. Over the course of the first few months, we saw I/O throughput errors, disk errors, and in one case the processor on one of the Edisons overheated. We worked our way through many of the issues and began digging into the data to try and do some more focused data analysis. That's when we found the most interesting bug of our alpha testing.

Perhaps obviously, the data that we were gathering depended on having accurate timestamps. The Intel Edison, however, doesn't have an onboard clock for keeping time separately from being on a network. This isn't unusual among microcomputers these days; the Raspberry Pi has the same limitation. But this meant that we needed a way to set the time on the sensors that didn't rely on them having access to the internet. Remember, these were never going to connect to the wider internet once installed; they were going to connect directly to a laptop or tablet that the librarians were using to monitor and download information. Our solution, which is the same used in another open source project I run called LibraryBox, is to scrape the time from the browser during the calibration step. When the initial connection to a laptop is made, each sensor would check the time the browser had and set the time on the board accordingly.

This seemed like a good solution to the issue, and in testing it seemed to work beautifully. We could set up a new sensor, start collecting data, download the data, and the timestamps were all correct. When we did the initial setup of the sensors in NYPL, we calibrated and tested the units and started collecting data, checked the data, and everything looked great. NYPL staff collected data over the next few days, and again in checking the data for dates (downloading and checking the beginning of the file and then scrolling to the end to compare timestamps), everything looked great—until, of course, we started doing visualizations. When we put the data into a visualization, the timestamps didn't make any sense at all, and so we dug in to see what was going on.

What we discovered was one of the strangest bugs that I've dealt with in my time building hardware like this. The sensors had, it turns out, been turned off at night with the lights in the room—they were on the same circuit, and when the master for the room was turned off, so were the sensors. They then came back on when the lights were turned on in the morning and began recording data again. But because they had been power-cycled, they no longer knew the correct time and so timestamped beginning with Linux start time (January 1, 1970)—until, of course, someone connected to the sensor, at which point the system took the browser time and began applying it, so that if you looked at the last several hundred data points, they would be timestamped correctly. This was a data bug that existed only when you weren't looking.

It became apparent that part of our troubleshooting of all of our alpha issues would have to be a careful analysis of the platform we had chosen. The Edison had fallen down on the processing side of things, and even with refined computer vision techniques, it was likely that we would run into other hardware issues. Meanwhile the Raspberry Pi foundation had announced the Model 3 version of its hardware in early 2016, and by fall they were finally becoming available for purchase. The Raspberry Pi Model 3 dealt with a lot of the issues that had caused us to decide against it early in our development, primarily by putting Wi-Fi onboard rather than relying on external adapters. With more processing power, more storage, and onboard Wi-Fi, the Raspberry Pi Model 3 seemed like the answer to our issues—except that we'd have to start almost from scratch in porting code from one platform to the other.

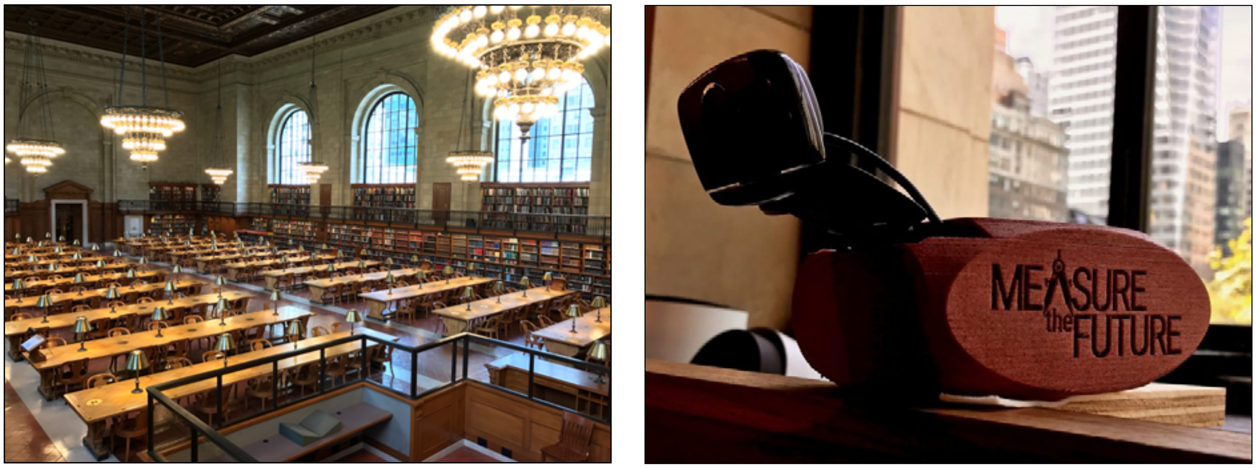

Figure 2.3
Measure the Future sensors at NYPL

After evaluating options, it became obvious that moving to the Raspberry Pi–based hardware configuration was indeed the best option, and so began the development of the Measure the Future Beta program.

Beta

Through the spring and summer of 2017, we focused on moving everything to the new hardware while ensuring that we solved the problems that were identified in the alpha testing. We solved the lack of a clock for accurate timestamping by adding one physically to the Raspberry Pi. One of the advantages of the platform is that it is so popular that it has a huge variety of additional components that can be added to the base model. Adding a battery-powered real-time clock gives us confirmed timestamps for all data collected, with no concerns about power cycling or other service interruptions. By late summer, we had tested our new sensor units and confirmed that they were ready for testing in the real world.

Enter our new beta partners, the libraries at the University of Rochester in Rochester, New York; the Carnegie Library in Pittsburgh, Pennsylvania; and the Boston University Law Library. They will join NYPL, Meridian, and SUNY Potsdam as testbeds for our beta hardware, which is rolling out over the course of the fall of 2017. In addition to the updated hardware, the beta development will continue on the software side, pushing toward the launch of the cloud-based visualization and analysis tool. This new visualization tool is needed for multiple reasons, most of which boil down to user experience and system capabilities.

For libraries with multiple sensors, having all of the data in a single place and interface is clearly a better experience. In addition, we want to be able to cross-reference sensor-to-sensor data and generally have a more holistic look at building usage, rather than individual room usage, as quickly as we can. There are also visualizations and analysis of the data that we simply can't do on the sensor unit itself. The Raspberry Pi is a big step up from the Edison, but it doesn't compare in processing power to a cloud-based server where we can throw almost unlimited amounts of processing power at a particular set of data. The data we're collecting grows pretty quickly, as you can imagine. Every second we're capturing the position and timestamp for everyone in the room, all day long. Over months and months, the only reasonable way to handle that much data and deal with it all at once is to put it onto a proper server and have much more powerful processors deal with it.

With more power to throw at the data, especially longitudinal data over months and eventually years, we hope to be able to surface patterns of use that would be invisible via other data collection methods. Our beta partners will be the first to see the power of that data, and over the next six months, we will be developing the next stage for Measure the Future.

Conclusion

At the time of writing, Measure the Future has one beta site live and is running on the latest iteration of our sensor hardware, with two more location scheduled to go live in the next two weeks and another two in the following month. By the end of 2017, we should have our latest hardware in all six of our partner libraries, all of them collecting data locally. Early in 2018, we will begin moving those that wish from local data collection and visualization over to our cloud service. It's possible that not all beta sites will want to share their data remotely in any way, which is totally understandable. If they wish to implement a local instance of the Measure the Future cloud, they will be able to do that because of our open source nature. I believe, however, that our security model will be such that most libraries will choose to share their data with the project through our cloud portal.

At that point, the goal will be to look for patterns of similarity and difference between libraries. Identifying patterns across libraries is something that I believe could be incredibly useful, especially for space planning for renovations and new library buildings. Ultimately, our hope is that the data leads to libraries being able to understand how their patrons want to use their spaces, allows for iterative testing of spaces to make them ever better for their local communities, and gives libraries the information they need to tell the stories needed to ensure their continued funding.

Notes

1. John Bracken, “Knights News Challenge: Libraries Closes Sept. 30,” September 10, 2014, Knight Foundation, https://www.knightfoundation.org/articles/bracken-knight-news-challenge-libraries-offers-25-million-innovative-ideas.
2. William Gibson, *Burning Chrome* (New York: Ace Books, 1982).