360 Time-Lapse Camera

Brian Bock
bbock@terpmail.umd.edu

ABSTRACT
The aim of this project is to build an omnidirectional camera capable of recording spherical footage. This Raspberry Pi based system will be used to take 360 time-lapses, which will be viewable via Google Cardboard and Youtube's 360 video viewer. Several wide-angle cameras will be connected to equally many Raspberry Pi Zeros, which will take pictures at regular intervals. These photosets will be stitched into the proper format by an additional Raspberry Pi, and then exported as a 360 video. Various methodologies and related projects are evaluated. The history of panoramic photography is explored, leading up to the invention of several omnidirectional cameras and 360 viewers.

INTRODUCTION
Unlike regular cameras, which capture a scene from a static, fixed point of view, 360 (also known as omnidirectional) cameras allow you to capture a scene from every angle at one point. This immersive technology puts you in the moment, and lets you change perspectives as you would by turning your head. The entire scene is available to be viewed. I want to build a 360 camera and use it to take time-lapses. Ultimately, these 360 footages could be paired with a VR headset, such as Google Cardboard, giving you the freedom to look around as a day passes by.
I do not yet have a finalized design for this project, and there are still many different design aspects that need to be considered. The current plan is to use several Raspberry Pi cameras connected to either a Raspberry Pi and multiplexer or multiple Pi Zeros. The cameras would be positioned to cover as much of a spherical field of view as possible. For time-lapses, each camera would take a photo simultaneously. These photos would then be stitched together into equirectangular format and saved as a frame of the time-lapse. This process would repeat at regular intervals (such as once per minute) for several hours. At the end, the equirectangular frames would be combined into an equirectangular video, which could then be viewed as a 360-spherical video.

PROJECT DESCRIPTION
The current plan is to have four wide angle cameras individually connected to Pi Zeros. With a viewing angle of 160 degrees, 4 cameras should be sufficient to cover a full spherical view with sufficient overlap for stitching. However, if testing reveals a gap in the footage, an additional camera will be added. Each of these Pi Zeros will be connected via USB to a "MasterPi", which will handle timing and communication. The MasterPi will be a Raspberry Pi 3 with built in Wifi connectivity, and serve as the point of contact with the outside world. The MasterPi will command the Zeros to take photos in sync; this collection of photos will be known as the photoset. Photosets will be taken at regular intervals for several hours.

After each photoset is taken it will need to be stitched into equirectangular format. This process could be handled by one of several programs, including Hugin, AutoStitch, and Photoshop. Normally, these programs analyze the photos, find matching regions to combine, and create a homography matrix for the transformation, which can be time consuming and processor intensive. In this approach, photosets would need to be offloaded from all the Pis, sorted, and then stitched together externally. In addition to being a very tedious and time consuming effort, this approach might not be easily automatable.

Fortunately, once I finalize the configuration of the cameras, each photoset will require the exact same transformations to turn it into an equirectangular frame. The homography matrix will not need to be recalculated with each frame, and I should be able to "blindly" reapply it to each photoset, dramatically reducing processing time. This looks like it can be done with several python scripts (Rosebrock, 2016). Ideally, the MasterPi will take in the photos from all the Zeros and apply the homography transformations while it would otherwise be idle. At a frequency of 1 photoset per minute, all the Pis will be dormant for at least 50 seconds/minute. After the photo-shoot, the MasterPi would have a folder of equirectangular frames, which could then be easily imported into Adobe Premiere and turned into a time-lapse.

HISTORY

While most of the history of omnidirectional photography is extremely recent, its roots can be traced back as far as the mid 1800's. Per the Library of Congress's Brief History of Panoramic Photography, panoramic photographs have existed since as early as 1851, and the first mass produced panoramic camera was introduced in 1898 (LoC, 2016). Panorama comes from the Greek roots *"pan",* and "*horama*", meaning "all seeing". Over the next 100 years, a variety of different panoramic cameras and techniques were invented (McBride). For the sake of brevity, these innovations will not be recounted here. They are the foundation upon which modern panoramic cameras are based, but their value past that is purely historic.

In 2012, Apple's iOS 6 came with panoramic support for the first time (Esposito, 2012). While there were already many existing iPhone panorama solutions, this was a major step towards the democratization of this technology. There are now an even greater variety of panorama taking and stitching applications as well as 360 cameras. VirtualRealityTimes has a list of popular commercial 360 cameras (Oscillada, 2016). In March 2015, Youtube began support for 360 videos (Lowensohn, 2015). Facebook also announced integration of 360 videos at around the same time (Etherington, 2015). The sudden surge in prevalence of this video format likely led to its increased interest in the public eye, which then encouraged more users to get involved. NYTimes has subsequently joined the 360/VR world, and now have a mobile app that lets you watch 360 stories in an incredibly immersive way (Watercutter, 2015)

When I first started thinking about this project late last spring, I had difficulty finding similar projects online. Many of the 360 cameras I saw were incredibly expensive and proprietary. Subsequently there seems to have been significant progress in this field. James Mitchell's Zero360 project, started in late June has a lot of similarities to the project I have in mind. I anticipate this project will be a valuable resource to me as a starting point, but I hope to take the idea further. Zero360 can only take cylindrical pictures and videos, basically full wrap around panoramas (Mitchell, 2016). I want to be able to take spherical pictures and videos, capturing a full 360x360 degree field of view.

In August, Tinkernut released this Raspberry Pi based 360 camera, which utilizes a spherical mirror to take cylindrical footage (Klosowski, 2016). While this project poses a creative solution to the "large number of cameras" problem I have been facing, the result is far lower quality than I would like.

SIGNIFICANCE AND DISTINCTION

From greyscale to color; from still to video; every pictographic innovation has been made with the goal of a more realistic and immersive experience. 360 footage, is in my opinion, the pinnacle of immersive innovation, giving photographers the new ability to embed their viewers directly in the experience they are trying to capture.
Time-lapse photography is special to me because it is a way for me to show people the world they know in a way they've never seen it before. We all have an internal understanding of time, but when we move through time at the same rate it passes, much of its impact is lost. Only by seeing vast amounts of time move by quickly can we begin to appreciate the time we have.

As mentioned in the history section, Mitchell's Zero360 and Tinkernut's projects are similar in principle to what I have in mind, but there are several important distinctions. Both projects can take only cylindrical footage, while I strive to capture full spherical footage. Tinkernut's spherical mirror design (similar in nature to a catadioptric camera system), allows one camera to capture a full cylindrical view. My design, by contrast, involves at least four cameras. This makes an important difference in quality. Even with the same resolution cameras, my project will have four times the pixel density and image quality of Tinkernut's. That said, I am in no way trying to diminish or belittle the works of my predecessors. In fact, without their progress to work off of, this project would be much lengthier and more difficult.

EXPERTISE AND SKILLS

I am already skilled with 3D design and printing, which will be very useful when I want to create a case for all the cameras. I also have extensive experience with photography, especially time-lapse photography.

This project will stretch my programming knowledge and skill. While I have some background in a few languages, I am not a very strong programmer. For basic operation of the Pi(s), I need a basic working knowledge of Unix command line and Python. At this point I believe I know enough of each to get by and can learn more as needed. I will want to build an interface to control and configure the cameras for each time-lapse. I'm not yet sure if this will be physical or web based. If it is web based, I'll need to brush up on my HTML and JavaScript. If, instead, I build a physical control panel, it will likely communicate with the Pi's through a combination of Bash and Python.
Several major steps in this project look like they will require more programing prowess than I currently possess. I am, however, hopeful that the resources available to me online and through my peers will reduce the difficulty of these steps.

After the photos are taken they will need to be stitched into equirectangular format. This process will likely be handled by one of several programs, including Hugin, AutoStitch, and Photoshop. Normally, these programs analyze the photos, find matching regions to combine, and create a homography matrix for the transformation, which can be time consuming and processor intensive. Once I finalize the configuration of the cameras, each photo set will require the exact same transformations to turn it into an equirectangular frame. I'm hoping that I'll then be able to "blindly" apply these transformations to dramatically reduce processing time. The homography matrix will not need to be recalculated with each frame. I initially thought that creating a fixed photo stitching application would be too far out of my skillset to be a realistic expectation for this project. However, I subsequently found tutorials on how to do this in Python (Rosebrock, 2016).

APPROACH

Much of this project will be tangible and hands on as I work to build a physical camera. I anticipate there will be a lot of trial and error while I refine my design. I will be drawing extensively from my own experience and learning "on the job" from online tutorials and the work of my predecessors. I was thrilled to discover that as I learn more about this topic, I am better able to research relevant topics and find more applicable information. I eagerly anticipate learning much more as I progress through this project.

WORK PLAN AND TIMELINE

The major steps of this project are outlined below. Steps in bold are expected to be significant challenges and require considerable time and thought. It is important to note that while these steps are listed in rough order they need to be completed, not all of them require the previous step. As such, some of the times listed can and will overlap. To get a head start on the Spring 2017 semester, this project will likely be started over Winter break, giving me more time for more careful progress.

Major Steps:

1. Order parts: 1-2 weeks + 1-2 weeks for reordered parts. From other projects, I already have a Raspberry Pi, Pi Zero, and Pi Cam, which can be used for initial testing and configurations while the actual parts ship.

2. **Make the Zeros talk with the MasterPi – 2-4 weeks.** I expect this will be simultaneously one of the most crucial and most difficult steps and will require the most learning outside of my existing skills.

    - **Communication over USB**

        - **One way communication from MasterPi to Zero**

        - One way communication from MasterPi to Zeros (plural)

            - Each Zero should have a unique name/identifier

        - **Two-way communication between MasterPi and Zero**

            - **MasterPi must be able to receive pictures from Zero. Zero should act as a drive(?)**

        - Two-way communication between MasterPi and Zeros (plural)

            - MasterPi must be able to receive pictures from each Zero. Each zero should act as a drive (?)

    - Determine if the MasterPi can power all the Zeros or if they must be powered individually – 1 day

3. Determine optimal camera configuration – 1 week

4. **Design and print case to keep cameras and Pis in place.** I already have the skills that this step requires, but I know from experience that this takes considerable time and careful thought as well as multiple prototypes.
    - Case for each camera with room for lens and cable - 2 weeks
    - **Larger structure to lock each camera in the proper orientation (camera holder) – 2 weeks**
    - Case for each Zero with room for the power and camera cables – 2 weeks
    - Case for the MasterPi with power and USB ports accessible. Needs room for CPU heatsink. 2 weeks
    - Case for portable power supply
    - Enclosure for entire system – can just be a way to **interlock** the above pieces
5. **Calculate homography matrix and program MasterPi to apply it – 2-3 weeks** Similarly to step 2, this step is crucial and outside of my existing skills. I'm hoping that online tutorials will make this less of a challenge than it initially seems.
    - Determine if MasterPi can stitch photosets faster than they are taken – this will be determined through experimental data
        - What's the fastest photosets can be taken before the MasterPi can't keep up?
        - Calculate required "idle time" post photoshoot to finish processing
    - Determine if MasterPi can handle homography transformations or if stitching must be external
    - Refine transformations as needed
    - Are other transformations or edits necessary for smooth equirectangular frames? Can the MasterPi handle them? If not, can they be applied externally in bulk?
6. Control interface for MasterPi
    - Web based – 1 week
        - Can label photoshoot
        - Repurpose PiCam control page. From another project, I have a web based interface to control a network connected Raspberry Pi and camera. I don't think it will be particularly difficult to recycle this page for this project.
    - Physical – 2 days
        - Connect LCD panel to MasterPi

    - Will it be necessary to ‘lock’ the control panel so that passersby cannot interfere?
  - Both(?)
  - Configure MasterPi accept parameters from control interface
    - Photoset interval
    - Photoshoot duration
  - If possible, control panel should monitor battery life (or estimate remaining based on known average battery life and running time)

7. Real world tests – 2-3 weeks
   - Determine expected battery life
   - Revise design as necessary
8. Real world use – 1-2 weeks, weather permitting
   - Take 3+ outdoor 360 time-lapses
9. Make 360 videos viewable on Google Cardboard - <1 day
10. BONUS (Time permitting)
    - Capture 360 video
      - Stitch equirectangular videos; will likely need to be handled externally

Jan 2-10

Order parts

Begin experimenting with interPi communication on existing parts

Jan 11-20

Continue working on interPi communication

Start working on case for camera module

Jan 21-30

Continue working on interPi communication

Print camera module case V1. Revise design as necessary

Jan 31-Feb 10

Begin work on two way interPi communication

Modify camera module case.

Begin working on Zero case.

Feb 11-20

Continue working on two way InterPi communication

Finalize camera configuration

Begin working on camera holder

Feb 21-28

Continue working on Zero case

Continue working on camera holder

Print camera holder. Revise design as necessary

March 1-10

Start working on homography matrix and transformations

Begin working on MasterPi case

Begin modifying PiCam interface for this project

March 11-19

Continue working on homography matrix and transformations

Continue modifying PiCam interface

Install and configure physical interface

Print MasterPi case. Revise design as necessary

Begin working on system enclosure

March 19-26 – Spring Break

March 27-April 10

Finalize automated homography transformations

Modify any 3D printed cases, as necessary

Print full system enclosure. Refine design as necessary

Assemble full system

Iron out any issues

April 11-20

Real world testing

April 21-30

Real world testing – time-lapse

May 1-10

Real world use – time-lapse

Port videos to Google Cardboard

Prepare for Capstone fair

AUDIENCE

There are several target audiences for this project. The DIY community revolves around the idea of building things yourself and creating cheap/creative alternatives to professional solutions. This project is not targeted at the professional user with high-end equipment and a large budget, but rather individuals who want to join the world of 360 capture with an entry-level system. I don't expect the result of this project to be able to compete with high-end 360 systems. Thus, I will not be marketing this project for profit. Instead, I will document my construction process carefully and make the code open source, allowing anyone to replicate or expand upon this project under the Creative Commons license. The goal of this project is to make omnidirectional camera technology widely and easily accessible by reducing the barriers to entry.

BUDGET

The budget for this project will be largely dependent on its final design. You can read some of my thoughts about different designs at http://dcc.umd.edu/portfolio/bbock/. I will layout this budget with several different configurations for comparison. This will necessarily include technical explanations of the pros and cons of each idea. I apologize if this section becomes more technical than it was originally intended to be, but I feel it is the only way to properly do it justice.

This project requires several cameras. At this point, it is unclear how many cameras will be needed, or their optimal configuration. Most likely I will utilize fisheye cameras, such as SainSmart's (http://www.sainsmart.com/sainsmart-wide-angle-fish-eye-camera-lenses-for-raspberry-pi-arduino.html). This camera costs $22 when shipping is accounted for. Each camera must be connected to a Pi, and this is where my designs begin to differ. Natively, each Raspberry Pi only supports one camera, but there are 3rd party add-ons that allow for additional connections. Two such examples can be found at https://hackaday.io/project/2847-ivport-raspberry-pi-camera-module-multiplexer and https://www.amazon.com/Arducam-Camera-Adapter-Compatible-Raspberry/dp/B012UQWOOQ?ie=UTF8&*Version*=1&*entries*=0. It's not yet clear if these multiplexers will be a cost-effective solution. While they allow more cameras to be connected to each Pi, they might only be compatible with the Pi models A, B, and B+, which range from $20-$40. I'll have to do more research to see if they are compatible with the $5 Pi Zero. If I don't use one or more of these multiplexers, I'll need to connect each camera to an individual Pi. With this setup, it will make more sense to use several $5 Pi Zeros. The Pi Zero has a smaller camera connector than the regular Pis, so a different cable is required to connect the camera. I'm building this budget table with liberal and conservative estimates for the quantities of each part required. As previously mentioned, the quantities are entirely dependent on the final design, which still requires more thought. Additional items that can yet be accurately budgeted for include:

- A portable power supply (Dependent on the power requirements of the full setup)
- Material Costs for a 3D Printed case (Dependent on final design, will likely require multiple revisions and prototypes). While I can print for free in the DCC lab, my printing requirements may necessitate that I print at McKeldin or Terrapin Works, where I am charged by the amount of material used. It is very difficult at this point to predict how large my prints will be but I'll estimate $5-10 per prototype. There will likely be multiple prototypes and revisions as my design evolves and improves through testing. To get precise measurements of the individual components for my designs I will utilize digital calipers.

As I am working with relatively fragile components, it is likely that one or more of them will break during the prototyping, construction, or testing of this project. The budget will therefore include extra to allow for material defects and damages.

Will I want this project to have network connectivity? This added functionality has some merit, but adds $13/P. Will all the Pis need Internet connectivity or can I get away with one that serves as a relay between the world and my Pi? I will want to build an interface to control this setup, but I'm not yet sure if that interface should be physical or web based. A web based control would probably be hosted for free on Heroku, while a physical control will require additional parts with cost. Please see the attached Budget spreadsheet for my detailed numerical budget.

OUTCOMES

I believe this project to be an amazing synergetic fusion of my passions, skills, and major as a photographer, and an engineer.

I hope that this project will inspire others to join the incredible new world of omnidirectional recording, and that my work can serve as the foundation for others to build even greater projects. As previously mentioned, I intend to document my process as carefully as possible and make my work readily available for those who are inspired by it and wish to take it further. This project does not end with its physical completion. After this system is finalized, it can and will be used repeatedly to capture beauty in a unique way. This project may also evolve after its completion – I hope to be able expand its functionality to include 360 video and possibly even live steamed 360 video.

BIBLIOGRAPHY

Materials
http://www.sainsmart.com/sainsmart-wide-angle-fish-eye-camera-lenses-for-raspberry-pi-arduino.html
https://hackaday.io/project/2847-ivport-raspberry-pi-camera-module-multiplexer
https://www.amazon.com/Arducam-Camera-Adapter-Compatible-Raspberry/dp/B012UQWOOQ?ie=UTF8&*Version*=1&*entries*=0
https://www.adafruit.com/products/3055
https://www.adafruit.com/product/3157
https://www.adafruit.com/products/2822
https://redbear.cc/product/iot-phat.html
https://www.adafruit.com/products/1099
https://www.adafruit.com/product/1110
https://www.amazon.com/Anker-PowerCore-20100-Capacity-Technology/dp/B00X5RV14Y/ref=sr_1_3?s=wireless&ie=UTF8&qid=1479582903&sr=1-3&keywords=20000+mah+portable+charger
https://www.adafruit.com/product/3055
https://www.adafruit.com/products/3082
https://www.amazon.com/Sabrent-6-Pack-Premium-Cables-CB-UM63/dp/B011KLFERG/ref=sr_1_15?s=pc&ie=UTF8&qid=1479585111&sr=1-15&keywords=micro+usb
https://www.amazon.com/SanDisk-microSDHC-Standard-Packaging-SDSQUNC-032G-GN6MA/dp/B010Q57T02/ref=sr_1_1?s=pc&ie=UTF8&qid=1479585280&sr=1-1&keywords=micro+sd
https://www.amazon.com/Neiko-01407A-Electronic-Digital-Caliper/dp/B000GSLKIW/ref=sr_1_3?ie=UTF8&qid=1479694073&sr=8-3&keywords=digital+calipers

Sources

Esposito, Dom. "New Feature In IOS 6: How To Use Panorama Camera Mode." AppAdvice. AppAdvice, n.d. Web. 15 Nov. 2016. <http://appadvice.com/appnn/2012/09/new-feature-in-ios-6-how-to-use-panorama-camera-mode>.

Etherington, Darrell. "Facebook To Support Spherical Video In News Feed And Oculus." TechCrunch. TechCrunch, 25 Mar. 2015. Web. 15 Nov. 2016. <https://techcrunch.com/2015/03/25/facebook-to-support-spherical-video-in-news-feed-and-oculus/>.

Klosowski, Thorin. "Make a Cheap 360 Video Camera Using a Raspberry Pi." Lifehacker. Lifehacker, 07 Oct. 2016. Web. 15 Nov. 2016. <http://lifehacker.com/make-a-cheap-360-video-camera-using-a-raspberry-pi-1787535629/>

LoC. "A Brief History of Panoramic Photography." The Library of Congress. The Library of Congress, n.d. Web. 15 Nov. 2016. <https://www.loc.gov/collections/panoramic-photographs/articles-and-essays/a-brief-history-of-panoramic-photography/>.

"Spinner 360° 35mm Panoramic Camera - Microsite - Lomography – History." Spinner 360° 35mm Panoramic Camera - Microsite - Lomography – History. Lomography, n.d. Web. 15 Nov. 2016. <https://microsites.lomography.com/spinner-360/history>.

Lowensohn, Josh. "YouTube Now Supports 360-degree Videos." The Verge. The Verge, 13 Mar. 2015. Web. 15 Nov. 2016. <http://www.theverge.com/2015/3/13/8203173/youtube-now-supports-360-degree-videos>.

McBride, Bill. "A Timeline of Panoramic Cameras." A Timeline of Panoramic Cameras. Panoramic Photo, n.d. Web. 15 Nov. 2016. <http://www.panoramicphoto.com/timeline.htm>.

Mitchell, James. "Introducing Zero360." Raspberry Jam Berlin. Raspberry Jam Berlin, 05 July 2016. Web. 15 Nov. 2016. <http://raspberryjamberlin.de/introducing-zero360/>.

Oscillada, John Marco. "List Of Consumer 360-Degree Cameras." Virtual Reality Times, 05 July 2016. Web. 15 Nov. 2016. <http://www.virtualrealitytimes.com/2016/07/05/consumer-360-degree-cameras-list/>

Rosebrock, Adrian. "Real-time Panorama and Image Stitching with OpenCV - PyImageSearch." PyImageSearch. PyImageSearch, 22 Jan. 2016. Web. 15 Nov. 2016. <http://www.pyimagesearch.com/2016/01/25/real-time-panorama-and-image-stitching-with-opencv/>.

Rosebrock, Adrian. "OpenCV Panorama Stitching - PyImageSearch." PyImageSearch. PyImageSearch, 08 Jan. 2016. Web. 15 Nov. 2016. <http://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/>.

Watercutter, Angela. "How The NY Times Is Sparking the VR Journalism Revolution." Wired.com. Conde Nast Digital, 27 Apr. 2015. Web. 15 Nov. 2016. <https://www.wired.com/2015/04/virtual-reality-journalism-nyt-mag/>.