Tracking a System of Multiple Cameras on a Rotating Spherical Robot

James Hasbany, Brian P. DeJong, Ernur Karadogan, Kumar Yelamarthi, Jonathan M. Smith
School of Engineering & Technology
Central Michigan University
Mount Pleasant, MI, United States

***Abstract*—For many applications in the operation of a spherical robot, it is necessary to use optical devices to observe, track and monitor the robot's surroundings and environment. Estimating the rotation of a multiple camera system is crucial and can also be complex and computationally expensive. Demonstrating the multiple camera system with a shared central point can be simpler and require less computation. In this paper, we will demonstrate that a multiple camera system can be developed, an estimation of the movement of a wheel and sphere can be tracked, and the movement can be observed from a remote location.**

***Keywords*—spherical robots; camera fusion; robotics**

I. Introduction

Recently, there has been a large growth in robotic and autonomous systems such as mobile robots, drones, and teleoperation [1]. The largest investments seen in robotic research and development is that of government spending with an attempt to stay ahead of the curve in technological advancements [2,3].

This growth has also sparked an interest in that of spherical robots. Notable applications for spherical robots are in aerospace exploration, patrol, agriculture, and the military due to the robots' holonomic nature, internally sealed mechanisms, ability to navigate obstacles, and robustness from collisions [4]. For example, spherical robots have been used in aerospace reconnaissance, data collection, material collection, and logistics of cargo [5,6]. The use of spherical robots in agriculture allows for an enhanced ability to track and treat crops, irrigation, and pests through both data collection and real-time observation [7]. Spherical robots also have application in military reconnaissance, espionage, transportation of cargo, and the precise and accurate delivery of payloads [4,9]. In Fig. 1, four different spherical robots are demonstrated.

Largely, the benefit of a spherical robot is in its encapsulation of all motor functions, computers, and control mechanisms. This encapsulation of the motors, computers and control of the robot allows for the robot to operate in extreme environments, rebound easily from collisions without damage, transition smoothly between different terrains, and allows for inherent consistency in the movement, tracking, and sensory perception of the robot [9].

Fig. 1. Four separate applications of a spherical robot [5-8].

Many spherical robots are single-pendulum designs where a single hanging mass is rotated about a central axis to change the robot's center of mass and induce rolling (e.g., [10]). The distinguishing benefits of having a single-pendulum spherical robot is the single axis of rotation, which accommodates simpler mounting, and the ability to load a majority of the spherical robot's weight onto the single pendulum, allowing for significantly higher velocity capabilities during translation of the robot. The downside of the single-pendulum spherical robot is its lack of omnidirectionality, which hinders a spherical robot's ability to navigate obstacles. Due to this limitation in mobility of the single-pendulum spherical robot design, investigation of multi-mass spherical robots has risen.

Multi-mass spherical robots use four or more masses in varying positions about the center of the sphere to allow for omnidirectionality (e.g., [11]). This enhanced mobility allows the robot to manage rough terrain and navigate clutters of obstacles more easily than a single-pendulum design. A limitation of the omnidirectional design is that the spherical robot has no axle upon which to mount forward-facing sensors, such as video cameras. (For example, note the two side-mounted cameras in the bottom-right picture of Fig. 1 – such configuration is not possible with an omnidirectional

978-1-5090-3202-0/17/$31.00 ©2017 IEEE

sphere.) Since the robot can roll in any direction, there is no location on the sphere upon which a sensor is guaranteed to face forward.

To overcome this, this paper explores the possibility of using multiple inexpensive cameras (video or image) distributed around the sphere. The robot can then intelligently pick at any given time the most-forward-facing camera to display to the user. As the sphere rolls, the images can be visually rotated in the user's display to maintain the proper field-of-view alignment. Such an implementation will require the tracking of the cameras as well as the manipulation of the picture feed as it is retrieved.

The multi-mass spherical robot in question is a design with four pendulums centrally mounted in tetrahedral formation (Fig. 2; [11]). This design allows for each pendulum to swing in full rotation without collision with each other and without an excessive loss in the rate of travel capable by the spherical robot. The four-pendulum spherical robot design has the greatest speeds without the sacrifice of mobility in regards to multi-pendulum spherical robots.

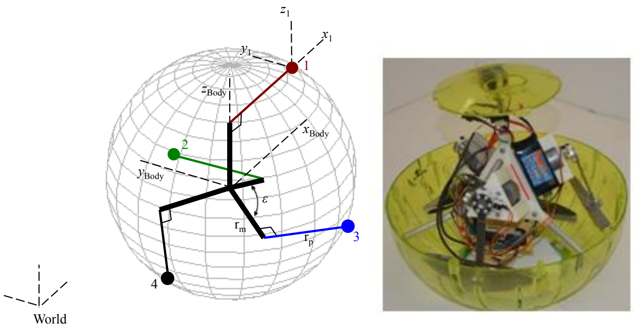

Fig. 2. Sketch and picture of the four-pendulum design [11].

The following sections discuss this camera fusion and present a preliminary prototype that achieves the fusion of cameras. A basic review of the previous design and research conducted on the four-pendulum system will be discussed. An understanding of the rotation of the spherical robot will be built. An overview of how the fusing and manipulation of the cameras will be discussed and demonstrated. A brief overview of the design of the camera system will be discussed. The computer as well as the communication process will be discussed and demonstrated. The mathematical tracking of cameras will be demonstrated with the use of encoders on the motor, image projection, and rotation matrices. The results will be demonstrated. Finally, the conclusion and future work will be discussed.

II. Design

In order to test the feasibility of the cameras, three cameras were mounted about a wheel on a structural frame, with onboard motor, battery, and Raspberry Pi 2 computer (see Fig. 3). The cameras used were USB cameras (30 FPS, 50Hz, 4P 2M lens F2.8). The wheel was rotated by a DC gear motor (200rpm, 170 oz.-in.) with an attached encoder (3200 CPR) identical to the motor operated within the spherical robot.

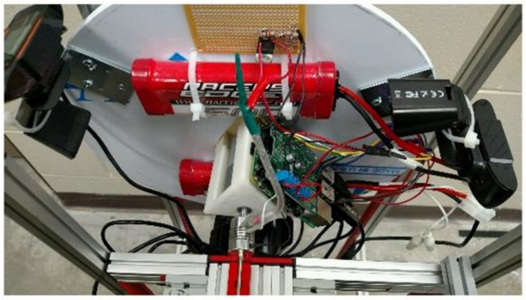

Fig. 3. Picture of cameras and mount

The motor was powered and controlled through a motor controller board that was managed via GPIO pins of the onboard computer. The motor controller received data via the encoder of the motor, which was then received and interpreted on the onboard computer. The interpreted encoder data was then used to estimate the rotation of the wheel mounted about the motor, which is later saved with each picture captured. This estimated rotation was then used to select the correct camera to use (see Fig. 4). In order to successfully use off-the-shelf webcams, each camera must be activated and then deactivated between the selection process. Only one camera can be activated at any given time.

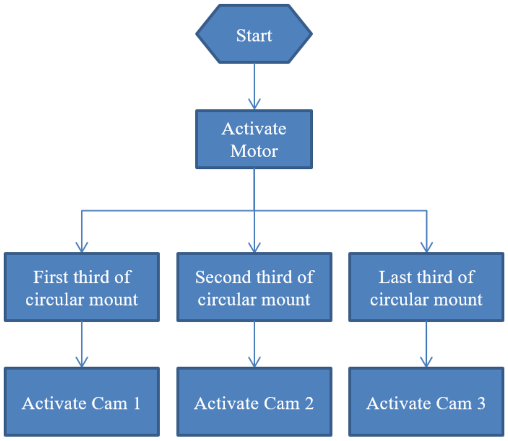
Start
Activate Motor
First third of circular mount
Second third of circular mount
Last third of circular mount
Activate Cam 1
Activate Cam 2
Activate Cam 3

Fig. 4. Flowchart of onboard computer

Once a camera was selected, pictures (320 x 240 pixels) were taken by the selected camera. The onboard Raspberry Pi computer used Linux folder mounting in order to save the pictures to the user's interface computer via a Wi-Fi connection. The onboard computer constantly checks the files that have been saved to the interface computer (Window 10;

16.0 GB of RAM; 8-core Intel i7-6700 3.40 GHz; Nvidia Quadro M2000 series video card) for being accessed and after the pictures are accessed, inspected, and displayed by the interface computer, the onboard computer has access to the pictures and deletes them. In Fig. 5 below, nine frames are captured within a 500 ms gap in time to demonstrate rotation of the wheel.

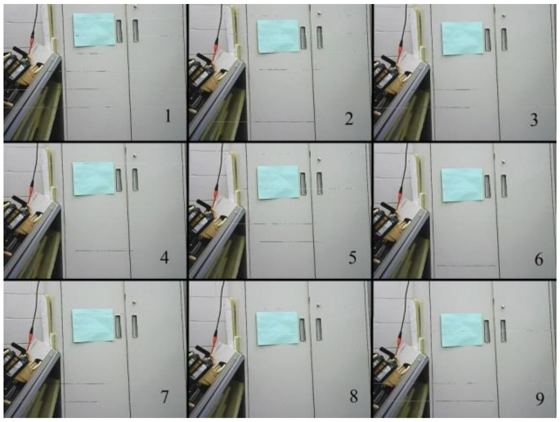

Fig. 5. Nine frames captured of green paper within 500 ms while the wheel is in rotation

In the process of collecting the pictures saved by onboard computer the interface computer flags them as being accessed while they are inspected and displayed (see Fig 6.). The pictures are multi-buffered to an interface and manipulated via the orientation of the camera estimated and saved via the onboard computer. The pictures are displayed to a user interface, which imitates a streaming video.

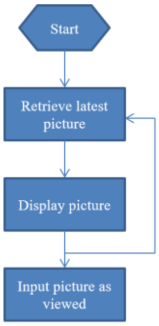

Fig. 6. Flowchart of interface computer

The design of the mounting structure, the camera placement, and the viewing capacity require an analytical understanding of the entire system. In Fig. 7, the design of the multiple camera system and view range is demonstrated.

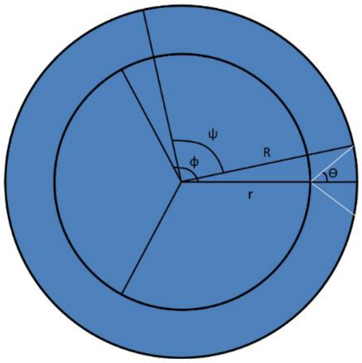

Fig. 7. Figure of angles and distances of the tested system

Equation (1) below finds the resultant ψ, where Θ is half the viewing angle capable by the cameras, ϕ is angle given by 360° divided by the number of cameras mounted, r is the radius of the structural mount, R is the radius of the designated view distance, and ψ is the angle of the dark region (region where pictures cannot be captured), respectively.

$$\psi \;=\; \phi + 2\left(\sin^{-1}\!\left(\frac{r}{R}\sin(\Theta)\right) - \Theta\right) \qquad (1)$$

III. Results

In Fig. 8, the communication of the system is demonstrated. The motor was started, depending on the encoder data, the cameras were selected, pictures were taken, the Raspberry Pi computer sent pictures to the interface computer via a mounted folder using Wi-Fi (100 Mbps), and the interface computer manipulates the pictures to imitate a streaming video using data collected from the motor encoders.

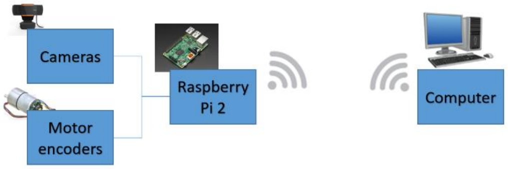

Fig. 8. Diagram of the experienced communication

A range of rotational speeds were tested in order to determine the optimal speed at which the cameras were producing the pictures. Fig. 9 demonstrates the rotational speeds, revolutions per second, and the capable frames per second demonstrated by the cameras. The trend seen in Fig. 9 demonstrates a higher rate of frames when the motor operates at slower speeds due to the programming necessary to switch between two cameras. From this data, the optimal rotational speed was found to be 1.096 revolutions per second producing 2.71 frames per second. Better framerates could be achieved at slower revolutions but are significantly too slow for the applications of spherical robots.

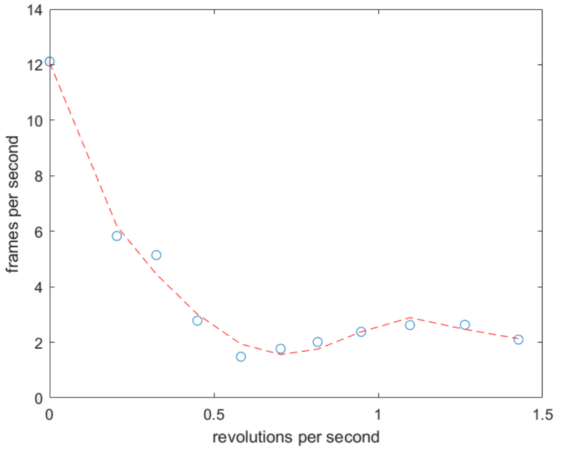

Fig. 9. Rotational speeds and frame rates

The analytics of the system was tested through the tracking of the dark region demonstrated by ψ in Fig. 7. In order to test the analytical dark region, the calculated revolutions per second were used to track the time a distinguished point cannot be seen in a set of pictures. For this testing, r was measured to be 14 inches, R was determined to be 20 inches, ϴ was found to be 35°, ϕ was found to be 120°, and the resultant ψ was found to be 117.84°. Discrepancies were found between the analytical and tested tracking of the dark region. The discrepancies found could be related to the low frame rates currently capable by the system. Optimizing the system with higher quality cameras, the addition of a multiplexer, or other improvements on the system should achieve more accurate results.

With the analytical calculations and the operation of the structure and cameras, testing was conducted on the manipulation of the pictures in order to mimic the motion of the cameras in a user interface. The pictures were displayed similar to a low framerate video as they were rendered. The motion of the pictures imitated the rotation of the wheel's rotation but slowly developed an error the longer the process lasted. The rendering and manipulation was done using Visual Studios.

With the design for testing the camera system being demonstrated, the number of distributed cameras needed to guarantee a constant forward-facing field of view was also simulated. The number of cameras necessary depends primarily on the cameras' viewing angle and the minimum radius desired for full viewing. For example, suppose the robot has a radius of 1 (normalized units) with 16 equally-distributed cameras of viewing angle 75 degrees (modeled here as a circular image plane), each mounted on the sphere's surface. Each camera as modeled can see a cone of space projecting out from their focal point – see Fig. 10. At some distance, those cones intersect; at some further distance, they intersect without any gaps thus forming a full viewing coverage. For sixteen 75-degree cameras, this corresponds to a minimum viewing distance of around 10 times the robot's radius.

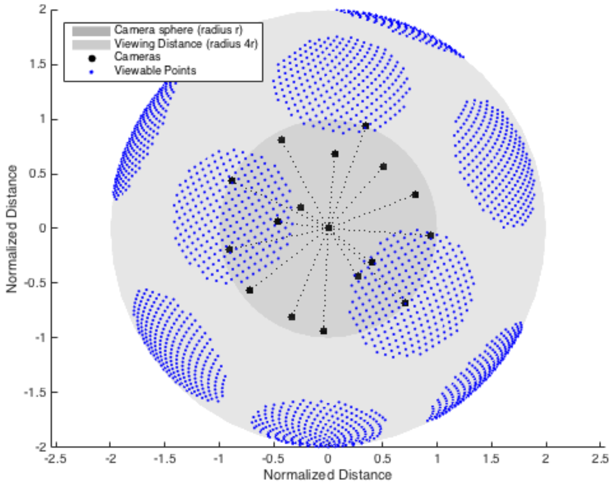

Fig. 10. Example visible regions on a sphere

IV. Conclusion & Future Work

The calculations are in place to estimate the motion of a spherical robot. The onboard computer demonstrated its capability to control, read, and manage the motors, select the cameras, take pictures and manage data cleanup. The onboard computer was able to mount the interface computer and pictures were sent to the interface computer via mounting using Wi-Fi. The interface computer demonstrated its capability to manipulate and display a streaming video. The calculations were done to demonstrate the number of cameras that would be needed by the rotational wheel mount and the four-pendulum spherical robot demonstrated in this paper.

Future work includes experimenting further with limitations of the system to later optimize, such as the process of selecting cameras, experimental data on resolution vs. fps, experimental data on number of cameras vs. fps, among other experimental data. Continual testing on adding more cameras onto the wheel mount to reduce the dark region angle to zero, which is represented by (1). Also, mounting cameras on a multi-axis frame and the four-pendulum spherical robot itself. Some other ways to optimize the system could be achieved through the addition of a multiplexer to allow for multiple cameras being active simultaneously or additions to the logic in activating individual cameras.

Acknowledgment

This work was funded in part by the Central Michigan University FRCE Grant #48868 and GSRCE Grant from the Office of Research and Graduate Studies.

REFERENCES

[1] R. Siegwart, I.R. Nourbakhsh, D. Scaramuzza, Introduction to Autonomous Mobile Robots, 2nd ed., 2010.

[2] The Knowledge Transfer Network, Robotics and Autonomous Systems, https://connect.innovateuk.org/documents/2903012/16074728/RAS%20UK%20Strategy?version=1.0

[3] National Robotics Initiative 2.0: Ubiquitous Collaborative Robots (NRI-2.0), Division of Information & Intelligent Systems, United States Department of Agriculture, NSF, USDA, DOE, DOD, NSF Announcement of 2014 https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=503641

[4] Richard Chase, Abhilash Pandya, *A Review of Active Mechanical Driving Principles of Spherical Robots*, Wayne State University, Robotics 2012, 1(1), 3-23; doi:10.3390/robotics1010003

[5] GuardBot, "Rotundus," http://www.rotundus.se/, September 2006.

[6] V. Kaznov and M. Seeman, "Outdoor Navigation with a Spherical Amphibious Robot," IEEE/RSJ International Conference on Intelligent Robots and Systems, October 2010.

[7] J. D. Hernández, J. Barrientos, J. del Cerro, A. Barrientos, D. Sanz. "Moisture measurement in crops using spherical robots". *Industrial Robot: An International Journal*, vol. 40, pp. 59–66, 2013

[8] E. Kirksey, M. Von Damitz III, *OrbSWARM*, Burning Man, 2007, http://blog.orbswarm.com/

[9] M. Seeman, M. Broxvall, A. Saffiotti, P. Wide, "An Autonomous Spherical Robot for Security Tasks," Orebro University, Sweden, CIHSPS 2006 - IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, Alexandria, VA, USA, 16-17 October 2006

[10] Z. Qiang, L. Zengbo, C. Yao, "A Back-stepping Based Trajectory Tracking Controller for a Non-chained Nonholonomic Spherical Robot," Chinese Journal of Aeronautics, Volume 21, Issue 5, October 2008, Pages 472-480, 2008

[11] B.P. DeJong, E. Karadogan, K. Yelamarthi, J. Hasbany. "Design and Analysis of a Four-Pendulum Omnidirectional Spherical Robot," Journal of Intelligent Robot Systems, 2016.