Light Field Array Camera

Rafe Denham

***Abstract*—The aim of this project was to design and build a parallel camera array that can capture a light field. This was completed through the use of Raspberry Pi cameras and Raspberry Pis, an Ethernet network to facilitate communication and a frame to hold it together. The final camera array lacked good calibration, but outputted clear, high resolution, images of still scenes. The resulting camera array facilitates further research into light field arrays and also opens the possibility of parallel processing though the use of the Raspberry Pi cluster.**

I. Background

Since the inception of robotics, autonomous imitations of human senses, actions and behaviours have proven considerably more difficult to implement than perceived by most. One of these enigma is computer vision, which is being rapidly broached in modern robotics and is being applied to automation in a wide range of fields. However, these advances in computer vision analysis and processing are consistently hindered by the need for things that humans do subconsciously.

An array of parallel cameras can retrieve a light field array which can be used to remove occlusions, remove shadows, 3D image, perceive depth and post-capture focus - to name a few applications among many [1] [2] [3] [4]. Applying this technique to robotics allows for better image filtering, object recognition and overall computer vision accuracy.

A. Scope

The project is restricted to building a scalable array of cameras with synchronised shutters which, upon request, send their individual image data streams to embedded computers for initial image processing and compiling. This synchronised data then needs to be sent to an external server for further processing. A frame will be required to facilitate the cameras, embedded computers and associated cable management.

B. Related Work

1) Previous planar camera arrays: Three prominent large monocular camera arrays feature in current literature. The Standford 100 camera array [5], the Massachusetts Institute of Technology (MIT) 64 camera array [6] and the Delaware 9 camera array [7].

The Stanford and Delaware arrays dealt with the high volume data streams associated with capturing several images simultaneously through brute force, while the MIT array used dynamic user input to select regions of interest and only captured the computed relevant regions of the images.

The synchronicity of the images depended upon whether explicit hardware synchronisation was used or not. The Stanford and Delware arrays used explicit hardware synchronisation resulting in a disparity of only 100-200µs [5] [7]. Alternatively, the MIT array used implicit synchronisation though software which caused a larger disparity of 5-10ms [6].

The robustness of these existing arrays is poor as they were designed for scientific use in a controlled environment. All frames consisted of an open steel design, allowing for easy access to manual adjustments and to the cabling. However this also exposed the cameras for incidental bumping requiring each to be manually adjusted and re-calibrated after being relocated.

2) Calibration: In a calibration method decribed by Zhang (2000) [8], each camera is treated individually with one being a reference point for the rest. A number of images are captured at different depths and orientations with reference points compared between them. Zhang describes a camera view through two extrinsic, rotation (R) and translation (t), and five intrinsic parameters, the coordinates of the principle point(u_0,v_0), axis scale factors (α, β) and skew (γ).

An alternative method is described by Vaish et al. [9]. This method is more simple and is only applicable to planar camera arrays. It only adjusts the camera positions by a translation, identifying the rest of the parameters described by Zhang [8] as redundant when applied to a planar array.

3) Light field photography techniques: The captured, decoded and calibrated light field can be used to compute new views of the scene. Through calculating the projected light rays, a synthetic reference plane can be moved, simulating a change in focus of the rendered image. This is visually evident in Figure 1, by varying *D*, while keeping the other parameters constant, the focal length is changed - resulting in a change in focus of the subject.

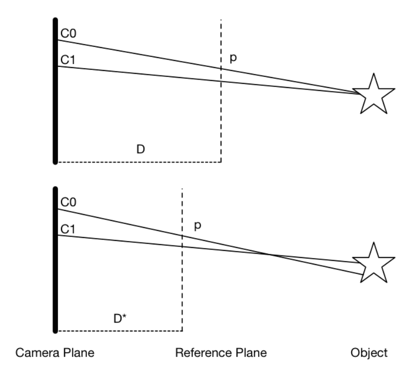

Fig. 1. Diagram of the simulated focal length change.

II. Implementation

A. Design

With reference to the overall structure defined in the Scope, a design was devised using Raspberry Pi camera modules each connected to a Raspberry Pi. The Raspberry Pis are networked together, allowing them to act as essentially one device. The captured images are sent over the network to the external server.

In the initial design, the networking was done by using one Raspberry Pi as a network switch and a capture device, with the remaining three Pis connecting to this network. Each Raspberry Pi has one Ethernet port and four USB 2.0 ports, totalling five usable Ethernet ports though the use of USB adapters - four out of these five are needed to connect the other Raspberry Pis and the external server. Providing a safe 5V 1A of power to each Raspberry Pi is a USB charging hub.

With regard to scalability, the design is simply scalable through increasing the number of cameras and Pis, while daisy chaining these network switch configured Raspberry Pis and increasing the power supply as necessary.

1) CompoundPi: Part way through the completion of the project, an open source project that is very early in development was discovered. CompoundPi, currently in Version 0.3, provides a means for controlling multiple cameras attached to Raspberry Pis all of which are attached to the same network. Broadcast User Datagram Protocol (UDP) packets are utilized to permit near-simultaneous triggering of all attached cameras.

B. Prototype

The prototype 2x2 camera array build was executed successfully, although it revealed minor adjustments that need to be made for the final design. It was initially assumed that the Raspberry Pi had the capabilities to act as a unmanaged network switch. However to implement this proved to be too much work to be used in place of a cheap unmanaged Ethernet switch - especially when it was still required to capture a synchronised image. In the adjusted design, each Raspberry Pi is connected to the Ethernet switch and the Ethernet switch to the external server.

The mechanical design did not include the projection of the ports of the Raspberry Pi and the SD card, and hence did not fit together as planned. This was adjusted for the final design.

The prototype was tested for image quality and synchronisation which produced encouraging results.

C. Final

The final array consisted of 16 cameras in a 4x4 planar configuration, refer to Figure 2 for the final design diagram. In order to upgrade the design from the prototype, three Ethernet switches were daisy chained together. A computer 250W power supply was used to supply 5V 1A to the Raspberry Pis and 5V 0.6A to the Ethernet switches.

After the initial testing of the prototype it became evident that mounting the cameras with the same orientation was important and the frame was updated relevantly. The fixing method also needed to be changed, and a mortise and tenon fixing was designed.

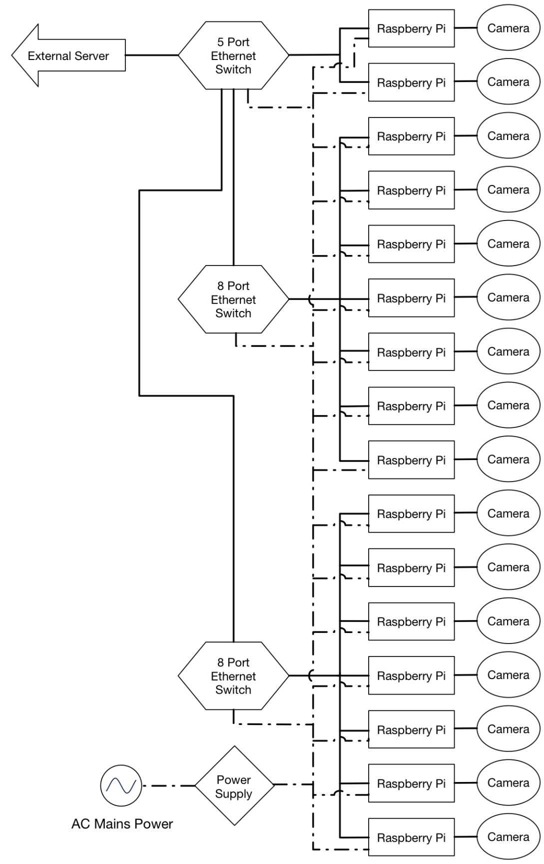

Fig. 2. Final design diagram.

III. Evaluation

The camera array performed as it was designed to, however one camera cable became unfixed - resulting in a hole in the light field array. This is discussed further in the associated Project Report.

The image quality and synchronisation of the final outputted images were tested, with good image quality results and relatively poor synchronisation results that would impact use in dynamic environments.

In comparison to the related camera arrays discussed previously, the array performed well. The image quality was higher, while the synchronisation disparity was also higher. With regard to the mechanical robustness, the final camera array was better protected in its enclosure. However, the unreliable camera connectors in combination with the lack of access to that part of the design causes problems that would have been easily fixed in an open implementation like the Stanford array.

A. Calibration and Image Processing

The calibration and image processing of the final array produced poor results. The cause of this is likely due to the calibration of the cameras in the array, possibly the implementation of the calibration algorithm was incorrect. For in-depth use of the light fields captured by this camera array, further and better calibration of the cameras must be completed.

IV. Future Work

The project provides solid infrastructure for further research into light field photography, calibration and image processing.

An interesting avenue to explore is the possiblity of on-board or parallel processing on the Raspberry Pis. An example of on-board processing is the calibration algorithm described by Zhang [8], this requires a transformation image to be applied to each images before it is decoded into a light field.

With regard to parallel processing, the Raspberry Pis could be configured to work as a computing cluster with high parallel processing capabilities. Parallel processing is especially good for image processing algorithms due to their intrinsically parallel nature.

V. Conclusion

The final design has been proven to output a light field successfully, however further calibration is needed to align this light field correctly in order for it to be used further.

This project highlighted importance of good initial planning and research (with regard to the lack of direct synchronicity of the RPi cameras), the importance of good light field calibration and also the attractive idea of parallel processing.

References

[1] *Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation*, vol. Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, 2013.

[2] D. G. Dansereau, "Plenoptic signal processing for robust vision in field robotics," Ph.D. dissertation, The University of Sydney, 2014.

[3] A. Yamashita, T. Kaneko, and H. Asama, "Precise extraction of visual information from images by image processing techniques."

[4] K. Venkataraman, D. Lelescu, J. Duparr, A. McMahon, G. Molina, P. Chatterjee, R. Mullis, and S. Nayar, "Picam: An ultra-thin high performance monolithic camera array," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 166, 2013.

[5] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 765–776, 2005.

[6] J. C. Yang, M. Everett, C. Buehler, and L. McMillan, "A real-time distributed light field camera." *Rendering Techniques*, vol. 2002, pp. 77–86, 2002.

[7] *Dynamic fluid surface acquisition using a camera array*, vol. Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

[8] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.

[9] *Using plane+ parallax for calibrating dense camera arrays*, vol. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on 1. IEEE, 2004.