Queensland University of Technology

Honours thesis

# Planar Calibration of Camera Arrays for Light Field Acquisition

*Ashley Stewart*

supervised by

Dr. Donald Dansereau

November 10, 2016

# Declaration

The work contained in this thesis has not been previously submitted to meet the requirements for an award at the Queensland University of Technology or any other higher education institution. To the best of my knowledge and belief, this thesis contains no material previously published or written by another person except where due reference is made.

**Ashley Wilton Stewart**

November 10, 2016

**Abstract**

Light field cameras are an emerging technology with unique post-processing capabilities. Their applications in computer vision are numerous, especially in robotics, navigation systems and surveillance. Within the medical field, there is a serious potential for occlusion removal in endoscopic imaging and surgical theatre videos using light field cameras. In order for results to be achieved in any such application, calibration is essential. Existing calibration procedures are either conceptually complex metric procedures, or non-metric procedures inflexible to camera orientation and planarity. We present a novel and conceptually simple non-metric calibration for light field acquisition, suitable for camera arrays with constant yet arbitrary camera poses. We also provide a quantitative measure of calibration accuracy, and use it to demonstrate the procedure's efficacy with our Raspberry Pi camera array. Additionally, we present qualitative results by rendering light fields at varying levels of focus and occlusion, and demonstrate success in capturing and rendering light field video - an area of particular interest for our applications. Finally, we reflect on implementation challenges and lessons learned.

# Acknowledgements

I would first like to express my sincere gratitude to my thesis supervisor Dr Donald Dansereau of the Science and Engineering Faculty at QUT, for his continual support, patience, enthusiasm and vast knowledge. He is always passionately reactive to my ceaseless volleys of questions, even when overseas. Thank you for being an excellent supervisor.

Second, I would like to thank Steven Martin, and acknowledge his essential contributions to the project, without which we would not be able to produce any results. Steven assisted immensely by organising the delivery of components, materials and designs.

Last but not least, I would like to express my very profound gratitude to my family and my fiancée, who have given me unfailing support and continuous encouragement throughout my years of study. Thank you. This accomplishment would not have been possible without you.

# Contents

# 1 Introduction

## 1.1 Motivation

Light field cameras are an emerging technology with unique post-processing capabilities. Along with providing useful spatial information for computer vision, light field cameras can identify objects, construct artificial views, and render with increased focus in poor visibility. Their applications in computer vision are numerous, especially in robotics, navigation systems and surveillance.

Light field technology in the medical field may facilitate the removal of occluders from surgical images, such as those obtained by an endoscope during arthroscopic procedures. Vision through an endoscope in such procedures is often hindered by floating debris, bubbles and equipment. The US corporations Stryker and Intuitive Surgical have both expressed interest in applying computer vision technology in these areas. In addition, light field technology could also be applied in the surgical theatre, providing useful views for training and education. This could be achieved using occlusion removal techniques with light field video.

To enable any light field camera to perform well in such applications, accurate calibration is essential. A calibrated light field camera is able to capture light fields because it is aware of the relationships between each of its views. Our implementation uses a camera array, and although several calibration procedures exist for camera arrays, current procedures are either conceptually complex 'full metric calibrations', or they are inflexible to camera orientation and non-planar setups.

## 1.2 Outcomes

We present a novel and conceptually simple calibration for light field acquisition, suitable for camera arrays with arbitrary camera poses. We also provide a quantitative measure of calibration accuracy, and use it to demonstrate the procedure's efficacy with our Raspberry Pi camera array. Additionally, we present qualitative results by rendering light fields at varying levels of focus and occlusion, as well as success in capturing and rendering light field video - an area of particular interest for our applications. Finally, we reflect on

implementation challenges and lessons learned.

# 2 Prior work

## 2.1 Camera array implementation

The Raspberry Pi camera array, when initially implemented by Rafe Denham, achieved poor light field rendering quality [3] (see figure 1). The calibration method used was one proposed by Vaish et al., and involves a *plane + parallax* framework [5]. In Denham's evaluation, a definitive reason for the poor rendering quality is not provided, but it is suggested that it may have been due to an incorrect implementation of Vaish's algorithm.

**Figure 1:** Light field calibrated using Vaish's plane + parallax procedure exhibiting poor focus. Adapted from Rafe Denham. Light field array camera. Technical report, Queensland University of Technology, June 2015

After some testing of the original, we discovered that although the cameras appeared to be planar with a uniform viewing direction, in reality they exhibited significant arbitrary rotation, which Vaish's method does not account for. The lack of conformity in poses may have been at least partially related to the original medium-density fibreboard front-plate, which has since been replaced (see section 3). However, even after replacing the front-plate, the cameras continue to retain significant arbitrary rotation (see figure 2).

**Figure 2:** Images taken from two horizontally adjacent cameras in our camera array. The red lines illustrate the variance in camera orientation. This variance must be accounted for if calibrated light fields are to be generated.

## 2.2 Calibration literature

### 2.2.1 Identifying the research gap

Calibration procedures that facilitate light field acquisition can be categorised as either metric or non-metric. Metric calibration procedures recover precise camera positions and orientations. These methods are conceptually complex and take time to implement. Non-metric procedures, while simple, only recover camera positions to some unknown scale. However, it turns out that this is acceptable for most light field applications including 3D reconstruction, synthetic aperture imaging, light field rendering and space-time view interpolation.

Vaish's *plane + parallax* calibration procedure is the only non-metric light field camera calibration procedure that we are aware of. The procedure is one of the simplest procedures for light field acquisition, and it is currently used

by Stanford University for their multi-camera array. The procedure requires each camera to be arranged on a plane. Images can then be projected onto a reference plane parallel to the camera plane, by applying a homography matrix. The $x, y$ positions of cameras (to an unknown scale) can then be recovered by measuring the parallax of a single feature point that does not lie on the reference plane. A limitation of this procedure is that it requires all cameras to lie on a plane and have a uniform viewing direction. As discussed, our cameras exhibit arbitrary rotation, which cannot be accounted for by this procedure (see figure 2). Vaish et al's experiments indicate that their non-metric method actually yields results better qualitative results than those acquired by a full metric calibration.

Perhaps the most appropriate existing calibration procedure for our camera array is one proposed by Xu et al [6]. The procedure extends Zhang's single-camera calibration [7] to provide a metric calibration for mobile camera arrays. This directly solves the camera pose problem, because the procedure works with camera arrays irrespective of their planarity or camera poses. However, this is a metric procedure that goes beyond our needs, and is conceptually complex - which is what Vaish's method, while inflexible, intended to address (with the addition of providing better qualitative results).

For our camera array, we have a need for a new calibration procedure that bridges the gap between Vaish's method and Xu's method - one that is flexible to orientation like Xu's, yet maintains the non-metric simplicity of Vaish's.

### 2.2.2 Leading the way to planar calibration

A *monocular co-registration* procedure described by Donald Dansereau can be extended and formalised as a novel calibration [2]. The procedure relates conventional 2D images via the plenoptic function, since any conventional image can be considered some subset of the function. Dansereau states that given a set of images captured by collinear cameras, it is possible to reproject the images into a common parametrisation, so long as the images overlap sufficiently. If we determine the geometric transforms separating images via some reference plane in advance, we can exploit this parametrisation to align images and construct light fields (see figure 3 and figure 4).

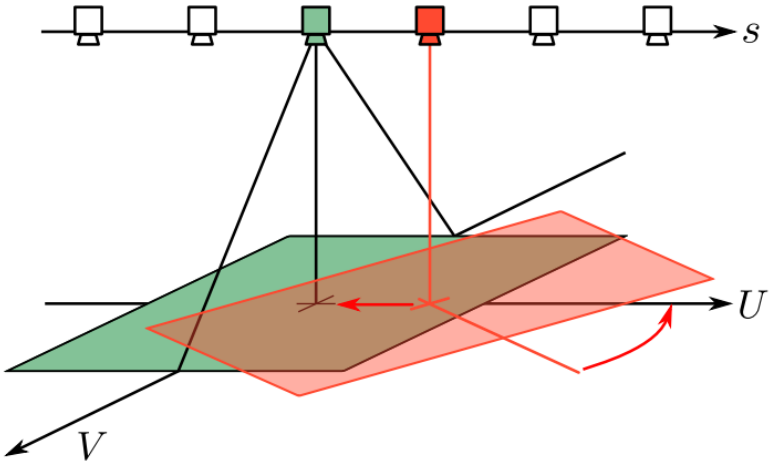



**Figure 3:** The simplified case of a planar scene and collinear cameras having only rotations about the principal axes and translations parallel to the scene. Adapted from Donald Gilbert Dansereau. *Plenoptic Signal Processing for Robust Vision in Field Robotics*. PhD thesis, The University of Sydney, 2014

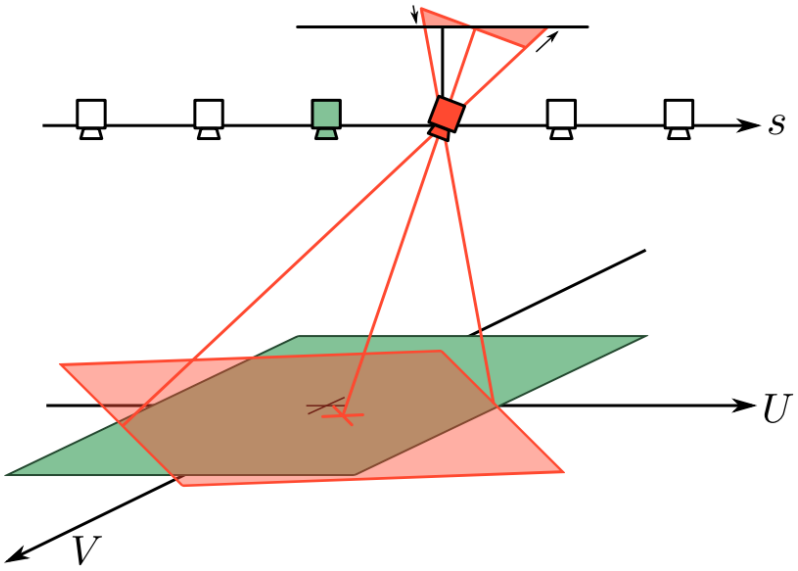



**Figure 4:** In the case of cameras having arbitrary rotations, images can be reprojected onto a plane parallel with the $U, V$ plane, based on an estimate of the camera's orientation. Adapted from Donald Gilbert Dansereau. *Plenoptic Signal Processing for Robust Vision in Field Robotics*. PhD thesis, The University of Sydney, 2014

# 3 Adjustments to the implementation

Our camera array's casing was originally built using medium-density fibreboard. After some testing, it was clear that the material significantly reduced the potential for a consistent calibration. This is because medium-density fibreboard is simply too bendy to maintain its form. This likely contributed to the poor quality of the original light field renders. Fortunately, we have been able to address this by designing, building, and installing a sturdier aluminium front-plate (see figure 5). The design specifications for the new front-plate are provided in appendix 10.1.

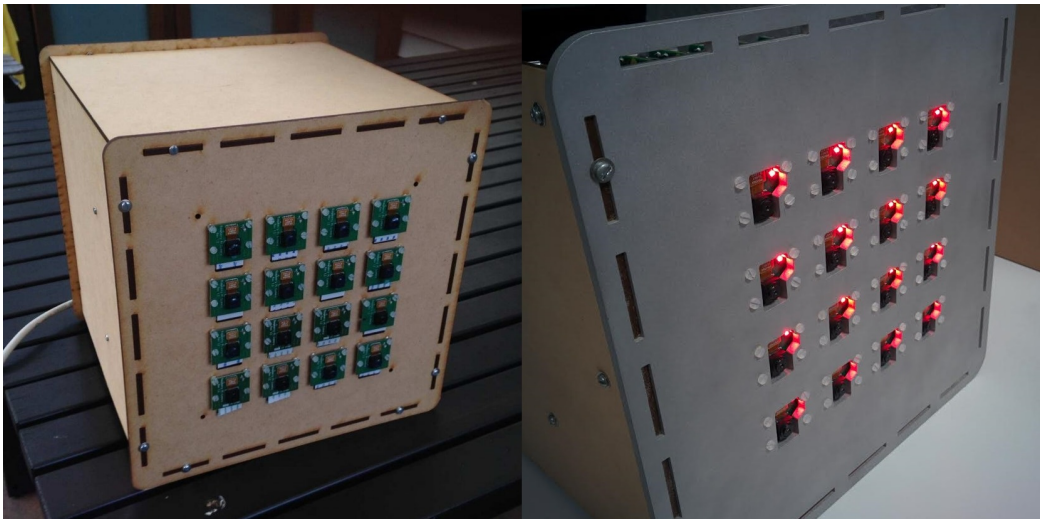


**Figure 5:** The original camera array (left) and the camera array with the new aluminium front-plate installed (right).

Although the front-plate is now sufficiently sturdy and maintains its form, the cameras themselves still retain some arbitrary rotation (see figure 2). This rotation may be due to the nylon nuts and screws used to mount the cameras, which do not fit as tightly as stainless steel materials would. Stainless steel nuts and screws were considered, but were ultimately rejected because they may have damaged the camera boards or caused short-circuits. The cameras themselves may also have some slight variance if they are not built to be absolutely identical. Fortunately, this rotation can be addressed by our novel calibration procedure.

# 4 Procedure

## 4.1 Calibration

The monocular co-registration procedure introduced in section 2.2.2 can be extended and formalised as a novel non-metric calibration procedure suitable for our camera array. The key part of the procedure lies in the estimation of geometric transforms that relate image pairs captured by the camera array, relative to some reference plane. To achieve this, we can capture a full array image of a calibration pattern positioned at the reference plane. This image set will be called the *calibration set*, and it will inform our estimation.

### 4.1.1 Capturing an effective calibration set

Our first step is to choose an appropriate reference plane at which to situate the calibration pattern. The reference plane should be one that provides sufficient overlap across camera views, so that common features can be matched (see figure 6).

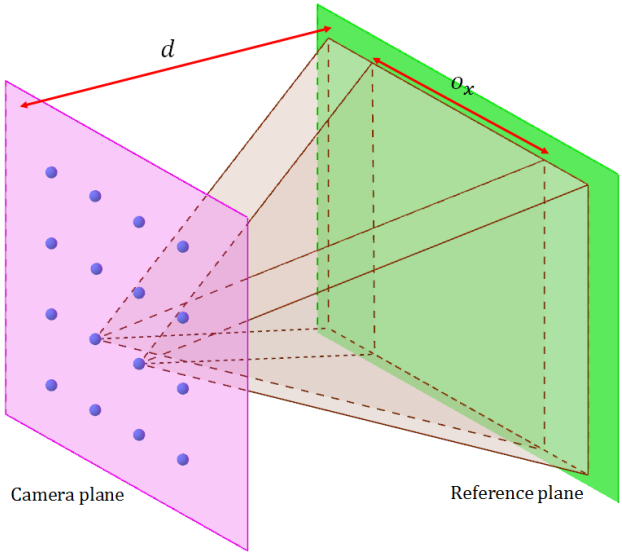



**Figure 6:** Diagram of our camera array and reference plane. Two camera views are projected onto the reference plane, which is set up at distance $d$ from the camera plane. Significant overlap in the $x$ direction is illustrated as $o_x$.

It is clear that the degree of overlap between any two camera images is a function of each camera's pose, field of view, translation, as well as the distance $d$ from each camera to the reference plane. For a coplanar setup with uniform viewing directions, overlap increases logarithmically with $d$, with total overlap occurring as $d$ approaches $\infty$. Thus we should consider that as $d$ increases for such setups, so does the required size of the calibration pattern. Additionally, if we consider the camera array a virtual camera, then the ratio between the synthetic aperture size and our synthetic focal length (this focal length is also given by $d$) will have a significant effect on the focal sensitivity of light fields. Clearly, the most practical calibration setup will depend on the resources available, the camera setup, and the intended use of the camera. For our calibration setup, see section 5.1.

After choosing an appropriate reference plane, we can choose a calibration pattern. Most calibration procedures involve a checkerboard pattern, but this is inappropriate for our procedure since we rely on the detection of *unique* features across images. Checkerboards and other repeating patterns have non-unique surface features which can easily be mismatched with other identical features. Therefore, the calibration pattern must be sufficiently detailed and non-repeating. The colourful impressionistic paintings of Leonid Afremov have proven effective for our setup.

Once we have a calibration set that meets the aforementioned requirements, we can begin to develop our transformation set for rectification.

### 4.1.2 Calculating the geometric transforms between reference plane views

For each image in the calibration set, we can identify surface features automatically via the Speeded Up Robust Feature detection algorithm (SURF) [1]. SURF has been shown to identify sufficient features quickly and effectively. We can then identify unique matching surface features between successive image pairs, and thus estimate their relative geometric transforms. Estimating geometric transforms can be achieved by applying a *RANdom SAmple Consensus* algorithm (RANSAC). We have opted to use *M-estimator SAmple and Consensus* (MSAC), which additionally evaluates the quality and likelihood of the consensus set [4]. We can exploit this to enforce a minimum confidence level required to produce a positive match. A MATLAB implementation of our calibration procedure is provided in appendix 10.2.1.

### 4.1.3 Measuring calibration accuracy

To quickly assess the accuracy of our transformation set, we can use the transformation set to project all the images from the *calibration set* onto a new image plane, effectively constructing a panorama. It turns out that constructing such a panorama provides an effective means to visually assess the quality of the transformation set (see figure 7). A good transformation set will result in a smoothly stitched panorama.


**Figure 7:** Jagged panorama of a calibration set (left) and smooth panorama of a calibration set (right). The calibration set used to construct the smooth panorama will produce better results for light field applications. The jagged panorama was built using a poor calibration set which was not of a fully planar scene. Ignore any clear differences in colour across the panoramas - this is due to automatic colour balancing and gamma correction in the camera modules.

To more accurately assess the quality of a transformation set, we can evaluate the positional consistency of surface features across images in the panorama, one image at a time. The closer each surface feature is to having a uniform position across images, the better the transformation set is for light field applications.

## 4.2 Rectification

A transformation set built using the procedure described in the previous section (section 4.1) can be used to rectify any set of images captured by the camera array. Rectification will bring images into light field alignment and enable popular light field applications such as synthetic aperture focusing.

In section 4.1.3, we suggested that projecting calibration images onto plane (effectively constructing a panorama) is useful as a quick way to assess the accuracy of a transformation set. This is very similar to the rectification process. To rectify an image, we can simply project only the image of interest onto the 'panorama plane' using the appropriate transformation (see figure 8). Repeating this for all images in a set will rectify the set for light field acquisition and rendering. This effectively brings all images into alignment so that when rendered as a light field, the focus is on the reference plane. Focus can then be easily adjusted to any other parallel plane. A MATLAB implementation of the rectification procedure is provided in appendix 10.2.2.

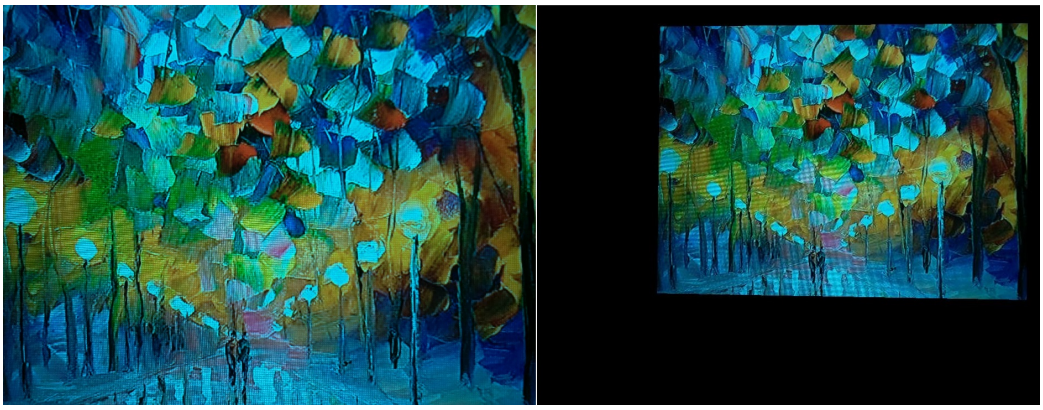


**Figure 8:** Original image (left) and rectified image (right).

Rectified images have a side effect of having potentially large areas of blackness. This may be inconvenient or significantly affect the file size of large sets. It is possible to crop rectified images, but this also requires that the relative positions of cameras be known. This is because the translations between cropped portions must correspond to the relative camera positions associated with each image. Cropped images must also be of uniform dimensions. We illustrate both possibilities (see figure 9).

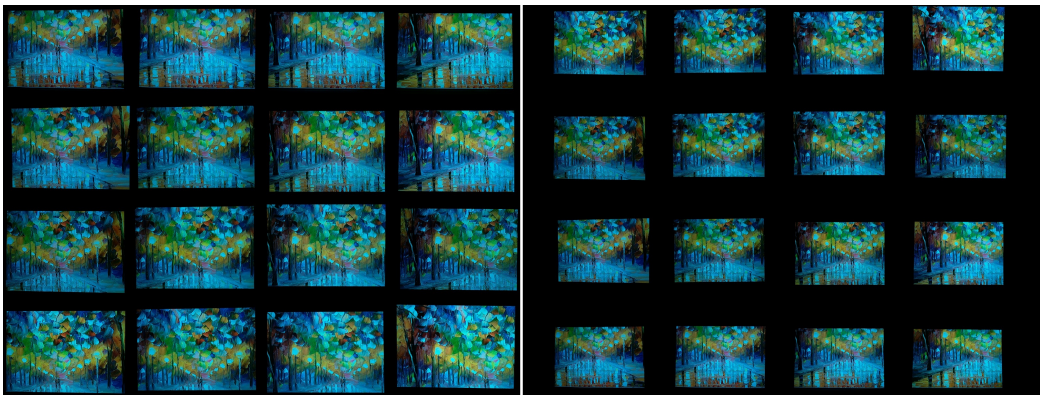


**Figure 9:** Cropped set (left) vs. uncropped set (right). In the uncropped set, each image is the size of a full panorama.

# 5 Results

This section presents results obtained using the proposed calibration procedure with the Raspberry Pi camera array.

## 5.1 Our implementation

So far, we have found that the most effective calibration patterns are detailed images such as paintings (see figure 10).

We have positioned our Raspberry Pi camera array parallel to a TV displaying such a pattern (see figure 11). This setup makes it easy to change the pattern while keeping the same reference plane distance.

**Figure 10:** A calibration pattern that achieved good results for our camera. The image is of Leonid Afremov's *Farewell to Anger*.

**Figure 11:** Our calibration setup. We have displayed our calibration image on a TV to ensure planarity, and aligned by visual inspection.

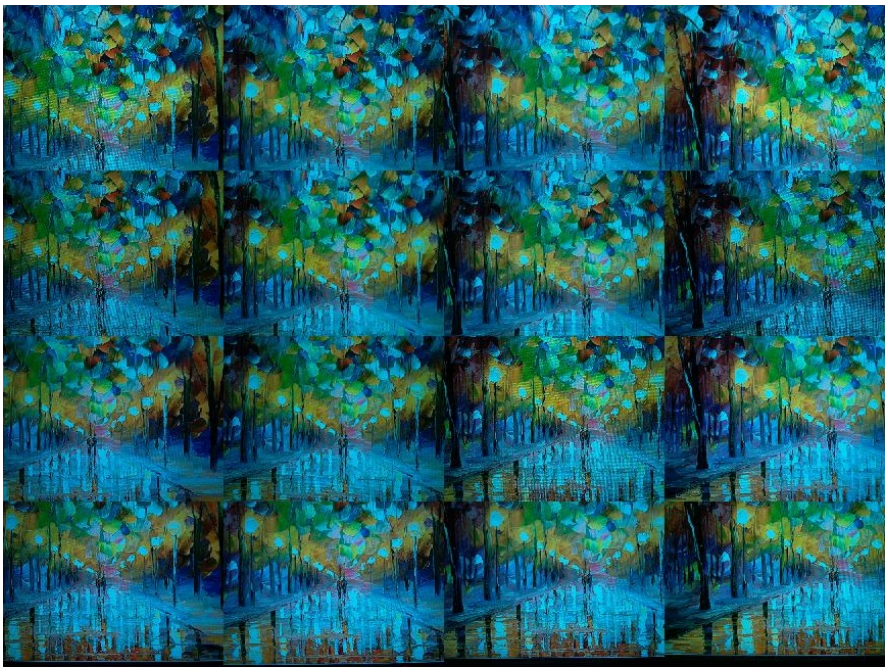


**Figure 12:** Mosaic of our calibration set

### 5.1.1 Approximate overlap

For our 4x4 camera array, we have camera translations of $t = 34.5$ mm, camera fields of view $\phi = (53.5°, 41.41°)$, and a distance to the reference plane $d \approx 300$ mm. Using basic geometry, and by assuming a precisely coplanar camera array and perpendicular viewing directions, we have:

$$\begin{aligned} \text{Projection width } P_x &= 2d\tan(\frac{\phi_x}{2}) \\ &= 2 \times 300 \times \tan(\frac{53.5}{2}) \\ &= 302.424 \text{ mm} \end{aligned} \tag{1}$$

$$\begin{aligned} x \text{ overlap} &= \frac{P_x - t}{P_x} \\ &= \frac{302.424 - 34.5}{302.424} \\ &= 88.59\% \end{aligned} \tag{2}$$

So we have approximately 88.59% overlap between camera views in the $x$ direction. Applying the same equations in the $y$ direction, we calculate approximately 84.78% overlap.

## 5.2 Calibration accuracy

In section 4.1.3, we suggest that calibration accuracy can be tested by comparing the relative coordinates of features across rectified images. For a planar scene (such as the calibration set scene), perfect calibration is achieved when coordinates of features match precisely. A MATLAB implementation of this assessment procedure is provided in appendix 10.2.3.

We have calculated such pixel inconsistencies over multiple passes of the calibration procedure (see table 1). Multiple passes are achieved by estimating and transforming already rectified images successive times.

| **# of passes** | **x-inconsistency (pixels)** | **y-inconsistency (pixels)** | **Overall inconsistency (%)** |
|---|---|---|---|
| 1 | 5.092 | 5.155 | 0.23 |
| 2 | 1.178 | 0.884 | 0.043 |
| 3 | 1.184 | 0.917 | 0.043 |
| 4 | 1.160 | 0.925 | 0.042 |

**Table 1:** Average pixel inconsistencies of reference plane features detected across rectified images

It is clear that after the second pass, we see significant diminishing returns. For our implementation, we have therefore opted for two passes. We achieve an overall final inconsistency of 0.043%, from 0.23% with only one pass (an improvement of a factor of 5.35). Calibrated light fields can therefore be considered to be 99.957% accurate in terms of their alignment and resultant focus.

It is critical to note that the average difference in feature positions does not necessarily correspond to reprojection error. Reprojection error is a common accuracy measure used in metric calibration procedures. Like Vaish et al's plane + parallax procedure, our procedure is non-metric, so we cannot calculate reprojection error. Vaish points out that non-metric procedures are not calculating the same intrinsic and extrinsic camera parameters, or making the same assumptions as metric calibration procedures [5].

## 5.3 Light field rendering

Light fields can be rendered by shifting rectified images or *light field slices* to a common depth, then adding the slices together to yield a single 2D output. Dansereau provides an implementation of this in his *Light Field Toolbox* for MATLAB as `LFFiltShiftSum`. Light fields will initially need be loaded via `LFReadGantryArray`, which can take a rectified image set as input.

We can demonstrate the effectiveness of our calibration procedure by attempting to render a planar scene. We juxtapose an uncalibrated light field render with its rectified counterpart (see figure 13). Note the increased clarity after rectification.

**Figure 13:** Uncalibrated images rendered as a light field (left) vs. light field calibrated using our procedure (right).

### 5.3.1 Synthetic aperture focusing and robustness to occlusion

We can demonstrate our calibration procedure's appropriateness for synthetic aperture focusing applications by adjusting light field focus for non-planar scenes. A scene demonstrating synthetic focus of three depth levels is shown (see figure 14).

**Figure 14:** Three focus levels for a non-planar scene. Focus on backdrop (left), focus on person (middle), focus on fist holding ruler (right).

An application of particular interest in light field imaging is occlusion removal, which synthetic focusing can demonstrate robustness to. An example illustrating such robustness is shown (see figure 15 and figure 16).

**Figure 15:** Two focus levels of a non-planar scene with a significantly occluding hand. Focus on occluded area containing office chair and computer monitor (left) and focus on hand (right).

**Figure 16:** Rectified image set used to construct the rendered light fields in figure 15. Note that in each camera view, a significant portion of the monitor or the office chair is occluded.

### 5.3.2 Light field video

Light field video is another area of particular interest to our applications, and very little work has been done in light field imaging that exploits the temporal axis. We have been able to construct light field video by extracting each frame of video, and thus construct a light field for each frame.

The Raspberry Pi camera array has been able to capture synchronised video with $< 1$ centisecond of synchronisation disparity. This has been achieved by recording a stopwatch across all cameras and commanding video capture by sending synchronised SSH (see figure 17).

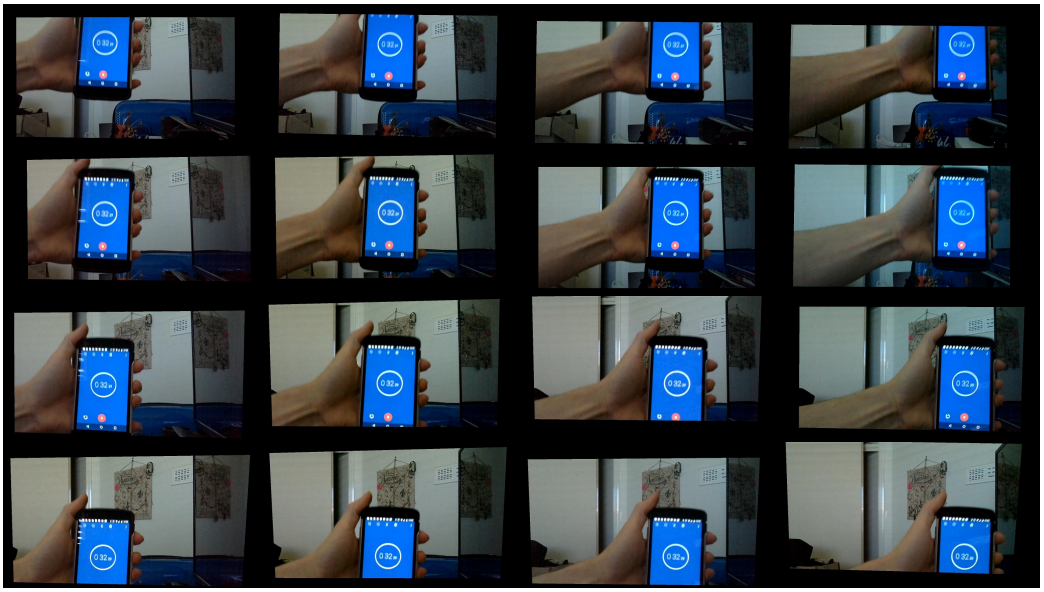


**Figure 17:** Recorded video has been shown to be synchronised to less than one centisecond. The camera modules exhibit significant motion blur over smaller timespans due to the shutter speed.

Additionally, we have demonstrated light field video that adjusts focus, to demonstrate aperture focusing with moving occluders (see figure 18).

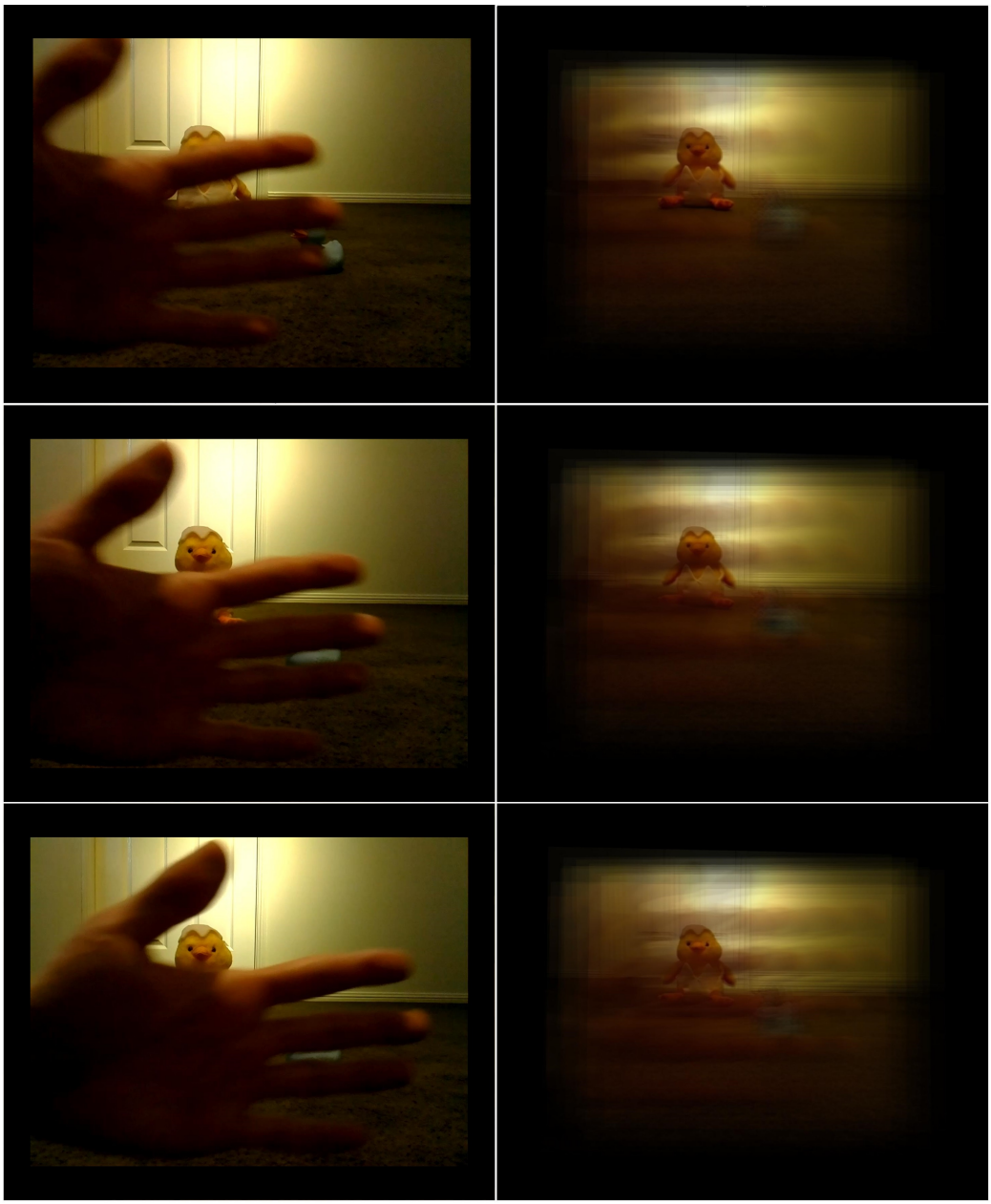


**Figure 18:** Three frames of video have been selected. The left-hand images show the frames from raw video captured by one of our camera modules. The right-hand images show the same frames, but rendered as a light field - the focus is on the stuffed toy.

# 6 Challenges and lessons learned

Several significant challenges were faced throughout the project, which we reflect on in this section as a reference for potential future work on the Raspberry Pi camera array.

## 6.1 Camera array power setup

A significant challenge which presented itself early in the project was the camera array's power setup. Initially, not all Raspberry Pi devices were powering on. This meant that at the start of the project, the cause of the failing Raspberry Pis needed to be identified and fixed. The power setup and lack of cable management meant that identifying the cause was a significant problem itself (see figure 19). A failing Raspberry Pi was also noted in Denham's technical report [3].

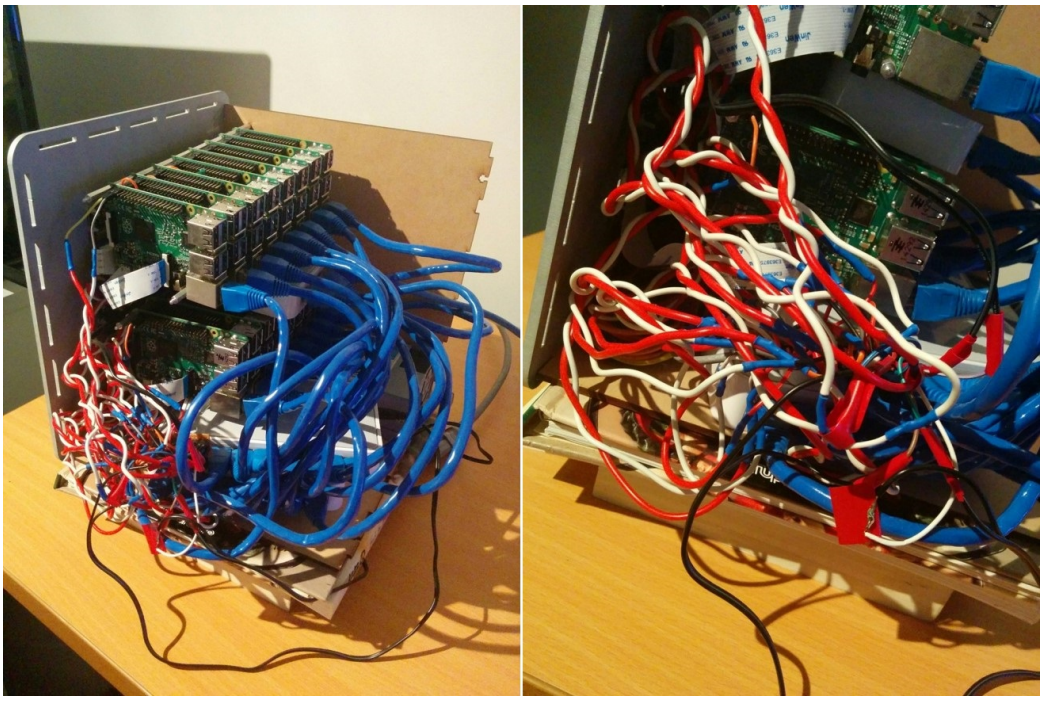


**Figure 19:** Raspberry Pi camera array power setup. Notice the poor cable management.

Solving the problem required the camera array to be pulled apart and re-constructed, ensuring all cables were plugged in securely. Unfortunately, the power setup was also extremely temperamental - power cables frequently came loose, and often required resoldering. As soldering was a new skill, training and practice was required.

Once all cameras were powering on and had the right software, such issues were relatively infrequent. For future work, it may prove beneficial to re-design the power setup of the array, so that cables are organised, safe and secure.

## 6.2 Camera array hardware setup

The way Raspberry Pis are positioned within the enclosure makes it difficult to access SD cards, power cables and camera board cables (see figure 20). If SD cards need re-imaging, or power or camera board cables need resitting, the entire array may need to be pulled apart.

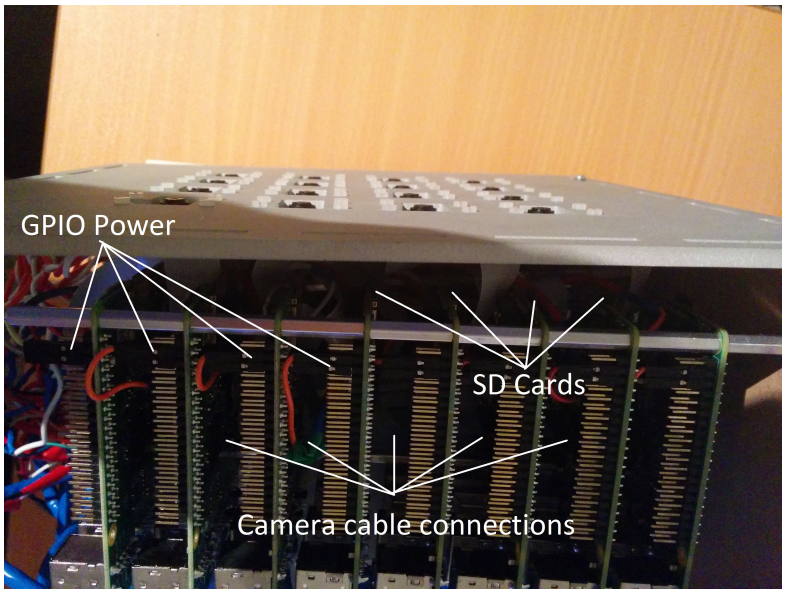



**Figure 20:** Top-down view of the camera array enclosure with the top removed. Key areas requiring regular access are labelled. It is difficult to access these areas without deconstructing the array of Raspberry Pis.

Again, once the camera array has all the necessary software and all cables are

secure, these issues do not cause serious problems. Nevertheless, we highly recommend redesigning the enclosure so that Raspberry Pis are not situated as they are presently.

## 6.3 Raspberry Pi software setup

In the early stages of the project, we wanted to demonstrate camera function via a live video stream to each Raspberry Pi camera module. It was discovered that the MATLAB support package for Raspberry Pi devices could achieve this. However, this requires Raspberry Pis to be set up with MATLAB's Raspbian distribution, since MathWorks does not provide the streaming software on its own. Setting up the new OS was a tedious process, because it required that the entire camera array be pulled apart so that SD cards could be retrieved, imaged, and put back into place.

After successfully testing camera functionality via live streams through MATLAB, our aim was to capture synchronised images and video. In Denham's original report, CompoundPi is suggested for synchronised capture. Installing the CompoundPi software package presented another challenge, since software needed to be downloaded and configured across Raspberry Pi devices. Since only one device can be connected to Wi-Fi at a time (we only have one Wi-Fi dongle), we attempted to install and configure CompoundPi on one device to start with. At this point, it was discovered that CompoundPi was incompatible with MATLAB's Raspbian distribution, so we decided to revert to a plain Raspbian distribution. Once we had one Raspberry Pi working with pure Raspbian and CompoundPi, we pulled apart the array to copy the image to the other 15 devices and reconstruct the array.

Later, we discovered that CompoundPi does not support synchronised video well. After some exploration, we found that the *SuperPutty* software package for Windows could synchronise commands to many devices via SSH. This could effectively replace CompoundPi so that we can capture synchronised images and video using the default capture commands `raspistill` and `raspivid`, without requiring any third-party software. This is our current software recommendation.

# 7 Conclusions

We have developed a non-metric calibration for light field acquisition, suitable for camera arrays, and flexible to viewing direction and planarity. The procedure is conceptually simple, yet demonstrates excellent results for light field applications, provided transform estimates are sufficiently accurate. Synthetic aperture focusing results compare well with other such setups (see figure 21).

**Figure 21:** Vaish et al's synthetic aperture focusing demonstration (top) and ours (bottom). The top images were adapted from Vaibhav Vaish, Bennett Wilburn, Neel Joshi, and Marc Levoy. Using plane + parallax for calibrating dense camera arrays. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–2. IEEE

It is important to recognise that the calibration procedure is non-metric, thus there is no inherent baseline for a correct result or reprojection error. However, our calibration is also unique in that it is a non-metric calibration whose accuracy can be easily calculated to a fine degree, by measuring positional consistency of features on the reference plane. We have achieved such a positional consistency of 99.957%.

The Raspberry Pi camera array is now significantly more capable as a light field camera, since camera poses are now retained by a new aluminium front-plate. Additionally, the camera array has demonstrated good performance with the new calibration procedure and synthetic aperture focusing applications. The camera array's potential for explorations in light field video has also been demonstrated, as synchronised capture and rendering of light field video has been achieved.

# 8 Future work

For any future work with the Raspberry Pi camera array, we highly recommend improvements be made in the design of the device before additional light field work takes place. Specifically, the power method and/or setup should be adjusted, so that power cables are secure, relatively accessible, and less exposed. Additionally, the enclosure should be redesigned so that SD cards, camera board ribbon cables, and power cables are accessible without pulling apart the device.

The camera array has successfully demonstrated effectiveness in light field applications such as synthetic aperture focusing and light field video, and is therefore fit for future work in plenoptic imaging. An area of particular interest is occlusion removal that exploits the temporal axis.

The nature of the Raspberry Pi camera array also means that there may be potential for parallel processing. Perhaps using OpenCV across devices, images could be rectified before being sent to a host machine for final processing into a light field. If the current calibration procedure is still used, it too could be implemented using OpenCV and run on each Raspberry Pi. This would allow the device to self-calibrate without the need for a host machine.

# 9 References


[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[2] Donald Gilbert Dansereau. *Plenoptic Signal Processing for Robust Vision in Field Robotics*. PhD thesis, The University of Sydney, 2014.

[3] Rafe Denham. Light field array camera. Technical report, Queensland University of Technology, June 2015.

[4] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

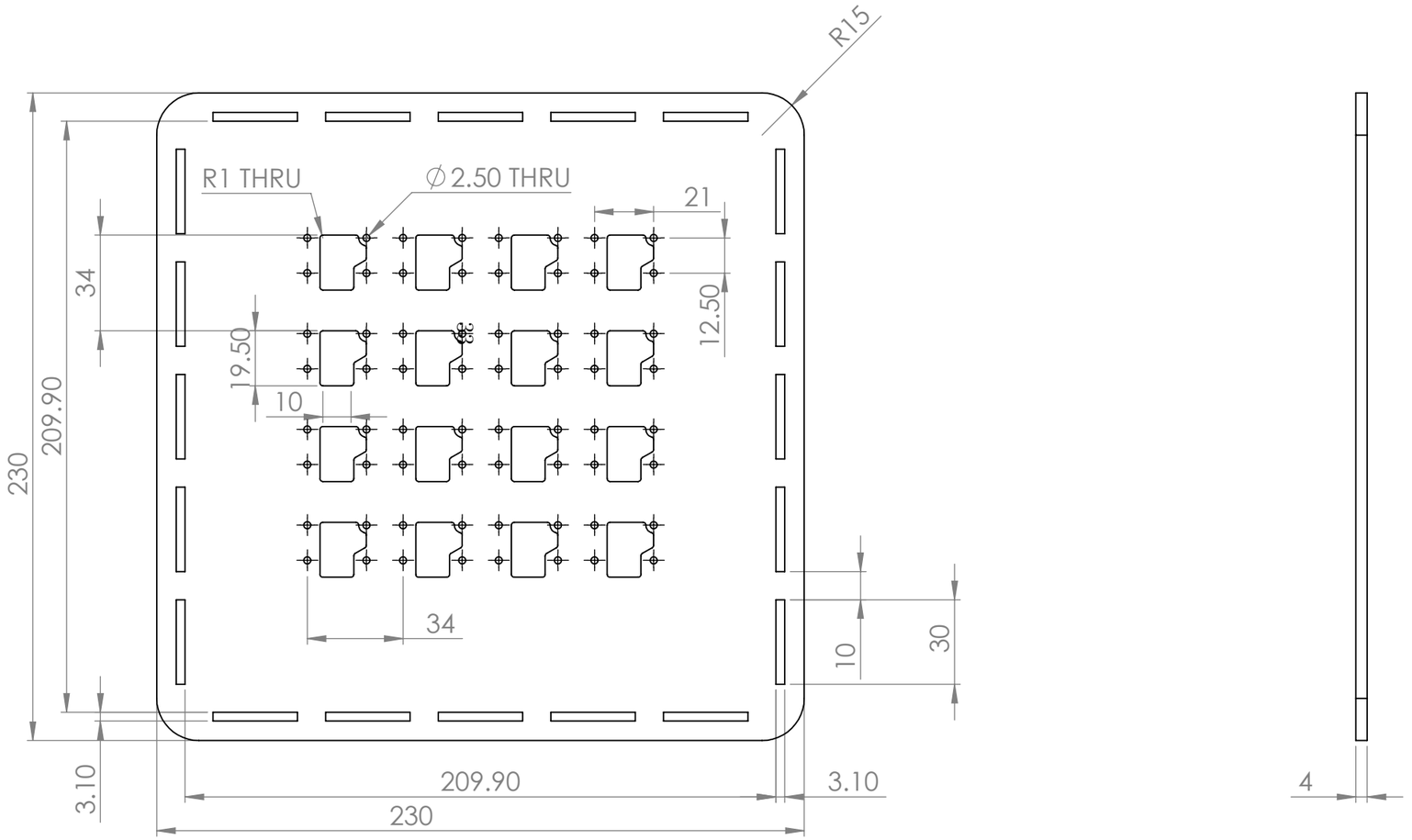
[5] Vaibhav Vaish, Bennett Wilburn, Neel Joshi, and Marc Levoy. Using plane + parallax for calibrating dense camera arrays. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–2. IEEE.

[6] Yichao Xu, Kazuki Maeno, Hajime Nagahara, and Rin-ichiro Taniguchi. Mobile camera array calibration for light field acquisition. *arXiv preprint arXiv:1407.4206*, 2014.

[7] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

# 10 Appendices

## 10.1 Replacement front plate




| | | | | | | | |
|---|---|---|---|---|---|---|---|
| UNLESS OTHERWISE SPECIFIED:<br>DIMENSIONS ARE IN MILLIMETERS<br>SURFACE FINISH:<br>TOLERANCES:<br>LINEAR:<br>ANGULAR: | | FINISH: | | | DEBURR AND BREAK SHARP EDGES | DO NOT SCALE DRAWING | REVISION |
| | NAME | SIGNATURE | DATE | | | TITLE: RaspPi Camera Array | |
| DRAWN | Steven Martin | | 11/04/2016 | | | | |
| CHK'D | | | | | | | |
| APPV'D | | | | | | | |
| MFG | | | | | | | |
| Q.A | | | | MATERIAL: Aluminium | | DWG NO. FrontPlate2 | A3 |
| | | | | | | | |
| | | | | | | | |
| | | | | WEIGHT: | | SCALE:1:2 | SHEET 1 OF 1 |

## 10.2 MATLAB Code

### 10.2.1 BuildTransformMatrixFromCalibrationImages

```
%%
% This file is an adaptation of MATLAB's 'Feature Based
% Panoramic Image Stitching' demonstration.
%
% Ashley Stewart
% Queensland University of Technology
% Australia
% 06/11/2016

%%
% BuildTransformMatrixFromCalibrationImages(calibrationSetDir)
%   calibrationSetDir:
%           The directory containing the calibration set
%
% Returns a series of geometric transformations that describe
% the relative transforms of images in a calibration set
% captured by a camera array, according to Ashley Stewart's
% planar calibration procedure for light field acquisition.
%
% The transformation matrix returned can be used to rectify
% future image sets captured by the same camera array.
%
% The transformation matrix will contain transforms in
% alphabetical order according to the calibration set
% filenames.

function tforms = BuildTransformMatrixFromCalibrationImages...
    (calibrationSetDir)

% Load calibration images
images = imageSet(calibrationSetDir);

% Initialise the first image and detect features
I = read(images, 1);
grayImage = rgb2gray(I);
points = detectSURFFeatures(grayImage);
```

```
[features, points] = extractFeatures(grayImage, points);

% Initialise transforms to the identity matrix
tforms(images.Count) = projective2d(eye(3));

% Iterate over remaining image pairs
for n = 2:images.Count

    % Store points and features for I(n-1)
    pointsPrevious = points;
    featuresPrevious = features;

    % Read I(n)
    I = read(images, n);

    % Detect and extract SURF features for I(n)
    grayImage = rgb2gray(I);
    points = detectSURFFeatures(grayImage);
    [features, points] = extractFeatures(grayImage, points);

    % Find correspondences between I(n) and I(n-1)
    indexPairs = matchFeatures(features, ...,
        featuresPrevious, 'Unique', true);
    matchedPoints = points(indexPairs(:,1), :);
    matchedPointsPrev = pointsPrevious(indexPairs(:,2), :);

    % Estimate the transformation between I(n) and I(n-1)
    tforms(n) = estimateGeometricTransform(matchedPoints,...
        matchedPointsPrev, 'projective',...
        'Confidence', 99.9, 'MaxNumTrials', 2000);

    % Compute T(1) * ... * T(n-1) * T(n)
    tforms(n).T = tforms(n-1).T * tforms(n).T;
end

%%
% At this point, all the transformations in |tforms| are
% relative to the first image.
%
% Using the first image as the initial reference does not
% produce the best result because it tends to distort most of
```

```
% the images. An improved result can be achieved by modifying
% the transformations such that the center of the scene is the
% least distorted. This is accomplished by inverting the
% transform for the center image and applying that transform to
% all the others.
%
% Start by using the |projective2d| |outputLimits| method to
% find the output limits for each transform. The output limits
% are then used to automatically find the image that is roughly
% in the center of the scene.
imageSize = size(I);  % all the images are the same size

% Compute the output limits  for each transform
for i = 1:numel(tforms)
    [xlim(i,:), ylim(i,:)] = outputLimits(tforms(i), ...
        [1 imageSize(2)], [1 imageSize(1)]);
end

%%
% Next, compute the average X limits for each transforms and
% find the image that is in the center. Only the X limits are
% used here because the scene is known to be horizontal. If
% another set of images are used, both the X and Y limits may
% need to be used to find the center image.

avgXLim = mean(xlim, 2);
[~, idx] = sort(avgXLim);
centerIdx = floor((numel(tforms)+1)/2);
centerImageIdx = idx(centerIdx);

%%
% Finally, apply the center image's inverse transform to all
% the others.
Tinv = invert(tforms(centerImageIdx));

for i = 1:numel(tforms)
    tforms(i).T = Tinv.T * tforms(i).T;
end

end
```

### 10.2.2 RectifyImagesViaTransforms

```
%%
% Ashley Stewart
% Queensland University of Technology
% Australia
% 06/11/2016

%%
% RectifyImagesViaTransforms(tforms, originalsDir, rectifiedDir)
%   tforms:
%       A transformation set built by
%       BuildTransformMatrixFromCalibrationImages
%   originalsDir:
%       The directory of the original images
%   rectifiedDir:
%       The directory to save rectified images to
%
% Rectifies an image set according to a series of
% transformations determined by analysing a calibration set
% via BuiltTransformMatrixFromCalibrationImages.
%
% Rectified image sets can be loaded via LFReadGantryArray.

function RectifyImagesViaTransforms(tforms, originalsDir, ...
    rectifiedDir)

% Changable parameters
scaleFactor = 0.33;
outputFormat = 'jpg';

% Load images
imageDir = fullfile(originalsDir);
images = imageSet(imageDir);

% Find the minimum and maximum output limits
imageSize = size(read(images, 1));
for i = 1:numel(tforms)
    [xlim(i,:), ylim(i,:)] = outputLimits(tforms(i), ...
        [1 imageSize(2)], [1 imageSize(1)]);
```

```
end

xMin = min([1; xlim(:)]);
xMax = max([imageSize(2); xlim(:)]);

yMin = min([1; ylim(:)]);
yMax = max([imageSize(1); ylim(:)]);

% Final dimensions of each image
width  = round(xMax - xMin);
height = round(yMax - yMin);
imageDimensions = [height width];

% Construct an empty reference plane view
referencePlaneView = imref2d(imageDimensions, [xMin xMax], ...
        [yMin yMax]);

% Calculate how many digits there are in the number of images
digits = numel(num2str(images.Count));

for imageNum = 1:images.Count
    I = read(images, imageNum);


    warpedImage = imresize(imwarp(I, tforms(imageNum),...
        'OutputView', referencePlaneView), scaleFactor);

    imwrite(warpedImage,strcat(rectifiedDir, '/',...
        sprintf(strcat('%0', num2str(digits), 'd'),...
        imageNum), '.', outputFormat));
end

end
```

### 10.2.3 CalculateRectifiedSetAccuracy

```
%%
% This file is an adaptation of MATLAB's 'Feature Based
% Panoramic Image Stitching' demonstration.
%
% Ashley Stewart
% Queensland University of Technology
% Australia
% 06/11/2016

%%
% CalculateRectifiedSetError(rectifiedSetDir)
%   rectifiedSetDir:
%           The directory containing the rectified set
%
% Returns the average pixel error calculated in the
% x and y directions for a rectified image set.
function [avgPixelError, pixelError] = ...
    CalculateRectifiedSetAccuracy(rectifiedSetDir)

% Load rectified images
images = imageSet(rectifiedSetDir);

% Initialise the first image and detect features
I = read(images, 1);
grayImage = rgb2gray(I);

points = detectSURFFeatures(grayImage);
[features, points] = extractFeatures(grayImage, points);

% Pixel error
pixelError = {};

% The pixel error beyong which we remove as an outlier
outlierError = 50;

% Iterate over remaining image pairs
for n = 2:images.Count
```

```
    % Store points and features for I(n-1)
    pointsPrevious = points;
    featuresPrevious = features;

    % Read I(n)
    I = read(images, n);

    % Detect and extract SURF features for I(n)
    grayImage = rgb2gray(I);
    points = detectSURFFeatures(grayImage);
    [features, points] = extractFeatures(grayImage, points);

    % Find correspondences between I(n) and I(n-1)
    indexPairs = matchFeatures(features, ...,
        featuresPrevious, 'Unique', true);
    matchedPoints = points(indexPairs(:,1), :);
    matchedPointsPrev = pointsPrevious(indexPairs(:,2), :);

    % Calculate pixel error
    thisError = abs(matchedPoints.Location - ...
        matchedPointsPrev.Location);

    % Remove outliers
    thisError(thisError(:,1) > outlierError,:) = [];
    thisError(thisError(:,2) > outlierError,:) = [];

    pixelError{n-1} = thisError;
end

% Find the average pixel error in x and y directions
avgPixelError = mean(vertcat(pixelError{:}));

end
```

### 10.3 Original Project Proposal


# Removing occlusions from light field data
# Research project proposal

Ashley Stewart

October 28, 2016

# Contents

# 1 Introduction

Computer vision is developing rapidly, empowering industrial robots, autonomous vehicles, surveillance cameras and other intelligent agents. In many applications, objects can obstruct a camera's view, preventing ideal function. This often occurs in surveillance and navigation during rainfall, hail or snow, among other scenarios. It also occurs in the surgical theatre for both external training videos and internal views such as those gathered by an endoscope. Occlusion removal enables the reconstruction of occluded segments to effectively 'see behind' such obstacles. This project proposes to implement a novel occlusion removal method using light field cameras.

Occlusion removal has already been demonstrated across a range of capture mediums (see section 2.3). However, most methods require either an elaborate, expensive, or stationary camera setup, only work in controlled environments, cannot be achieved quickly, automatically, or in real-time, or are designed to work only for images rather than video.

Light field cameras are a relatively new technology enabling the capture of light fields emanating from scenes. Little work has been done on occlusion removal specifically for light field cameras, with only one paper that we know of directly touching on the topic; one which is also inflexible for general applications [7]. A method to remove occlusions using light field cameras would be convenient for a number of reasons. Firstly, because light field cameras capture and produce a model of a light field, artificial views have a high level of metric accuracy, making them appropriate for robotic vision or applications where accuracy is important. The use of light field cameras also provides a layer of abstraction; an occlusion removal method that can be applied to one light field camera could potentially be applied to any light field camera. This means it does not necessarily require a custom or elaborate camera setup. Finally, light field video is an area that has not yet seen a great deal of research. It is possible that the properties of light field video present benefits for occlusion removal.

As well as contributing to research in occlusion removal and light field video, this project may build upon methods for object tracking, background segmentation (or *foreground detection*) and synthetic aperture focusing.

## 1.1 Aim

The aim of this project is to explore the properties of light field video signals and how they can be applied to achieve occlusion removal. Specifically, the project aims to answer the following research questions:

- How can the properties of light field video be exploited for occlusion removal?
- How can occlusion removal be applied to light fields?
- How can occlusion removal be applied to light field video?
- How can occlusion removal be achieved in real-time via a light field video stream?
- Can existing occlusion removal methods be made more robust for light field cameras?
- Could such occlusion removal methods be integrated into real-world applications using light field cameras?

A series of demonstrations have been proposed to illustrate a novel occlusion removal implementation for light field video, the results of which may lead to answers to some of the above research questions (see section 1.4.1). The effectiveness of the implementation can be tested via the proposed demonstrations using the following criteria:

- Robustness to occluders of varying shapes and sizes
- Robustness to occluders at varying distances
- Robustness to occluders moving at various speeds (video only)
- Robustness to occluded objects moving at various speeds (video only)
- Continued effectiveness during camera motion (video only)
- Computation time (video only)

Finally, in light of project findings, the project aims to make justified comments about the occlusion removal potential for several applications of interest (outlined in section 1.3).

## 1.2 Scope

Although occlusion removal techniques are useful across a wide range of applications, the project will only consider a small selection (see section 1.3). A series of demonstrations have been proposed which illustrate properties relevant to those applications.

Additionally, demonstrations are limited by the available resources. They will only be tested via the available Raspberry Pi Light Field Array Camera, which will be calibrated via Xu et al's procedure for mobile light field cameras [20]. Demonstration of occlusion removal in conditions such as rainfall, hail and snow or via an endoscope will also not be possible given the available time and resources. Instead, the demonstration design will illustrate key properties of those applications (e.g. small, fast-moving occluders to illustrate effectiveness during rainfall).

## 1.3 Applications

Occlusion removal has many applications, including surveillance, navigation, robotic vision, as well as surgical applications, among others. As previously stated, demonstrations have been designed to illustrate the possible effectiveness in a limited range of specific applications. Three main applications of interest have been described.

Scenarios where camera vision is hindered, such as during rainfall or in hail leads to applications of interest. If occlusion removal can be demonstrated in such scenarios, the implications would be significant. This would provide great benefits to automated and piloted navigation systems as well as surveillance technology.

Other applications of interest are found from scenarios where solid objects are obstructing an area of interest. There is particular interest for such technology in the surgical theatre. In a knee arthroscopy, for example, the surgeon must insert an arthroscope and endoscope into incisions on either side of the patient's knee. The area within the incision is then filled with fluid, and vision through the endoscope is obscured by floating debris, bubbles, and the arthroscope. An endoscope with occlusion removal capabilities would allow the surgeon to have increased awareness of the inside of the knee, and provide useful images for analysis and training.

The final application of interest involves the use of a light field camera in the surgical theatre, as a replacement for the traditional top-down camera. This would allow larger obstructions such as medical equipment to be removed from view for training purposes.

## 1.4 Outcomes

The proposed outcomes take the form of a scientific research article along with a series of demonstrations.

The research article will document the implementation, demonstration results, along with any findings related to the project aims. Comments will also be made about findings that shed light on the potential for occlusion removal in applications of interest, along with an outline for possible future work. The article may be published at a conference such as the Australasian Conference on Robotics and Automation. The implementation may also be made publicly available by integrating it into Dansereau's *Light Field Toolbox* for MATLAB [2] to be used for a range of applications.

### 1.4.1 Demonstrations

A series of practical demonstrations have been outlined, from which useful findings and results will be recorded. Additionally, justified comments will be made about the demonstration results and potential effectiveness for applications of interest. The demonstrations have been designed so that they can be achieved in the early and late stages of the project lifetime, and each are closely tied to major project milestones. These demonstrations will act on light fields, light field video, and a live light field stream.

The first demonstration will illustrate the effectiveness of calibration via digital refocusing. This demonstration can be achieved easily in the early stages of the project using the light field toolbox. The demonstration scene will be indoors and will contain objects of varying distances to show changing clarity as the focal plane is adjusted. Digital refocusing is a key feature of synthetic aperture focusing, a method which can be used to passively remove occlusions [7].

The second demonstration will occur later in the project, once an initial implementation of an occlusion removal algorithm has been developed. Ideally this will be Xu et al's camera selection algorithm [20], but if this is found to be impractical, it may need to be a variation of a background modelling or synthetic aperture focusing method. The demonstration can be assessed against some of the relevant criteria specified in the project aims (see section 1.1).

If the results of the second demonstration were successful, then the third demonstration will be pursued. This demonstration will work with light field video, and will thus require modifications to be made to the occlusion removal algorithm from the second demonstration (see section 4.3). This will allow the testing of full effectiveness criteria specified in the project aims (see section 1.1). There are risks associated with this demonstration that should be considered (see section 5).

If the results of the third demonstration were successful, a fourth and final demonstration will be pursued. This demonstration will work with a light field video stream, and thus may require optimisations to be made to the previous implementation. The same criteria as the previous demonstration will be used to test effectiveness, with a stricter cut-off for computation time. As with the previous demonstration, associated risks should be considered (see section 5).

# 2 Preliminary Literature Review

## 2.1 Introduction

Implementing occlusion removal for a light field camera is a complicated problem. In order to effectively implement and test a method, suitable camera hardware must be calibrated to capture accurate light fields. Several calibration methods exist, each with their own advantages and disadvantages. These must be examined in order to sensibly select an appropriate method. Additionally, existing occlusion removal literature must be analysed in order to identify possible starting points for the project. The following is an abridged extract from the full literature review available in appendix 7.3.

## 2.2 Camera calibration

### 2.2.1 Procedures for single camera calibration

The classic photogrammetry approach [11] solves the calibration problem for a single camera using a two-step procedure. The first step is to estimate the intrinsic and extrinsic camera parameters linearly via a closed-form solution. The second step uses nonlinear minimisation to obtain the final values, generally via the Levenberg-Marquardt algorithm [9].

Significant research has been achieved which further develops this two-step approach in different ways. Zhang, Heikkilä, and Tsai have made such developments, and their methods and source code are publicly available [6, 15, 23]. Zhang's method is particularly flexible, as a good estimate of the camera parameters can be made by capturing images of a planar pattern from at least two orientations (usually the pattern is a checkerboard - the corners of each square act as convenient feature points). Sun and Cooperstock provide an overview and empirical evaluation of the accuracy of these well-known methods [13].

### 2.2.2 Procedures for light field camera calibration

Single camera calibration approaches such as Zhang's are on their own unsuitable for light field cameras. Rigid transformations between pairs of viewpoints become inconsistent when the cameras are calibrated independently [20]. This inconsistency causes inaccurate estimations of the distances between each camera. Specialised calibration methods are therefore required.

Ueshiba and Tomita describe an extension to Zhang's method for multi-camera systems, which recover the rigid displacements between cameras, as well as the intrinsic parameters [16]. The handiness and flexibility of Zhang's method is maintained, as only two captures of a known planar pattern at different orientations are needed. The algorithm presents a homography matrix to act between the camera image and planar pattern. This leads to a measurement matrix with an unknown scale. This matrix can then be factorised to find camera and plane parameters. However, lens distortion is not considered in this method.

Svoboda et al. present a convenient method to calibrate multi-camera arrays, which also uses this factorisation approach [14]. However, instead of using a planar reference pattern, the calibration object is a freely moving bright spot, such as one generated by a laser pointer. This method was designed for virtual environment applications, and deals with a fixed volume and static camera system, and therefore is inflexible to a more dynamic camera system such as the one in our project.

Xu et. al. present a method to calibrate a mobile camera array in which the working volume and viewpoints need not be fixed [20], which also performs in Zhang's style by moving a checkerboard pattern. The method is flexible enough to allow the user to assign the number of viewpoints, and global optimisation of the intrinsic parameters is optional. The method also models radial distortion and achieves accurate results.

Dansereau et al. describe a method to calibrate a lenslet-based camera [1]. The light field camera's initial pose is estimated by taking the mean or the median of each image's pose estimate by following a conventional single camera approach [6, 15, 23]. Following this, the camera's intrinsic parameters are estimated through a closed-form solution for the camera's intrinsic matrix. The estimates are then refined through an optimisation such as those used in conventional approaches. Finally, distortion parameters are introduced and a full optimisation takes place. This method also introduces a practical

4D intrinsic matrix and distortion model which relate the indices of pixels to corresponding spatial rays. The source code for this method is publicly available from Dansereau's *Light Field Toolbox* for MATLAB.

Vaish et al. present a method that uses a plane plus parallax framework to calibrate large camera arrays [17]. Assuming all cameras lie on a plane parallel to the reference plane, camera positions can be recovered (such as in Ueshiba and Tomiba's approach [16]). This is achieved by measuring the parallax of a single scene point that is not on the reference plane. The light field can then be parameterised as a light slab (Levoy's two-plane parameterisation) [8]. Since the method assumes that all cameras are on precisely the same plane, and only calculates projection to a reference plane in advance of calibration, the accuracy is somewhat diminished for certain setups. This approach is therefore suitable for applications such as synthetic aperture photography, where planar cameras are commonly used.

## 2.3 Occlusion Removal

### 2.3.1 Applied to video from a single view

Occlusion removal can be achieved on still video (e.g. on surveillance cameras) by exploiting the temporal axis to perform *background subtraction* (also called *foreground detection*). A common background subtraction method for video data involves thresholding the error between estimates of images with and without occlusions. This generally involves computing a confidence level for each pixel with past and future frames. This allows a background model, and therefore an occlusion layer to be built. The numerous approaches to this problem differ in the type of background model used, and the procedure used to update the model [5, 10, 12, 19]. For example, Stauffer and Grimson's method models each background pixel as a mixture of Gaussians, updating them via an on-line approximation [12].

### 2.3.2 Applied to light fields

An occlusion removal method specific to light fields involves exploiting focusing techniques via a synthetic aperture. Given enough views and a sufficiently wide synthetic aperture, focussing on a region of interest can effectively blur out occluders in the reconstructed image to the point that they disappear. Vaish et al's plane plus parrallax calibration technique shows improved occlusion removal results via synthetic aperture focussing, compared to results from metric calibration techniques [17].

Vaish et al. have also explored occlusion removal through 3D reconstruction, using cost functions that are robust to occluders. This has been shown to improve the occlusion removal quality of synthetic aperture focusing [18]. Although this technique improves on the results of ordinary synthetic aperture focusing, its performance drops significantly in the case of complicated or severe occlusion.

To overcome these issues with severe occlusion, Yang et al. consider occluded object imaging a problem of light ray selection from optimal camera views [21]. An optimal camera selection algorithm and greedy optimisation is used to to propagate visible ray information from depth focus planes. This approach leads to a much clearer reconstruction of occluded objects.

### 2.3.3 Applied to camera array video

For the case where an array of cameras is used (though not necessarily a light field camera or cameras even on the same plane), synthetic aperture focusing can be combined with object detection and tracking algorithms. Joshi et al. uses a straightforward approach built from existing single camera tracking algorithms, which tracks moving objects through severe occlusions [7]. The method tracks objects with up to 70% occlusion on all cameras via detection aggregation across views.

Similarly, Yang et al. describe a method to remove occlusions from video via object tracking and synthetic aperture focusing [22]. A synthetic aperture imaging system is used to model precise locations of objects in a controlled scene (see Figure 1). Yang's method also enables the seamless interaction among detection, imaging and tracking modules via a hybrid framework, and introduces an improved synthetic aperture focusing method. However, its use is limited to controlled scenes such as in their setup.

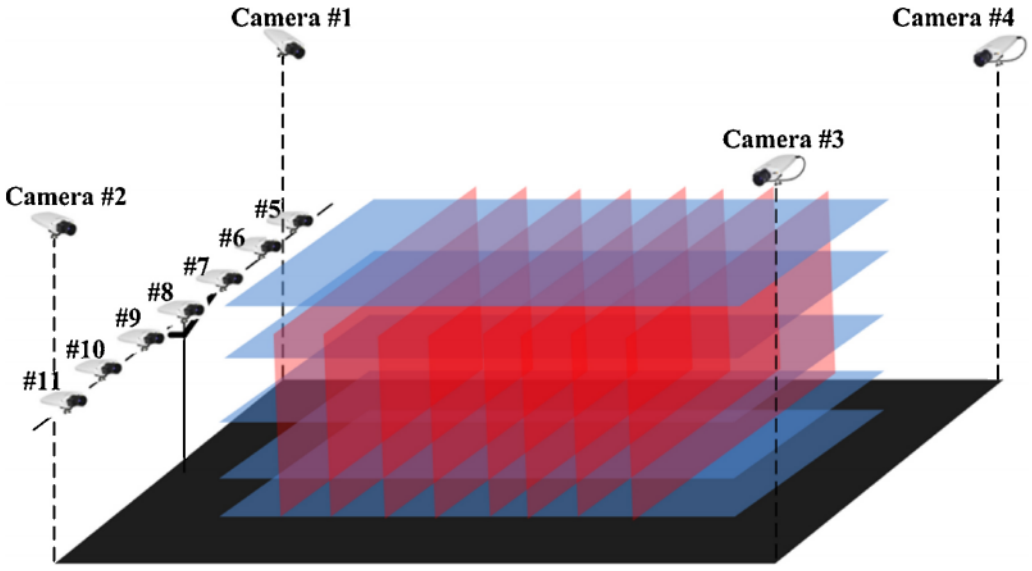



**Figure 1:** Hybrid synthetic aperture imaging system. Adapted from Tao Yang, Yanning Zhang, Xiaomin Tong, Xiaoqiang Zhang, and Rui Yu. A new hybrid synthetic aperture imaging model for tracking and seeing people through occlusion. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(9):1461–1475, 2013.

### 2.3.4 Applied to light field video

Research on occlusion removal on light field video is limited. However, it has been shown that some occlusion removal can be achieved through depth-velocity filtering. Edussooriya et al. have analysed the spectrum of a light field video corresponding to a Lambertian object with constant depth and velocity. They have showed that the object can be enhanced and modelled when occluded based on its depth and velocity via a 5D depth-velocity filter [4]. This method is applicable to a small set of applications, as it only works when objects are modelled at constant velocity and depth.

## 2.4 Review Conclusion

An overview of camera calibration literature and methods have been presented, which is necessary in order to sensibly choose a calibration procedure for the available light field camera. A similar overview for occlusion removal procedures has also been given, some of the methods of which will provide the foundations for our implementation.

It is clear from the review that current research in light field video is limited. The existing methods described are also relatively inflexible to more dynamic scenes. However, plenty of work has been done on occlusion removal in other contexts, and their concepts may be applied to produce a novel increment or new method. There is a great opportunity for a valuable addition to this area.

# 3 Resources required

In order to complete the project, several important physical and human resources are required. Fortunately, all such resources have been made available for the duration of the project.

A light field camera will be needed to perform the demonstrations and execute the proposed methodology. As such, a light field camera built as a final year undergraduate project has been made available (see figure 2). A summarised version of the full technical report [3] for the device has been provided in appendix 7.1. The device uses a 4x4 array of Raspberry Pi camera modules each connected to a Raspberry Pi 2 (RPi). The RPis are networked together, allowing them to act as essentially one device. Captured images can be sent through the network via an Ethernet cable to a server machine for further processing.

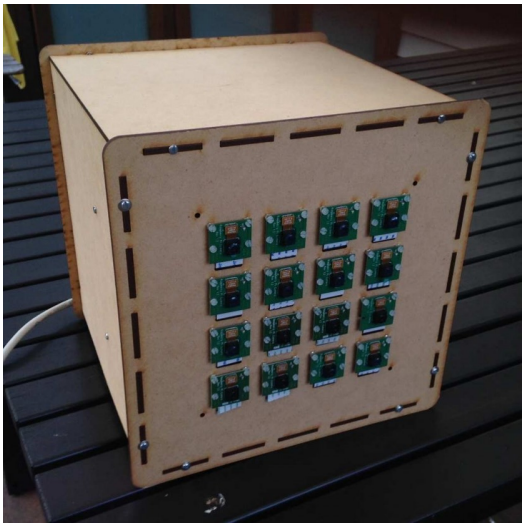


**Figure 2:** Raspberry Pi Light Field Array Camera

In addition to the light field camera, a workstation will be needed to interact with the camera and develop the necessary implementation. Workstation details are available in the MAPS risk assessment associated with the project.

For effective completion of the project, several human resources will be needed. The supervisory team will provide useful guidance and advice about the project's direction and problems encountered. A technical team is also

needed in case any hardware related issues occur. Their details are also provided from the MAPS risk assessment page.

# 4 General methodology

This section details the proposed methodology that the project will follow. It is important to note that the methodology ties very closely with the proposed demonstrations outlined in section 1.4.1.

## 4.1 Camera setup

The casing of the Raspberry Pi Light Field Array Camera is made from Medium Density Fibreboard (MDF), which is quite malleable. Unfortunately, this means that small forces applied to the frame can seriously affect calibration accuracy. To overcome this, a sturdier aluminium front plate has been designed to replace the MDF plate which will maintain calibration accuracy. The replacement plate can be considered the first milestone in the project which then enables the completion of camera calibration. The design for the new plate is available in appendix 7.2.

## 4.2 Calibration Procedure

Calibration is a necessary step in computer vision which ensures that captured images are consistent with the actual geometry of the scene. Many algorithms have been developed to calibrate conventional cameras and light field cameras, and are discussed in the literature review (see section 2). It is important to note that calibration algorithms each have their advantages and disadvantages. As such, the chosen method should suit the available light field camera, as well as any requirements necessary for occlusion removal.

The main objective of calibration in terms of enabling occlusion removal is metric accuracy. Accurate geometric information about the scene must be recovered if occlusions are to be actively removed. Passive removal of occlusions via synthetic aperture focusing should also be possible. Other objectives include simplicity and flexibility of the method due to time and budget limitations. Additionally, the calibration method must be suitable

for a mobile light field camera array, since that is the available camera's classification (rather than a lenslet-based or fixed camera setup).

Vaish et al. describe a method to calibrate a light field camera via a plane plus parallax framework [17]. This method yields clearer results for synthetic aperture focusing, but loses the metric information and camera parameters calculated via other methods such as Zhang's. This may limit the possible applications and occlusion removal features. In their paper, they indicate that metric calibration techniques are better suited to general camera configurations and in applications like metric reconstruction. This may include active occlusion removal, as it ideally involves accurate metric reconstruction of occluded segments.

The chosen procedure is a variation of Zhang's metric calibration method, adapted to suit mobile camera arrays with the goal of light field acquisition. It was proposed and tested by Xu et al. [20]. The method considers one of the cameras in the array a reference camera, and the others relative to the reference, rather than independent viewpoints. However, each camera is initially considered independent in order to obtain initial estimates of intrinsic and extrinsic parameters. Since the relative camera positions are theoretically constant between camera array poses, a full optimisation of camera parameters takes place in terms of each camera's median relative pose, rather than independent poses. This allows the rigid transformations between viewpoints to remain consistent.

An implementation of Dansereau et al's calibration method for lenslet-based plenoptic cameras [1] is available for the project via the light field toolbox. Xu et al's method will be implemented as an adjustment to this method. It will then be used to calibrate the camera array.

Calibration will be achieved by first capturing synchronised images from several camera poses with a planar pattern in view via the CompoundPi software package. CompoundPi provides a means for controlling cameras attached to Raspberry Pis in the same subnet, as recommended in the Raspberry Pi Array Camera documentation [3]. The images will then be downloaded to the workstation machine so that the intrinsic and extrinsic parameters can be calculated via Xu et al's method. Accuracy can then be tested by calculating the re-projection error in calibrated light field shots. Effectiveness can also be illustrated through digital refocusing, a simple and popular application of light field imaging, already handled by the light field toolbox.

## 4.3 Occlusion Removal

Since one of the aims of this project to *explore* the properties of light field video signals in the context of occlusion removal, the implementation of occlusion removal will likely change across the project lifetime as new findings are made. Therefore, a starting point based on existing research will be identified, along with possible alternatives and other features for possible experimentation.

Yang et al propose and demonstrate a method to remove occlusions via an optimal camera selection algorithm [21]. This is different from the more traditional background model [5, 10, 12, 19] and synthetic aperture focusing methods [7, 17, 18, 22]. It considers occlusion removal a problem of optimal light ray selection in order to produce a 'mosaic' of occluded segments using light rays from ideal viewpoints. This produces results that are much clearer than those obtained through synthetic aperture focusing (see figure 3). It has also not been demonstrated with light field video, nor optimised to take advantage of the properties of light field video signals. It therefore lends itself very well as a starting point for this project.

**Figure 3:** Occluded view from camera array (left), synthetic aperture focusing (middle) vs. light ray mosaic (right). Adapted from Tao Yang, Yanning Zhang, Xiaomin Tong, Wenguang Ma, and Rui Yu. High performance imaging through occlusion via energy minimization-based optimal camera selection. *International Journal of Advanced Robotic Systems*, 10, 2013.

Yang et al's method will serve as a starting point for demonstrating occlusion removal with the RPi light field camera. The first occlusion removal milestone will be to demonstrate occlusion removal on a previously captured light field using an implementation of the algorithm on MATLAB. Following this, the algorithm will be adapted to function with light field video and finally live light field video. See section 1.4.1 for more details about the pro-

posed demonstrations. If possible and practical, the demonstrations may be achieved in combination with synthetic aperture focusing, background modelling or object tracking features to demonstrate and document their effect on occlusion removal.

# 5 Risks

There are a number of potential risks associated with the project that relate to the implementation and available resources.

The CompoundPi software package is used to capture synchronised images and videos from the Raspberry Pi camera modules. Although a hardware level synchronisation is preferable, it is unfortunately not possible because the RPi board is not open-source. The synchronisation disparity using the software alternative has been measured to be between 20 and 20 milliseconds [3]. This is a significant disparity which could cause light fields to have blurred movement if objects in the scene are not completely still. It may also mean that recorded light field videos need to be manually sliced so that they have precisely the the same start and end points. A simple way to do this would be to have a running stopwatch visible in all scenes. Recorded video could then be sliced post-capture, ensuring the stopwatch is synchronised between all views.

The synchronisation disparity may also make live occlusion removal difficult. Obviously, the stopwatch solution cannot be applied live. If the synchronisation disparity is significant enough, the live demonstrations may have to be limited further. A possible alternative would be to attempt live occlusion removal with very slow movement, or completely still scenes. Small movements in still scenes may then be made before the scene becomes still again, so that the occlusion removal can act live on the updated scene. Another potential risk with live occlusion removal is related to the high network bandwidth necessary for image transfer, as well as as the computation time required for occlusion removal. In light of these risks, if the live occlusion removal demonstration is unsuccessful, it may be considered an area for future work. However, implementation details and proposed algorithms for live occlusion removal may still be provided.

# 6 Timeline

The figure below (figure 4) illustrates the proposed timeline for the project. The timeline is closely tied to the methodology (section 4) and demonstration outcomes (section 1.4.1). Each delivery is dependent on the previous delivery (therefore a Gantt style chart would be superfluous). The start date is the 13th of July, 2016, and the final thesis due date is the 28th of October, 2016. Exact dates of other project milestones are not included, as there is a degree of uncertainty. Instead, approximate delivery times have been marked.


**Figure 4:** Proposed timeline

# References


[1] D. G. Dansereau, O. Pizarro, and S. B. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1027–1034, June 2013.

[2] Donald Dansereau. *Light Field Toolbox v0.4*, 2015 (accessed June 10, 2016). `http://www.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4`.

[3] Rafe Denham. Light field array camera. Technical report, Queensland University of Technology, School for Electrical Engineering and Computer Science, 06 2015.

[4] Chamira US Edussooriya, Donald G Dansereau, Len T Bruton, and Panajotis Agathoklis. Five-dimensional depth-velocity filtering for enhancing moving objects in light field videos. *Signal Processing, IEEE Transactions on*, 63(8):2151–2163, 2015.

[5] Nir Friedman and Stuart Russell. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 175–181. Morgan Kaufmann Publishers Inc., 1997.

[6] Janne Heikkila and Olli Silvén. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112. IEEE, 1997.

[7] Neel Joshi, Shai Avidan, Wojciech Matusik, and David J Kriegman. Synthetic aperture tracking: tracking through occlusions. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[8] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.

[9] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[10] Christof Ridder, Olaf Munkelt, and Harald Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. In *Proceedings of International Conference on recent Advances in Mechatronics*, pages 193–199. Citeseer, 1995.

[11] Chester C. Slama, Charles Theurer, Soren W. Henriksen, and American Society of Photogrammetry. *Manual of photogrammetry*. American Society of Photogrammetry, Falls Church, Va, 4th edition, 1980.

[12] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[13] Wei Sun and Jeremy R. Cooperstock. An empirical evaluation of factors influencing camera calibration accuracy using three publicly available techniques. *Machine Vision and Applications*, 17(1):51–67, 2006.

[14] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence*, 14(4):407–422, Aug 2005.

[15] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987.

[16] Toshio Ueshiba and Fumiaki Tomita. Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 966–973. IEEE, 2003.

[17] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy. Using plane + parallax for calibrating dense camera arrays. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–2–I–9 Vol.1, June 2004.

[18] Vaibhav Vaish, Marc Levoy, Richard Szeliski, C Lawrence Zitnick, and Sing Bing Kang. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2331–2338. IEEE, 2006.

[19] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-time tracking of the human body.

*Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, 1997.

[20] Yichao Xu, Kazuki Maeno, Hajime Nagahara, and Rin-ichiro Taniguchi. Mobile camera array calibration for light field acquisition. *arXiv preprint arXiv:1407.4206*, 2014.

[21] Tao Yang, Yanning Zhang, Xiaomin Tong, Wenguang Ma, and Rui Yu. High performance imaging through occlusion via energy minimization-based optimal camera selection. *International Journal of Advanced Robotic Systems*, 10, 2013.

[22] Tao Yang, Yanning Zhang, Xiaomin Tong, Xiaoqiang Zhang, and Rui Yu. A new hybrid synthetic aperture imaging model for tracking and seeing people through occlusion. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(9):1461–1475, 2013.

[23] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.