A Novel Light Field Camera Calibration Algorithm Applied for Stereo-vision

Yu Ding[1], Junhui Deng[2]
[1] Shanghai American School
[2] Tsinghua University

Abstract

3D vision plays a fundamental role in many engineering and medical fields. Stereo-vision is one of the most popular method, which reconstructs depth information from binocular images. Camera calibration is crucial for the reconstruction geometry accuracy.

Conventional pinhole camera model represents each pixel as a ray through the optical center and the pixel. In reality, due to the manufacture error, the optical lense and the digital sensors have various distortions, the physical camera rays may not converge to a single point. Conventional calibration method, such as Zhengyou Zhang's algorithm, uses limited number of parameters to describe the intrinsic distortions, such as effective focal length, slant parameter, principle center and radial and tangential distortions, which are incapable of describing the complicated distortions or more general cameras, such as cell phone cameras and Lytro light field cameras.

Recently, the light field camera becomes popular, which associates each pixel with a ray in the physical world, and there is no optical center for the rays. This model is able to describe all types of complex intrinsic distortions, and represent any types of cameras. Furthermore, a light field camera can synthesize virtual pinhole cameras. But, so far there is no calibration algorithm for light field camera.

This work develops a practical method for light field camera calibration, which measures each camera ray individually based on Principle Component Analysis (PCA) and using a simple set up (including a linear rail, a LCD panel and a laser pointer), and stores the representations of all the rays. Comparing to conventional calibration method, the proposed method greatly simplifies the optimization process, and improves the precision.

The proposed calibration method is applied for a stereo-vision project to scan geometric objects in the real world. Our experimental results demonstrate that the proposed calibration algorithm can handle complicated geometric or textural features with high precision and outperforms the conventional calibration method.

keywords — stereo-vision, camera calibration, pinhole, ray, principle component analysis, least square

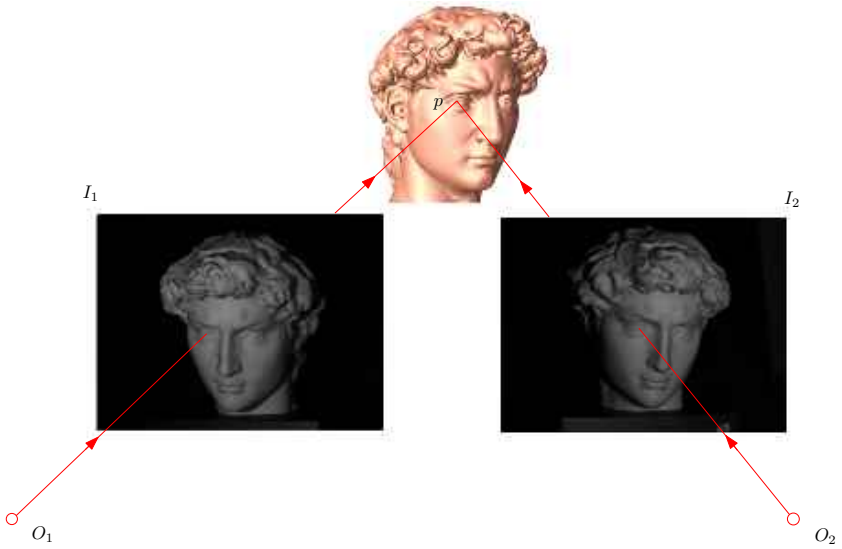
1 Introduction

3D vision plays a fundamental role in many fields, such as medical imaging, industrial inspection, 3D facial recognition in public security and recently for autopilot.

Stereo-Vision and Auto-pilot Stereo vision aims at recovering 3D geometry from a pair of 2D images captures from cameras in different view angles. Stereo-vision is the center for computer vision for decades, and becomes more and more important recently, especially for auto-pilot.For example, Tesla, refresh the definition of driving cars should by human. Beside of the traditional pilot system [2], Tesla also include a new auto-pilot system that free peoples hand. The CEO of Tesla, Elon Musk claim the new Auto-pilot would introduce a new method of detecting objects that enhance the accuracy of AI. In practice, this new system also presents an ideal dodging ability.In the recent Q1 2021 report, Tesla provide an incredible number of car accident during he 1st quarter of 2021:"In the 1st quarter, we registered one accident for every 4.19 million miles driven in which drivers had Autopilot engaged. For those driving without Autopilot but with our active safety features, we registered one accident for every 2.05 million miles driven."

[1]Referring back to the goal of Tesla,"to be the safest cars in the world".

In order to improve the safety of auto-pilot and improve the quality of the services based on 3D-vision techniques, it is crucial to improve the accuracy and efficiency of stereo-vision algorithms, especially camera calibration.



(a). Triangulation principle

(b). Reconstructed point cloud

Figure 1: Stereo-vision triangulation principle for reconstruction. The left and right camera images are I_1 and I_2, the optical centers are O_1 and O_2 respectively.

Figure 2: A Tesla car accident caused by auto-vision error

Light Field Camera Recently, light field camera [13, 12, 7, 3, 19, 16] becomes more and more popular. In principle, a light field camera captures the intensity of all the rays in the physical space, which gives much more comprehensive information about the object. The light field includes the pinhole camera images from all possible view angles. Therefore it allows users "shoot first, and focus later". Furthermore, light field camera has great potential for visualization, virtual reality and augmented reality applications. However, there is no mature calibration method for light field cameras, therefore there is no stereo-vision system based on this type of camera.

Challenges Conventional stereo-vision methods are based on pinhole camera model, which associates each pixel with a ray and all the rays converge to the single optical center. Hence the model only uses a few parameters to describe the relative position and orientation of the camera, and the intrinsic distortions, including effective focal lengths, slant parameter, principle center and radial and tangential distortions. However, in reality, many types of cameras can not be modeled as pinhole cameras, such as the cell phone camera and the Lytro light field camera, furthermore, due to the manufacture inaccuracy, the distortions of optical lenses and digital sensors are very complicated. Conventional pinhole camera model can not fully describe all the distortions. Furthermore, conventional calibration process is inaccurate due to the distortion of the calibration target board, the sparsity of the feature points on the board and so on. Theoretically, conventional camera calibration algorithm is equivalent to a non-linear optimization problem, which may get stuck at the local optima. Therefore the conventional calibration method doesn't meet the requirements in many practical applications nowadays.

Our Solutions In order to tackle these difficulties, in this work, we adopt the more advanced light field camera model and develop a calibration algorithm for it.

The light field camera model treats a camera as a set of rays, each pixel represents a ray in the physical world. These camera rays unnecessarily merge to a single point, therefore the concept of optimal center is unnecessary. The camera rays can be interpolated and reorganized to synthesize pinhole cameras and generate images with variant focal lengths.

The calibration of a light field camera means to measure each individual ray associated with the corresponding pixel. Each ray requires 4 parameters to describe, and the total number of calibration parameters for a light field camera is 4 times the number of pixels. Therefore there could be millions of parameters for a light field camera, much more than that of a pinhole camera. Hence a light field camera can handle all types of distortions and greatly improves the accuracy.

As shown in Fig. (3), we design special mechanical device for the calibration purpose. We mount our stereo-camera system on a linear rail, and use a LCD panel as the calibration target board. By displaying special patterns on the LCD, we can obtain the intersection point between the camera ray with the plane of the LCD panel.By sliding the camera system, or equivalently the LCD panel along the rail, we obtain multiple intersection points of each camera ray with different LCD panel planes. We compute the camera ray using the Principle Component Analysis. Then the calibrated camera system is applied for capturing 3D shapes with complicated geometry using stereo-matching algorithm.

Contributions In this work, we propose a novel algorithms for light field camera calibration, which improves the accuracy for stereo-vision. Our contributions can be summarized as follows:

- Replace the pinhole camera model by a light field camera model in order to represent complicated distortions and improve the system accuracy.
- Develop a light field camera calibration algorithm based on Principle Component Analysis. The theoretic formulation is much simpler than conventional camera calibration model.
- Design the hardware setup for the calibration method, which includes a linear rail, a laser pointer and a LCD panel. The conventional calibration target board is replaced by the LCD panel to reduce the physical distortion and increase the number of markers;
- Apply the proposed calibration algorithm for stereo-matching and geometric reconstruction. The experimental results demonstrate the system can capture real objects with complicated geometric and textural characteristics, and improves the reconstruction accuracy.

The paper is organized as follows: in section 3, we review the conventional pinhole camera model and the classical Zhang's calibration algorithm; in section 4, we introduce our novel ray field camera model and the calibration algorithm; in section 5, we explain the stereo-matching and geometric reconstruction algorithm. The experimental results are reported in section 7, and the work is concluded in section 8.

2 Previous Works

Camera calibration refers to the problem of finding the mapping between the 3D world and the image plane. There has been many research works on camera calibration [8]. In most of the algorithms, some set of features are extracted from images, the intrinsic camera parameters as well as camera pose and orientation (extrinsic camera parameters) are estimated by a minimization of an over all cost function.

In many existing calibration techniques, good estimates for external and internal camera parameters are first obtained by a pinhole camera model neglecting lens distortion. Then distortion calibration is performed while holding the other parameters fixed [4, 10, 21]. Many calibration techniques use both nonlinear minimization and closed form solutions as in [9, 5].

Some methods [4, 5] extract the 3D line segments from the images, by the fact that protective transformations preserve lines, the linear constraints can be added to the energy, and the camera parameters can be estimated by the optimization. Some methods [15] rely on point correspondences. Given a set of 3D points, the associated epipolar and trilinear (among three cameras) constraints are arranged into a tensor, the distortion parameters can be optimized by minimizing the reprojection error in an iterative manner. The method in [6] a directly finds the camera calibration parameters by incorporating lens distortion. Recent variational approach [18] proposes a joint region-based image segmentation and simultaneous 3D stereo reconstruction technique.

Most existing methods are based on pinhole camera model, and use Taylor expansions to approximate the non-linear lens distortions. Unfortunately, the light field camera can not be covered by these mathematical models, therefore can not be calibrated by these algorithms either. This motivates us to develop the current algorithm to calibrate more general cameras.

3 Pinhole Camera Model and Calibration

Stereo-vision and Triangulation Principle As shown in Fig. (1), stereo-vision systems mimic human eyes to reconstruct 3D geometry from two planar images captures by the left and the right eyes. Our stereo-vision system is shown in Fig. (3), two gray scale Flir cameras and one IDS color camera are mounted on a koolehaoda camera rail, and the camera rail is mounted on a linear stage. In order to help stereo-matching, structured light techniques are commonly applied as well. As shown in Fig. (3), a digital projector is added to the system, which projects structured lights with special patterns to improve the matching efficiency and accuracy. As shown in

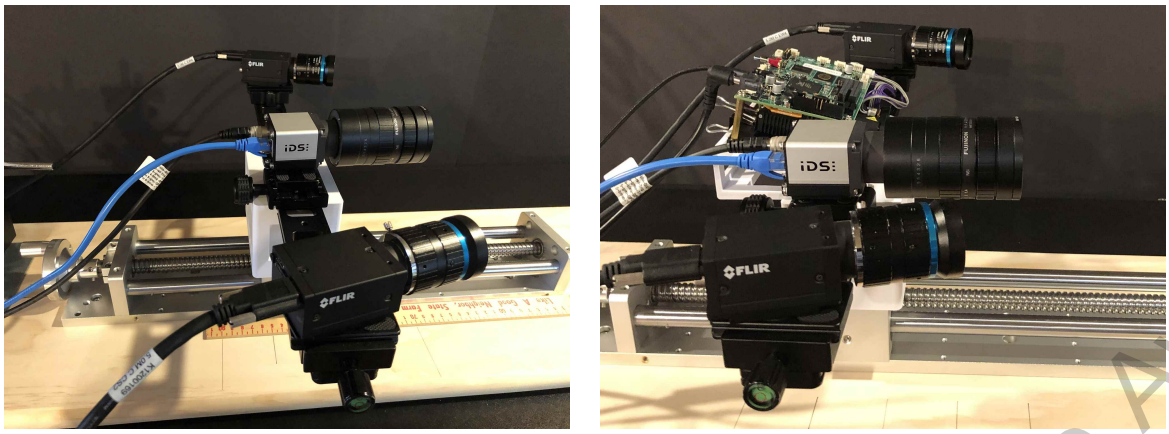


Figure 3: Our experimental stereo-camera system withmultiple cameras and a digital projector.

Fig. (1), both gray scale cameras capture the images of the 3D object simultaneously. Each pixel in the left image is matched to a pixel in the right image. Each pixel represents a ray through the optical center and the pixel. Two rays intersect at a 3D point on the object. This is the so-called *triangulation* principle for stereo-vision. For the purpose of geometric reconstruction by stereo-matching, the cameras need to be calibrated to obtain corresponding geometric relations and internal distortions. For example, the precision of the intersection point depends on the geometric accuracy of the camera rays, which are obtained by the calibration process. In the following, we briefly review the mathematical model of pinhole camera and the conventional Zhengyou Zhang's calibration method [21].

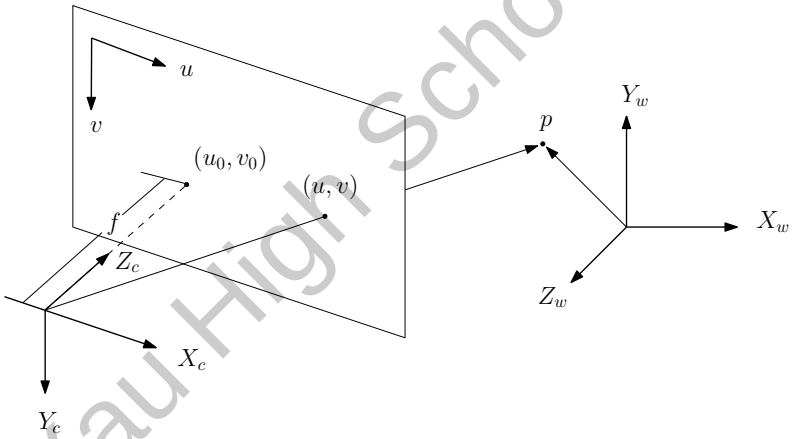


Figure 4: Pinhole camera model.

3.1 Pinhole Camera Mathematical Model

Conventional cameras are modeled as a *pinhole* camera, where each pixel captures the light coming through a spacial ray, and all the rays intersect at a unique common point, the so-called *optical center*.

World to Camera Coordinate Transformation Fig. 4 shows the mathematical model of a pinhole camera. (X_w, Y_w, Z_w) is the world coordinates, (X_c, Y_c, Z_c) the camera coordinates, (u, v) image coordinates. The optical center is the origin of the camera coordinate frame. A point p in the world coordinate system is (X_w, Y_w, Z_w), in the camera coordinate system is (X_c, Y_c, Z_c), then the transformation from the world coordinates to the camera coordinates is given by

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T. \tag{1}$$

where R is the rotation matrix from the world coordinate system to the camera coordinate system, T is the translation vector.

Protective Transformation The projection from the camera coordiantes to the camera projective coordinates (without considering distortions) are given by:

$$\begin{cases} x_c &= fX_c/Z_c \\ y_c &= fY_c/Z_c \end{cases} \tag{2}$$

where f is the focal length.

Distortions Model In practice, the lense of the camera introduces distortions, hence the imaging process does not satisfy the ideal pinhole camera model. In the calibration process, the distortions need to be carefully considered. In general, the distortion include both radial distortion $(\delta_{xr}, \delta_{yr})$ and tangential distortion $(\delta_{xt}, \delta_{yt})$. The radial distortion $(\delta_{xr}, \delta_{yr})$ are represented as

$$\begin{cases} \delta_{xr}(x_c, y_c) &= x_c(k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots), \\ \delta_{yr}(x_c, y_c) &= y_c(k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots), \end{cases} \tag{3}$$

where $r^2 = x_c^2 + y_c^2$, $k_1, k_2, k_3, \cdots$ are the radial distortion parameters. The tangential distortion $(\delta_{xt}, \delta_{yt})$ can be represented as

$$\begin{cases} \delta_{xt}(x_c, y_c) &= 2p_1 x_c y_c + p_2(r^2 + 2x_c^2), \\ \delta_{yt}(x_c, y_c) &= p_1(r^2 + 2y_c^2) + 2p_2 x_c y_c, \end{cases} \tag{4}$$

where p_1, p_2 are tangential distortion parameters.
After considering the camera distortion, the distorted camera projective coordinates (x_d, y_d) of the point p can be represented as

$$\begin{cases} x_c^d &= x_c + \delta_{xr}(x_c, y_c) + \delta_{xt}(x_c, y_c) \\ y_c^d &= y_c + \delta_{yr}(x_c, y_c) + \delta_{yt}(x_c, y_c) \end{cases} \tag{5}$$

After the projective transformation, the camera image coordinates of the point p can be represented as

$$\begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c^d \\ y_c^d \\ 1 \end{bmatrix} = A \begin{bmatrix} x_c^d \\ y_c^d \\ 1 \end{bmatrix} \tag{6}$$

where f_u, f_v are the effective focal lengths along u and v directions respectively, s is the *slant parameter* of the coordinate axis, (u_0, v_0) are the coordinates of principle point, the intersection point between the optical axis of the camera and the image plane.

Mathematical Model In practice, the mathematical model for camera and projector can be described using the following pipeline:

$$(X_w, Y_w, Z_w) \xrightarrow{\varphi_1} (X_c, Y_c, Z_c) \xrightarrow{\varphi_2} (x_c, y_c) \xrightarrow{\varphi_3} (x_c^d, y_c^d) \xrightarrow{\varphi_4} (u_c, v_c)$$

The top row shows the image formation process of the camera, the bottom row shows the image formation of the projector.

1. The map $\varphi_1 : (X_w, Y_w, Z_w) \to (X_c, Y_c, Z_c)$ transforms from the *world coordinates* to the *camera coordinates*, which is a rotation and a translation, as shown in Eqn. (1);
2. $\varphi_2 : (X_c, Y_c, Z_c) \to (x_c, y_c)$ is the pinhole camera projection, maps from *camera coordinates* to the *camera projective coordinates*, as shown in Eqn. (2);
3. $\varphi_3 : (x_c, y_c) \to (x_c^d, y_c^d)$ is the camera distortion map in Eqn. (5), transforms from *camera projective coordinates* to the *distorted camera projective coordinates*, the distortion includes both *radial distortion* Eqn. (3) and *tangential distortion* Eqn. (4);
4. $\varphi_4 : (x_c^d, y_c^d) \to (u_c, v_c)$ is the projective transformation in Eqn. (6), which maps from the *distorted camera projective coordinates* to the *camera image coordinates*.

3.2 Pinhole Camera Calibration

Camera calibration aims at find all the parameters of the camera, including extrinsic parameters: rotation R_c, translation T_c; intrinsic parameters: effective focal lengths f_u, f_v; slant parameter s, principle center (u_0, v_0); and the distortion parameters: radial distortion parameters k_1, k_2, k_3, tangential distortion parameters p_1, p_2. In practice, intrinsic parameters also include distortion parameters. Generally, k_3 and s are small enough, and usually treated as 0's. We denote all the extrinsic and intrinsic parameters as

$$\mu = (R_c, T_c, f_u, f_v, s, u_0, v_0),$$

and all the distortion parameters as

$$\lambda = (k_1, k_2, k_3, p_1, p_2).$$

Calibration Board The target board for calibration is a planar plate with a checker board pattern. The corners of the checkers are detected using corner detector [**?**]. The top left corner is the origin of the world coordinates system, the horizontal and vertical directions are along X_w and Y_w axis, the normal to the target plane is the Z_w axis.

During the calibration process, each time we fix the position of the target board plane π, the local coordinates system of the target plane is treated as the world coordinates system, the plane equation is $Z_w = 0$, the corners of the checkers are known, denoted as

$$\{(X_w^1, Y_w^1), (X_w^2, Y_w^2), \cdots, (X_w^n, Y_w^n)\},$$

the image coordinates of each star center is captured

$$\{(u_1, v_1), (u_2, v_2), \cdots, (u_n, v_n)\}.$$

From the mapping $\{(X_w^i, Y_w^i)\} \rightarrow \{(u_i, v_i)\}$, by using the method in [21], we can estimate the extrinsic and intrinsic parameters μ.

Parameters Estimation The image formation mapping, also called the *forward projection*, depends on the extrinsic and the intrinsic parameters,

$$\varphi_{\mu,\lambda} : (X_w, Y_w, Z_w) \rightarrow (u, v), \quad \varphi_{\mu,\lambda} = \varphi_4 \circ \varphi_3 \circ \varphi_2 \circ \varphi_1.$$

The calibration problem is formulated as an optimization problem:

$$\min_{\lambda,\mu} E(\lambda, \mu) = \min_{\lambda,\mu} \sum_{i=1}^{n} \|\varphi_{\lambda,\mu}(X_w^i, Y_w^i) - (u_i, v_i)\|^2.$$

we first use Zhang's algorithm [**?**] to estimate μ, the extrinsic and intrinsic parameters; then fix μ, optimize $E(\lambda, \mu)$ with respect to λ; third, fix λ and optimize $E(\lambda, \mu)$ with respect to μ. By alternating optimizations, we can reach the optimum

$$(\lambda^*, \mu^*) = argmin_{\lambda,\mu} E(\lambda, \mu).$$

The optimization can be carried out using gradient descend algorithm:

$$\frac{\nabla E}{\partial \lambda} = \left[\frac{\partial E}{\partial k_1}, \frac{\partial E}{\partial k_2}, \frac{\partial E}{\partial k_3}, \frac{\partial E}{\partial p_1}, \frac{\partial E}{\partial p_2}\right]^T.$$

4 Light Field Camera Model and Calibration

In this work, we adapt a general camera model, the *light field camera*. As shown in Fig. (5), each pixel (i, j) receive the light signal along a ray $\gamma(i, j)$. A light field camera can be treated as a set of rays $\{\gamma(i, j)\}$ parameterized by the image pixels.

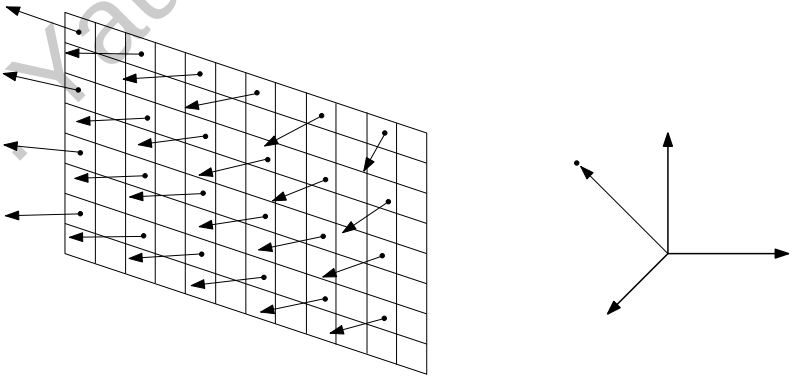
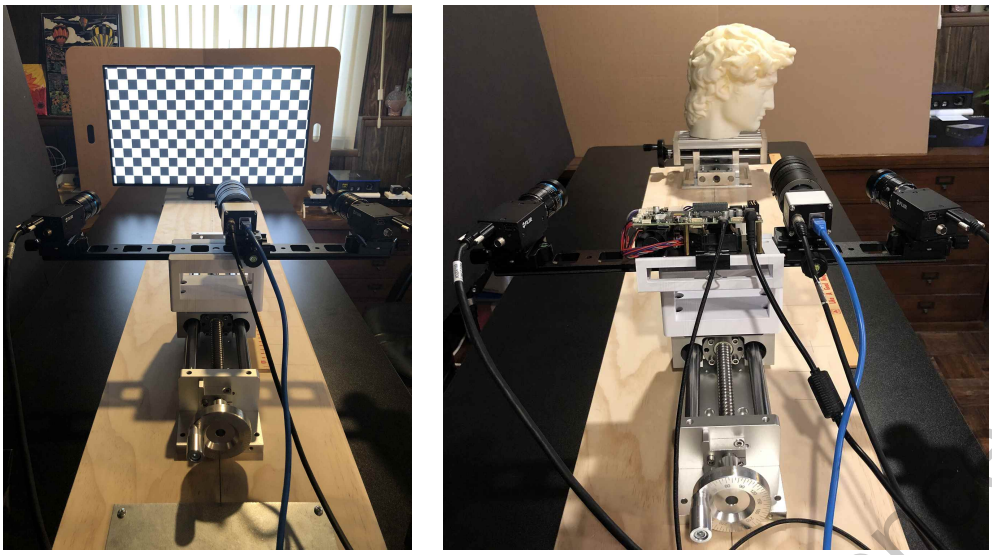


Figure 5: A light field camera model, each pixel represents a ray in the physical world independently.

In the pinhole camera model, all the rays intersect at the optical center. In contrast, in the light field camera model, the rays are independent, they may or may not share any common point.Therefore, the light field model doesn't require the concept of optical center, and stores the rays of all pixels instead of a few parameters. The light field camera model is much more general, and much more accurate than the conventional pinhole model.



(a). calibration mode. (b). capture mode.

Figure 6: The calibration mode and the capture mode for our stereo-camera system.

System Setup Fig. (6) illustrates our hardware system setup for the light field camera calibration. The camera system is mounted on a linear rail (sliding table) and can freely slide along it. The rail is orthogonal to a LCD panel. Different patterns are displayed on the LCD screen and captured by both left and right cameras. The world coordinates system is set as follows: the LCD panel is the plane of $Z_w = 0$, the center of the LCD panel is the origin of the world, the horizontal and vertical directions of the LCD panel are the X_w and Y_w directions, and the Z_w direction is along the linear rail.

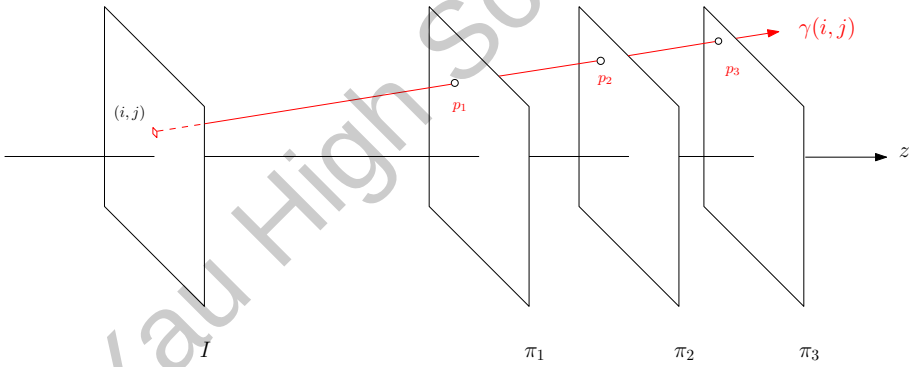


Figure 7: Line fitting based on principle component analysis.

We can either slide the camera system along the linear rail as shown in Fig. (6), or equivalently move the LCD panel as shown in Fig. (7). In our experiments, we move the camera system to different planes π_1, π_2 and π_3 respectively. The equations for the planes are:

$$\pi_1 : Z_w = 750, \quad \pi_2 : Z_w = 800, \quad \pi_3 : Z_w = 850,$$

where the unit is millimeter. Suppose the ray through the (i, j)-th pixel of one light field camera is $\gamma(i, j)$. As shown in Fig. (7), $\gamma(i, j)$ intersects π_k at p_k, $k = 1, 2, 3$. We estimate $\gamma(i, j)$ from $\{p_k\}$ using the Principle Component Analysis (PCA) method.

Comparing to the conventional method, this method has many advantages: a). it replaces the target board by the LCD panel, the panel has much less physical distortion than the board; b). each pixel on the LCD panel can be used as the markers for the calibration, the number of pixels is much more than the that of the corners of the checker board in the conventional method; c). the mathematical formulation is much simpler than that in the conventional method, the PCA method finds the global optimum instead of the local optimum in the conventional nonlinear optimization; d). the ray associated for each pixel can be estimated independently, the whole algorithm is intrinsically parallel; e). the proposal calibration method has much more parameters than the conventional method and can represent complicated distortions, hence greatly improve the accuracy.

Line Fitting Based on Principle Component Analysis Suppose we are give a set of points $\{p_1, p_2, p_3, \ldots, p_k\}$ in $\mathbb{R}^3$, our goal is to find a best fitting linear space π, which is called the principle analysis of the point set. Each point p_i is projected to the linear space to obtain the projection $\tilde{p}_i$. Our goal is to minimize the approximation error,

$$E(\pi) := \sum_{i=1}^{k} \|p_i - \tilde{p}_i\|^2, \tag{7}$$

which is a least square problem. For 0 dimension, the best fitting point c is obtained by optimizing

$$\min_c \sum_{i=1}^{k} \|p_i - c\|^2 = \min_{i=1}^{k} \langle p_i - c, p_i - c \rangle.$$

By differentiating the energy, we obtain $2\sum_{i=1}^{k}(p_i - c) = 0$, and the center formula

$$c = \frac{1}{k}\sum_{i=1}^{k} p_i.$$

Then we shift every p_i to $p_i - c$. For the best fitting line $\gamma(t)$, it is represented as

$$\gamma(t) = c + td, \quad d \in \mathbb{S}^2,$$

where d is the unit direction vector. The projection of the vector $p_i - c$ to the line is $\langle p_i - c, d\rangle d$, and the error vector is

$$e_i := (p_i - c) - \langle p_i - c, d\rangle d.$$

Since the center c is fixed, minimizing the length of the error vector e_i is equivalent to maximizing the length of the projection component $\langle p_i - c, d\rangle d$. The signed length of the projection component is given by

$$\langle p_i - c, d\rangle = d^t(p_i - c) = (p_i - c)^t d,$$

then the least square problem is formulated as maximizing the projection component:

$$\max_{d\in\mathbb{S}^2} \sum_{i=1}^{k} d^T(p_i - c)(p_i - c)^T d = \max_{d\in\mathbb{S}^2} d^T \left[\sum_{i=1}^{k}(p_i - c)(p_i - c)^T\right] d = \max_{d\in\mathbb{S}^2} d^T \Sigma d, \tag{8}$$

where Σ is the covariance matrix. By definition, for any vector $v \in \mathbb{R}^3$,

$$v^T \Sigma v = \sum_{i=1}^{3} \langle p_i - c, v\rangle^2 \geq 0.$$

Suppose we assume $p_i - c$ span the whole $\mathbb{R}^3$, and v is not equal to 0, then the $v^T\Sigma v$ is positive. This shows the covariance matrix Σ is positive definite. Then it can be decomposed as

$$\Sigma = O^T \Sigma O, \quad O^T O = Id,$$

where O is a rotation matrix, formed by the eigen vectors e_i's of Σ,

$$O = (e_1, e_2, e_3), \quad Oe_i = \lambda_i, \quad \lambda_1 \geq \lambda_2 \geq \lambda_3.$$

The eigen vectors form an orthonormal frame of $\mathbb{R}^3$, $\langle e_i, e_j\rangle = \delta_{ij}$. Then the direction vector d can be represented as $d = xe_1 + x_2e_2 + x_3e_3$, hence

$$d^t\Sigma d = \lambda_1 x_1^2 + \lambda_2 x_2^2 + \lambda_3 x_3^3 \leq \lambda_1(x_1^2 + x_2^2 + x_3^2) = \lambda_1.$$

The equality holds if and only if d equals to the first eigen vector e_1 of Σ. This shows the solution to the least square problem in Eqn. (8) is the first eigen vector of the covariance matrix. Hence the best fitting line is given by

$$\gamma(t) = c + te_1. \tag{9}$$

This process is also called the *Principle Component Analysis* method.

Virtual Camera Once a light camera has been calibrated, we can use it to synthesize virtual pinhole cameras, then we can the virtual camera and conventional vision algorithms directly.

First, we define a virtual optical center. Since all the camera rays may not intersect at a common point, we define the optical center as the one minimizing the total squared distance to all camera rays. For each camera ray $\gamma_{ij} := \gamma(i, j)$, it is represented as

$$\gamma_{ij}(t) := a_{ij} + n_{ij}t,$$

where a_{ij} is the base point, n_{ij} the unit direction vector. The squared distance from a point $p \in \mathbb{R}^3$ to the line of γ_{ij} is given from Pythagoras:

$$d_{ij}^2 = |p - a_{ij}|^2 - \langle p - a_{ij}, n_{ij}\rangle^2 = (p - a_{ij})^t(p - a_{ij}) - [(p - a_{ij})^T n_{ij}]^2,$$

where $(p - a_{ij})^t n_{ij}$ is the projection of $p - a_{ij}$ on the line. The sum of distance to the square to all lines is:

$$\sum_{i,j} d_{ij}^2 = \sum_{i,j} \left[(p - a_{ij})^T(p - a_{ij}) - [(p - a_{ij})^t n_{ij}]^2\right]$$

In order to minimize the total squared distance, we differentiate it with respect to p.

$$\sum_{i,j}(p - a_{ij}) - [(p - a_{ij})^T n_{ij}]n_{ij} = 0$$

Namely

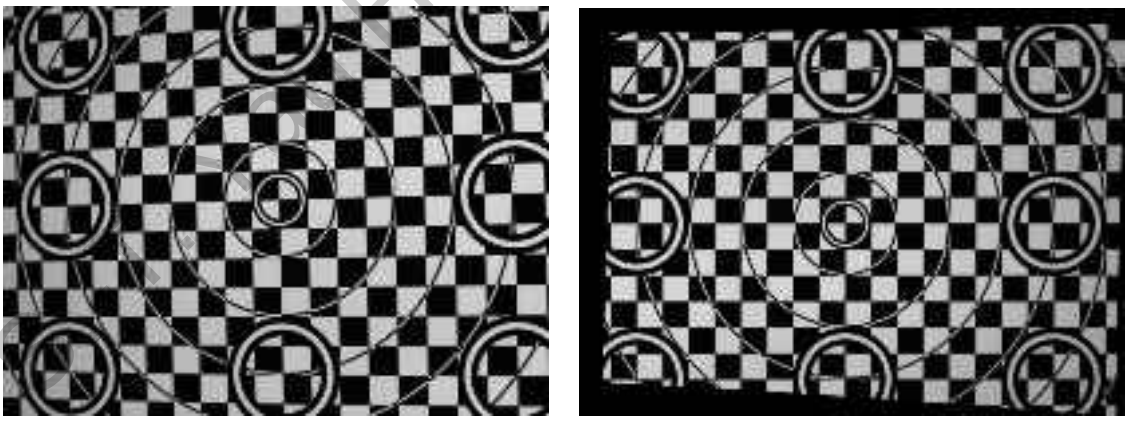
$$\sum_{i,j}(p - a_{ij}) = \sum_{i,j}[(p - a_{ij})^T n_{ij}]n_{ij} = \sum_{i,j} n_{ij}[(p - a_{ij})^T n_{ij}] = \sum_{i,j} n_{ij}n_{ij}^t(p - a_{ij})$$

Then we obtain a linear equation

$$\left[\sum_{i,j}(n_{ij}n_{ij}^T - I)\right] p = \sum_{i,j}(n_{ij}n_{ij}^T - I)a_{ij}, \tag{10}$$

where I is the identity matrix. The linear system can be solved using Eigen library []. The virtual optical center is denoted as O_v.

Seond, we define the virtual camera image plane π_v as $Z_w = 0$. Each ray $\gamma(i, j)$ in the light field camera intersects π_v at $p(i, j)$. The (i, j)-th pixel in the physical camera is mapped to $p(i, j)$ in the virtual camera image plane. We use a piecewise linear mapping to map the image captured by the physical camera to the virtual camera image. The mapping can be implemented using texture mapping method directly using OpenGL. Fig. (8) shows a virtual camera construction result. The left frame is the real image captured by our physical camera, some circles are distorted and look like ellipses. The right frame is the virtual camera image, all the circles are corrected and look much more circular.



(a) physical camera image (b) virtual camera image

Figure 8: Comparision between the real image captured by our physical camera and the the image of our virtual camera at $Z_w = 0$. On the physical image, some circles are distorted to ellipses, on the virtual image, they are corrected to circles.

In this way, both the left and the right physical cameras are converted to virtual pinhole cameras. Next, we can use conventional algorithms for stereo-matching and reconstruction.

5 Stereo-matching and Reconstruction

In this section, we review the conventional algorithms in stereo-vision, including rectification, stereo-matching and triangulation (reconstruction). The stereo-matching algorithm matches each pixel in the left image to a unique pixel in the right image. The triangulation algorithm computes the intersection between the camera rays through the pair of matched pixels to obtain the depth information.

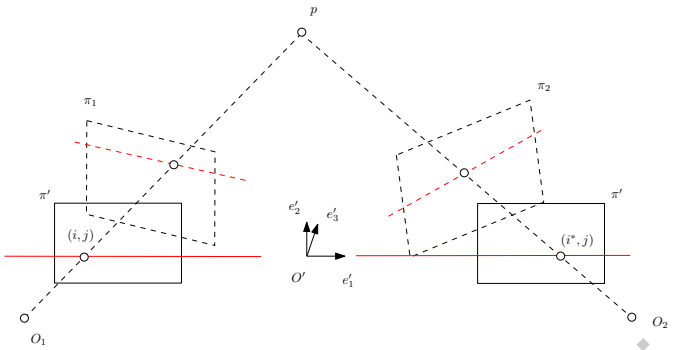


Figure 9: Epipolar rectification.

Figure 10: Epipolar rectification results.

Epipolar Rectification Finding matches in stereo vision is restricted by epipolar geometry: as shown in Fig. (10), each pixel's match in another image can only be found on a line called *epipolar line*. If the two images are coplanar, and their optical centers differ by a horizontal translation, then each pixel's epipolar line is horizontal and at the same vertical position as that pixel. In general settings, the epipolar lines are slanted. *Image epipolar rectification* warps both images to make their epipolar lines to be horizontal, therefore simplifies the matching process.

Suppose we have two virtual cameras with optical centers O_1 and O_2, and image planes π_1 and π_2 respectively.

We choose the middle point of the optical centers as the origin of the rectified coordinate system,

$$O' = \frac{1}{2}(O_1 + O_2),$$

the new x-direction is given by

$$e_1' = \frac{O_2 - O_1}{|O_2 - O_1|},$$

the rectified z-direction equals to the original z-direction, $e_3' = e_3$, and the rectified y-direction is given by

$$e_2' = e_3' \times e_1',$$

where $\times$ is the cross product of vectors. Then we can transform the coordinates to the rectified frame $\{O' : e_1', e_2', e_3'\}$ by a rigid motion,

$$(e_1', e_2', e_3') \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + O' = (e_1, e_2, e_3) \begin{pmatrix} x \\ y \\ z \end{pmatrix} + O.$$

Hence the coordinate transformation is given by

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + T,$$

where the translation vector is given by $T = O' - O$, and rotation R matrix is $R = (e_1', e_2', e_3')$, namely the coordinates of the axis vectors of the rectified frame.

In the rectified coordinate systems, we choose the image plane π' as $z' = c'$. Consider the left virtual camera $\{\pi_1, O_1\}$, each (i, j)-th pixel on the image plane represents a ray $\gamma(i, j)$, which intersects π' at (u, v). This gives a *projective transformation* $\varphi_1 : \pi_1 \to \pi'$, using the optical center O_1 as the projection center. Suppose in the rectified coordinate systems,

$$O_1' = (x_1', y_1', 0),$$

then the projection

$$\varphi_1(x', y', z') = \frac{1}{c'}(x' - x_1', y' - y_1').$$

Similarly, we compute the projective transformation $\varphi_2 : \pi_2 \to \pi'$, $O_2' = (x_2', y_2', 0)$,

$$\varphi_2(x', y', z') = \frac{1}{c'}(x' - x_2', y' - y_2').$$

The projected images are the rectified images. The projective transformation from the virtual camera image to the rectified image can be carried out using texture mapping in OpenGL. Fig. (10) shows the results of epipolar rectification algorithm.

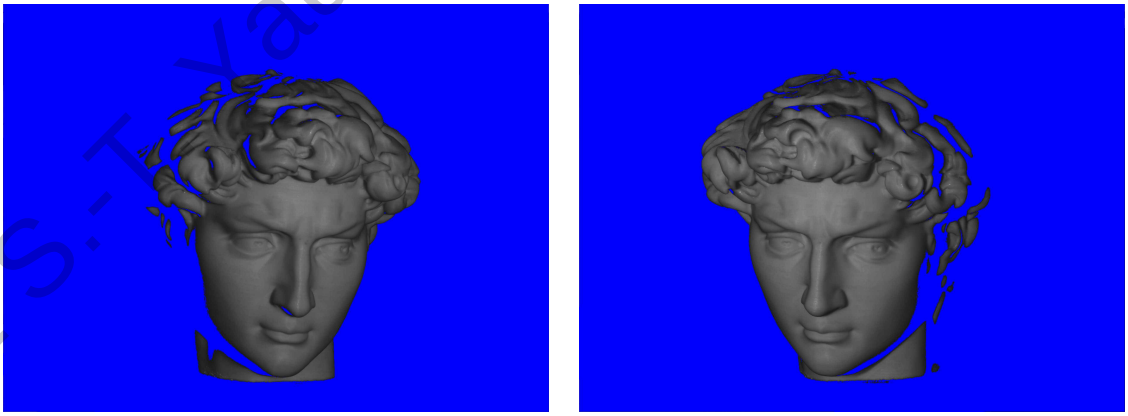


Figure 11: Segmentation results.

Stereo-matching The rectified ambient image is segmented into foreground and the background by simple intensity threshold. For all pixels with intensity less than a threshold ε, they are labeled as background, otherwise labeled as foreground. Fig. (11) shows the segmentation results, the background is in blue color. The stereo-matching is conducted between the foreground pixels only.

For each foreground pixel (i, j) On the left rectified image, its corresponding epipolar line on the right rectified image is the horizontal line with equal height. We search in the epipolar line pixel by pixel, and find the matched pixel with the minimal difference. In practice, we choose a small neighborhood of each pixel, and measure the total squared difference between the neighorhoods. The *disparity map* is defined as the difference between the horizontal indices of the matched pair. Fig. (12) shows the disparity map computed by this algorithm.

Figure 12: Disparity Map.

Reconstruction (Triangulation) In the rectified coordinates system, the optical centers of the left and the right virtual cameras are $(-l, 0, 0)$ and $(+l, 0, 0)$ respectively, both camera planes are $z = c$. The centers of the left and the right camera images are $(-l, 0, 0)$ and $(+l, 0, 0)$. Suppose the size of the camera image is $w \times h$, the distance among the pixels are Δx and Δy. Then the coordinates of the pixel in the left image (i, j) and the coordinates of the pixel in the right image $(i + f(i, j), j)$ are

$$((i - w/2)\Delta x + l, (j - h/2)\Delta y, c) \quad ((i + f(i, j) - w/2)\Delta x - l, (j - h/2)\Delta y, c)\,,$$

where $f(i, j)$ is the disparity for pixel (i, j). Their intersection point between two rays is given by

$$(t(i - w/2)\Delta x - l, t(j - h/2)\Delta y, tc), \quad t = \frac{2l}{f(i, j)}\Delta x. \tag{11}$$

Thus the 3D point cloud can be obtained accordingly. Fig. (1) frame (b) and Fig. (13) show the reconstructed 3D point cloud using the stereo-matching and triangulation.

6 Phase Shifting Algorithm

Light field camera calibration and stereo-matching are based on phase shifting algorithm. In calibration process, the coordinates of each pixel on the LCD panel is encoded by the absolute phase by Eqn. (13), and converted to intensities by Eqn. (12) and Eqn. (14). The cameras capture the fringe images shown on the LCD panel and compute the coordinates of pixels of the LCD panel. For stereo-matching, the 3D object is lit by the projected fringe image. Each point on the 3D object corresponds to a pixel on the image of the digital projector. For each pixel on the camera image, its corresponding projector pixel can be computed using the phase shifting algorithm. The pixels on the left camera image and those on the right camera image are matched by their corresponding projector pixel.

Projection Fringe Pattern A digital projector is used as the light source, then projector displays fringe patterns with different wave lengths and base phase. In our experiment, the horizontal fringe patterns for the projector are given by

$$H_k(u, v) = a + b\cos(\Phi(u, v) + \Delta\varphi_k) \tag{12}$$

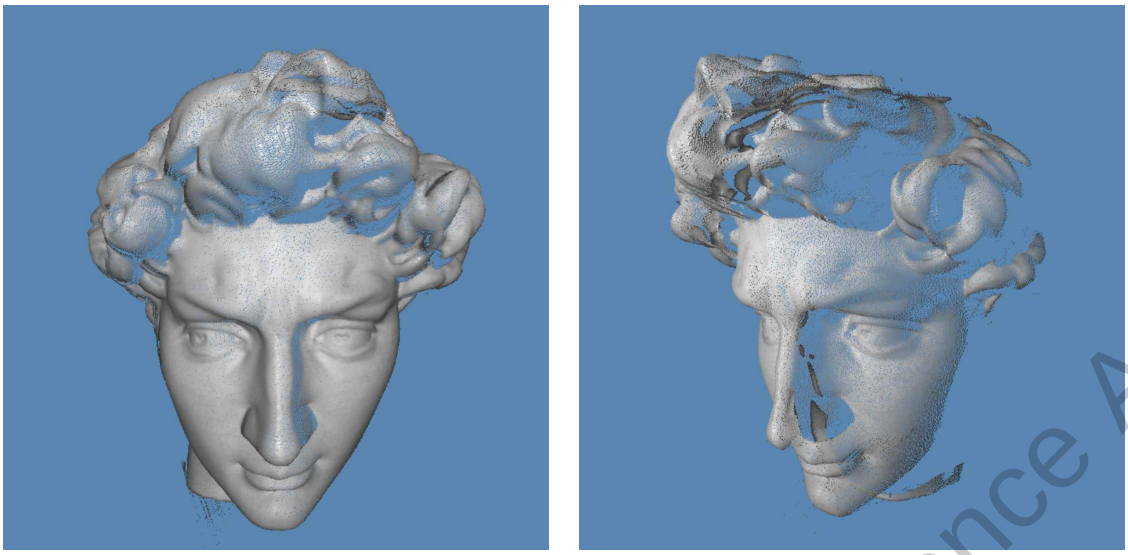


Figure 13: Reconstructed 3D point cloud, captured from different view angles.

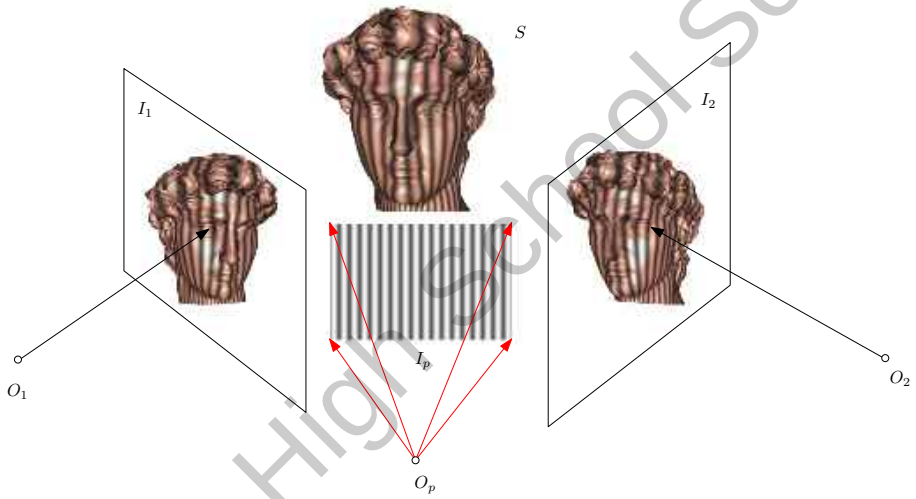


Figure 14: The fringe pattern for the digital projector or the LCD display. The left and right camera optical centers and image planes are (O_1, I_1) and (O_2, I_2) respectively. The projector optical center and image plane are (O_p, I_p).

where a is the ambient component, b the modulation,

$$(\Delta\phi_1, \Delta\varphi_2, \Delta\phi_3) = \left(-\frac{2\pi}{3}, 0, \frac{2\pi}{3}\right),$$

the absolute phase

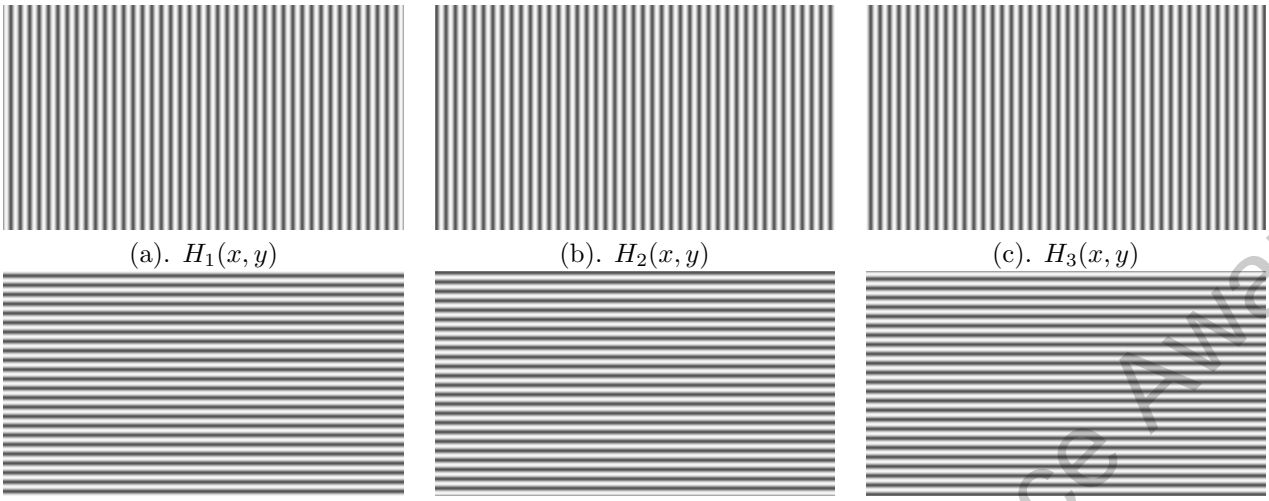
$$\Phi(u, v) = \frac{2\pi u}{\lambda}, \tag{13}$$

λ is the wavelength. The absolute phase $\Phi(u, v)$ is solely determined by the horizontal coordinate u. Similarly, the vertical fringe patterns are given by

$$V_k(u, v) = a + b\cos(\Phi(u, v) + \Delta\varphi_k), \tag{14}$$

where $\Phi(u, v) = 2\pi v/\lambda$, which is solely determined by the vertical coordinate v. The fringe pattern for the digital projector is shown in Fig. (15). In the calibration process, the LCD panel also displays the similar fringe patterns.

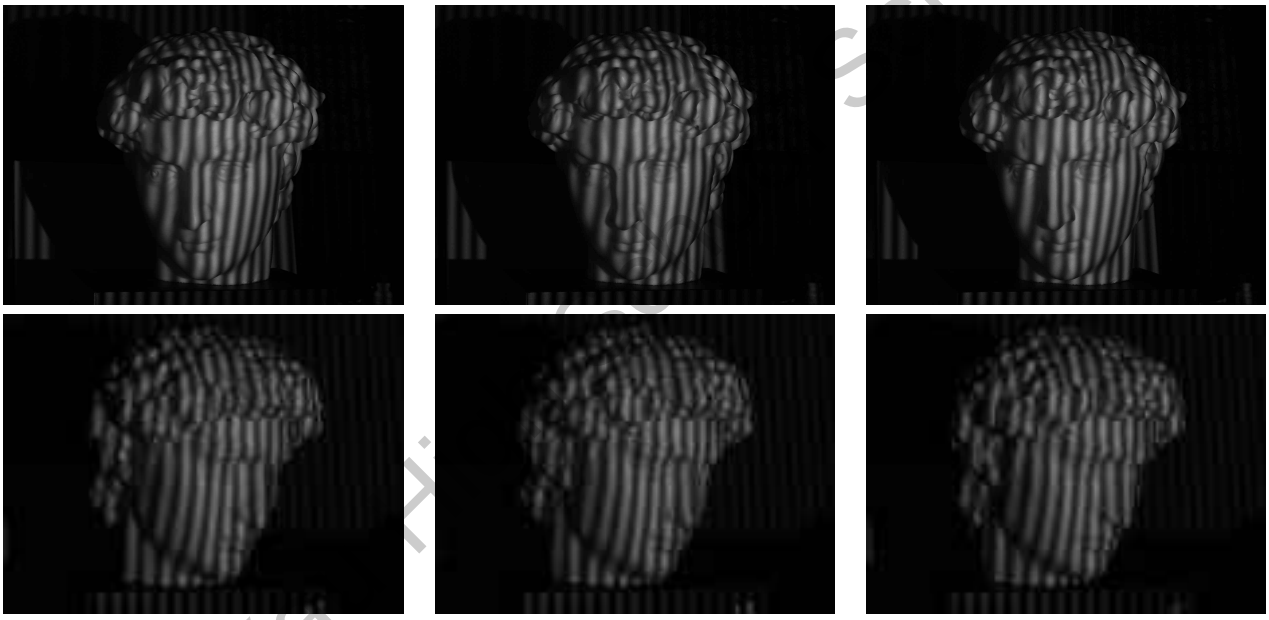
Phase Shifting Algorithm Phase-shifting algorithms are widely used in 3D vision to capture depth from the asolute phase. For the single wavelength phase-shifting algorithm, a number of fringe images with certain phase shift are used to obtain the phase. The 3D coordinates are computed from the phase based on calibration.



(a). $H_1(x,y)$ (b). $H_2(x,y)$ (c). $H_3(x,y)$

(d). $V_1(x,y)$ (e). $V_2(x,y)$ (f). $V_3(x,y)$

Figure 15: The fringe pattern for the digital projector or the LCD display.



(a). $I_1(x,y)$ (b). $I_2(x,y)$ (c). $I_3(x,y)$

Figure 16: Fringe images in the phase-shifting algorithm, the top row shows the images captured by the left camera, the bottom row show shows those by the right camera.

A three step phase-shifting algorithm with a phase shift of $2\pi/3$ can be written as,

$$\begin{aligned} I_1(x,y) &= I'(x,y) + I''(x,y)\cos[\Phi(x,y) - 2\pi/3] \\ I_2(x,y) &= I'(x,y) + I''(x,y)\cos[\Phi(x,y)] \\ I_3(x,y) &= I'(x,y) + I''(x,y)\cos[\Phi(x,y) + 2\pi/3] \end{aligned} \tag{15}$$

where $I'(x,y)$ is the *ambient*, $I"(x,y)$ the *intensity modulation*, and $\Phi(x,y)$ the *absolute phase*. Fig. 16 show three phase-shifting images.

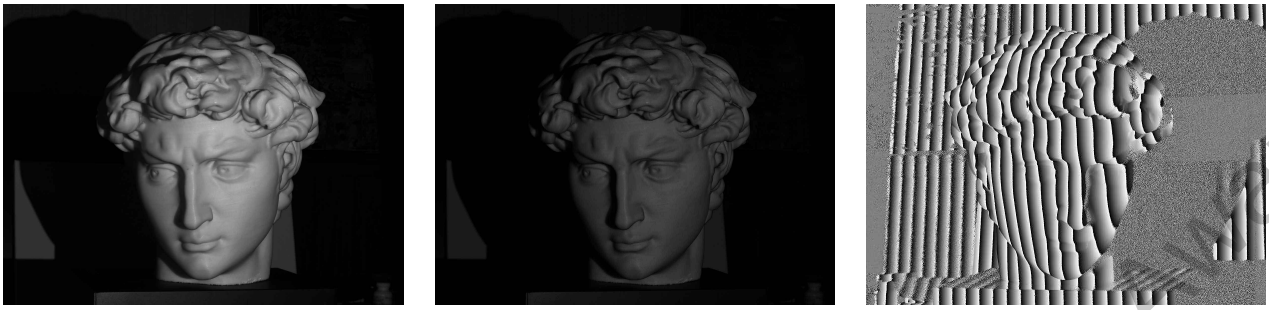
From the three equations, we can obtain

$$\begin{aligned} I'(x,y) &= \frac{1}{3}[I_1(x,y) + I_2(x,y) + I_3(x,y)] \\ I''(x,y) &= \frac{1}{3}\sqrt{3(I_1 - I_3)^2 + (2I_2 - I_1 - I_3)^2} \end{aligned} \tag{16}$$

The *relative phase* or *wrapped phase* is given by:

$$\varphi(x,y) = \tan^{-1}\frac{\sqrt{3}(I_1 - I_3)}{2I_2 - I_1 - I_3}, \tag{17}$$

$\varphi(x, y)$ ranging from $-\pi$ to π. Fig. (17) shows the ambient, modulation and wrapped phase components computed from the fringe images in Fig. (16).



(a). ambient $I'(x, y)$ (b). modulation $I''(x, y)$ (c). wrapped phase $\varphi(x, y)$

Figure 17: The ambient, modulation and wrapped phase computed from the fringe images in Fig. 16.

Phase Unwrapping The actual *absolute phase* is given by

$$\Phi(x, y) = 2\pi k(x, y) + \varphi(x, y), \tag{18}$$

where $k(x, y)$ are integers. The process of recovering $k(x, y)$ is called *phase unwrapping*. In this work, we use *dual wave length phase unwrap* method.

Suppose we capture the fringe images of two wave lengths λ_1 and λ_2, and compute the wrapped phase $\phi_1(x, y)$ and $\phi_2(x, y)$. Then according to Eqn. (13),

$$\Phi_1(x, y) - \Phi_2(x, y) = \frac{2\pi u}{\lambda_1} - \frac{2\pi u}{\lambda_2} = \frac{2\pi u}{\lambda_{eq}}, \tag{19}$$

where λ_{eq} is the equivalent wavelength,

$$\lambda_{eq} = \frac{\lambda_1 \lambda_2}{|\lambda_1 - \lambda_2|}. \tag{20}$$

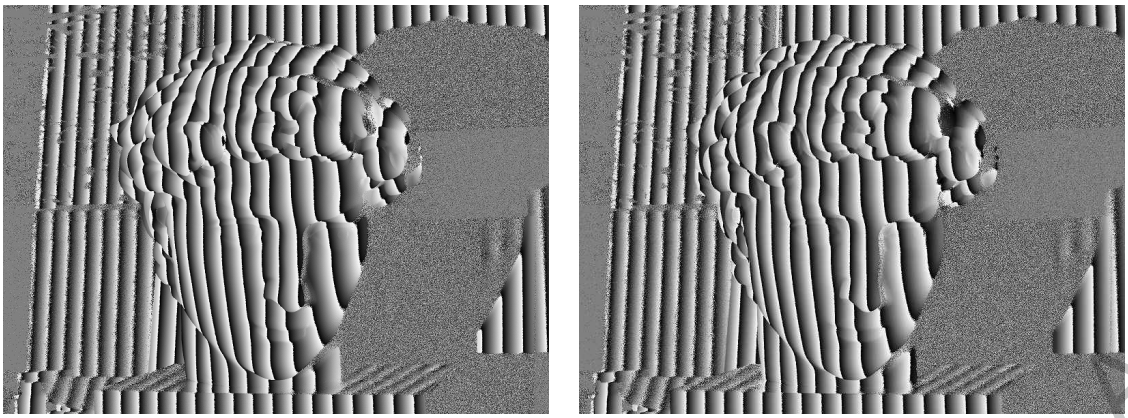
Suppose λ_1 and λ_2 are close enough, therefore λ_{eq} is very big so that the whole field of view (FOV) is covered by a single wavelength, therefore the wrapped phase $\varphi_1(x, y) - \varphi_2(x, y)$ equals to the absolute phase. By the unwrapped phase of equivalent wavelength λ_{eq}, we can recover the unwrapped phase for λ_1 and λ_2.

7 Experiments

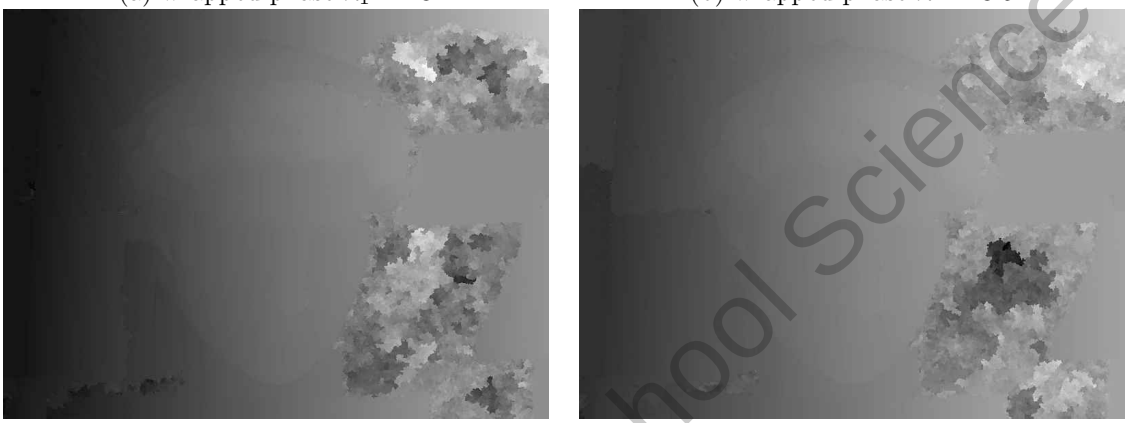
Hardware System The two gray scale cameras are new point Grey/Flir Grasshopper-20s4M camera with IEEE 1394B FireWire interface, and Arducam C-Mount Lens for 12MP IMX477 Raspberry Pi HQ Camera, 16mm Focal Length with Manual Focus and Aperture Adjustment. The image resolution is 1600×1200. The color camera is IDS CCD 3.0 C-mount camera UI-6280SE-C-HQ with GigE interface, and a Fujinon HF25SA-1 lense. The cameras are mounted on a koolehaoda Aluminium 480mm Professional Rail, the camera system is mounted on a 400mm CNC Sliding Table with a cross slide linear stage and a ballscrew. The digital project is TI Lightcrafter 4500 Education module with resolution 1140×912. The cameras and the projector are synchronized, and the projector sends out the trigger signals. As shown in Fig. 19, various clamps and adapters are designed use Fusion 360, and printed using Creality 3D Ender V2 3D printer. The scanned object is scanned from Michelangelo's David sculpture and 3D printed.

Software System The algorithms are implemented using generic C++ on Windows Visual Studio platform. The image processing is based on OpenCV library, the graphics display uses OpenGL and freeglut, the linear systems are solved using Eigen library. The digital camera program is based on FlyCapture SDK, the digital projector coding is based on TI Lightcrafter SDK.

System Alignment We use the laser pointer to align the system, as shown in the right frame of Fig. 20. In the first step, we mount a small screen on the linear stage of the sliding table, then slide the stage and observe the intersection between the laser beam and the screen. By adjusting the orientation of the laser pointer, we ensure the intersection point is invariant when we slide the stage and change the position of the screen. This means the laser beam is parallel to the linear rail. In the second step, we attach a first surface mirror on the LCD panel, and adjust the orientation of the LCD panel, such that the laser beam hits the mirror and is reflected to the lense of the laser pointer. This means the LCD panel is perpendicular to the laser beam, therefore to the linear rail as well.



(a) wrapped phase $\lambda_1 = 18$ (b) wrapped phase $\lambda = 18.5$



(c) unwrapped phase $\lambda_1 = 18$ (b) unwrapped phase $\lambda = 18.5$

Figure 18: Dual wavelength method for phase unwrapping.

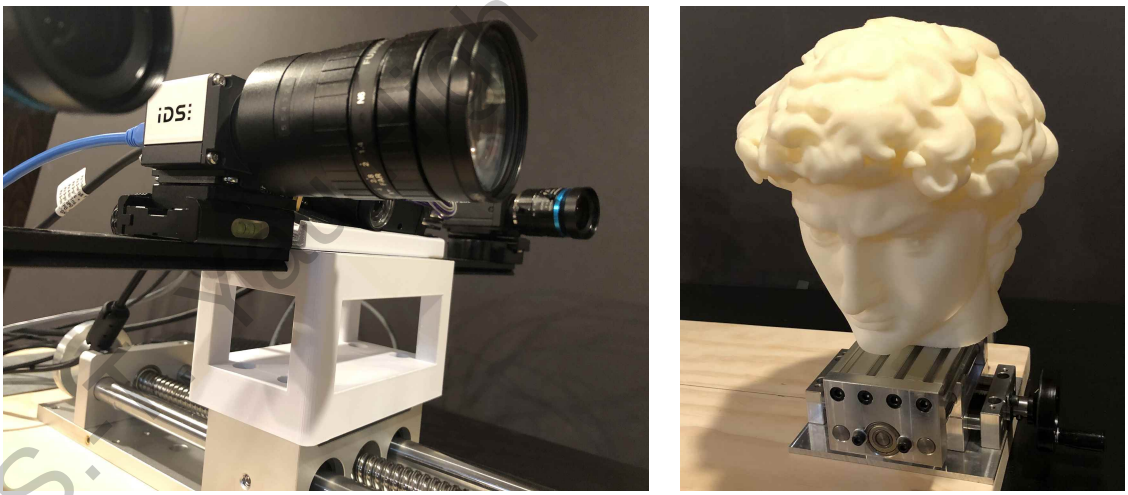


Figure 19: The clamps and adapters are 3D printed. The scanned object is also 3D printed.

Exermpiental Results We have tested our calibration algorithm and stereo-vision system by scanning real objects in physical world. Fig. (13) and Fig. (22) show the point clouds of the 3D printed David sculpture scanned from different view angles. Fig. (23) illustrates the point clouds of a Chinese dragon sculpture. From the scanning results, we can see that all the geometric details, such as the scales, the teeth of the sculpture are accurately reconstructed. Fig. (24) shows the scanned results for various objects, including the fruits and vegetables. The small bumps on the cucuber are clearly detected. The experiment for scanning a watermelon in Fig. (**??**) is more challenging, because the melon surface has dark areas and highly reflective regions. We use multiple-exposure technique for the scan and obtain good reconstruction quality.

The scanned point clouds are merged together using iterative closest point (ICP) method [17, 20, 14]. The

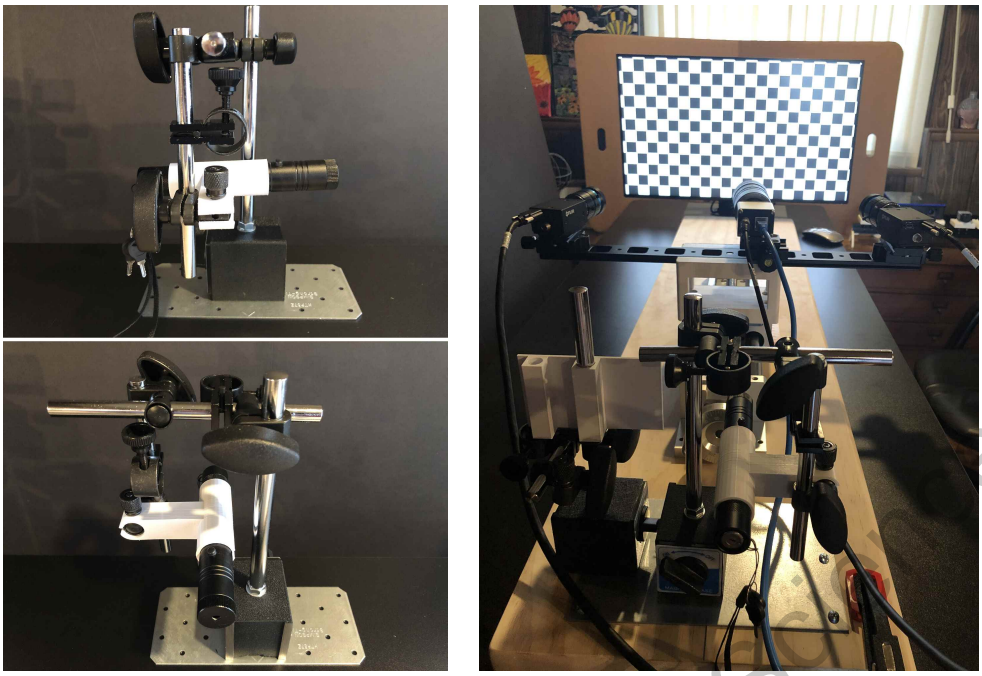


Figure 20: The laser pointer is mounted on a multi-position magnetic base with two degree of freedoms. The system is aligned uses the laser beam.

merged point clouds are reconstructed as triangle mesh using Poisson reconstruction algorithm [11]. The reconstruction is carried out using MeshLab software. The reconstructed meshes are 3D printed directly.

Furthermore, we use our system to scan a planar object. The best fitting plane is calculated using the PCA method. Then we measure the average distance from the reconstructed points to the best fitting plane, the error is about $0.3mm$. The error obtained based on conventional calibration method is about $2.4mm$. This shows our proposed method outperforms the convention one.

8 Conclusion

This work proposes a novel algorithm for light field camera calibration. Comparing to conventional pinhole camera model, the light field camera has much more parameters and can describe complicated distortions of the lense and the sensors. The classical calibration method is based on non-linear optimization method, the proposed method is based on PCA, which has unique global optimum, and is much simpler and purely in parallel. Our experimental results shows that our algorithm achieves better performance.

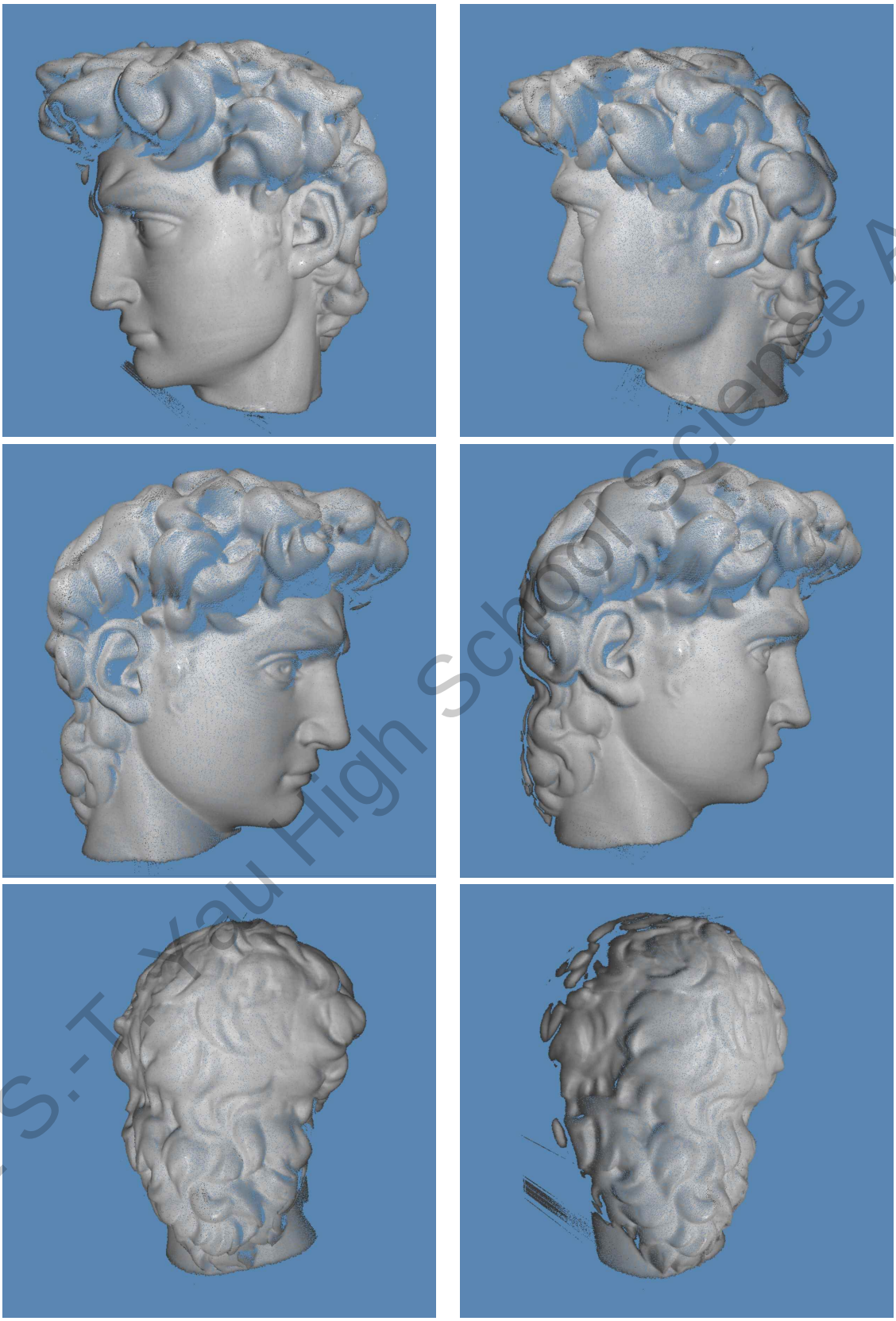


Figure 21: 3D scanned point clouds from different view angles.

Figure 22: Reconstructed 3D Surface from scanned point clouds and the 3D printed model with the original object. It can be seen that the scanning quality is high and the printed model preserves most geometric features of the original surface.

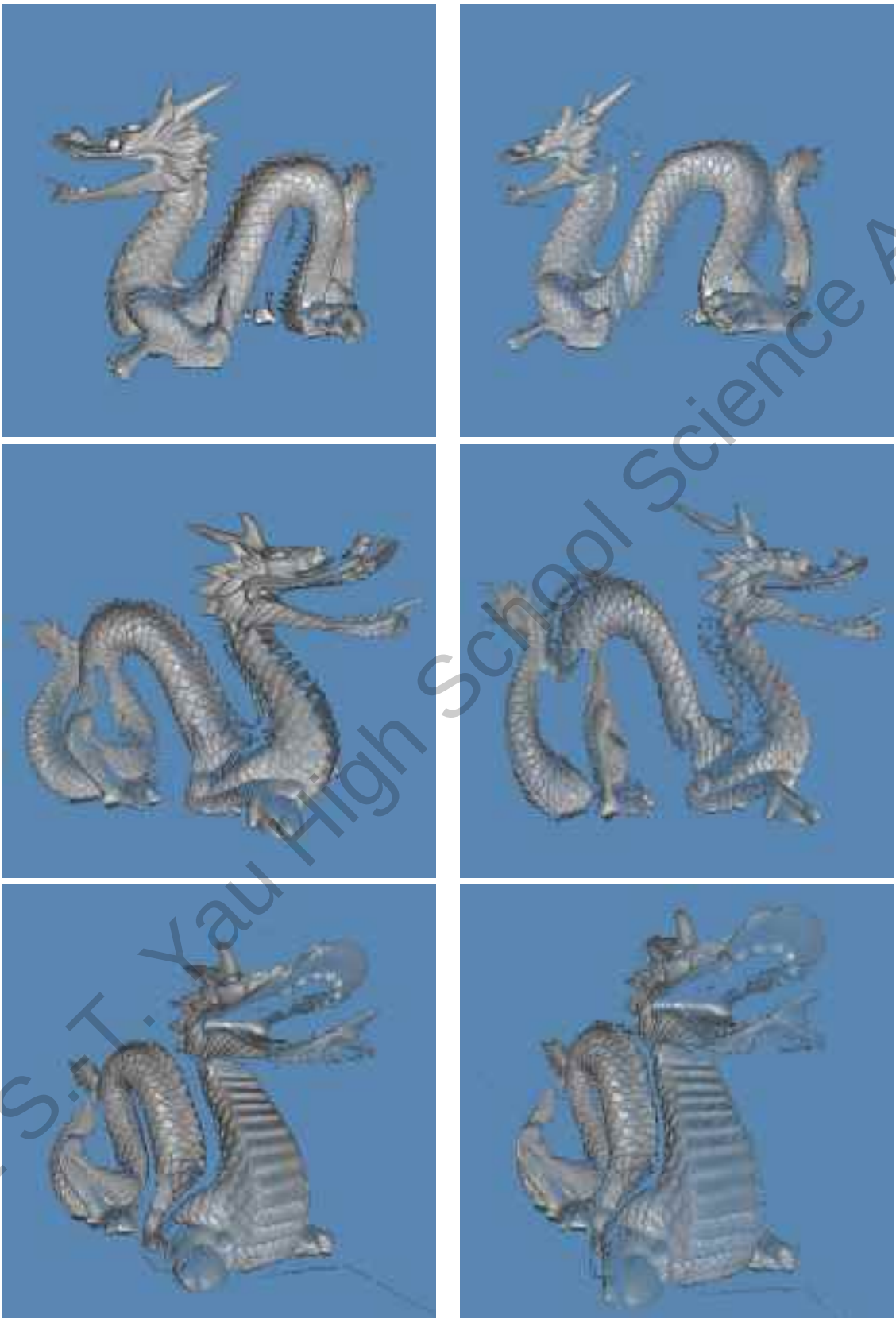


Figure 23: 3D scanned point clouds for the dragon model from different view angles.

Figure 24: 3D scanned point clouds from different view angles. Top rows vase model, middle row: pear and peach, bottom row: cucumber.

Figure 25: 3D scanned point clouds from different view angles, a watermelon.

Figure 26: 3D reconstructed teeth fossils.

Figure 27: 3D reconstructed bone fossil.

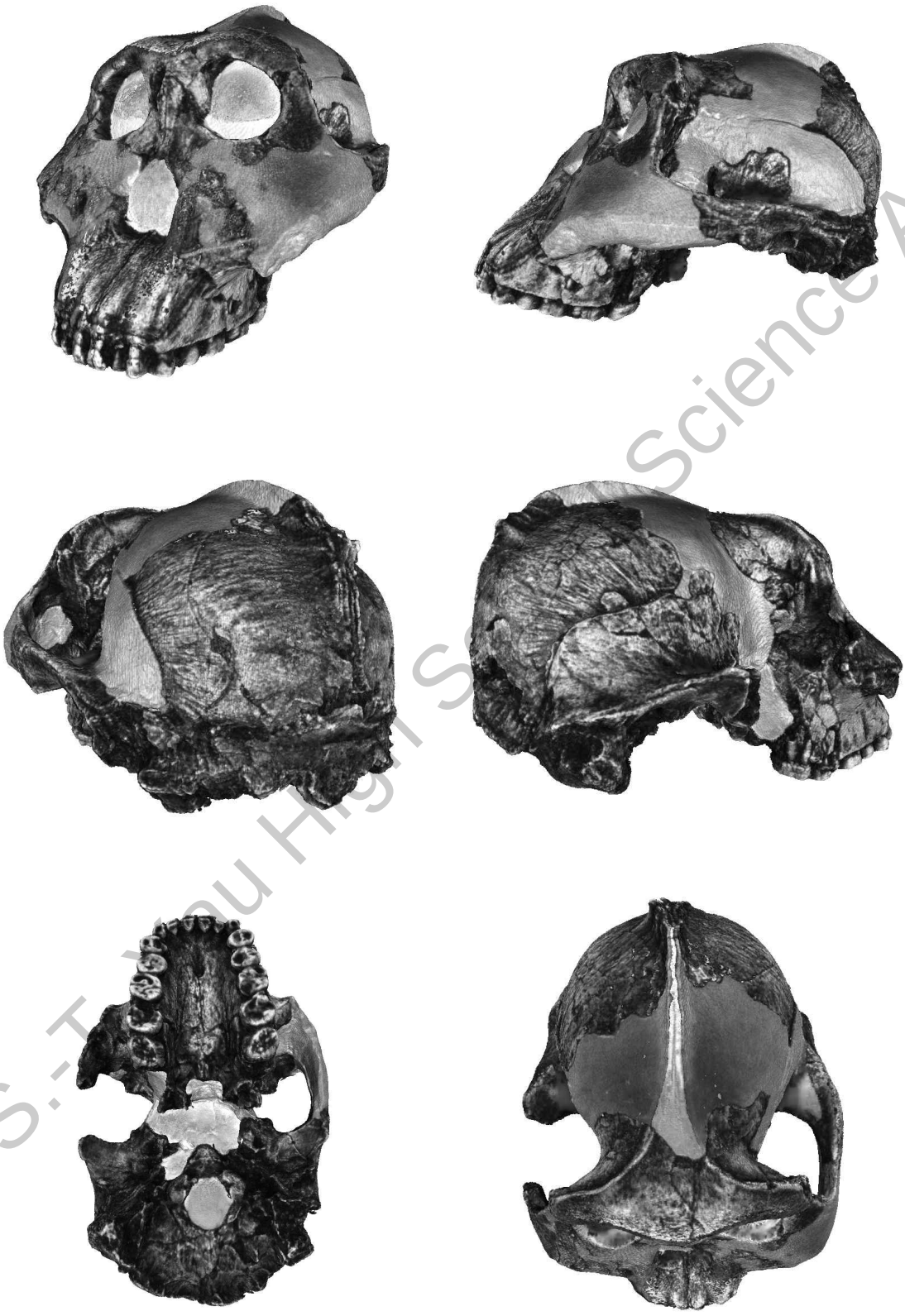


Figure 28: 3D reconstructed skull fossils.

Figure 29: 3D reconstructed girl sculpture and the 3D printed model.

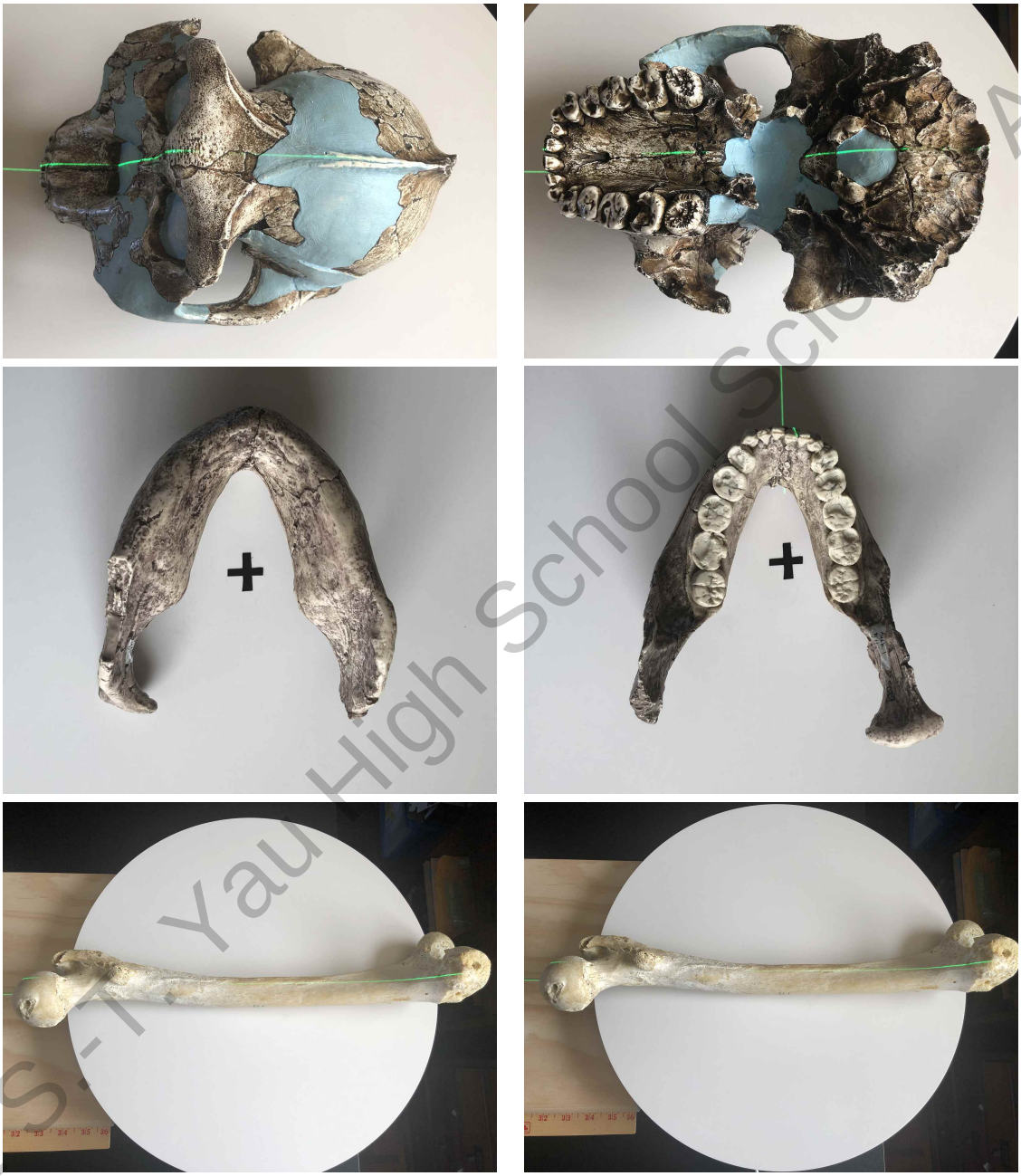


Figure 30: Real objects for the scanning tests.

References

[1] Q1 report. https://www.tesla.com/VehicleSafetyReport.

[2] D. Adams. *Tesa Invistigation*. San Val, 1995.

[3] J.-X. Chai, X. Tong, C.-C. Chan, and H.-Y. Shum. Plenoptic sampling. In *SIGGRAPH 2000 Conference Proceedings*, pages 307–318, 2000.

[4] F. Devernay and O. Faugeras. Straight lines have to be straight. *Machine Vision and Application*, 13, 2001.

[5] M. T. El-Melegy and Aly A. Farag. Nonmetric lens distortion calibration: Closed-form solutions, robust estimation and model selection. In *International Conference on Computer Vision (ICCV)*, 2003.

[6] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Computer Vision and Pattern Recognition (CVPR)*, 2001.

[7] S.J. Gorlter, R. Grzeszczuk, R. Szeliski, and M.F. Chohen. The lumigraph. In *SIGGRAPH 1996 Conference Proceedings*, pages 43–54, 1996.

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[9] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition (CVPR)*, 1997.

[10] P. Cohen J. Weng and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, 14:965–980, 1992.

[11] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, 2013.

[12] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH 1996 Conference Proceedings*, pages 31–42, 1996.

[13] Zhouchen Lin and Heung-Yeung Shum. A geometric analysis of light field rendering. *International Journal of Computer Vision*, 58:121–138, 2004.

[14] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, page 145–152, 2001.

[15] G. P. Stein. Lens distortion calibration using point correspondences. In *Computer Vision and Pattern Recognition (CVPR)*, 1997.

[16] Ren Wu. Fourier slice photography. In *SIGGRAPH 2005 Conference Proceedings*, page 735–744, 2005.

[17] Chen Yang and Gerad Meioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, 1991.

[18] Yezzi and Soatto. Stereoscopic segmentation. *International Journal of Computer Vision (IJCV)*, 53:31–43, 2003.

[19] Jingyi Yu, Leonard McMillan, and Steven Gortler. Surface camera (scam) light field rendering. *International Journal of Image and Graphics*.

[20] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(12):119–152, 1994.

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