STRUCTURED LIGHT VISION SYSTEMS USING A ROBUST LASER STRIPE SEGMENTATION METHOD

by

Zhankun Luo

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering at Purdue Northwest

Hammond, Indiana

May 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Lizhe Tan, Chair

Department of Electrical and Computer Engineering

Dr. Khair Al Shamaileh

Department of Electrical and Computer Engineering

Dr. Colin Elkin

Department of Electrical and Computer Engineering

Approved by:

Dr. Vijay Devabhaktuni

This thesis is dedicated to my parents

ACKNOWLEDGMENTS

Firstly, I would like to express my deep gratitude and appreciation to my academic advisor Dr. Lizhe Tan for all his constructive instruction and valuable advice during the research at Purdue University Northwest.

I would also like to extend my thanks to my parents, uncle, and host family for their consistent support and encouragement while studying abroad.

Furthermore, I sincerely appreciate the help from my friends, Liming Wu, Yaan Zhang, Jintao Hou, and Changshi Yang. They provide warm-hearted help on my career and life constantly.

In addition, I am truly grateful to Dr. Bin Chen, Dr. Sidike Paheding, and Dr. Weihua Ruan for their valuable suggestions on my research and study, and especially to Dr. Chenn Zhou for her funding of my being a research assistant at the Center for Innovation through Visualization and Simulation (CIVS).

Last but not least, I would like to thank the other committee member, Dr. Khair Al Shamaileh and Dr. Colin Elkin for their participation and helpful comments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A	camera intrinsic matrix
R	extrinsic parameter: rotation matrix from camera coordinates to world coordinates
t	extrinsic parameter: translation from camera coordinates to world coordinates
$X = (x\ y\ z)^T$	the point in the world coordinate
$\bar{X} = (x\ y\ z\ 1)^T$	the homogeneous form of X
$X_{ck} = (x_{ck}\ y_{ck}\ z_{ck})^T$	the point in the k-th camera coordinate
$\bar{X}_{ck} = (x_{ck}\ y_{ck}\ z_{ck}\ 1)^T$	the homogeneous form of X_{ck}
$I_{pk} = (u_k\ v_k)^T$	the point in the pixel coordinate from the k-th camera
$\bar{I}_{pk} = (u_k\ v_k\ 1)^T$	the homogeneous form of I_{pk}

ABBREVIATIONS

3D	Three dimensional
CCD	Charge-coupled Device
SSD	Solid State Drive
RANSAC	Random Sample Consensus
MLRANSAC	Multi-Level Random Sample Consensus
AI	Artificial Intelligence
CNN	Convolutional Neural Network
GPU	Graphic Processing Unit
MLP	Multi-Layer Perceptron
FCN	Fully Convolutional Networks
BP	Back propagation
BCELoss	Binary Cross Entropy Loss
SGD	Stochastic Gradient Descent
Adam	Adaptive moment estimation
IoU	Intersection over Union

ABSTRACT

In thesis, we propose a structured light vision system equipped with multi-cameras and multi-laser emitters for object height measurement or 3D reconstruction. The proposed method offers a better accuracy performance over a single camera system. The structured light method may fail the interference of reflection and scattering of light. We use U-Net to extract the laser region, obtain the laser stripe center after erosion and dilation, and finally reconstruct the point cloud corresponding to the laser stripe. Our experiments demonstrate that our structured light system with the U-Net can perform effectively and robustly in a complex environment.

1. INTRODUCTION

1.1 Literature Review

Today, structured light systems are widely utilized in computer vision for 3D reconstruction[1]–[3]. Due to the high accuracy and efficiency, people usually chose the structured light vision systems for various industrial applications, including robotics, mechanical fault detection, autonomous driving, architecture, archaeology, agriculture [4]–[10] and Mars exploration [11]–[13]. Besides, it can be used to predict phenotyping features in the target plants [14], reconstruct sub-surface 3D depth images [9] and curve welding seam [15], [16]. Traditionally, a vision system with beam sources and cameras installed at different viewpoints can provide enough information for a basic 3D reconstruction [17]. Typically, a vision system with multiple beam sources and cameras installed at different angles can provide accurate information for 3D reconstruction [17], [18]. In addition, laser emitters consisting of beam laser lines and projectors with encoded patterns [2], [19], [20] are also frequently used to perform 3D reconstructions.

1.2 Motivation

Nevertheless, the structured light system could be affected by the inference of reflection and scattering under complex scenarios. U-Net is a type of neural network that was firstly proposed for biomedical image segmentation task [21]. This network could be utilized to filter out the image corruption caused by reflection and scattering of light.

1.3 Thesis Scope

In this study, we eliminate the effect of reflective noise in the background for a structured light system using multiple laser emitters and numerous cameras similar to [22]. Eventually, we validate the proposed methods. The important contributions of this project are the derivation of a framework with multiple laser emitters and multiple cameras, and the proposed method to extract laser regions with reflective interference in complex environments

using U-Net. Finally, we adopt multiple cameras to improve the measurement accuracy over the recently developed system [23].

2. BACKGROUND

2.1 Pinhole Camera Model

The pinhole camera model represents the transformation from world coordinates to pixel coordinates (see Fig. 2.1). We define the corresponding symbols in Table 2.1.

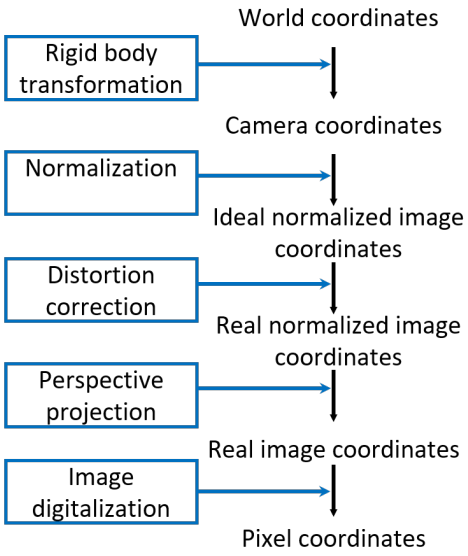


Figure 2.1. Transformations from world coordinates to pixel coordinates.

Table 2.1. Symbol table for pinhole camera model

symbol	definition
$(x\ y\ z)^T$	the world coordinates
$(x_c\ y_c\ z_c)^T$	the camera coordinates
$(x_u\ y_u)^T$	the ideal normalized image coordinates
$(\breve{x}\ \breve{y})^T$	the distorted real normalized image coordinates
$(x_d\ y_d)^T$	the distorted real image coordinates
$(u\ v)^T$	the pixel coordinates
f	the focal length
k_1, k_2	the parameters of radial distortion
dx, dy	the physical scales for pixel on x, y axes
u_0, v_0	the pixel coordinates of pinhole
θ	the skewed angle between x, y axes of the pixel coordinates

2.1.1 Rigid body transformation

We denote the rotation matrix and translation between the world coordinates $(x\ y\ z)^T$ and the camera coordinates $(x_c\ y_c\ z_c)^T$ as R, t.

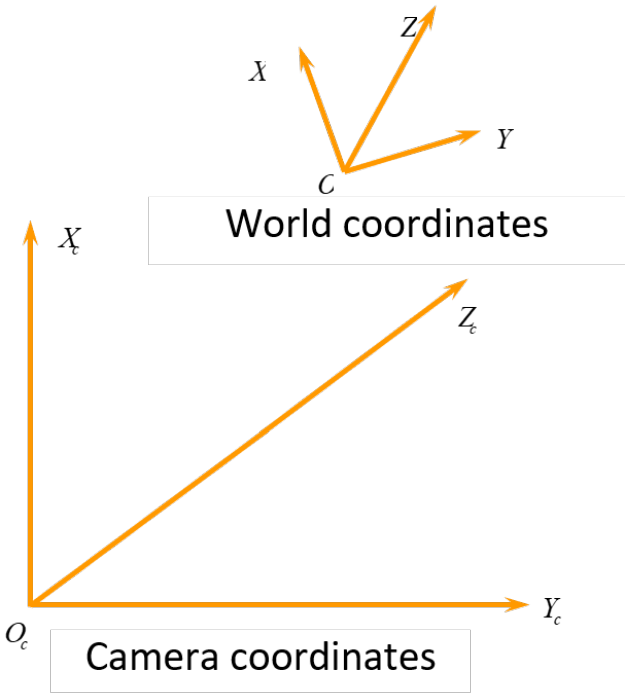


Figure 2.2. Rigid body transformation from world coordinates to camera coordinates.

The rigid body transformation from the world coordinates $(x\ y\ z)^T$ to the camera coordinates $(x_c\ y_c\ z_c)^T$ can be given by:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = R \begin{pmatrix} x \\ y \\ z \end{pmatrix} + t, \quad \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R & t \\ 0_3^T & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ x \\ 1 \end{pmatrix} \tag{2.1}$$

2.1.2 Normalization and distortion correction

$(x_u\ y_u)^T$ the ideal normalized image coordinates are defined as below.

$$\begin{pmatrix} x_u \\ y_u \\ 1 \end{pmatrix} = \frac{1}{z_c} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} \tag{2.2}$$

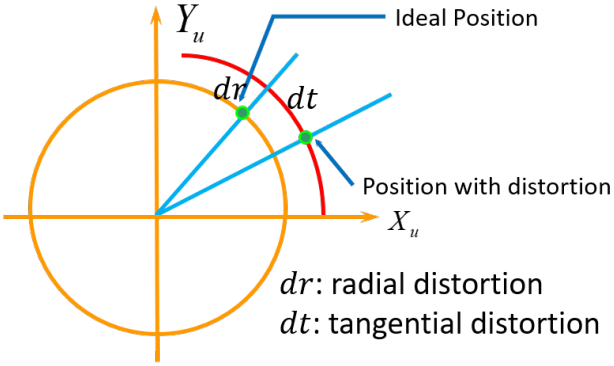


Figure 2.3. Distortion between ideal normalized image coordinates and real normalized image coordinates

Here, if we only consider the radial distortion, the distorted real normalized image coordinates $(\breve{x}\ \breve{y})^T$ can be computed as follows.

$$\begin{pmatrix} \breve{x} \\ \breve{y} \end{pmatrix} = \left(1 + k_1 r^2 + k_2 r^4\right) \begin{pmatrix} x_u \\ y_u \end{pmatrix}, \quad r = \sqrt{x_u^2 + y_u^2} \tag{2.3}$$

2.1.3 Perspective projection

The perspective projection from the real normalized image coordinates $(x_c\ y_c\ z_c)^T$ to the real image coordinates $(x_u\ y_u)^T$ can be given by:

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = f \begin{pmatrix} \breve{x} \\ \breve{y} \end{pmatrix} \tag{2.4}$$

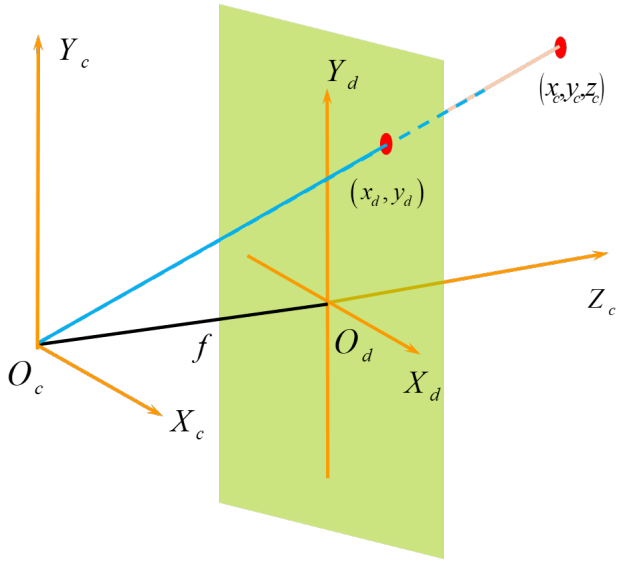


Figure 2.4. Perspective projection from real normalized image coordinates to real image coordinates.

2.1.4 Image digitalization

Image digitization is the process of converting real image coordinates into pixel coordinates (see Fig. 2.5).

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{dx} & -\frac{1}{dx \tan\theta} & u_0 \\ 0 & \frac{1}{dy \sin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_d \\ y_d \\ 1 \end{pmatrix} \tag{2.5}$$

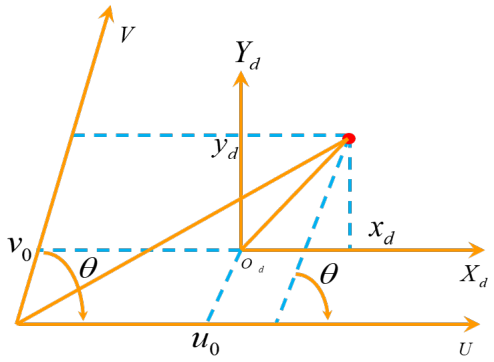


Figure 2.5. Image digitalization from real image coordinates to pixel coordinates.

The formula above describes the image digitalization from real image coordinates to pixel coordinates. Here k_1, k_2 indicate the parameters of radial distortion, dx, dy represent the physical scales for pixel on x, y axes, u_0, v_0 define the pixel coordinates of pinhole, θ is the skewed angle between x, y axes of the pixel.

2.1.5 Summary

It has been shown that we can describe the pinhole camera model with following formulas

$$
\begin{aligned}
&\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} R & t \end{pmatrix} \begin{pmatrix} x \\ y \\ x \\ 1 \end{pmatrix} \\
&\begin{pmatrix} \breve{x} \\ \breve{y} \\ 1 \end{pmatrix} = \frac{1}{z_c} \begin{pmatrix} 1 + k_1 r^2 + k_2 r^4 & 0 & 0 \\ 0 & 1 + k_1 r^2 + k_2 r^4 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix}, \quad r^2 = \left(\frac{x_c}{z_c}\right)^2 + \left(\frac{y_c}{z_c}\right)^2 \\
&\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{f}{dx} & -\frac{f}{dx \tan\theta} & u_0 \\ 0 & \frac{f}{dy \sin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \breve{x} \\ \breve{y} \\ 1 \end{pmatrix}
\end{aligned}
\tag{2.6}
$$

The intrinsic matrix A for camera is defined below, where α and β are the scaling factors between camera coordinates and pixel coordinates in x and y axes, c is the skewness in for two image axes.

$$
A = \begin{pmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{f}{dx} & -\frac{f}{dx \tan\theta} & u_0 \\ 0 & \frac{f}{dy \sin\theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.7}
$$

Furthermore, the rotation matrix R and the translation t indicate the extrinsic parameters between the world coordinates and the camera coordinates. In the meantime, k_1, k_2 represent the parameters of radial distortion for the camera lens.

2.2 Zhengyou Zhang's Calibration

The camera is calibrated using Zhang's method [24], where the basic equation are written as follows when the distortion of camera lens is neglected, where r_{i} is the i-th column of rotation matrix.

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = A \begin{pmatrix} R & t \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = A \begin{pmatrix} r_1 & r_2 & r_3 & t \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{2.8}$$

Because z of the points on the checkerboard are always 0, the equation is simplified as follows. Here $\tilde{m}$ is the homogeneous form of the pixel coordinates $(u \ v \ 1)^T$, $\tilde{M}$ is the homogeneous form of the world coordinates $(x \ y \ 1)^T$, and s indicates the depth to the camera pinhole, i.e. z_c in the pinhole model.

$$s\tilde{m} = A \begin{pmatrix} R & t \end{pmatrix} \tilde{M} = A \begin{pmatrix} r_1 & r_2 & t \end{pmatrix} \tilde{M} \tag{2.9}$$

2.2.1 Estimation of homography H

Let's suppose that there are N points on the $z = 0$ plane of world coordinates. Let's denote the pixel coordinate set of N points as $\mathcal{U} = (\tilde{m}_1, \ldots, \tilde{m}_N)$, and the world coordinate set as $\mathcal{X} = \left(\tilde{M}_1, \ldots, \tilde{M}_N\right)$. There is a relationship between the $\tilde{m}_k$ and $\tilde{M}_k$ related to the homography matrix H.

$$\tilde{m}_k = \frac{1}{\begin{pmatrix} h_{31} & h_{32} & h_{33} \end{pmatrix} \cdot \tilde{M}_k} H \tilde{M}_k, \quad H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} = \lambda A \begin{pmatrix} r_1 & r_2 & t \end{pmatrix} \tag{2.10}$$

Note that λ is an arbitrary nonzero number.

To improve the numerical stability of the computation of homography H, we normalize $\mathcal{U}, \mathcal{X}$ for preprocessing.

$$\begin{aligned} \mathcal{X}' &= \text{normalize}(\mathcal{X}) = \left(N_{\text{X}} \cdot \tilde{M}_1, \ldots, N_{\text{X}} \cdot \tilde{M}_N\right) = \left(\tilde{M}'_1, \ldots, \tilde{M}'_N\right) \\ \mathcal{U}' &= \text{normalize}(\mathcal{U}) = (N_{\text{U}} \cdot \tilde{m}_0, \ldots, N_{\text{U}} \cdot \tilde{m}_N) = (\tilde{m}'_1, \ldots, \tilde{m}'_N) \end{aligned} \tag{2.11}$$

where the normalization matrix $N_{\text{X}}, N_{\text{U}}$ are

$$N_{\text{X}} = \begin{pmatrix} \frac{1}{\sigma_x} & 0 & -\frac{\bar{x}}{\sigma_x} \\ 0 & \frac{1}{\sigma_y} & -\frac{\bar{y}}{\sigma_y} \\ 0 & 0 & 1 \end{pmatrix}, \quad N_{\text{U}} = \begin{pmatrix} \frac{1}{\sigma_u} & 0 & -\frac{\bar{u}}{\sigma_u} \\ 0 & \frac{1}{\sigma_v} & -\frac{\bar{v}}{\sigma_v} \\ 0 & 0 & 1 \end{pmatrix} \tag{2.12}$$

Here $\bar{x}, \bar{y}, \sigma_x, \sigma_y$ are defined as follows, $\bar{u}, \bar{v}, \sigma_u, \sigma_v$ are calculated in this way as well.

$$\begin{aligned} \bar{x} &\leftarrow \tfrac{1}{N} \textstyle\sum_{k=1}^{N} x_k, \quad \sigma_x^2 \leftarrow \tfrac{1}{N-1} \textstyle\sum_{k=1}^{N} (x_k - \bar{x})^2 \\ \bar{y} &\leftarrow \tfrac{1}{N} \textstyle\sum_{k=1}^{N} y_k, \quad \sigma_y^2 \leftarrow \tfrac{1}{N-1} \textstyle\sum_{k=1}^{N} (y_k - \bar{y})^2 \end{aligned} \tag{2.13}$$

The correspondence becomes

$$\tilde{m}'_k = \frac{1}{(h'_{31}\ h'_{32}\ h'_{33}) \cdot \tilde{M}'_k} H' \tilde{M}'_k, \quad H' \equiv N_{\text{U}} H {N_{\text{X}}}^{-1} \tag{2.14}$$

Then we denote $\text{h}' = (h'_{11}\ h'_{12}\ h'_{13}\ h'_{21}\ h'_{22}\ h'_{23}\ h'_{31}\ h'_{32}\ h'_{33})^T$ as the vector form of the homography matrix.

$$\begin{aligned} (x'_k\ y'_k\ 1\ 0\ 0\ 0\ -x'_k u'_k\ -y'_k u'_k\ -u'_k) \cdot \text{h}' = 0 \\ (0\ 0\ 0\ x'_k\ y'_k\ 1\ -x'_k v'_k\ -y'_k v'_k\ -v'_k) \cdot \text{h}' = 0 \end{aligned} \tag{2.15}$$

(2.15) holds for each pair of $\tilde{m}'_k, \tilde{M}'_k$ $(k = 1, \cdots, N)$, stack all the equations for $k = 1, \cdots, N$, yield a system of $2N$ homogeneous equations

$$\begin{pmatrix} x'_1 & y'_1 & 1 & 0 & 0 & 0 & -x'_1u'_1 & -y'_1u'_1 & -u'_1 \\ 0 & 0 & 0 & x'_1 & y'_1 & 1 & -x'_1v'_1 & -y'_1v'_1 & -v'_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_N & y'_N & 1 & 0 & 0 & 0 & -x'_Nu'_N & -y'_Nu'_N & -u'_N \\ 0 & 0 & 0 & x'_N & y'_N & 1 & -x'_Nv'_N & -y'_Nv'_N & -v'_N \end{pmatrix} \cdot \mathrm{h}' = \vec{0} \tag{2.16}$$

In the matrix-vector form, where W is a $2N \times 9$ matrix

$$W \cdot \mathrm{h}' = \vec{0} \tag{2.17}$$

We may assume that $2N > 9$, then we can solve the homogeneous system h' by finding the corresponding eigenvector for the smallest eigenvalue of W^TW. Afterwards, we can rearrange h' to the matrix form H'. Eventually, we obtain the homography H by de-normalization.

$$H = N_{\mathrm{U}}^{-1} H' N_{\mathrm{X}} \tag{2.18}$$

After all, we can refine the homography H by minimizing the error function below with the initial guess that we obtained in previous formula (2.18).

$$H \leftarrow \mathrm{argmin}_H \sum_{k=1}^{N} ||m_k - (\hat{u}_k \ \hat{v}_k)^T||^2 \quad \text{where } (\hat{u}_k \ \hat{v}_k \ 1)^T = \frac{1}{(h_{31}h_{32}h_{33}) \cdot \tilde{M}_k} H\tilde{M}_k \tag{2.19}$$

2.2.2 Estimation of intrinsic camera matrix A

By comparing (2.9) and (2.10), we formulate the relationship between A and the homography H, where λ' is an arbitrary nonzero number.

$$H = (h_1 \ h_2 \ h_3) = \lambda' A \, (r_1 \ r_2 \ t) \tag{2.20}$$

Based on $r_1^T \cdot r_2 = 0,\ r_1^T \cdot r_1 = r_2^T \cdot r_2 = 1$, this yields two constraints for A.

$$h_1^T A^{-T} A^T h_2 = (\lambda')^2 r_1^T \cdot r_2 = 0, \quad h_1^T A^{-T} A^T h_1 = h_2^T A^{-T} A^T h_2 = (\lambda')^2 r_2^T \cdot r_2 = (\lambda')^2 \quad (2.21)$$

We may assume that $B = \lambda A^{-T} A^{-1}$, where $\lambda \neq 0$ is an arbitrary nonzero number.

$$B = \lambda A^{-T} A^{-1} = \begin{pmatrix} B_0 & B_1 & B_3 \\ B_1 & B_2 & B_4 \\ B_3 & B_4 & B_5 \end{pmatrix} \quad (2.22)$$

which is symmetric and consists of 6 different values.

$$\begin{aligned} B_0/\lambda &= \tfrac{1}{\alpha^2}, & B_1/\lambda &= -\tfrac{c}{\alpha^2\beta}, \\ B_2/\lambda &= \tfrac{c^2}{\alpha^2\beta^2} + \tfrac{1}{\beta^2}, & B_3/\lambda &= \tfrac{v_0 c - u_0\beta}{\alpha^2\beta} \\ B_4/\lambda &= -\tfrac{c(v_0 c - u_0\beta)}{\alpha^2\beta^2} - \tfrac{v_0}{\beta^2}, & B_5/\lambda &= \tfrac{(v_0 c - u_0\beta)^2}{\alpha^2\beta^2} + \tfrac{v_0^2}{\beta^2} + 1 \end{aligned} \quad (2.23)$$

Using the vector form of B

$$\mathrm{b} = (B_0, B_1, B_2, B_3, B_4, B_5)^T \quad (2.24)$$

Rewrite Equation (2.21) as a pair of linear equations $\begin{pmatrix} \mathrm{v}_1^T \\ \mathrm{v}_2^T \end{pmatrix} \cdot \mathrm{b} = \vec{0}$, where the coefficients $\mathrm{v}_1 = (h_{11}h_{12},\ h_{11}h_{22} + h_{21}h_{12},\ h_{21}h_{22},\ h_{31}h_{12} + h_{11}h_{32},\ h_{31}h_{22} + h_{21}h_{32},\ h_{31}h_{32})^T$ and $\mathrm{v}_2 = (h_{11}^2 - h_{12}^2, 2(h_{11}h_{21} - h_{12}h_{22}), h_{21}^2 - h_{22}^2, 2(h_{11}h_{31} - h_{12}h_{32}), 2(h_{21}h_{31} - h_{22}h_{32}), h_{31}^2 - h_{32}^2)^T$

Stack M pairs of equations for each homography H of all M views. We denote V as the coefficient matrix V of size $2M \times 6$.

$$V \cdot \mathrm{b} = \vec{0} \quad (2.25)$$

We may assume that $2M \geq 6$, then we can solve the homogeneous system b by finding the corresponding eigenvector for the smallest eigenvalue of $V^T V$. Afterwards, we can rearrange

b to the matrix form B. With the Cholesky decomposition, we can conclude the intrinsic matrix A, but we have to obtain λ firstly.

$$\frac{\lambda^3}{(\alpha\beta)^2} = \det(\lambda A^{-T}A^{-1}) = \begin{vmatrix} B_0 & B_1 & B_3 \\ B_1 & B_2 & B_4 \\ B_3 & B_4 & B_5 \end{vmatrix} = B_0B_2B_5 - B_1^2B_5 - B_0B_4^2 + 2B_1B_3B_4 - B_2B_3^2 \tag{2.26}$$

Moreover, notice that the 3-rd row and 3-rd column element of A is 1, consider the second order principal submatrix of B by deleting 3-rd row and 3-rd column.

$$\frac{\lambda^2}{(\alpha\beta)^2} = \det\left(\lambda \begin{pmatrix} \alpha & c \\ 0 & \beta \end{pmatrix}^{-T} \begin{pmatrix} \alpha & c \\ 0 & \beta \end{pmatrix}^{-1}\right) = \begin{vmatrix} B_0 & B_1 \\ B_1 & B_2 \end{vmatrix} = B_0B_2 - B_1^2 \tag{2.27}$$

We denote w, d as the determinant of B and the second order principle submatrix of B, and therefore compute the nonzero arbitrary number λ.

$$\lambda = \frac{w}{d}, \text{where } w \equiv B_0B_2B_5 - B_1^2B_5 - B_0B_4^2 + 2B_1B_3B_4 - B_2B_3^2,\ d \equiv B_0B_2 - B_1^2 \tag{2.28}$$

With Equation (2.23), we propose the closed-form expression of A

$$\begin{aligned} \alpha &= \sqrt{w/\left(d \cdot B_0\right)} \\ \beta &= \sqrt{w/d^2 \cdot B_0} \\ c &= \sqrt{w/\left(d^2 \cdot B_0\right)} \cdot B_1 \\ u_0 &= \left(B_1B_4 - B_2B_3\right)/d \\ v_0 &= \left(B_1B_3 - B_0B_4\right)/d \end{aligned} \tag{2.29}$$

Therefore, we calibrate the intrinsic matrix A with the homograhy H for of all M views.

$$A = \begin{pmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.30}$$

2.2.3 Estimation of extrinsic camera parameters R, t

After obtaining the intrinsic matrix A and λ, we can calculate the corresponding extrinsic parameters R, t for each homography H of all M views respectively.

$$r_1 = \lambda \cdot A^{-1} \cdot h_1, \quad r_2 = \lambda \cdot A^{-1} \cdot h_2, \quad t = \lambda \cdot A^{-1} \cdot h_3, \tag{2.31}$$

Then we can normalize r_1, r_2 to make sure $|r_1| = 1$

$$r_1 \leftarrow \frac{r_1}{|r_1|}, \quad r_2 \leftarrow \frac{r_2}{|r_1|}, \quad t \leftarrow \frac{t}{|r_1|} \tag{2.32}$$

Notice that $R = (r_1\ r_2\ r_3)$ is orthonormal, r_3 is the cross product of r_1, r_2

$$r_3 = r_1 \times r_2 \tag{2.33}$$

The estimated $Q = (r_1\ r_2\ r_3)$ may not satisfy the constraint of rotation matrix $R^T R = I$. Then, we solve the best rotation matrix R with the smallest Frobenium norm of $R - Q$

$$\min_R \|R - Q\|_F^2 \quad \text{subject to } R^T R = I \tag{2.34}$$

Notice that

$$\|R - Q\|_F^2 = 3 + \operatorname{trace}\left(Q^T Q\right) - 2\operatorname{trace}\left(R^T Q\right) \tag{2.35}$$

With the singular value decomposition of $Q = U_Q S_Q V_Q^T$, and where $S_Q = \operatorname{diag}(\sigma_1, \sigma_2, \sigma_3)$. If we define an temporary orthogonal matrix $Z_Q = V_Q^T R^T U_Q$

$$\operatorname{trace}\left(R^T Q\right) = \operatorname{trace}\left(R^T U_Q S_Q V_Q^T\right) = \operatorname{trace}(Z_Q S_Q) = \sum_{\mathrm{i}=1}^{3} z_{\mathrm{ii}}\sigma_{\mathrm{i}} \leq \sum_{\mathrm{i}=1}^{3} \sigma_{\mathrm{i}} \tag{2.36}$$

The maximal is achieved when the orthogonal matrix $Z_Q = I$, that is $R = U_Q V_Q^T$. Consequently, we obtain the best rotation matrix.

$$R = U_Q V_Q^T \tag{2.37}$$

2.2.4 Estimation of camera lens distortion parameters k_1, k_2

Let $(\hat{u}\ \hat{v})^T$ be the ideal (nonobservable distortion-free) pixel image coordinates, and $(u\ v)^T$ the real pixel coordinates. With Equation (2.6) in camera pinhole model, we conclude

$$\begin{pmatrix} u-\hat{u} \\ v-\hat{v} \end{pmatrix} = \left(k_1 r^2 + k_2 r^4\right) \begin{pmatrix} \alpha & c \\ 0 & \beta \end{pmatrix} \begin{pmatrix} x_u \\ y_u \end{pmatrix}, \quad r = \sqrt{x_u^2 + y_u^2} \tag{2.38}$$

where the ideal normalized image coordinates $(x_u\ y_u)^T$ is defined below

$$\begin{pmatrix} x_u \\ y_u \\ 1 \end{pmatrix} \equiv \frac{1}{z_c} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \frac{1}{z_c} \left[R \begin{pmatrix} x \\ y \\ z \end{pmatrix} + t \right] \tag{2.39}$$

From the definition of $(\hat{u}\ \hat{v})^T$, we have

$$\begin{pmatrix} \hat{u}-u_0 \\ \hat{v}-v_0 \end{pmatrix} = \begin{pmatrix} \alpha & c \\ 0 & \beta \end{pmatrix} \begin{pmatrix} x_u \\ y_u \end{pmatrix}, \quad r = \sqrt{x_u^2 + y_u^2} \tag{2.40}$$

Thus, we find the following relationship

$$\begin{pmatrix} u-\hat{u} \\ v-\hat{v} \end{pmatrix} = \left(k_1 r^2 + k_2 r^4\right) \begin{pmatrix} \hat{u}-u_0 \\ \hat{v}-v_0 \end{pmatrix}, \quad r = \sqrt{x_u^2 + y_u^2} \tag{2.41}$$

Write in the form with $(k_1\ k_2)^T$ as an unknown vector

$$\begin{pmatrix} u-\hat{u} \\ v-\hat{v} \end{pmatrix} = \begin{pmatrix} (\hat{u}-u_0)r^2 & (\hat{u}-u_0)r^4 \\ (\hat{v}-v_0)r^2 & (\hat{v}-v_0)r^4 \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \end{pmatrix}, \quad r = \sqrt{x_u^2 + y_u^2} \tag{2.42}$$

When A, R, t are fixed and N points in M views given, we can stack equations to obtain total $2MN$ equations, in matrix form as $\mathrm{d} = D \cdot (k_1\ k_2)^T$. The solution with the least-squares method is given by

$$\begin{pmatrix} k_1 \\ k_2 \end{pmatrix} = \left(D^T D\right)^{-1} D^T \mathrm{d} \tag{2.43}$$

2.2.5 Non-linear refinement for all parameters

After all, we refine all the parameters $A, k_1, k_2, R_{\mathrm{i}}, t_{\mathrm{i}} (\mathrm{i} = 1, \cdots, N)$ by minimizing the error function defined below with the initial guess that we obtain in formulas (2.29), (2.37) and (2.43).

$$\underset{A,k_1,k_2,R_{\mathrm{i}},t_{\mathrm{i}}}{\operatorname{argmin}} \sum_{\mathrm{i}=1}^{M} \sum_{k=1}^{N} \left\| m_{\mathrm{i}k} - \hat{m}\left(A, k_1, k_2, R_{\mathrm{i}}, t_{\mathrm{i}}, M_{\mathrm{i}k}\right) \right\|^2 \quad \text{subject to } R_{\mathrm{i}}^T R_{\mathrm{i}} = I \tag{2.44}$$

But before minimizing the error function with the Levenberg-Marquardt algorithm, we have to rewrite the rotation matrix in a vector form to remove constraints $R_{\mathrm{i}}^T R_{\mathrm{i}} = I$. Because of the Rodrigues' rotation formula, the rotation matrix R with the unit rotation axis $\mathrm{u} = (u_1 \ u_2 \ u_3)^T$ and the rotation angle θ can be written as below

$$\begin{aligned} R &= I + \sin\theta W + (1 - \cos\theta) W^2 = I + \sin\theta W + (1 - \cos\theta)\left[\mathrm{u}\mathrm{u}^T - I\right] \\ &= \mathrm{u}\mathrm{u}^T + \cos\theta\left[I - \mathrm{u}\mathrm{u}^T\right] + \sin\theta W = \exp(\theta W) \end{aligned} \tag{2.45}$$

where W is the matrix form of $\mathrm{u}\times$, and $W^2 = \mathrm{u}\mathrm{u}^T - I, Wz = \vec{0}$.

$$W \equiv \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix} \tag{2.46}$$

Thus, we can represent R with an vector $\rho = \theta \mathrm{u}$. The relationship between the rotation matrix R and the vector ρ is

$$\begin{aligned} \operatorname{trace}(R) &= \operatorname{trace}\left(\cos\theta I + \sin\theta W + (1 - \cos\theta)\mathrm{u}\mathrm{u}^T\right) = 3\cos\theta + (1 - \cos\theta) \\ R - R^T &= 2\sin\theta W \end{aligned} \tag{2.47}$$

Similarly, we can convert the rotation matrix R_{i} to the rotation vector ρ_{i} for all the M views. The optimization problem in equation (2.44) is converted to the following problem.

$$\underset{A,k_1,k_2,\rho_{\mathrm{i}},t_{\mathrm{i}}}{\operatorname{argmin}} \sum_{\mathrm{i}=1}^{M} \sum_{k=1}^{N} \left\| m_{\mathrm{i}k} - \hat{m}\left(A, k_1, k_2, \rho_{\mathrm{i}}, t_{\mathrm{i}}, M_{\mathrm{i}k}\right) \right\|^2 \tag{2.48}$$

2.3 Triangulation

2.3.1 The expression of the 3D point M

For conciseness of illustration, we define following symbols in Table 2.2 below

Table 2.2. Symbol table for triangulation

symbol	definition
r'_1	unit direction vector of ray 1
r'_2	unit direction vector of ray 2
I_1	intersection point of the camera 1 pixel plane and ray 1
I_2	intersection point of the camera 2 pixel plane and ray 2
M_1	the closest point on ray 1 to ray 2
M_2	the closest point on ray 2 to ray 1
M	the mid point of M_1 and M_2
k_1	the distance from I_1 to M_1
k_2	the distance from I_2 to M_2

Our goal is to represent the mid point $M = \frac{M_1+M_2}{2}$ with the known parameters r'_1, r'_2, I_1, I_2. We start from the definitions of M_1, M_2, the vector $M_1 - M_2$ must be perpendicular to r'_1, r'_2

$$\begin{aligned} r'^T_1(M_1 - M_2) = 0 \\ r'^T_2(M_1 - M_2) = 0 \end{aligned} \tag{2.49}$$

Because the distance from I_1 to M_1 and from I_2 to M_2 are denoted by k_1, k_2

$$\begin{aligned} k_1 r'_1 \equiv M_1 - I_1 \\ k_2 r'_2 \equiv M_2 - I_2 \end{aligned} \tag{2.50}$$

To solve k_1, k_2, we firstly replace M_1, M_2 in (2.49) with the k_1, k_2 in (2.50)

$$\begin{aligned} r'^T_1\Big([I_1 - I_2] + k_1 r'_1 - k_2 r'_2\Big) = 0 \\ r'^T_2\Big([I_1 - I_2] + k_1 r'_1 - k_2 r'_2\Big) = 0 \end{aligned} \tag{2.51}$$

Equation (2.51) is equivalent to

$$\begin{aligned}[r'^T_1 r'_1]k_1 - [r'^T_1 r'_2]k_2 &= -r'^T_1[I_1 - I_2] \\ -[r'^T_2 r'_1]k_1 + [r'^T_2 r'_2]k_2 &= r'^T_2[I_1 - I_2]\end{aligned} \tag{2.52}$$

We write Equation (2.52) in the matrix form to solve both k_1 and k_2

$$\begin{pmatrix} [r'^T_1 r'_1] & -[r'^T_1 r'_2] \\ -[r'^T_1 r'_2] & [r'^T_2 r'_2] \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} = \begin{pmatrix} -r'^T_1[I_1 - I_2] \\ r'^T_2[I_1 - I_2] \end{pmatrix} \tag{2.53}$$

With Cramer's rule, the expressions of k_1, k_2 with the known parameters r'_1, r'_2, I_1, I_2 are

$$\begin{aligned} k_1 &= \frac{\begin{vmatrix} -r'^T_1[I_1 - I_2] & -[r'^T_1 r'_2] \\ r'^T_2[I_1 - I_2] & [r'^T_2 r'_2] \end{vmatrix}}{\begin{vmatrix} [r'^T_1 r'_1] & -[r'^T_1 r'_2] \\ -[r'^T_1 r'_2] & [r'^T_2 r'_2] \end{vmatrix}} = r'^T_2 \left[\frac{1}{1 - [r'^T_1 r'_2]^2} \Big(r'_1 r'^T_2 - r'_2 r'^T_1 \Big)[I_1 - I_2] \right] \\ k_2 &= \frac{\begin{vmatrix} [r'^T_1 r'_1] & -r'^T_1[I_1 - I_2] \\ -[r'^T_1 r'_2] & r'^T_2[I_1 - I_2] \end{vmatrix}}{\begin{vmatrix} [r'^T_1 r'_1] & -[r'^T_1 r'_2] \\ -[r'^T_1 r'_2] & [r'^T_2 r'_2] \end{vmatrix}} = r'^T_1 \left[\frac{1}{1 - [r'^T_1 r'_2]^2} \Big(r'_1 r'^T_2 - r'_2 r'^T_1 \Big)[I_1 - I_2] \right] \end{aligned} \tag{2.54}$$

Consequently, the midpoint M is represented with the known parameters r'_1, r'_2, I_1, I_2

$$\begin{aligned} M \equiv \frac{M_1 + M_2}{2} &= \frac{I_1 + I_2}{2} + \frac{k_1 r'_1 + k_2 r'_2}{2} \\ &= \frac{I_1 + I_2}{2} + \frac{1}{2}\Big(r'_1 r'^T_2 + r'_2 r'^T_1 \Big) \left[\frac{1}{1 - [r'^T_1 r'_2]^2} \Big(r'_1 r'^T_2 - r'_2 r'^T_1 \Big)[I_1 - I_2] \right] \\ &= \frac{I_1 + I_2}{2} + \frac{1}{2} \frac{1}{1 - [r'^T_1 r'_2]^2} \Big[r'_1 [r'^T_2 r'_1] r'^T_2 - r'_2 [r'^T_1 r'_2] r'^T_1 \Big][I_1 - I_2] \\ &= \frac{I_1 + I_2}{2} + \frac{1}{2} \frac{[r'^T_1 r'_2]}{1 - [r'^T_1 r'_2]^2} \Big(r'_1 r'^T_2 - r'_2 r'^T_1 \Big)[I_1 - I_2] \end{aligned} \tag{2.55}$$

2.3.2 The other expression of M with cross product

Consider the cross product of $a = (a_1\ a_2\ a_3)^T$ and $b = (b_1\ b_2\ b_3)^T$, and its cross product in matrix form

$$a \times b = \begin{pmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{pmatrix} \tag{2.56}$$

Rearrange the vector $a \times b$ in the matrix form

$$(a \times b)\times = \begin{pmatrix} 0 & a_2b_1 - a_1b_2 & a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 & 0 & a_3b_2 - a_2b_3 \\ a_1b_3 - a_3b_1 & a_2b_3 - a_3b_2 & 0 \end{pmatrix} = ba^T - ab^T \tag{2.57}$$

We represent the 3D point M with cross product using $\left(r'_1 r'^T_2 - r'_2 r'^T_1\right) = -(r'_1 \times r'_2)\times$

$$M = \frac{I_1 + I_2}{2} - \frac{1}{2}\frac{[r'^T_1 r'_2]}{1 - [r'^T_1 r'_2]^2}(r'_1 \times r'_2) \times [I_1 - I_2] \tag{2.58}$$

2.4 U-Net

2.4.1 Architecture of U-Net: encoder and decoder

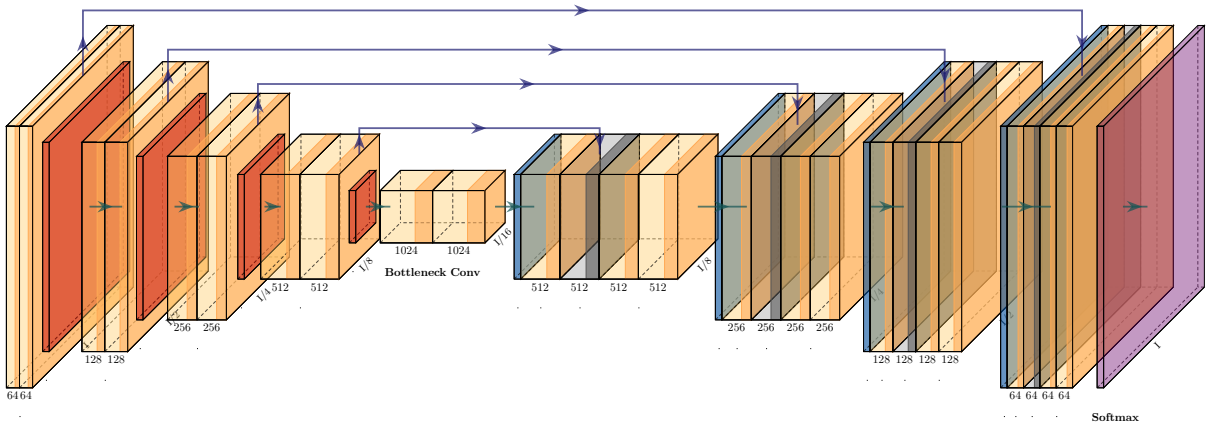


Figure 2.6. U-Net Architecture.

The U-Net architecture includes the encoder and the decoder. The encoder is composed of 4 down-sampling modules, and the decoder consists of 4 up-sampling module. For the down-sampling module, it is made up with the repeated combination of a convolution layer, a batch normalization layer and a ReLU layer, which is followed by max pooling layer with stride size=2. All the convolution layers are set up with stride size=3, kernel size=1 and padding size=1. For the up-sampling module, it is made up with a transposed convolution layer, a concatenation with the correspondingly feature map from the skip-connection path and the same repeated combination of a convolution layer, a batch normalization layer and a ReLU layer. Furthermore, there are the same repeated combination of a convolution layer, a batch normalization layer and a ReLU layer between the encoder and decoder.

Note that we can address the relationship of the input feature map size and the output feature map size of a convolution layer as follows.

$$o = \left\lfloor \frac{\mathrm{i} + 2p - k}{s} \right\rfloor + 1 \tag{2.59}$$

where i, o indicate the height/width of the input feature map and the output feature map respectively. In addition, kernel size, stride size, and padding size of a convolution layer are denoted by k, s, p.

Furthermore, we can also write down the relationship of the input feature map size and the output feature map size of a transposed convolution layer, i.e. deconvolution layer.

$$o' = (\mathrm{i}' - 1)\, s + k - 2p \tag{2.60}$$

where i', o' indicate the height/width of the input feature map and the output feature map respectively. In addition, kernel size, stride size, and padding size of a transposed convolution layer are denoted by k, s, p.

2.4.2 Optimization method

RMSprop optimization method was firstly proposed by Geoffrey Hinton in his Coursera course, as one of the adaptive learning rate methods, and extension of Stochastic Gradient Descent (SGD) and momentum method.

$$\begin{aligned} E\left[\boldsymbol{g^2}\right](t) &= \alpha E\left[\boldsymbol{g^2}\right](t-1) + (1-\alpha)\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} \odot \frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} \\ \boldsymbol{w}(t) &= \boldsymbol{w}(t-1) - \frac{\eta}{\sqrt{E\left[\boldsymbol{g^2}\right](t)+\epsilon}} \odot \frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} \end{aligned} \tag{2.61}$$

Where $\odot$ means the element-wise multiplication, η is the learning rate, α is the moving average parameter, the default value of ϵ is 10^{-8}, $E\left[\boldsymbol{g^2}\right]$ represents the moving average of squared gradients and $\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}$ means the gradient of loss function $J(\boldsymbol{w})$ with respect to $\boldsymbol{w}$ the weights of neural networks.

If we replace the loss function $J(\boldsymbol{w})$ with the regularization loss during L2 regularization $\hat{J}(\boldsymbol{w})$ in (2.61), here λ is the weight decay parameter.

$$\begin{aligned} \hat{J}(\boldsymbol{w}) &= J(\boldsymbol{w}) + \tfrac{\lambda}{2}\|\boldsymbol{w}\|_2^2 \\ \tfrac{\partial \hat{J}(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \tfrac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} + \lambda \boldsymbol{w} \end{aligned} \tag{2.62}$$

Thus, equation (2.61) becomes the equation (2.63) below

$$\begin{aligned} E\left[\boldsymbol{g^2}\right](t) &= \alpha E\left[\boldsymbol{g^2}\right](t-1) + (1-\alpha)\left(\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} + \lambda \boldsymbol{w}\right) \odot \left(\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} + \lambda \boldsymbol{w}\right) \\ \boldsymbol{w}(t) &= \boldsymbol{w}(t-1) - \frac{\eta}{\sqrt{E\left[\boldsymbol{g^2}\right](t)+\epsilon}} \odot \left(\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} + \lambda \boldsymbol{w}\right) \end{aligned} \tag{2.63}$$

For the purpose of making the values of $\boldsymbol{w}(t) - \boldsymbol{w}(t-1)$ more stable, we introduce the momentum factor β to ensure

$$\left[\boldsymbol{w}(t) - \boldsymbol{w}(t-1)\right] = \beta\left[\boldsymbol{w}(t-1) - \boldsymbol{w}(t-2)\right] - \frac{\eta}{\sqrt{E\left[\boldsymbol{g^2}\right](t)+\epsilon}} \odot \left(\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} + \lambda \boldsymbol{w}\right) \tag{2.64}$$

Thus, it leads to

$$\begin{aligned}
&E\left[\boldsymbol{g^2}\right](t) = \alpha E\left[\boldsymbol{g^2}\right](t-1) + (1-\alpha)\left(\tfrac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} + \lambda \boldsymbol{w}\right) \odot \left(\tfrac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} + \lambda \boldsymbol{w}\right) \\
&\boldsymbol{v}(t) = \beta \boldsymbol{v}(t-1) + \tfrac{1}{\sqrt{E\left[\boldsymbol{g^2}\right](t)+\epsilon}} \odot \left(\tfrac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} + \lambda \boldsymbol{w}\right) \\
&\boldsymbol{w}(t) = \boldsymbol{w}(t-1) - \eta \boldsymbol{v}(t)
\end{aligned} \tag{2.65}$$

2.4.3 Criterion for loss function

Let's consider the binary classification problem, there are only the classes 0 and 1. Here $\boldsymbol{y} = (y^{(1)}, \cdots, y^{(L)})^T,\ y^{(k)} \in \{0, 1\}$ represents the probability of belonging to the class 1 for the corresponding input $\boldsymbol{x} = (x^{(1)}, \cdots, x^{(L)})^T$. We hope to find a mapping of probability with neural networks, where $\boldsymbol{w}$ is the weights of the neural networks.

$$\hat{\boldsymbol{y}} = (\hat{y}^{(1)}, \cdots, \hat{y}^{(L)})^T = \left(\hat{P}_{\boldsymbol{w}}^{(1)}(y^{(1)} = 1|\boldsymbol{x}), \cdots, \hat{P}_{\boldsymbol{w}}^{(L)}(y^{(L)} = 1|\boldsymbol{x})\right) \tag{2.66}$$

We may assume that each component in $\boldsymbol{y} = (y^{(1)}, \cdots, y^{(L)})^T,\ y^{(k)} \in \{0, 1\}$ is conditional independent to $\boldsymbol{x} = (x^{(1)}, \cdots, x^{(L)})^T$.

$$\hat{P}_{\boldsymbol{w}}(\boldsymbol{y}|\boldsymbol{x}) = \prod_{k=1}^{L} \hat{P}_{\boldsymbol{w}}^{(k)}(y^{(k)}|\boldsymbol{x}) \tag{2.67}$$

Moreover, notice that we can write $\hat{P}_{\boldsymbol{w}}^{(k)}(y^{(k)}|\boldsymbol{x})$ with all the possible options for $y^{(k)}$.

$$\hat{P}_{\boldsymbol{w}}^{(k)}(y^{(k)}|\boldsymbol{x}) = \hat{P}_{\boldsymbol{w}}^{(k)}(y^{(k)} = 1|\boldsymbol{x})^{y^{(k)}} \cdot \hat{P}_{\boldsymbol{w}}^{(k)}(y^{(k)} = 0|\boldsymbol{x})^{1-y^{(k)}} = \left(\hat{y}^{(k)}\right)^{y^{(k)}} \cdot \left(1 - \hat{y}^{(k)}\right)^{1-y^{(k)}} \tag{2.68}$$

Thus, we conclude

$$\hat{P}_{\boldsymbol{w}}(\boldsymbol{y}|\boldsymbol{x}) = \prod_{k=1}^{L} \left(\hat{y}^{(k)}\right)^{y^{(k)}} \cdot \prod_{k=1}^{L} \left(1 - \hat{y}^{(k)}\right)^{1-y^{(k)}} \tag{2.69}$$

Let's take a batch of N samples $(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{y}_{\mathrm{i}}),\ \mathrm{i} = 1, \cdots, N$ for the variables $(\boldsymbol{x}, \boldsymbol{y})$. We may assume for each pair of $(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{y}_{\mathrm{i}})$, they are independent to each other. Thus, we can write down the likelihood function to maximize $\hat{P}_{\boldsymbol{w}}(\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N | \boldsymbol{x}_1, \cdots, \boldsymbol{x}_N)$

$$\hat{P}_{\boldsymbol{w}}(\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N | \boldsymbol{x}_1, \cdots, \boldsymbol{x}_N) = \prod_{\mathrm{i}=1}^{N} \hat{P}_{\boldsymbol{w}}(\boldsymbol{y}_{\mathrm{i}} | \boldsymbol{x}_{\mathrm{i}}) = \prod_{\mathrm{i}=1}^{N} \left(\prod_{k=1}^{L} \left(\hat{y}_{\mathrm{i}}^{(k)} \right)^{y_{\mathrm{i}}^{(k)}} \cdot \prod_{k=1}^{L} \left(1 - \hat{y}_{\mathrm{i}}^{(k)} \right)^{1 - y_{\mathrm{i}}^{(k)}} \right) \tag{2.70}$$

Thus, the negative logarithm of $\hat{P}_{\boldsymbol{w}}(\boldsymbol{y}_1, \cdot, \boldsymbol{y}_N | \boldsymbol{x}_1, \cdot, \boldsymbol{x}_N)$ is

$$\begin{aligned} -\log \left(\hat{P}_{\boldsymbol{w}}(\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N | \boldsymbol{x}_1, \cdots, \boldsymbol{x}_N) \right) &= -\sum_{\mathrm{i}=1}^{N} \left(\sum_{k}^{L} y_{\mathrm{i}}^{(k)} \log \left(\hat{y}_{\mathrm{i}}^{(k)} \right) + \sum_{k}^{L} (1 - y_{\mathrm{i}}^{(k)}) \log \left(1 - \hat{y}_{\mathrm{i}}^{(k)} \right) \right) \\ &= -\sum_{\mathrm{i}=1}^{N} \left(\boldsymbol{y}_{\mathrm{i}}^{T} \log \left(\hat{\boldsymbol{y}}_{\mathrm{i}} \right) + (\vec{1} - \boldsymbol{y}_{\mathrm{i}})^{T} \log \left(\vec{1} - \hat{\boldsymbol{y}}_{\mathrm{i}} \right) \right) \end{aligned} \tag{2.71}$$

In the end, we obtain the cross entropy definition by dividing it with N. We use the cross entropy as the criterion for loss function in the minimization problem, i.e. the maximization problem for $\hat{P}_{\boldsymbol{w}}(\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N | \boldsymbol{x}_1, \cdots, \boldsymbol{x}_N)$

$$J(\boldsymbol{w}) \equiv -\frac{1}{N} \sum_{\mathrm{i}=1}^{N} \left(\boldsymbol{y}_{\mathrm{i}}^{T} \log \left(\hat{\boldsymbol{y}}_{\mathrm{i}} \right) + (\vec{1} - \boldsymbol{y}_{\mathrm{i}})^{T} \log \left(\vec{1} - \hat{\boldsymbol{y}}_{\mathrm{i}} \right) \right) \tag{2.72}$$

Similarly, we can define the cross entropy for multiple classes, where the number of classes $C \geq 3$ and $\sum_{c=1}^{C} \hat{\boldsymbol{y}}_{\mathrm{i}}[c] = \sum_{c=1}^{C} \boldsymbol{y}_{\mathrm{i}}[c] = \vec{1}$ always holds.

$$J(\boldsymbol{w}) \equiv -\frac{1}{N} \sum_{\mathrm{i}=1}^{N} \sum_{c=1}^{C} \left(\boldsymbol{y}_{\mathrm{i}}^{T}[c] \log \left(\hat{\boldsymbol{y}}_{\mathrm{i}}[c] \right) + (\vec{1} - \boldsymbol{y}_{\mathrm{i}}[c])^{T} \log \left(\vec{1} - \hat{\boldsymbol{y}}_{\mathrm{i}}[c] \right) \right) \tag{2.73}$$

The loss function of cross entropy can be used to regulates voxelwise binary prediction.

2.4.4 Metrics of evaluation

Let's denote the ground truth image mask with $\boldsymbol{y} = (y^{(1)}, \cdots, y^{(L)})^{T},\ y^{(k)} \in \{0, 1\}$. In addition, the corresponding predicted output mask is $\hat{\boldsymbol{y}} = (\hat{y}^{(1)}, \cdots, \hat{y}^{(L)})^{T},\ \hat{y}^{(k)} \in [0, 1]$

We can use the dice coefficient to measure the similarity between the ground truth mask and the predicted mask generated by neural networks.

$$\text{dice coefficient}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = \frac{2 \cdot \boldsymbol{y}^T \hat{\boldsymbol{y}} + \epsilon}{\boldsymbol{y}^T \boldsymbol{y} + \hat{\boldsymbol{y}}^T \hat{\boldsymbol{y}} + \epsilon} \tag{2.74}$$

In addition, the dice coefficient also can be written in the other form

$$\text{dice coefficient}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = \frac{2 \cdot \sum\left(\boldsymbol{y} \& \left(\hat{\boldsymbol{y}} > 0.5\right)\right) + \epsilon}{\sum \boldsymbol{y} + \sum\left(\hat{\boldsymbol{y}} > 0.5\right) + \epsilon} \tag{2.75}$$

The corresponding metric, i.e. dice is defined as below

$$\text{dice}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = 1 - \text{dice coefficient}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = \frac{\left(\boldsymbol{y} - \hat{\boldsymbol{y}}\right)^T \left(\boldsymbol{y} - \hat{\boldsymbol{y}}\right)}{\boldsymbol{y}^T \boldsymbol{y} + \hat{\boldsymbol{y}}^T \hat{\boldsymbol{y}} + \epsilon} \tag{2.76}$$

When ϵ is a very small positive number and the ground truth mask $\boldsymbol{y}$ is nonzero, the closer the dice is to 0, the closer the dice coefficient is to 1, the closer the predicted mask $\hat{\boldsymbol{y}}$ is to the ground truth mask $\boldsymbol{y}$, and the better performance the neural network model has.

The other metric that could be used to measure the similarity the ground truth mask and the predicted mask generated is Intersection over Union (IoU), i.e. Jaccard distance.

$$\text{IoU} = J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{2.77}$$

We can write IoU of the predicted mask $\hat{\boldsymbol{y}}$ and the ground truth mask $\boldsymbol{y}$ as a differentiable function as below

$$\text{IoU}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = \frac{\boldsymbol{y}^T \hat{\boldsymbol{y}} + \epsilon}{\sum \boldsymbol{y} + \sum \hat{\boldsymbol{y}} - \boldsymbol{y}^T \hat{\boldsymbol{y}} + \epsilon} \tag{2.78}$$

In addition, IoU also can be written as in the other form

$$\text{IoU}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = \frac{\sum\left(\boldsymbol{y} \;\&\; \left(\hat{\boldsymbol{y}} > 0.5\right)\right) + \epsilon}{\sum\left(\boldsymbol{y} \mid \left(\hat{\boldsymbol{y}} > 0.5\right)\right) + \epsilon} \tag{2.79}$$

3. METHODOLOGY

3.1 Structured light vision system with multiple laser emitters and multiple cameras

The structured light system measurement system is shown in Fig. 3.1. It is composed of N laser emitters with green, red, and other colors, a holding platform and K high-resolution cameras. As demonstrated in Fig. 3.1, both the camera coordinates and world coordinates are represented by $o_{ck}x_{ck}y_{ck}z_{ck} \quad k = 1, 2 \cdots, K$ and $oxyz$ respectively. In addition, we may assume that the projections of the triangular planes on the horizontal planes exist and can be captured by the cameras.

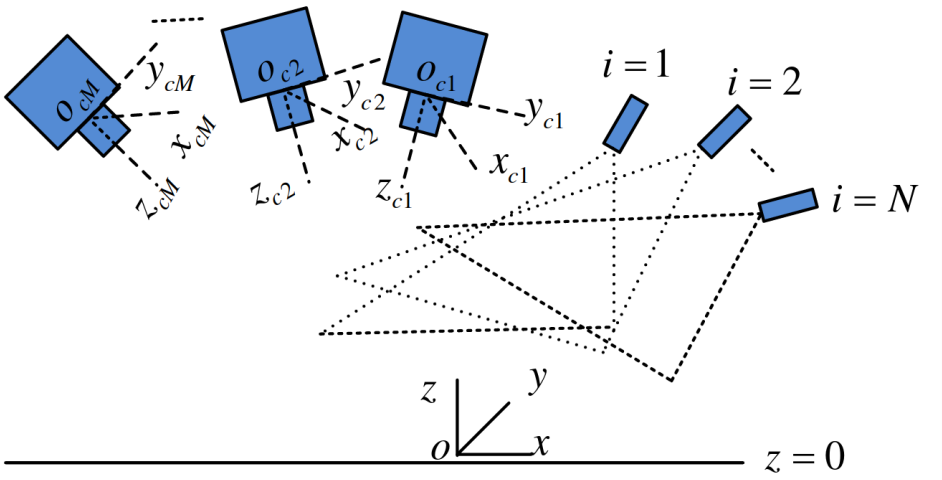


Figure 3.1. Height measurement system.

For conciseness of illustration, we define following symbols in Table 3.1.

Table 3.1. Symbol table for the structured light system

symbol	definition
$X = (x \ y \ z)^T$	the point in the world coordinate
$\bar{X} = (x \ y \ z \ 1)^T$	the homogeneous form of X
$X_{ck} = (x_{ck} \ y_{ck} \ z_{ck})^T$	the point in the k-th camera coordinate
$\bar{X}_{ck} = (x_{ck} \ y_{ck} \ z_{ck} \ 1)^T$	the homogeneous form of X_{ck}
$I_{pk} = (u_k \ v_k)^T$	the point in the pixel coordinate from the k-th camera
$\bar{I}_{pk} = (u_k \ v_k \ 1)^T$	the homogeneous form of I_{pk}

3.1.1 Measurement method

In our project, we can formulate the expression of camera coordinates for camera k as below (3.1) and (3.2) based on the camera pinhole model.

$$X_{ck} = z_{ck} A_k^{-1} \bar{I}_{pk} \tag{3.1}$$

$$\bar{X}_{ck} = \begin{pmatrix} R_k & t_k \\ 0 & 1 \end{pmatrix} \bar{X} \tag{3.2}$$

where the intrinsic matrix A_k of k-th camera can be expressed in the form of below

$$A_k = \begin{pmatrix} \alpha_k & c_k & u_{0k} \\ 0 & \beta_k & v_{0k} \\ 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

where for camera k: α_k and β_k are the scaling factors between camera coordinates and pixel coordinates in x and y axes. Besides, u_{0k} and v_{0k} are the pixel coordinates of the optical axis. The c_k represents the skewness between the pixel axes and the real image axes. This $[R_k \ t_k] = [r_{1k} \ r_{2k} \ r_{3k} \ t_k]$ denotes the rotation and translation with respect to the camera coordinates and the world coordinates.

For each characteristic j-th point on the i-th laser plane, it can be denoted by $\bar{X}_{clk}(\mathrm{i},\mathrm{j}) = (x_{cLk}(\mathrm{i},\mathrm{j}) \quad y_{cLk}(\mathrm{i},\mathrm{j}) \quad z_{cLk}(\mathrm{i},\mathrm{j}) \quad 1)^T$. Furthermore, the i-th laser plane can be designated by $\boldsymbol{\pi}_{\mathrm{i}} = (a_{\mathrm{i}} \quad b_{\mathrm{i}} \quad c_{\mathrm{i}} \quad -1)^T$. We can derive the following equations

$$X_{clk}(\mathrm{i},\mathrm{j})/z_{ck}(\mathrm{i},\mathrm{j}) = A_k^{-1} \bar{I}_{pk}(\mathrm{i},\mathrm{j}) \tag{3.4}$$

$$\boldsymbol{\pi}_{\mathrm{i}}^T \bar{X}_{cLk}(\mathrm{i},\mathrm{j}) = 0, \quad \mathrm{j} = 1, 2, \cdots, J \tag{3.5}$$

For each given projected point $\bar{I}_{pk}(\mathrm{i},\mathrm{j})$, with (3.4), we can achieve

$$\frac{X_{cLk}(\mathrm{i},\mathrm{j})}{z_{cLk}(\mathrm{i},\mathrm{j})} = \begin{pmatrix} \frac{x_{cLk}(\mathrm{i},\mathrm{j})}{z_{cLk}(\mathrm{i},\mathrm{j})} & \frac{y_{cLk}(\mathrm{i},\mathrm{j})}{z_{cLk}(\mathrm{i},\mathrm{j})} & 1 \end{pmatrix}^T \tag{3.6}$$

For the calibrated triangular plane i, with (3.5), we can eventually produce

$$z_{cLk}(\mathrm{i},\mathrm{j}) = \frac{1}{\left(a_{\mathrm{i}} \frac{x_{cLk}(\mathrm{i},\mathrm{j})}{z_{cLk}(\mathrm{i},\mathrm{j})} + b_{\mathrm{i}} \frac{y_{cLk}(\mathrm{i},\mathrm{j})}{z_{cLk}(\mathrm{i},\mathrm{j})} + c_{\mathrm{i}}\right)} \tag{3.7}$$

From (3.1), $X_{ck}(\mathrm{i},\mathrm{j}) = X_{cLk}(\mathrm{i},\mathrm{j})$ can be computed. Finally, we obtain

$$\bar{X}(\mathrm{i},\mathrm{j}) = \begin{pmatrix} R_k & t_k \\ 0 & 1 \end{pmatrix}^{-1} \bar{X}_{ck}(\mathrm{i},\mathrm{j}) \tag{3.8}$$

With the calibrated intrinsic A and extrinsic parameters R_k, t_k and all the laser planes $\boldsymbol{\pi}_{\mathrm{i}k}$, we can perform the 3D reconstruction successfully.

3.1.2 Calibration of the laser plane π

We describe the camera model in the form of Zhang's method [24], where the pixel coordinates and the world coordinates are represented by $\tilde{m} = (u\ v\ 1)^T \quad M = (x\ y\ z\ 1)^T$. With a pinhole camera model, we can find the relationship between pixel coordinates and the world coordinates as below

$$s\tilde{m} = A_k \begin{pmatrix} R_k & t_k \end{pmatrix} M = A_k \begin{pmatrix} r_{1k}\ r_{2k}\ r_{3k}\ t_k \end{pmatrix} M \tag{3.9}$$

We may assume $z = 0$ (floor plane) in the model, thus (3.9) becomes

$$s\tilde{m} = A_k \begin{pmatrix} r_{1k}\ r_{2k}\ t_k \end{pmatrix} \tilde{M} \tag{3.10}$$

where s indicates the depth to the camera pinhole and $\tilde{M} = (x\ y\ 1)^T$. The details of Zhang's method to calibrate the intrinsic matrix A and extrinsic parameters R, t can be found in [24].

Each laser plane $\boldsymbol{\pi}_{\mathrm{i}} = (a_{\mathrm{i}}\ b_{\mathrm{i}}\ c_{\mathrm{i}}\ -1)\,,\ \mathrm{i} = 1, \cdots, N$ needed in [5] can be calibrated independently. As shown in Fig. 3.1, the floor plane $z = 0$ in the world coordinate can be indicated by the plane equation coefficients $(0\ 0\ 1\ 0)^T$. Thus, we can compute the plane equation coefficients of the floor plane in the camera coordinate as

$$\boldsymbol{\pi}_{0k} = \begin{pmatrix} R_k & t_k \\ 0 & 1 \end{pmatrix}^{-T} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} a_{0k} & b_{0k} & c_{0k} & d_{0k} \end{pmatrix}^T \tag{3.11}$$

Furthermore, the projected j-th points $X_{ck}(\mathrm{i},\mathrm{j})$ must be on the plane of the checkerboard in the camera coordinates during calibration, namely,

$$\boldsymbol{\pi}_{0k}^T X_{ck}(\mathrm{i},\mathrm{j}) = 0 \text{ for } \mathrm{j} = 1, 2, \cdots, J \tag{3.12}$$

In other words, we can write it in a different form

$$\begin{gathered} z_{ck}(\mathrm{i},\mathrm{j}) \left(a_{0k} \frac{x_{ck}(\mathrm{i},\mathrm{j})}{z_{ck}(\mathrm{i},\mathrm{j})} + b_{0k} \frac{y_{ck}(\mathrm{i},\mathrm{j})}{z_{ck}(\mathrm{i},\mathrm{j})} + c_{0k} \right) = -d_{0k} \\ \text{for } \mathrm{j} = 1, 2, \cdots, J \end{gathered} \tag{3.13}$$

Moreover, both $\frac{x_{ck}(\mathrm{i},\mathrm{j})}{z_{ck}(\mathrm{i},\mathrm{j})}$ and $\frac{y_{ck}(\mathrm{i},\mathrm{j})}{z_{ck}(\mathrm{i},\mathrm{j})}$ in (3.13) can be computed with the projected pixel points

$$\begin{gathered} \left(\frac{x_{ck}(\mathrm{i},\mathrm{j})}{z_{ck}(\mathrm{i},\mathrm{j})} \quad \frac{y_{ck}(\mathrm{i},\mathrm{j})}{z_{ck}(\mathrm{i},\mathrm{j})} \quad 1 \right)^T = \frac{X_{ck}(\mathrm{i},\mathrm{j})}{z_{ck}(\mathrm{i},\mathrm{j})} = A_k^{-1} \bar{I}_{pk}(\mathrm{i},\mathrm{j}) \\ \text{for } \mathrm{j} = 1, 2, \cdots, J \end{gathered} \tag{3.14}$$

Then we can obtain $z_{ck}(\mathrm{i},\mathrm{j})$ in the camera coordinates with (3.13). Consequently, we can compute characteristic points as below

$$X_{cLk}(\mathrm{i},\mathrm{j}) = X_{ck}(\mathrm{i},\mathrm{j}) = z_{ck}(\mathrm{i},\mathrm{j}) A_k^{-1} \bar{I}_p(\mathrm{i},\mathrm{j}) \tag{3.15}$$

Similarly, for i-th triangular laser plane $\boldsymbol{\pi}_{\mathrm{i}k} = (a_{\mathrm{i}k} \;\; b_{\mathrm{i}k} \;\; c_{\mathrm{i}k} \;\; -1)^T$ in the k-th camera coordinate, the characteristic points are known

$$\boldsymbol{\pi}_{\mathrm{i}k}^T \bar{X}_{cLk}(\mathrm{i},\mathrm{j}) = 0, \quad \mathrm{j} = 1, 2, \cdots, J \tag{3.16}$$

With M trials for different checkerboard orientations on the floor plane $z = 0$, we derive

$$\boldsymbol{\pi}_{\mathrm{i}k}^T \bar{X}_{cLk}(\mathrm{i},\mathrm{j})_m = 0, \quad \mathrm{j} = 1, 2, \cdots, J, m = 1, 2, \cdots, M \tag{3.17}$$

Finally, we use the least-squares method to solve the i-th triangular plane in the k-th camera coordinate, i.e., $\boldsymbol{\pi}_{\mathrm{i}k} = (a_{\mathrm{i}k} \;\; b_{\mathrm{i}k} \;\; c_{\mathrm{i}k} \;\; -1)^T$.

3.2 Training process of neural networks

Our dataset contains totally 500 images with the reflective light and the scattering light. We randomly split 90% of dataset (450 images) and 10% of dataset (50 images) as the training subset and the test subset respectively.

We use the ReduceLROnPlateau strategy for the learning rate scheduler, i.e. reduce the learning rate $\eta \leftarrow 0.1 \cdot \eta$ by multiplying a factor 0.1 once learning stagnates. For the binary segmentation task, we choose the max mode for the learning rate η, i.e. η will be reduced when the metric, i.e. dice coefficient on the test dataset has stopped increasing for 2 epochs. Moreover, we set the smooth constant (the moving average parameter) $\alpha = 0.99$, the weight decay parameter $\lambda = 10^{-8}$, the momentum factor $\beta = 0.9$ and the initial learning rate $\eta_0 = 10^{-3}$ for the RMSprop optimizer.

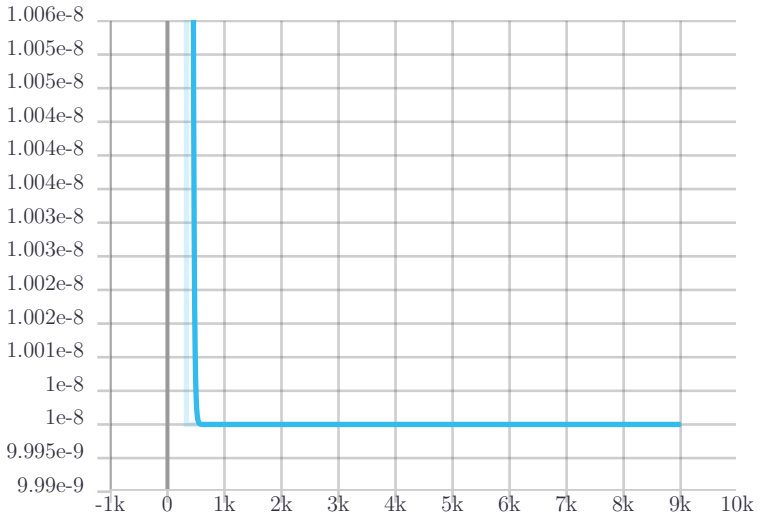


Figure 3.2. The learning rate η during training.

3.3 Post processing of image

3.3.1 Converting RGB images to grayscale images

Firstly, we convert the RGB masked image from the RGB color space to the grayscale color space by

$$I = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{3.18}$$

Where I indicates the luminance in the grascale color space, and R, G, B denote the red, green and blue components in the RGB color space. Besides, we can also use the red component for the red laser, the green laser for the green laser directly to convert the original image to the grascale image I.

3.3.2 Adaptive contrast enhancement

For the next step, we may use the adaptive contrast enhancement method [25] to eliminate the effect of dark environment.

$$\hat{I}_{\mathrm{ij}} = G_{\mathrm{ij}} \left(I_{\mathrm{ij}} - M_{\mathrm{ij}} \right) + M_{\mathrm{ij}} \tag{3.19}$$

where $G_{\mathrm{ij}}, I_{\mathrm{ij}}, M_{\mathrm{ij}}$ is the local gain, the luminance and the local mean for the pixel of the grayscale image at (i, j), and the enhanced luminance at (i, j) denotes $\hat{I}_{\mathrm{ij}}$. In the paper [25], it is set up to be $G_{\mathrm{ij}} = \alpha \frac{M}{\sigma_{\mathrm{ij}}},\ 0 < \alpha < 1$, where M is the global mean value, σ_{ij} is the local standard deviation. The local gain G_{ij} is spatially adaptive: we hope that at the edges of the image or other areas with drastic changes, the G_{ij} is smaller so that no ringing effect is produced; nevertheless, G_{ij} is large in smooth area, which causes the amplification of noise. Thus, the maximum value of G_{ij} should be limited. In our work, we set G_{ij} to be

$$G_{\mathrm{ij}} = \min \left(\frac{\sigma}{\sigma_{\mathrm{ij}}},\ G_{\max} \right) \tag{3.20}$$

where the definitions of $M_{\mathrm{ij}}, \sigma_{\mathrm{ij}}, M, \sigma$ are shown below, note that the image of grayscale I is padded with n pixels before calculating $M_{\mathrm{ij}}, \sigma_{\mathrm{ij}}$. Here we set the window size to be

$(2n+1)\times(2n+1)=17\times17$, the maximal gain to be $G_{\max}=4$, and N, M denote the width and height of images.

$$
\begin{aligned}
M_{\mathrm{ij}} &= \frac{1}{(2n+1)^2}\sum_{\Delta \mathrm{i}=-n}^{n}\sum_{\Delta \mathrm{j}=-n}^{n} I_{\mathrm{i}+\Delta \mathrm{i},\mathrm{j}+\Delta \mathrm{j}} \\
\sigma_{\mathrm{ij}} &= \sqrt{\frac{1}{(2n+1)^2-1}\sum_{\Delta \mathrm{i}=-n}^{n}\sum_{\Delta \mathrm{j}=-n}^{n}\left(I_{\mathrm{i}+\Delta \mathrm{i},\mathrm{j}+\Delta \mathrm{j}}-M_{\mathrm{ij}}\right)^2} \\
M &= \frac{1}{N\cdot M}\sum_{\mathrm{i}=1}^{N}\sum_{\mathrm{j}=1}^{M} I_{\mathrm{ij}} \\
\sigma &= \sqrt{\frac{1}{N\cdot M-1}\sum_{\mathrm{i}=1}^{N}\sum_{\mathrm{j}=1}^{M}\left(I_{\mathrm{ij}}-M\right)^2}
\end{aligned}
\tag{3.21}
$$

3.3.3 Binarization and morphological operation

Afterwards, we binarize the enhanced image $\hat{I}$ to the binary image $\tilde{I}$ by the thresholding method, where the typical value of $\mathrm{th}_{\mathrm{binary}}$ ranges from 0.9 to 0.99.

$$
\tilde{I}=\begin{cases}1 & \text{if } \hat{I}\geq \mathrm{th}_{\mathrm{binary}}\times 255\\ 0 & \text{otherwise}\end{cases}
\tag{3.22}
$$

Then, we do an opening operation on the binarized image $\tilde{I}$ to remove the small objects. Namely, opening operation is equivalent to doing the erosion operation first then doing the dilation operation for the next step.

$$
\tilde{I}\circ S=(\tilde{I}\ominus S)\oplus S
\tag{3.23}
$$

The erosion operation is expected to remove the small object on the foreground, and the dilation operation to expand the shapes contained in the binary image $\tilde{I}$. The erosion and dilation of the binary image $\tilde{I}$ by a structuring element S are defined by

$$
\begin{aligned}
\tilde{I}\ominus S &= \bigcap_{s\in S}\tilde{I}_{-s} \\
\tilde{I}\oplus S &= \bigcup_{s\in S}\tilde{I}_{s}
\end{aligned}
\tag{3.24}
$$

where $\tilde{I}_s, \tilde{I}_{-s}$ are the translation of $\tilde{I}$ by s and $-s$, and the structuring element S is set to be a disk-shaped $(2r-1) \times (2r-1)$ structuring element, where r specifies the radius. In our work, we set $r = 6$ and the disk-shaped structuring element S to be

$$S = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \tag{3.25}$$

3.3.4 Extracting laser stripe centers

Finally, we simply compute the coordinates by averaging the clusters of 1s for every Δv rows. Besides, the gap between two adjacent distinct clusters should be greater than the maximum in-cluster gap, i.e. $\mathrm{gap}_{\max}$. In addition, we can use the Steger's method [26] to extract the sub-pixel coordinates of the laser stripe center for further improvement.

$$\begin{aligned} \mathcal{U}_k &:= \left\{ \mathrm{i} \;\middle|\; \tilde{I}_{\mathrm{i},(k-1)\Delta v+1} = 1 \right\} \quad k = 1, \cdots, \left\lfloor \tfrac{M-1}{\Delta v} \right\rfloor + 1 \\ \mathcal{U}_k &\xRightarrow{\text{split into}} \bigcup_{c=1}^{C_k} \mathcal{U}_k^{(c)} \text{ where for any } \mathrm{i}_1 \in \mathcal{U}_k^{(c_1)}, \mathrm{i}_2 \in \mathcal{U}_k^{(c_2)} \\ &\qquad \begin{cases} |\mathrm{i}_1 - \mathrm{i}_2| \le \mathrm{gap}_{\max} & \text{if } c_1 = c_2 \\ |\mathrm{i}_1 - \mathrm{i}_2| > \mathrm{gap}_{\max} & \text{if } c_1 \ne c_2 \end{cases} \end{aligned} \tag{3.26}$$

where C_k indicates the number of clusters, $\mathcal{U}_k^{(c)}$ denote the subset of the c-th cluster subset, and all the subset are disjoint $\mathcal{U}_k^{(c_1)} \cap \mathcal{U}_k^{(c_2)} = \emptyset$ here $c_1 \ne c_2$

For each set partition $\mathcal{U}_k^{(c)}$ $c = 1, \cdots, C_k$, we average the values in every subset to obtain the x component of the pixel coordinates, and set $(k-1)\Delta v + 1$ to be the y components of the pixel coordinates. To obtain more accurate pixel coordinates of the extracted laser stripe centers, we use the gray weighted method to improve the accuracy. Consequently, we get the set of all the pixel coordinates $\mathcal{U}$

$$\mathcal{U} \coloneqq \bigcup_{k=1}^{\lfloor \frac{M-1}{\Delta v} \rfloor + 1} \bigcup_{c=1}^{C_k} \left\{ \begin{pmatrix} \dfrac{\sum\limits_{\mathrm{i}=\min\left(\mathcal{U}_k^{(c)}\right)}^{\max\left(\mathcal{U}_k^{(c)}\right)} \mathrm{i} \times I_{\mathrm{i},(k-1)\Delta v+1}}{\sum\limits_{\mathrm{i}=\min\left(\mathcal{U}_k^{(c)}\right)}^{\max\left(\mathcal{U}_k^{(c)}\right)} I_{\mathrm{i},(k-1)\Delta v+1}}, & (k-1)\Delta v + 1, & 1 \end{pmatrix}^T \right\} \tag{3.27}$$

3.4 Accuracy evaluation of structured light vision system

First, we calculate the corresponding camera coordinates $(\hat{x}_c \ \hat{y}_c \ \hat{z}_c)^T$ for each pixel coordinate $(u, v, 1)^T \in \mathcal{U}$ using the method described above. Then, we convert the reconstructed camera coordinates $(\hat{x}_c \ \hat{y}_c \ \hat{z}_c)^T$ to the estimated world coordinates $(\hat{x} \ \hat{y} \ \hat{z})^T = R^T \cdot \left((\hat{x}_c \ \hat{y}_c \ \hat{z}_c)^T - t \right)$ with the calibrated extrinsic parameters R, t. We can set all the points to be in the same plane parallel to $z = 0$, i.e. they all have the same z component (negative number of the height) in the real world coordinate system. Thus we can define the metric of error as below

$$E = z - \frac{\sum\limits_{(u \ v \ 1)^T \in \mathcal{U}} \hat{z}(u, v)}{|\mathcal{U}|} \tag{3.28}$$

where $\hat{z}(u, v)$ indicates the reconstructed z component with respect to the pixel coordinates $(u \ v \ 1)^T \in \mathcal{U}$, and $|\mathcal{U}|$ means the cardinality of set $\mathcal{U}$.

In addition, the percentage of error E_p is denoted by

$$E_p = \frac{E}{|z|} \times 100\% \tag{3.29}$$

4. RESULTS

4.1 Experiment platform

The designed measurement system includes, red and green laser emitters [27], 9×10 square checkerboard, a processing platform and two high-resolution cameras (Basler acA2500-14gc GigE camera with ON Semiconductor MT9P031 CMOS sensor, 14 frames per second, 5MP resolution). We used two laser emitters to conduct experiments. The system setup we used for the measurement is shown in Fig. 4.1.

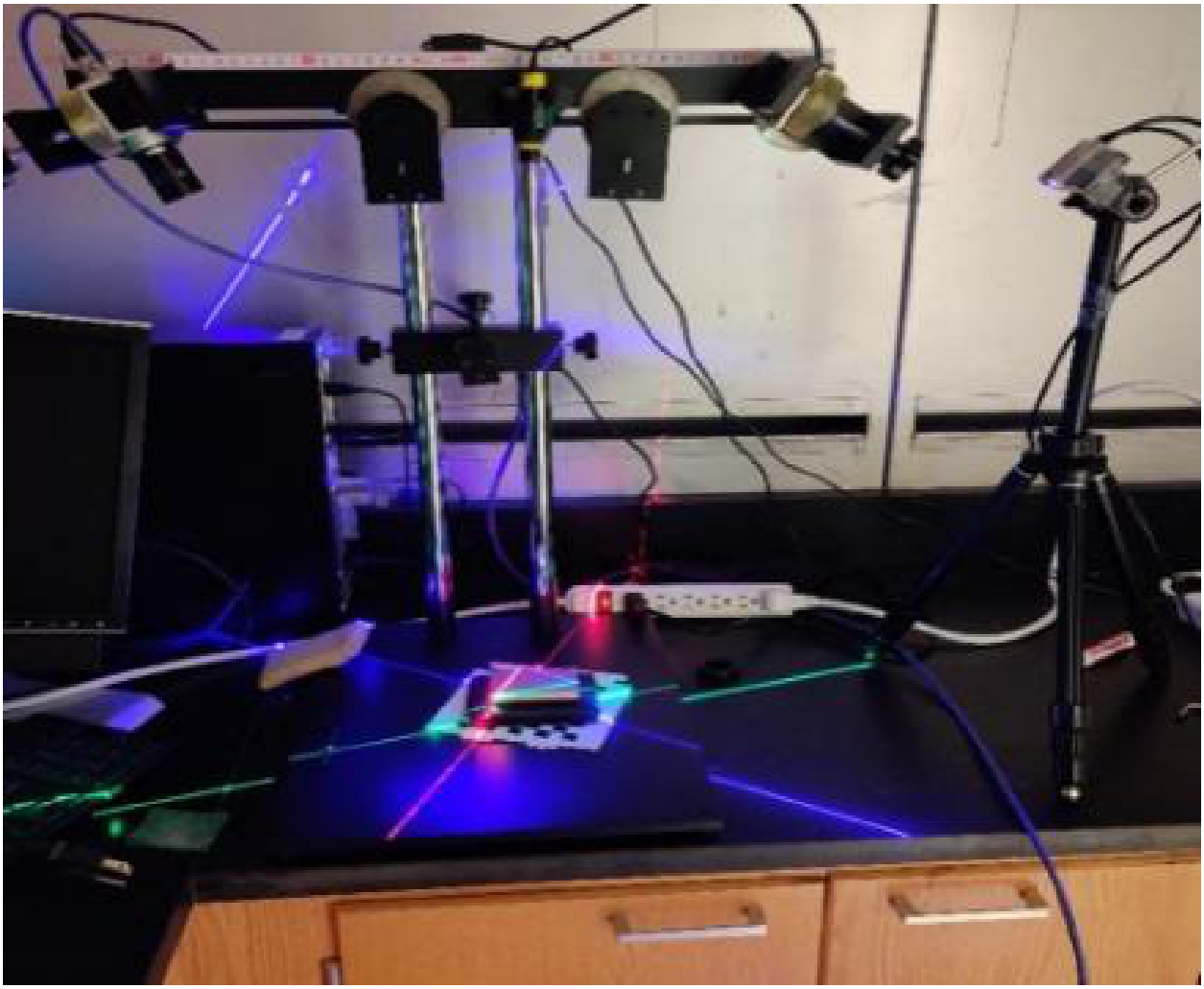


Figure 4.1. Proposed structured-light measurement system.

4.2 Result of metrics for neural networks

We trained the neural networks for 100 epochs, the dice coefficient on the test dataset and the values of loss function with cross entropy on the training dataset are shown below. All the ground truth masks were labeled manually with the MATLAB Image Labeler toolbox.

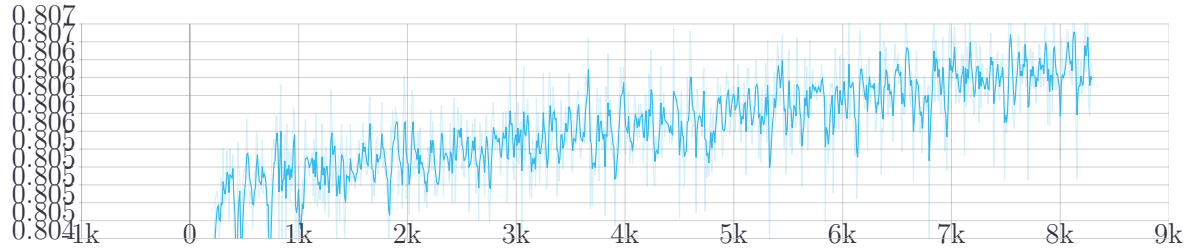


Figure 4.2. The dice coefficient on test dataset during training.

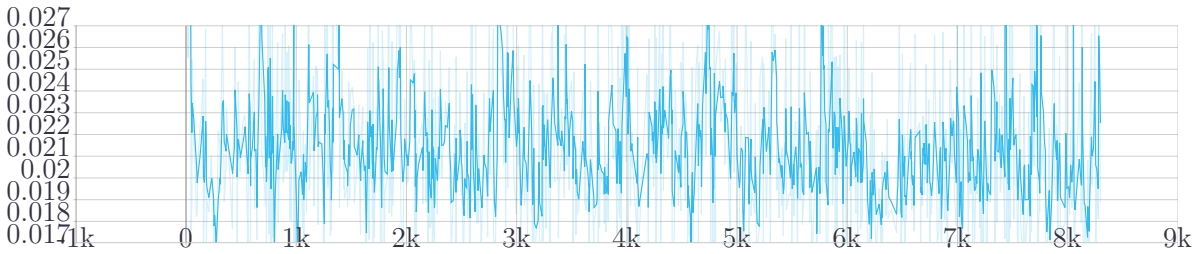


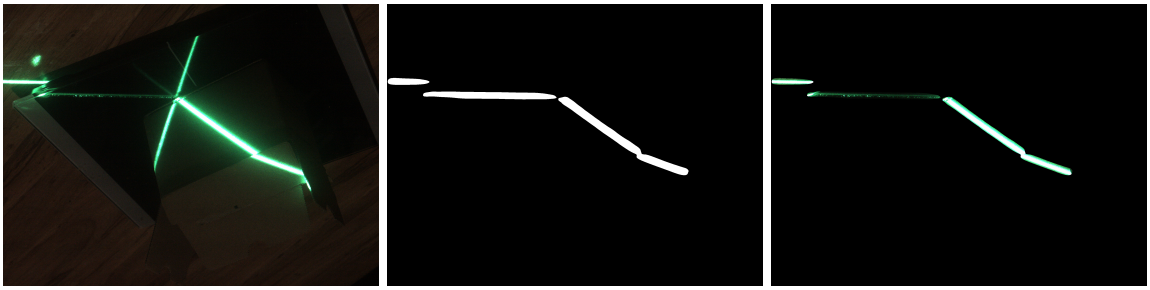
Figure 4.3. The values of loss function during training.

After training, we evaluate the performance of the trained neural networks with metrics. It shows that the dice coefficient for the overall dataset is 0.8108, Intersection over Union (IoU) is 0.6900 in Table 4.1.

Table 4.1. The metrics for neural metworks.

metric	dice coefficient	IoU
	0.8108	0.6900

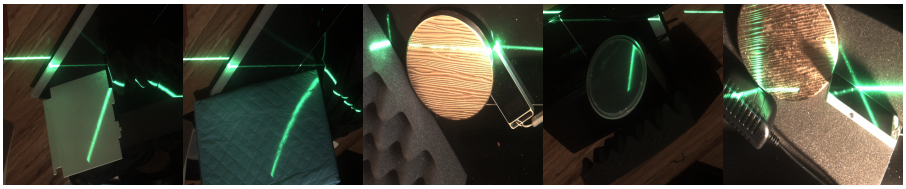
Thus, the neural networks would be able to provide us a mask for each input image, that only keeps laser stripes and removes the reflective noise in the original input image. We only retain the areas in the input image where the generated mask is enabled, the other areas are set to be 0.



(a) The input image (b) The generated mask (c) The masked image

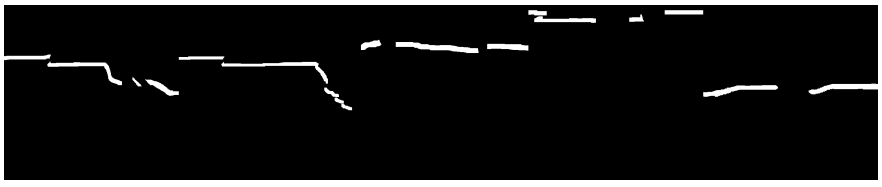
Figure 4.4. A example image, its corresponding generated mask and the masked image.

The input image, the predicted mask by the neural networks and the ground truth mask are demonstrated as follows. Fig. 4.5a shows 5 original input images, Fig. 4.5b displays the corresponding masks generated the neural networks, and Fig. 4.5c is the concatenated image with 5 ground truth mask.



(a) The original input images used for training neural networks

(b) The generated predicted masks while training neural networks



(c) The ground truth masks used for training neural networks

Figure 4.5. A input image, its predicted mask and the ground truth mask.

4.3 Result of measurement evaluation

Before combining the neural network part for segmentation and the structured light system part for reconstruction, we have to check the accuracy of the structured light system by comparing the actual heights of objects and the calculated heights with the structured light system.

4.3.1 System calibration results

With MATLAB Camera Calibration toolbox, we calibrated both the intrinsic extrinsic parameters $[R \;\; t]$ and the intrinsic parameter A for each cameras based on Zhang's methodology [24]. The parameters for the first camera are shown in (4.1).

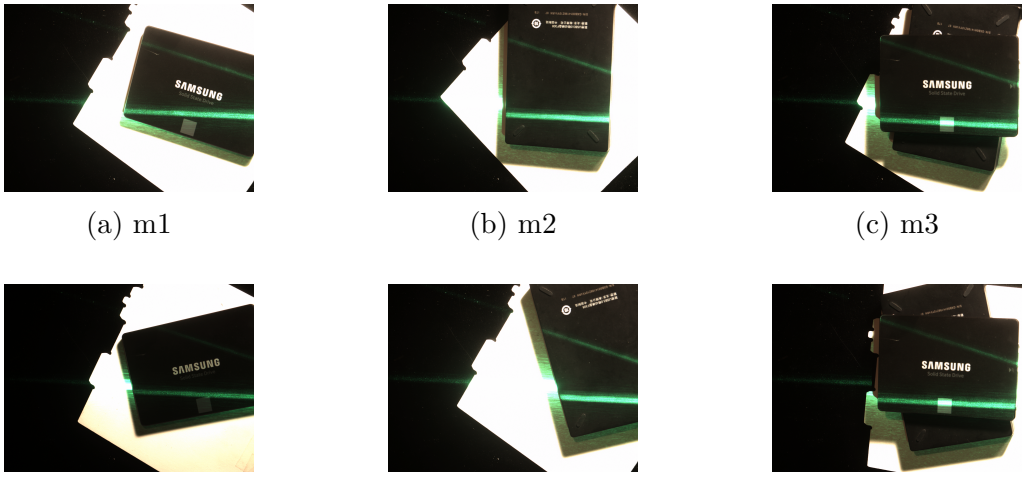
$$\begin{aligned} A &= \begin{pmatrix} 7.643543 \times 10^3 & 0 & 1.067068 \times 10^3 \\ 0 & 7.663082 \times 10^3 & 1.032315 \times 10^3 \\ 0 & 0 & 1 \end{pmatrix} \\ k_1 &= -0.231085, k_2 = 0.526775 \\ R &= \begin{pmatrix} -0.047294 & -0.934260 & 0.353440 \\ 0.991818 & -0.085926 & -0.094416 \\ 0.118579 & 0.346082 & 0.930680 \end{pmatrix} \\ t &= \begin{pmatrix} 0.652110 \times 10^2 \\ -0.529570 \times 10^2 \\ 5.255506 \times 10^2 \end{pmatrix} \end{aligned} \tag{4.1}$$

Besides, the plane equation of the laser plane in the left camera coordinates is

$$\boldsymbol{\pi}_1 = \begin{pmatrix} 0.536388 \times 10^{-3} \\ 3.200885 \times 10^{-3} \\ 1.803748 \times 10^{-3} \end{pmatrix} \tag{4.2}$$

4.3.2 Segmentation results

Fig. 4.6 displays the objects labeled from m1 to m6 whose height measurements were performed using our structured light vision system.

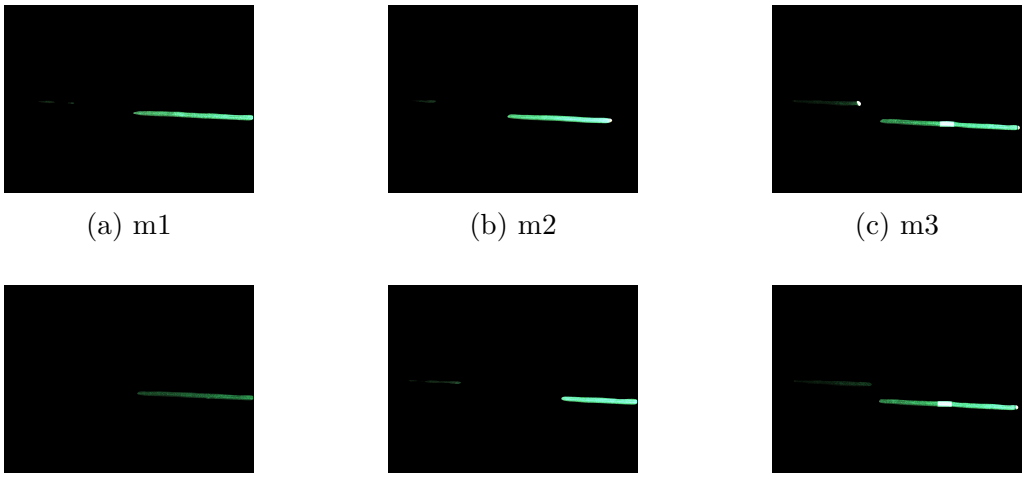


(a) m1 (b) m2 (c) m3

(d) m4 (e) m5 (f) m6

Figure 4.6. Measurements for m1 to m6.

Fig. 4.7 shows the segmented masks for m1 to m6 by the trained U-Net neural networks. The neural network removes the reflective noise in the background and keeps the laser stripe.

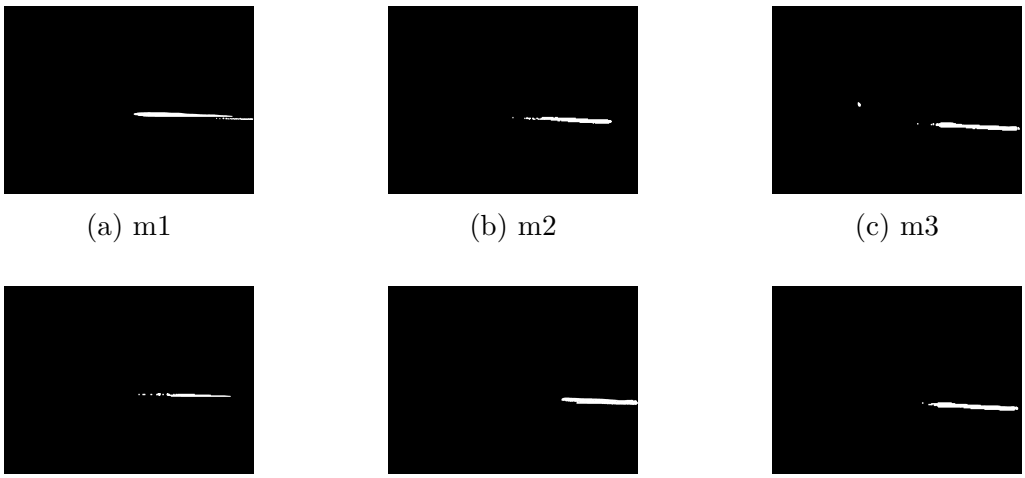


(a) m1 (b) m2 (c) m3

(d) m4 (e) m5 (f) m6

Figure 4.7. Masks for m1 to m6.

Fig. 4.8 shows the extracted laser stripe for m1 to m6 after the post processing operations. These operations includes adaptive contrast enhancement, binarization and morphological opening operation.



(a) m1 (b) m2 (c) m3

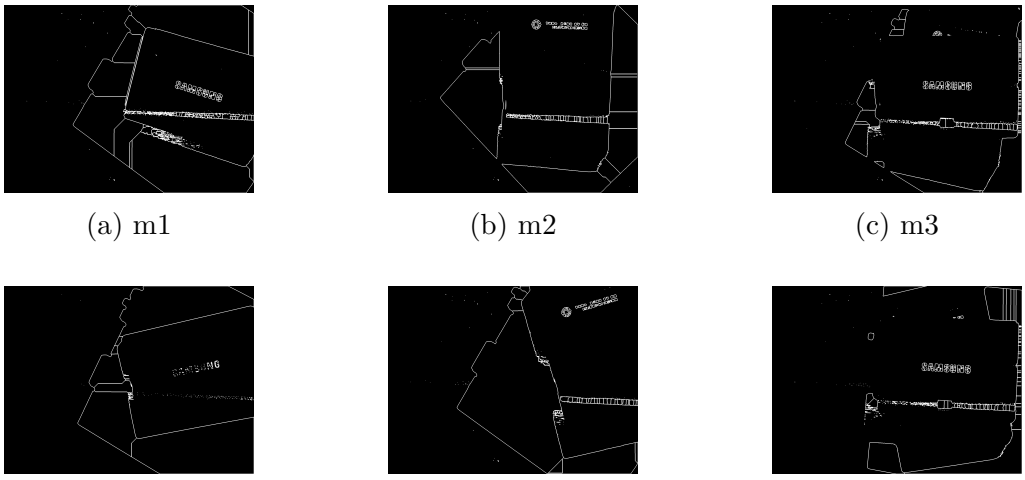
(d) m4 (e) m5 (f) m6

Figure 4.8. Extracted laser stripes for m1 to m6.

4.3.3 Comparison with segmentation results using Watershed

For comparison, we use the marker-controlled Watershed with distance transform [28] to find out the boundary of different parts in the image. It is clearly that the laser stripes must be included in the boundaries (see Fig. 4.9). With morphological operations, we can remove the clear boundary between the darker parts and lighter parts, and keep the laser stripes. Fig. 4.10 demonstrates the Watershed images with morphological operations.

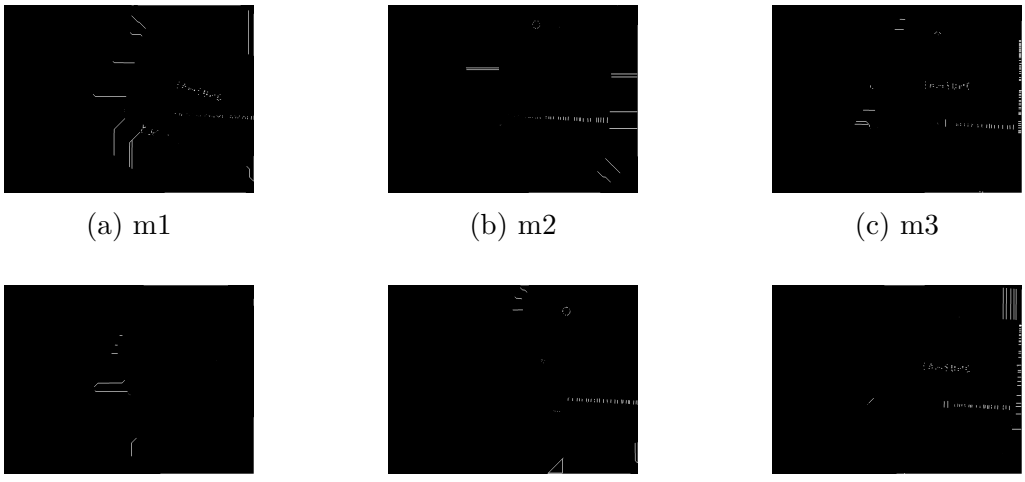
As we can see, the complex environments have a great impact on segmentation results of the conventional Watershed method with morphological operations. On the contrary, the neural networks with U-Net architecture can provide the robust and accurate results for the segmentation task. Hence, we choose the method with U-Net instead of the conventional methods like Watershed and morphological operations as our backbone of segmentation procedure.



(a) m1 (b) m2 (c) m3

(d) m4 (e) m5 (f) m6

Figure 4.9. Segmentation with Watershed for m1 to m6.



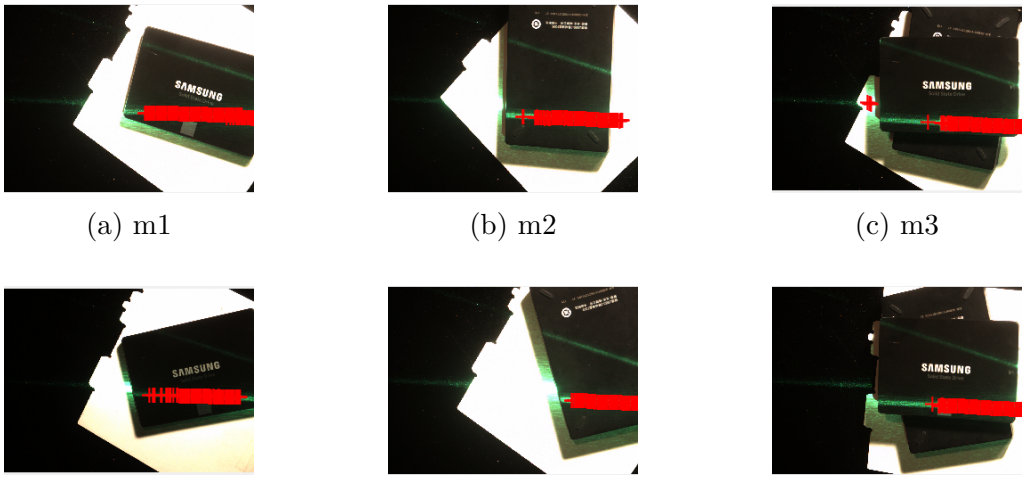
(a) m1 (b) m2 (c) m3

(d) m4 (e) m5 (f) m6

Figure 4.10. Watershed images with morphological operations for m1 to m6.

4.3.4 Height measurement results

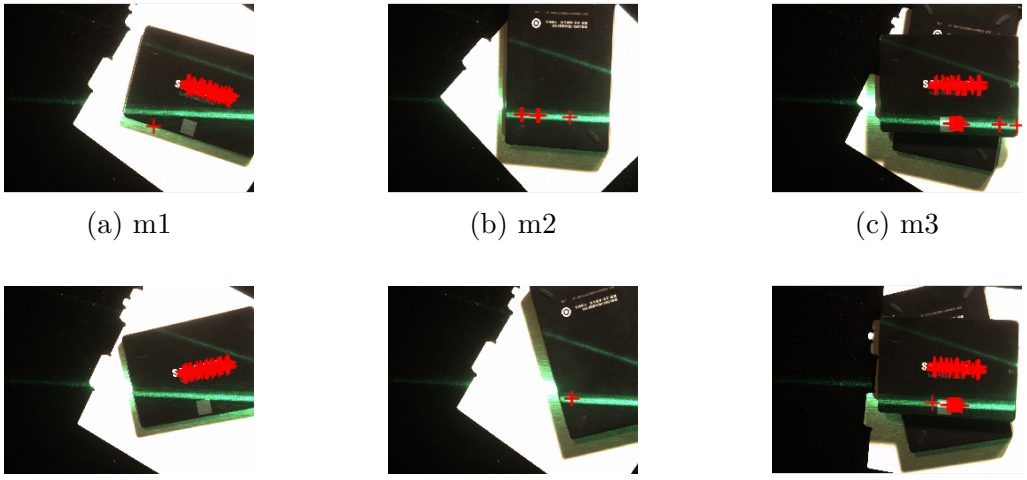
We can compare the extracted laser stripe centers with U-Net and that without U-Net as follows. For instance, the white word in the pictures and the scattering light in Fig. 4.12 affect the performance of the structured light system dramatically. It usually cannot extract sufficient correct points, sometimes even cannot extract any laser stripe centers in the complex environment.



(a) m1 (b) m2 (c) m3

(d) m4 (e) m5 (f) m6

Figure 4.11. Extracted laser strip centers with U-Net method for m1 to m6.

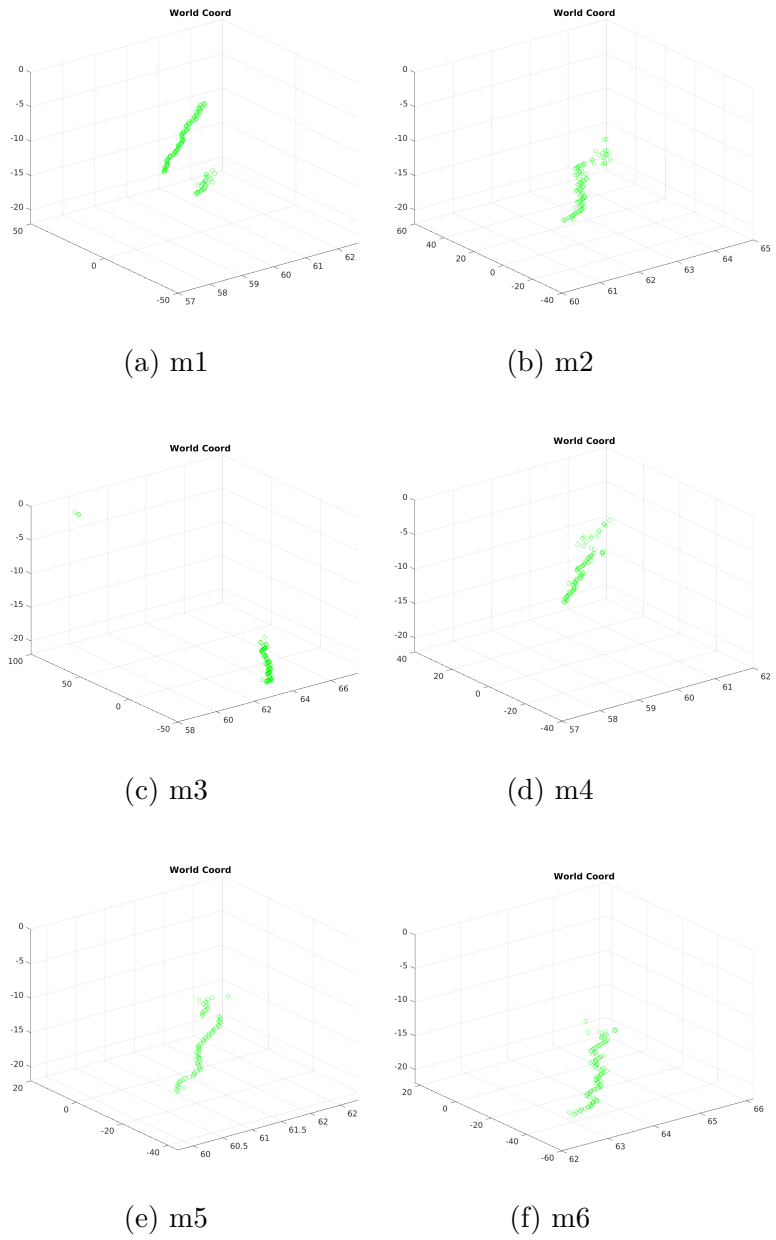


(a) m1 (b) m2 (c) m3

(d) m4 (e) m5 (f) m6

Figure 4.12. Extracted laser strip centers without U-Net method for m1 to m6.

The reconstructed cloud points for m1 to m6 are shown below in Fig. 4.13 and Fig. 4.14.



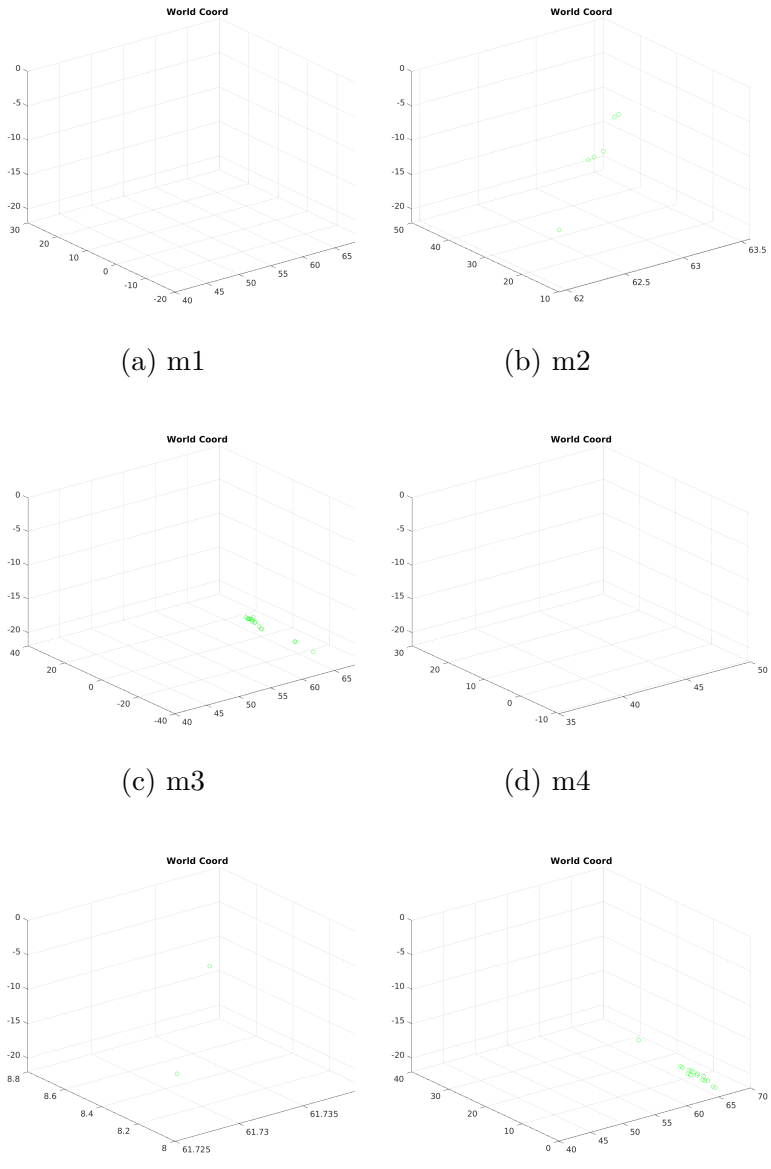
(a) m1 (b) m2

(c) m3 (d) m4

(e) m5 (f) m6

Figure 4.13. Extracted laser strip centers with U-Net method for m1 to m6.

We can notice that the cloud points of our proposed method are consistently and densely distributed. However, the cloud point distribution of the method without U-Net is sparse and incoherent.



(a) m1 (b) m2

(c) m3 (d) m4

(e) m5 (f) m6

Figure 4.14. Extracted laser strip centers without U-Net method for m1 to m6.

The height measurement results with our proposed method are listed in Table 4.2, while Table 4.3 demonstrates the results without U-Net segmentation. The first column in Table 4.2 and Table 4.3 indicate the actual heights via physical measurements; meanwhile, the heights measured with the structured light system are shown in the second column of Table 4.2 and Table 4.3. The results in Table 4.3 are the average height of reconstructed cloud points fter filtering with a threshold of 5 mm.

For m1, the method without U-Net has a large error due to the misidentification of scattered light. Besides, it cannot provide a predicted height of m4 due to its weak ability to identify laser stripes in complex environments. In the meantime, our proposed method can always provide robust height measurements with a range of error 1-2 mm in complex environments.

Table 4.2. Table of measurement error for the structured light system with U-Net

Measurement Number	Actual height / mm	Measure height / mm	Absolute error / mm	Relative error %
1	6.700	8.864	2.164	32.29%
2	12.800	14.182	1.382	10.8%
3	19.500	20.713	1.213	6.22%
4	6.700	8.530	1.830	27.31%
5	12.800	13.967	1.167	9.11%
6	19.500	20.732	1.232	6.32%

Table 4.3. Table of measurement error for the structured light system without U-Net

Measurement Number	Actual height / mm	Measure height / mm	Absolute error / mm	Relative error %
1	6.700	25.675	18.975	283.20%
2	12.800	12.913	0.113	0.89%
3	19.500	20.232	0.732	3.75%
4	6.700	NA	NA	NA
5	12.800	12.627	-0.173	-1.35%
6	19.500	20.639	1.139	5.84%

In our conference paper [29], the height measurement results of two cameras are listed in Table 4.4, while Table 4.5 includes the worst results from the single camera. The first column in Table 4.4 and Table 4.5 indicate the actual heights via physical measurements; meanwhile,

the heights measured with the structured light system are shown in the second column of Table 4.4 and Table 4.5. The structured light system has 3.65% obtained from averaging the relative errors from the six objects. It is clear that the performance of a multi-camera system exceeds that of a system equipped with a single camera.

Table 4.4. Table of measurement error for the structured light system with two cameras

Measurement Number	Actual height / mm	Measure height / mm	Absolute error / mm	Relative error %
1	6.700	6.839	0.139	2.08 %
2	12.800	12.386	0.414	-3.23 %
3	19.500	18.165	1.335	-6.85 %
4	6.700	6.942	0.242	3.61 %
5	12.800	12.373	0.427	-3.34 %
6	19.500	18.488	1.012	-5.19 %

Table 4.5. Table of measurement error for the structured light system with a single camera

Measurement Number	Actual height / mm	Measure height / mm	Absolute error / mm	Relative error %
1	6.700	6.787	0.087	1.30 %
2	12.800	11.851	0.949	-7.41 %
3	19.500	17.524	1.976	-10.13 %
4	6.700	7.027	0.327	4.87 %
5	12.800	11.849	0.951	-7.43 %
6	19.500	17.666	1.834	-9.40 %

5. CONCLUSION

In this paper, we have derived a framework for 3D reconstruction and object height measurement using multiple cameras and multiple laser emitters. We have developed a U-Net based approach to tackle the problem cased by the refection and scattering of light in complex environment. Our experiments demonstrate that the system with multiple cameras and U-Net laser stripe extraction method improves the accuracy of height measurement over the single camera and strengthen the stability of system. For the future work, we can collect more images from different perspectives with reflected light and scattering light. Thus, we are able to further improve the accuracy of our model for segmentation task with sufficient information. Besides, we may replace the U-Net architecture with the UNet++ [30] architecture which has more skip pathways to reduce the semantic disparity between the encoder feature maps and that of the decoder.

REFERENCES

[1] G. Godin, F. Blais, L. Cournoyer, J. A. Beraldin, J. Domey, J. Taylor, M. Rioux, and S. El-Hakim, "Laser range imaging in archaeology: Issues and results," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, vol. 1, 2003, pp. 11–16, ISBN: 0769519008. DOI: 10.1109/CVPRW.2003.10002.

[2] H. Ha, T.-H. Oh, and I. S. Kweon, "A multi-view structured-light system for highly accurate 3D modeling," in *2015 International Conference on 3D Vision*, IEEE, 2015, pp. 118–126.

[3] C. Ho, "Machine vision based 3D scanning system," in *2009 9th International Conference on Electronic Measurement & Instruments*, IEEE, 2009, pp. 4–445.

[4] J. Li, G. Liu, and Y. Liu, "A dynamic volume measurement system with structured light vision," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, IEEE, 2016, pp. 251–255.

[5] M. Garrido, M. Perez-Ruiz, C. Valero, C. J. Gliever, B. D. Hanson, and D. C. Slaughter, "Active optical sensors for tree stem detection and classification in nurseries," *Sensors (Switzerland)*, vol. 14, no. 6, pp. 10 783–10 803, 2014, ISSN: 14248220. DOI: 10.3390/s140610783.

[6] D. Li, H. Zhang, Z. Song, D. Man, and M. W. Jones, "An automatic laser scanning system for accurate 3D reconstruction of indoor scenes," in *2017 IEEE International Conference on Information and Automation (ICIA)*, IEEE, 2017, pp. 826–831.

[7] J. Deng, B. Chen, X. Cao, B. Yao, Z. Zhao, and J. Yu, "3D reconstruction of rotating objects based on line structured-light scanning," in *2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, IEEE, 2018, pp. 244–247.

[8] Y. Zheng, J. Li, and L. Wu, "Tree Radial Growth Measurement System Based on Line Structured Light Vision," in *2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, IEEE, 2019, pp. 338–342.

[9] C. Roman, G. Inglis, and J. Rutter, "Application of structured light imaging for high resolution mapping of underwater archaeological sites," in *OCEANS'10 IEEE SYDNEY*, 2010, pp. 1–9. DOI: 10.1109/OCEANSSYD.2010.5603672.

[10] F. Mokhayeri, E. Granger, and G.-A. Bilodeau, "Domain-specific face synthesis for video face recognition from a single sample per person," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 757–772, 2018.

[11] B. Ayhan and C. Kwan, "Mastcam Image Resolution Enhancement with Application to Disparity Map Generation for Stereo Images with Different Resolutions," *Sensors*, vol. 19, no. 16, p. 3526, Aug. 2019, ISSN: 1424-8220. DOI: 10.3390/s19163526. [Online]. Available: https://www.mdpi.com/1424-8220/19/16/3526.

[12] "Enhancing Stereo Image Formation and Depth Map Estimation for Mastcam Images," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2018*, 2018, pp. 566–572, ISBN: 9781538676936. DOI: 10.1109/UEMCON.2018.8796542.

[13] C. Kwan, B. Chou, and B. Ayhan, "Stereo Image and Depth Map Generation for Images with Different Views and Resolutions," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2018*, 2018, ISBN: 9781538676936. DOI: 10.1109/UEMCON.2018.8796832.

[14] T. T. Nguyen, D. C. Slaughter, N. Max, J. N. Maloof, and N. Sinha, "Structured light-based 3d reconstruction system for plants," *Sensors*, vol. 15, no. 8, pp. 18 587–18 612, 2015, ISSN: 1424-8220. DOI: 10.3390/s150818587. [Online]. Available: https://www.mdpi.com/1424-8220/15/8/18587.

[15] W. Xiuping, F. Ying, L. Ziteng, and B. Ruilin, "Detecting and reconstructing curve welding seam using structured light stereovision," in *2015 Chinese Automation Congress (CAC)*, 2015, pp. 381–384. DOI: 10.1109/CAC.2015.7382529.

[16] Z. Wang, "An imaging and measurement system for robust reconstruction of weld pool during arc welding," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, pp. 5109–5118, 2015. DOI: 10.1109/TIE.2015.2405494.

[17] Y. Zou and R. Lan, "An End-to-End Calibration Method for Welding Robot Laser Vision Systems with Deep Reinforcement Learning," *IEEE Transactions on Instrumentation and Measurement*, 2019.

[18] J. Geng, "Structured-light 3d surface imaging: A tutorial," *Adv. Opt. Photon.*, vol. 3, no. 2, pp. 128–160, Jun. 2011. DOI: 10.1364/AOP.3.000128. [Online]. Available: http://aop.osa.org/abstract.cfm?URI=aop-3-2-128.

[19] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, IEEE, vol. 1, 2003, pp. I–I.

[20] M. Vo, S. G. Narasimhan, and Y. Sheikh, "Texture illumination separation for single-shot structured light reconstruction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 390–404, 2015.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[22] C. Wang, Y. Li, Z. Ma, J. Zeng, T. Jin, and H. Liu, "Distortion rectifying for dynamically measuring rail profile based on self-calibration of multiline structured light," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 3, pp. 678–689, 2018.

[23] Y. Zhang, Z. Luo, J. Hou, L. Tan, and X. Guo, "Computer vision techniques for improving structured light vision systems," in *2020 IEEE International Conference on Electro Information Technology (EIT)*, 2020, pp. 437–442. DOI: 10.1109/EIT48999.2020.9208332.

[24] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proceedings of the seventh ieee international conference on computer vision*, Ieee, vol. 1, 1999, pp. 666–673.

[25] P. M. Narendra and R. C. Fitch, "Real-time adaptive contrast enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-3, no. 6, pp. 655–661, 1981. DOI: 10.1109/TPAMI.1981.4767166.

[26] C. Steger, "An unbiased detector of curvilinear structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 113–125, 1998. DOI: 10.1109/34.659930.

[27] Z. Feng, D. Man, and Z. Song, "A Pattern and Calibration Method for Single-Pattern Structured Light System," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 3037–3048, 2019.

[28] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.

[29] Z. Luo, Y. Zhang, and L. Tan, "Multi-level random sample consensus method for improving structured light vision systems," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 2020, pp. 0577–0582. DOI: 10.1109/UEMCON51285.2020.9298161.

[30] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*, Springer, 2018, pp. 3–11.

VITA

Zhankun Luo received the Bachelor of Engineering (B.E.) degree in telecommunication engineering from Beijing Institute of Technology in 2019. He is a master student with the Department of Electrical and Computer Engineering at Purdue University Northwest. He is currently a Research Assistant with Center for Innovation through Visualization and Simulation (CIVS), Hammond, IN. His research interests include computer vision, image processing, and deep learning. He would continue to pursue his Ph.D. degree at Purdue University, West Lafayette.

PUBLICATIONS

[1] Z. Luo, Y. Zhang, and L. Tan, "Multi-level random sample consensus method for improving structured light vision systems," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 2020, pp. 0577–0582.

[2] Y. Zhang, Z. Luo, J. Hou, L. Tan, and X. Guo, "Computer vision techniques for improving structured light vision systems," in *2020 IEEE International Conference on Electro Information Technology (EIT)*, 2020, pp. 437–442.