RGBD Cameras’ Per-Pixel Calibration and Natural 3D Reconstruction on GPU

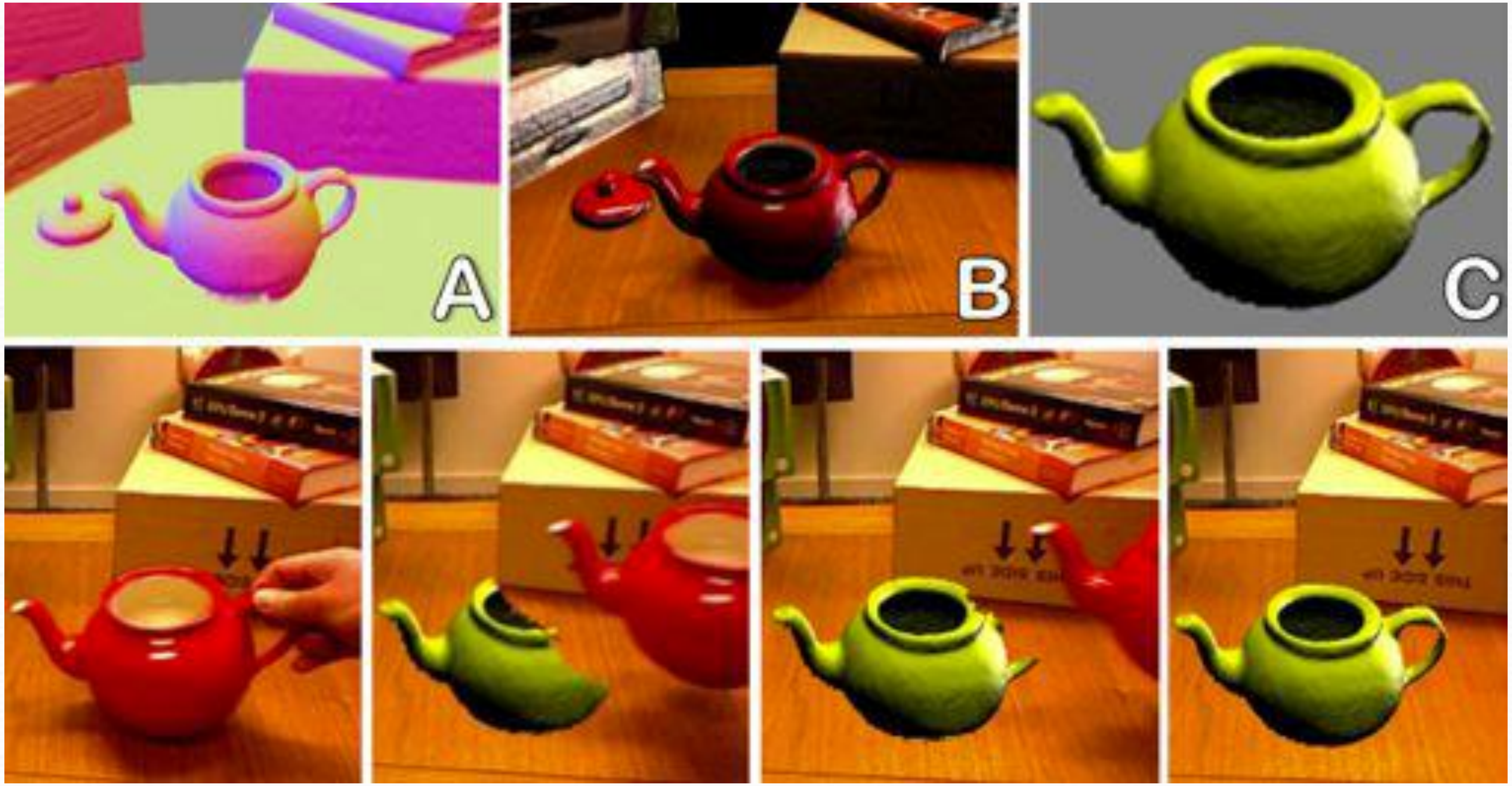
Dr. Lau, advisor
Dr. Hastings, committee member
Dr. Cheung, committee member

- Presented By: Sen Li

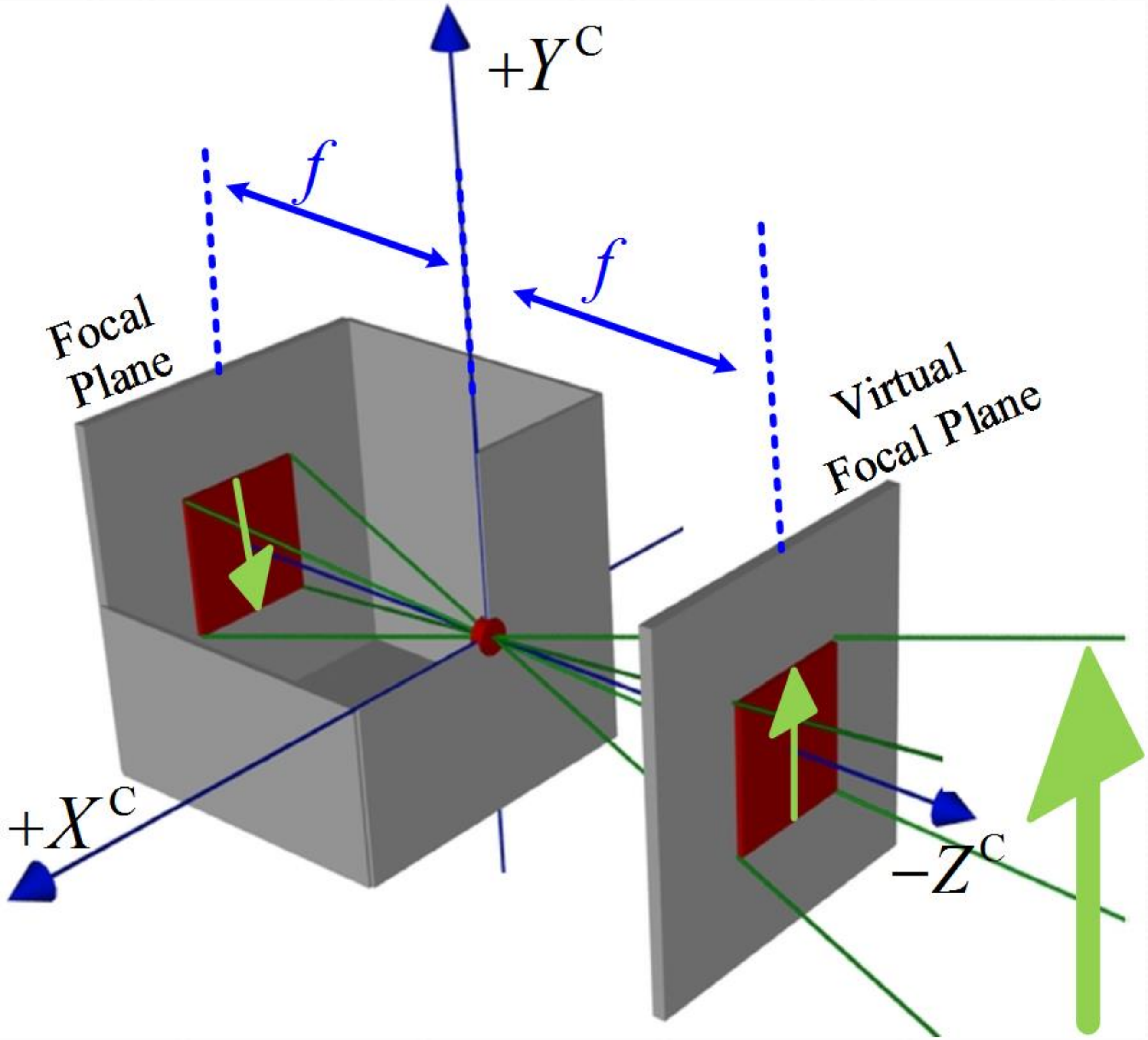
Outline

- Introduction
- Academic Background:
  - –Pinhole Camera Model (Camera calibration / 3D reconstruction)
  - –Kai’s per-pixel 3D reconstruction on GPU
  - –Inspiration
- Natural GPU Calibration and Reconstruction Method:
  - –Calibration system
  - –Calibration procedures
  - –Undistorted 3D Reconstruction
- Conclusion

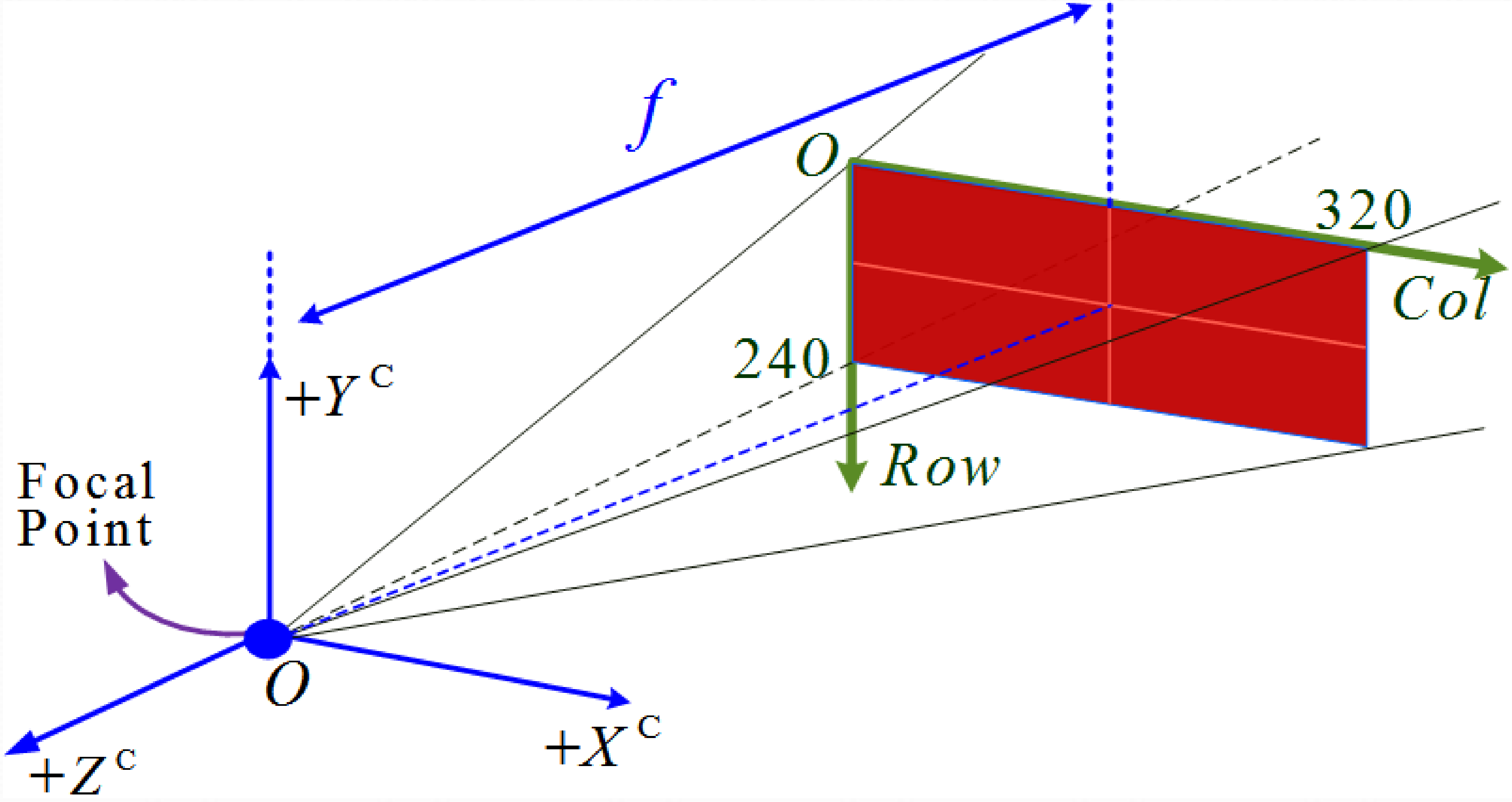
Object Segmentation in KinectFusion



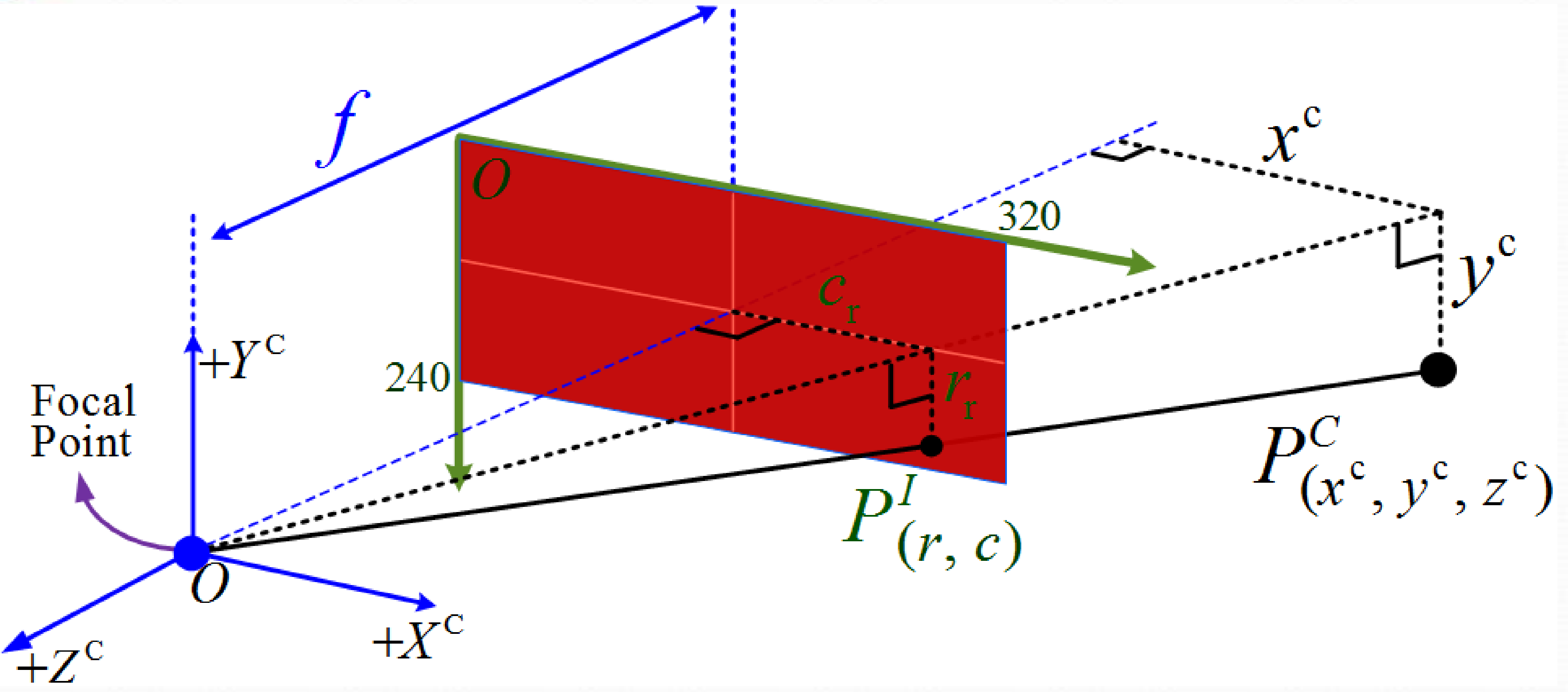
Pinhole-Camera Model



Pinhole-Camera Model



Pinhole-Camera Model



$$\begin{bmatrix} \alpha_c f & 0 & \alpha_c c_h \\ 0 & \alpha_r f & \alpha_r r_h \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{K} = \begin{bmatrix} f_c & s & t_c \\ 0 & f_r & t_r \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} c \\ r \end{bmatrix} = f \begin{bmatrix} x^c/z^c \\ y^c/z^c \end{bmatrix} + \begin{bmatrix} c_h \\ r_h \end{bmatrix}$$

$$z^c \begin{bmatrix} c \\ r \\ 1 \end{bmatrix} = \begin{bmatrix} f x^c \\ f y^c \\ z^c \end{bmatrix} + \begin{bmatrix} z^c c_h \\ z^c r_h \\ 0 \end{bmatrix} = \begin{bmatrix} f & 0 & c_h \\ 0 & f & r_h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix}$$

- Traditional Reconstruction (Calibrate $\boldsymbol{M}$ and then Inverse)

$$M = K\begin{bmatrix} R_{3*3} & T_{3*1} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}$$

$$k\begin{bmatrix} R \\ C \\ 1 \end{bmatrix} = \boldsymbol{M}_{3*4}\begin{bmatrix} X^{\mathrm{W}} \\ Y^{\mathrm{W}} \\ Z^{\mathrm{W}} \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} X^{\mathrm{W}} \\ Y^{\mathrm{W}} \\ Z^{\mathrm{W}} \\ 1 \end{bmatrix} = [\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M}]^{-1}\boldsymbol{M}^{\mathrm{T}}\begin{bmatrix} kR \\ kC \\ k \end{bmatrix}$$

- Kai's : Natural Shader Reconstruction on GPU

Pinhole Camera Model $M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}$ $\longrightarrow$

$$X^W[m,n] = a[m,n]Z^W[m,n] + b[m,n]$$

$$Y^W[m,n] = c[m,n]Z^W[m,n] + d[m,n]$$

- KinectV2: Raw 3D Reconstruction on GPU based on $\theta_{\mathrm{h}} / \theta_{\mathrm{v}}$ (Intrinsic $\boldsymbol{K}$)

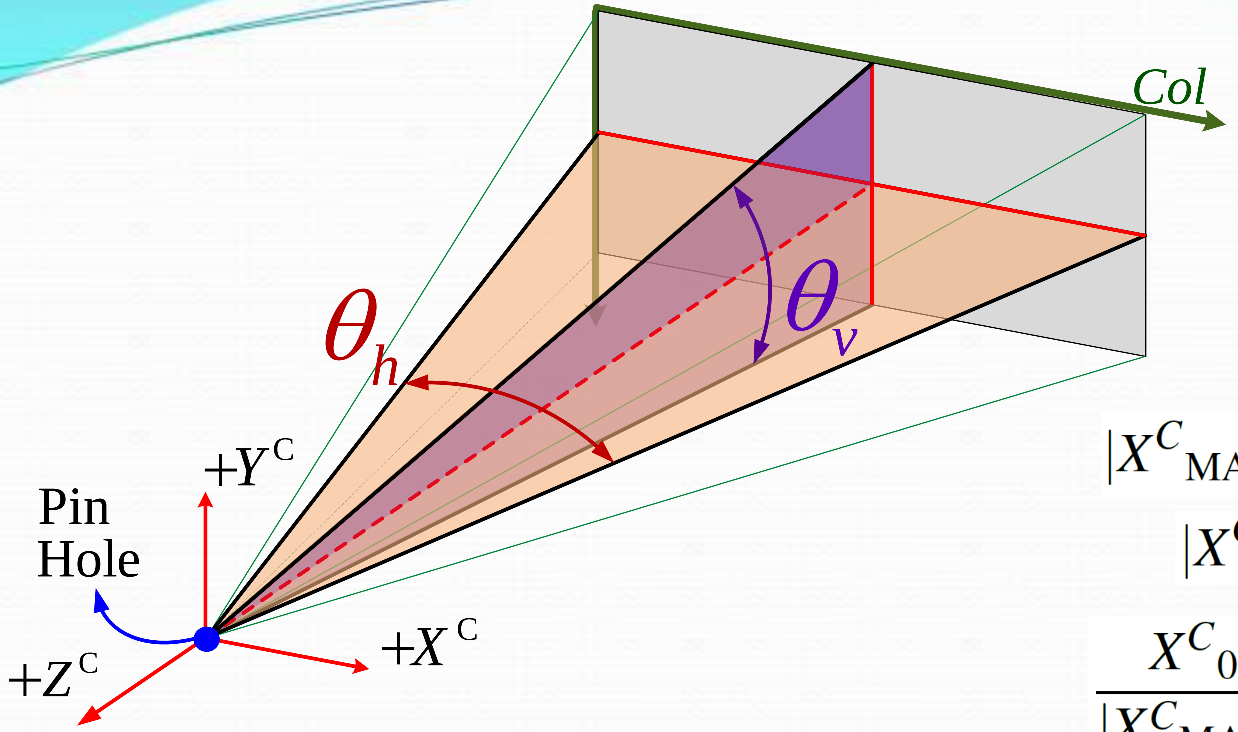
$$Z^C[m,n] = -\beta D[m,n] \qquad \beta = 65535.0$$

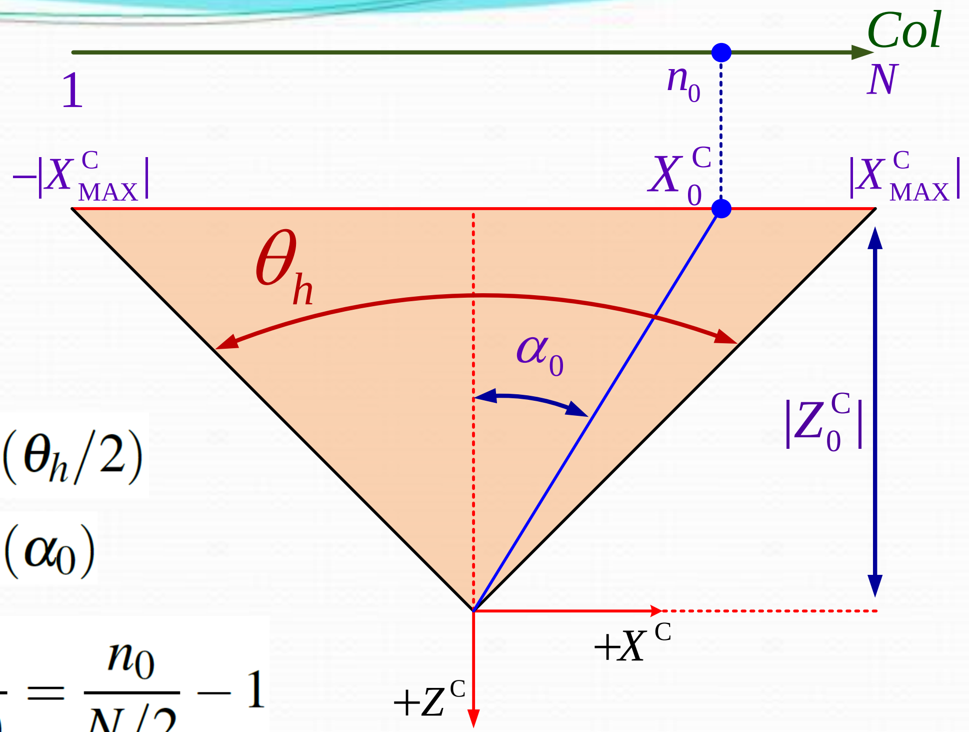
θ_{h}, θ_{v} $\longrightarrow$

$$X^C[m,n] = a[m,n] \cdot |Z^C[m,n]|$$

$$Y^C[m,n] = b[m,n] \cdot |Z^C[m,n]|$$

- Raw 3D Reconstruction on GPU based on θ_h / θ_v (Intrinsic $\boldsymbol{K}$)





$$|X^C{}_{\mathrm{MAX}}| = |Z^C{}_0| \cdot tan(\theta_h/2)$$

$$|X^C{}_0| = |Z^C{}_0| \cdot tan(\alpha_0)$$

$$\frac{X^C{}_0}{|X^C{}_{\mathrm{MAX}}|} = \frac{tan(\alpha_0)}{tan(\theta_h/2)} = \frac{n_0}{N/2} - 1$$

$$X^C[m,n] = tan(\theta_h/2) \cdot (\frac{n}{N/2} - 1) \cdot |Z^C[m,n]|$$

$$Y^C[m,n] = tan(\theta_v/2) \cdot (\frac{m}{M/2} - 1) \cdot |Z^C[m,n]|$$

$$Z^C[m,n] = -\beta D[m,n]$$

65535.0

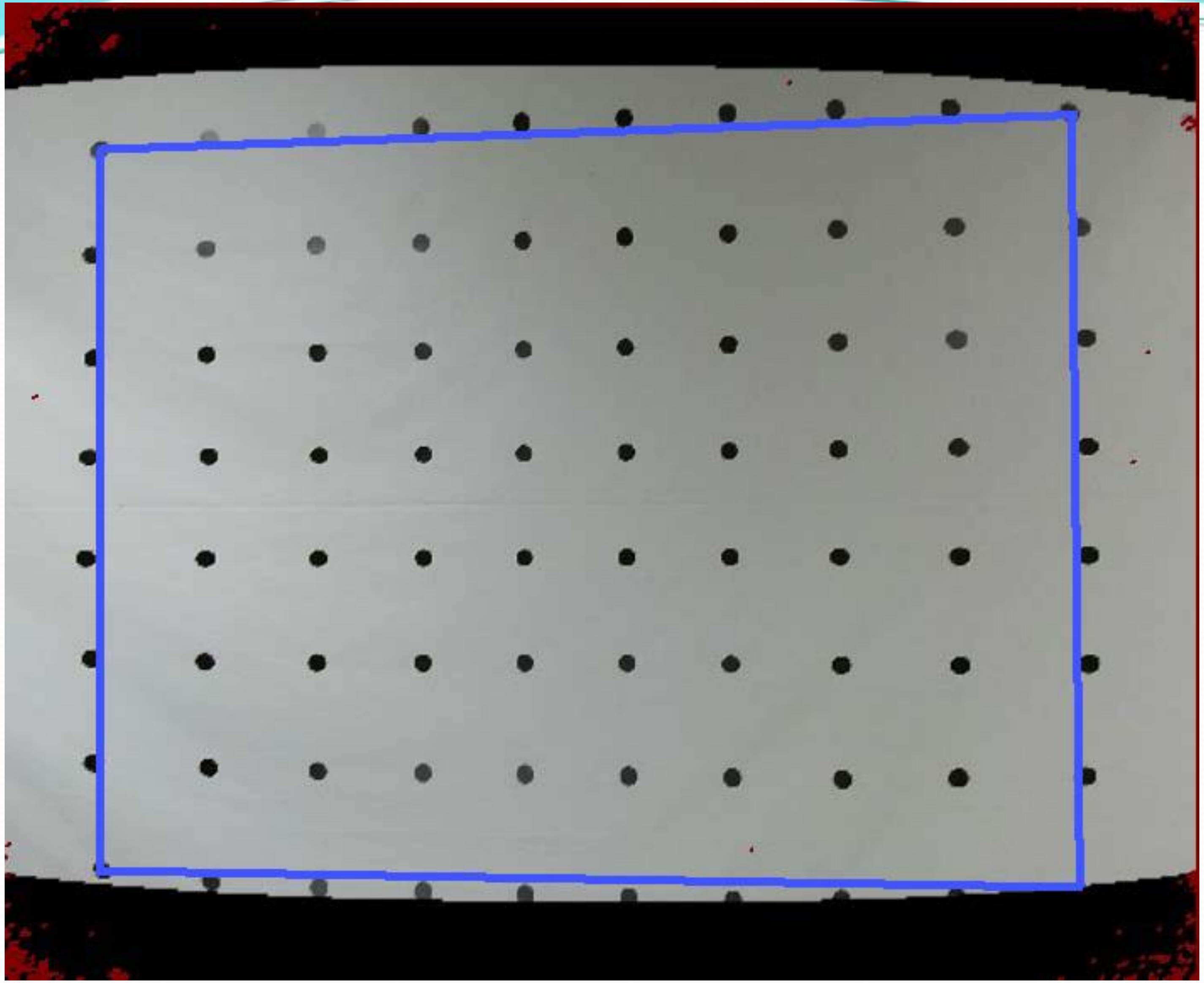
$$a[m,n] = tan(\theta_h/2) \cdot (\frac{n}{N/2} - 1)$$

$$b[m,n] = tan(\theta_v/2) \cdot (\frac{m}{M/2} - 1)$$

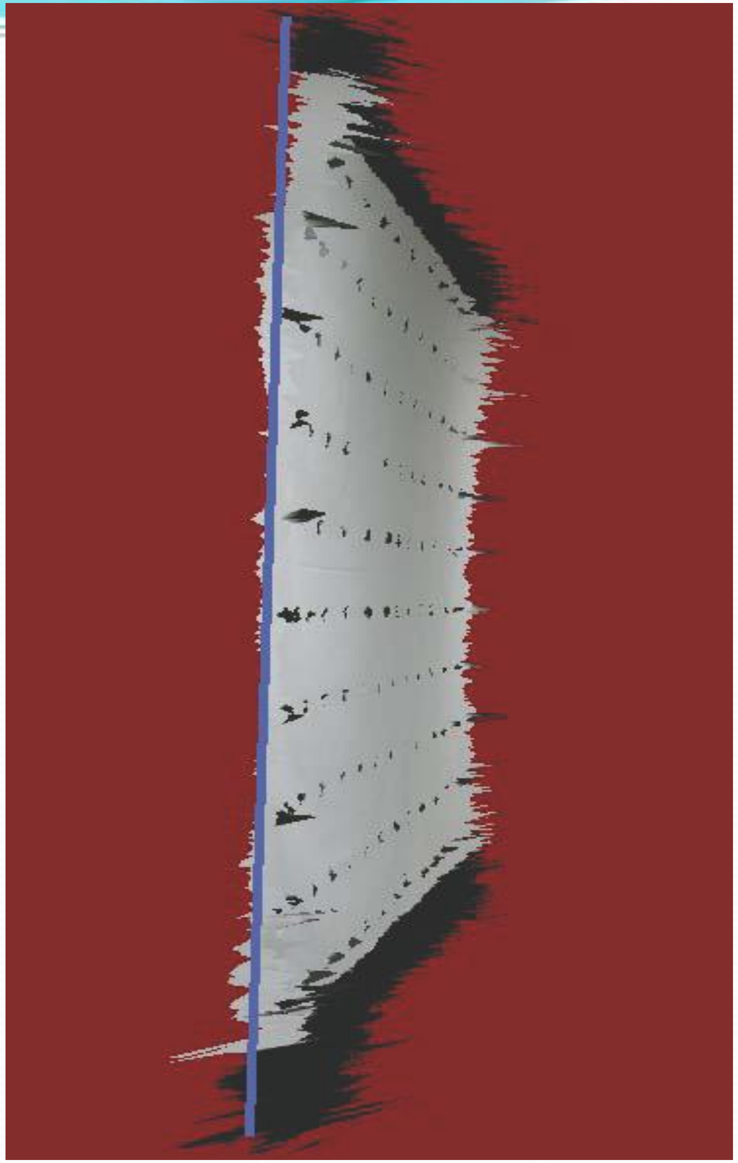
$$X^C[m,n] = a[m,n] \cdot |Z^C[m,n]|$$

$$Y^C[m,n] = b[m,n] \cdot |Z^C[m,n]|$$

Raw KinectV2 3D Reconstruction

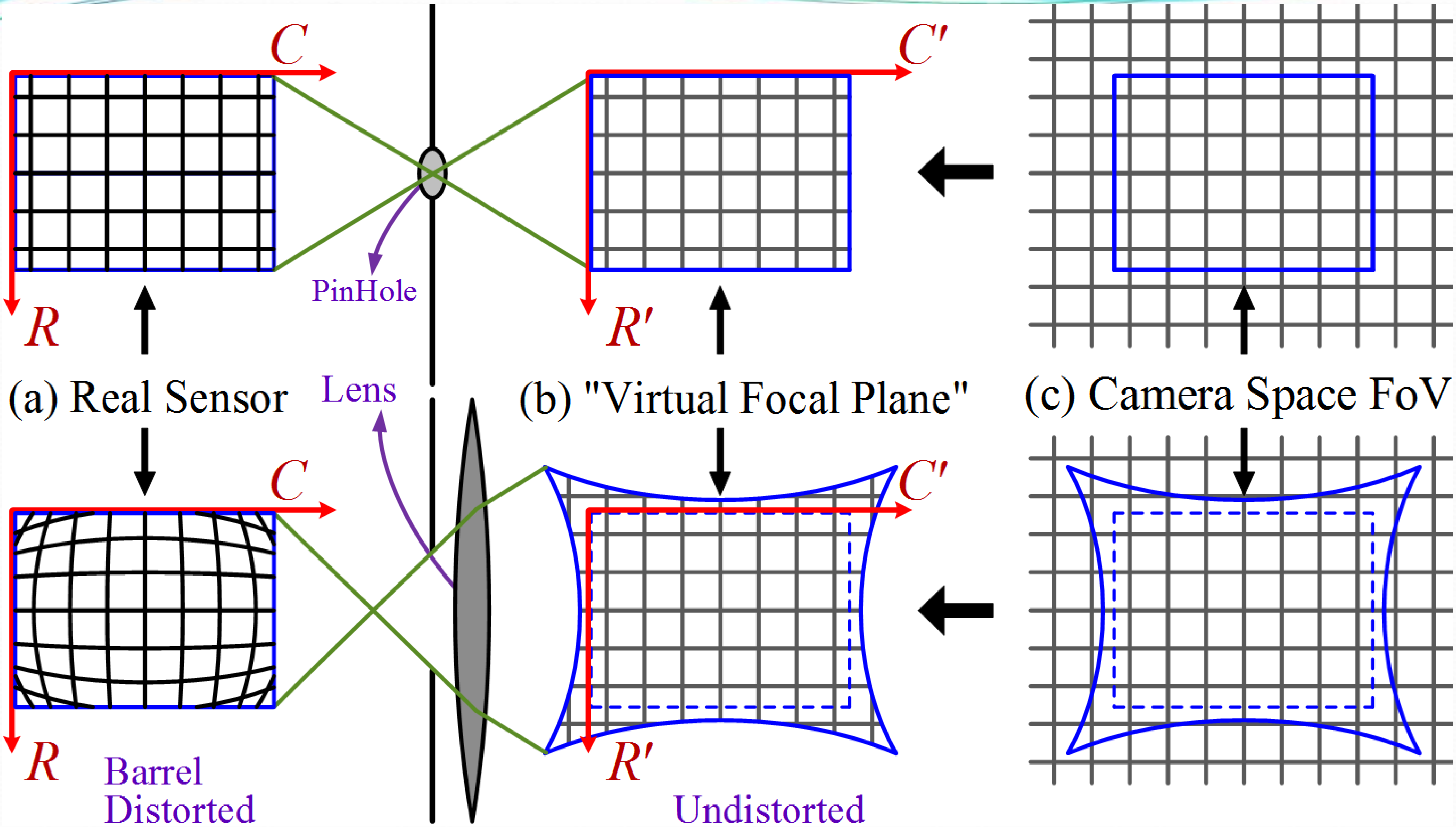


lens Distortion



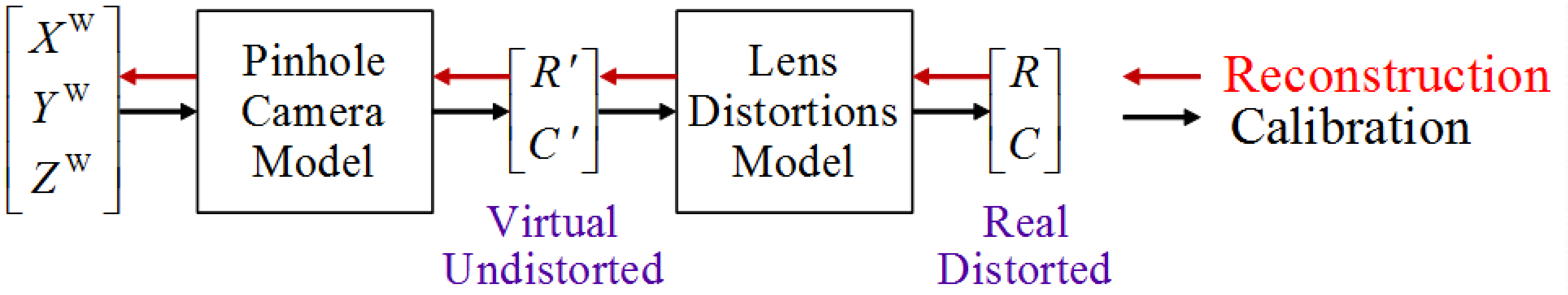
Depth Distortion

- From Camera Space to Image Space with Lens Distortions



Academic Background

- Pinhole-Camera Model (Calibration and Reconstruction)



Intrinsic Matrix $K = \begin{bmatrix} f_c & s & t_c \\ 0 & f_r & t_r \\ 0 & 0 & 1 \end{bmatrix}$

$$C' = C(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + [p_1(r^2 + 2C^2) + 2p_2 CR]$$
$$R' = R(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + [p_2(r^2 + 2R^2) + 2p_1 CR]$$

Extrinsic Matrix $\begin{bmatrix} R_{3*3} & T_{3*1} \end{bmatrix}$

Distortion Parameters: $k_1/k_2/k_3/p_1/p_2$

Pinhole Camera Matrix :

$$M = K \begin{bmatrix} R_{3*3} & T_{3*1} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}$$

$$k \begin{bmatrix} R' \\ C' \\ 1 \end{bmatrix} = \boldsymbol{M}_{3*4} \begin{bmatrix} X^W \\ Y^W \\ Z^W \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} X^W \\ Y^W \\ Z^W \\ 1 \end{bmatrix} = [\boldsymbol{M}^T \boldsymbol{M}]^{-1} \boldsymbol{M}^T \begin{bmatrix} kR' \\ kC' \\ k \end{bmatrix}$$

- Traditional Undistorted Reconstruction (High Order and Inverse)

Distortion Parameters: $k_1/k_2/k_3/p_1/p_2$

$$C' = C(1+k_1r^2+k_2r^4+k_3r^6)+[p_1(r^2+2C^2)+2p_2CR]$$
$$R' = R(1+k_1r^2+k_2r^4+k_3r^6)+[p_2(r^2+2R^2)+2p_1CR]$$

$$k\begin{bmatrix} R' \\ C' \\ 1 \end{bmatrix} = \boldsymbol{M}_{3*4}\begin{bmatrix} X^W \\ Y^W \\ Z^W \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} X^W \\ Y^W \\ Z^W \\ 1 \end{bmatrix} = [\boldsymbol{M}^T\boldsymbol{M}]^{-1}\boldsymbol{M}^T\begin{bmatrix} kR' \\ kC' \\ k \end{bmatrix}$$

- Kai's : "Natural" Shader Reconstruction ($k_1/k_2/k_3/p_1/p_2$ High Order)

$$m' = m*(1+k_1r^2+k_2r^4+k_3r^6)+[p_1(r^2+2m^2)+2p_2*m*n]$$
$$n' = n*(1+k_1r^2+k_2r^4+k_3r^6)+[p_1(r^2+2n^2)+2p_2*m*n]$$

$$X^W[m,n] = a[m',n']Z^W[m,n]+b[m',n']$$
$$Y^W[m,n] = c[m',n']Z^W[m,n]+d[m',n']$$

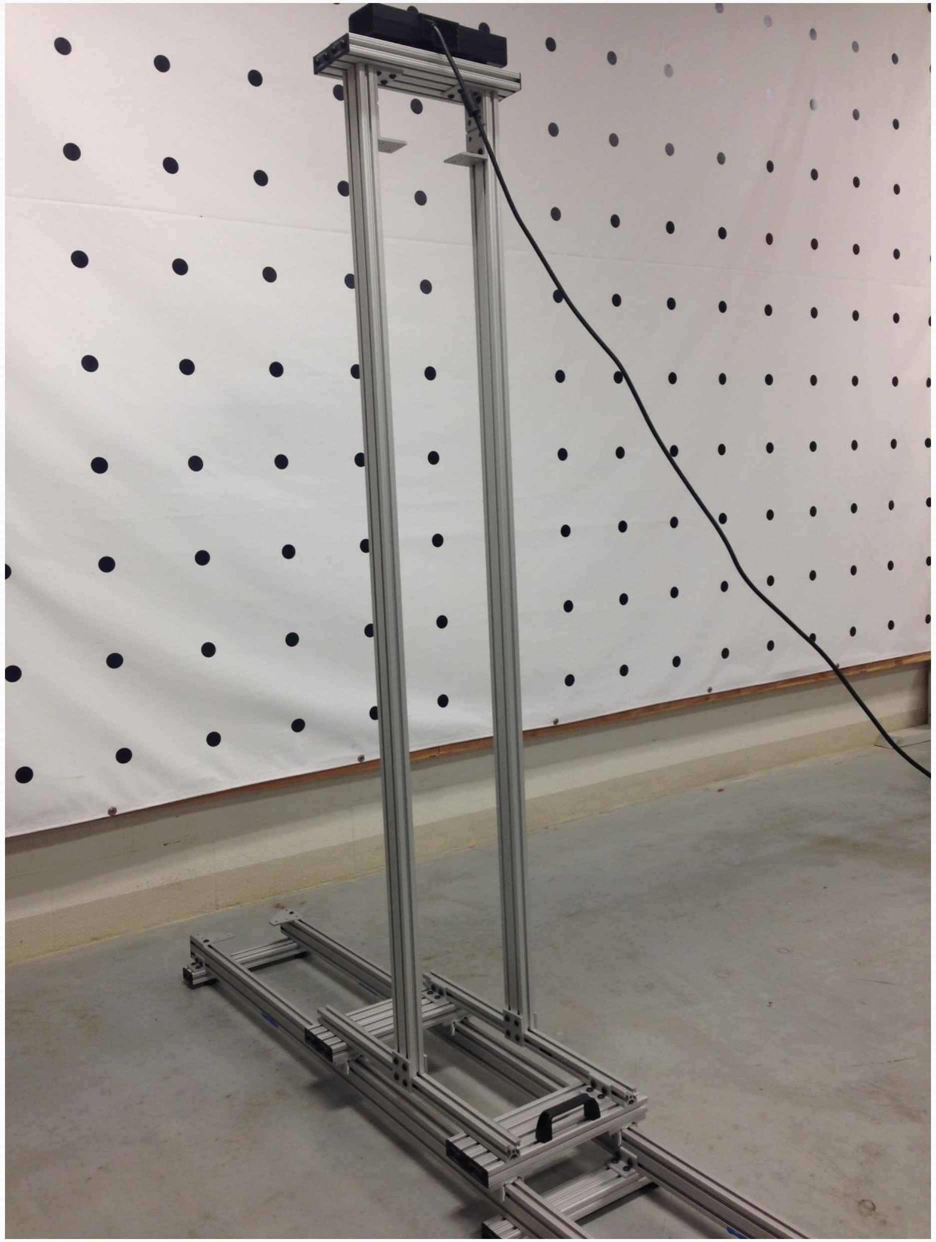
- Want: Undistorted Natural 3D Reconstruction on GPU

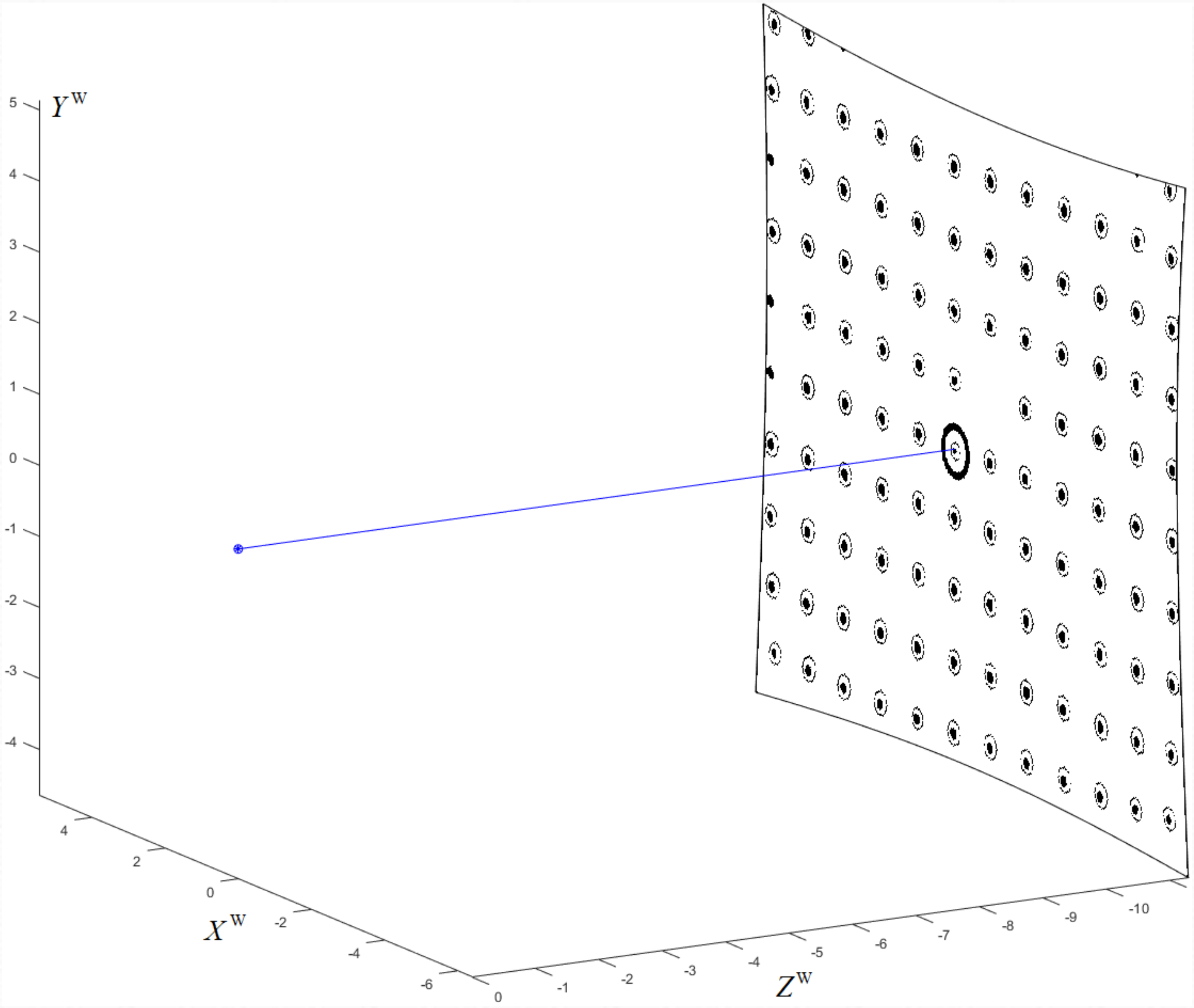
$$Z^C[m,n] = -\beta D[m,n]$$
$$\Big\downarrow \ 65535.0$$
$$Z^W[m,n] = e[m,n]D[m,n]+f[m,n]$$

$$X^W[m,n] = a[m,n]Z^W[m,n]+b[m,n]$$
$$Y^W[m,n] = c[m,n]Z^W[m,n]+d[m,n]$$

- Inspiration: find a way to get per-pixel Z^W

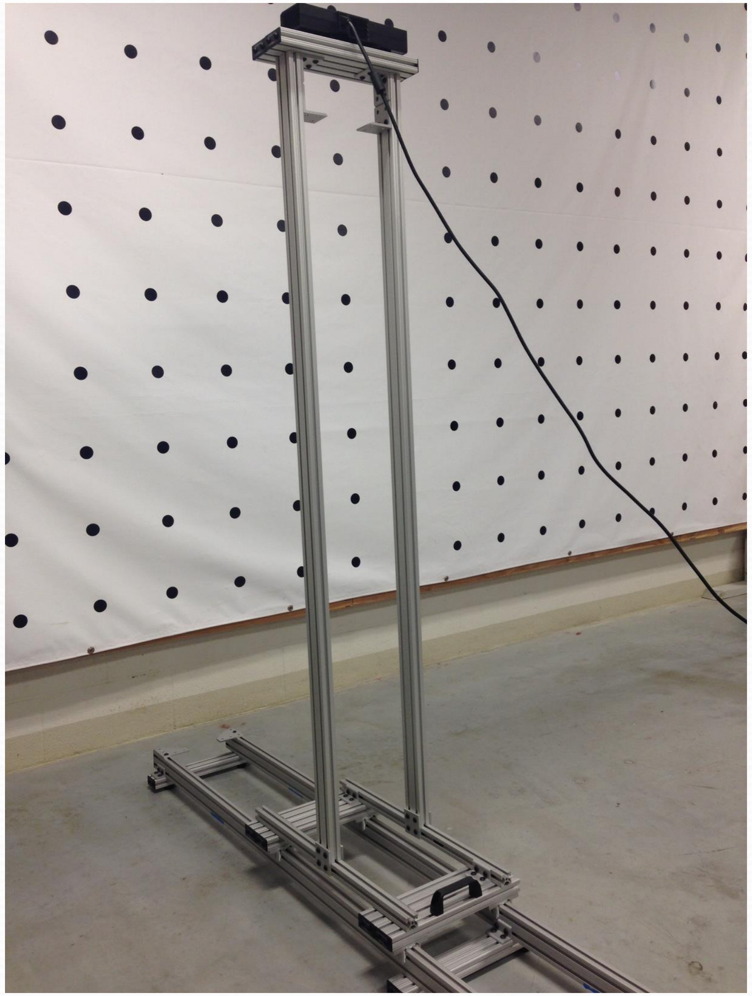
Rail Calibration System





Data Collection

- Mount: camera and laser distance measurer
- Want:
  - NIR: $X^W Y^W Z^W ID$
  - RGB: $X^W Y^W Z^W RGBD$
- Bound:
  - Z^W: Laser Distance Measurer
  - $X^W Y^W$: Uniform Round Dot Pattern
- Found (algorithms):
  - Calibration Points (*Row*, *Col*)s Extraction;
  - Corresponding World Space Address Assignment;
  - Non-Linear Dense Transformation



Calibration Points (*Row, Col*) Extraction;

- Gray-Scaling

$$I_0[m,n] = 0.21R[m,n] + 0.72G[m,n] + 0.07B[m,n]$$

- Histogram Equalization

$$I_1[m,n] = \frac{I_0[m,n] - I_{\mathrm{MIN}}}{I_{\mathrm{MAX}} - I_{\mathrm{MIN}}}$$

- Adaptive Thresholding

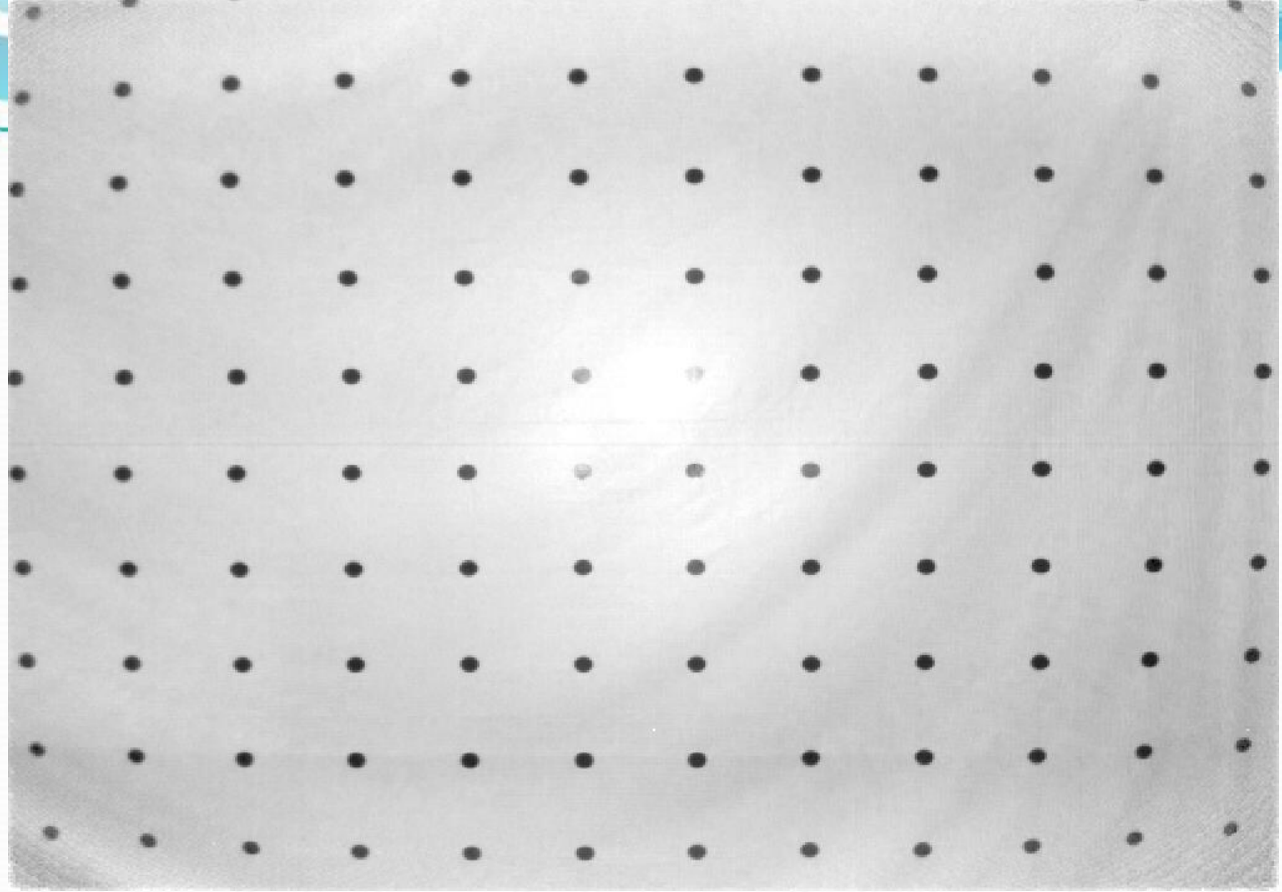
$$I_2[m,n] = \begin{cases} 1, & I_1[m,n] - I_{\mathrm{b}}[m,n] - C_0 > 0 \\ 0, & \mathit{else} \end{cases}$$

- Round Dot Tracking

E	*A*	*F*
B	*O*	*C*
G	*D*	*H*

$$I_3[m,n] = I_2[m,n] * \frac{(128I_{\mathrm{A}} + 64I_{\mathrm{B}} + 32I_{\mathrm{C}} + 16I_{\mathrm{D}} + 8I_{\mathrm{E}} + 4I_{\mathrm{F}} + 2I_{\mathrm{G}} + I_{\mathrm{H}})}{255}$$

Raw NIR Image



After Historgram Equalization

$$I_0[m,n] = \frac{g[m,n]}{G} * (1.0 - 0.0) + 0.0 = \frac{g[m,n]}{G}$$

$$CDF(g) = \frac{\sum_{l=1}^{g} PMF[l]}{M \times N}$$

$$CDF(g_{\min}) = 0.01$$

$$CDF(g_{\max}) = 0.99$$

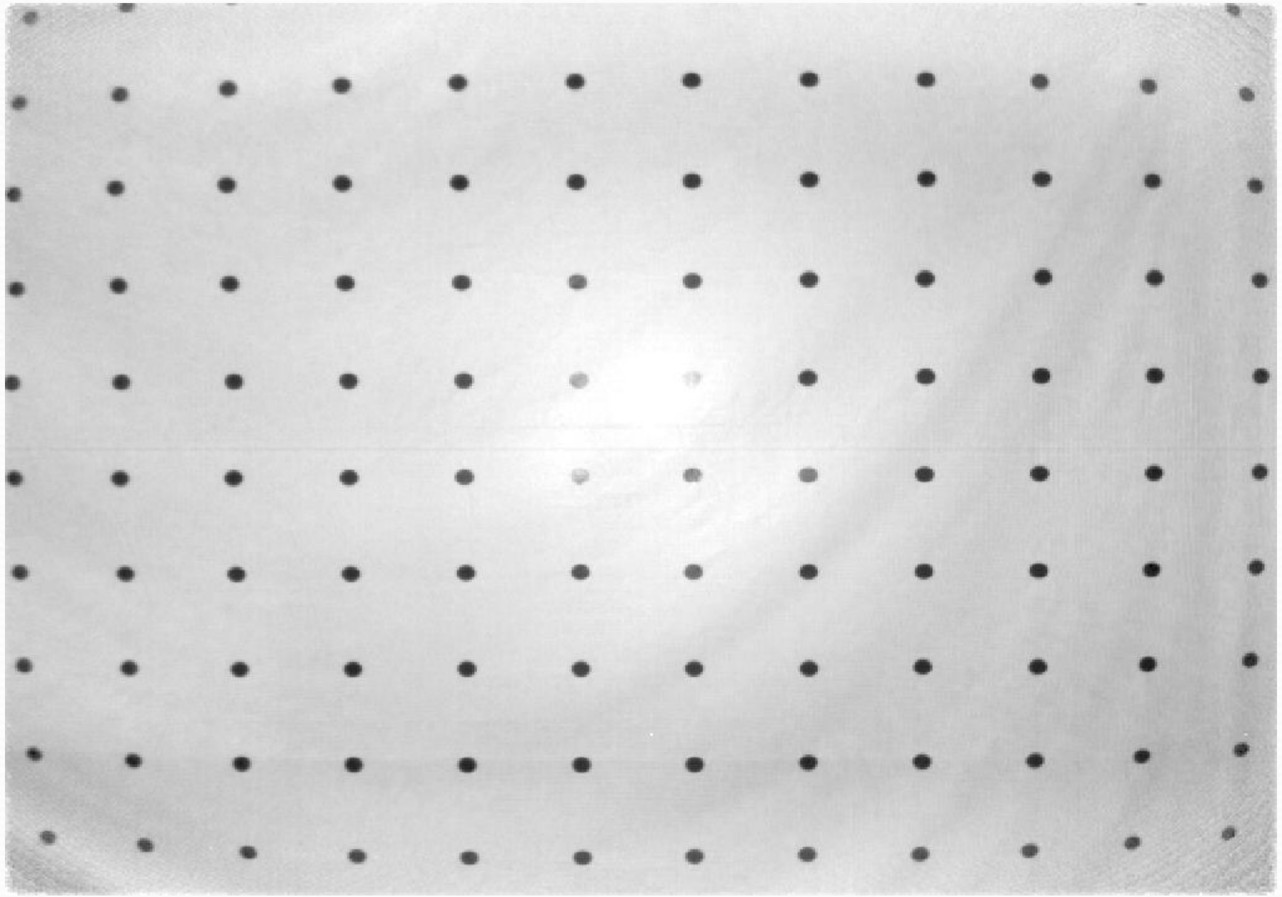
$$I_{\mathrm{MIN}} = g_{\min} / G$$

$$I_{\mathrm{MAX}} = g_{\max} / G$$

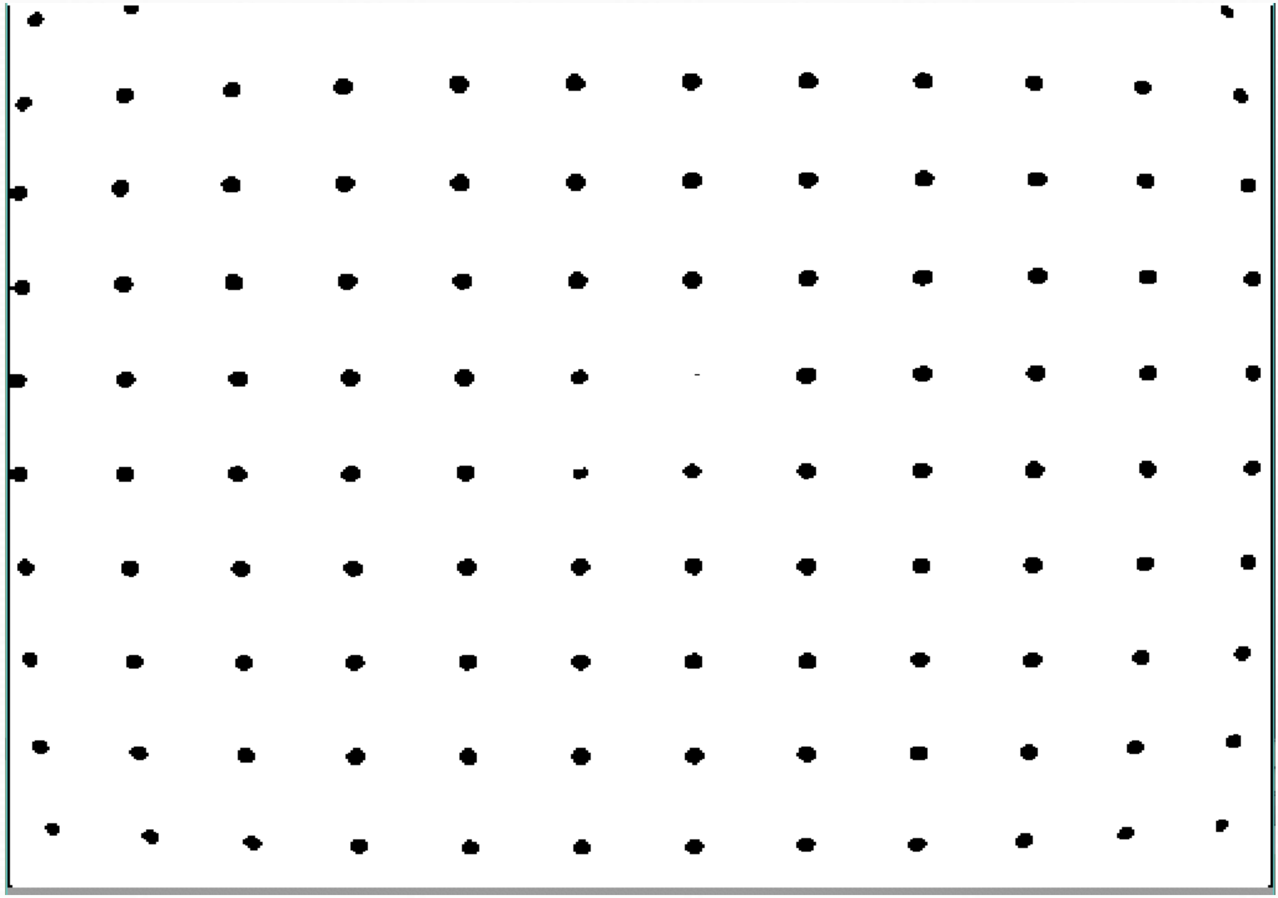
$$I_1[m,n] = \frac{I_0[m,n] - I_{\mathrm{MIN}}}{I_{\mathrm{MAX}} - I_{\mathrm{MIN}}}$$

- Adaptive Thresholding

$$I_2[m,n]=\begin{cases}1, & I_1[m,n]-I_{\mathrm{b}}[m,n]-C_0>0\\ 0, & else\end{cases}$$



Historgram Equalized



Binarized

- Round Dot Tracking

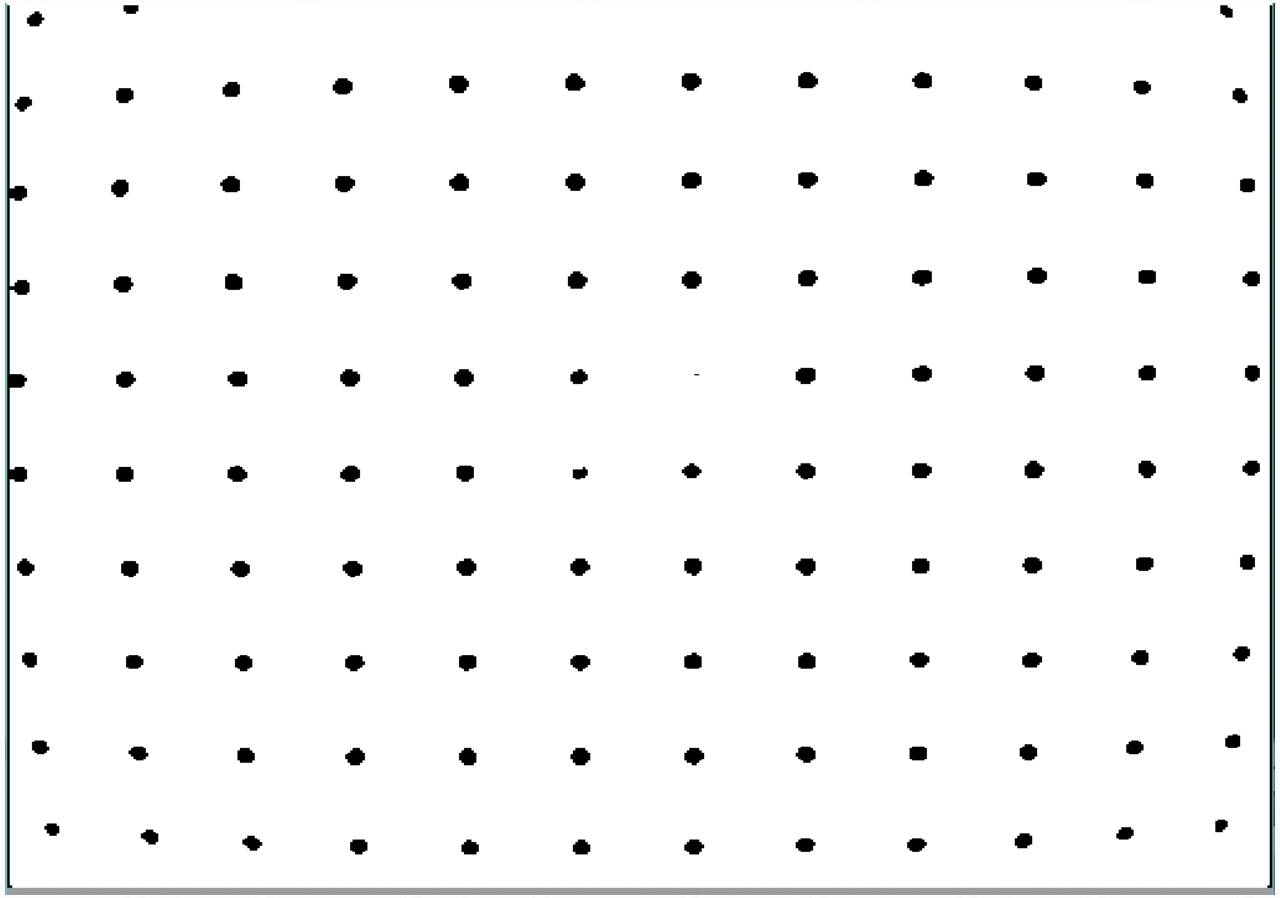
E	A	F
B	O	C
G	D	H

$$I_3[m,n] = I_2[m,n] * \frac{(128I_\mathrm{A} + 64I_\mathrm{B} + 32I_\mathrm{C} + 16I_\mathrm{D} + 8I_\mathrm{E} + 4I_\mathrm{F} + 2I_\mathrm{G} + I_\mathrm{H})}{255}$$

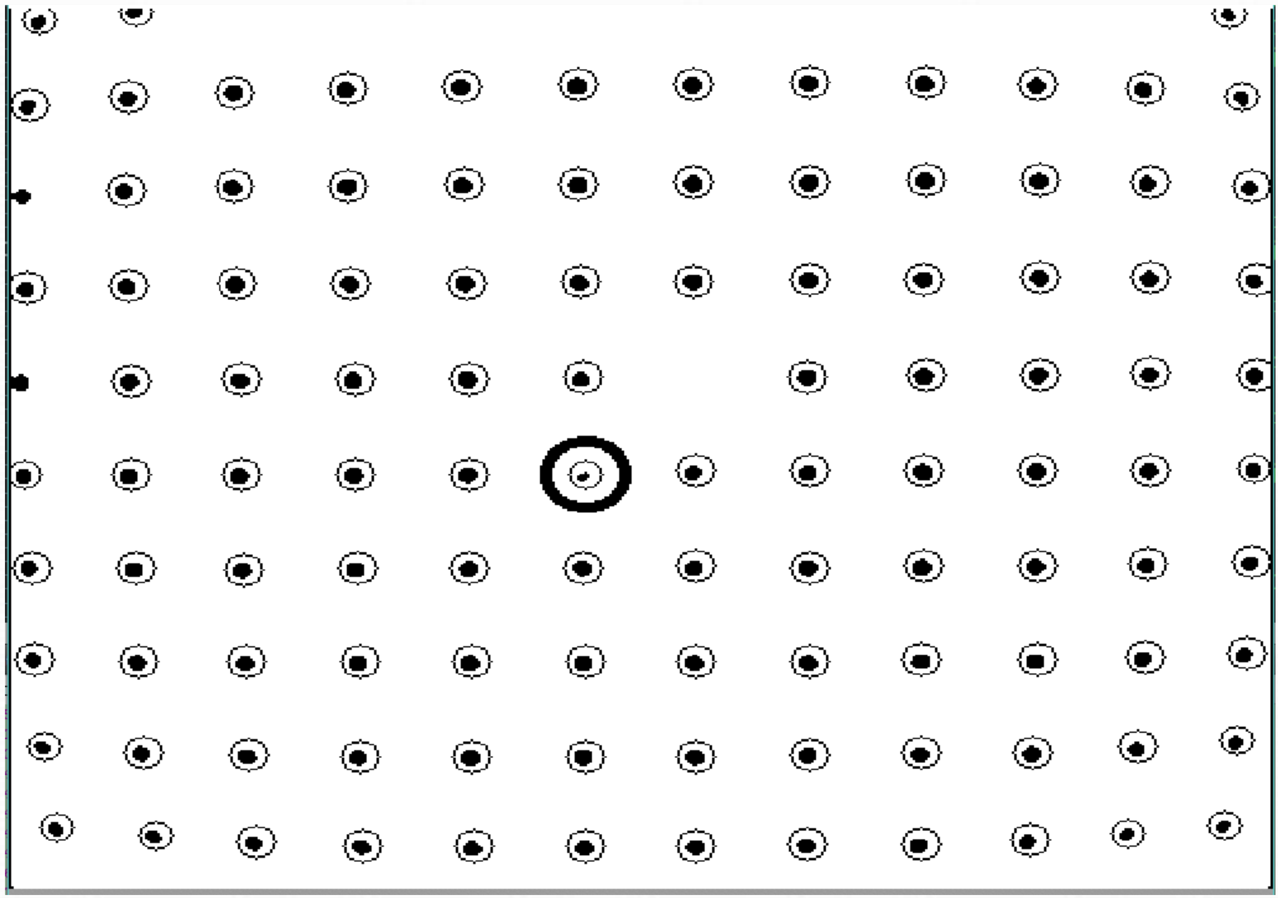
$$\begin{aligned}
&if\ (I_\mathrm{O}\ \&\ 0\mathrm{x}80 == 1), && then, && \text{marker } A \text{ is valid} \ \ (\text{ go Up })\\
&if\ (I_\mathrm{O}\ \&\ 0\mathrm{x}40 == 1), && then, && \text{marker } B \text{ is valid} \ \ (\text{ go Left})\\
&if\ (I_\mathrm{O}\ \&\ 0\mathrm{x}20 == 1), && then, && \text{marker } C \text{ is valid} \ \ (\text{ go Right })\\
&if\ (I_\mathrm{O}\ \&\ 0\mathrm{x}10 == 1), && then, && \text{marker } D \text{ is valid} \ \ (\text{ go Down })
\end{aligned}$$

Anchor !=0 → Detect Valid → Flip Valid → Get Area + Bounding Box

Traverse and Extract on CPU



Binarized Image

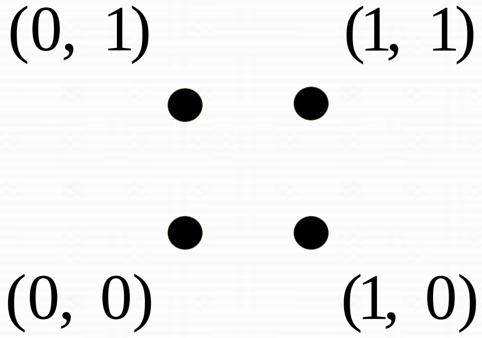


Dots' Centers Extracted

World Space Address Assignment

- Travers every 4 Square-Shaped points and find best $\boldsymbol{A_0}$, which corresponds to the best-fit “Unit One” (228mm) in Image Space:

Get Linear Mapping matrix $\boldsymbol{A_0}$ → Transform all (R, C) to (X^W, Y^W) → Count (N_v) valid points with integer X^W/Y^W → Find best $\boldsymbol{A_0}$ who generates the largest N_v.



$$\begin{bmatrix} zX^W \\ zY^W \\ z \end{bmatrix} = \boldsymbol{A}_0 \begin{bmatrix} C \\ R \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} C \\ R \\ 1 \end{bmatrix}$$

- Translate Origin to the Center Dot

$$c_h = (512-1)/2 = 255.5$$
$$r_h = (424-1)/2 = 211.5$$

$$\begin{bmatrix} zx_h \\ zy_h \\ z \end{bmatrix} = \boldsymbol{A}_0 \cdot \begin{bmatrix} c_h \\ r_h \\ 1 \end{bmatrix}$$

$$c_h = round(c_h)$$
$$r_h = round(r_h)$$

$$\boldsymbol{A}_1 = \boldsymbol{T} \cdot \boldsymbol{A}_0 = \begin{bmatrix} 1 & 0 & -x_h \\ 0 & 1 & -y_h \\ 0 & 0 & 1 \end{bmatrix} \cdot \boldsymbol{A}_0$$

World Space Address Assignment

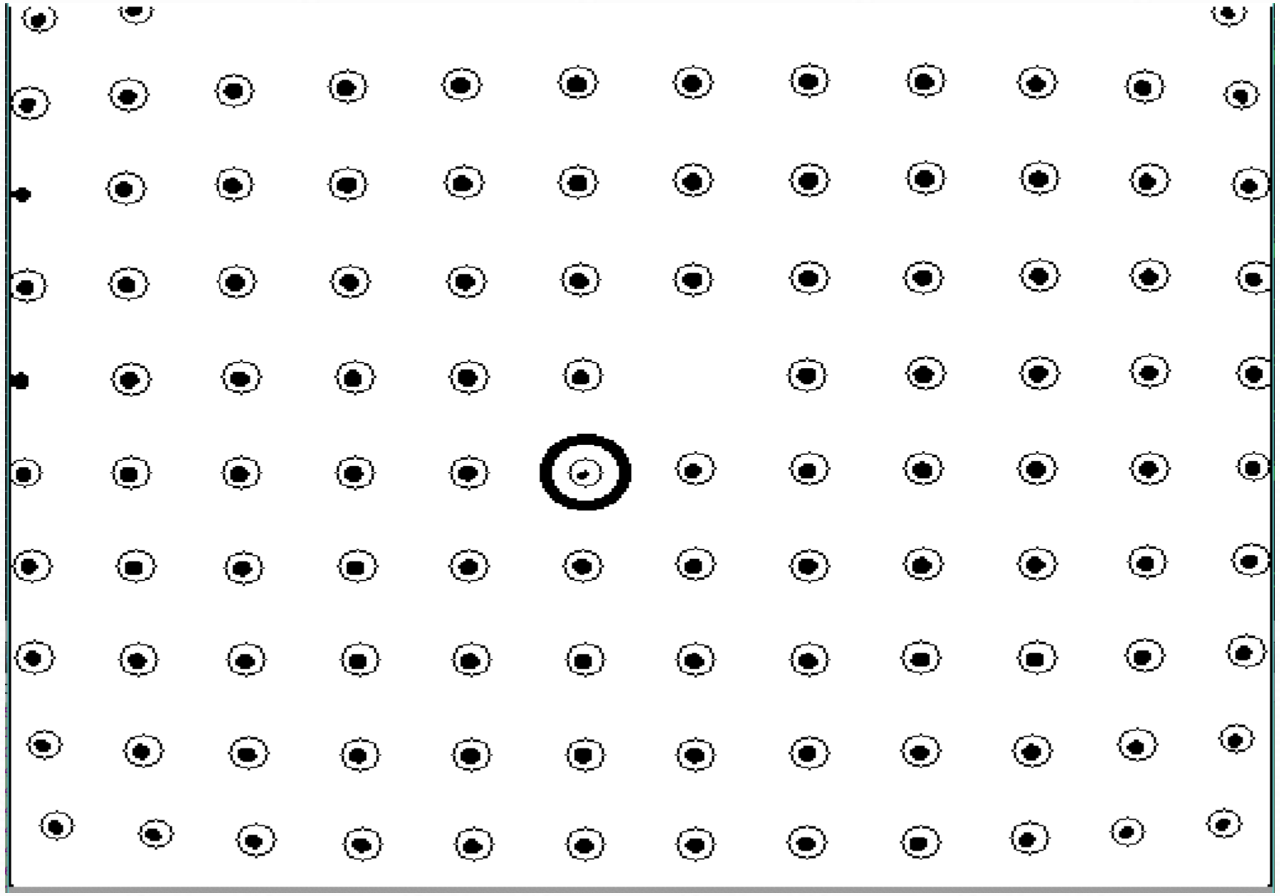
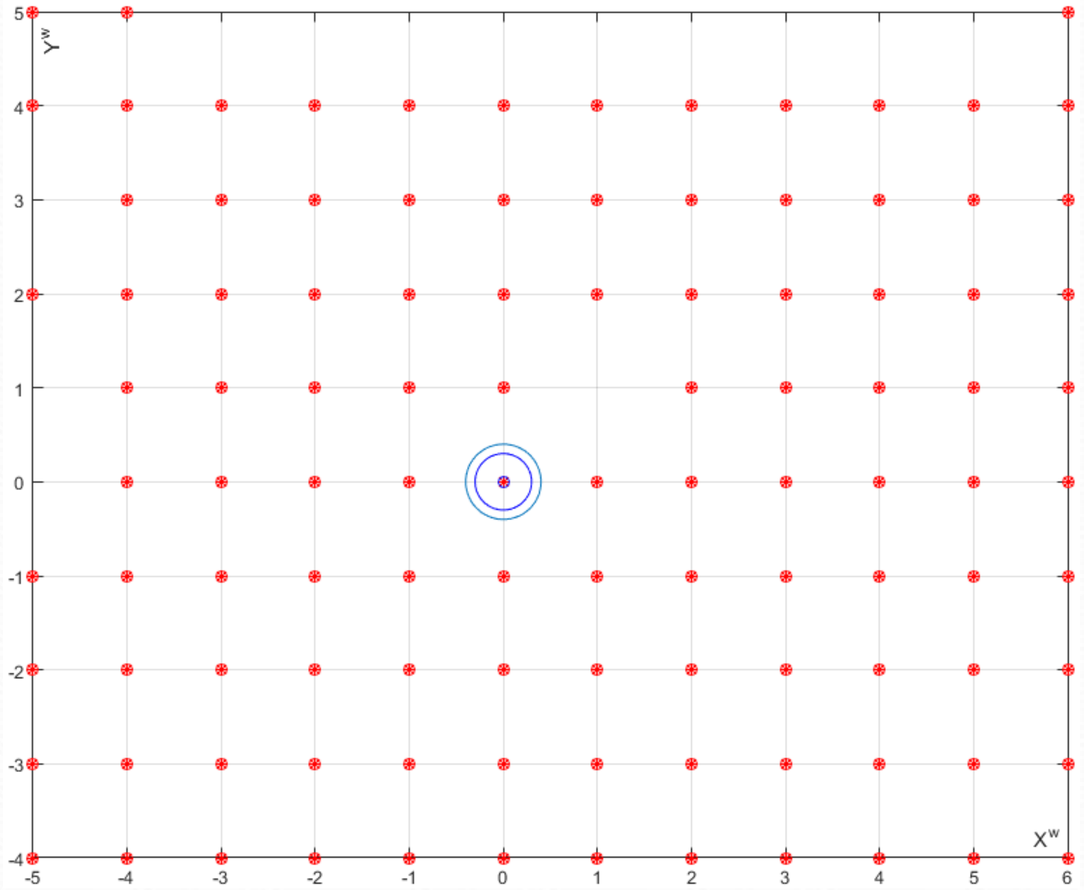


Image Space Calibration Points



Assigned World Space $X^w Y^w$

Two Dimensional Polynomial Mapping

- (First Order Perspective Transformation)

$$\begin{bmatrix} zX^W \\ zY^W \\ z \end{bmatrix} = \boldsymbol{A}_0 \begin{bmatrix} C \\ R \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} C \\ R \\ 1 \end{bmatrix}$$

- Second Order

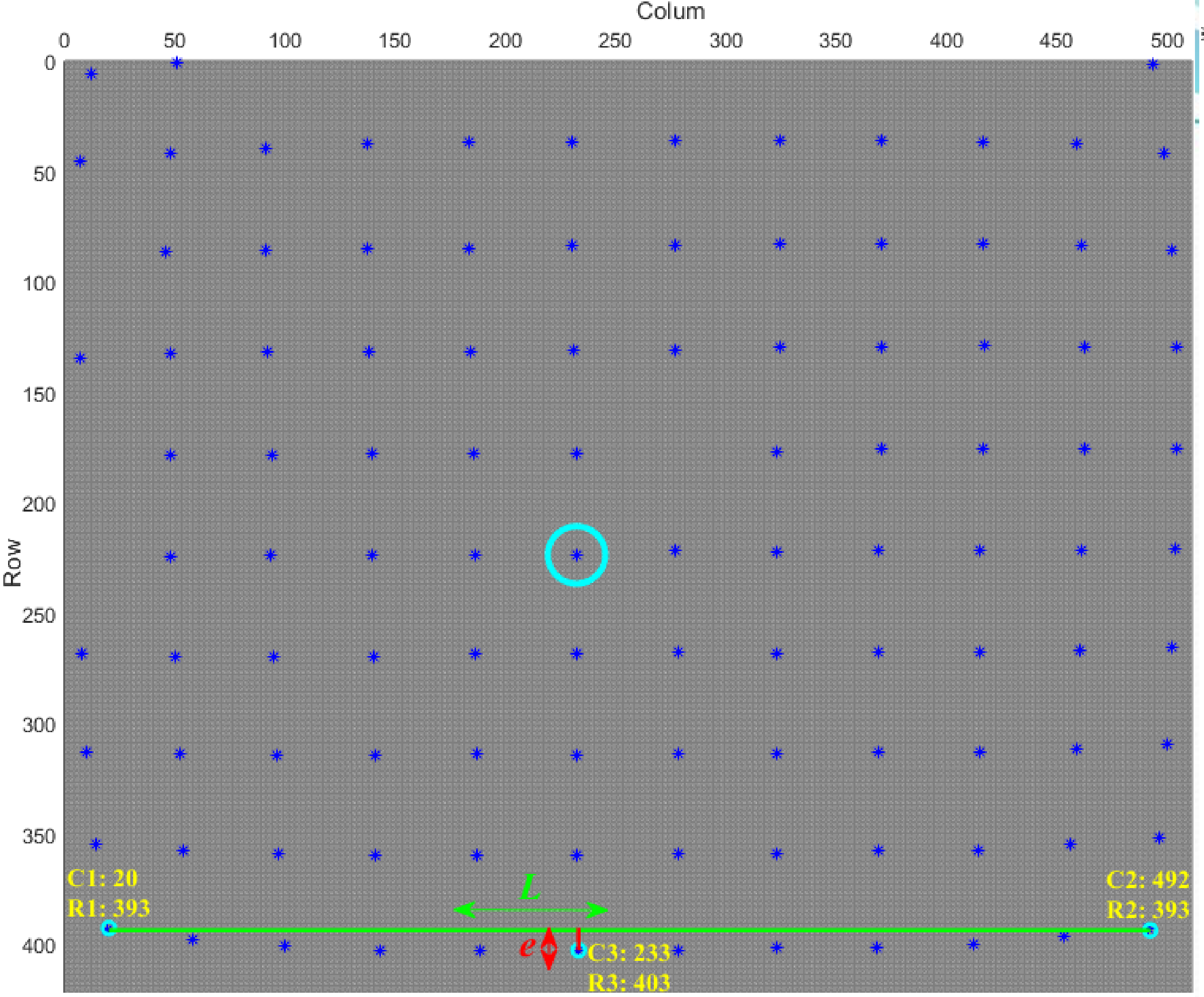
$$X^W = a_{11}C^2 + a_{12}CR + a_{13}R^2 + a_{14}C + a_{15}R + a_{16}$$

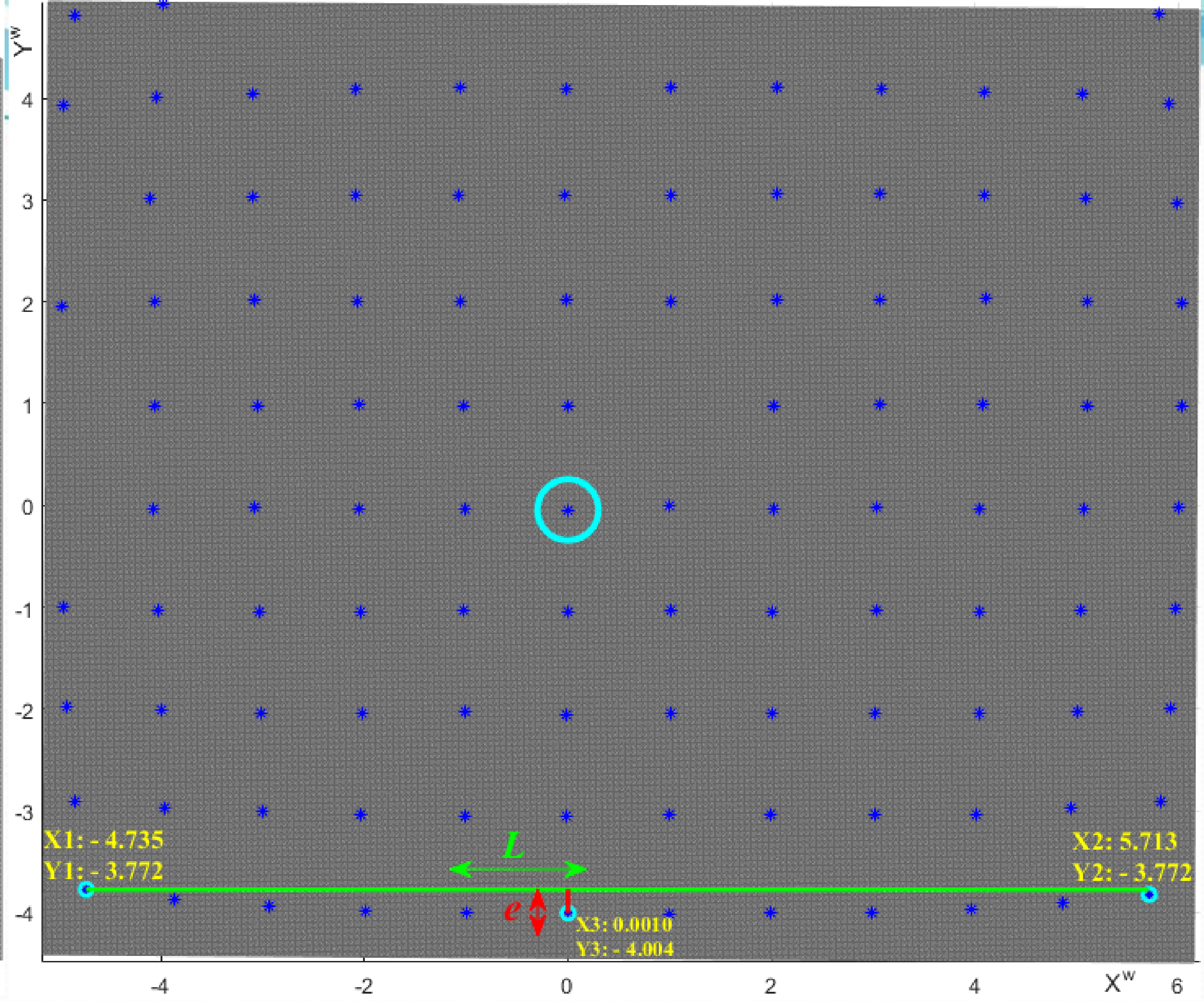
$$Y^W = a_{21}C^2 + a_{22}CR + a_{23}R^2 + a_{24}C + a_{25}R + a_{26}$$

- Fourth Order

$$X^W = a_{11}C^4 + a_{12}C^3R + a_{13}C^2R^2 + a_{14}CR^3 + a_{15}R^4 + a_{16}C^3 + a_{17}C^2R \quad \ldots$$
$$+ a_{18}CR^2 + a_{19}R^3 + a_{110}C^2 + a_{111}CR + a_{112}R^2 + a_{113}C + a_{114}R + a_{115}$$

$$Y^W = a_{21}C^4 + a_{22}C^3R + a_{23}C^2R^2 + a_{24}CR^3 + a_{25}R^4 + a_{26}C^3 + a_{27}C^2R \quad \ldots$$
$$+ a_{28}CR^2 + a_{29}R^3 + a_{210}C^2 + a_{211}CR + a_{212}R^2 + a_{213}C + a_{214}R + a_{215}$$

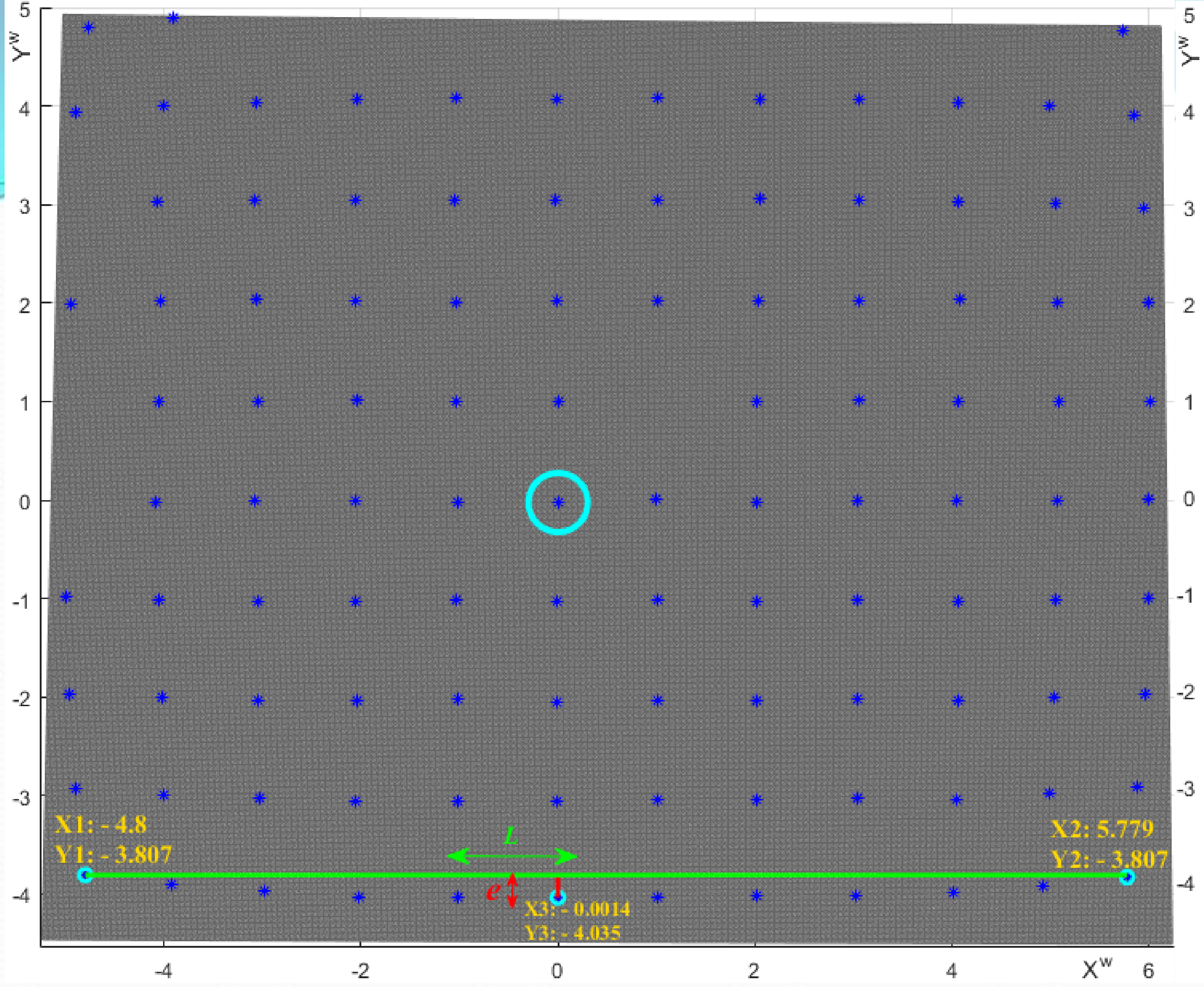


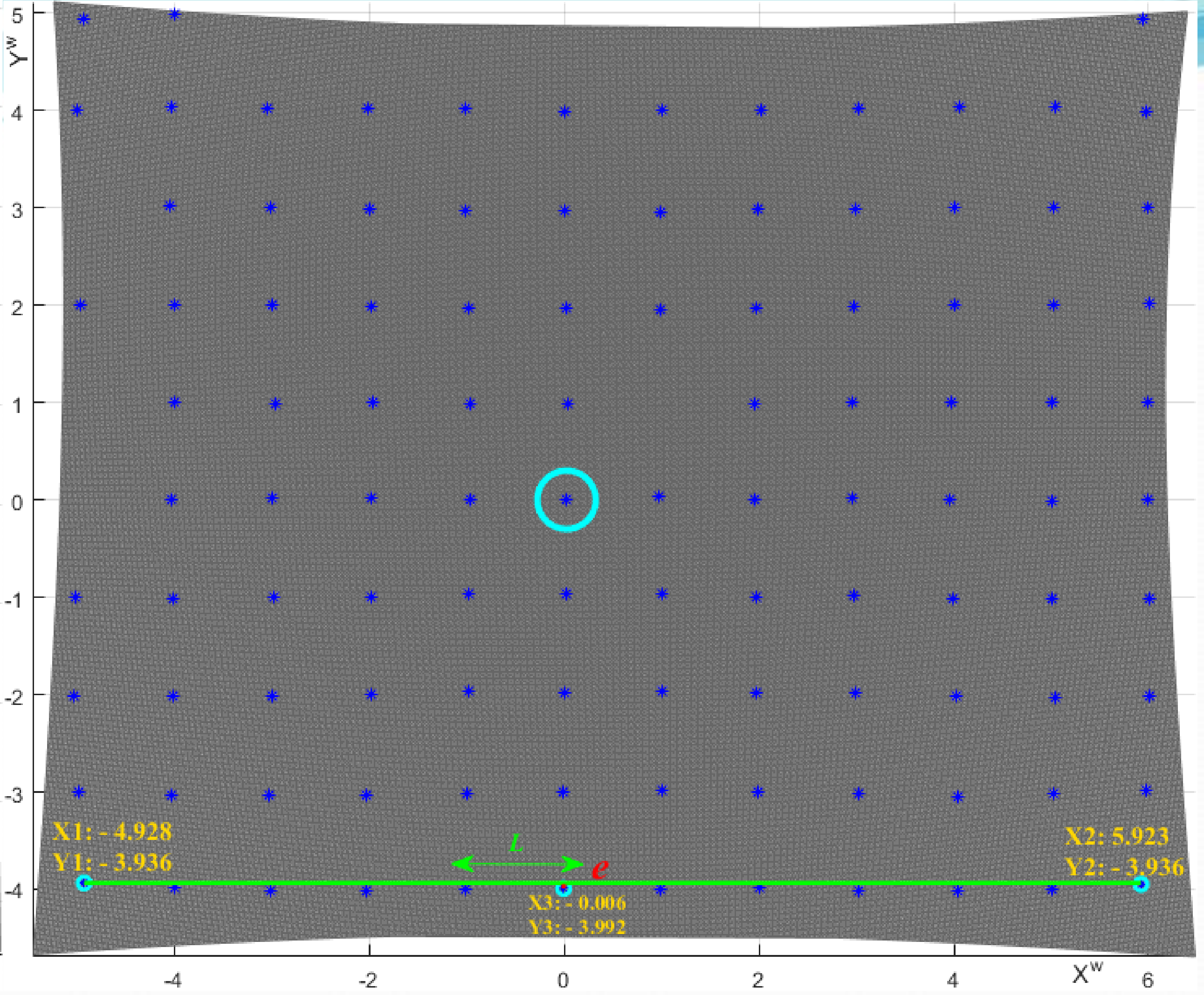


$$d(\%) = e / L$$

$$d_0 = (\mathrm{R3} - \mathrm{R1}) / (\mathrm{C2} - \mathrm{C1}) = (403 - 393) / (492 - 20) = 2.1\%$$

$$d_1 = (\mathrm{Y1} - \mathrm{Y3}) / (\mathrm{X2} - \mathrm{X1}) = [-3.772 + 4.004] / [5.713 + 4.735] = 2.2\%$$

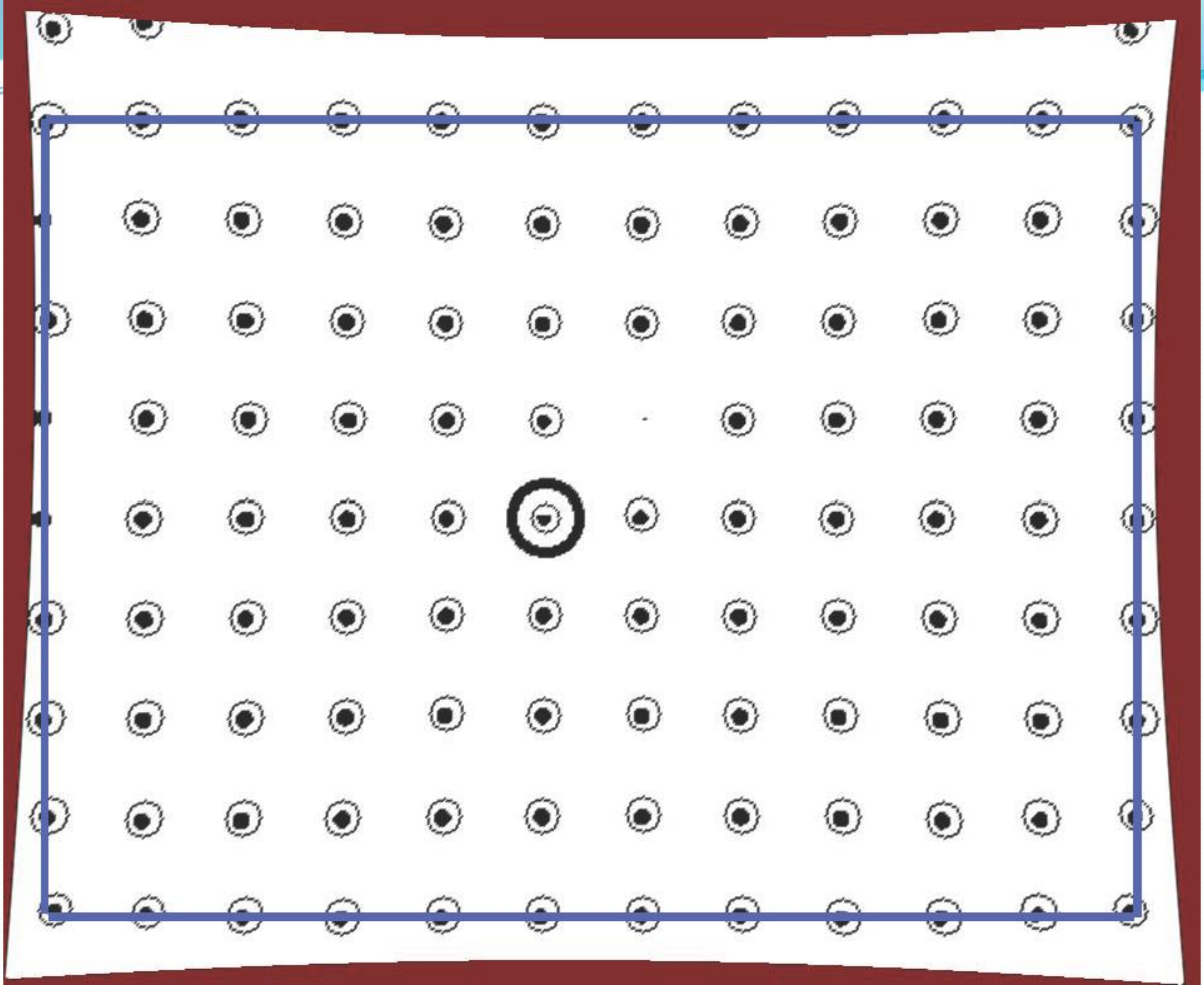




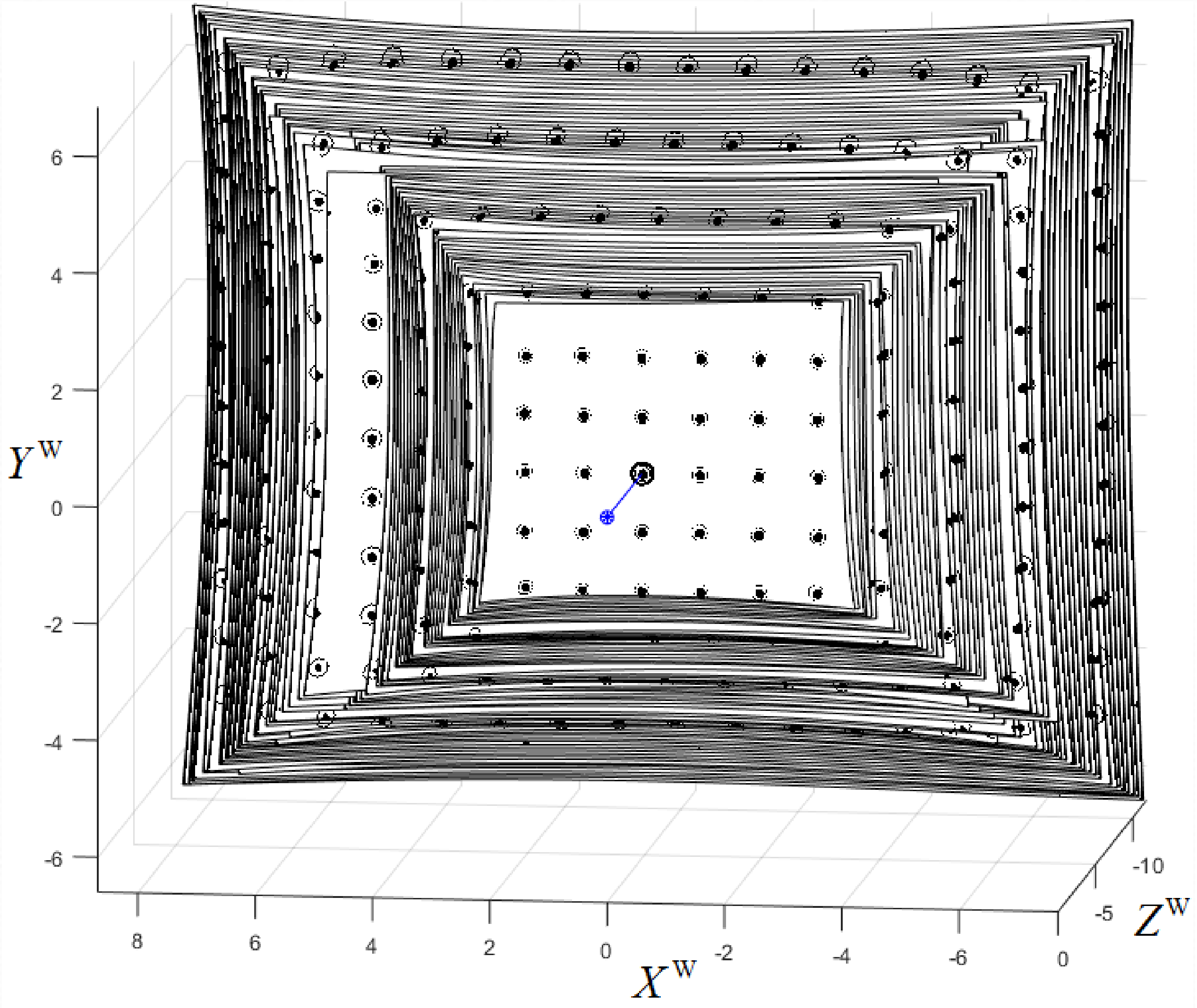
$$d(\%) = e / L$$

$$d_2 = [-3.807 - (-4.035)] = [5.779 - (-4.8)] = 2.1\%$$

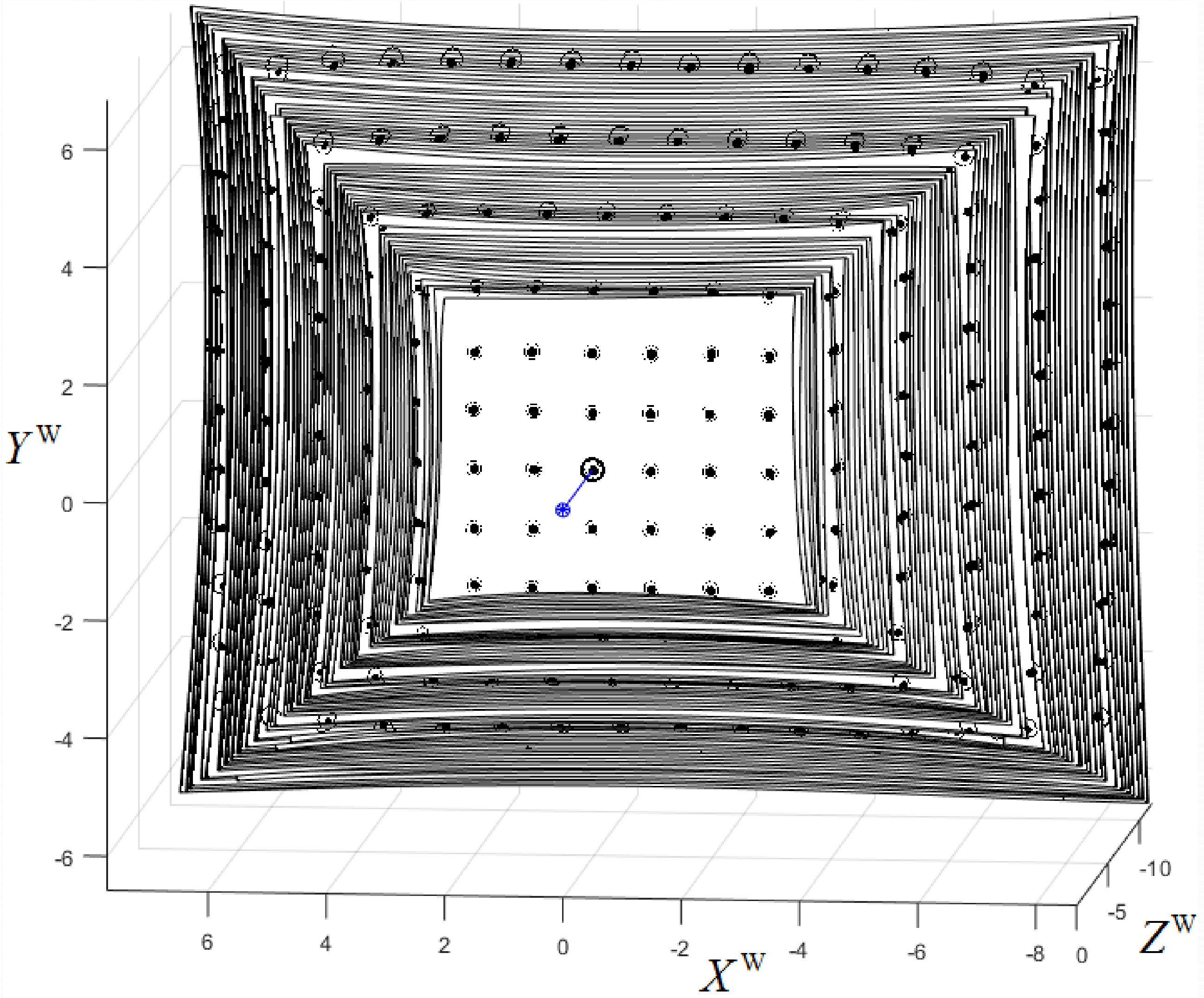
$$d_4 = [-3.936 - (-3.992)] = [5.923 - (-4.928)] = 0.516\%$$



63 frames from 1.165m to 2.565m, 25mm / frame

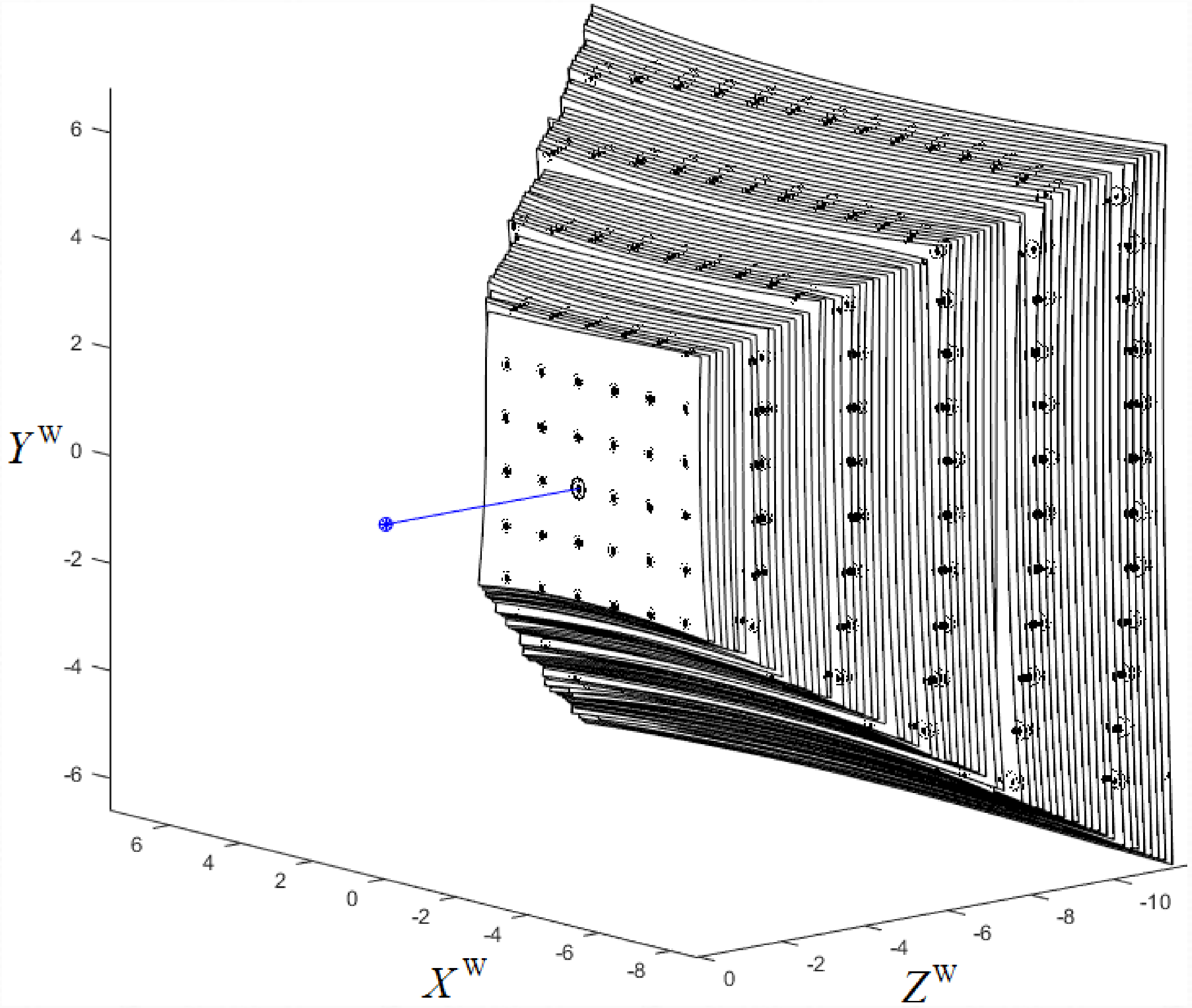


Staggered

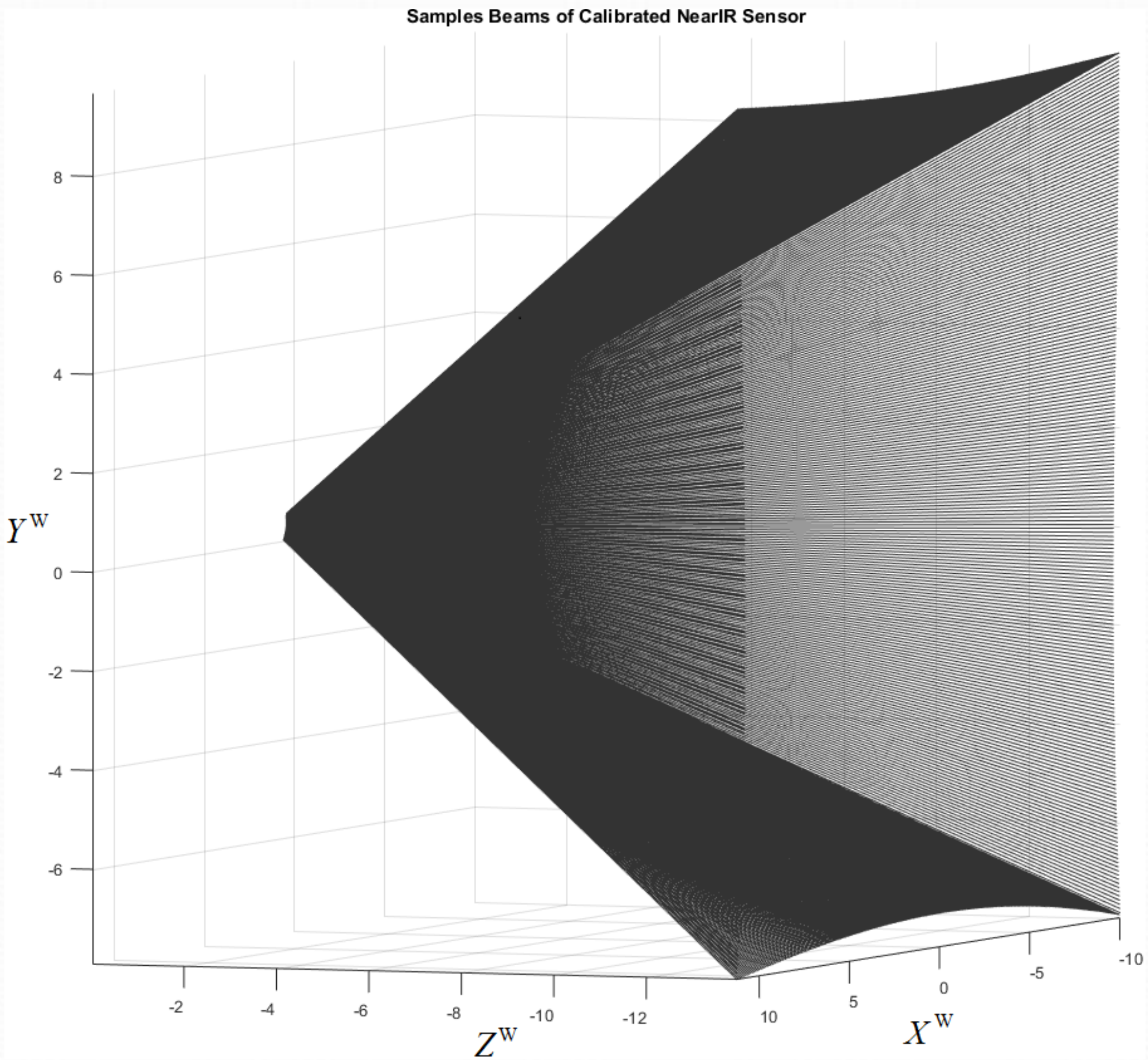


Unified

63 frames, generate Pixels' Beam Equations

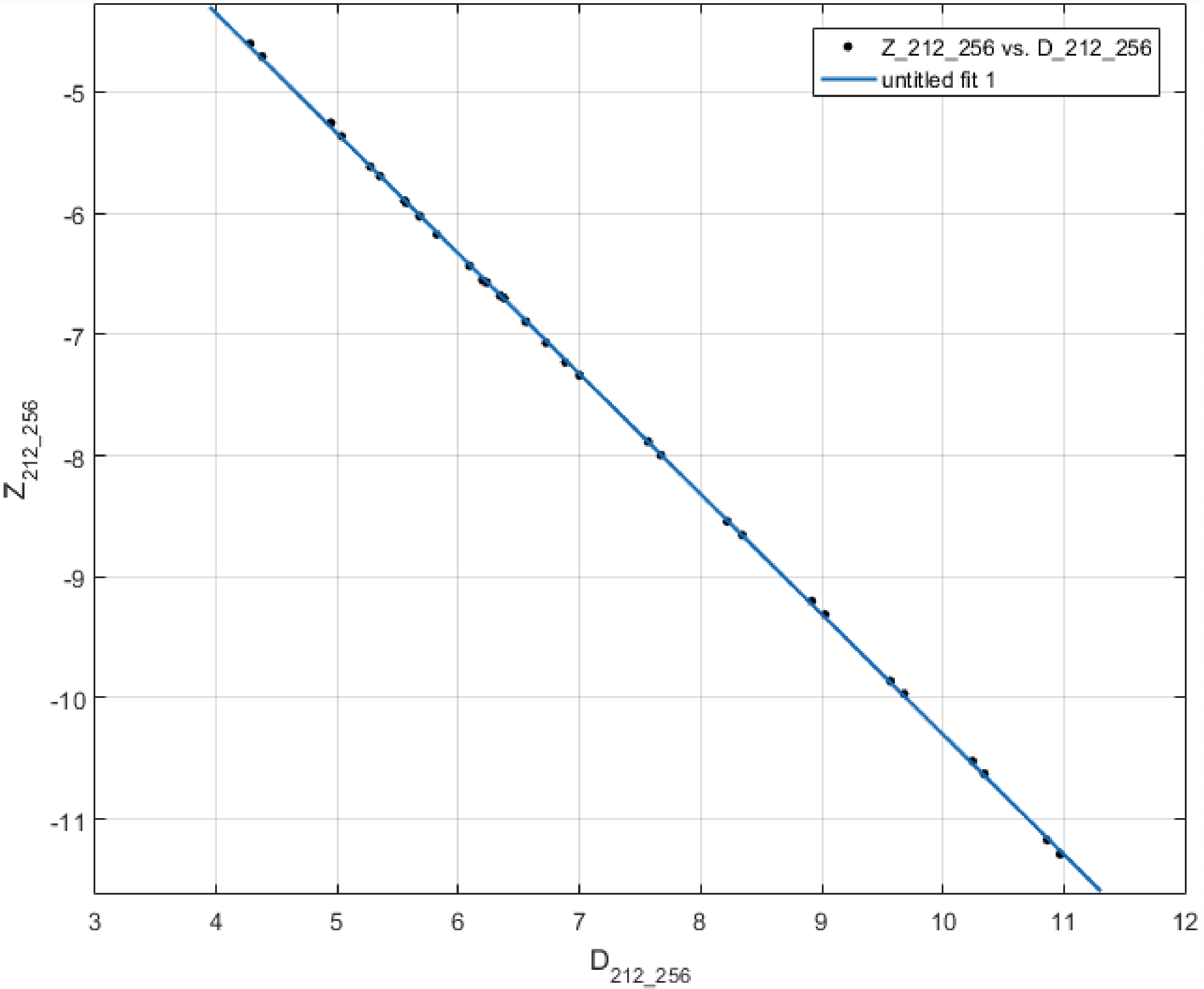


Frustum



Linear Z^W to $X^W Y^W$: Pixels' View

Per-Pixel D to Z^{W} mapping



D to Z^{W} Polynomial Fit

Data at pixel (212, 256) from 32 frames

↓

Linear in this example

↓

$$Z^{W}[m,n] = e[m,n]D[m,n] + f[m,n]$$

Generate Look-Up Table

- Size: *Width* - *Height* - 6 (512*424*6)
- Data: $X^{W} Y^{W} Z^{W} D$
- Pre-Process: (for every frame)
  - Find best-fit plane equation $D = aX^{W} + bY^{W} + c$
  - Throw away 10% pixels of worst D
- Mappings:

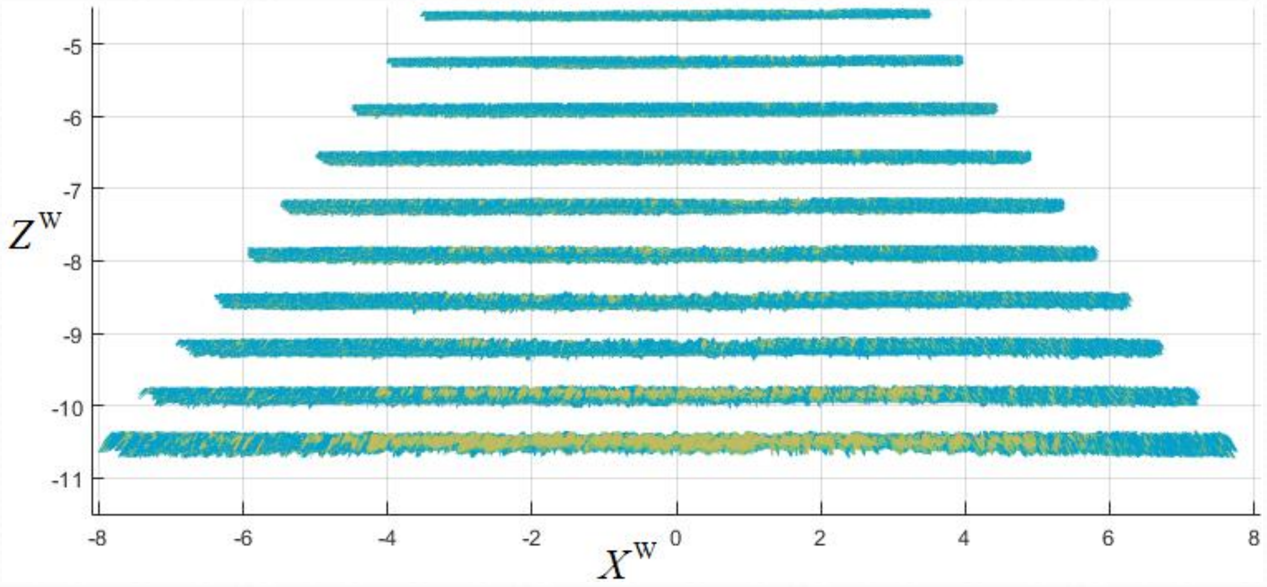
  Fragment Shader: 3 per-pixel linear mappings, 6 parameters

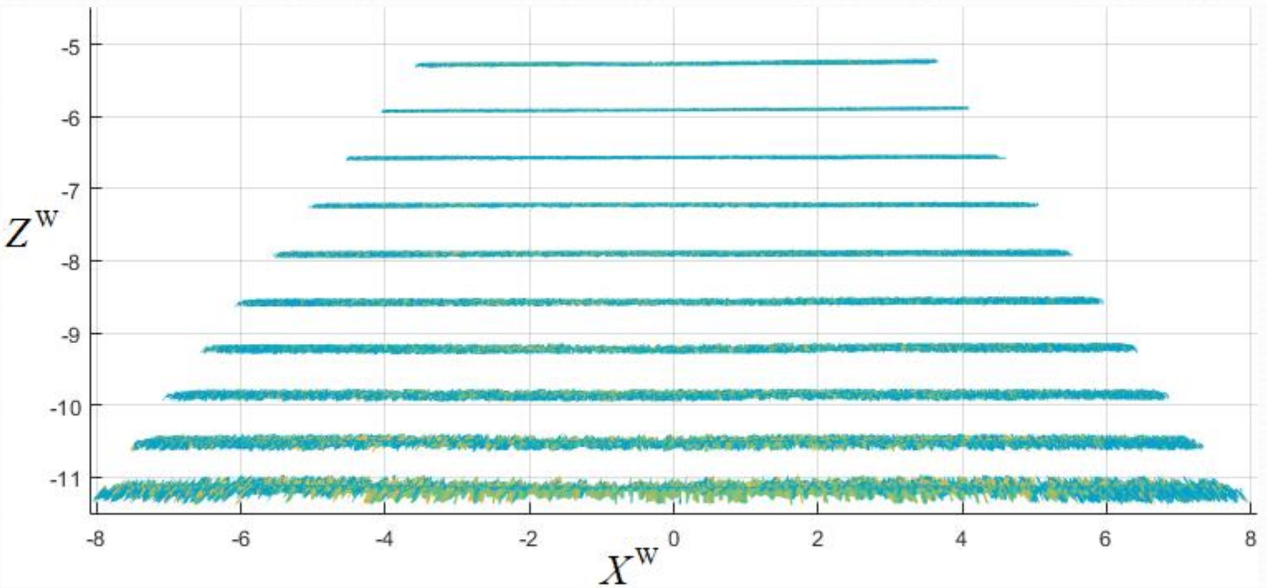
$$X^{W}[m,n] = a[m,n]Z^{W}[m,n] + b[m,n]$$

$$Y^{W}[m,n] = c[m,n]Z^{W}[m,n] + d[m,n]$$

$$Z^{W}[m,n] = e[m,n]D[m,n] + f[m,n]$$

Depth Distortion Correction

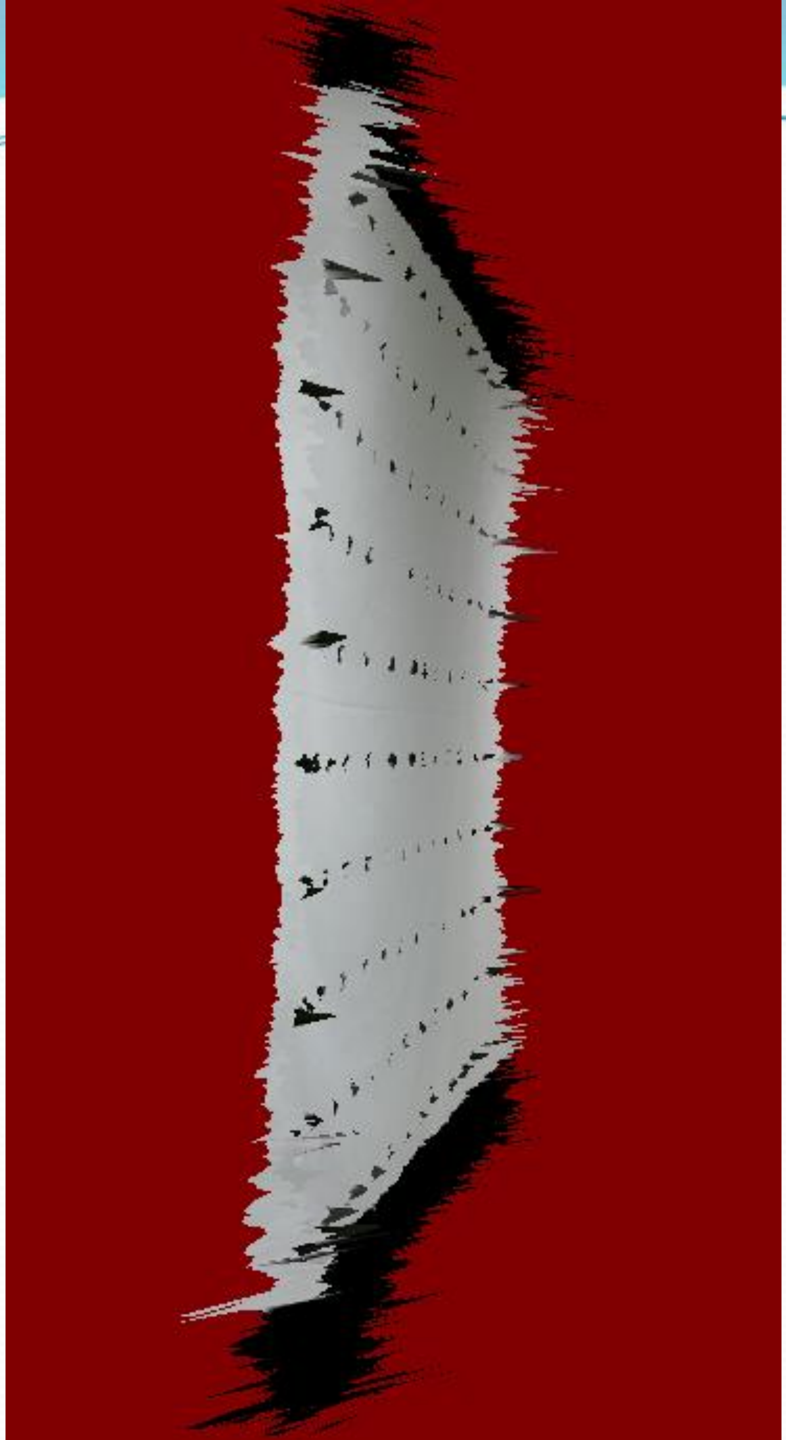




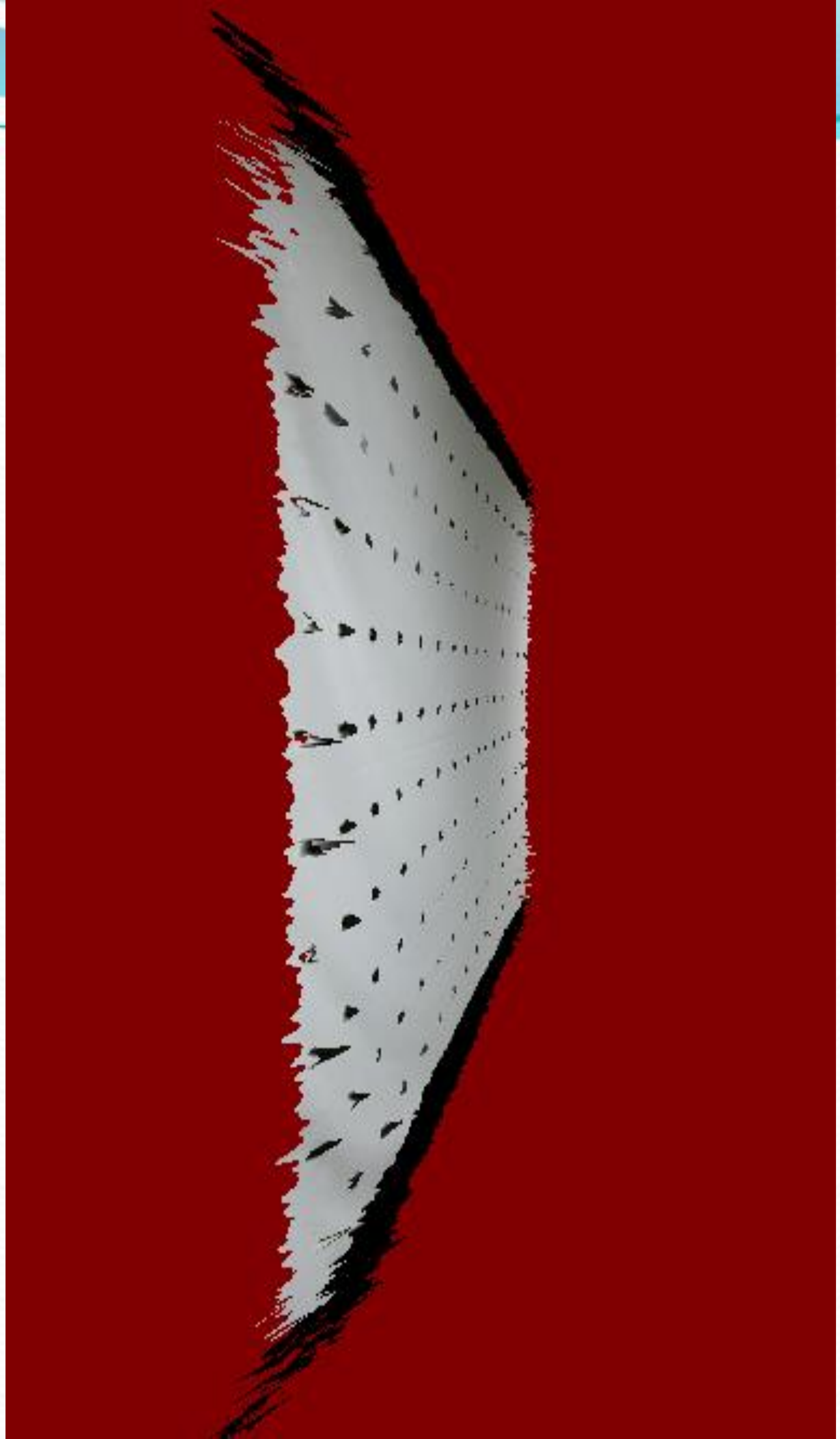
Raw Pin-Hole Reconstructions

Transformed
into World Space
By
Best-Fit *Rotation* and *Translation*

Calibrated LUT Reconstructions

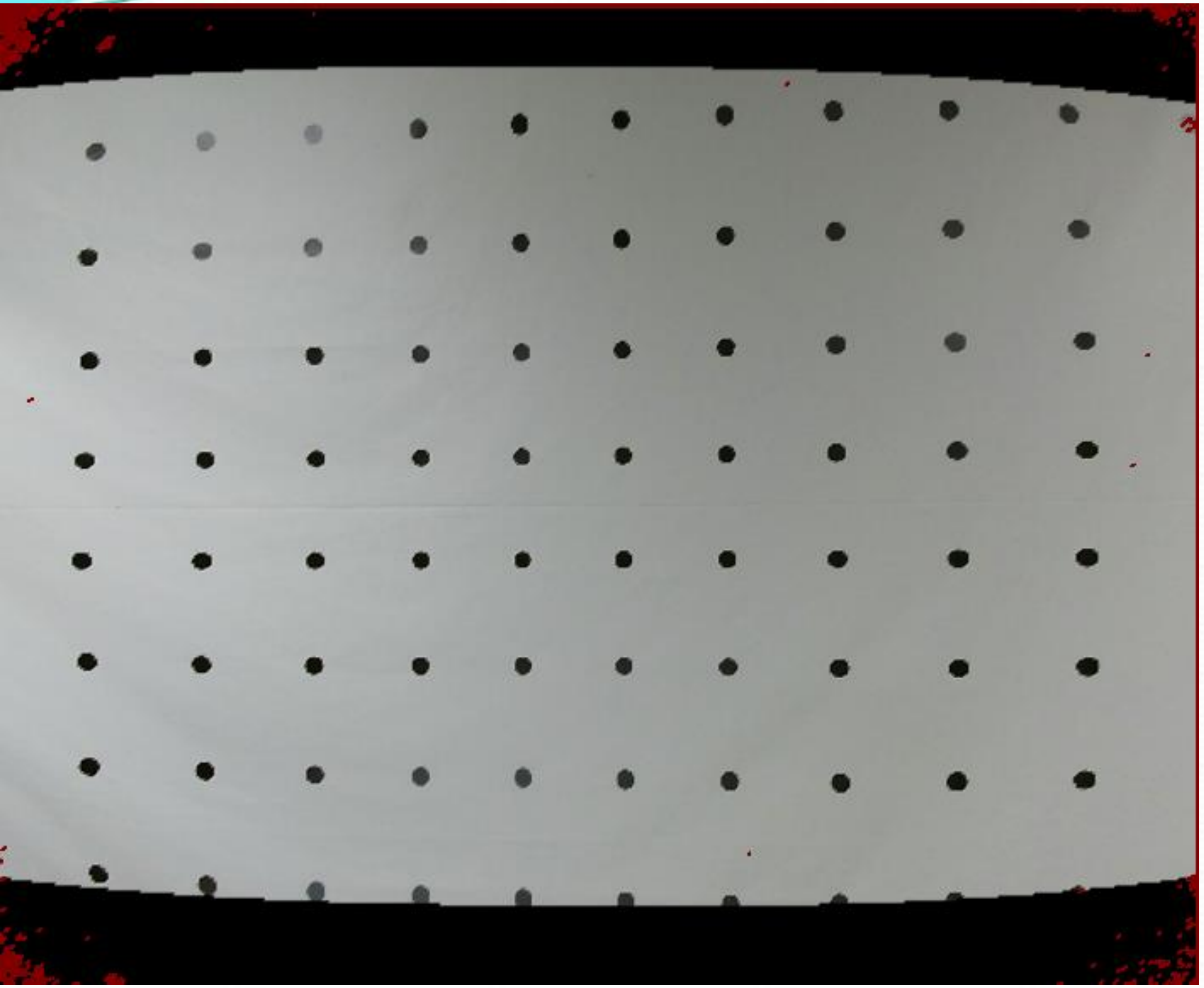


Before Calibration

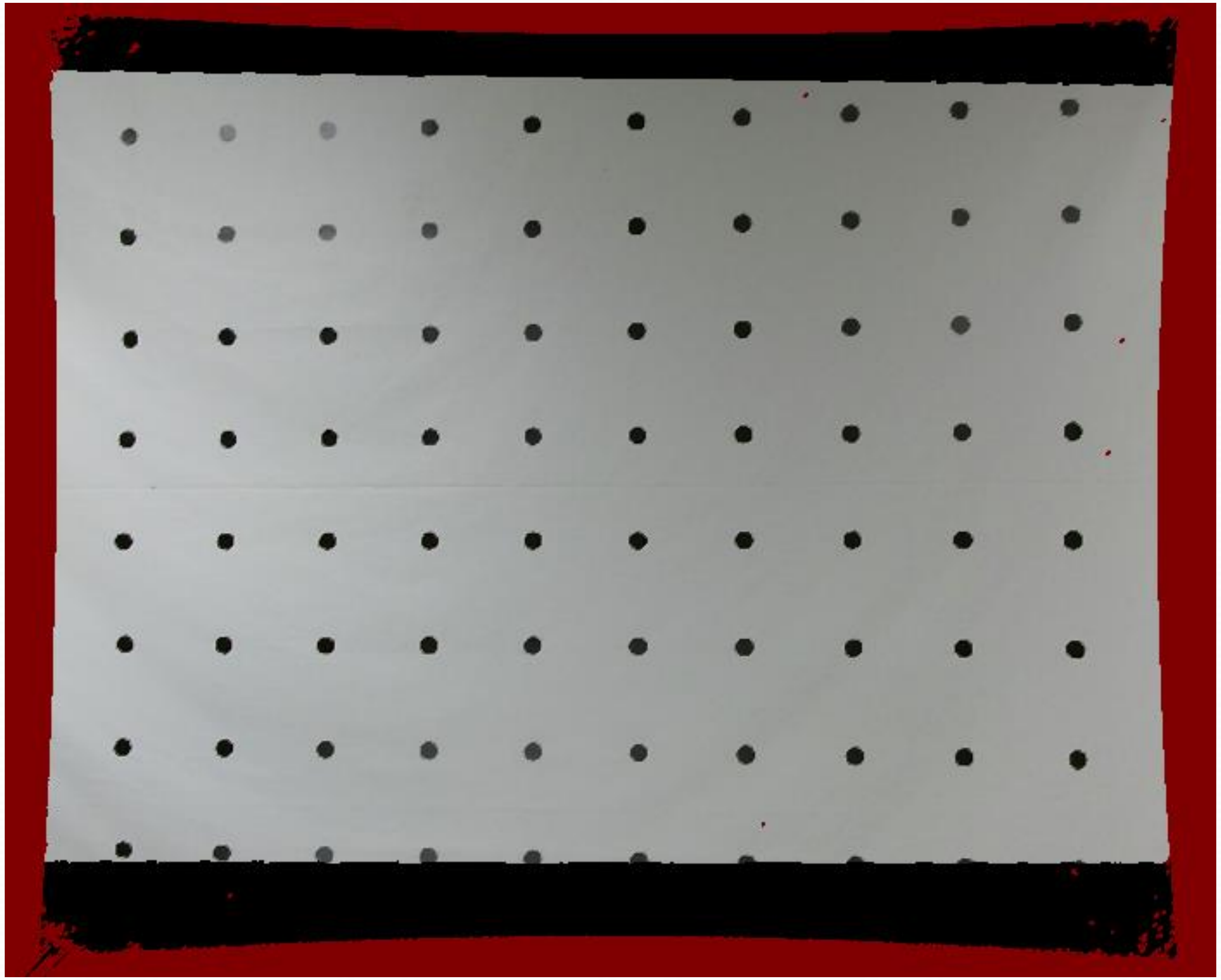


Calibrated

Lens Distortion Correction



Before Calibration



Calibrated

Conclusion

- Rail System

  Infinite frames of data, dense calibration points, per-pixel D to Z^W mapping
- Data Collection (Per-Frame)
  - Get Z^W from laser distance measurer ;
  - Robust calibration points' extraction ;

    Histogram Equalization, Adaptive Thresholding, Round Dots' Tracking
  - Assign world space coordinates to calibration points ;
  - Determine two-dimensional fourth order polynomial mapping ;
  - Generate dense undistorted $X^W Y^W$.
- Pre-Process
  - Unify staggered frames
  - Throw away 10% noise pixels
- Generate LUT: *Width* - *Height* - 6 (512*424*6)
- 3D Reconstruction: Undistorted 3D Reconstruction

Future Works

- Hard Ware
  - Longer rail: singular D to Z^W linear mapping to segmented mapping ;
  - Pattern Size and Distribution: based on resolution ;
  - 2D pattern to 3D pattern: in case NIR streams cannot be used, 3D pattern for depth streams analysis ;
  - Tracking module on rail: to substitute laser distance measure, such that it is possible to input Z^W record frames automatically.
- Software
  - Better DIP techniques ;
  - Higher order polynomial Z^W to $X^W Y^W$ mapping.

Questions?