

## Kinect: hoe werkt het?

Joris Stork, Jeroen Zuiddam en Lucas Swartsenburg

Kinect



# Kinect



Vandaag beginnen we met:

1.  
correspondence
2. applicatie

# Kinect



Vandaag beginnen we met:

1. **correspondence**
2. applicatie
1. **Mellin transformatie**
2. **light coding**
3. **schaal identificatie**
4. **extra referentie beelden**

# Kinect

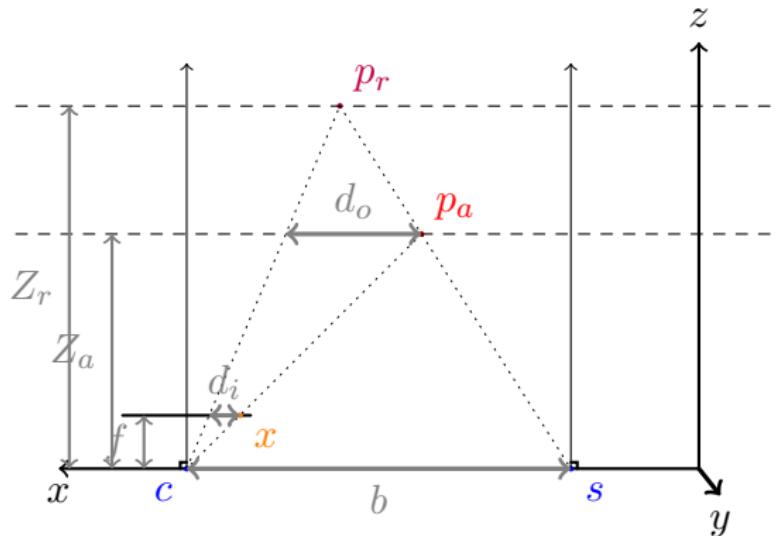


Daarna:

- |                      |                              |
|----------------------|------------------------------|
| 1.                   | 1. <b>stereo calibratie</b>  |
| correspondence       | 2. <b>creating a model</b>   |
| 2. <b>applicatie</b> | 3. <b>point cloud viewer</b> |

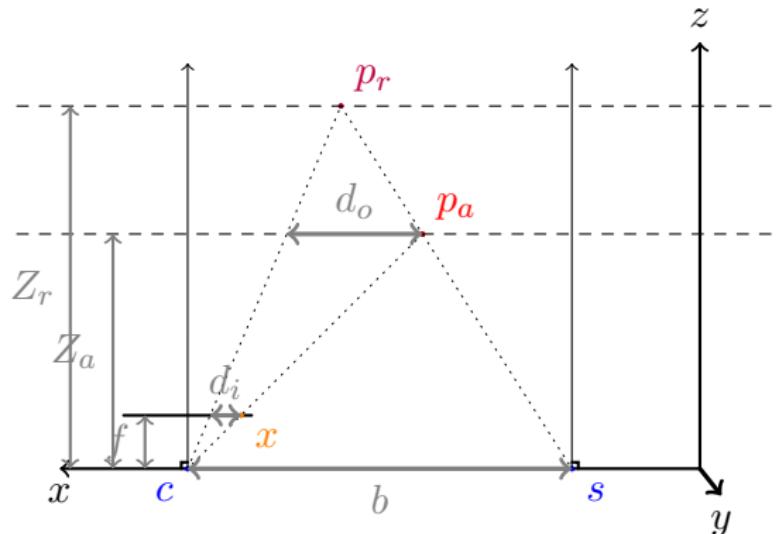
# (Actieve) Stereotriangulatie

Bepaalt afstand  $Z_a$



# (Actieve) Stereotriangulatie

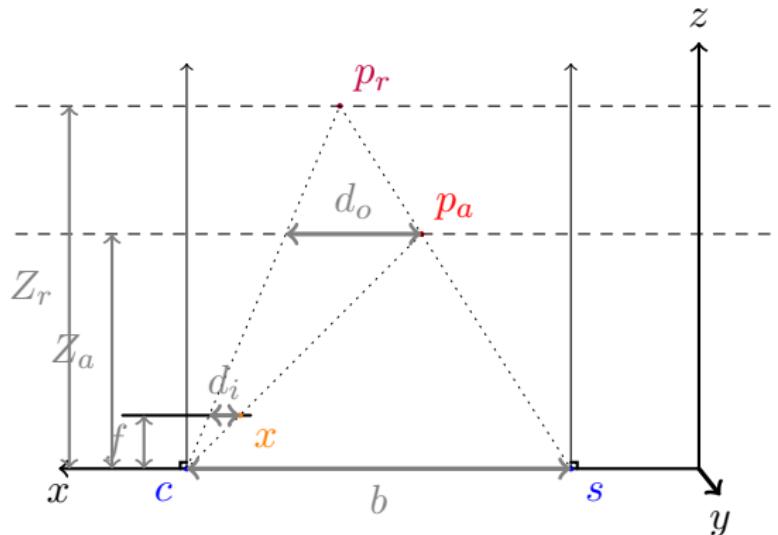
Bepaalt afstand  $Z_a$



Rekenvoorbeeld: twee parallele pinhole cameras

# (Actieve) Stereotriangulatie

Bepaalt afstand  $Z_a$

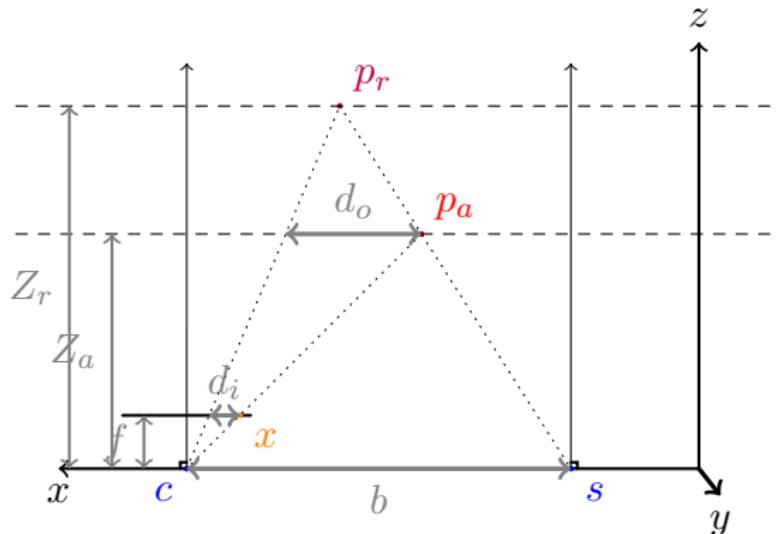


Rekenvoorbeeld: twee parallele pinhole cameras

$$\frac{z_1}{b} = \frac{z_1 - f}{b - x_l + x_r}$$

# (Actieve) Stereotriangulatie

Bepaalt afstand  $Z_a$



Rekenvoorbeeld: twee parallele pinhole cameras

$$\frac{z_1}{b} = \frac{z_1 - f}{b - x_l + x_r} \quad \Rightarrow \quad z_1 = \frac{bf}{x_l - x_r}$$

# Actieve stereotriangulatie

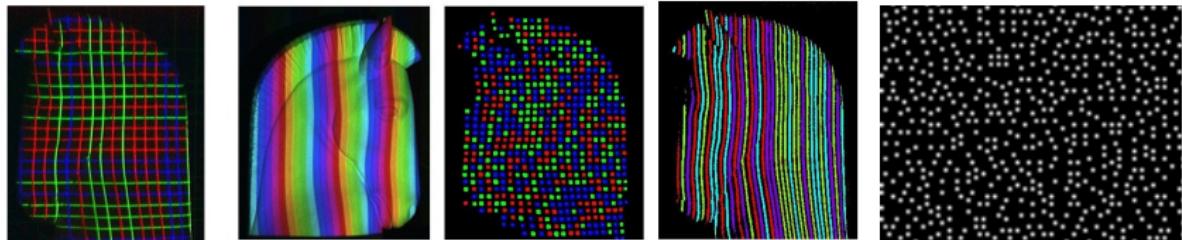
## Algemene situatie

1. camera
2. projector, laser or lamp

# Actieve stereotriangulatie

## Algemene situatie

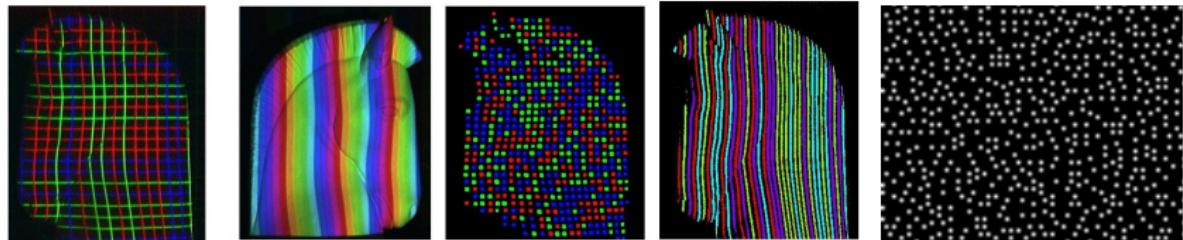
1. camera
2. projector, laser or lamp



# Actieve stereotriangulatie

## Algemene situatie

1. camera
2. projector, laser or lamp

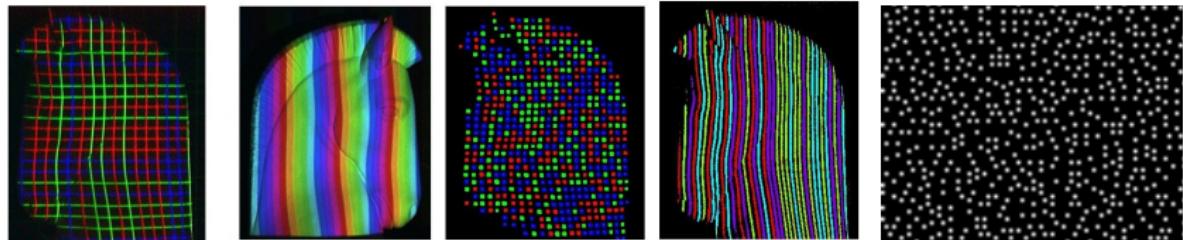


Rekenvoorbeeld: 'Kinect' met één geprojecteerd punt

# Actieve stereotriangulatie

## Algemene situatie

1. camera
2. projector, laser or lamp



Rekenvoorbeeld: 'Kinect' met één geprojecteerd punt

$$z_1 = \frac{z_0 b f}{b f + r_0 d}$$

# Achieve stereotriangulatie



# Correspondence problem

Correspondence problem

Welke referentiepunt hoort bij mijn punt?

# Correspondence problem

## Correspondence problem

Welke referentiepunt hoort bij mijn punt?

Een oplossing

1. maak referentiebeelden



# Correspondence problem

## Correspondence problem

Welke referentiepunt hoort bij mijn punt?

Een oplossing

1. maak referentiebeelden
2. neem camerabeeld



# Correspondence problem

## Correspondence problem

Welke referentiepunt hoort bij mijn punt?

Een oplossing

1. maak referentiebeelden
2. neem camerabeeld
3. voor elk punt



# Correspondence problem

## Correspondence problem

Welke referentiepunt hoort bij mijn punt?

Een oplossing

1. maak referentiebeelden
2. neem camerabeeld
3. voor elk punt
  - a. neem regio



# Correspondence problem

## Correspondence problem

Welke referentiepunt hoort bij mijn punt?

Een oplossing

1. maak referentiebeelden
2. neem camerabeeld
3. voor elk punt
  - a. neem regio
  - b. bepaal schaal



# Correspondence problem

## Correspondence problem

Welke referentiepunt hoort bij mijn punt?

Een oplossing

1. maak referentiebeelden
2. neem camerabeeld
3. voor elk punt
  - a. neem regio
  - b. bepaal schaal
  - c. correleer regio over referentiebeeld met die schaal



# Correspondence problem

## Correspondence problem

Welke referentiepunt hoort bij mijn punt?

Een oplossing

1. maak referentiebeelden
2. neem camerabeeld
3. voor elk punt
  - a. neem regio
  - b. bepaal schaal
  - c. correleer regio over referentiebeeld met die schaal
  - d. punt met hoogste correlatie is referentiepunt



## Modifications to Freenect

- ▶ Freenect: opensource driver Kinect
- ▶ Geen infrarood beeld
- ▶ Hack nodig voor calibratie

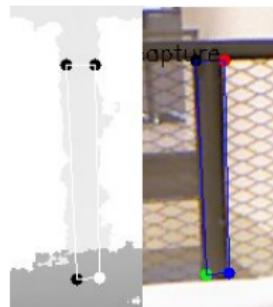
# Modifications to Freenect

- ▶ Freenect: opensource driver Kinect
- ▶ Geen infrarood beeld
- ▶ Hack nodig voor calibratie



# Stereo calibratie

- ▶ Undistort
- ▶ Rectify
- ▶ Translate



# Stereo calibratie

$$sm' = A [R|t] M'$$

or

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$f_x, f_y$ : focal lengths

$c_x, c_y$ : coordinaten van principal point

# Open CV

We gebruiken:

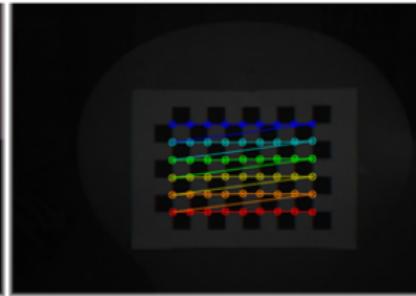
- ▶ `FindChessboardCorners`
- ▶ `DrawChessboardCorners`
- ▶ `StereoCalibrate`
- ▶ `StereoRectify`
- ▶ `FindHomography`

# Open CV

Task 2/5 shots. Press any key to continue.



Task 5/5 shots. Press any key to continue.



# Echte diepte

Dr. Stéphane Magnenat

- $depth = 12.36 * \tan(rawdepth/2842.5 + 1.1863)$

# 3D model maken

Eerst:

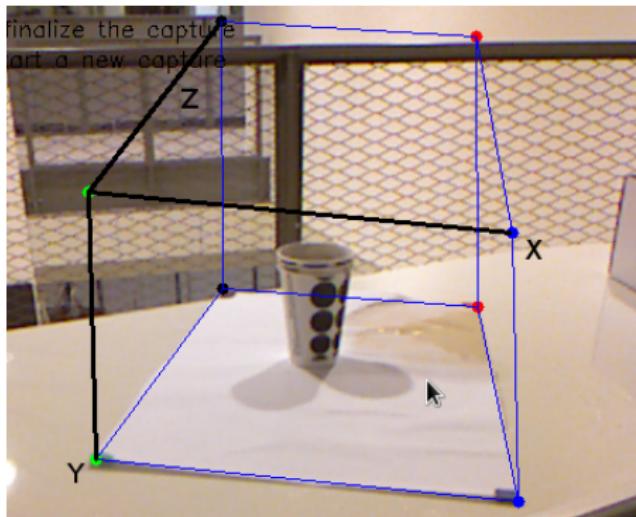
- ▶ InitUndistortRectifyMap
- ▶ WarpPerspective

## Extrinsic matrix

Extrinsic matrix met `FindExtrinsicCameraParams2` en  
`Rodriques2`

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

# Extrinsic matrix



# De punten verkrijgen

Bepaal de waarde voor  $s$  door  $depth(u, v)$ .

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}^{-1} \begin{bmatrix} us \\ vs \\ s \end{bmatrix} - \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

# Point cloud viewer

VPython

