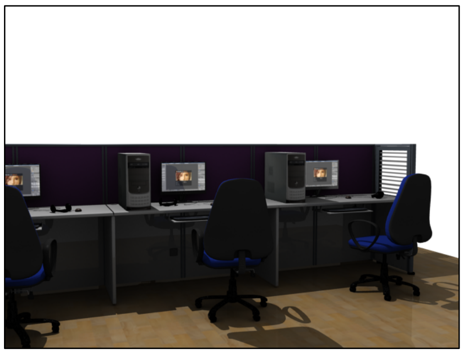
Outline

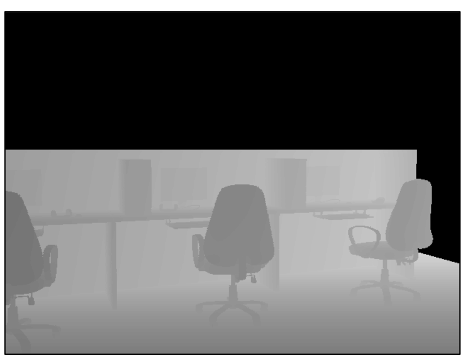
- Introduction
- Part I: Basics of Mathematical Optimization
  - Linear Least Squares
  - Nonlinear Optimization
- Part II: Basics of Computer Vision
  - Camera Model
  - Multi-Camera Model
  - Multi-Camera Calibration
- Part III: Depth Cameras
  - Passive Stereo
  - Structured Light Cameras
  - Time of Flight Cameras

Depth Cameras

- **Aim:** Measure distances to scene, compute distance image/depth map (depth = Z coordinate)



Color image $\mathbf{I}$



Distance image $\mathbf{D}$

$d_{\min}$ $d_{\max}$

$D(u,v) = 0$: undefined depth

Depth Cameras

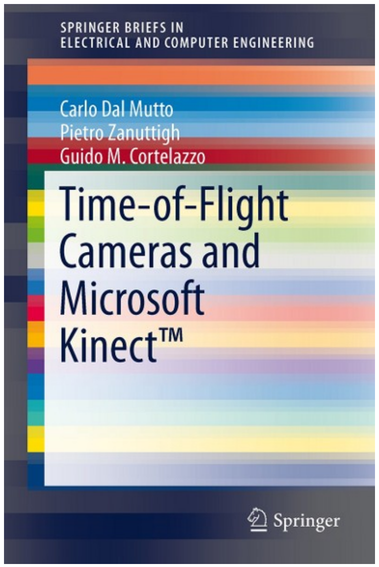
- **Aim:** Measure distances to scene, compute distance image/depth map
- **Technologies:**
  - Stereo Vision (passive): Analyze depth of scene from images captured at different vantage points
  - Structured Light camera (active stereo vision): Analyze known light pattern projection into scene
  - Time of Flight sensor: Measure light travel time

Depth Cameras

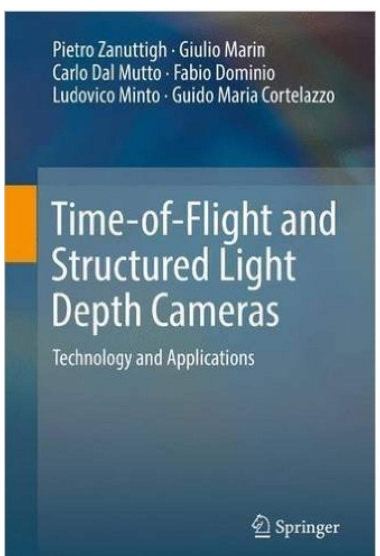
- **Aim:** Measure distances to scene, compute distance image/depth map
- Consider active optical reflection-based methods here
- Active = generate own electromagnetic radiation (*e. g.*, with lasers, lamps, LEDs) and analyze reflections from observed objects

<table>
<tr><td colspan="3">non-contact / reflection-based methods</td></tr>
<tr><td colspan="2">optical</td><td>non-optical</td></tr>
<tr><td>active</td><td>passive</td><td rowspan="2">• Radar
• Sonar
• ...</td></tr>
<tr><td>• Structured Light
• Time of Flight
• ...</td><td>• Stereo Vision
• Structure from Motion
• Shape from Shading
• Shape from Silhouette
• ...</td></tr>
</table>

Literature on Depth Cameras



- Carlo Dal Mutto, Pietro Zanuttigh, Guido M. Cortelazzo: *Time-of-Flight Cameras and Microsoft Kinect™*. Springer Briefs in Electrical and Computer Engineering. Springer, 2012.



- Pietro Zanuttigh, Guilio Marin, Carlo Dal Mutto, Fabio Dominio, Ludovico Minto, Guido M. Cortelazzo: *Time-of-Flight and Structured Light Depth Cameras. Technology and Applications.* Springer, 2016.

Literature on Depth Cameras

- Andrea Corti, Silvio Giancola, Giacomo Mainetti, Remo Sala:
  A Metrological Characterization of the Kinect V2 Time-of-Flight Camera.
  Robotics and Autonomous Systems **75**, 2016.
- Simone Zennaro, Matteo Munaro, Simone Milani, Pietro Zanuttigh, Andrea Bernardi, Emanuele Menegatti:
  Performance Evaluation of the 1st and 2nd Generation Kinect for Multimedia Applications.
  IEEE International Conference on Multimedia and Expo (ICME), 2015.
- Diana Pagliari, Livio Pinto:
  Calibration of Kinect for Xbox One and Comparison between the Two Generations of Microsoft Sensors. Sensors **15**, 2015.
- Hamed Sarbolandi, Damien Lefloch, Andreas Kolb:
  Kinect Range Sensing: Structured-Light versus Time-of-Flight Kinect.
  Journal of Computer Vision and Image Understanding **13**, 2015.

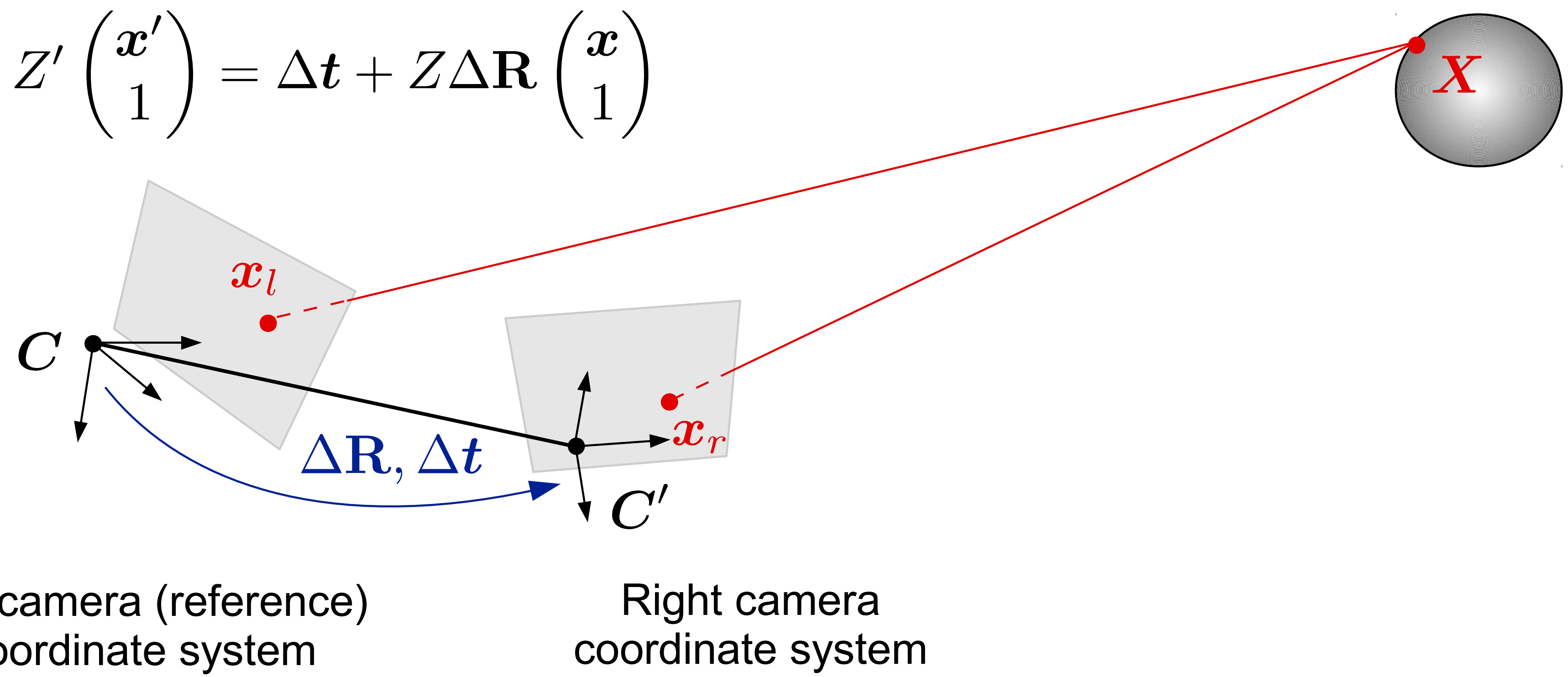
Outline

- Introduction
- Part I: Basics of Mathematical Optimization
  - Linear Least Squares
  - Nonlinear Optimization
- Part II: Basics of Computer Vision
  - Camera Model
  - Multi-Camera Model
  - Multi-Camera Calibration
- Part III: Depth Cameras
  - Passive Stereo
  - Structured Light Cameras
  - Time of Flight Cameras

Depth Cameras: Passive Stereo

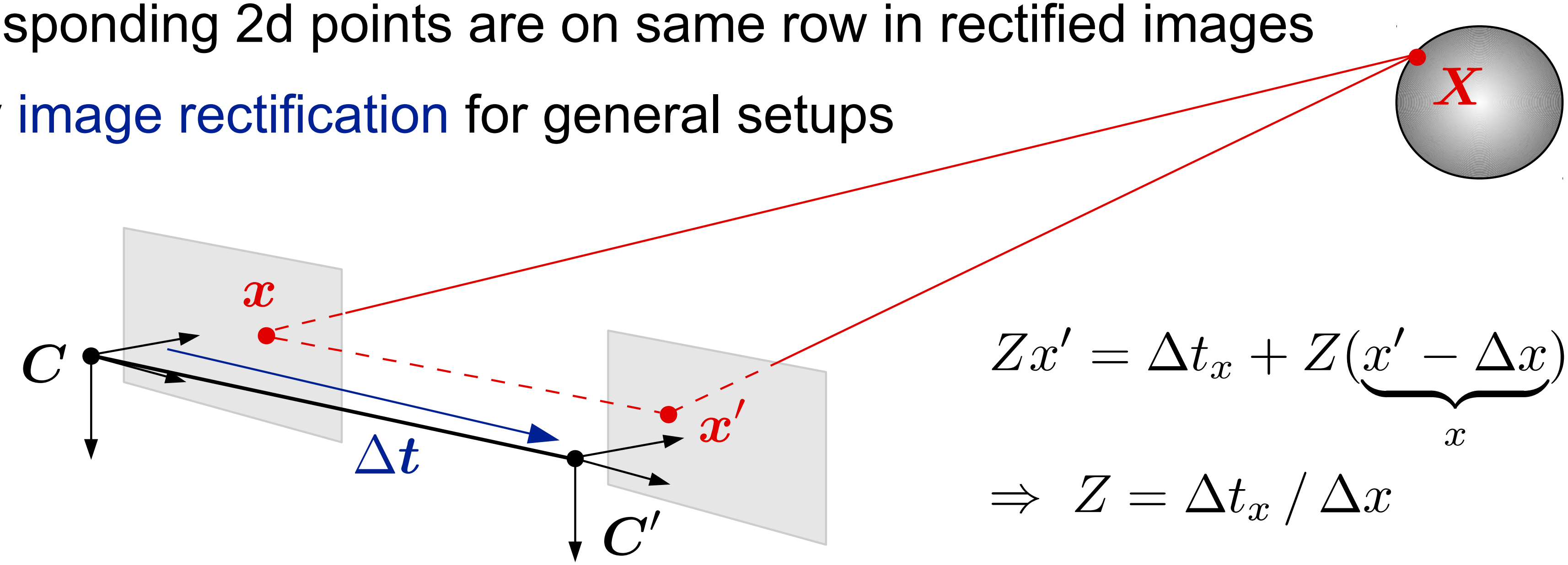
- **Stereo camera:** Consider two rigidly coupled cameras
- Intrinsic parameters: Camera functions K and K'
- Extrinsic parameters: Rotation and translation from left (reference) to right camera coordinate system $(\Delta\mathbf{R}, \Delta\boldsymbol{t})$
- Compute depth via triangulation from 2D/2D correspondences $(\boldsymbol{x}, \boldsymbol{x}')$

$$\boldsymbol{X}' = Z' \begin{pmatrix} \boldsymbol{x}' \\ 1 \end{pmatrix} = \Delta\boldsymbol{t} + Z\Delta\mathbf{R} \begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix}$$



Depth Cameras: Passive Stereo

- **Rectified stereo camera:** Special setup to facilitate stereo vision
  - Camera image planes are coplanar ($\Delta\mathbf{R} = \mathbf{I}$)
  - Camera displacement only along x-axis ($\Delta\boldsymbol{t} = (\Delta t_x, 0, 0)$)
  - Camera have same intrinsic parameters ($K = K'$)
- Corresponding 2d points are on same row in rectified images
- Apply image rectification for general setups



$$Zx' = \Delta t_x + Z(\underbrace{x' - \Delta x}_{x})$$

$$\Rightarrow \quad Z = \Delta t_x \,/\, \Delta x$$

Left rectified camera coordinate system

Right rectified camera coordinate system

Outline

- Introduction
- Part I: Basics of Mathematical Optimization
  - Linear Least Squares
  - Nonlinear Optimization
- Part II: Basics of Computer Vision
  - Camera Model
  - Multi-Camera Model
  - Multi-Camera Calibration
- Part III: Depth Cameras
  - Passive Stereo
  - **Structured Light Cameras**
  - Time of Flight Cameras

Depth Cameras: Structured Light Cameras

Intel RealSense R200 (480 × 360 px, 70° × 46°, 0.5 – 4 m)

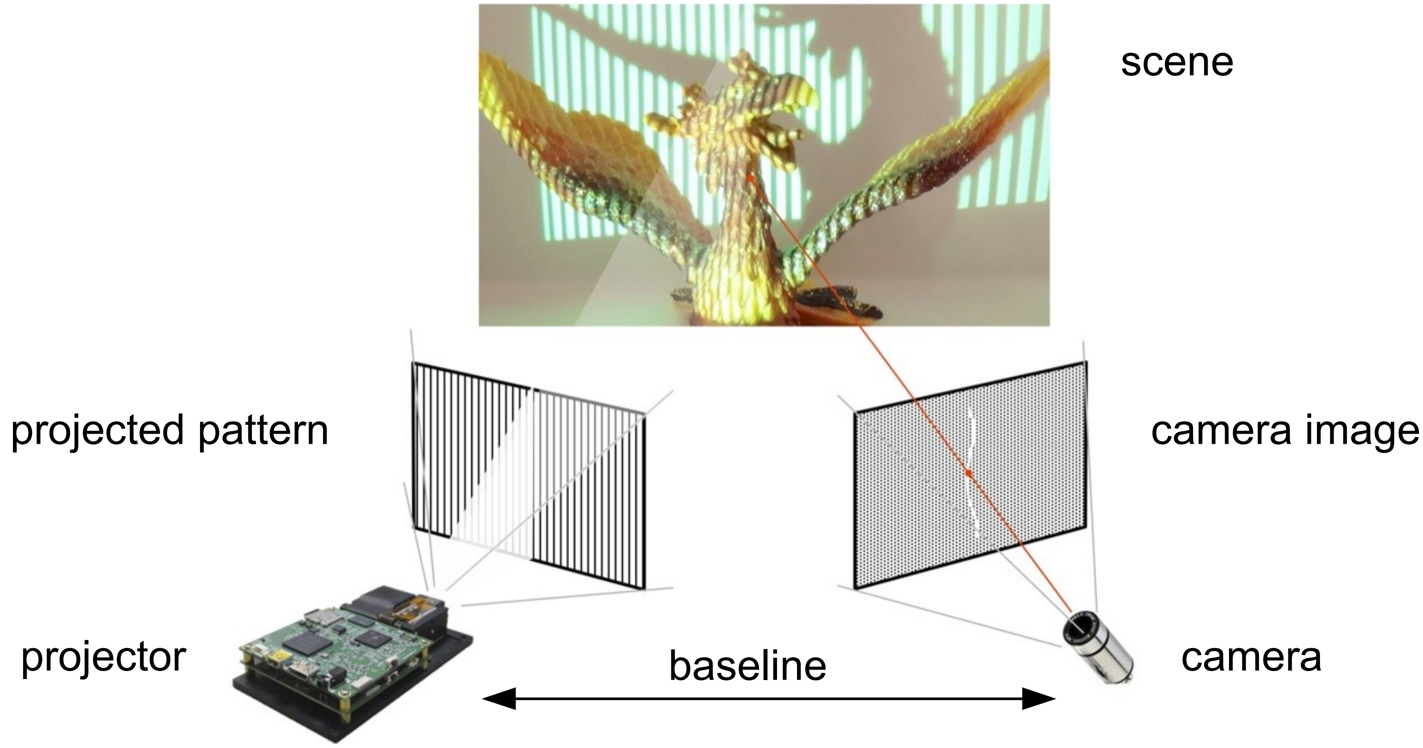
Intel RealSense SR300 (640 × 480 px, 72° × 55°, 0.2 – 2 m)

HP 3D Scanner Pro S3 (6 – 50 cm, up to 0.05 mm accuracy, > 3000 €)

Microsoft Kinect v1 (640 × 480 px, 57° × 43°, 0.4 – 4 m / 8 m)

Depth Cameras: Structured Light Cameras

- **General approach:** Project light pattern into scene, compute depth from distortion of light pattern observed with camera from different position
- Common patterns: stripe pattern, random dot pattern (often infrared light)
- Alignment of camera and projector must be known
- Similar to passive stereo vision, but does not depend on scene texture

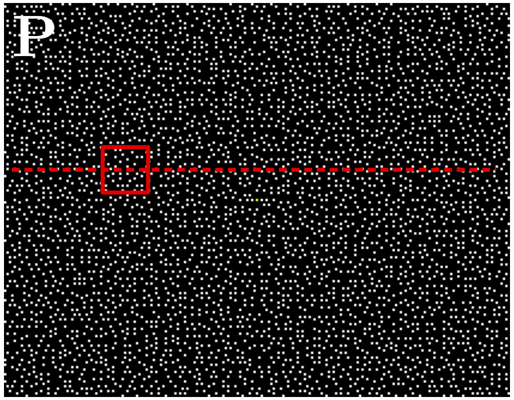


Depth Cameras: Structured Light Cameras

- Camera and projector are in horizontally aligned (rectified stereo)
- Compute depth Z via triangulation from disparity Δx, focal length f_x, and baseline b:

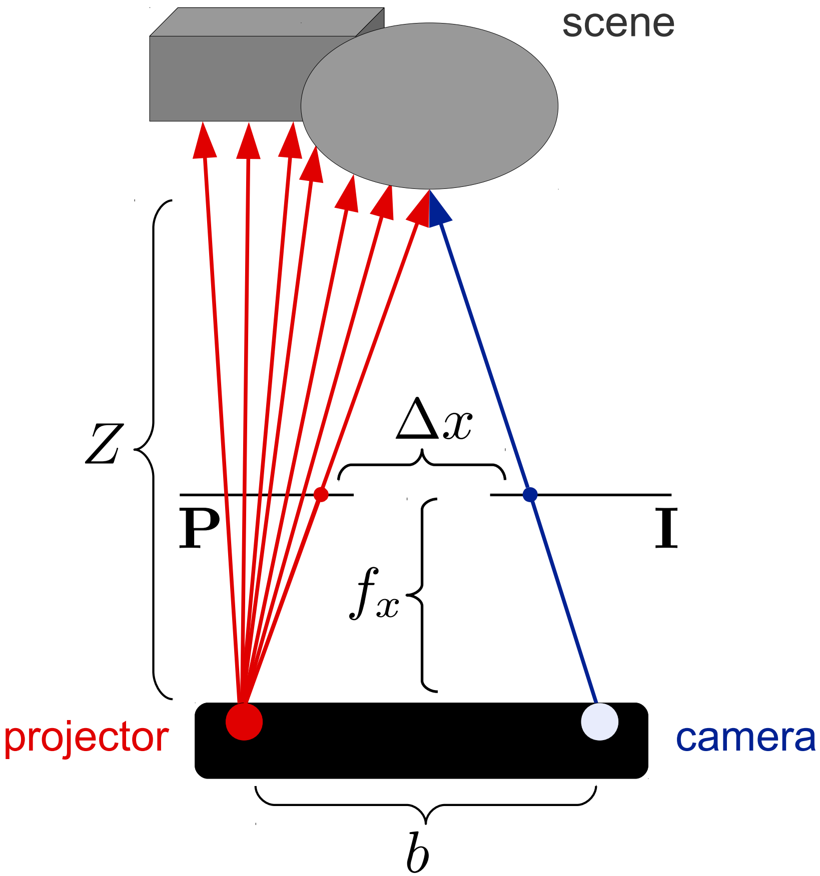
$$Z = \frac{f_x\, b}{\Delta x}$$

- Compute (horizontal) disparity between camera image I and pattern image P for each pixel via correlation window



projected pattern

camera image



Outline

- Introduction
- Part I: Basics of Mathematical Optimization
  - Linear Least Squares
  - Nonlinear Optimization
- Part II: Basics of Computer Vision
  - Camera Model
  - Multi-Camera Model
  - Multi-Camera Calibration
- Part III: Depth Cameras
  - Passive Stereo
  - Structured Light Cameras
  - Time of Flight Cameras

Depth Cameras: Time of Flight Cameras

PMD CamCube 3.0 (200 × 200 px, 40° × 40°, 0.3 – 7 m)

Mesa Swiss Ranger 4000 (176 × 144 px, 43° × 34° / 69° × 56°, 0.8 – 8 m)

Creative Senz3D / DepthSense 525 (320 × 240 px, 74° × 58°, 0.15 – 1 m)

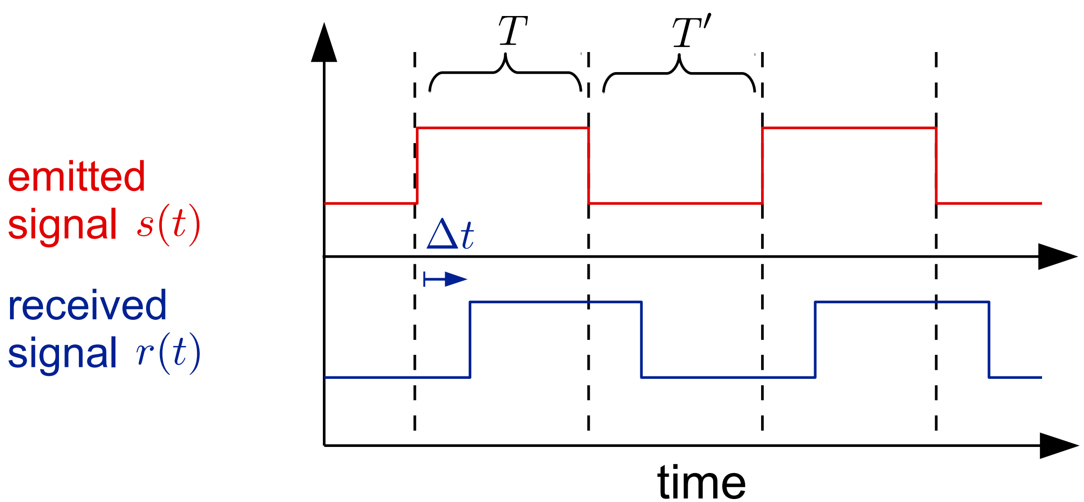
Microsoft Kinect v2 (512 × 424 px, 70° × 60°, 0.8 – 4.5 m / 18.75 m)

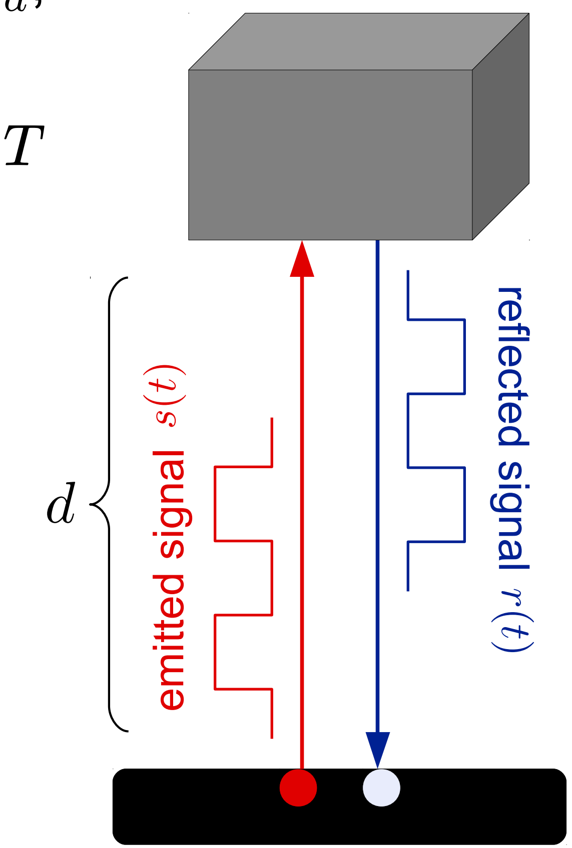
Depth Cameras: Time of Flight Cameras

- **General approach:** Measure absolute time needed by a light pulse to travel from light emitter to object and back to detector (= time of flight)
- Speed of light is $c \approx 300\,000\,000\ \mathrm{m/s}$ (approx. $0.3\ \mathrm{m/ns}$)
- Different ToF measurement approaches:
  - Pulsed modulation
  - Continuous wave modulation
- **Pulsed modulation:** Measure travel time of very short light pulses (*e. g.,* LED, laser), either directly (*e. g.,* with avalanche photodiodes) or indirectly (range-gated imaging)
- **Continuous wave intensity modulation:** Measure phase difference between sent and received signal (*e. g.,* sinusoidal or square wave with known modulation frequency)
- Emitted light typically in near-infrared range (~850 nm wavelength)

Depth Cameras: Time of Flight Cameras

- Compute distance from time of flight: $\Delta t = \frac{2d}{c} \;\Rightarrow\; d = \frac{c}{2}\Delta t$,
  e. g.: $\Delta t = 26.67\,\mathrm{ns} \;\Rightarrow\; d = 4\,\mathrm{m}$
- Temporal measurement resolution of $\delta_t = \frac{2\delta_d}{c}$ needed for distance measurement resolution of δ_d,
  e. g.: $\delta_d = 1\,\mathrm{mm} \;\Rightarrow\; \delta_t = 0.0067\,\mathrm{ns}$
- Maximal distance for periodic pulse with duration T and period $T + T'$ is limited to $d_{\max} = \frac{c}{2}T'$,
  e. g.: $T = T' = 53.33\,\mathrm{ns} \;\Rightarrow\; d_{\max} = 8\,\mathrm{m}$

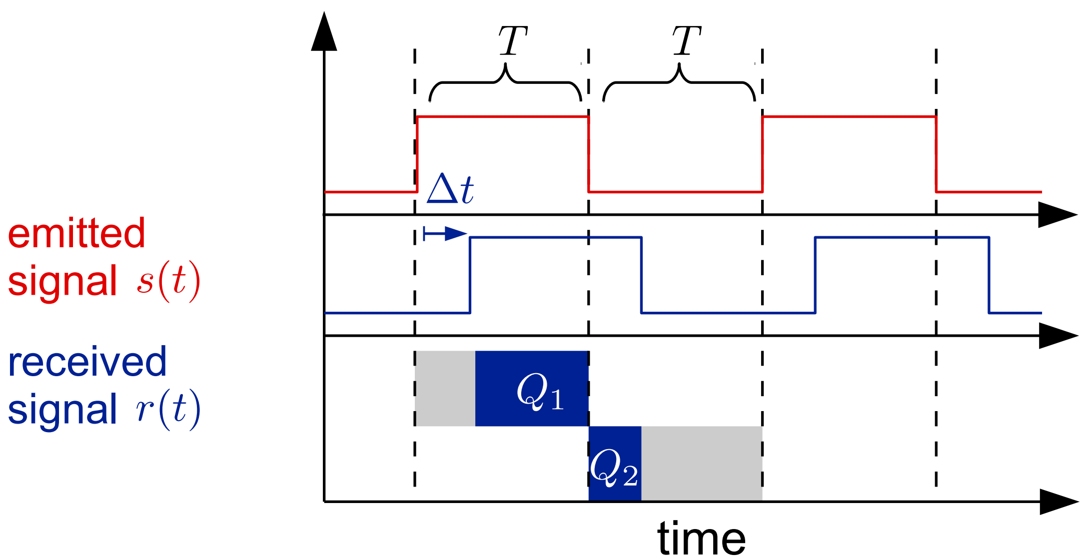


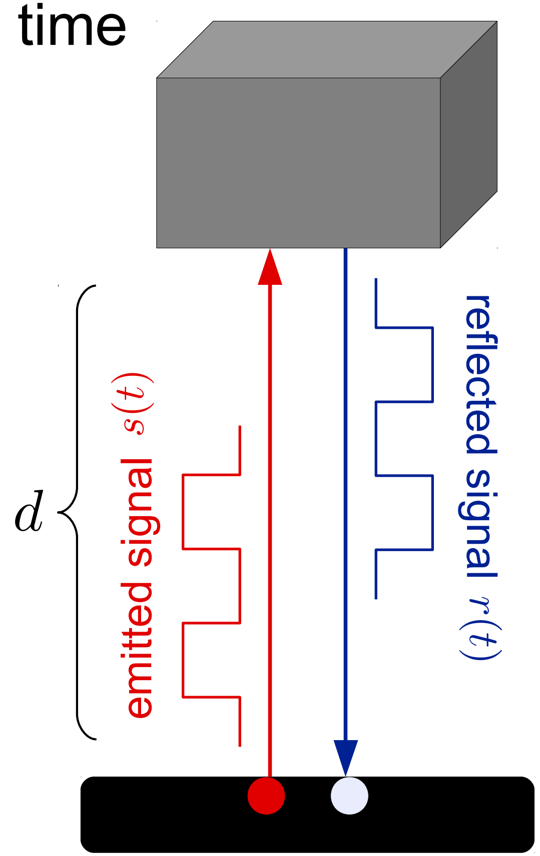


Depth Cameras: Time of Flight Cameras

- Pulsed modulation approach: Illuminate scene for brief duration T
- Compute delay of reflection with fast photo-detector, *e. g.* single-photon avalanche diode (SPAD)
- Alternative: Integrate reflected light received within time intervals $[0, T)$ and $[T, 2T)$ to compute time delay:

$$\Delta t = \frac{Q_2}{Q_1 + Q_2} T$$

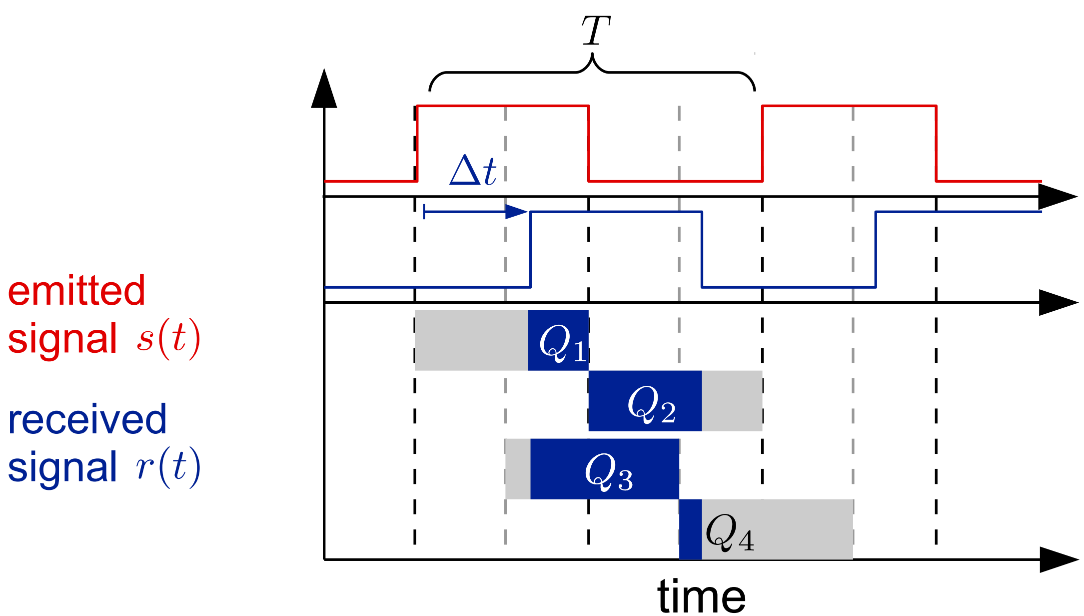


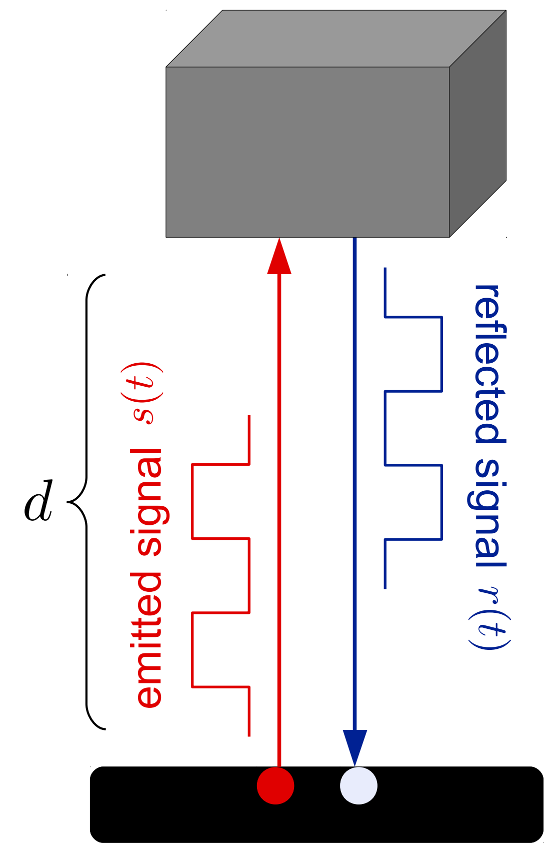


Depth Cameras: Time of Flight Cameras

- Continuous wave intensity modulation approach:
  Illuminate scene with modulated light with frequency $f = \frac{1}{T}$
- Compute distance from phase shift: $\Delta\varphi = 2\pi f \Delta t \Rightarrow d = \frac{c}{4\pi f}\Delta\varphi$
- Compute phase angle shift from 4 time intervals
  $[0, \frac{1}{4}T), [\frac{1}{4}T, \frac{1}{2}T), [\frac{1}{2}T, \frac{3}{4}T), [\frac{3}{4}T, T)$:

  $$\Delta\varphi = \mathrm{atan2}(Q_3 - Q_4, Q_1 - Q_2)$$



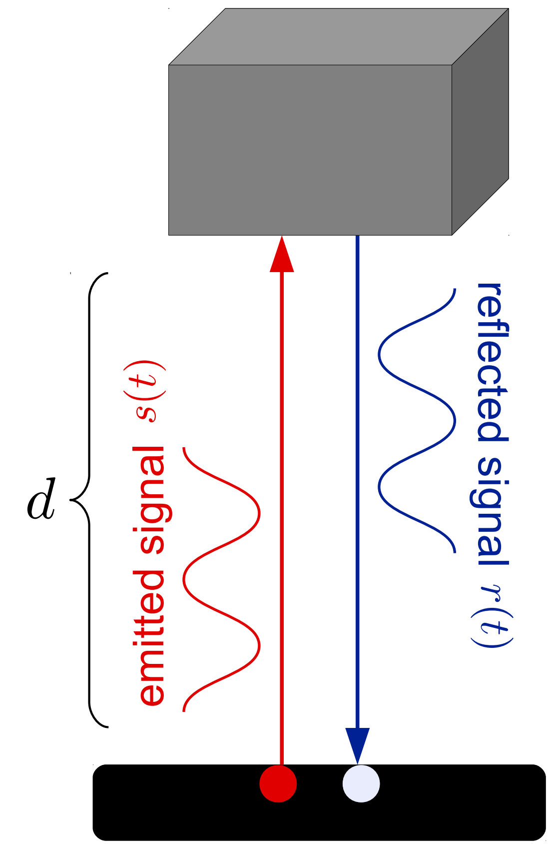


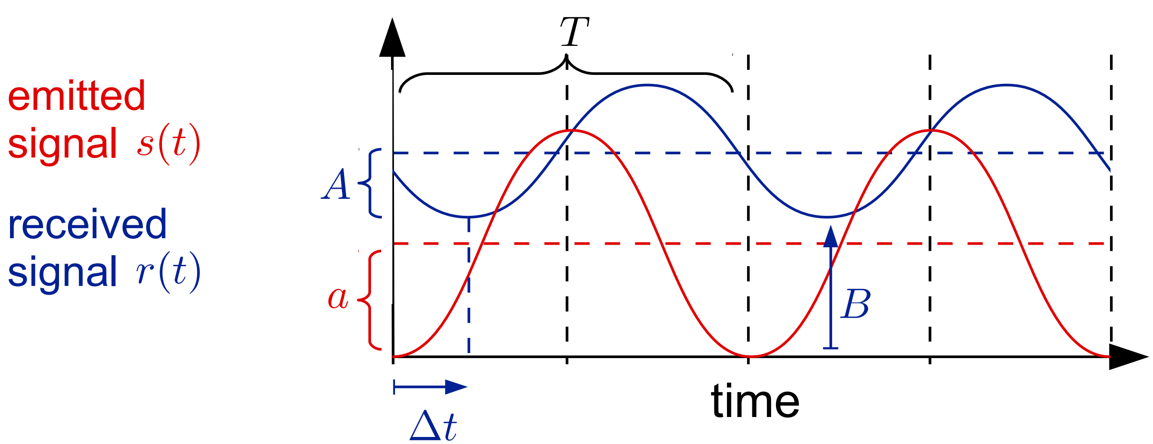
Depth Cameras: Time of Flight Cameras

- Emit sinusoidal signal with amplitude a and modulation frequency f:
  $s(t) = a \cdot (1 + \sin(2\pi f t))$
- Receive reflected signal with delay Δt, attenuated amplitude A, and intensity offset B (*e. g,* due to ambient light):

$$\begin{aligned} r(t) &= A \cdot (1 + \sin(2\pi f (t - \Delta t))) + B \\ &= A \cdot (1 + \sin(2\pi f t - \Delta\varphi)) + B \end{aligned}$$

- Compute $\Delta\varphi$, A, and B from samples of $r(t)$





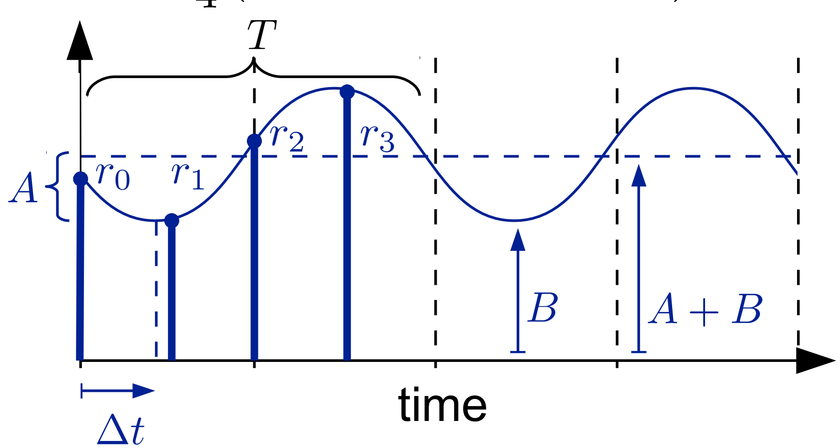
Depth Cameras: Time of Flight Cameras

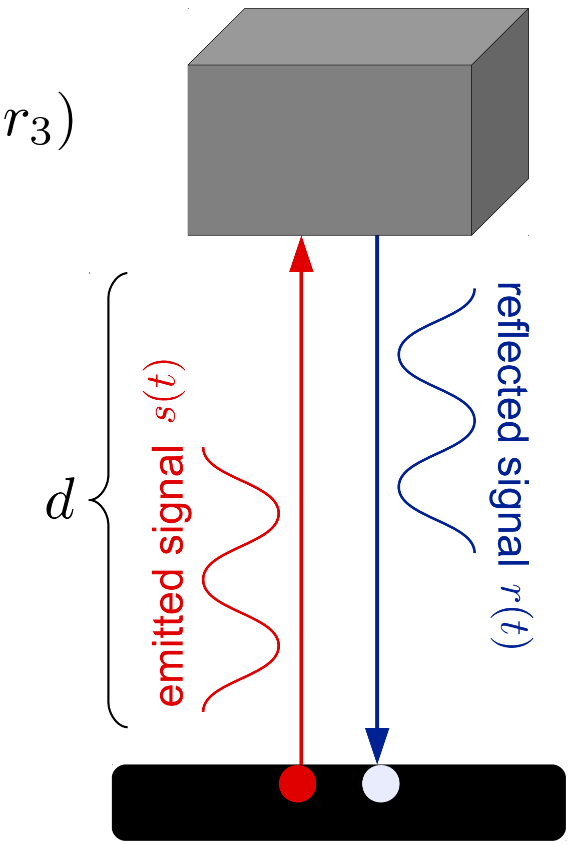
- Estimate amplitude A, offset B, and phase angle shift difference $\Delta\varphi$ of $r(t)$ from 4 samples at: $t_0 = 0,\ t_1 = \frac{1}{4}T,\ t_2 = \frac{1}{2}T,\ t_3 = \frac{3}{4}T$

$$\Rightarrow\ r_i = r(t_i) = A \cdot \sin(\underbrace{2\pi f(t_i - \Delta t)}_{\frac{i\pi}{2} - \Delta\varphi}) + (A + B),\ i = 0, \ldots, 3$$

  - Phase shift: $\Delta\varphi = 2\pi f \Delta t = \operatorname{atan2}(r_2 - r_0, r_1 - r_3)$
    $\rightarrow$ distance: $d = \frac{c}{4\pi f}\Delta\varphi$
  - Amplitude: $A = \frac{1}{2}\sqrt{(r_0 - r_2)^2 + (r_1 - r_3)^2}$
  - Offset: $B = \frac{1}{4}(r_0 + r_1 + r_2 + r_3) - A$

received signal $r(t)$





Depth Cameras: Time of Flight Cameras

- **Exercise:** Compute parameters of function $s(t) = a \cdot \sin(2\pi f t + \varphi) + b$ from 4 samples $s_i = s(t_i),\ t_i = \frac{i}{4} \cdot T,\ i = 0, \ldots, 3$ with $T = \frac{1}{f}$

$$s_0 = s(t_0) = a \cdot \sin(\varphi) + b$$

$$s_1 = s(t_1) = a \cdot \sin(\varphi + \tfrac{\pi}{2}) + b = a \cdot \cos(\varphi) + b$$

$$s_2 = s(t_2) = a \cdot \sin(\varphi + \pi) + b = -a \cdot \sin(\varphi) + b$$

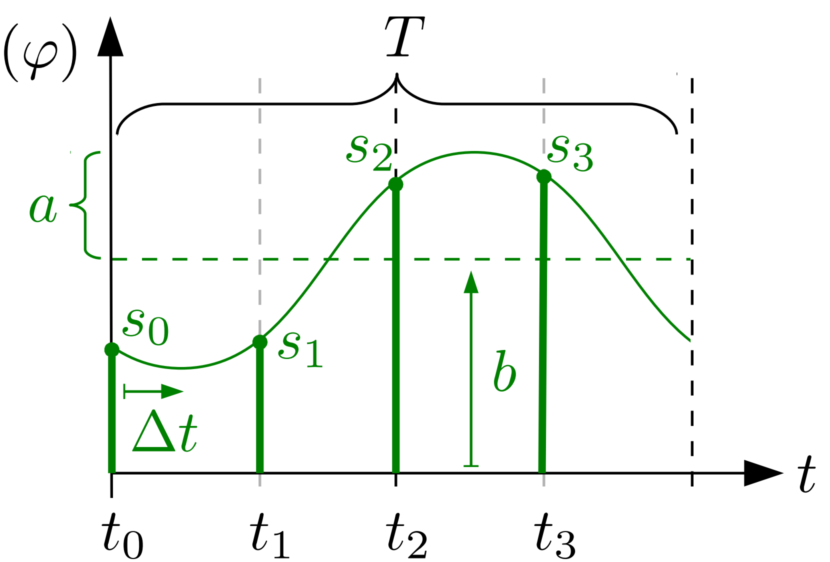
$$s_3 = s(t_3) = a \cdot \sin(\varphi + \tfrac{3\pi}{2}) + b = -a \cdot \cos(\varphi) + b$$

$$\Rightarrow\ s_0 - s_2 = 2a \cdot \sin(\varphi),\ s_1 - s_3 = 2a \cdot \cos(\varphi)$$

$$\Rightarrow\ (s_0 - s_2)^2 + (s_1 - s_3)^2 =$$
$$4a^2 \cdot \left(\sin(\varphi)^2 + \cos(\varphi)^2\right) = 4a^2$$

$$\Rightarrow\ \frac{s_0 - s_2}{s_1 - s_3} = \frac{\sin(\varphi)}{\cos(\varphi)} = \tan(\varphi)$$

$$\Rightarrow\ s_0 + s_1 + s_2 + s_3 = 4b$$



Depth Cameras: Time of Flight Cameras

- Approximation of distance measurement error from photon shot-noise as Gaussian noise with

$$\sigma_d = \frac{c}{4\sqrt{2}\pi f} \cdot \frac{\sqrt{A+B}}{c_d \cdot A}$$

  where modulation contrast c_d describes how well the ToF sensor separates and collects the photoelectrons
- Negative influence: High amount of ambient light (intensity offset B)
- Positive influence: High signal amplitude A, high modulation frequency f, high modulation contrast c_d
- But: Modulation contrast will attenuate at high modulation frequencies due to physical limitations
- Maximal distance is $d_{\max} = \frac{c}{2f}$ due to phase wrapping at $\Delta\varphi = 2\pi$,
  e. g.: $d_{\max} = 8\,\mathrm{m} \;\Rightarrow\; f = 18.75\,\mathrm{MHz}$

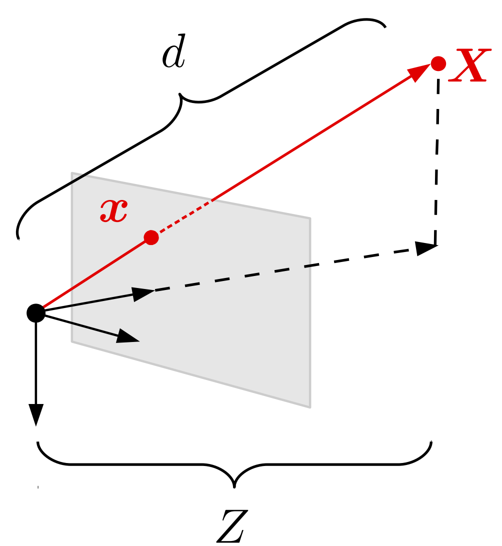
Depth Cameras: Error Sources

- **General error sources** for Structured Light and ToF cameras:
  - Inaccurate intrinsic camera calibration
  - Ambient background light
  - Multi-device interference
  - Temperature drift
  - Systematic distance error (*e. g.,* due to quantization)
  - Errors at depth discontinuities
    - Structured Light: Invalid pixels due to occlusion
    - ToF: Mixed measurement from light reflected partly by objects in foreground and background (“flying pixels”)

see also: Sarbolandi, Lefloch & Kolb: *Kinect Range Sensing: Structured-Light versus Time-of-Flight Kinect* (section 2.3), 2015.

Depth Cameras: 3D Structure Computation

- **Output of depth cameras:** Disparity Δx (for structured light cameras), depth Z or distance d (ToF cameras) for each pixel (u, v)
  - From disparity to depth: $Z(u,v) = f_x\, b \,/\, \Delta x(u,v)$
- **Compute 3D points** using pinhole camera model for IR camera
  - Camera parameters are (f_x, f_y, c_x, c_y)
  - Mapping from pixel to normalized image coordinates:
    $$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} (u - c_x) \,/\, f_x \\ (v - c_y) \,/\, f_y \end{pmatrix}$$
  - Mapping from depth to 3D point:
    $$\boldsymbol{X} = Z(u,v) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$
  - Mapping from distance to 3D point:
    $$\boldsymbol{X} = \frac{d(u,v)}{\sqrt{x^2+y^2+1}} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$
  - Take also distortion into account!



Depth Cameras: 3D Structure Computation

- **Further issues:**
  - Compute dense 3D structure / surface from sparse 3D points
  - Fusion of depth maps from multiple range sensors
  - Combine image-based 3D reconstruction with depth maps
- **Next topic:**
  - Fusion of multi-view color information and depth maps for 3D-TV