
# Depth Cameras: A State-of-the-Art Review

Carlo Dal Mutto

May 23, 2018

aQUIFI

# Aquifi

## Aquifi Constellation™

## Aquifi Discovery™

## Aquifi Endeavour™ Services

**3D Reconstruction**

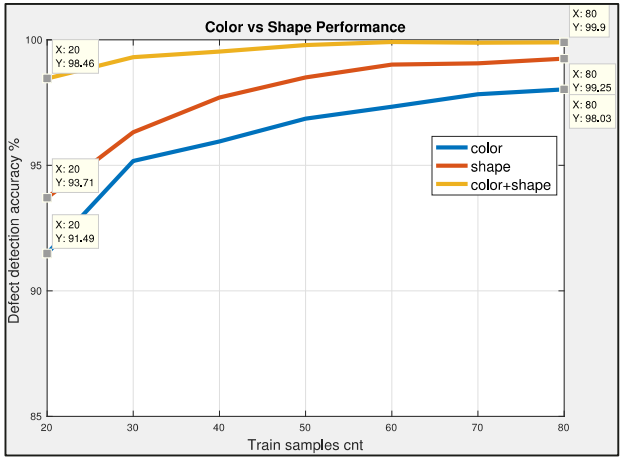
**3D + Color**

**3D Processing and AI**

**Object Sizing**

**Object ID**

**Object inspection**



# Agenda

- Intro to depth cameras
- Technology fundamentals
- Triangulation-based depth cameras
  - Stereo vision systems
  - Active stereo systems
  - Structured-light cameras
- Time-of-flight cameras
- Technology review and comparison

# Intro to Depth Cameras

# What is a depth camera?

- A depth camera acquire images that encode depth (z) at each pixel

- Important concepts:
  - Spatial resolution: # of pixels
  - z-resolution: quantum of z measurements
  - z-range: [minZ, maxZ]
  - SNR
  - Precision (repeatability)
  - Accuracy (offset)

# Which applications?

- AR/VR
- Contactless authentication
- Computational imaging (*e.g.*, Bokeh effect)
- Robotics
- Video-conferencing
- Gaming


# Which are the core technologies?

- Time-of-Flight cameras
- Structured-light cameras
- Active stereo cameras
- Passive stereo cameras


Note: this is not a complete list of technologies, but it is relevant for the scope of this talk.

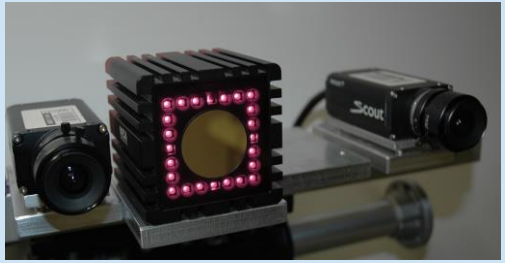
# Some players

- Microsoft
- Intel
- Apple
- PMD
- Occipital
- ZED
- Orbbec
- (Primesense)
- …




Note: this is not a complete list of players, and some might not be active anymore.

# My experience with depth cameras



ToF and Stereo Data Fusion

ToF

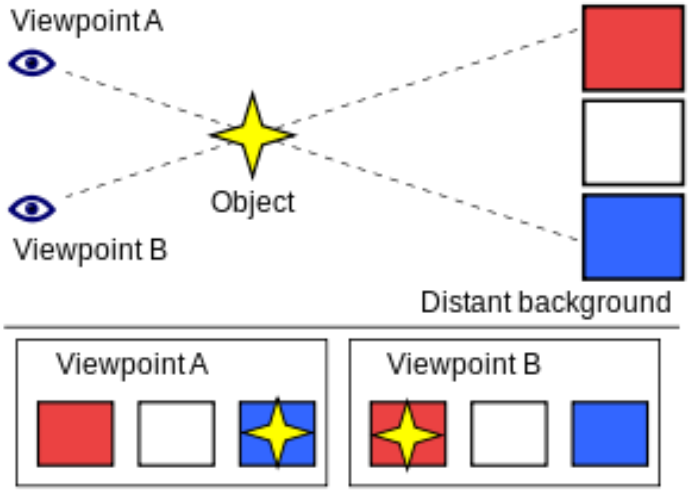
Passive stereo

Active stereo

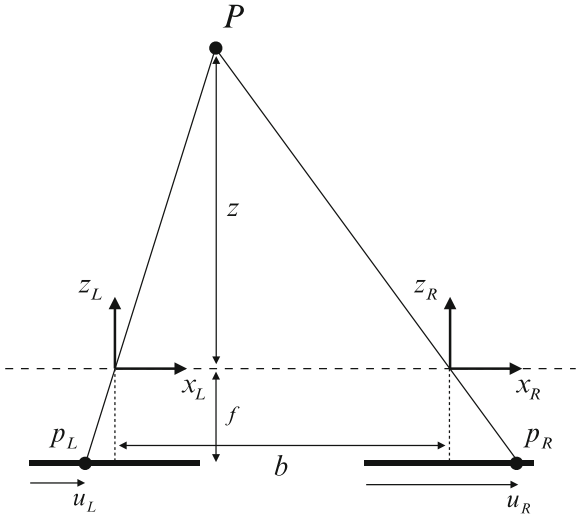
# Technology Fundamentals

# Physics101: measuring distance by triangulation

Parallax




Triangulation



# Physics101: measuring distance with time-of-flight

- Indirect distance measurement: time-of-flight

$$distance = speed \times time$$

$$speed = c = 3 * 10^{8}[m/s]$$

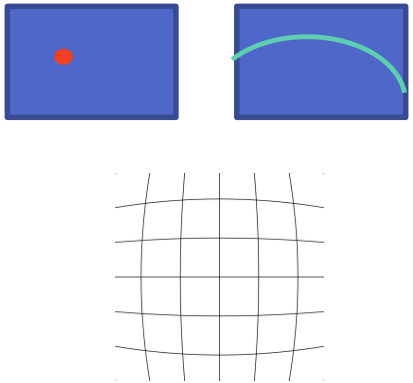
- Now it is just a matter of measuring time $10^{-9}[s]$

# Triangulation-based depth cameras

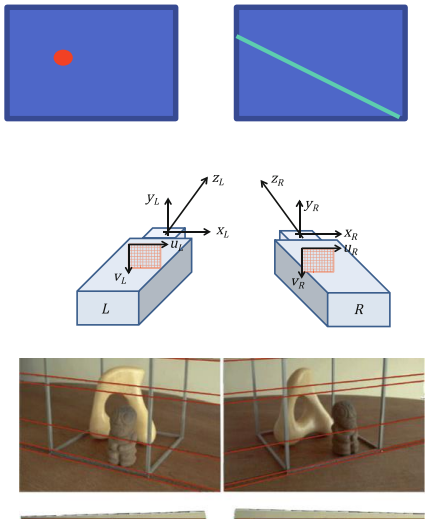
## Stereo vision systems

# Basics of epipolar geometry

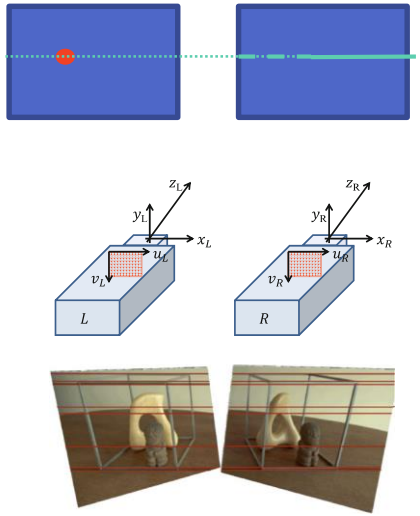
- Finding correspondences (conjugates) in stereo images



No un-distortion
No rectification

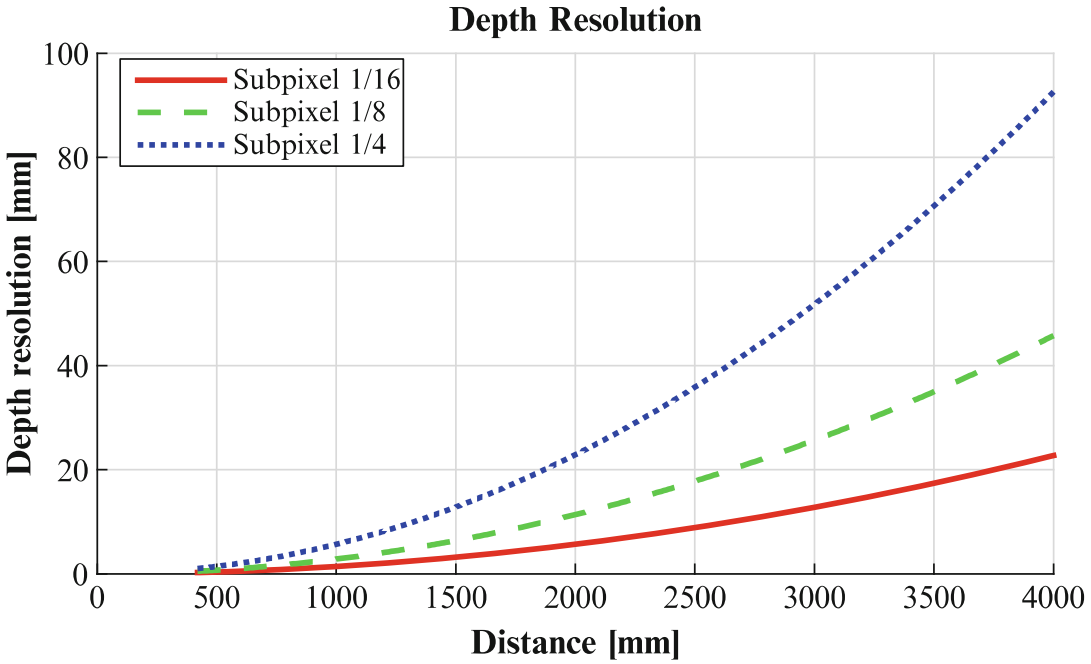



Un-distortion
No rectification




Rectification
(un-distortion)

# Depth resolution




$$\Delta Z = \frac{Z^2}{bf}\Delta d$$

$\Delta Z$ : depth resolution

$b$ : baseline

$f$ : focal

$\Delta d$ : disparity resolution

# Stereo matching

- Finding conjugates on the epipolar lines

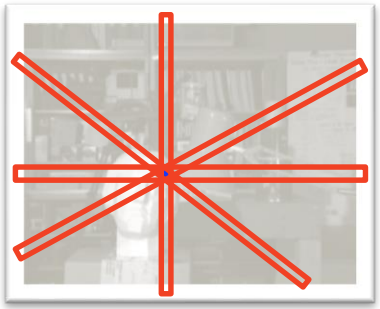

- Matching cost (score): dissimilarity (similarity) within the red window (aggregation support)

# Stereo matching

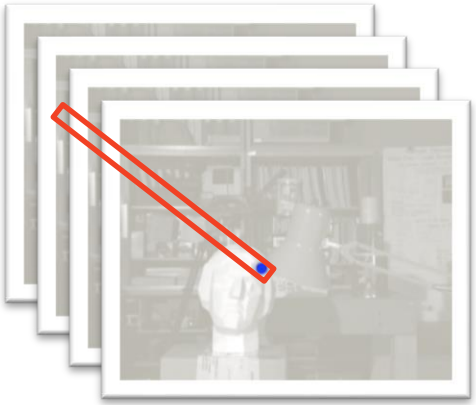
- Type of windows

No aggregation

Local spatial aggregation



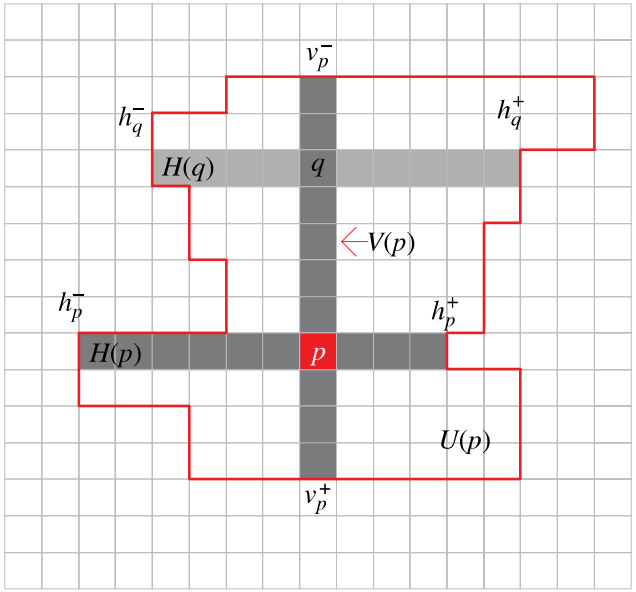
Semi-global spatial aggregation



Temporal aggregation (space-time)

# Stereo matching

- Engineering the **aggregation window**
  (*Zhang, Lu, Lafruit, "Cross-based local stereo matching using orthogonal integral images"*)




1) **Local cross**:
   Length of the arms determined by coarse color similarity after smoothing.

2) **Shape-adaptive support region**:
   integrate multiple horizontal lines

*Figure from: Zhang, Lu, Lafruit, "Cross-based local stereo matching using orthogonal integral images"*

# Stereo matching

- Classical matching costs:
  - Sum of absolute differences (**SAD**)

$$SAD(u,v,d) = \sum_{(k,l)} |I_1(u+k, v+l) - I_2(u+k+d, v+l)|$$

  - Sum of squared differences (**SSD**)

$$SSD(u,v,d) = \sum_{(k,l)} (I_1(u+k, v+l) - I_2(u+k+d, v+l))^2$$

- Classical matching costs (continued):
  - Normalized cross-correlation

$$NCC(u,v,d) = \sum_{(k,l)} \frac{I_1(u+k,v+l)I_2(u+k+d,v+l)}{\sqrt{I_1(u+k,v+l)^2}\sqrt{I_2(u+k+d,v+l)^2}}$$

  - Census transform

| | | |
|---|---|---|
| 89 | 63 | 72 |
| 67 | 55 | 64 |
| 58 | 51 | 49 |

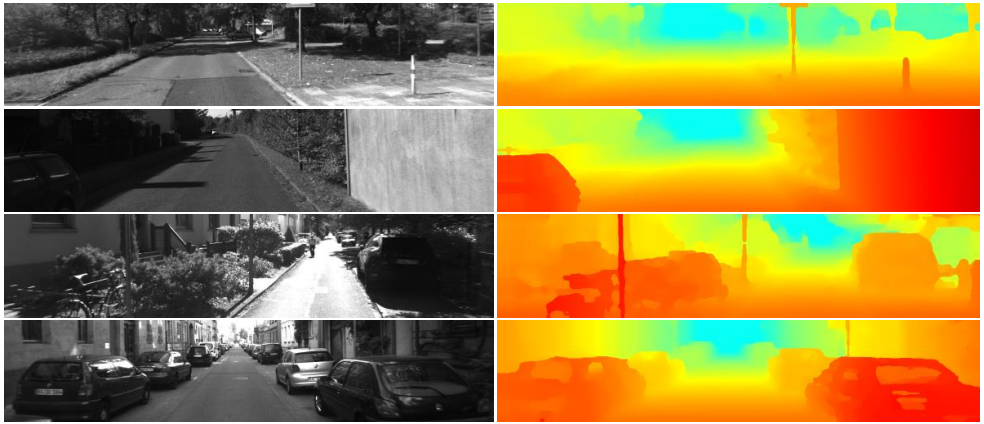
$\mathcal{C}$ ⇨ 00000011

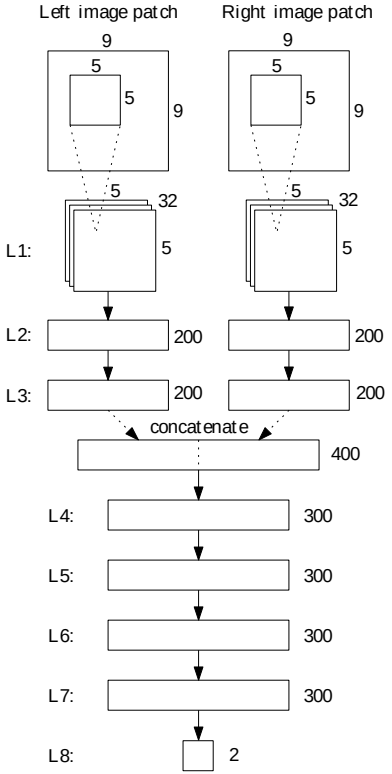
$$\xi(I,p,p\prime) = \begin{cases} 1 & \text{if } I(p) < I(p\prime) \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{C}[I(p)] = \bigodot_{p' \in S(P,\beta)} \xi(I,p,p')$$

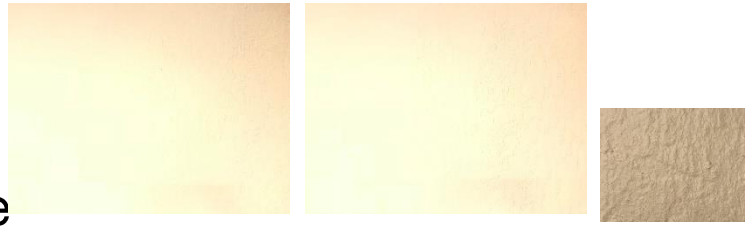
# Stereo matching

- Matching cost function (deep-learning)
- Zbontar, LeCun, “Computing the Stereo Matching Cost with a Convolutional Neural Network,” 2015





# Stereo matching



- Issue: aperture problem
  - Matching on a white wall?
  - …even though sometimes is not white
- Other issues: perspective and photometric distortions…

- Stereo uniqueness
  - For a given point
  - Ratio between lowest cost and the second lowest cost
  - Min across entire image

# Stereo cameras

- Stereo cameras in the market
  - Special effects on phones
  - Outdoor applications (drones)
- Measurements
  - Spatial resolution: very high
  - Z-resolution: high, but scales with distance
  - Z-range: good (no emitted light)
  - SNR: scene dependent, poor indoor
  - Power consumption: very high, dominated by **computations**

*e.g.*, Iphone world facing camera

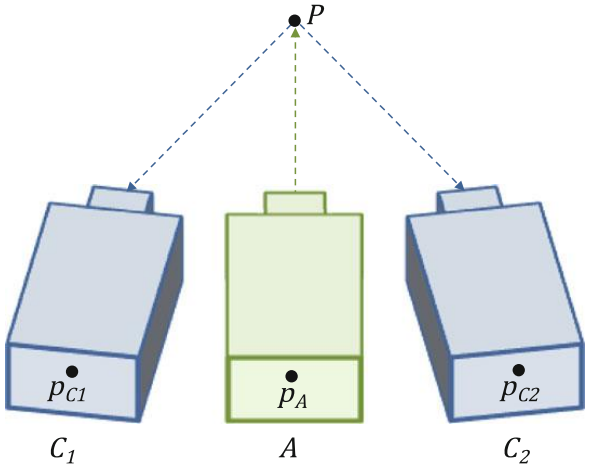

*e.g.*, ZED stereo camera

Note: the systems showed in the figures are only for illustration purposes and are not to be considered “recommended” systems w.r.t. competitors.

# Active stereo systems

# Active stereo

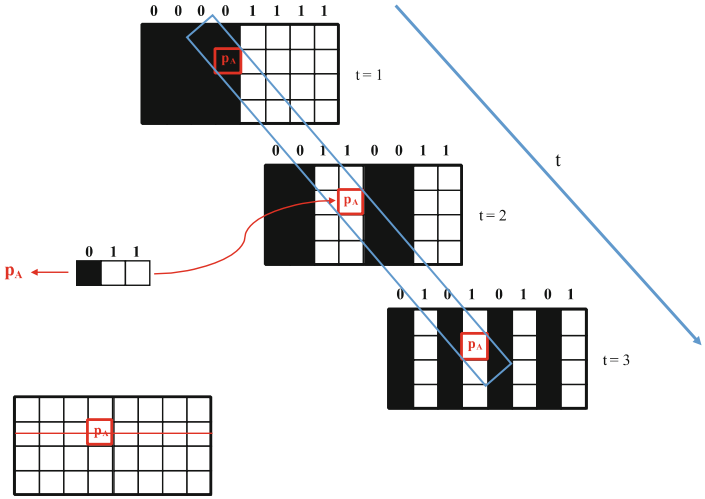
- Use a projector to solve the aperture problem

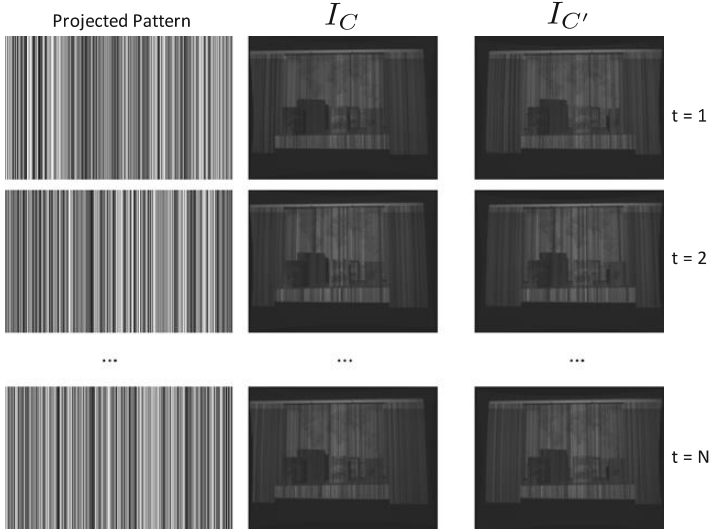



- Which type of pattern? Time multiplexing, space multiplexing

# Time-multiplexing (1)

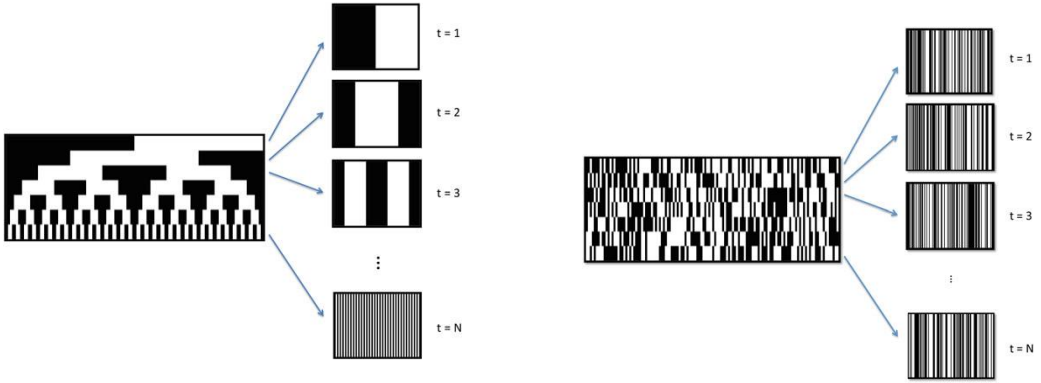
- This technique is often called space-time stereo



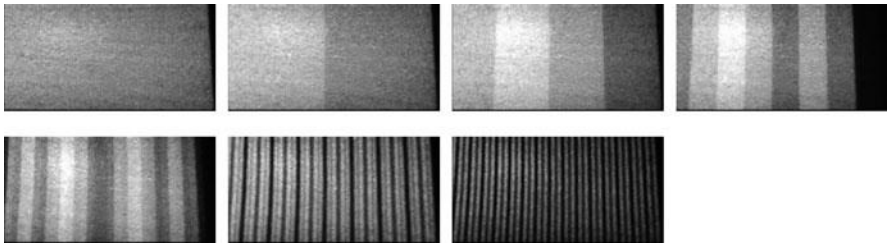



# Time-multiplexing (2)

- Pattern: Gray code




- Intel S200



# Time multiplexing (3)

- Data quality
- Multiplexing time→ temporal sampling → ghosting

# Time multiplexing cameras

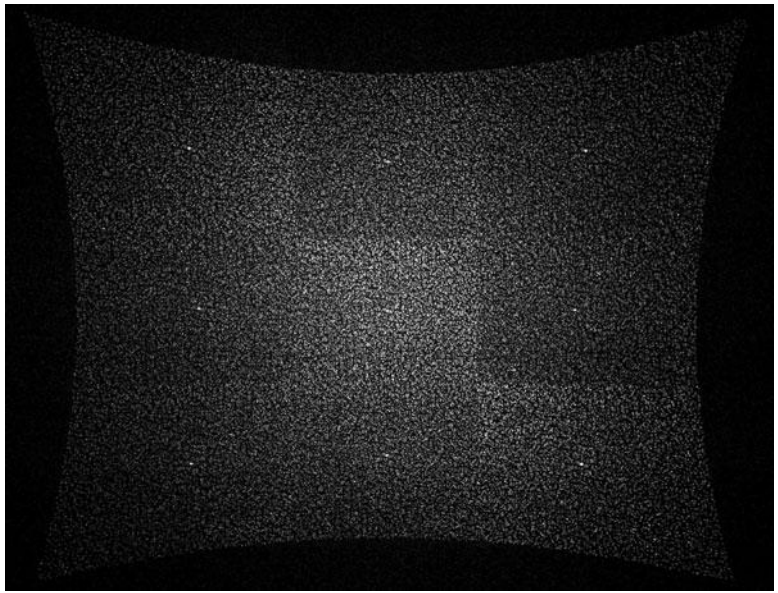
- TMUX cameras in the market
  - Gesture interaction
  - Indoor applications (the pattern is diffused)
- Measurements
  - Spatial resolution: high
  - Z-resolution: reasonable, but scales with distance
  - Z-range: poor
  - SNR: reduced temporal resolution due to temporal sampling
  - Power consumption: dominated by illumination and computations

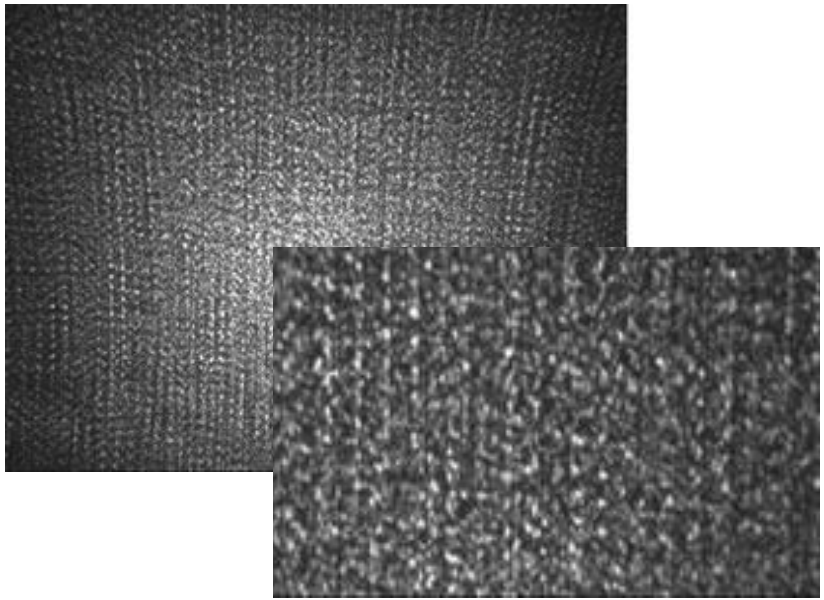

*e.g.*, Intel Realsense F200

Note: the systems showed in the figures are only for illustration purposes and are not to be considered “recommended” systems w.r.t. competitors.

# Space multiplexing (1)

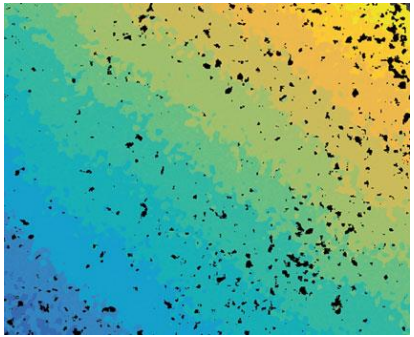
- Help stereo uniqueness by projecting a pattern



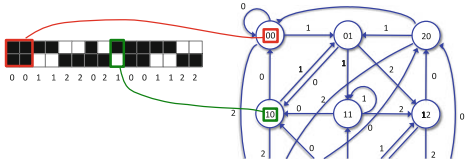


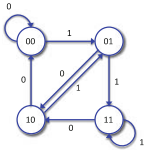
# Space multiplexing (2)

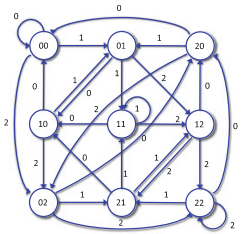
- Type of patterns
  - Collimated vs un-collimated

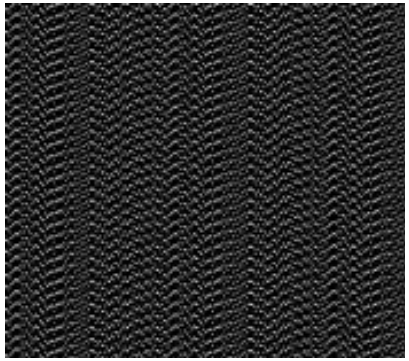


  - De-Brujin pattern



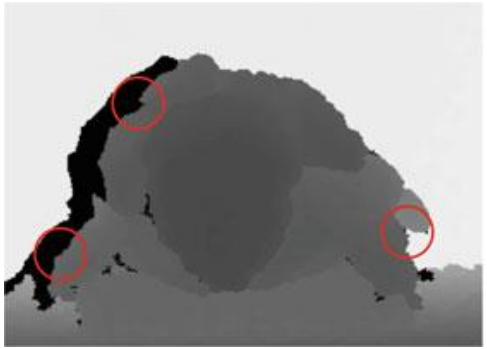



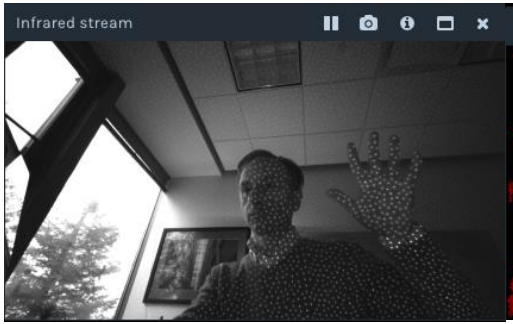



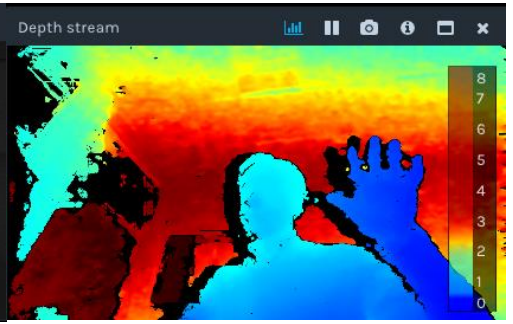



# Space multiplexing (3)

- Data quality
- Multiplexing range → x-y sampling





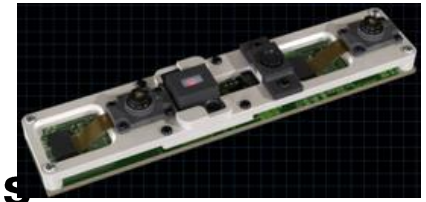
# Space multiplexing cameras

- SMUX cameras in the market
  - Indoor/outdoor applications
- Measurements
  - Spatial resolution: lower than sensor (block artifacts)
  - Z-resolution: ok, but scales with distance
  - Z-range: good
  - SNR: trade-off range/spatial resolution (energy is split among dots)
  - Power consumption: dominated by **computations**

*e.g.*, Intel Realsense R200

*e.g.*, Intel Realsense D435

*e.g.*, Intel Realsense D415



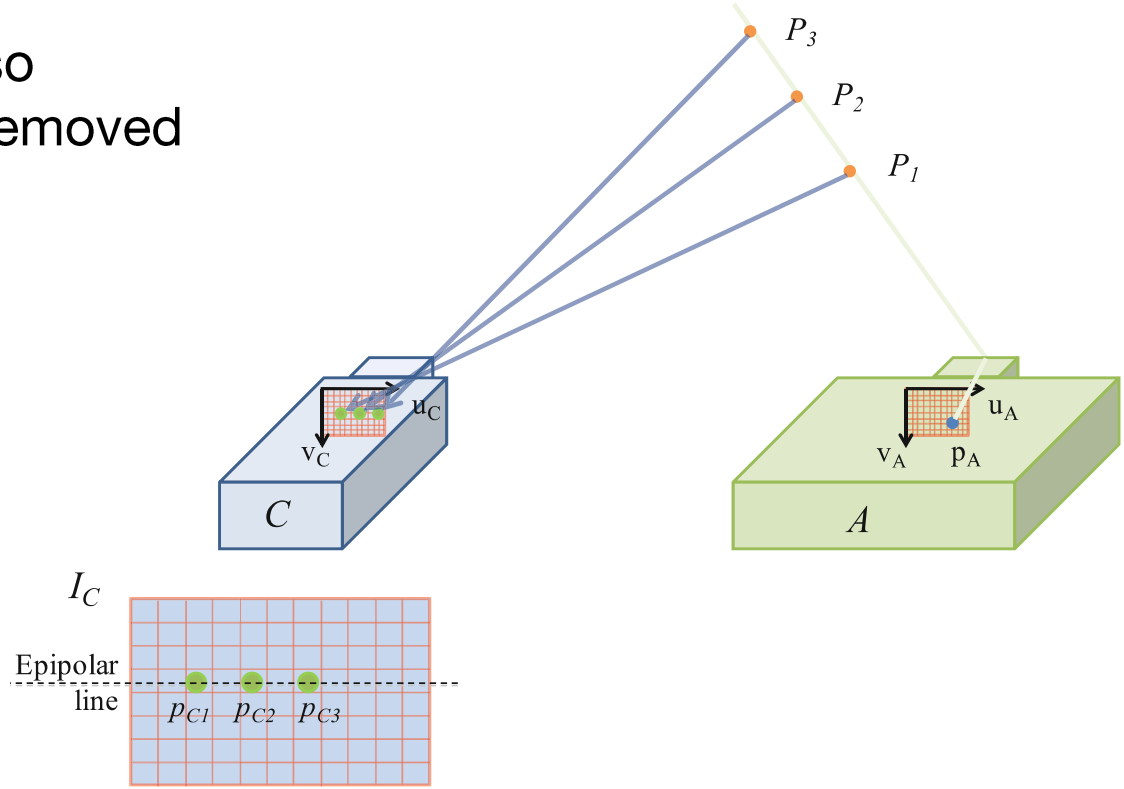
*e.g.*, Occipital Structure Core

Note: the systems showed in the figures are only for illustration purposes and are not to be considered “recommended” systems w.r.t. competitor

## Structured-light cameras

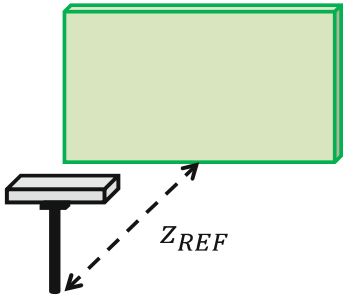
# Camera virtualization

- Epipolar geometry also works if a camera is removed

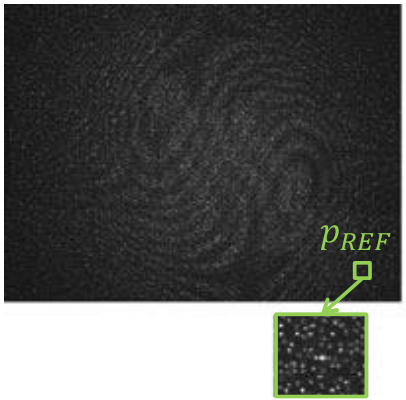


# Reference image trick

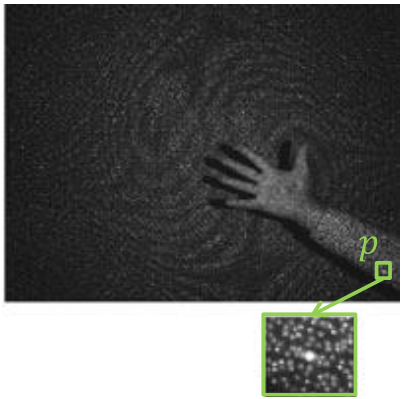
CALIBRATION SETUP




REFERENCE IMAGE




ACQUIRED IMAGE




$$d = d_{REF} + d_{REL}$$

$$\boldsymbol{p}_{REF} = \begin{bmatrix} u_{REF} \\ v_{REF} \end{bmatrix}$$

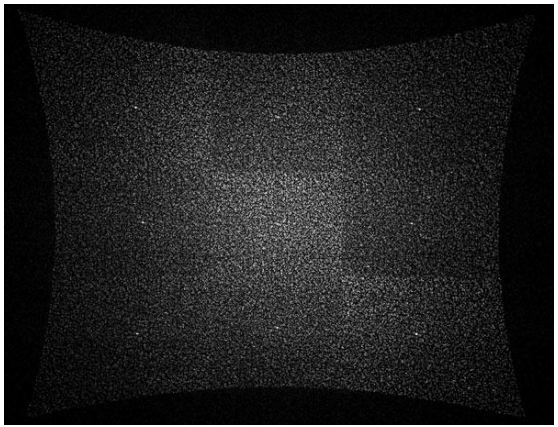
$$d_{REF} = \frac{bf}{z_{REF}}$$

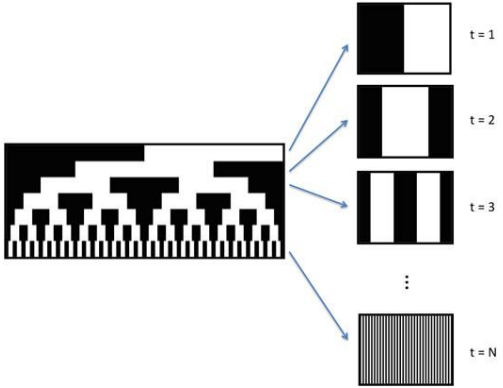
$$\boldsymbol{p} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_{REF} + d_{REL} \\ v_{REF} \end{bmatrix}$$

$$d_{REL} = u - u_{REF}$$

# Typical projected patterns

- Same patterns as active stereo systems

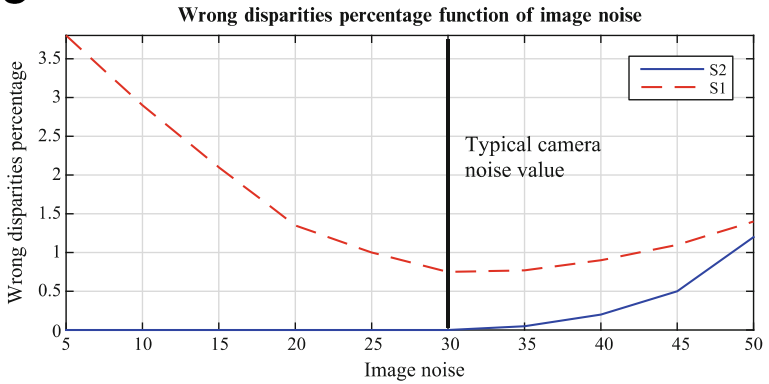


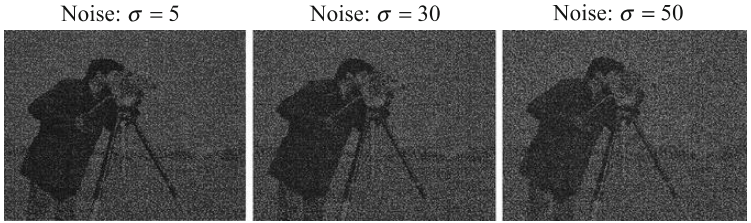


# Single-cameras issues

- Lower SNR: scene intrinsic texture becomes noise.

- Experiment:
  - Pick a reference image
  - Simulate pattern projection
  - Add noise to one image
  - Simulate one and two camera systems
  - Compute disparity errors as function of noise



# Structured–light cameras

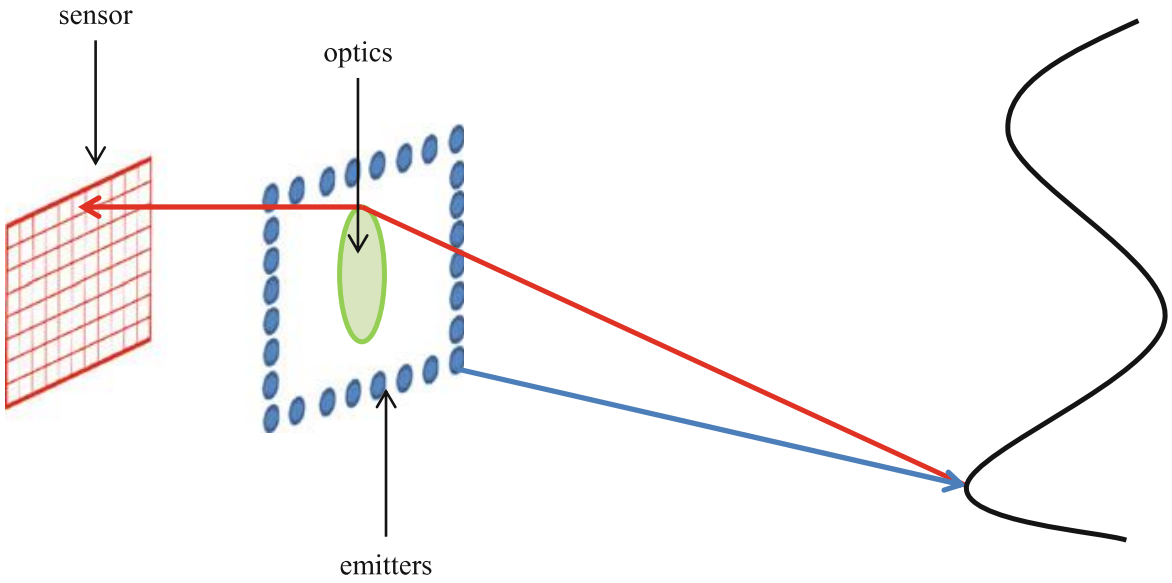
- Structured light cameras in the market
  - Interaction (touchless ID, gaming)
  - Indoor applications or shorter range
  - Smaller footprint (only one camera)
- Measurements
  - Spatial resolution: lower than sensor (block artifacts)
  - Z-resolution: ok, but scales with distance
  - Z-range: good
  - SNR: trade-off range/spatial resolution
    (energy is split among dots, pattern has to be seen)
  - Power consumption: dominated by illumination and **computations**


*e.g.*, Iphone X FaceID camera
Microsoft Kinect v1
Occipital structure sensor
Orbec3D sensor family

Note: the systems showed in the figures are only for illustration purposes and are not to be considered “recommended” systems w.r.t. competitor

## Time-of-Flight depth cameras

# ToF camera system architecture




**Blue circles**: emitters

**Blue arrow**: propagation toward the scene

**Red arrow**: back-propagation toward the sensor

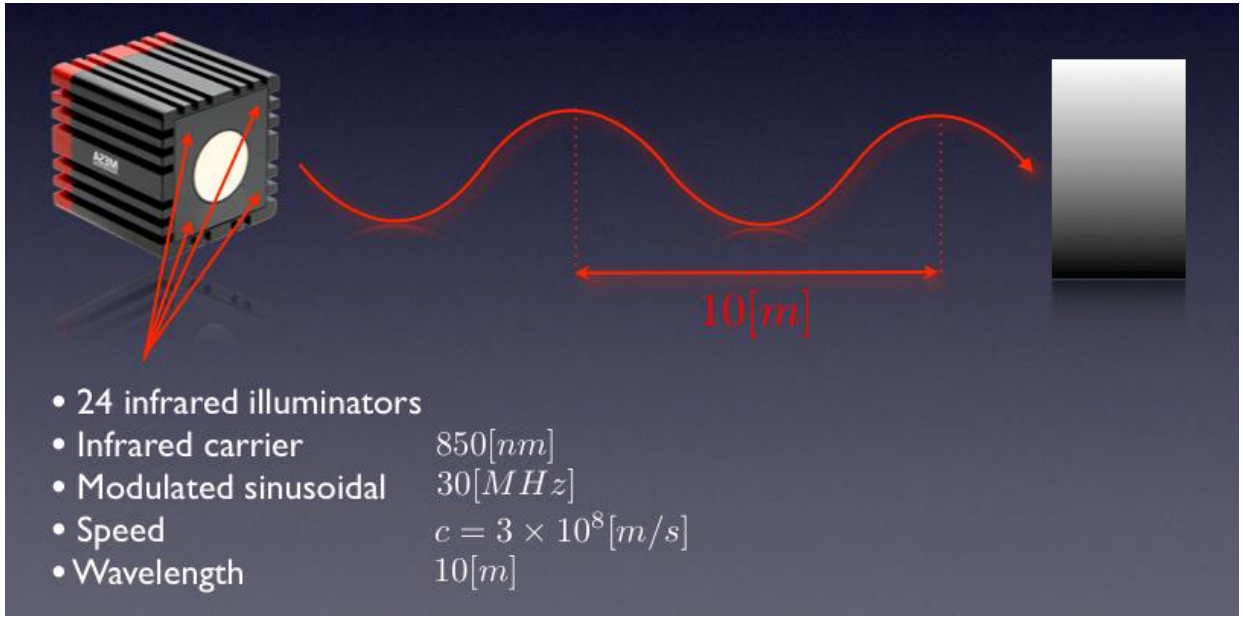
**Green circle**: camera optics

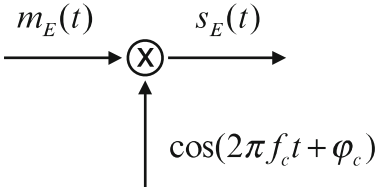
**Red matrix**: ToF sensor

Figure from: C. Dal Mutto, P. Zanuttigh, G.M. Cortelazzo, "Time-of-Flight Cameras and Microsoft Kinect™", Springer Briefs, 2012

# ToF camera: illumination

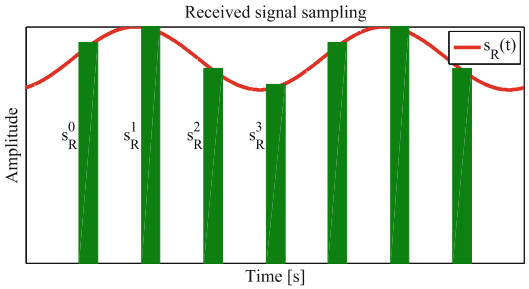
- AM modulation performed by illuminators

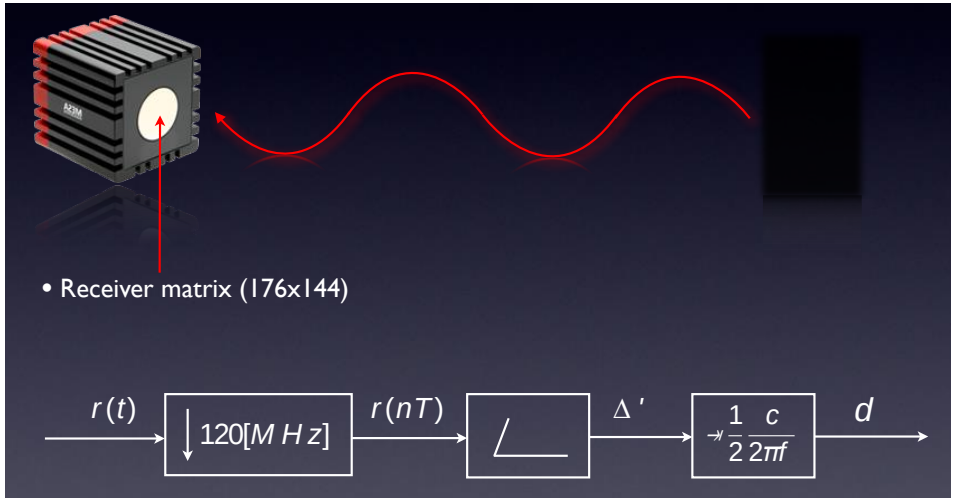


# ToF camera: acquisition

- AM de-modulation performed at each pixel



# Acquired data

$D_T$ Distance map
(radial distance from camera)

$A_T$ Amplitude image

# Acquired data

$A_T$

Amplitude image

$$\hat{A} = \sqrt{\left(\frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n} \rightarrow \frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n+2}\right)^2 + \left(\frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n+1} \rightarrow \frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n+3}\right)^2}{2}$$

$D_T$

Distance map

Step 1: phase estimation

$$\widehat{\varphi} = \operatorname{atan2}\left(\frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n} \rightarrow \frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n+2}, \frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n+1} \rightarrow \frac{1}{N}\sum_{n=0}^{N-1} c_R^{4n+3}\right)$$

Step 2: phase → distance

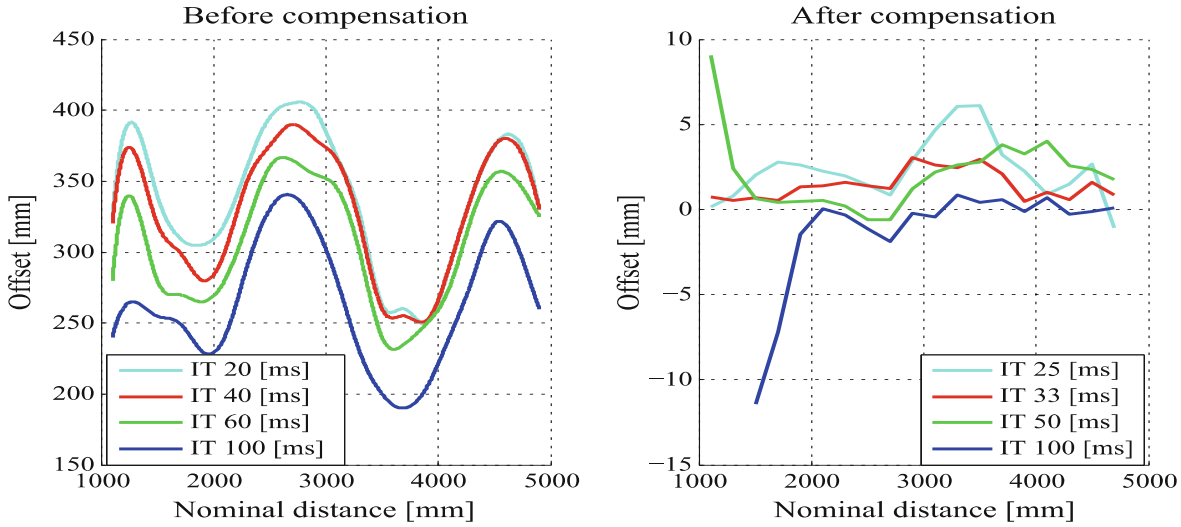
$$\hat{z} = \widehat{\Delta\varphi} * \frac{1}{2}\frac{c}{2\pi f}$$

# Data quality (1)

- Noise model

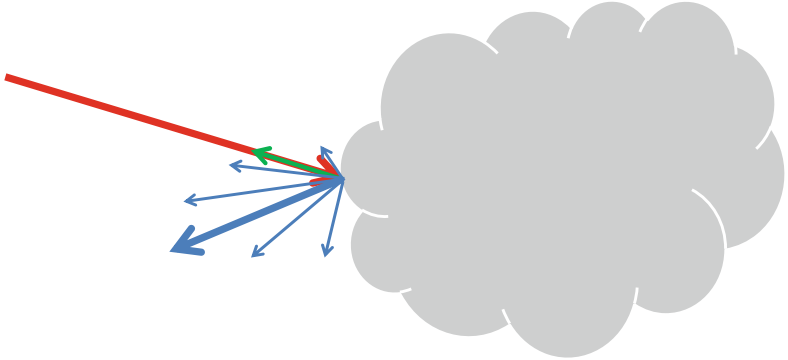
$$= \frac{c}{4 \; f_m \sqrt{2}} \frac{\sqrt{B}}{A}.$$

- Systematic offset

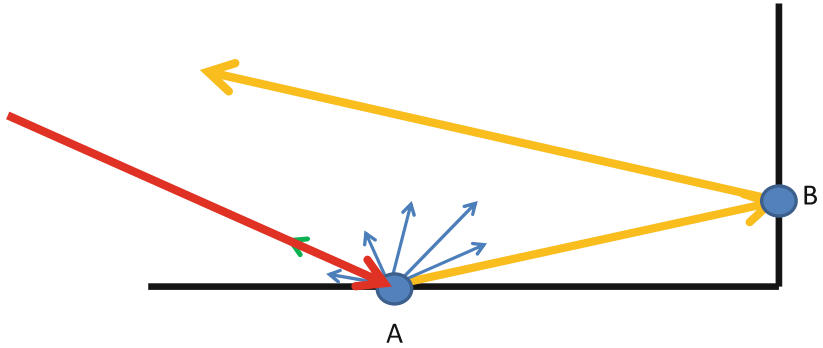


# Data quality (2)

- Scattering



- Multi-path



# ToF cameras

- ToF cameras in the market
  - Gaming
  - 3D reconstruction
  - Small footprint (one camera, low z-height of illuminator)
- Measurements
  - Spatial resolution: same as sensor
  - Z-resolution: good, linear with distance
  - Z-range: good
  - SNR: proportional to the emitted power (no collimation, attenuation with $z^2$)
  - Power consumption: dominated by **illumination** and **computations**

*e.g.*, Google Tango devices.

*e.g.*, Microsoft Kinect v2.

*e.g.*, Microsoft Kinect for Azure.

Note: the systems showed in the figures are only for illustration purposes and are not to be considered “recommended” systems w.r.t. competitor

# Technology review and comparison

# Comparison table

| Tech | Resolution (space/time) | z-res factors | z-range | SNR factors | Power | Footprint |
|---|---|---|---|---|---|---|
| Stereo | Very high/high (1-5 MP, ~60 fps) | $k*z^2$, k>> | b, f [~20 cm, ~20 m] | Pixel (aperture) | Sensors: >100 mW<br>Process: ~5-50 W | 2 sensors |
| Active stereo (TMUX) | High/low (~1 MP, ~30 fps) | $\propto z^2$ | b, f, power (diffuse) [~20 cm, ~1 m] | Power, Pixel | Sensors: ~100 mW<br>Illuminat.: ~500 mW<br>Process: ~1 W | 2 sensor + illuminator |
| Active stereo (SMUX) | Low/high (~VGA, ~60 fps) | $\propto z^2$ | b, f, power (collim.) [~30 cm, <10 m] | Power, Pixel | Sensors: ~100 mW<br>Illuminat.: ~200 mW<br>Process: ~5 W | 2 sensor + illuminator |
| Structured-light | Low/high (~VGA, ~60 fps) | $\propto z^2$ | b, f, power (collim.) [~40 cm, <5 m] | Power, Pxl, Ext. Illumin | Sensors: ~100 mW<br>Illuminat.: ~300 mW<br>Process: ~5 W | 1 sensor + illuminator |
| ToF | Med/med (<VGA, 30-60 fps) | $\propto z$ | Power (diffuse) [~50 cm, ~5 m] | Power, Pxl, Ext. Illumin | Sensors: >100 mW<br>Illuminat.: 1-5 W<br>Process: 1-5 W | 1 sensor + illuminator |

Note: expressed quantities don't refer to specific implementations, but to technology capabilities.

## Thank you!

# Resources

- www.carlodalmutto.ml
- C. Osterwood, "How to Choose a 3D Vision Technology," Embedded Vision Summit 2017
- C. Dal Mutto, P. Zanuttigh, G.M. Cortelazzo, "Time-of-Flight Cameras and Microsoft Kinect™," Springer Briefs, 2012
- P. Zanuttigh, G. Marin, C. Dal Mutto, F. Dominio, L. Minto, G.M. Cortelazzo, "Time-of-Flight and Structured Light Depth Cameras: Technology and Applications"
- M. Hansard, S. Lee, O. Choi, R. Horaud, "Time-of-Flight Cameras," Springer Briefs, 2013
- J. Salvi, J. Pages, J. Battle, "Pattern Codification Strategies in Structured Light Systems"
- R. Szeliski, "Computer Vision: Algorithms and Applications"
- Zbontar, LeCun, "Computing the Stereo Matching Cost with a Convolutional Neural Network"