CEPTON

*Intelligence at the Speed of Light™*

# Calibrate and Align Cameras and Lidars

Cepton Webinar Series #2

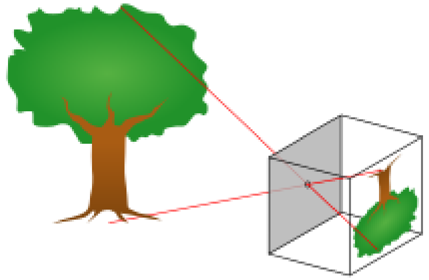
2022-02-18

# Table of Contents



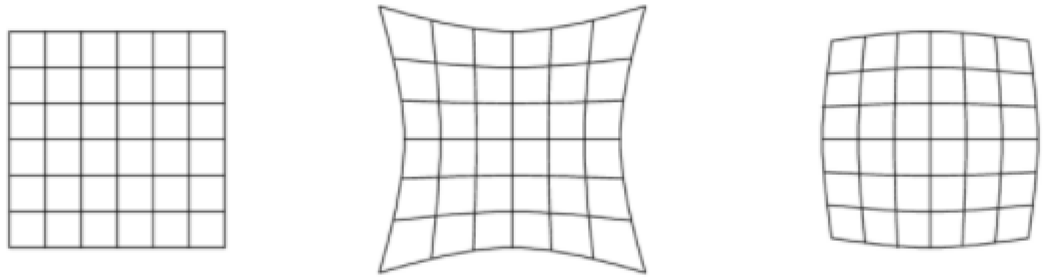
 Cepton is hiring interns and new college grads. Check out our LinkedIn or Handshake job page

# Calibrate Cameras



**Why do we need to calibrate?**

- Ideal lens: Pin-hole model
  - All angles are preserved. Lines in real world remain as lines in image: Grid is not distorted
- All lenses have distortion
  - Pincushion and barrel distortions (Radial distortions only)
  - DNN usually doesn't care.
  - Vital to alignment!



**Select camera for best calibrated results**

- Avoid "smart" cameras (e.g., GoPro, iPhone). Esp. avoid motion compensation.
- Avoid cameras with wide FOV (no fisheye). Too much distortion is hard to correct
- Auto-focus or auto-white balance can be trouble. Lens or aperture change may change distortion.
  - Choose small aperture so the whole picture is focused.
- For moving applications, important to use fast shutter speed.

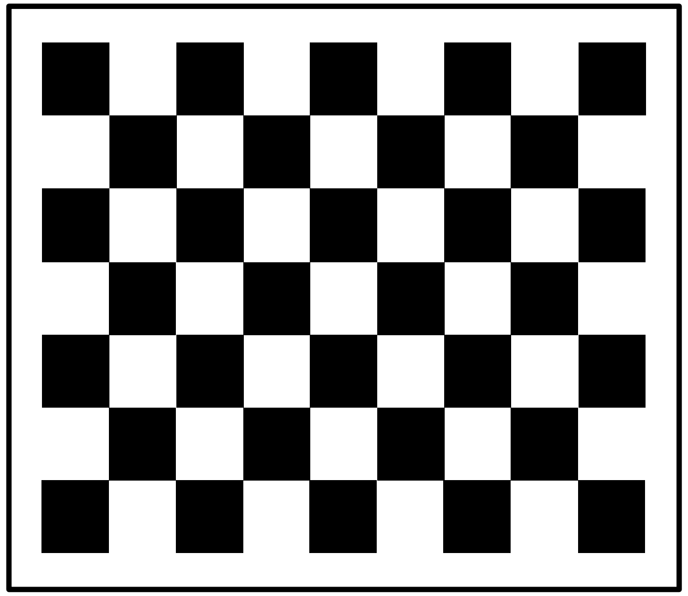
 Cepton is hiring interns and new college grads. Check out our LinkedIn or Handshake job page

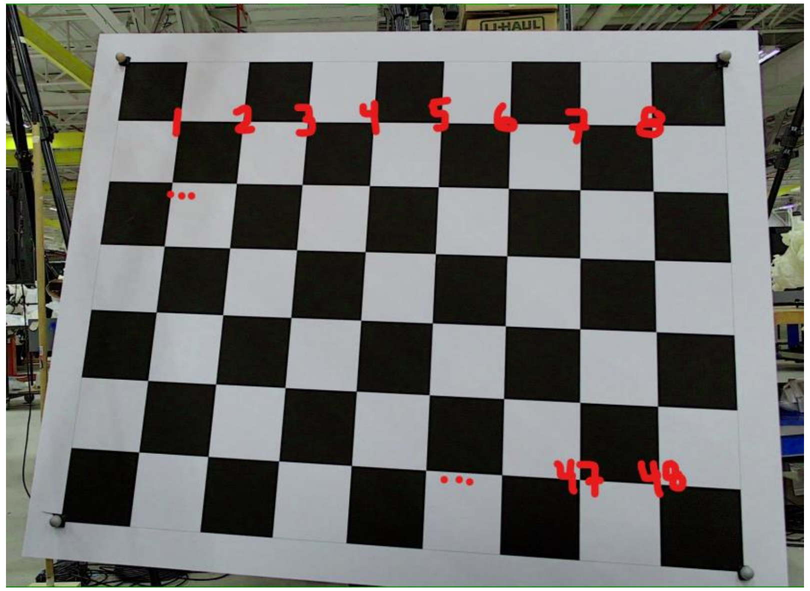
# Calibrate Cameras (continued)

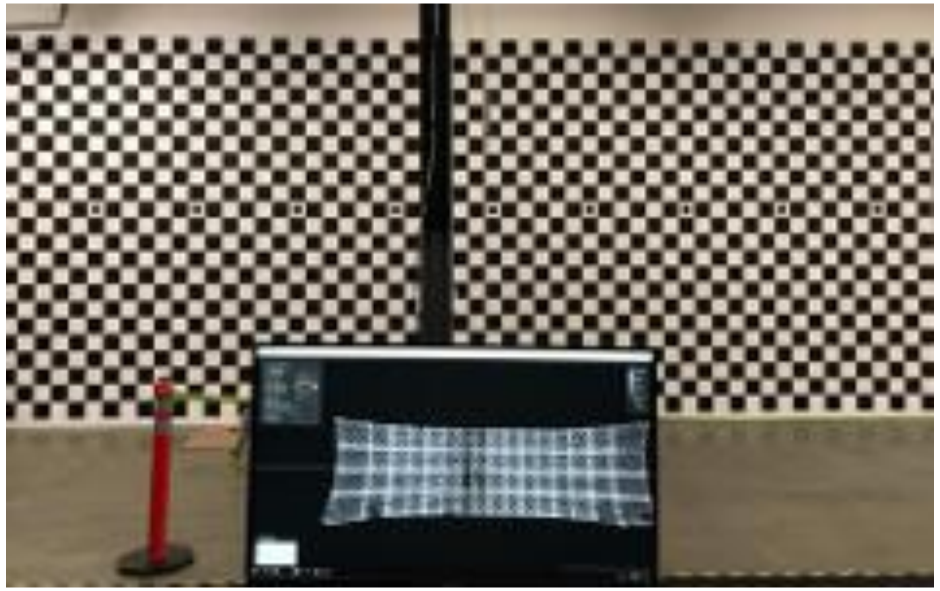
**How to calibrate Cameras?**

- Use checkerboards
- Feature extraction of checkerboard corners:
  - Avoid bias and ambiguity with a bigger convolution kernel
  - Don’t look at individual pixels or reconstruct lines.
- Establish a loss function on the extracted features. (e.g. sum of squares of the error compared with ground truth)
- Introduce rotation/translation (6 degrees of freedom) as free parameters or strictly control your target placement.
- Beware of “period skip”. Introduce “anchor” into your checkerboard if needed.

$$LOSS = \sum_i (C_i - Distort(Transform(C_i^{gt}))^2$$





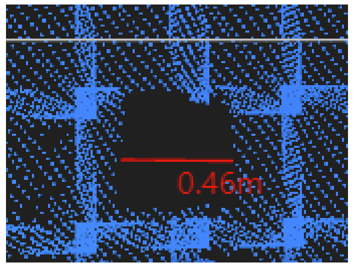



 Cepton is hiring interns and new college grads. Check out our LinkedIn or Handshake job page

# Calibrate Lidars

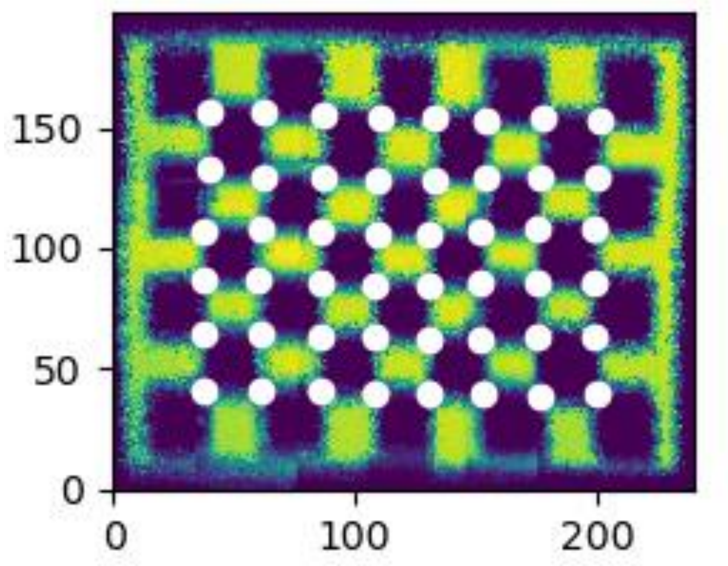
### Do I need to calibrate lidars?

- All lidars from Cepton are already calibrated out of the box.
- A good idea to confirm correctness:
  - No need to do a full space calibration
  - Use a laser pointer measurement tool to try out a few points of interest.
  - You can do this with CeptonViewer’s measurement feature.
- You do need to calibrate if light is optically distorted, e.g., going through a windshield.
  - Use the same checkerboard.




### How to locate the checkerboard on Lidar?

- You need to get Lidar’s “read” of the checkerboard for alignment with camera.
- Understand the lidar point cloud data for calibration:
  - They are sparse.
  - Aggregate enough data (10 frames or more).
- Map 3D to 2D after figuring out the target plane.




 Cepton is hiring interns and new college grads. Check out our LinkedIn or Handshake job page

# Camera-Lidar Fusion

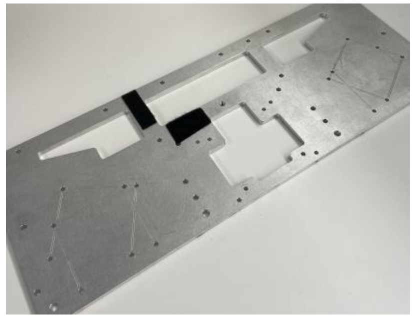
## Geometrical Alignment

- Work with calibrated data only.
- Goal: Figure out the relative rotation and translation between Lidar and camera coordinates.
  - Usually put into a 4x4 matrix (just like in computer graphics)
- Best to choose a world coordinate that matches the perception system (center of car)
  - Figure out the transformation from each sensor to the "world".

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

## How to do camera-lidar alignment





- Cannot talk about alignment without fixture.
- Fixtures lock each sensor tightly onto the "system" (the car).
- Naïve measurements of 6-DOF works for low accuracy.
- Fixed checkerboard placement relative to "system" allows reuse of calibration code.
- Automatic calibration is possible. (More on this later)

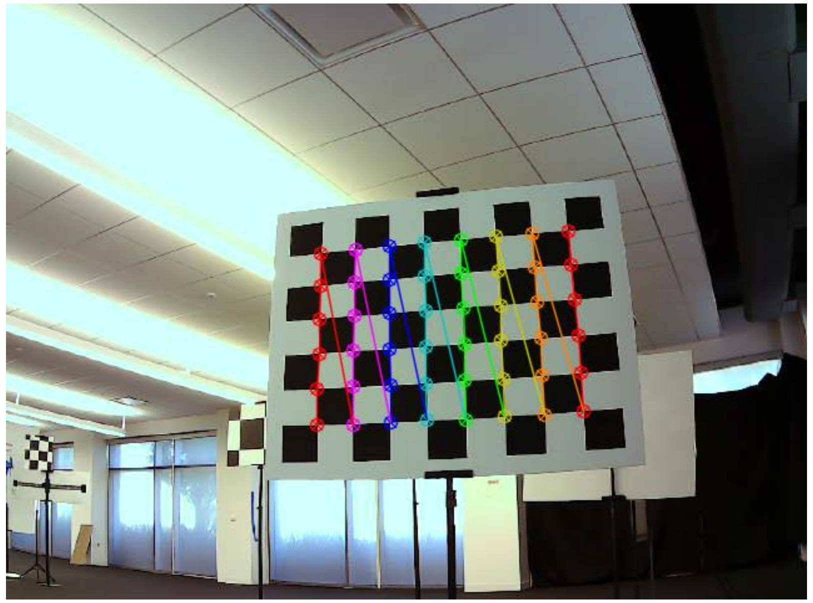
# Camera-LiDAR Fusion

- Camera-LiDAR platform
  - An X-90 Sensor + A camera

Front View

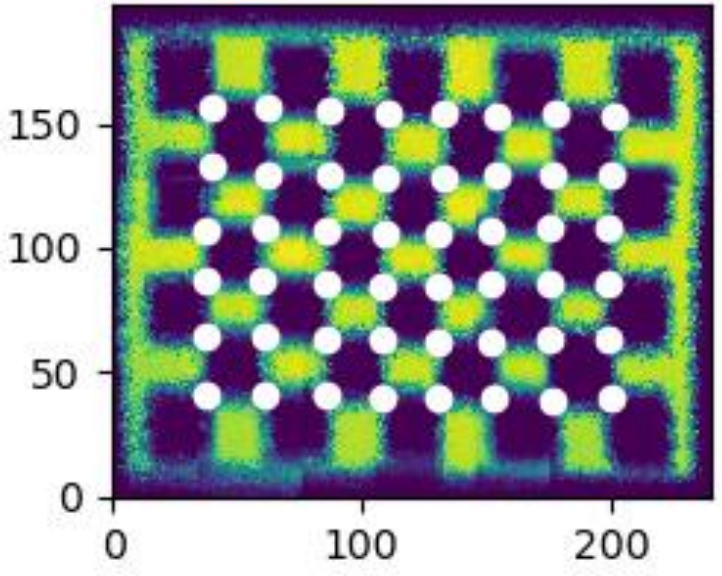

Top View

- Camera-LiDAR Calibration
  - Camera Intrinsic Calibration + Camera-LiDAR Calibration with Checkerboard

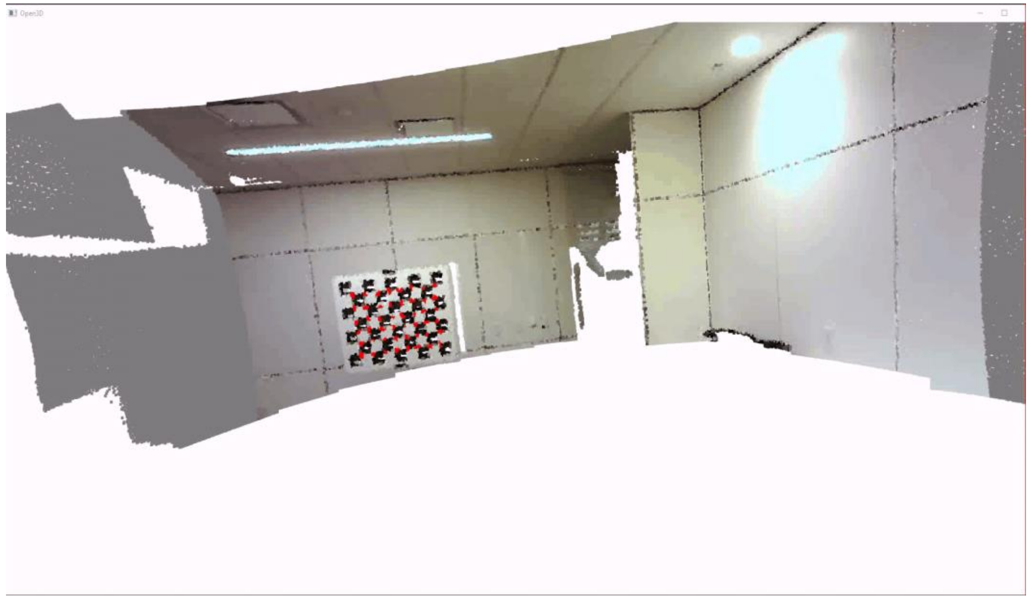


Detected 2-D Corners in Image Frame

Point Correspondences




Detected 3-D Corners in LiDAR Frame



Final Calibration Result

# Camera-LiDAR Fusion

- Camera-LiDAR platform
  - An X-90 sensor + a camera

- Lane Detection Results
  - Detect lane markings on the highway.

Front View

Top View

Our Lane Detection Results Re-projected onto Image

# Align Multiple Sensors

**When there is sufficient overlap (e.g., Camera + Lidar case)**

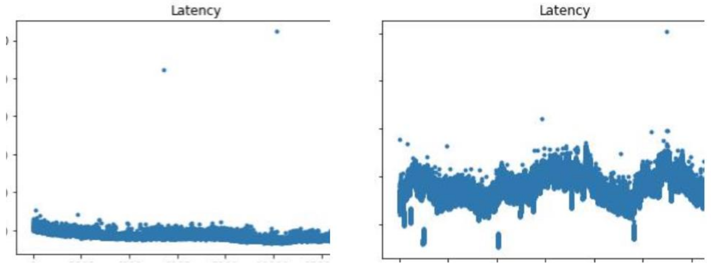
- Strategy: (with a moveable checkerboard)
  - Align pairwise first by moving the checkerboard around
  - Optional global optimization step to “close the loop”
- Strategy: (with scenery)
  - Alignment between cameras is well established (photo stitching algorithms)
  - Alignment between lidars can use direct ICP with normalized point cloud
  - Alignment between camera and lidar still relies on the “feature extract” technique.
  - Use checkerboard, corner of a room, door frame and other extractable features that have both geometry and color differences.
- Strategy: (machine learning)
  - Black box.

# Align Multiple Sensors (continued)

**When there is very small overlap or no overlap**

- Strategy: (with global ground truth)
  - Define world coordinate by ground truth (e.g., a room with several existing checkerboard walls)
  - Align each sensor independently to world coordinate.
  - Take care of precise positioning of your “car” for repeatability. Think about a “fixture” with location screws.
- Strategy: (moving targets)
  - Moving target with constant linear speed.
  - Require correct time synchronization
  - Can use large amount of aggregated data (e.g., driving data, using roadside objects to align)
- Strategy: (rotating sensors)
  - If sensors can be put on a rotational platform with constant rotation, same as above with slightly different math.

# Time Synchronization



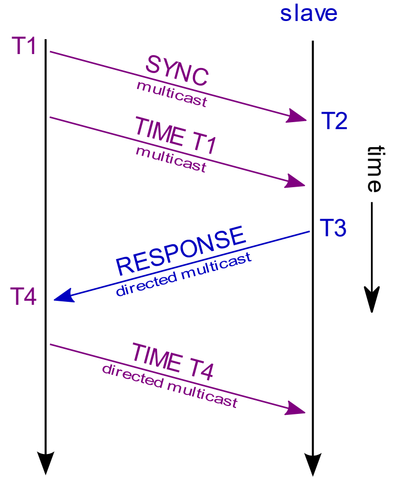

Latency of direct vs. switched network

**Why do we need to synchronize?**

- ➢ Cameras and Lidars do not take measurements at the same time.
- ➢ Data arrival times on computer are highly unreliable esp. with high bandwidth usage.
  - ➢ It may work when you tested and fail in the field.
  - ➢ Performance is different for different ethernet devices. (including switches in the middle)
  - ➢ Don't leave it to luck.
- ➢ Synchronization is not needed for static scene, e.g., some calibration scenarios.
- ➢ Synchronization is import for any moving scenario.

**How to synchronize?**

- ➢ Use (Precision Time Protocol) PTP for Lidar. 802.1AS or other IEEE1588 based mechanisms.
- ➢ No new wires, all based on the ethernet.
- ➢ Linux open source support by default (ptp4l)
- ➢ Some cameras also support PTP
- ➢ For the cameras that don't support PTP, use a fixed frame rate and calibrate the time difference.




 Cepton is hiring interns and new college grads. Check out our LinkedIn or Handshake job page

# Time Synchronization (continued)

**Concept of a "frame"**

| | Camera | Lidar |
|---|---|---|
| What is a "frame" | A picture | A sequence of measurements that cover the whole field of view. |
| Timestamp | Usually global shuttered, with a single timestamp | Measured point-by-point, with each point carrying its own timestamp. |
| Trigger | Electrically controlled, either fixed time intervals or dynamically triggered. | Mechanically controlled. Usually cannot guarantee specific timing. |

- Frame is a natural unit of perception.
- Different sensors have different frame timings and different frame rates.

**How to do sensor fusion at frame level**

- Use "most recent frame" from all sensors, do this at fixed interval.
  - Introduce long latency up to a frame
  - Easiest to implement
- Use lidar frames as they arrive and take most recent camera data
  - Identify the "primary" lidar, as lidars don't have synchronized frames between each other. (Usually the front facing one)
  - Camera frame rate can be tuned to be fast enough to avoid any latency caused problems.
- Use "fixed time interval" for lidar sensor (usually based on camera frame rate):
  - Needs good understanding of how the lidar scan works
  - Requires normalization of point cloud density

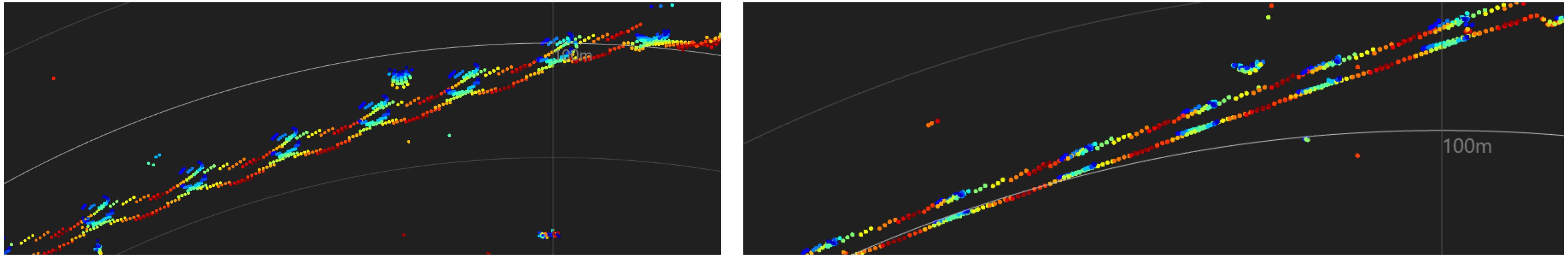
# Time Synchronization (continued)

## Motion Compensation

- Every camera frame and every single lidar point has its own timestamp.
- Perception algorithms work best with a fixed point in time.
- Project measurement positions to a fixed time by assuming constant speed of motion (object or sensor itself)

$$\vec{S}(t) = \vec{S_0} + \vec{v}\Delta t$$

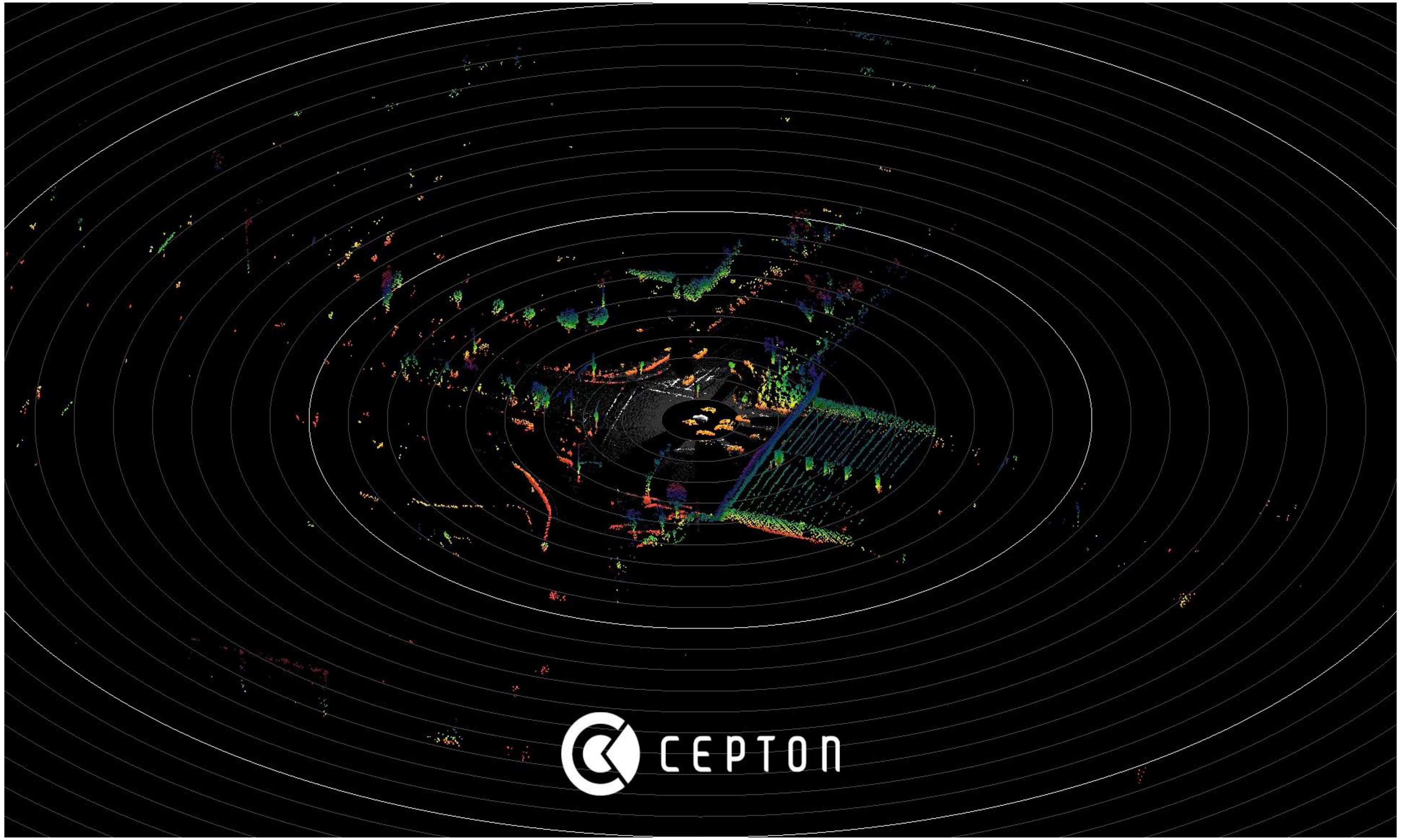
- This is a bigger topic beyond the scope of this talk:
  - Connected to ego-motion algorithm
  - Turning and linear motions are different

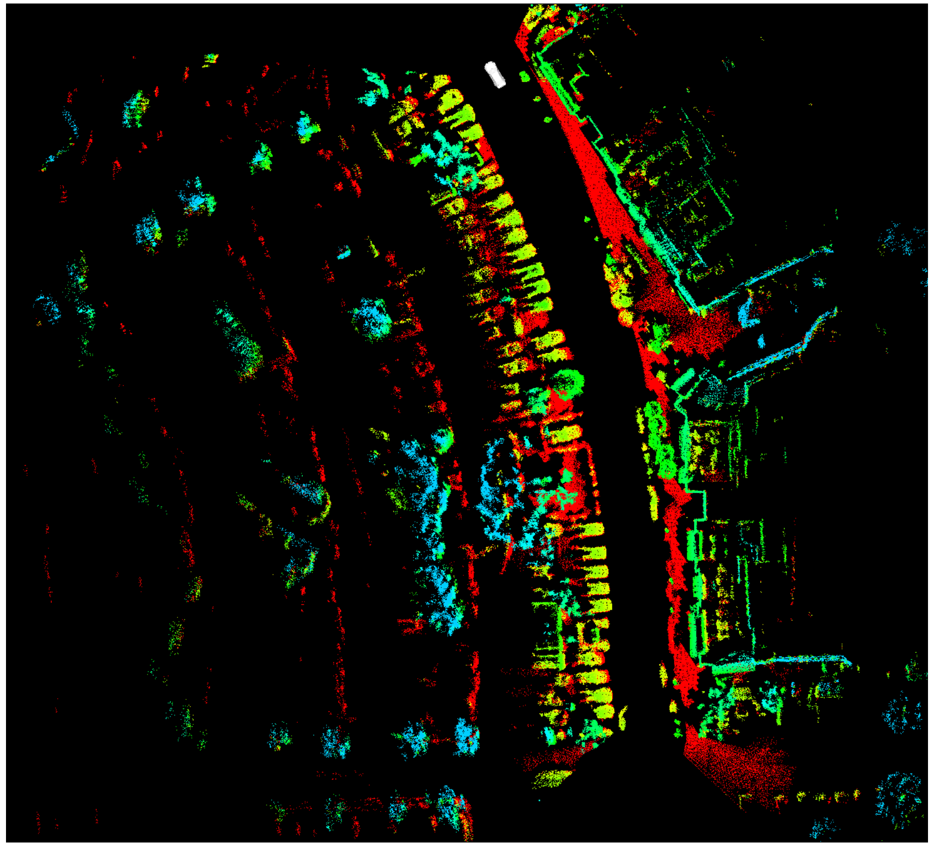



Visualized time-within-frame when passing a bridge. Before and after motion compensation.

 Cepton is hiring interns and new college grads. Check out our LinkedIn or Handshake job page

# System with 6 Lidars



 Cepton is hiring interns and new college grads. Check out our LinkedIn or Handshake job page

# SLAM Aggregation (Single Lidar)

# Future Topics For Webinar

- ROS2 integration in-depth.
- Python SDK and offline data processing.
- Advanced SDK and SDK internals.
- MMT and scan pattern.
- Cepton’s perception system and CR file.


Cepton is hiring interns and new college grads. Check out our LinkedIn or Handshake job page

# Resources

- Developer Center (https://developer.cepton.com coming soon…)
  - This and all other webinars
  - Download SDK package
  - Download Cepton Viewer executable
- Official cepton.com
- JOB postings: LinkedIn and Handshake


Cepton is hiring interns and new college grads. Check out our LinkedIn or Handshake job page