Lab 3

Depth Images

This lecture is part of the RACECAR-MN introductory robotics course.
You can visit the course webpage at mitll-racecar-mn.readthedocs.io.

Objectives

Main Objective: Use the depth camera to provide more information about the environment

Learning Objectives

- Use gaussian blur to reduce noise
- Use depth images to identify the distance at a point
- Use depth images to identify the closest object
- Combine depth and color images to identify and measure the distance of objects

Color Images

Group activity

- **Previous approach**: Use color thresholds to identify contours, calculate center and area
- What are some potential issues with this approach?

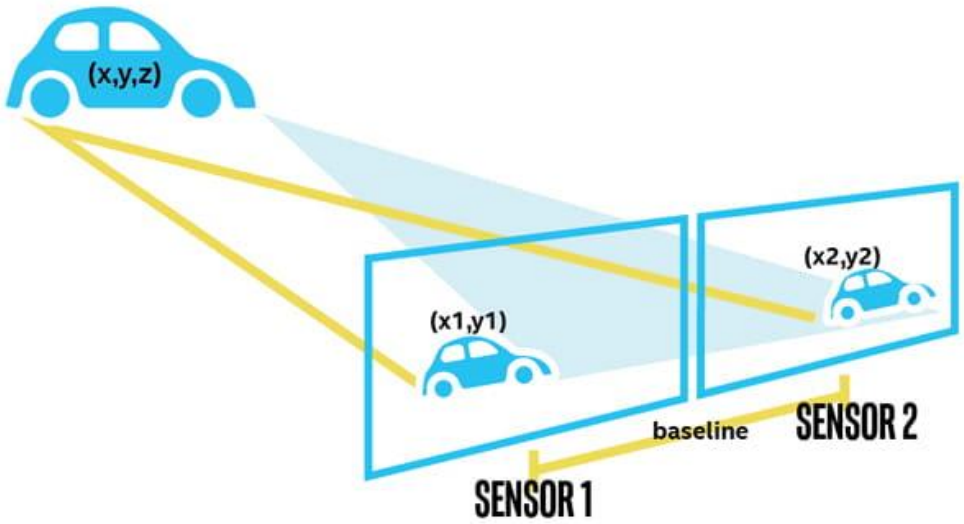
Color Images

Group activity

- **Previous approach**: use color thresholds to identify contours, calculate center and area
- What are some potential issues with this approach?
  - Multiple objects of the same color
  - Different sized objects of the same color
  - Complex objects with many colors
  - Etc.

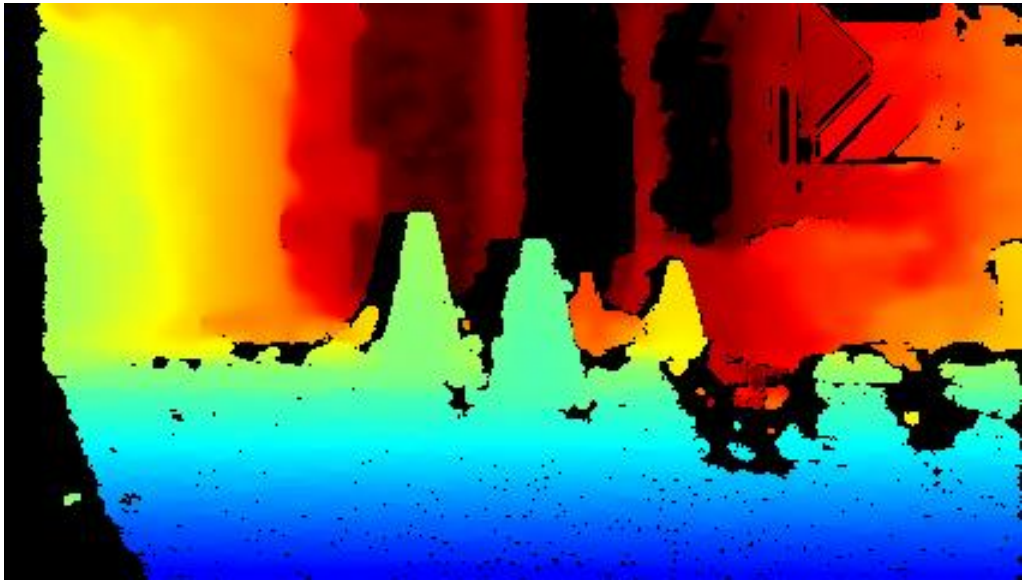
Stereoscopic Vision

- Two sensors capture an image at the same time and compare corresponding points
  - The greater the difference, the closer the object



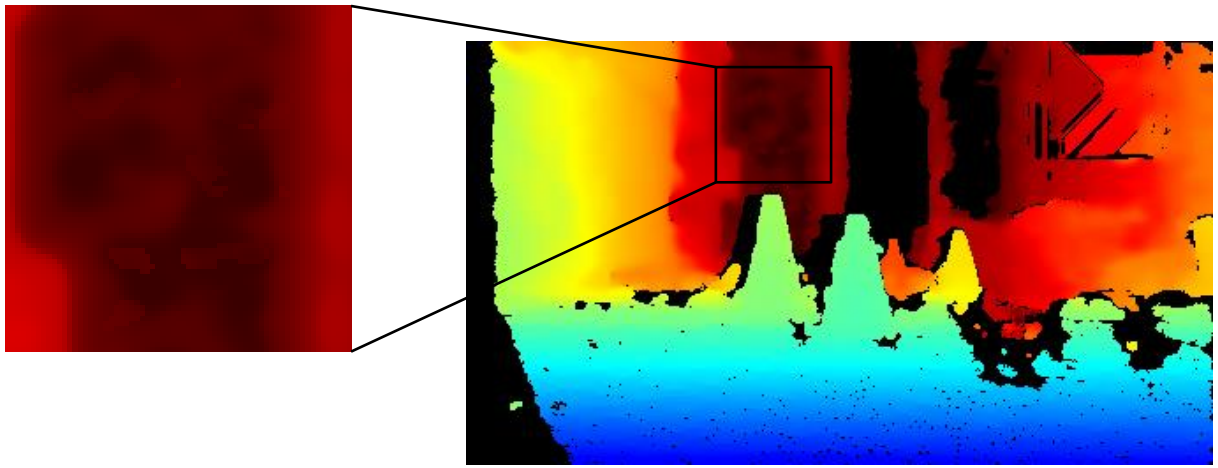
Depth Image

- Just like a color image, but each "pixel" stores a distance rather than a color
  - Can visualize by mapping distance to color



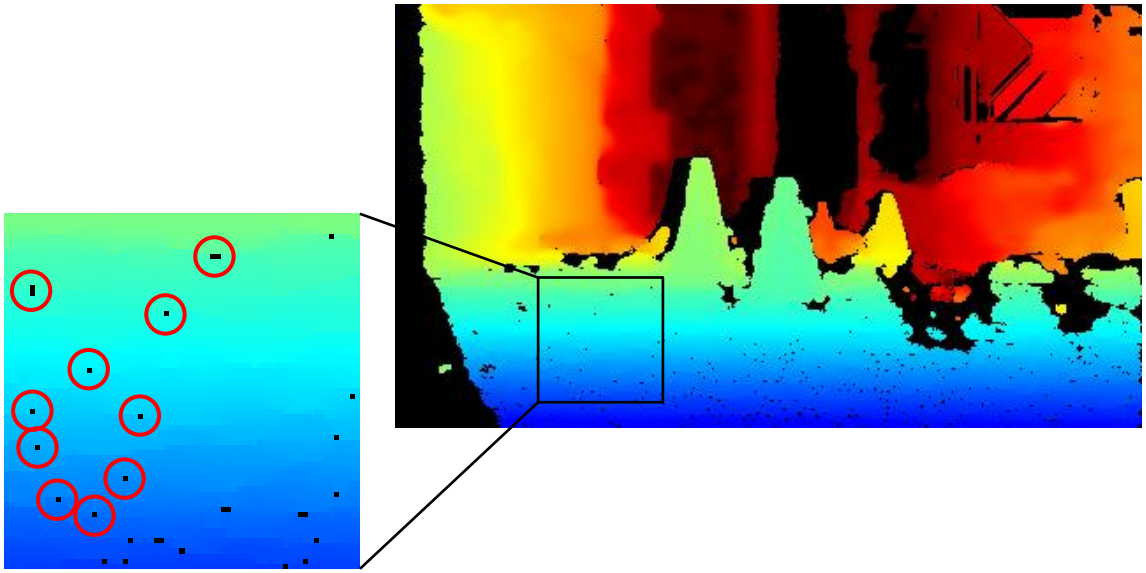
Noise

- Each pixel measurement has random error due to imperfections in the sensors
  - Depends on distance, material, lighting, temperature, etc.



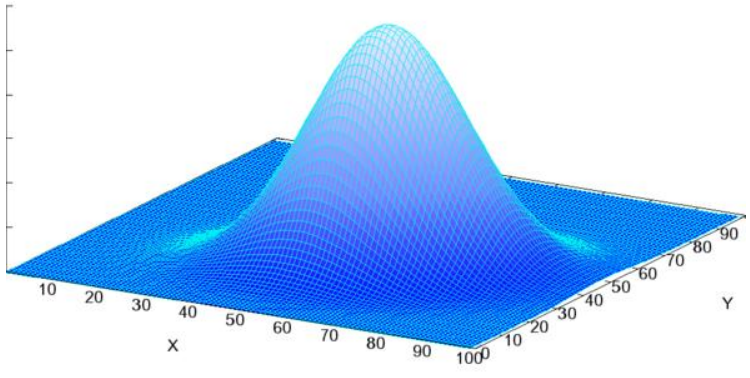
Null Values

- Pixels without data are stored as 0.0 cm
  - This appears black on the visualization



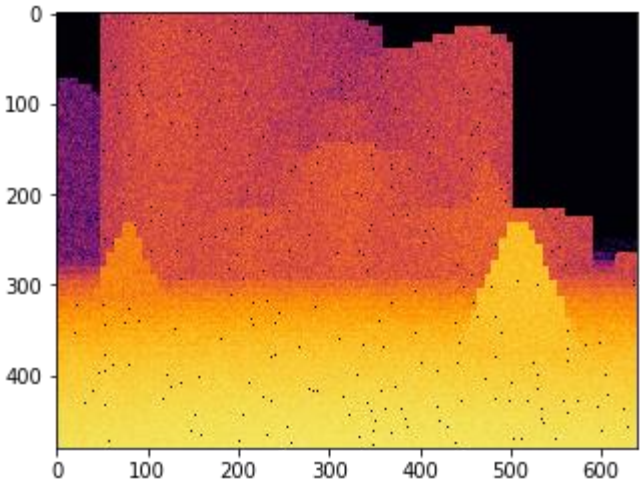
Gaussian Blur

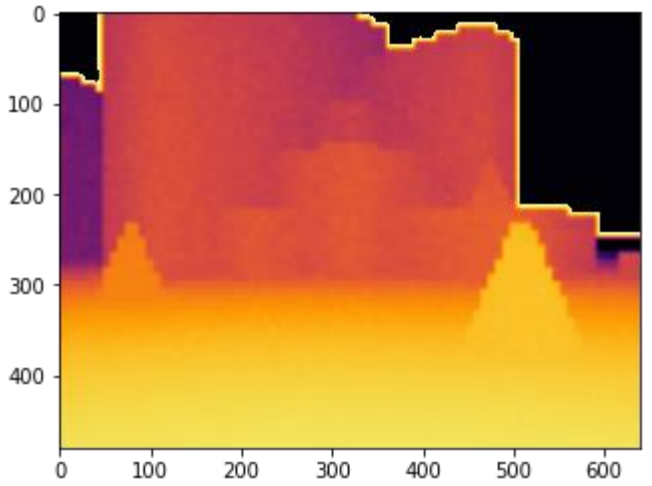
- **Problem:** If we focus on a single pixel, noise and null values can create major issues
- **Solution:** Average each pixel with its neighbors
- **Gaussian blur** = each pixel is updated with a weighted average of its neighbors
  - **kernel** = the area used in the calculation
  - Larger kernel means more blurred



Gaussian Blur

- An image before and after Gaussian blur
  - Kernel size = 11 pixels





Camera Module

- Retrieves color and depth images from the camera
- Public Interface
  - `get_color_image()`
  - **`get_depth_image()`**
  - `get_width()`
  - `get_height()`
  - **`get_max_range()`**

Display Module

- Displays data and images to the screen
  - Simulation: Creates a window on your computer
  - RACECAR: Creates a window on the mini-monitor
- Public Interface
  - `show_color_image()`
  - **`show_depth_image()`**
  - `show_lidar()` (we'll use this later)

Examples 1 and 2

Group activity

```
# Example 1
depth_image = rc.camera.get_depth_image()
depth_image = depth_image[0 : rc.camera.get_height() * 2 // 3, :]

# Example 2
depth_image = rc.camera.get_depth_image()
depth_image = (depth_image - 0.01) % rc.camera.get_max_range()
```

Example 3

Group activity

```
# Example 3
depth_image = rc.camera.get_depth_image()
temp = np.copy(depth_image)

for r in range(1, rc.camera.get_height() - 1):
    for c in range(1, rc.camera.get_width() - 1):
        temp[r][c] = (
            depth_image[r - 1][c - 1]
            + depth_image[r - 1][c]
            + depth_image[r - 1][c + 1]
            + depth_image[r][c - 1]
            + depth_image[r][c]
            + depth_image[r][c + 1]
            + depth_image[r + 1][c - 1]
            + depth_image[r + 1][c]
            + depth_image[r + 1][c + 1]
        ) / 9

depth_image = temp
```

Example 3

Group activity

```
# Example 3
depth_image = rc.camera.get_depth_image()
temp = np.copy(depth_image)

for r in range(1, rc.camera.get_height() - 1):
    for c in range(1, rc.camera.get_width() - 1):
        temp[r][c] = (
            depth_image[r - 1][c - 1]
            + depth_image[r - 1][c]
            + depth_image[r - 1][c + 1]
            + depth_image[r][c - 1]
            + depth_image[r][c]
            + depth_image[r][c + 1]
            + depth_image[r + 1][c - 1]
            + depth_image[r + 1][c]
            + depth_image[r + 1][c + 1]
        ) / 9

depth_image = temp
```

- Why do we create a temporary copy?
- Why do we not iterate over the entire range?

Lab 3 Objectives

- **Jupyter Notebook**: Write helper functions to measure distance, reduce noise, and find the closest point
- **Lab 3A**: Safety stop
- **Lab 3B**: Cone parking (revisited)
- **Lab 3C**: Wall parking