An Introduction to Course Project

2022/09/19

RAP: Robotic rAndom bin Picking

A **dual-stage** project, consisting of:

Stage 1: Extracting the information behind the **media**.

Try to establish your own stereo matching system for 3D perception.

Stage 2: Designing the action based on the **cognition**.

Maneuvering the robot in the 3D world upon request.

Just like a common **intelligent system**……

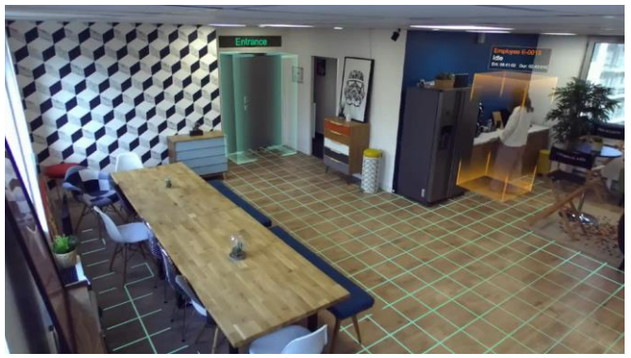
Media acquisition

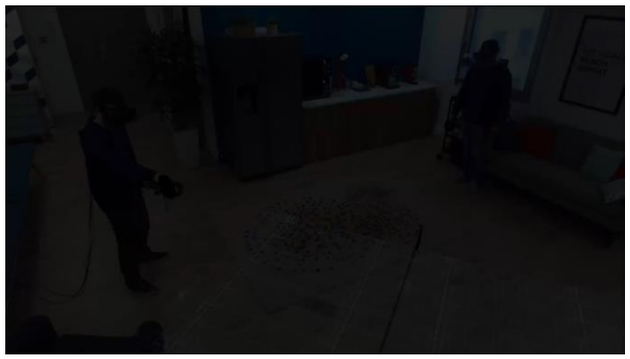
Cognition, decision making & action

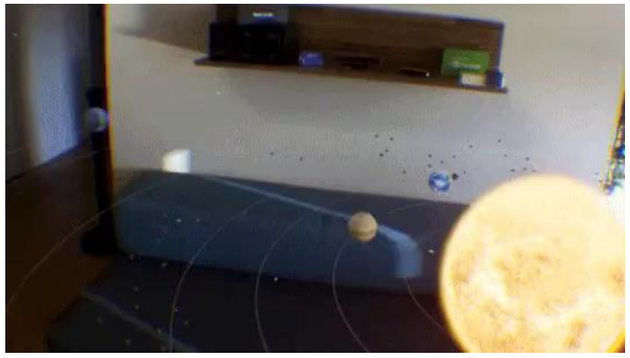
Stage 1 **Media Acquisition:** Stereo Vision

3D perception: background

Distance perception, 3D estimation have become a supportive & fundamental technique in several applications.
Broadcasting, surveillance, industrial, entertainments……







Building your own depth estimation system

We will provide two homogeneous camera (AR0330 sensor, 70 degree FOV lens, USB link, copper via for mounting).

You may need to:

1. Choose the baseline distance for your camera set, and 3D print the frame holder (in our lab or in iSpace);
2. Calibration and rectification for your stereo camera;
3. Designing or choosing the matching algorithm (traditional methods or learning based methods);
4. Testing the stereo system in our benchmark scenario, and further in the robotic system……

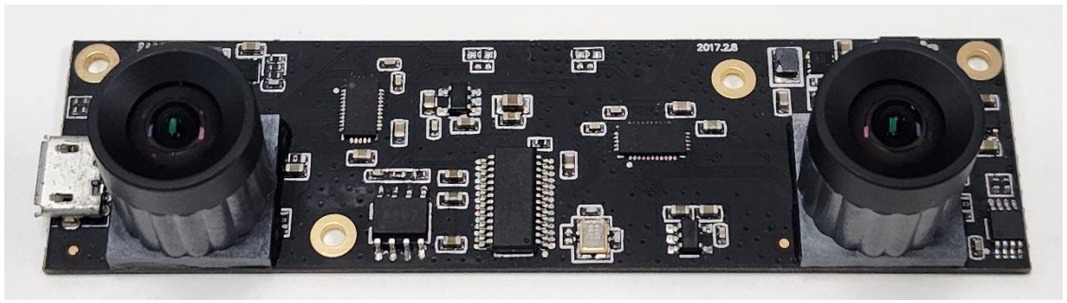
Something maybe useful:

Camera intrinsic and extrinsic matrix and how to calibrate them;
Image distortion and color calibration;
Some useful software e.g. MATLAB Stereo Toolbox, OpenCV……
Benchmark metrics e.g. accuracy (MSE, SSIM…), speed, robustness (extreme cases like repeating patterns, weak texture regions…)……

Building your own depth estimation system

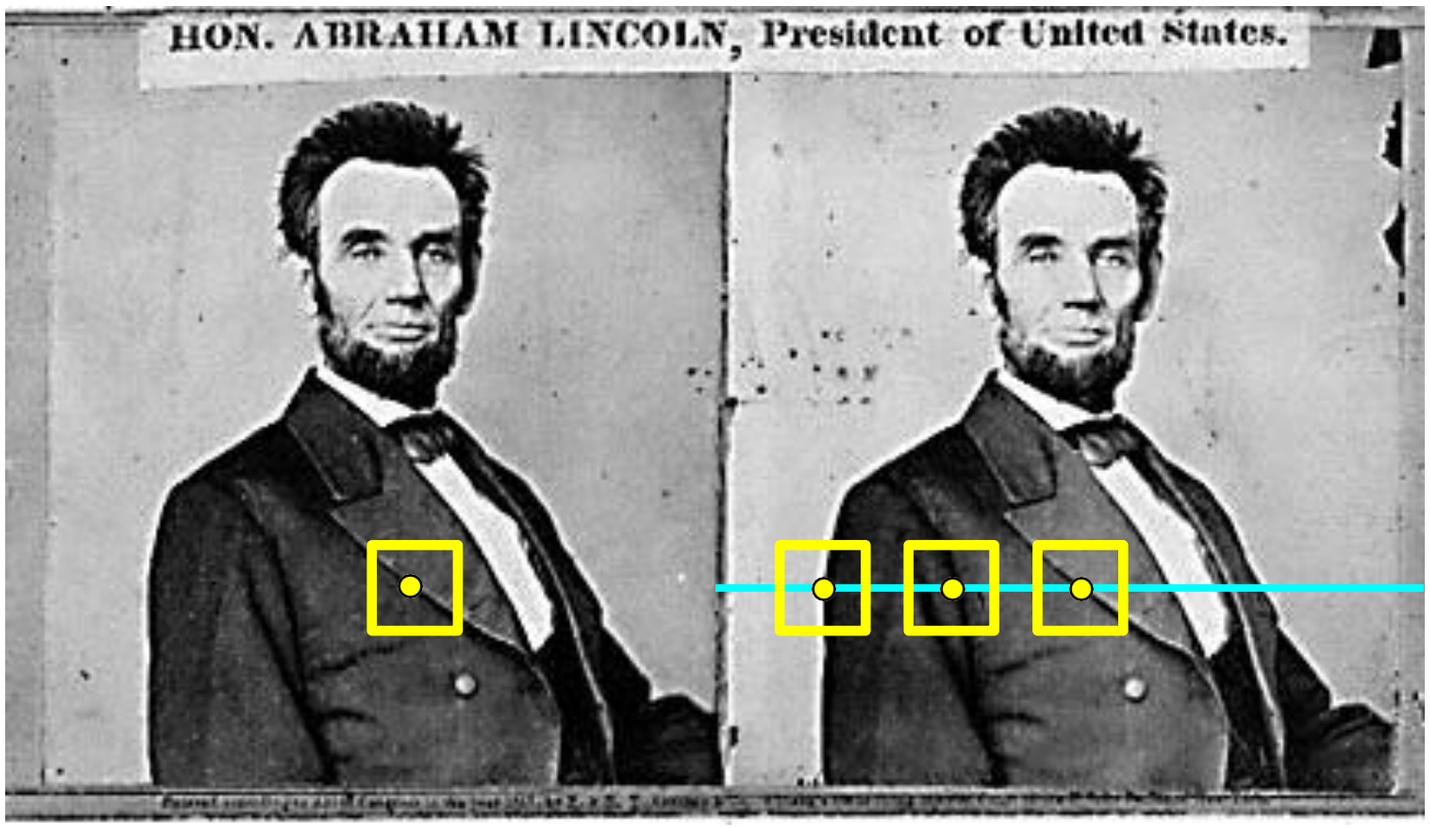
We have separated cameras. ☺

But integrated camera is also available, if you could bear a score discount. ☹



*Work in teams of **3 to 4** people.*
*End-of-term **presentation** and assessments.*

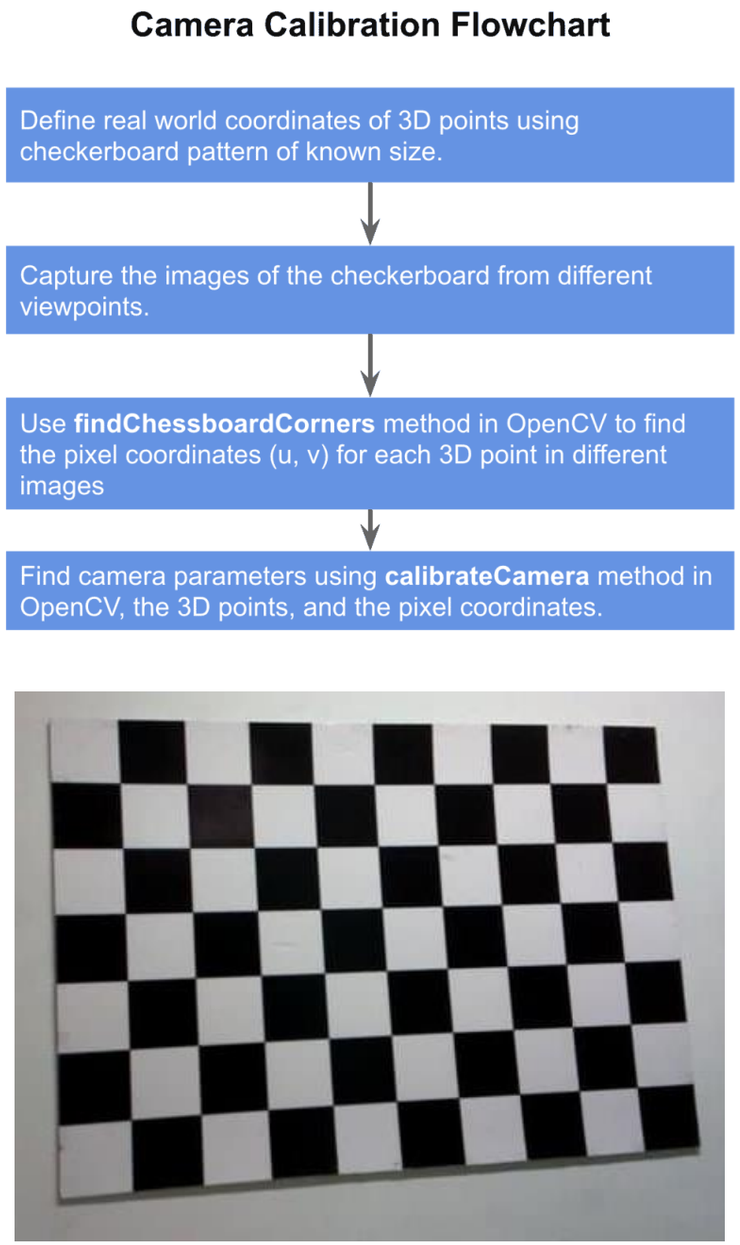
Example pipeline

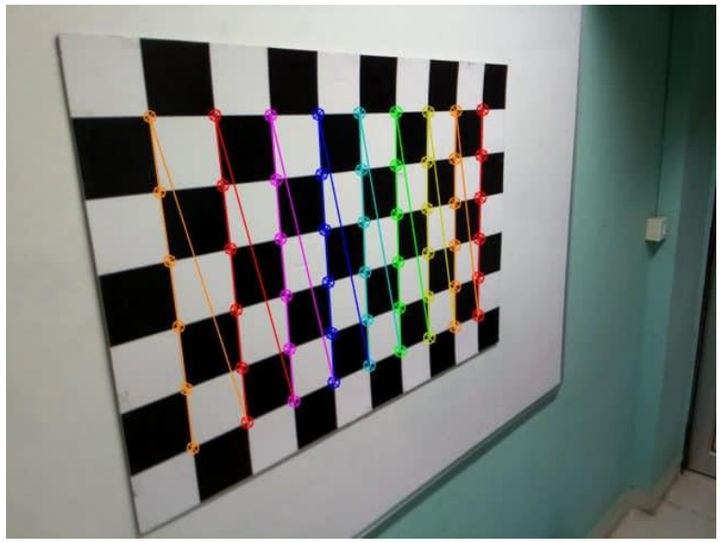


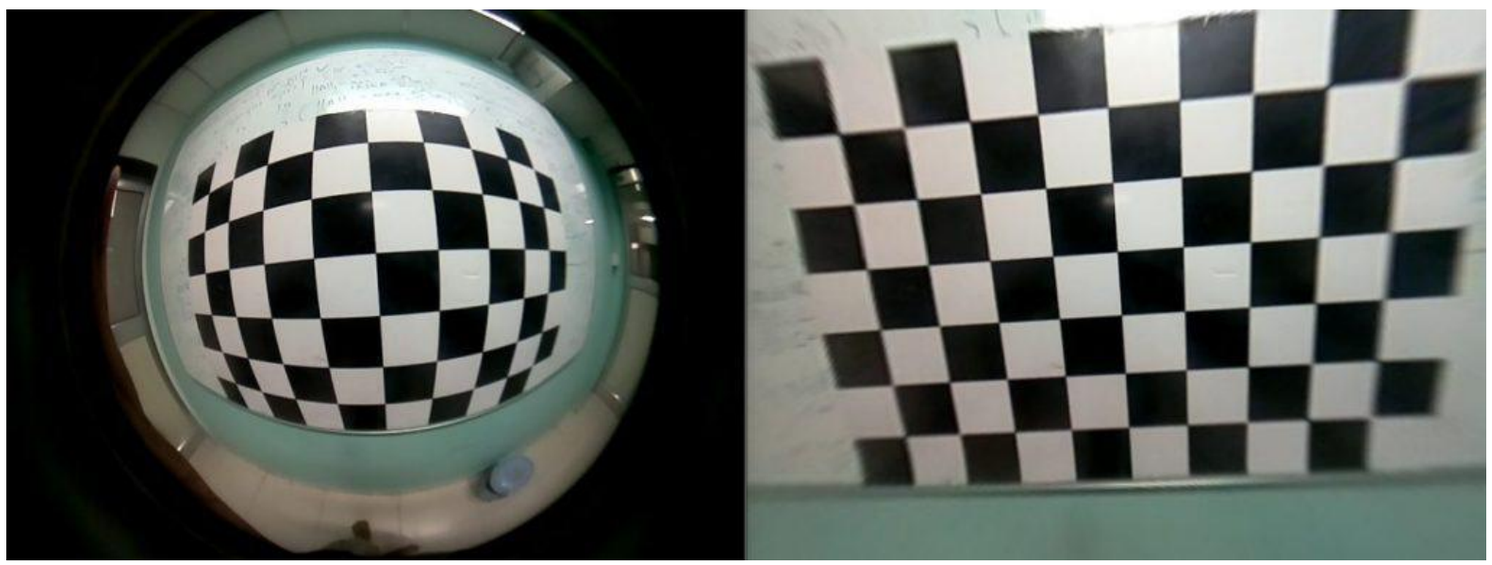
1. Rectify images
   (make epipolar lines horizontal)
2. For each pixel
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity

Calibrate your camera

An example: using the checkerboard

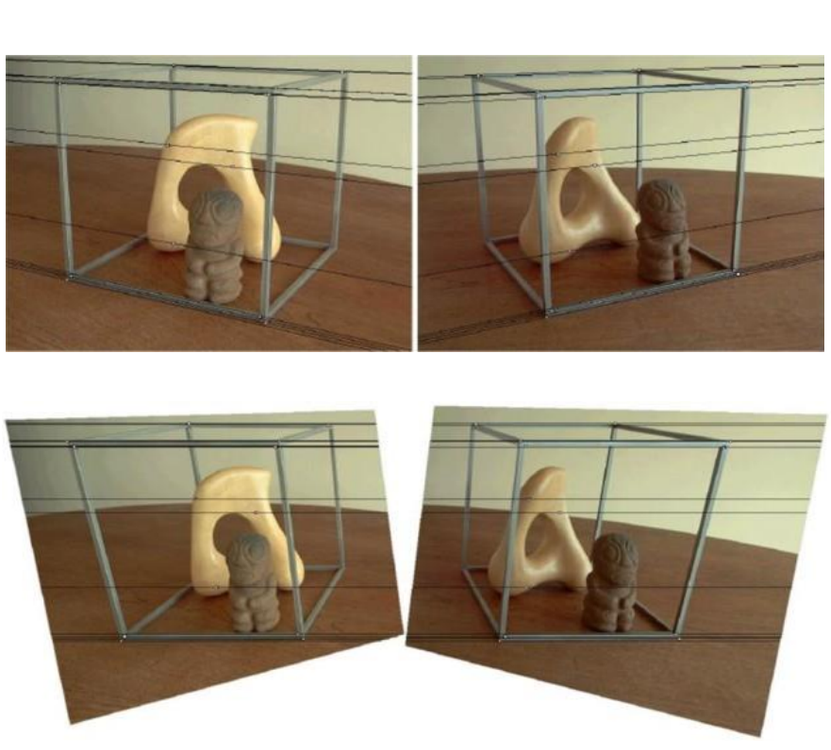


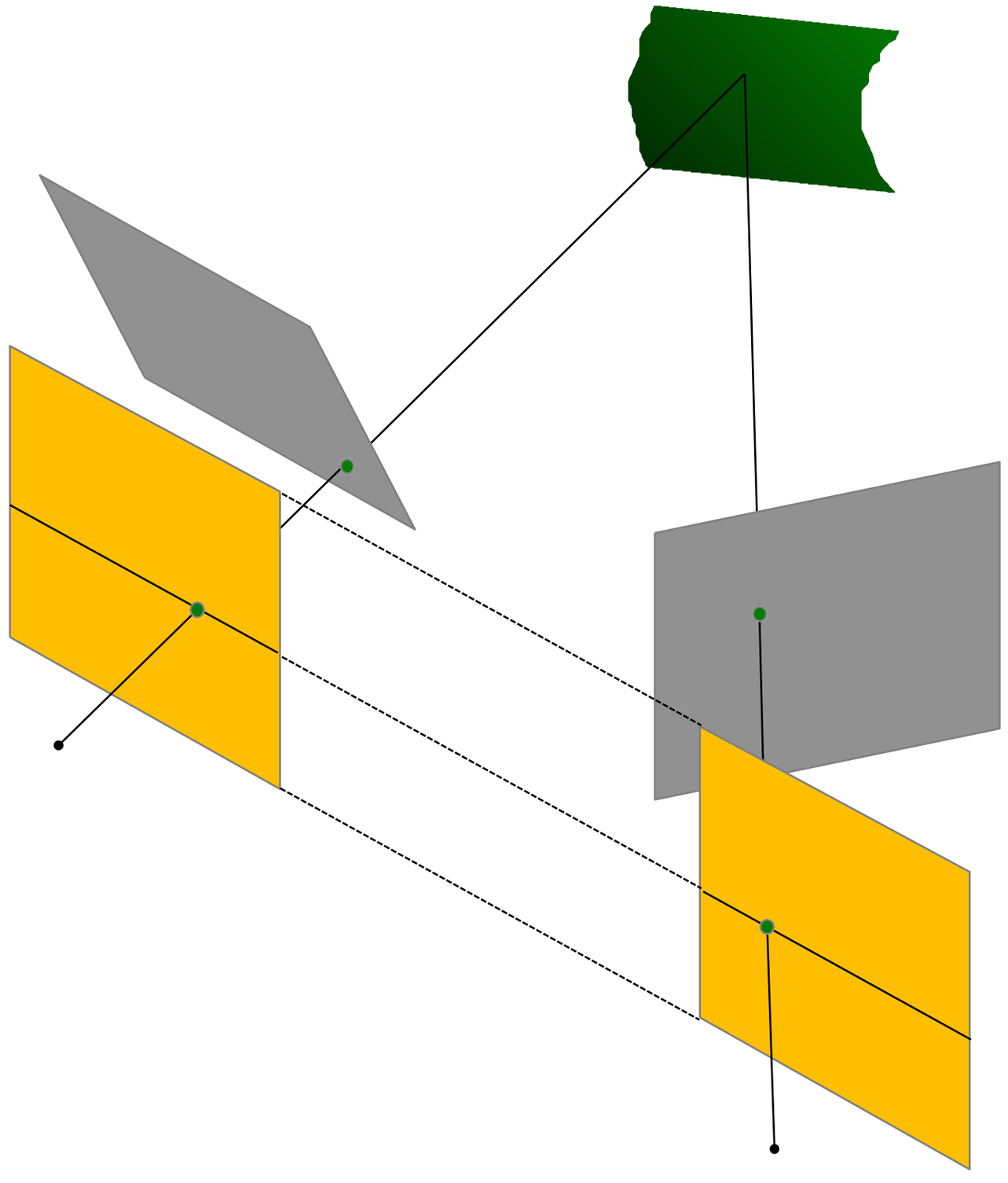




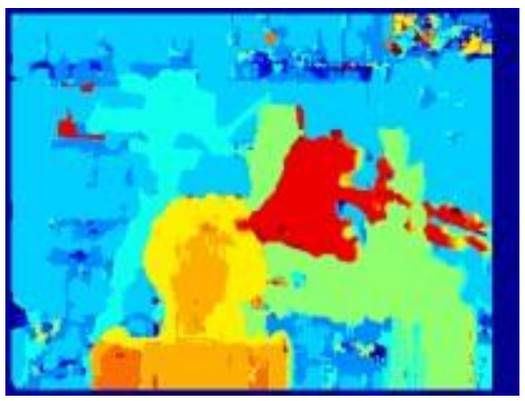
→ *Stereo rectification*

Reproject image planes onto a **common plane** parallel to the line between camera centers

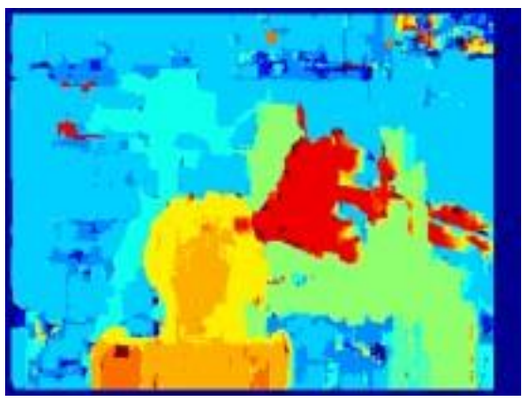




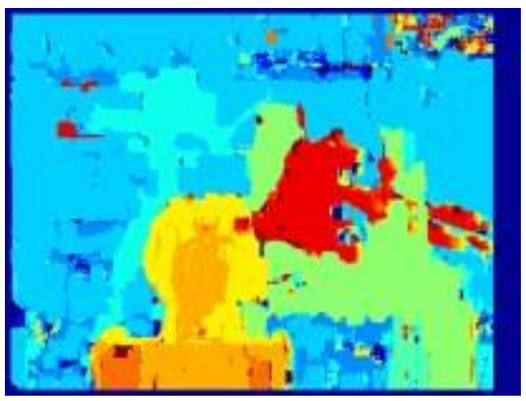
Similarity Measure	Formula
Sum of Absolute Differences (SAD)	$\sum_{(i,j)\in W} \lvert I_1(i,j) - I_2(x+i,y+j)\rvert$
Sum of Squared Differences (SSD)	$\sum_{(i,j)\in W} \left(I_1(i,j) - I_2(x+i,y+j)\right)^2$
Zero-mean SAD	$\sum_{(i,j)\in W} \lvert I_1(i,j) - \bar{I}_1(i,j) - I_2(x+i,y+j) + \bar{I}_2(x+i,y+j)\rvert$
Locally scaled SAD	$\sum_{(i,j)\in W} \lvert I_1(i,j) - \frac{\bar{I}_1(i,j)}{\bar{I}_2(x+i,y+j)} I_2(x+i,y+j)\rvert$
Normalized Cross Correlation (NCC)	$\frac{\sum_{(i,j)\in W} I_1(i,j).I_2(x+i,y+j)}{\sqrt[2]{\sum_{(i,j)\in W} I_1^2(i,j).\sum_{(i,j)\in W} I_2^2(x+i,y+j)}}$



SAD



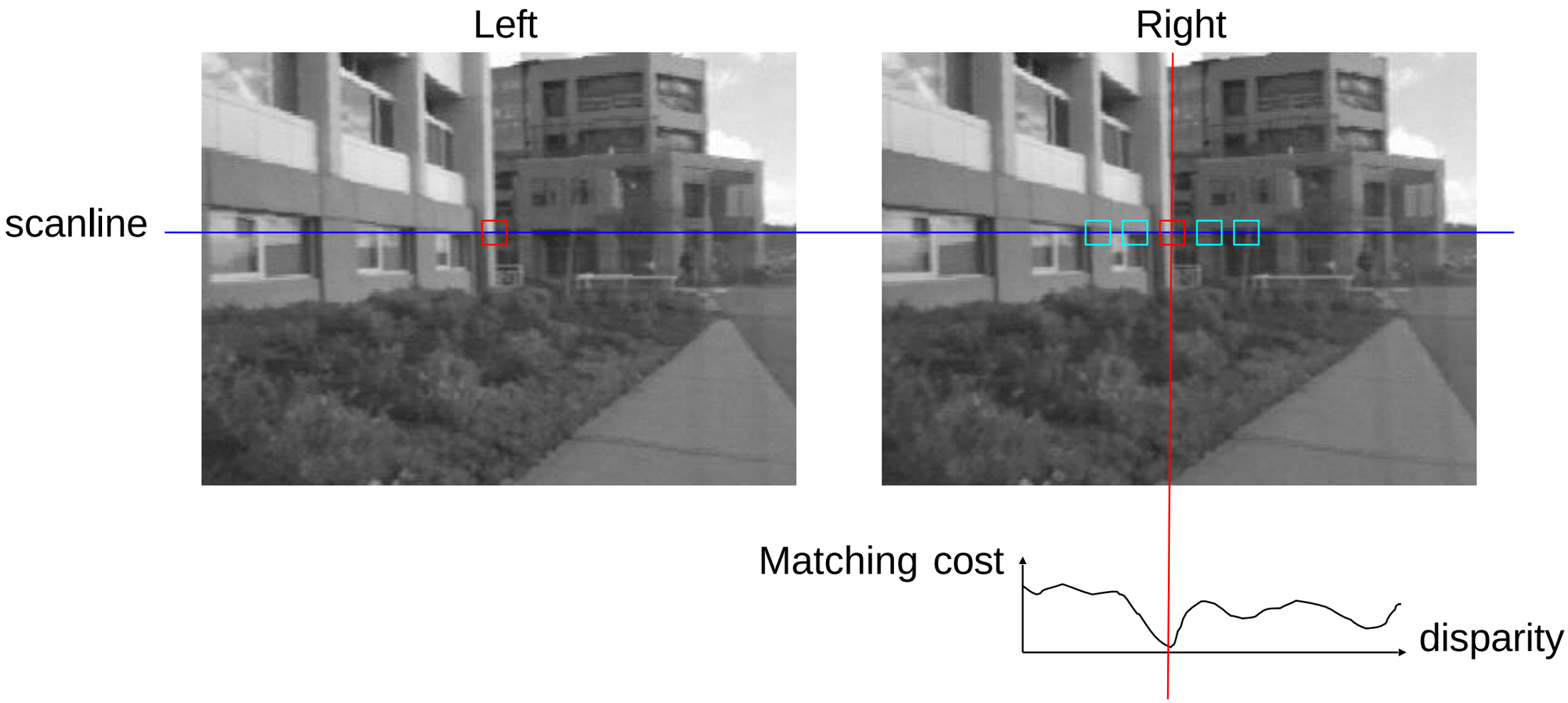
SSD



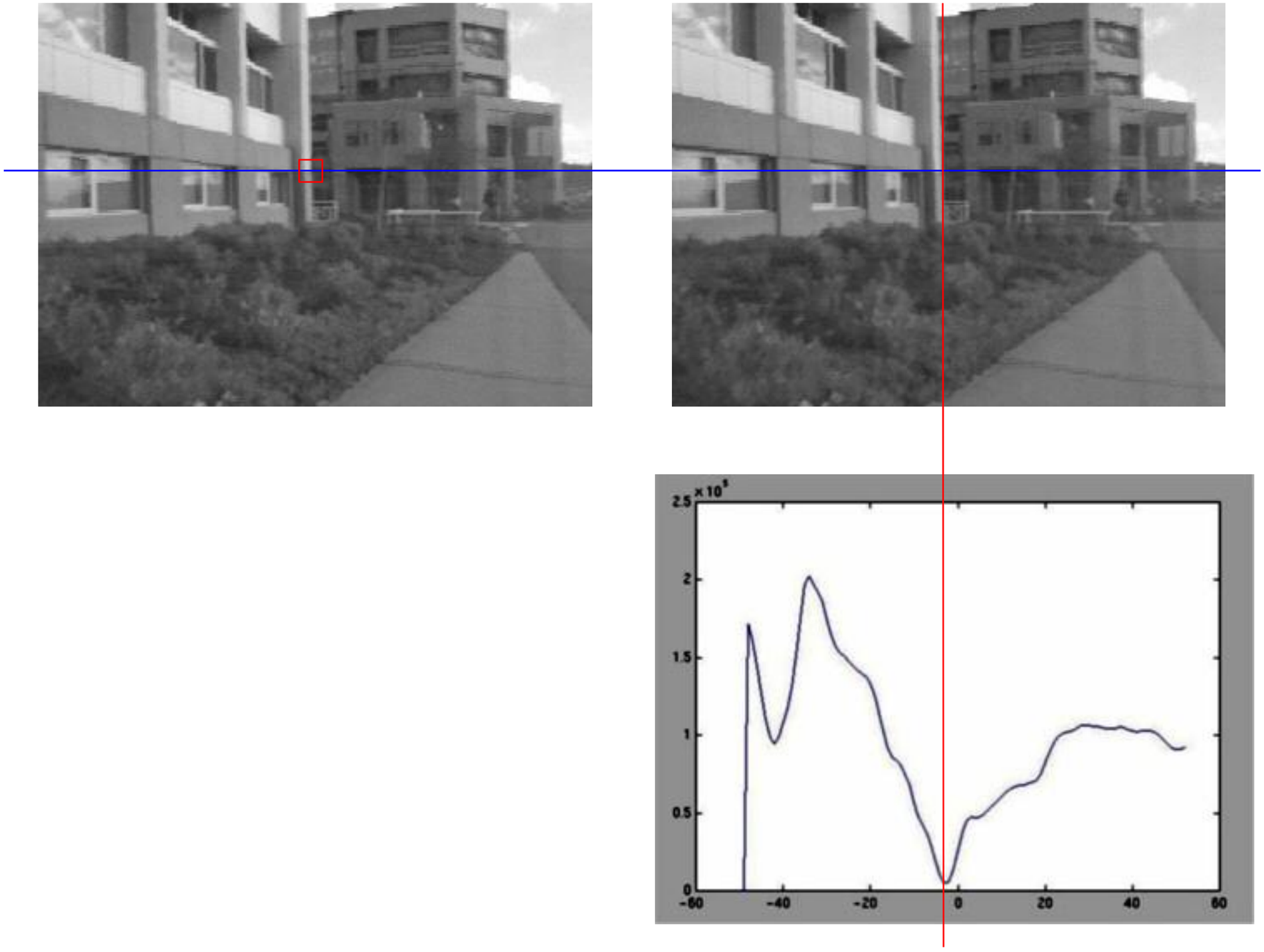
NCC

Ground truth

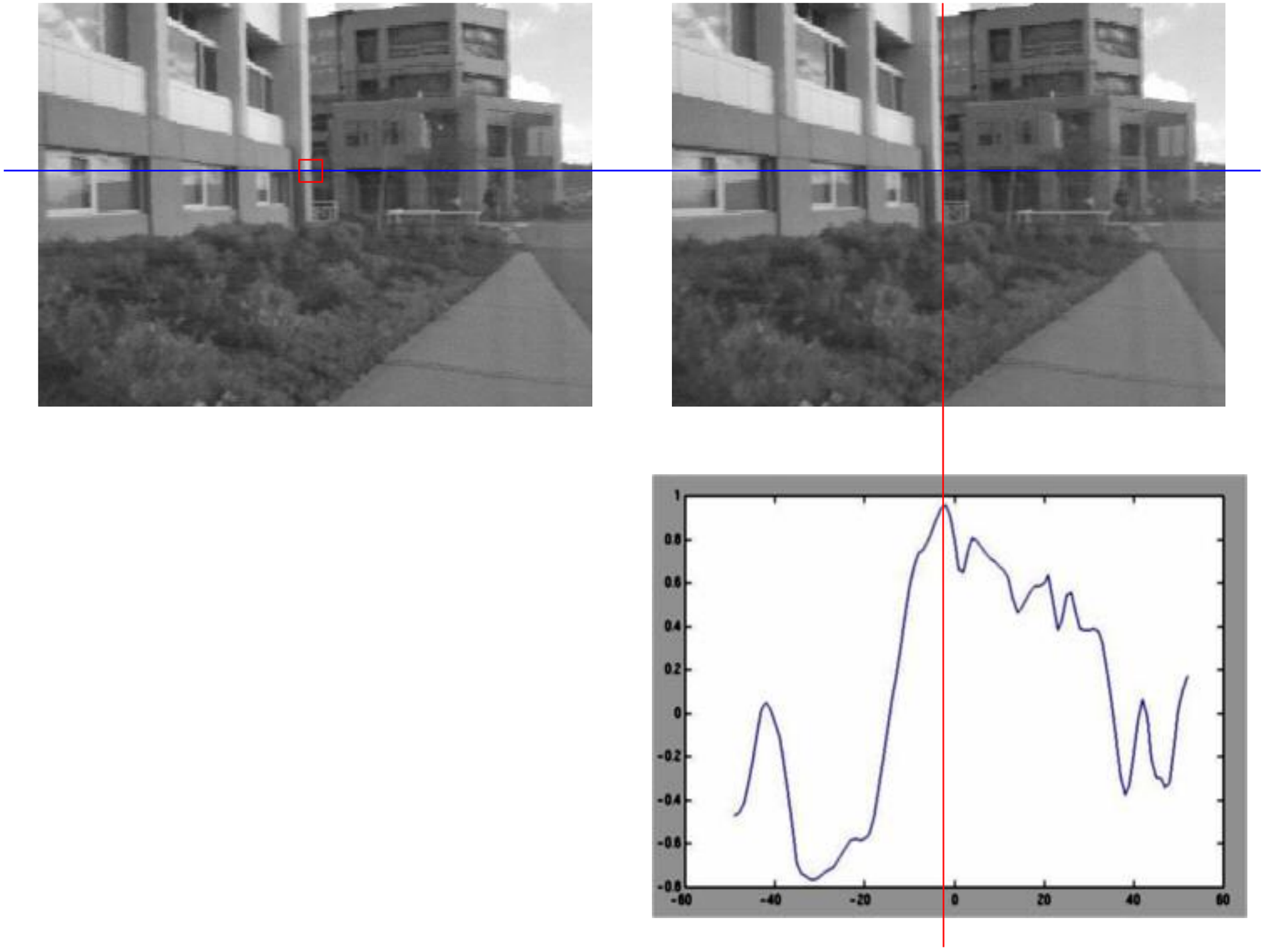
Stereo Block Matching



- Slide a window along the epipolar line (right)
- Compare contents of that window with the reference window (left)
- Matching cost: Sum of Squared Differences or Normalized Cross Correlation



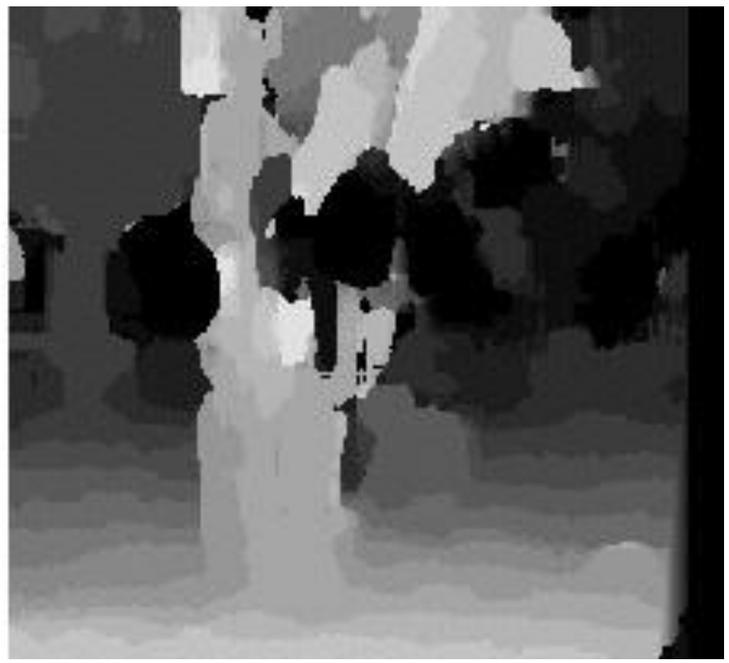
Sum of Squared Differences (SSD)



Normalized Cross Correlation (NCC)

Effect of window size

W = 3



W = 20

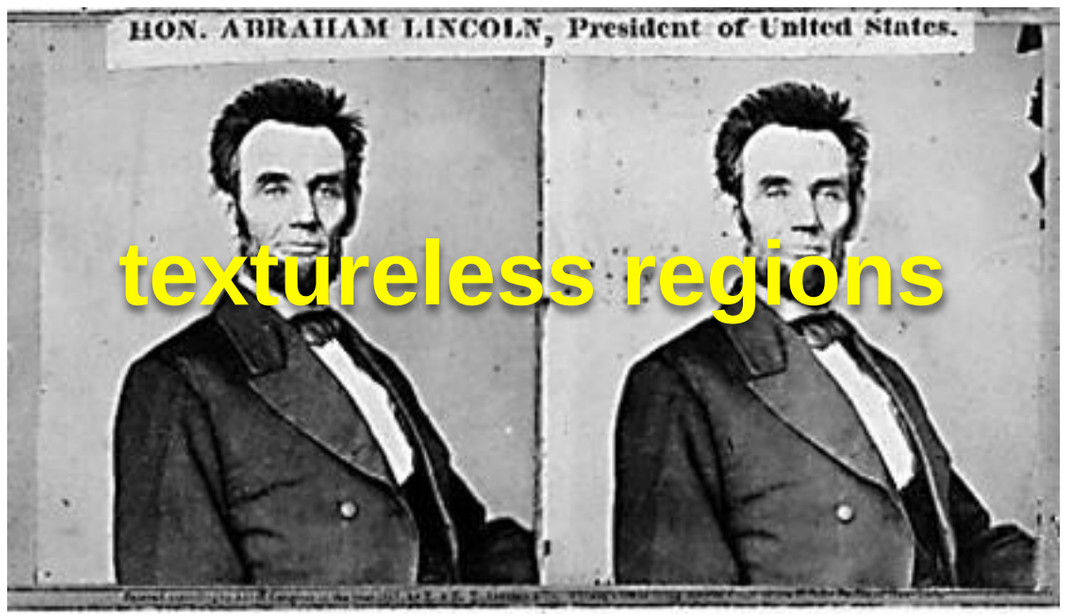
Smaller window

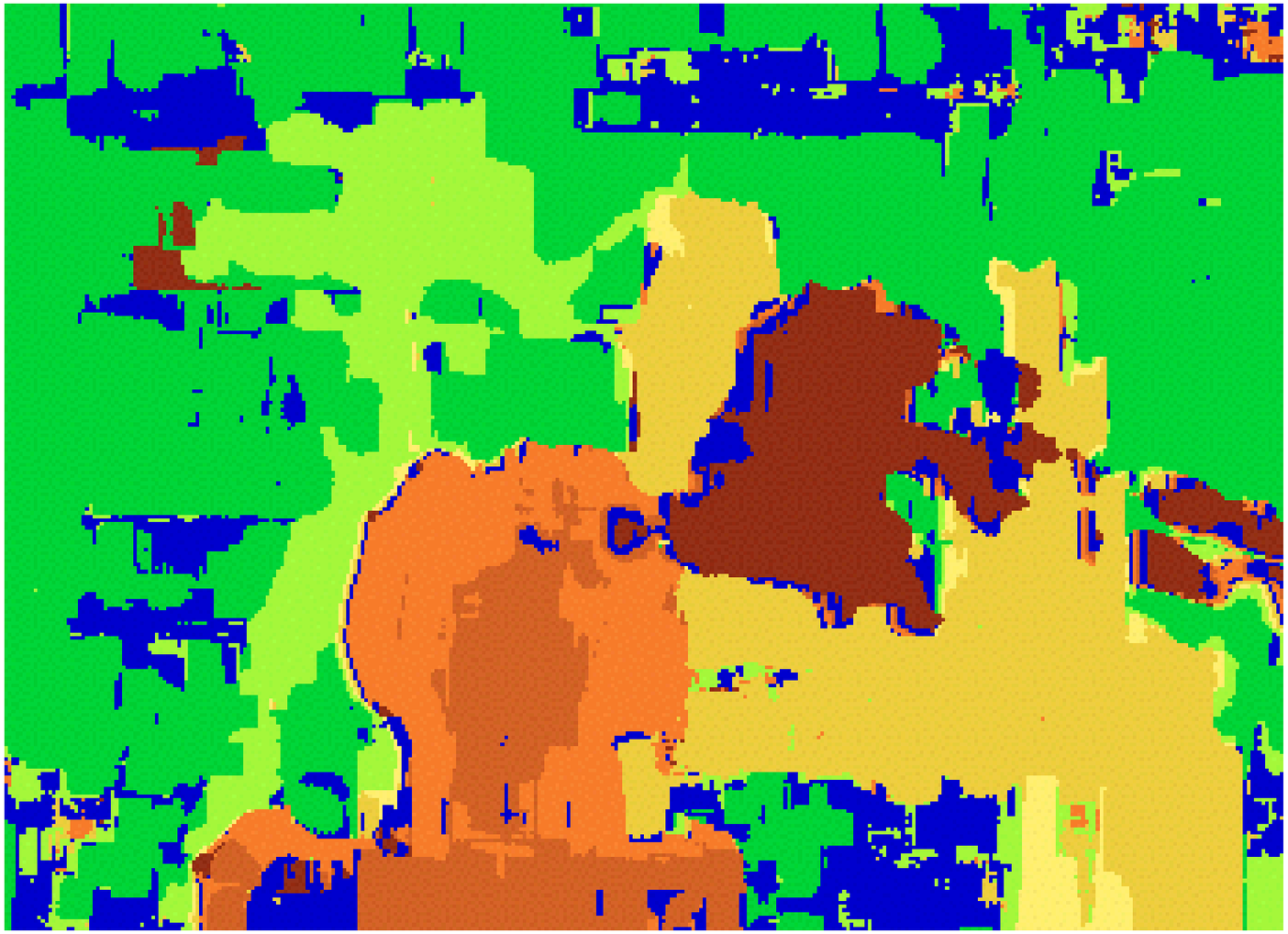
+ More detail
- More noise

Larger window

+ Smoother disparity maps
- Less detail
- Fails near boundaries

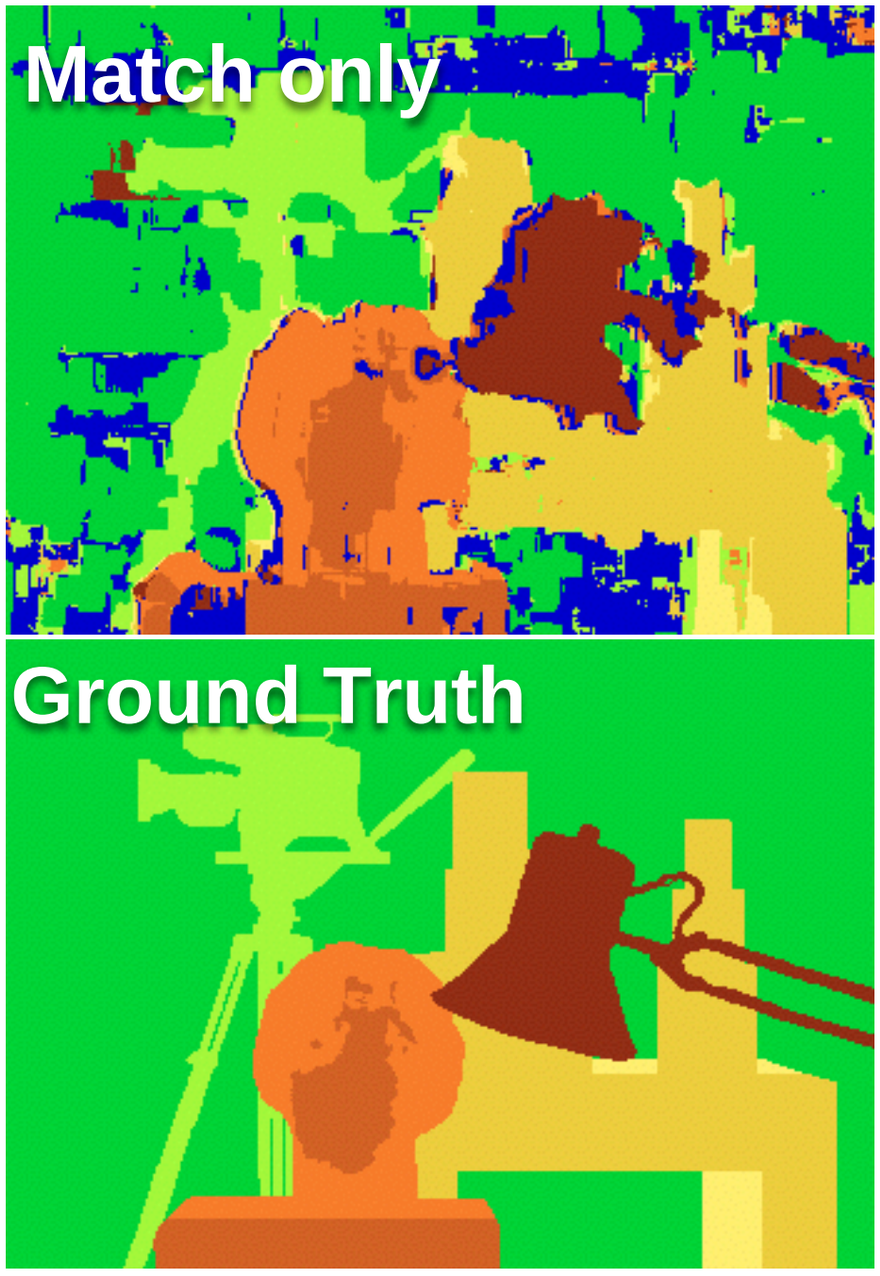
When will stereo block matching fail?

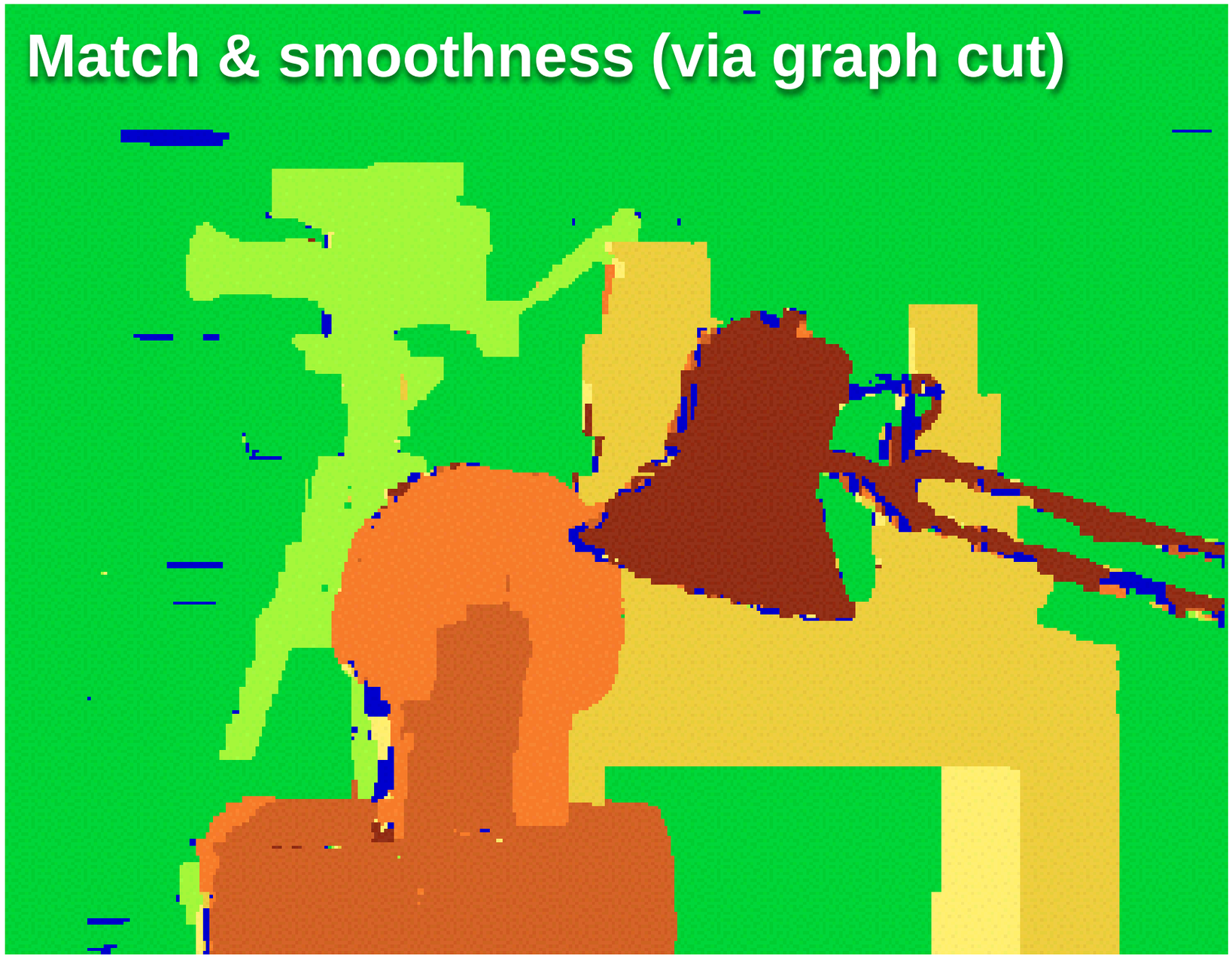




How can we improve depth estimation?

Too many discontinuities.
We expect disparity values to change slowly.

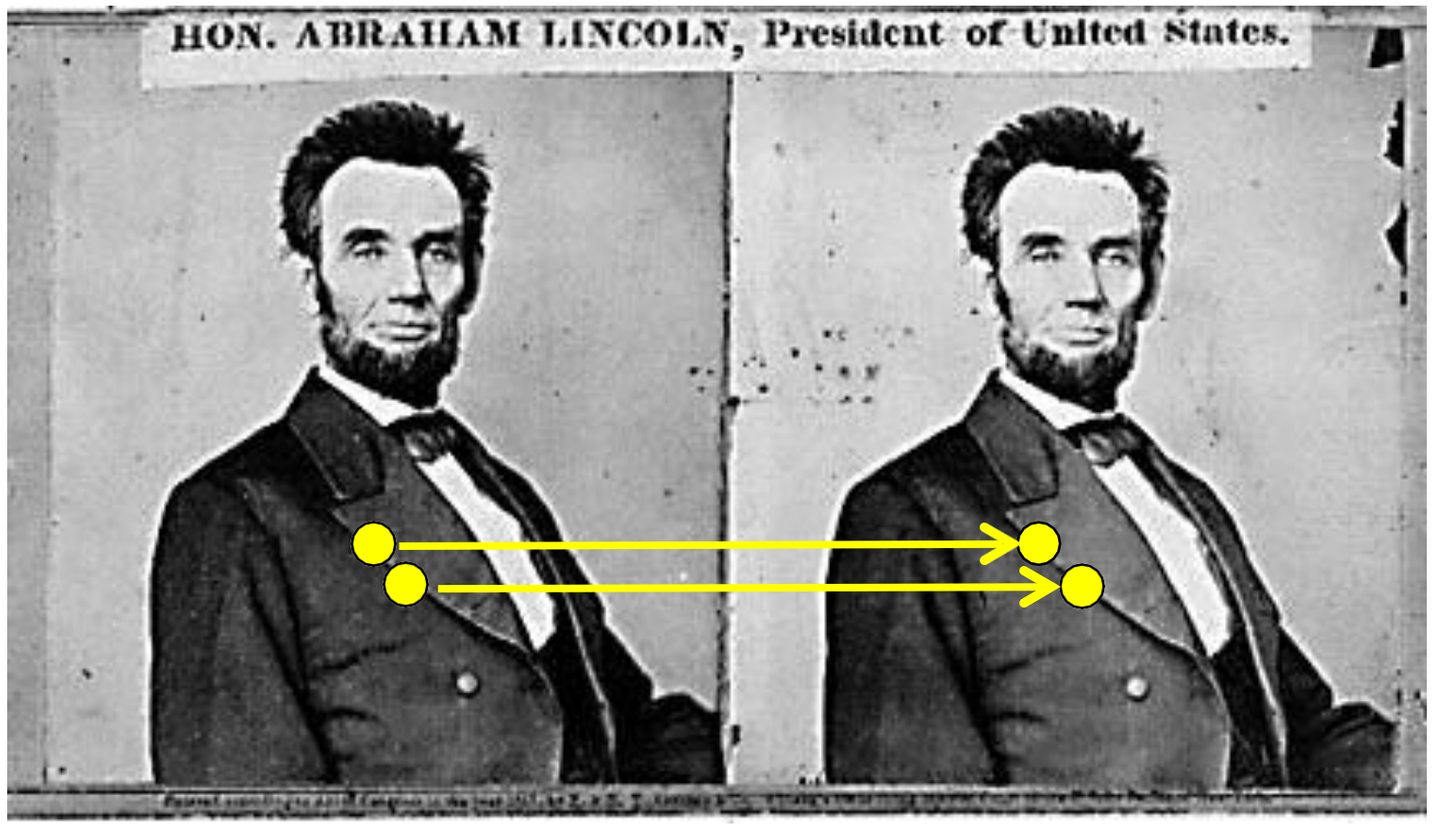




Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001

Stereo matching as …

Energy Minimization



What defines a good stereo correspondence?

1. **Match quality**
   - Want each pixel to find a good match in the other image
2. **Smoothness**
   - If two pixels are adjacent, they should (usually) move about the same amount

energy function
(for one pixel)

$$E(d) = \underbrace{E_d(d)}_{\text{data term}} + \underbrace{\lambda E_s(d)}_{\text{smoothness term}}$$

Want each pixel to find a good match in the other image

(block matching result)

Adjacent pixels should (usually) move about the same amount

(smoothness function)

$$E(d) = E_d(d) + \lambda E_s(d)$$

$$E_d(d) = \sum_{(x,y)\in I} C(x, y, d(x, y))$$

data term

SSD distance between windows
centered at I(x, y) and J(x+ d(x,y), y)

$$E(d) = E_d(d) + \lambda E_s(d)$$

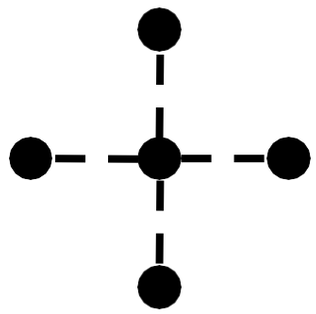
$$E_d(d) = \sum_{(x,y) \in I} C(x, y, d(x, y))$$

SSD distance between windows centered at I(x, y) and J(x+ d(x,y), y)

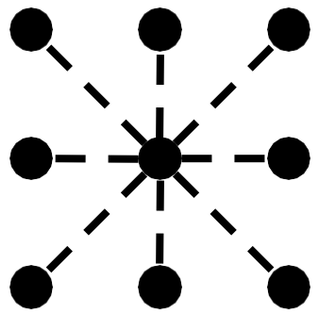
$$E_s(d) = \sum_{(p,q) \in \mathcal{E}} V(d_p, d_q)$$

smoothness term

$\mathcal{E}$: set of neighboring pixels



4–connected neighborhood



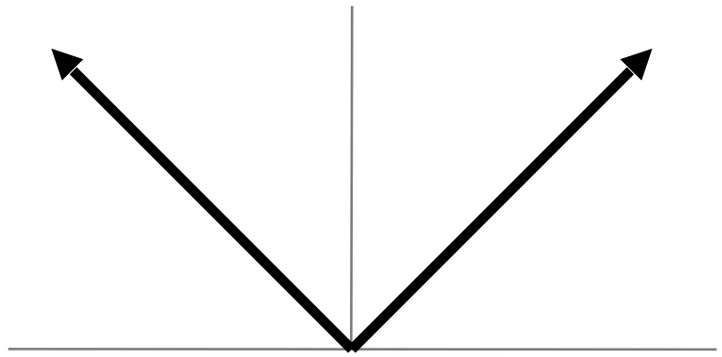
8–connected neighborhood

$$E_s(d) = \sum_{(p,q)\in\mathcal{E}} V(d_p, d_q)$$

smoothness term

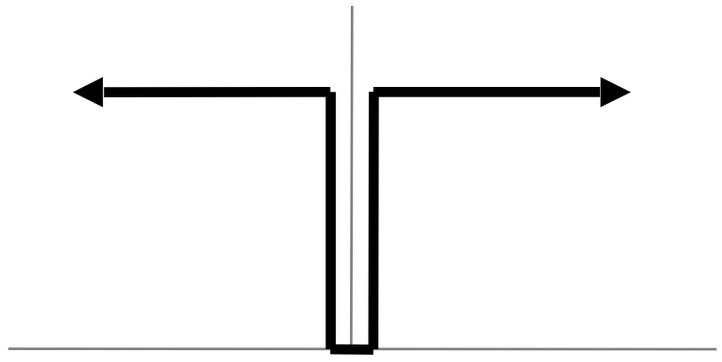
$$V(d_p, d_q) = |d_p - d_q|$$

L_1 distance

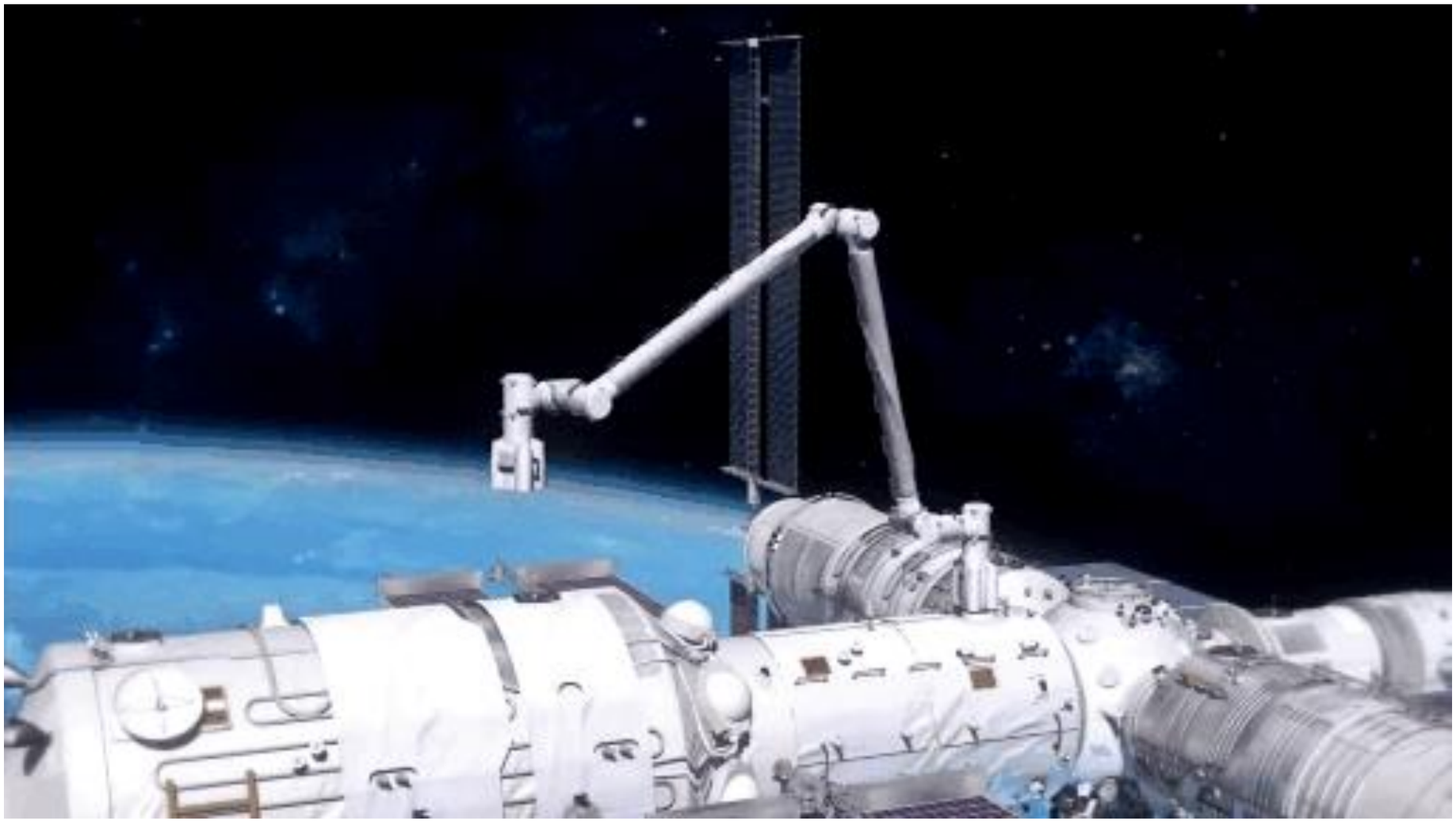


$$V(d_p, d_q) = \begin{cases} 0 & \text{if } d_p = d_q \\ 1 & \text{if } d_p \neq d_q \end{cases}$$

“Potts model”

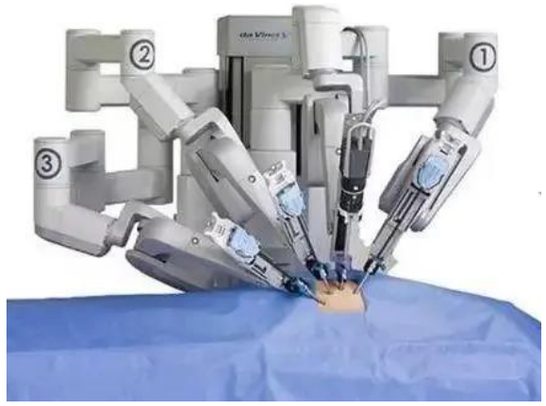


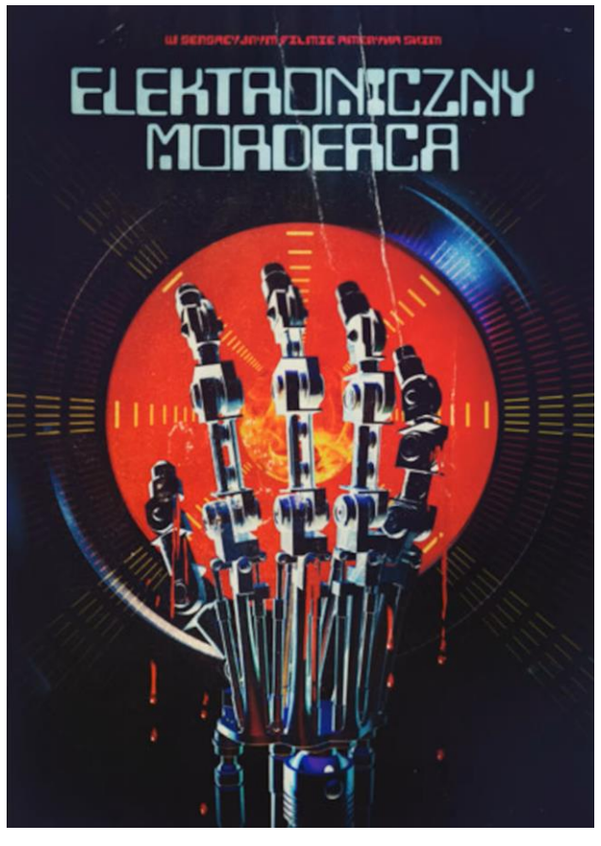
Stage 2 **Cognition & Action:** Intelligent Robot

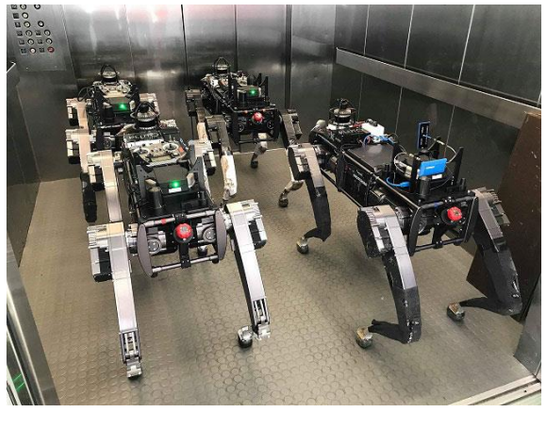


Variant kinds of Robotic Arms

Accuracy, Speed & Intelligence

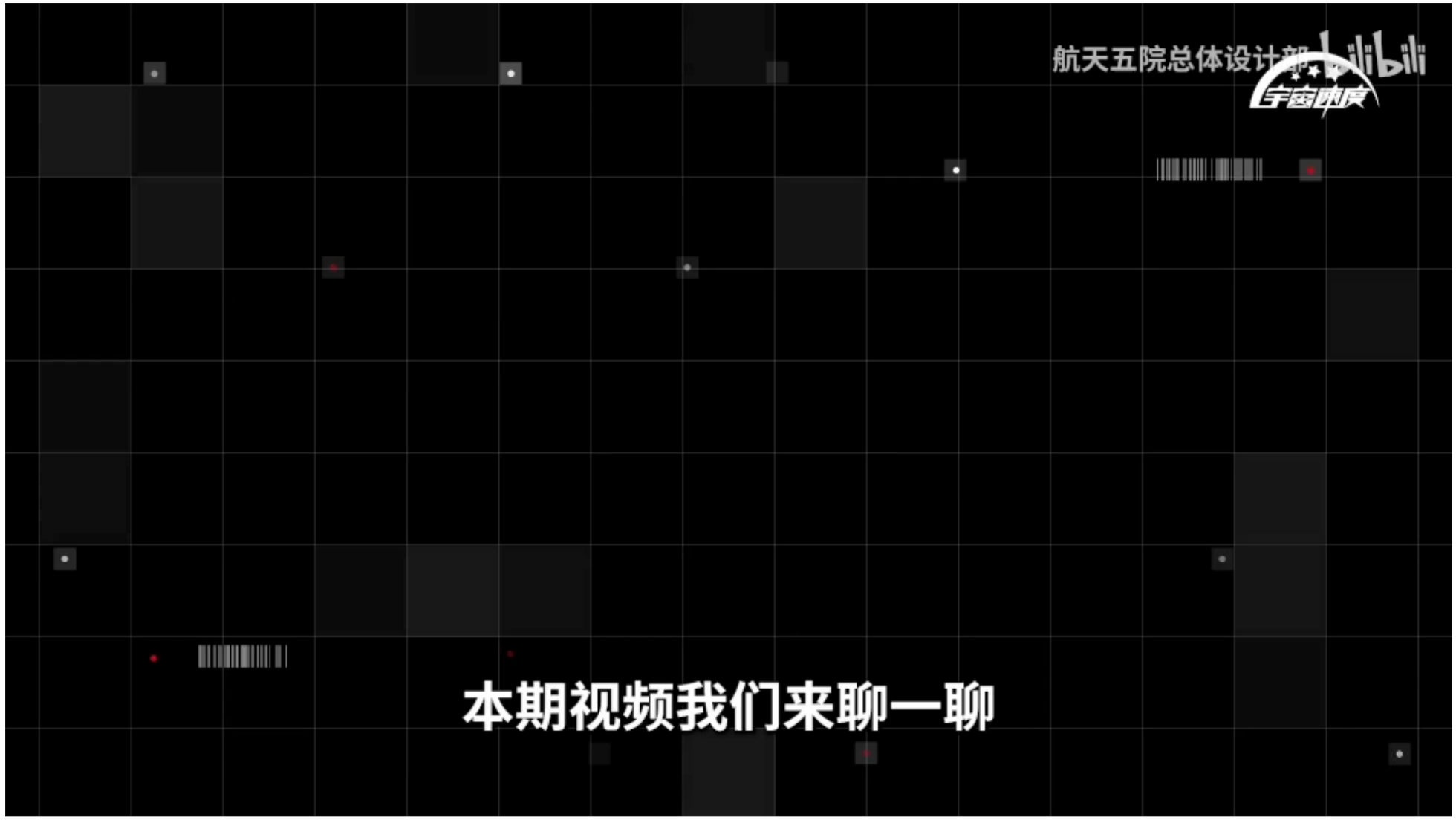






Tianhe Robotic Arm

- High precision requirements
- extreme environment: low temperature, communication efficiency



https://www.bilibili.com/video/BV1wY411p7MJ?spm_id_from=333.337.search-card.all.click&vd_source=8f294b7f2fbc721bf639fdb82d80ba08

Robotic Arms in a Bullet Production Line

- High precision requirements
- extreme environment: high temperature

Demand for Intelligent Robotic System

- From high efficiency and high accuracy, to the pursuit of high intelligence
- Adapt and respond to more complex scenarios and tasks

Where is the gap?

- Need of an intelligent brain
  - Multimedia: RGB image, Depth sensor, Language instruction, …
  - Cognition: Task understanding, planning and decision making.

Rapid developing --- Dynamic Robotics

Merge the scenario **cognition** with **intelligent robot**

Purpose: Deploy the depth camera on the robotic arm to form a complete system, and design the program to manipulate a robotic arm to complete the task of object sorting.

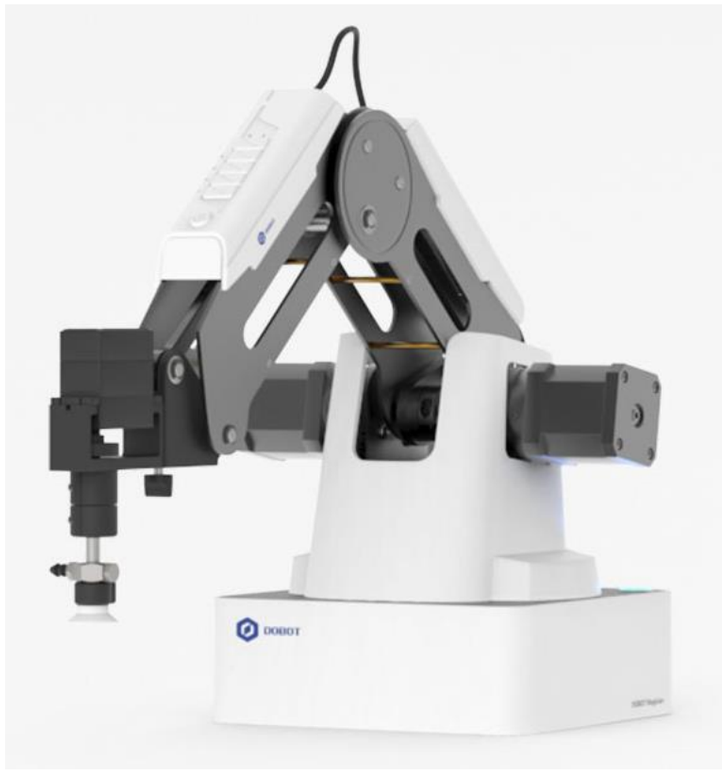
Requirements:

1. **Localize objects using RGB and depth image.**
    1. RGB image: recognize the target object with specific properties.
    2. Depth image: locate the position of the target object.
1. **Control the robotic arm to sort the target objects.**
    1. Control the robotic arm to reach the 3D position.
    2. Catch the object to the specified target location.
2. **Evaluation**
    1. Accuracy, efficiency(speed), robustness(extreme cases), etc.

Work in teams of 3 to 4 students.

End-of-term presentations and assessments.

Detail of the Robotic Arm (which will be used)

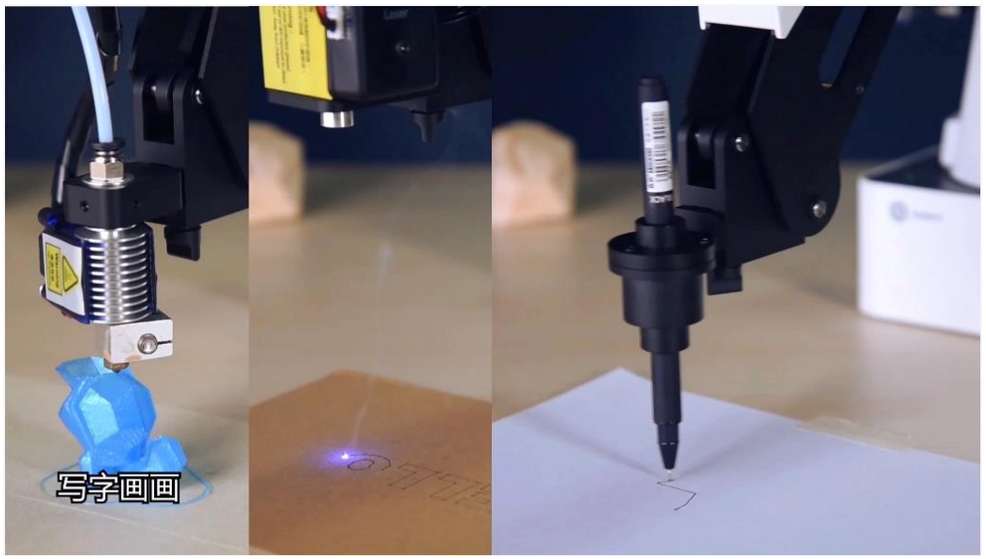


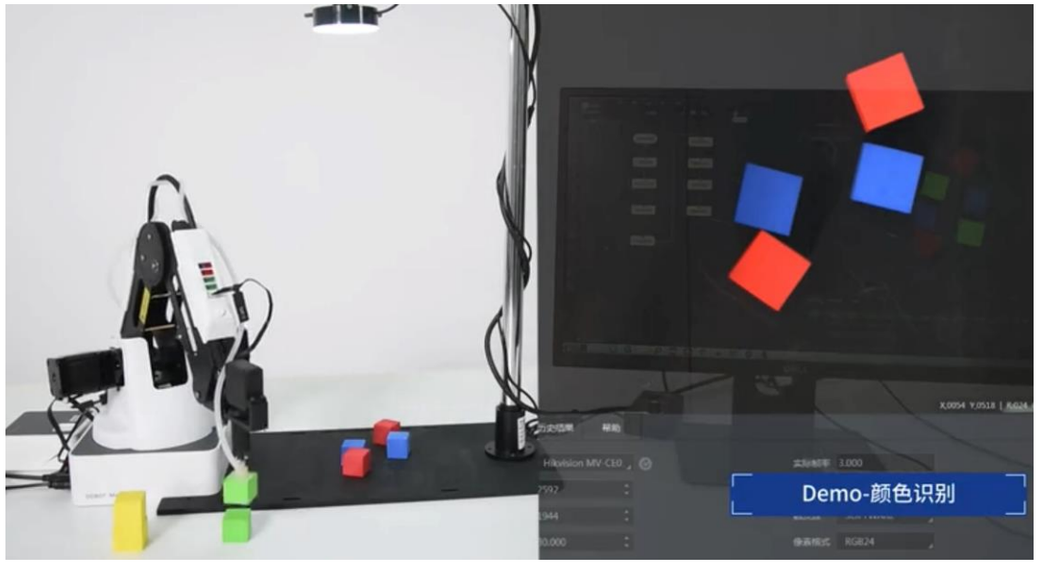
Desktop robotic arm

- 4 degrees of freedom

Axis	Range	Max Speed (250g workload)
Joint 1 base	-120° ~+120°	320° / s
Joint 2 rear arm	-5° ~+90°	320° / s
Joint 3 forearm	-15° ~+90°	320° / s
Joint 4 rotation servo	-140° ~+140°	480° / s

- Development Platform：Python、C++





https://www.dobot.cn/products/education/magician.html

Sample working scenarios

Lego toys **stretching & classification**,
based on their spatial position, shape, scale, or color.

https://www.lego.com/en-us/product/creative-building-bricks-11016

Good luck on the grand tour!

Q & A