

Computer Vision: Line detection

Siheng Chen 陈思衡

Announcement

1. Homework 2 is online; due on May 30th
2. Project group size: 1-4 members
3. Final presentation: Upload videos (5 mins)
4. Project team-up report; due on June 10th
5. We have another TA

Teaching Assistant: Yueyuan Li (李悦圆)

Ph.D student at Shanghai Jiao Tong University

- Email: li.yueyuan@outlook.com (Daily check at 9am)
- Tel: +86 13701856658
- Office hour: Fri, 19:00-21:00 or by appointment
- Feishu Room ID (Virtual): 978476156
- Hobby: reading, painting, video games
- Research interest:
 - Reinforcement learning
 - Self-driving decision-making
 - Traffic simulation



Line detection

Finding boundaries

Line parameterization

Hough transform

Deep Hough transform

Finding boundaries



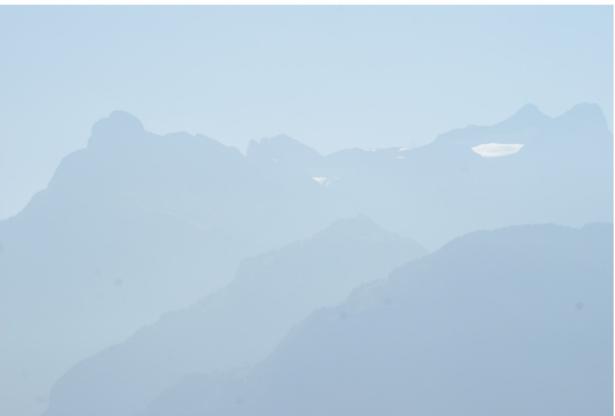
Finding boundaries



Finding boundaries



Finding boundaries



Finding boundaries

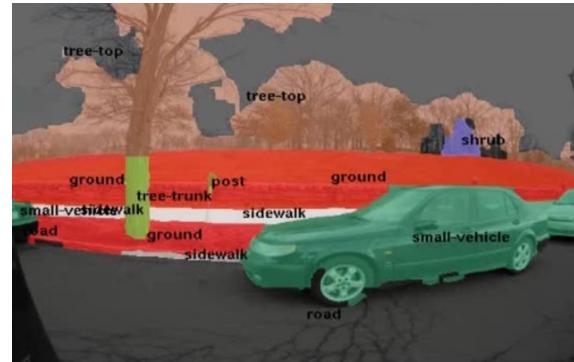


Defining boundaries is nontrivial!

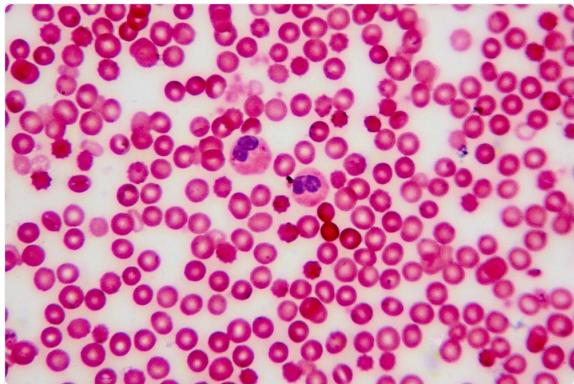
Finding boundaries



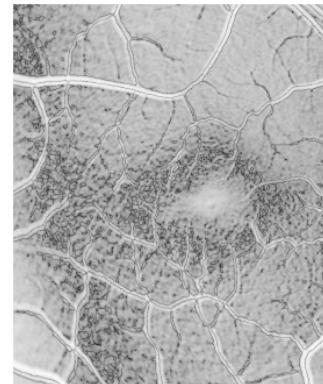
Lane detection



Scene segmentation



Cell counting



Blood vessel detection

Finding boundaries

Start from something very basic! Line detection



Finding boundaries

Otto, the self-driving truck company owned by Uber, recently completed what it claims was the first commercial delivery by a self-driving vehicle. OCTOBER 25, 2016



Finding boundaries

Lines can be disconnected



How to represent a line ?

Line detection

Finding boundaries

Line parameterization

Hough transform

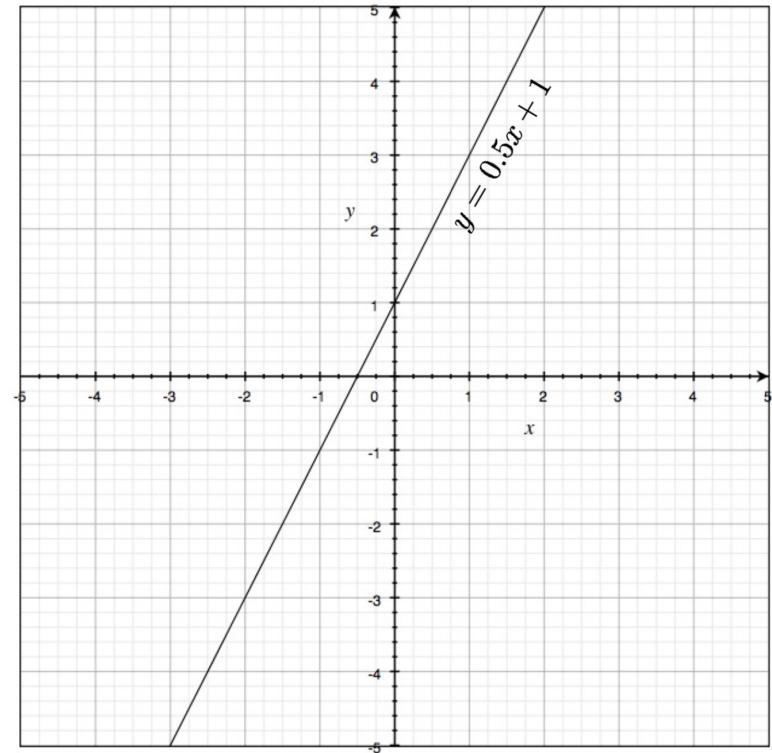
Deep Hough transform

Line parameterization

Slope intercept form

$$y = mx + b$$

↑ ↑
slope intercept



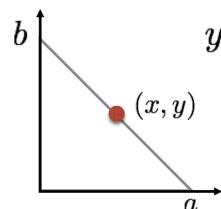
Line parameterization

Double intercept form

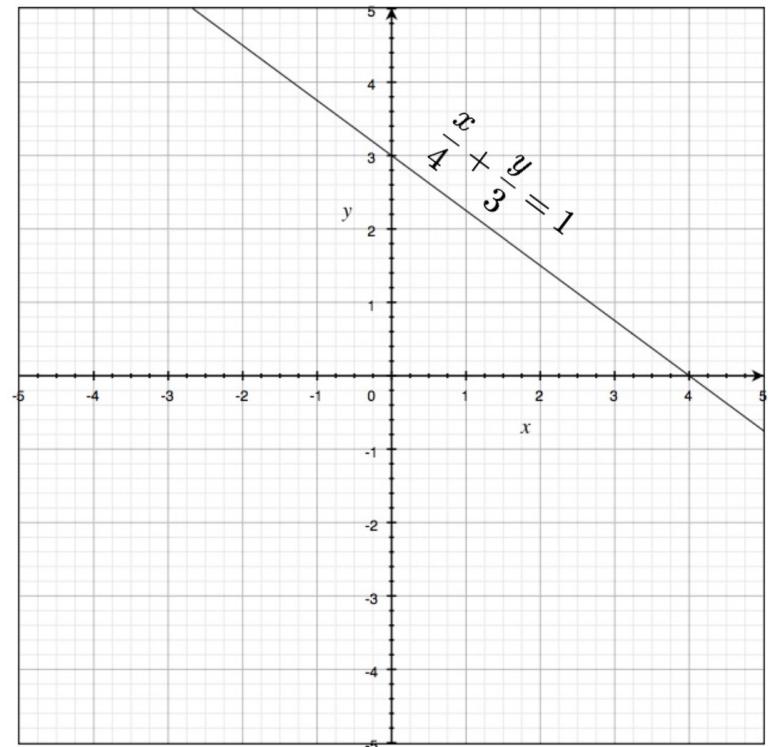
$$\frac{x}{a} + \frac{y}{b} = 1$$

 
x-intercept y-intercept

Derivation:



$$\begin{aligned} & \text{(Similar slope)} \quad \frac{y - b}{x - 0} = \frac{0 - y}{a - x} \\ & ya + yx - ba + bx = -yx \\ & ya + bx = ba \\ & \frac{y}{b} + \frac{x}{a} = 1 \end{aligned}$$

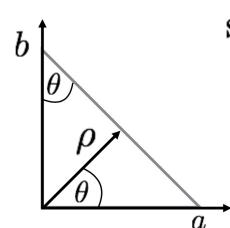


Line parameterization

Normal form

$$x \cos \theta + y \sin \theta = \rho$$

Derivation:

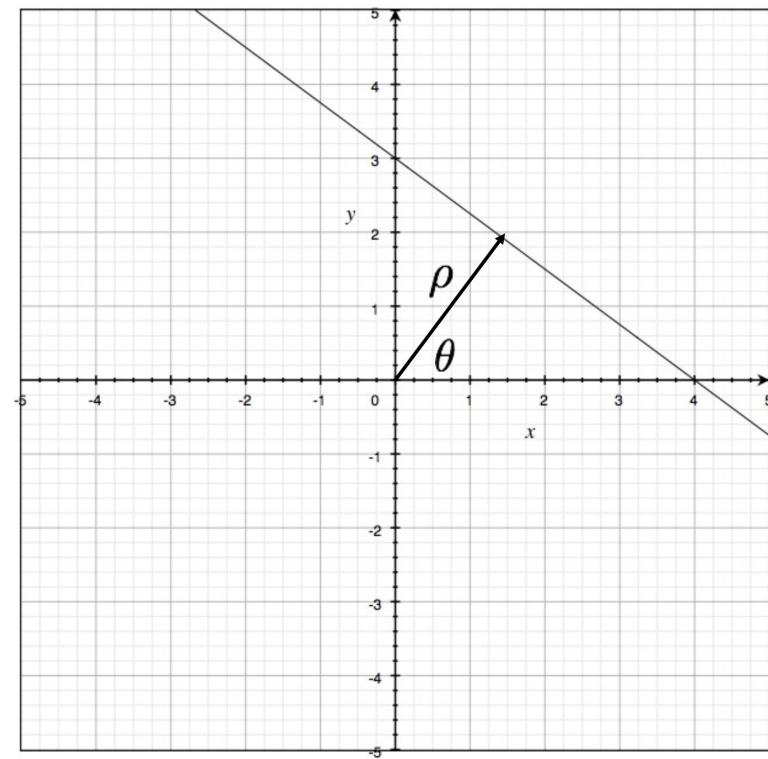


$$\cos \theta = \frac{\rho}{a} \rightarrow a = \frac{\rho}{\cos \theta}$$

$$\sin \theta = \frac{\rho}{b} \rightarrow b = \frac{\rho}{\sin \theta}$$

$$\text{plug into: } \frac{x}{a} + \frac{y}{b} = 1$$

$$x \cos \theta + y \sin \theta = \rho$$



Line detection

Finding boundaries

Line parameterization

Hough transform

Deep Hough transform

Hough transform

Lines can be disconnected



How to represent a line ?

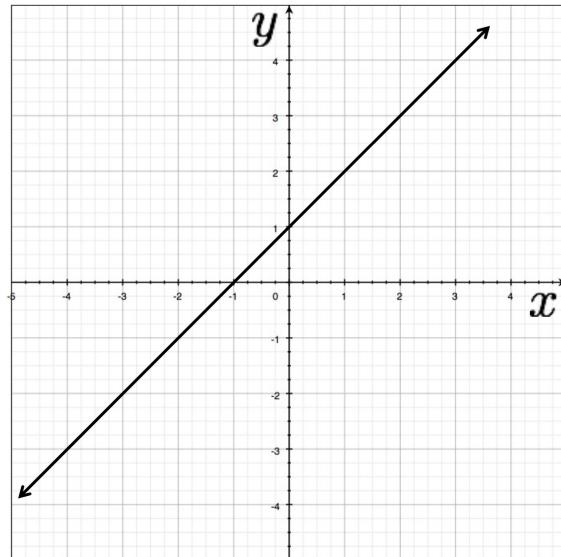
Hough transform

Dual representation

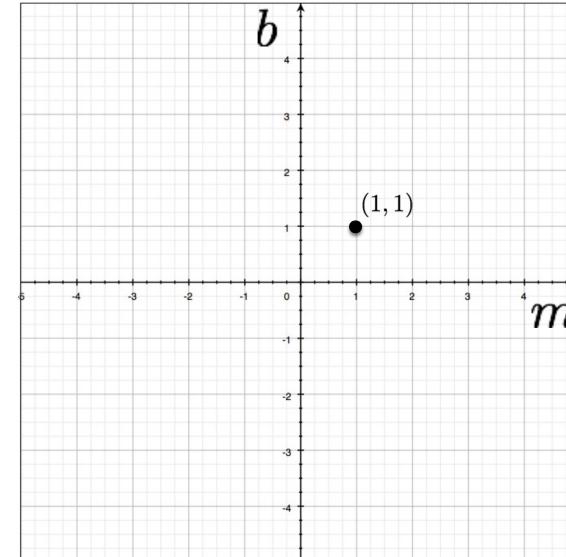
$$y = mx + b$$

parameters

data



data space



parameter space

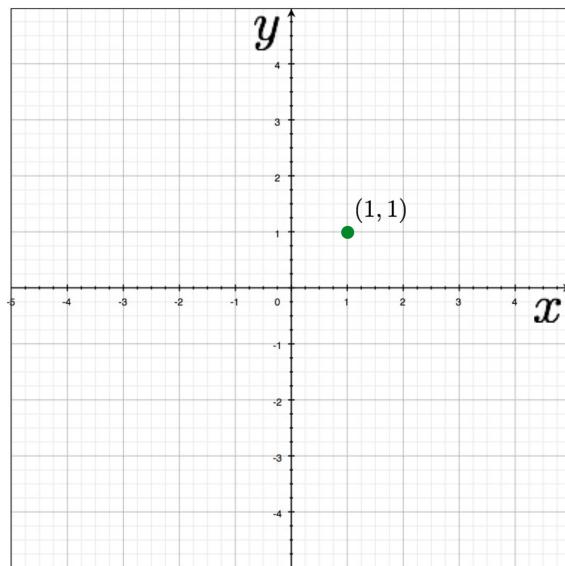
Hough transform

Dual representation

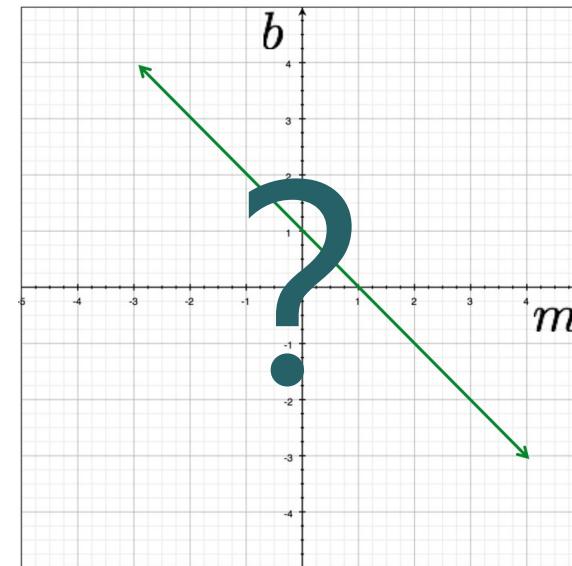
$$y = mx + b$$

parameters

data



data space



parameter space

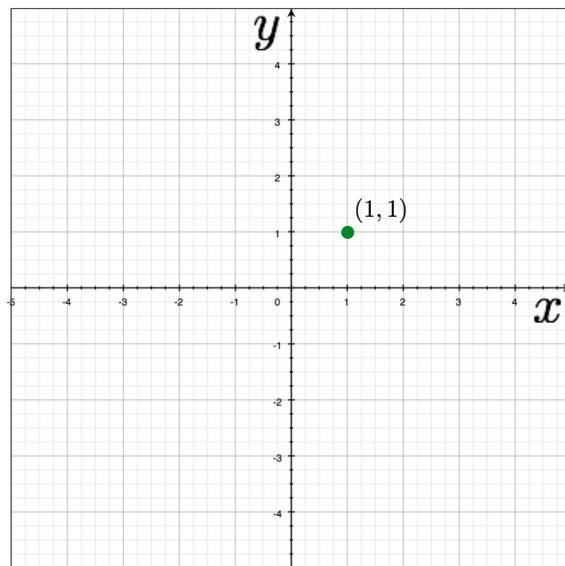
Hough transform

Dual representation

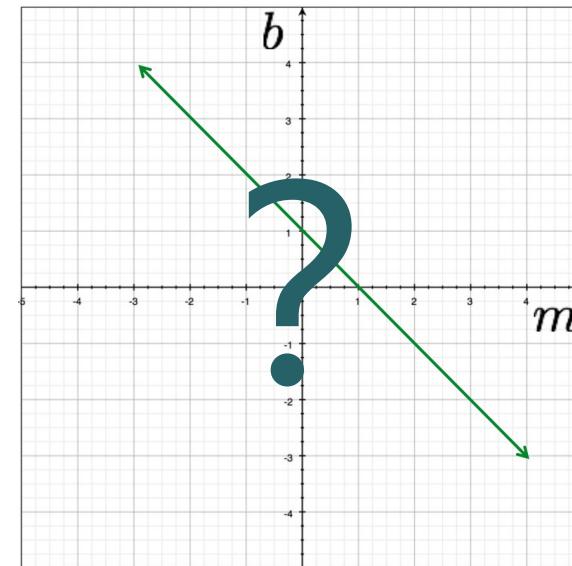
$$y = mx + b$$

parameters

data



data space



parameter space

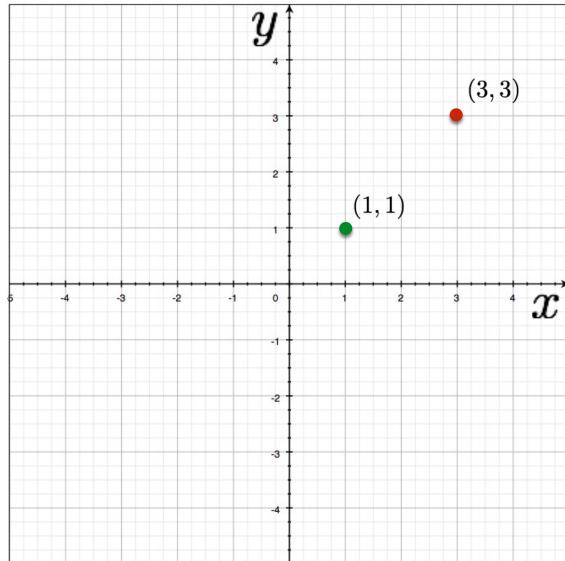
Hough transform

Dual representation

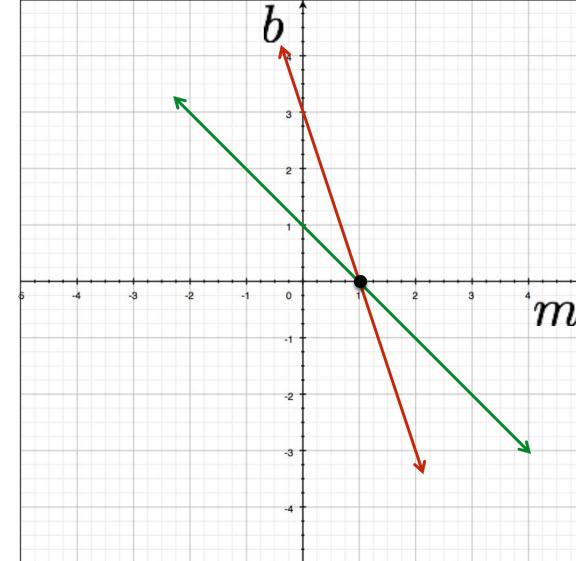
$$y = mx + b$$

parameters

data



data space



parameter space

Hough transform

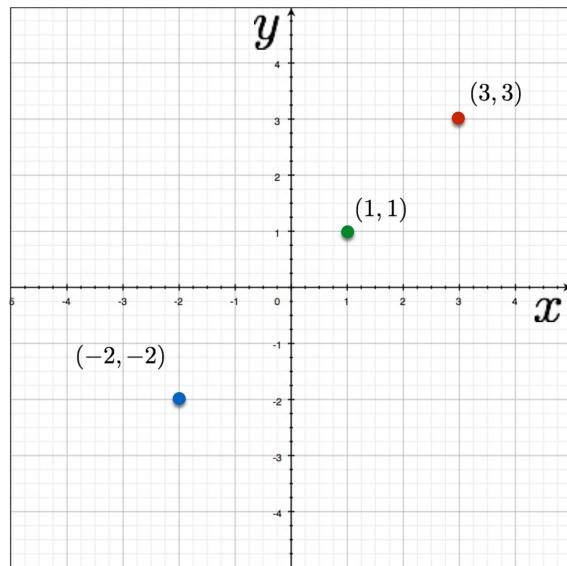
Dual representation

parameters

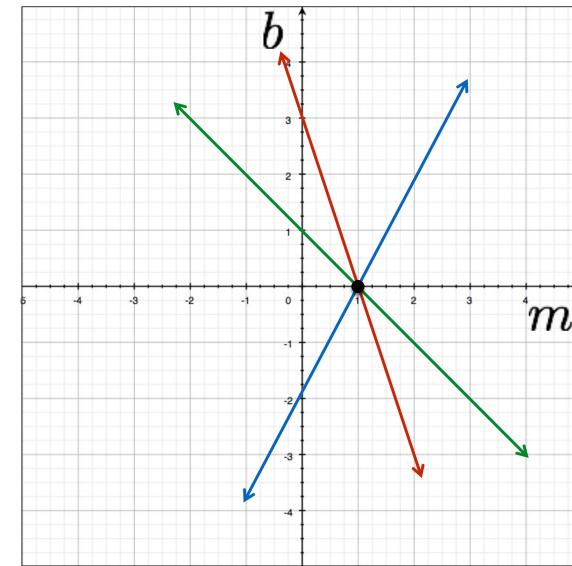
$$y = mx + b$$

data

robust to outliers?



data space



parameter space

Hough transform

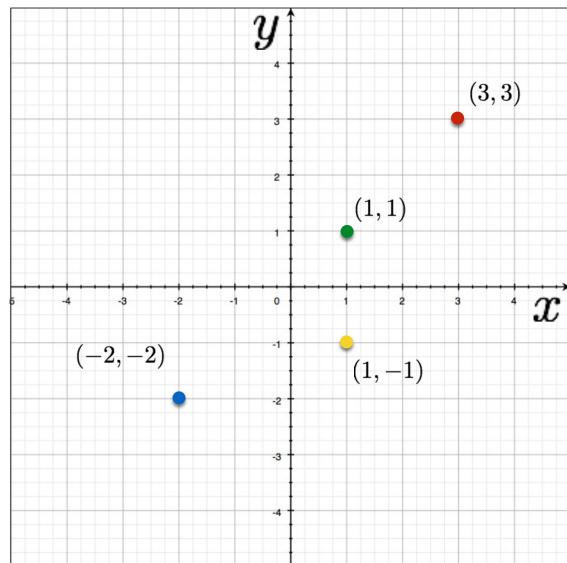
Dual representation

$$y = mx + b$$

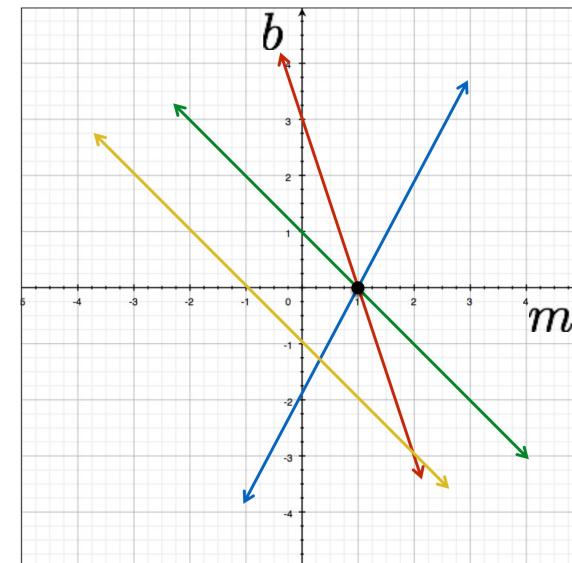
parameters

data

robust to outliers?



data space



parameter space

Hough transform

Lines can be disconnected



Each pixel would contribute one line

Hough transform

Algorithm:

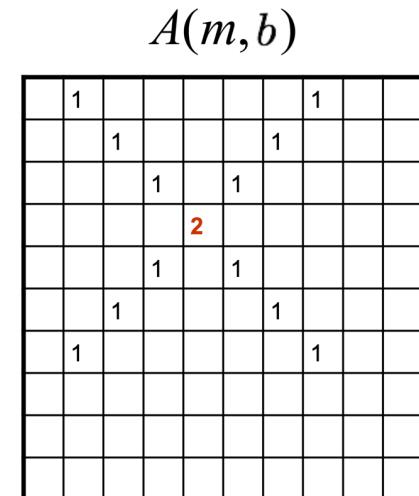
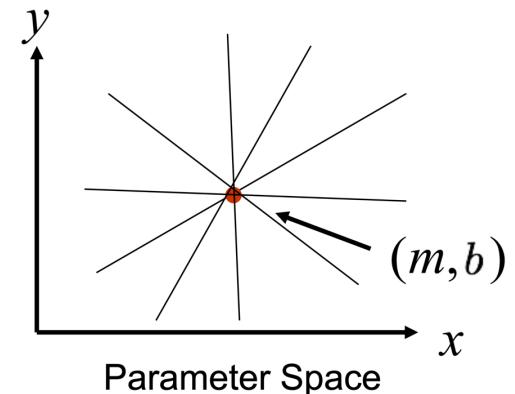
1. Quantize Parameter Space (m, b)
 2. Create Accumulator Array $A(m, b)$
 3. Set $A(m, b) = 0 \quad \forall m, b$
 4. For each image edge (x_i, y_i)
For each element in $A(m, b)$
If (m, b) lies on the line:
Increment $A(m, b) = A(m, b) + 1$
 5. Find local maxima in $A(m, b)$

What is the size of parameter size?

$$-\infty \leq m \leq \infty$$

$$-\infty \leq b \leq \infty$$

Infeasible to implement



Hough transform

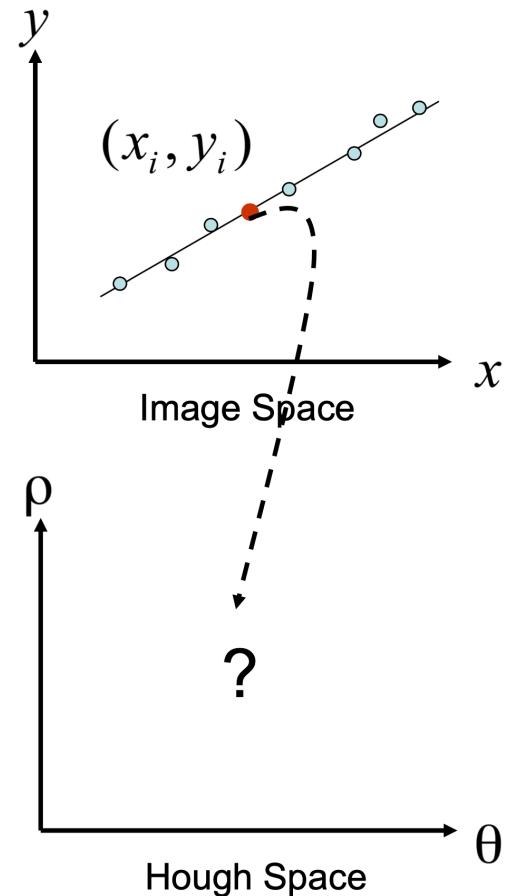
Consider normal form

$$x \cos \theta + y \sin \theta = \rho$$

Given points (x_i, y_i) find (ρ, θ)

$$0 \leq \theta \leq 2\pi$$

$$0 \leq \rho \leq \rho_{\max}$$

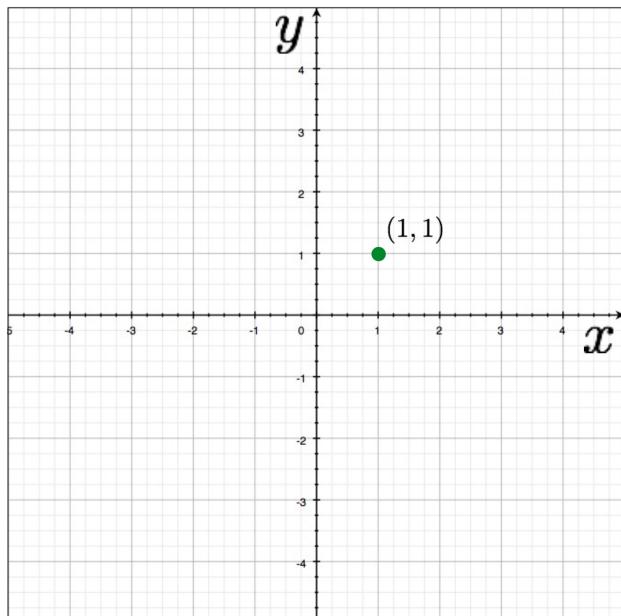


Hough transform

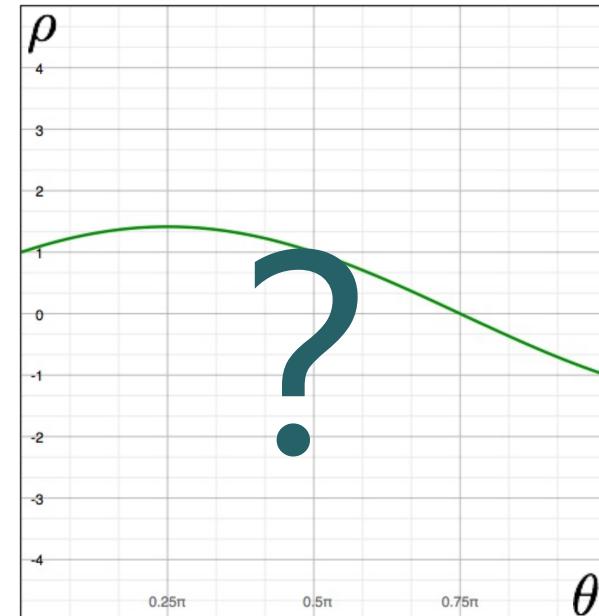
Consider normal form

$$x \cos \theta + y \sin \theta = \rho$$

parameters
↓
↑ data



data space



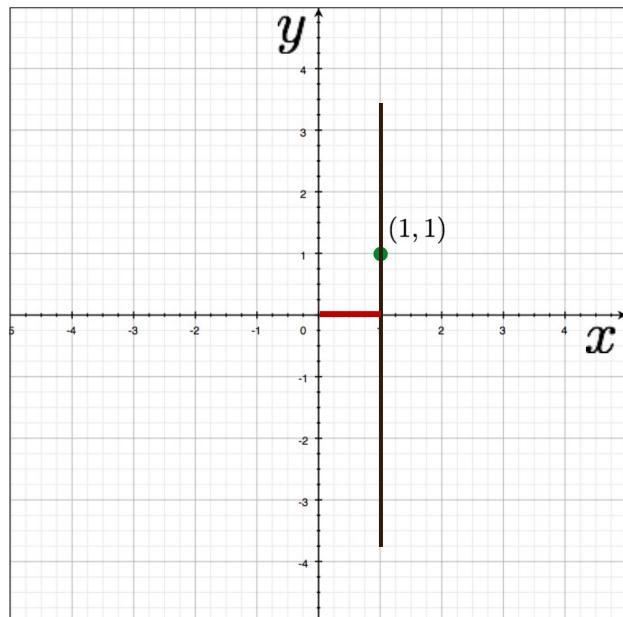
parameter space

Hough transform

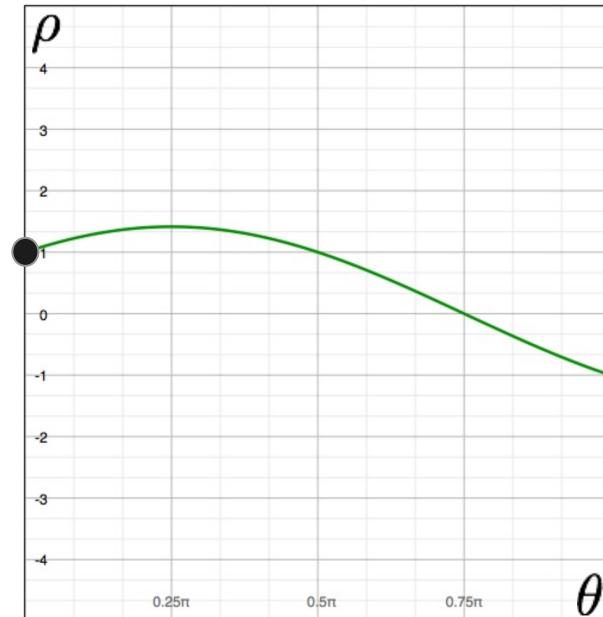
Consider normal form

$$x \cos \theta + y \sin \theta = \rho$$

parameters
data



data space



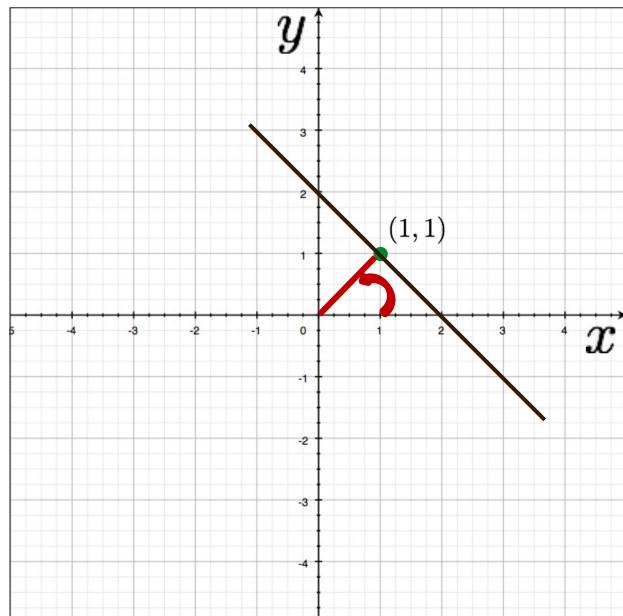
parameter space

Hough transform

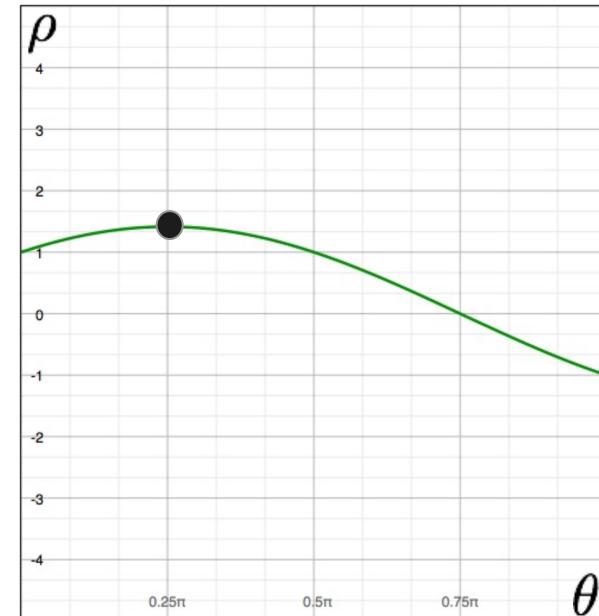
Consider normal form

$$x \cos \theta + y \sin \theta = \rho$$

parameters
↓
↑ data



data space



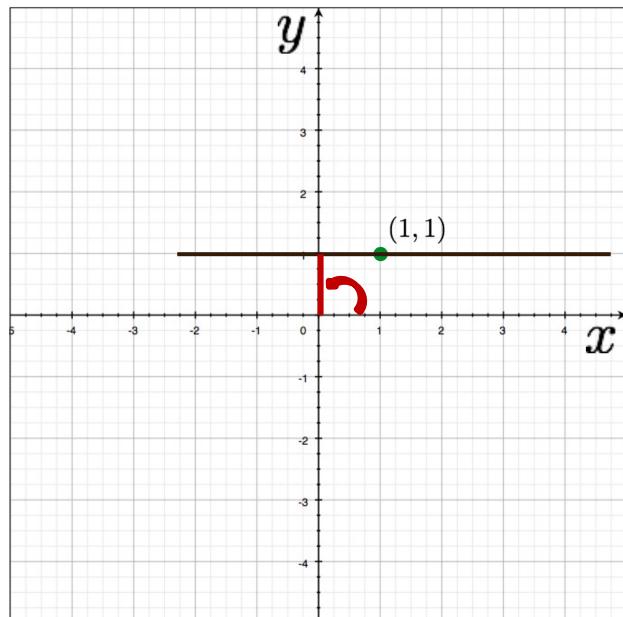
parameter space

Hough transform

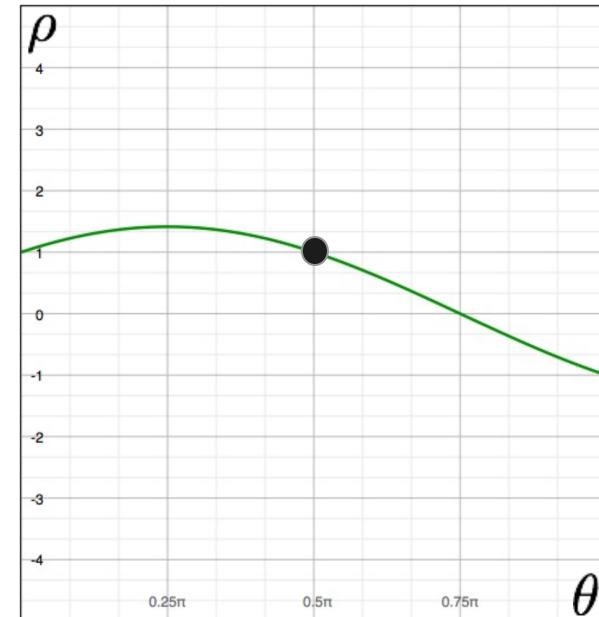
Consider normal form

$$x \cos \theta + y \sin \theta = \rho$$

parameters
↓
↓
data ↓



data space



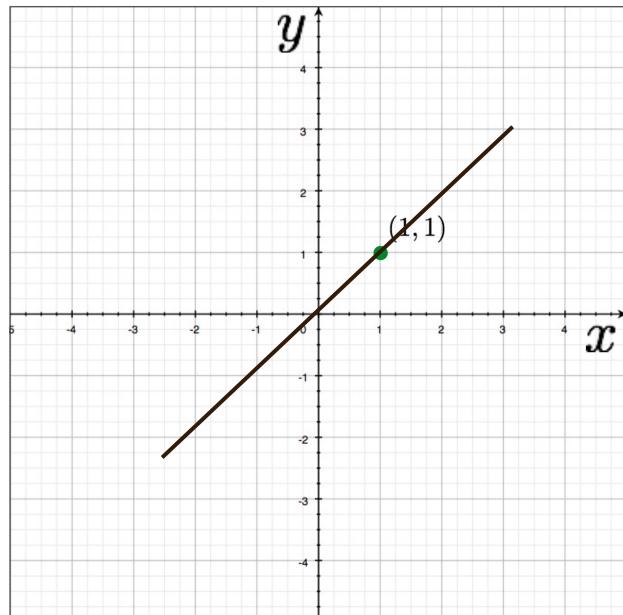
parameter space

Hough transform

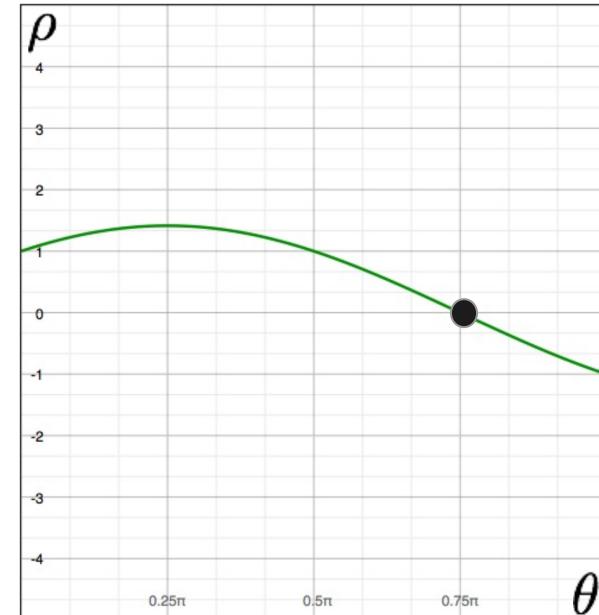
Consider normal form

$$x \cos \theta + y \sin \theta = \rho$$

parameters
↓
↑ data



data space



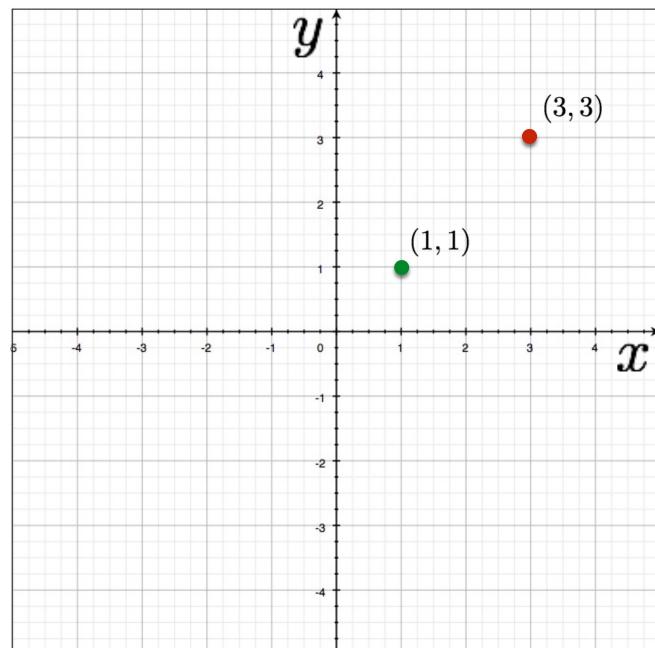
parameter space

Hough transform

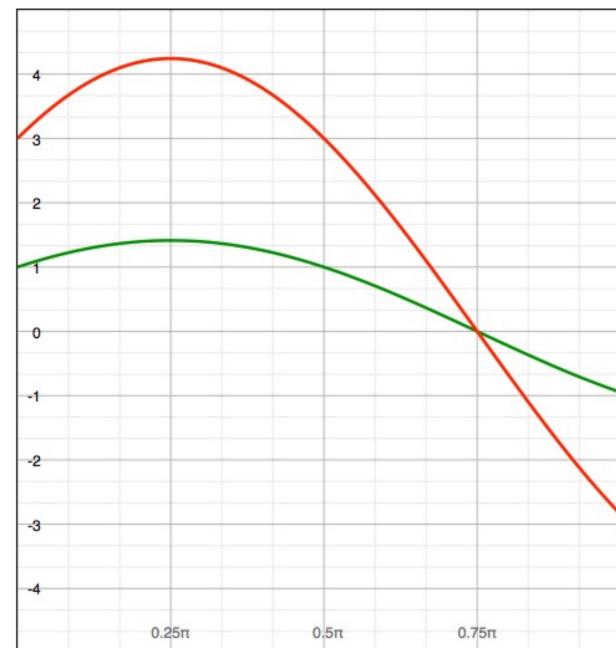
Consider normal form

$$x \cos \theta + y \sin \theta = \rho$$

parameters
↓
data



data space



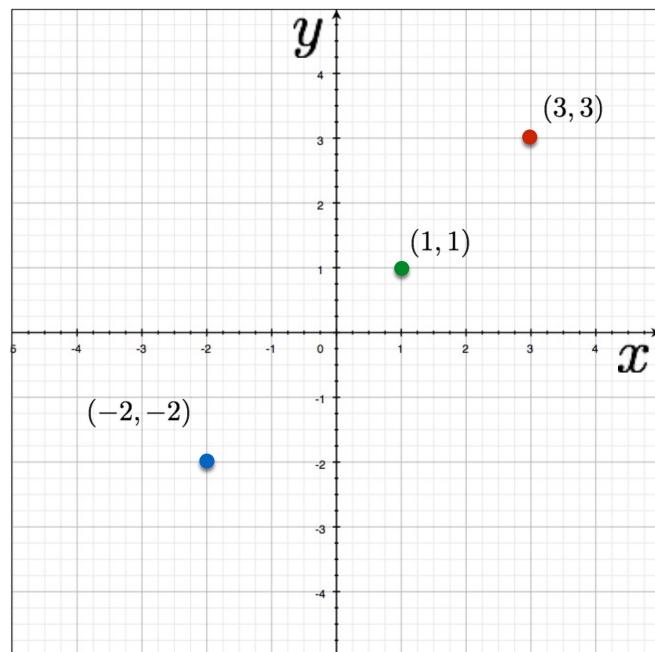
parameter space

Hough transform

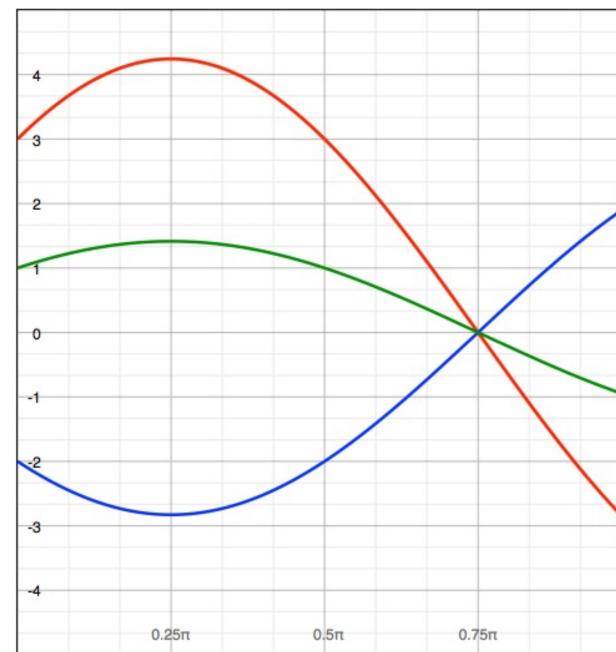
Consider normal form

$$x \cos \theta + y \sin \theta = \rho$$

parameters
↓
data



data space



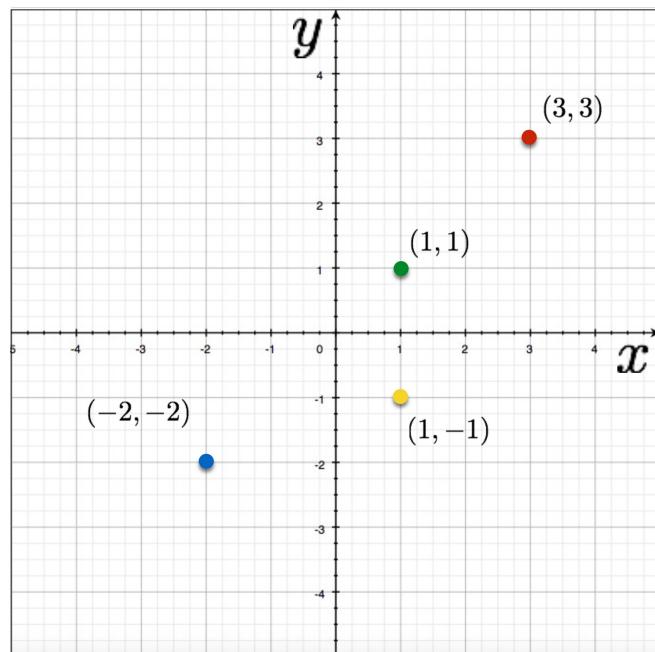
parameter space

Hough transform

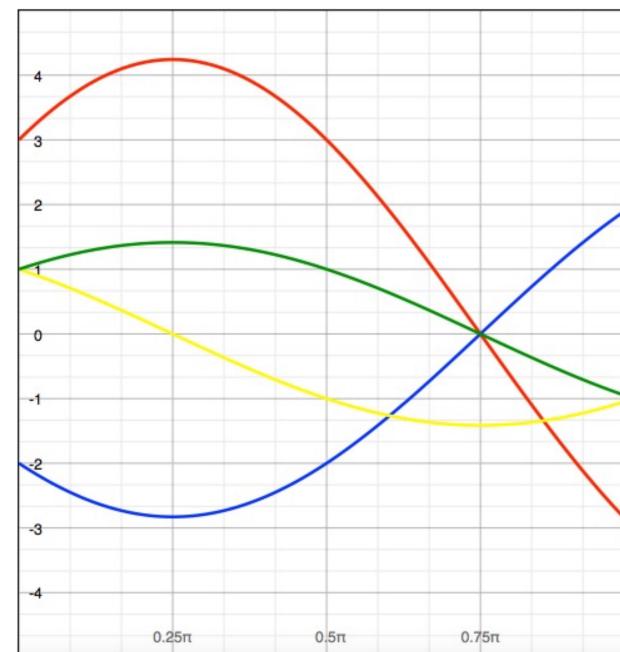
Consider normal form

$$x \cos \theta + y \sin \theta = \rho$$

parameters
↓
↓
data →



data space

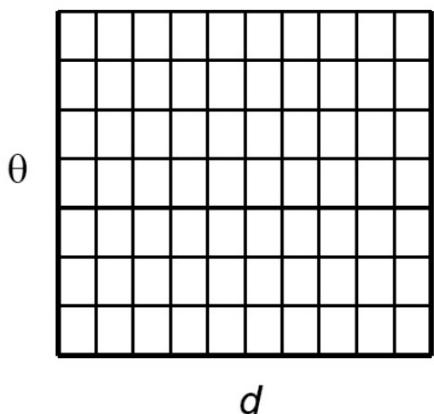


parameter space

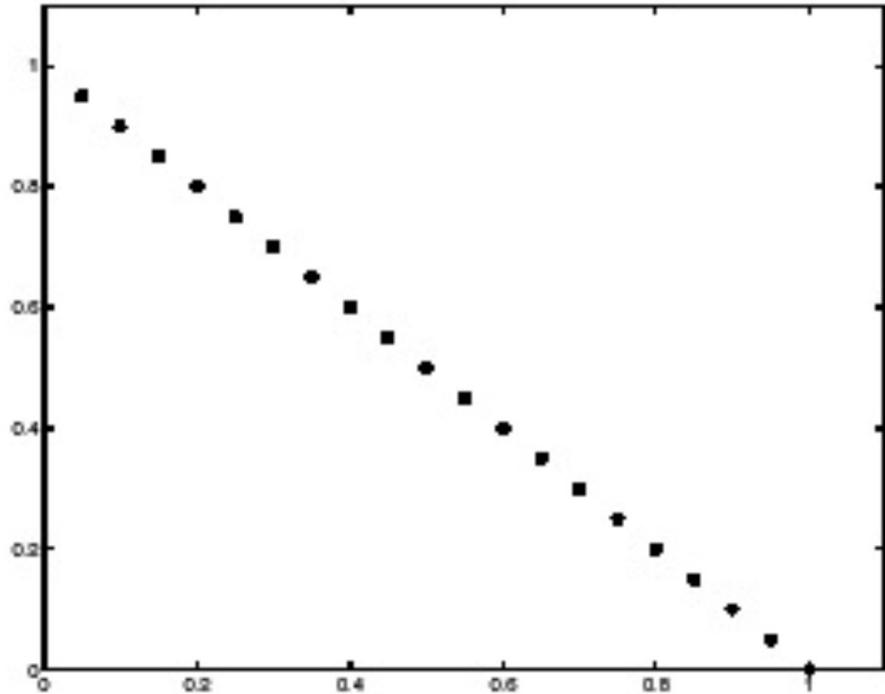
Hough transform

1. Initialize accumulator H to all zeros
2. For each edge point (x, y) in the image
 For $\theta = 0$ to 180
 $\rho = x \cos \theta + y \sin \theta$
 $H(\theta, \rho) = H(\theta, \rho) + 1$
 end
end
3. Find the value(s) of (θ, ρ) where $H(\theta, \rho)$ is a local maximum
4. The detected line in the image is given by
$$\rho = x \cos \theta + y \sin \theta$$

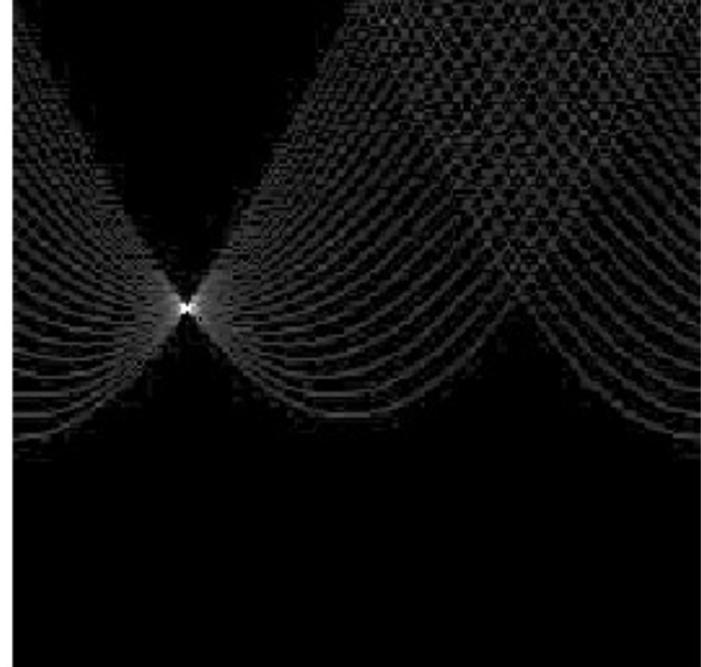
H : accumulator array (votes)



Hough transform

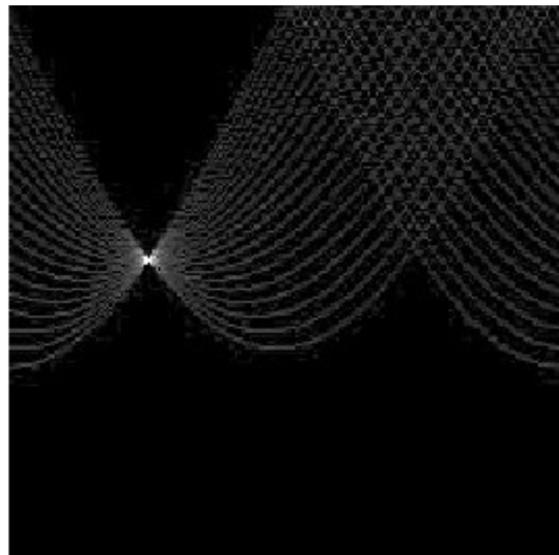


data space

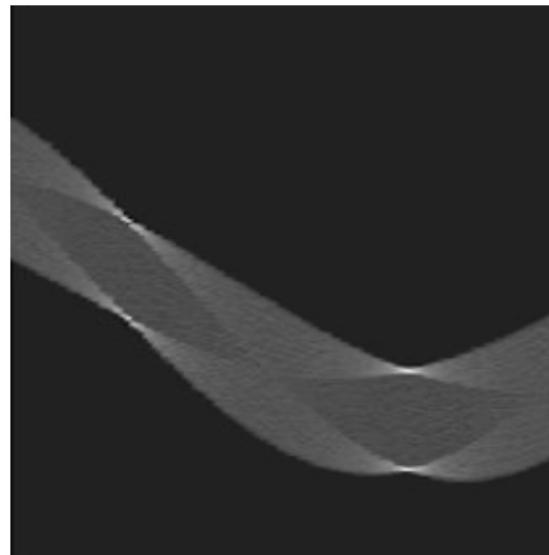


parameter space

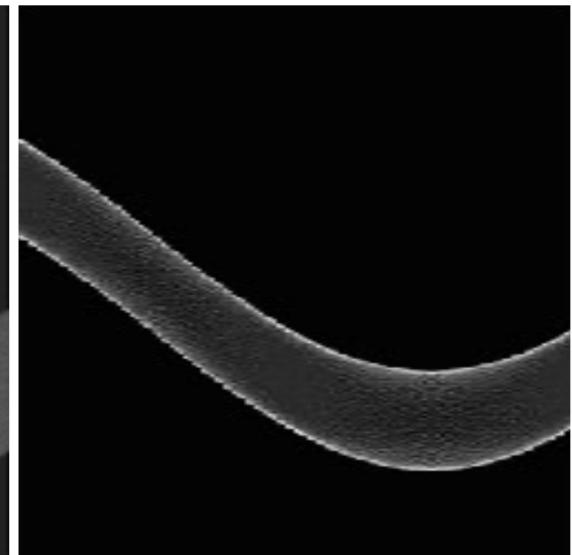
Hough transform



line



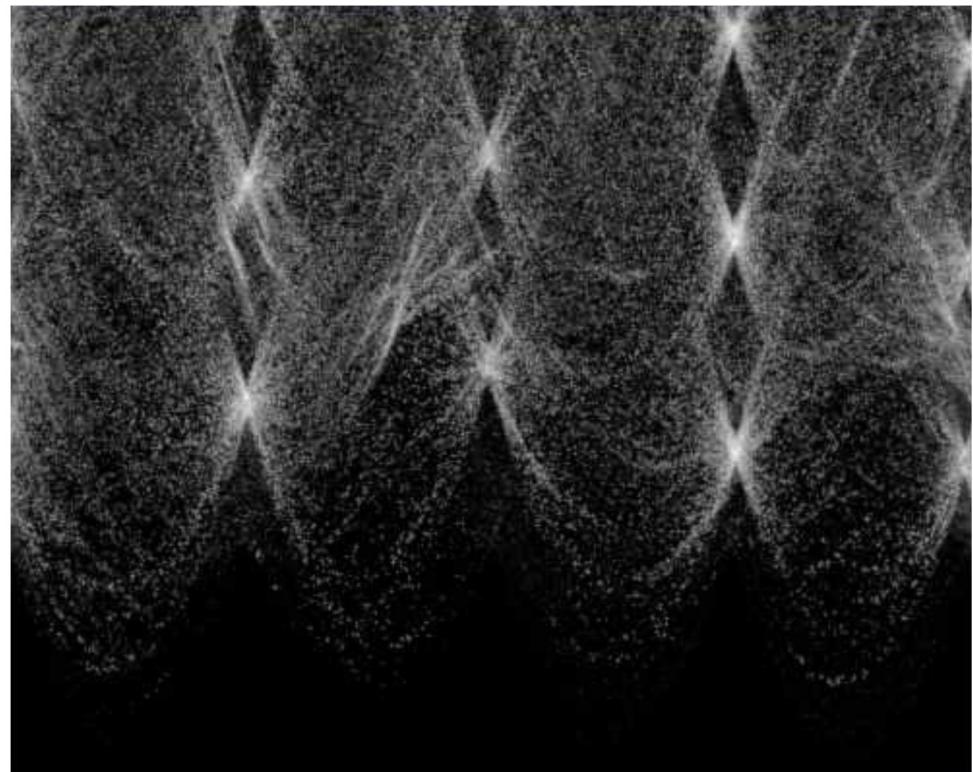
rectangle



circle

Hough transform

Real-world example

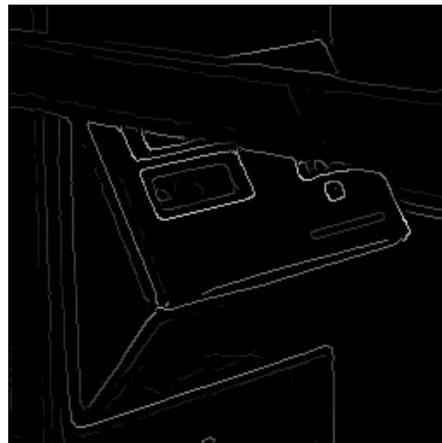


Hough transform

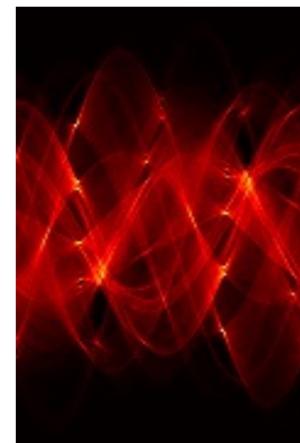
Real-world example



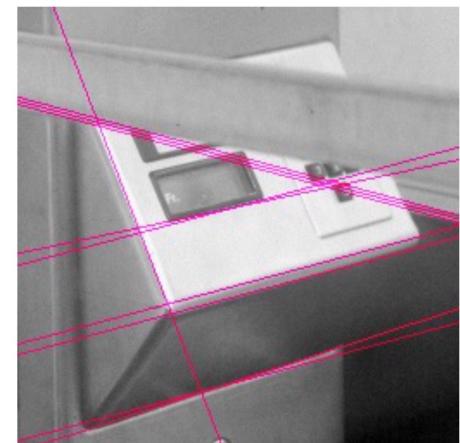
original



edges



parameter
space



Hough
lines

Line detection

Finding boundaries

Line parameterization

Hough transform

Deep Hough transform

Deep Hough transform

Propose an end-to-end framework for incorporating the feature learning capacity of CNN with Hough transform, resulting in an efficient real-time solution for **semantic line** detection.



Fig. 1. Example pictures from [12] reveal that semantic lines may help in the photographic composition. (a): a photo was taken with an arbitrary pose. (b): a photo fits the golden ratio principle [21], [33] which is obtained by the method described in [12] using so-called 'prominent lines' in the image. (c): Our detection results are clean and comprise only a few meaningful lines that are potentially helpful in the photographic composition. (d): Line detection results by the classical line detection algorithms often focus on fine detailed straight edges.

Deep Hough transform

Propose an end-to-end framework for incorporating the feature learning capacity of CNN with Hough transform, resulting in an efficient real-time solution for **semantic line** detection.

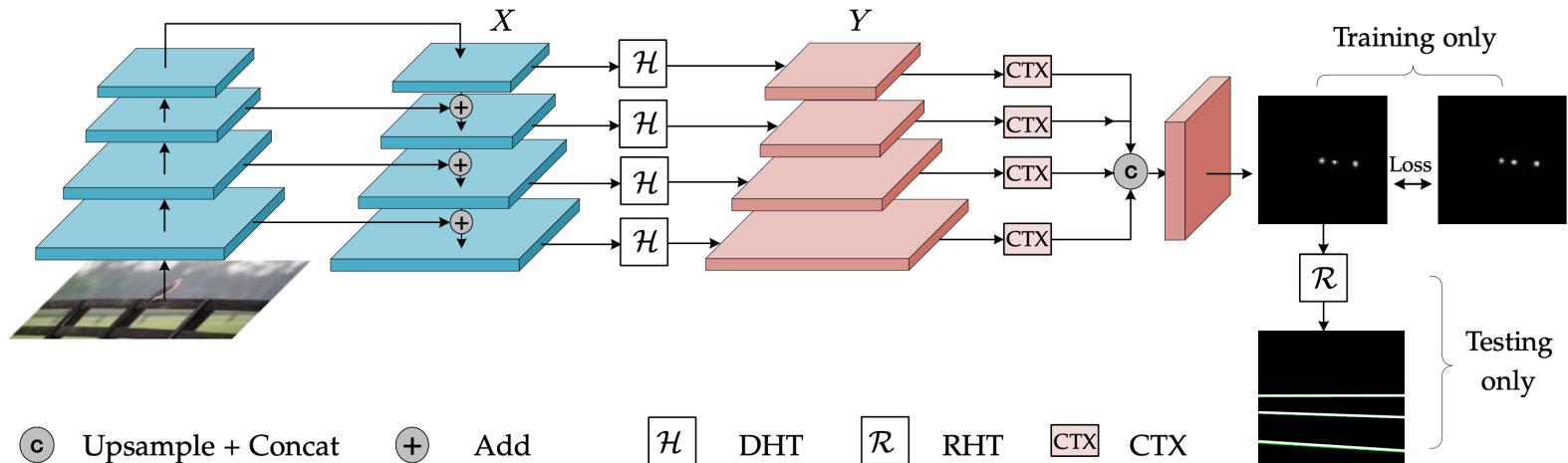


Fig. 2. The pipeline of our proposed method. DHT is short for the proposed Deep Hough Transform, and RHT represents the Reverse Hough Transform. CTX means the context-aware line detector which contains multiple convolutional layers.

Deep Hough transform

Propose an end-to-end framework for incorporating the feature learning capacity of CNN with Hough transform, resulting in an efficient real-time solution for **semantic line** detection.

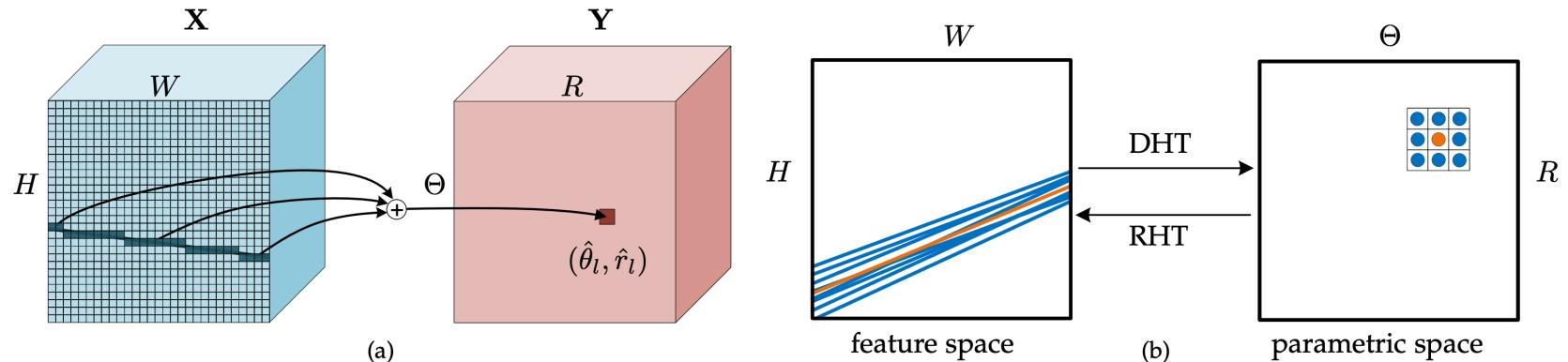


Fig. 4. (a): Features along a line in the feature space (blue, left) are accumulated to a point $(\hat{\theta}_l, \hat{r}_l)$ in the parametric space (red, right). (b): Illustration of the proposed context-aware feature aggregation. Features of nearby lines in the feature space (left) are translated into neighbor points in the parametric space (right). In the parametric space, a simple 3×3 convolutional operation can easily capture contextual information for the central line (orange). Best viewed in color.

Deep Hough transform

Propose an end-to-end framework for incorporating the feature learning capacity of CNN with Hough transform, resulting in an efficient real-time solution for **semantic line** detection.

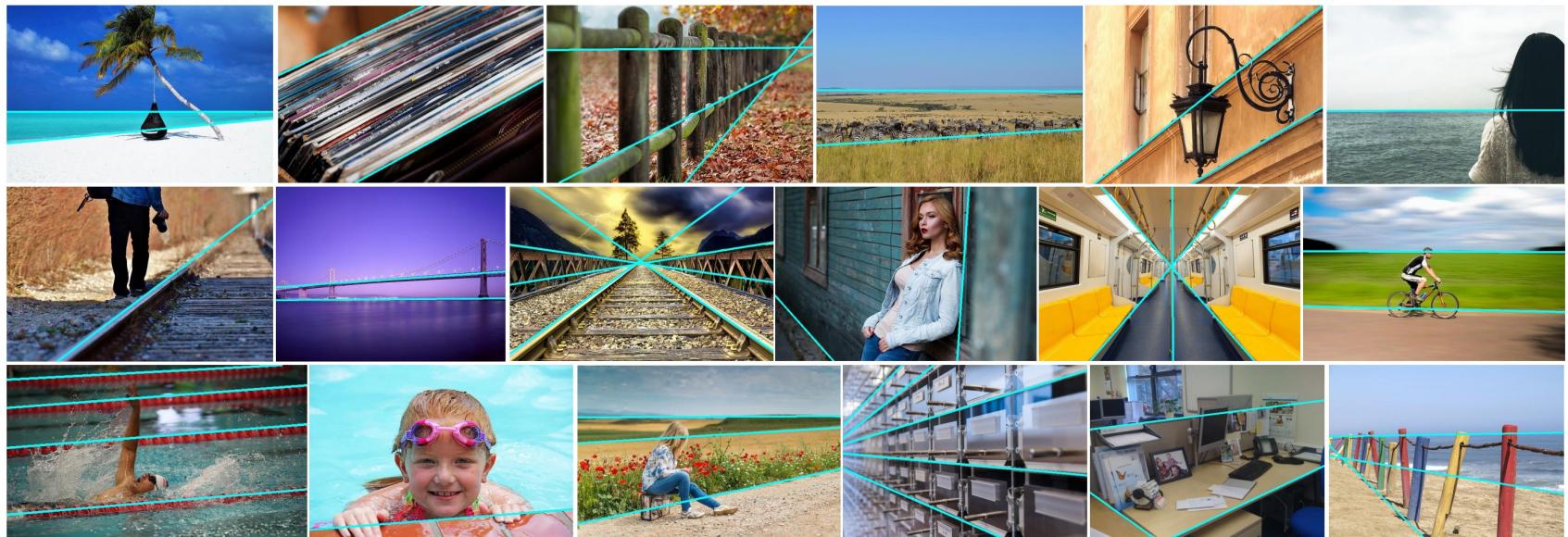


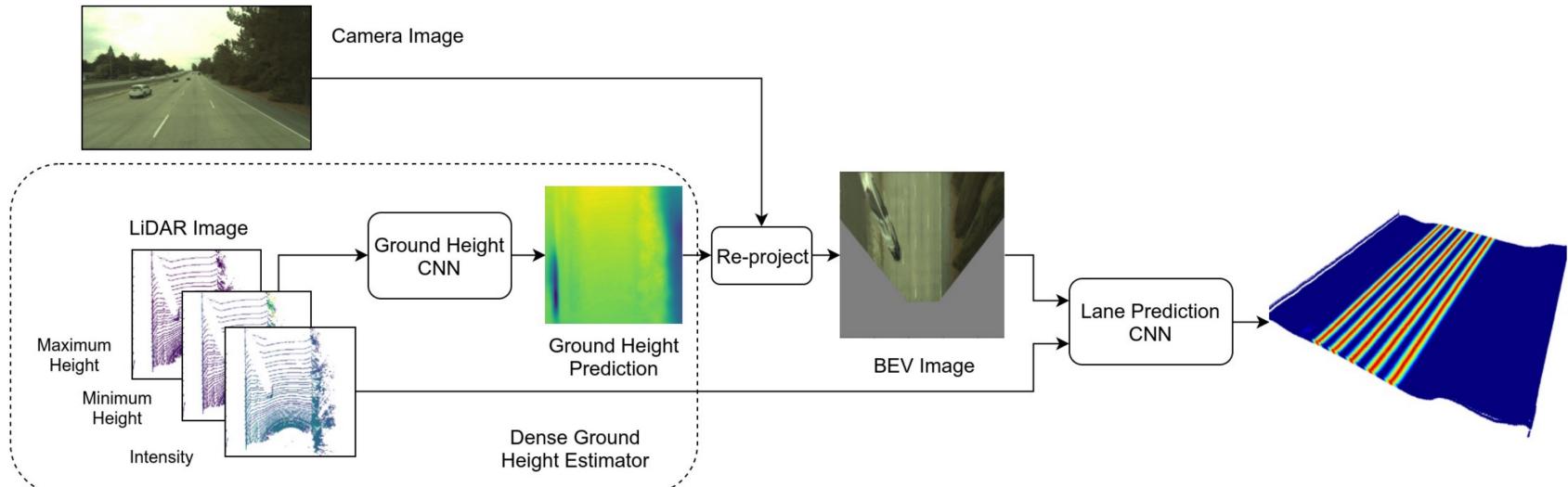
Fig. 14. Detection results of our method on the NKL dataset. Our method produces results that are visually compatible with human perception.

Deep Hough transform

Try this <http://mc.nankai.edu.cn/dht/>

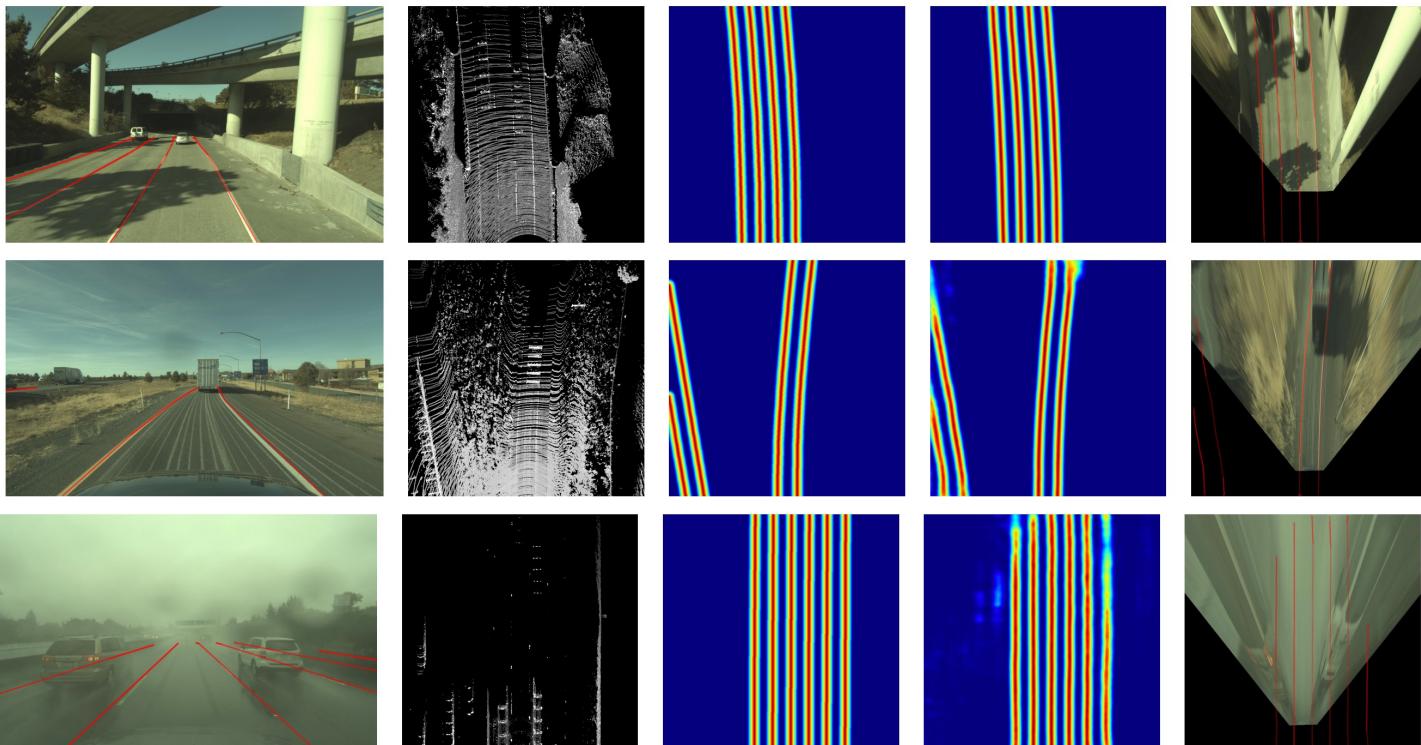
Deep multi-sensor lane detection

Propose a novel deep neural network that takes advantage of both LiDAR and camera sensors and produces very accurate estimates directly in 3D space



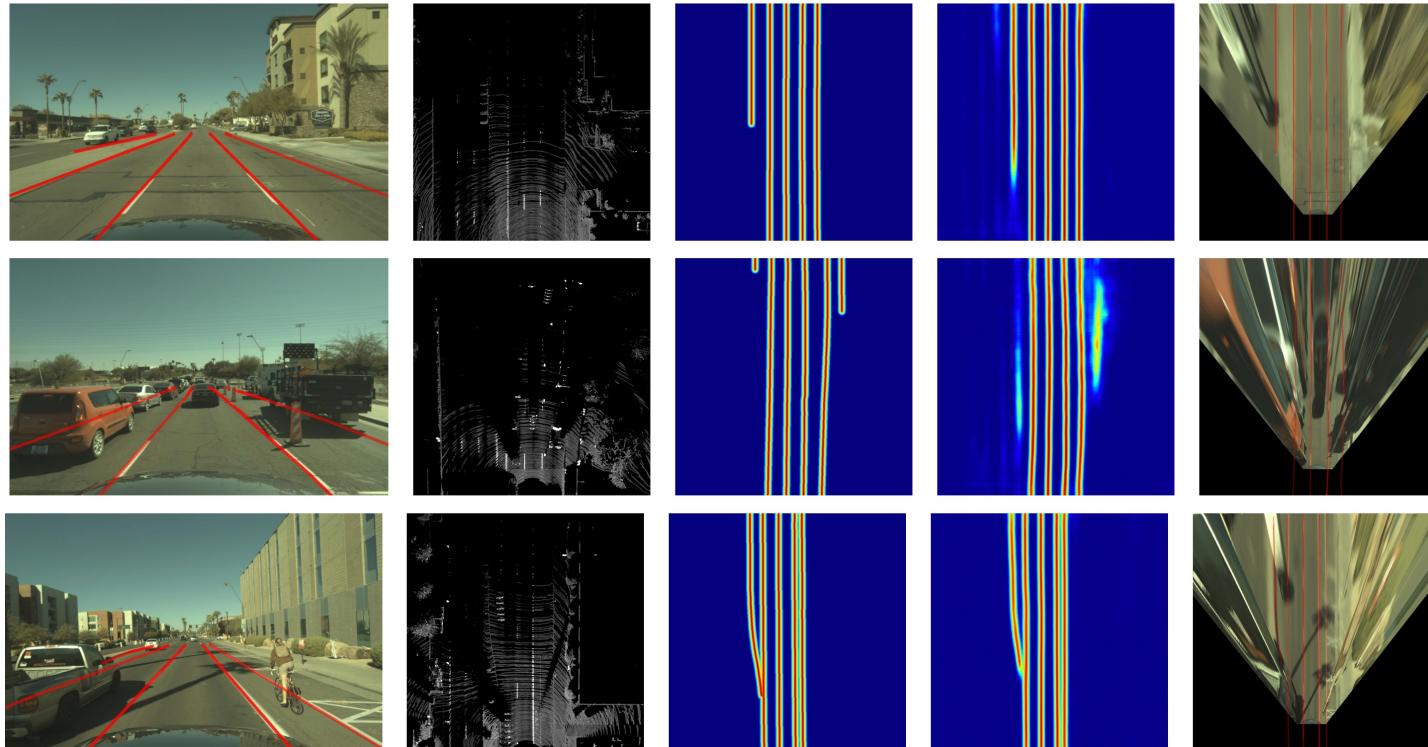
Deep multi-sensor lane detection

Output using both LiDAR and camera on **highways**

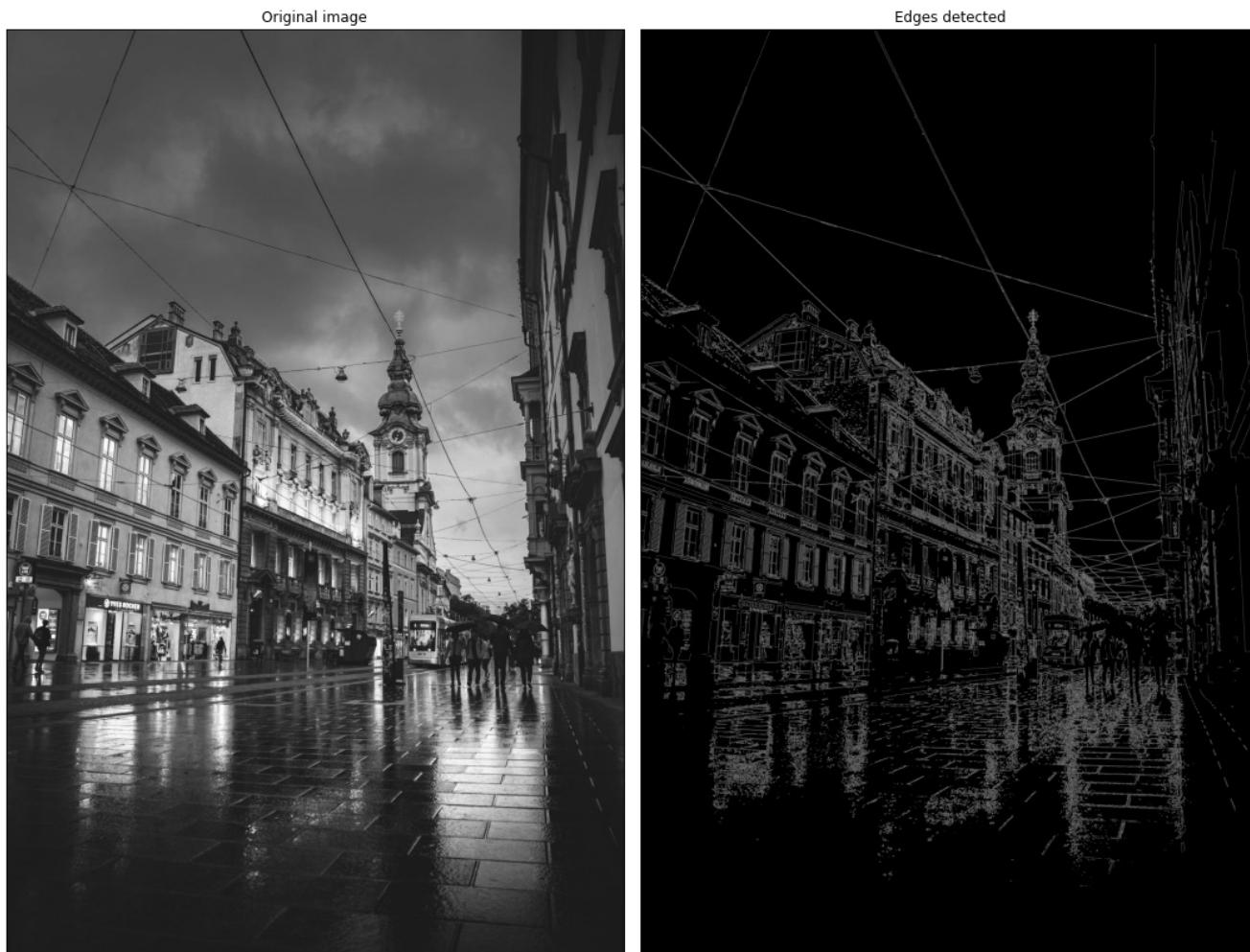


Deep multi-sensor lane detection

Output using both LiDAR and camera on **urban areas**



Next lecture: Edge detection



Thank you very much!

sihengc@sjtu.edu.cn