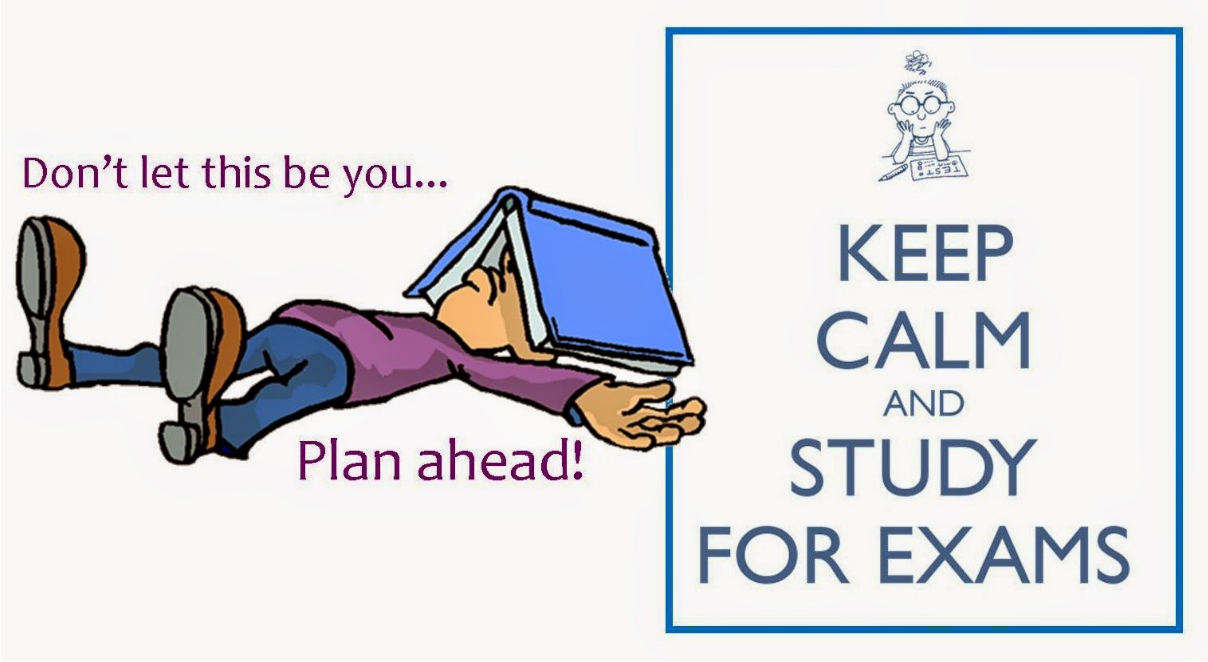
CPSC 425: Computer Vision



Lecture 35: Review

Menu for Today (**November 30, 2018**)

Topics:

— **Final** Review

Redings:

— **Today's** Lecture: N/A

— **Next** Lecture: N/A

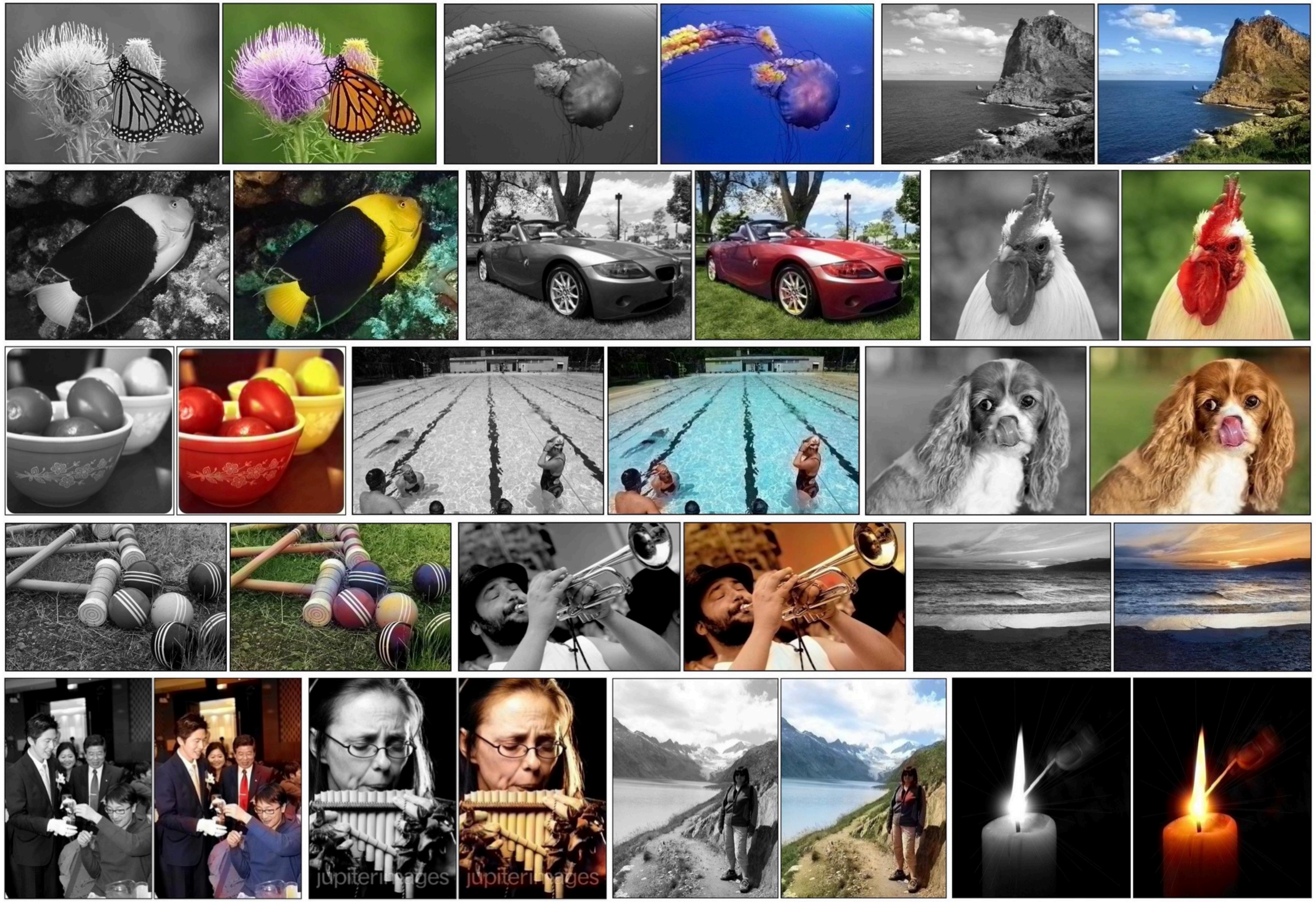
Reminders:

— **Assignment 5**: Scene Recognition with Bag of Words due **today**

— **Midterm** is now graded from **47 points** (+6% for everyone)

— Additional office hours, Piazza, etc.

Today’s “**fun**” Example: Colorful Image Colorization



Final Exam Details

2.5 hours

Closed book, **no** calculators

— Equations will be given

Format similar to midterm exam

— Part A: Multiple-part true/false

— Part B: Short answer

No coding questions

How to study?

— Look at the Lectures Notes and Assignment and think critically if you **truly** understand the material

— It easy to look at the slides and think — “This all makes sense”

— Look at each algorithm, concept,
  - — what are properties of the algorithm / concept?
  - — what does each step do?
  - — is this step important? can you imagine doing it another way?
  - — what are parameters? what would be the effect of changing those?

Course Review: Reading

Lecture **slides**

Assigned **readings** from Forsyth & Ponce (2nd ed.)
— Paper "Texture Synthesis by Non-parametric Sampling"
— Paper "Distinctive Image Features from Scale-Invariant Keypoints"

Assignments 1-5

iClicker questions

Lecture exercises

Practice **problems** (with solutions) — 4 more sets will be up tonight.

Course Review: Cameras and Lenses

Pinhole **camera**

Projections (and projection equations)

— perspective, weak perspective, orthographic

Lenses

Human **eye**

Course Review: Filters

Correlation and **convolution**

Box, pillbox, Gaussian filters

Separability

Non-linear filters: median, bilateral

Template matching

Course Review: Edge and Corners

Estimating the **image gradient**

Canny edge detection

Marr/Hildreth edge detection

Boundary detection

Harris corner detection

Course Review: Texture

Texture representation

Laplacian pyramid, oriented pyramid

Texture **synthesis** (Efros and Leung paper)

Course Review: Colour

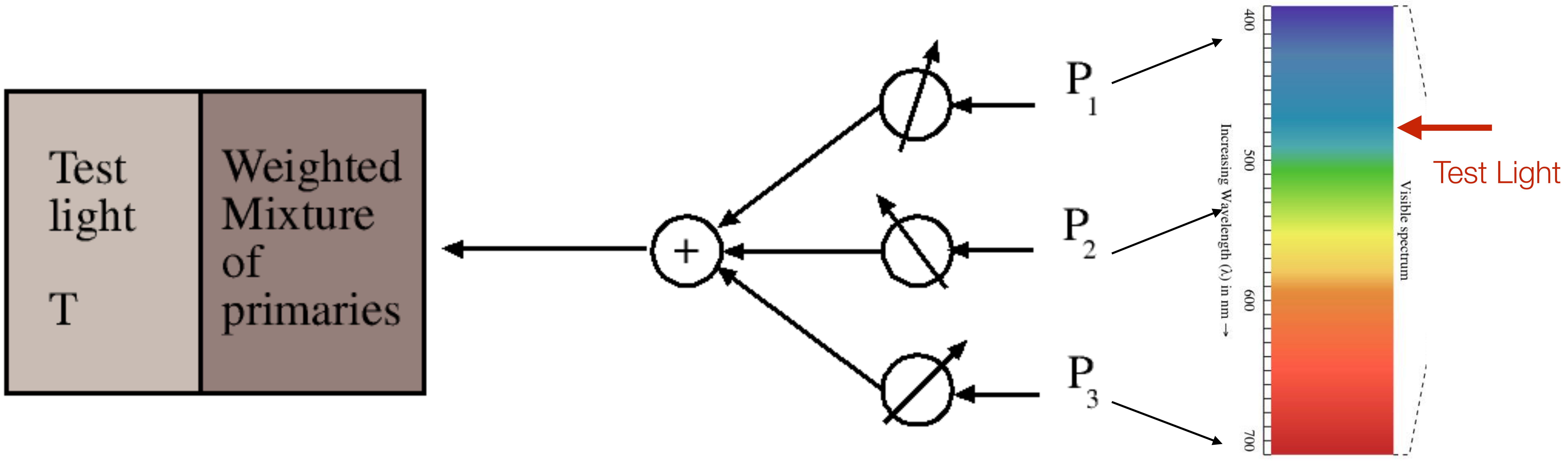
Human colour perception

RGB and **CIE XYZ** colour spaces

Uniform colour space

HSV colour space

Color Matching Experiments

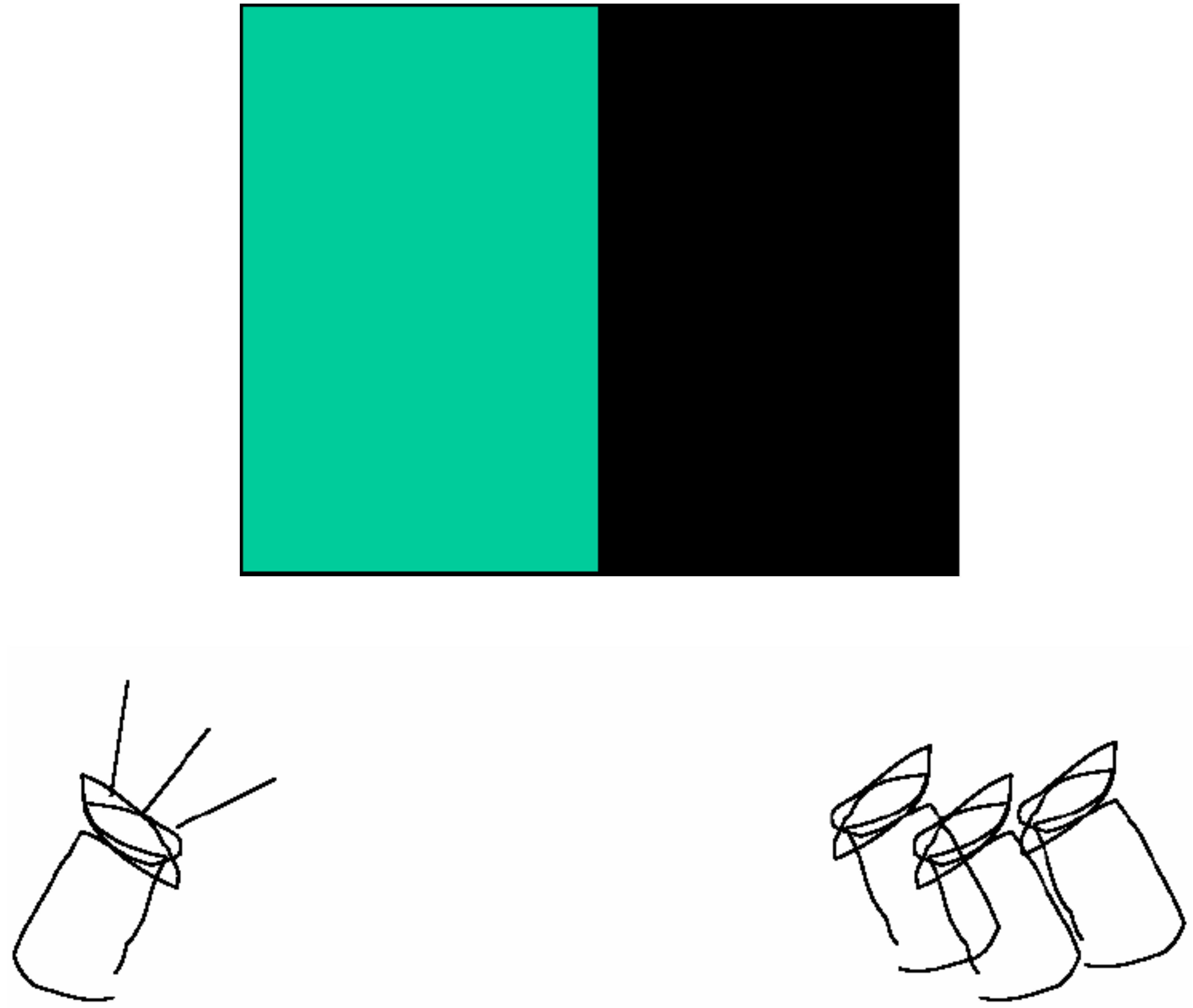


Forsyth & Ponce (2nd ed.) Figure 3.2

Show a split field to subjects. One side shows the light whose colour one wants to match. The other a weighted mixture of three primaries (fixed lights)

$$T = w_1 P_1 + w_2 P_2 + w_3 P_3$$

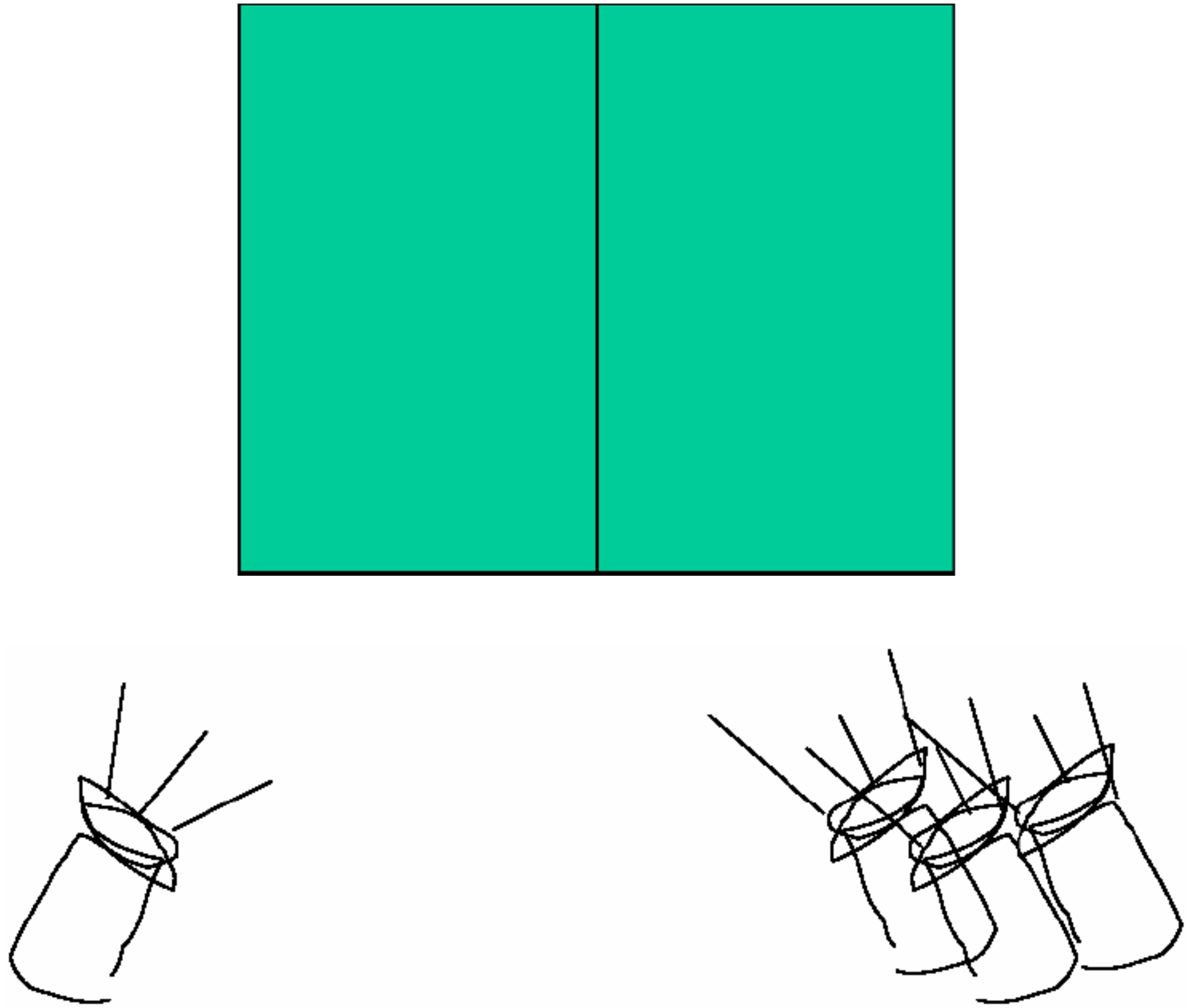
Example 1: Color Matching Experiment

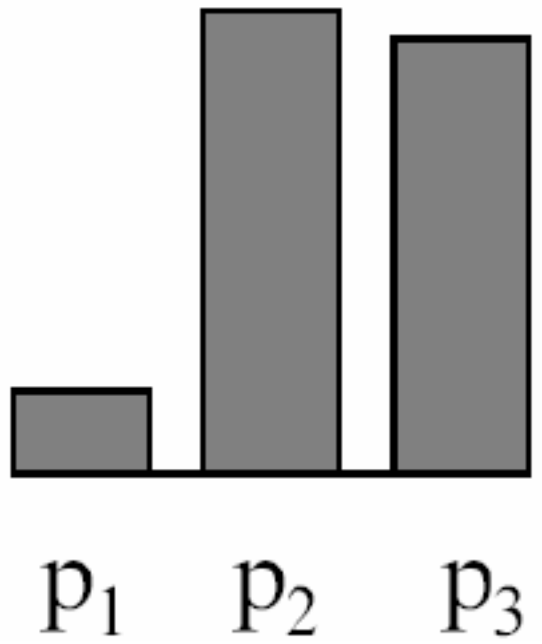


knobs here

Example Credit: Bill Freeman

Example 1: Color Matching Experiment

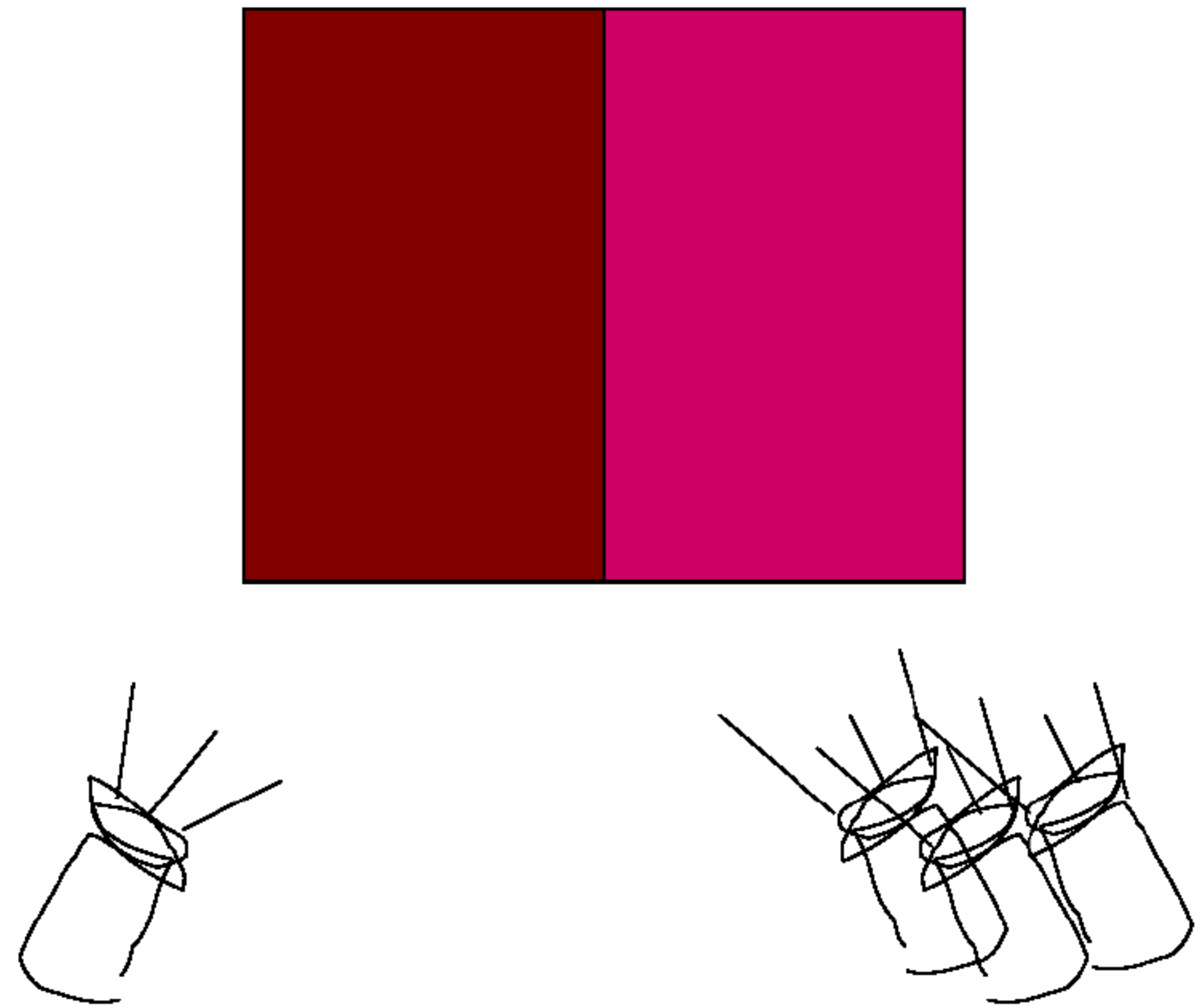


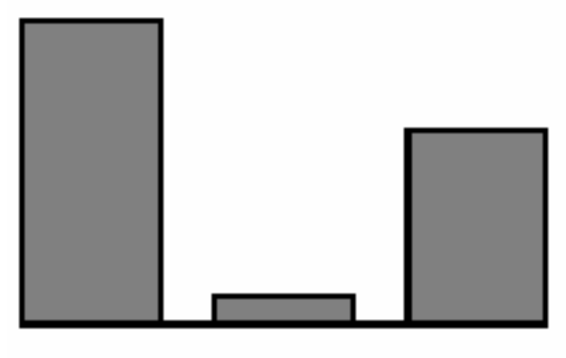


knobs here

Example Credit: Bill Freeman

Example 2: Color Matching Experiment

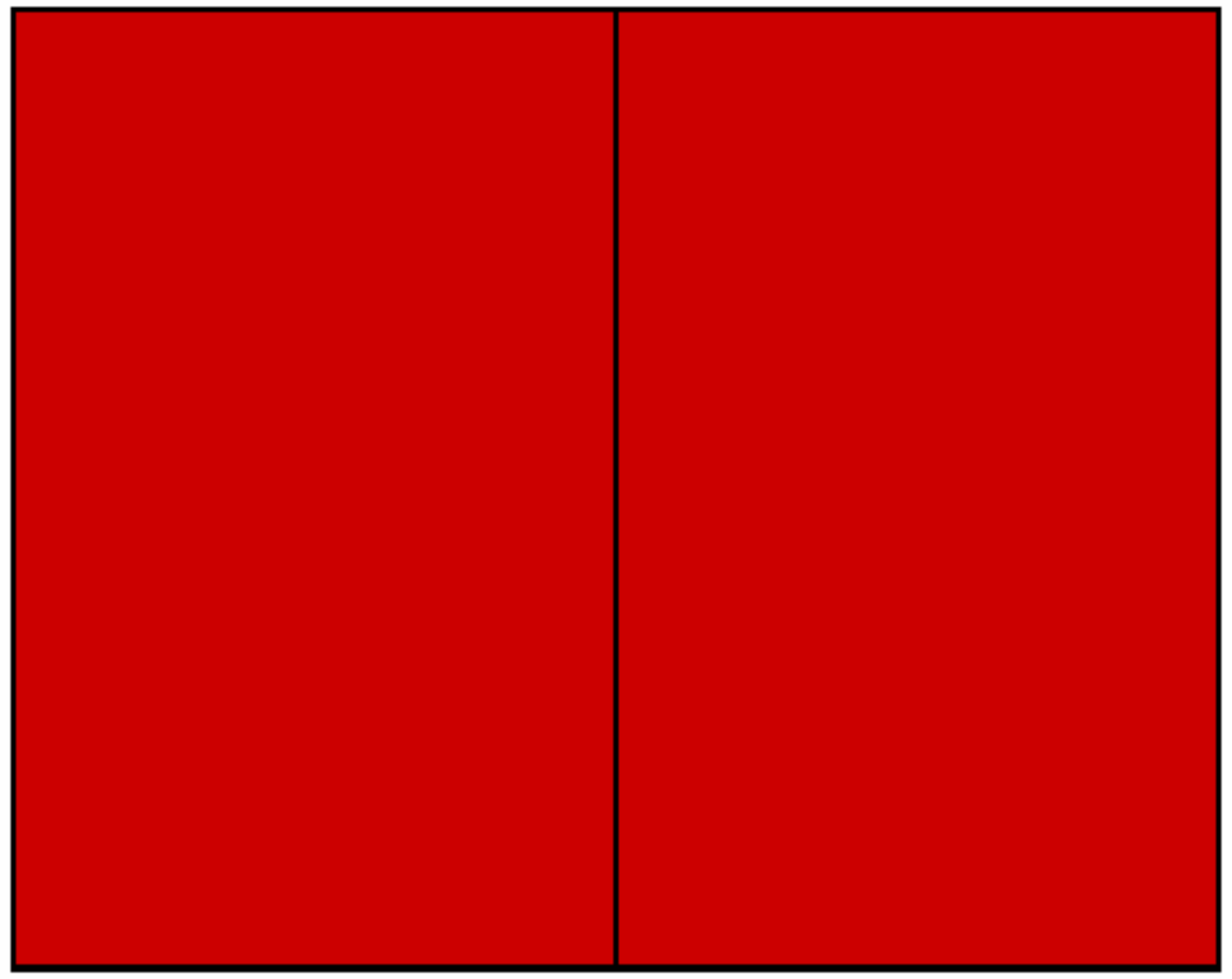




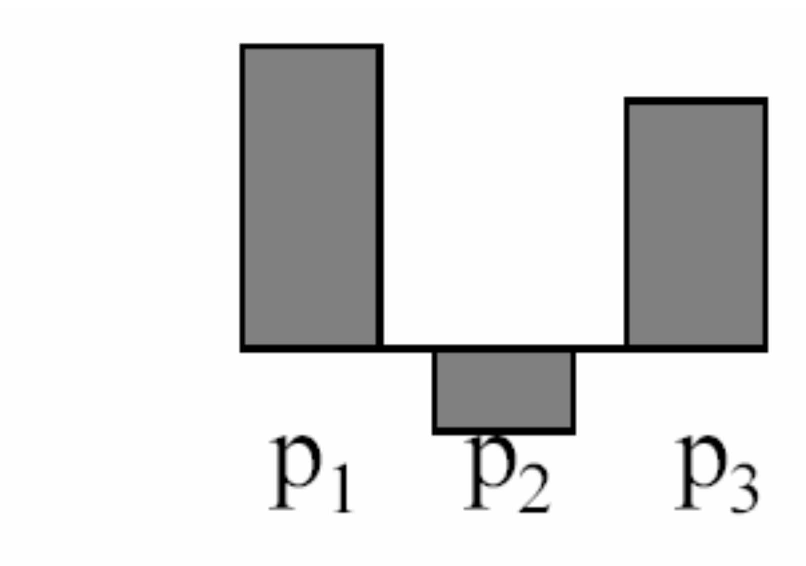
Example Credit: Bill Freeman

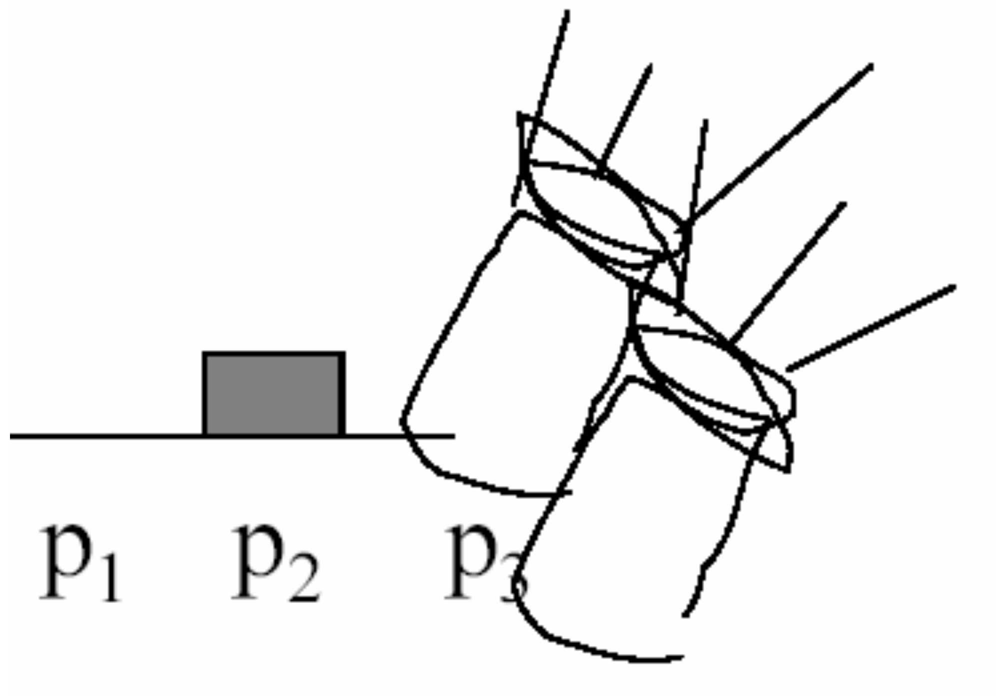
Example 2: Color Matching Experiment

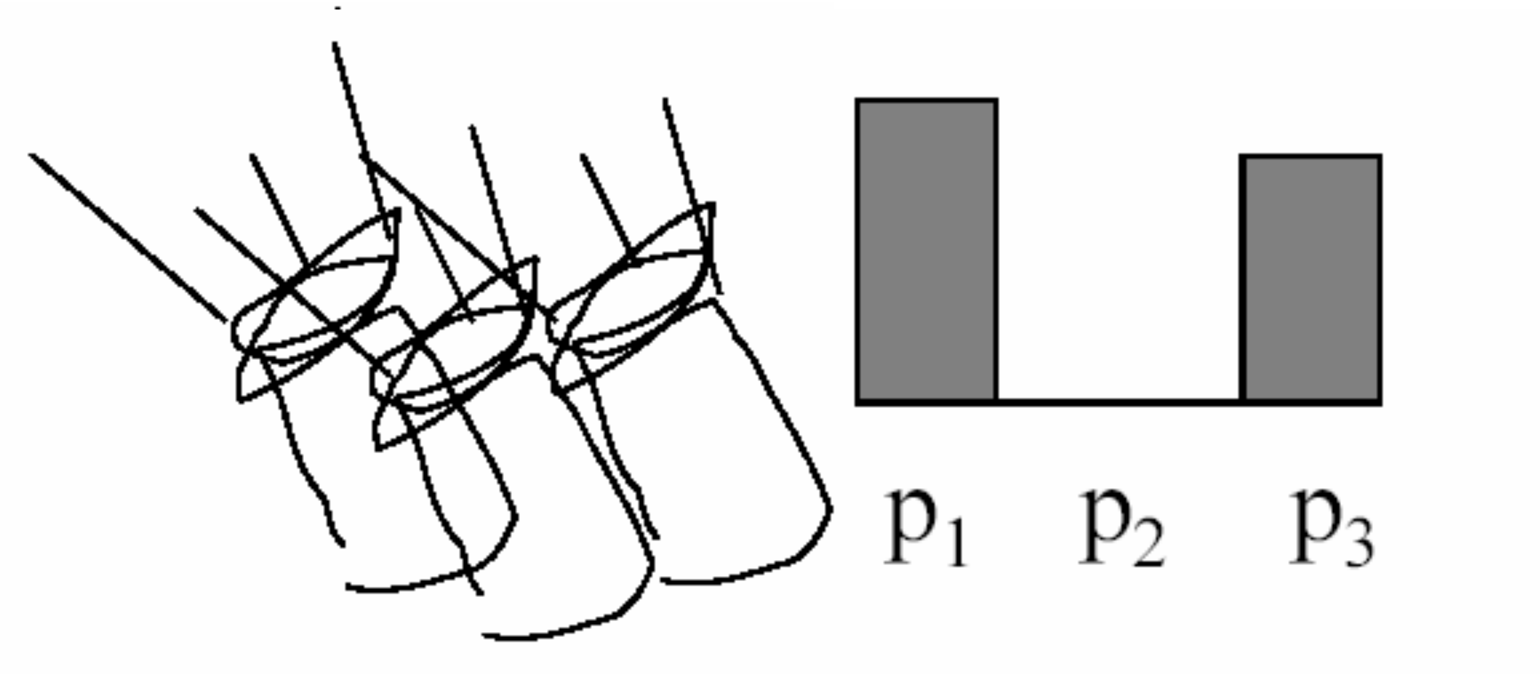
We say a "negative" amount of P_2 was needed to make a match , because we added it to the test color side



The primary color amount needed to match:



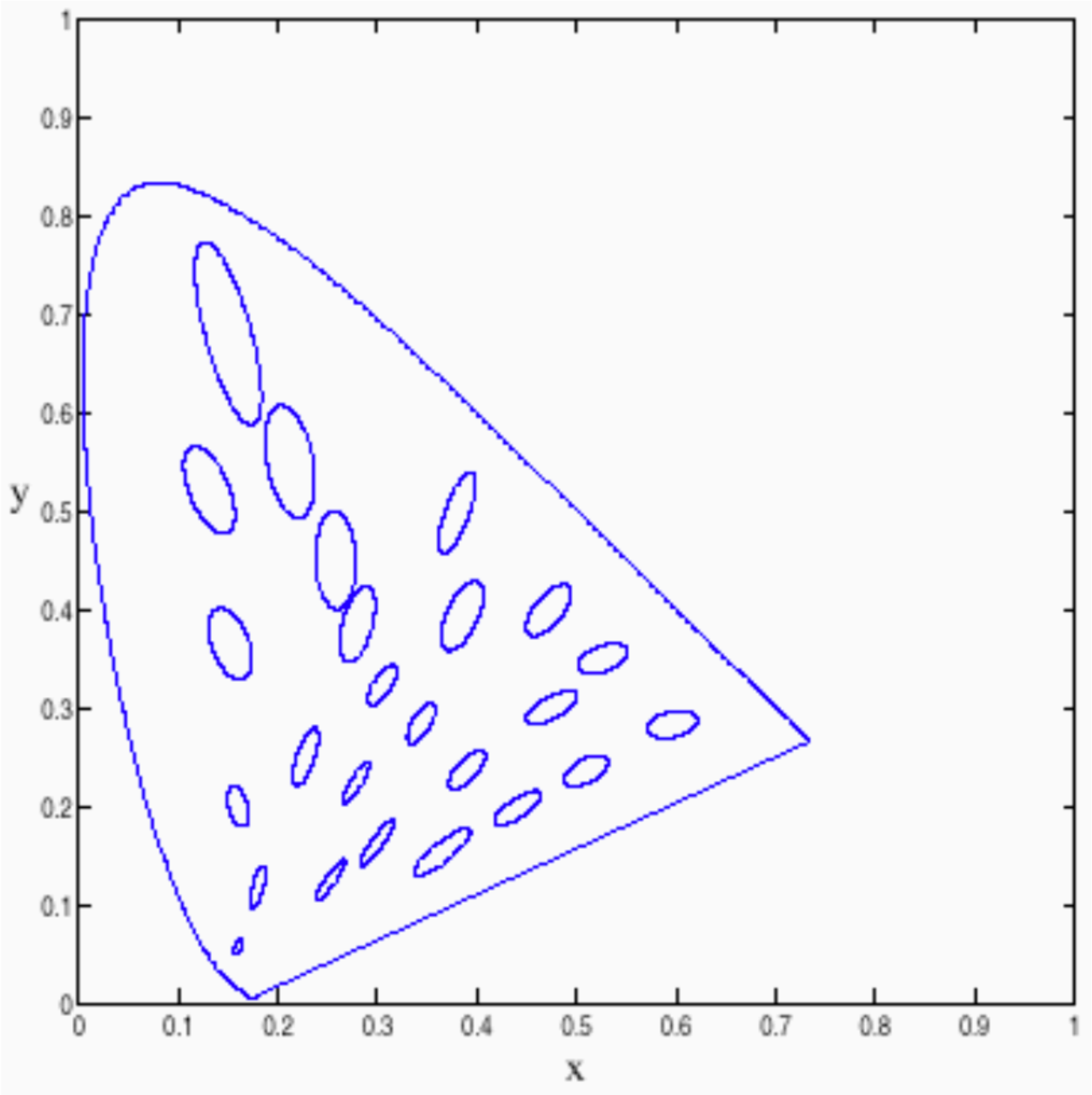




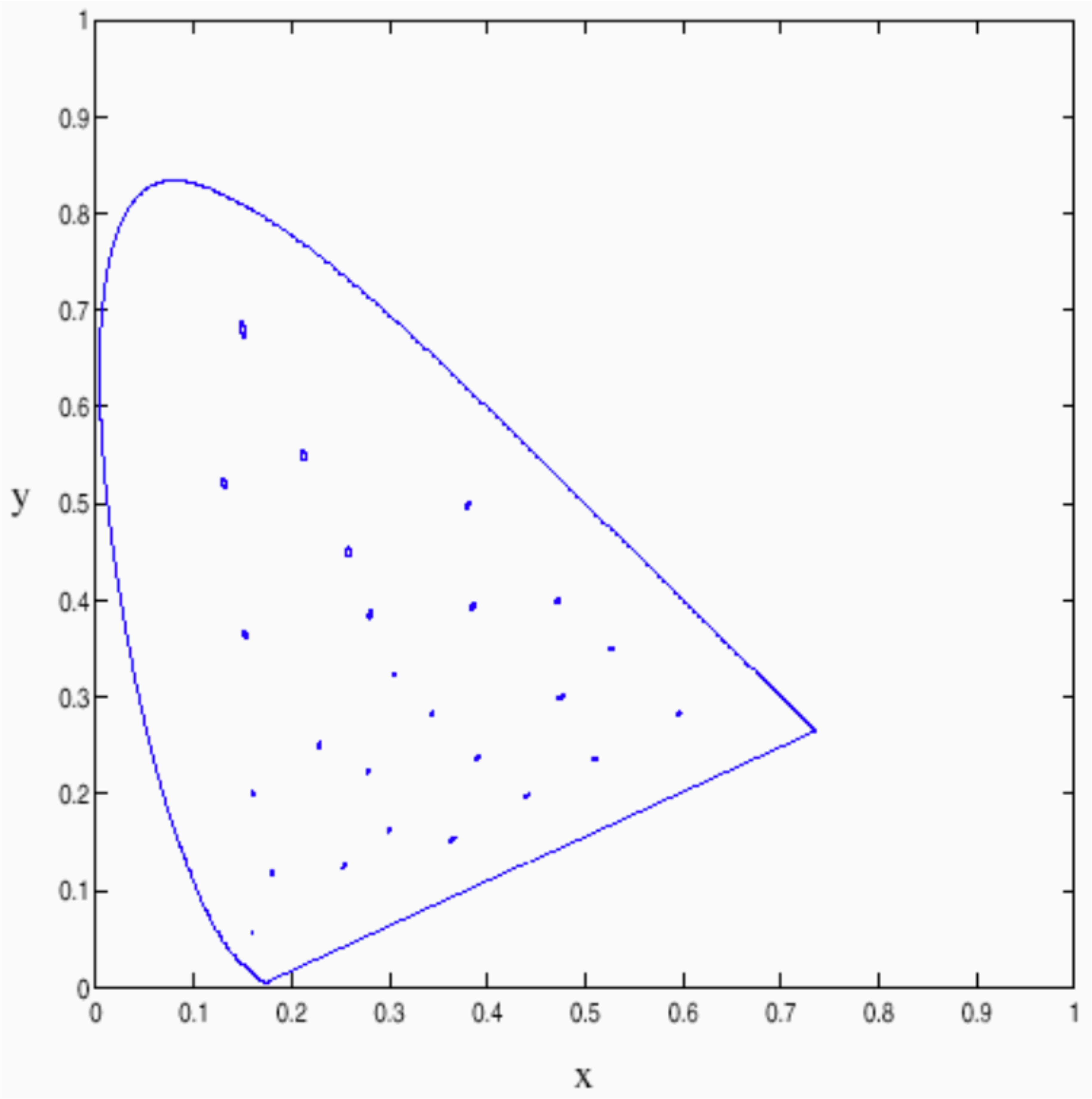
Example Credit: Bill Freeman

Uniform Colour Spaces

McAdam Ellipses: Each ellipse shows colours perceived to be the same



10 times actual size



Actual Size

Forsyth & Ponce (2nd ed.) Figure 3.14

Uniform Colour Spaces

McAdam ellipses demonstrate that differences in x , y are a poor guide to differences in perceived colour

A **uniform colour space** is one in which differences in coordinates are a good guide to differences in perceived colour

— example: CIE LAB

HSV Colour Space

More natural description of colour for human interpretation

Hue: attribute that describes a pure colour
— e.g. ’red’, ’blue’

Saturation: measure of the degree to which a pure colour is diluted by white light
— pure spectrum colours are fully saturated

Value: intensity or brightness

Hue + saturation also referred to as **chromaticity**.

Course Review: Local Invariant Features

Keypoint detection using Difference of Gaussian pyramid

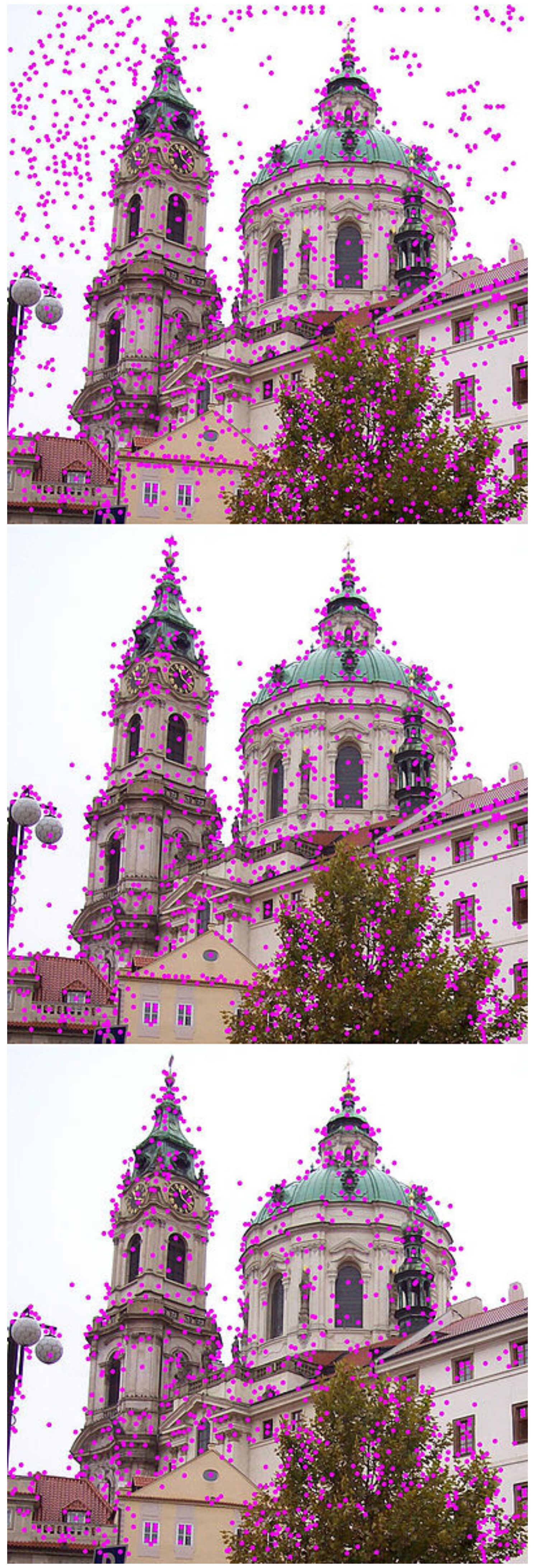
Keypoint **orientation assignment**

Keypoint **descriptor**

Matching with nearest and second-nearest neighbors

SIFT and object recognition

Scale Invariant Feature Transform (**SIFT**)

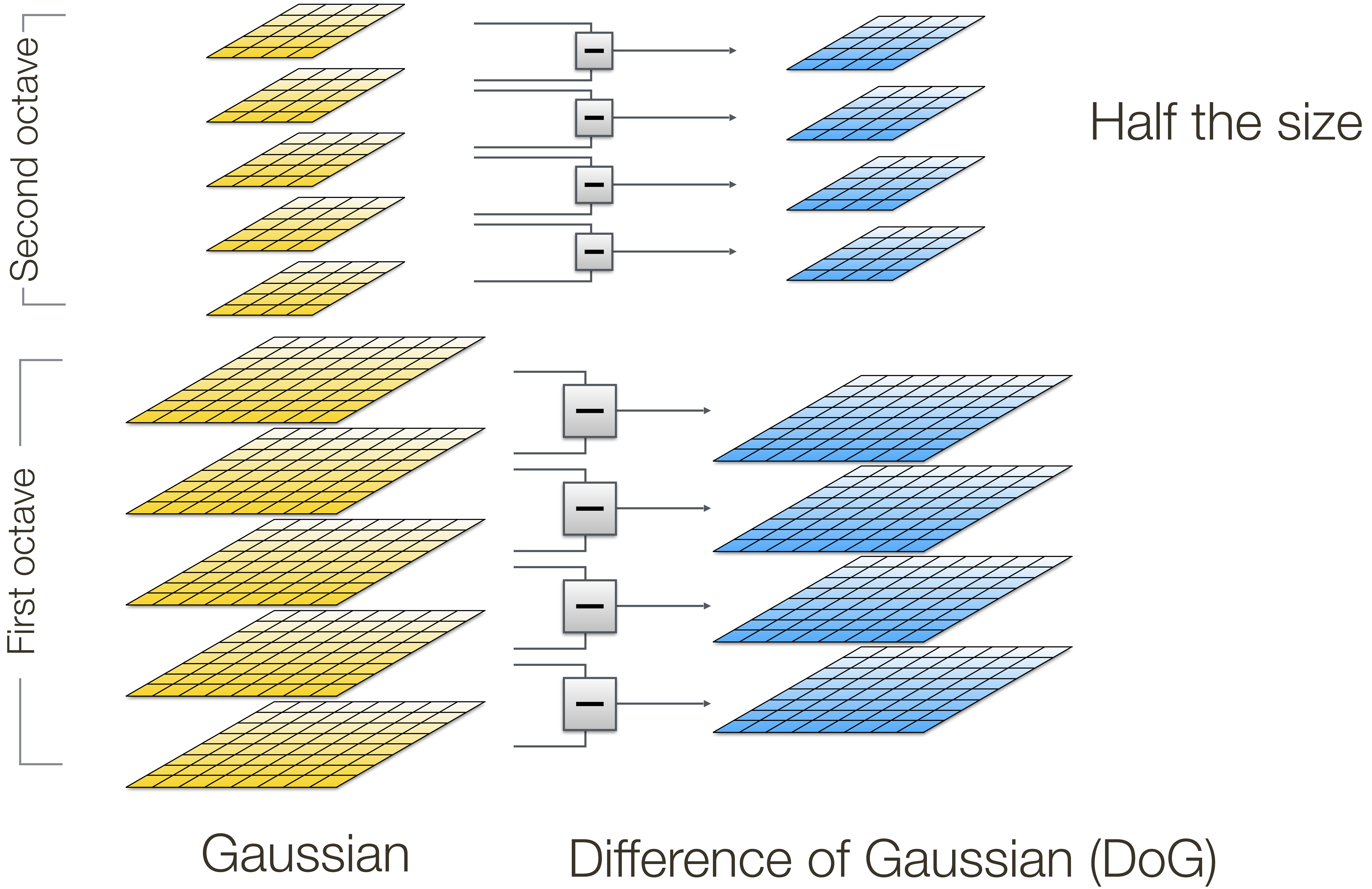


SIFT describes both a **detector** and **descriptor**

1. Multi-scale extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

1. Multi-scale Extrema Detection

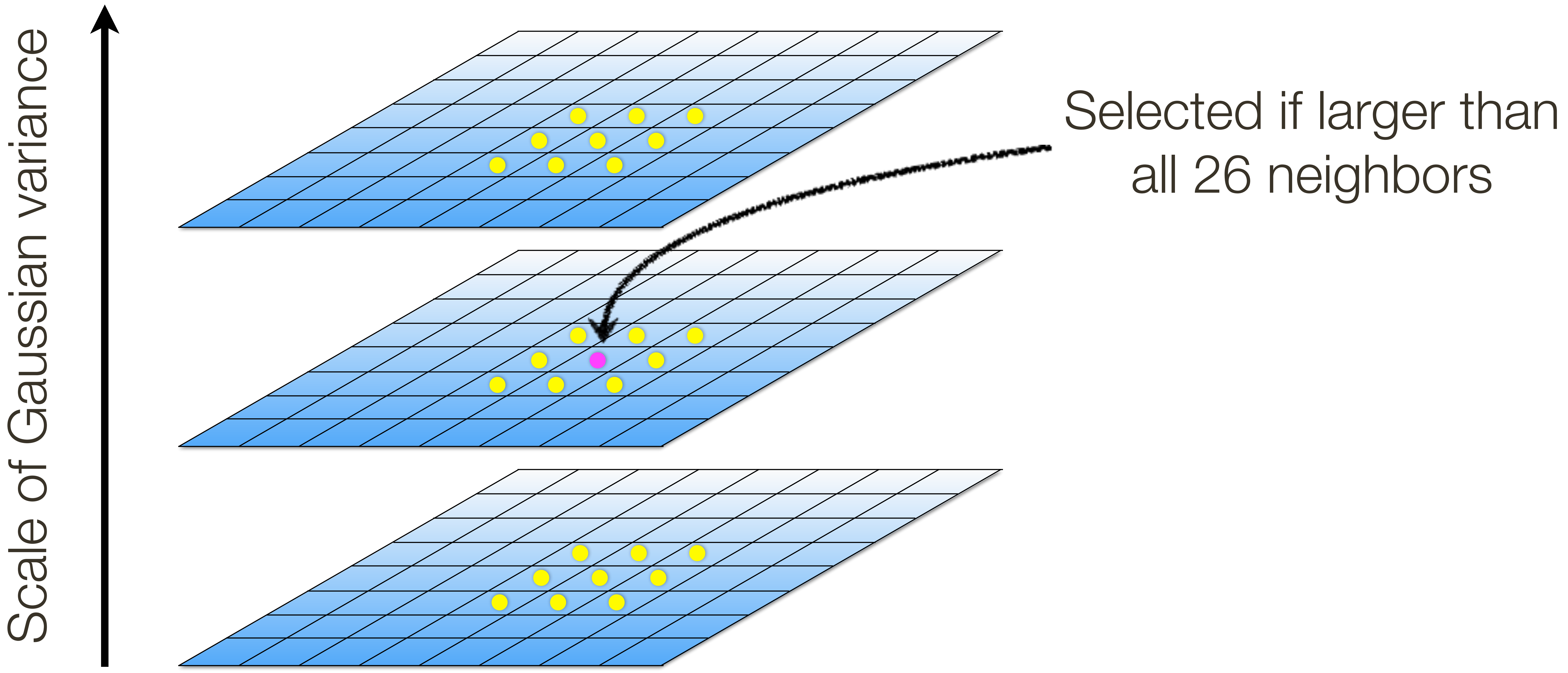


Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

1. Multi-scale Extrema Detection

Detect maxima and minima of Difference of Gaussian in scale space

$$\sigma = 2^{1/s}$$



Difference of Gaussian (DoG)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

2. Keypoint Localization

— After keypoints are detected, we remove those that have **low contrast** or are **poorly localized** along an edge

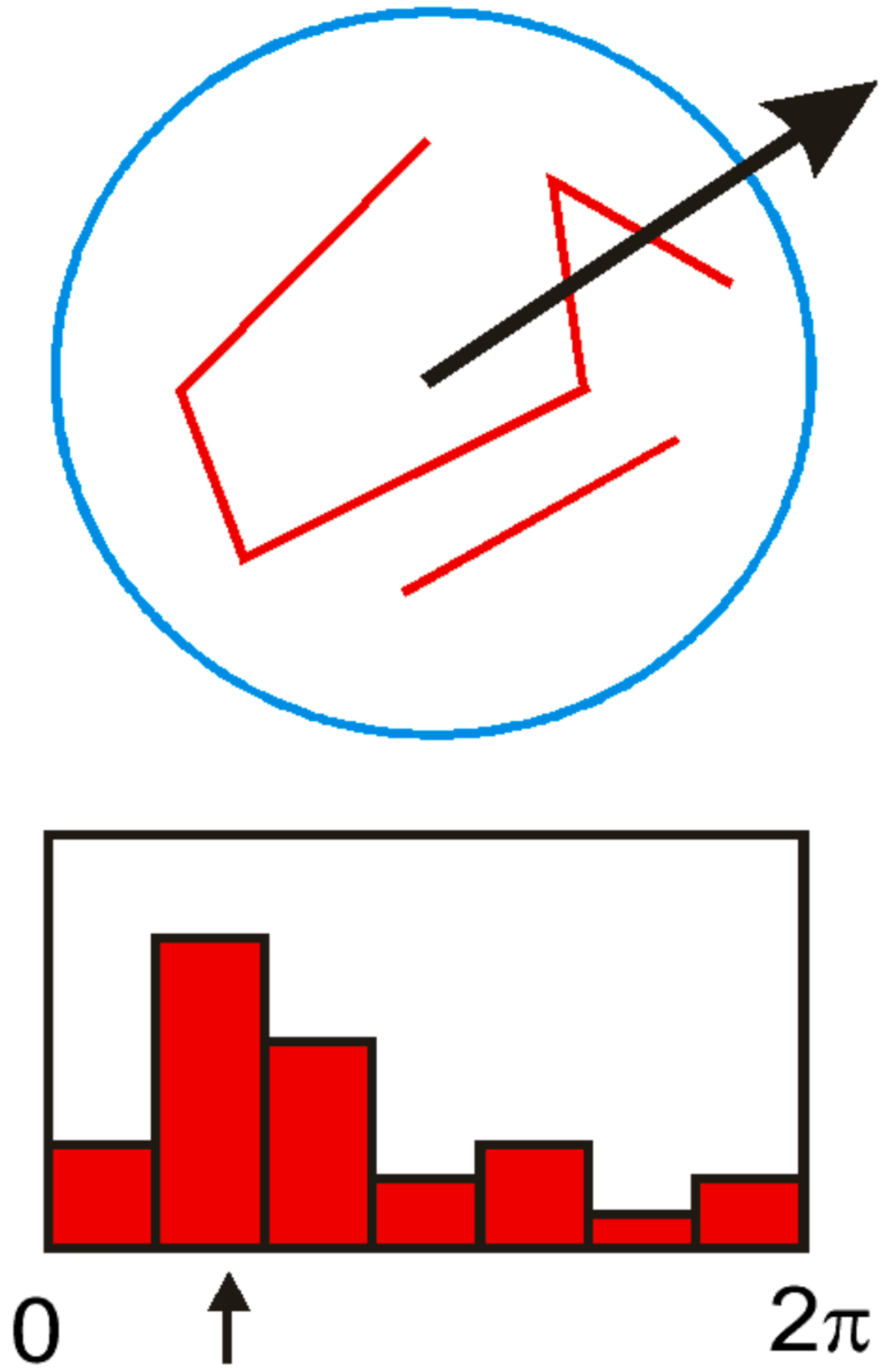
— Lowe suggests computing the ratio of the eigenvalues of **C** (recall Harris corners) and checking if it is greater than a threshold

3. Orientation Assignment

— Create **histogram** of local gradient directions computed at selected scale

— Assign **canonical orientation** at peak of smoothed histogram

— Each key specifies stable 2D coordinates (x , y , scale, orientation)

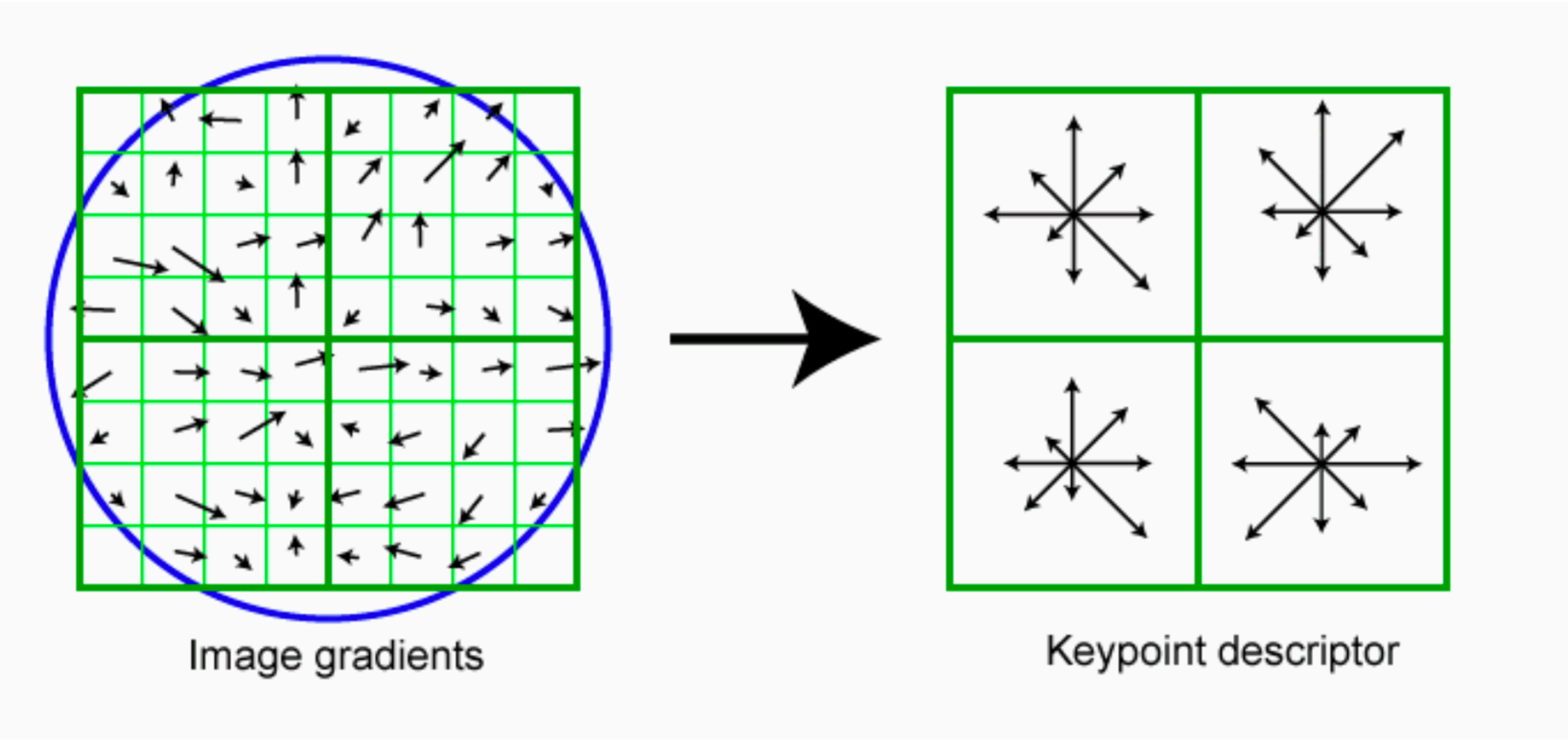


4. SIFT Descriptor

— Thresholded image gradients are sampled over 16 × 16 array of locations in scale space (weighted by a Gaussian with sigma half the size of the window)

— Create array of orientation histograms

— 8 orientations × 4 × 4 histogram array



Course Review: Fitting Data to a Model

RANSAC

Hough transform

RANSAC (**RAN**dom **SA**mple **C**onsensus)

1. Randomly choose minimal subset of data points necessary to fit model (a **sample**)

2. Points within some distance threshold, t, of model are a **consensus set**. Size of consensus set is model's **support**

3. Repeat for N samples; model with biggest support is most robust fit
   — Points within distance t of best model are inliers
   — Fit final model to all inliers

Slide Credit: Christopher Rasmussen

RANSAC: How many samples?

Let ω be the fraction of inliers (i.e., points on line)

Let n be the number of points needed to define hypothesis
($n = 2$ for a line in the plane)

Suppose k samples are chosen

The probability that a single sample of n points is correct (all inliers) is

$$\omega^n$$

The probability that all k samples fail is

$$(1 - \omega^n)^k$$

Choose k large enough (to keep this below a target failure rate)

RANSAC: *k* Samples Chosen (p = 0.99)

Sample size	Proportion of outliers						
n	**5%**	**10%**	**20%**	**25%**	**30%**	**40%**	**50%**
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Figure Credit: Hartley & Zisserman

Discussion of RANSAC

Advantages:

— General method suited for a wide range of model fitting problems

— Easy to implement and easy to calculate its failure rate

Disadvantages:

— Only handles a moderate percentage of outliers without cost blowing up

— Many real problems have high rate of outliers (but sometimes selective choice of random subsets can help)

The Hough transform can handle high percentage of outliers

Hough Transform

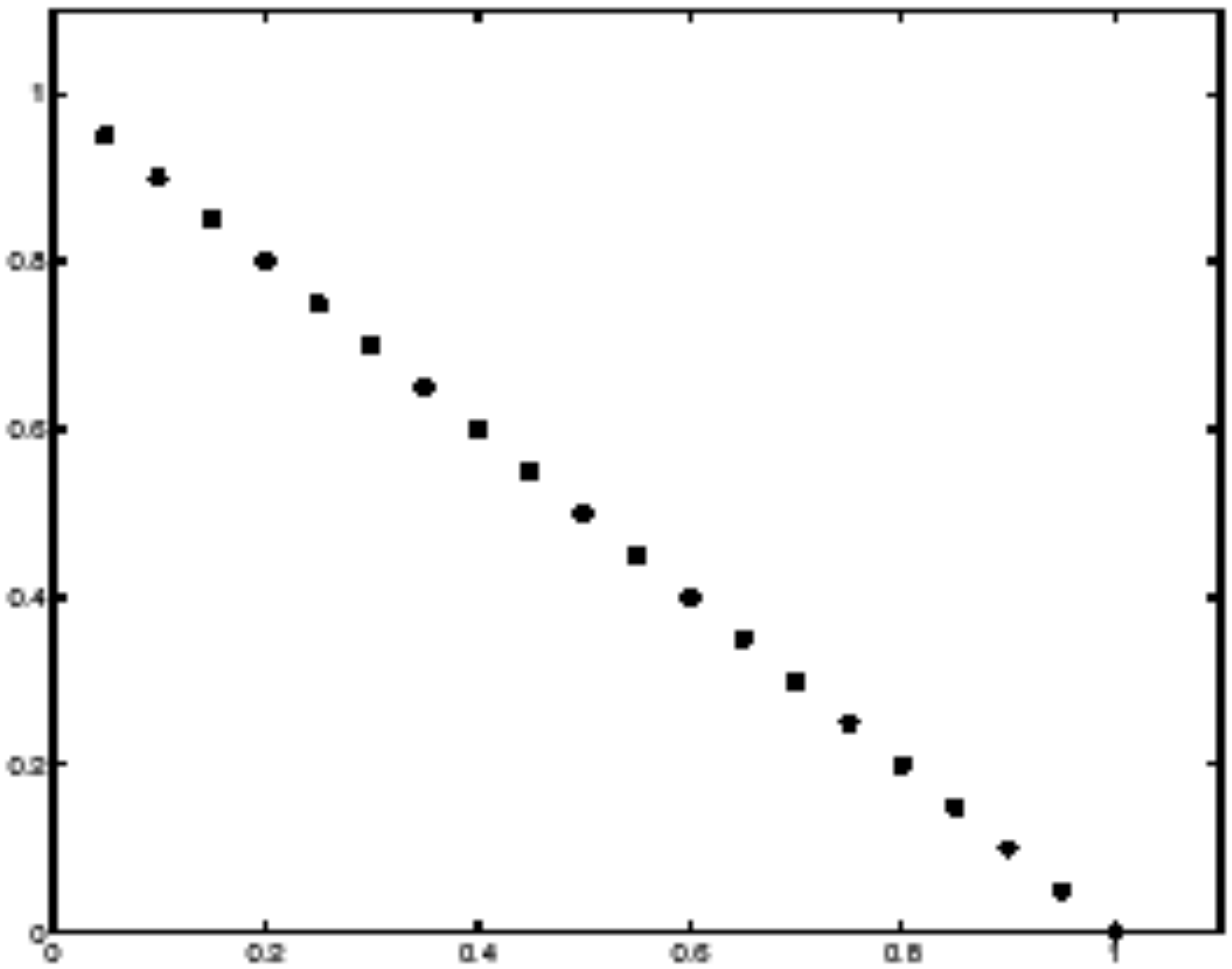
Idea of **Hough transform**:

— For each token vote for all models to which the token could belong

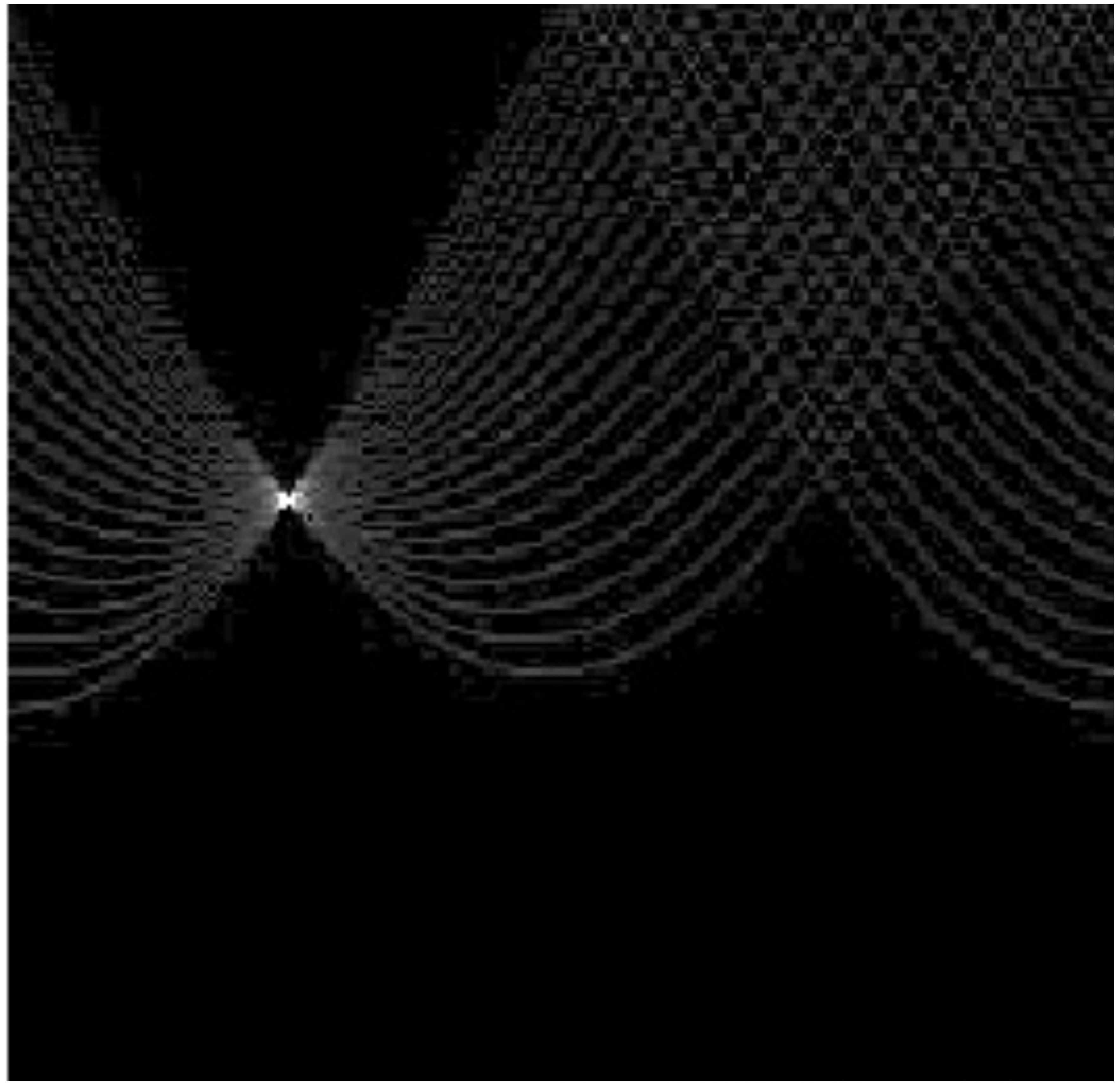
— Return models that get many votes

Example: For each point, vote for all lines that could pass through it; the true lines will pass through many points and so receive many votes

Example: Clean Data



Tokens



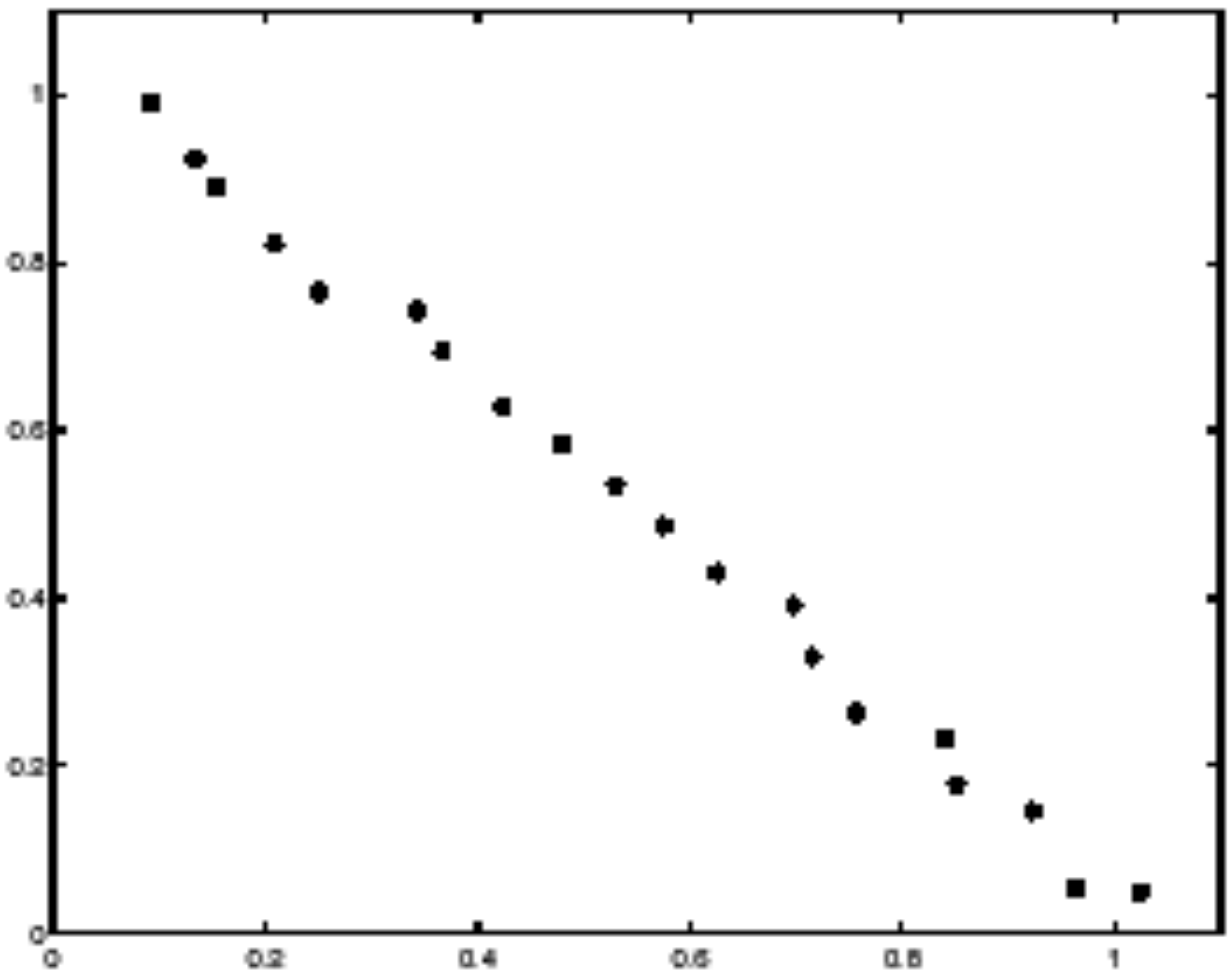
Votes

Horizontal axis is θ

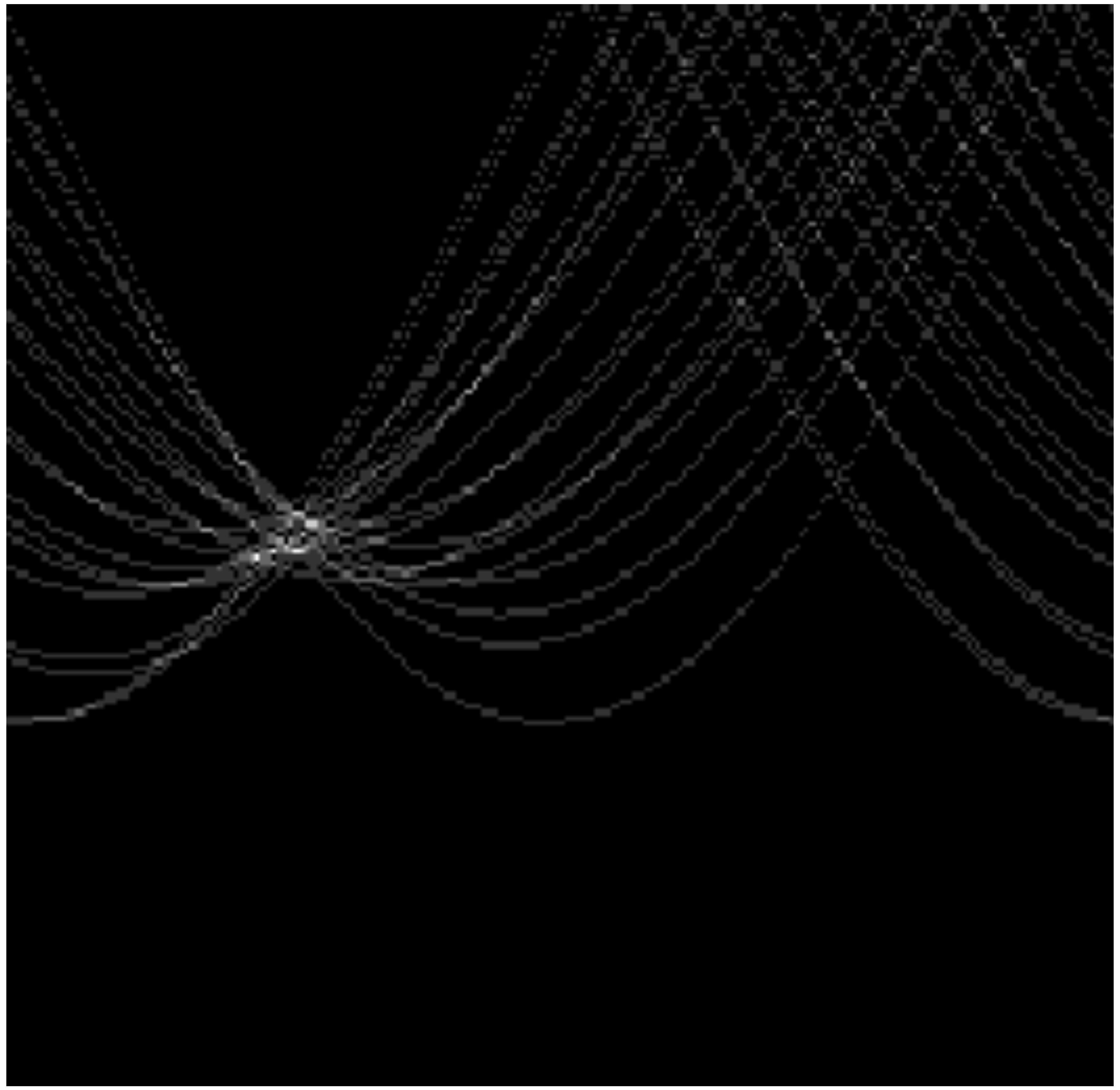
Vertical Axis is r

Forsyth & Ponce (2nd ed.) Figure 10.1 (Top)

Example: Some Noise



Tokens



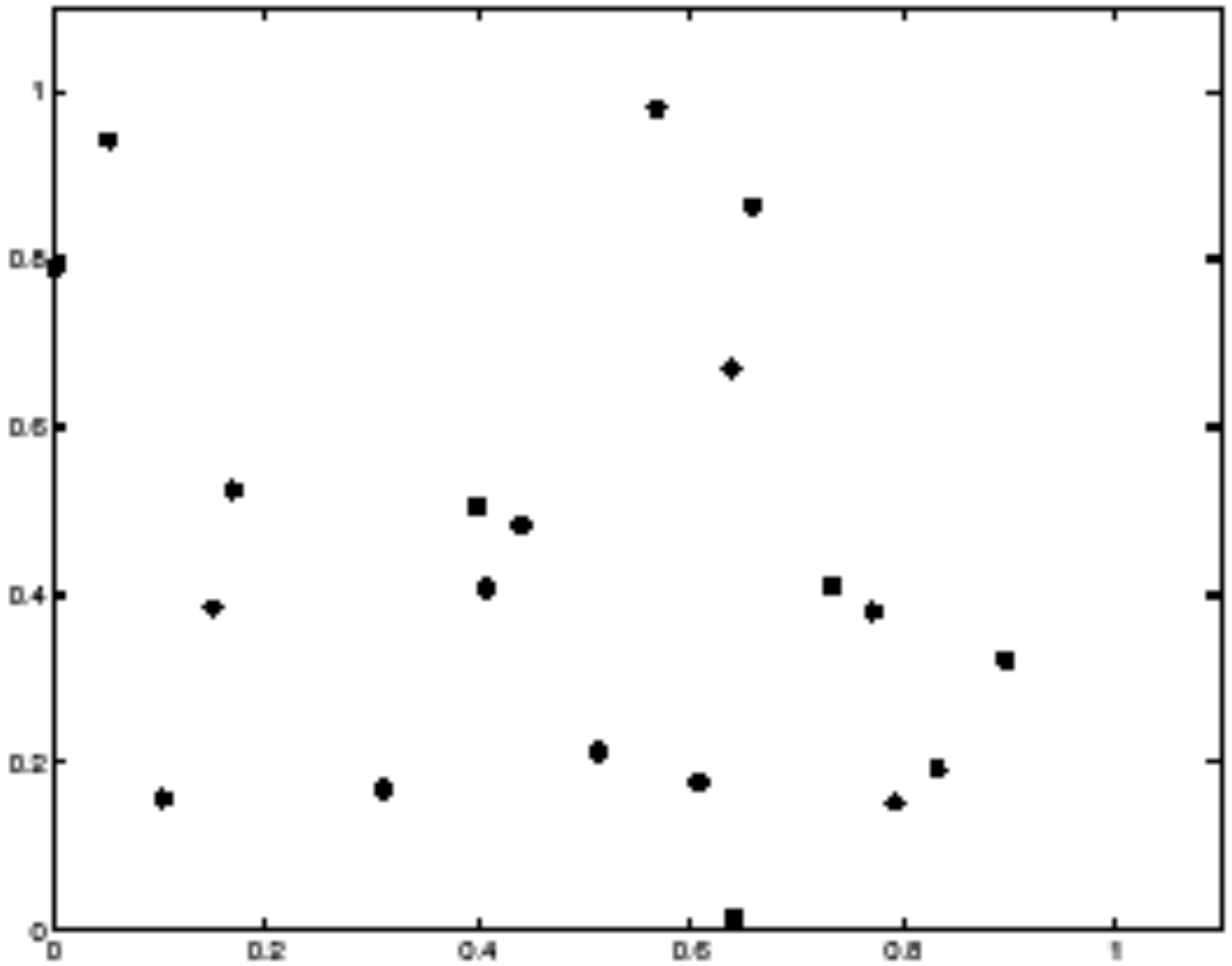
Votes

Horizontal axis is θ

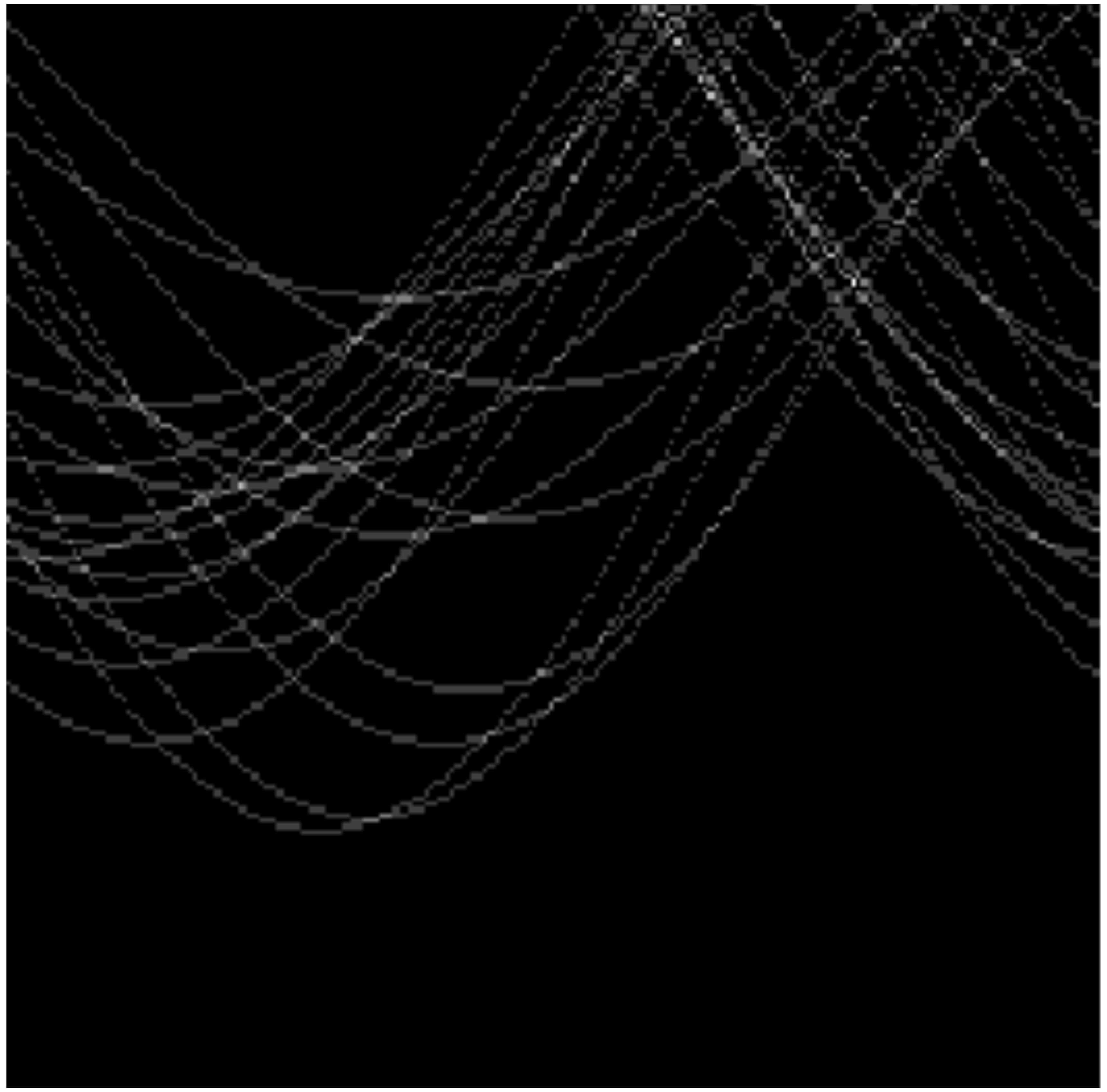
Vertical Axis is r

Forsyth & Ponce (2nd ed.) Figure 10.1 (Bottom)

Example: Too Much Noise



Tokens



Votes

Horizontal axis is θ

Vertical Axis is r

Forsyth & Ponce (2nd ed.) Figure 10.2

Sample Question

In his SIFT paper, why did Lowe choose to use a Hough transform rather than RANSAC to recognize clusters of 3 consistent features?

Course Review: Stereo

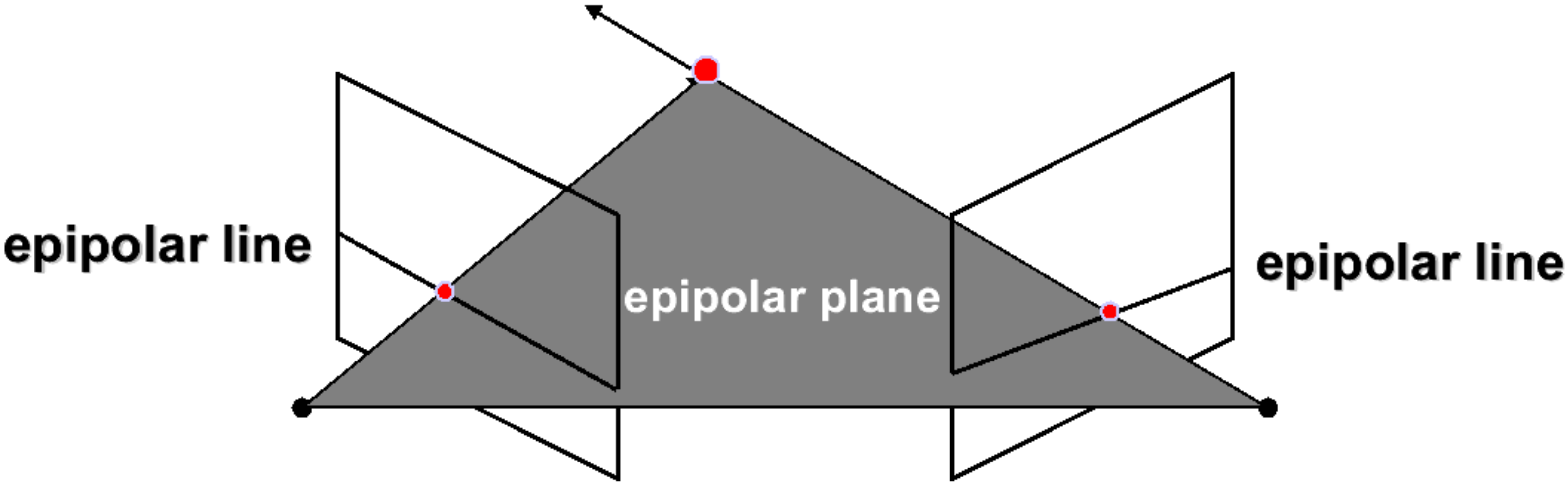
Epipolar constraint

Rectified images

Computing **correspondences**

Ordering constraint

The **Epipolar** Constraint



Matching points lie along corresponding epipolar lines

Reduces correspondence problem to 1D search along conjugate epipolar lines

Greatly reduces cost and ambiguity of matching

Slide credit: Steve Seitz

Simplest Case: **Rectified** Images

Image planes of cameras are **parallel**

Focal **points** are at same height

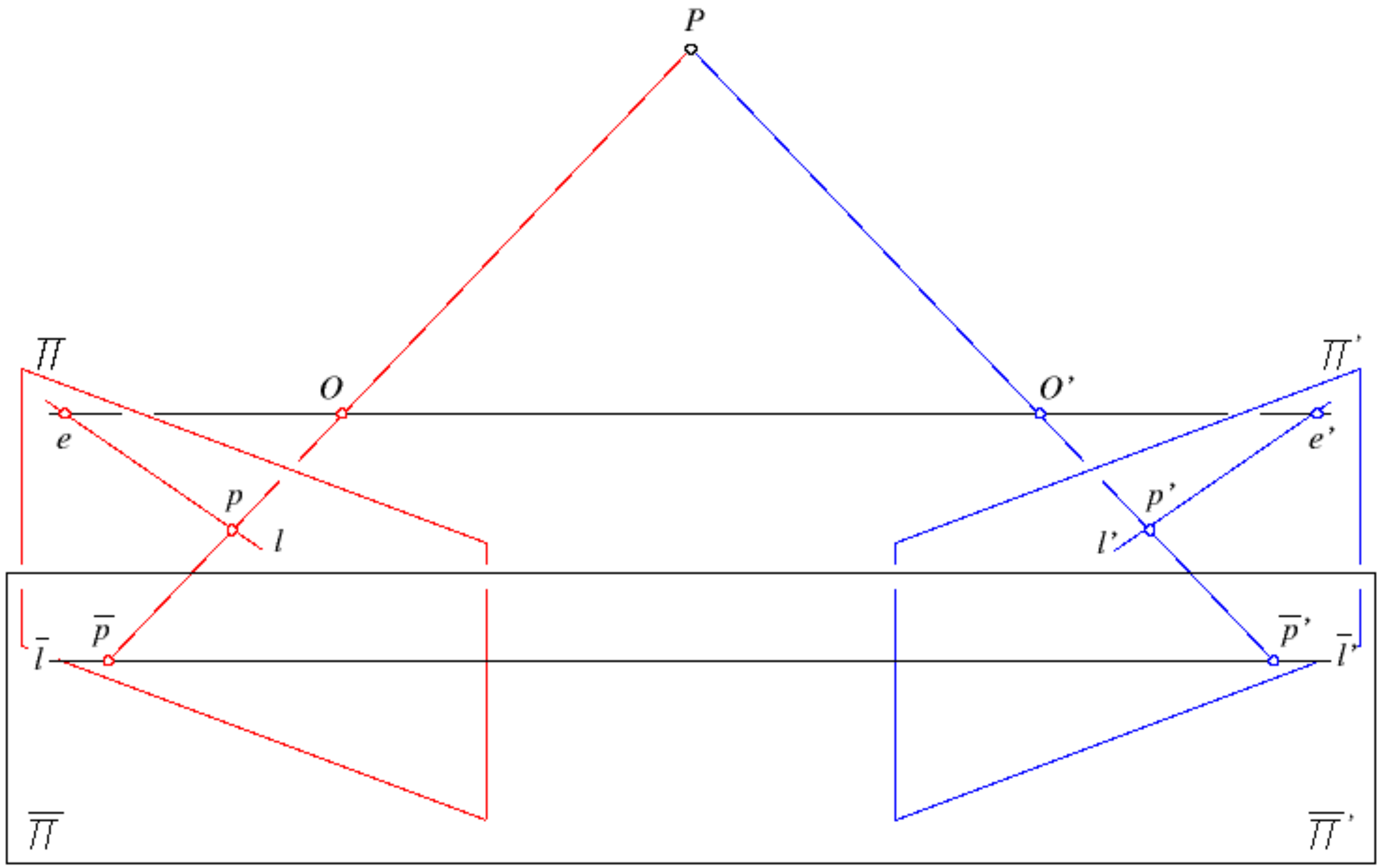
Focal **lengths** same

Then, **epipolar lines** fall along the **horizontal scan lines** of the images

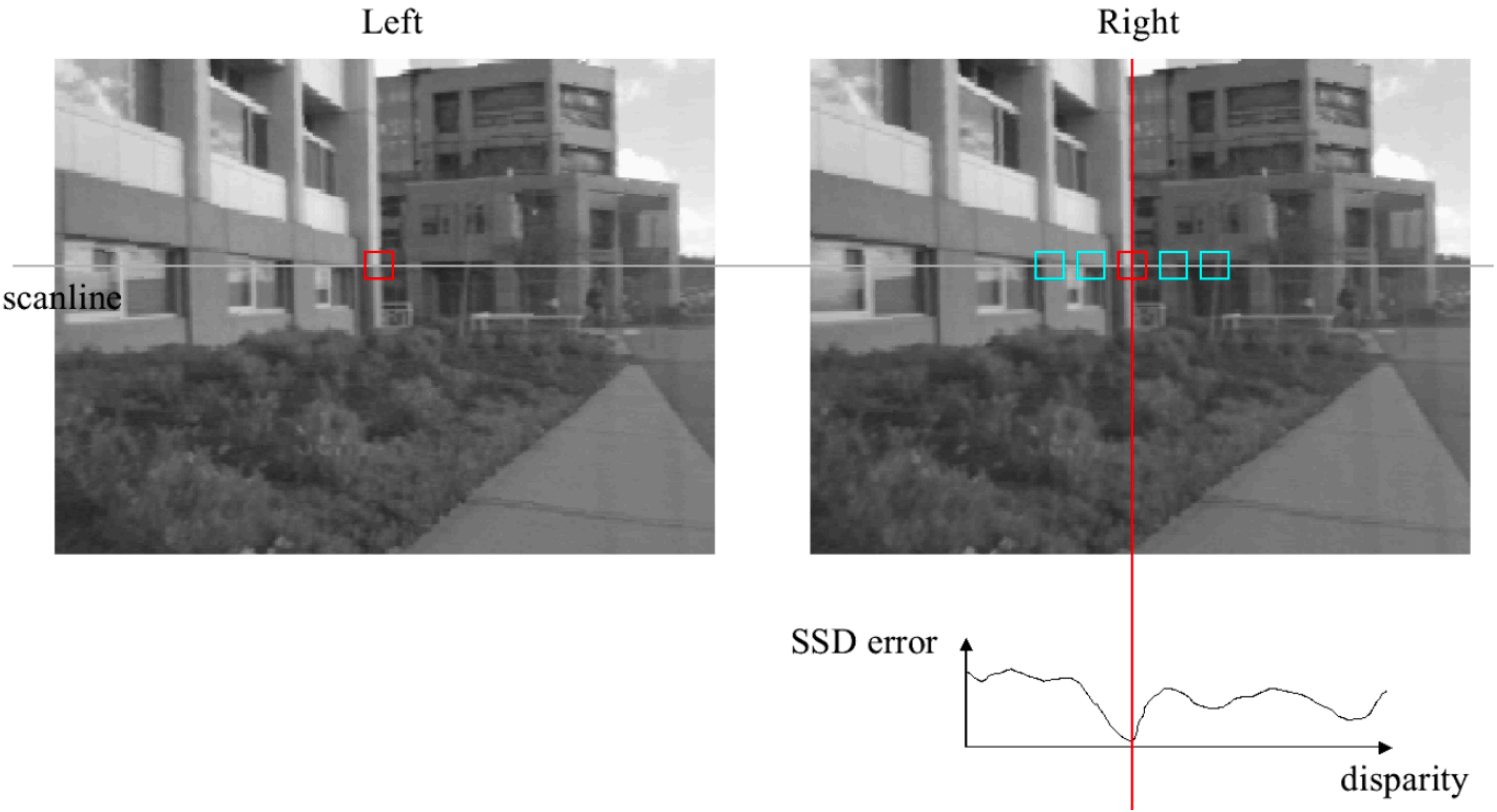
We assume images have been **rectified** so that epipolar lines correspond to scan lines

- Simplifies algorithms
- Improves efficiency

Rectified Stereo Pair

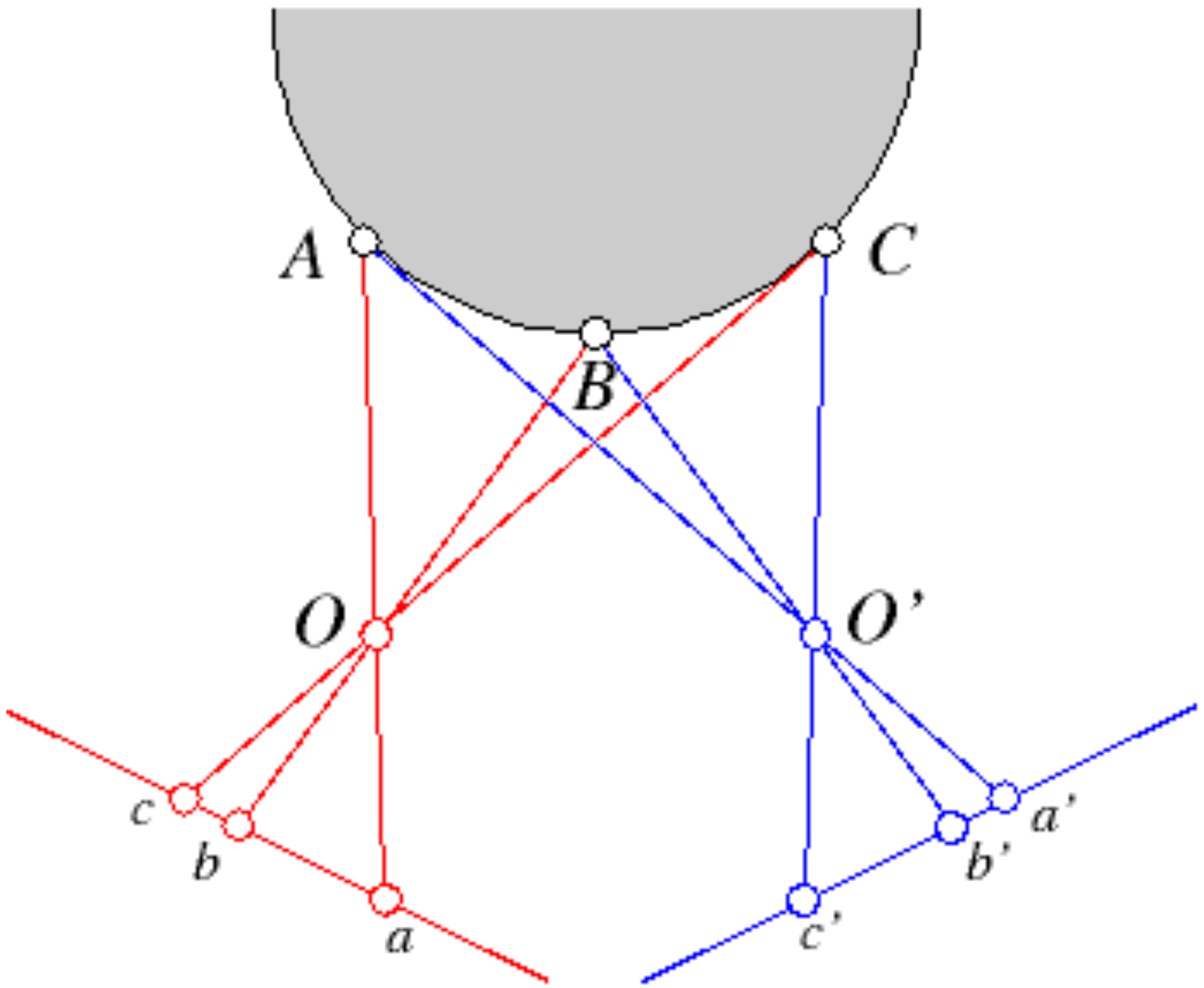


Method: Correlation

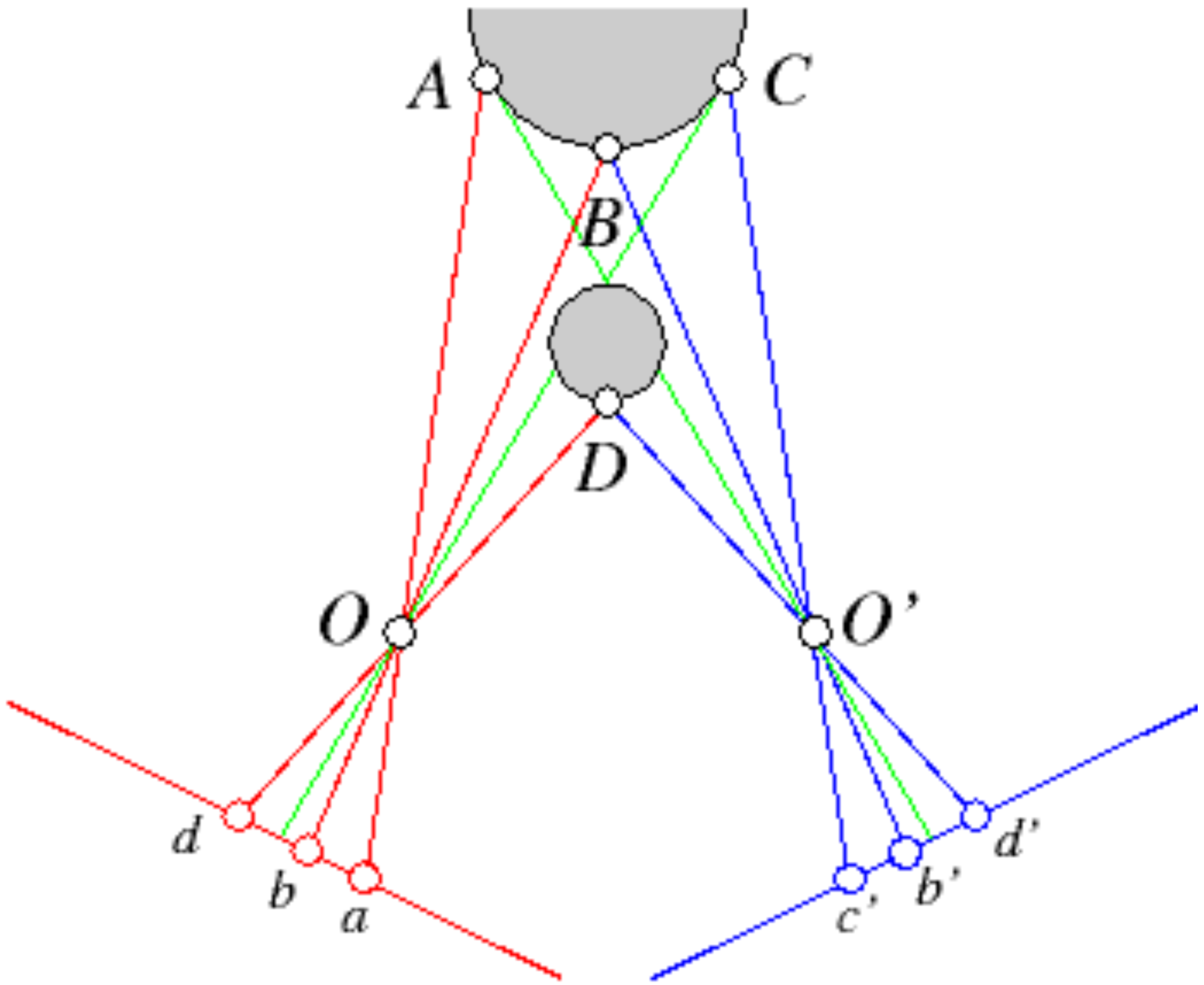


Ordering Constraints

Ordering constraint …



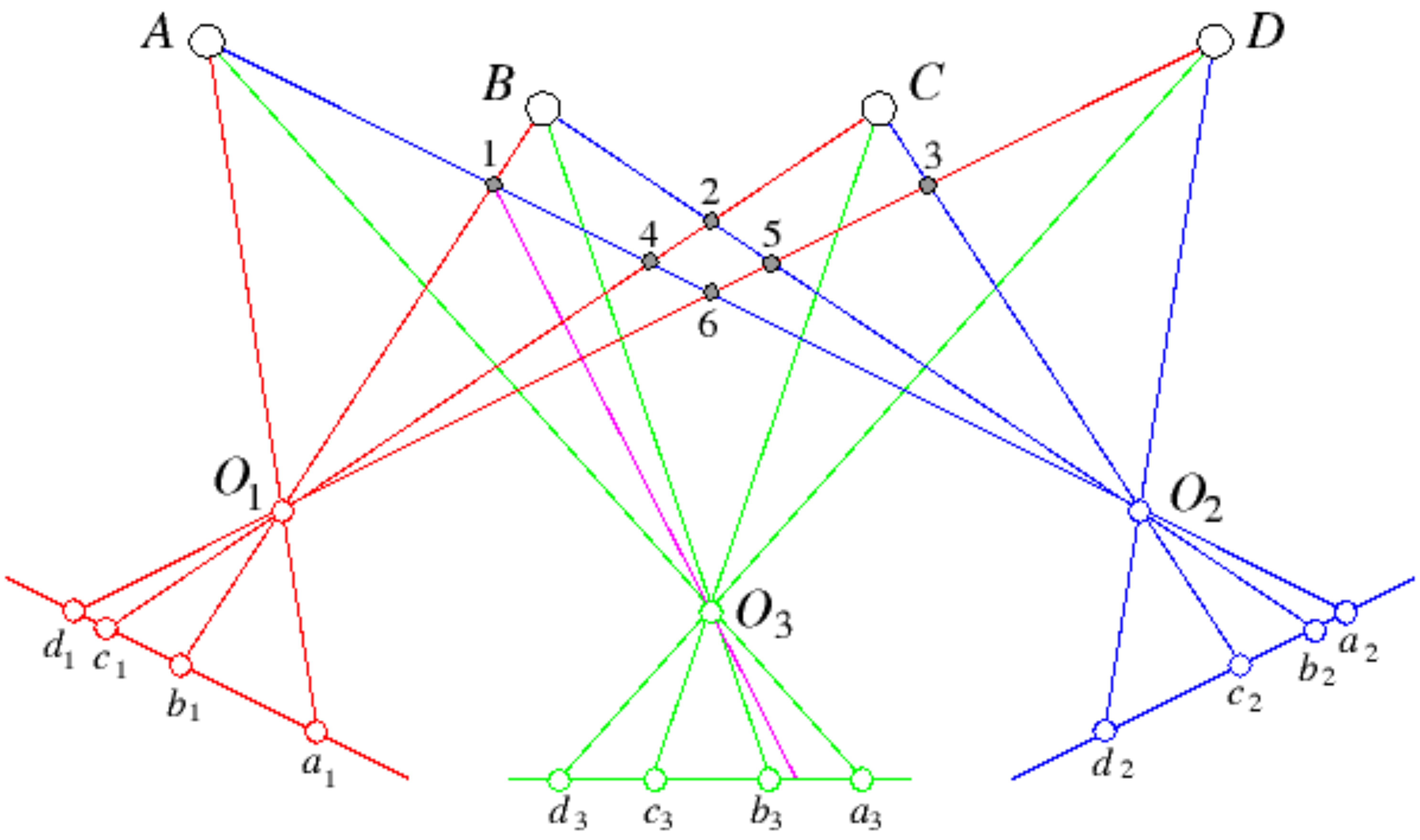
…. and a **failure** case



Forsyth & Ponce (2nd ed.) Figure 7.13

Idea: Use More Cameras

Adding a third camera reduces ambiguity in stereo matching



Forsyth & Ponce (2nd ed.) Figure 7.17

Please get your **iClickers** — Quiz

Sample Question

True or **false**: The ordering constraint always holds in stereo vision.

Course Review: Motion and Optical Flow

Motion (geometric), optical flow (radiometric)

Optical flow constraint equation

Lucas-Kanade method

Optical Flow **Constraint Equation**

Consider image intensity also to be a function of time, t. We write

$$I(x, y, t)$$

Applying the **chain rule for differentiation**, we obtain

$$\frac{dI(x, y, t)}{dt} = I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t$$

where subscripts denote partial differentiation

Define $u = \dfrac{dx}{dt}$ and $v = \dfrac{dy}{dt}$. Then $[u, v]$ is the 2-D motion and the space of all such u and v is the **2-D velocity space**

Suppose $\dfrac{dI(x, y, t)}{dt} = 0$. Then we obtain the (classic) **optical flow constraint equation**

$$I_x u + I_y v + I_t = 0$$

How do we **compute** …

$$I_x u + I_y v + I_t = 0$$

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

Forward difference
Sobel filter
Scharr filter
…

Frame differencing

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Frame Differencing: Example

$$t+1 \qquad\qquad t \qquad\qquad I_t = \frac{\partial I}{\partial t}$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	10	10	10
1	1	10	10	10
1	1	10	10	10

-

1	1	1	1	1
1	1	1	1	1
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10

=

0	0	0	0	0
0	0	0	0	0
0	-9	-9	-9	-9
0	-9	0	0	0
0	-9	0	0	0
0	-9	0	0	0

(example of a forward temporal difference)

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

t

1	1	1	1	1
1	1	1	1	1
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10

$t+1$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	10	10	10
1	1	10	10	10
1	1	10	10	10

$$I_x = \frac{\partial I}{\partial x}$$

-	0	0	0	-
-	0	0	0	-
-	9	0	0	-
-	9	0	0	-
-	9	0	0	-
-	9	0	0	-

-1 0 1

$$I_y = \frac{\partial I}{\partial y}$$

-	-	-	-	-
0	0	0	0	0
0	9	9	9	9
-	0	0	0	0
0	0	0	0	0
-	-	-	-	-

-1
0
1

$$I_t = \frac{\partial I}{\partial t}$$

0	0	0	0	0
0	0	0	0	0
0	-9	-9	-9	-9
0	-9	0	0	0
0	-9	0	0	0
0	-9	0	0	0

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

How do we **compute** …

$$I_x u + I_y v + I_t = 0$$

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

$$u = \frac{dx}{dt} \quad v = \frac{dy}{dt}$$

optical flow

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

Forward difference
Sobel filter
Scharr filter
…

How do you compute this?

Frame differencing

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Lucas-Kanade **Summary**

A dense method to compute motion, $[u, v]$ at every location in an image

Key Assumptions:

1. Motion is slow enough and smooth enough that differential methods apply (i.e., that the partial derivatives, I_x, I_y, I_t, are well-defined)

2. The optical flow constraint equation holds (i.e., $\frac{dI(x,y,t)}{dt} = 0$)

3. A window size is chosen so that motion, $[u, v]$, is constant in the window

4. A window size is chosen so that the rank of $\mathbf{A}^T\mathbf{A}$ is 2 for the window

Please get your **iClickers** — Quiz

Sample Question

Describe two examples of imaging situations where motion and optical flow do not coincide.

Course Review: Clustering

Two basic approaches: **agglomerative** and **divisive** clustering

Dendrograms

Inter-cluster distance measures

K-means clustering

Segmentation by clustering

Course Review: Classification

Bayes' risk, loss functions

Underfitting, overfitting

Cross-validation

Receiver Operating Characteristic (**ROC**) curve

Parametric vs. **non-parametric** classifiers
— K-nearest neighbour
— Support vector machines
— Decision trees

Course Review: Image Classification

Visual words, codebooks

Bag of words representation

Spatial pyramid

VLAD

Sample Question

How do we construct a codebook (vocabulary) of local descriptors, say SIFT?

Course Review: Object Detection

Sliding window

Viola-Jones face detection

Deformable **part model**

Object **proposals**

Course Review: Convolutional Neural Networks

Neuron, activation function

Backpropagation (you only need to know properties)

Convolutional neural network architecture

Convolutional neural network layers

R-CNN

Hope you enjoyed the **course**!