EECS 531
Computer Vision

Introduction and Overview

Course Information

- **Instructor**:
  Dr. Michael Lewicki, EECS/CWRU
  Office: Olin 508
  Office hours: TBA
  email: michael.lewicki@case.edu
- **Teaching Assistants:**
  Jing Chen: jxc761@case.edu
  Office hour: Friday and Saturday 1:00 - 3:00 PM in Olin513

  Morgan Zhu: hxz244@case.edu
  Office hour: Tuesday and Thursday 1:00 - 3:00 PM in Olin513
- **Class meeting times**:
  MW 3:20 - 4:35 PM in White 411.
- **Web page**:
  canvas: https://canvas.case.edu/courses/7804
  redmine (git) server: https://csevcs.case.edu/

Course Goals

- develop a practical understanding of problems in computer vision
- reason scientifically about problems and issues
- understand mathematics of essential computations and algorithms
- develop and test effective algorithms
- discuss, analyze, and critique of course readings and topics

Course design

- Exercise and project based. No exams. You learn by doing.
  - Assignments: 5 assignments, 15% each, 75% total
  - Class Project: 25% (required for graduate students, optional for undergrads, but required for an A)
- Goal is for you to be able to teach yourself.
- Emphasize creative exploration
- Main difficulty: breaking down complex concepts into manageable chunks.
  - no one should get stuck!
  - wide range of interests and backgrounds
  - will introduce necessary concepts in earlier assignments

Assignments and Grading

- Assignments will consist of a small number of simple exercises.
  - Use git repository for exercises
  - Preferred form for assignments will be notebooks (default python in jupyter)
    - Latex/Markdown + other programming language
    - Needs to have version control. No binary formats are allowed.
  - 2 submissions and 2 discuss sessions
- The notebooks you will create for each exercise in an assignment will consist of
  - conceptual background (text)
  - mathematical background (equations, must use latex)
  - code (e,g., python)
  - plots or results (could be table)
- Grading based on
  - The final portfolio and your progress
  - The feedback and review you get and given
  - How you addressed feedback from discussion sessions

Projects and Grading

- Explanation and motivation (50%)

  What problem are you solving? What issue or topic are you investigating? Why is it interesting? What does success mean? How do we know if you are successful?

- Approach and rationale (30%)

  What approach are you taking? Why does your this make sense? Why did you choose this approach. How did you have to simplify the problem or limit the scope of the project?

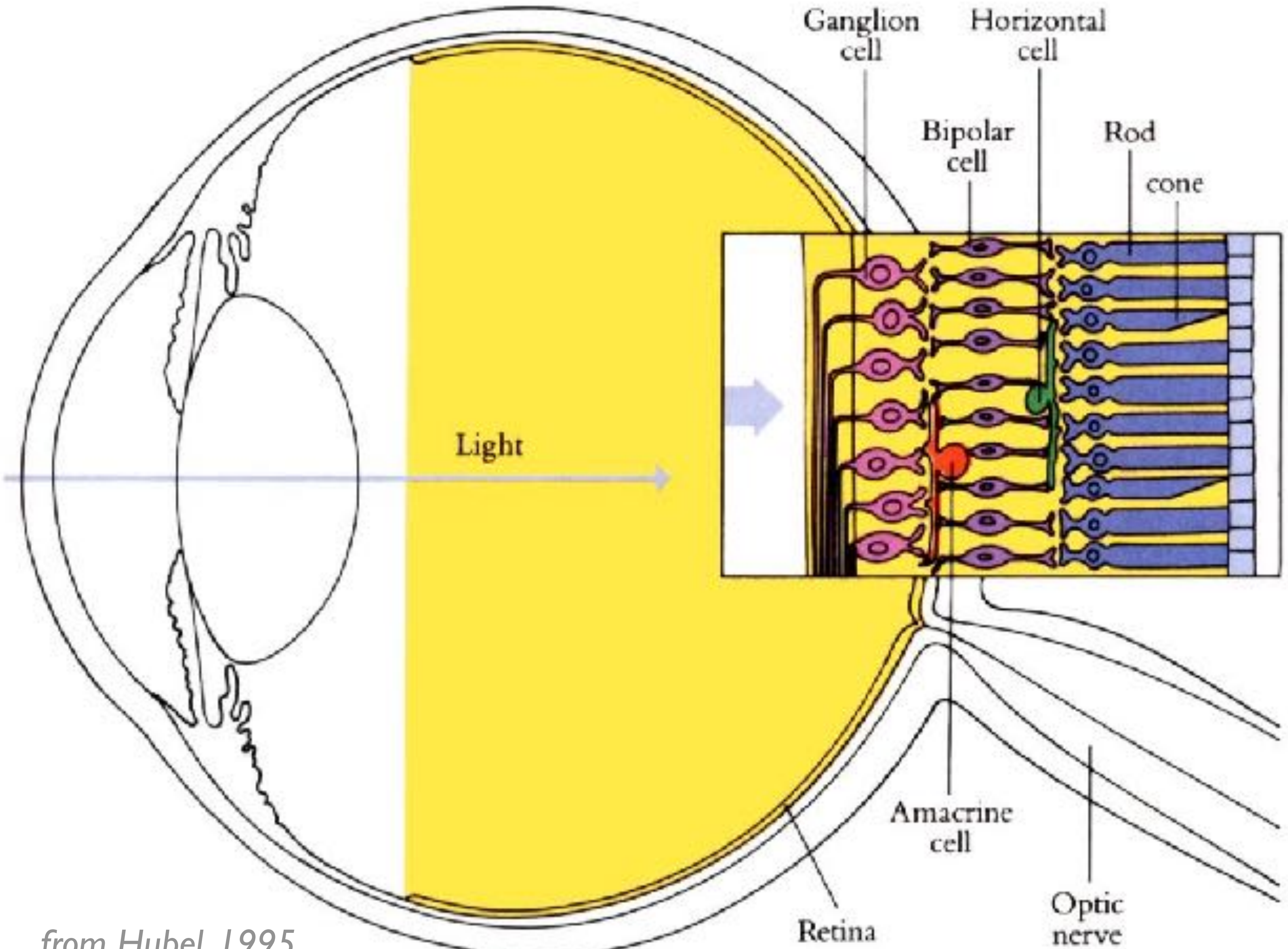
- Findings and results (20%)

  What are the take home messages from your project? Did your approach "work" ? Why or why not? How could it be improved? What did you learn from the project? What other approaches might you consider if did further work on it.

Collaboration and Cheating

- Collaboration to discuss the assignments and problems is encouraged.
- You *must* generate and write up the solutions yourself
- We expect that you can write on the explanation and code yourself, starting from a blank page without reference.
- Anything less is cheating.
- You are not allowed to copy code.

Eye anatomy

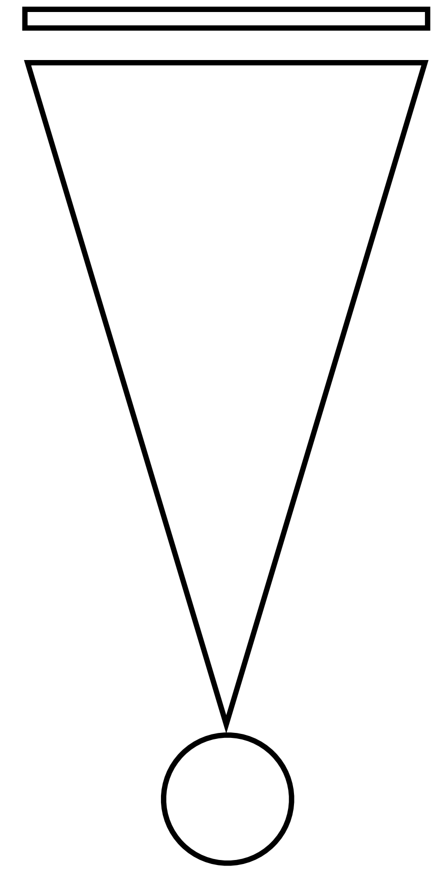


from Hubel, 1995

- Photoreceptors: rods (night vision) and cones (day vision)
- Other layers: processing to enhance SNR and maximize information transmission
- All visual information conveyed to brain through ~1 million optic nerve fibers

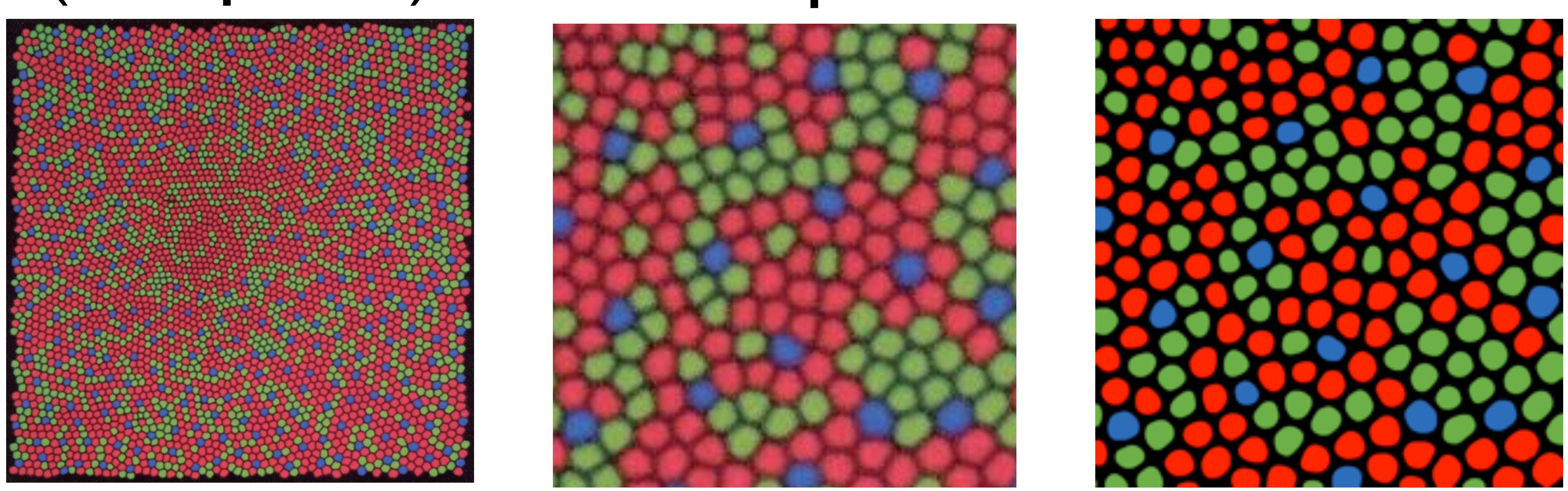
The image on this screen spans a visual angle of roughly 5-15 degrees.

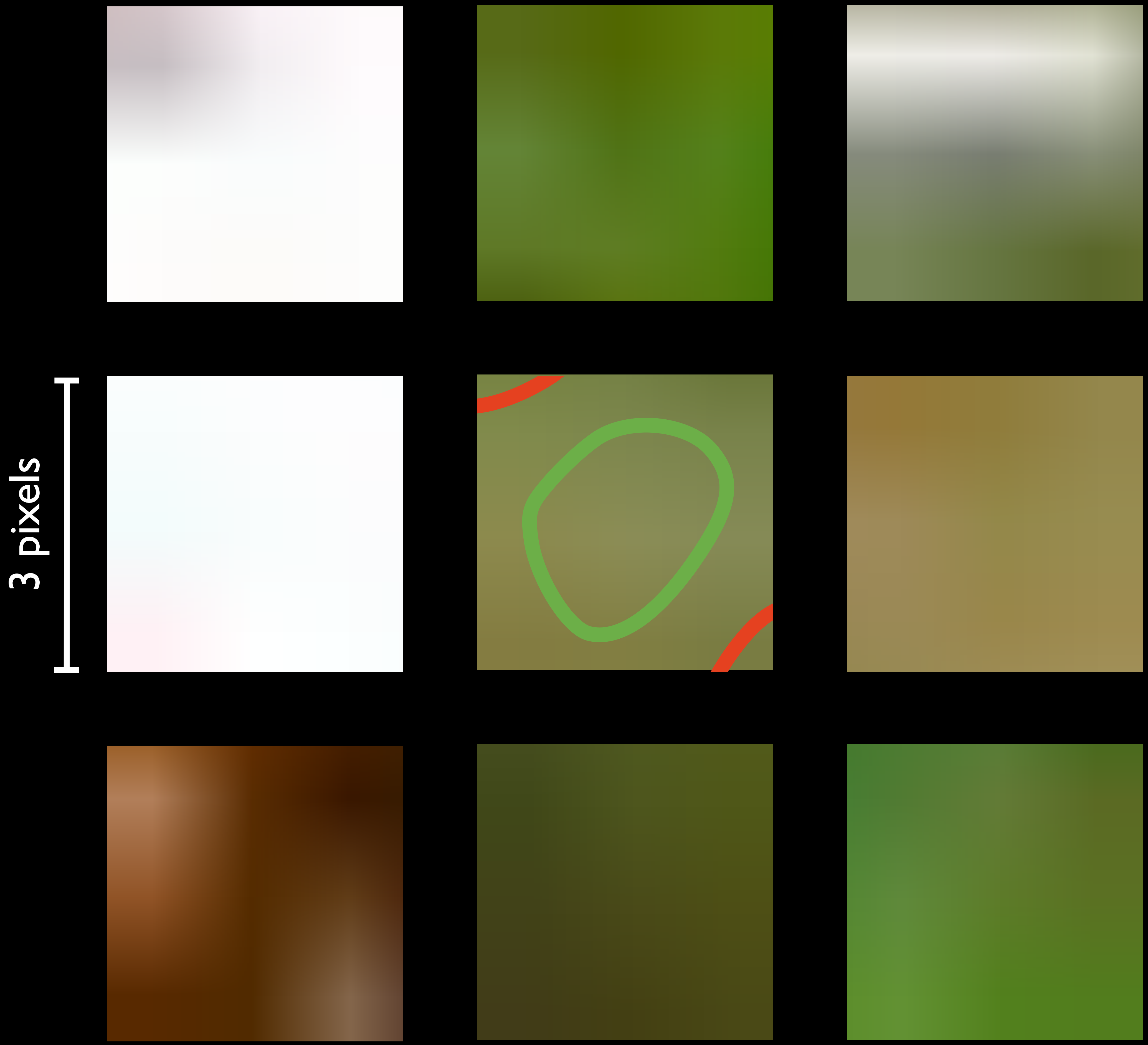
This is approximately 0.005 and 0.015 degrees per pixel.
So 1° is between 66 and 200 pixels.



1 degree
(200 pixels)

40x40 pixels




3 pixels

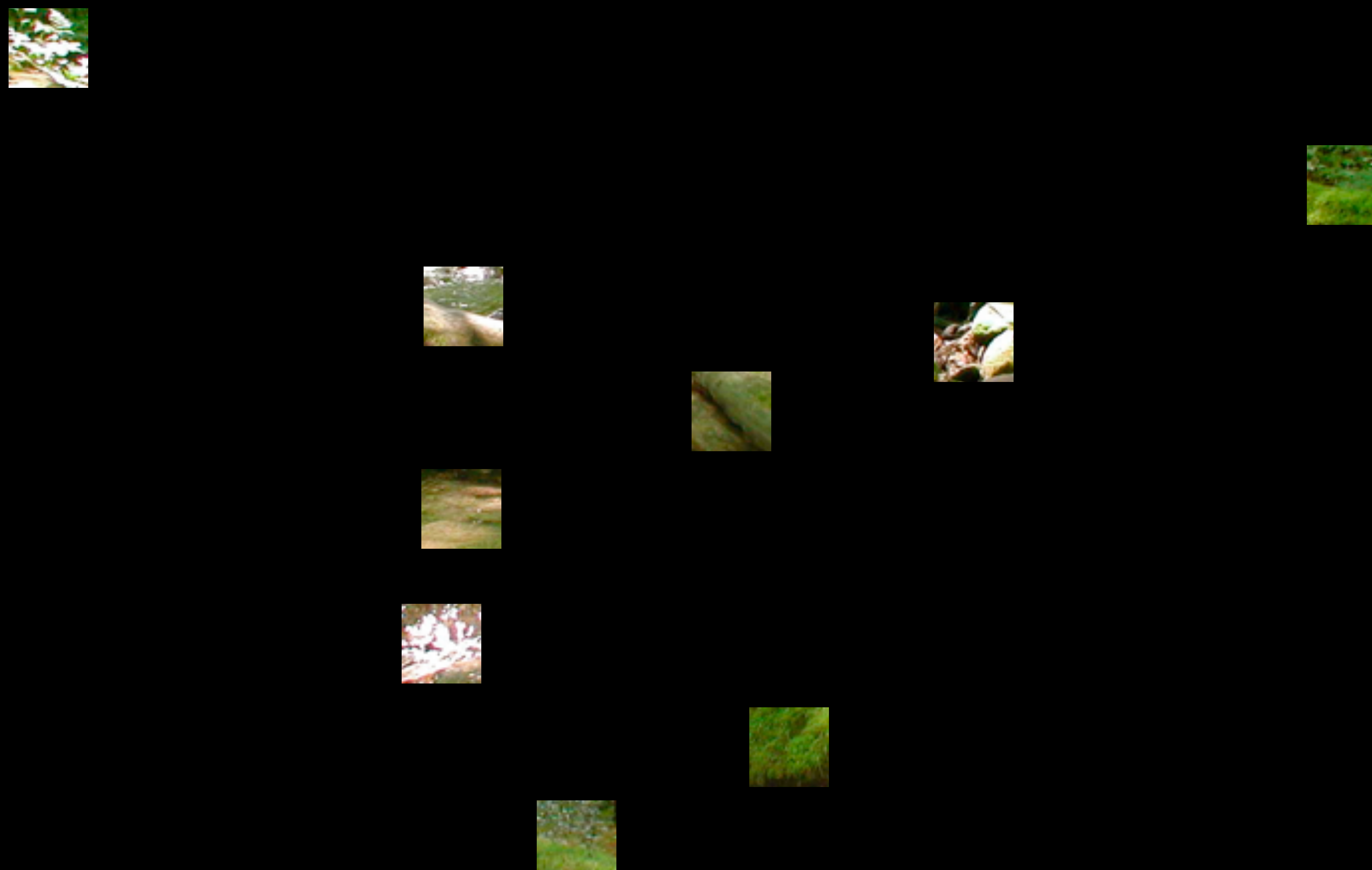

5 pixels

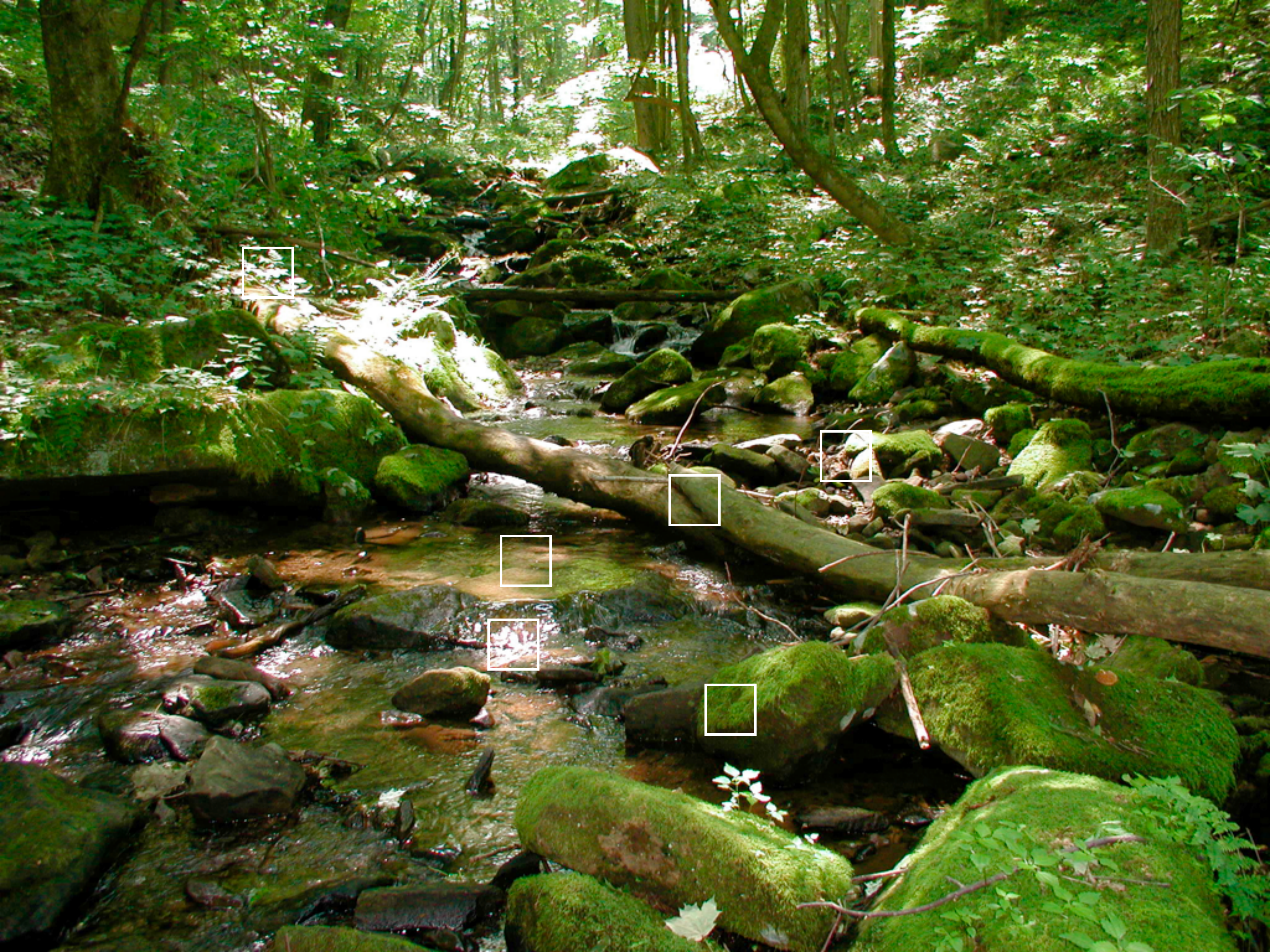

10 pixels

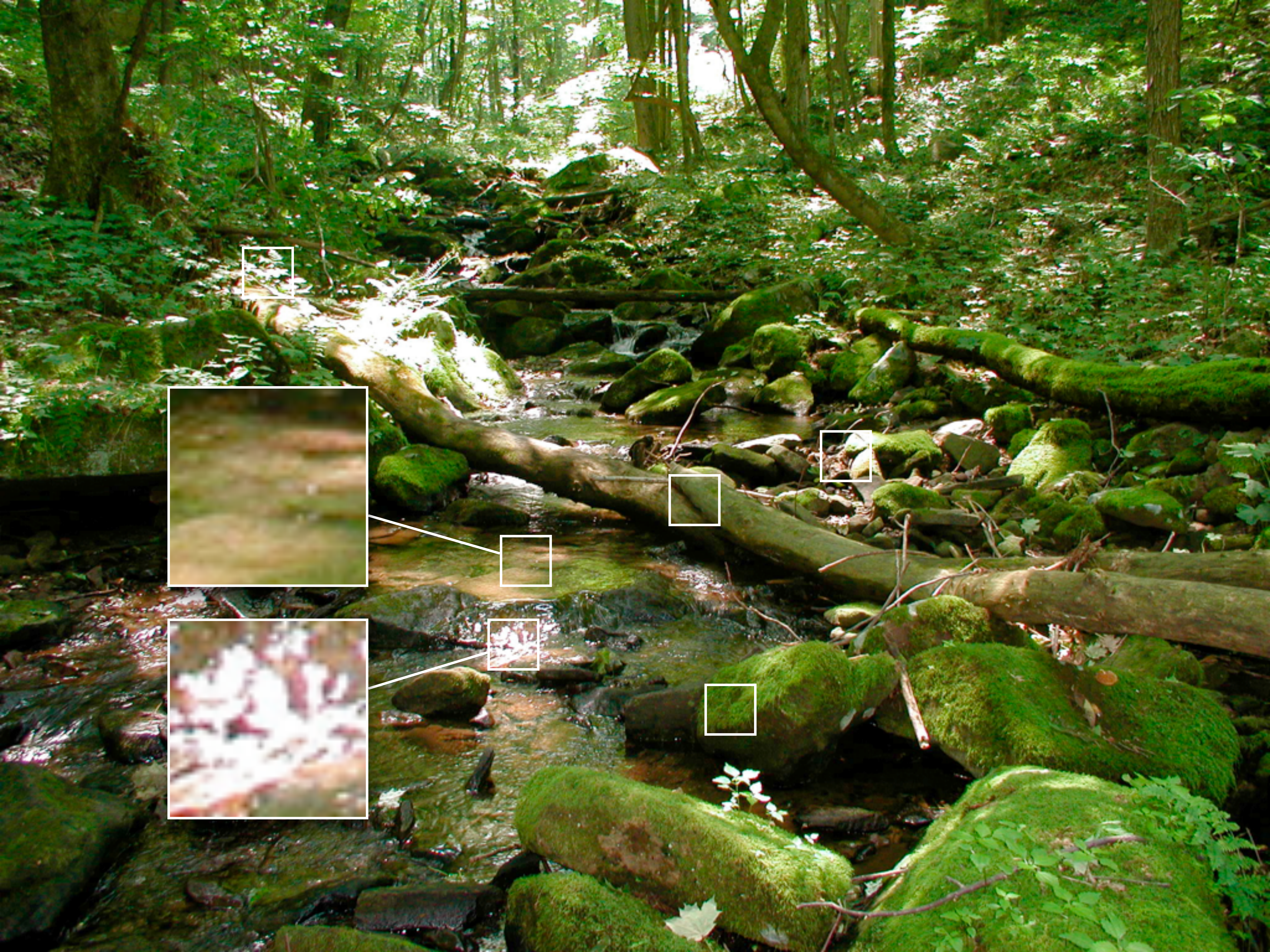

20 pixels

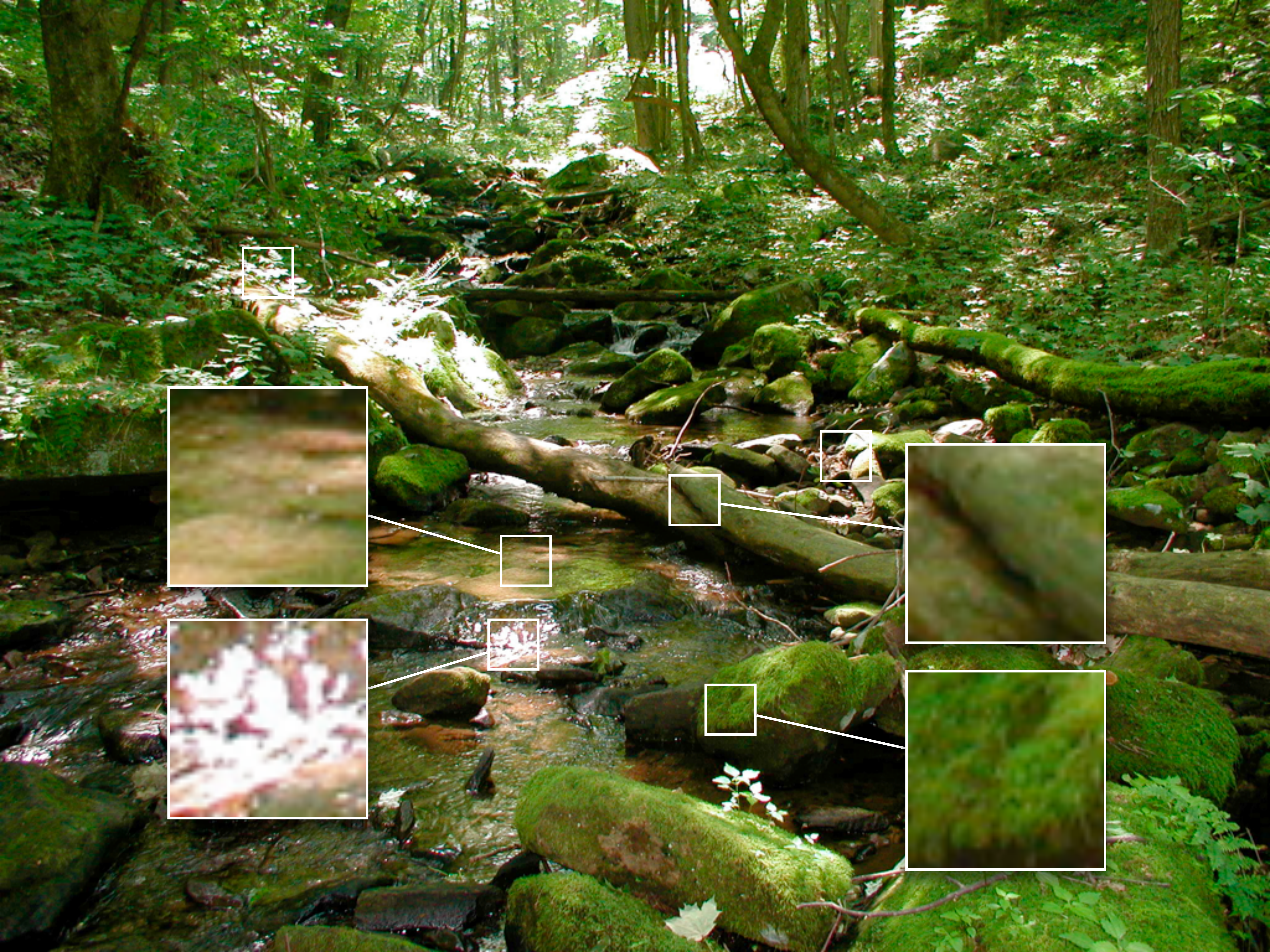

30 pixels

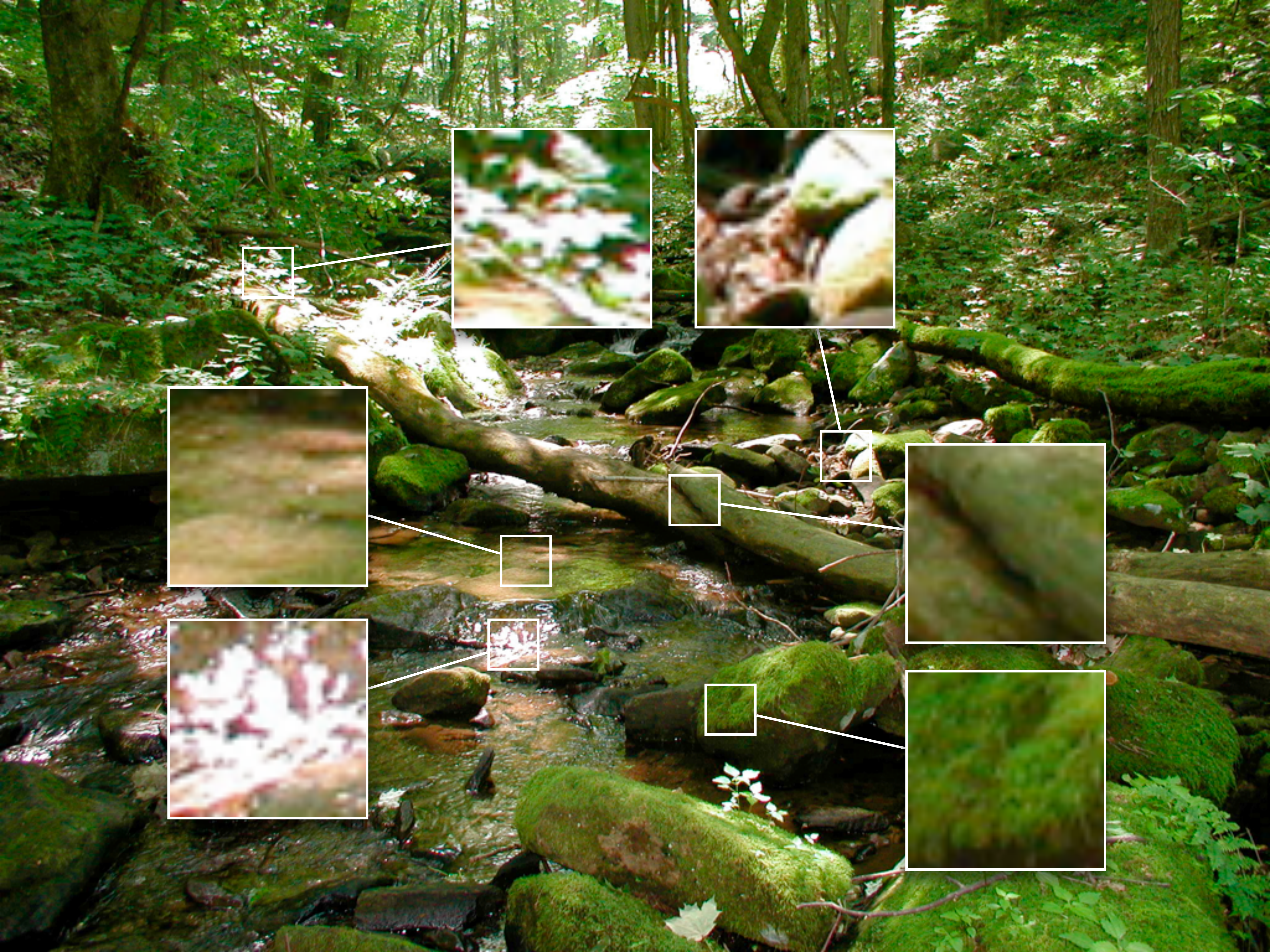

40 pixels



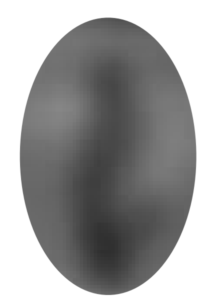








Contextual inference



example courtesy of Antonio Torralba

Contextual inference

example courtesy of Antonio Torralba

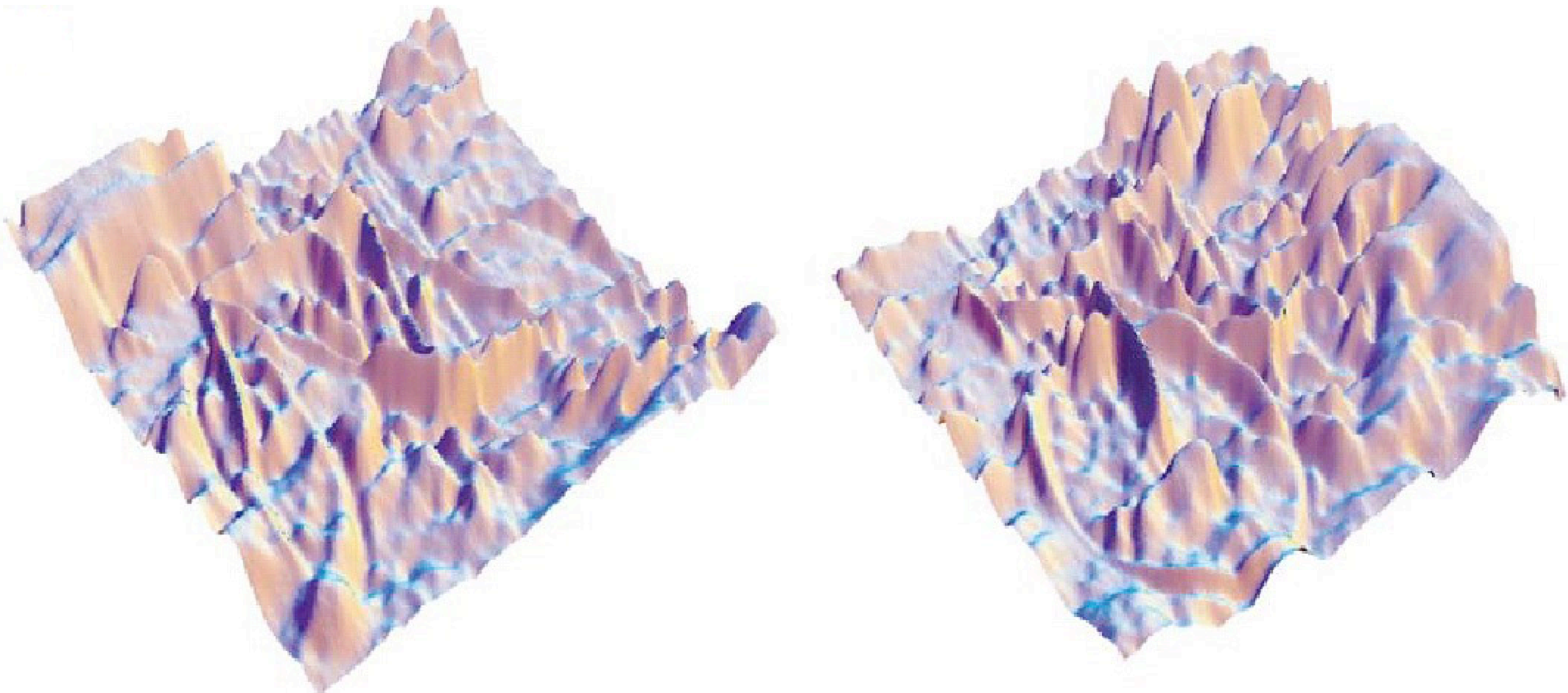
Contextual inference

example courtesy of Antonio Torralba

Contextual inference

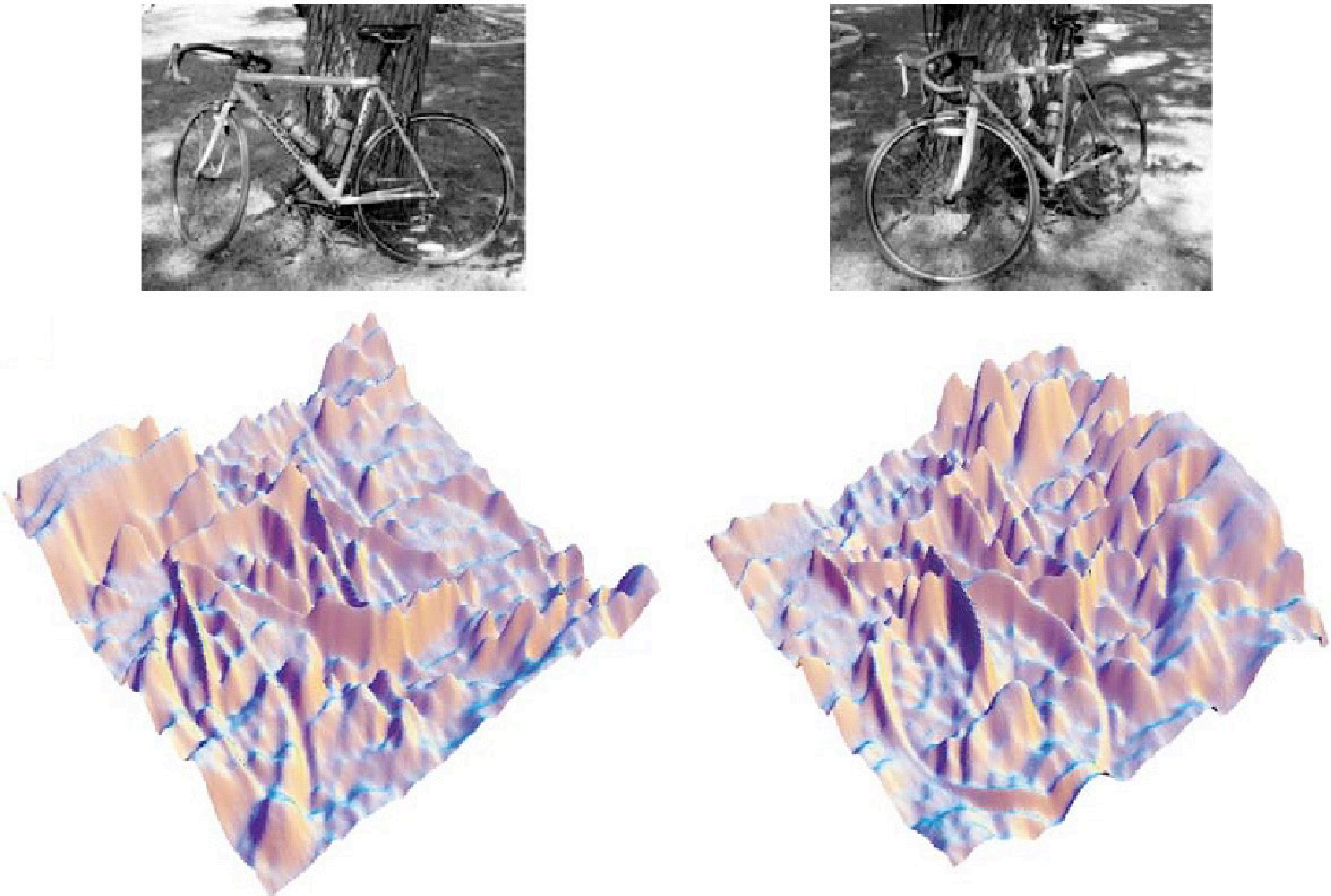
example courtesy of Antonio Torralba

Where are these objects?



from Kersten and Yuille, 2003

How should “parts” be grouped?



from Kersten and Yuille, 2003

Grouping parts: more difficult than it appears

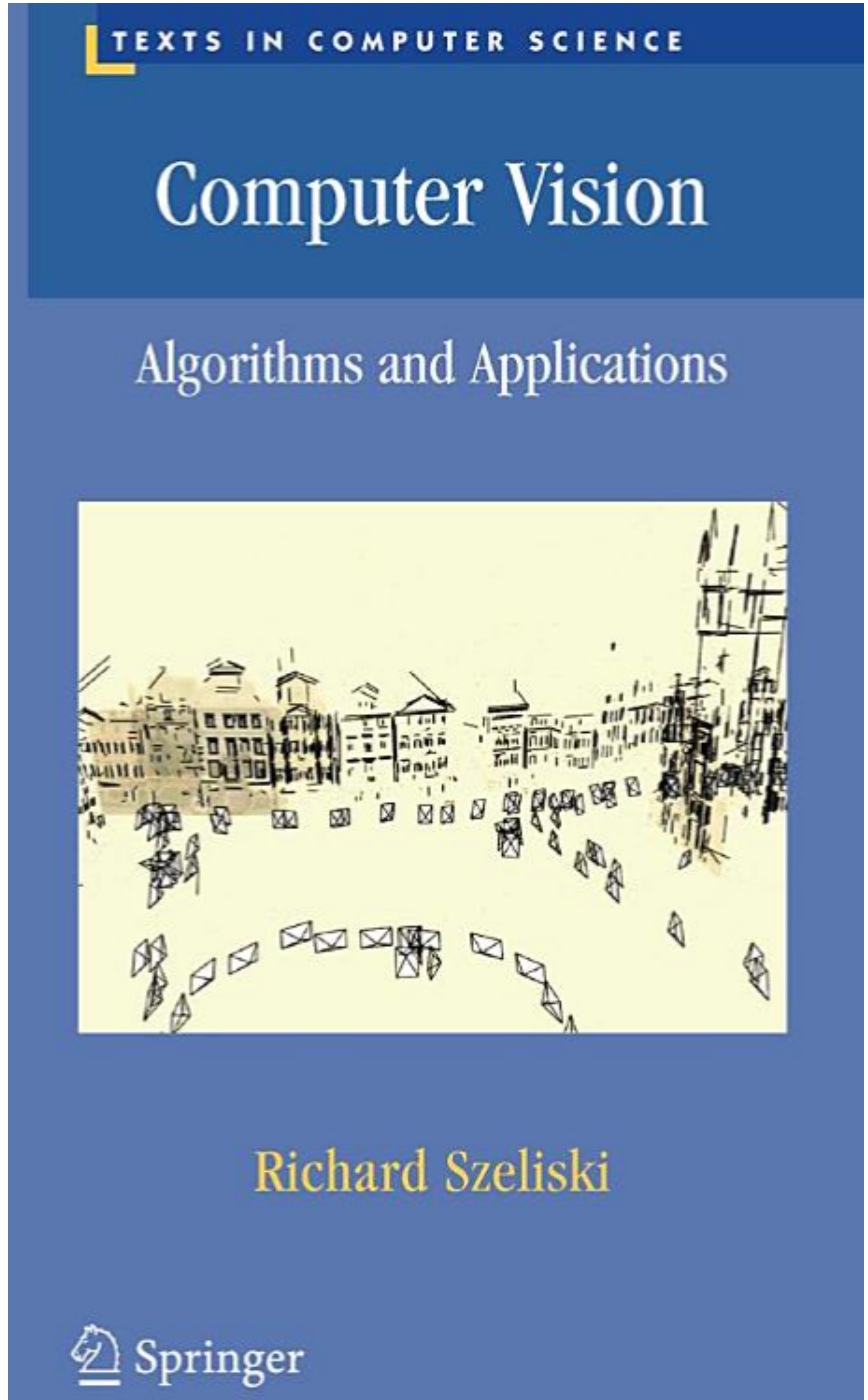
We’re still just passing this information along to our visual system,
which is where the real work is done.

Vision is *inference* about the external world.

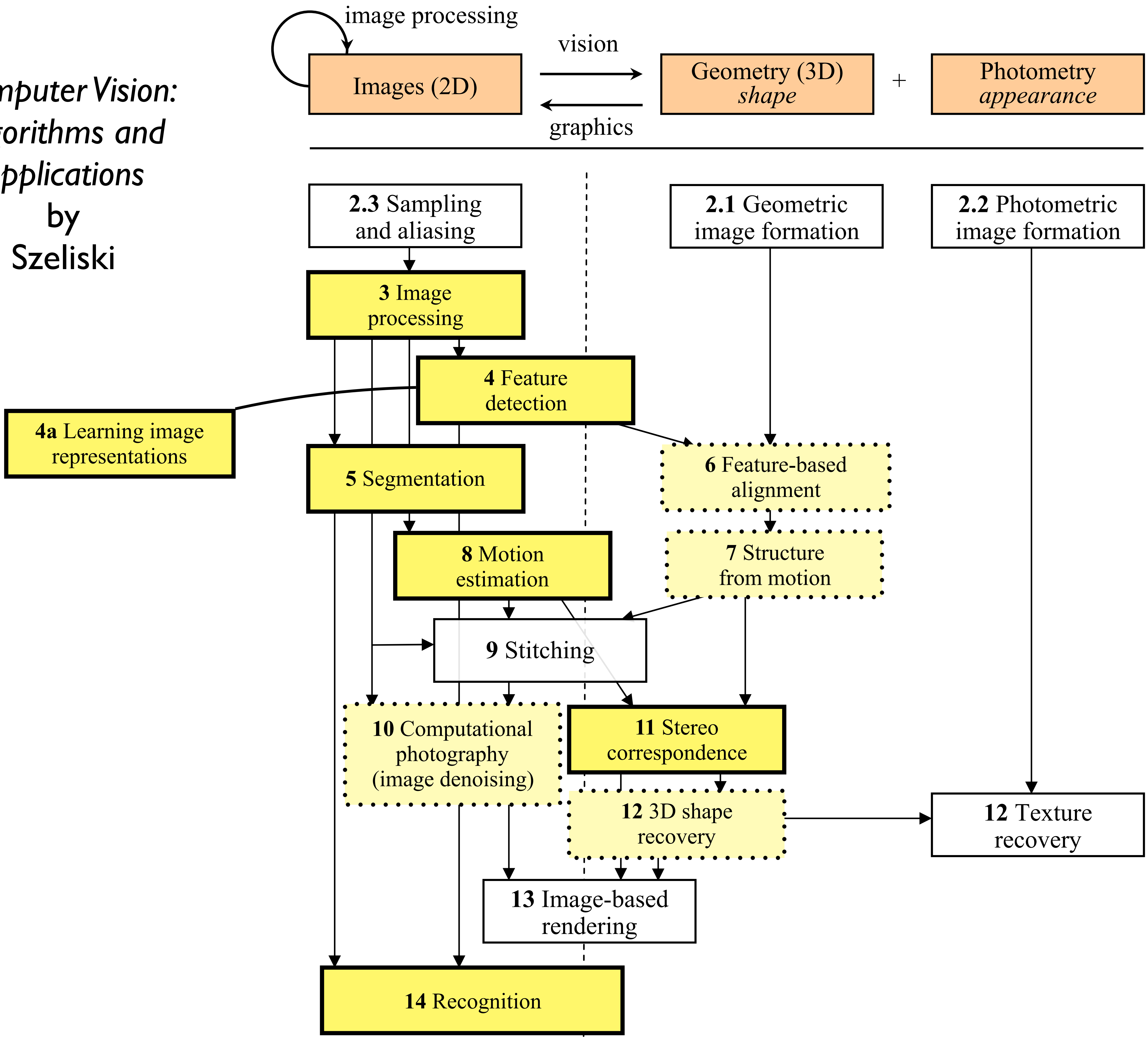
It is *not* simply feature detection.

Our main textbook.

Available electronically:
http://szeliski.org/Book/



Computer Vision: Algorithms and Applications
by
Szeliski



Another, recent (2012) computer vision textbook …
based on probabilistic inference and machine learning

But the author states:
"I am aware that most people will not learn computer vision from this book alone."

"I should say that I found most of the ideas in this book very hard to grasp when I was first exposed to them. My goal was to make this process easier for subsequent students following the same path."

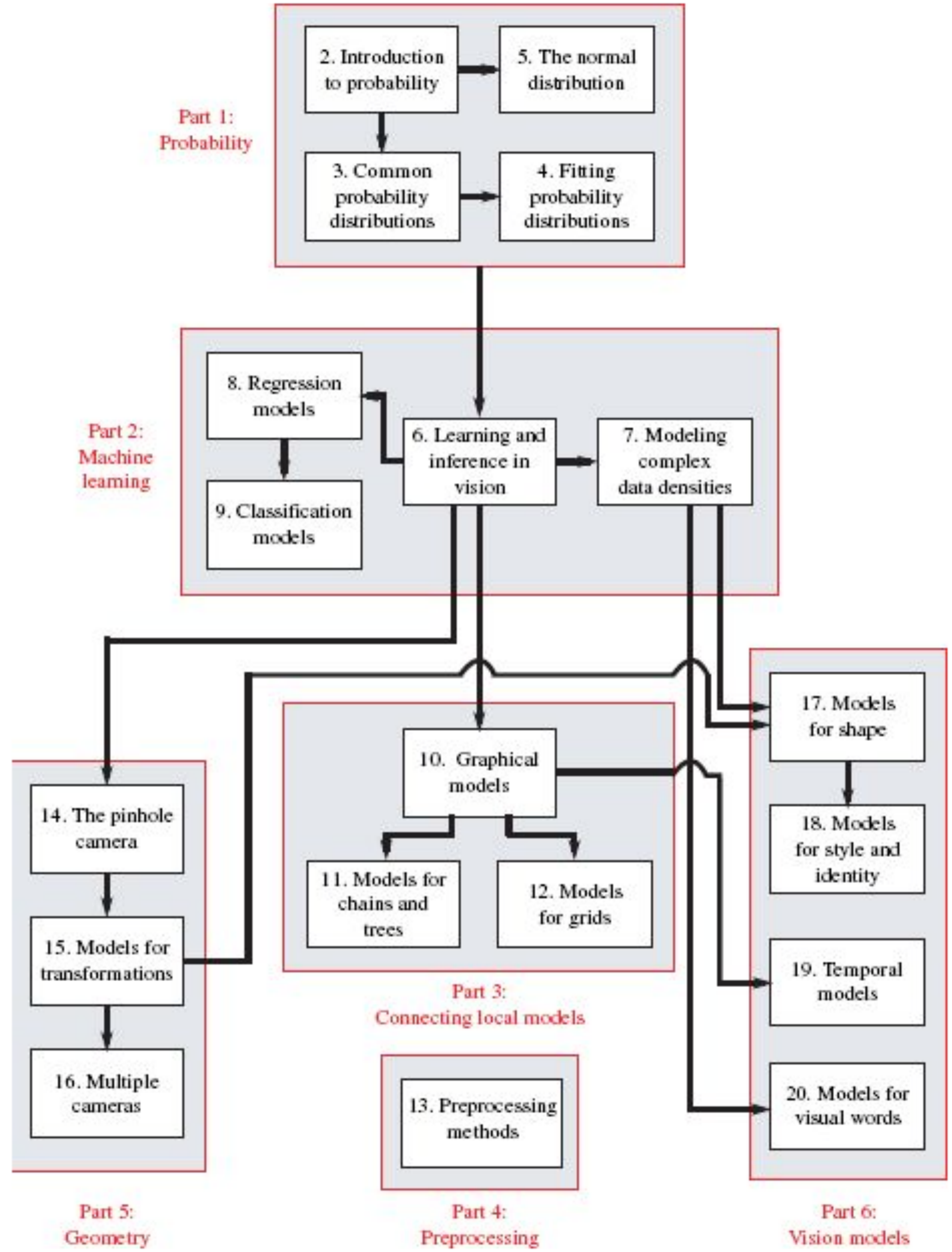


Image Processing

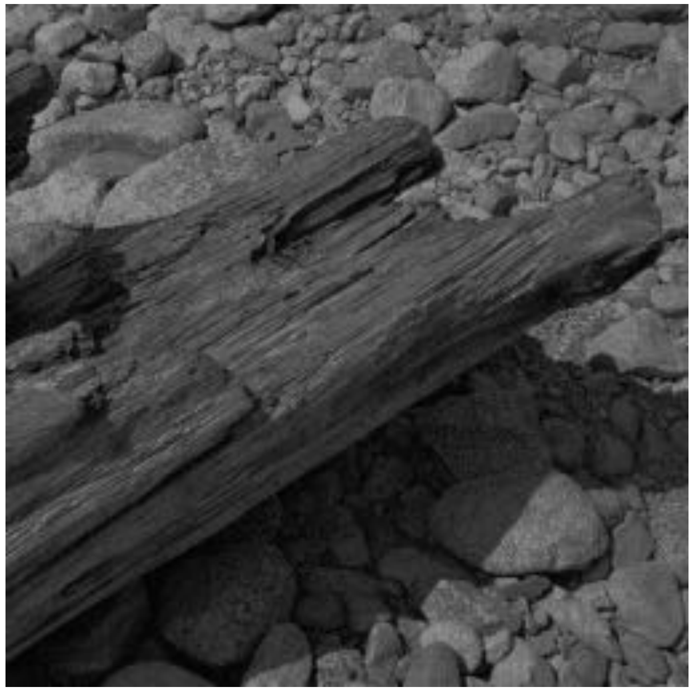
- image sampling
- image transformations, filtering, and compression (affine, Fourier, wavelet)
- image enhancement and restoration (sharpening, smoothing, color normalization, denoising)
- feature detection (points, lines, edges)
- image segmentation
- image classification
- computational photography: image stitching, HDR imagining, super-resolution

Computer Vision

- imagine formation from 3D scene, computer graphics rendering, camera models
- feature detection (edges, contours, 3D features, faces)
- image segmentation
- stereo, multiple-view geometry
- 3D structure from motion, shading, texture
- visual motion, optical flow, and object tracking
- visual navigation
- visual scene analysis

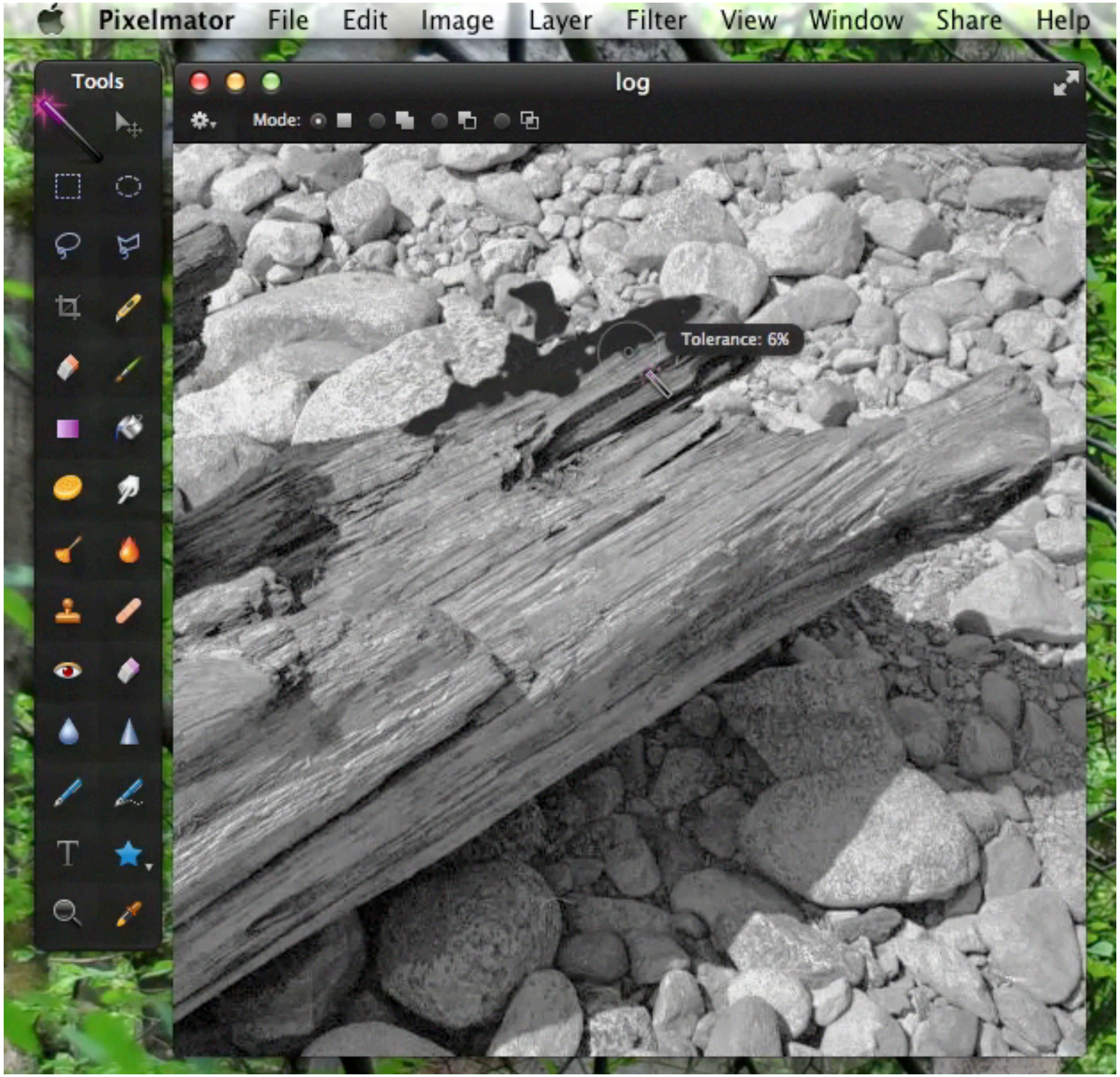
While there is some overlap, image processing is about the *transformation* of the 2D image, whereas computer vision is about the *inference* of (3D) scene properties.

Image Processing

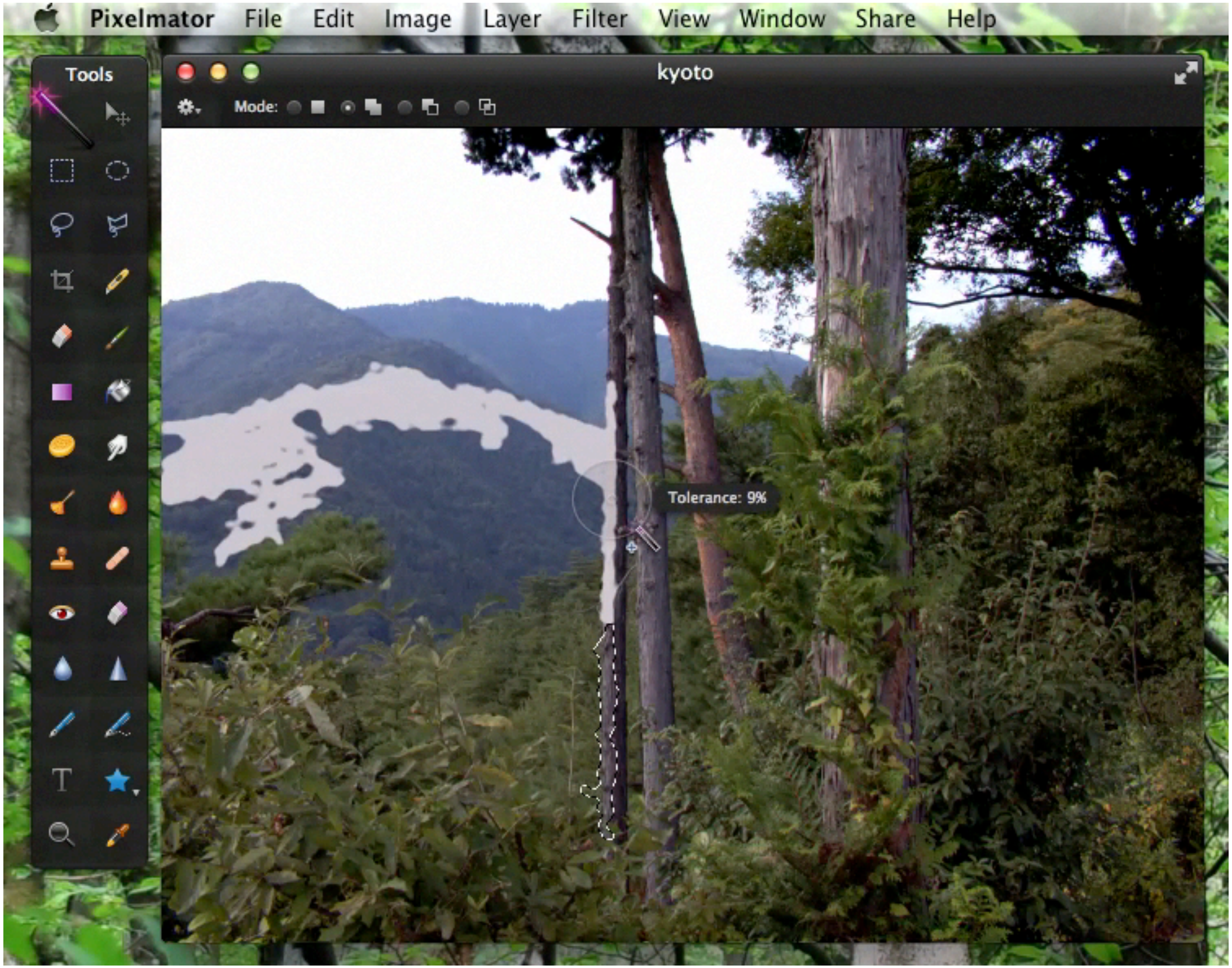




Detecting object boundaries



Does color help?



Where are object contours?

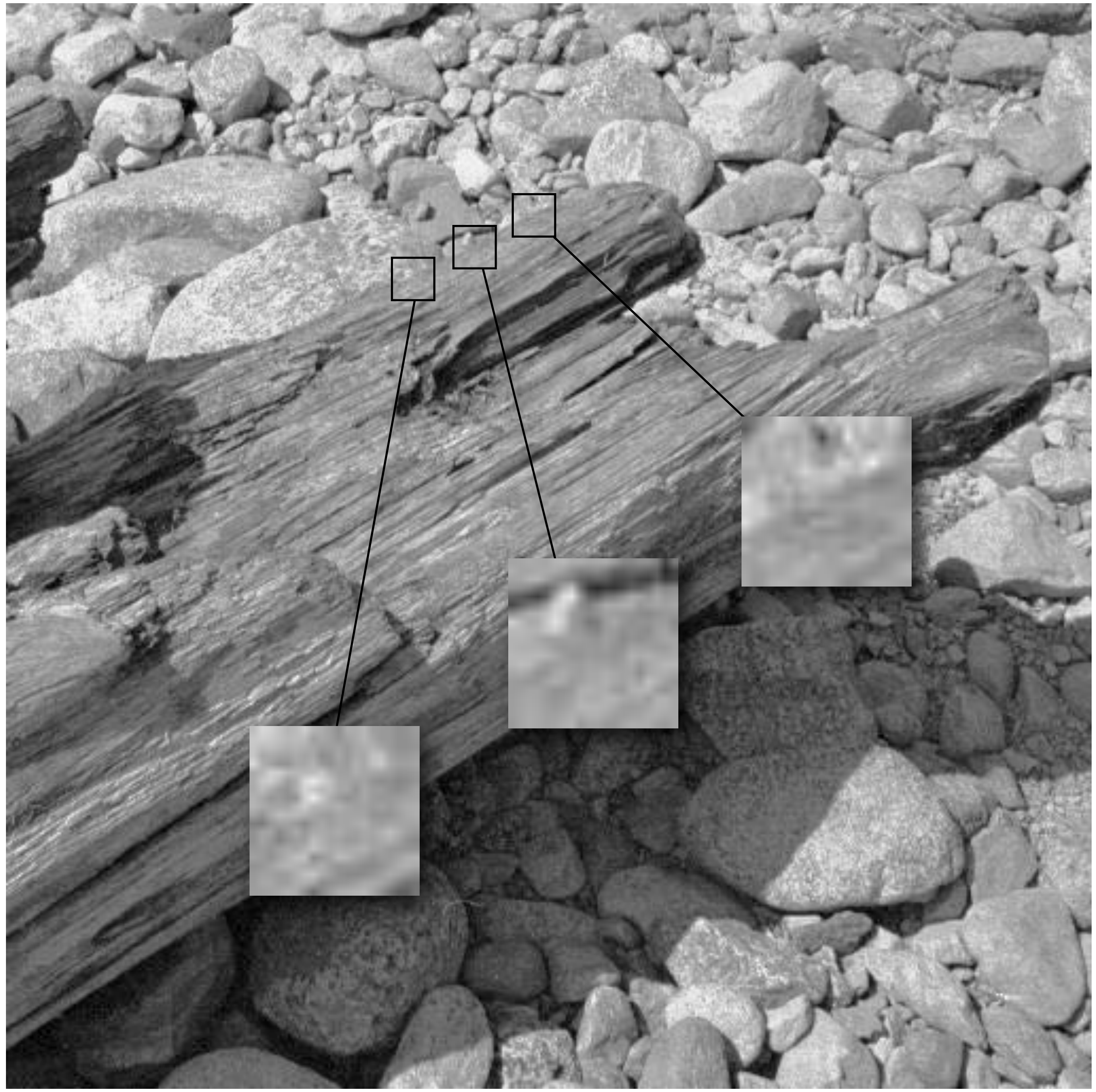
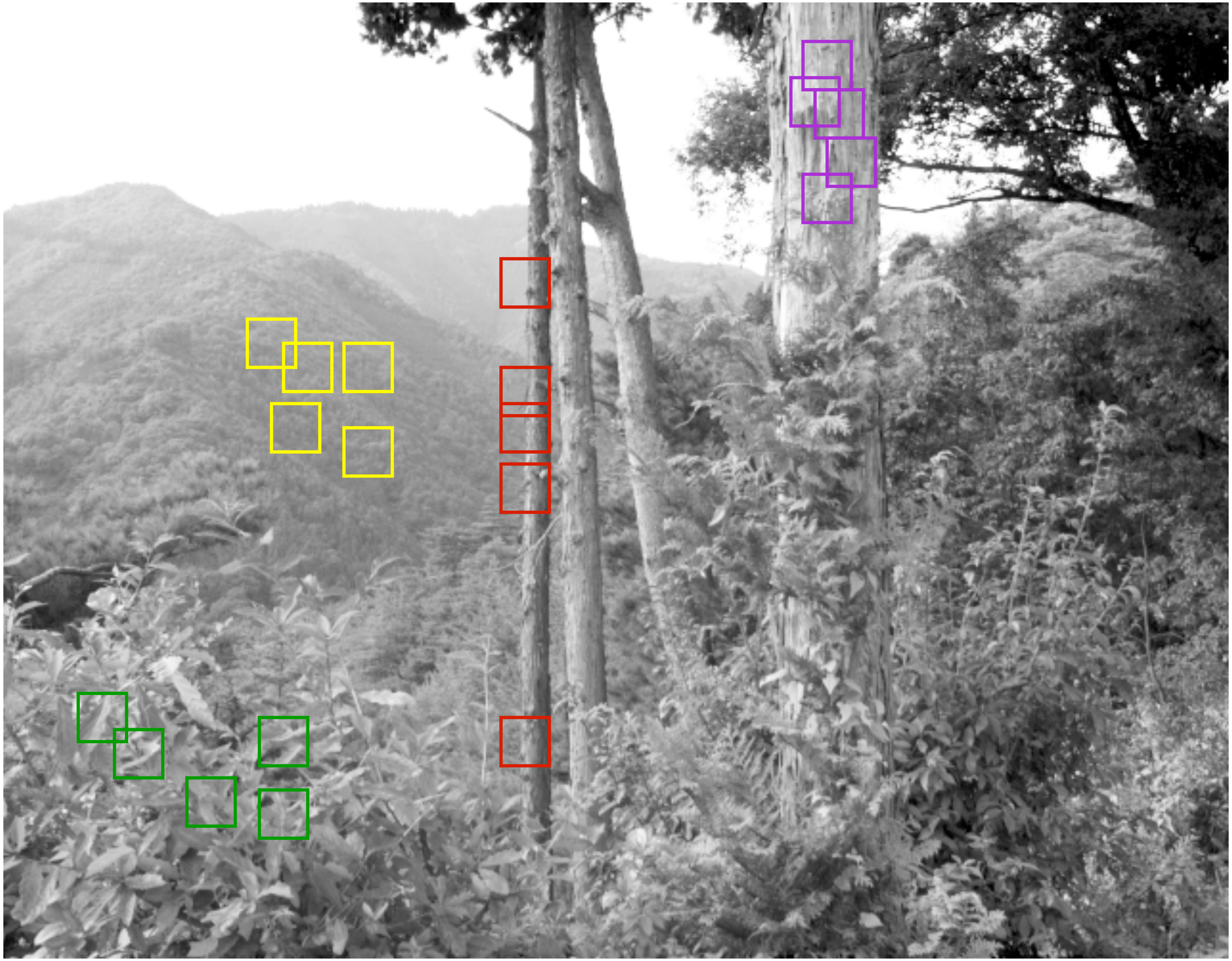


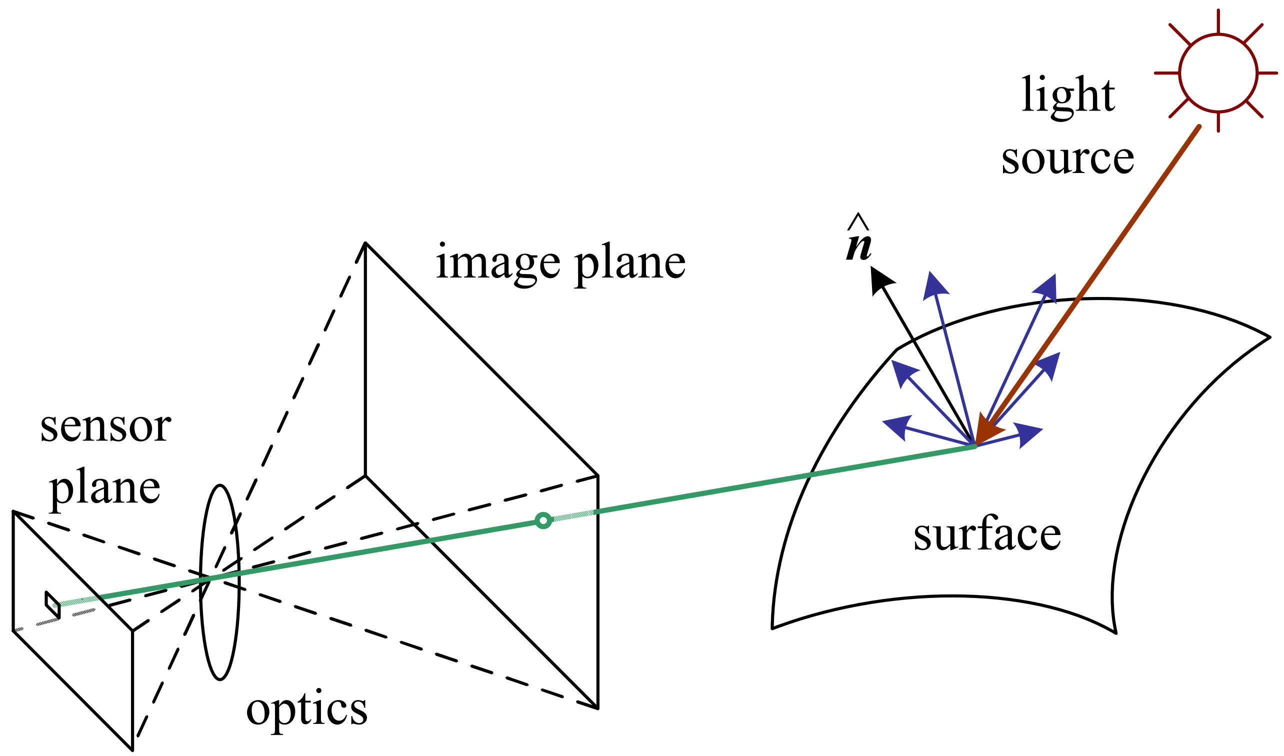
image from Field (1994)

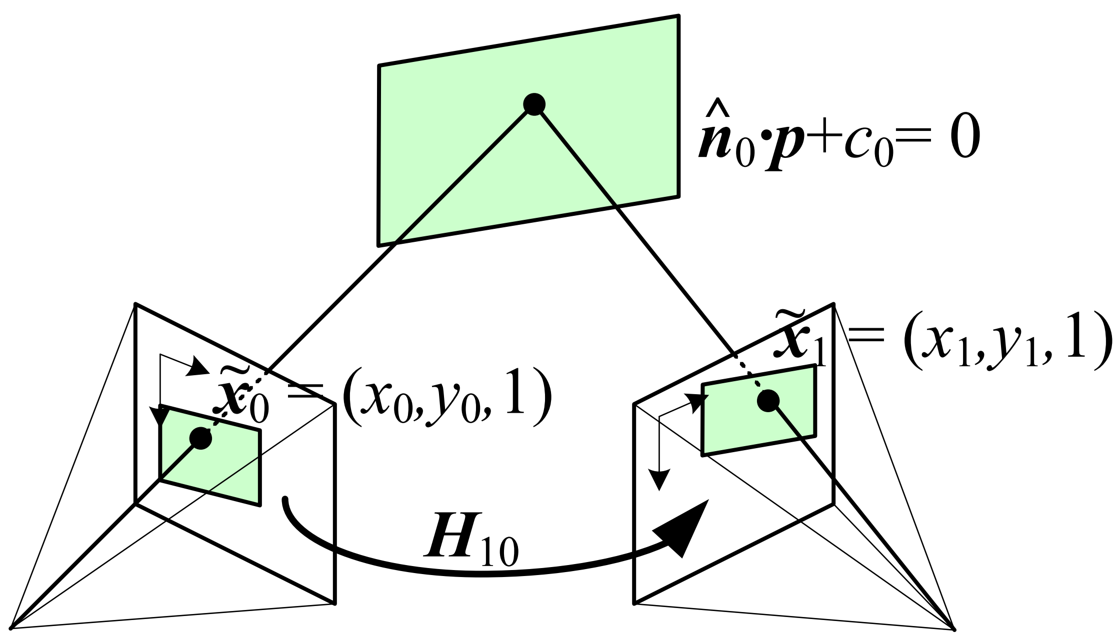
Are these patches part of the same surface or edge?

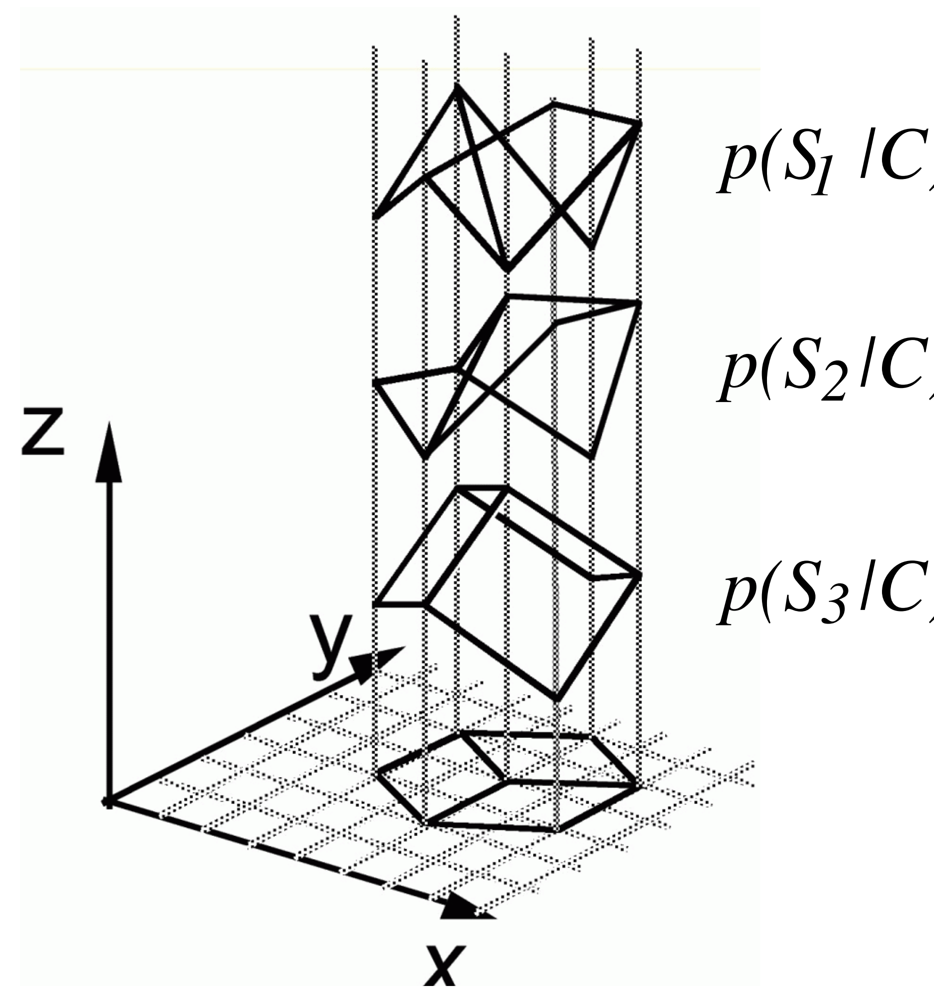


Edges in natural scenes

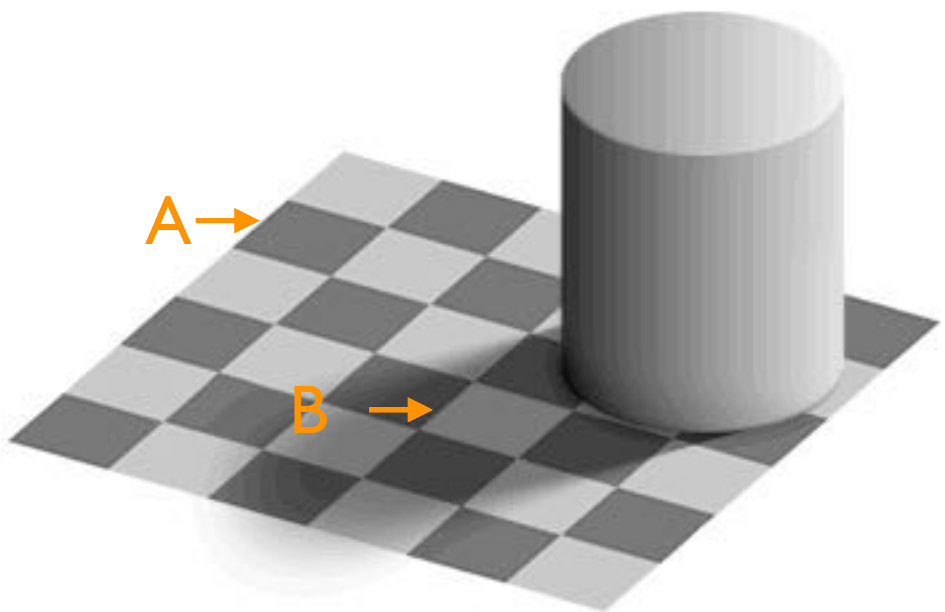
Vision infers scene structure from the projected image





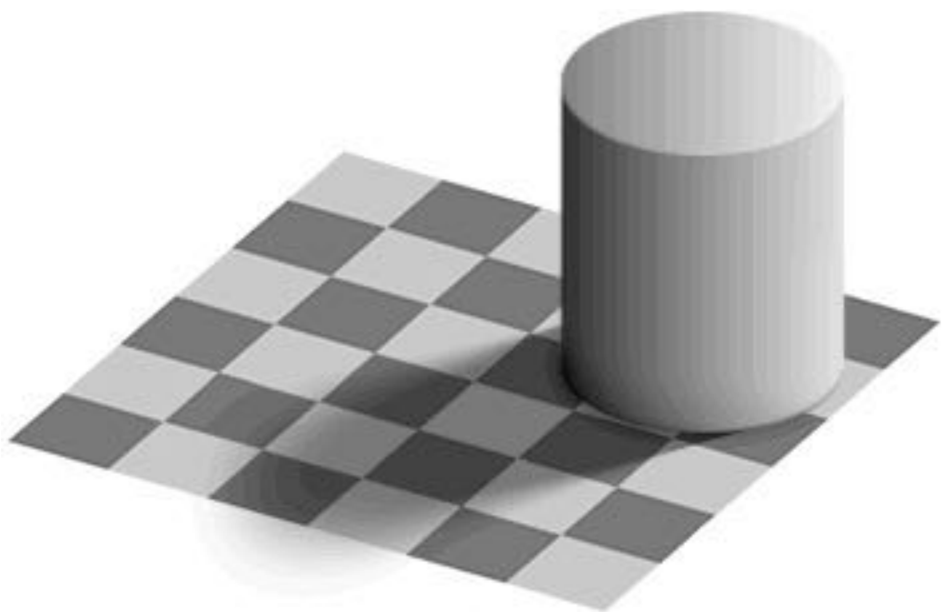


Inference of surface properties: What color are the squares?

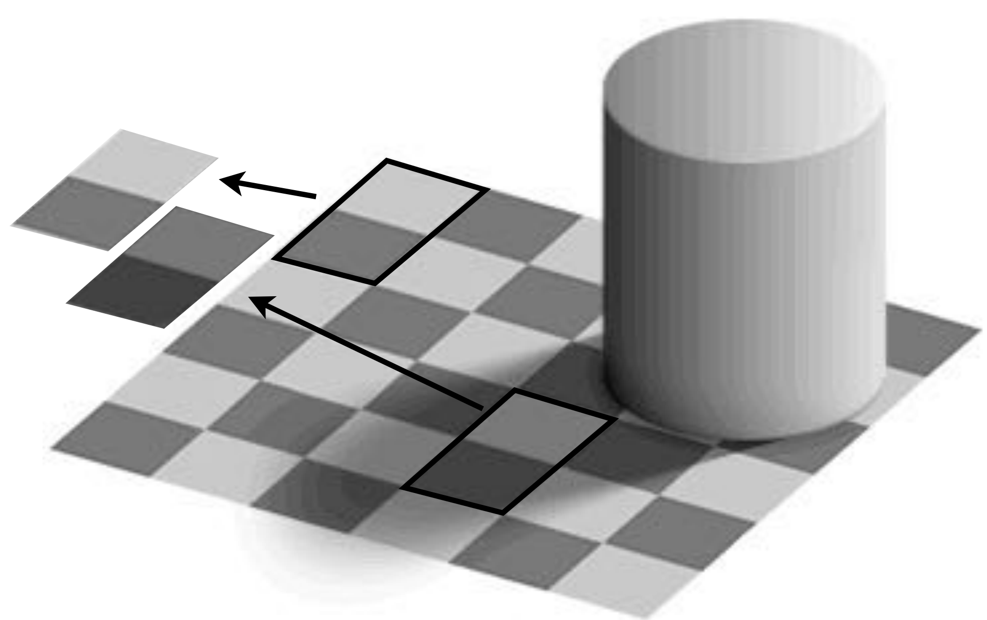


- The light square in the shadow has the same gray value as the dark squares out of the shadow. (Really!)

Shadows influence lightness perception



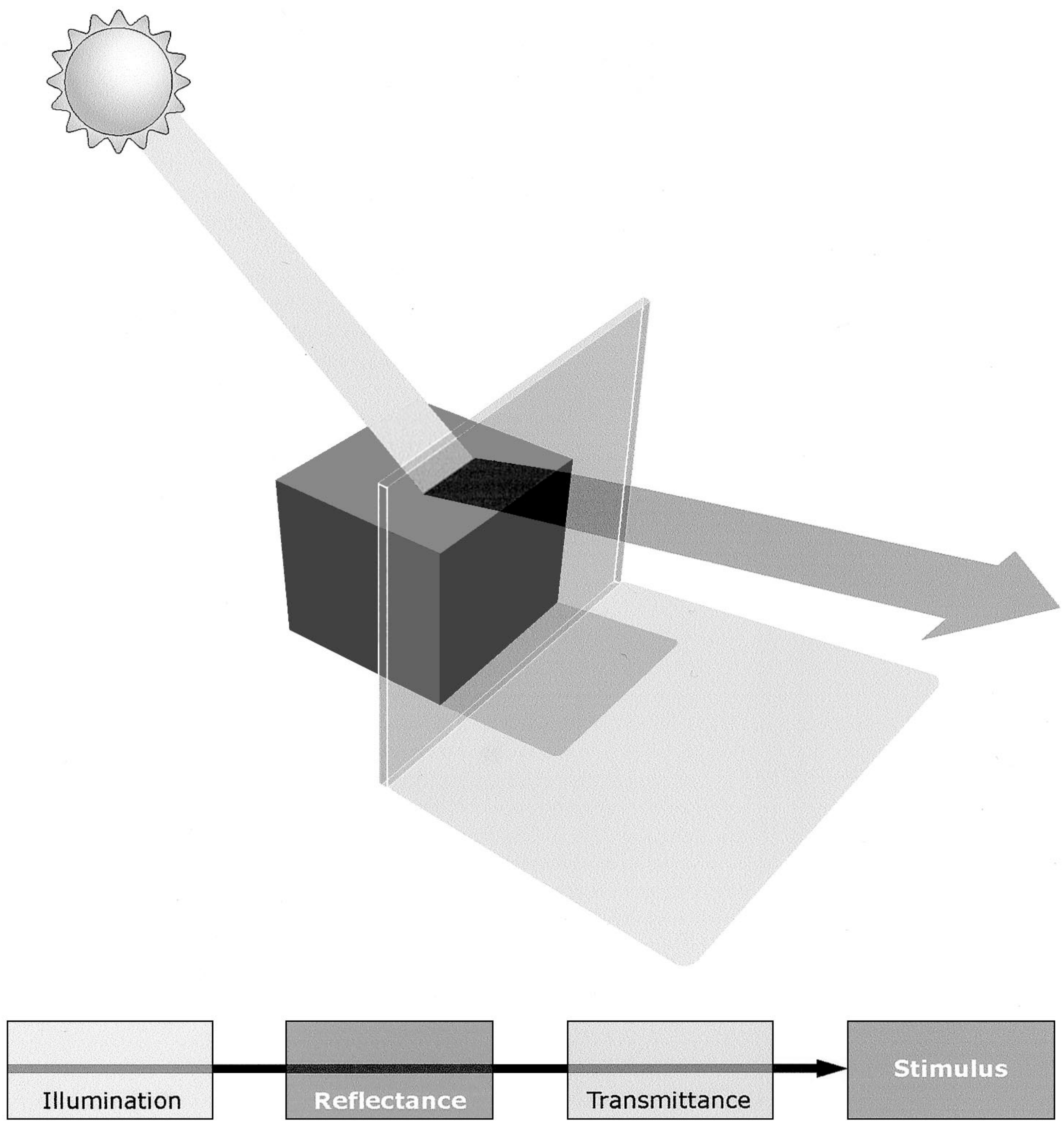
- The light square in the shadow has the same gray value as the dark squares out of the shadow. (Really!)



Vision is inference about the real world

Figure 1.13 from Frisby and Stone. Photograph by Len Hetherington.

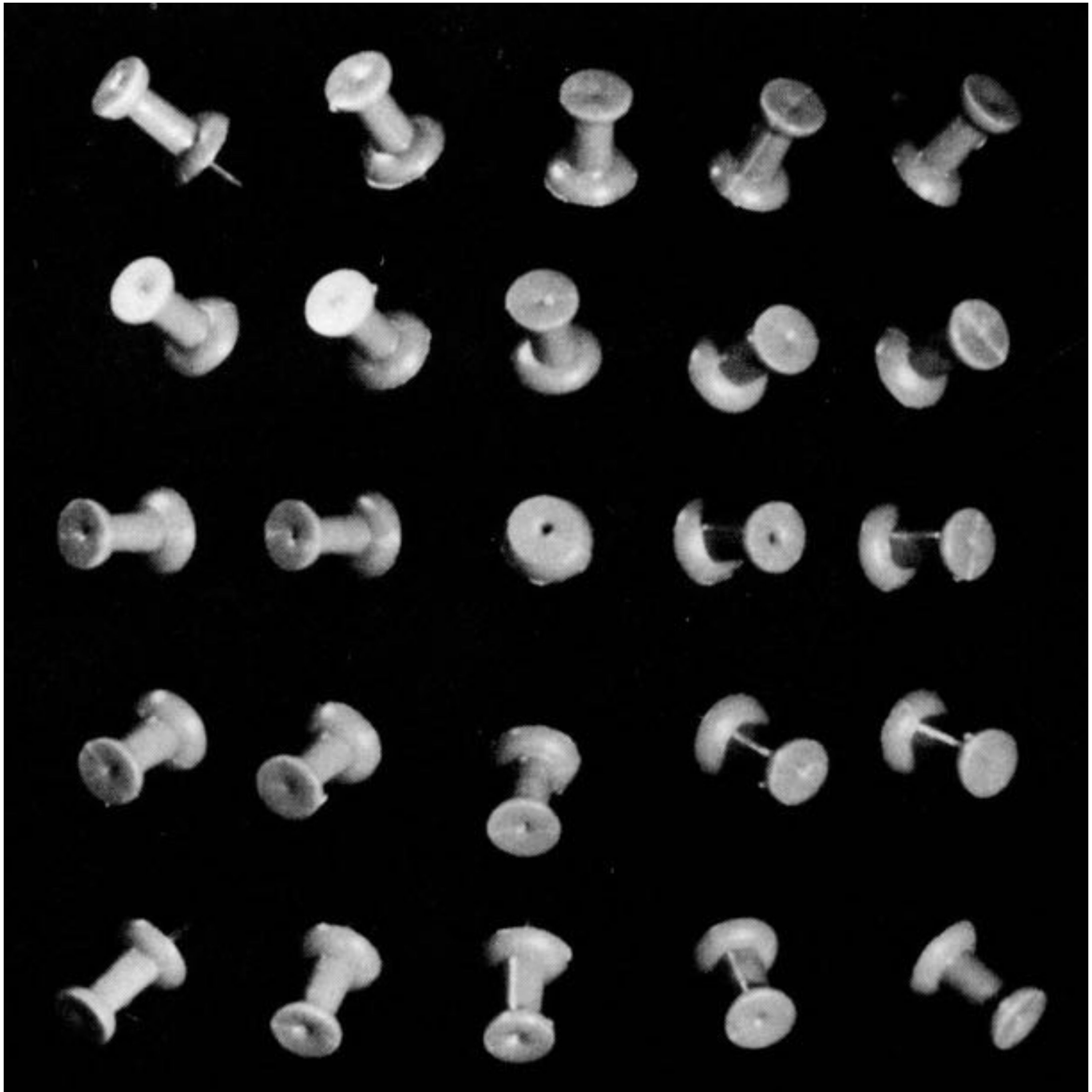
Primary factors in luminance



- Constancy is fundamentally an inference problem
- Light reaching the eye is determined by many factors:
  - illumination of object
  - reflectance of surface
  - surface orientation and curvature
  - surface depth and structure
  - transmittance between object and observer
- None of these can be derived directly from the retinal stimulus

from Purves etal, 2004

Shape from shading



- We easily recognize these as the same shape
- But: retinal images vary greatly
- Shadows and brightness gradients provide most shape information
- How can we recover structure (or shape) from intensity patterns?
- Harder problem: How do we determine similar 3D structures?

How do we compare images?



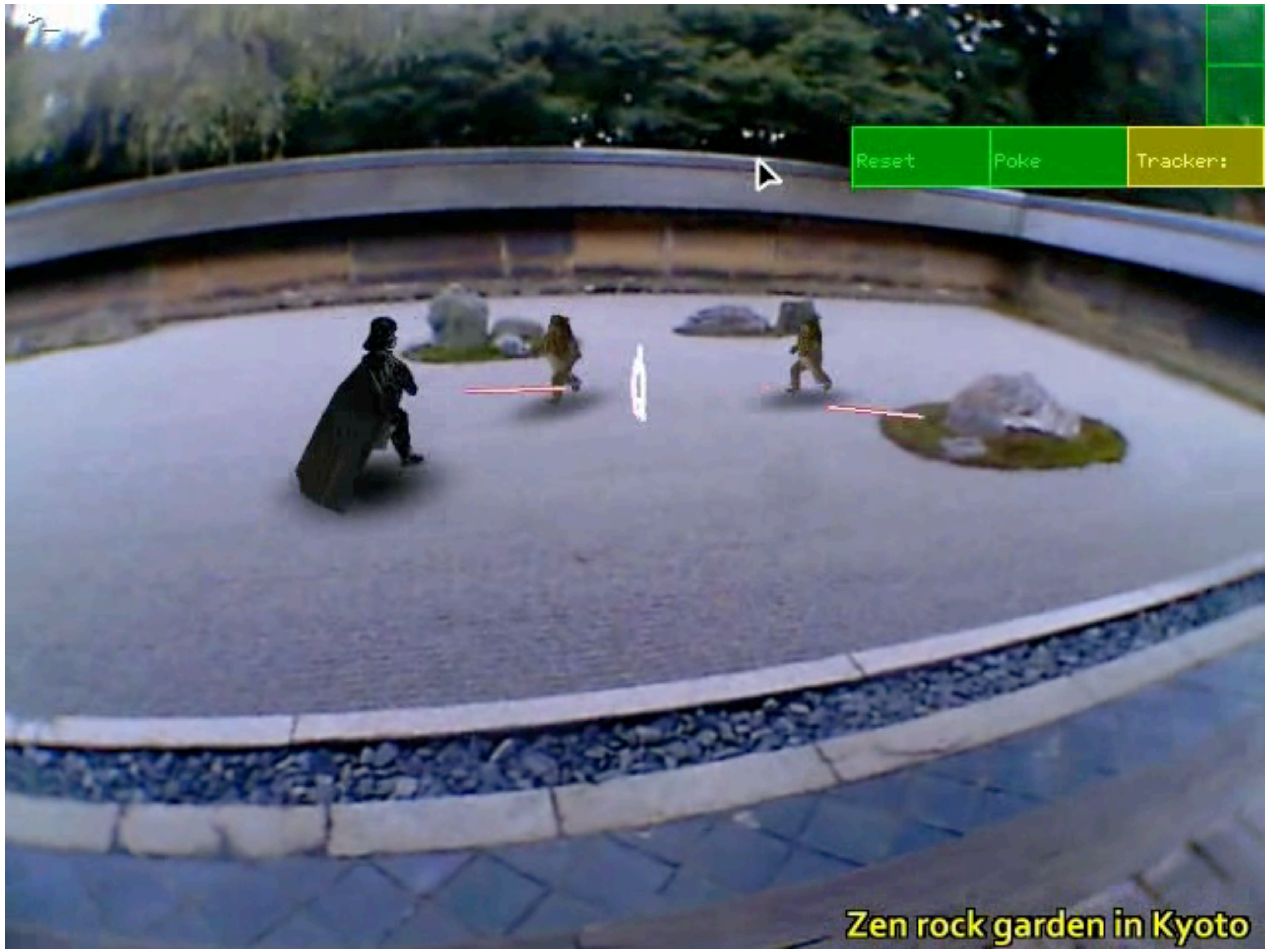

We need to compare in the space of *scene* parameters
(but we will learn there are many shortcuts)

Is this the same person?

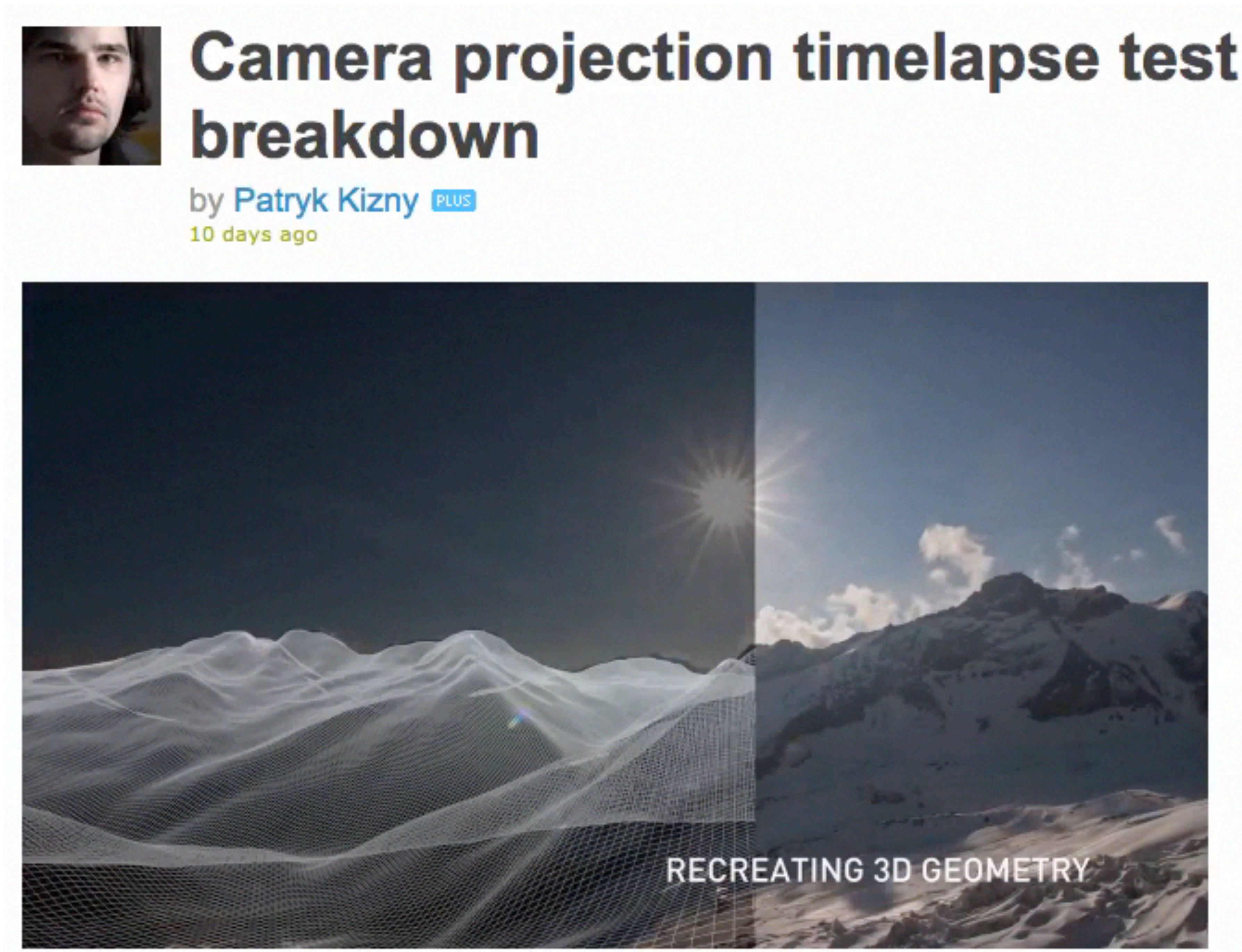
Inferring the 3D structure of the face provides an invariant representation for comparisons.
Note: this is not *necessarily* the way we do face recognition.

Augmented reality is possible with structure inference

Klein and Murray (2007)



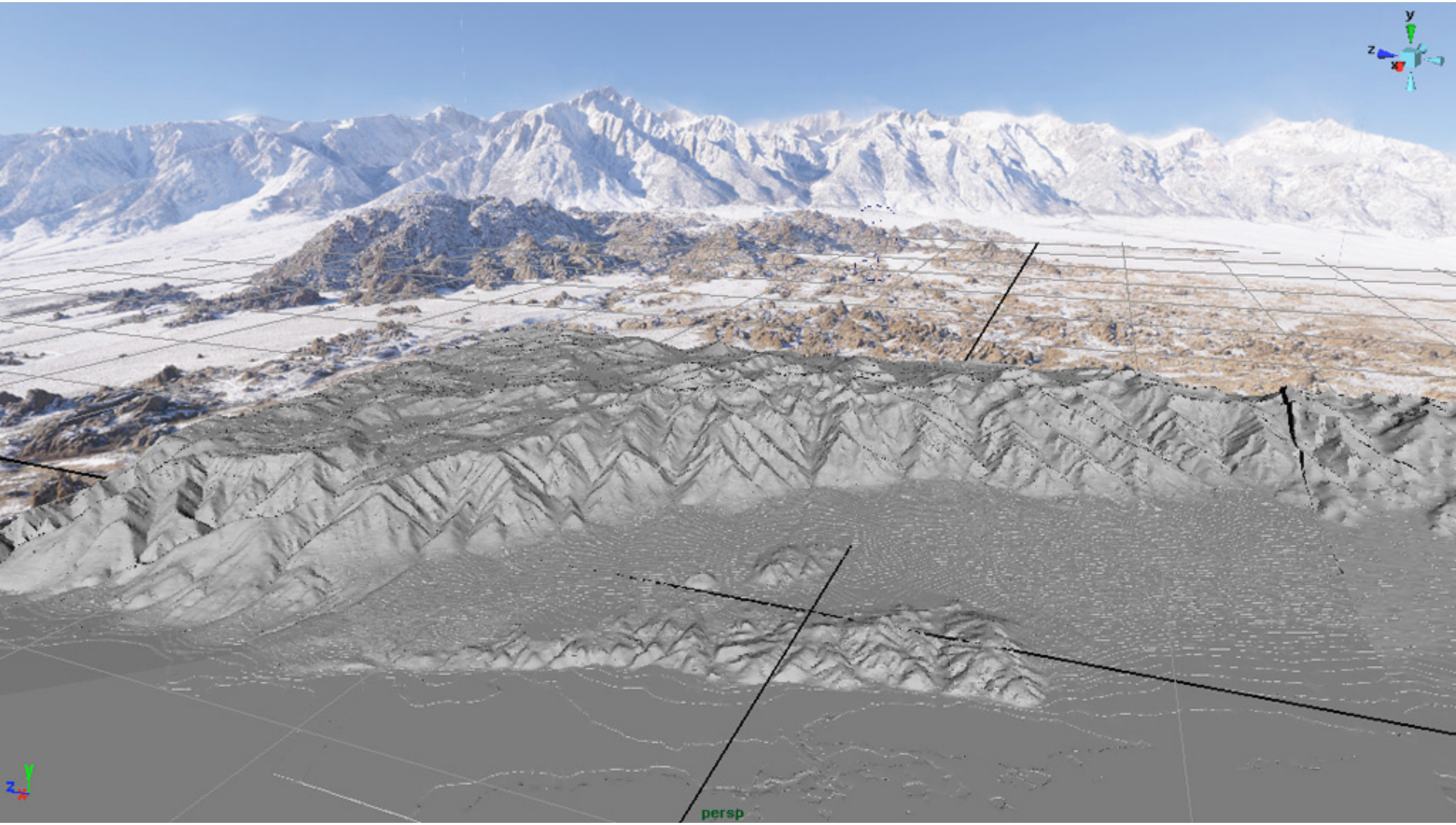
Applications of computer vision?



http://vimeo.com/34698590

http://www.xrez.com/

xres.com



It’s not really computer vision, they’re clever and use digital elevation models.

Motivating questions

- Can perception be understood in terms of computational principles?
- Can we view different components of perceptual processes as solving specific computational problems?

Generic computations

- non-linear threshold
- summation / pooling
- normalization / division / scaling
- linear transformation
- convolution

Computations by themselves tell us nothing about the algorithm or computational goal.

Methodology

- **Theory**: modeling, abstraction, idealization
  How do we formulate the problem?

- **Engineering**: mathematics, computation, algorithms
  How do we solve the problem?

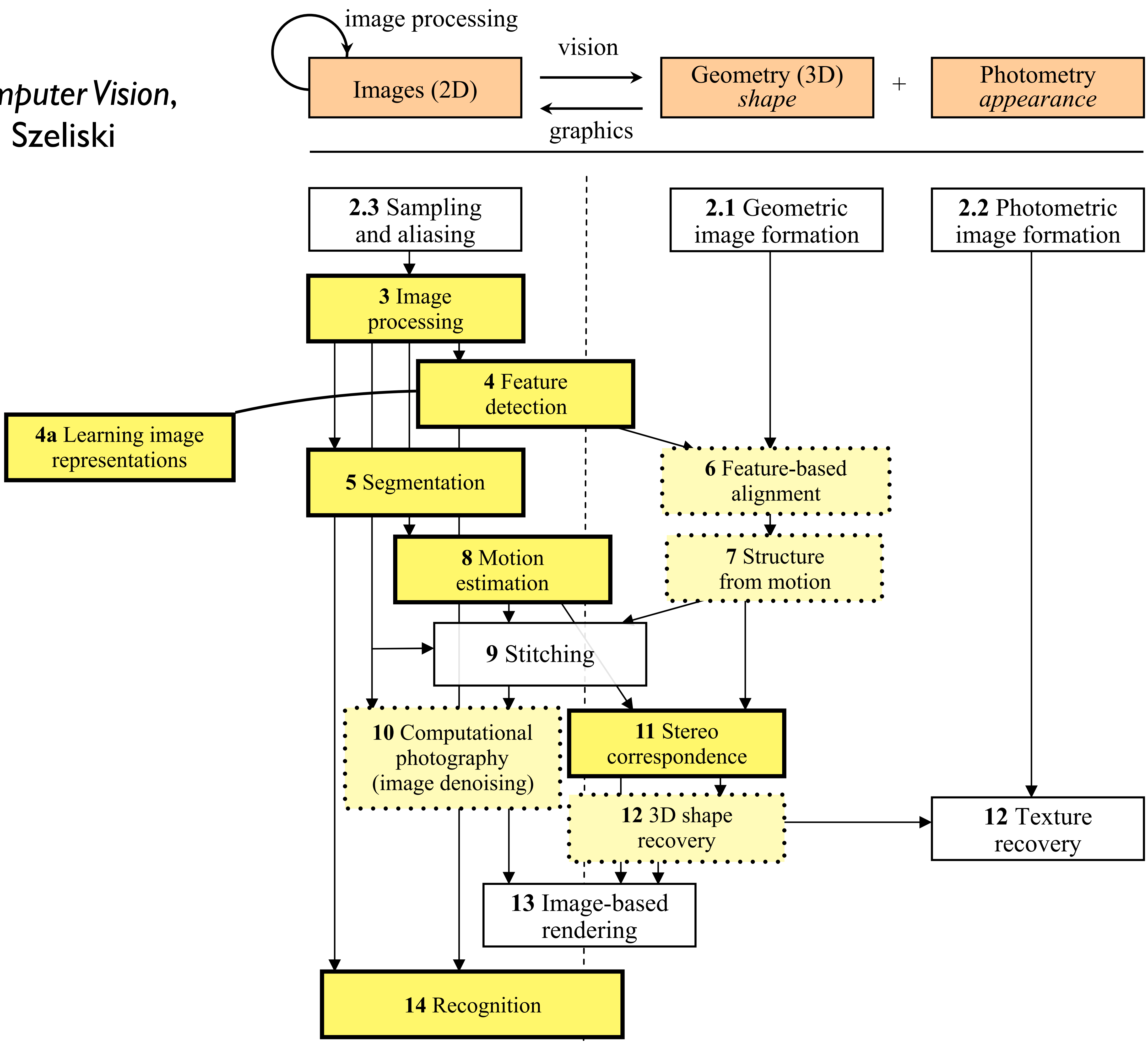
- **Science**: psychology, psychophysics, ethology
  Are we solving the right problem?

We’re still just passing this information along to our visual system, which is where the real work is done.

Vision is *inference* about the external world.

It is *not* simply Feature Detection/Machine Learning/Deep Learning.

Computer Vision, Szeliski



- **Main Textbook**
  Computer Vision: Algorithms and Applications by Richard Szeliski. This is also available electronically at http://szeliski.org/Book/. (S.x.y)
- **Supplemental Texts**
  - *Computational Vision: Information Processing in Perception and Visual Behavior* by Hanspector A. Mallot. (M.x.y)
  - *Seeing: The Computational Approach to Biological Vision*, 2nd edition, by John P. Frisby and James V. Stone. (FS.x,y)
  - *Multiple View Geometry in Computer Vision*, 2nd edition, by Richard Hartley and Andrew Zisserman. (HZ.x.y)
  - *Computer Vision: A Modern Approach*, by David A. Forsyth and Jean Ponce. (FP.x.y)

- Log into redmine (git) server with your case network ID

  https://csevcs.case.edu/

- Check canvas regularly

  https://canvas.case.edu/courses/7804