Computer Vision 01 – Traditional Approaches

PRAKARN UNACHAK

OpenCV

- Open Source Computer Vision Library (https://opencv.org/)
  - Contains many computer vision algorithms
  - For python: https://docs.opencv.org/master/d6/d00/tutorial_py_root.html
  - Require numpy and matplotlib
  - Can install using pip

```
pip install opencv-python
```

- If using Google Colab, the module is already installed, can import right away

```python
import cv2 as cv
```

OpenCV Tutorial of Notes

- *Getting Started with Images* for reading (*imread*) and, showing image (*imshow*) writing (*imwrite*) images
  - https://docs.opencv.org/master/db/deb/tutorial_display_image.html
- Some image processing techniques in *Image Processing in OpenCV*
  - https://docs.opencv.org/master/d2/d96/tutorial_py_table_of_contents_imgproc.html
- And many in Feature Detection and Description
  - https://docs.opencv.org/master/db/d27/tutorial_py_table_of_contents_feature2d.html

Outline

- Computer Vision
- Image Formation
- Image Processing
- Some Tasks in Object Recognition

Perception and Computer Vision

- Perception: getting information about the world by interpreting response from sensors
  - Such as switch, camera, microphone, GPS
  - Can also using active sensing, that send out signals. Example: radar, ultrasound, LIDAR
- For this course, we will focus on computer vision (CV): how to extract useful information and understanding from visual data, such as images and videos.
- In this slides, most algorithms work on grayscale image

Examples of Tasks in Computer Vision

- Object Detection/ Localization
  - Locate objects of interest in an image
  - Face, car, lane in the road, people, etc.
- Object Recognition
  - Classify objects using visual information
  - Face-to-person, facial expression, optical character recognition (OCR), id'ing objects on the road
- Others, such as image labelling, model construction from images

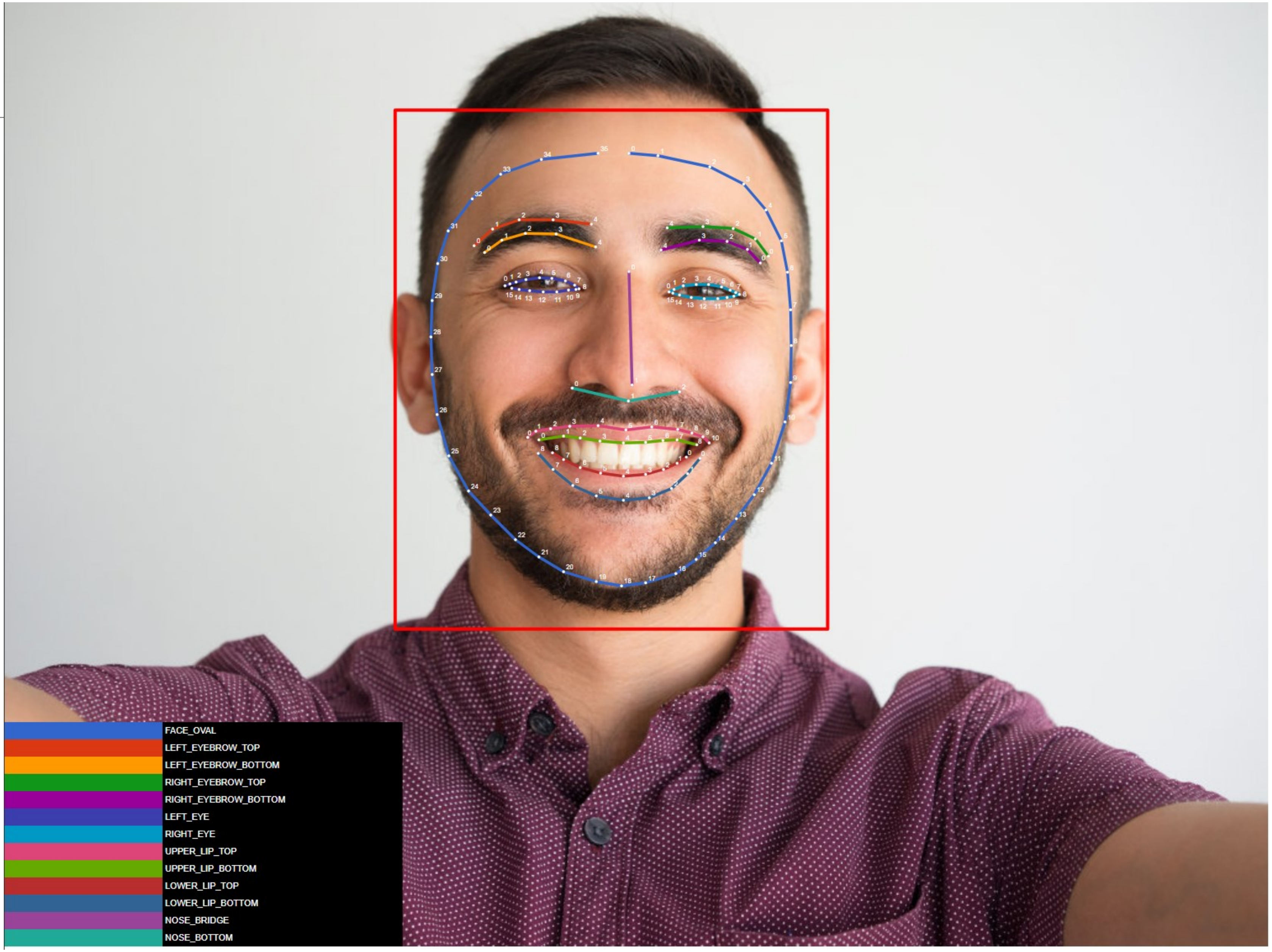
Optical Character Recognition (OCR)

- Identify characters in images

4YCH428

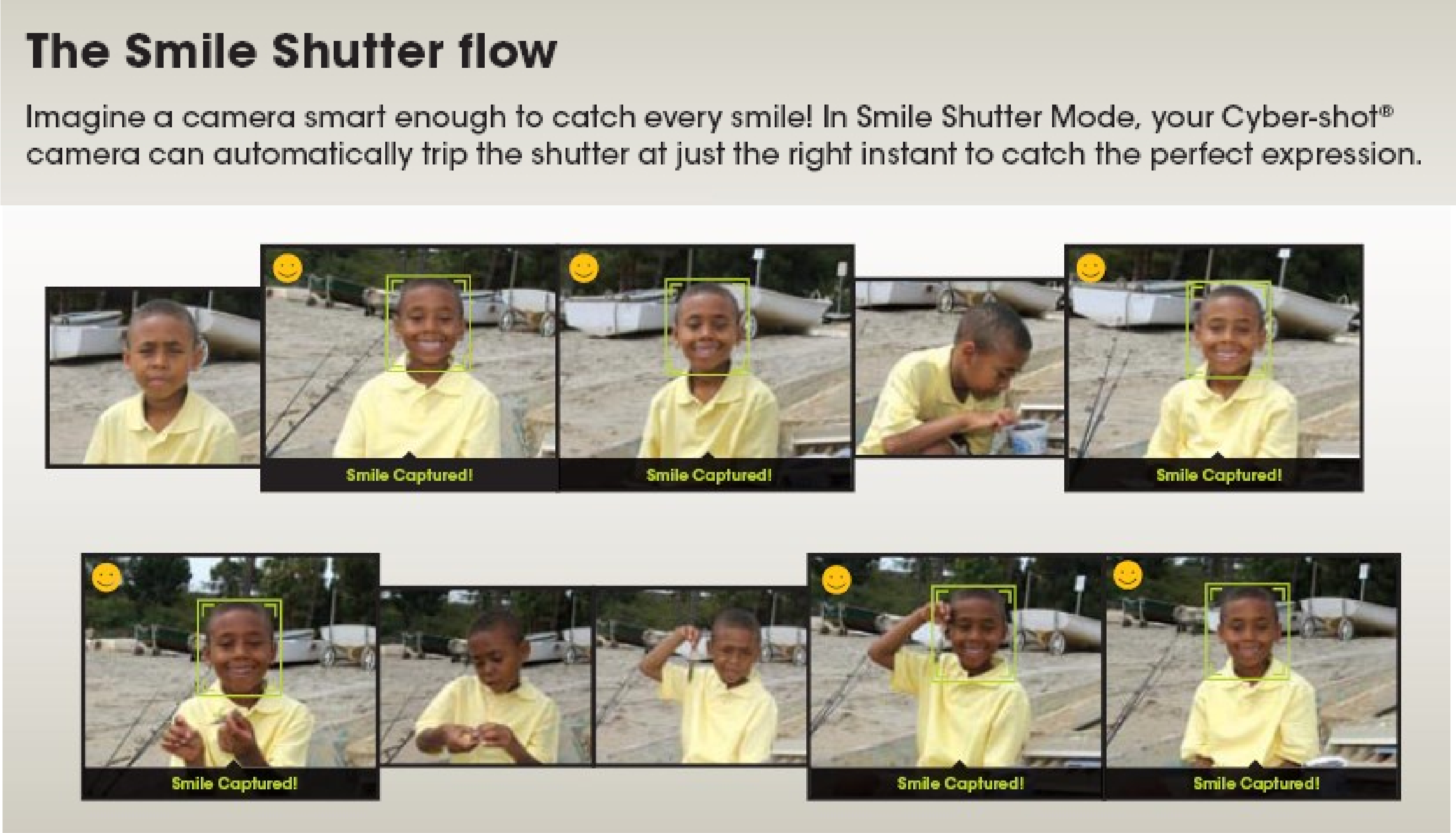
Source: Wikipedia

Face (and Facial Feature) Detection



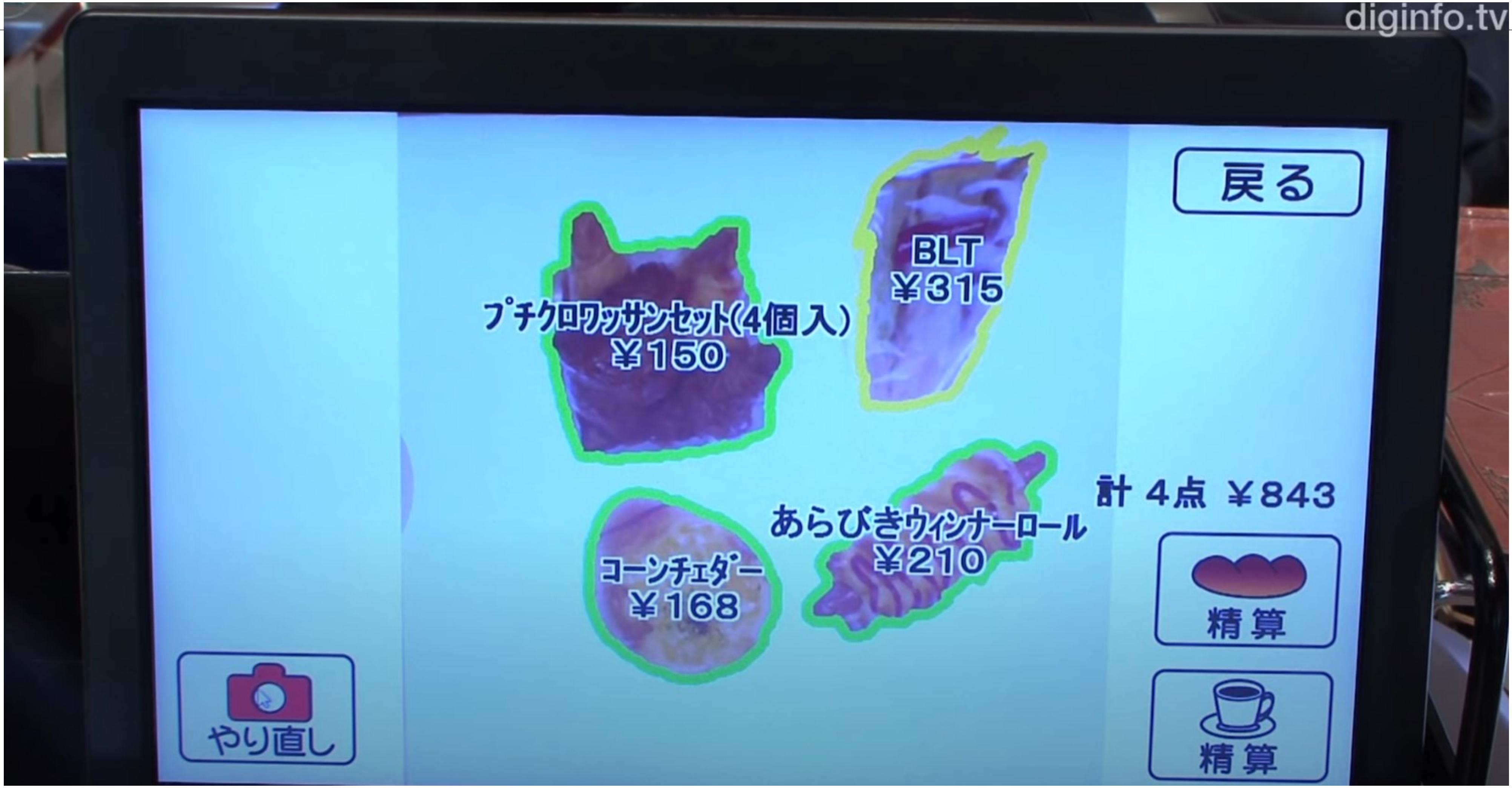
Source: Google ML Kits

Facial Expression Detection



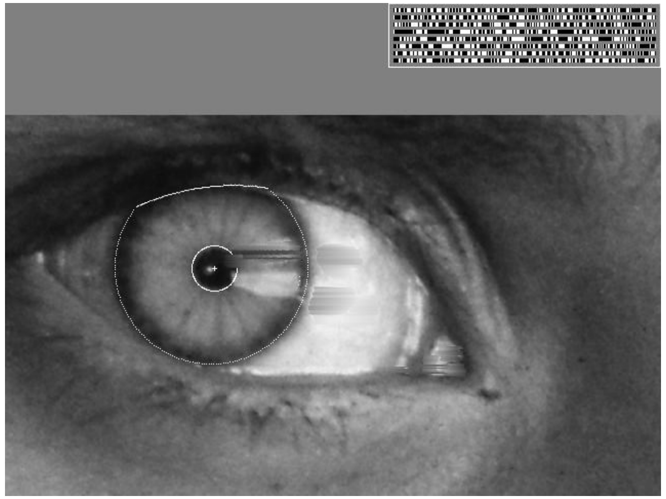
[Sony Cyber-shot® T70 Digital Still Camera]

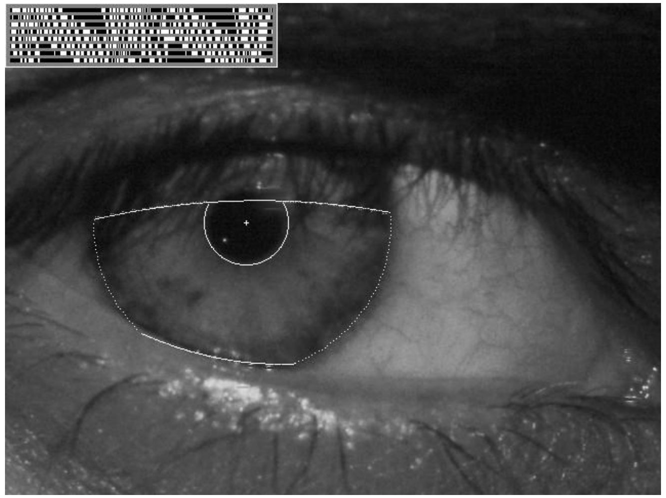
Object Recognition



https://www.youtube.com/watch?v=HdZ3y3tVtTU

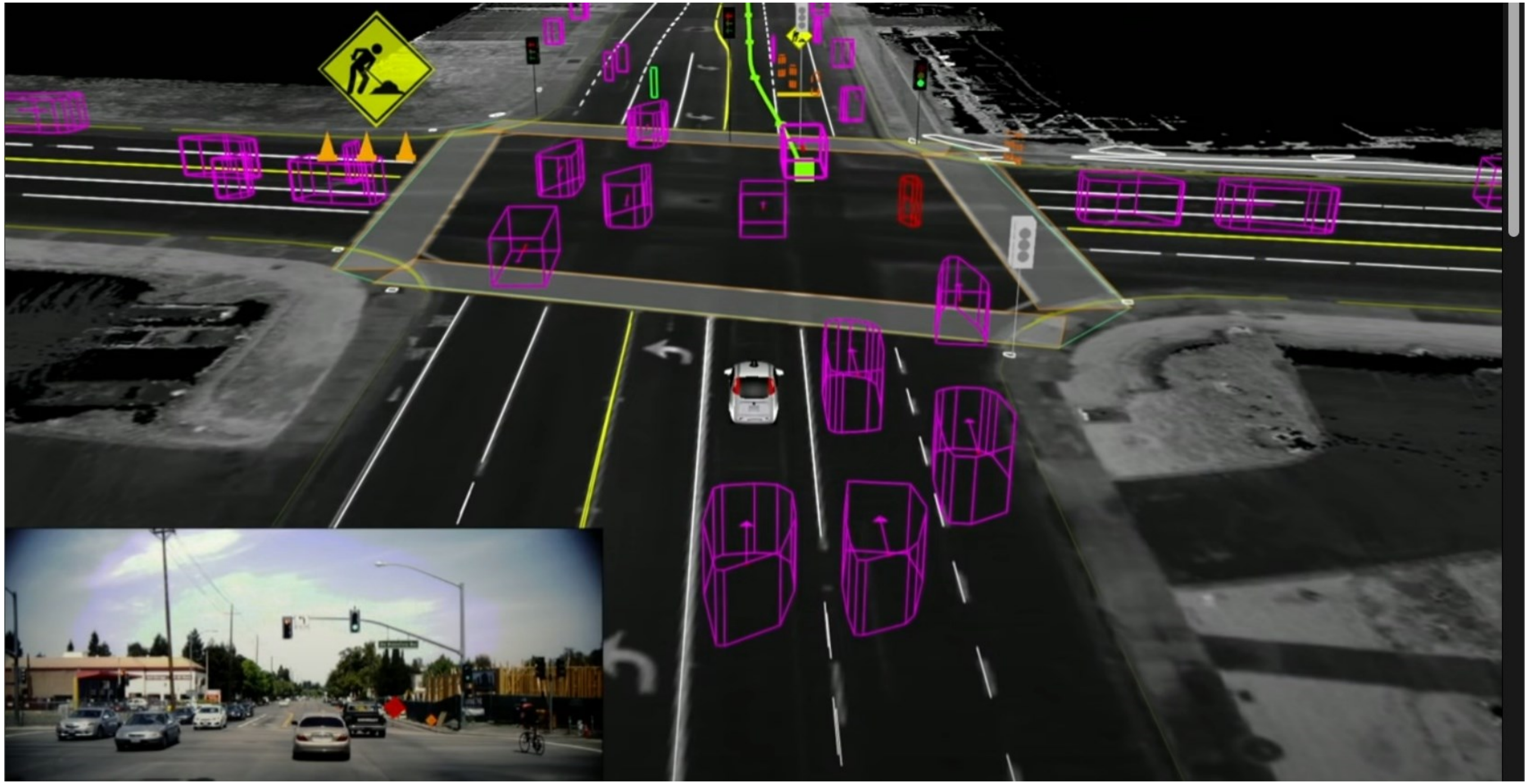
Vision-based Biometrics





"*How the Afghan Girl was Identified by Her Iris Patterns*" Read the story
wikipedia

Autonomous Car



Source: New York Times, Youtube

Autonomous Car (cont.)

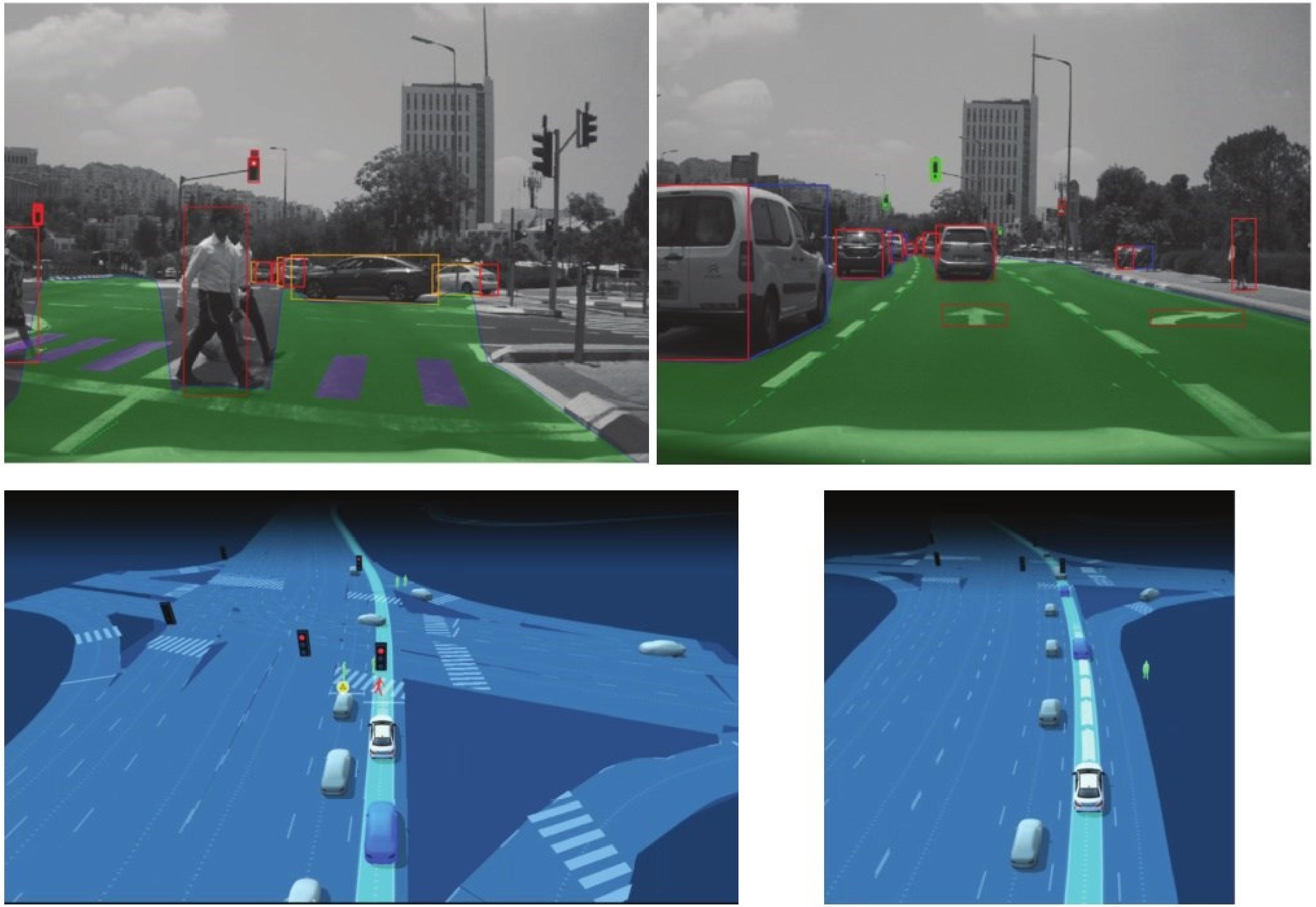
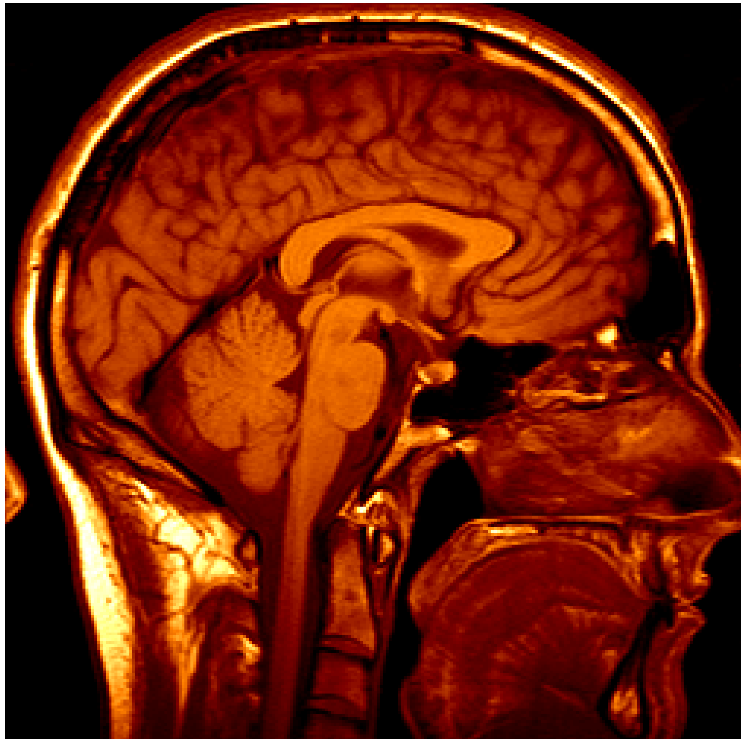


Figure 25.28 Mobileye's camera-based sensing for autonomous vehicles. **Top row**: Two images from a front-facing camera, taken a few seconds apart. The green area is the free space—the area to which the vehicle could physically move in the immediate future. Objects are displayed with 3D bounding boxes defining their sides (red for the rear, blue for the right side, yellow for the left side, and green for the front). Objects include vehicles, pedestrians, the inner edge of the self-lane marks (necessary for lateral control), other painted road and crosswalk marks, traffic signs, and traffic lights. Not shown are animals, poles and cones, sidewalks, railings, and general objects (e.g., a couch that fell from the back of a truck). Each object is then marked with a 3D position and velocity. **Bottom row**: A full physical model of the environment, rendered from the detected objects. (Images show Mobileye's vision-only system results). Images courtesy of Mobileye.

Constructing 3D World from 2D Image

Medical Imaging



3D imaging
MRI, CT

Image guided surgery
Grimson et al., MIT

CV in Medicine

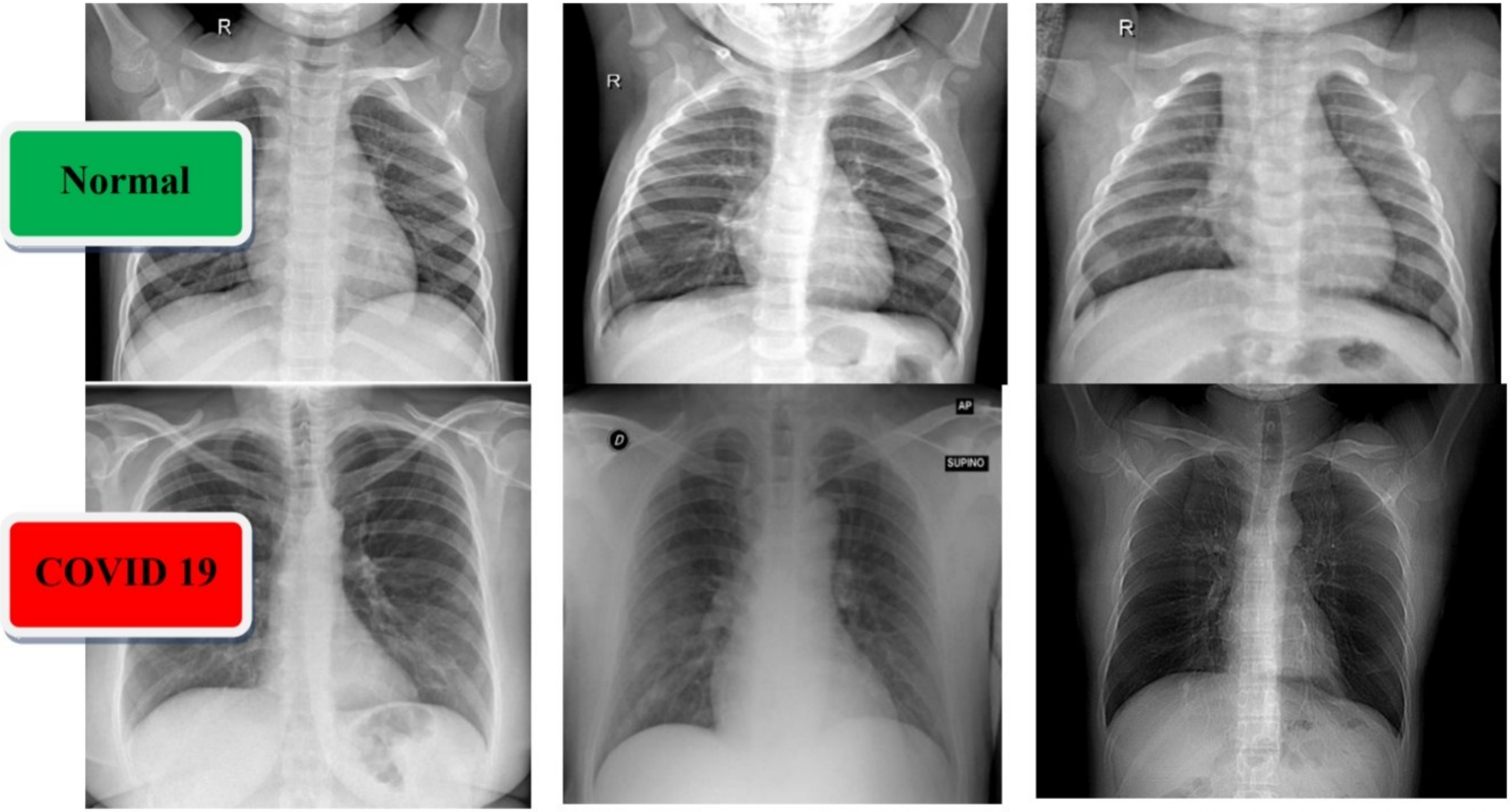
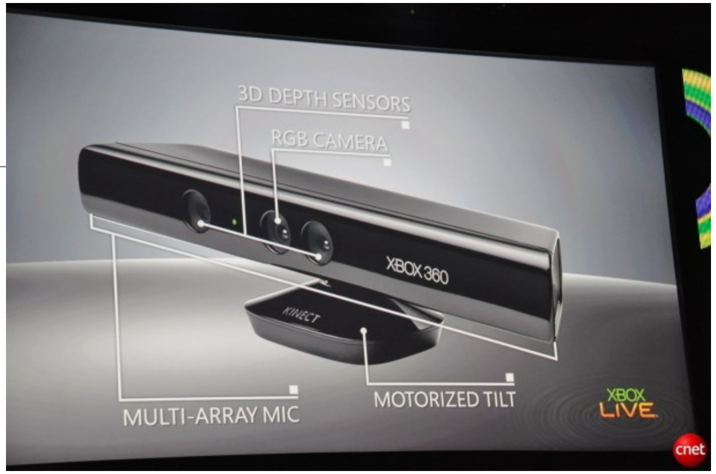


Fig 1. A sample of X-ray images dataset for normal cases (first row) and COVID-19 patients (second row).

Mahdy et. al, Automatic X-ray COVID-19 Lung Image Classification System based on Multi-Level Thresholding and Support Vector Machine “”

CV in Gaming



Kinect

- Pose Estimation
  - https://www.youtube.com/watch?v=33AsuE-WP64
- Object Recognition
  - https://www.youtube.com/watch?v=fQ59dXOo63o

Why is Computer Vision Difficult?

1. 2D $\longrightarrow$ 3D: The actual world is 3D. Need to inverse optics (how 3d objects are placed on 2D image) and graphics (how images are rendered from 3D objects, involving occluded (blocked) surfaces, shading, gradients, perspective, etc.)
   - No unique solution, especially on inverse optics. There may be most likely solution, though.

Easy for Human, not so for CV

- Some tasks are easy for us, even if we can't explain how we do it

Source: Prof. Daugman's Slides

Which two pictures show the same person?

Why is Computer Vision Difficult? (cont.)

2. Few visual tasks can be performed in a purely data-driven way (“bottom-up” image analysis)
   - May need model of the worlds to interpret images correctly

- A noisy and monochromatic image

Source: Prof. Daugman’s Slides

Why is Computer Vision Difficult? (cont.)

From that low-quality image, you need to:

- perform the figure-ground segmentation of the scene (into its objects, versus background clutter)
- infer the 3D arrangements of objects from their mutual occlusions
- infer surface properties (texture, color) from the 2D image statistics
- infer volumetric object properties from their 2D image projection
- And in real time?

Difficulty for Autonomous Car in Australia

- Kangaroo can jump up to 3 meters
- How will this affect autonomous car trying to determine how far away it is?

Source: Getty Images

Image Formation

Image Formation

- 3D $\longrightarrow$ 2D: From object model to rendering model
  - Object model describes objects that inhabit the visual world
  - Rendering model describes the physical, geometric, and statistical processes that produce the stimulus (image) from the world
- Rendering model can be ambiguous
  - Ambiguity can be managed with prior knowledge

Interpreting Image

Unknown Source

Pinhole Camera

- Light from the scene (Q) go through pinhole and create image (P') at sensor (screen)
- Perspective projection causes farther object to appear smaller
- Line disappears at vanishing point
- Larger pinhole results in image out of focus, smaller pinhole results in darker image
  - More exposure time can allow more light in, but result in motion blur

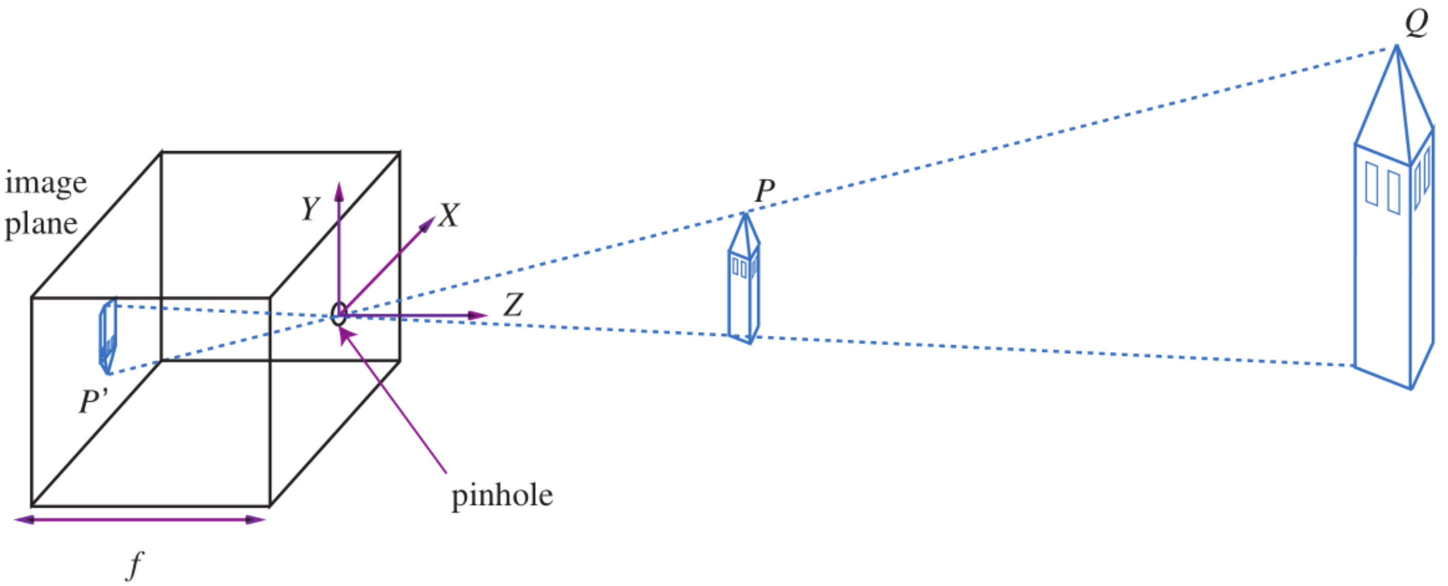


Figure 25.2 Each light sensitive element at the back of a pinhole camera receives light that passes through the pinhole from a small range of directions. If the pinhole is small enough, the result is a focused image behind the pinhole. The process of projection means that large, distant objects look the same as smaller, nearby objects—the point P' in the image plane could have come from a nearby toy tower at point P or from a distant real tower at point Q.

Perspective Projection:

$$-\frac{\mathrm{x}}{\mathrm{f}}=\frac{\mathrm{X}}{\mathrm{Z}}, -\frac{\mathrm{y}}{\mathrm{f}}=\frac{\mathrm{Y}}{\mathrm{Z}} \Rightarrow x=-\frac{fX}{Z}, y=-\frac{fY}{Z}$$

Source: AIMA

Lens Systems

- Lens system allows more light while keeping image in focus
- Have limited depth of field: can only focus on certain depths around focal plane

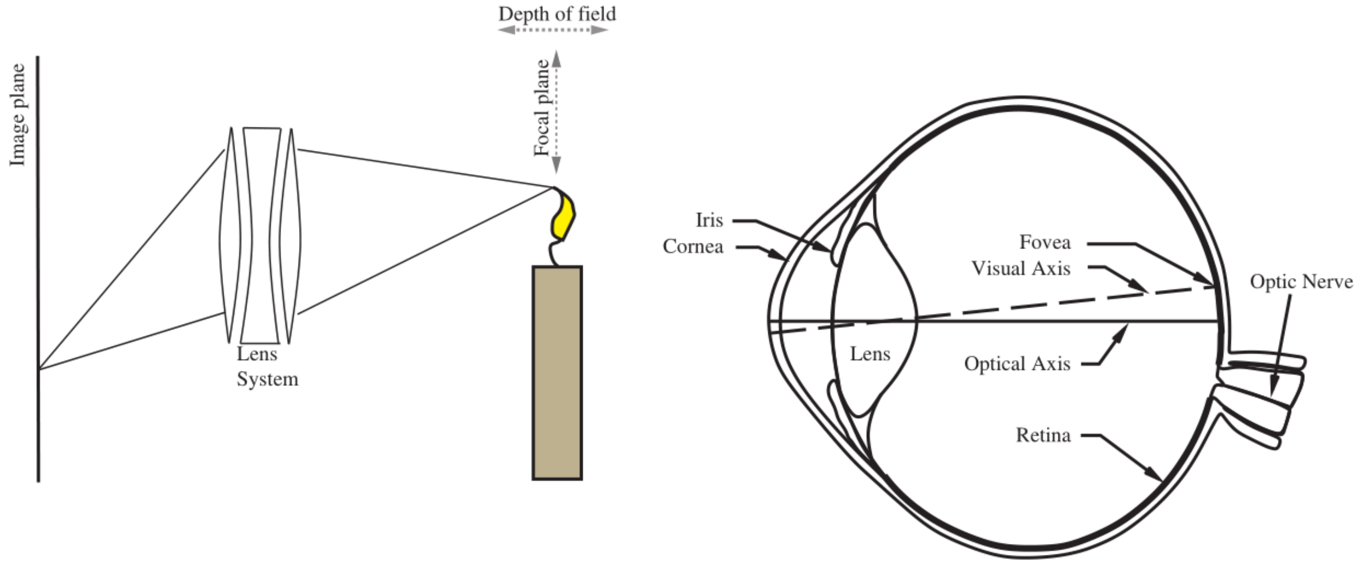


Figure 25.3 Lenses collect the light leaving a point in the scene (here, the tip of the candle flame) in a range of directions, and steer all the light to arrive at a single point on the image plane. Points in the scene near the focal plane—within the depth of field—will be focused properly. In cameras, elements of the lens system move to change the focal plane, whereas in the eye, the shape of the lens is changed by specialized muscles.

Source: AIMA

Scaled Orthographic Projection

- Perspective effects aren't pronounced if objects are relatively closed to each other compared to the distance from sensor to those objects
- If the depth Z of points on the scaled orthographic projection object varies within $Z_0 \pm \Delta Z$, with $\Delta Z << Z_0$, then the perspective scaling factor f/Z can be approximated by a constant $s = f/Z_0$
- Good for a scene with not much depth variation ($\pm \Delta Z$)

Light and Shading

- The brightness of a pixel is a function of the brightness of the surface patch in the scene that projects (reflect) to the pixel
- Diffuse reflection spread evenly
- Specular reflection is directional mirror effect
- Shadow is the area not reached by (some) light sources

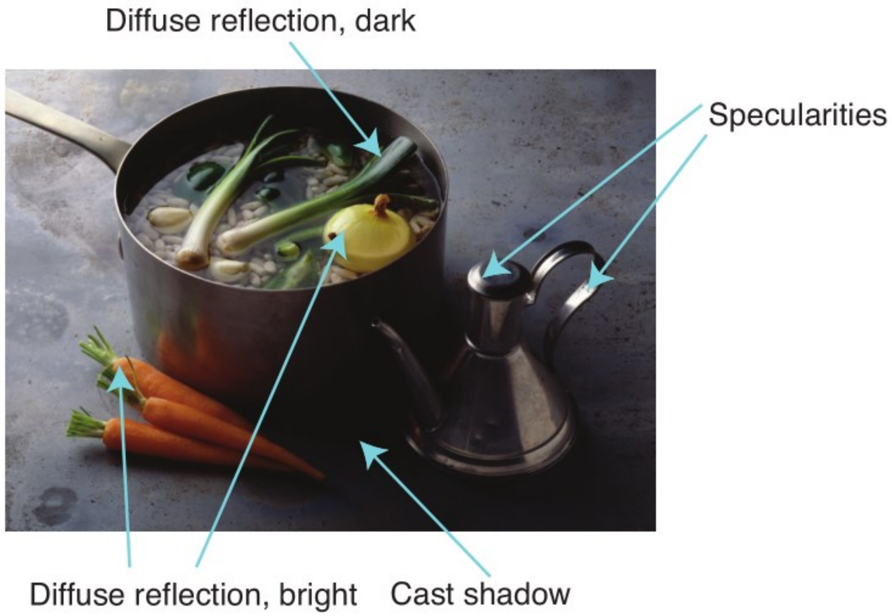
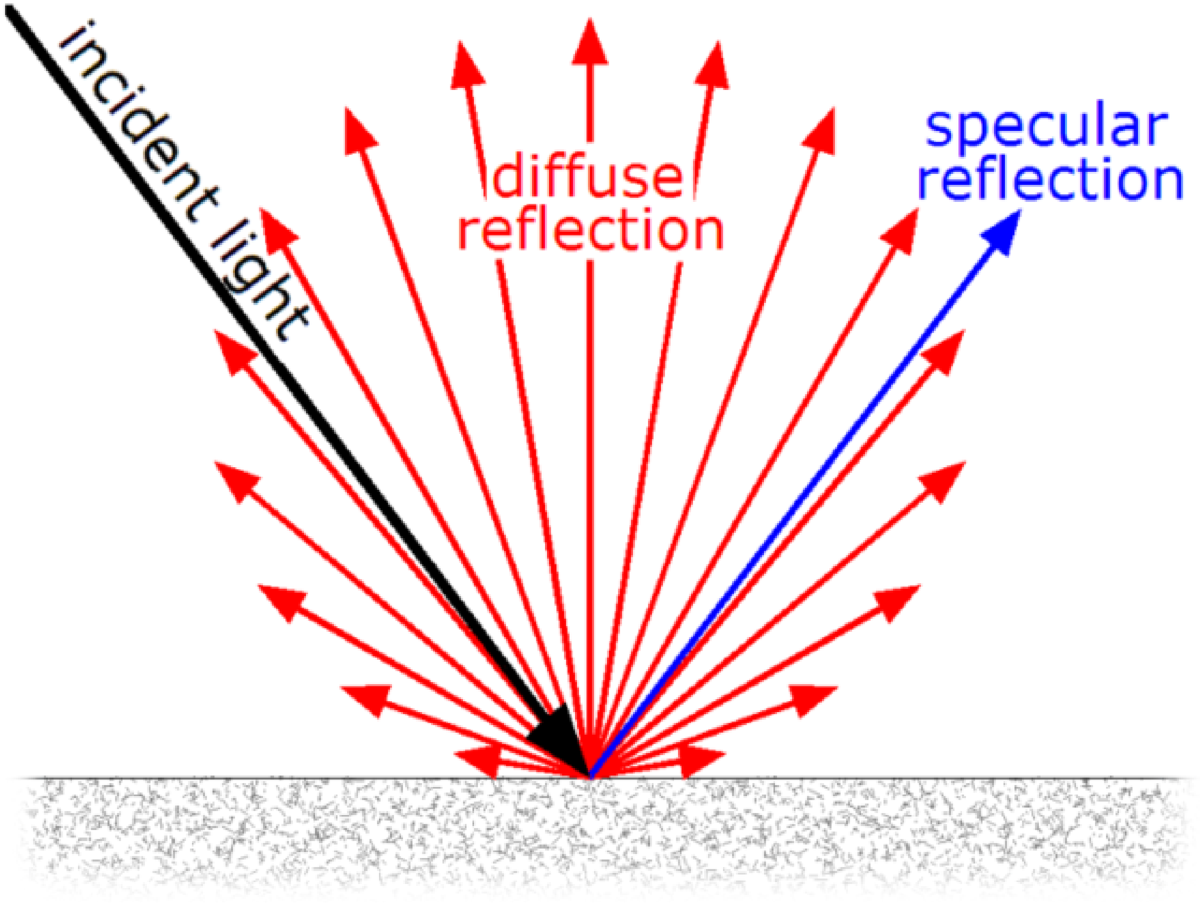


Figure 25.4 This photograph illustrates a variety of illumination effects. There are specularities on the stainless steel cruet. The onions and carrots are bright diffuse surfaces because they face the light direction. The shadows appear at surface points that cannot see the light source at all. Inside the pot are some dark diffuse surfaces where the light strikes at a tangential angle. (There are also some shadows inside the pot.) Photo by Ryman Cabannes/Image Professionals GmbH/Alamy Stock Photo.

Source: AIMA

Light Reflection



Source: Prof. Daugman's Slides

Color Constancy

- Perceived colors hardly depend on the wavelengths of illumination, but depend greatly on local context
- The brown tile at the center of the illuminated upper face of the cube, and the orange tile at the center of the shadowed front face, are actually returning the same light to the eye

Source: Prof. Daugman's Slides

Other Properties of Digital Image

- Pixel arrays - scanned, drawn, or captured by digital camera
- Spatial Resolution
  - Pixel density
- Luminance (and Color) Resolution
  - How many bits per pixel
- Some image formats can also use compression to reduce the size of the file
  - lossy: .jpg, .gif
  - lossless: .png
  - Non-compressive: .tiff, .bmp

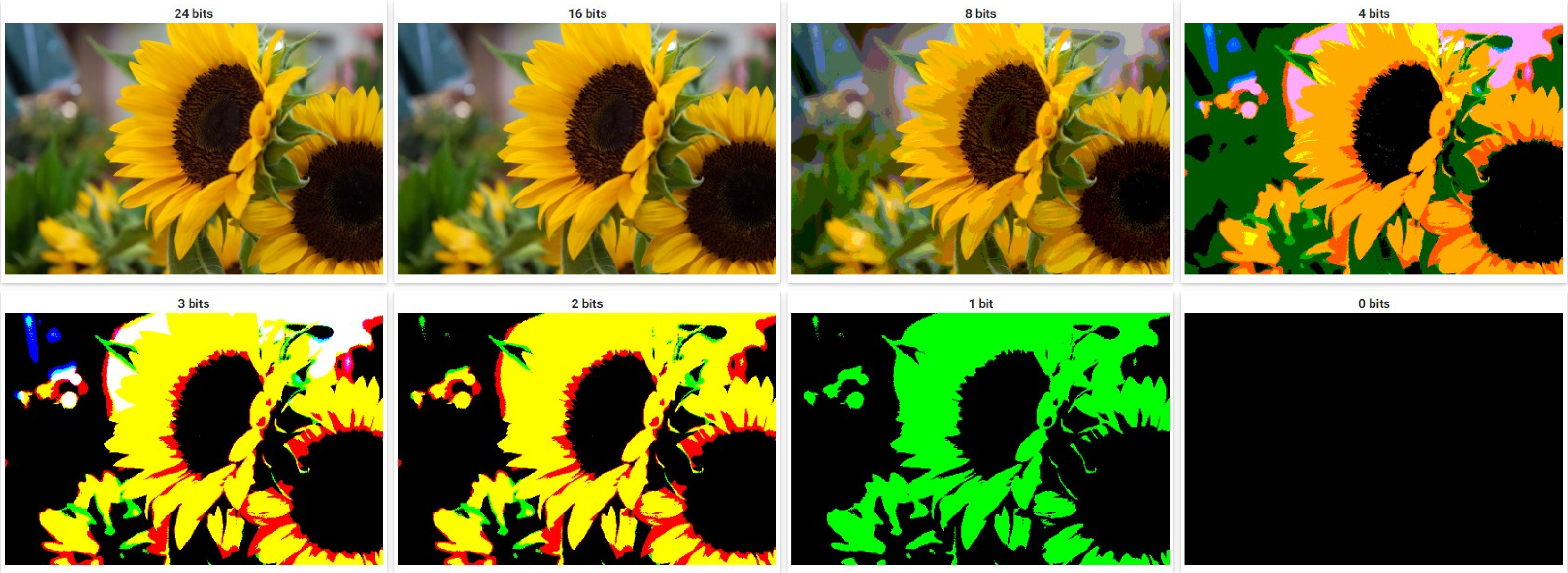
Effects of Spatial Resolution

108 × 108

1080 × 1080

Unknown Source

Effects of Color Resolution



Unknown Source

Colors in Digital Image

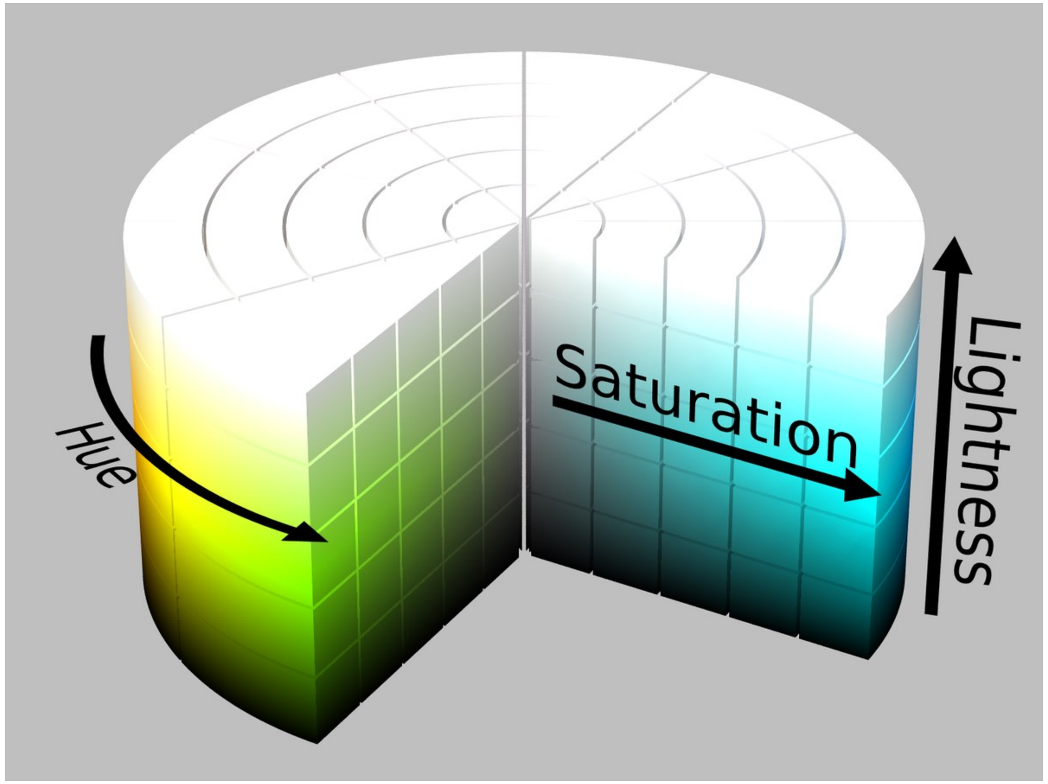
- Principle of Trichromancy: it is possible to construct another spectral energy density (color) consisting of a mixture of just three colors – for human perception
- RGB – Red, Green, Blue
  - Pixel color consists of these three components
  - Each component value ranged from 0 (none) to MAX (full)
  - Related is CMY (Cyan, Magenta, Yellow)
- Image can also be in grayscale, where each pixel has one value from 0 (black) to MAX (white)

Color in Digital Image (cont.)

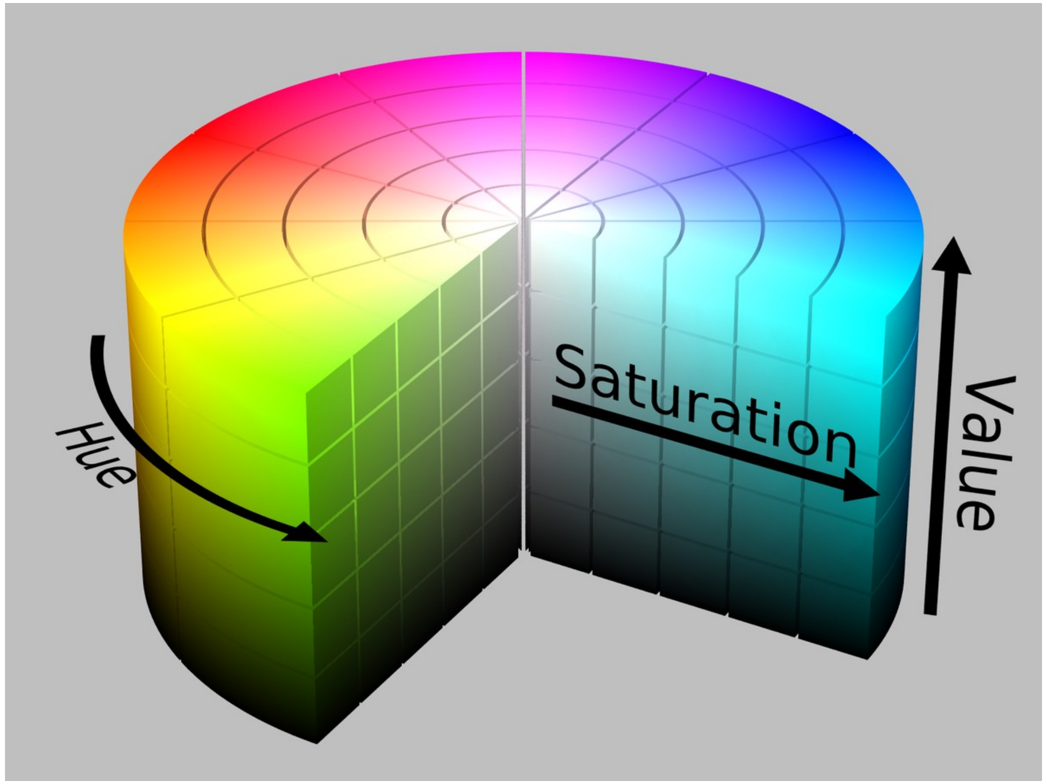
- HSV – Hue, Saturation, Value
  - Hue represents different color in radial slice
  - Saturation: how full the color is. S at 0 mean there's no color (gray)
  - Value: amount of light shining to the color.
    - V at 0 means no light (black)
    - V and max is color at full brightness.
  - Related is HSL, where L is lightness, amount of white (or black) color in the mix.
    - Color at max L is pure white and 0 is pure black.

Color: HSL vs HSV

HSL



HSV



Source: Wikipedia

Image Histogram

- Graphical Representation of frequency of tone or intensity of pixels in an image

Source: Getty Images

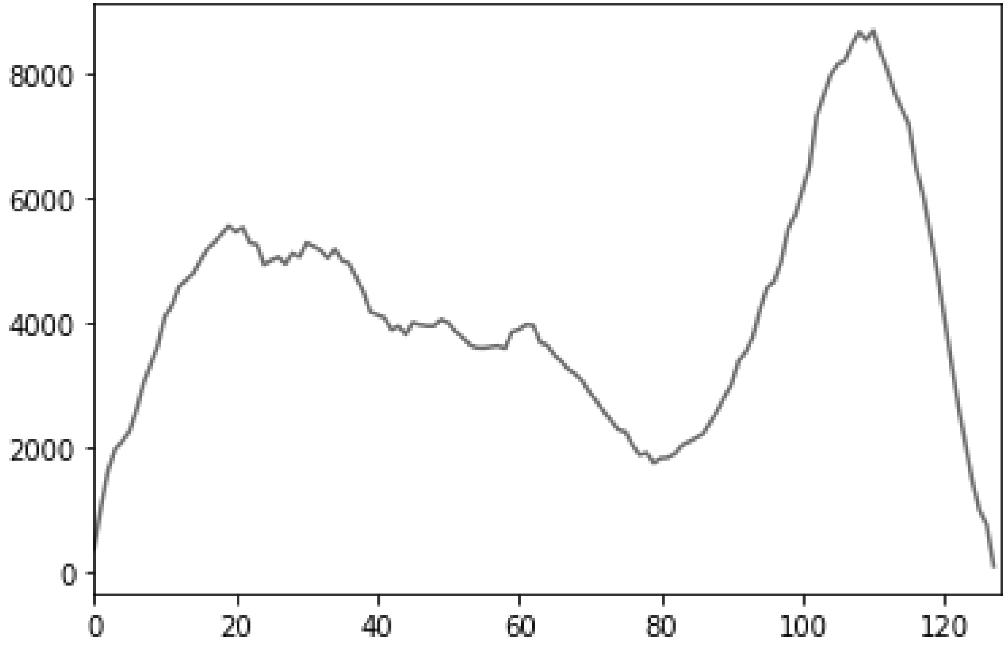


Image Processing

Image Processing

- “A method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it.”
- A signal processing where:
  - Input: an image
  - Output: an image, or characteristics/features associated with that image (the latter can also be called feature detection/extraction)

Convolution

- Apply filter, or function, to pixels in an image.
  - Creating new image based on information of the original image
  - Example: Gaussian blur
- Type of functions is denoted by kernel
- Convolution will be used for many image processing tasks
  - Smoothing
  - Thresholding
  - Edge detection
  - etc.

Smoothing

- Blur the image by convolving the image with a low-pass filter kernel.
  - Remove high frequency content such as noise and edges
  - Averaging - average the pixel value by mean of nearby pixels, using box kernel

$$K = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

  - Gaussian - average with Gaussian kernel (x and y are distance in axis x and y of the point from origin point)

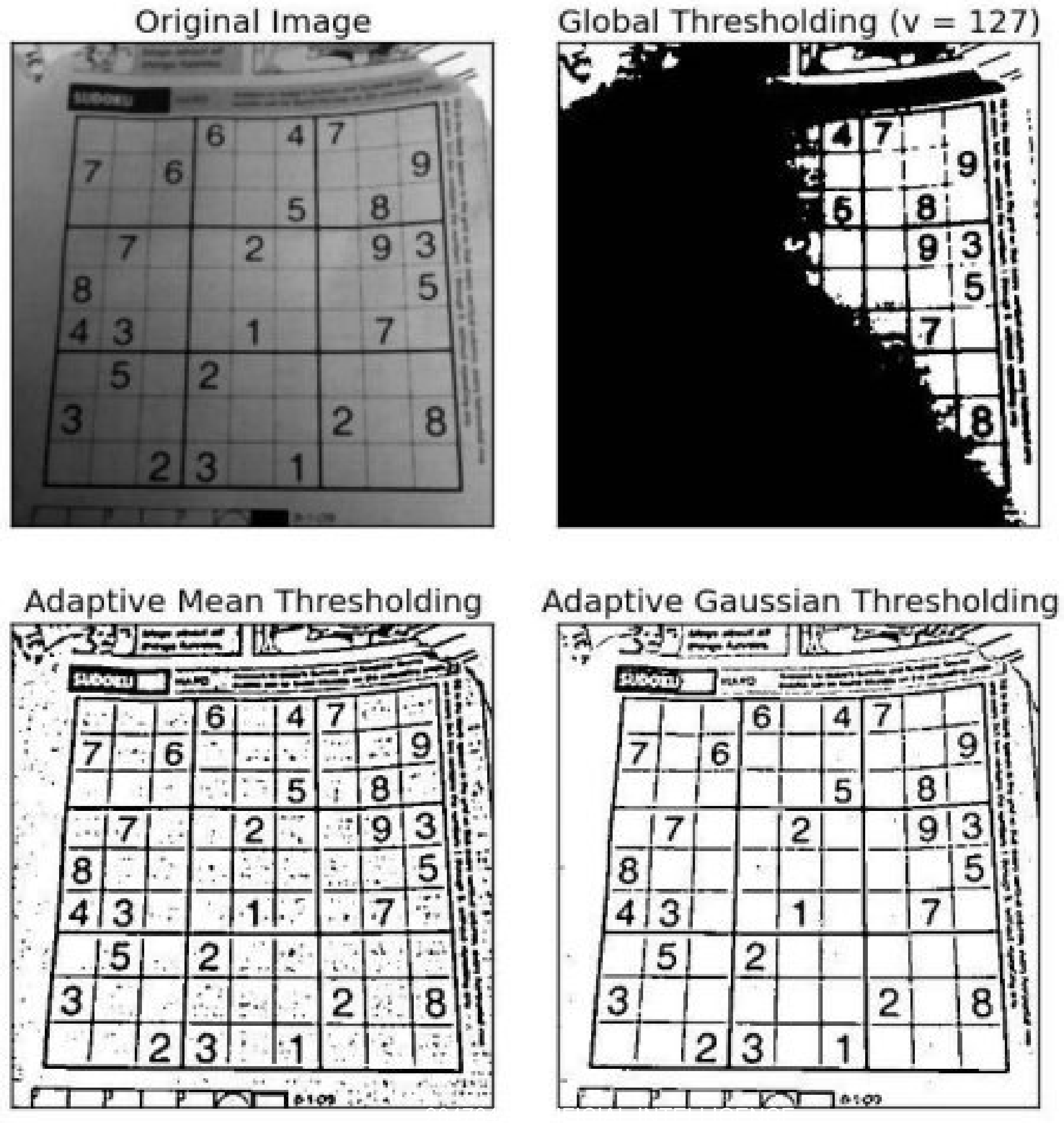
$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

  - Median - average the pixel value by median of nearby pixels
  - Bilateral – modified Gaussian smoothing that preserve edges

Thresholding

- If a pixel has value less than threshold, new pixel has value of 0 (black), maximum value (white) otherwise
  - Simple Thresholding – use one threshold value (global threshold) throughout
  - Adaptive Thresholding – determine threshold value based on small region around it, can help with lighting conditions
  - Otsu’s Method – find optimal threshold value for global threshold first
- Benefit from smoothing to remove noises before thresholding

Thresholding (cont.)



Source: OpenCV

Edge Detection

- Edges: straight lines or curves in the image which there is a "significant" change in image brightness
- Edge detection doesn't care about the type of the edge

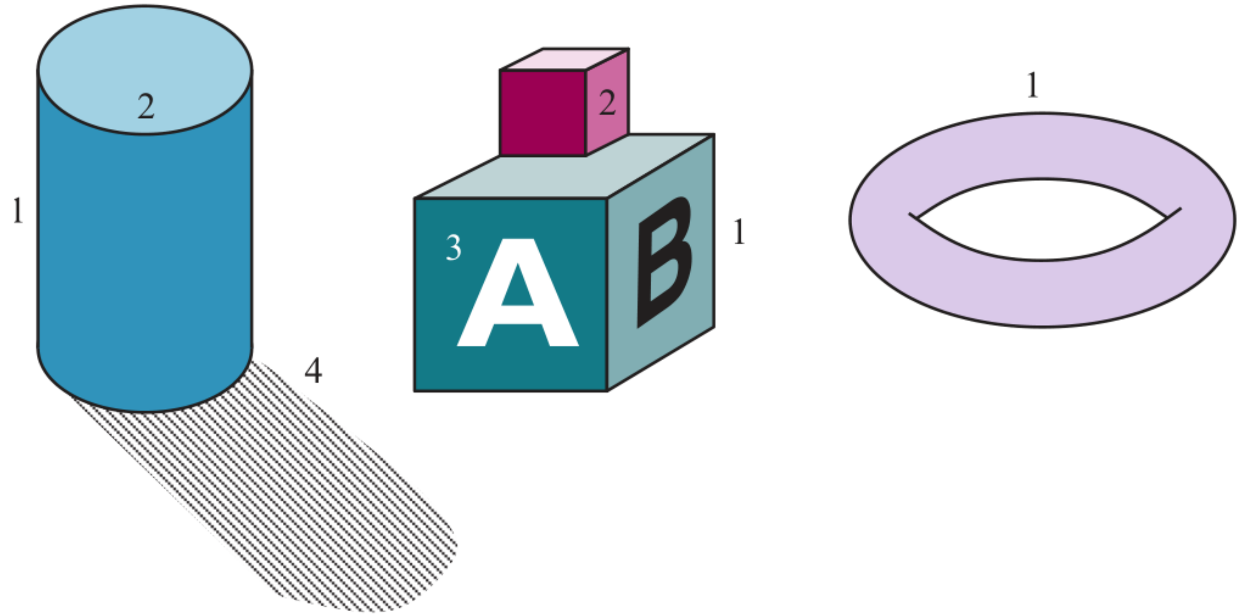


Figure 25.6 Different kinds of edges: (1) depth discontinuities; (2) surface orientation discontinuities; (3) reflectance discontinuities; (4) illumination discontinuities (shadows).

Source: AIMA

Edge Detection (cont.)

- Edges correspond to locations in images where the <u>brightness (Intensity, *I*(*x*)) undergoes a sharp change</u>
- Look for where derivative of intensity (*I*'(*x*)) is large for edge point
- Also benefit from convolution with smoothing

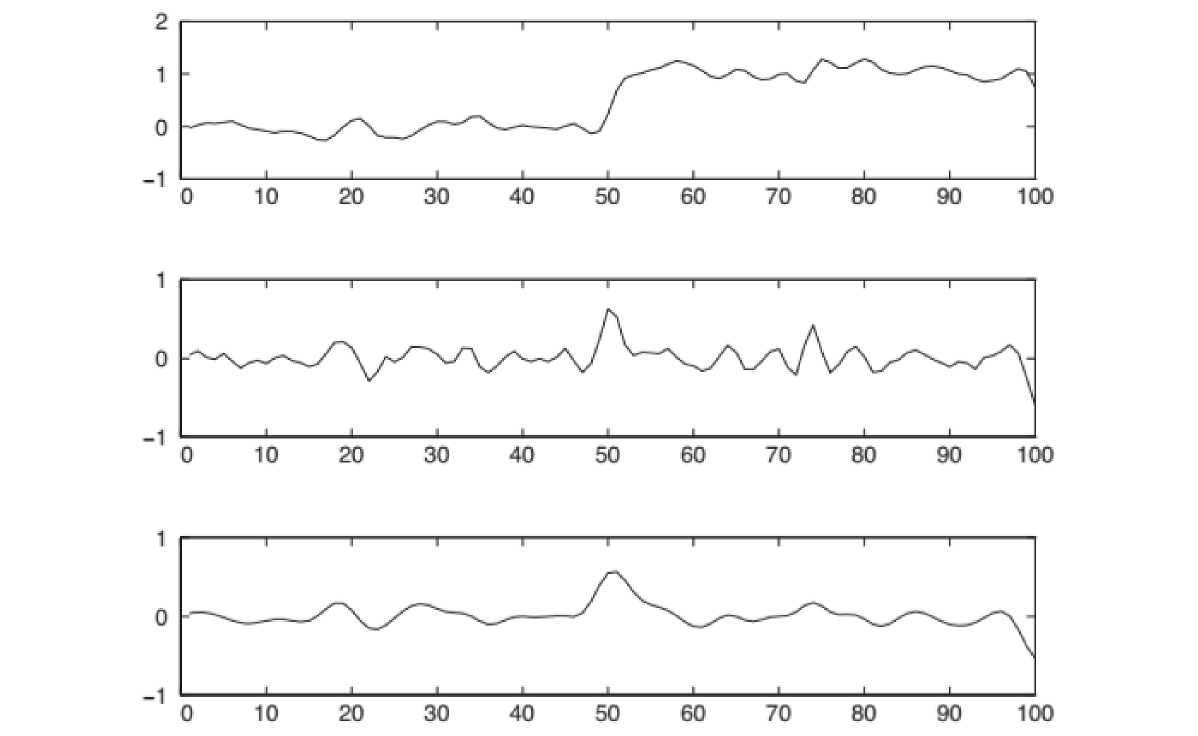
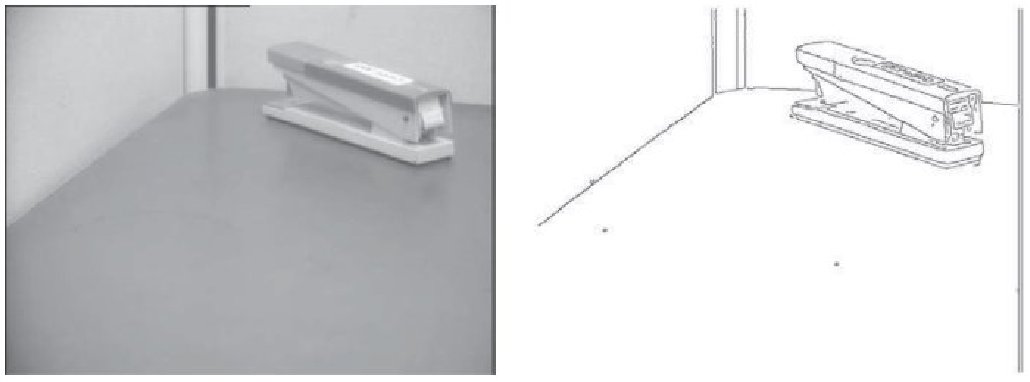


Figure 24.8 Top: Intensity profile $I(x)$ along a one-dimensional section across an edge at $x = 50$. Middle: The derivative of intensity, $I'(x)$. Large values of this function correspond to edges, but the function is noisy. Bottom: The derivative of a smoothed version of the intensity, $(I * G_\sigma)'$, which can be computed in one step as the convolution $I * G'_\sigma$. The noisy candidate edge at $x = 75$ has disappeared.



(a) (b)

Figure 24.7 (a) Photograph of a stapler. (b) Edges computed from (a).

Source: AIMA

Edge Detection (cont.)

- In two dimension (x, y), find gradient of intensity vector, with angle θ, or edge orientation

$$\nabla I = \begin{pmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{pmatrix} = \begin{pmatrix} I_x \\ I_y \end{pmatrix}$$

- Where the direction of gradient:

$$\frac{\nabla I}{||\nabla I||} = \begin{pmatrix} I_x \\ I_y \end{pmatrix}$$

- How to find gradient depends on the algorithm
- We can obtain edges by finding edge points and linking them together.
  - Edge points will be ones with the highest magnitude in the local area, an above a suitable threshold
  - Link edge points to the same edge curves: two neighboring edge pixels with consistent orientations

Example: Canny Edge Detection

- Multi-stage: smoothing is done first, then computing intensity gradient, then filtering edge pixels and link them together
- Intensity gradient is computed by edge detector operator, such as Sobel operator (A is the image)

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \text{ and } \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

- Local maxima edge pixels are filtered using double threshold (high and low)
  1. Pixels with gradient value below low threshold are ignored
  2. Pixels with gradient value above high threshold (strong edge pixels) are accepted
  3. Pixels with gradient value between high and low threshold are accepted iff they links with pixels from 2.

Example: Canny Edge Detection (cont.)

- OpenCV Tutorial:
  - https://docs.opencv.org/master/da/d22/tutorial_py_canny.html
- Code Example:
  - https://colab.research.google.com/drive/1vTtxUNbjO9xsq03nOA5QXD-T9lnxiobY?usp=sharing

Texture

- Visual feel of the surface
  - Can affect edge detection that use intensity gradient
- However, texture gradient orientation is invariant to changes in illumination
- Can use edge detection by looking for differences in texture properties

Texture Example

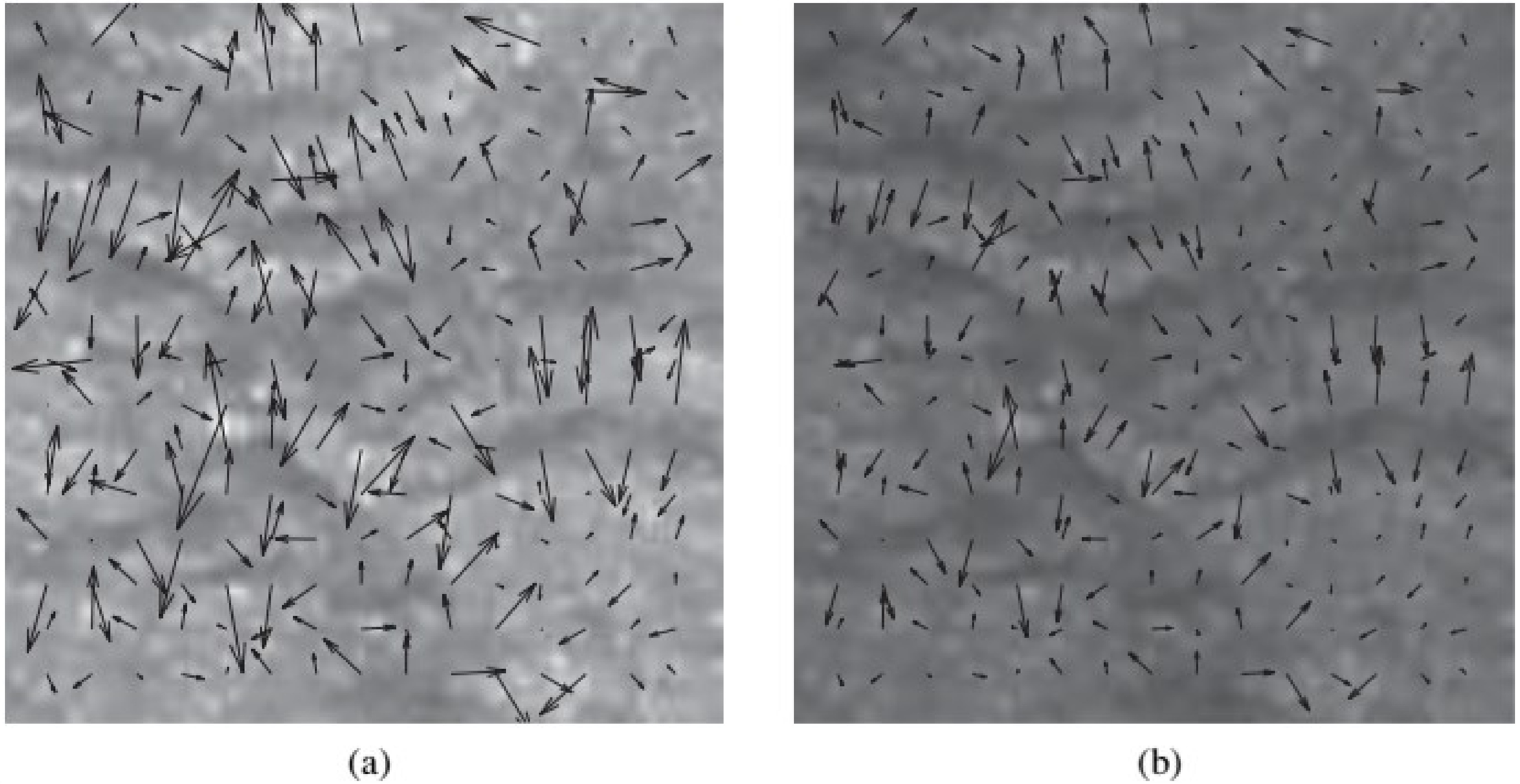


Figure 24.9 Two images of the same texture of crumpled rice paper, with different illumination levels. The gradient vector field (at every eighth pixel) is plotted on top of each one. Notice that, as the light gets darker, all the gradient vectors get shorter. The vectors do not rotate, so the gradient orientations do not change.

Source: AIMA

Optical Flow

- For video sequence, show movement of objects

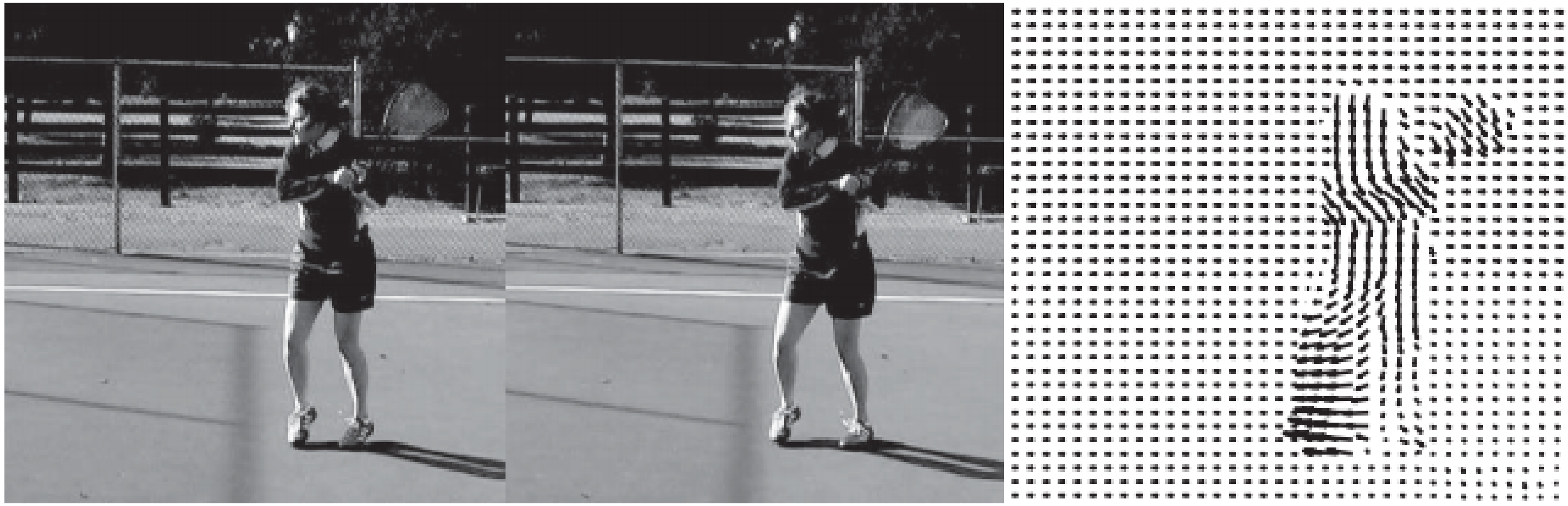


Figure 24.10 Two frames of a video sequence. On the right is the optical flow field corresponding to the displacement from one frame to the other. Note how the movement of the tennis racket and the front leg is captured by the directions of the arrows. (Courtesy of Thomas Brox.)

Source: AIMA

Image Segmentation

- Breaking an image into regions of similar pixels
  - Similar in brightness, color, or texture
  - Regions should represent objects
- Two general approaches
  1. Focusing on detecting boundaries of regions
  2. Focusing on detecting the regions themselves: clustering pixels into regions
- Can benefit from high-level knowledge, such as likely objects in the scene
  - Oversegment images into superpixels, then used knowledge-based algorithm to put superpixels togethers

Image Segmentation (cont.)

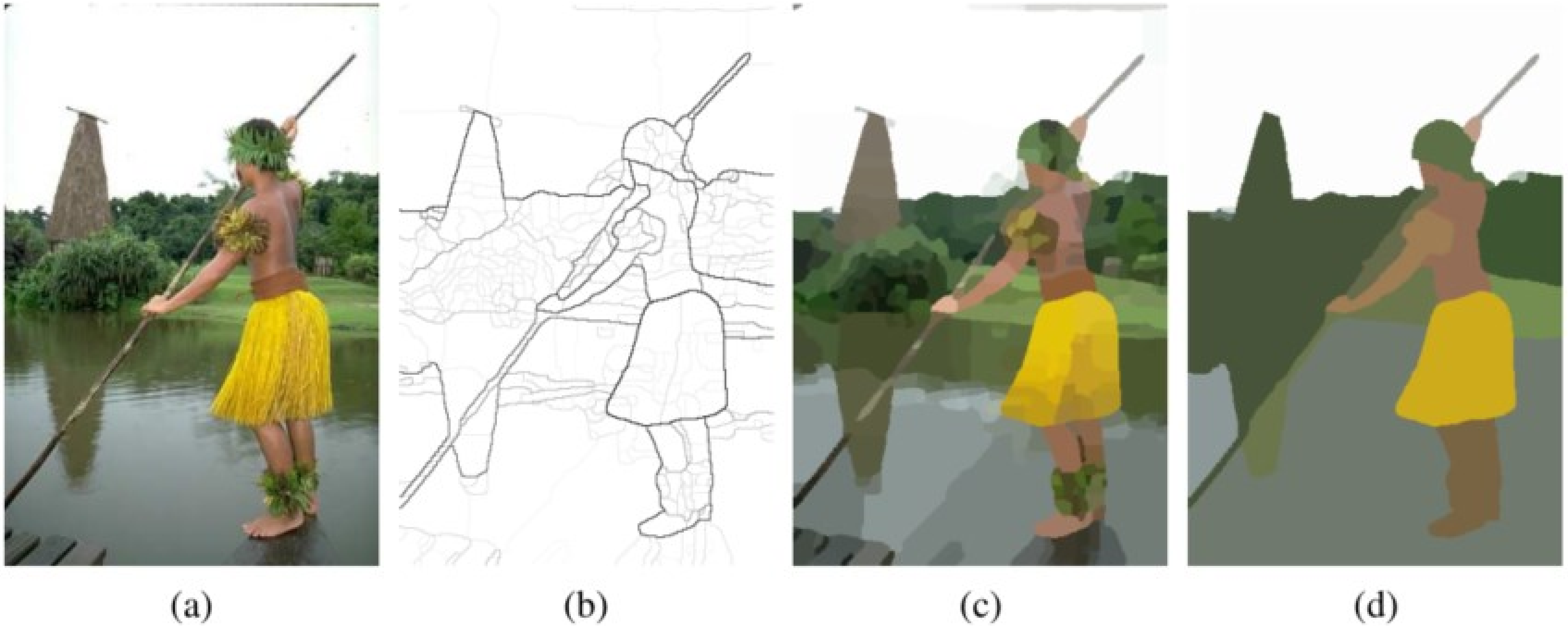


Figure 25.10 (a) Original image. (b) Boundary contours, where the higher the P_b value, the darker the contour. (c) Segmentation into regions, corresponding to a fine partition of the image. Regions are rendered in their mean colors. (d) Segmentation into regions, corresponding to a coarser partition of the image, resulting in fewer regions. (Images courtesy of Pablo Arbelaez, Michael Maire, Charless Fowlkes and Jitendra Malik.)

Segmentation Example:

- Watershed Algorithm
  - Work on grayscale level
  - Use topographic surface with height based on grayscale level
  - Steps
    - Water start filing from local minima (valleys)
    - Build barrier to prevent water from different sources to join
    - Once the water reach all local maxima (peaks), the barriers are the segmentation results
  - https://people.cmm.minesparis.psl.eu/users/beucher/wtshed.html
  - https://docs.opencv.org/master/d3/db4/tutorial_py_watershed.html