16. Perception

16.1 Speech processing

- Study of auditory (speech) signals and processing methods for these signals.
- Process of converting speech signal to a sequence of words, or a sequence of words to speech.
- Intersection of digital signal processing and natural language processing.
- Goals and categories:
  - Speech recognition for analysis of linguistic content.
  - Speaker recognition (voice recognition)
  - Signal improvement via noise reduction
  - Compression and transmission of speech (e.g., in telecommunication)
  - Speech synthesis: computer generated speech
  - Voice analysis for medical purposes, e.g., dysfunction of vocal cord.

16.2 Speech recognition

- Goal: find the most probable word sequence given the acoustic stream.
- Requires several levels of processing.
- Words are transmitted as sound waves, which is subjected to signal processing:

  → Feature extraction: amplitude and frequency

    → Features are mapped to sound units called *phones*.

      → Translate the sequence of phones into words.

        (→ NL understanding)
- Applications
  - Dictation
  - Voice dialing
  - Simple data entry
  - Telephone booking

16.3 Speech generation

- Artificial production of human speech
  - Map words into sequence of phones.
  - From ordinary text or some symbolic linguistic representation.
- Performance criteria
  - Intelligibility — ease of understanding
    - Professionals, e.g., process or vehicle control
    - Highly motivated users with special needs, e.g., visually impaired, hostile environment.
  - Naturalness (or pleasentness) — similarity to human voice
    - General public applications, e.g., telephone information retrieval

- Articulatory synthesis
  - Physical model of human vocal tract.
  - Physiological model of speech production.
  - Mostly of academic interest until recently.
- Formant synthesis
  - Not based on human speech but models the main acoustic features of the speech signal.
  - Robotic sounding speech
  - Intelligible even at high speed
  - Smaller systems ← no database

- Concatenative synthesis
  - Based on natural speech databases.
  - Unit selection synthesis
    - Large database of recorded speech utterances
      - segmented into individual phones, syllables, words, or phrases.
      - stored by pitch, duration, neighboring phones, etc.
    - Most natural sounding ← minimal signal processing.
  - Diphone synthesis
    - Small database of all diphones = sound to sound transitions.
    - Depends on language, for instance Spanish has 800 diphones, German 2500.
    - Only one instance of each diphone in the database.
    - At runtime (generation) signal processing applied.

16.4 Challenges

- Dictate text directly into document
  - Requires extensive training to familiarize the system to individual's voice, and a clean environment.
  - Recognition errors can be corrected afterwards.
  - Claim: 98-99% accuracy.
- A long way from dictation to a system that responds to:
  - "Back up all the program files for the projects I have worked on today."
  - Requires natural language understanding.
  - Uncertainty in recognition of individual words.
    - Segmentation –– we are dealing with continuous input.
    - Many words sound the same (in English): mail vs. male, fare vs. fair, here vs. hear, etc.

16.5 Human communication

- Humans are good at making guesses about unidentified words from
  - Context
  - Experience
  - Expectations
  - Gestures, facial expressions, tone.
- Rephrasing and corrections are often possible.
- Human are good at distinguishing different sound sources (compare to cocktail party effect).

16.6 Phone

- Individual unit of sound.
- Smallest identifiable unit in a stream of speech.
- Pronounced in the defined way.
- Phones need to be identified and grouped to ensure that all the words of a language can be distinguished from each other.
- Examples:
  - [b] bin
  - [p] pin
  - [th] thin
  - [l] lip
  - [ay] iris
- Glottal stop [?]
  - Consonant sound
  - Vocal cords pressed together to stop the airflow.
  - Examples: [Anna? elää](in Finnish), cat [kæ?t] (in English)

Phones may have different sounds in different contexts:

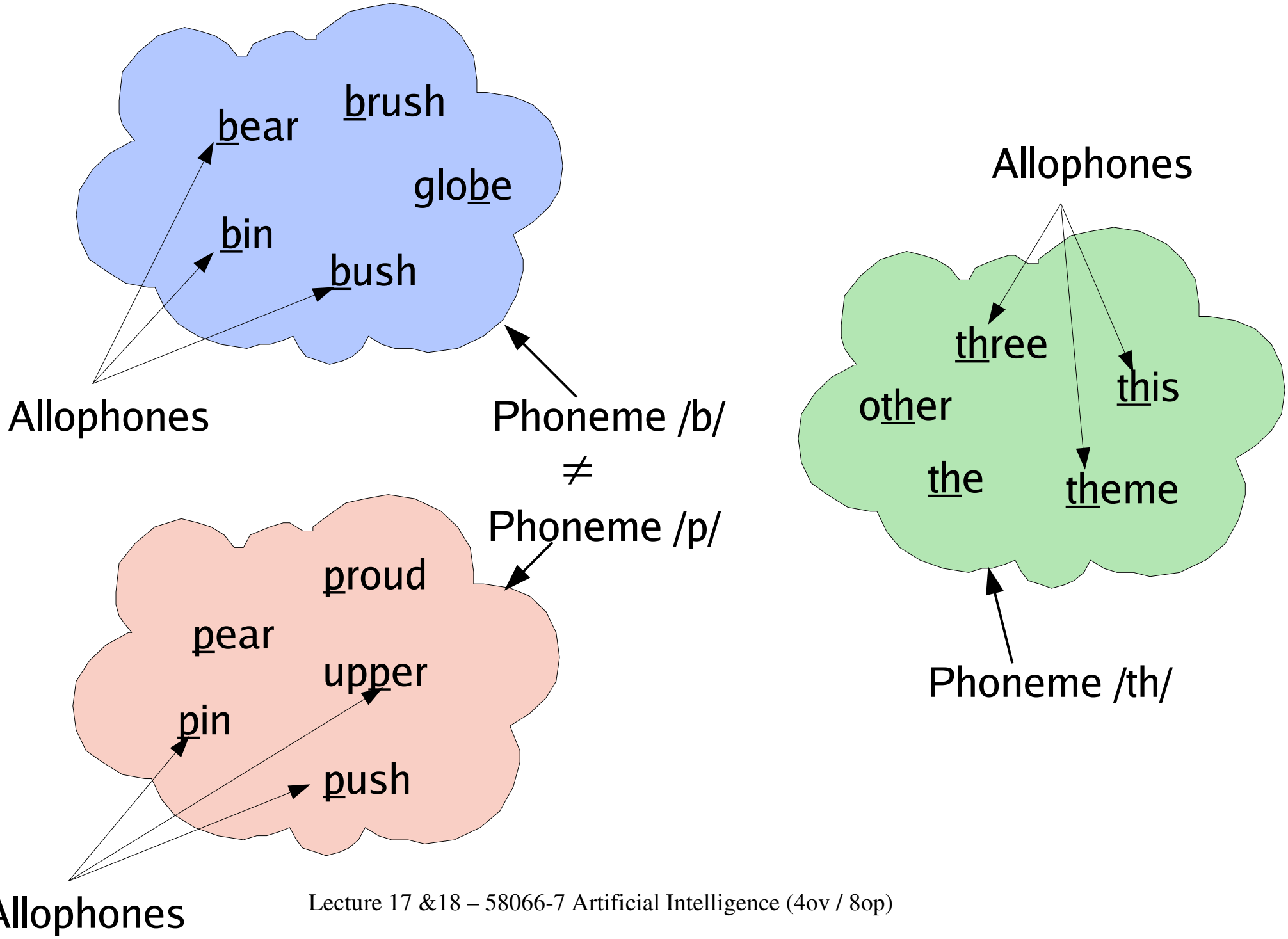
three — then

Variations are called allophones.

16.7 Phoneme

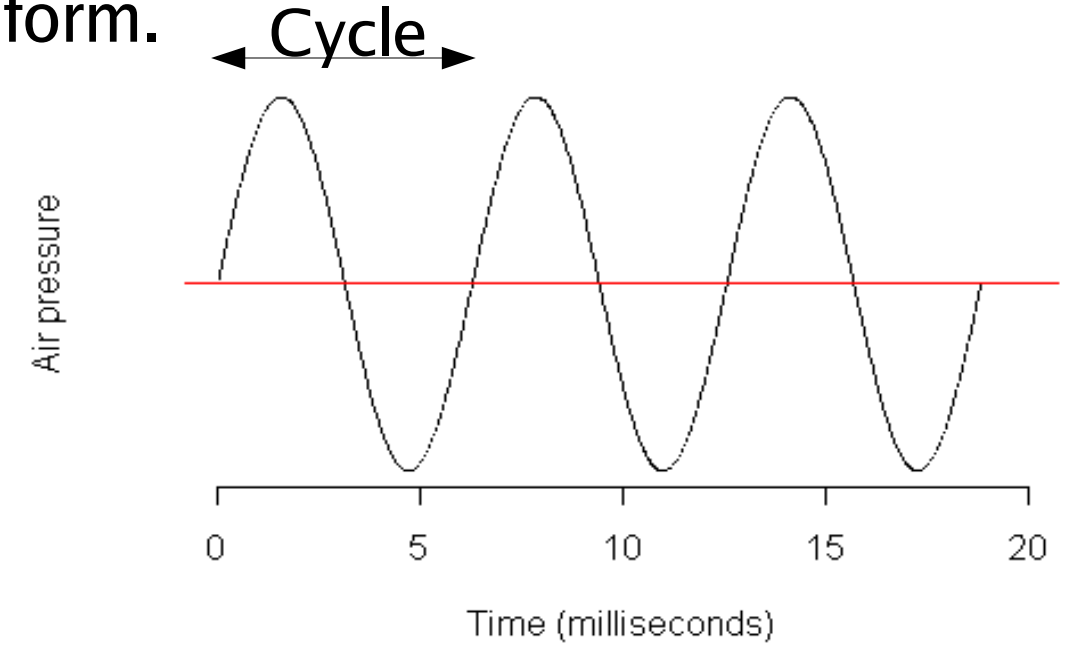
- The smallest contrastive unit in a sound system of a particular language.
- Minimal unit that serves to distinguish between meanings of words.
- Contains all the variations (allophones) of a phone, for instance /th/

  → may be pronounced in various ways, depending on the number of allophones.
- Examples: /b/, /r/, /l/.


brush
bear
globe
bin
bush
Allophones
Phoneme /b/
≠
Phoneme /p/
Allophones
three
this
other
the
theme
Phoneme /th/
proud
pear
upper
pin
push
Allophones

16.8 Sound waves

- Variations in the air pressure. Two key features:
  - *Amplitude* = measure of air pressure at one time.
  - *Frequency* = How many times the wave repeats itself in a second. Each repeat is a *cycle*.
- Taking of measurements = *sampling*.
- Identification of phones not possible from the visual waveform.

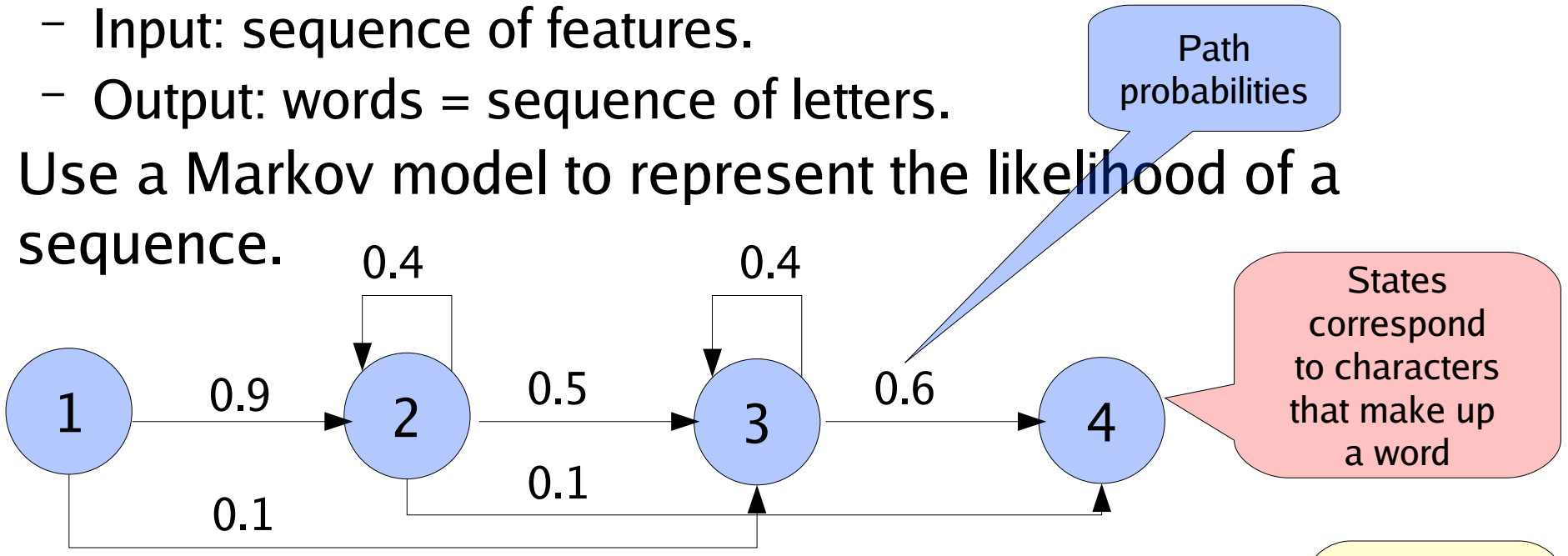


16.9 Signal processing

- Break signal into number of overlapping chunks (*frames*) $\rightarrow$ feature extraction.
- *Spectrum*
  - represents different frequency components of the wave.
  - identifies dominant frequencies that can be matched to phones to produce likelihood estimates.
- Fourier transform
  - Represent sound wave as a composition of sine waves.
  - Fourier transform can identify the dominant frequencies and amplitudes that make up the wave.
- Linear predictive coding (LPC)
  - Each sample is represented as a linear combination of the previous samples.
  - Coefficients are estimated by minimizing the distance between the predicted signal and the actual signal.

16.10 Recognition

- Find out which word is represented by the extracted features.
  - Input: sequence of features.
  - Output: words = sequence of letters.
- Use a Markov model to represent the likelihood of a sequence.

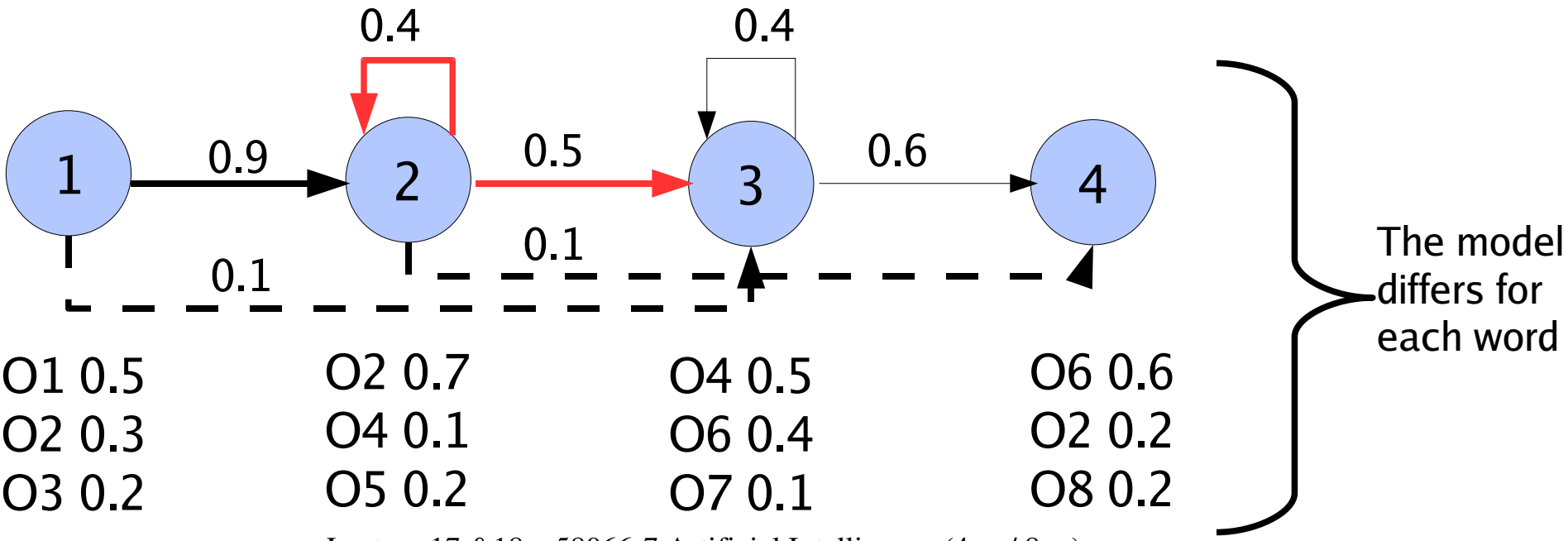


16.11 Hidden Markov Model

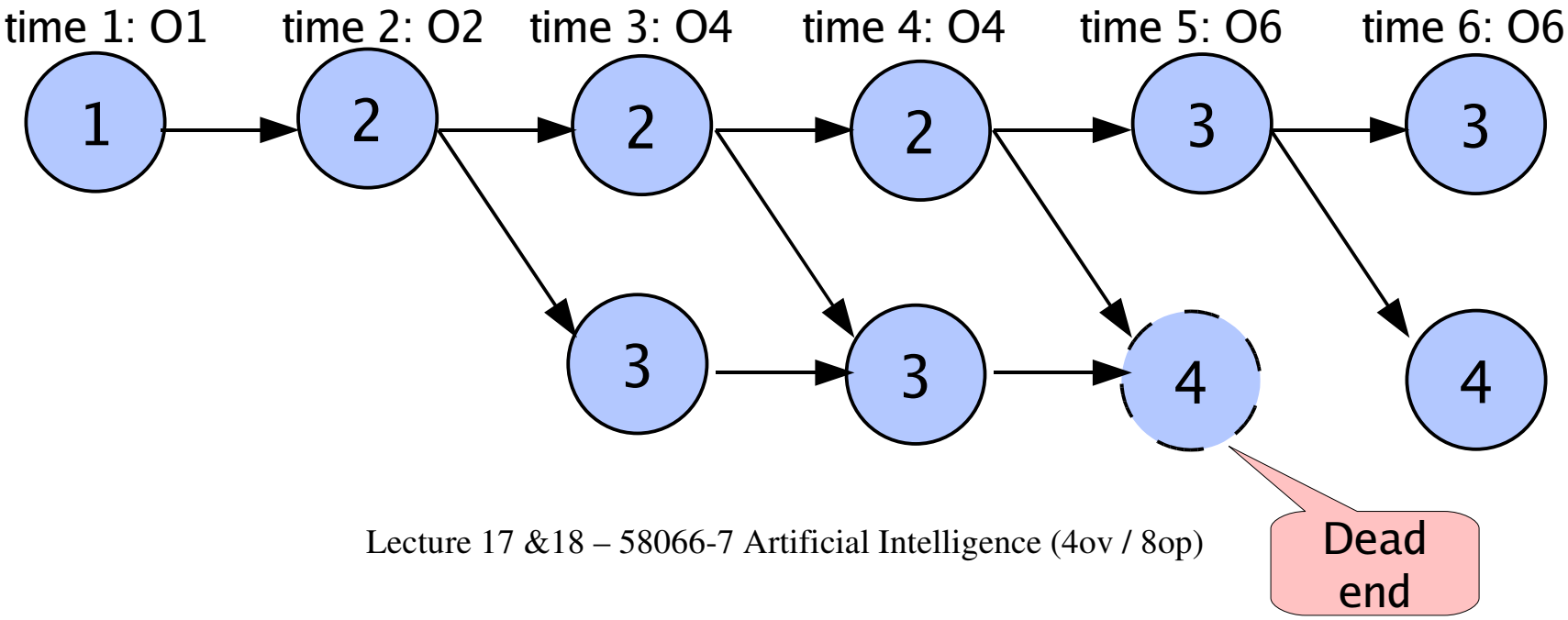
- The state which the machine is in is hidden from the user.
  - We do not not know which state the input features correspond to.
  - The features may not correspond exactly to the states of HMM.
  - Different phones may have different sounds, and phones may share some sound features.
- We know which states more likely correspond to the feature $\rightarrow$ some states can be ruled out.
  - Some features co-occur more often.
  - More likely that 'y' comes after 'ph' than 't'.

16.12 HMM example

- Consider the sequence O1 O2 O4 O4 O6 O6.
- Symbols are observed at time steps t_1, t_2, ..., t_6.
- The recognizer does not know whether the first occurrence of O4 is generated by state 2 or 3.



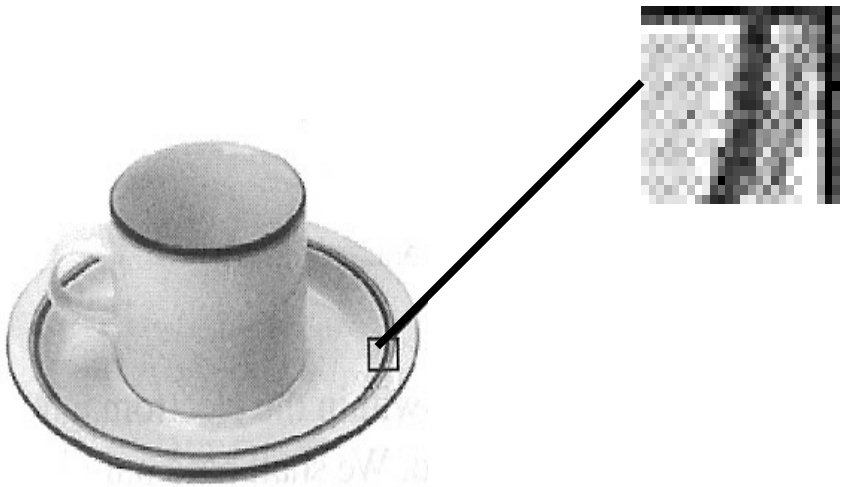
- Which path most likely generates the sequence of observations O1 O2 O4 O4 O6 O6?
- 6 possible paths. For instance: $p_1 = 1 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 3 \rightarrow 4$
- $Pr(p_1) = Pr(O1|s_1) \times 0.9 \times Pr(O2|s_2) \times 0.4 \times Pr(O4|s_2) \times 0.4 \times Pr(O4|s_2) \times 0.5 \times Pr(O6|s_3) \times 0.6 \times Pr(O6|s_4) = 3.6 \times 10^{-5}$
- $\sum_{i=1,6} Pr(p_i)$ is the probability that the word was generated by this model.



16.13 Word model

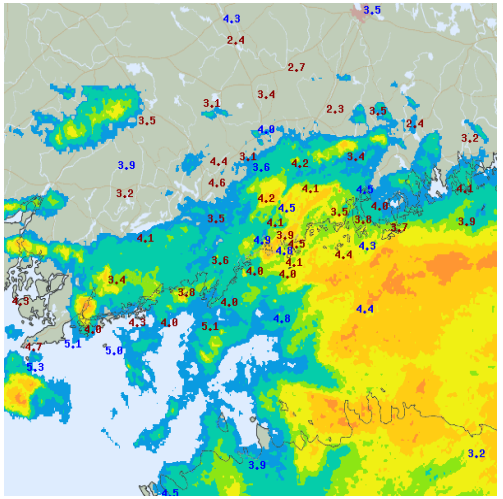
- The task of the speech recognizer is to determine which word model is the most likely.
- Assume no knowledge of the start or terminating state.
- When an observation is received, the state it corresponds to is unknown.
- An algorithm for finding the best path is called **Viterbi algorithm**.

16.14 Visual processing



243	243	243	242	241	242	242	240	241	245	236	17	235	240
243	243	243	243	242	242	242	241	243	243	109	17	236	7
243	243	243	243	242	243	242	241	243	188	20	21	239	247
245	243	243	243	243	242	242	243	245	236	67	109	188	236
243	245	243	243	243	243	241	243	241	18	67	237	238	247
245	243	243	243	243	243	242	245	247	21	19	188	7	188
245	245	243	243	243	242	243	240	18	67	237	240	247	243
243	245	243	243	243	243	245	237	67	18	240	239	239	245
245	245	243	243	243	243	7	20	19	239	188	237	242	243
245	243	243	243	243	238	19	20	247	240	237	241	245	243
245	243	243	244	241	234	21	236	242	237	188	244	243	245
245	243	243	243	248	21	109	240	239	239	244	243	245	243
243	245	243	236	21	109	188	7	239	243	245	245	243	245
245	243	236	21	18	188	238	247	242	245	243	243	245	245
243	236	21	21	18	188	238	247	242	245	243	243	245	245
255	255	255	255	255	255	255	255	255	255	255	255	255	255

- Image processing
  - Manipulation of digital image to generate a second image that differs somehow from the the first.
- Computer vision
  - Extraction of numerical or symbolic information from images:
    - Recognize objects (their invariant properties) on robot's way.
    - Motion detection:
      - Motion of other objects
      - Apparent motion of the environment caused by agent's own motion.



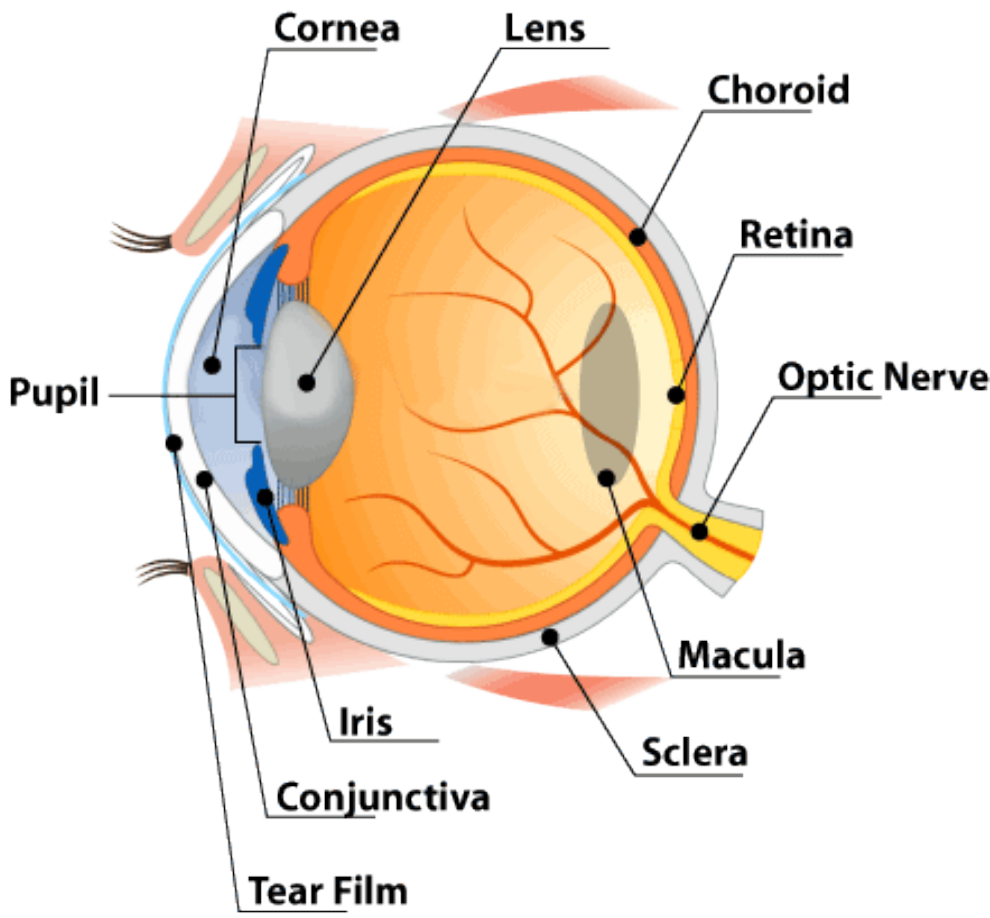
"It's raining in Helsinki."
- each pixel in the image corresponds to rain intensity at the specific location.

16.15 Applications of computer vision

- Optical character recognition
  - Interpretation of printed, hand written or cursive text
  - For example, USPS automatically recognizes hand-written postal addresses in envelopes.
- Biometry
  - Identification of humans from their physical characteristics
  - Use images of the characteristics, e.g., iris or fingerprints.
- Aerial photography
  - Mapping, remote sensing, land-use analysis.
- Content-based image retrieval

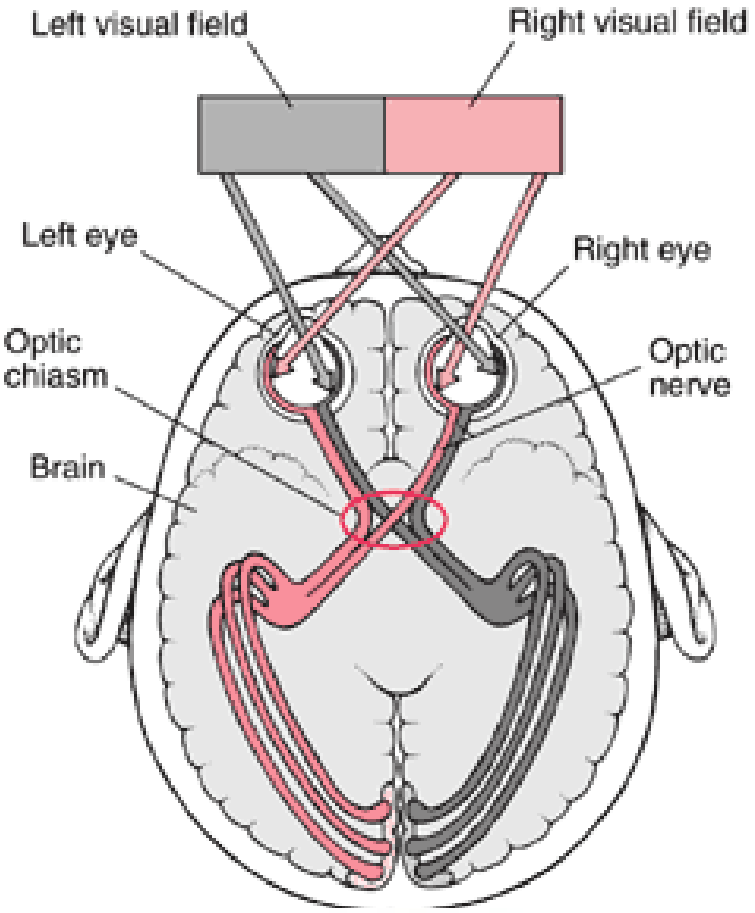
16.16 Human eye

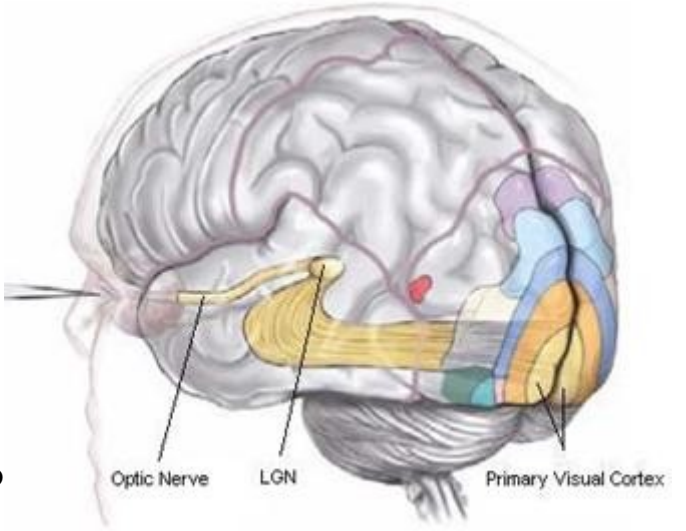
- Light reflects from surfaces
  → Detected by light-sensitive cells in the retina
    →Translated to electrical signals
      →Transmitted to brain through the optic nerve.
        →The image is interpreted on the visual cortex
- Iris regulates the amount of light entering the eye.



- Retina contains two types of receptors:
  - Rods
    - Sensitive to light: respond to dim light and dark conditions.
    - Not sensitive to color
  - Cones
    - Have high visual acuity
    - Different cones respond to different wavelengths of light $\rightarrow$ color vision
  - Rods are more numerous (about 120 million) than cones (6-7 million).
  - Cones concentrated in the central retina, rods more evenly spread.

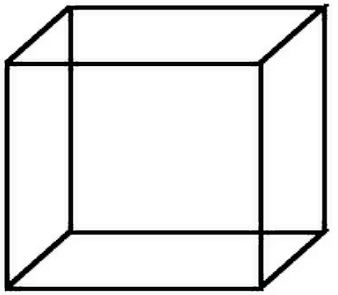
- Two forward-looking eyes
- Information from right visual field (left side of the retina of both eyes) is projected to the left hemisphere.
- Information from left visual field (right side of the retina of both eyes) is projected to the right hemisphere.

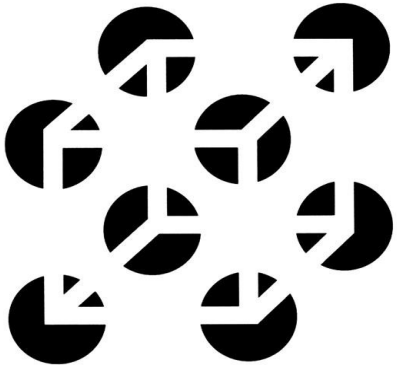


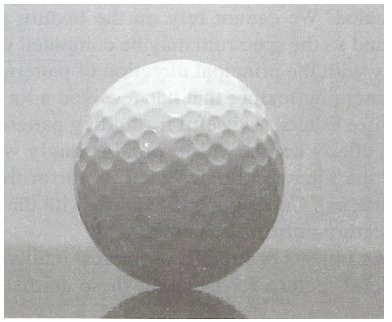


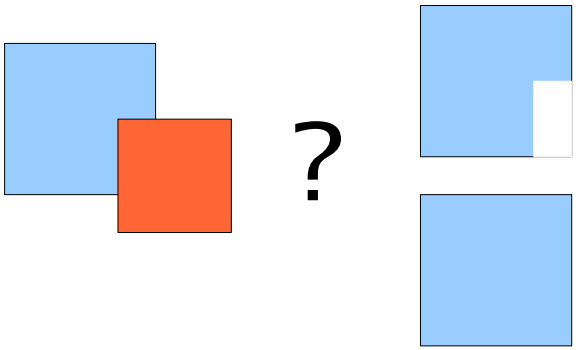
16.17 Challenges

- Segmentation
  - Visual scene is represented in pixels
  - Which pixels belong to the same object?
- Part-whole
  - Which objects belong together?
- Object consistency
  - Shadow
  - Occlusion
  - Movement
- Face recognition
- Depth perception

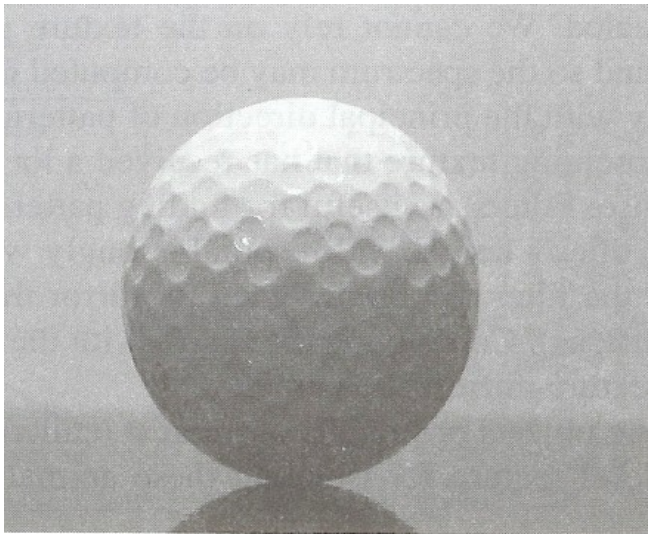




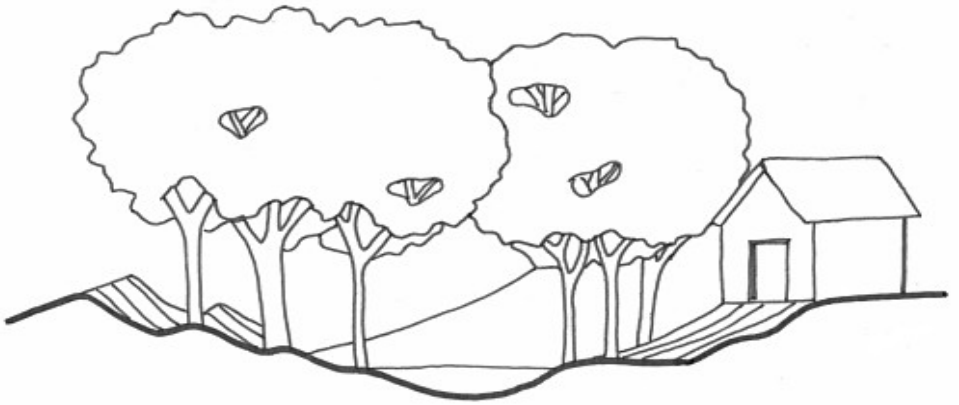




16.18 Visual cues



- Features that help identify an object.
- Color
- Shading, shadows
  - Variation in light intensity.
  - Determined by the geometry and reflectance properties of surfaces.
  - Model the effect of light sources.
- Contour



- Texture
  - Qualitative perception combining seeing and touching.
  - Regularly repeating pattern
    - Examples: windows on a building, stitches in a sweater, pebbles on a beach, or crowd of people in a stadium.
    - Regularity can be statistical.
    - Use rules to represent.
    - Apply filters to find texture patterns.
  - Function of scale
  - Tells not only about the material, but also about movement, shape, and orientation.

16.19 Motion

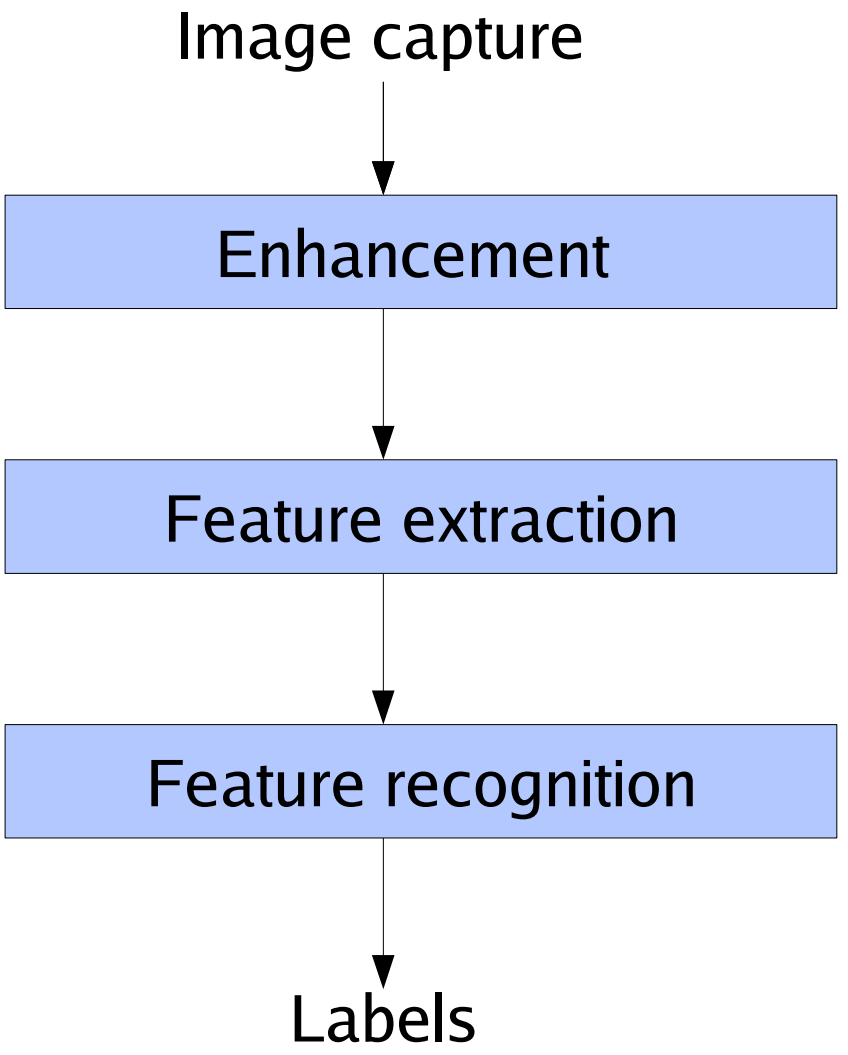
- Optical flow = apparent motion in the image
  - The direction and speed of motion of features in the image.
  - Relative motion between the viewer and the scene.
  - Tells also about distance.
  - Need to find corresponding points in two consecutive time frames ← similar intensity.
- Binocular stereopsis
  - Instead of using images separated by time, use one or more images separated in space.

16.20 Object recognition process

Remove any degradation introduced by the capture process.

Recognize structures potentially useful in the recognition of objects in the image:
- edge detection
- segmentation

Object recognition:
- Use features to assign labels to the image or its parts.

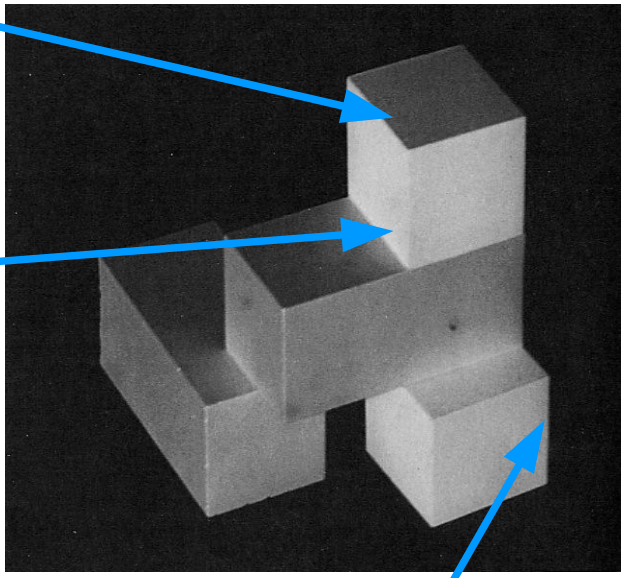


16.21 Object recognition

- Extract information from the image in a representational format that is useful for the task.
- The goal is to understand the contents of the image.
- Images are high-dimensional data
  - More training time required.
  - More training instances needed.
- Brightness-based
  - Per pixel: index to a feature vector.
  - Highly redundant: illumination invariance($\rightarrow$ PCA)
- Feature-based
  - Spatially localized features, such as regions and edges: there are fewer edges than pixels.
  - Location is the feature.

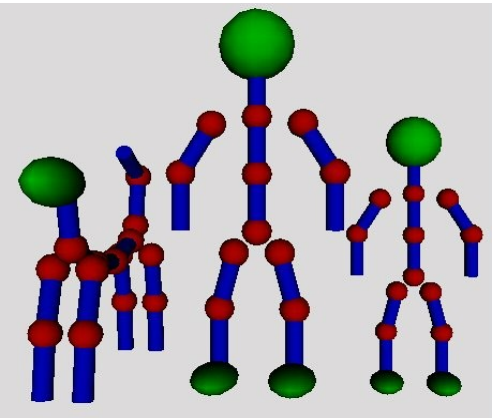
16.22 Edge detection

- Preprocessing
  - Images contain noise, modeled as Gaussian process.
  - Smoothing: each pixel is replaced by a weighted average of its neighbors.
- Objects have solid edges.
- Edged denote object boundaries.
- Formed by
  - abrupt change in brightness (illumination discontinuity, e.g., shadows)
  - depth discontinuity
  - surface orientation discontinuity
  - reflectance discontinuity: texture or pattern (e.g., zebra)



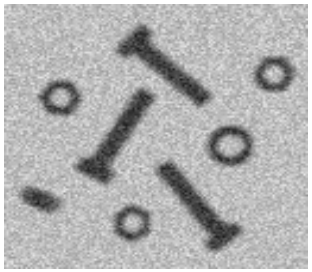
16.23 Segmentation

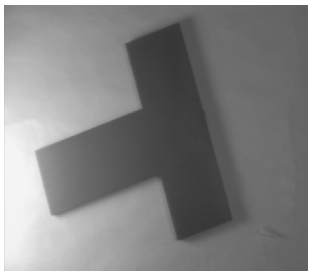
- Divide the image to homogeneous areas with no edges within.
- Group image elements based on some measure of similarity: color, texture, intensity.
- A general term that contains many forms of information extraction:
  - Edge detection can be seen as a subtask of segmentation.
  - Shape detection: human bodies may be detected as connected cylinder-like objects in the image.
  - Binary thresholding
    - Pixels that relate to an object share some characteristics.
    - Example: find a threshold for gray-level values that most accurately discriminates the pixels belonging to foreground and pixels belonging to background.
  - Splitting and merging
    - Divide non-homogeneous areas into two or more.
    - Combine two adjacent and similar areas (e.g., neighboring pixels) into one area.

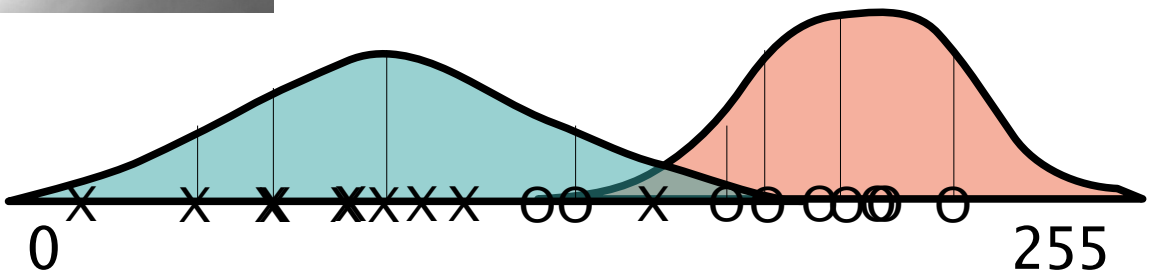


16.24 Binary thresholding

Observed data D	Latent data Z
12	0
46	1
5	0
178	1
x_5	z_5
...	...
x_N	z_N







- Trying to find parameters Θ that maximize $P(D|\Theta)$.
- Parameters for two-component Gaussian mixture model:
  - α_0,α_1: the probabilities of latent values 0 and 1.
  - μ_0, μ_1: mean values for components 0 and 1.
  - σ_0^2, σ_1^2: variances for components 0 and 1.
- $P(D|\Theta) = \sum_z P(D|Z,\Theta)P(Z|\Theta)$
- $P(D|Z,\Theta) = \prod_{i=1,...,N} P(x_i|z_i,\Theta)$

16.25 Expectation Maximization (EM)

- Initialize parameters Θ
- While progress:
  - E: Calculate $P(Z|D,\Theta)$
  - M: find maximizing parameters Θ' for

$$\sum_Z P(Z|D,\Theta)\ln P(D,Z|\Theta')$$

  - $\Theta \leftarrow \Theta'$

- $\alpha_0=\alpha_1=0.5$, $(\mu_0,\mu_1)=(10,100)$, $\sigma_0^2=\sigma_1^2=1000$.
- While progress:
  - for all k and i

E-step

$$P(z_i=k|x_i,\Theta)\propto\frac{\alpha_k}{\sigma_k\sqrt{2\pi}}e^{\frac{(x_i-\mu_k)^2}{-2\sigma_k^2}}$$

  - for all k:

M-step

$$\alpha_k\propto\sum_{i=1}^{N}P(z_i=k|x_i,\Theta)$$

$$\mu_k=\frac{1}{\alpha_k}\sum_{i=1}^{N}P(z_i=k|x_i,\Theta)x_i$$

$$\sigma_k^2=\frac{1}{\alpha_k}\sum_{i=1}^{N}P(z_i=k|x_i,\Theta)(x_i-\mu_k)^2$$

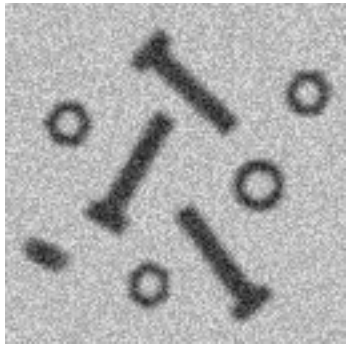
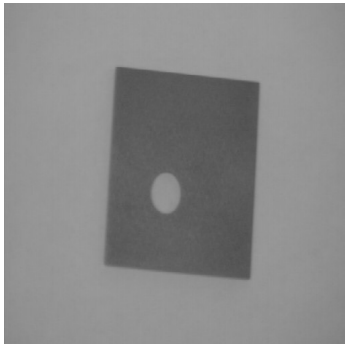
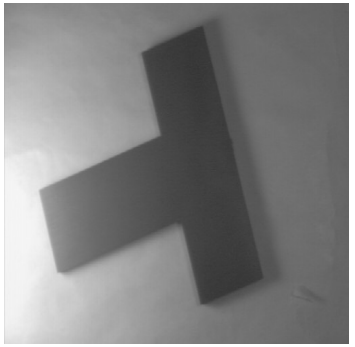
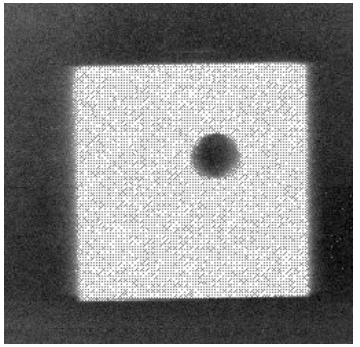
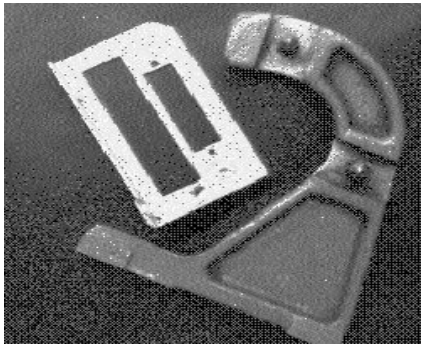
Disclaimer:
- No proof given here why this works, i.e. maximizes $P(D|\Theta)$, but each iteration improves it.
- EM may converge to a local maximum, not the global one.

16.26 Binary thresholding example

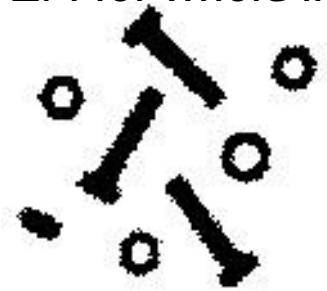
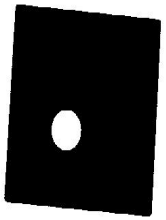
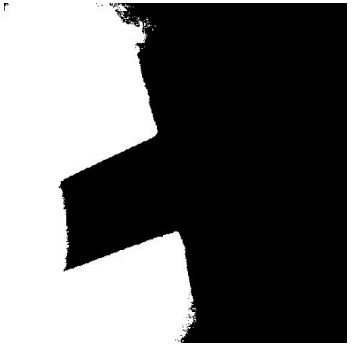
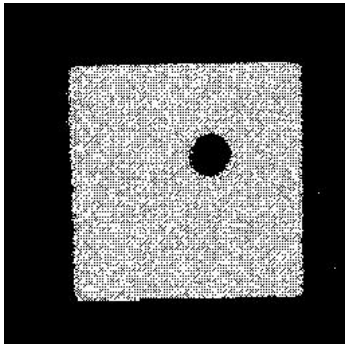
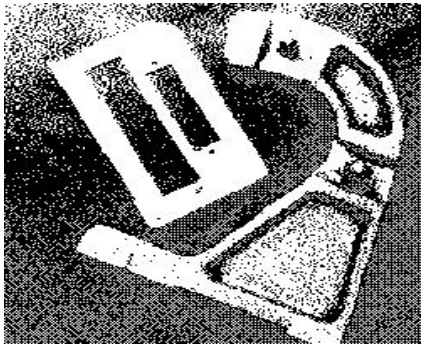
- Five gray-scale images with varying illumination.
- Four approaches are tried:
  - Run EM for the whole image to find the distribution of foreground and background pixels, and classify the pixel according to which distribution's mean its value is closer to.
  - Divide image to fixed number of sub-images and run EM for each of them independently.
  - For each pixel define a fixed neighborhood and run EM for that neighborhood.
  - Instead of running EM, calculate the mean gray-value in the pixel's neighborhood, and assign the pixel to foreground or background depending on whether its value is below or above the mean.

Results

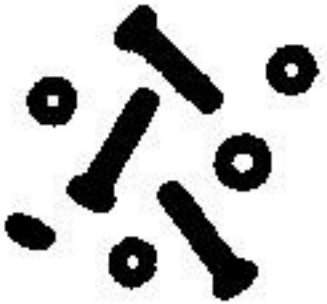
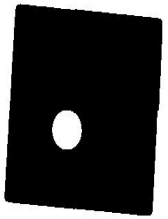
Original images

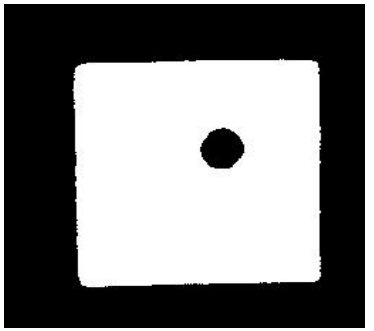
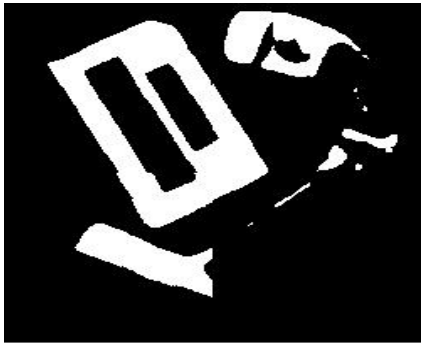






EM for whole images

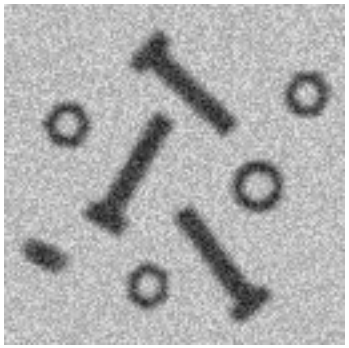
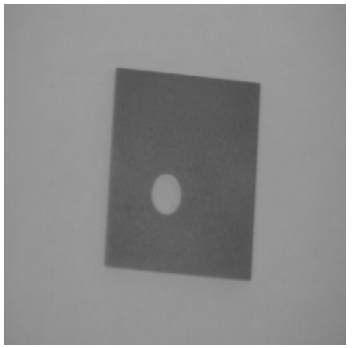
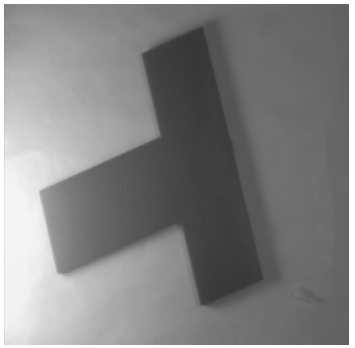
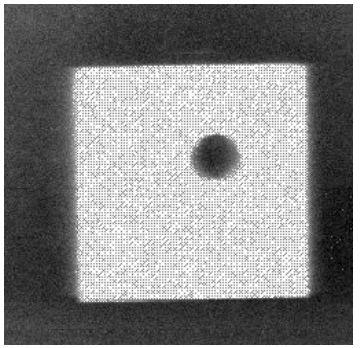
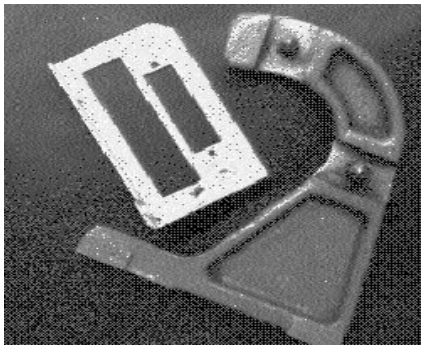






EM for 4 sub-images

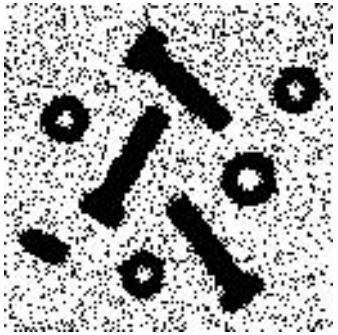
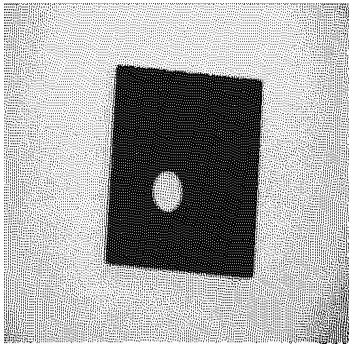
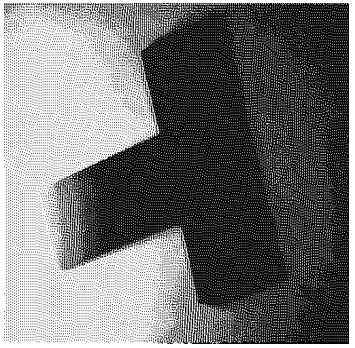
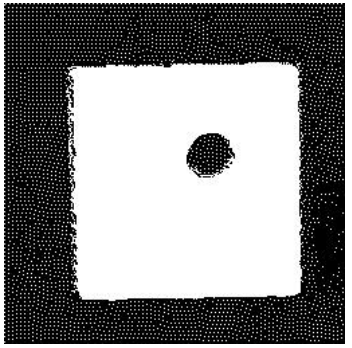
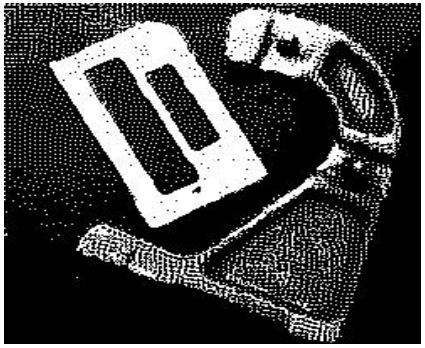



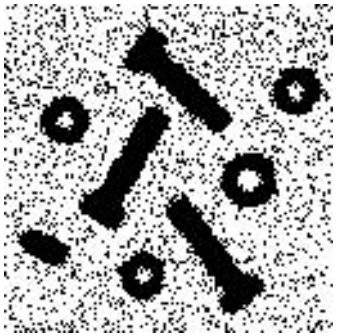
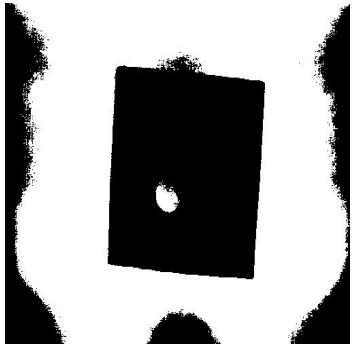
Original images

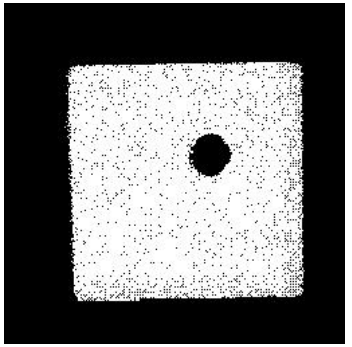
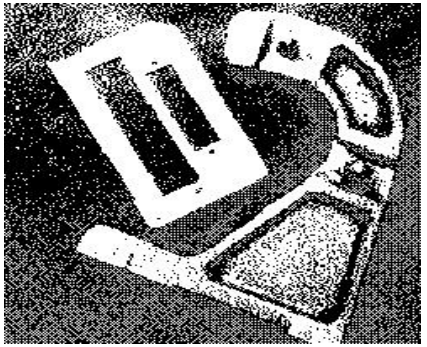






EM in each pixel's neighborhood

Comparison to mean gray-level in each pixel's neighborhood

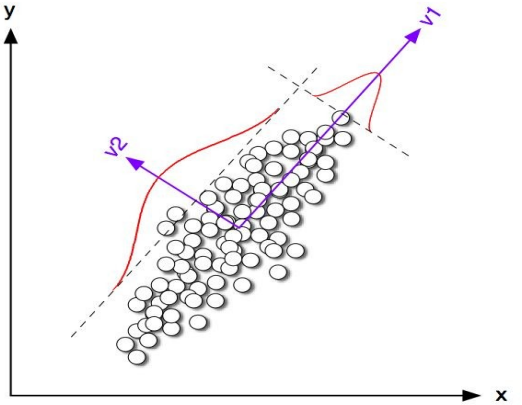



(T. Laine, 2001)

16.27 PCA in recognition

- Use Principal Component Analysis (PCA) to classify images → Reduce dimension by retaining those characteristics of the images that account most of the variance.
- Linear transformation that moves the data to a new coordinate system so that the highest variation is along the first dimension, the second highest along the second dimension, and so on.



- In image recognition and classification:

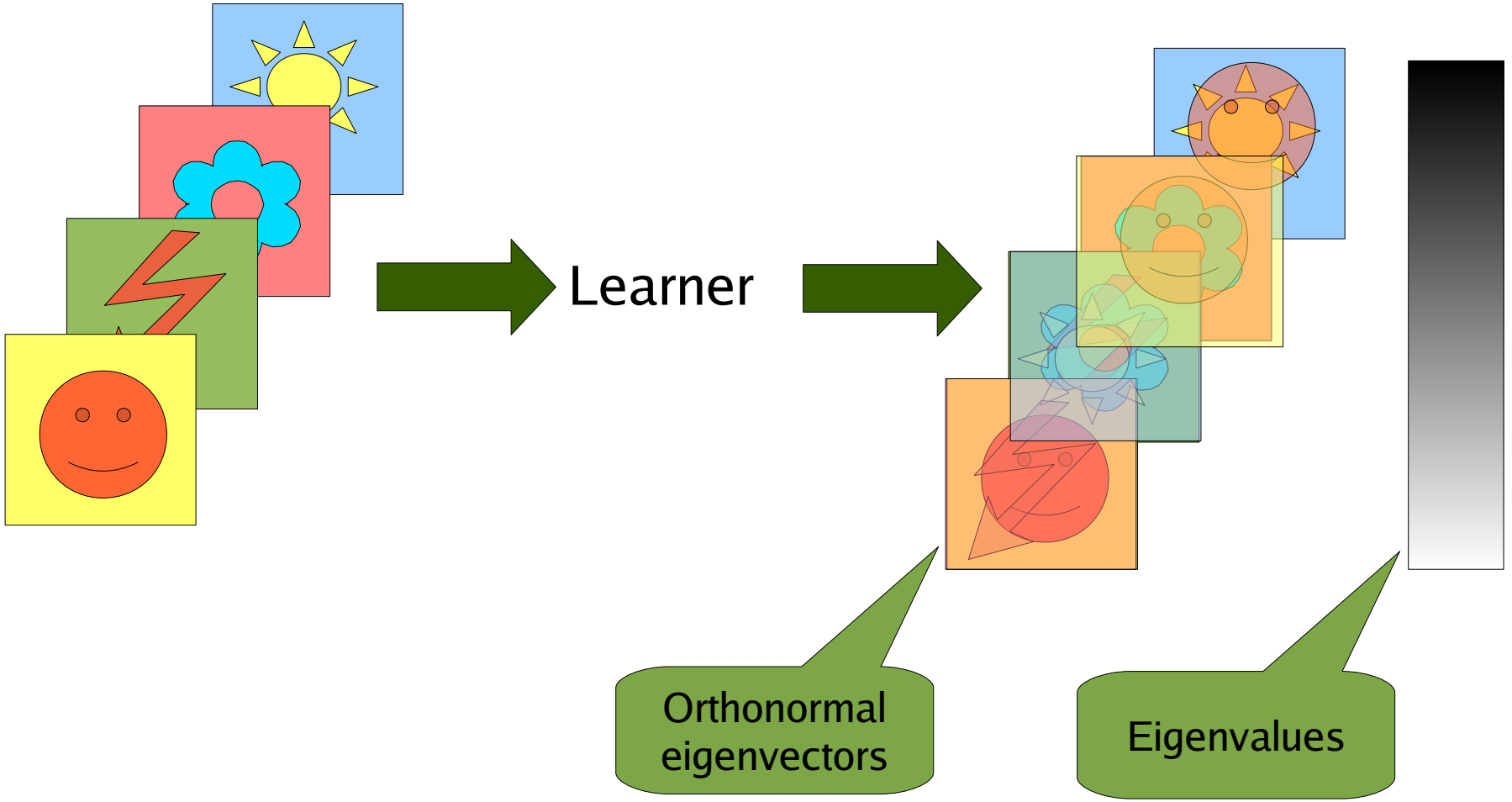
  Find relevant features in the images

  → Encode it efficiently

  → Compare the encoding of the image to the database of similarly encoded images.

- We want to find the *principle components* of the distribution of images, i.e., the *eigenvectors* and the *eigenvalues* of the covariance matrix of the set of images.

16.28 PCA idea



16.29 PCA training phase

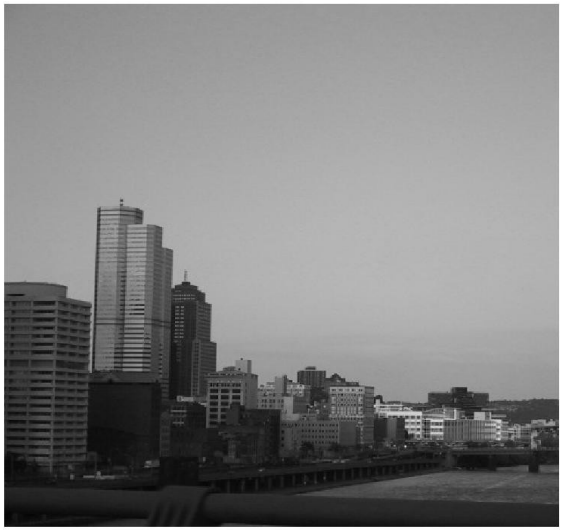
- Let the training set be images $\Gamma_1, \Gamma_2, ..., \Gamma_M$ of size $N \times N$.
- Average image is defined by $\Psi = 1/M \sum_{i=1,M} \Gamma_i$.
- Each image i differs from the mean by $\Phi_i = \Gamma_i - \Psi$.
- PCA looks for a set of orthonormal vectors u_k, and their associated eigenvalues λ_k that best describes the distribution of images.
- u_k and λ_k are the eigenvectors and eigenvalues, respectively of the covariance matrix:
  $C = 1/M \sum_{i=1,M} \Phi_i \Phi_i^T = AA^T$, where A = $[\Phi_1, \Phi_2, ..., \Phi_M]$, i.e.,
  $Cu_k = \lambda_k u_k$.
- $|C| = N^2 \times N^2 \rightarrow$ reduce the dimensionality to $M \times M$.

16.30 PCA recognition phase

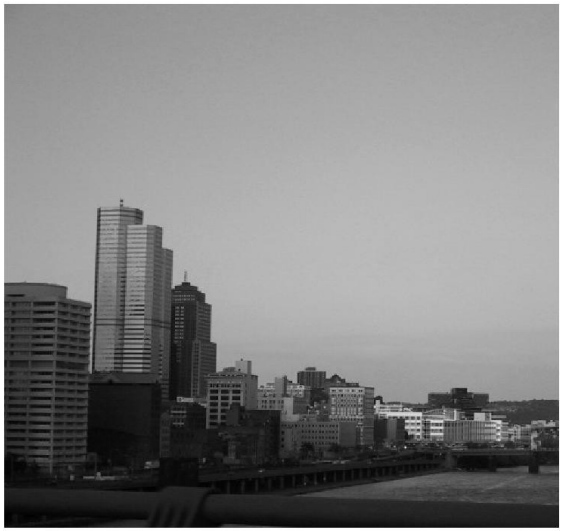
- Once the eigenvectors and values are calculated, identification is a pattern recognition task:
  - M' significant eigenvectors with the highest associated eigenvalues are chosen.
  - A new image is Γ projected into the eigen components by a simple operation:
    - $\omega_k = u_k^T (\Gamma - \Psi)$, for $k = 1, ..., M'$, where each weight ω_k represents the contribution of k^{th} eigenvector in recognition of the input image.

16.31 PCA example

Original image



Best eigenvectors for 16 sub-images



Reconstructed image

Instead of using a number of different images, divide the original one into sub-images, and find eigenvectors for them.

(T.Laine, 2001)

16.32 Face recognition

- Faces are complex, meaningful stimuli.
- Face recognition is a high-level visual task.
- Traditional approaches in recognition
  - Focus on detecting individual features, such the eyes, nose, mouth and the face outline, and defining a face by position, size and relation among these.
  - Difficult to extend to multiple views.
  - Insufficient to model human strategies in face identification.
  - Ignores the issue of which aspects of the face stimulus are important for identification.

- In recognition, significant features may not be related to visible face features, such as nose or eyes.
- Face recognition in PCA terms:
  - Extract relevant information in a face image
    $\rightarrow$ Encode it efficiently
    $\rightarrow$ Compare one face encoding to the database of faces encoded similarly.
  - Capture the variation in face images by finding the principal components of the distribution of faces $\rightarrow$ these are the eigenvectors of the 'face space', so called *eigenfaces*.
  - Each image can be represented as a linear combination of eigenfaces.
  - A new face image is projected to the space of eigenfaces, and classified by comparing its position to the positions of the existing faces.