Image Processing for feature extraction

Outline
- Rationale for image pre-processing
- Gray-scale transformations
- Local pre-processing (general notions)

Image (pre)processing for feature extraction
- Early vision: pixelwise operations; no high-level mechanisms of image analysis are involved
- Types of pre-processing:
  - enhancement (contrast enhancement for contour detection)
  - restoration (aim to suppress degradation using knowledge about its nature; i.e. relative motion of camera and object, wrong lens focus etc.)
  - compression (searching for ways to eliminate redundant information from images)

Image (pre)processing for feature extraction (cont’d)
- Pre-processing does not increase the image information content
- It is useful on a variety of situations where it helps to suppress information that is not relevant to the specific image processing or analysis task (i.e. background subtraction)
- The aim of pre-processing is to improve image data so that it suppresses undesired distortions and/or it enhances image features that are relevant for further processing

What are image features?
- Image features can refer to:
  - Global properties of an image:
    - i.e. average gray level, shape of intensity histogram etc.
  - Local properties of an image:
    - We can refer to some local features as image primitives: circles, lines, texels (elements composing a textured region)
    - Other local features: shape of contours etc.

Reading
Sonka, Hlavac, and Boyle. Chapter 5. Image preprocessing. (available on the course web site)
Some examples of image features (1)

- Global feature: shape of the histogram is useful in locating a threshold which will segment the image

Some examples of image features (2)

- Circumscribed (benign) lesions in digital mammography
- Spiculated lesions in (digital mammography)

The feature of interest: shape of contour; regularity of contour

- Can be described by Fourier coefficients
- We can build a feature vector for each contour containing its Fourier coefficients

Image features

- Are local, meaningful, detectable parts of an image:
  - Meaningful:
    - Features are associated to interesting scene elements in the image formation process
    - They should be invariant to some variations in the image formation process (i.e. invariance to viewpoint and illumination for images captured with digital cameras)
  - Detectable:
    - They can be located/detected from images via algorithms
    - They are described by a feature vector

Preprocessing

- Pixel brightness transformations
  - Also called gray scale transformations
  - Do not depend on the position of the pixel in the image
  - Operations on neighborhoods (filtering)

Gray-scale transformations work on histograms

- A histogram $H(r)$ counts how many times each quantized value occurs
- It is a 1D array
- $H(i)=$number of pixels in image having intensity level $i$
- Total area of image=total area under the histogram
- Can convert from the histogram (counting values) to probabilities (percentages) by just dividing by the area:
- This produces a probability density function.

$P(r) = \frac{H(r)}{\text{Area}}$

Histogram equalization

- Is a gray-scale transformation for contrast enhancement
- If we properly allocate our output levels proportional to the frequency of occurrence for our input levels, the histogram of the output should be uniform.
- The aim: create an image with equally distributed brightness levels over the whole brightness scale
Histogram equalization

- Ideal case
- Real case

Histogram equalization: how it works

- 1. Initialize the array $H = \text{zeros}(1, 256)$
- 2. Generate the histogram: scan each pixel and place it in the appropriate bin
- 3. Generate the cumulative histogram $H_c$ (which is the approximation of the discrete distribution function)
- Map input level $p$ into output level $T[p]$ using the formula
  
  $$T[p] = \text{round} \left( \frac{G-1}{NM} H_c(p) \right)$$

- Where does this formula come from?

Histogram equalization in Matlab

- `Imhist` for creating the histogram
- `Histeq` for equalizing it
- Examples

Adaptive histogram equalization

- Allows localized contrast enhancement:
  - Shadows
  - Background variations
  - Other situations where global enhancement wouldn’t work
- Remap based on local, not global histogram
- Example: $7 \times 7$ window around the point
- Problem: ringing artifacts

Histogram specification

- Histogram equalization: uniform output histogram
- We can instead make it whatever we want it to be
  - Comparing two images
  - “Stitching” multiple images
  - Image-compositing operations
- Use histogram equalization as an intermediate step

Histogram specification

- First, equalize the histogram of the input image:
  
  $$\tau(x) = \int \mu(x) \, dw$$

- Then histogram equalize the desired output histogram:
  
  $$\tau(x) = \int \mu(x) \, dw$$

- Histogram specification is
  
  $$T(x) = \tau(x) \cdot T(x)$$