Faculty of Informatics
Eötvös Loránd University
Budapest, Hungary

Basic Algorithms for Digital Image Analysis: a course

Dmitrij Csetverikov

with help of Attila Lerch, Judit Verestóy, Zoltán Megyesi, Zsolt Jankó and Levente Hajder

`http://visual.ipan.sztaki.hu`

Lecture 2: Intensity Transformations

Image enhancement by point processing

- Spatial domain and frequency domain methods
- Neighborhood operations and intensity transformations
- Gray-level histogram
- Some useful intensity transformations:
  - Contrast stretching and intensity normalisation
  - Image negation
  - Nonlinear compression of dynamic range
  - Intensity slicing
  - Histogram equalisation

Spatial domain and frequency domain methods

Goal of image enhancement:

- Image is composed of informative pattern modified by non-informative variations.
- **Enhance informative pattern** based on image data.
  - Examples: noise filtering, geometric correction.

Two types of image enhancement methods

- **Spatial domain methods**: Procedures that operate directly on image pixels.
- **Frequency domain methods** : Procedures that operate in a transformed domain.
  - Examples: power spectrum (Fourier), wavelets, Gabor transformation

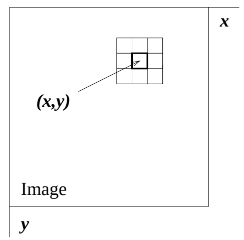
This course mainly deals with *spatial domain methods*.

Neighborhood operations

Output value in image point (x, y) is determined by pixels belonging to a neighborhood of (x, y):

$$g(x, y) = T[f(x, y)]$$

- $f(x, y)$: input image; $g(x, y)$: processed (output) image;
- T: operator on f, defined over some neighborhood of (x, y).



A 3×3 neighbourhood (window) about a point (x, y) in an image.

Intensity (grey-level) transformation, or mapping

- Neighborhood is of 1×1 size: reduces to point (x, y) itself (point processing).
- Output value depends only on intensity in (x, y).

For simplicity of notation: $r \doteq f(x, y)$, $s \doteq g(x, y)$.
Intensity transformation T maps r onto s:

$$s = T(r)$$

Basic properties of intensity transformations:

- Same intensities transform in the same way: **position independent**.
- Even local context neglected: **no structure** taken into account.
- Can only **reduce noise** when noise intensity is distinct; otherwise, can even amplify noise.

Use of intensity transformations:

- **Contrast stretching**: Increase contrast.
- **Intensity range normalisation**: Make intensities fall into specified range (usually, $[0, 255]$).
- **Background removal**: Remove irrelevant intensities.
- **Pattern enhancement**: Enhance relevant intensities.
- **Normalise images**
  - to compare images
  - to compare image descriptions

Gray-level histogram

Gray-level histogram $p(k)$ is occurrence probability (frequency) of grey level k in an image:

$$p(k) = \frac{n_k}{n}$$

where

- $k = 0, 1, \ldots, L-1$: grey level (L is number of grey levels);
- n_k: number of pixels with grey level k;
- n: total number of pixels.

Intensity histogram is **computed** in one scan of image $f(x, y)$

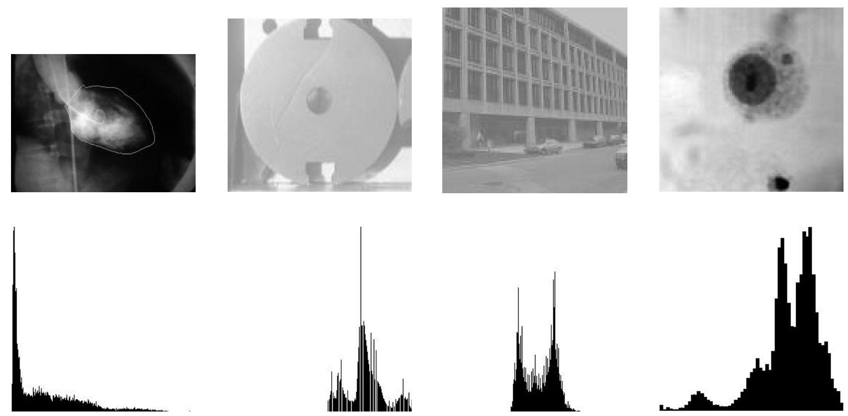
- Allocate array $p(k)$, set all $p(k) = 0$.
- Scan pixels (x, y). When $f(x, y) = k$, increment $p_{new}(k) = p_{old}(k) + 1$.

Histogram provides global description of appearance: **No structural information**

- Take structured image (shape or periodic pattern).
- Re-arrange pixels in random way.
- Obtain random image with **same histogram**.

Basic types of grey-level histograms:

- **Too dark**: grey levels concentrate at the dark end.
- **Too light**: grey levels concentrate at the light end.
- **Low contrast** (narrow dynamic range): grey levels concentrate in the middle.
- **Good contrast**: significant spread of histogram.



too dark | too light | low contrast | good contrast

Images and their histograms.

Some useful intensity transformations

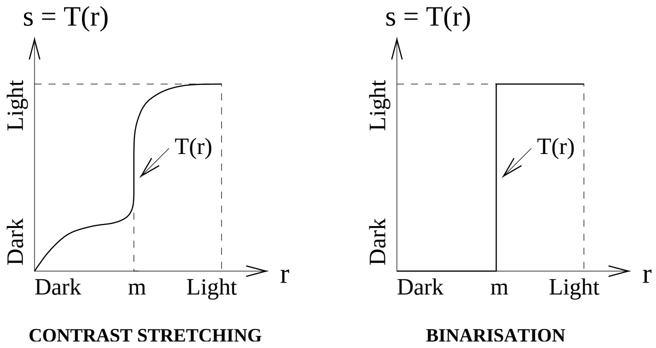
- Contrast stretching
- Intensity normalisation
- Image negation
- Nonlinear compression of dynamic range
- Intensity slicing
- Histogram equalisation

Contrast stretching

- **Purpose**: Increase dynamic range of intensities in low-contrast images.
- Why low contrast?
  - poor illumination
  - low dynamics of sensor
  - wrong setting of lens aperture

Basic properties of contrast stretching:

- Contrast stretching transformation function must be **single-valued** and **monotonically increasing**
  - Preserve order of grey levels: no artefacts
- The greater the **slope** the higher the **contrast** (spread) at that range.



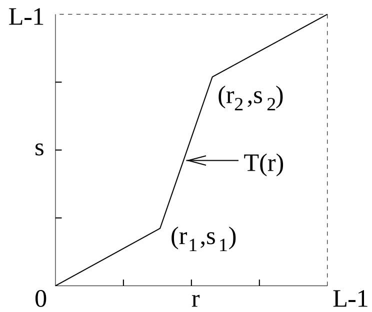
Gray-level transformation functions for contrast stretching and binarisation.

Binarisation (thresholding) is a special case of contrast stretching.

- Effect: Obtain binary image by setting to zero all intensities below a threshold and to maximum value all other intensities.
- Meaning: Separate object from background assuming 2 distinct intensity classes.

Piecewise-linear contrast stretching

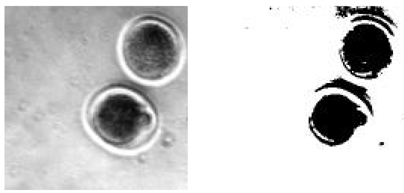
- Points (r_1, s_1) and (r_2, s_2) control shape of transformation function.
- **Binarisation** is special case of piecewise-linear contrast stretching: $r_1 = r_2$, $s_1 = 0$, $s_2 = L - 1$.



Piecewise-linear approximation of contrast stretching.

low contrast stretched

Example of contrast enhancement.



cell image binarised cell

Example of binarisation (thresholding).

Intensity normalisation (rescaling)

- **Purpose**: Normalise range of intensities to make their values fall into a standard range.
- **Reason**: Image processing operations may produce values that are beyond the initial range of intensities. It may be desirable to preserve the range and storage type (e.g., byte/pixel).

Solution: Let $r \in [r_1, r_2]$ be the original intensity and $s = T(r)$ a transformation that modifies the range $[r_1, r_2]$ to $[s_1, s_2]$, where

$$s_1 = \min_{r \in [r_1, r_2]} T(r)$$
$$s_2 = \max_{r \in [r_1, r_2]} T(r)$$

The output intensity s should be normalised so as to fall in the original range:

$$s' = T^{'}(s) \in [r_1, r_2]$$

This is achieved by the following linear transformation:

$$T^{'}(s) = \frac{r_2 - r_1}{s_2 - s_1} \cdot [T(s) - s_1] + r_1$$

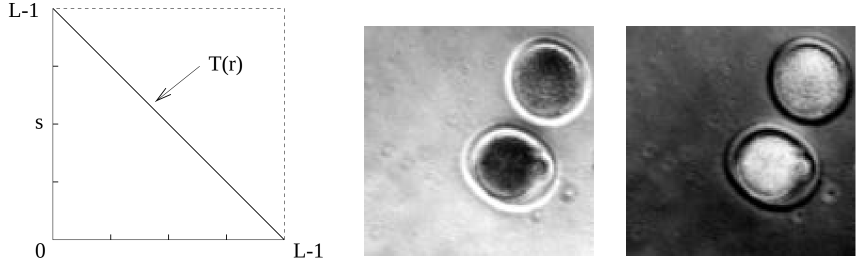
Before applying the transformation, check that $s_2 \neq s_1$.

Prove that $T^{'}(s) \in [r_1, r_2]$!

Image negation

- **Action**: Reverse the order of intensities.

- **Application**: Negate image of positive film and use the resulting negative as normal slide.



transformation image negative

Obtaining the negative of an image.

Compression of dynamic range

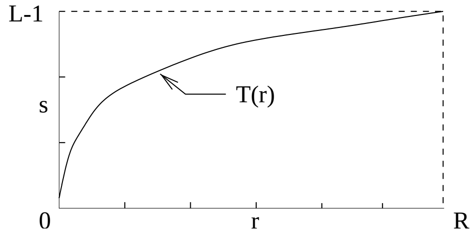
- **Action**: Compress dynamic range of image.

- **Application**: Dynamics of processed image may exceed the capabilities of display or film.
  - Only brightest parts visible, dark parts suppressed.
  - Typical for some medical imagery (e.g., x-ray) and Fourier spectra.

Solution:

$$s = c \cdot \log(1 + |r|)$$

where c is a scaling constant.



grey-level transformation function

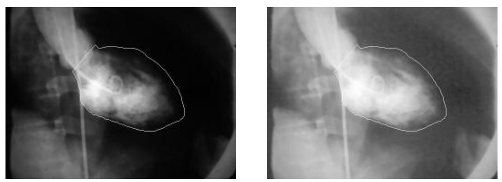


image transformation

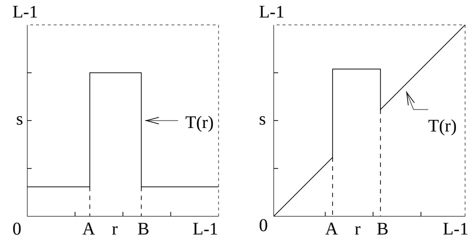
Compressing the dynamic range: Stretching the dark intensities of an image with dark intensities suppressed.

Gray-level slicing

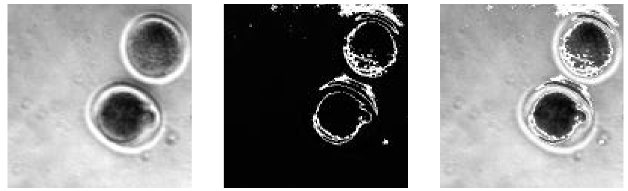
- **Action**: Highlight a specific range of intensities and/or suppress other intensities.
- **Application**: Background removal and segmentation.
  - Example: Highlight edge pixels when their intensities fall into a narrow range.

Two basic versions of slicing:

- Highlight a range, **diminish** other levels.
  - Thresholding is a particular case of this.
- Highlight a range, **preserve** other levels.



Transformation functions for intensity slicing. Left: Highlight range $[A, B]$ while suppressing other levels. Right: Same, but preserving other levels.



Examples of intensity slicing.

Histogram specification and equalisation

- **Histogram specification**: Transform intensities so as to obtain a specified shape of histogram of output image.
- Example: Humans perceive best the images that have **hyperbolic** histogram.

The most important case of histogram specification: **Histogram equalisation**, or histogram flattening.

- **Action**: The output image histogram becomes as **uniform** as possible.
- **Purposes**:
  - Increase dynamic range of image.
  - Normalise image histograms prior to comparison of
    - images
    - image descriptions

Histogram equalisation: continuous case

Gray-levels r are normalised continuous quantities: $r \in [0, 1]$.

Consider the transformation function

$$s = T(r) = \int_0^r p_r(w) \cdot dw \qquad 0 \leq r \leq 1$$

- $p_r(w)$: probability density function (PDF, distribution) of original intensity r.
- $0 \leq T(r) \leq 1$: cumulative distribution function (CDF) of r.
- $T(r)$: single-valued and monotonically increasing.

For this transformation function, the output intensity s has **uniform** distribution. (Prove!)

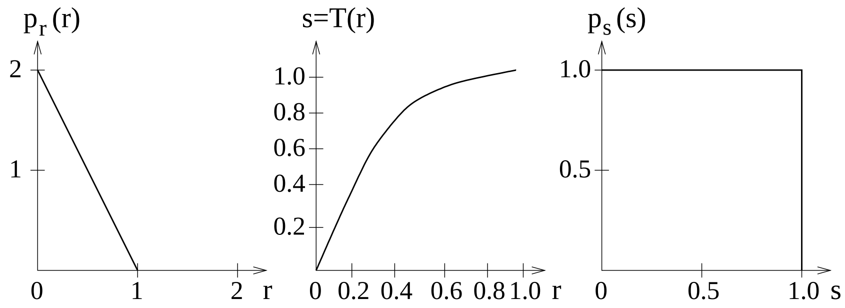


Illustration of histogram equalisation (flattening).
Left: Original probability density function.
Center: Transformation function.
Right: Resulting uniform density.

Histogram equalisation: discrete case

The transformation is obtained by summing up the bins of the **grey-level histogram**:

$$p_r(r_k) = \frac{n_k}{n} \qquad 0 \le r_k \le 1 \text{ and } k = 0, 1, ..., L-1$$

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} p_r(r_j)$$

$$r_k = T^{-1}(s_k) \qquad 0 \le s_k \le 1$$

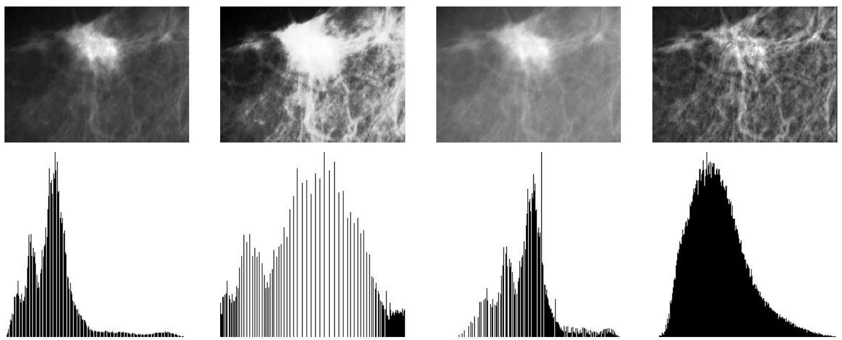


image | equlisation | stretching dark | adaptive equalis.

- Dynamic range increased, details better visible.
- Visual graininess and 'patchiness' increased because of too few grey levels.
- Noise amplified.
- Adaptive version is better.