University of Ljubljana, Faculty of Electrical Engineering
Robot Vision (RV)

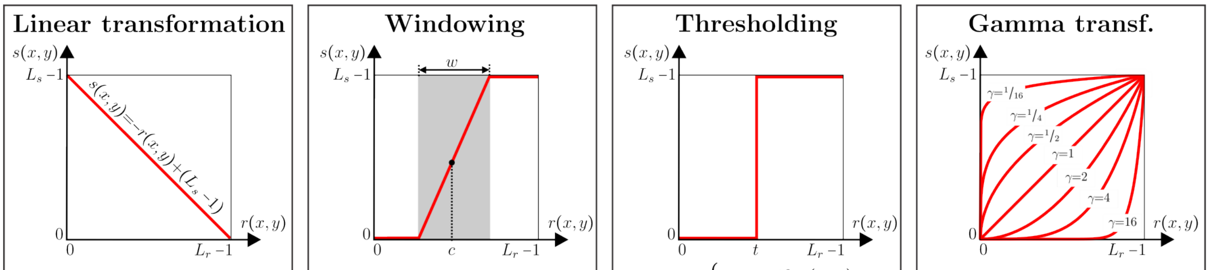
March 29/30, 2016

Exercise 3: Grayscale and color transforms

Created by: Žiga Špiclin | http://lit.fe.uni-lj.si/RV | Homework deadline: April 5/6, 2016

Instructions

Grayscale image transformations are arbitrary transformations of the form $\mathcal{T} : \mathbb{R} \to \mathbb{R}$ of the dynamic range $[0, L_r - 1]$ of the reference image $r(x,y)$ into dynamic range $[0, L_s - 1]$ of the transformed image $s(x,y)$. Transformation is performed on the values of all pixels in the reference image as $s(x,y) = \mathcal{T}(r(x,y))$. One of the main purposes of grayscale transformations is **contrast enhancement** of the structures of interest, while they are also used for **dynamic range adjustment** prior to image display. During this exercise you will get familiar with linear, windowing, thresholding and Gamma transformations.



$s(x,y) = a r(x,y) + b$ | $s(x,y) = \frac{L_s - 1}{w}\left[r(x,y) - (c - \frac{w}{2})\right]$ | $s(x,y) = \begin{cases} L_s - 1 & \text{if } r(x,y) \geq t \\ 0 & \text{if } r(x,y) < t \end{cases}$ | $s(x,y) = (L_s - 1)\left[\frac{r(x,y)}{L_r - 1}\right]^\gamma$

Color of a pixel is generally defined by three, and sometimes even with two, values or components. Incident light is transformed into digital form by three wavelength-sensitive sensors at about 700 nm (R), 550 nm (G) and 450 nm (B), thus the most common color space consists of three components RGB. There exist other color spaces that are more suitable for the analysis of digital color image, such as the HSV and $L^*a^*b^*$. In general, the RGB color image may be transformed into any other color space using (non)linear transformation of its three components.

Transformation from RGB to HSV

Precalculations:

$$
\begin{aligned}
&r = R/255,\ g = G/255,\ b = B/255 \\
&C_{max} = \max(r, g, b) \\
&C_{min} = \min(r, g, b) \\
&\Delta = C_{max} - C_{min}
\end{aligned}
$$

Hue $H \subset [0^\circ, 360^\circ]$:

$$
H = \begin{cases}
0^\circ \times \left(\frac{g-b}{\Delta} \bmod 6\right) & \text{if } \Delta \equiv 0 \\
60^\circ \times \left(\frac{g-b}{\Delta} \bmod 6\right) & \text{if } C_{max} \equiv r \\
60^\circ \times \left(\frac{b-r}{\Delta} + 2\right) & \text{if } C_{max} \equiv g \\
60^\circ \times \left(\frac{r-g}{\Delta} + 4\right) & \text{if } C_{max} \equiv b
\end{cases}
$$

Saturation $S \subset [0, 1]$:

$$
S = \begin{cases}
0 & \text{če } \Delta = 0 \\
\frac{\Delta}{C_{max}} & \text{sicer}
\end{cases}
$$

Brightness or *value* $V \subset [0, 1]$:

$$V = C_{max}$$

Transformation from HSV to RGB

Precalculations:

$$
\begin{aligned}
&C = V \times S \\
&X = C \times (1 - |H/60^\circ \bmod 2 - 1|) \\
&m = V - C
\end{aligned}
$$

Components $(r,\ g,\ b)$ are determined according to $H \subset [0^\circ, 360^\circ]$:

$$
(r,\ g,\ b) = \begin{cases}
(C,\ X,\ 0) & \text{if } 0^\circ \leq H < 60^\circ \\
(X,\ C,\ 0) & \text{if } 60^\circ \leq H < 120^\circ \\
(0,\ C,\ X) & \text{if } 120^\circ \leq H < 180^\circ \\
(0,\ X,\ C) & \text{if } 180^\circ \leq H < 240^\circ \\
(X,\ 0,\ C) & \text{if } 240^\circ \leq H < 300^\circ \\
(C,\ 0,\ X) & \text{if } 300^\circ \leq H < 360^\circ
\end{cases}
$$

Components to RGB are obtained by:

$$(R,\ G,\ B) = (r + m,\ g + m,\ b + m)$$

Output values RGB are in range $[0, 1]$.

During this exercise you will write various functions for transforming grayscale images and an RGB to HSV, and vice verse, color space transformation. Functions will be tested on the given color image.

1. Load the RGB color image `slika.jpg` into Spyder-Python environment using function `open()` in the library package `PIL.Image` and transform the image into unsigned 8-bit grayscale image according to $S = 0,299R + 0,587G + 0,114B$. Verify that the pixel values were correctly rounded and typecasted before saving the image.

2. Write a function for a linear transformation of a grayscale image `iImage`:

```
def scaleImage( iImage, iSlopeA, iIntersectionB ):
        return oImage
```

   where `iSlopeA` and `iIntesectionB` are parameters a and b of the linear transformation. Function should return a linearly transformed grayscale image `oImage`, which is of the same pixel-value type as the input image. Verify the function by applying an arbitrary linear transformation to the grayscale image.

3. Write a function for a windowing transformation of a grayscale image `iImage`:

```
def windowImage( iImage, iCenter, iWidth ):
        return oImage
```

   where `iCenter` and `iWidth` are parameters c and w of the windowing transformation. Function should return a windowed grayscale image `oImage`, which is of the same pixel-value type as the input image. The minimal and maximal values of the integer types can be obtained by `numpy.iinfo( dtype ).min` and `numpy.iinfo( dtype ).max`, respectively. Verify the function by applying an arbitrary windowing transformation to the grayscale image.

4. Write a function for thresholding transformation of a grayscale image `iImage`:

```
def thresholdImage( iImage, iThreshold ):
        return oImage
```

   where `iThreshold` is the threshold t. Function should return a thresholded grayscale image `oImage`. Verify the function by applying an arbitrary t to the grayscale image.

5. Write a function for Gamma transformation of a grayscale image `iImage`:

```
def gammaImage( iImage, iGamma ):
        return oImage
```

   where `iGamma` is the parameter γ. Function should return a Gamma-transformed grayscale image `oImage`, which is of the same pixel-value type as the input image. Verify the function on the grayscale image by using $\gamma > 1$ and $\gamma < 1$ and analyse its influence on the grayscale histogram of the image.

6. Write a function for transforming color image `iImage` between RGB and HSV color spaces:

```
def convertImageColorSpace( iImage, iConversionType ):
        return oImage
```

   where `iConversionType` is set either to `'RGBtoHSV'` or `'HSVtoRGB'` and determines the direction of the transformation from RGB to HSV or from HSV to RGB, respectively. Verify the function by transformtion the given RGB color image from RGB to HSV color space and, then, transform the obtained HSV image into an RGB image. The obtained RGB image should equal the given (input) RGB color image.

Homework Assignments

Homework report in the form of a Python script entitled `NameSurname_Exercise3.py` should execute the requested computations and function calls and display requested figures and/or graphs. It is your responsibility to load library packages and provide supporting scripts such that the script is fully functional and that your results are reproducible. The code should execute in a block-wise manner (e.g. `#%% Assignment 1`), one block per each assignment, while the answers to questions should be written in the corresponding block in the form of a comment (e.g. `# Answer: ...`).

1. Determine the value `iSlope` (a) and `iIntersection` (b) of the linear grayscale transformation such that the brightest point in the grayscale image will map into the darkest point and vice versa. Display the scaled image.

2. Determine the values of `iCenter` (c) and `iWidth` (w) of the grayscale windowing transformation such that 10% of the darkest and 10% of the brightest pixels will be saturated to the respective dark and bright values. Display the windowed image.

3. Determine the value `iThreshold` (t) of the threshold transformation such that exactly 50% of the brightest pixels in the input grayscale image will be above the threshold. Display the thresholded image.

4. Discuss the influence of choosing `iGamma` (γ) of the Gamma transformation onto the contrast of the output grayscale image (i) in the case of $\gamma < 1$ and (ii) in the case of $\gamma > 1$. Display an examples of Gamma-transformed images for both two cases.

5. Write a function for Gamma transformation of an RGB color image such that the RGB image is first transformed into HSV colorspace, then, Gamma transformation is applid to component $V$of the HSV color space. Finally, transform the HSV image into RGB color space. Similarly to previous assignment, display two example of Gamma-transformed images for $\gamma < 1$ and $\gamma > 1$. Does this approach to Gamma transformation have the same effect on the color image as in the case of a grayscale image?

Components of the color image displayed in RGB and HSV color space