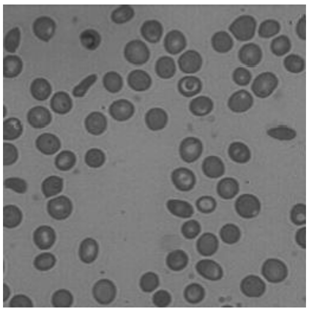
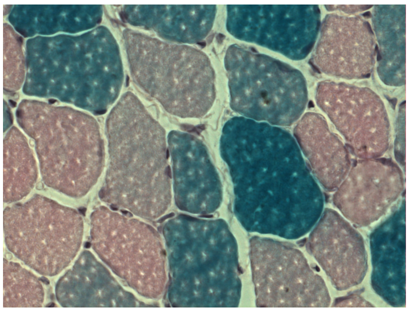
Tutorial: Histogram-based image segmentation

This tutorial aims to implement some image segmentation methods based on histograms (thresholding and "k-means" clustering).

The different processes will be applied on the following images:



(a) Cells.



(b) Muscle cells, author: Damien Freyssenet, University Jean Monnet, Saint-Etienne, France.

1 Manual thresholding

The most simple segmentation method is thresholding.

?

- Visualize the histogram of the grayscale image 'cells'.
- Make the segmentation with a threshold value determined from the image histogram.

2 k-means clustering

Let $X = \{x_i\}_{i\in[1;n],n\in\mathbb{N}}$ be a set of observations (the points) in $\mathbb{R}^d$. The k-means clustering consists in partitioning X in $k\,(k<n)$ disjoint subsets $\tilde{S}$ such that:

$$\tilde{S} = \arg\,\min_{S=\{S_i\}_{i\leq k}} \sum_{i=1}^{k} \sum_{x_j\in S_i} \|x_j - \mu_i\|^2 \qquad (1)$$

where μ_i is the mean value of the elements in S_i. The k-means algorithm is iterative. From a set of k initial elements $\{m_i^{(1)}\}_{i\in[1;k]}$ (randomly selected), the algorithm iterates the following (t) steps:

- Each element of X is associated to an element m_i according to a distance criterion (computation of a Voronoi partition):

$$S_i^{(t)} = \left\{ \mathbf{x}_j : \left\| \mathbf{x}_j - \mathbf{m}_i^{(t)} \right\| \leq \left\| \mathbf{x}_j - \mathbf{m}_{i^*}^{(t)} \right\|, \forall i^* \in [1;k] \right\} \tag{2}$$

- Computation of the new mean values for each class:

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j \tag{3}$$

  where $|S_i^{(t)}|$ is the number of elements of $S_i^{(t)}$.

3 Grayscale image, $k = 2$ in one dimension

The objective is to binarize image image 'cells', which is a grayscale image. The set X is defined by $X = \{I(p)\}$, for p being the pixels of the image I.

?

- Implement the algorithm proposed below (Alg. 1).
- Test this operator on the image 'cells'.
- Test another method of automatic thresholding (defined by Otsu in [1]).
- Compare the values of the thresholds (manual and automatic).

Data: Original image A
Data: Stop condition ε
Result: thresholded image
Initialize T_0, for example at $\frac{1}{2}(\max(A) + \min(A))$;
$done \leftarrow False$;
while *NOT done* **do**
  Segment the image A with the threshold value T;
  it generates two classes G_1 (intensities $\geq T$) and G_2 (intensities $< T$);
  Compute the mean values, denoted μ_1, μ_2, of the two classes G_1, G_2, respectively;
  Compute the new threshold value $T_i = \frac{1}{2}(\mu_1 + \mu_2)$;
  if $|T_i - T_{i-1}| < \varepsilon$ **then**
    $done \leftarrow True$
  end
end
Segment the image with the estimated threshold value.

Algorithm 1: K-means algorithm for automatic threshold computation of grayscale images.

The Matlab function `graythresh` computes the automatic treshold by the method of Otsu.

For use with python, use the module skimage. filter and the function threshold_otsu.

4 Simulation example, $k = 3$ in two dimensions

The objective is to generate a set of 2-D random points (within $k = 3$ distinct classes) and to apply the k-means clustering for separating the points and evaluating the method (the classes are known!).

?

Write a function for generating a set of n random points around a point with coordinates (x, y).

We will use the Matlab function randn.

We will use the python function randn from the module numpy.random.

?

- Use this function to generate 3 set of points (in a unique matrix) around the points $(0, 0)$, $(3, 4)$ and $(-5, -3)$.
- Use the builtin kmeans function for separating the points. The result is presented in Fig. 1.

Use the Matlab function kmeans. Verify the utility of the option replicate.

Use the python function sklearn.cluster.KMeans. Verify the utility of the option n_init=10.

5 Color image segmentation using K-means: $k = 3$ in 3D

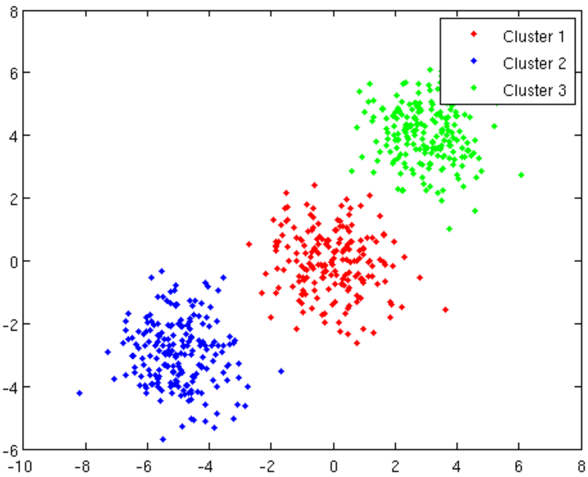
The k-means clustering is now used for segmenting the color image representing the muscle cells 'Tv16.png'.

?

Which points have to be separated? Transform the original image into a vector of size $N \times 3$ (where N is the number of pixels) which represent the 3 components R, G and B of each image pixel.

The MATLAB® function reshape can perform the transformation.

Figure 1: Resulting clustering of random points.



The python function numpy.reshape does the same transformation.

- Visualize the 3-D map (histogram) of all these color intensities.
- Make the clustering of this 3-D map by using the K-means method.
- Visualize the corresponding segmented image.

References

[1] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62–66, Jan 1979. 2