Assignment 24/04/2019

Graph cuts for interactive image segmentation

You will need a graph cuts library for this assignment: `pip install PyMaxflow`

Introduction

In this exercise you will learn to build a Markov Random Field (MRF) model for the binary image segmentation problem and to use a mincut solver in order to minimize the MRF energy, which will return the segmentation labels of the MRF problem. The binary segmentation problem of this exercise is to label each pixel of a gray scale image as belonging on the foreground or the background.

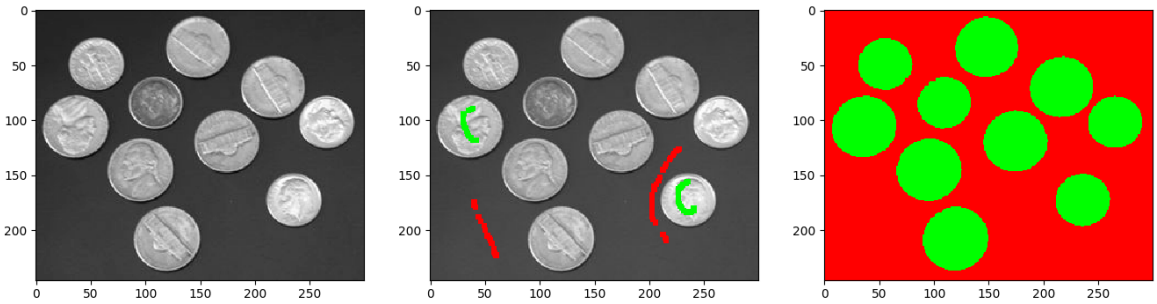

Figure 1: Input image, user input scribbles, and final segmentation.

The MRF energy of the segmentation is

$$E(Y|X) = \sum_{p} U_p(y_p|x_p) + \sum_{p \sim q} w_{pq} \cdot V_{pq}(y_p, y_q), \tag{1}$$

with $X = \{x_p\} \in [0, 255]^{width \times height}$ is the input image and $Y = \{y_p\} \in \{0, 1\}^{width \times height}$ is the segmentation result. In order to estimate the segmentation labels, we must minimize the energy $E(Y|X)$ with respect to Y:

$$\tilde{Y} = \arg\min_{Y} E(Y|X). \tag{2}$$

Your task is to properly define and implement the unary potentials $U_p(.)$, binary potentials $V_{pq}(.,.)$, and edge weigths w_{pq}.

From the pre-labeled input, histograms of red (background/label 0) and green (foreground, label 1) pixels are computed, P_0 and P_1. They are then normalized to behave as probability distributions. The unary potentials are:

$$U_p(y_p|x_p) = \begin{cases} -\log(k + P_0[x_p]), & \text{for } y_p = 0 \\ -\log(k + P_1[x_p]), & \text{for } y_p = 1 \end{cases} \tag{3}$$

with $k = 10^{-5}$. The pairwise potentials are defined according to the Potts model $V_{pq}(a, b) = \delta_{a \neq b}$.

Question 1

The function `binary_restore` of the code shows you the usage of the graph cut library. Explain what this function does, what energy is being minimized, and what the functions of the `Graph` class do.

Question 2

Explain the logics of (3).

Question 3

In the provided code, compute the probability distributions P_0 and P_1.

Question 4

The scribbler lets you define red and green areas with left and right clicks. Build the graph and assign weights. The pre-labeled pixels must keep their label at the end of the graph cut. For that, putting large enough weights to the appripate terminal nodes ensure that property. Take $w_{pq} = \lambda$ (a constant), and experiment with different values of λ, with the provided image `coins.png` and other images of your choice. Record the results.

Question 5

Change the weights w_{pq} so that the cuts along strong edges are not penalized much. Compare with fixed weights. You will need to use the powerful method `Graph.add_grid_edges` (see its documentation on the web).

Guidelines

Each student should submit a single zip file that will contain all the required code as well as a PDF document with the responses to the questions. Record also the input images, pre-labeled pixels, and segmentation results of your experiments. Include these files in the zip. For queries contact monasse@imagine.enpc.fr.