Practical session 3: Poisson image editing

Optimization Techniques, UPF

May, 2023

Deadline P101: Tuesday, May 16th (at 23:55)

Deadline P102: Friday, May 19th (at 23:55)

Grading: The evaluation is based on the report documenting your work (with figures), results, conclusions. For example,

- The goal of the lab.
- Summary with your own words of the topic.
- Conclusions for each exercise.

Nneka Okolo: `nnekamaureen.okolo@upf.edu`

1 An energy for image editing

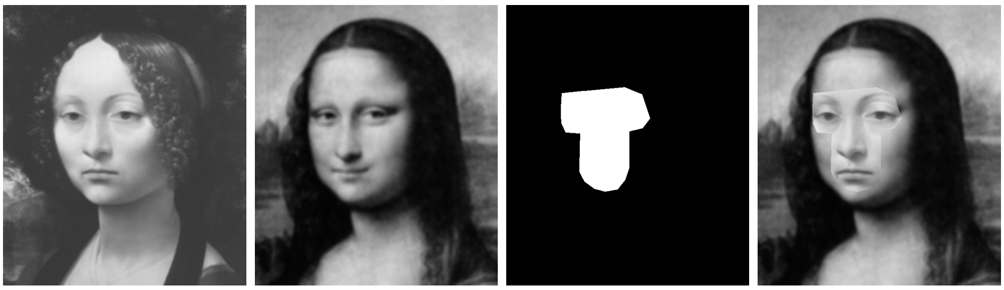

Figure 1: One problem in image editing. From left to right: u_1, portrait of Ginevra de' Benci; u_2, portrait of Lisa Gherardini; m face mask; result of composing images by editing the gray levels.

As an application of the conjugate gradient method for minimizing quadratic energies, we will present a method for image edition called Poisson editing. Poisson editing is based on editing the gradient of an image instead of working directly with the gray levels (or with the colors). As we will see, this simplifies many things for certain editing tasks.
We will focus on the following problem: we have two famous portraits by Leonardo Da

Vinci, and we would like to interchange the faces of Ginevra de' Benci and Lisa Gherardini. Both portraits have been aligned and scaled, so that the faces are the same size and are in the same position of the image. There is also a mask, which selects the part of the image we want to interchange.
Let $u_1, u_2 :\to \mathbb{R}$ and $m : \omega \to \{0, 1\}$ be both images, and the binary mask. Ω is a discrete rectangular 2D grid, the positions of the image pixels: $\Omega = 1, ..., M \times 1, ..., N$ (M columns and N rows). Say we want the face of u_1 in u_2. Let us call u the resulting image.

1.1 Editing directly the gray levels

The simplest thing to try is to define u as follows:

$$u_{ij} = m_{ij} u_{1,ij} + (1 - m_{ij}) u_{2,ij}, \text{ for } (i, j) \in \Omega.$$

The result is shown in Figure 1. Notice that the editing region is clearly visible. The reason is that one of the images is darker than the other, and this creates a discontinuity with the shape of the mask.

1.2 Editing the gradients

A better alternative is the following. First we compute the gradient of both images,

$$\nabla^+ u_{1,ij} = \begin{bmatrix} \nabla_i^+ u_{1,ij} \\ \nabla_j^+ u_{1,ij} \end{bmatrix} \text{ and } \nabla^+ u_{2,ij} = \begin{bmatrix} \nabla_i^+ u_{2,ij} \\ \nabla_j^+ u_{2,ij} \end{bmatrix}.$$

Here ∇_i^+ and ∇_j^+ refer to the forward differences partial derivatives in the direction of i (rows) and j (columns), as we defined them in the previous assignment. We define a new vector-valued image by composing these gradients as follows: when $m_{ij} = 0$ (non-face pixel) we use the gradient of u_2, and when $m_{ij} = 1$ (face pixel) we use the gradient of u_1. We store this in a new vector valued image v :

$$v_{ij} = m_{ij} \nabla^+ u_{1,ij} + (1 - m_{ij}) \nabla^+ u_{2,ij}, \qquad \text{for } (i, j) \in \Omega.$$

Now we look for an image u whose gradient coincides (or is similar) with v: $\nabla^+ u \approx v$. Also, the image should be u_2 outside the face (when $m_{ij} = 0$). We do that by solving a quadratic energy, very similar to the one we saw for image denoising!

The energy is the following:

$$E(u) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} |\nabla^+ u_{ij} - v_{ij}|^2_{\mathbb{R}^2} + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} \beta_{ij} (u_{ij} - u_{2,ij})^2,$$

where $\beta : \Omega \to \mathbb{R}$ is a coefficients image which controls the attachment to $u_{2,ij}$. We will use $\beta_{ij} = \beta_0 (1 - m_{ij})$, where $\beta_0 \in \mathbb{R}$ is a constant. This means that β_{ij} is zero on the face region, only acting outside the face. Recall that $|\cdot|_{\mathbb{R}^2}$ denotes the Euclidean in $\mathbb{R}^2$.

Similar to what we did for denoising, we can express the energy in matrix notation, using our vector representation of images and discrete gradients:

$$E(u) = \frac{1}{2} \langle \nabla^+ u - v, \nabla^+ u - v \rangle_{\mathcal{Y}} + \frac{1}{2} \langle B(u - u_2), u - u_2 \rangle_{\mathcal{X}}. \tag{1}$$

where B is a $MN \times MN$ diagonal matrix, which in its diagonal has the image β_{ij}. The rest of the notation is as in the previous assignment, $\mathcal{X} = \mathbb{R}^{MN}$ denotes the set of gray-scale images and $\mathcal{Y} = \mathbb{R}^{2MN}$ denotes the set of vector-valued images. $\langle \cdot, \cdot \rangle_{\mathcal{X}}$ is the scalar product in $\mathbb{R}^{NM}$ and $\langle \cdot, \cdot \rangle_{\mathcal{Y}}$ is the scalar product in $\mathbb{R}^{2NM}$. Finally,

Assignments

1. *Complete the code required in the cells by following the comments provided in the code*

2. *Once you have completed all the code above, now implemented into the function* `lisa_ginevra_test` *and explain with your own words the main steps of the algorithm*

3. *Run the Python script* `lisa_ginevra_test` *using different parameters and explain the differences.*

4. *Take a picture of two different faces (u_1 and u_2) and apply Poisson editing (it will be better if the faces are from the two components of the group). Try using u_1 as a background and the face of u_2 and the way around. Notice that you will need to create your own mask.*