

Assignment 1

Nalet Meinen
13-463-955

Image blending

1. Exercise 1

We can use a forward approach as it much simple to implement and it is sufficient for our needs. This derivation was discussed in the lecture.

$$\|\Delta u\|_2 \simeq \sum_{i,j} \sqrt{u[i+1,j] - u[i,j]^2 + (u[i,j+1] - u[i,j])^2}$$

2. Exercise 2

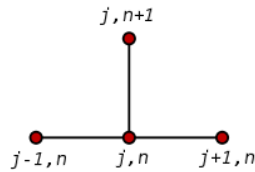


Fig. 1: Standard stencil as template

$$|u_C - g_C|_\Omega^2 = \sum_{i,j} \Omega[i,j] |u_C[i,j] - g_C[i,j]|_2^2$$
$$\frac{\partial}{\partial u[p,q]} |u_C - g_C|_\Omega^2 = \frac{\partial}{\partial u[p,q]} \sum_{i,j} \Omega[i,j] |u_C[i,j] - g_C[i,j]|_2^2$$

$$\frac{\partial}{\partial u[p, q]} \sum_{i, j} \Omega[i, j] \|u_C[i, j] - g_C[i, j]\|_2^2 = 2 \cdot \Omega[i, j] \cdot (u_C[i, j] - g_C[i, j])$$

$$\begin{aligned} \frac{\partial \|u_C - g_C\|_\Omega^2}{\partial u[i, j]} &= \Omega[i, j] \cdot (2 \cdot u_C[i, j] - u_C[i + 1, j] - u_C[i, j + 1] \\ &\quad - 2 \cdot g_C[i, j] + g_C[i - 1, j] + g_C[i, j - 1])^2 \end{aligned}$$

3. **Implementation.** For each of the 3 solvers (gradient descent, Linearization+Gauss-Seidel, Linearization+SOR):

For GD the results are quite good. However, for LGS and LSOR the numbers of iterations are a fixed value. A better result is achivable with a while loop in GD. In this case the running time for this notebook is already 40 minutes.

Hint: The numbers over the gradient images are the iterations for a image.

- **Gradient Descent**

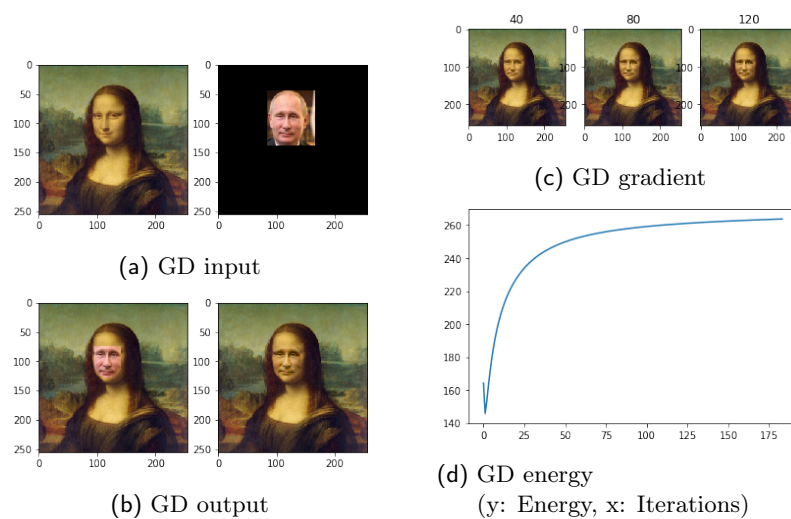


Fig. 2: Result Gradient Descent

- **Linearization+Gauss-Seidel**

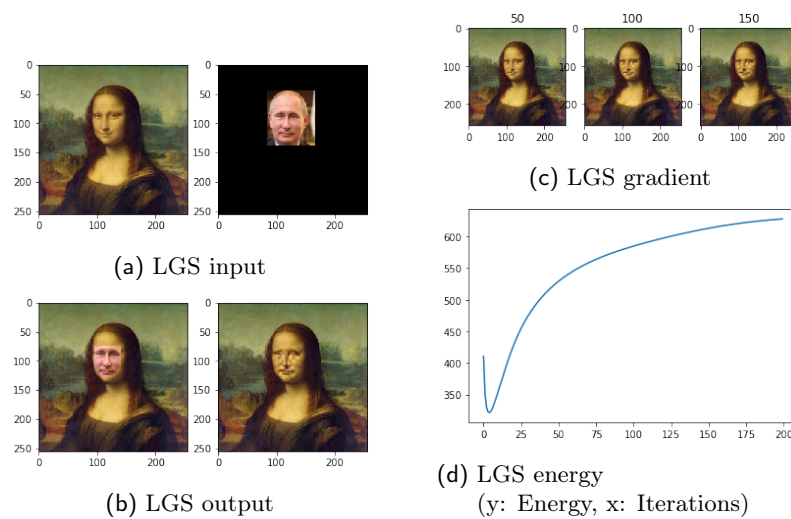


Fig. 3: Result Gradient Descent

• **Linearization+SOR**

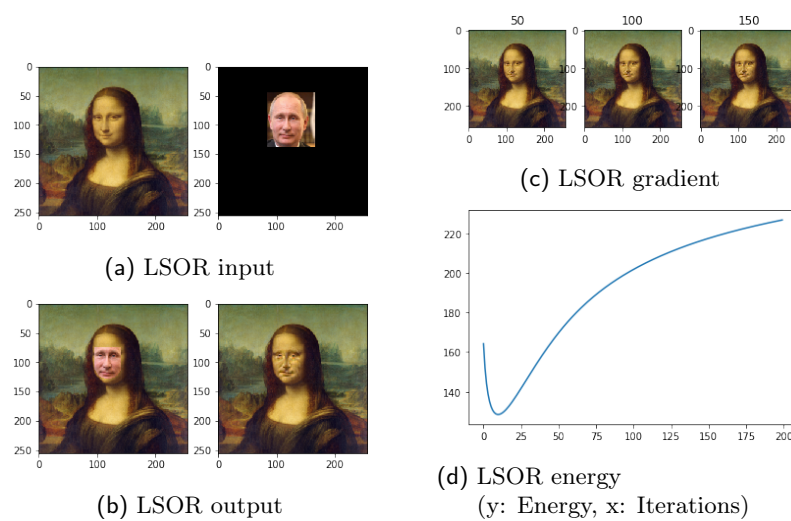


Fig. 4: Result Gradient Descent

4. State which of the 3 solvers you choose. Show images obtained by very high, very low and manually-tuned (approximately optimal) λ . In this section you should:

Very high lamda GD: 20, LGS: 50, LSOR: 20

• Gradient Descent

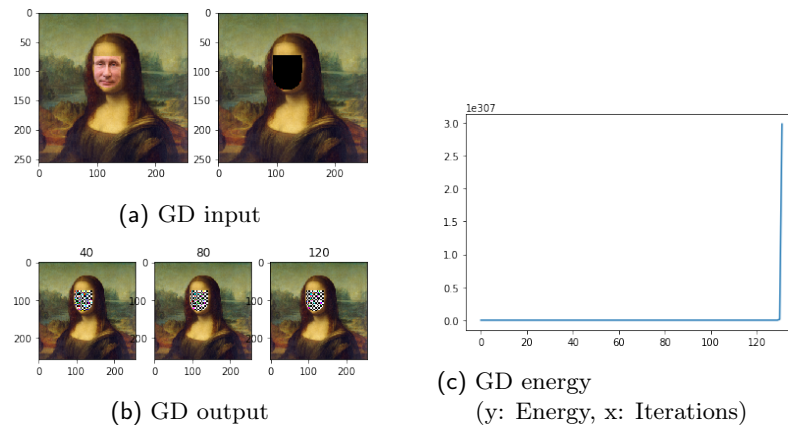


Fig. 5: Result Gradient Descent

• Linearization+Gauss-Seidel

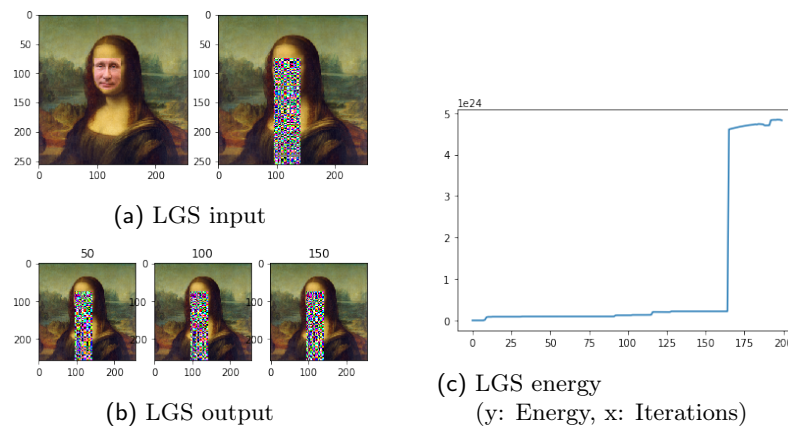


Fig. 6: Result Gradient Descent

• **Linearization+SOR**

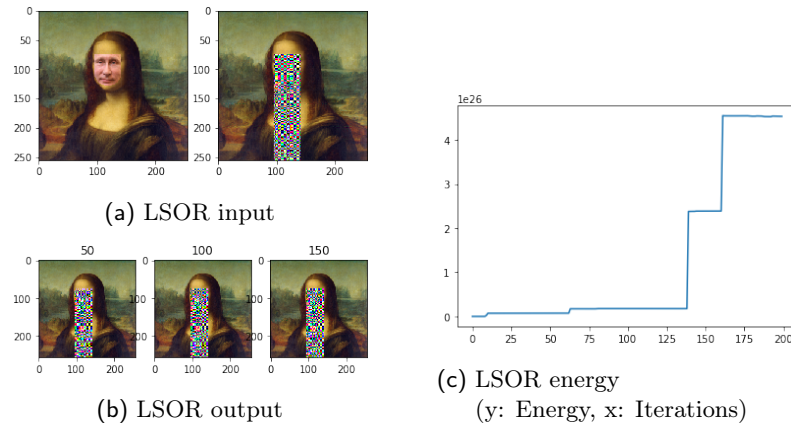


Fig. 7: Result Gradient Descent

Very low lamda GD: 2/100, LGS: 5/100, LSOR: 2/100

• **Gradient Descent**

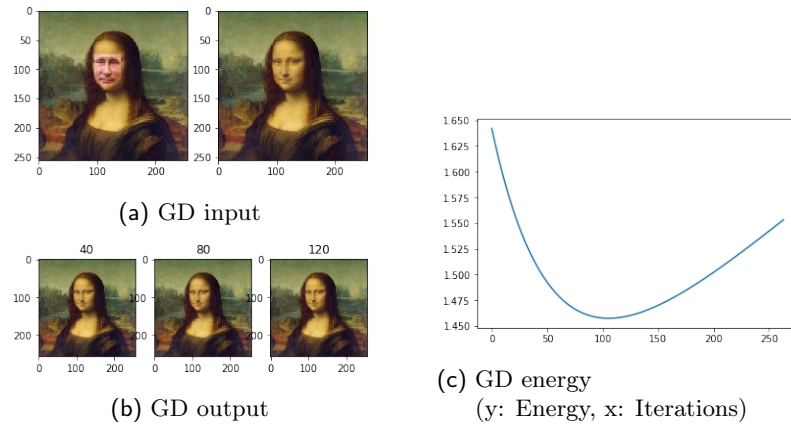


Fig. 8: Result Gradient Descent

- Linearization+Gauss-Seidel

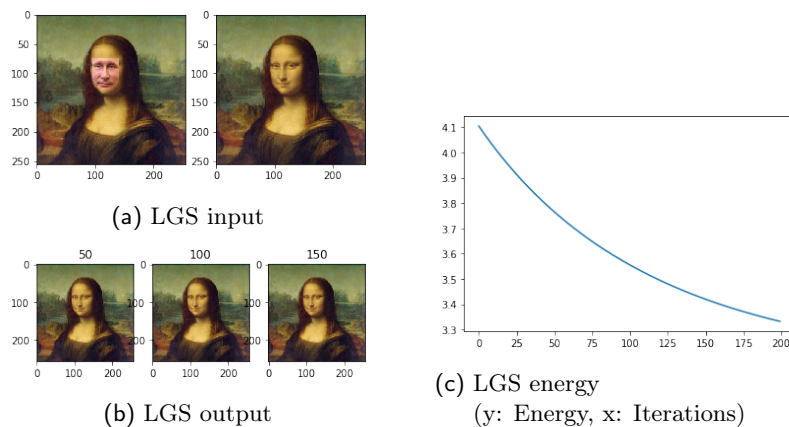


Fig. 9: Result Gradient Descent

- Linearization+SOR

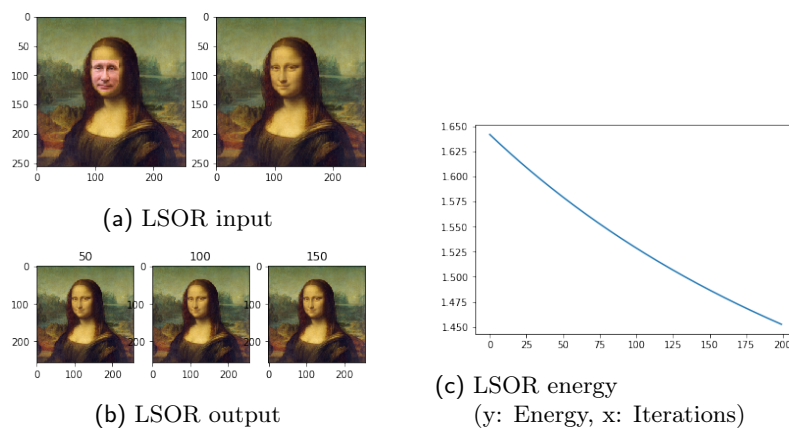


Fig. 10: Result Gradient Descent

- Describe the effect of λ on the solution.
 λ describes the change made to an image in each iteration step. Large λ will lead to a large step size, which the solver cannot process. Small λ would show no change at all at the images, as seen in the examples above.

5. Image blending:

- Display your own image composition here along with the foreground, background and mask images.
- **Gradient Descent**

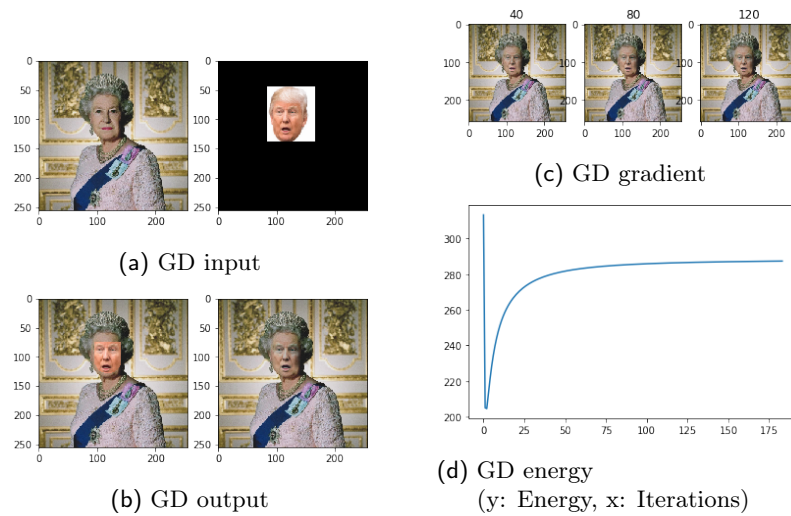


Fig. 11: Result Gradient Descent

- Describe how you used or modified the code to create your image(s). In this case the same images specs where used, so only the loading of the images needed to change.