Computational Photography Assignment 4

Single Michael
08-917-445

1 Poisson Solver

The following examples of *Seamless Cloning*, *Gradient Mixing* and *Highlight Removal* are based on the so called *Poisson equation* using a different gradient field.

In order to run a Matlab demo of these examples, please run the script *start4.m* from my submitted source code.

All the images (and also some more), can be found in the folder *outputs* in the zip file that contains my submitted Matlab code for this Project.

You can find the implementation of my Poisson solver in the Matlab file *poissonSolver.m*.

1.1 Seamless Cloning

The gradient field of a certain region (determined by a mask) in a given target image is supposed to be replaced by the (masked) gradient field from a source image. The task is to fit the source gradient field to the gradient field of the target image at the masked location in the image.

For this task I have implemented three examples shown below.

You can find the implementation of my Seamless Cloning driver in the Matlab file *seamlessCloning.m*.

1.1.1 Map Example

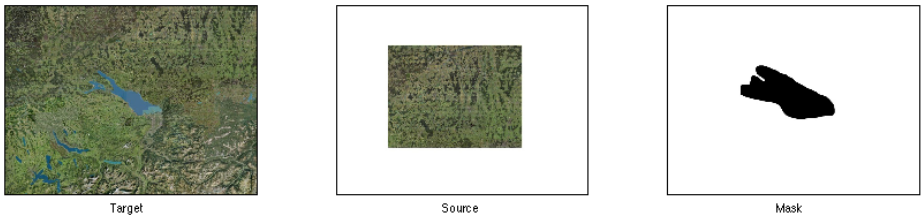

Fig. 1: Visualization of Input images: Target Image is a texture map (legt), the source image is a patch of another map (center), and the corresponding mask (right) is a sub-selection of the source patch-map.

Fig. 2: Visualization of Gradient field along dx and dy resulting from the gradient mixing gradient field applied on each color channel encoded as an RGB image.

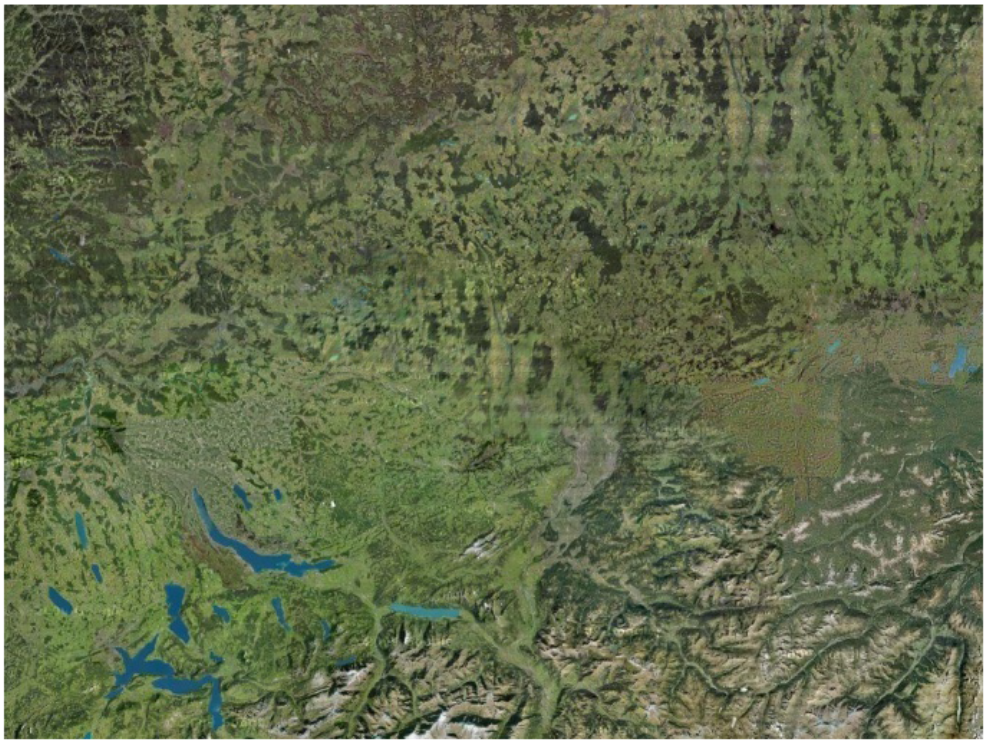

Fig. 3: Output resulting from the seamless cloning method applied on our given input images using the gradient field as shown above.

1.1.2 Plane Example

Fig. 4: Visualization of Input images: Target Image illustrates a landscape (left), is an image of a plane (center), and the corresponding mask (right) is a box capturing the whole plane image.

Fig. 5: Visualization of Gradient field along dx and dy resulting from the gradient mixing gradient field applied on each color channel encoded as RGB images.

Fig. 6: Output resulting from the seamless cloning method applied on our given input images using the gradient field as shown above.

1.1.3 Monster Example

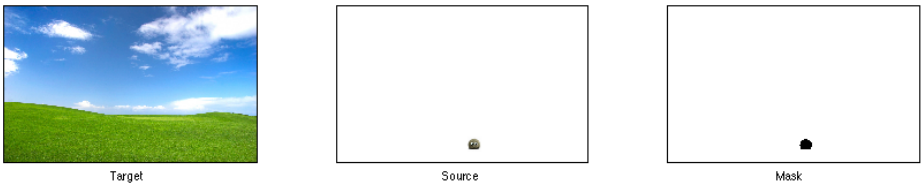

Fig. 7: Visualization of Input images: Target Image is again a hilly grass landscape (left), the source image is a hairy monster (center), and the corresponding mask (right) is the contour of the monster.

Fig. 8: Visualization of Gradient field along dx and dy resulting from the gradient mixing gradient field applied on each color channel encoded in RGB images.

Fig. 9: Output resulting from the seamless cloning method applied on our given input images using the gradient field as shown above.

1.2 Gradient Mixing

In this experiment we are interested in combining two images: One that is supposed to depict a foreground - usually a texture - and another image acting as the background image supposed to enhance some details in the given image. The method of gradient mixing will combine the gradient fields of the two images by selecting the gradient with the larger magnitude at each pixel. This will give us an output image having an additional detail layer following the gradient of the target image. Results are shown below.

You can find the implementation of my Seamless Cloning driver in the Matlab file *gradientMixing.m*.

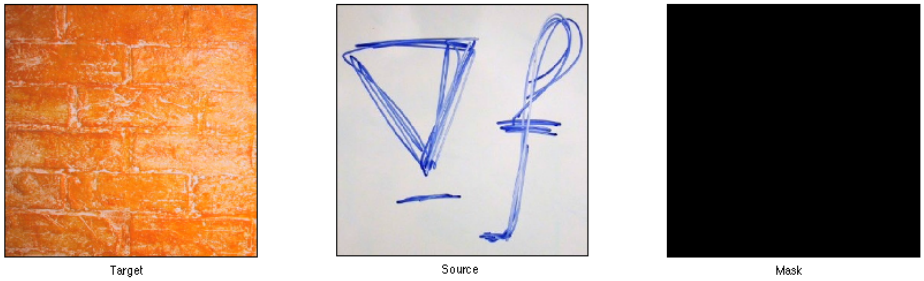

Fig. 10: Visualization of Input images: Target Image is a texture of a Wall (left), the source image is a mathematical scribble (center), and the corresponding mask (right) having a border of value one (known pixels) and everywhere else it is equal to zero.

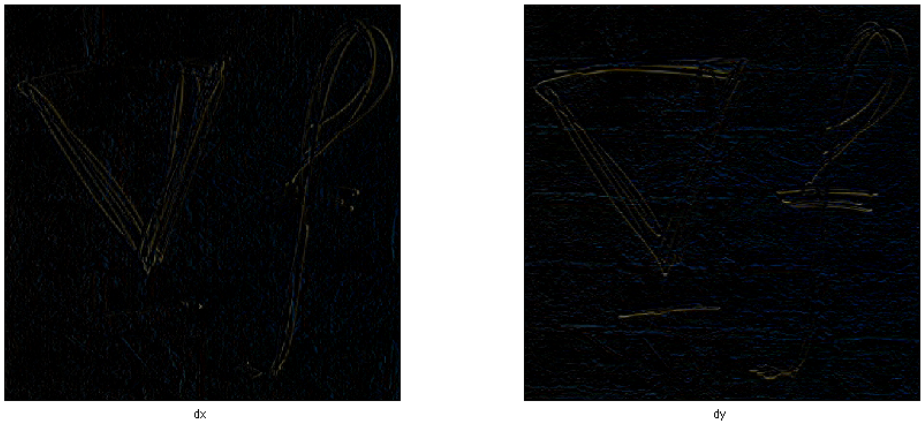

Fig. 11: Visualization of Gradient field along dx and dy resulting from the gradient mixing gradient field applied on each color channel encoded in RGB images.

Fig. 12: Output resulting from the gradient mixing method applied on our given input images using the gradient field as shown above.

1.3 Highlight Removal

In this experiment we want to adjust the local contrast of a given image by the method of highlight removal. For this purpose we mask the region in the target image that exhibits a way too bright specular spot. By solving the Poisson equation we are going to adjust the contrast within the masked region in the target image.

During this experiment I used two different gradient fields in order to reduce the local contrast:

- Alpha Compression: $v = f^{*\alpha}$: In This experiment I used an α value equal to 1.5. Please have a look at figure 15
- Gamma Compression: $v = \alpha^{\beta}|\nabla f^*|^{-beta}\nabla f^*$: In This experiment I used an α value equal to 0.005 and β equal to 0.4. Please see figure 16

Note that I use a different terminology: In the exercise sheet, we refer to my so called gamma compression, when talking about the *Alpha Compression*. However, during the lecture slides, we called this the gradient gamma compression. This is why I used this naming convention.

The results from figure 16, i.e. results from the gamma compression, are way more convincing compared to those produced using the alpha compression shown in figure 15. Simply speaking, the alpha compression (in the sense of the terminus I am relying on) produces renderings that are blurring out the textures within the masked region. Both methods, however, are capable to reduce the local contrast.

You can find the implementation of my Highlight Removal driver in the Matlab file *highlightRemoval.m*.

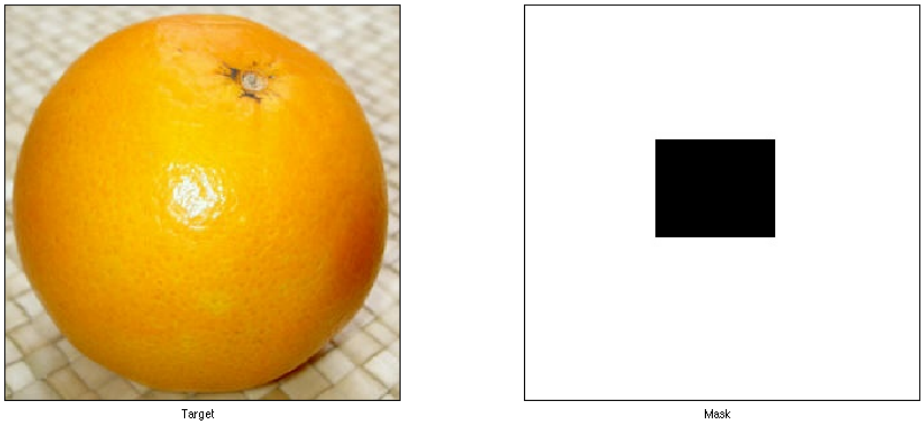

Fig. 13: Visualization of Input images: Target Image is an image of an orange exhibiting a specular spot (left) and the corresponding mask (right) that encloses the highlight

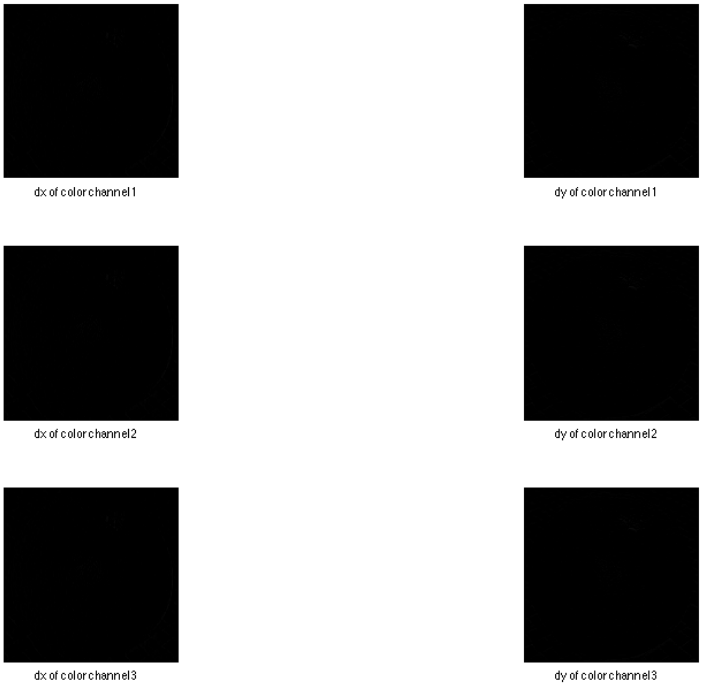

Fig. 14: Visualization of Gradient field along dx and dy resulting from the gradient mixing gradient field applied on each color channel separately.

Fig. 15: Output resulting from the gradient mixing method applied on our given input images using the gradient field as shown above.

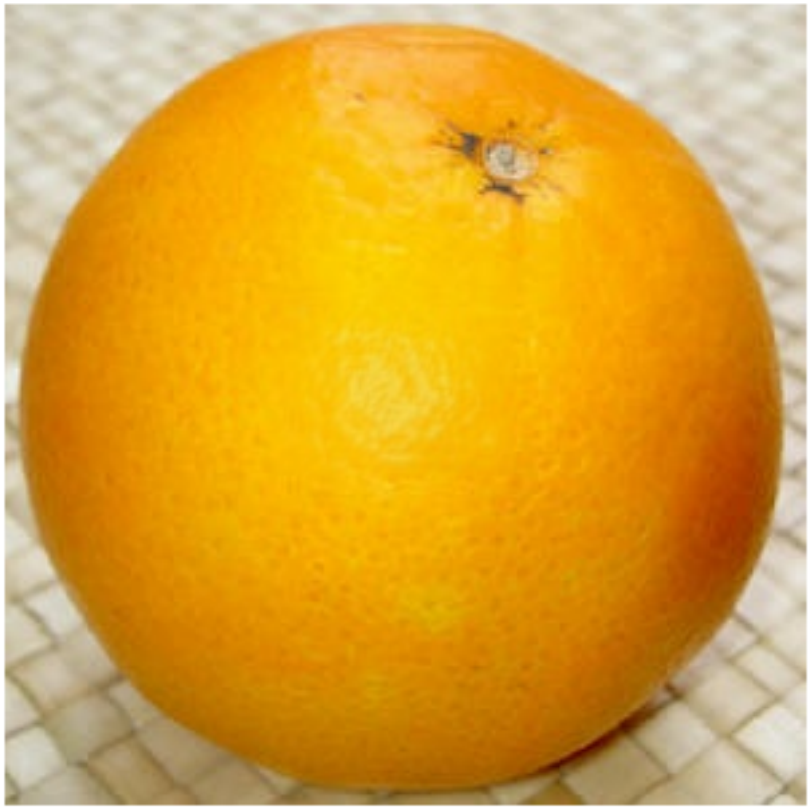

Fig. 16: Output resulting from the gradient mixing method applied on our given input images using the gradient field as shown above.

2 Image Segmentation

In this part of the project I have implemented an interactive image segmentation tool using graph cut opimization.

2.1 Sheep Example

Selection foreground sheep, background grass and some hay in the background gives us quite nice segmentation results.

In the following the results from some experiments I performed. For the smoothness therm I used a gamma parameter value equal to 20. In the *outputs* folder are additional examples using different values for gamma.

Note that a user can modify the value of gamma which is determing the smoothness penalty in the file *computeGraphSmoothness.m*.

You can start a Matlab demo of my image segmentation code by running the script *start4.m*. In order to run my image segmentation code on an arbitrary, user-specified image, you have to run the Matlab function *imageSegmentation.m*. This function expects being provided by an input image - this image is supposed to be segmented into a foreground-and a background image.

When running this function, a user has to specify foreground-and background masks. When running my function, a GUI will pop up, allowing a user to specify this masked regions. Please follow the instructions given by the Matlab console, when the GUI is running. Type *'f'* in order to select a foreground mask, type *'b'* to specify a background mask in the image or enter *'e'* to exit the mask selection process.

After the user has provided masking regions, the program will run by its own and computes the image segmentation. In addition it also will show some other helpful information: The user-specified masks, the probability images that a certain pixel belongs to the fore-or background, the mean colors of the fore-and background images, the smoothness term image and last, the segmentation of the image into a foreground and background image.

Fig. 17: Used Input (left) and Selection provided by user (right) whereas a green selection indicates foreground and a blue selection indicates the background.

Fig. 18: On the left the Foreground Mask and on the right the Background mask. For the foreground mask a white regions indicate that such a region should be interpreted as foreground. Similarly for the background mask.

Fig. 19: On the left the foreground mean colors and on the right the mean background colors.

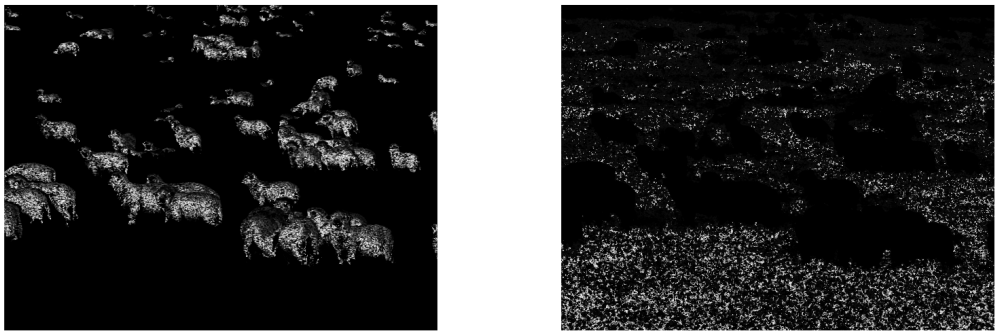

Fig. 20: On the left the probability a pixel belongs to the foreground and on the right the probability a pixel belongs to the background. The whiter the higher the probability is.

Fig. 21: Illustration of the smoothness term.

Fig. 22: The segmentation of the image: On the left the Foreground image and on the right the background image

2.2 Zebra Example

Selection foreground sheep, background grass and some hay in the background gives us quite nice segmentation results.

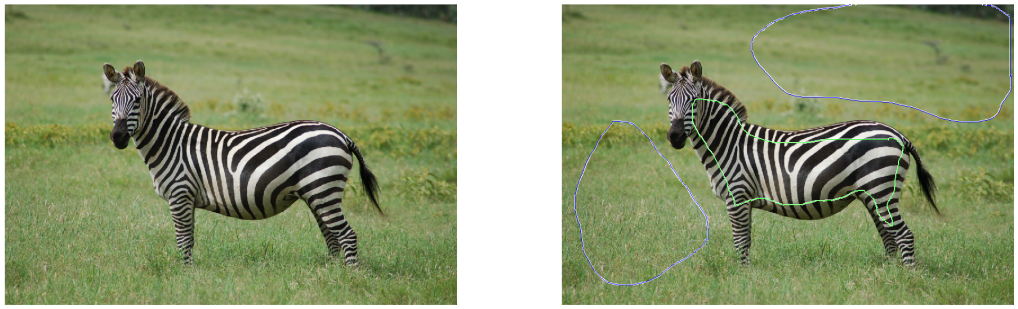

Fig. 23: Used Input (left) and Selection provided by user (right) whereas a green selection indicates foreground and a blue selection indicates the background.

Fig. 24: On the left the Foreground Mask and on the right the Background mask. For the foreground mask a white regions indicate that such a region should be interpreted as foreground. Similarly for the background mask.

Fig. 25: On the left the foreground mean colors and on the right the mean background colors.

Fig. 26: On the left the probability a pixel belongs to the foreground and on the right the probability a pixel belongs to the background. The whiter the higher the probability is.

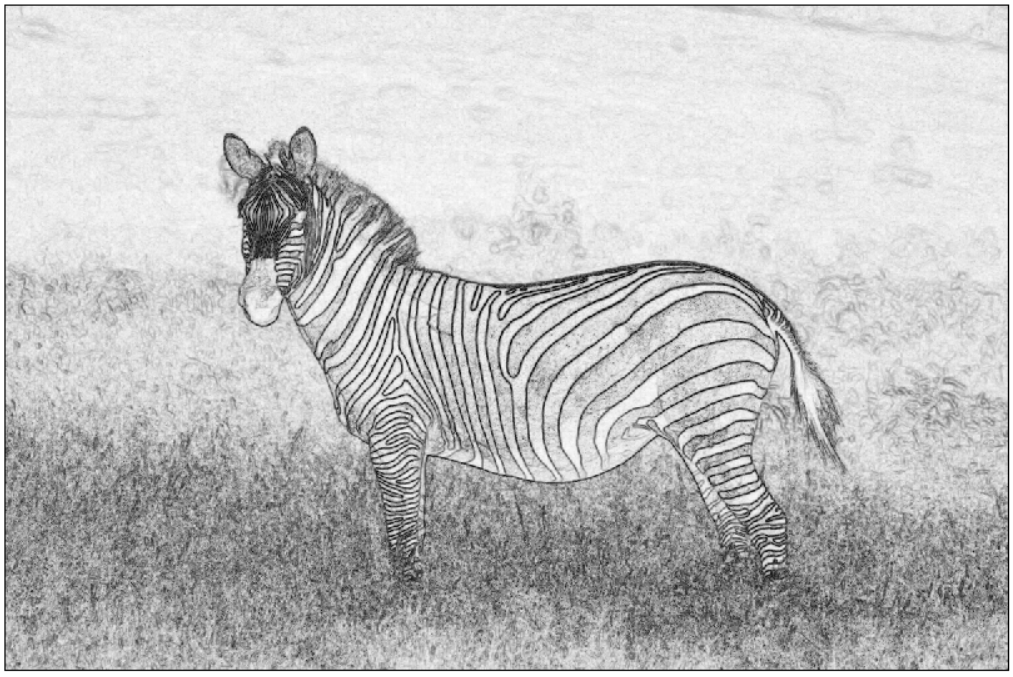

Fig. 27: Illustration of the smoothness term.

Fig. 28: The segmentation of the image: On the left the Foreground image and on the right the background image

2.3 Second Zebra Example

Selection foreground sheep, background grass and some hay in the background gives us quite nice segmentation results.

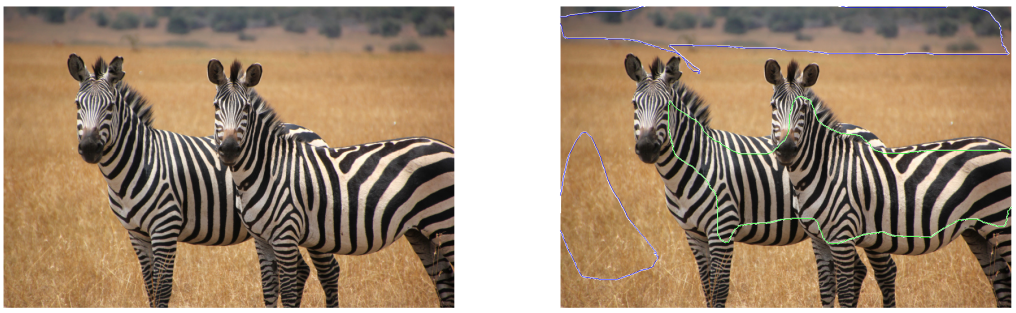

Fig. 29: Used Input (left) and Selection provided by user (right) whereas a green selection indicates foreground and a blue selection indicates the background.

Fig. 30: On the left the Foreground Mask and on the right the Background mask. For the foreground mask a white regions indicate that such a region should be interpreted as foreground. Similarly for the background mask.

Fig. 31: On the left the foreground mean colors and on the right the mean background colors.

Fig. 32: On the left the probability a pixel belongs to the foreground and on the right the probability a pixel belongs to the background. The whiter the higher the probability is.

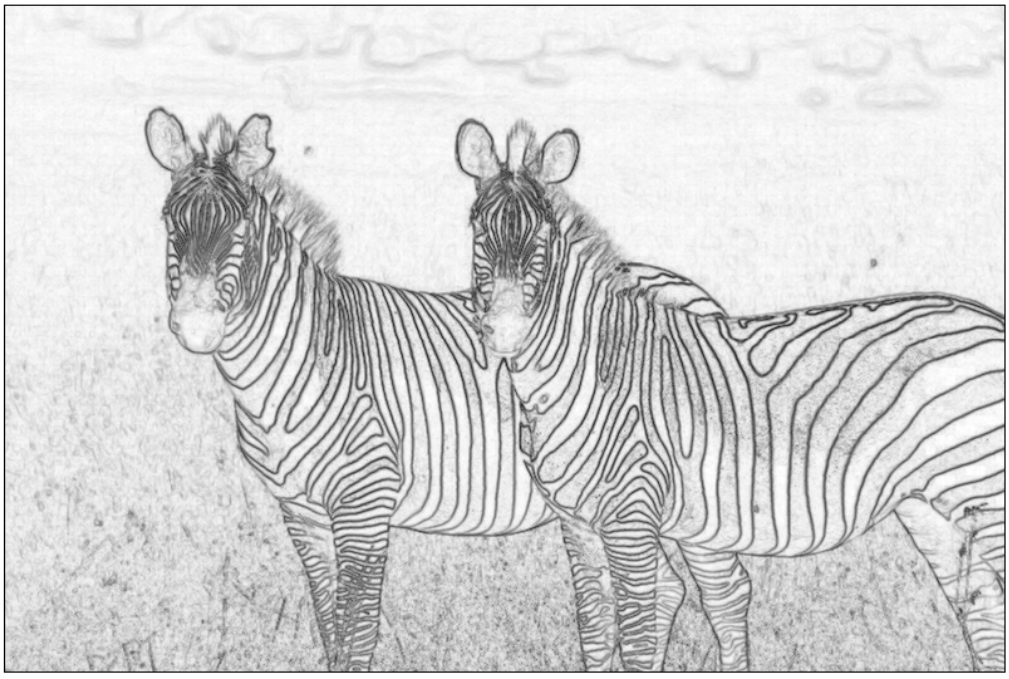

Fig. 33: Illustration of the smoothness term.

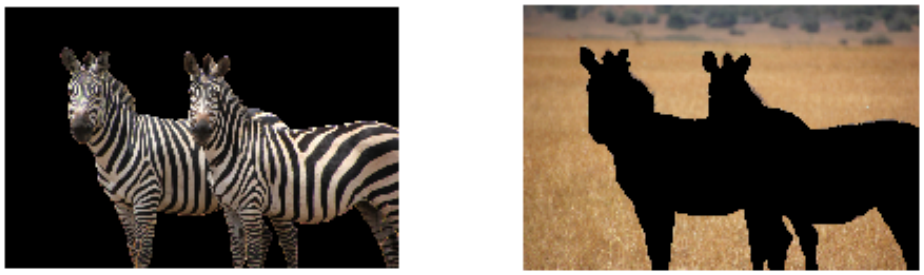

Fig. 34: The segmentation of the image: On the left the Foreground image and on the right the background image