MO446 – Introduction to Computer Vision
Project 1

Breno Leite
Guilherme Leite

28/08/2017

Question 1 - Input images

Throughout this project some images were used as input to test the algorithms, Figure **??** was used as input to the pyramids in exercises **2.1, 2.2, 2.3** and **3.1**, it's dimensions are 400x300, the blending exercises **2.4** and **3.2** used Figures **??**, **??** and **??**, all of them with dimensions 540x392, as input. The project was implemented using Python3.

Question 2 - Spatial blending

2.1) In the spectrum of image processing, convolution can be understood as the action of applying a mask to an image, in this case every pixel in the image is affected by the weighted sum of the filter applied to the pixel neighborhood. This approach requires some special attention when dealing with the edges of the image, since the application of the mask will overshoot the image borders, our solution to this problem was to extend the original image filling it's new borders with zeros, as seen in Figure **??**, by doing so the convolution simply ignore those values and don't account them into the result, a minor downside to this approach is the lessen effect of the mask around the edges, this solution was chosen for it's simplicity to implement. Additionally the convolution was tested with three smoothing masks, 3x3, 7x7 and 15x15, in comparison to the embedded solution by OpenCV our convolution was noticeably slower as seen in Table 1. As expected the masks smooth the edges around the image and suffers with a darkening close to the borders as seeing in Figure **??**, with bigger masks the image suffered more distortion to the point of loosing all of its fine details, see Figure **??**, the loss of such details is accounted by the size of the neighborhood that affect each pixel, thus a larger mask smooths more than a smaller one.

Important note: *The borders seen in the figures are not part of the image, they are figurative information about the starting and ending points of the image. Moreover, all the image scales in this report were changed in order to make the text more readable.*

Table 1: Convolution time comparison (ms)

	3x3	7x7	15x15
Convolution time	5.1238	4.86811	5.2383
OpenCV Conv. time:	0.0066	0.0048	0.0192

2.2) To store the gaussian and laplacian pyramids of this project we chose a list, in which each node of it holds a level of the pyramid. Since the data structure is a list the access method is as in a array, $pyramid[i]$ returns the image in the ith pyramid level, disregarding then, the necessity to implement an access function. The strategy chosen to implement the interpolation was the bilinear interpolation. These decisions were taken regarding the time to implement and simplicity to understand. It is also worth noting that due to some confusion about the meaning of Up and Down, these functions were renamed as follow: Expand referring to the action of expand the image width and height, and Contract analogously. Figure **??** shows the pyramid formed with the implemented function.

2.3) The laplacian pyramid demanded more attention to it's details, such as the fact that the last level of the pyramid is a copy of the same level at the gaussian pyramid. This particular level is used to perform the reconstruction of the original image, first the image is extended and interpolated, to match the previous level size, then it is summed with the previous level of the pyramid, the repetition of these steps result in the original image that was compressed in the pyramid, Figure **??** shows the laplacian pyramid.

2.4) Blending

Question 3 - Frequency blending

3.1) Blending

3.2) Blending