Laplacian Pyramids and Image Blending

Alan Zhuolun Zhao, Pepper Shiqin Wang

2/26/17

ENGR027 Project #2

Professor Zucker

ENGR027 Computer Vision

Procedure:

The program has 3 modes: pyramid, blend and hybrid.

If the user chooses pyramid mode, the program takes the input image and builds up the pyramid, and then attempts to reconstruct the original image from the pyramid. The user can compare the original image with the reconstructed image. The output is shown in Figure 1, 2, 3 below.

Figure 1. Original image

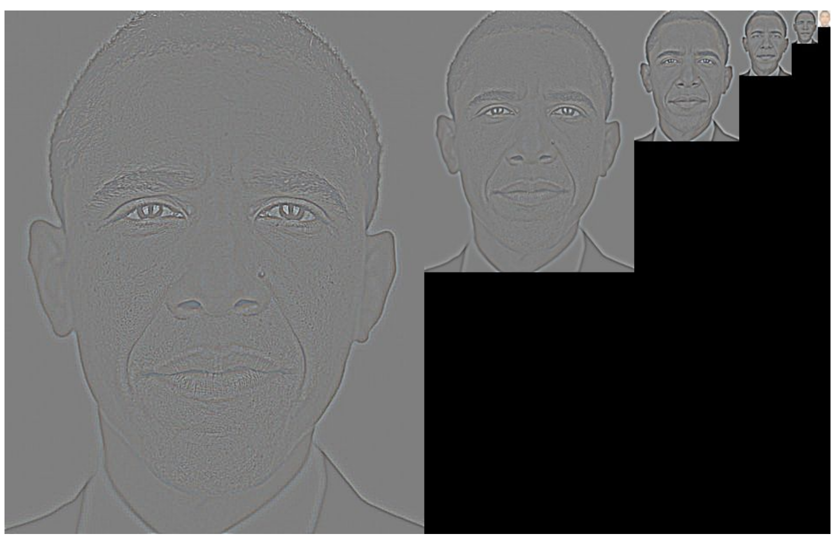

Figure 2. Pyramid built from the original image

Figure 3. Reconstructed image from the pyramid

If the user chooses blend mode, the program takes two input images and aligns them to the same size using an align function, which uses cv2.resize. Then the user can choose to either click the top, right, bottom, left vertices of an ellipse or specify the width, height, center coordinate and angle of an ellipse to construct the mask. The user can also enter the kernel size and kernel sigma for the Gaussian blur. It would then build up two pyramids based on the two input images, alpha blend each layer, and reconstruct to generate a new blended image. The program would also perform a direct alpha blend for reference.

Figure 4. The original image for pyramid blending

Figure 5. Pyramid alpha blending(left) vs. Direct alpha blending(middle),

sigma = 5, kernel size = 35

If the user chooses hybrid mode, the program takes two input images and produces the low-pass filtered image for the first and the high-pass filtered image for the second. Then the program generates a weighted sum of the two filters using two specified weights. The result is shown in Figure 6.

Figure 6. Hybrid output

Discussion:

Who did what for this project?
We worked together on both the coding part and the report.

How did you obtain and align your images for each of the two tasks? Did you use any third- party software (e.g. Paintbrush, Photoshop), or write a program to help prepare the images or mask?
We wrote a helper function to align two images. The function allows users to choose three critical points in both images, which outline the positions of the mask in both images. For example, if we want to blend two faces, then we choose points on the contours of the figure faces. The function then uses affine transformation to map image B to image A by matching these critical points. We choose to use affine transformation instead of perspective transformation because perspective transformation performs much worse than affine transformation when the two faces have largely different features, and it does not perform much better than affine transformation in other situations. The downside of this method is that if the two target objects have very different features, then it is impossible to align them together without distorting the object B to some extent.

What depth did you choose to build your Laplacian pyramid to, and why?
we chose to stop building the Laplacian pyramid when the minimum dimension of the current image from Laplacian pyramid is shorter than or equal to 8 pixels. That means the smallest image produced has both width and height no smaller than 8 pixels. Because the kernel used inside pyrDown is 5 pixels by 5 pixels, if we choose to make even smaller image, then it would be smaller than the kernel and raise the border-handling issue.

Why does Laplacian pyramid blending blend low-frequency content over a larger distance than high-frequency content? See if you can illustrate this with some carefully chosen input image examples.
When producing the Laplacian pyramid, the high-frequency features become less and less obvious. This is because high-frequency features get averaged out during the process. As shown in the Barack Obama example, the ripples on his face get progressively smoothed, and his eyebrows get blurred gradually. However low-frequency features get more obvious during the process as the edges gradually disappear. In the case of Barack Obama images, the color areas start appearing as the image gets smaller. Therefore, we can conclude that larger Laplacian pyramid images have more high-frequency content and smaller Laplacian pyramid images have more low-frequency content. Because at the coarsest level, one single pixel on the images corresponds to a much larger square in the full sized image, low-frequency content is blended over a large distance. In contrast, one pixel on the larger Laplacian pyramid corresponds to a smaller square in the full sized image, so high-frequency content is blended over a smaller distance.

How did you arrive at good values for the constants A, B, kA, and kB for the hybrid image generation? Describe the process.
We started with sigma_a = 10, sigma_b = 5, kA = 1, kB = 1. We realized that we want image A to be more blurred for us to recognize image B in close distance so we increased sigma_a to 50 and sigma_b to 30. We also decreased kA to 0.8 and kB to 1.2 to make image A less recognizable at close distance. In addition, we also change the kernel size of A to (55,55) and kernel size of B to (15,15).

If you display your hybrid image at full size on your computer screen, how close do you need to be in order to primarily see image B? How far away do you need to get before you only see features from image A? Are these distances fairly consistent between you, your lab partner, and any unsuspecting friends you show your image to?
We found that we need to stand in front of the computer(about 1-2 feet) to primarily see image B. When we are about 10 feet away, we only see features from image A. These distances are fairly consistent between us and some friends we showed to.

What does the Laplacian pyramid of your hybrid image look like?
The Laplacian pyramid of our hybrid image is shown in Figure :

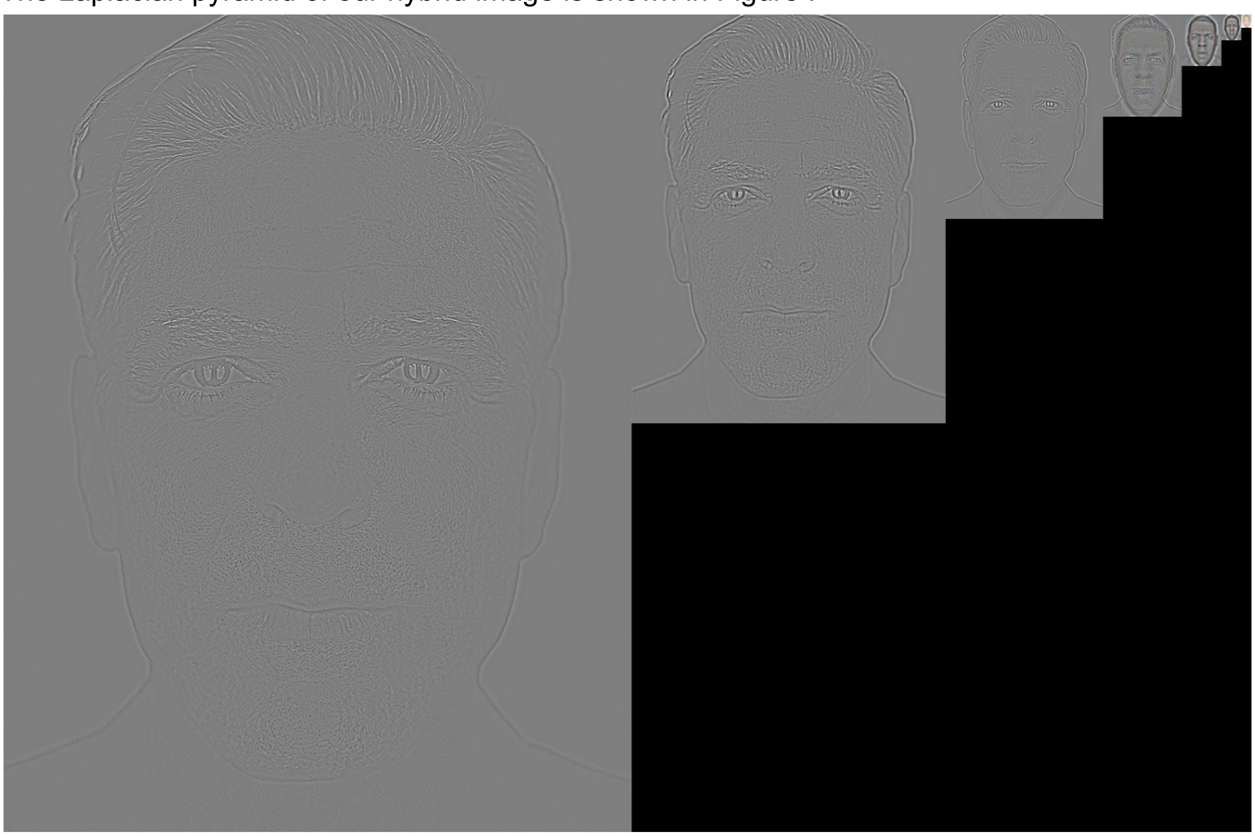

Figure 7. Laplacian pyramid of the hybrid image

As we can see, higher frequency contents(George Clooney) are dominant in the first several layers and as we get to the coarsest level, the lower frequency contents(Jay-Z) starts to take over.

Acknowledgments:

Responsibility were distributed evenly between lab partners. We would like to thank Professor Zucker for his help throughout the project.