Lab 1B: Pyramid Blending

CM30080

Dr Matthew Brown

Introduction

In this lab you will implement Laplacian Pyramid Blending [BA83] to smoothly combine two images. You will use MATLAB to specify an area in one of two equally sized images, then your software will then smoothly combine the sections of these to create a seamlessly blended result, see Figure 1.

To start, revise the lecture notes and chapter 3.5 in Szeliski's book [Sze11]. Make sure you have also completed the introduction to MATLAB Lab 1A.

Skeleton Code

A template is included in the file `PyramidBlend.m`. Run this example by typing `PyramidBlend` at the MATLAB command prompt. Draw a polygon around an object by clicking points and double clicking to finish. This should splice the two images together based on the area that you have selected (similar to Figure 1(c) below).

Gaussian Pyramid

First, check that you can blur an image with a separable kernel using the convolution code from Lab 1A. You can use `[1 4 6 4 1] / 16` as an approximation to the Gaussian function.

Now write a function `function pyr = GaussianPyramid(im)` which forms a Gaussian Pyramid `pyr` from the colour input image `im` (use the built-in MATLAB function `imresize`).

Laplacian Pyramid

Test creating a bandpass image by subtracting a blurred version of the image from itself. Note that this image will contain negative values, so you will need to use the `NormaliseImage` function from Lab 1A to view it.

Now write a function `function pyr = LaplacianPyramid(im)` to compute a Laplacian pyramid from the image `im`. Ouptut the sequence of images and verify that they show increasingly larger scale details.

(a) (b)

(c) (d)

Figure 1: Pyramid blending: (a)-(b) source images, (c) cut and paste, (d) blended result.

Reconstruction

Check that you can reconstruct the original image from your Laplacian Pyramid by successively upsampling factor 2 and adding levels from bottom to top. Note that you may need to remove the last row or column from the image after upsampling if the current pyramid level has an odd number of rows or columns[1].

Pyramid Blending

You will now write code to smoothly combine the different frequencies encoded in your Laplacian Pyramids to give the result in Figure 1(d). Remember that each frequency should be blended over a range proportional to spatial wavelength, so you you will need to use different splines for each level of the pyramid. You can obtain the blending weights by constructing a Gaussian Pyramid from the original user-defined mask. Test the effect of using variable numbers of pyramid levels in your reconstruction.

References

[BA83] P. Burt and E. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[Sze11] Richard Szeliski. *Computer vision: algorithms and applications*. Springer, 2011.

[1] this is because `imresize` rounds up when factor-2 downsampling an image with odd numbered side