Homework 08, UST

Downsampling and Upsampling of an Image using Pyramids (pyrUp and pyrDown Methods)

Mr. Saeed Ullah,
Korean Institute of Science & Technology Information (KISTI),
University of Science & Technology (UST)
Saeedonline12@gmail.com

Summary—This report presents Downsampling (Zoom out) and Upsampling (Zoom In) applied on an image. Tools used in this project are OpenCV 3.2 Library which is a library used for Computer Vision, and Visual Studio 2015 (64 bit).

I. INTRODUCTION

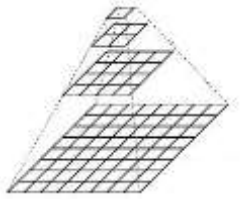
In computing, zooming is the ability to zoom in and out a document or image at specified level. It is usually found in all most all of the computer applications as it improves accessibility for people with visual impairment and people using mobile devices which have a relatively small screen.
Some of the important concepts for this project are:

- **Image Pyramid**

An image pyramid is a collection of images - all arising from a single original image - that are successively downsampled until some desired stopping point is reached.
There are two common kinds of image pyramids:
Gaussian pyramid: Used to downsample images
Laplacian pyramid: Used to reconstruct an upsampled image from an image lower in the pyramid (with less resolution).
In this tutorial I have used the Gaussian pyramid.

- **Gaussian Pyramid**

Imagine the pyramid as a set of layers in which the higher the layer, the smaller the size.



Every layer is numbered from bottom to top, so layer $(i+1)$ (denoted as G_{i+1} is smaller than layer i (G_i).
To produce layer $(i+1)$ in the Gaussian pyramid, I did the following steps:

I. Convolve G_i with a Gaussian kernel:

$$\frac{1}{16}\begin{bmatrix}1 & 4 & 6 & 4 & 1\\ 4 & 16 & 24 & 16 & 4\\ 6 & 24 & 36 & 24 & 6\\ 4 & 16 & 24 & 16 & 4\\ 1 & 4 & 6 & 4 & 1\end{bmatrix}$$

II. Removed every even-numbered row and column. It can easily be noted that the resulting image will be exactly one-quarter the area of its predecessor. Iterating this process on the input image G_0 (original image) produces the entire pyramid. Now
III. First, upsize the image to twice the original in each dimension, with the new even rows and columns filled with zeros (0)
IV. Perform a convolution with the same kernel shown above (multiplied by 4) to approximate the values of the "missing pixels".

These two procedures re implemented by the OpenCV functions pyrUp and pyrDown.

II. OPENCV SOURCE CODE

```
/// Load image
src = imread("..\\data\\lena.jpg");
// Perform an infinite loop waiting for user
input.
while (true)
{
int c;
c = waitKey(10);
if ((char)c == 27)
{ break; }
if ((char)c == 'i')
{
pyrUp(tmp, dst, Size(tmp.cols * 2, tmp.rows *
2));} else if ((char)c == 'o') {
pyrDown(tmp, dst, Size(tmp.cols / 2, tmp.rows /
2)); }
imshow(window_name, dst);
tmp = dst;}
```

III. RESULTS

Fig. 1. Original Image

Fig. 2. Image with Zoom In

Fig. 3. Image with Zoom Out

CONCLUSION: - You can also perform Zoom In and Zoom Out on an Image by using resize() function or some other user defined functions but OpenCV comes with built-in powerful functions i.e. pyrUp() and pyrDown() by using which one can perform zooming tasks more easily and efficiently.