IMAGE BLENDING: MIX N' MATCH

Gursimran Kaur (2018336), Hrithik Malhotra (2018236)

B.Tech, Indraprastha Institute of Information Technology, CSSS 22'
B.Tech, Indraprastha Institute of Information Technology, CSAM 22'

ABSTRACT

The aim of this project is to develop a CLI Interface to blend two input images. The motivation for this project arose due to an extremely rare occurrence while capturing your memories through photos by almost everyone: the pursuit of the *perfect* photo. It is a most common occurrence to find your pictures where either your eyes are not open, or you are not smiling nicely, or a thousand other issues. We have come at a solution to face swap your face from pictures where you look fantastic on pictures where some disturbance has occurred. For various purposes, face swapping synthesizing is becoming famous. We aim to analyse feature selection using image blending techniques and preserve the principal parts of the two overlapping images.[2]

Image blending, Gaussian Pyramids, Laplacian Pyramids, Facial Detection, ORB and CascadeClassifier

1. INTRODUCTION

Recently, the image blending has taken up applications even beyond the conventional uses. Image-based rendering applications and computer graphics have seen varied applications over the past centuries. Now, we have plenty of filters to use for different purposes ranging from sharpening, blurring to extracting important information from the images. Resizing the pictures from which the faces have to be swapped needs to be achieved effectively, which we have done using Laplacian and Gaussian pyramids. We aim to blend such images by weighting according to the mask created by recognising faces in an image. Our results show that the proposed model is working effectively.

2. THEORY

2.1. Gaussian Pyramid

The Gaussian pyramid is a technique in image processing that breaks down an image into successively smaller groups of pixels to blur it. When a digital photograph is blurred in this way, edges of objects are easier to detect, enabling a computer to identify them automatically. The "pyramid" is constructed by repeatedly calculating a weighted average of the neighboring pixels of a source image and scaling the image down. It can be visualized by stacking progressively smaller versions of the image on top of one another. This process creates a pyramid shape with the base as the original image and the tip a single pixel representing the average value of the entire image.

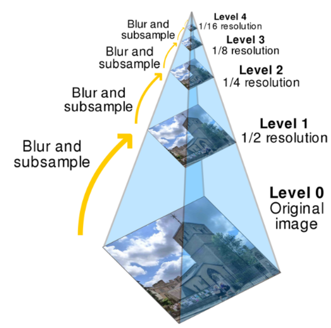

Figure 1: The process of generating a Gaussian Pyramid.

2.2. Laplacian Pyramid

The Laplacian pyramid, is a bandpass image decomposition derived from the Gaussian Pyramid which is a multiresolution image representation obtained through a recursive reduction (lowpass filtering and decimation) of the image data set. The image is recursively lowpass filtered and downsampled to generate a lowpass sub-band, which is re-expanded and subtracted pixel by pixel from the original image to yield the 2-D detail image having zero-mean. The output of a separable 2-D filter is downsampled along rows and columns to yield the next level of approximation. Again, the detail is given as the difference between the original image and an expanded version of the lowpass approximation.[1]

Thanks to AV Sir for teaching this fantastic course to us.

2.3. Feature selection using ORB and CascadeClassifier

We have used ORB for feature selection and CascadeClassifier for detecting faces in the digital images to generate our mask. ORB is a fusion of FAST keypoint detector and BRIEF descriptor with many modifications to enhance the performance. First it uses FAST to find keypoints, then apply Harris corner measure to find top N points among them. It also uses pyramid to produce multiscale-features. It computes the intensity weighted centroid of the patch with located corner at center. The direction of the vector from this corner point to centroid gives the orientation. To improve the rotation invariance, moments are computed with x and y which should be in a circular region of radius r, where r is the size of the patch.
Object Detection using Haar feature-based cascade classifiers is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Now, for each feature, it finds the best threshold which will classify the faces to positive and negative.We select the features with minimum error rate, which means they are the features that most accurately classify the face and non-face images. [4]

3. METHODOLOGY

The algorithm implemented to obtain the blended image is as follows:-

1. Two Laplacian pyramids are constructed for the source and the target image respectively. LA and LB are the Laplacian pyramids built from A and B respectively.

2. The feature extraction for face would take place to generate the required binary mask, where 1 would signify the overlap of the first image pixel on the second one, and 0 would ignore the second image pixel value.

3. The two pyramids are then blended based on a corresponding Gaussian pyramid for the mask. GP - Gaussian Pyramid is built from the required regions from original pictures.

4. To construct the blended image, the algorithm starts from the top level in the blended Laplacian pyramid, combines the top most two levels, takes the combined image to combine it with the third level image, and so on, until all the levels in the blended Laplacian pyramid have been combined to produce the blended image. Now we formulate a combined pyramid CP, efficiently from LA and LB (calculated in step-1) weighing it with the nodes of the gaussian pyramid. Mathematics behind the final calculation of combined pyramid for the resultant image would be:
   $CP(i,j) = GP(i,j) * LA(i,j) + (1 - GP(i,j)) * LB(i,j)$

This methodology has several advantages, like preserving the color of the source image and being computationally inexpensive. The computation cost is very low for such blending techniques since each level's creation results in a convolution's cost (for convolution with the Gaussian kernel). This task is optimised by us by convoluting with the kernel in the Fourier domain and then taking an inverse Fourier transform to obtain the resultant image. One observation of this method is that the effect depends on the size of the Gaussian filter, but not monotonically. If the size of the filter is too small, you might get a clear boundary between the source and the target image. If the size of the filter is too large, you might get artifacts at the positions of the boundary of the source image

4. RESULTS

We tested our code using several testing images, some of which are shown below. Since the aim of our project was to get 'better' images, we have tested using common errors which occur while clicking pictures. Several issues which occur while clicking photos are composed of if not restricted to closing your eyes, not smiling right etc. The input to the algorithm has to be the file names of the two original images, where the later has to be the image whose mask has to be created.
The command to run the algorithm is simply
python SourceCode.py
After running this command in the directory where the code and images are present, we would be greeted with an input choice of the form
Enter the name of the first image(with extension): image1.jpeg (E.g)
Enter the name of the second image(with extension): image2.jpeg (E.g)

As you can observe, using a mask for the face of the person whose eyes are closed in the first image, our algorithm blended this with the second image to obtain the desired image.
In this testing case, even if the person is standing in different poses or having different facial features, our algorithm is smart enough to learn the matching features and obtain the desired outcome.

5. CONCLUSION

The entire algorithm is running extremely well, accommodating different test cases. We have successfully accomplished the task at hand which was to blend two images of people in order to obtain an image where the *face swap*

Figure 2: Original Images when Eyes were closed.

Figure 3: Blended Image of Figure 2

is almost impossible to notice with the naked eye. Several adaptations were made to this final code in order to decrease its running time like optimising while shifting the Gaussian Pyramid Level computation to the Fourier Domain.

6. REFERENCES

[1] Li xingmei, Chen Liang and Wang Jin, *The application of Laplacian pyramid in image super-resolution reconstruction*, 2nd International Conference on Signal Processing Systems, IEEE, China, July 2010.

[2] R.T. Whitaker, *A level-set approach to image blending*, IEEE Transactions on Image Processing, vol. 9, no. 11, pp. 2-3, USA, Nov. 2000.

[3] Wei Wang and Michael K. Ng, *A Variational Method for Multiple-Image Blending*, IEEE Transactions on Image Processing, vol. 21. no. 4, China, April 2012.

[4] Zhang Xingjie, Joongseok Song and Jong-Il Park, *The image blending method for face swapping*, International Conference on Network Infrastructure and Digital Content, IEEE, China, pp. 6-9, Sept. 2014.

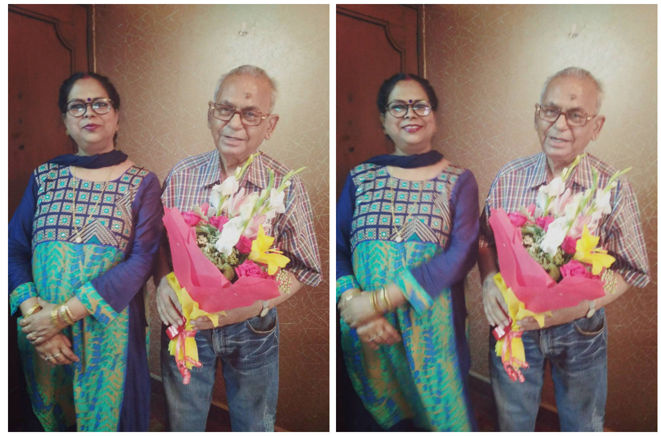

Figure 4: Original Images when posture was different

Figure 5: Blended Image of Figure 4