Automatic Face Swapping

Batuhan Yaman
Mehmet Alperen Derin
Hacettepe University
Ankara,Turkey
batuhan.yaman@hacettepe.edu.tr
mehmet.derin@hacettepe.edu.tr

Abstract

Face swapping is a commonly used concept in entertainment business. This project aims to achieve to swap face from two different images automatically. We detected faces using Viola-Jones algorithm and after that we did shifting and re-sizing. Lastly faces are swapped using Poisson image editing techniques.

1 Introduction

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images.[1] Nowadays, face detection is important part of image processing and computer vision. Addition to its critical usage, face detection is also used in entertainment business. Combined with image manipulation, bunch of softwares offer face swap applications. In this project our goal is to swap faces in two images that were given to our program. Every operation will be automatic. So user only needs to provide target and source images. To achieve this, we need to detect both faces in images. Then extract the necessary features and transfer them into target image. We need to transform our source image's colors to target image's color. For better fitting we are re-sizing the source image.

2 Related Work

- Photomontage
  "The history of photomontage is nearly as old as the history of photography itself. Photomontage has been practiced at least since the mid-nineteenth century, when artists like Oscar Rejlander [1857] and Henry Peach Robinson [1869] began combining multiple photographs to express greater detail. Much more recently, artists like Scott Mutter [1992] and Jerry Uelsmann [1992] have used a similar process for a very different purpose: to achieve a surrealistic effect. Whether for realism or surrealism, these artists all face the same challenges of merging multiple images effectively." [2]

- Neural Network-Based Face Detection [3]
  In this research, they used neural network based algorithm to detect faces in gray scale images. Their methods designed to be more general, with little differences in faces. They used $20x20$ pixel regions to determine if the face is present or absent. For larger faces than this region, sampling is applied. In the following steps, lighting correction and histogram equalization is applied. With result images neural network is trained to give an output ranging from -1 to 1, which represents absence or presence of the face. "... the system is able to detect 90.5 percent of the faces over a test set of 130 complex images, with an acceptable number of false positives."

- Poisson Image Editing [4]
  This research uses Poisson equation as well as many other similar works. They propose a guided interpolation, with the guidance being specified by the user. "The mathematical tool at the heart of the approach is the Poisson partial differential equation with Dirichlet boundary conditions which specifies the Laplacian of an unknown function over the domain of interest, along with the unknown function values over the boundary of the domain."

- Coordinates for Instant Image Cloning [5]
  "All gradient domain techniques eventually solve a large sparse linear system, the Poisson equation.". Instead of solving the Poisson equation, they propose coordinate-based

approach. Without solving the equation, algorithm needs less computational power is required and has small memory footprint. With this easy-to-implement algorithm, they achieved real-time seamless cloning and healing of still images and video sequences on the CPU, as well as on the GPU.

3 The Approach

In order to swap faces, first we need to detect them. Here we are using Matlab's built-in class *Cascade Object Detector* which uses Viola-Jones algorithm. This algorithm detects faces rapidly with high success rate, but it is weak when the faces are rotated or not directly pointed to the camera.
This algorithms uses Haar Features[6] and eliminates feature redundancy with Ada Boosting[7]. When a feature is proven its relevancy for finding specific region, in this case we are using "Eye Area" and "Mouth", these features tells us that there might be a face. Before we tried using "Left Eye" and "Right Eye" instead of only "Eye Area". Although sometimes the precision of detecting eyes individually was better, mostly detection was way harder or was not possible at all. So in order to generalize and increase the possible input image domain, we decided to continue with "Eye Area". For obtaining the final and only face, we used *MergeThreshold* which is a threshold value to merge our useful features.
First, we found both target and source images eyes and mouth. We compared their width and height then re-sized source image accordingly. In order to do Poisson image editing well we had to re-size our source image according to target image but in the same time size of the face had to remain same for best swap. So we find the center point of the face in the source image and cropped the corners with centering to that point.

Figure 1: Uncropped / Cropped

We obtained 4 different coordinates:

- Left eye's top left corner - x1,y1
- Right eye's top right corner - x2,y2
- Mouth's bottom left corner - x3,y3
- Mouth's bottom right corner - x4,y4

With these coordinates we formed our mask. We used following formula for applying our mask and transfer that area of source image to target image:

$SourceImage \times Mask + TargetImage \times (1 - Mask)$

Transferred image and target image are aligned to their face centers. Then we applied two Poisson image editing technique, *PoissonJacobi* and *PoissonGaussSeidel*.

4 Results

For experimental part, we did many tests but finally obtained 5 test results.

Figure 2: Target / Source

Figure 3: Jacobi / GaussSeidel

Figure 4: Target / Source

Figure 5: Jacobi / GaussSeidel

Figure 6: Target / Source

Figure 7: Jacobi / GaussSeidel

As we can clearly see from the results, Gauss-Seidel algorithm provides better results with nice color transition. But at least from our experiments, it takes more time than Jacobi. Our algorithm works best with similar sized images and as we stated before, our weakness is poorly lighted and not straightly faced images.

Figure 8: Target / Source

Figure 9: Jacobi / GaussSeidel

Figure 10: Target / Source

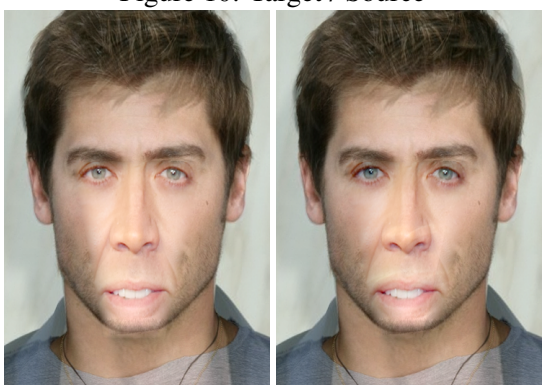

Figure 11: Jacobi / GaussSeidel

5 Conclusion

Face swapping is one of the most common used techniques in image processing, specifically in photomontage. We implemented a face swapping application using MATLAB. We detected faces

with ready-to-use, implemented Viola-Jones algorithm, we have done some intermediate process on images to make them ready for Poisson image editing techniques and we applied two Poisson image editing techniques using a 3rd party MATLAB toolbox. We got really satisfactory results in sufficiently lighted and straightly faced images. If these two are not satisfied, we cannot get a result since we fail at detecting faces. Another work can overcome these difficulties.

References

[1] https://facedetection.com/

[2] Agarwala, Aseem, et al. "Interactive digital photomontage." ACM Transactions on Graphics (TOG). Vol. 23. No. 3. ACM, 2004.

[3] Rowley, Henry A., Shumeet Baluja, and Takeo Kanade. "Neural network-based face detection." IEEE Transactions on pattern analysis and machine intelligence 20.1 (1998): 23-38.

[4] Prez, Patrick, Michel Gangnet, and Andrew Blake. "Poisson image editing." ACM Transactions on Graphics (TOG). Vol. 22. No. 3. ACM, 2003.

[5] Farbman, Zeev, et al. "Coordinates for instant image cloning." ACM Transactions on Graphics (TOG). Vol. 28. No. 3. ACM, 2009.

[6] Whitehill, Jacob, and Christian W. Omlin. "Haar features for facs au recognition." 7th International Conference on Automatic Face and Gesture Recognition (FGR06). IEEE, 2006.

[7] Freund, Yoav, Robert Schapire, and N. Abe. "A short introduction to boosting." Journal-Japanese Society For Artificial Intelligence 14.771-780 (1999): 1612.