

Your Final Project Title

Team Members
Department Names
UNiversity of Wisconsin-Madison
authors@wisc.edu

Team Members
Department Names
UNiversity of Wisconsin-Madison
authors@wisc.edu

1. Optimal Seam Search

After we warped the two images, we need to find an optimal seam along which the area extracted from the source image can be determined. There are three main concerns for the search. First, we want to avoid the the important features in face, such as eye, month, nose and so on. Second, the pixels on the boundary of the extracted area should be similar between the two images so that it will be more natural when we composite them together. Third, we want to reduce the computation complexity.

Our approach is based on Kwatra et al's work [1]. The algorithm works by formulating the image as a connected graph with edge weights based on the difference between neighboring pixels. This graph is then treated as a max-flow/min-cut problem where the sources are any pixels to be taken only from the first image and the sinks are any pixels to be taken only from the second image.

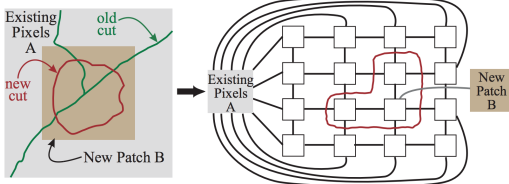


Figure 1. Graph representation of image, from [1]

As the figure 1 shows, in our project, the pixels in patch A must from our target image while the pixels in patch B must from source image. The goal is to find the optimal cut between patch A and patch B. We represent the image graph as an adjacency matrix while each entry in the adjacency matrix can be the flow. We use the sum of absolute difference between the source image and the target image plus the sum of absolute difference of the source image and the target image at the neighboring pixel.

$$M(c, n, S, T) = \|S(c) - T(c)\|_1 + \|S(n) - T(n)\|_1 \quad (1)$$

where c is the current pixel, n is the neighbor pixel, S is the source image and T is the target image. The $L1$ norm is

the sum of absolute difference of these two pixels in three channels. Note that the weight between the pixels in patch A and its neighbors must be infinity. This is same for patch B.

In our implementation, we first define the permissible path region by generating two masks based on the landmarks generated from the first step, shown in 2. The permissible path region is the area we want to find the optimal seam. All the pixels which are outside the right larger mask must from target image, i.e. patch B, and all the pixels inside the left smaller mask must from source image, i.e. patch A. Therefore, the permissible path region is the area which is inside the larger mask but outside the smaller mask. As the forehead region of a face is usually more flat than the other regions. A good permissible region is crucial to the following steps. First, the permissible path should not be too strict, then it's highly possible that we cannot find the optimal seam within the strict area. Second, the larger the permissible path region is, the higher computation complexity will be. Thus, when constructing the two masks, a convex hull is constructed by the landmarks. Then we split the face into two regions: the forehead region and the bottom region. For the forehead region, as the convex hull is right above the eyes while the forehead region is relatively flat, the mask is eroded so that the permissible path region will cover larger area. For the bottom region, because the convex hull is almost the boundary of the face, it is not reasonable to extend the permissible path outside the face, the mask is dilated. After generating two masks for each region via erosion and dilation, we stitch them together to get the final permissible path region.

For the construction of the adjacency matrix, we extend the patch A and patch B by including all the pixels which are immediate neighbors. In other words, if the neighbor of a pixel is in patch A or B, then the weight between them is also infinite. If the weight between two pixels is not infinite, then these two pixels must be 1) not in patch A and patch B and 2) their neighbors are also not in patch A and patch B.

The result is shown in Fig.3. The pink area is the permissible path region and the red circle in the left figure is

the result.

```
%mask_dst is initialized as the convex
    hull of the landmarks in target
    image
top_mask = mask_dst;
bottom_mask = mask_dst;
%split the mask into forehead region
    and bottom region
[v, indx] = max(mask_dst,[],1);
bd = indx(find(v==1,1));
top_mask(bd+1:end,:) = 0;
bottom_mask(1:bd,:) = 0;
%dilate for the forehead region
SE_disk = strel('disk',10,6);
top_mask_outer = imdilate(top_mask,
    SE_disk);
top_mask_inner = top_mask;
%erode for the bottom region
bottom_mask_outer = bottom_mask;
SE_disk = strel('disk',10,6);
bottom_mask_inner = imerode(bottom_mask
    , SE_disk);
%stitch the masks for the two regions
    together
mask_outer = logical(top_mask_outer +
    bottom_mask_outer);
mask_inner = bwconvhull(logical(
    top_mask_inner + bottom_mask_inner))
    ;
```

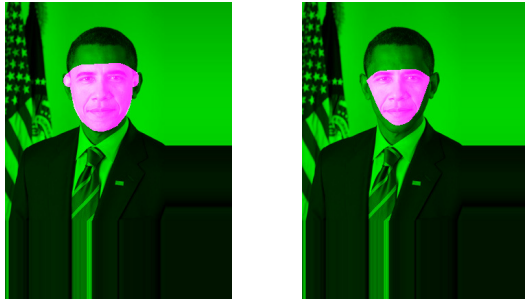


Figure 2. Mask for optimal seam search

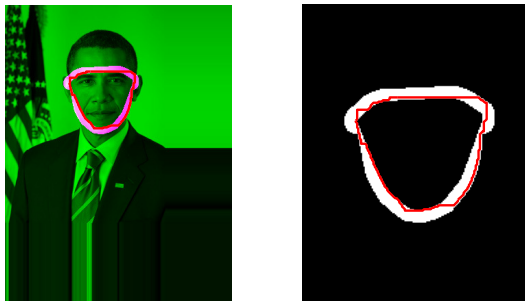


Figure 3. Result for optimal seam search

```
%mask is the area where we want to get
    the optimal seam
mask = mask_outer - mask_inner;
```

2. Face Blending

The optimal seam shows us what is the best area which should be extract from the source image. If we simply cut that area from the source image and paste it to the target image, the result will not be satisfiable, as the left image shown in Fig 5. The problem is the color of these two images are very different from each other. Therefore, we can see the color change abruptly around the boundary. In order to avoid the abrupt color changes, we use Poisson blending approach.

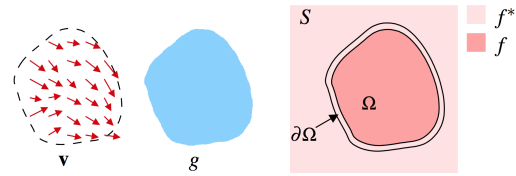


Figure 4. Poisson blending notation, [2]

Poisson blending is one of gradient domain image processing methods. As Fig.4 shown, v is gradient of a region in an image, g is selected region of source, f^* is a known functions that exist in domain S , f is an unknown functions that exist in domain Ω , Ω is a region g that is now placed on domain S (target background), $\partial\Omega$ is boundaries between the source and target regions. The goal is given v , find the value of f in the unknown region that optimize

$$\min_f \iint_{\Omega} |\nabla f - v|^2 \quad (2)$$

$$\text{subject to } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (3)$$

where $\nabla = [\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]$ is the gradient vector. Its solution is the unique solution of the following Poisson equation with Dirichlet boundary conditions:

$$\Delta f = \text{div } v \quad \text{over } \Omega \quad (4)$$

$$\text{subject to } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (5)$$

where $\text{div } v = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ is the divergence of $v = (u, v)$, and Δ is the Laplacian operator as follows:

0	-1	0
-1	4	-1
0	-1	0

This is the fundamental machinery of Poisson editing of color images: three Poisson equations of the above equation are solved independently in the three color channels of the chosen color space.

References

- [1] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graph-cut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003. [1](#)
- [2] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 313–318. ACM, 2003. [2](#)



Figure 5. Poisson blending