GROUP 1

Members:

Bárbara Darques Barros - 7243081
Juliana de Mello Crivelli - 8909303
Kaue Ueda Silveira - 7987498

Title: Conversion of static images into parallax animations

Abstract: Starting with a static image we try to retrieve objects at the same depth and separate them into images, each of them representing a different plane from the scene. Then we interpolate the blank spaces caused by the separation, extending the background. In order to produce the parallax effect we overlap the extended planes with slightly different displacements.

Roadmap:

1) Identify and separate planes
   We intend to detect regions applying Region Adjacency Graph (RAG) Merging over the original image. Each region will be put in a separate image, surrounded with transparent pixels.
2) Enlarge retrieved planes area
   We zoom in the planes in order to fill the areas covered by the foreground planes, avoiding blanks spaces to be displayed to the viewer.
3) Overlap new images with different relative positions
   The extended planes generated in the previous steps will be overlapped with a variable position difference, generating various frames that once put together simulate the relative movement of the scene objects.
4) Generate gif
   We'll use imageio modules in order to convert the output frames into a gif animation.

Input images:

The images chosen for input are landscape photos taken by the group students, such as Figure 1, and can be found at the following link: https://drive.google.com/drive/folders/0BxHj0zhF4J8-TXZYMmp3VDNXbW8?usp=sharing. We selected images with big contiguous regions, like lakes or forests with small variation in color and details.

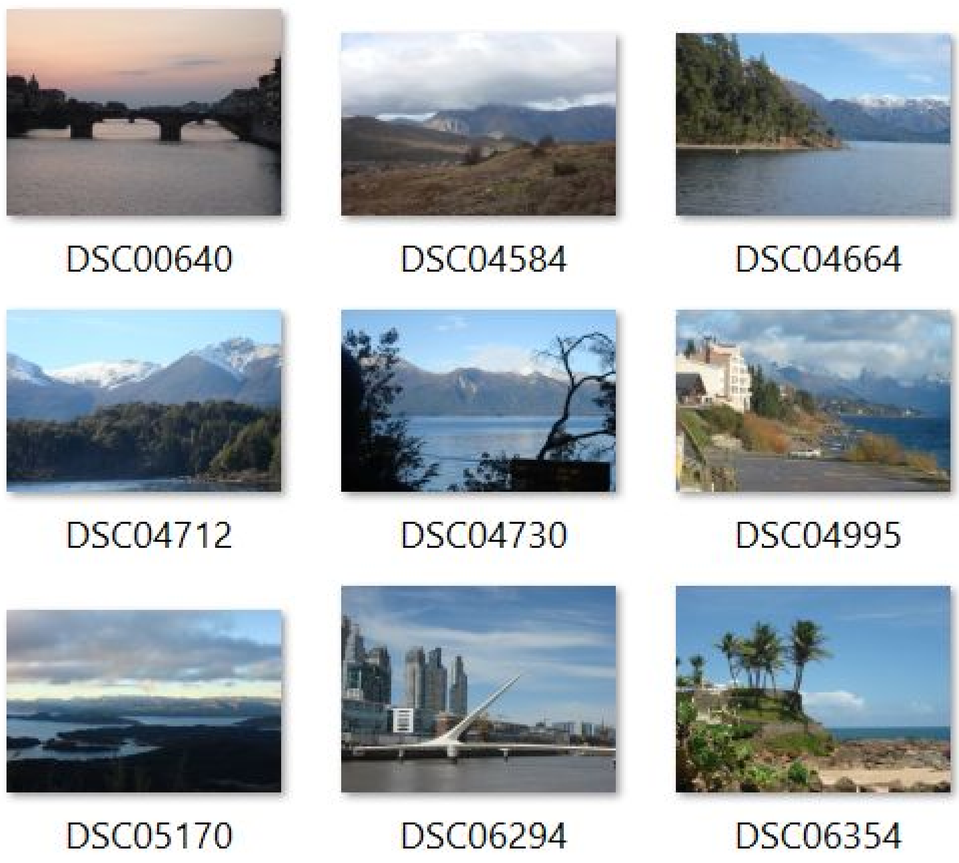

Figure 1: input image samples

Identify and separate planes:

We abandoned the original idea of using Fourier Transform, since it is best suited for black and white images. After several tests, such as watershed transform, a graph-based method proposed by Felzenszwalb & Huttenlocher and Random Walker algorithm, and Otsu-thresholds we were able to obtain a result similar to the desired one, using RAG Merging (Figure 2). The final result is displayed in Figure 3.

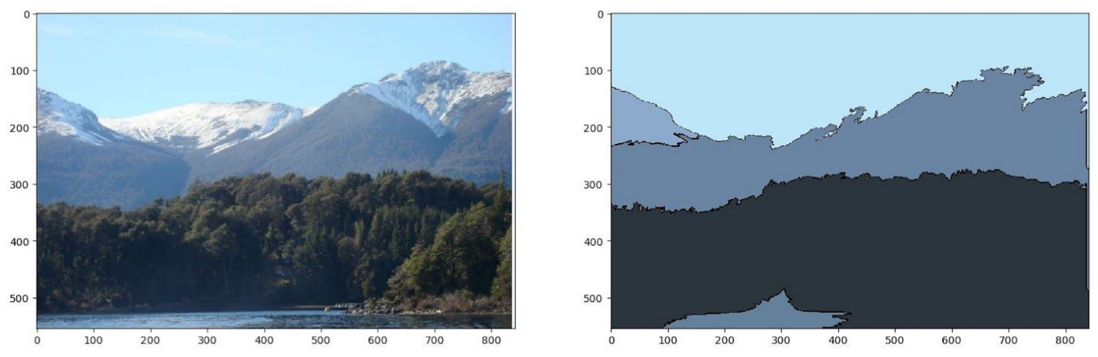

Figure 2: Image segmentated with RAG Merging

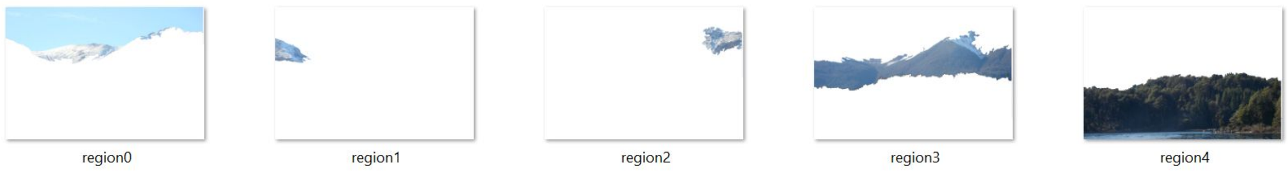

Figure 3: Retrieved planes

Enlarge retrieved planes area:

Searching for a simpler solution, we chose to zoom each plane instead of using some kind of interpolation method, such as the Papoulis-Gerchberg Algorithm. To make this decision we handcrafted a gif with the manually enlarged planes on Photoshop and observed that the resulting parallax was within our expectations. The only disadvantage was losing some of the resolution. To program this step we used opencv functions.

Overlap new images with different relative positions:

To obtain a frame we combined each plane in different phases, for instance setting the plane #0 three pixels to the left of plane #1. The number of generated frames considers how much you can shift the planes before getting any anomaly.

Generate gif:

Using the function `imageio.mimsave` and the frames generated in the previous step we convert the frames to an animation and save it as a gif.

Conclusions:

Each image has their own optimal parameters for segmentation and layer order. Even though it would be possible to figure out the best parameters through an artificial intelligence this application would work well as a semi-automated parallax animation generator, where the user can do some tweaking of the parameters or choose the best between a set of animations based on the same input image.

Code and Generated Animations:

https://github.com/jumc/img_parallax_gif

References:

https://jaydenossiterghostart.wordpress.com/2016/05/12/the-history-of-the-parallax-effect/ (*The History of the Parallax Effect*)
http://www.nooganeer.com/his/projects/image-processing/making-a-gif-with-opencv-and-scikit-image-in-python (*Making a GIF with OpenCV and Scikit-Image in Python*)
http://sweet.ua.pt/pjf/PDF/Ferreira94e.pdf (*Interpolation and the discrete Papoulis-Gerchberg Algorithm* - Paulo J. S. G. Ferreira)
revistas.ua.pt/index.php/revdeti/article/download/1720/1597 (T*eaching signal and image reconstruction algorithms* - Paulo J. S. G. Ferreira)
http://cs.brown.edu/~pff/papers/seg-ijcv.pdf (*Efficient Graph-Based Image Segmentation* - P. Felzenszwalb, D. Huttenlocher)
http://scikit-image.org/docs/dev/auto_examples/segmentation/plot_rag_merge.html