Magic Sky: Sky Replacement

Jing Wu
2017301200171
Wuhan University

Yuan Liu
2017301110209
Wuhan University

Zhao Shen
2017301200295
Wuhan University

***Abstract*— The sky is a common background in photos, but because of the the time of photographing, it is usually not very interesting. In this report, we propose an automatic background replacement algorithm that can generate realistic, artifact-free images with a diverse styles of skies. The key idea of our algorithm is to use semantic segmentation to guide the entire process, including sky segmentation, replacement, color transfer and guide filter.**

Keywords: sky segmentation, sky replacement, image fusion, appearance transfer, guided filter.

I. INTRODUCTION

The sky is one of the most common backgrounds in photos. However, we cannot control the weather or lighting conditions when shooting. As a result, many interesting and valuable photos have uninteresting or poorly exposed sky areas. Professional photographers use sophisticated tools to solve this problem by manually and accurately depicting the sky area, and finally adjusting the foreground to match the new synthetic sky. This is time-consuming and requires expertise that novice users cannot have. Therefore, in this project, we implemented an automatic sky replacement tool that can take input images and skies, and then generate a picture that combines the selected sky and original image.

To achieve this goal, we address two challenging questions in this work. Can we accurately segment the sky region from the image? Can we match the appearance of the input image and the new sky image to create a realistic composition?

In this work, we choose U-Net as our network architecture for sky segmentation so that we can get an image mask. The full steps are as follow:

- Sky segmentation
- Sky replacement
- Color transfer
- Post process

Plus, we implement our algorithm into video process. Even it can't be played real time, the outcome still performs well.

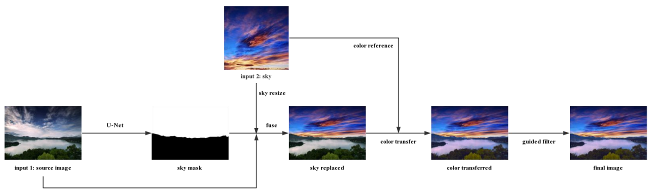


Figure 1: Pip line of our work

II. THEORY AND METHOD

A. Sky Segmentation and Replacement

The appearance of the sky varies greatly between images, and if you don't understand the scene layout, it may not be distinguishable from non-sky areas (e.g. reflections in water).

We use Convolutional Networks for Biomedical Image Segmentation (U-net) to parse the input image and generate dense pixel-level predictions of semantic labels. More specifically, as illustrated in Figure 2, there are two parts in this net. The left half part zooms out the picture and extract the features we want in multi-scales. The right half zooms in the picture step by step and at last retrieves it back to the original scale. In the last layer, we get a 64-dimentional feature vector.

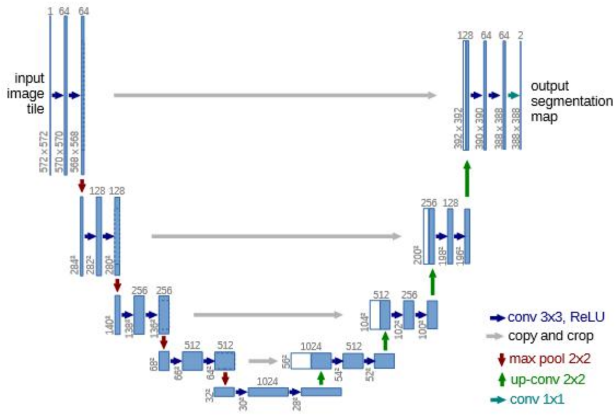


Figure 2: Architecture of U-Net

By binarizing it, we get the last mask that will be used in the subsequent steps.

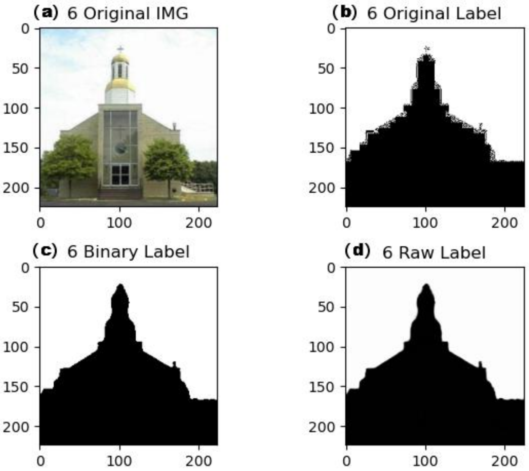


Figure 3: (a) is the input, (b) is the raw output of U-Net, (c) is the label by binarizing (b), (d) is the given label used for training

By understanding the overall layout of the scene, despite the different colors, shapes, sizes and attributes, the proposed algorithm can still locate the sky area robustly.

Now we choose a sky image and the original image as inputs and take the binarized label as a mask. By getting the element-wise production of input and mask, we obtain the robust composition of selected sky and the foreground objects in the original image.

B. Color Transfer

Our implementation of color transfer is (loosely) based on *Color Transfer between Images* [Reinhard et al, 2001]. In this paper, Reinhard and colleagues demonstrate that by utilizing the LAB color space and the mean and standard deviation of each L, A, and B channel, respectively, that the color can be transferred between two images.

There are many other methods that can perform much better than the above method. However, as mentioned in Section I, we aim to implement our algorithm into video process, which means we need to take consumed time into account. In, Reinhard's method, we only compute the mean and the standard deviation of each image, which costs less time than other algorithms.

In fact, to make the method fit in our project, we did some modification and optimization on the basis of the original algorithm. Specifically, given the sky image and the sky-replaced image which is obtained in Section A, we first convert them into LAB color space and then calculate the mean and derivation of L, A, B channel respectively. Since the number of pixels in an image is big enough, we can consider they belong to Gaussian Distribution. While the specific mean and derivation of a distribution denote different attributes of the luminance and chroma, we can easily transfer the color style of the source image to that of the sky image by the following formula:

$$\frac{I_{src}-\mu_{src}}{\sigma_{src}}=\frac{I_{sky}-\mu_{sky}}{\sigma_{sky}}$$

Where I is the value of each pixel in each channel, μ is the mean, σ is the derivation.

Unlike the original method, we don't take all pixels of the sky-replaced image into computation. When calculating the parameters of the source image, we only include those who belong to foreground objects, which correspond to the dark region in the mask. We do so because the sky in our source image is exactly the sky we chose in Section A. The foreground region in this image is the part that need to be transferred. Also, we add weight parameter into the formula so that the transferred color of foreground objects looks more harmonious.

$$I_{transfer}=\frac{I_{src}-\mu_{src}}{\sigma_{src}}\times\sigma_{\exp}+\mu_{\exp}$$

$$\mu_{\exp}=\alpha\times\mu_{sky}+(1-\alpha)\times\mu_{\mathrm{src}}$$

$$\sigma_{\exp}=\alpha\times\sigma_{sky}+(1-\alpha)\times\sigma_{\mathrm{src}}$$

In the above formulas, I_{transfer} is the outcome we want, α is a parameter in [0,1] which denotes the weights of sky image in color transfer. Setting α at 0 means that we don't change the color while 1 means that we transfer the color totally the same as the sky. After many times of experiment, we recommend α should be neither too small nor too big. In our work, we set α at 0.5. As figure 4 illustrates, the modified algorithm can keep some of its original color when implementing color transfer and the outcome in the proposed method looks more realistic.

(a)source (b)original method (c)proposed method

Figure 4:comparison of color transfer method

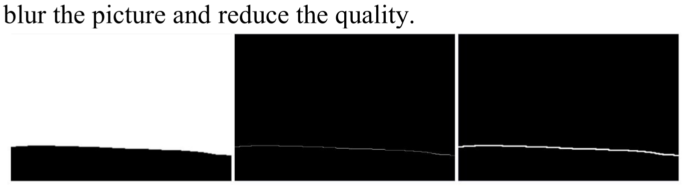
C. Post Process

Our trained model cannot guarantee a perfect sky mask. This will lead to an artifact edge between the sky and the foreground. Figure 5 shows the sky mask and the sky-replaced image. It is clear to see that the edge is a zigzag rather than a smooth line. A lot filters function as smoothing filters, such as mean filter, median filter Gaussian filter and so on. In our work, we choose guided filter [Kaiming He et al,2013]to smooth the zigzag edge.

(a)mask (b)image with zigzag (c)zigzag

Figure 5:the zigzag was formed because of the imperfect mask

Guided filtering algorithm is a kind of filtering algorithm that can keep the edge while smoothing the image. But in our project, we hope to smooth the edge and keep the difference of chroma between the sky and the foreground simultaneously. Plus, we don't need to process the whole image with the filter. This will blur the picture and reduce the quality.



(a)mask (b)extracted edge (c)dilated edge

Figure 6:(a) is the sky mask, (b) contains the edge between sky and foreground, (c) is the thicker edge after dilating

Therefore, we first decide the area that needs to be processed. Concretely, we extract the edge by filtering the sky mask with

Canny operator and then dilate it with a 5 by 5 matrix so that the edge region becomes a little larger. As Figure 6 shows, the white region is exactly the area that should be filtered. In Figure 7, it is clear that after filtering, the edge is smoother and the picture looks much more harmonious.

(a)non filtered (b)filtered (c)non filtered edge (d)filtered edge

Figure 7:comparison between filtered image and non filtered

D. *Video process*

Our algorithm can be applied in video process at a high speed. Given a video, we read each frame and then process one per time. At first, our algorithm is very time consuming and the time of processing one frame on average is about 7s. By optimizing the algorithm, we at last make the average time at about 0.57s. more details would be introduced in Section III.E.

III. EXPERIMENT AND RESULT

A. *Training Phase*

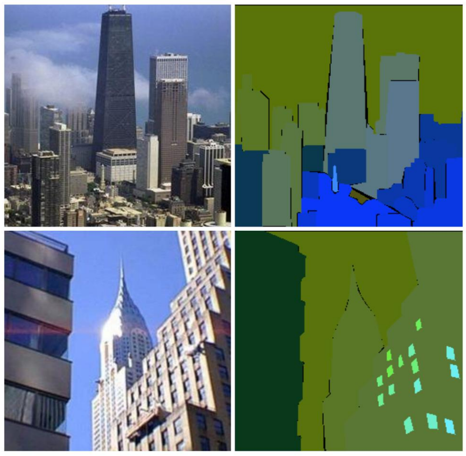


Figure 8:the original image and the correspond label, the color of sky label and building label is very close, which may lead to the wrong label convert

(a)original image (b)mask (c)sky (d)sky-replaced (e)color transferred (f)guided filtered

Figure 9:the outcome of each process and the final result

Our data set is chosen from ADE20k. It has more than 25,000 images (20k train, 2k val, 3k test), which are densely annotated with an open dictionary label set. To fit it into our project, we first select 903 images that contain sky label and then re-label them in the criteria that sky is 1 and others are 0. We split the whole data set into training set of 813 images and test set of 90 images.

Our model was trained for 200 epochs. It should be noted that the labels in ADE20k are not perfect. As Figure 8 shows, some pixels which should be labeled as sky indeed are assigned to foreground objects, which to some degree incurs the inaccuracy of our trained model. In addition, since pictures in our data set are all about natural scenes and buildings, it performs well when the input is related to scenery. However, if the edge between sky and foreground is far too meander, there may be some mismatched pixels. Overall, the final model performs 95.1% accuracy rate and the mean of loss is 0.6249.

B. Sky Segmentation

The second column in Figure 9 illustrates the sky mask we get from U-Net. On average, the model can accurately segment the sky from the picture, especially in scenery pictures. There are some factors that can significantly influence the accuracy of segmentation. First, when the edge gets meandering, or the edge gets very complicated, some pixels near the edge cannot be categorized precisely. As shown in the second row of Figure 10, pixels near the tree were not detected as part of sky. Second, when the gradient of chroma between the sky and foreground has very little difference, the model may give us a bad sky mask, which we cannot use. The first row in Figure 10 shows this phenomenon well.

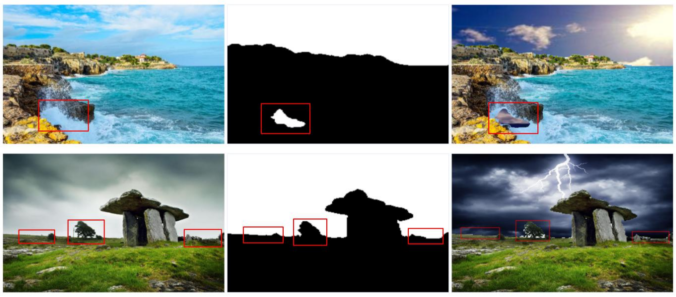


Figure 10:some bad segmentation examples

Even if the sky segmentation is not perfect at all, we indeed can add some post process to make it look better. The guided filter in Section II.C can fix this shortcoming in some degree.

C. Color Transfer

As we've said previously, from the perspective of effect, the chosen algorithm is not the best. But it is less time consuming in video process. Indeed, in most occasions the improved color transfer method can still performs well as Figure 9 shows. However, when the luminance and chromatic difference between the sky and the original picture is too big, the transferred image will look a little strange and have a strong sense of artifact. Figure 11 shows this problem. The original picture was shot in a

Figure 11:bad color transfer

Sunny day. When we replace the sunny sky with a starry dark sky, the foreground should be adjusted as if the photo was taken during the night so that it looks more harmonious. But the luminance and the color of yellow flowers look very strange. The yellow color is still very bright. We can't see this at night. In a word, our color transfer method can't behaves well when solving the uneven illumination exposure problem or when the color style of sky is totally different from the original image.

D. Guided Filtering

The guided filter is used for fixing the problem of micro inaccurate sky segmentation. When we look at the whole filtered picture, we can hardly feel the imperfect of its effect in details.

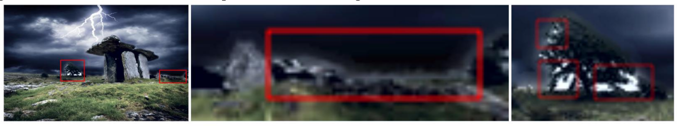


Figure 11:imperfect guided filter

But when we zoom in the area that was filtered, we can see there is still some traces along the edge showing that it was artifact. Besides, when the sky segmentation is not doing well along the complicated edge, the wrong categorized area can be very big versus the filter size. Therefore, even guided filter cannot resolve the big mismatched area.

E. Algorithm Implementation in Video

To implement our algorithm in video process, we need to optimize our algorithm so that the running time can be reduced.

	Before optimizing	optimized
Mask getting	0.22	0.218
Sky replace	1.02	**0.044**
Color transfer	2.37	**0.074**
Guided filter	2.63	**0.246**
total	6.24	**0.582**

Figure 11:time comparison between the optimized and non optimized

In the original algorithm, we used *for-loop* syntax very frequently, by optimizing them with more efficient syntax, the time of sky replacement is 25 times less than before, time of color transfer is 32 times less and the time of filtering is about 10 times less than before. On average, the time of processing reduces from 6.24s to 0.582s. If we apply parallel processing technology in the video process, we perhaps can realize the real-time video processing.

IV. CONCLUSION

In computer vision area, a lot people have done a very well work in sky segmentation. As students, this is the first time that we step into this area and realize the project. We are not doing it without any setbacks. Indeed, there was a very long time that we got stuck and were extremely anxious. But we finally did it after paying a lot of time and energy and got a not-bad result. We know that compared with the work that professional people have done in this aspect, our result still seems awkward and needs to be improved. Therefore, we will keep on learning and exploring more and more possibilities of solving the problems we meet. Many thanks for the guidance we got from the papers, the help in our open source communities. And many thanks especially for Teacher Jia Yan, who led us to the gate of image process. Without the help from all of you, we cannot achieve our goals. At last, many thanks for all of you.

References

[1] Scene Parsing through ADE20K Dataset. Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso and Antonio Torralba. Computer Vision and Pattern Recognition (CVPR), 2017.

[2] Tsai Y H , Shen X , Lin Z , et al. Sky is Not the Limit: Semantic-Aware Sky Replacement[J]. ACM Transactions on Graphics, 2016, 35(4):149.1-149.11.

[3] Ronneberger O , Fischer P , Brox T . U-Net: Convolutional Networks for Biomedical Image Segmentation[J]. 2015.

[4] Long J , Shelhamer E , Darrell T . Fully Convolutional Networks for Semantic Segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 39(4):640-651.

[5] He K , Sun J , Tang X . Guided Image Filtering[J]. IEEE Transactions on Software Engineering, 2013, 35(6):1397-1409.

[6] E. Reinhard, M. Adhikhmin, B. Gooch and P. Shirley, "Color transfer between images," in IEEE Computer Graphics and Applications, vol. 21, no. 5, pp. 34-41, July-Aug. 2001, doi: 10.1109/38.946629.