Sketch2Real

A new photoediting tool for digital alchemy.

Valay Y. Shah
Department of Computer Science
Brown University
Advisor: James Hays

valay_shah@brown.edu

Abstract. Sketch2Real is a system for inserting new objects into existing photograph by simply drawing the sketches of the objects on the photograph. The idea is to free the user from searching through a large number of objects to find a potential object to be inserted into the photograph. Our system provides an interface to select an image to modify and tools to draw a sketch on the image. The user is allowed to draw sketch of objects and the system recognizes the sketch as belonging to a particular object category. It retrieves a few best matching object images based on the shape by searching into the object database for that category. The retrieved objects are then filtered based on the illumination with respect to the background image; the resulting objects are seamlessly blended in the photograph thus making photos more interesting. We compare different algorithms used for shape matching and two different object databases for finding the potential objects.

1 Introduction

What better to express an imagination than a photograph? Photography has advanced in a way that has allowed general public to edit existing photographs, as well as composite novel photographs. Today, there are many softwares, which aids users to manipulate photographs in many different sense. But very few people are aware of their usage because of the complexity. Also, it's an arduous task to edit out small details along the boundaries of the object you placed on the top of the image. Hence a major goal of the computer vision and graphics community is to provide a medium that allows the general public to use their imagination to manipulate photographs or even generate new photographs. Though the field has been progressing but there is a long way to go.

The photography industry has become much advanced, and almost everyone now has a digital camera. Due to the increase in the number of photographs taken every day, there has been a huge growth in the photo storage and sharing websites. Thus, the amount of visual information available is huge but it's still difficult for most people to use the softwares for creation and manipulation of the images. Thus, the main goal of this project is to provide an interface that makes photo manipulation task simple and easy.

Fig. 1: Figure on the left shows a background image to be edited. Figure in the middle shows the background image along with the sketch of the object to be inserted into it. Figure on the right shows the final result after compositing.

In this paper, we present a system that allows naive users to create novel visual content by leveraging the amount of visual information available these days on

the Internet. We propose a system of sketch based object insertion. The idea is to provide the user with an interface, which allows him to select a photograph to insert objects into it. Then user simply draws a freehand sketch of the object at the position where he wants to place that object. The system recognizes the sketch and retrieves the best matching object photographs from the object library and seamlessly composite those into the photograph to provide a ranked output. Several compositions are automatically generated and provided to the user to choose from. Figure 1 demonstrate the general idea behind this project.

2 Related Work

There has been much work related to generating novel images and its a well-studied area. Perez et al. [9] showed how to seamlessly blend images by interpolating the gradients along the boundaries. Jia et al. [10] devised an algorithm to compute optimised boundaries around the source patch and target image used for compositing. Wang and Cohen [12] showed how simultaneously optimizing matting and compositing task produce successful matte for compositing. All of this work assume that you have a fixed source image or patch you want to blend on the target image.

A couple of previous works motivated our project. Our sketch recognition part is motivated from and uses the work of Mathias et al. [4], while the idea of automatic object search and blending has been motivated from the work of Chen et al. [1]. Chen et al. [1] in their Sketch2Photo paper generates a novel photograph based on sketches of objects along with their text labels. They collected data from the Internet based on text labels describing the background as well as the sketches for the scene items. Then they extracted the objects from images collected from the Internet and seamlessly composite them with the background image. We differ from Chen et al. in three ways. We gave freedom to users from specifying text describing the sketch as we depend on the sketch recognition part for to get the sketch category; we used two different object databases Label Me and ImageNet for object selection instead of relying on a huge amount of unfiltered Internet images. We allow the user to modify their own images rather than the algorithm selecting a background image for you.

From a naive users perspective, Sketch2Photo system does not allow them to manipulate their own photos. Thus, it's only useful to artists or graphics designer, who want to generate novel photographs by drawing sketches while not being useful for the general public who wanted to manipulate their existing photographs in order to make them look more appealing. Why would anyone want to generate a photograph with random objects (e.g. different people) in it? While all the fancy things are done by the system, the user's role is to just draw sketches.

Our system allows users to be creative and lets their imagination be converted to real. Hence users can manipulate their own photographs by inserting objects in it.

Photo Clip Art [2] is another closely related system, which manipulates existing images by adding objects into the photograph. In this system user has to go through a huge list of object images and select the object to be inserted into the photograph while also specifying the position to insert the object at. Our system is in a way related to Photo Clip Art with the difference that instead of dragging and dropping objects in photograph, our system allows user to draw sketch of an object to be inserted into the photograph and the object will be inserted at position where the sketch has been drawn. The principle advantage of this system is that the object to be inserted will be similar in shape and size to the sketch of the object (also color future work). Hence it frees user from looking into a database for the object that best matches his preference.

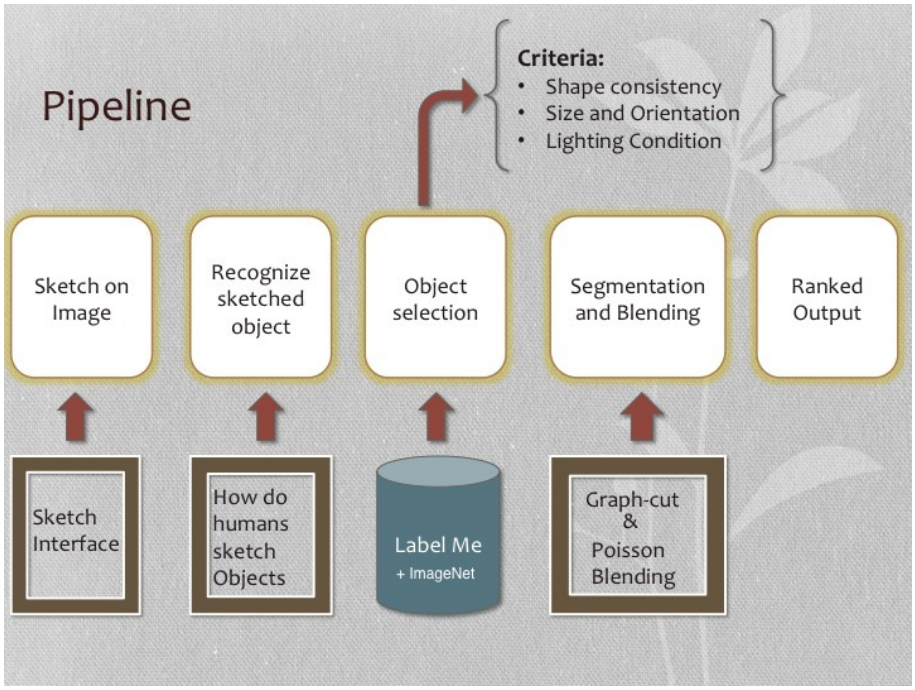

Fig. 2: The Sketch2Real Pipeline

3 Overview

In this project, we propose a system for editing existing photographs by inserting objects. The goal of this project is to provide users with a tool to generate novel images using their imagination and to shield them from the arduous task of compositing images. It uses a background image, user drawn sketches of the objects and their respective locations as the input. Our system then performs tasks like sketch recognition to figure out the object category, shape matching to find the object which best matches with the sketch, illumination context matching to re-arrange objects with similar illumination to the background image to the top. After getting the best matching objects, it seamlessly blends the objects on the background to create a novel image. The outline of the idea is shown in figure 2, along with the algorithms and database we used to perform each task.

The user is provided with a sketch interface program, which allows the user to select a background image and supports sketching on the selected background image. User then draws sketches of the objects to be inserted. The system will recognize the sketch category with our sketch recognition part and search the library of that category for the objects images. The retrieved objects images are later filtered based on shape, size, and orientation of the object as well as the illumination context between objects original image and the background on which it is to be inserted and this is followed by compositing part. Our system uses Poisson blending as described by Perez et al. [9] for seamlessly blending an object with the background.

4 Sketch Recognition

Chen et al. [1] in sketch2photo uses the text labels to search for the object images from the Internet. In this project, we free the user from labelling the object as well as from specifying the task the object is performing. We did that by introducing sketch recognition system into the pipeline. Our sketch recognition part is based on the paper How do human sketch objects? [4] and uses the sketch database created by them.

The sketch database has 20000 sketches divided into 250 categories. So each category has 80 sketches. Once the user has drawn sketch on the background image, the system starts recognizing the sketch as belonging to one of the object categories. Currently, we restrict users to draw sketches of objects belonging to these 250 categories. To determine the sketch category, we computed shog features with soft encoding for each sketch in the database and then used those to train our one vs. all binary SVM thus training 250 SVMs individually for each object category. We used radial basis function with Gaussian Kernel and the parameters we used were = 17.8 and C = 3.2 which are the best parameter

Fig. 3: Sketch Recognition: Sketches of elephant and airplane are correctly classified, while a motor-bike is incorrectly classified as a bicycle.

as suggested in the paper [4]. When we encounter a new sketch, we compute shog features on it and run it through all the 250 SVMs. The sketch is given a classification score from all 250 classes and the category with the highest score is assigned to the sketch.

We found that the sketch recognition accuracy using the SVMs was roughly 59%. But we also noticed that 80% of the time the correct sketch category appeared in the list of top five recognized categories and roughly 87% of the time in top ten. So in order to get the correct category as the output we consult the user to select the correct category from the top ten recognized categories. With the addition of more sketches and more categories, we believe that the sketch recognition part can be more accurate. Figure 3 demonstrate the sketch recognition part of the project.

5 Creating an Object Library

One of the important part of this project is to find few best matching object images based on the sketch and its category that is obtained from the sketch recognition part. Sketch2photo [1] uses the Internet and search for object images based on the text label and a verb provided by the user that describes the action.

As our project does not use any text labels specified along with a sketch, our algorithm is solely based on the sketch for finding matching objects. So to reduce the search space we perform sketch recognition. Also instead of searching for objects on the internet we create an object library to store the objects. Figure 4 shows some of the objects in the bird category in our library.

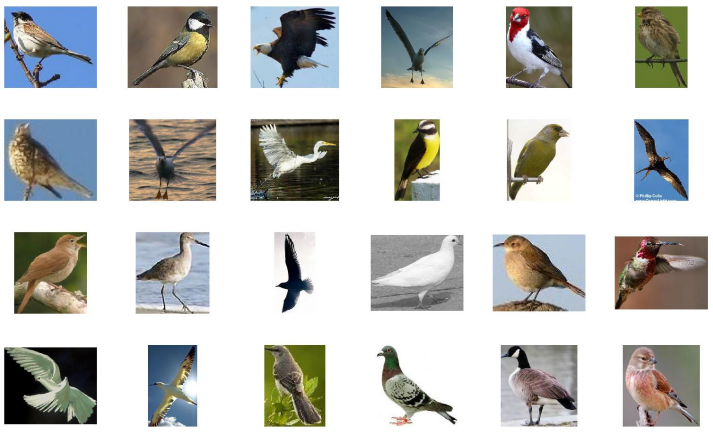

Fig. 4: Object Library: this figure shows some sample object images in the bird category.

Photo Clipart [2] uses LabelMe database to create their photo clip art library because LabelMe provides segmented and annotated objects. So we initially used LabelMe database to create the object library. But we soon realised that LableMe has very limited number of objects in most of the categories as penguin, flying birds, zebra, cannon, etc. while having huge number of objects in categories like car, trees, building, etc. It suggests that LabelMe database is more biased towards outdoor images such as streets and roads. Thus, we focused our attention to ImageNet that is much larger and a richer dataset.

Although ImageNet has lot of images for each of the object category, it does not have segmentations. But the good thing is it provides bounding boxes for most of the object categories we have, and most of the images have foreground objects in focus with a blurred background. Thus, we can use grab cut segmentation using the bounding boxes to extract foreground (objects) out of the background. We created object library from LabelMe as well as ImageNet that consist of segmented objects along with their masks. We will discuss the two image databases we use to create this object library.

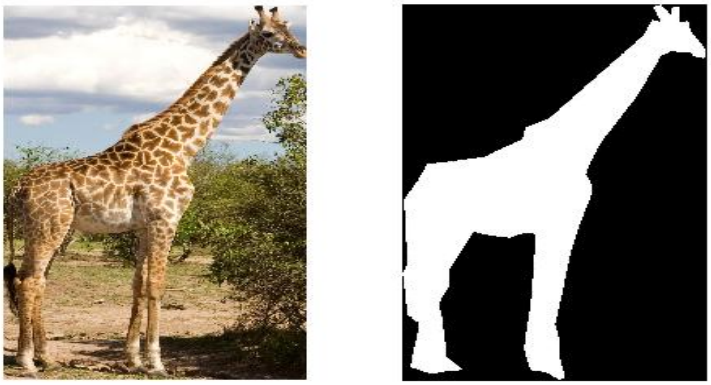

Fig. 5: LabelMe: An object with corresponding segmentation mask . The segmentation and annotation in LabelMe is crowd-sourced. This image shows the goodness of the segmentation performed by the crowd.

5.1 LabelMe

As the project is completely data driven we needed a large dataset of object images. Label Me [3] data set contains a huge set of annotated images and it's easy to get the crops of the object from user segmentation. We queried Label Me database for 250 object categories and created an object library for those categories. The LabelMe database has reasonably good segmentation of objects but there are other problems like synonyms, occluded objects, cast shadows, etc. So we filtered out objects whose annotation says occluded and shadowed. We filtered out those objects that are small in dimension after segmentation. We also filtered out manually the objects having occluded boundaries and were under shadows but were not stated so in their annotation. Finally, the library encompassed object images along with their masks to be used in the project. As we mentioned above LabelMe has a limited amount of objects in most categories, we moved to a different image database ImageNet [11]. A good segmentation of giraffe performed by the user has been shown in figure 5.

5.2 ImageNet

ImageNet [11] is a database of images collected and arranged according to the Wordnet hierarchy. They have an average of 500 images per category or node in the word net. Having lot if images for each object category along with the bounding boxes, ImageNet was an ideal choice for this project.

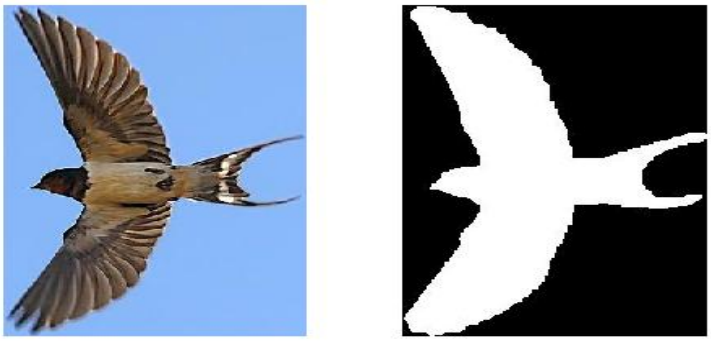

Fig. 6: ImageNet: An object image with corresponding alpha matte. The alpha matte was genereated using Grabcut algorithm with bounding box as a prior.

We queried ImageNet to download images of objects for each of our object categories. We used the bounding boxes on the images to provide them as input to the Grabcut algorithm along with the corresponding images. Grabcut [14] is an efficient tool to seperate foreground object from an image. It is an iterative algorithm based on graph-cut [15] technique for segmentation and uses color and edge contrast information for segmentation. This algorithm gives us an alpha matte for foreground and background seperation. Figure 6 is an example of alphe matte generated by the Grabcut algorithm. After getting the segmented objects, they are filtered based on size.

6 Object Filtering and Selection

We searched into the object library to find object images that best matches the sketch and filter out the irrelevant images. The criteria for best match were based on three things (i) Shape consistency, (ii) Size and orientation, (iii) Lighting condition. The first two criteria were taken care by our algorithm for Shape Matching while we perform Illumination context matching to find an object with correct lighting condition as of the background image. We will now discuss the algorithms we used for object filtering.

Fig. 7: A sample sketch

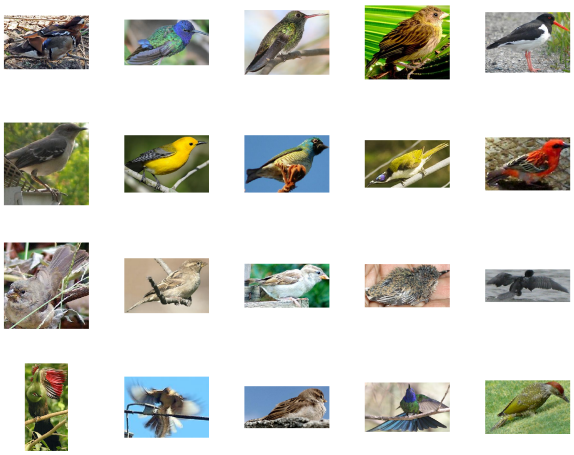

Fig. 8: Shape Matching: The corresponding object images found by the system. Most of them closely matches the shape of the sketch given in figure 7.

6.1 Shape Matching

One of the important part of the system is to find the best matching object in terms of shape. The core idea is that the object should match in terms of shape to the sketch as close as possible. For example, a flying bird will be completely different in terms of shape from a standing bird. In this paper [5], they performed sketch based image retrieval by finding the scenes that best matches the sketch in terms of shape. There can be many features used to perform shape matching as Gist [6], HoG [7], but [5] have shown that SHoG works the best for shape matching. But we found that for finding objects from a limited set of images in a particular object category gist works pretty well. We also compared the results of HOG, Gist and Local Self Similarity descriptor [13], but gist comes out to be the best. In this project, we used Gist descriptor. We apply canny edge detector averaged over multiple thresholds and sigma, on the crop of the objects and compute gist descriptor on edge detected object crops as well as the sketch. Sum of Square Difference method is used to compare the descriptors and to find the error scores. At last few objects with minimum error score were selected. The figure shows Gist descriptor works well in general. The size and orientation matching were implicitly covered with the Shape Matching part. As there are a large number of objects in the library, its not difficult to find objects that best matches the sketch in terms of size and orientation. Figure 7 and 8 demonstrate the shape matching part. Most of the objects in figure 8 has similar shape to the

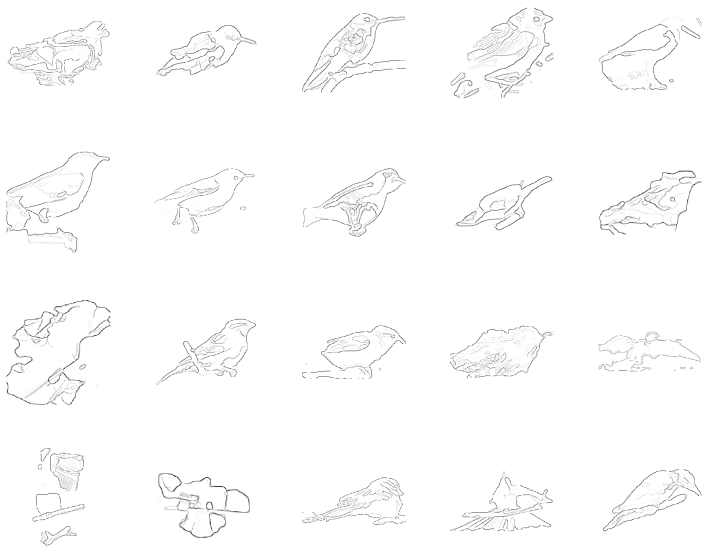

Fig. 9: Shape Matching: This figure shows the edge detected boundary images corresponding to the images in figure 8. These boundary images are used to perform shape matching with the sketch.

sample sketch in figure 7. Figure 9 shows the corresponding boundary images used for shape matching.

Fig. 10: The swan inserted in this image has similar lighting condition to the background image.

Fig. 11: This image looks incorrect due to mis-match in the lighting conditions of the swan and the background.

6.2 Illumination Context Matching

Its really important that the lighting conditions of the object be same as the lighting conditions of the background image. Figure 10 and 11 shows that the photo looks incorrect if there is a large difference between illumination of object and the background image. We followed the approach similar to [2] that is to compute a coarse environment map from a single image. The idea is to generate a rough 3D structure of the scene as done by [8] and use this to estimate lighting condition of each surface (ground, vertical and sky) independently. We used geometric context of Hoiem et al. [8] to estimate three major surface types i.e. sky, vertical and ground as shown in figure 12. Illumination for each surface is computed as joint 3D histogram of pixel colors in CIEL*a*b* color space. Thus three histograms are generated for each channel. Though the illumination context is too global, it works quite well. Also as we have a large number of objects, the probability of finding an object having similar lighting condition to the background is very high. We compare illumination of the background scene with the illumination of the scene from which the object is cropped using the standard chi square distance between the histograms. These object images are those discovered from shape matching part. Finally we have few filtered objects ready to be seamlessly blended in to the photograph.

7 Compositing

After re-filtering the objects based on lighting conditions we composite the object and the background scene to create a novel image. As we have object crops and their corresponding masks available from object library, its not difficult to perform blending. Different methods are available for blending images as alpha blending, Poisson blending [9], drag and drop pasting [10], etc. We used Poisson blending in this project.

7.1 Poisson Blending

Since the illumination context matching part takes care of matching the lighting condition objects with the scene lighting conditions, Poisson blending performs pretty well in this process. We followed paper from Perez et al. [9] for this part. Poisson Image editing is a generic machinery based on solving the Poisson equations with Dirichlet boundary conditions. The basic idea is to interpolate between the gradients at the boundary between the source and the target image. It works well if there is not much difference between object and background scene in terms of color. The result of Poisson blending is shown in the figure 15 as compared the naive pasting in figure 14.

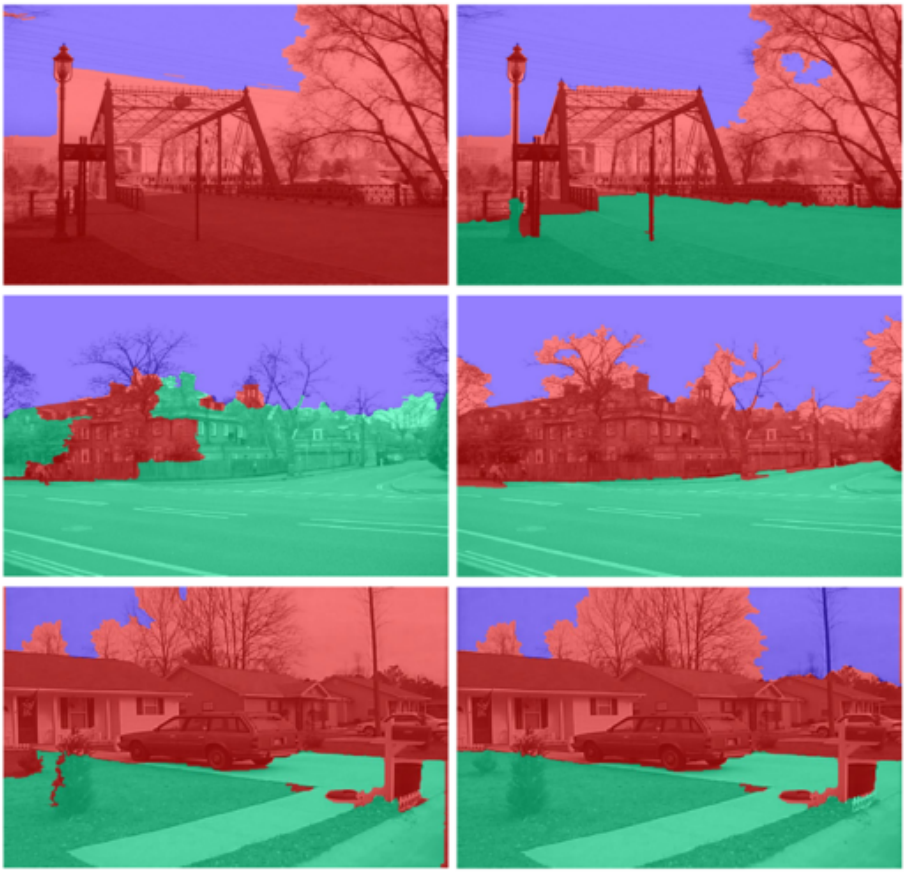

Fig. 12: Geometric Context: All images are divided into three major surfaces (sky, vertical and ground) as shown in figure.

8 Results and Discussion

Figure 15,16 and 17 shows result of sketch based object insertion. After filtering the objects based on the shape, size, orientation and lighting conditions we composite the object onto the background scene as shown in the figure. We then ask the user to select best out of the ranked outputs. Figure 18 shows some of the failure cases of the project. These are mostly due to artifacts of Grabcut algorithm, color bleeding in poisson blending and unable to find objects with matching lighting with the background.

Fig. 13: Compositing: An example image with a sketch of a flying bird

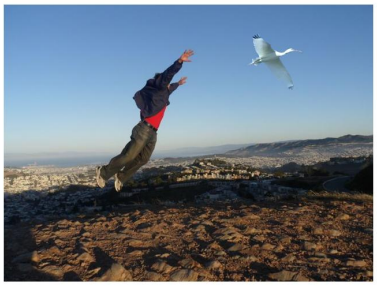

Fig. 14: Naive Pasting vs Poisson Blending: The Image on the left is generated by naively pasting the bird on the background image while the image on the right have the bird seamlessly blended onto the photograph using Poisson blending.

9 Future Work

There is a lot more progress to make on this project. There are many possible future directions. Starting from sketches, we can include color information in sketches to match them better with objects and retrieve objects with a similar color as the sketch. It will add one more dimension of information into consideration and will improve object retrieval. These days hand-held devices like mobile phones has many apps that allow to draw sketches with color so we can leverage color information for better object selection. It shows that the system is portable for the hand-held device easily.

The other thing is to improve the performance of the sketch recognition system. Currently, due to a limited number of sketches into each category, our system is limited to the shapes of the sketches we have in the database. Thus, adding more sketches of different view point of the objects and collecting large number

of sketches for each category will improve the performance of sketch recognition.

Also, various other descriptors can be tried for sketch recognition as well as for shape matching so that the performance of the system can be improved. A local metric on estimating lighting condition of the object is also helpful.

Finally, we can use other techniques for segmenting objects perfectly from the Label Me database. The annotation's boundaries are very rough and dont seem to be much useful. Hence grab cut segmentation can be used for finding good segmentation boundary, which aids Poisson blending. We can also combine the objects from these two different categories to create a large object library. Also hybrid compositing techniques such as alpha plus Poisson blending can be used to blend images seamlessly. Thus, there are many directions but we believe that this project idea might be useful for an artist as well as a general public for expressing their imagination.

10 Acknowledgements

We would like to thank Mathias Eitz for sketch recognition code and other useful insights on this project. We thank Antonio Torralba and Derek Hoiem for their code on Gist descriptor and Geometric Context from a single image. And special thanks to Krishna Nand K. for his comments and insight on various parts of the project.

References

1. Tao Chen, Ming-ming Cheng, Ping Tan, Ariel Shamir and Shi-min Hu, Sketch2photo: Internet image montage. ACM SIGGRAPH 2009.
2. Jean-Francois Lalonde, Derek Hoiem, Alexei A. Efros, Carsten Rother, John M. Winn, Antonio Criminisi: Photo clip art. ACM Trans. Graph. 26(3): 3 (2007)
3. B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman, LabelMe: a database and web-based tool for image annotation. MIT AI Lab Memo AIM-2005-025, September 2005.
4. Eitz, Mathias and Hays, James and Alexa, Marc, How Do Humans Sketch Objects? ACM Trans. Graph. (Proc. SIGGRAPH) 2012.
5. Eitz, Mathias and Hildebrand, Kristian and Boubekeur, Tamy and Alexa, Marc, Sketch-Based Image Retrieval: Benchmark and Bag-of-Features Descriptors. IEEE Transactions on Visualization and Computer Graphics 2011.
6. Aude Oliva and Antonio Torralba, Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope International Journal of Computer Vision 2001
7. N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. InCVPR, 2005.

8. D. Hoiem, A. Efros, and M. Hebert. Geometric Context from a Single Image. In ICCV, 2005.
9. Patrick Perez, Michel Gangnet, Andrew Blake, Poisson image editing, ACM SIGGRAPH 2003 Papers, July 27-31, 2003, San Diego, California
10. Jiaya Jia, Jian Sun, Chi-Keung Tang, Heung-Yeung Shum, Drag-and-drop pasting, ACM SIGGRAPH 2006 Papers, Boston, Massachusetts.
11. Deng, J. and Dong, W. and Socher, R. and Li, L.-J. and Li, K. and Fei-Fei, Li, ImageNet: A Large-Scale Hierarchical Image Database, CVPR 2009
12. Wang, J., and Cohen, M. 2006. Simultaneous matting and compositing. Tech. Rep. MSR-TR-2006-63.
13. Eli Shechtman and Michal Irani, Matching Local Self-Similarities across Images and Videos, IEEE Conference on Computer Vision and Pattern Recognition 2007 (CVPR'07)
14. C. Rother, V. Kolmogorov, A. Blake. GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts. ACM Transactions on Graphics (SIGGRAPH'04), 2004
15. Vivek Kwatra and Arno Schodl and Irfan Essa and Greg Turk and Aaron Bobick. Graphcut Textures: Image and Video Synthesis Using Graph Cuts, ACM Transactions on Graphics, SIGGRAPH 2003

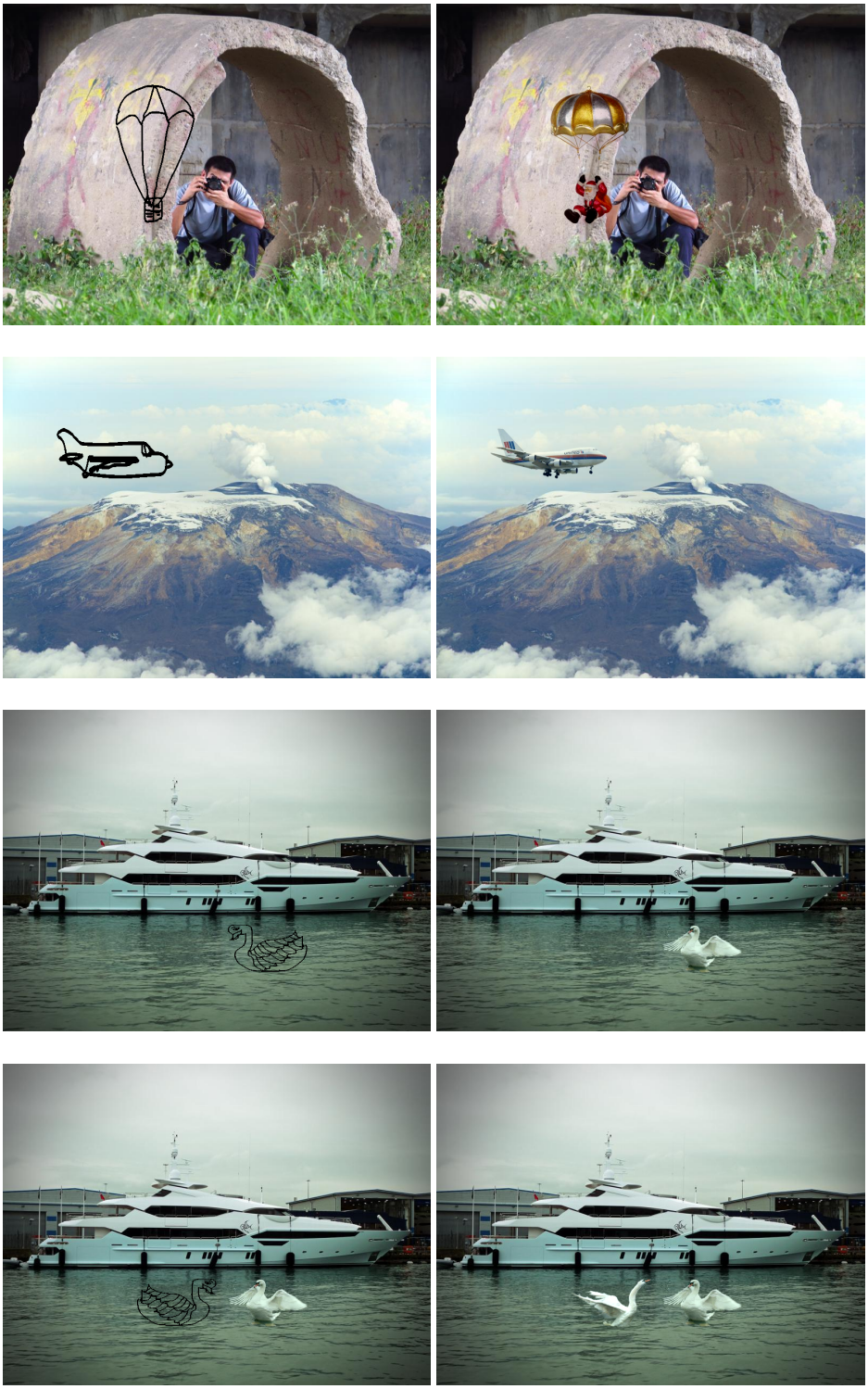

Fig. 15: Sketch2Real: The figure shows some of the novel images generated by our system.

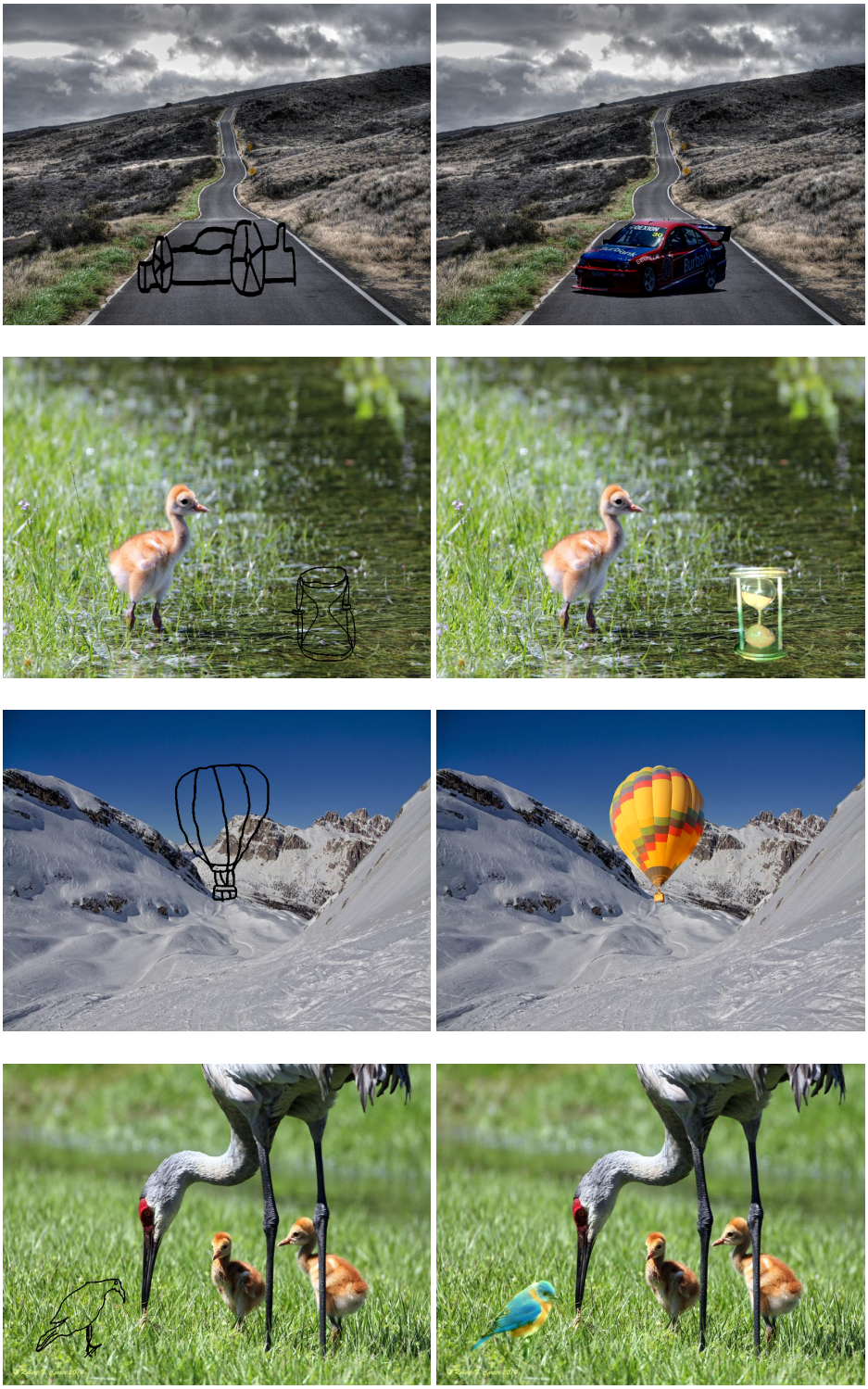

Fig. 16: Sketch2Real: The figure shows some of the novel images generated by our system.

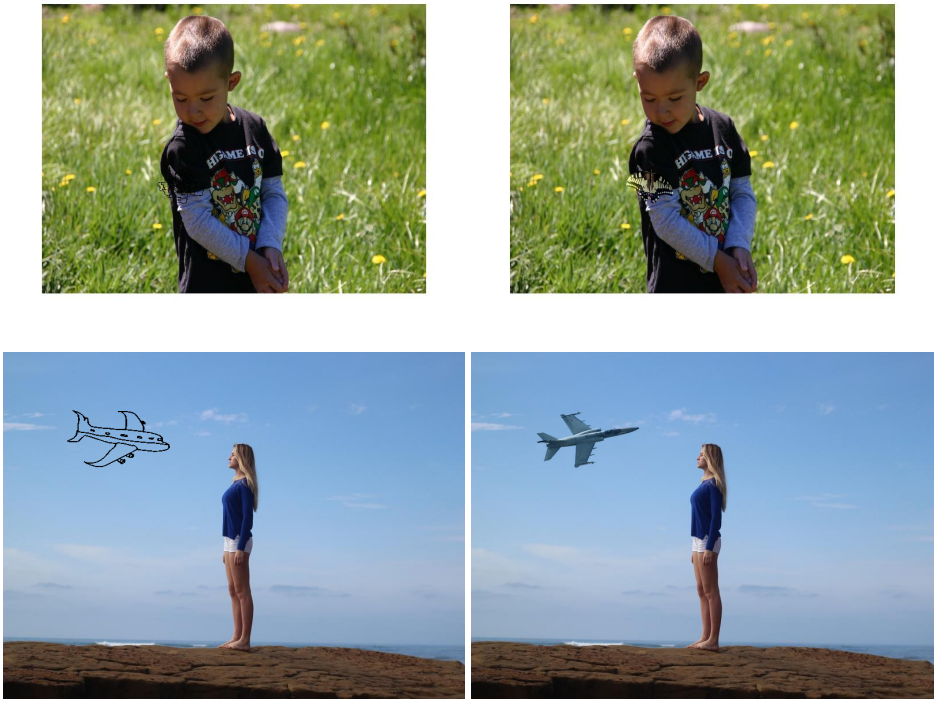

Fig. 17: Sketch2Real: The figure shows some of the novel images generated by our system.

Fig. 18: Sketch2Real: The figure shows some of the failure cases. These are the cases, where our system is unable to find object images with similar illumination to that of background. The space shuttle in the bottom right image is not segmented correctly which is a disadvantage of Grabcut algorithm. Also poisson blending has a disadvantage of color bleeding that is visible in the middle image.