MMAI Final Project - Peeping Montage

Che-An Lu, Chin-Hui Chen, Yu-Ta Lu

Abstract

In this project, we implement a broadly used art "image mosaic", and we use this popular method in different ways to create some visual art pieces which are different from traditional mosaic images. We also propose methods to create mosaic movies as well as image composed of movies. We then experiment on these methods using different size of dataset to see the results. Finally, we propose some prospective future works that provide good ways for further researching.

Index Terms

MMAI, mosaic, montage, art.

I. Introduction

The idea of our project, Peeping Montage, comes from image mosaic. Image mosaic is a technique that help create an image composed of other images. In this project, we will implement the image mosaic technique, and then extend this idea to other two parts, which are creating a movie composed of images and creating a image composed of images. If we see into the detail of the output, we can see there existing many meaningful images or movies. That is why we use the word "peeping". We will elaborate our methods in detail in the following sections.

II. Problem Definition

We can separate our project into three parts:

A. Creating a Mosiac Image

Input an image I_{in}, then output an image I_{out} that is composed of many small images selected from a prepared image dataset. I_{out} looks almost the same as the original image when looking at it distantly enough.

B. Creating a Mosaic Movie

Input a movie M_{in}, then output a movie M_{out} that is composed of many small images selected from an prepared image dataset. M_{out} looks almost the same as the original movie when looking at it distantly enough.

C. Creating A Dynamic Mosaic Image

Input a movie M_{in} and an image I_{in}, then output an image I_{out} that is composed of many parts of the input movie M_{in}. I_{out} looks almost the same as the original image when looking at it distantly enough. In addition, when looking close to I_{out}, you can see many parts of the input movies playing.

III. Methodology

We first focus on describing the method of creating a mosaic image composed of many small images in the following parts. Then we extend the ideas to devise more applications.

A. Notation

Before diving into the methodology, we first define the notations used in the following text. As we are using many small images to compose another image as the effect of mosaic, we have a corpus, C, that stores all the images we have. Moreover, for an image I, we denote the 3 color channels over rgb color space of the pixel of the i-th row and j-th column to be $I_r(i,j)$, $I_g(i,j)$, and $I_b(i,j)$.

B. Indexing

In this part, we aim to cluster images of similar average colors together.

1) Building Index: We build an index file for C in order to provide faster searching speed when processing. For each $I \in C$, we take the average color over all pixels of the images. That is,

$$Avg_{I_r} = \sum_{i,j} I_r(i,j)$$

where Avg_{I_r} is the average color of all pixels on r channel. Avg_{I_g} and Avg_{I_b} is computed in the same way.

After we take the average over all images, we build an index of color which maps a color to a set of images. For example, if two images $I_1, I_2 \in C$ have the same average color ($Avg_{I_1r} = Avg_{I_2r} = r_1, Avg_{I_1g} = Avg_{I_2g} = g_1, Avg_{I_1b} = Avg_{I_2b} = b_1$), then the entry for $r1, g1, b1$ in the index will point to these two images. We build the list for all different r, g, b values, each of which ranging from 0 to 255. However, having so many entries seems unreasonable, for images look alike when their colors are not too distant from each other using the Manhattan Distance metric. Therefore, we do color quantization to avoid such problem.

2) Color Quantization: Instead of having entries for all r, g, b values, we first quantize the average color of an image to a certain number of bins and then build index only over these bins. Color quantization prevents similar images from being distributed to different entries and thus provides more choices when we want to find the set of images of a specific kind of color (which will be described later).

3) Image Interpolation: We need at least one image in each entry for later use. Therefore, for those entries not pointing to any images, we have to assign some images to the entries. We solve the problem by copying the images in the nearest (still, we use Manhattan Distance metric) entries to this currently empty entry. Thus all entries point to at least one image in the index.

C. Creating A Mosaic Image

Now that we have the index for C, we can create a mosaic image for a given input image I. For each pixel of I, we quantize the color of that pixel using the same quantization method when indexing, and then find the set of images that is pointed to by the entry in the index corresponding to this quantized color. We randomly selected one of the images in the set and put this image to the position of the pixel. Finally, we generate the resulting image according to this mapping of pixels to images. The reason of random selection is that we don't want the image to be too vapid. For example, suppose that the input image has only one color. By selecting a image randomly, we can make the resulting image be composed of different images that have the same quantized color as the color of the original image.

D. Extensions

1) Creating A Mosaic Movie: For an input movie, we slice the video into frames and adapt our mosaic method to each frames. We then combine these frames back to video as output.

The remained issue is continuity among frames. We created first version of our video, but the result is quite poor. Human can hardly feel comfortable about the rapid image stream. The reason is that the same color block in the n-th frame and the $n+1$-th frame use different images and will make the movie "jump" from frame to frame, resulting in uncomfortable visual effects when watching. In order to solve

this problem , we keep the former mosaic information when computing the mapping of the current frame. If a pixel of the current frame has the same color as the corresponding pixel of the previous frame, we don't select another small image to fit in this pixel. Use these information, we can solve the problem of discontinuity between frames.

2) Creating A Dynamic Mosaic Image: We want to create an image composed of many movie shots from a certain movie. Therefore, we use shot detector to extract movie shots of the movie first. We then build the index using the key frames of all the shots of the movie. Now we can deal with the input image. We build the map from image pixels to these key frames just as when creating a mosaic image do. With this mapping, we can put the shots to the position in the map where the corresponding key frames reside. That is, for example, we put a shot to a pixel of the image if that pixel's color is like the key frame's color representing this shot. With the assumption that the frames in the same shot have similar colors and therefore are appropriate to be put on the same position of the image.

The main problem is "how to implement". Because each image pixel is replaced with shot, there are many video data in one image. Since we have no quite powerful computer to display this dynamic image, we use an alternative method to display the result. We compress each shot into a gif file and then scrape these gif files together. We write a Web UI to demonstrate our work.

IV. Experiment

A. Dataset

As we need thousands of images to be our mosaic image pool, we choose Google Image and Flickr to be our data source. First, we choose Google Image API. we manually adapt some keywords to search for specific class images , such as "Spring" and "Fall." But, unfortunately, the images from Google search is quite poor. There are a lot of noise which can be classified into 3 catalogs. The first one is the problem of image resolution. The second, there are many icons and symbols. The the last, we also have some gif animations. In order to solve these problems , we choose the latter one data source, Flickr. Someone developed phpFlickr, a kind of Flickr API , which helps us download thousands of images. The image quality from Flickr is quite well. The only remained problem is that we use a lot of disk space! Finally, we crawl approximately 35000 images, consisting of 70 classes with each class contains 500 images.

B. Implementation Details

First we use only 5000 images in dataset, and we implement it with Matlab. The processing time is pretty long (about 10 minutes per image), and the result is not so good due to lack of image data. After we increase the size of dataset to be 35000, which is 7 times to our original dataset, and use c++ to implement it, the process time is much faster than Matlab (about 10 seconds per image), and the result is better than the previous ones.

C. Mosaic Image

We experiment our method on different kinds of images, including photos(1), paints(2), and drawings(3). The results are nice and the small images are random enough to make the big image more like masaic.

D. Mosaic Movie

We adapt the image dataset (35000 images) and download videos from YouTube.COM and from MMAI homework 1 dataset(I). We Slice video into frames and set fps 29 in the output.

The result of the first video can be viewed on the web site:
`http://ir.csie.ntu.edu.tw/~chean/mmai08/video_of_images.avi`

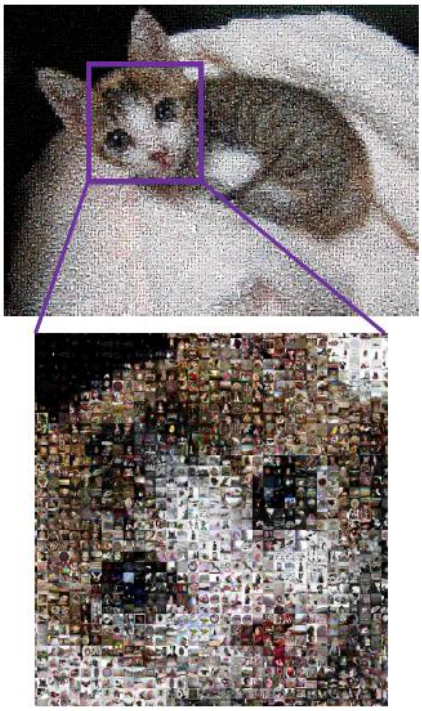

Fig. 1. Mosaic on Photo

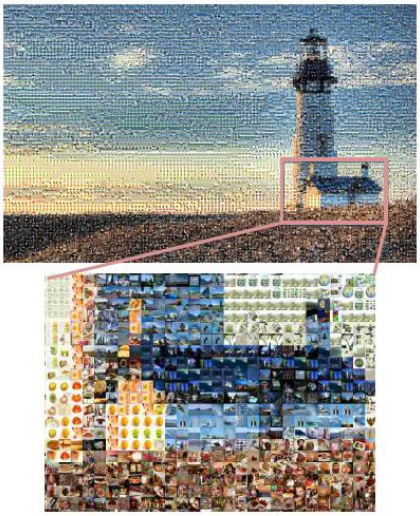

Fig. 2. Mosaic on Painting

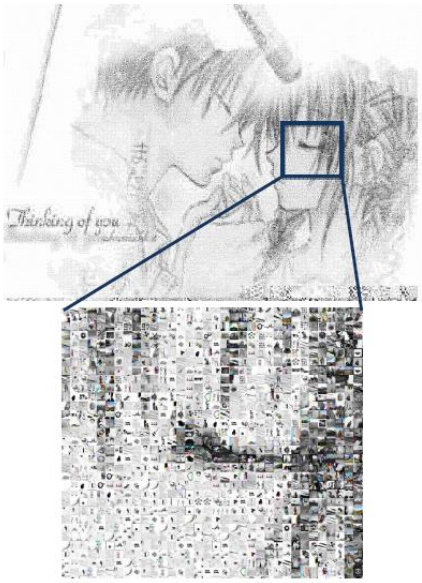

Fig. 3. Mosaic on Drawing

No	Video Name	Source
1	90-Second Animation For Global Day of Action	http://tw.youtube.com/watch?v=IOAtbWHWJqk
2	Nice Pants! Lego Animation	http://tw.youtube.com/watch?v=tlkd45W4TWU
3	A penguin's joke	http://tw.youtube.com/watch?v=YnC_HiUFZQs
4	Bruno Bozzetto animation	http://tw.youtube.com/watch?v=ms3UFVmUycY
5	Yellow Lemon Tree - Der Clip	http://tw.youtube.com/watch?v=hTTyYHScsck
6	07.mpg	MMAI homework 1 dataset

TABLE I
VIDEO LIST IN OUR EXPERIMENT

E. Dynamic Mosaic Image

First, we choose "Hoodwinked" movie (http://www.imdb.com/title/tt0443536/) to be the movie for shots. We extract 1768 shots from the movie and use our method to build a dynamic mosaic image. The way we present the final result is that we convert each shot to a gif file, and create a web-based UI listing thousands ofsuch gif files to make up the image. Because the gif files are compressed to small size, the computation complexity is quite less. However, the result of this part is not so good as we expect, because we use the movie's shot to be the dataset and only found 1768 shots in our input movie, the number of different colors is not abundant enough to substitute the pixels of the original images and will make the resulting image look different from the original one. Using movies of more shots may reduce such difference. The result is at our demo site (http://ir.csie.ntu.edu.tw/~chinhui/mmai/) - Peeping Montage - (Motion-Static).

V. FUTURE WORK

First, we can try more to speed up the processing time of our current work, such as implement other efficient indexing methods and compare the pros and cons of them. Second, we can create movie composed of movies. This may need some additional algorithms to solve problems such as discontinuity over frames

and finding best sequence of small frames for a specific position of a sequence of frames of the original movie. Finally, we can implement image mosaic with different size of image data set, in which we analyze the input image first and then decide which parts of the image is more important. With the knowledge of the important part of the input image (ex: peoples face or buildings), we can put many small images in these parts to make the details clearer. In the contract, we put few big images in the less important part (ex: background). Therefore the output image will emphasize the important part and blur the less information part of the input image.

VI. Conclusion

Although the idea of montage art has been proposed for a long period of time, however, with the new application over dynamic objects such as movies, creative art pieces can be designed. More kinds of visual presentation of images will be devised. Our work provides a preliminary view into such issues. Though there is still space for improvement, we believe that this work is prospective and may be popular in the future.

References

[1] Irani, M. and Anandan, P., 1995. Mosaic based representations of video sequences and their applications. ICCV
[2] Li, Z. and Yee-Hong, Y., 1999. Mosaic image method: a local and global method. Elsevier Science B.V., 1421-1433

Team Members

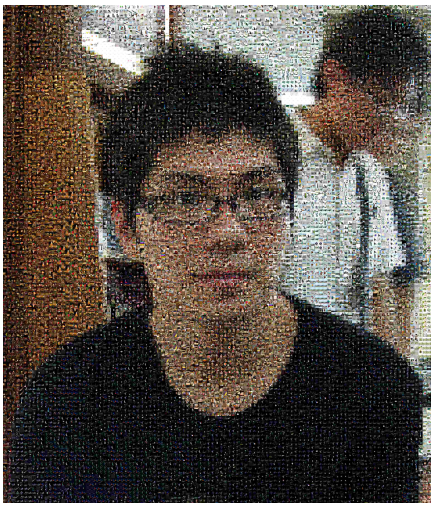

Fig. 4. Che-An Lu

Fig. 5. Chin-Hui Chen

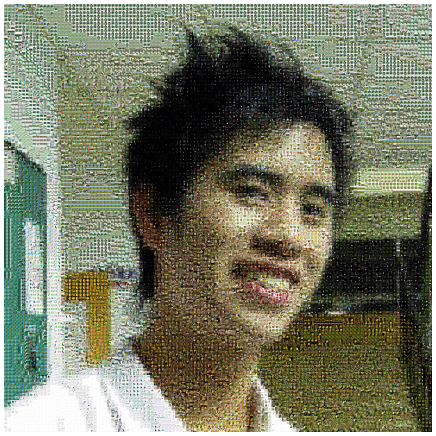

Fig. 6. Yu-Ta Lu