Discovering Panoramas in Web Videos

Lingfeng Huang lhuang58@wisc.edu

Fang Wang fwang64@wisc.edu

12/2016

Note

Our website is available here. lhuang58.github.io
Due to the file size of video test segments, we are not able to upload those to the Learn@UW. We send the video segments to you by email. The name of the video corresponds to the Figure number in this report.

Abstract and Problem Statement

Panoramas have been widely used in may applications in multimedia, but the main constraint for panoramas is that they must be taken by people who physically present at the place. In this project, we implement our version of Discovering Panoramas in Web Videos by Liu et al.[1]. Our goal is to solve two problems: First, the program should be able to select optimal segments within a given web videos to synthesis a panorama using visual quality measurement introduced in Liu et al.[1] Second, the program should be able to generate a set of panoramas if multiple panoramas can be synthesised in a given video. The whole procedure is a optimization problem where we optimize the three criteria which are wide field of view, mosaicability, and high image quality.

1 Introduction

The emerge of the idea "Panorama" has to be dated back to early 20 A.D. and was a means of generating an 'panoptic' view of a vista[2]. Nowadays, with the help of advancement of technology, people are able to create desired panorama by simply rotating their cell phones and clicking the shot button. The process of synthesizing panorama is relatively straight forward. First step is to take successive photos from the same optical center. The next

step is finding the alignment between each image and warping accordingly, and the final step is interpolating the warped image and applied certain blending to remove the visible seams. However, the problem with creating panoramas using above approach is that people are required to physically appear at the place where they take the images, which means if people want to take a panorama of Time Square in New York, they have to fly over New York to do so.

Compared to a sequence of images, videos are still shot from the same optical center and cover a wide field-of-view although some segments within videos have relatively low image quality and there might also have moving objects. Liu et al.[1]suggests an approach that synthesizing panoramas by identifying proper segments within videos as panorama source. They convert the problem into a optimization problem, and set up three constraints in order to evaluate the video segments.[1] ei al. indicates that in order to be a appropriate panorama source, a video segment should cover a wide field-of-view based on the definition of panorama imagery, be "mosaicable" and the frames should have high image quality[1].

2 Visual Quality Measure

We measure the visual quality of a single frame based on two terms, one is the incorrectness of the motion model $E_{vm}(S_i)$ and the other is the source image visual quality $E_{vv}(S_i)$. Then by setting up the visual quality distortion $E_v(S_i)$, we can obtain the visual quality measure.

$$E_v(S_i) = \alpha_m E_{vm}(S_i) + \alpha_v E_{vv}(S_i) \tag{1}$$

By default, we set both weights α_v and α_m to be 1.0.

2.1 Source Image Visual Quality

The source image visual Quality $E_{vv}(S_i)$ is defined as how blurry and how blocky the image is. We use the idea from Tong et al.'s [4] method of measuring blurring artifacts by using Haar Wavelet Transform. Tong et al. stated that by utilizing the ability of Haar Wavelet Transform, their method can be used not only on judging if an image is blurred or not, but also on determining the extent of blurriness of an image. Their method is discussed firmly in their paper and we will skip this part for now.

The blockiness is measured by using the method of Wang et al., which estimates the average difference across block boundaries modulated by image activities [3]. They came up with a computationally inexpensive and memory efficient feature extraction method to measure the blockiness which is ideal to our computational intensive procedures.

(a) Blurriness: 0.8086

(b) Blurriness: 0.3648

Figure 1: Blurriness measure compare

After obtain the blurriness and blockiness from all the frames within a given video segment, we can proceed to calculate the visual distortion for this segment as follows:

$$E_{vv}(S_i) = \sum_{I_k \in S_i} \gamma q_{bk}(I_k) + (1-\gamma) q_{br}(I_k) \tag{2}$$

where $q_{bk}(I_k)$ is the measurement of blockiness of given frame, and $q_{br}(I_k)$ is the measurement of blurriness. Weight γ is set to 0.45 according to Liu et al. [1]

2.2 Incorrectness of Motion Model

In order to achieve the "mosaicablity", we use a homography to model the motion between frames. By matching SIFT feature points, we are able to locate significant points between frames and thus obtain the homography. In practice, getting a high quality panorama from video requires the inter-frame motion is closed to its homography and few casual videos can achieve that. Therefore, we measure the error using the real motion vector from SIFT feature points and the predicted value by homography between two successive frames.

$$E_{vm}(S_i) = \sum_{I_k \in S_i} \frac{1}{n_k} \sum_{p_{j,k} \in S_i} \| mv(p_{j,k}), mv_h(p_{j,k}) \| \tag{3}$$

First, for each adjacent frame I_k and I_{k+1} ,we find its matching SIFT feature pairs, then we calculate the homography using RANSAC based on these feature pairs. The notation $mv(p_{j,k})$ is the motion vector of j^{th} SIFT feature point of frame I_k and $mv_h(p_{j,k})$ is the predicted motion vector by homography at j^{th} feature point. Then the error of j^{th} feature

(a) Blockiness: 0.204

(b) Blockiness: 0.479

Figure 2: Blockiness measure compare

point is taking the L1 norm of these two terms. Then we average the errors of all feature pair in each frame and obtain the incorrectness of motion model by summing up all the average.

3 Extent of Scene

The extent of scene $\varepsilon(S_i)$ is defined as the scene covered by the video segment. Although it could be better if we maximizing the covering area to obtain a larger field of view, larger field of view often leads to more distortion. Hence, we decide to choose the reference frame where the distortion is minimum, in other words, we want the minimum area covered by segment S_i

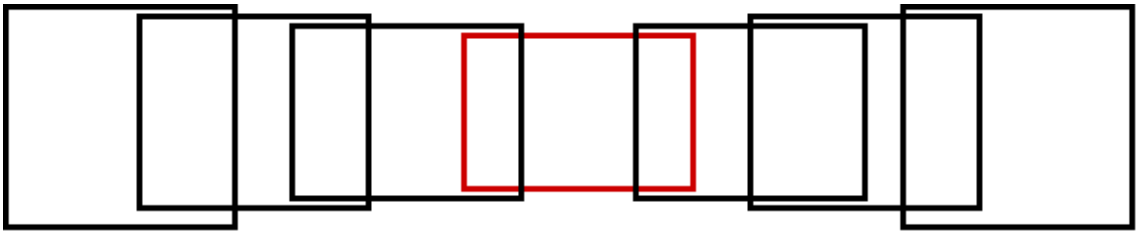

Figure 3: All the other frames in the video are alignment to the reference frame, and we want to find the optimal reference frame so that the area covered by aligning all the frames to be minimized

In order to find the optimal reference frame, we search for all possible combination of

panorama alignment as follows:

$$r = \arg\min_{f_r \in S_i} \bigcup_{f \in S_i} I(f, f_r) \tag{4}$$

Then the extent of scene is defined as:

$$\varepsilon(S_i) = \bigcup_{f \in S_i} I(f, r) \tag{5}$$

Liu et al. [1] used generic polygon clipping for finding the minimum area covered. However, due to the complexity of their method, we chose another simplified approach. Although in this approach the accuracy declines, the complexity is reduced a decent amount. The simplified version of finding extent of a scene is done by finding the distance between the center of reference frame to all other frames:

$$\frac{1}{n} \sum_{f \in S_i} \| C_r, C_f \| \tag{6}$$

Where n is the number of frames within the segment, C_r is the center pixel of the reference frame and f_r is the center pixel in a given frame. Figure 4 demonstrates when reference frame is the most left frame (the first frame in the image sequence), the middle frame and the right most frame. However, due to the relatively small distortion since this image sequence only has 5 images, the visual difference between three panoramas is subtle.

4 Experiments and Results for Determining a Panorama

We randomly select some videos from Youtube and manually cut the video into several segments. Liu et al.[1] use histogram based shot boundary detection to divide the video, while we use another approach which we describe in the next section. In order to test the correctness of the three panorama synthesis criteria, we choose to feed our algorithm with preprocessed video segments.

In figure 5, the motion error of the video estimated by our program is 598 (rounded to nearest integer) and since this segment has over 130 frames, so on average each frame has 4.6 error which is considered a good score in our standard. The image quality error for this segment is 23 which is also a relatively good score for a 640*360 (360p for short) resolution video. Therefore, we obtained a decent panorama from this segment. Figure 6 shows a shot from Shanghai, and the computed motion error is 587. Because the drone is very stable in the original video, the motion error is also what we expected. We also discovered that the

(a) The reference frame is set at the left most frame

(b) The reference frame is set at the center frame

(c) The reference frame is set at the right most frame

Figure 4: Difference reference frame results in different look of panorama

image quality error of 360p videos from Youtube usually ranges from 20 to 45. Therefore, the panorama synthesized is also in good quality with only a subtle distortion.
Figure 7 shows a failure case. In the segment, we observed a large amount of object motion which is from the waves and the boats on the lake between frame to frame. As a result, the motion error for this segment is 1521. As a conclusion, our implementation of judging a segment works as expected and is able to distinguish between segments with different quality.

5 Video Segment Fetching

Liu et al.[1] pre-process the given video by applying histogram based method for shot boundary detection and shot boundary is detected whenever the histogram intersection between two adjacent frames is below a threshold. However, we detect scene boundary by using the visual quality distortion.The idea is that in order to calculate the homography

Figure 5: Stitched using a video segment of a wood house

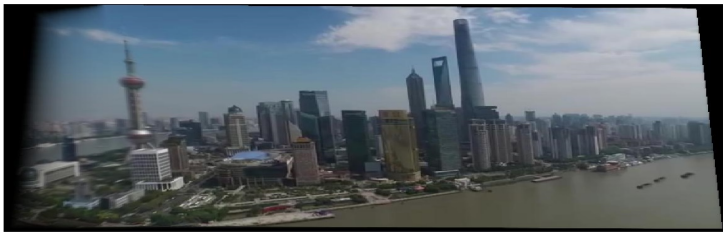

Figure 6: Stitched using a video segment of Shanghai city view taken by a drone

between two adjacent frames, we should have at least 4 SIFT matching pairs. According to our experiments, the video segments we have do not always have enough pairs to calculate homography when two frames are at scene boundary and even when they have enough pairs to calculate the visual quality error tend to be high. Thus, to solve the second problem, which is the program should be able to separate video segments to make a set of panoramas, we cut the whole video into segments with 20 frames each (since frame per second of a video is usually between 23 to 30, 20 frames segment roughly represents one second in the video which is neither too long nor too short), then we compute visual quality error for each segments. During the visual quality computation, there are two cases that clearly indicate a transition occurs in a given segment. One is that the transition frame is too sharp to have more than 4 SIFT matching pairs to the last frame. In this case, by setting the visual quality error to a large positive number (infinity in our case) we prevent the program from crashing due to not having enough matching pairs to calculate homography. Furthermore, since the larger the visual quality error is, the most likely it is a transition segment, setting the transition to infinity is acceptable. Another case is that when the transition occurs "smoothly", we need to set a threshold to separate segments. By examining the visual quality error of Figure 5 to 7, we decided to set the threshold to be 250. If any segment has visual quality error larger than 250, then it is likely to be a transition segment. In the final step of going through all segments to generate panoramas, we specify that there should be at least 3 consecutive segments to synthesis a valid panorama because a two second video is usually not having enough information (usually narrow in view). Since there may have several transition segments, we are able to generate several panoramas using the threshold of visual quality

Figure 7: Stitched using a video segment of West Lake

error as the signal of ending current panorama.

6 Experiments and Results for fetching multiple Panoramas

We define the types of videos that we might encounter while fetching as follows:
Case 1: A scene that is not capable of synthesizing panorama (a bad scene) followed by a scene that is a proper source for panorama (good scene) or vise versa. In both cases the program should only obtain one panorama from the good scene:

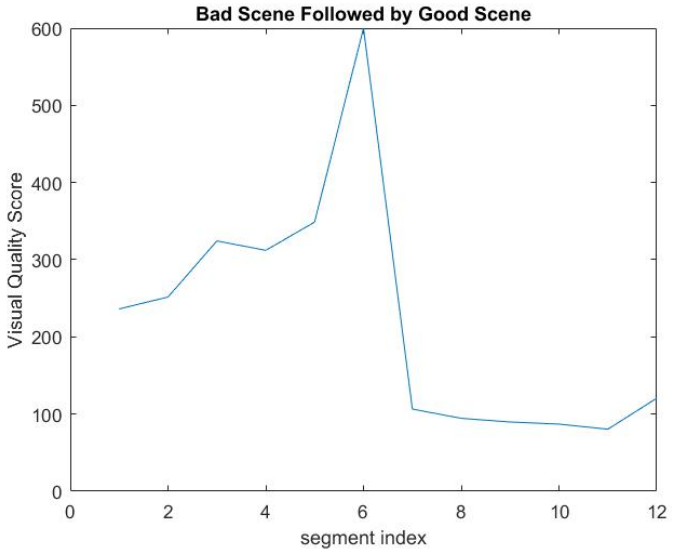

Figure 8: Visual quality error for Case 1

Above figure shows that the overall visual quality error for bad scene segments are high (over 250) and we have this sudden increase in the error at segment 6, thus we can say segment 6 has a high possibility of containing a scene transition. Thus, we want to drop all the segments before segment 7 due to high error. Then synthesizing the panorama use only segment 7 to 12.

Case 2: A good scene followed by a good scene. In this situation, the program should generate two panoramas :

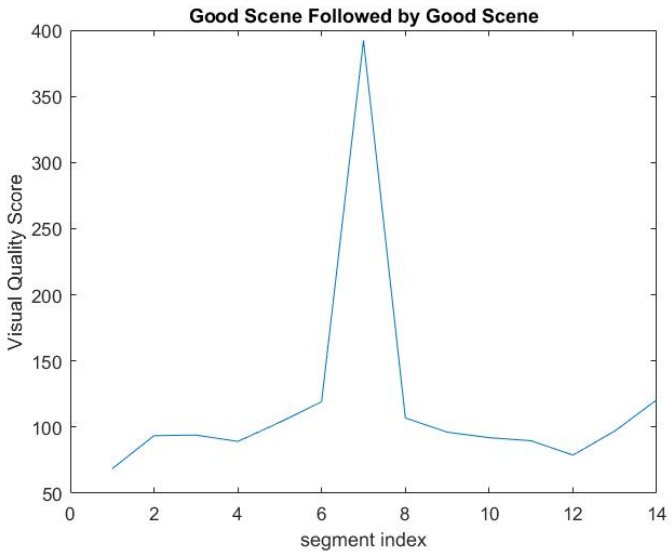

Figure 9: Visual quality error for Case 2

It can be observed that before and after segment 7 there are two smooth segments series and a sudden increase in visual quality error at segment 7. In this case, we drop segment 7 and synthesize two panoramas from segment 1 to 5 and segment 7 to the rest segments.
Case 3: A bad scene followed by a bad scene. In this situation, we drop both scenes, and we can expect the overall visual quality errors are relatively high across all the segments.

7 Conclusion and Limitations

The result of this experiment is positive, we are able to solve the two problems which are determining a panorama and synthesis multiple panoramas. We learned a lot by reading the paper and searched for related domain knowledges. Since our experiment consists of several technique parts and we choose to implement in a simpler way, there are still limitations that need to be solved. For example, the whole process is significantly time-consuming. Because we are computing the visual quality score in each segments due to our segmentation algorithm, the complexity tends to be very high. During our experiments, a 250 frames video will take roughly 40 seconds to process. Another limitation is that because we are dropping the whole 20 frames segment with scene transition the field of view of the final panoramas are reduced in small amount but the effect is subtle if the the panorama is made by a longer video segment.

8 Overview of our project

Credit to: part of the synthesis and blending stage of panorama synthesis is from project 4 CS534. TODO: blurness and blockiness credit if necessary
Number of lines: TODO: 400lines?
Contribution: Lingfeng Huang: establish the skeleton of the program and the report, mainly focus on solving the first problem which is determine a panorama. Fang Wang: focus on solving the second problem which is how to separate different panoramas. Suggested several algorithms to implement and speed up the code when working with Lingfeng to solve the first problem. Revised and reformed the report.

References

[1] Feng Liu, Yu Hen Hu, and Michael Gleicher. "Discovering panoramas in web videos". In: *Proceedings of the 16th International Conference on Multimedia 2008, Vancouver, British Columbia, Canada, October 26-31, 2008*. 2008, pp. 329–338. DOI: 10.1145/1459359.1459404. URL: http://doi.acm.org/10.1145/1459359.1459404.

[2] *Panorama*. URL: https://en.wikipedia.org/wiki/Panorama.

[3] Hamid R. Sheikh, Zhou Wang, and Alan C. Bovik. "No-reference perceptual quality assessment of JPEG compressed images". In: *ICIP (1)*. 2002, pp. 477–480. DOI: 10.1109/ICIP.2002.1038064. URL: http://dx.doi.org/10.1109/ICIP.2002.1038064.

[4] Hanghang Tong et al. "Blur detection for digital images using wavelet transform". In: *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo, ICME 2004, 27-30 June 2004, Taipei, Taiwan*. 2004, pp. 17–20.