Discovering Panoramas in Web Videos
lhuang58.github.io

Lingfeng Huang, Fang Wang

11/2016

Abstract

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 will implement our version of Discovering Panoramas in Web Videos by Liu et al. to solve the problem by selecting optimal segments within a given web videos, then perform synthesizing to obtain panoramas. This whole procedure is basically 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 [**wikipedia**]. 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 and next step is finding the alignment between each image and warping accordingly, and 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 that 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 sequence of images, although some segments within videos have relatively low image quality and also moving object, they are still shot from the same optical center and cover a wide field-of-view. Lui et al. suggests an approach that synthesizing panoramas by identifying proper segments within videos as panorama source [**Lui**]. They convert the problem into a optimization problem, and set up three constraints in order to evaluate the video segments. Lui 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[**Lui**].

2 Visual Quality Measure

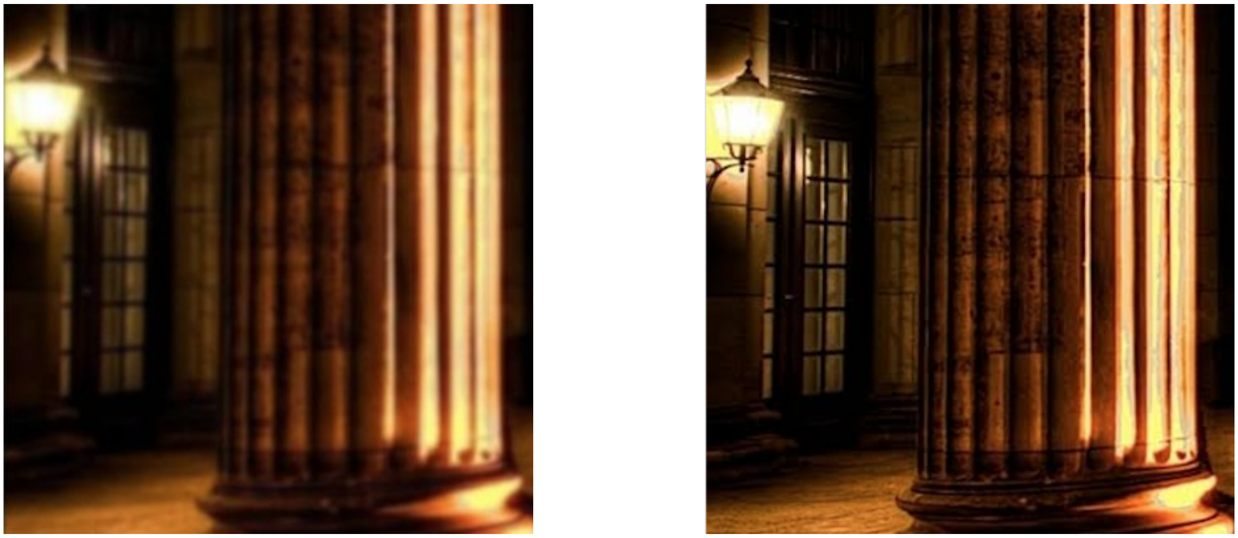
We measure the visual quality of a single frame based on two terms, one is incorrectness of the motion model $E_{vm}(S_i)$ and the source image visual quality $E_{vv}(S_i)$. Then by setting up the visual quality distortion $E_v(S_i)$, we can obtain the visual quality measure.

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

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 blocky the image is. We use the idea of Tong et al.'s method of measuring blurring artifacts by using Haar Wavelet Transform [**Tong**].

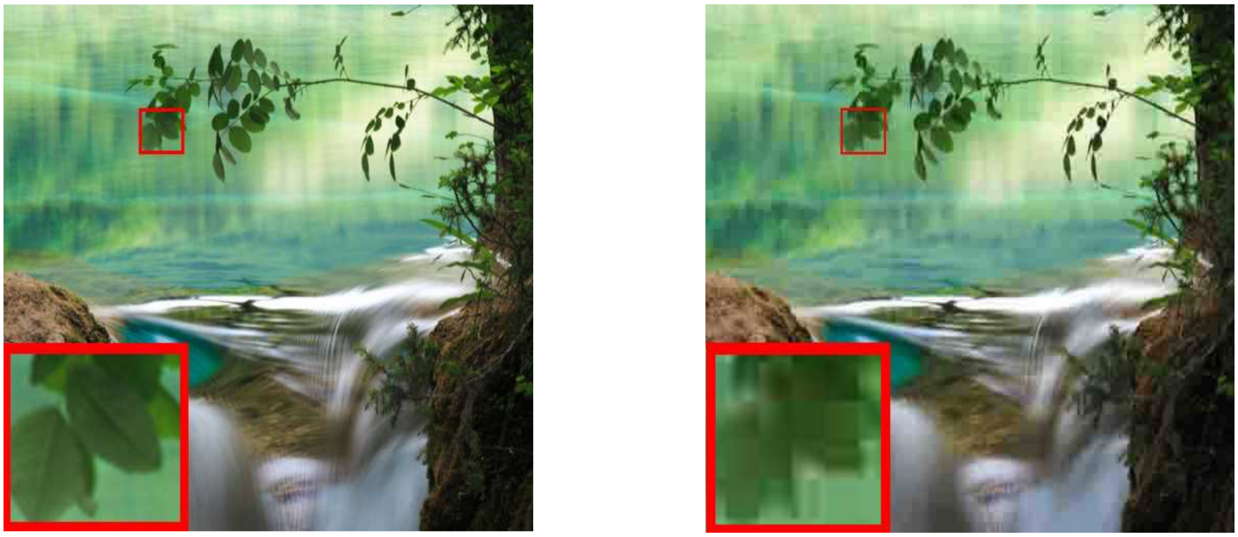


(a) Blurriness: 0.8086

(b) Blurriness: 0.3648

Figure 1: Blurriness measure compare

The blockiness is measured by using the method of Wang et al. which estimates the average difference across block boundaries modulated by image activities [**Wang**].



(a) Blockiness: 0.204

(b) Blockiness: 0.479

Figure 2: Blockiness measure compare

After obtain the blockiness and blurriness from all the frames within a given video segment, we 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.

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.

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

We first for each adjacent frame I_k and I_{k+1} find its matching SIFT feature pairs, and calculate 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 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 good if we maximizing the covering area to obtain a larger field of view, larger field of view often means more distortion. Hence, we want to choose the reference frame where the distortion is minimum, in other words, we want the minimum area covered by segment S_i

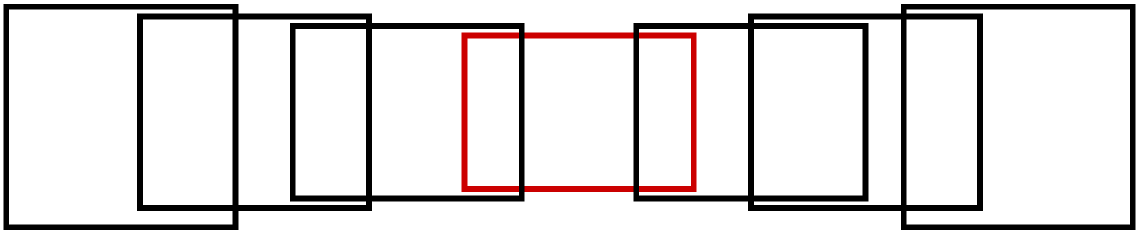


Figure 3: All the other frames in the video are alignment to the reference frame, and we want to find the optimal reference frame so that the area covered by aligning all the frames to be minimized

In order to find the optimal reference frame, we want to search for all possible combination of panorama alignment as follows:

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

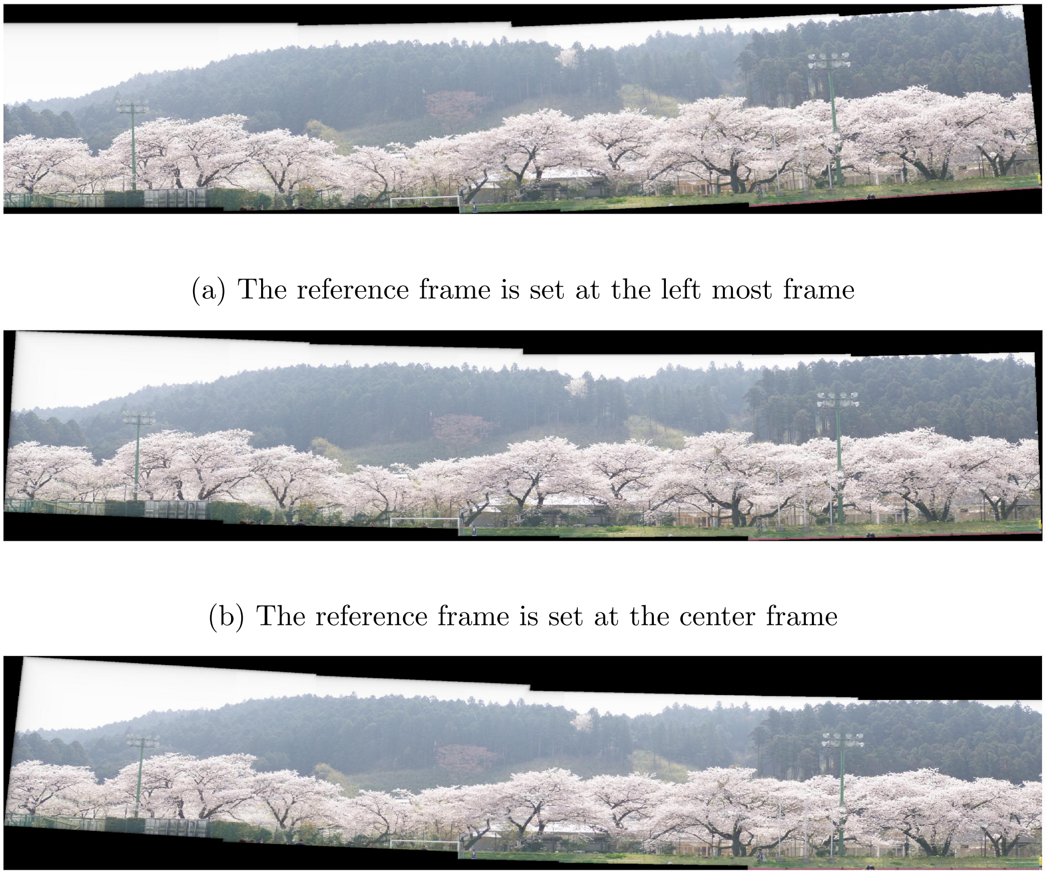
Then the extent of scene is defined as:

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

In Lui et al. paper, they used generic polygon clipping for finding the minimum area covered. However, we used another approach. Although the accuracy declines, the complexity reduces in decent amount, we choose to find the distance between the center of reference frame and that of all other frames as follows:

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

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



(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

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 Panorama

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

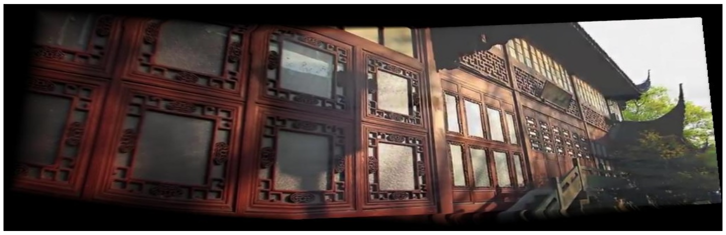


Figure 5: Stitched using a video segment from West Lake

In figure 5, the motion error we estimated 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 360p video. Therefore, we obtained a decent panorama from this segment. Figure 6 shows a shot from Shanghai, and the computed motion error is 587.

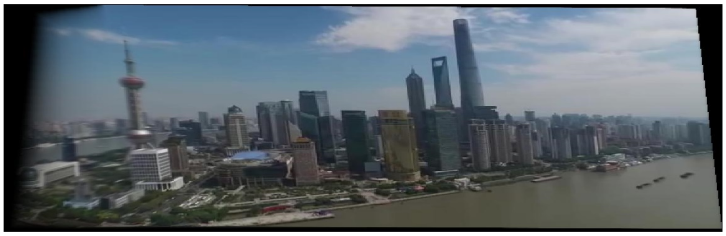


Figure 6: Stitched using a video segment from Shanghai shot by drone

Because the drone is very stable in the original video, the motion error is also what we expected. We also discovered that the 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.

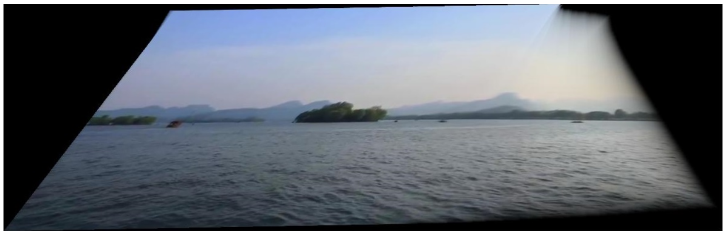


Figure 7: Stitched using a video segment from West Lake

Figure 7 is a failure. In the segment, we observed a large amount of object motion which is from the waves and the boats on the lake. As a result, the motion error for this segment is 1521. In addition, due to the panorama. 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

Lui et al. preprocess 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 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, we cut the whole video into 20 frames segments, then compute visual quality error for each segments and by setting the visual quality error to infinity when encounter scene transition segment and dropping that segment we are able to append all adjacent segments to obtain a optimal segement for panorama stiching. 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 (bad scene) followed by a scene that is proper source for panorama (good scene) or a good scene followed by a bad scene, in both cases we want to only obtain one panorama from the good scene:

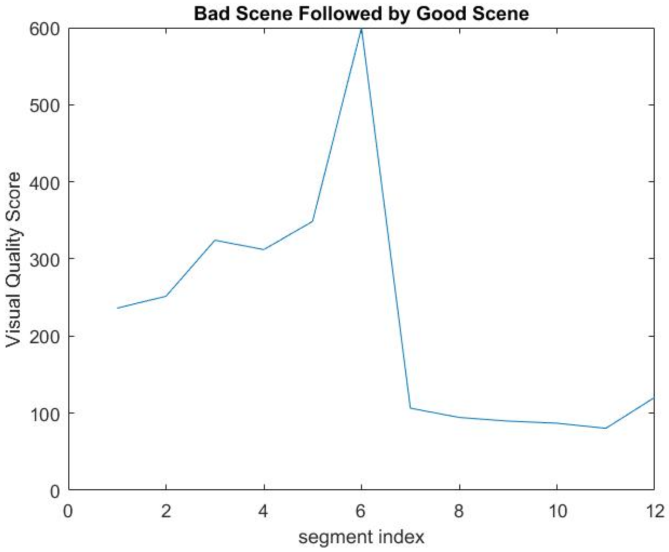


Figure 8: Visual quality error for Case 1

Above figure shows that the overrall 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. And we want drop all the segments before segment 7 due to high error and synthesize panorama use only segment 7 to 12.

Case 2: A good scene followed a good scene followed. In this situation, we want to get two panoramas :

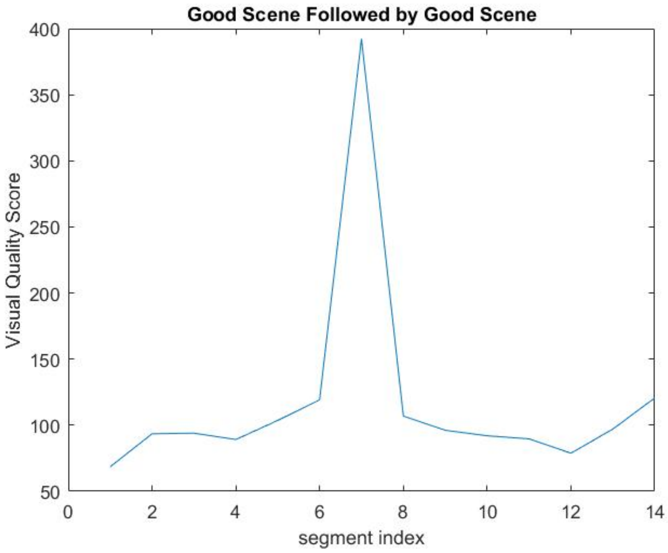


Figure 9: Visual quality error for Case 2

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 want to 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 want to drop both scenes, and we can expect the overrall visual quality errors are relatively high across all the segments.

6 Limitations

First of all, the whole process is significantly very time consuming. Because we are computing the visual quality score in each segments due to our segmentation algorithm, the complexity tends to be very large. During our experiments, a 250 frames video will take raghly 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.