Obtaining High-quality Panorama from Videos

Zhang Shuheng
The Chinese University of Hong Kong, Shenzhen
120090270@link.cuhk.edu.cn

Zhao Songlin
The Chinese University of Hong Kong, Shenzhen
120090346@link.cuhk.edu.cn

Abstract

Panorama construction is an essential research topic in digital image processing. Researchers have done a lot on panorama construction methods, most of which focus on image stitching, but few consider video stitching. This project studies panorama construction from videos. In terms of frame selection, we proposed a binary search method based on the similarity between neighboring frames. We further achieved cylindrical projection to eliminate distortion of the panorama. Finally, The obtained panorama is deblurred by analyzing the camera's motion.

1 Introduction

Image stitching is a popular technology for photographers to produce high resolution pictures with a wide field of view. Existing research mainly focus on panorama construction from separate photos. Our project achieves video stitching. Algorithms of panorama construction were applied in video stitching, such as SIFT, RANSAC, and projective mapping [1, 3]. A novel frame selection algorithm is advanced in this project. Additionally, our project also filled the gap of reducing the motion blur when constructing panorama from video.

One of the major challenge of the previous method in panorama construction is that, as the total field of view increase, the stitched image will be severely distorted. Thus, cylindrical transformation is introduced to our project [4]. As a result, the panorama will be less distorted even if the field of view is large. The second challenge of panorama construction from video is that when the camera is moving, the video will easily suffer from motion-blur. The blur is especially obvious in the low illumination circumstance, where the shutter speed of the camera is forces to be low. To reduce the motion-blur, our project estimate the horizontal motion of the camera, and construct a spacial filter based on the camera motion. Then the motion blur in the frequency domain can be effectively reduced by Wiener's filter.

1.1 Related Work

Interest point detection and match SIFT algorithm can be used for the corner point detection. It can localize the key points and give a spacial analysis to the key points [5]. In the match step, the RANSAC algorithm can used for the analysis features between two 2D images and find the match pairs, which had been tested by previous researchers for hundreds of times [1].

Image warping and transformation Images should undergo warping before they are stitched to construct the panorama. Projective mapping is a mapping from one plane onto another plane [3]. A

Paper Report of Final Project for CIE4512 - Digital Image Processing.

projective mapping has eight degrees of freedom, and at least four pairs of match points on the image are needed for solving a projective mapping [3]. If there are more than four match points, the least square is used to estimate the projective mapping matrix.

Motion-blur reduction When the motion and the exposure time of the camera are known, the degradation filter of the motion blur in the spatial domain can be estimated, and then transformed into frequency domain by the discrete Fourier transform. The motion blur of the original image can be reduced in the frequency domain by Wiener's filter [2]. The Wiener's filter will be helpful in deblurring, where the noise to signal noise can be estimated [2].

2 The Proposed Algorithm

2.1 Pipeline

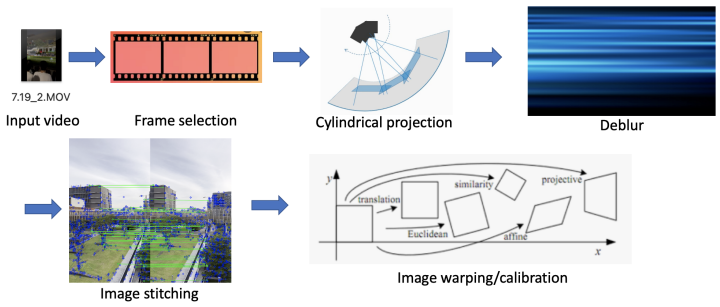


Figure 1: The overall logic of our project

2.2 Selecting frames from a video

Algorithm 1 Frame Selection

Require: indexes of two frames where $idx1 < idx2$.
Ensure: Selected indexes into an array.

$selectedFrames \leftarrow \{\}$; ▷ create empty array
$selectedFrames \leftarrow idx1$; ▷ add the start index
$selectedFrames \leftarrow idx2$; ▷ add the end index
function SEARCH($idx1$,$idx2$)
  if $idx2 - idx1 < 2$ And REACHCRITERION$(idx1, idx2)$ **then**
    return
  else
    $selectedFrames \leftarrow \frac{idx1+idx2}{2}$; ▷ add the middle index
    SEARCH$(idx1, \frac{idx1+idx2}{2})$;
    SEARCH$(\frac{idx1+idx2}{2}, idx2)$;
  end if
end function
$selectedFrames \leftarrow$ SORTED$(selectedFrames)$; ▷ sort the indexes

Since the efficiency should be guaranteed as there are large number of frames to be select from, our project propose the proxy frames, with sizes compressed compared with the original ones, and the criterion is applied on the proxy set. In algorithm 1, the inputs are the indexes of the start frame and the end frame $(idx1, idx2)$. The algorithm will search recursively until all the consecutive frames in the selected set satisfy the criterion, which is defined as:

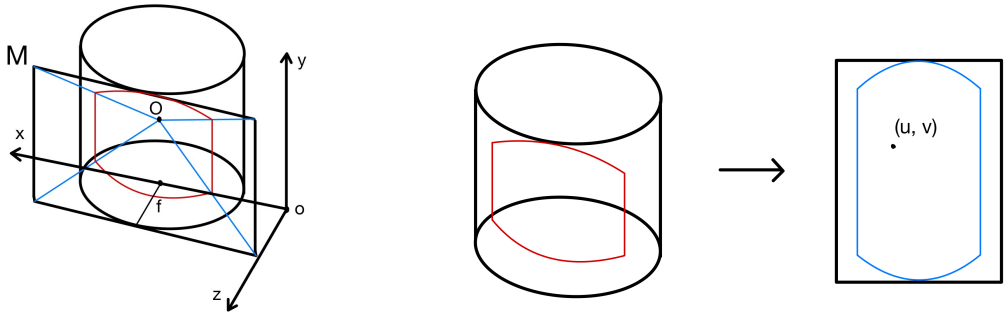
$$\#\text{ of good match}(idx1, idx2) > n \times \text{thres} \times \frac{idx2 - idx1}{L}. \quad (1)$$

Criterion 1 returns a boolean value, indicating whether criterion is reached or not. The variable $\#$ of good match is calculated by the previous RANSAC algorithm, thres is the hyper-parameter

for the criterion, and L is the total number of frames in the video. The last term with respective to indexes and video length is introduced in the criterion. Thus, when the two indexes are far from each other, there is smaller possibility that the two frames match each other, while the frames closing to each other will easily match. variable n is usually a small integer. For example, when $n = 2$ and $idx1 - idx2$ is $\frac{L}{2}$, the threshold $thres$ is exactly the number of good match.

2.3 Projection onto a cylinder

The video is captured by rotating the camera at a fixed point. So the projective transformation will produce a heavily distorted panorama. The solution is to project the images onto a cylinder before stitching them.



(a) 3D illustration of projection onto a cylinder

(b) Correspondence between cylinder and unrolled image

Figure 2: Representation of the cylindrical projection

In figure 2(a), the center of the cylinder is O, while the radius of the cylinder is f, which is the focal length measured by number of pixels. The focal length in pixel scale can be derived from the optical focal length and the camera parameters. Assume the optical focal length (equivalent to 35mm film) of the camera $\hat{f}$ is known, then:

$$f := \text{Rounding}\left[\frac{W \cdot P_w}{43.23 \cdot R \cdot W_p} \cdot \hat{f}\right] \tag{2}$$

where W is the width of the video, P_w is the number of pixels in the camera sensor along the video's width side; R is the ratio of camera sensor's width and height, and W_p is the effective pixels in the video taking along the width of the sensor.

The cylinder to be projected is tangent to the original image M. The center of the image on xy plane is (c_x, c_y). For every point (x, y) in original image M, there is a projected point P on the surface of the cylinder that can be measured by two parameters: θ and h, where

$$\theta = \arctan\frac{x - c_x}{f} \quad \text{and} \quad h = \frac{f(y - c_y)}{\sqrt{f^2 + (x - c_x)^2}}. \tag{3}$$

After projection, we have to unroll the cylinder to get a flat image denoted by N (as shown in figure 2(b). Here N is the same shape as the original image.

For every pixel (u, v) in image N, we can find its correspondence with original image M:

$$\begin{aligned} u &= c_x + f\frac{\theta}{2\pi} = c_x + \frac{f}{2\pi}\arctan\frac{x - c_x}{f} \\ v &= c_y + h \quad = c_y + \frac{f\,(y - c_y)}{\sqrt{f^2 + (x - c_x)^2}}. \end{aligned} \tag{4}$$

2.4 Motion-deblur of the panorama

2.4.1 Estimating the motion of camera

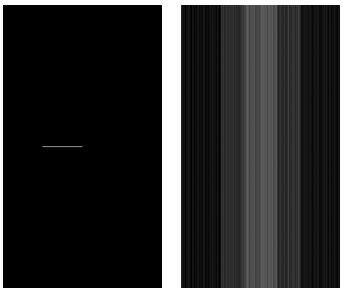
For a given video of length t seconds, the total number of frames is $N = \frac{t}{fps}$, where fps is frame rate. Then the video can be represented by a set $\{M_i \text{ for } i = 1, 2, 3, ..., N\}$. Based on the stitched image on the cylinder, we can estimate the total angle that the camera rotated during the entire video by $\theta = \frac{W}{f}$, where W is the width of the stitched image on the cylinder and f is the focal length.

We assume that the camera rotates at a single direction (from left to right), and during the exposure time at each frame, the rotating velocity is constant. Thus, for any frame M_i, we extract its neighboring frames M_{i-1} and M_{i+1}. Its velocity can be estimated as

$$v_i := w_i \cdot f = \frac{\text{fps} \cdot \Delta\theta}{2} \cdot f = \frac{\text{fps} \cdot (\theta_{i+1} - \theta_{i-1})}{2} \tag{5}$$

2.4.2 Deblurring in the frequency domain

Assume the camera rotates from left to right, and the exposure time T of each frame is already fixed, once the velocity of the i-th frame v_i is derived, the blur length can be calculated: $L_i := v_i T$. Then the spacial filter and frequency filter can be derived (as in fig. 2.4.2, note that the length of white line in fig. 3(a) is L).



(a) Spatial filter (binary image) (b) Frequency filter H_i (magnitude)

Figure 3: Sketch of the ideal filter of horizental motion blur

After that, the Wiener's filter is used to deblur the image in the frequency domain (as in eq. 6).

$$\hat{F}_i(u, v) = \left[\frac{1}{H_i(u, v)} \frac{|H_i(u, v)|^2}{|H_i(u, v)|^2 + K} \right] G_i(u, v). \tag{6}$$

where G_i represents the blurred image, K is a constant representing the ratio of noise and signal. $\hat{F}$ is the deblurred image desired. After inverse discrete Fourier transformation to $\hat{F}$, the deblurred image is derived. Then our project performs image stitching on the deblurred images.

3 Experiments

3.1 Dataset and Implementation Details

The videos are all taken in the campus of the Chinese University of Hong Kong, Shenzhen, with mobile phone or camera. There some requirements for the video dataset:

* The video must be 1080p (1920×1080) 8-bit color depth with frame rate 30.
* When taking the video, the camera must be placed vertically, and the rotation direction of the camera must be from left to right

* The camera user should know the equivalent optical focal length $\hat{f}$ and set exposure time T for the camera, then input them to our program. Our code takes $\hat{f}$ and T as the input variables.

This project implements all the algorithms in Python. We first implement the frame selector, where a video is the input. Three parameters should be input by user:

$$(proxy\_compress,\ sift\_thres,\ interest\_thres)$$

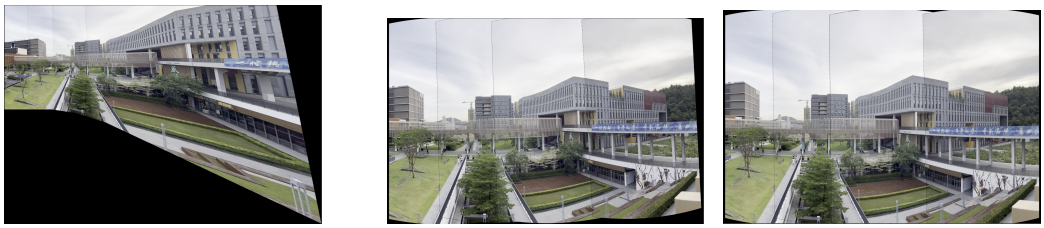
which are respectively the compress ratio for frame selection, the threshold for the ratio test in the SIFT detector and the threshold in the criterion for selecting frames.

In the cylindrical transformation process, the user must input the information of the camera in eq. 2. The codes will derive the focal length in pixel scale and does cylindrical transformation to each frame selected. In the deblurring procedure, the user is supposed to input T: exposure time of each frame (e.g., 1/30s at night). The code will calculate the angular velocity of the rotating camera and the blur length L. If L is large, the cylindrically warped frames will be deblurred. If L is smaller than 5, which means the motion blur is very little, the code will skip the deblur procedure to save the computational resources. The last step is the image stitching. The code will automatically construct panorama image by previously cylindrically transformed and deblurred frames. This project tests the sequential stitching and the divide-and -conquer stitching.

3.2 Performance evaluation of our algorithm

3.2.1 Frame selection, image warping and stitching

This project tries different criterions and different parameters in the criterion of selecting frames. We find that criterion 1 with $n = 6$ is a decent choice. We test different values of parameters and find (3, 0.5, 30) works well in selecting frames.



(a) projective warping (b) cylinderical warping (c) cylinderical warping with corner calibration

Figure 4: Panorama constructed from video with different warping methods

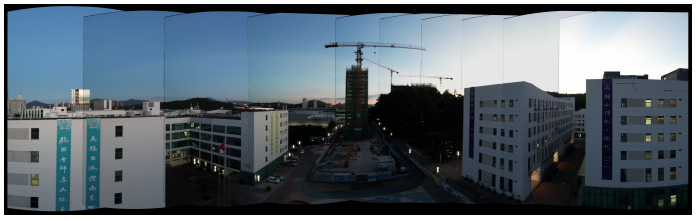
Figures 4(a) and 4(b) compares the effect of projective and cylindrical transformation. It is obvious that when the field of view get larger when the camera rotates, the cylindrical warping outperforms the projective warping as the distortion of image gets relieved. Figures 4(b) and 4(c) compare the image stitching with and without calibration. When the camera is not hold stably, it is easy to construct the panorama like 4(b). We calculated the warped locations and relocate the corner points to derive 4(c). By cylindrical warping and calibration, this project managed to construct panorama of wide field of view, even the angle get close to 360°, high quality panorama can be achieved with enough stability (as in 5).

Figure 5: Panorama construction without deblurring ($T = 1/1000$, proxy=3, siftThres=0.5, interestThres=30)

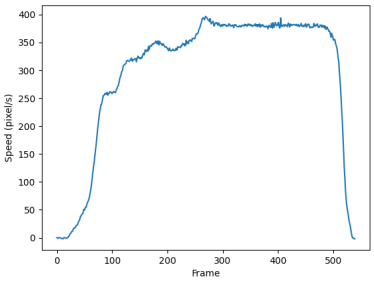
3.2.2 Motion deblurring

This project achieved the motion deblur process by analyzing the rotation of the camera and applying the Wiener's filter. Finally, the blur-reduced panorama is constructed (as in 6(a)). The velocity of the camera is calculated by two neighboring frames of the selected frame. Figure 6(b) shows the camera motion in the source video of 6(a). It is clear that the camera stand still at the start and end of the shooting, and rotates when recording.

By Wiener's filter, deblurred panorama is achieved. We tried different values of K in the Wiener's filter 6, and find 0.1 be a good choice for our data set. Figures 6(c) and 6(d) compares the zoomed in area of the deblurred and non-deblurred panorama constructed from the source video of 6(a). The effect is obvious as the lights get thinner in the deblurred panorama.



(a) panorama construction with deblur (without blending)



(b) velocities of the frames in the source video of panorama 6(a)

(c) zoom in an area of 6(a)

(d) zoom in the same area without deblur

Figure 6: Panorama construction with motion deblurring ($T = 1/30$, proxy=3, siftThres=0.5, interestThres=30)

4 Discussions

* Our program can only handle the construction from left to right horizontally. In the image stitching procedure, the latter frame is stitched on the former frame. If the camera moves towards the inverse direction (i.e., from right to left), then the frame on the left will be stitched onto the left frame, which will give a negative position in the projective transformation process, but out algorithm cannot handle the negative position in the stitching. During video capturing, if the camera is not vertical to the ground, there will be loss of information in the stitched image.
* The SIFT detector we used in image stitching is only sensitive to corner points, so there will be matching errors in images with few corner features.
* The deblurring process by Wiener's filter is not perfect, as there is ring effect on sharp edges.

References

[1] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 510–517. IEEE, 2005.

[2] R. C. Gonzalez, R. E. Woods, et al. Digital image processing. 2002. *Google Scholar Google Scholar Digital Library Digital Library*, 2007.

[3] P. Heckbert. Projective mappings for image warping. *Image-Based Modeling and Rendering*, 869, 1999.

[4] M. Lin, G. Xu, X. Ren, and K. Xu. Cylindrical panoramic image stitching method based on multi-cameras. In *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 1091–1096. IEEE, 2015.

[5] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.