Moving Object Removal/ Motion Reconstruction in Stereo Panoramas

Danyang Wang
danyangw@stanford.edu

Xiaoshi Wang
xiaoshiw@stanford.edu

Chenjie Yang
yangcj@stanford.edu

Abstract

In this paper we purposed an approach to remove ghosting artifact generated by moving objects and reconstruct the movement of these objects in panorama images. Our approach contains three steps. First is to detect moving objects in the image flow with moving background. Second is to replace the moving objects with corresponding background scenes. Third is to reconstruct the movement of moving object we detected in step 1. Finally we stitch the processed images together to generate panorama images with no ghosting artifacts and we generate movies containing reconstructed movement of moving objects.

1. Introduction

When people try to take panoramas, especially in some famous tourist sites, ghosting artifact is a very common and sometimes severe problem, doing harm to the quality of result images. It is impossible to let everything be stable when someone want to take the set of images, thus we need some other ways to post-process the images such that the final panorama after stitching have no ghosting artifacts. In addition, it is also interesting to recover the moving paths of moving objects in the panorama scene. In this paper we would like to purpose a method to both remove ghosting artifacts in panorama and reconstruct the moving path of moving objects in video.

1.1. Previous Works

The previous researches in removing image ghosting artifact focus on images with stationary backgrounds such as HDR images. Two different types approaches are with explicit tracking of moving objects or not. For example, Kang et al.[3] applied a method to track the moving objects in frames for HDR images generation using gradient-based optical flow and then remove the ghosting artifacts. Khan et al.[4] described a another way to remove ghosting artifacts in HDR images using weights computed for each pixel in image flows, without explicit moving object detection.

For removing objects in panorama images, the situation is different. In order to generate a panorama image, we need to take a sequence of images with non-stationary background. Because of this, previous methods with stationary backgrounds do not work very well. There are not many previous researches in removing ghosting artifacts in panorama images. Wan et al.[6] described an approach to pre-stitch two consecutive images, detect the moving objects in overlapping area by color difference and rearrange final blending region not to blend moving object in the two images. This method is not very robust and in some situations the same moving object can be shown several times in the panorama, which is not desired. Yingen Xiong[7] applied another approach that first constructs a composed gradient field in moving object regions to remove moving objects, then recover those regions with the best-fit contents found in the other part of the image by the gradient domain region filling operation.

1.2. Equipment and Dataset

The dataset was collected with the 360 degree stereo panorama camera(Figure 1) built by Stanford Computational Imaging Group. In each dataset we took 400 images within a 360 degree revolution. When taking the dataset we tried to cover many different scenarios including multiple moving objects, moving object from different distances and moving object with different speed.

The original procedure of generating stereo panorama with the Stereo Panorama Camera is to undistort, rotate and crop each raw frame. The next step is to feed the cropped images into a software called AutoStitch [1] to get the final panorama. This approach will generate ghosting artifact if there are moving objects between frames. In this paper, we propose a method to remove ghosting artifacts in stereo panorama images and also reconstruct the motion of the moving objects.

2. Approaches

2.1. Moving Object Detection(Fast-MCD)

In our approach we use the method purposed by K M Yi et al.[8] to detect moving object with moving background.

Figure 1. Stereo Panorama Camera

In their paper the authors call the method "fast Minimum covariance determinant"(fast-MCD). We will also refer this method as "fast-MCD" in our paper. We choose this method because it is fast and the detection results are reasonable. Figure 2 illustrates the work flow of the fast-MCD method. We would like to briefly introduce the method and then how we applied it in our approach.

2.1.1 Framework of Yi et al.'s method

There are three building blocks in the Yi et al.'s method: Single Gaussian Model(SGM) with age, Dual-Mode SGM and Motion Compensation.

Single Gaussian model with age uses a Gaussian distribution to keep track of the change of moving background. The general idea is that if in a new frame the pixel intensities in a specific grid are very different comparing with the corresponding grid in the previous frames(far from the mean of Gaussian), then the method think there is a moving object covers this grid. The way we update the mean and variance of the Gaussian model is:

$$\mu_i^{(t)} = \frac{\tilde{a_i}^{(t-1)}}{\tilde{a_i}^{(t-1)}+1}\tilde{\mu_i}^{(t-1)} + \frac{1}{\tilde{a_i}^{(t-1)}+1}M_i^{(t)} \tag{1}$$

$$\sigma_i^{(t)} = \frac{\tilde{a_i}^{(t-1)}}{\tilde{a_i}^{(t-1)}+1}\tilde{\sigma_i}^{(t-1)} + \frac{1}{\tilde{a_i}^{(t-1)}+1}V_i^{(t)} \tag{2}$$

$$\alpha_i^{(t)} = \tilde{\alpha_i}^{(t-1)} + 1 \tag{3}$$

Where M and V are the mean and variance of all pixels in grid i, α_i is the age of the grid i, referring to the number of consecutive frames this grid is shown. parameters with tilde refer to the parameters values of the corresponding grid in previous frames(remember that the background is changing so the we need to match the "same" grid in different frames). We will introduce how the method find the corresponding grids in different frames later in Motion Compensation.

Dual-Mode SGM is to use two SGM to record the grid related to background and foreground(moving objects) separately such that the pixel intensities in foreground do not contaminate the parameter values in background Gaussian model. In more detail, for each grid we keep track of two SGM B and F and each time we only update one model. We start from updating B(assume it as the background model), until

$$(M_i^{(t)} - \mu_{B,i}^{(t)})^2 >= \theta_s \sigma_{B,i}^{(t)}$$

Where θ_s is a threshold parameter. Then we update F, similarly until

$$(M_i^{(t)} - \mu_{F,i}^{(t)})^2 >= \theta_s \sigma_{F,i}^{(t)}$$

. Also we change swap the model for recording foreground and background model if the number of consecutive updates of F is larger than that of B, that is

$$\alpha_{F,i}^{(t)} > \alpha_{B,i}^{(t)}$$

This is because if the "foreground" stay longer in frames than "background", than the foreground is probably the real background.

Motion Compensation is used to match grids in consecutive frames. Because the background is moving in different frames, this step is very important. The Motion Compensation method Yi et al. purposed is using a mixing model. For all grids G(32×24) in time stamp t, the method first performs the Kanade-Lucas-Tomasi Feature Tracker(KLT)[5] on corners of each grid $G_i^{(t)}$ to extract features of these points. Then the method performs RANSAC[2] to generate transformation matrix $H_{(t,t-1)}$ from frame at t to $t-1$. Then for each grid G_i^t, the method find the matching grid $G_i^{(t-1)}$ by $H_{(t,t-1)}$ and applies a weighted summation for grids in frame $t-1$ that $G_i^{(t-1)}$ covers to generate the parameter values of $G_i^{(t-1)}$, which are the tilde values we mentioned in SGM($\tilde{\mu_i}^{(t-1)}, \tilde{\sigma_i}^{(t-1)}, \tilde{\alpha_i}^{(t-1)}$).

2.1.2 Our application of Yi et al.'s method

We used the code published with this paper at `https://github.com/kmyid/fastMCD/` to perform moving objected detection. First we generated a movie using all our frames and then tuned the model parameters in fast-MCD to generate a binary mask for each frame with reasonable quality. The most important parameter we tuned is

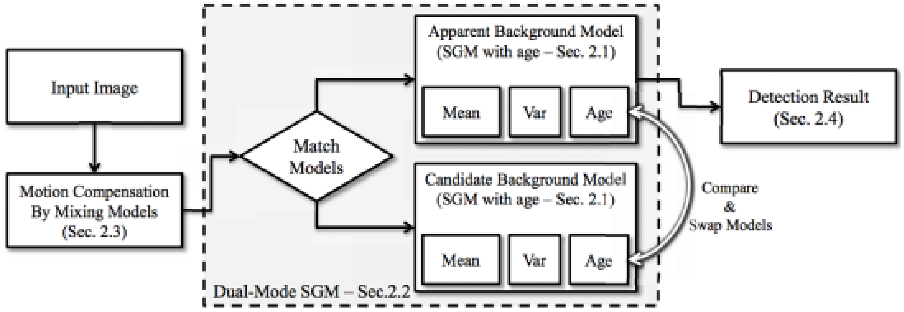

Figure 2. The flow of fast-MCD method purposed by Yi et al.

the threshold θ_s which we introduced before. Finally, for each set of images flows, we can get a set of corresponding binary masks with moving object detection results.

2.2. Mask Optimization

The original masks generated from fast-MCD has lots of problems. First it contains some noise due to the limitation of its performance. In addition, a more serious problem is that it cannot detect the whole shape of moving objects, such that there will be some part of moving objects not being substituted if we simply apply the original masks. This is the reason we come up with an algorithm to remove the noises and generate the bounding boxes around the detection result to cover whole moving objects. Figure 3 shows the original mask and the new mask after optimization.

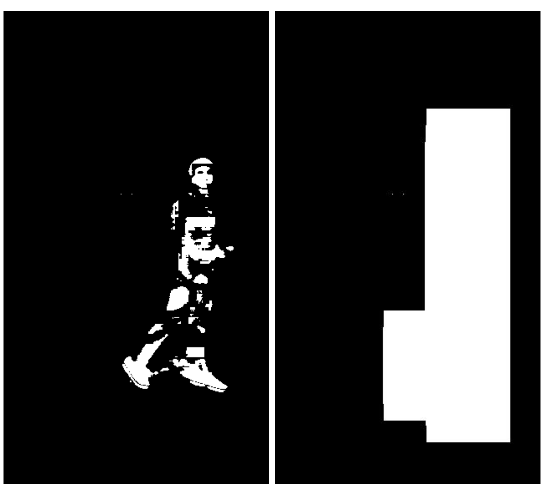

Figure 3. Sample results before and after mask optimization

The detail of the mask optimization method is first to remove small regions in the original mask. Then we scan from left to right of the image to locate moving objects and generate bounding boxes. First we locate the left boundary by finding a column with summation of its pixel intensities larger than a threshold and also the summation of previous 5 columns is less than a threshold. Then we locate the corresponding right boundary using same method, except changing the "previous" to "next". Then we locate the top and bottom boundaries in the area separated by left and right boundaries with same idea. Here we implemented two choices. The conservative method is to find one top boundary start from first row and one bottom boundary start from last row such that finally there is only one big bounding box be generated given the left and the right boundaries. The more aggressive choice is to generate several smaller bounding boxes given the left and the right boundaries. Finally we chose the conservative model according to the qualities of original masks and our testing results. Then for each bounding box we locate, we expend the it slightly larger to be more safe. This is because the original mask may not include the edge part of moving objects, for example, hand, foot and head top of a person. Finally we mark all pixels in the bounding box be 1. There are a bunch of parameters that we can tune to generate best results for different sets of images. For example, the area limitation in small object removing and the thresholds for boundaries. In general, smaller the moving objects, smaller the threshold values we would like to set. However, with smaller threshold values, the algorithms will be more sensitive to noise.

2.3. moving object substitution

After generating the bounding box for each moving object, we performed the following algorithm to substitute pixels of the moving object with pixels of the same location from neighboring frames.

At First, we map all images onto the same surface. Since the camera circles around a fixed vertical axis, it's better to use a cylindrical surface. After projecting the images onto the cylindrical surface, it's possible to perfectly align the images by using only horizontal translations. So at the beginning, we attempt to estimate the focal length of the camera: If the images are projected using the right focal length, there should be a set of horizontal translations that perfectly align the consecutive images. For other focal lengths, the alignment won't be that good and will result in large mean-squared error if we try to overlap them.

So we choose some focal lengths, for each focal length, we make the projection and then look for the best set of horizontal translations that minimize the mean-squared error of overlapped areas between consecutive images. The focal length that gives the minimum mean-square error will be our estimated focal length. We then project all images onto the same cylindrical surface and align them using horizontal translations.

Next, we select a proper search radius. This parameter is the number of neighboring frames to search for substitution pixels. It needs to be adjusted based on the density of frames for a given dataset to achieve best efficiency and substitution result. Since we have already projected all the images and their corresponding bounding box masks to the same cylindrical surface and coordinate system, to substitute the moving objects in a frame (denote as frame A), we search the neighboring frames within the search radius and pick out the pixels of the same location that have not been masked out. We then assign a weighted interpolation method to interpolate these pixels. The weights are inversely proportional to the absolute distance from these neighboring frames to frame A. Before interpolation, we normalize the weight vector to a total weight of 1 such that pixel intensity is preserved.

At last, we replace the pixels of moving objects in frame A with a blending of pixels from neighboring frames multiplied by their corresponding normalized weight. The result of the substitution algorithm is shown in Figure 4. As shown in the figure, the substitution algorithm can replace pixels of the detected moving object with minimal artifacts.

2.4. Image stitching

After removing the moving objects and substitute the corresponding pixels from neighboring frames. We stitched these modified images together to get the panorama. Since we already mapped the images to the same cylindrical surface and align them using horizontal translations, we just need to blend the overlapping parts of the consecutive images to merge them into a panorama. We apply alpha blending in this step. To better demonstrate the effect, here we compare the results got from our method with results got using AutoStitch [1] algorithm. The results are shown in figure 5. As we can see in the figure, ghost artifacts are very obvious in the left image, while in the right image the moving objects as well as the ghosting artifacts are effectively removed.

In fig 6, we also give a complete panorama of a dataset of 400 consecutive images we collected. As we can see, moving objects and ghosting artifacts are perfectly removed.

2.5. Movement reconstruction

Finally, we generate videos containing reconstructed movement of moving objects. The background of the video is the complete panorama we get and is fixed. For each frame, we paste the corresponding moving objects onto the background. Take the dataset we used for example: we first generate the panorama using the method described in the previous sections. Then we set this panorama as the background image of the video. The dataset has 400 consecutive images, so the video will have 400 frames. For the *ith* frame, we paste moving object of the *ith* image on the corresponding location of the fixed background. Repeating this process for all frames give us the video that reconstructs the object movement. Here we show a sample of video frames in fig 7.

This reconstruction method can be used in virtual reality applications. Suppose someone is watching the panorama using a card board. We can show different parts of the moving object trajectory according to the deflection angle of his/her head. For example, the panorama is formed by 360 pictures. So each degree of angle corresponds to one picture. If the person heads the direction of *ith* degrees, then we show the moving objects in the *ith* picture. Then when the person heads around, he/she can the reconstructed moving path of moving objects.

3. Conclusion and Analysis

In this paper, we proposed a new method to remove moving object in stereo panorama images and also reconstruct the object movement across the panorama. This method is relatively robust with moving objects with various sizes, various colors and various numbers(multiple objects in one scene). We believe that this method can effectively solve many ghosting artifacts in panorama images and the motion reconstruction procedure can be helpful to display images with object motion in VR devices.

After testing with various datasets we found our proposed method works for most scenarios. However, we also observed some situations where the method cannot perform perfectly.

The object removal method relies heavily on the detection quality of fast-MCD algorithm, if a moving object is very small or hard to distinguish from the background then Fast-MCD method is not able to detect the whole object such as upper body of a person. In this case the substitution method will not be able to replace all moving objects in each frame.

In another case, If a moving object such as a person

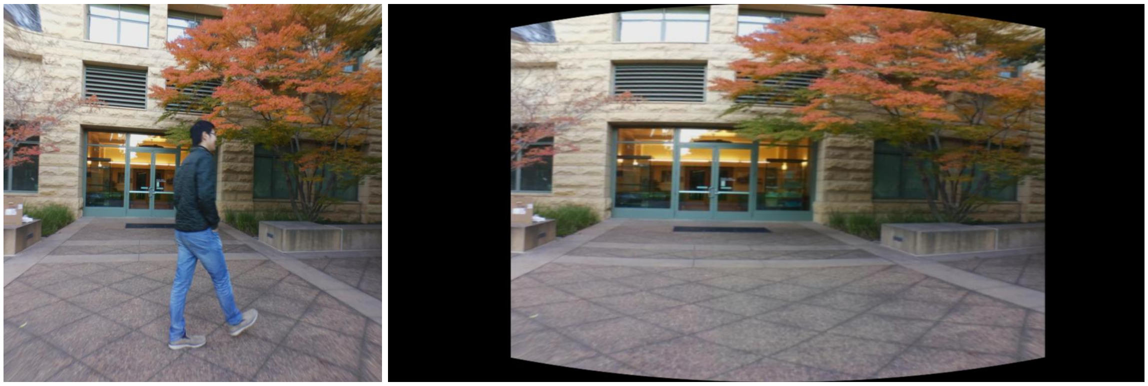

Figure 4. Substitution Result for a Single Frame

For the above example we used a search radius of 5, left side is the original image, right side is the image after substitution(projected on common cylindrical surface).

Figure 5. Moving Object Removal Result

left side is the result got from AutoStitch algorithm, right side is the result got from our method.

stays in the same location without moving a lot for many frames, Fast-MCD will treat the person as part of the background. If the person then starts to move, the substitution algorithm may take the background previously blocked by the person as foreground(moving object). In this case the resulting panorama may leave some artifacts.

Finally, our substitution algorithm requires the neighboring frames to have enough overlap. If the number of frames per 360 degree is too few then the algorithm may not able to proper pixels to perform substitution.

4. Acknowledgement

This project receives support from Stanford Computational Imaging Group, especially from Professor Gordon Wetzstein, Donald Dansereau and Aniq Masood. We would like to thank them for their valuable insights throughout the project and instructions of how to capture and process data with the 360 Stereo Panorama Image Camera.

References

[1] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

[2] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[3] S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High dynamic range video. *ACM Trans. Graph.*, 22(3):319–325, July 2003.

Figure 6. Final Panorama After Stitching

[4] E. A. Khan, A. O. Akyuz, and E. Reinhard. Ghost removal in high dynamic range images. In *2006 International Conference on Image Processing*, pages 2005–2008, Oct 2006.

[5] C. Tomasi and T. Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[6] Y. Wan and Z. Miao. Automatic panorama image mosaic and ghost eliminating. In *2008 IEEE International Conference on Multimedia and Expo*, pages 945–948, June 2008.

[7] Y. Xiong. Eliminating ghosting artifacts for panoramic images. In *2009 11th IEEE International Symposium on Multimedia*, pages 432–437, Dec 2009.

[8] K. M. Yi, K. Yun, S. W. Kim, H. J. Chang, and J. Y. Choi. Detection of moving objects with non-stationary cameras in 5.8ms: Bringing motion detection to your mobile device. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 27–34, June 2013.

Figure 7. Video frames of movement reconstruction