Generating Panorama Photos

Yining Deng* and Tong Zhang
Hewlett-Packard Labs, 1501 Page Mill Road, Palo Alto, CA 94304

ABSTRACT

Photo or video mosaicing have drawn a lot of interests in the research field in the past years. Most of the existing work, however, focuses on how to match the images or video frames. This paper presents techniques to handle some practical issues when generating panorama photos. We have found from the experiments that a simple translational motion model gives more robust results than affine model for horizontally panned image sequences. Realizing the fact that there would always be some misalignments between two images no matter how well the matching is done, we propose a stitching method that finds a line of best agreement between two images, to make the misalignments less visible. Also shown in this paper are methods on how to correct camera exposure changes and how to blend the stitching line between the images. We will show panorama photos generated from both still images and video.

Keywords: Photo panorama, image stitch, video mosaic

1.Introduction

People have always been fascinated about capturing the entire view of the scenes. Before the era of digital cameras, wide-angle view is captured using special optical lens. However, these lenses are usually mounted on SLR cameras which most people do not have. Plus, lens distortion is often introduced in these pictures and even with the wide-angle lenses we are still unable to obtain the full 360 degree view. A new generation of digital cameras based on the line scanning technologies, such as the ones produced by Panoscan.com, allows us to capture incredible 360 degree views of the scenes. The pictures produced from these cameras often have very high quality. The drawback is that those cameras are very expensive and far beyond the reach of average consumers.

The idea of using a simple camera to take a few pictures or pan a video of a scene, and being able stitch those pictures or video frames using image processing techniques, to produce a panorama photo, has drawn a great amount of research interests in the past years. Many new ideas were proposed in the literature in the late nineties, such as [1-9]. Lately, there have been some new interests in stitching video frames directly from MPEG data [11-13] and this involves estimation of the global motion from the MPEG motion vectors [10].

One of the major advantages of using image processing is affordability as anyone can install a piece of software on a PC and is able to process the data to produce the panorama photos. However, since the images are taken at multiple moments while the camera is panning around the scene, they need to be registered to each other in order to obtain the final result. This registration or motion matching has proven to be a difficult problem and that is what most of the research work in this field has focused in the past.

In a perfect world where we can have the camera placed horizontally and panning exactly around its focal point, if we know the tilt angle and how much the camera has panned, we can warp all the images to a sphere based on the focal field of view of the camera model. In the case when the tilt angle is zero, a cylinder is a good substitute for the sphere. Theoretically, all the images can be warped to such a common reference sphere or cylinder and we can then reproduce the entire field of view from this sphere or cylinder. This is known as spherical or cylindrical warping, a well-known technique in the textbook.

In reality, without knowing any camera angles and camera focal field of view, a correct estimation for this kind of warping is difficult to obtain. Instead, people have been mostly trying to use 2D planar matching techniques to obtain

*yining.deng@hp.com

relative matching between two images, such as affine matching. However, without correct warping, there would always be errors during introduced during the matching due to the perspective changes from the 3D scene to the 2D image. One interesting idea is to use only narrow center strips of video frames [3]. This approach works for high frame-rate video data. It, in fact, is mimicking the line-scan cameras mentioned earlier. The line-scan cameras scan one vertical line at a time and there is no geometric distortion. However, issues still remain. What if there are objects moving in the scene? The strips would likely cut the moving objects into parts. Plus, this approach would not work on still photo stitching.

The work in the literature has mostly focused on how to match images in the general cases of transformation, i.e. in the case when the camera pan, rotate, and tilt in any directions. We realize that no matter how well the matching is done, there will be some misalignments between the images. This could happen because the camera is drifting away from its initial focal point position, as is always the case for hand-held cameras. It could also happen because there are moving objects in the scene or because of the 3D to 2D transformation that can not be accounted by 2D image matching. So instead, this work rather focuses on the stitching side to avoid such kind of misalignments or make them less visible.

We do assume that people capture the data with the panorama photos in mind. This means that the camera is held roughly horizontally and the panning is done consistently along the horizontal or vertical direction, rather than the general scenario when the camera can be moved in any directions. We will present some interesting observations in motion matching assuming this panorama mode.

We will also show how to deal with some other practical issues in generating good panorama photos. One problem we have faced is the change of exposure in camera settings, since most cameras are in automatic mode and adjust to lights when taking pictures. As a result, one picture could be significantly brighter than another. And this needs to be corrected before the final stitching process. Another issue is on how to blend two images. We have found that a combination of Gaussian pyramid and alpha blending methods produce better results than either one by itself.

2.General Scheme

Our general scheme can handle both image and video data. We use a two-pass approach. In the first pass, images are matched to each other. In the second pass, these matched images are corrected based on their relative intensity gain and then stitched together. If the input data is video, the first pass also selects a subset of video frames for final stitching and discards the rest of the frames. The criterion to select which frame is based on the overall estimated displacement between the current frame and the previous selected frame which acts as a reference. If there is a significant displacement, the current frame is selected and set as new reference. This speeds up both the motion estimation and stitching process. In the motion estimation stage, only coarse-level motion estimation on the low-resolution versions of the video frames is needed to give a rough estimation of how far the two frames are apart. If the frame is selected as the new reference, refined motion estimation can be done at the original-resolution. This reduces the amount of motion estimation time. In the stitching process, since we have dropped most of frames and only have to stitch a small subset, we can also stitch them much faster.

The diagrams for the two-pass process are shown in the following figures. Images or video frames are first fed into motion estimation block. Then relative intensity gain is calculated on the overlapping area based on the motion estimation. This step is used to correct the changes in pixel values due to camera exposure variations. Another advantage of having two-pass approach, in addition to the speed-up in processing video data, is that we can estimate relative intensity gains for all the images and correct average intensity of each image to an average reference. With one pass process, we can only set the first image as the reference. Sometimes, the first image is too bright or too dark and this causes the rest of the images to be too bright or too dark as well.

Another advantage of having two pass is that we can determine the order of how we stitch those images. Since we have relative positions of those images and we can reposition them according to a global spatial reference point. It is not necessary to stitch the images in terms of their temporal order or how they are fed into the system. There have been studies on how to stitch image based on their spatial patterns, in order to obtain better overall matching among all the images. In our current approach, the stitching order still follows the same as the input, because we are assuming that acquired data is intended for panorama stitching. Therefore, the camera pan is along either horizontal or vertical direction. We are planning to add grid-style layout image stitching in the future.

One special case can be handled well in our system is the zigzag panning pattern when the images or video are taken along horizontal direction and then in reverse direction at different camera tilt angle to catch more view along the vertical direction. An example of such a zigzag pattern is shown in the following figure. It can be seen that while it is easier to estimate matching between the adjacent images, there will be likely be big mismatches between the top row and bottom row when the camera comes back to the same position after the panning. One way to match those images is to use the global reference system as mentioned earlier and re-estimate all the motion vectors. However, complexity of such an approach is high and there is no guarantee that a robust estimate can be obtained.

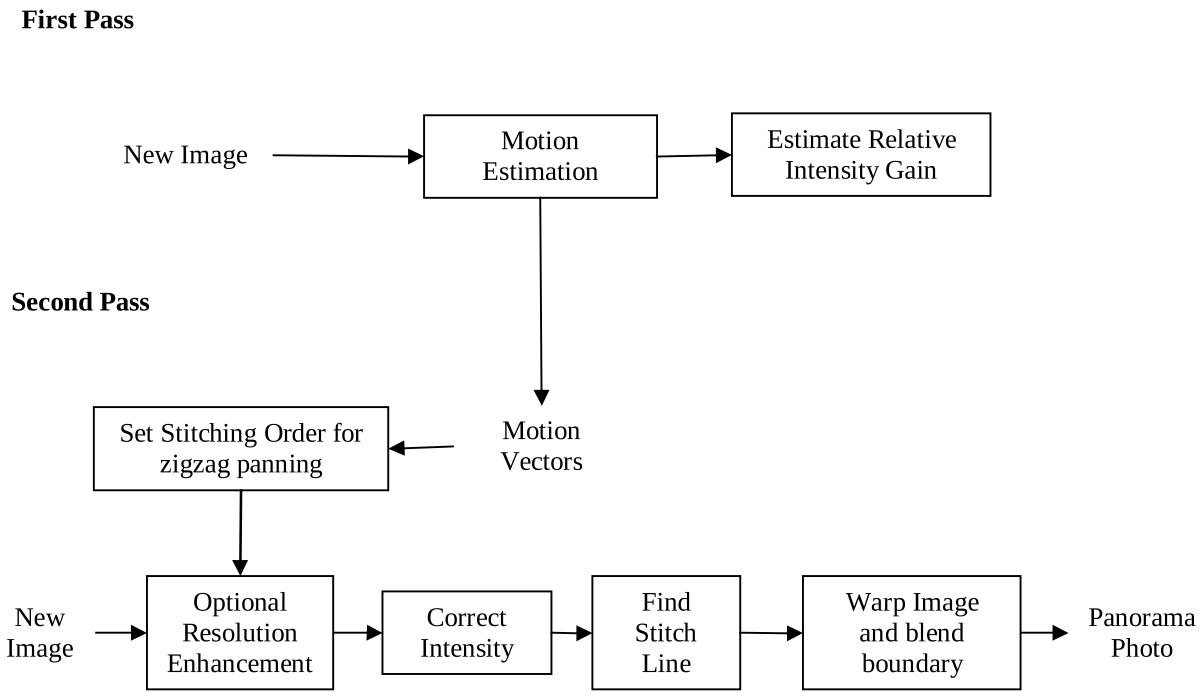
First Pass
New Image
Motion Estimation
Estimate Relative Intensity Gain
Second Pass
Motion Vectors
Set Stitching Order for zigzag panning
New Image
Optional Resolution Enhancement
Correct Intensity
Find Stitch Line
Warp Image and blend boundary
Panorama Photo

Fig. 1 Two pass scheme to generate panorama

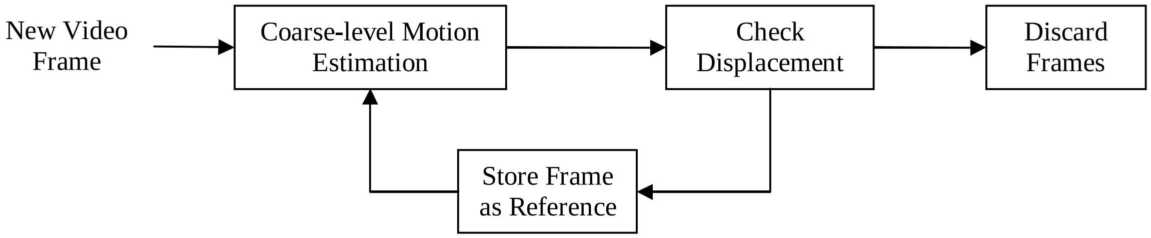

Fig. 2 Motion Estimation for video data.

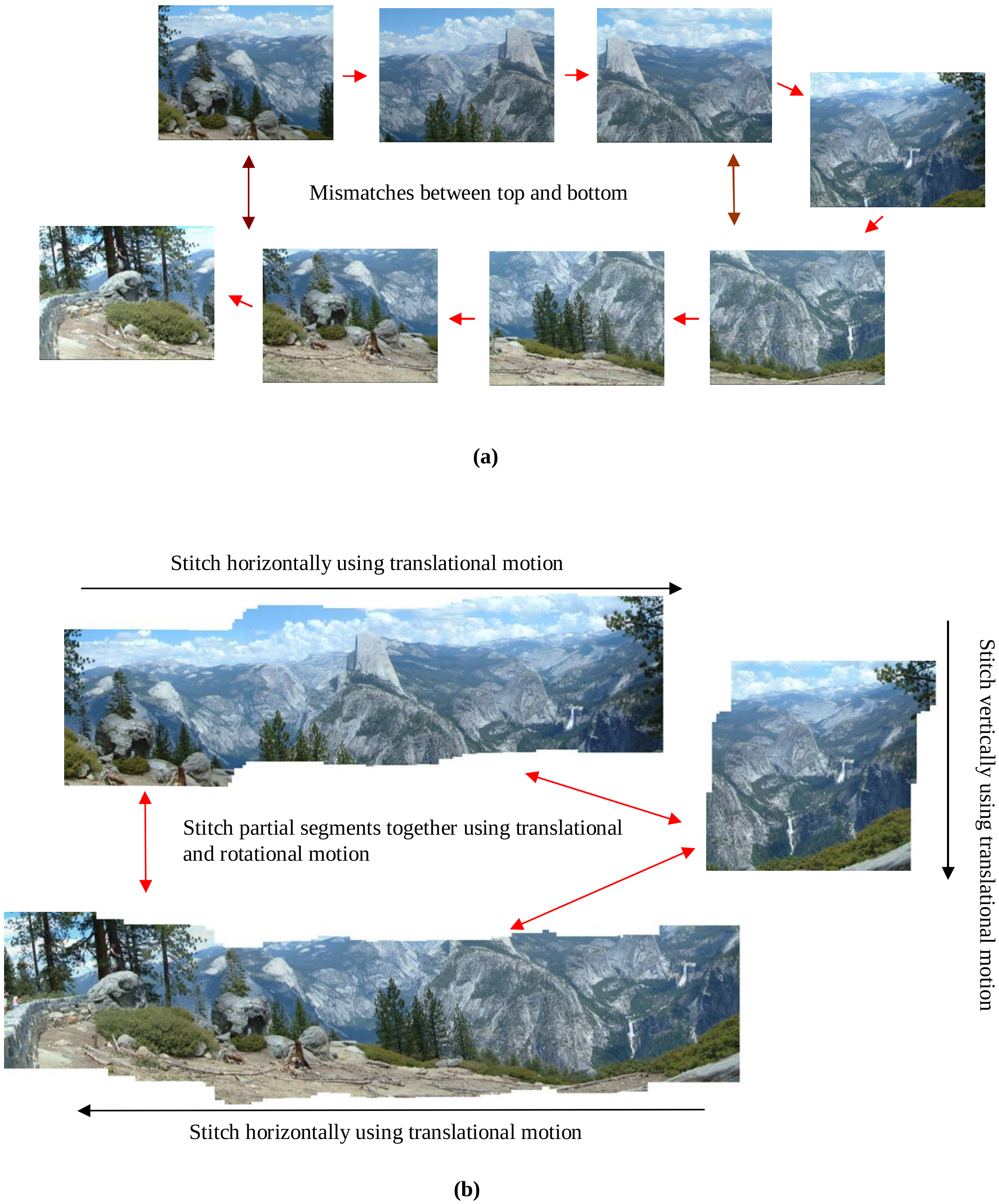
Mismatches between top and bottom
(a)
Stitch horizontally using translational motion
Stitch vertically using translational motion
Stitch partial segments together using translational and rotational motion
Stitch horizontally using translational motion
(b)

Fig. 3. (a) Zigzag panning pattern. (b) Stitch by separating images into groups.

Our approach is simple as shown in the above figure. We separate all the images into separate groups based on their motion direction. Images within each group are stitched first and this can guarantee good matching among those images. The mosaiced images from all the groups are then match to each other. Within each group, only translational motion is used as discussed in Section 3. To register the mosaiced images between the groups, higher-complexity motion model is used to obtain better match.

In addition to all the necessary image matching and stitching components, an optional resolution enhancement component has been added to the system. This is specifically for video data, where the highest resolution is only VGA at 640x480. In order to achieve good printing quality for the panorama photos, resolution enhancement can be performed before the second pass stitching process.

3.Motion Matching

We have used global frame matching to estimate the spatial transform between two images. A simple translational model turns out to be good enough when the main camera motion is only panning and the scene is remote so that the perspective distortion is less severe. The motion matching searches through a window and finds the displacements along x and y axes that give the minimum error. For image sequences, the size of the window is the entire image, because we do not know where the matching location is. For video, the search window is much smaller because the motion vectors of the previous frame can be used as an initial estimation. Also in the video, instead of matching two consecutive frames, we are matching the current frame with the reference frame being selected earlier. The motion matching is done in a typical hierarchical way where the images are downsampled to a very low resolution to speed up the process. And the motion vectors are gradually refined at higher resolutions.

We have noticed in our experiment that in the absence of sphere or cylindrical warping, affine matching gives more accurate results for a small number of images. However, the perspective distortions, due to the recording of the 3D scene to a 2D imaging plane, would cause the affine model to generate an incorrect rotation angle to fit the data. This is because that the camera usually has a small tilt angle. Putting together all the camera imaging planes at different panning angle, they form a strip on a facet cone. Imagining flattening out that strip, it will bend either up or down depending on whether the camera is tilted up or down. For a long image sequence or video, a typical result of using affine model would have a bending artifact, like the one shown in the following figure. This is quite undesirable. On the other hand, the translational model forces the images to go horizontal direction and surprisingly achieve better results than the affine model. Of course, without a correct warping that takes care of the camera tilt angle, even if we only use the translational model, we see the stitched image slowly drifting up or down for a very long image sequence or video. Some methods have been proposed in the literature to solve this problem if the images wrap around 360 degrees and matching accuracy can be double checked. The accumulated motion errors can then be compensated by distributing the total vertical shift evenly among all the images.

The exposure variation is one major cause for inaccurate motion estimation. That is one of the disadvantages of using the frame matching technique, which in general cases is more robust than feature or point based approaches. Because the exact motion displacement is unknown, there is no way of knowing the relative intensity gain between two images before motion estimation. In this work, correction of intensity gain is done at each search step in the search window. The relative intensity gain is first calculated between the overlapping areas in the two images. One of the overlapping areas is then corrected to have the same average intensity as the other one. Their pixel average difference is calculated for this set of x and y motion vectors. The one that gives the minimum error based on adjust intensity values is determined to be final motion vectors.

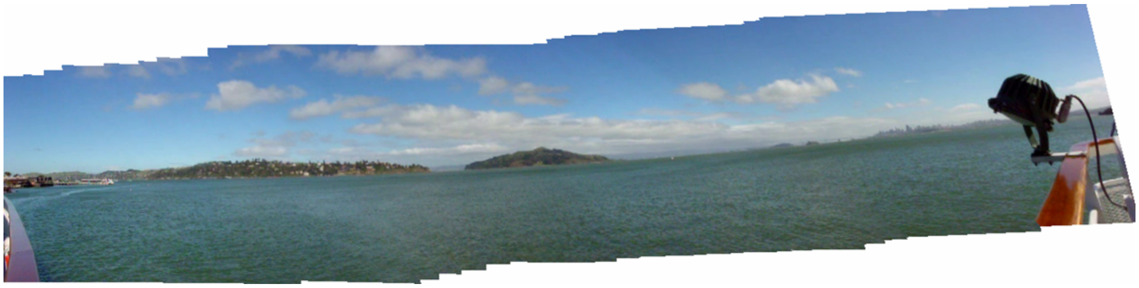

Fig. 4. Bending artifact using affine model.

4.Stitching

4.1. Finding Stitching Line

As mentioned earlier, stitching is an important step to generate good panorama photos. In order to avoid misalignments and objects moving in the scene, we propose a method here that attempts to stitch across the best agreement. This method involves two steps. First it finds a strip in the overlapping region that gives the minimum error. Second, it finds a stitching curve within this strip that bypasses all the possible misalignments. The first step might not be applicable if the two images are matched diagonally.

Fig. 5 best illustrates this approach. Two images are aligned to each other but objects are moving the images and the alignments are not perfect at pixel level. Strips are drawn in each image and compared to the ones at the same location in the other image. The one with the least difference is chosen because it indicates that there are fewer errors in this region. It can be seen that there are still a few mismatches in these two strips. The next step is to find a curve that cuts through the matched pixels and bypasses those mismatches. The final stitched result shows no misalignments.

The approach taken in Step 2 is similar as watershed methods. It works as follows. The differences between two strips are calculated and smoothed out by a Gaussian filter. This forms a difference image. The values in this difference image are sorted from highest to lowest. Higher value means that there is more difference at this location and a potential mismatch we should try to avoid. The method starts with the pixel with highest intensity, and check to see if it is neighboring to any of two original images. If not, that pixel is classified to a third category. The next highest valued pixel is then checked. If the pixel is neighboring to one of the original images, it is assigned to that image. All the third category pixels neighboring to this pixel are assigned that image as well. After we go through all the pixels in the difference image, they will be assigned to either one of the original images. Because of the way high-valued pixels are grouped together, the resulting boundary between the two images does not cut through any high-value pixels and therefore is less likely to cut through misalignments as well.

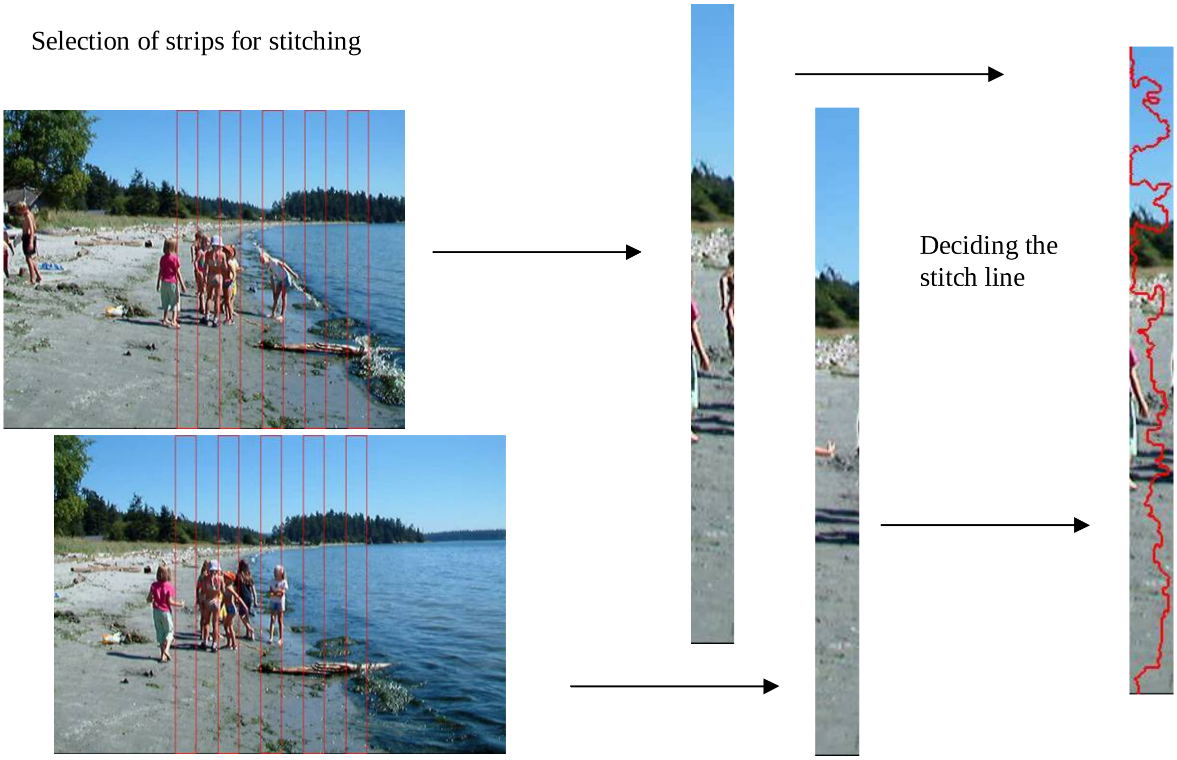

Fig. 5(a) Stitching across the best agreement.

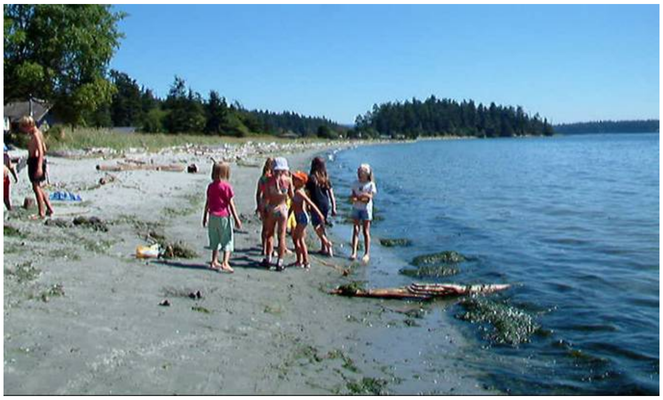

Fig. 5(b) Stitching Result.

4.2. Blending

After the stitching line is determined, blending is applied across the stitch so that the stitching would be seamless. There are two popular ways of blending the images. One is called alpha blending, which takes weighted average of two images. The weighting function is usually a ramp. At the stitching line, the weight is half and half, while away from the stitching line one image is given more weights than the other. The cases that alpha blending works extremely well is when image pixels are well aligned to each other and the only difference between two images is the overall intensity shift. Alpha blending will merge two images seamlessly. However, if the images are not aligned well, the disagreements will show in the blended image.

Another popular approach is Gaussian pyramid [15]. This method essentially merges the images at different frequency bands and filters them accordingly. The lower the frequency band, the more it blurs the boundary. Gaussian pyramid blurs the boundary while preserving the pixels away from the boundary. It does not work well, however, if the two images are at significantly different intensity levels. The transition is not as smooth as alpha blending for this case.

In this work, a combination of alpha blending and Gaussian pyramid is used. The images are first downsampled to lower resolutions using the Gaussian pyramid approach. At the lowest resolution, alpha blending is used to merge the two images. The difference in intensity between the images can be blended well at the lowest frequency band. At higher frequency band, no alpha blending is applied so that the misalignments, usually in the form of double edges, are avoided.

5.Results

We show here some examples of panorama photos generated from both still photos and video. Fig. 6 and 8 are generated from video and Fig. 7 is generated from the still images. It is worth to mention that in both Fig. 6 and 7, overall brightness changes from images to images. It can be seen from the middle portion of Fig. 7 that the sky is so bright that it is out of the pixel value range and turns to white. In both figures, intensity correction is done and blending is added to smooth out the transition between the images.

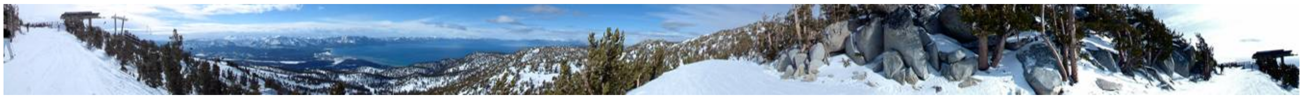

Fig. 6. A 360 degree view of lake Tahoe. Generated from video.

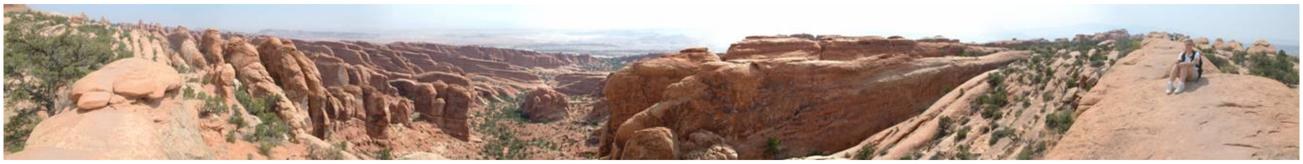

Fig. 7. Arch national park in Utah. Generated from still photos.

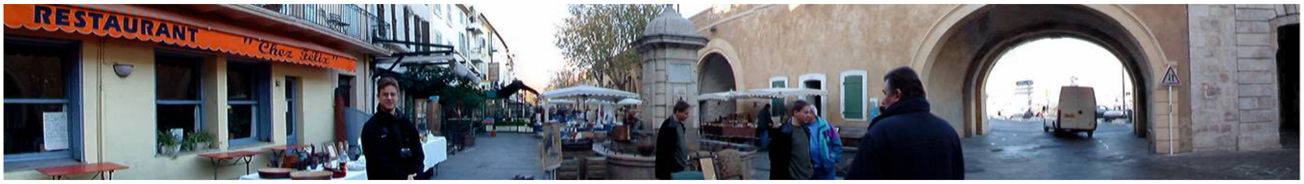

Fig. 8. Streets of London. Generated from video.

In Fig. 8, people are moving in the scene, which is the reason that there is one person who appears twice in the stitched photo. Normal stitching would cut the moving people. By using the proposed stitching method, we obtain a smooth panorama photo without any visually disturbing artifacts. This approach does not prevent a moving object from appearing more than once in the image. But then multiple appearances make the picture more dynamic and more interesting.

6.Conclusions

This paper presents techniques to handle some practical issues when generating panorama photos. Realizing the fact that there would always be some misalignments between two images no matter how well the matching is done, we propose a stitching method that finds a line of best agreement between two images, to make the misalignments less visible. Also shown in this paper are methods on how to correct camera exposure changes and how to blend the stitching line between the images. In the future, we plan to add grid-layout stitching to the system.

REFERENCES

1. R. Szeliski and H.-Y. Shum, "Creating full view panoramic image mosaics and texture-mapped models," SIGGRAPH, p. 251-258, 1997.

2. H.-Y. Shum and R. Szeliski, "Construction and refinement of panoramic mosaics with global and local alignment," ICCV, p. 953-58, 1998.

3. S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet, "Mosaicing on adaptive manifolds," PAMI, p. 1141-54, October, 2000.

4. B. Rousso, S. Peleg, I. Finci, and A. Rav-Acha, "Universal mosaicing with pipe projection," ICCV, p. 945-952, 1998.

5. B. Rousso, S. Peleg, I. Finci, "Mosaicing with generalized strips," DARPA Image Understanding Workshop, pp. 255-260, May 1997.

6. S. Peleg and J. Herman, "Panoramic mosaics with VideoBrush," DARPA Image Understanding Workshop, pp. 261-264, May 1997.

7. S. Peleg and J. Herman, "Panoramic mosaics by manifold projection," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 338-343, June 1997.

8. S. Mann, C. Manders, and J. Fung, "Painting with Looks: Photographic images from video using quantimetric processing," ACM Multimedia, 2002.

9. H. Sawyhney and S. Ayer, "Compact representation of video through dominant and multiple motion estimation," PAMI, p. 814-830, August, 1997.

10. D. Capel and A. Zisserman, "Automated mosaicing with super-resolution zoom," Proc. of CVPR, page 885-891, June, 1998.

11. M. Pilu, "On using raw mpeg motion vectors to determine global camera motion," SPIE Electronic Imaging Conference, 1998.

12. Y. Li, Li, Xu, G. Morrison, C. Nightinggale, and J. Morphett, “Robust panorama from mpeg video,” vol. I, p. 81-84, ICME, 2003.

13. J.W. Hsieh, “Fast stitching algorithm for moving object detection and mosaic construction,” vol. I, p. 85-88, ICME, 2003.

14. M. Robertson, “Mosaic from MPEG-2 video,”, SPIE Electronic Imaging Conference on Computational Imaging, #5016-31, 2003.

15. E. Adelson, C. Anderson, J. Bergen, P. Burt, and J. Ogden, “Pyramid methods in image processing,” RCA Engineer, 29-6, 1984.

16. Y. Altunbasak, A. Patti, O. King, and E. Miloslavsky, “CAST: Camera scanning technology,” Hewlett-Packard Labs technical report, 1999.