FAST STITCHING OF VIDEOS CAPTURED FROM FREELY MOVING DEVICES BY EXPLOITING TEMPORAL REDUNDANCY

Motaz El-Saban, Mostafa Izz and Ayman Kaheel

Cairo Microsoft Innovation Lab

ABSTRACT

We investigate the problem of efficient panoramic video construction based on time-synchronized input video streams. No additional constraints are imposed regarding the motion of the capturing video cameras. The presented work is, to the best of our knowledge, the first attempt to construct in real-time a panoramic video stream from input video streams captured by freely moving cameras. The main contribution is in proposing an efficient panoramic video construction algorithm that exploits temporal information to avoid solving the stitching problem fully on a frame by frame basis. We provide detailed experimental evaluation of different methodologies that employ previous frames stitching results such as tracking interest points using optical flow and using areas of overlap to limit the search space for interest points. Our results clearly indicate that making use of temporal information in video stitching can achieve a significant reduction in execution time while providing a comparable effectiveness.

Index Terms— Efficient Stitching, Video Stitching.

1. INTRODUCTION

The goal of panoramic video construction is to construct a single panoramic video output by stitching together N input video streams. This technology is important in several domains, such as security (through surveillance cameras) and entertainment. In the case of security applications, the cameras are mostly fixed in location, while in entertainment applications video feeds, simultaneously captured by different users in the same scene, are obtained by free-to- move capturing devices. The latter is particularly interesting given the wide availability of consumer-level capturing devices, such as mobile phones and digital camcorders. This ubiquitous presence increases the chance that two or more capturing devices are simultaneously shooting the same scene, possibly from different angles. In this paper, we investigate the more challenging problem of efficient construction of panoramic video output from independently, but simultaneously, captured video streams. Our target is an efficient technology, possibly real-time, for automated panoramic video construction to enable applications such as streaming videos from multiple capturing devices and allowing another party to watch the constructed panoramic video in real-time. This would offer the remote party a better viewing experience, in terms of wider field-of-view. In this paper, we constrain ourselves to stitching two incoming video streams, leaving the scaling aspect as a topic for future investigation.

The main contributions of this paper are: 1) proposing novel algorithms for usage of temporal information for efficient panoramic video construction from two independently and freely moving capturing devices and 2) experimentally validating that usage of temporal information achieves a significant reduction in execution time while maintaining a comparable effectiveness.

The paper is organized as follows. Section 2 discuses related work. An overview of the proposed algorithm is given in Section 3. In Section 4, different methodologies are presented for incorporating temporal information. Section 5 presents our experimental results. Conclusions and future research points are given in Section 6.

2. RELATED WORK

Very little work has been done on the front of producing a video composite out of multiple video streams, especially for the case of freely moving capturing devices, as opposed to the case where the cameras are fixed[3]. The work in [1][1] described fast techniques for stitching videos, however, the authors only showed objective alignment results for pure translation transformation videos which is not a realistic situation for freely moving cameras. In our case, we report objective results for the full affine case on a large video dataset collected under varying conditions. The authors in [8][11] proposed a system for stitching videos streamed from mobile phones, however, the proposed system treats the video stitching problem as essentially an image stitching problem on individual frames independent of each other. Solving the image stitching problem has been extensively studied in the literature[2][15]. Considering video stitching as a series of image stitching sub-problems completely ignores the potential benefits of considering temporal redundancy. The authors in [5] propose a video stitching system for streams coming from webcams. Their system requires an initialization phase that is not real-time and is needed whenever the webcams positions change.

3. OVERVIEW OF THE PROPOSEDVIDEO STITCHING ALGORITHM

The starting point of the proposed video stitching algorithm is a set of two time synchronized video streams. For the first pair of frames, the stitching algorithm implements the two main steps, alignment and compositing as is normally done in image stitching algorithms; for the alignment part, we detect IPs and form corresponding descriptors using the recently proposed SURF descriptor [7] which has been shown to be faster than SIFT [6] while having comparable detector and descriptor

performance (SIFT shown in many studies to outperform other detectors and descriptors [9] and was utilized for image stitching [12]). Being fast is a key requirement for achieving an efficient video stitching algorithm. After IP detection and description is performed on the pair of video frames, IP matches are identified between frame pairs using Euclidean distance in an exhaustive manner. A perspective geometric transform is estimated for best alignment using a least squares approach and a RANSAC [16] procedure to reduce outliers' effect. Since the main point of this work is the use of time information for performing efficient video stitching, we have resorted to the use of a simple, yet acceptable in most cases, approach for blending, namely feathering and using a flat compositing surface, as opposed to more elaborate but time demanding blending techniques [4][14]. Feathering is based on pixel weighting using pixel distance from image boundary.

For subsequent frames, we utilize previous frames information to aid stitching in the current frames, as explained in the next section. It is important to note that for each pair of frames, we test whether these two frames are stitchable or not (i.e. the algorithm judges it can stitch them or not). This is crucial for the end user experience, since it is quite unpleasant for a user to watch a composite video where the stitching is done incorrectly. The inability of the algorithm to stitch frame pairs can be due to either: a) there is little or no overlap between frames, b) there is overlap, but the algorithm computes an incorrect geometric alignment. We proceed with the regular phases of stitching, up to computing a geometric transform. We signal two frames as stitchable if we find a minimum percentage of IPs agreeing with the estimated geometric transform.

4. UTILIZING TEMPORAL REDUNDANCY

In this section we investigate three approaches for enhancing the efficiency of the video stitching process using time information with little or no effect on effectiveness:

1) Exploit the area of overlap from previous frame
2) Use motion vectors for transformation calculation
3) Track interest points (IPs) from previous frames

We started this investigation by a study of the time spent in the various stitching steps to identify the most time-consuming steps to target during algorithm development. Based on conducted experiments, it was found that IP detection and descriptor computation (for SURF) takes more than 80% of the time in stitching two frames. Hence, most of the investigated approaches in this paper are aiming at minimizing the time spent in IP detection and description. In the subsequent discussion, frames at the current time step will be referenced as frames (***n***) and frames from the previous time step will be referenced as frames (***n-1***).

4.1. Area of overlap from previous frames

The premise in this approach is that knowledge of the area of overlap (in case that the video stitching algorithm declares frames as stitchable) from frames (***n-1***) can potentially limit the search space for IPs in frames (***n***). The stitching algorithm performs the same steps as in the case of the first frame pairs but instead of trying to detect IPs in the new whole frame, they are detected in the area of overlap found in frames (***n-1***) only, plus some buffer region in the frames (***n***) (best value experimentally determined to be a 20 pixel band). The efficiency of the buffer-based approach performance is inversely proportional to the size of the overlap area between frames (***n-1***) and ***(n)***.

4.2. Using motion vectors for speed up

This approach was discussed earlier in [1]. The basic idea in this approach is to use global motion estimates of IPs between frames (through motion vector estimation) to avoid re-computation of the new transformation matrix. Some improvements are applied to the original algorithm in [1] including motion vectors estimation using SURF descriptors matching. Our conducted experiments have shown that it is better to use the first frame in computing motion vectors for other frames. Besides, we introduce another improvement by limiting the number of created descriptors to speed up the stitching process by generating descriptors one by one and matching them until we get certain number of matches between current frame and first frame. The found matches are used to calculate motion vectors. Finally as in [1], the geometric transformation matrix in frame (n), R_n, is updated using the transformation matrix at first frame, R_1 and global motion estimates, V_A and V_B, in both pair of video frames at the current frame: $R_n = V_A R_1 (V_B)^{-1}$

4.3. Using Optical Flow for tracking IPs

The main idea behind this approach is to avoid re-computation of the expensive stages of IP detection and descriptors in the alignment process between incoming frames. For that purpose, we track IPs from frames (***n-1***) to find their 2D locations in frames (***n***) using Lucas-Kanade optical flow[10]. Once we find the new location of IPs in frames (***n***), their descriptors are obtained from frames (***n-1***) in order to avoid re-computation. Tracked IPs are filtered by discarding foreground moving objects, which leads to a more stable stitching using background IPs. IP filtering is based on finding the 2D global motion (d_x, d_y) of all IPs by averaging 2D motion parameters of all IPs and then removing IPs falling outside the 2σ range from the mean value in either d_x, or d_y. It is worth noting that beyond IP filtering process, the estimated global motion is not used in the stitching process itself in frames (***n***).

Using IP descriptors from previous frames is not perfect as it neglects illumination and 3D viewpoint variations and may lead to error accumulation over time. Hence, a criterion is used to signal when we require to do an image-based stitching for a given frame pair (without usage of previous frames information). The suggested criterion is based on the number of IPs that can be successfully matched between the frame pairs of frames (***n***).

5. DATA SET AND EVALUATION RESULTS

Since there are no existing suitable data sets for testing our proposed techniques, we have resorted to collecting our own dataset using commonly available mobile phones with video capturing capabilities. Human data collectors were asked to capture time-stamped videos simultaneously at multiple locations, at different time of the day and while performing various camera motions to enable a general assessment, for

varying video content, shooting distance, lighting condition and camera motion. It is worth noting also that instructions were given to the shooters to try to shoot for a common object so as to maximize chance of overlap. Videos were captured using mobile phone cameras with a CIF video resolution (352x288) and with an average frame rate of 11 frames/sec. Tables 1, 2 and 3 highlight some of the statistics of the dataset based on time of capture, object motion, and camera motion respectively.

Table 1. Dataset statistics based on time of capture

condition	Total number of frames used for evaluation
Day time	879
Night	297
Sunrise	99

Table 2. Dataset statistics based on objects' motion

condition	Total number of frames used for evaluation
Slow	484
Static	593
Fast	198

Table 3. Dataset statistics based on cameras' motion

condition	Total number of frames used for evaluation
Walking with both cameras	396
Tracking a moving object (car, ship, …etc)	198
In plane rotation for each camera alone	198
One static and the other converging /diverging from some object	187
Both cameras are static	98
Panning cameras in same direction	198

Example frames are shown in Fig. 1 to illustrate some of the variability within the dataset.

Fig. 1 Sample frames from the dataset

Upon data collection, a human judge was asked to manually label a sample of 1275 pairs of frames. The human labeler was asked to mark the corresponding video frames as correctly stitched by the algorithm or not (both alignment and blending); and whether they could be stitched by a human or not. Although the output of our system is a stitched video and not frames, we opted to label individual frame pairs to get an upper bound on stitching errors, since a human can miss small stitching issues if watching an output video compared to the case of watching a single video frame.

The evaluation aims at measuring precision/recall values for stitching frame pairs as well as stitching time for various descriptors and time information usage methodologies. Note that we are not evaluating the matching capability of the different descriptors (that has been already extensively studied in [9] among other works). Experiments were conducted on an Intel® Core™2 Duo CPU E8400 @ 3.00GHZ, with 4.0 GB RAM. In the dataset used, the total number of possibly stitched frame pairs is 892 pairs leaving a relatively small percentage of non-stitchable pairs, as the experiments have shown that a negligible fraction (<5%) of the non-stitchable frames were judged, erroneously, as stitchable by the algorithm. Table 4 shows the evaluation results for the complete data set using different time information methodologies and descriptors.

Table 4 Performance evaluation of different methodologies for video stitching

	recall	precision	Average time (ms)
SIFT + no time info usage	0.37	0.46	414
SURF + no time info usage	0.32	0.48	162
SURF + Motion Vector estimation	0.23	0.38	128
SURF + overlapping region	0.27	0.48	149
SURF + Optical Flow	**0.32**	**0.47**	**103**

It is obvious that SIFT has better recall compared to all other techniques, but the required time is much more than all others. Our aim here is to find a good compromise between accuracy and execution time. Hence, we choose the SURF descriptor as our baseline for the experiments as it is shows experimentally comparable precision/recall values as the SIFT with almost 40% of the execution time. From Table 4, SURF and optical flow method performed the best achieving 37% relative execution time reduction compared to our baseline with comparable accuracy. It is worth noting that we may need to do a pure image-based stitching on some frames, in case time information is not useful. The image-based stitching is invoked when the previous frames cannot be stitched or when the current frame inliers number is less than six (experimentally determined). On average, the image-based stitching algorithm was performed every 15 frames.

5.1. Results per shooting condition

In this section, more experiments are conducted to investigate the effect of content variability on precision/recall. Table 5 shows the variation in performance with respect to the time of capture. Results show that night time shooting performs worse than the

daytime/sunrise cases due to lack of interest points using artificial lights at night time. We further test the effect of different amount of object's motion on performance in while cameras were mostly fixed. As expected when objects are static, we get the best performance, followed by the faster ones.

Table 5 Effect of different capture times

condition	technique	recall	precision	time
daytime	SURF	0.33	0.47	156
	OPTICAL FLOW	0.34	0.46	97
night	SURF	0.14	0.27	199
	OPTICAL FLOW	0.15	0.30	136
sunrise	SURF	0.65	0.81	106
	OPTICAL FLOW	0.62	0.73	61

As for the shooting distance variations, results were as expected. The medium range (5 m up to 100 m) gave the best stitching performance with very far shooting leading to loss of features and very close shooting leading to high disparity in 3D viewpoints. The effect of close range shooting is most pronounced in indoor settings, where the performance is worst across all methods.

5.2. Error analysis

The most common stitching errors, incurred by the proposed algorithm, were classified into their different sources. These errors were 1) recall-type, and 2) precision-type errors. In recall-type errors, stitching errors are divided into geometric alignment and blending errors. Most of alignment errors are due to large 3D viewpoint changes between cameras, occlusion occurrences between views and compression artifacts, accounting for 20%, 25% and 14% respectively. Compression errors are considered here since videos were captured using unavoidably limited resources mobile phones. These three errors sources are apart from the obvious alignment error reason when there is either little overlap in views or not enough texture or details. As for blending errors, these are only considered when there are no geometric alignment errors to isolate error sources. Blending errors account for 32% of missed stitchable pairs. This suggests the usage of a more elaborate blending technique that is not very computationally expensive. As for the precision-type stitching errors when the algorithm claims it can stitch pairs that are not stitchable by human, experimental evidence has shown that these are a negligible fraction (<5%) and are mostly due to similarly looking features in frame pairs.

6. DISCUSSION AND FUTURE WORK

In this work, we have presented a set of novel video stitching techniques utilizing time information for efficient panoramic video construction. We have shown that indeed time information can be utilized in building an efficient video stitching method without a severe compromise on precision/recall. Besides, we have conducted comparative experiments of the proposed methods and have shown that a combination of a global motion estimation method, based on optical flow, and the SURF descriptor yielded the best tradeoff between stitching accuracy and execution time on a newly collected dataset.

As far as future work is concerned, more investigation is needed to compensate for large difference in 3D viewpoints, occlusion problems discrepancies and better compositing techniques (as in [13]) while not adversely affecting efficiency. In addition, further investigations are warranted regarding video streams with different frame rates and focal lengths. Combining the different proposed methodologies of using time information in stitching would be another interesting future direction.

7. REFERENCES

[1] T. Shimizu, A. Yoneyama and Y. Takishima, "A fast video stitching method for motion-compensated frames in compressed video streams", International Conference on Consumer Electronics, 2006.

[2] R. Szeliski, "Image Alignment and Stitching: A Tutorial", MSR Tech Report (last updated 2006)

[3] Kolor autopano, http://www.autopano.net/blog-en/tag/video-stitching/, retrieved Feb 7, 2010.

[4] A. Zomet, A. Levin and S. Peleg, "Seamless Image Stitching by Minimizing False Edges", IEEE Transactions in image processing, , vol. 15, no4, 2006.

[5] M. Zheng, X. Chen and L. Guo, "Stitching Video from Webcams", Lecture Notes In Computer Science; Vol. 5359 archive Proceedings of the 4th International Symposium on Advances in Visual Computing, Part II, 2008.

[6] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", IJCV 04.

[7] H. Bay, T. Tuytelaars and L. Gool, "SURF: Speeded Up Robust Features", ECCV, 2006.

[8] M. El-Saban, M. Refaat, A. Kaheel and A. Abdul Hamid, " Stitching videos streamed by mobile phones in real-time", ACM-MM 09.

[9] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors", TPAMI, October 2005.

[10] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", *Proceedings of Imaging understanding workshop*, 1981

[11] A. Kaheel, M. El-Saban, M. Refaat, and M. Izz, "Mobicast - A system for collaborative event casting using mobile phones", in ACM MUM '09.

[12] M. Brown and D. Lowe, "Automatic Panoramic Image Stitching using Invariant Features", ICCV 2007.

[13] C. Doutre and P. Nasiopoulos, "Fast vigneting correction and color matching for panoramic image stitching", ICIP 2009.

[14] A. Agarwala, "Efficient gradient-domain compositing using quadtrees", ACM Transactions on Graphics, 2007.

[15] M. Bujnak and R. Sara, "A Robust Graph-Based Method for The General Correspondence Problem Demonstrated on Image Stitching", ICCV 2007.

[16] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", Communications of the ACM, 24(6), 1981.