Normalized VideoSnapping: A Non-Linear Video Synchronization Approach

Ankit Tripathi, Benu Changmai, Shrukul Habib, Nagaratna B. Chittaragi and Shashidhar G. Koolagudi
Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal, Mangalore - 575 025, Karnataka, India
Email:{13co113, 13co114, 13co143, nagaratna.cs15f09}@nitk.edu.in, koolagudi@nitk.ac.in

Abstract

Video synchronization is the task of content-based alignment of two or more videos depicting the same event with spatial variations or the same object with temporal differences. This is one of the most fundamental tasks when it comes to manipulations with temporally or spatially multi-perspective video-shots. In this paper, we have identified the state-of-the-art models which deal with the synchronization problem and put forward a model which efficiently tackles problems arising during synchronizing two videos. Here, the videos are dealt with, at the frame level with features from each frame forming the basis of alignment. Features are matched and mapped to generate a cost matrix of similarities among the frames of the videos in concern. A modified version of Dijkstra's algorithm yields an optimal path through the matrix. Through an optimal path, events are grouped into contiguous regions following which temporal warpings are introduced into the videos to achieve the best possible alignment among them. Our model has proven to be efficient and compatible to all classes of quality levels of videos.

***Index Terms*—Video Synchronization, Cost Matrix, Normalization, Dijkstra's Algorithm.**

I. Introduction

Video Synchronization refers to the problem of aligning, which may have various pace and may be recorded at different time, such that the resulting videos appear to move through the similar objects together. The problem of Video Synchronization has been studied extensively in the past. Traditionally, most of the existing solutions used spatial alignments between individual frames of two videos to achieve synchronization. However, these methods will not give accurate results if the videos are not in temporal alignment. Hence, we have proposed an approach that takes into consideration both spatial and temporal cues for synchronizing the two videos.

Video synchronization has a lot of applications such as in Movie Productions and visual effects, Video mosaicking, stitching and action recognition and so on. In movies having twin or duplicate roles, the actor is made to shoot two different sequences for the two characters separately. Then, the two videos are synchronized and then superimposed such that the resulting video consists of both the characters in the same frame.

A video is a collection of frames. Each frame is nothing but a still from the video sequence just like an image. In order to synchronize any two videos without any human intervention, it is necessary for the model in question to recognize similarities between any two frames from their corresponding frames. These similarities are identified in terms of features, in simple words they are the points of interest in a frame.

In the proposed approach, the frames of the two videos needs to be synchronized are extracted and a cost matrix which gives the cost of aligning any two frames of the two videos is generated. Both the spatial and temporal aspects are taken into consideration when generating the cost matrix. Lesser the cost, the better the alignment between two frames under consideration. Hence, we compute shortest path in the Cost matrix by using a modified version of Djikstra's algorithm, that gives the path having the lowest cost and hence the best alignment between the two videos is achieved. Next, Bezier's algorithm is used for smoothening the curve nothing but the alignment curve obtained between two videos.

The remaining part of this paper is organized as followed, section II describes the related works in the area of synchronization of videos, section III covers methodologies which discusses the base paper that has been referred [1] for implementation of basic model, normalization details which covers the novel method to improve the VideoSnapping algorithm. Results details are covered in section IV, section V discuss on conclusions and future works.

II. Related Work

The works on video synchronization began with resolving temporal offsets and introducing dynamic time warping. But these techniques do not ensure robustness. Moreover, only the temporal alignment has been taken into consideration leaving out the spatial context. This problem of spatio-temporal alignment is first taken into consideration with the technique proposed by Caspi and Irani [2]. It computes an affinity-driven temporal warping and a fixed spatial homography per video. But it does not facilitate flexibility in camera motion and perspective and is unresponsive towards depth variation and camera placement.

Videos in which there is a significant difference in the content of the scenes due to difference in perspectives or camera angles suffered from the inability to align frames

as the existing techniques could not cater to the problem of feature matching invariant of scale or angle. Later, with the advent of sophisticated feature extraction and matching techniques like SIFT, ORB, SURF etc. techniques and the extension of their application towards aligning frames resolved the issue. With many iterative improvements that followed, linear synchronization of [3] led to linear temporal mappings. Li and Chellappa allow for nonlinear time warps but require xed time spans and full video overlap [4].

The work carried out by Disney Research et. al [1] termed Video Snapping that evolved non-linear time warping among multiple videos brought about a new dimension to the solution of the existing problem. The results achieved are robust and could align multiple videos with contrasting spatial and temporal specifications. The model is suitable to be applied to professional use and is eventually used for animations and visual effects which demanded synchronization. However, the model produced good results only on high quality frames which are unaffected by motion jitter and other noises.

It was found that most of the previous research work deals with padding frames into the video for synchronization. Our contribution has focused on padding and removing frames, appropriately. At the same time, our work improvise on the performance of the model on noisy and low quality videos by normalization of cost matrix. Our contribution has mainly focused on the task of finding a path through the cost matrix which ensures greatest overall similarity of the matches along the path. This path yields the desired sequence of the frames to be played invoking necessary frame stalls so as to achieve optimal synchronization.

III. Methodology

Two videos to be synchronized should have the inherent property of partial overlap to support video editing. It is also expected that the transformation is symmetric in nature, i.e., aligning $v1$ and $v2$ or $v2$ and $v1$ results in the same output. It is essential that the degree of warping should change with respect to the context.

Videos consists of sequence of frames. Each frame is formed to combining three matrices each belonging to one of the colour channel. Hence, each pixel can be imagined as a three dimensional vector point.

Let the two videos be $v1$ and $v2$. $V_i(j)$ represents jth frame of $i-th$ video. Our aim is to generate a mapping between frames of $v1$ to frames of $v2$. Mathematically, let p be the function such that $p(t)$ represents a mapping of the frames of the video such that $t = (j, k)$ where j is a frame from $v1$ and k is a frame from $v2$. The inherent property of p is that it should be temporally continuous. In other-words, p is the flow of frames that has to be displayed.

First task is to extract features from frames of both the videos and match each of the feature from $v1$ to the most suitable match in the features of $v2$. This task is performed using Scale-invariant feature transform (SIFT) [5] algorithm which produces Scale and Rotation invariant features. These features are basically edges and corners. Each feature a has

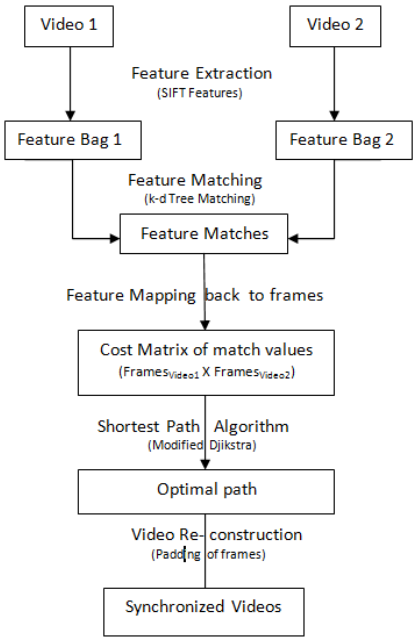

Fig. 1. Working model

a descriptor $d(a)$ which is either a collection of histograms bins that collect information about the pixels surrounding the keypoint or it is the set of pixels surrounding the keypoint. The mapping of features is done through Nearest Neighbour algorithm implemented using KD-Trees. A matrix M is generated such that each cell (i, j) represents ith feature extracted from v1 and jth feature extracted from $v2$. The value at the cell can either be 0 if j is not the nearest neighbour of i otherwise its value is 1. The information about the spatial location of each keypoint is also extracted as $x(a)$.

The mapping that we have produced is between the features but we need to synchronize videos, therefore, the mapping between the frames is needed. A matrix h is created such that each cell (j, k) represents the degree of alignment of frame j of v1 to frame k of v2. Let $(a, b) \in M$ represent if feature a of v1 is nearest neighbor of b of v2. H_{jk} is calculated using the formula

$$h_{jk} = \sum_{(a,b)\in M_{jk}} \rho_d(||d_1(a) - d_2(b)||)\rho_s(||x_1(a) - x_2(b)||) \quad (1)$$

here ρ is the decay weighing function which gives a higher weight to closer matches enhanced by its non-linear behavior.

$$w(x) = e^{-w.x} \quad (2)$$

where ρ_d and ρ_s control the rate of decay for descriptor and spatial weights.

From the h_{jk}, the cost matrix C is produced such that each cell (j, k) gives the cost of aligning frame j of v1 to frame k of v2. It is calculated as

$$c_{jk} = \left(1 - \frac{h_{jk}}{h_{max}}\right)^{\alpha} \tag{3}$$

where h_{max} is the maximum value of the matrix h_{jk}. α value can be varied depending upon the quality of the videos that are given as inputs.

We apply modified Djikstra's algorithm on the cost matrix so that frames mapping p is monotonic in nature and produces the least cost path. In other words the shortest path mapping video frames is not allowed to move backward wth respect to each of the videos. Figure 1 explains the work flow model of the approach used in this work. The performance of the existing model is best suited for the high quality high definition(HD) videos that produce many detailed keypoints in each frame and hence, the cost matrix generated has distinct difference between adjacent matches of the frames. The output generated shows sensitivity towards minute differences between frames which is incompatible with videos constituted with blurry or noisy frames. The differences among the frames in the cost matrix fade out due to which it is not possible to generate a distinct path.

A. Normalization of Cost Matrix

The h_{jk} values generated by the previous model had large differences causing many c_{jk} values to approach nearly zero value. The h_{jk} values are thus accumulated over and each of these values are normalized to a range of $(0, 100)$ against this cumulative sum. The process though changed the ratio of the values to each other but the relativity remains the same. In simple words, the smaller values remained smaller while the larger ones remained larger. The processed c_{jk} values hence generated are significant enough to provide a clear path through the matrix. So, the modified c_{jk} value is as below

$$h_{jk} = \sum_{(a,b)\in M_{jk}} \rho_d(||d_{ab}||)\rho_s(||x_{ab}||) \tag{4}$$

where d_{ab}, x_{ab} are calculated as

$$d_{ab} = \frac{||d_1(a) - d_2(b)|| \times N_f}{\sum_{(a,b)\in M} ||d_1(a) - d_2(b)||} \tag{5}$$

$$x_{ab} = \frac{||x_1(a) - x_2(b)|| \times N_f}{\sum_{(a,b)\in M} ||x_1(a) - x_2(b)||} \tag{6}$$

N_f is the Normalization factor. In our experiments, good results are found when N_f is 100. The proposed algorithm is named as NormalizedVideoSnapping algorithm.

B. Identification and mapping of events

The existing techniques use padding of frames to achieve synchronization. In this paper, frames are selectively added/eliminated to/from the video to remove the stucking frame effect. This stucking effect is caused due to the repetition of frames in the padding process. The idea is described below:

First, we compute the standard deviation of each pair of adjacent frames based on their pixel values. From the curves

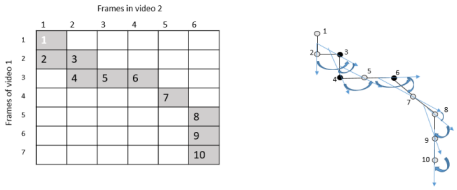

Fig. 2. Approach-I, all the points in the path of Dijkstra's algorithm are plotted and tangents are calculated and rotation is observed at each point to find the cutpoints

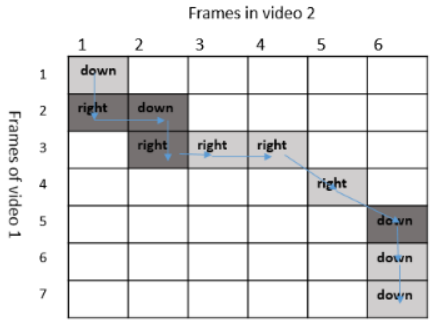

Fig. 3. Approach-II, each point is associated with the type of movement that occurs as one proceeds in the Dijkstra's shortest path.

obtained, each region between every adjacent pair of local extrema is defined as *localized event*s. Now, we map the *localized event*s from Video1 to that of Video2. The mapping can be done by two approaches.

Now that the *localized event*s are identified and mapped the *localized event*s, the following is done:

1) The Dijkstra's algorithm on the cost matrix provides the path that has least cost and this path is considered for finding the regions/events that are occurring in the two videos. A tangent is computed for each point on this path such that it passes through the mid-point of previous two points in the path and the current point. Extrema are found on the path by observing points that have different type of rotations(clockwise/anticlockwise) of tangents before and after the point. These points are sorted and corresponding regions are located in the original videos such that there is one-to-one mapping of regions in the two videos.
2) In the cost matrix, traverse along the minimum cost path computed by the modified dijkstra's algorithm. When traversing, there can be two directions: down and right. If the current direction of traversal is down followed by a transition to the right, then the point at which the change of direction occurred is added to the cutpoint list. Traversing diagonally is not considered as a change of direction. This way, all the points at which the change of direction occurs are called cut points. The region between adjacent cutpoints are the *localized event*s.

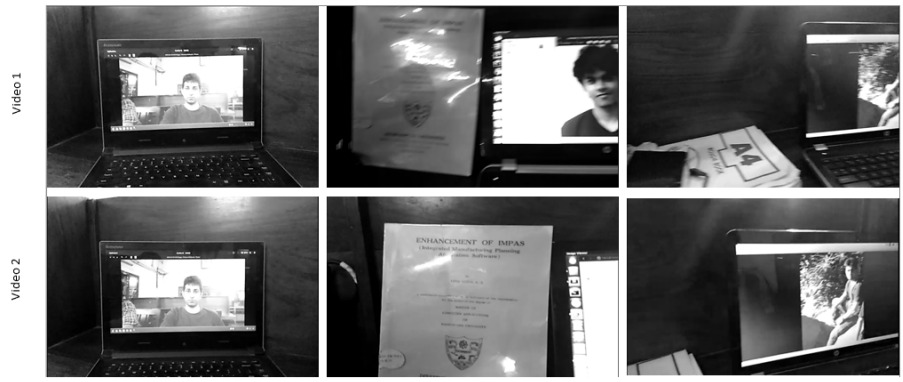

Fig. 4. Original video sequence snapshots taken at different instants

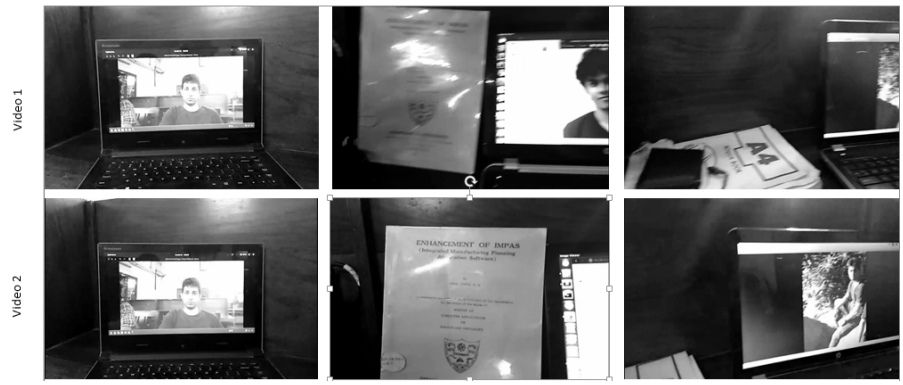

Fig. 5. VideoSnapping tool results snapshots taken at different instants

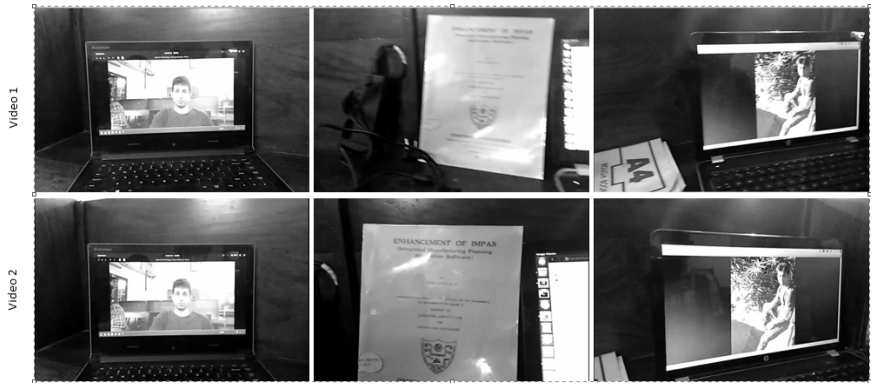

Fig. 6. Results of NormalizedVideoSnapping algorithm taken at different instants

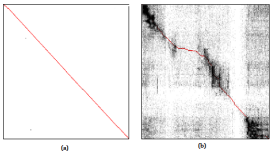

Fig. 7. Cost Matrix with shortest path in red:(a) For VideoSnapping, (b) NormalizedVideoSnapping algorithm

C. Synchronization of videos

The videos can be synchronized either by padding frames or by removing appropriately. The two methods are discussed below:

1) **Remove:** There exists a one-to-one mapping of the *localized event*s between the two videos. But the length

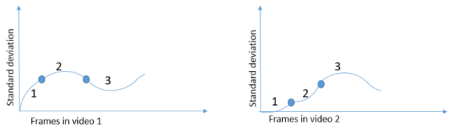

Fig. 8. There is one-to-one mapping of regions in the two videos as plotted and addition or removal of frames happen based on the corresponding regions.

of these regions may not be the same. If the length of the corresponding regions are same, we don't do anything. If the length is not the same, we selectively remove the frames from the region with more number of frames. We remove those frames from the region that have low standard deviation. The idea behind this is that, adjacent frames having lower standard deviation have a lot of similarity and hence removing one of them will not adversely affect the continuity of the video. The is done till the length of both the regions become equal. However, if the standard deviation is greater than the value alpha, then we add frames to the region with lower number of frames.

2) **Add:** Here, frames are added to the region having lower number of frames. A frame is inserted between two adjacent frames having lower standard deviation. This is because, frames having lower standard deviation are very similar and hence adding a similar frame in the middle will not change the detail depicted by that region. The frame added will be the mean of the two frames computed pixel-wise. This is done till the length of both the regions become equal.

IV. Results

The preprocessing stage consisted of extracting features from each frame. SIFT algorithm is used for extraction from openCV library and the matching is done by N-Nearest Neighbor algorithm implemented in KD-Tree in Python programming language. The time taken to run the algorithm is substantially concentrated in extracting features and feature matching. Another considerable chunk of time is spent in finding the shortest path in the cost matrix.

Figure 5 shows the corresponding frames of the original videos arranged one above the other. Both the videos are synchronized at the beginning but Video1 moves along faster than Video 2. Hence we see that synchronization is lost in frame 2 and frame 3 of the two videos.

Figure 6 shows the corresponding frame of the Videos generated after applying the Video Snapping algorithm. Again, both the videos seem to be synchronized in the beginning. But synchronization is lost again in frame 2 and frame 3. Looking at the Cost matrix, we see that its almost a linear line. Hence, the videos generated are highly similar to the original videos.

Figure 7 shows the frames of the videos generated after applying the Normalized Video Snapping algorithm. We can clearly see that all the three frames appear to be synchronized. The same criteria of tests are carried out on several other videos of varying qualities and noise level and the results are promising.

V. Conclusion and Future Work

For High Definition (HD) Videos, Normalized VideoSnapping algorithm is found to be on par with the existing solutions. For Low Quality or Noisy Videos, such as the ones shot by Smartphone cameras, we found that Normalized VideoSnapping algorithm gives considerably better synchronization. An algorithm removes or pads frames based on the event the frames represent. We would like to explore the area of dynamic frame rate to further tune the videos.

References

[1] Oliver Wang, Christopher Schroers, Henning Zimmer, Markus Gross, and Alexander Sorkine-Hornung. Videosnapping: Interactive synchronization of multiple videos. *ACM Transactions on Graphics (TOG)*, 33(4):77, 2014.

[2] Yaron Caspi and Michal Irani. Spatio-temporal alignment of sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(11):1409–1424, 2002.

[3] Jan Ruegg, Oliver Wang, Aljosa Smolic, and Markus Gross. Spatio-temporal video compositing, September 9 2013. US Patent App. 14/022,048.

[4] Ruonan Li and Rama Chellappa. Aligning spatio-temporal signals on a special manifold. In *Computer Vision–ECCV 2010*, pages 547–560. Springer, 2010.

[5] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[6] Nicholas J Bryan, Paris Smaragdis, and Gautham J Mysore. Clustering and synchronizing multi-camera video via landmark cross-correlation. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 2389–2392. IEEE, 2012.

[7] Ferran Diego, Daniel Ponsa, Joan Serrat, and Antonio M López. Video alignment for change detection. *Image Processing, IEEE Transactions on*, 20(7):1858–1869, 2011.

[8] Prarthana Shrstha, Mauro Barbieri, and Hans Weda. Synchronization of multi-camera video recordings based on audio. In *Proceedings of the 15th international conference on Multimedia*, pages 545–548. ACM, 2007.

[9] Georgios D Evangelidis and Christian Bauckhage. Efficient subframe video alignment using short descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(10):2371–2386, 2013.

[10] Ferran Diego, Joan Serrat, and Antonio M López. Joint spatio-temporal alignment of sequences. *Multimedia, IEEE Transactions on*, 15(6):1377–1387, 2013.

[11] Simon Baker and Iain Matthews. Equivalence and efficiency of image alignment algorithms. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–1090. IEEE, 2001.

[12] Yaron Caspi and Michal Irani. A step towards sequence-to-sequence alignment. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 682–689. IEEE, 2000.

[13] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 494–501. ACM, 2007.

[14] Carlo Tomasi and Takeo Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[15] Yaron Ukrainitz and Michal Irani. *Aligning sequences and actions by maximizing space-time correlations*. Springer, 2006.

[16] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[17] Eli Shechtman and Michal Irani. Space-time behavior based correlation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 405–412. IEEE, 2005.

[18] Chong-Wah Ngo, Yu-Fei Ma, and Hong-Jiang Zhang. Video summarization and scene detection by graph modeling. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(2):296–305, 2005.

[19] Cen Rao, Alexei Gritai, Mubarak Shah, and Tanveer Syeda-Mahmood. View-invariant alignment and matching of video sequences. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 939–945. IEEE, 2003.

[20] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):978–994, 2011.

[21] G Kang. Digital image processing. *Quest, vol. 1, Autumn 1977, p. 2-20.*, 1:2–20, 1977.

[22] Yaron Caspi and Michal Irani. Aligning non-overlapping sequences. *International Journal of Computer Vision*, 48(1):39–51, 2002.

[23] Yaron Caspi, Denis Simakov, and Michal Irani. Feature-based sequence-to-sequence matching. *International Journal of Computer Vision*, 68(1):53–64, 2006.