SUBFRAME VIDEO SYNCHRONIZATION BY MATCHING TRAJECTORIES

Yuanyuan Wu[1]*, Xiaohai He*[1]*, and Truong Q. Nguyen*[*] [2]

[1]College of Electronics and Information Engineering, Sichuan University, Chengdu China
[2]ECE Department, University of California San Diego, La Jolla CA 92093

ABSTRACT

We propose a novel approach to align unsynchronized video sequences of the same dynamic scene that can be subframe accurate and is applicable for different frame rate problem. The proposed approach relies on matching motion trajectories and it is assumed that the object moves on a planar surface. By exploring the invariants of planar trajectories under projective transformation, the cross ratio as an invariant feature is computed for each point along the trajectories and the similarity between invariant features of different trajectories is measured with a distance that takes into account the statistical properties of the cross ratio. Then the smooth high frame rate trajectory is synthesized for searching subframe temporal displacement under our alignment framework. The experimental results with synthetic and real-world sequences show that our approach achieves fairly accuracy and efficiency in subframe temporal alignment of the multiple unsynchronized video sequences.

Index Terms— Video synchronization, Subframe temporal alignment, Projective invariants, Matching trajectories.

1. INTRODUCTION

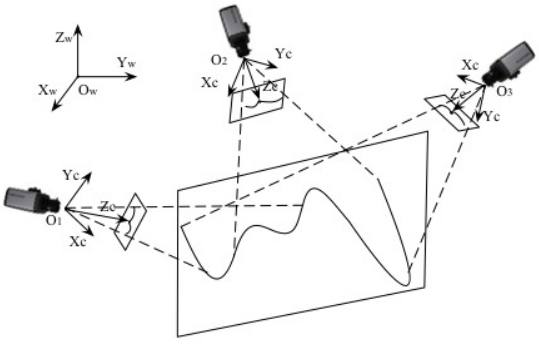


Fig. 1: A multi-camera system setup. An object moving on a planar surface is recorded by several cameras at different viewpoint. There is an overlapping among the three videos.

This work was supported by NSAF (Grant No. 61071161) , China Scholarship Council and it was carried out while Yuanyuan Wu was visiting the Video Processing Lab at University of California San Diego.

Along with the rapid development of software and hardware technology of computer vision, many videos needed to be analyzed and processed, such as those from visual surveillance system, industry control equipment, medical testing instruments, traffic management center, military and science research field. Furthermore, the number of video sequences capturing the same scene has also increased from 1 to N (N is typically larger than 2), as illustrated in Fig.1. Typically, when a group of cameras are placed at different viewpoints, the captured videos may look very different from each other. Moreover, when cameras start asynchronously, even with different frame rate, there would be an unknown time offset between them. How to determine the temporal relationship between multiple video cameras effectively and accurately has attracted considerable attention of researchers. In recent years, many algorithms have been proposed to resolve this problem for specific domain, such as video super-resolution reconstruction, sport motion analysis, camera calibration, 3D visualization and gait recognition.

Several works have been proposed such as the work by Caspi and Irani [1] which estimates the temporal and spatial shift parameters jointly in an iterative framework; and the work by Padua and Kutulakos [2], which uses a linear video to align multiple video sequences. A nonlinear dynamic time warping function is computed to find temporal shift between video sequences [3]. In [4] audio sequences as addtional information is used to synchronize videos, by detecting and matching flashes present generated by still cameras. Most previous approaches can only estimate the integer frame accurate offset, while a few methods achieve subframe accurate offset. However, very few previous works consider the problem of temporal alignment among videos with different frame rates.

With respect to the above approaches, our method is more suitable for the task of arbitrary time shift and frame rate differences, in particular it's highly robust with respect to noise that often occur along trajectories. We capitalize a new geometric invariant feature (cross ratio) to synchronize videos, which was not considered in earlier studies (models such as 2D homography, fundamental matrices, 3D rotations or affine projections [5] have been used). While the present study is related to recent approaches in invariant feature-based trajectory recognition [6] and dynamic depth recovery from unsyn-

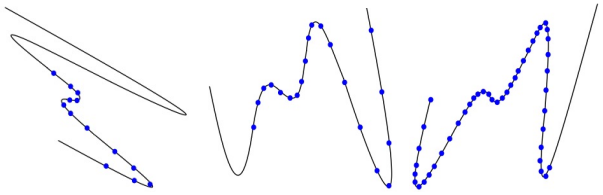
chronized video streams [7]. The main difference between the proposed method and those in [6, 7] is how the cross ratio is constructed. Moreover, the proposed framework creates a smooth trajectory with high frame rate. Consequently, the proposed method does not need manual alignment of the coarse temporal offset (as done in [7]).

The rest of the paper is organized as follows. Section 2 formulates the problem of subframe temporal alignment. Section 3 describes the projective invariants of planar trajectories and the method of comparing the invariants. Section 4 presents our alignment algorithm framework for subframe temporal alignment. Section 5 discusses the simulation and real-world experiment results. Finally, the conclusion and future work is presented in Section 6.

2. PROBLEM FORMULATION

In the multi-camera system, the classic pinhole camera model is adopted to describe the image acquisition process, which defines the geometric relationship of mapping a 3D point onto a 2D image plane. Typically, any two images of the same planar surface in space are related by a projective transformation in projective geometry. Fig.2 shows an example of the motion trajectories obtained from 3 videos. Although they record the same dynamic scene, they seem to vary widely due to their different start time, viewpoint, and frame rate.

NTSC (30fps (frames per second)) and PAL (25fps) are the two common formats for video recording with various kinds of video frame rates: 12/24/48/50/60/100/120/240fps. When the reference video and the second video are recorded in different frame rates, this issue should be taken into account in sub-frame video synchronizing algorithm. The other issue is how to automatically find the temporal shift between two videos without apriori knowledge, i.e. knowing the order of which video is recorded first as well as which video has ended first, or whether the two videos are recorded at almost the same time.



(a) Trajectory 1 (b) Trajectory 2 (c) Trajectory 3

Fig. 2: Multi-view images of a planar scene: (a) Trajectory obtained in video1, with a low frame rate. (b) Trajectory obtained in video2, with a low frame rate. (c)Trajectory obtained in video3, with a high frame rate.

For the case with two cameras, let P_1:$\{p_1(t)\}, t = [1...N]$ be the trajectory point sequence obtained from the reference video S, which is recorded at $f_1$fps, where $p_1(t) = (x(t), y(t))$ denotes the image coordinates of the trajectory point, $t = [1...N]$ denotes the frame index number. Similarly, P_2:$\{p_2(t')\}$, $t' = [1...M]$ is defined as the trajectory point sequence obtained from the second video S', which is recorded at $f_2$fps. The temporal displacement between S and S' can be expressed as follows:

$$t' = R \cdot t + \Delta t \tag{1}$$

where, $R = \frac{f_2}{f_1}$, and Δt is the subframe shift.

3. PROJECTIVE INVARIANTS OF TRAJECTORIES

3.1. Computing Cross Ratio

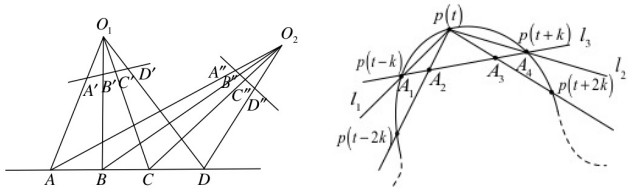
Cross ratio is the most important projective invariant in the sense that it is preserved by the projective transformations of a projective line. In particular, given four distinct collinear points A, B, C and D in R^2 (shown in Fig.3(a)), the Euclidean distance between two points A and C is denoted as $\overline{AC}$. Then, one definition of the cross ratio is:

$$(A, B; C, D) = \frac{\overline{AC} \cdot \overline{BD}}{\overline{AD} \cdot \overline{BC}} \tag{2}$$

According to planar projective transformation theory, we have the following conclusion:

$$\begin{cases} (A, B; C, D) = (A', B'; C', D') \\ (A, B; C, D) = (A'', B''; C'', D'') \end{cases}$$

$$\Rightarrow (A', B'; C', D') = (A'', B''; C'', D'') \tag{3}$$



(a) (b)

Fig. 3: Cross ratio of collinear points. (a) depicts the invariance property cross ratio under projective trasformation. (b) the proposed method to compute cross ratio along trajectory.

As mentioned before, since cross ratio is invariant under projective transformation, we create unique cross ratio for each point along the trajectory, as shown in Fig.3 (b). The cross ratio $\tau(t)$ is calculated for point $p(t)$ by using its neighbouring points $p(t-2k)$,$p(t-k)$,$p(t)$,$p(t+k)$,$p(t+2k)$ where k is an integer value chosen according to the trajectory. The detail of calculating process is expressed as follows:

$$\tau(t) = g(p(t)) = \frac{\overline{A_1A_3} \cdot \overline{A_4A_2}}{\overline{A_1A_2} \cdot \overline{A_4A_3}} \tag{4}$$

- $A_1 \leftarrow p(t-k), A_4 \leftarrow p(t+k)$
- $l_1 = p(t) \times p(t-2k)$, line through $p(t)$ and $p(t-2k)$,
- $l_2 = p(t) \times p(t+2k)$, line through $p(t)$and $p(t+2k)$,
- $l_3 = p(t-k) \times p(t+k)$, line through $p(t-k)$ and $p(t+k)$,
- $A_2 = l_1 \times l_3$, intersection between l_1 and l_3,
- $A_3 = l_2 \times l_3$, intersection between l_2 and l_3

3.2. Measure the Similarity

It is shown in [8] that a probability density function for the cross ratio can be computed in closed form, together with the corresponding cumulative density function. A distance measure derived from this function has been proposed in [9] in the context of object recognition. The distance is computed with respect to the cumulative distribution function:

$$d(\tau_1, \tau_2) = \min\left(|F(\tau_1) - F(\tau_2)|, 1 - |F(\tau_1) - F(\tau_2)|\right) \quad (5)$$

where $F(x)$ is defined as follows:

$$F_X(x) = P(X < x) \begin{cases} F_1(x) + F_3(x) & if \ \ x < 0 \\ 1/3 & if \ \ x = 0 \\ 1/2 + F_2(x) + F_3(x) & if \ \ 0 < x < 1 \\ 2/3 & if \ \ x = 1 \\ 1 + F_1(x) + F_2(x) & if \ \ 1 < x \end{cases} \quad (6)$$

where

$$F_1(x) = \frac{1}{3}\left(x \cdot (1-x) \cdot \ln(\frac{x-1}{x}) - x + \frac{1}{2}\right)$$

$$F_2(x) = \frac{1}{3}\left(\frac{x - x \cdot \ln(x) - 1}{(x-1)^2}\right)$$

$$F_3(x) = \frac{1}{3}\left(\frac{(1-x) \cdot \ln(1-x) + x}{x^2}\right)$$

In the proposed method of calculating the cross ratio, there are a few special situations, such as $p(t-2k)$, $p(t-k)$, $p(t)$ are collinear, so the point A_1 and A_2 would be the same point. Meanwhile the distance value of $\overline{A_1A_2}$ is zero. In that situation, $\overline{A_1A_2}$ will be replaced by an infinitely small quantity, then according to the formula, $\tau(t)$ is considered to be infinite. In our experiment, if $\overline{A_1A_2}$ or $\overline{A_4A_3}$ equals to zero, we use a large value $\left(1 \times 10^6\right)$ instead of infinite value as the calculating result of $\tau(t)$.

The selection of the parameter k is closely related to the length and the variability of trajectory. If a trajectory changes quickly, a smaller value for k is an appropriate choice, whereas a slowly varying trajectory would utilize a larger value for k. As mentioned before, a trajectory segment with length of $4k$ is used to compute the cross ratio for each trajectory point, so this length should reflect the unique representation between different parts of the same trajectory.

4. SUBFRAME TEMPORAL ALIGNMENT

4.1. Synthesis High Frame Rate Trajectory

Objects in nature change or move in a continuous way. When people use a camera to record the dynamic scene, it is actually a sampling of a continuous signal, while the sampling rate is the camera frame rate. It's tempting to think that if we can get more sufficient sample data points, we can form a more realistic trajectory to help us find the relationship between the videos. In many engineering practice and scientific experiments, one usually use either data fitting or data interpolation to obtain more sample points. We adopt the cubic spline interpolation method to address this problem. A series of unique cubic polynomials are fitted between the adjacent two trajectory points and with the stipulation that the curve obtained be continuous and appeared smooth. This data interpolation process should be done in X and Y image coordinates separately, then a smooth high frame rate trajectory will be produced. In our paper, the high frame rate trajectory P_2':$\{p_2'(t')\}$ for the second trajectory P_2:$\{p_2(t')\}$ is generated through this method, which contains more trajectory points.

4.2. Temporal Alignment Framework

Let f_1, f_2 and f_3 denote the frame rates of trajectory P_1, P_2 and P_2' respectively. f_3 is chosen using (7), where $LCM(f_1, f_2)$ is the least common multiple of f_1 and f_2, a is a positive integer value and selected for practical need. In particular, the higher the value of the f_3 is, the higher the calculation accuracy.

$$f_3 = LCM(f_1, f_2) \cdot a, a \in N^* \quad (7)$$

the formula can be expressed as follows:

$$\frac{1}{f_1} = b \cdot \frac{1}{f_3}, \frac{1}{f_2} = c \cdot \frac{1}{f_3}, b, c \in N^* \quad (8)$$

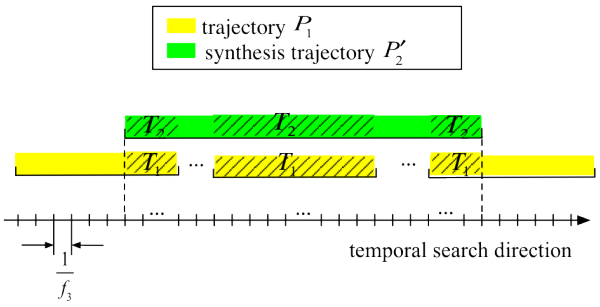


Fig. 4: Structure of temporal alignment method.

Obviously, there is an overlapping in time between two sequences to provide enough information for the aligning algorithm. Let T_1 be the overlapping region in trajectory P_1, and T_2 be the corresponding overlapping region in trajectory P_2'. Moreover, the overlapping time should not be too

Table 1: Multi-video synchronization results.

Trajectory	Ground truth (R,Δt)	Average frame error \Maximum frame error (10% noise)	Average frame error \Maximum frame error (20% noise)	Average frame error \Maximum frame error (30% noise)	Average frame error \Maximum frame error (40% noise)
Tra1	(0.5, 5.63)	0.0478\ 0.1300	0.0983\ 0.3000	0.1881\ 0.5700	0.2015\ 0.6900
Tra2	(5/6, 7.12)	0.0375\ 0.1167	0.0807\ 0.2750	0.1338\ 0.4750	0.1677\ 0.5000
Tra3	(1.0, -4.35)	0.0512\ 0.2000	0.0978\ 0.3200	0.1504\ 0.8700	0.1547\ 0.6600
Tra4	(1.2, 2.56)	0.0413\ 0.1680	0.0695\ 0.1920	0.1118\ 0.3840	0.1288\ 0.5040
Tra5	(1.6, -3.89)	0.0666\ 0.2020	0.1256\ 0.3300	0.1622\ 0.4740	0.2072\ 0.6140
Tra6	(2.0, 4.27)	0.0460\ 0.1700	0.1294\ 0.4100	0.1506\ 0.5500	0.2042\ 0.6900

short, because smaller overlapping time tends to create a similar part of trajectory that causes inaccurate alignment result. The aligning process is shown in Fig.4. Since there are several possibilities of (T_1, T_2), we perform an exhaustive search over time shifts to calculate the average distance between T_1 and T_2, which is denoted as follows:

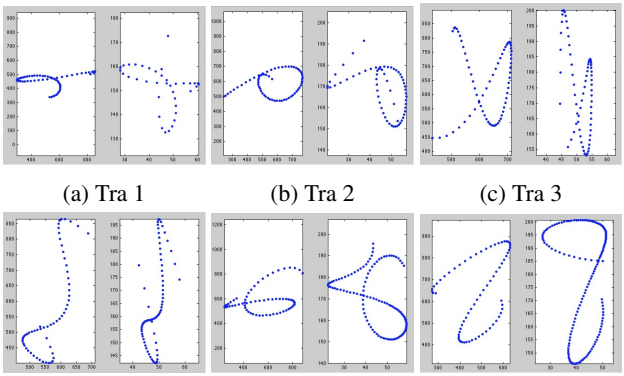
$$d(T_1, T_2) = \frac{1}{n}\sum_{i=1}^{n} d(\tau_1(t), \tau_2(t)), T_1 \subseteq P_1, T_2 \subseteq P_2' \quad (9)$$

Here, n is the number of points contained in trajectory T_1. The value (T_1, T_2) is closer to the real overlapping regions, the smaller $d(T_1, T_2)$ will be.

5. EXPERIMENT RESULTS

In the first experiment, 6 pairs of multi-view planar trajectories (shown in Fig.5) with provided ground-truth parameters $(R, \Delta t)$ were generated and corrupted with an along-trajectory noise to simulate the tracking error. The parameter k was selected as 6, 10, 6, 6, 8, 7 for Tra1 to Tra6. The experiment was repeated 100 times respectively with different level of noise variance, from 10% to 40% of the average distance between points. The estimated $\Delta t'$ was compared with the ground-truth Δt, the average frame error $\frac{1}{100}\sum_{i=1}^{100}|\Delta t_i' - \Delta t|$ and the maximum frame error $max(|\Delta t_i' - \Delta t|), i = 1...100$ were recorded in Table 1. From Table 1, we can see that our method is highly robust to noise.

In the second experiment, a dynamic scene of moving a baseball on the wall was recorded by 3 cameras from multi-view points with 80fps, 60fps, 30fps respectively, shown in Fig.6. We tracked the centroid of the baseball to obtain trajectories by block track method, and the three trajectories consisted of 320, 260 and 150 points individually. Fig.6(c) was assumed to be the reference video and we set k=9. We improved the classical temporal alignment method [1] to suit for different frame rates problem and the calculated temporal displacement were Δt=-74.5133 for Fig.6(a) and Δt=-35.07 for Fig.6(b). By contrast, our results were Δt=-74.5367 and Δt=-35.09. Both alignment results were very similar, which proves that our method is effective and practical.



(a) Tra 1 (b) Tra 2 (c) Tra 3

(d) Tra 4 (e) Tra 5 (f) Tra 6

Fig. 5: 6 pairs of multi-view simulation trajectories with different frame rate and subframe temporal shift.

(a) The left view (b) The middle view (c) The right view

Fig. 6: An example of multi-video.

6. CONCLUSION AND FUTURE WORK

In this paper, we present a method to synchronize multi-view videos with different frame rates that can achieve high subframe accuracy. The proposed method uses theory of collinearity of points and the invariance of cross ratio under projective transformation to compare trajectories. The overlapping region of two trajectories is then analyzed carefully and a smooth high frame rate trajectory is synthesized for searching the subframe temporal offset. The results of experiment verify that our method is effective and robust. Future research plan consists of investigating how to efficiently determine the parameter k, as well as extending the proposed algorithm from 2D planar scenes to general 3D scenes.

7. REFERENCES

[1] Yaron Caspi, Denis Simakov, and Michal Irani, "Feature-based sequence-to-sequence matching," *Int. J. Comput. Vision*, vol. 68, no. 1, pp. 53–64, June 2006.

[2] Flavio Padua, Rodrigo Carceroni, Geraldo Santos, and Kiriakos Kutulakos, "Linear sequence-to-sequence alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 304–320, Feb. 2010.

[3] Cheng Lu, Meghna Singh, Irene Cheng, Anup Basu, and Mrinal Mandal, "Efficient video sequences alignment using unbiased bidirectional dynamic time warping," *J. Vis. Comun. Image Represent.*, vol. 22, no. 7, pp. 606–614, Oct. 2011.

[4] Prarthana Shrestha, Hans Weda, Mauro Barbieri, and Dragan Sekulovski, "Synchronization of multiple video recordings based on still camera flashes," in *Proceedings of the 14th annual ACM international conference on Multimedia*, New York, NY, USA, 2006, MULTIMEDIA '06, pp. 137–140, ACM.

[5] Congxia Dai, Yunfei Zheng, and Xin Li, "Subframe video synchronization via 3d phase correlation.," in *ICIP*. 2006, pp. 501–504, IEEE.

[6] Walter Nunziati, Stan Sclaroff, and Alberto Del Bimbo, "An invariant representation for matching trajectories across uncalibrated video streams," in *Proceedings of the 4th international conference on Image and Video Retrieval*, Berlin, Heidelberg, 2005, CIVR'05, pp. 318–327, Springer-Verlag.

[7] Chunxiao Zhou and Hai Tao, "Dynamic depth recovery from unsynchronized video streams," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, June, vol. 2, pp. II–351–8 vol.2.

[8] Kale Aastrom and Luce Morin, "Random Cross Ratios," Rapport Technique IMAG-RT - 92-088 ; LIFIA - 92-014, 1992.

[9] Patrick Gros, "How to use the cross ratio to compute projective invariants from two images," in *Proceedings of the Second Joint European - US Workshop on Applications of Invariance in Computer Vision*, London, UK, UK, 1994, pp. 107–126, Springer-Verlag.