Grouping Crowd-Sourced Mobile Videos for Cross-Camera Tracking

Nathan Frey
Systems & Technology Research
Woburn, MA, USA
nathan.frey@STResearch.com

Matthew Antone
Sciopic Technologies
Winchester, MA, USA
antone@sciopic.com

Abstract

Public adoption of camera-equipped mobile phones has given the average observer of an event the ability to capture their perspective and upload the video for online viewing (e.g. YouTube). When traditional wide-area surveillance systems fail to capture an area or time of interest, crowd-sourced videos can provide the information needed for event reconstruction. This paper presents the first end-to-end method for automatic cross-camera tracking from crowd-sourced mobile video data. Our processing (1) sorts videos into overlapping space-time groups, (2) finds the inter-camera relationships from objects within each view, and (3) provides an end user with multiple stabilized views of tracked objects. We demonstrate the system's effectiveness on a real dataset collected from YouTube.

1. Introduction

Recent advancements in wide-area surveillance devices (e.g., surveillance camera networks, aerial sensor arrays) have provided the video data sources required to track single or multiple objects across large spatial and temporal extents. Many techniques for object tracking [1,2] and inferring inter-camera relationships [3] from stationary sensor networks have been developed to exploit these systems; however, utilization of these techniques to survey a particular object or event requires that the camera system be deployed in the area and at the time of interest, which in turn requires a-priori knowledge of the event.

Wide adoption of camera-equipped mobile phones has given average observers the ability to record an event from their perspective and upload the recording for public consumption (e.g., YouTube). Observers of highly unique, exciting, or unusual events now can—and frequently do—capture their own perspectives using mobile cameras. When such events occur, the coverage in approximate space-time vicinity of the event becomes large enough to group overlapping videos, identify correspondences between views, calibrate inter-camera relationships, and temporally align the videos. With these relationships known, cross-camera object tracking can be performed over larger space-time volumes than those captured by a single camera.

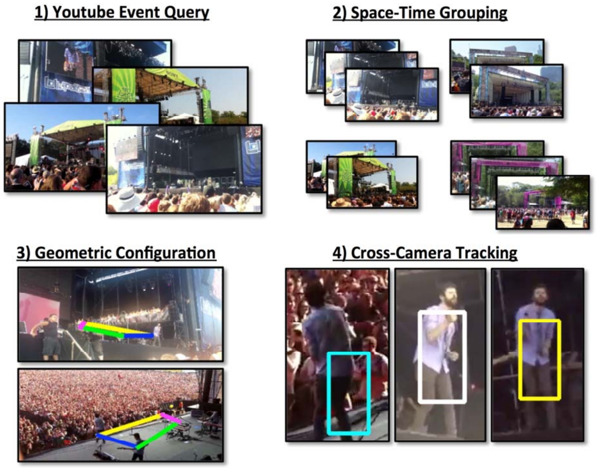

Figure 1. Processing overview, progressing from a broad video search (1) to sorted videos (2), then establishing inter-camera relationships (3) to stabilized cross-camera cueing (4).

This paper introduces the first end-to-end system to allow monitoring and tracking within a wide area utilizing only publicly available, user generated video (Figure 1). We demonstrate the effectiveness of the system using a real dataset collected from YouTube.

1.1. Challenges

Crowd-sourced videos of public events exhibit a number of characteristics that reduce the effectiveness of traditional methods for discovering and exploiting inter-camera relationships, such as [4]. In particular, geometric event coverage can be sparse with very wide viewpoint diversity and frequent occlusion; videos can contain substantial capture artifacts (motion blur, jitter, compression); and extracted visual keypoints tend to favor dominant portions of the scene (background) which often do not overlap across camera fields of view (Figure 2).

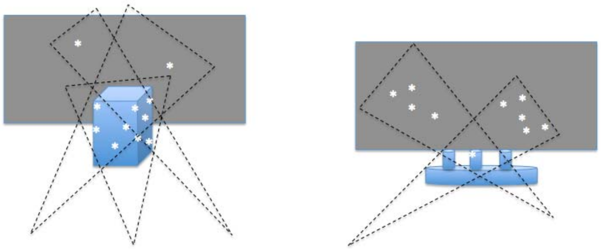

Figure 2: Traditional camera alignment methods (left) assume a large overlap region and many commonly-observed keypoints (white asterisks) on an object of interest (blue box). Crowd-sourced event camera views (right) typically have much smaller foreground overlap regions (blue) with most keypoints detected on non-overlapping background regions (gray).

Despite these challenges, crowd-sourced public event videos do provide additional information (many frames per camera, synchronized audio streams), additional constraints (minimum translational movement) and strong priors (textual video descriptions, inter-frame motion) that can drastically reduce the search space.

1.2. Related Work

While a number of methods have been developed for cross-camera tracking and automatic calibration of multi-sensor networks [5,6], very few techniques are applicable to opportunistic, crowd-sourced data sets gathered from a heterogeneous collection of sensors, from different viewpoints, and possibly containing unrelated content, such as would be returned by a YouTube query.

Image and video categorization techniques (e.g., [7]) are able to group images into broad, pre-learned categories, but lack the spatial and temporal structure and specificity to associate data from particular events. Sivic et al [8] demonstrated a system for detecting visually similar objects from video in a query-retrieval system; more recent work in large-scale structure-from-motion (e.g., [4]) and visual SLAM (e.g., [9]) applies similar techniques—namely SIFT coupled with high-performance descriptor indexing and weak geometric constraints—to achieve excellent performance in recognizing and clustering visually similar locales. All such techniques rely on substantial overlap of visual content.

Camera network calibration has been explored in a variety of contexts using a variety of methods, including explicit calibration targets visible to multiple sensors [10], local motion feature correlation [11], long-term observation of trajectory shapes and track co-occurrences [3], and centralized or distributed feature-based bundle adjustment [4,12]. Although in some cases cameras are allowed to pan, tilt, and zoom, all such techniques are designed to operate within a stationary sensor network installation; they benefit from fixed camera positions, large spatial overlap, known temporal synchronization, and long-term observation. Calibration of moving camera networks has also been explored in the context of motion capture and visual effects; some techniques rely on explicit calibration targets placed in the scene [13,14], precluding crowd-sourced scenarios, while more recent systems are markerless [15] but require substantial single-camera motion and visual overlap across cameras.

Cross-camera tracking systems have also been widely explored, again mainly for stationary networks [16]. Shah [17,18] and others have even extended multi-camera to the case of no spatial overlap, but still rely on known time synchronization and work best with isolated movers in uncluttered scenes.

1.3. Contributions

In this paper we introduce the first end-to-end method for ingesting a large unsorted collection of videos (e.g., from a search result) to produce tracks of objects from multiple views. We accomplish this by (1) sorting videos into groups that were captured in the same approximate region and time, (2) finding spatial and temporal inter-camera relationships using audio streams, visual scene appearance, and common object detections across views, and (3) creating and fusing a stabilized track window from multiple viewpoints.

2. System Overview

Our approach to cross-camera tracking can be formulated as the optimization of a single cost function that attempts to account for all features across all views. To solve for the ideal alignment, we decouple the problem into three distinct and independent steps: camera grouping, spatial-temporal alignment, and establishing inter-camera relationships.

Textual web-based video queries for particular events invariably result in a mixture of both relevant (space-time overlapping) clips and irrelevant (non-correlated outlier) clips. For more spatially and/or temporally distributed events, videos may form a number of locally overlapping but mutually distinct sub-groups, as well as outliers. The first step in our pipeline determines these general clip groupings in order to perform alignment and fusion only across relevant camera sets.

Once video clusters have been established, inter-camera relationships within each cluster must be estimated. Figure 2 illustrates a common viewing scenario in which visual background scene similarity cannot be used to determine a geometric relationship between cameras, despite those cameras observing common foreground content. Therefore, we also utilize ensemble tracking data from object detectors (e.g., pedestrians), which provides sets of salient features within and across views. Track correlation, inter-frame homographies, and temporal alignment provide the means to determine inter-camera relationships across the time-period and field-of-view of overlap.

Lastly, with inter-camera spatial-temporal alignment in place, single-object tracks can be correlated across views. For instance, a user may be interested in maintaining view of a particular object over a long period of time, but within any single camera this object may leave the field of view, become occluded, or have low resolution coverage at any point during the event. Using the inter-camera relationships at each frame, we show that the subject can be identified and segmented from multiple views given the current state of the camera network.

2.1. Camera Network Assumptions

We make several assumptions about crowd-sourced event coverage, based on empirical observations, that reduce problem complexity. First, we assume that moving objects (e.g., performers, presenters, and other subjects of attention) are visible from multiple camera viewpoints. Presumably, the very existence of these videos is predicated upon this assumption: multiple observers wish to capture the event of interest. Correlated cross-camera motion tracks form the basis of our geometric alignment technique and allow post-coverage of subjects from multiple angles.

Second, we assume that a relatively distinctive audible signal is present in overlapping videos, which is typically the case for concerts, speeches, plays and other staged events. This audio assists in grouping clips and determining their temporal alignment. We do not, however, require that this common source always dominate the audio stream—our methods are robust to typical disturbances such as sound dropouts, sensor-local speech, crowd noise, and muffled microphones.

Third, we assume that each camera source is restricted (roughly) to pan-tilt-zoom motions. In many mobile videos the camera operator remains largely stationary, simply rotating and zooming the camera to follow objects of interest and capture the event. This allows the intra-camera image motion to be well-described by projective homographies.

Finally, we assume that all tracked individuals move on an approximately planar surface, which allows inter-camera homographies to be estimated between views that account for most moving objects in the scene. This assumption is minimally restrictive, since most event action occurs over locally flat surfaces (e.g., man-made environments).

2.2. Alignment Problem formulation

Given camera pair $\{\mathrm{C_i}, \mathrm{C_j}\}$ we wish to estimate the pair's global temporal offset t_{ij}^{off} and spatial alignment homography $\mathrm{H_{ij}}$. Knowing that the camera's viewpoint relationships will have areas of *mutual support* (ms) for *audio features* (s^a) and *object tracks* (s^o) we can formulate an alignment cost function:

$$H_{ij} = \underset{H_{ij}, t_{ij}^{\mathrm{off}}}{argmin} \left[\sum_{t \in ms(i,j)} \left(d^a\left(s_i^a, s_j^a, t_{ij}^{\mathrm{off}}, t\right) \right) + \sum_{t \in ms(i,j)} \left(d^o\left(s_i^o, s_j^o, t_{ij}^{\mathrm{off}}, H_{i,j}, t\right) \right) \right] \quad (1)$$

where the audio feature alignment score is

$$d^a\left(s_i^a, s_j^a, t_{ij}^{\mathrm{off}}, t\right) = \left\| s_j^a(t) - s_i^a\left(t + t_{ij}^{\mathrm{off}}\right) \right\| \quad (2)$$

and the track object alignment score is

$$d^o\left(s_i^o, s_j^o, t_{ij}^{\mathrm{off}}, H_{i,j}, t\right) = \left\| s_j^o(t) - H_{i,j} s_i^o\left(t + t_{ij}^{\mathrm{off}}\right) \right\| \quad (3)$$

To solve equation (1), we decouple the alignment problem into isolated steps. The first step is temporal alignment and spatial grouping of the cameras to find areas of mutual support. After temporal alignment and grouping, pedestrians are detected within each viewpoint, tracked, and matched across cameras. The corresponding track states (positions and velocities) provide the final constraints needed to determine H_{ij} with a relative projective offset at each time t.

Once H_{ij} are known for all camera pairs, any particular subject tracked in one camera can be simultaneously observed within all overlapping views via reprojection, providing users with multiple perspectives of the subject.

3. Camera Grouping and Temporal Alignment

When conducting a text based query for an event on a public video website (e.g. YouTube), a wide variety of results are typically returned. Results may fall outside of the user's specified space-time region (outliers) or may make up clusters of sub-events within the queried region. Because the results of a web query do not typically all have overlapping space-time FOVs, we first group cameras that potentially overlap our reference camera using visual and audible features. We also use audio features to determine temporal alignment between views.

3.1. Audio Feature Matching

Each video clip is accompanied by an audio track that can be decoded to a raw PCM sample stream. The transport wrapper encodes timestamps and rates, so that both the audio sampling frequency and timing relative to the video stream are accurately known. Temporal alignment among multiple clips can thus be reduced to the estimation of a single parameter—namely, the time offset t^{off}—for each clip. Furthermore, knowledge of the recorded video rate and timestamps allow this time offset to be easily related to a particular video frame (i.e., with more useful units of frames rather than seconds).

To estimate t^{off} between two particular clips, we first note that videos observing the same event (e.g., speech, performance, parade) are likely to share audio content. Therefore, if temporally localized audio features can be extracted and associated across clips, each such association forms a single constraint on the time offset. Ambient noise, varying sound intensity levels, and differing "foreground" audio content across clips all preclude application of cross-correlation or similarly simplistic stream-to-stream alignment methods such as those used in [10]; we instead operate in the time-frequency domain, computing features derived from the short-time Fourier transform.

An audio feature consists of a real-valued magnitude-only spectrogram $S(\omega,t)$ centered at a particular instant and computed over a fixed set of overlapping time windows. Here, ω and t are discrete indices representing frequency and time window center, respectively. Thus, feature $S(\omega,t)$ can be thought of as a temporally local slice of the entire audio waveform's spectrogram. A feature S_i from one clip is applied as a template against the entire spectrogram S_j from another clip to compute the best time offset t_{off} according to

$$t_{ij}^{\text{off}} = \text{argmax}_T \sum_t \sum_\omega W^2(\omega)\tilde{S}_i(\omega, t+T)\tilde{S}_j(\omega, t) \qquad (4)$$

akin to normalized cross-correlation, where

$$\tilde{S}(\omega, t) = [S(\omega, t) - \mu_t(\omega)]/\sigma_t(\omega) \qquad (5)$$

and μ_t and σ_t denote the mean and standard deviation of S, with respect to time. Each frequency band is weighted by W, the inverse total energy in that band over the entire clip, so that "non-informative" frequencies have smaller influence on the correlation score. The correlation function (4) can be computed very efficiently in the frequency domain as the inverse of a product of fast Fourier transforms.

A set of local time offsets is thus estimated across all clip pairs, with each offset casting a vote for the most likely global time offset for a particular pair. A beneficial side effect of the correlation functions is evaluation of the degree of audible similarity in the form of an overall pairwise link score $q_{ij} = a_m\ a_s\ a_u$, with terms denoting absolute correlation magnitude, local sharpness, and global uniqueness of each peak. We form a weighted undirected association graph whose nodes represent the clips and whose edge weights are derived from q_{ij}.

3.2. Spatial Camera Matching

Because we wish to apply cross-video tracking and association only over those clips that view common content, we associate clips with one another according to their visual appearance. To determine visual similarity, we extract SIFT keypoints and descriptors [19] from a set of representative frames, efficiently match keypoint sets across frames using approximate nearest-neighbor search, and apply weak geometric constraints in the form of the fundamental matrix to arrive at a set of spatially consistent keypoint matches. We then form a second weighted undirected association graph whose nodes represent the clips and whose edge weights are determined according to the degree of visual similarity between clips.

Similarity in appearance is a sufficient but not a necessary condition for strong spatial and temporal association. In many cases, multiple cameras may be co-located or view the same foreground scene, but from such different perspectives that there is no direct visual similarity—particularly on the background (Figure 3 and Figure 4). We therefore simultaneously incorporate both audible and visual cues by fusing edge weights of the two association graphs described above. Candidate associations are then considered if either their appearance or audio content is strongly correlated, and rejected outright if neither condition is met.

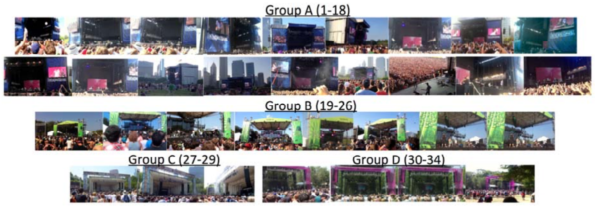

Figure 3. Above shows screenshots of our testing dataset (videos 1-34) and their respective spatial groupings.

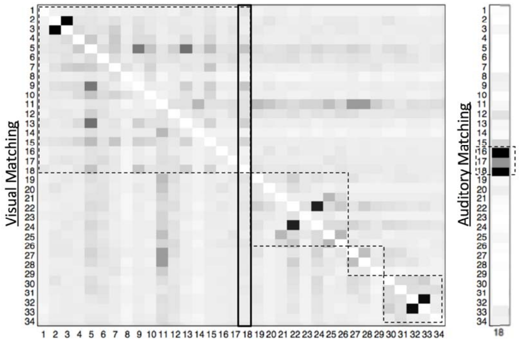

Figure 4. Above (left) is the confusion matrix of overlapping spatial matching among each of the videos (darker is better). The dashed boxes show the truthed groupings. Above (right) shows the audio match of each video to our reference video (18). Note that while some visual matching occurs among each spatial overlapping group, audio matching provides a strong indicator of spatial-temporal overlap and allows us to find 3 overlapping videos.

4. Camera Alignment

After removing outlier clips outside the space-time window of interest, inter-camera geometric relationships must be determined among the remaining overlapping views. As previously discussed, failure of keypoint matching in view-disparate scenarios requires the use of tracks to align cameras. In crowd-sourced event videos, pedestrians (e.g. performers, actors, presenters) within the scene are typically the focus of the crowd's attention and are visible within each of the cameras in the network.

4.1. Pedestrian Detection

Finding objects that are visible within the overlap region of pairs of cameras using traditional motion-based detection/tracking techniques is difficult in crowd-sourced video data for several reasons. In most videos near-field objects (other event observers) typically occlude the view of the camera for brief periods of time. These near-field objects are indistinguishable from potential moving tracks in a motion detection image. In addition, the videos often suffer from jitter and rolling shudder effects that further prohibit effective use of motion information. For these reasons we utilized still image pedestrian detection techniques that are more robust against these effects.

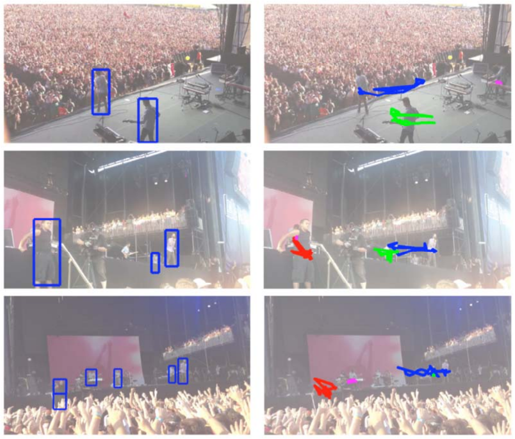

Figure 5. Three different viewpoints of a single time instant depicting single-frame pedestrian detections (left column) and pedestrian tracks projected back into a reference frame using frame-to-frame registration (right column).

In this work we applied [20] to locate persons of interest within the overlap area. Modern algorithms such as this produce robust detections of isolated individuals even in cluttered environments; however, the presence of crowds in typical event scenes still causes occasional spurious detections. To filter these, we enforce temporal consistency of the detector over time, by stitching detections from individual frames into tracks and removing tracks that fail to meet a minimum length criterion. We used a simple gating algorithm, based on a constant velocity and size assumption, to connect detections over time. Our assumption of purely projective (pan-tilt-zoom) image motions allows utilization of image registration to estimate geometric frame-to-frame relationships in the form of plane projective homographies; this assists consistent tracking via stabilization of detections with respect to a common reference frame.

In addition to filtration, connecting individual detections into tracks also serves to aid in pedestrian matching across views. Figure 5 shows an individual frame of pedestrian detections as well as tracks projected back into the same frame.

4.2. Projective Alignment

Each pair of views can now be aligned using the pedestrian tracks established in each individual view. In [3], views were aligned across a stationary camera network using a minimum number of track correspondences to constrain a particular camera pair. We extend this algorithm to the moving camera case by incorporating the frame-to-reference registration computed previously:

$$d^o\left(s_i^o, s_j^o, t_{ij}^{\text{off}}, H_{ij}, H_i^{\text{ref}}, H_j^{\text{ref}}\right) = \left\| H_j^{\text{ref}}(t) s_j^o - H_{ij} H_i^{\text{ref}}(t + t_{ij}^{\text{off}}) s_i^o \right\| \quad (6)$$

Without any a-priori knowledge of track correspondences, an exhaustive search over all possible track pairs is required to establish the projective relationships between cameras. To reduce the exponential time requirements of such a search, we incorporate features derived from each track to produce track match likelihoods across views and rank potential match candidates. Using track motion information of each pedestrian track within each camera we calculated the correlation of the targets. The motion of each object was normalized and compared with other temporally overlapping tracks. Figure 6 shows a sample confusion matrix of motion correlations between views; by ranking the highest likelihood assignments between pedestrian tracks within each view, we greatly reduce the number of correspondence tests required to find an acceptable alignment.

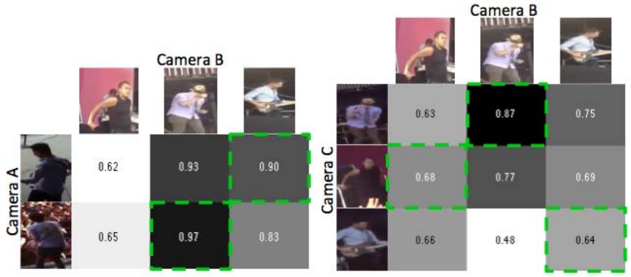

Figure 6. Confusion matrix for two pairs of views based on correlated motion over time. True matches are indicated by green dashed borders.

5. Cross-Camera Tracking

Inter-camera spatial-temporal alignment and intra-camera stabilization homographies allow projection of image locations from any frame (within the planar region) into any of the other cameras' viewpoints. These relationships may be used, for example, to provide a user with all additional views of a particular subject that are available at any given time (see Figure 7).

Within camera C_i the user nominates a subject of interest at location (x,y). At the nomination timestamp, parallel views can be computed at C_j using:

$$C_j(x_j, y_j) = H_j^{\text{ref}}(t)^{-1} H_{ij} H_i^{\text{ref}}(t + t_{ij}^{\text{off}}) C_i(x_i, y_i) \qquad (7)$$

where H_i^{ref} is the projective transform from the current frame to the reference frame within a camera and H_{ij} is the projective transform from the nominated camera to the additional viewpoint.

In this work we utilized the Multiple Instance Learning (MIL) tracker [1] to maintain appearance matching of the subject over time. We modified the MIL tracker to support repositioning of track locations using inter-frame registration, so that only relative object motion need be considered. We did not utilize pedestrian detection information from the camera alignment stage, because a user may wish to track non-pedestrian subjects or people that were filtered out by temporal consistency constraints.

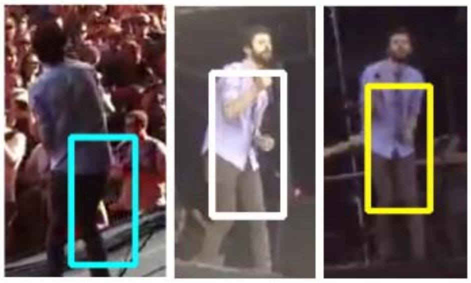

Figure 7. Three simultaneous views of a selected subject. The subject is tracked in Camera B using the MIL tracker (center), and the track box is projected into two other views (left and right).

6. System Performance

We tested our system on a dataset collected from YouTube that comprised 34 clips from a popular public music festival obtained as the result of a keyword search. The videos span many different stages and performance groups over several days. We evaluated the performance of each step in the grouping and alignment process, and demonstrated efficacy of alignment by automatically providing multiple views of a user-selected subject over time.

6.1. Camera Grouping and Audio Alignment

After selecting a subject from one of the 34 videos (Camera B) in our dataset, we utilized the combination of visual and auditory features to find two additional clips with overlapping space-time regions (Cameras A and C). All three videos were collected at 30fps from handheld devices (most likely mobile smartphones) and contain large amounts of rotational motion and jitter. Cameras A and B captured at 720p resolution, while camera C captured at 1080p.

Using the audio-based registration techniques discussed previously, we were also able to temporally align the three overlapping videos. Table 1 reports the time offsets with respect to Camera A as well as errors in offsets as compared with manually generated ground truth; the maximum error between any pair of videos was 1/10th of a second, part of which may be due to sound propagation delays between disparate observers.

Camera	Truth Offset (frms|*sec.*)	Computed Offset (frms|*sec.*)	Error (frms|*sec.*)
A	0|*0*	0|*0*	**N/A**
B	-1481|*49.37*	-1483|*49.43*	**2|*0.06***
C	-288|*9.6*	-291|*9.7*	**3|*0.1***

Table 1. True temporal offsets and our automatically computed offsets for each camera. All errors were within 100ms of ground truth.

6.2. Projective Alignment and Cross-Camera Tracking

Pixel-level registration accuracy is desirable for providing accurate cross-camera views of an object being tracked within the initialization camera. During the alignment process pedestrian detectors are used to find and track individuals for correlation (Figure 8).

The average image-relative reprojection error of selected object centroids after projective alignment is shown in Table 2. Though the alignment solution produces errors on the order of tens of pixels, it is important to note that (1) the average pedestrian size within each view is approximately 75x200 pixels; (2) the images are HD-resolution (1280x720 or 1920x1080); (3) the stage surface is viewed from an inherently unstable slant geometry, and (4) the pedestrian detection algorithm produces centroids that are strongly affected by partial occlusion and motion blur.

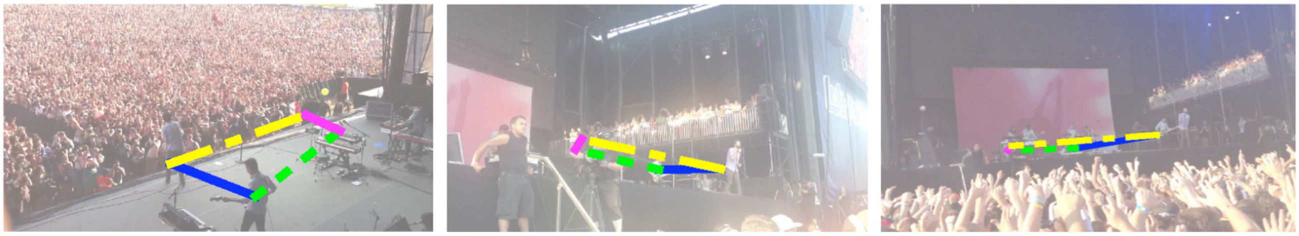

Figure 8. A plane drawn in Camera A (left) of our test sequence is reprojected accurately in Camera B (center) and Camera C (right).

Target	Avg. Dist. B-A	Avg. Dist. B-C	Avg. Target Size in B
Subject 1	15.5px	93.2px	108x283 px
Subject 2	35.9px	57.7px	51x127 px
Subject 3	78.7px	73.21px	89x174 px

Table 2. For three subjects, the average association errors (in pixels) of automatic pedestrian tracks used to align the cameras.

To evaluate the reprojection accuracy of targets over time, we truthed two individuals in all three cameras over 1200 frames and plotted the errors (see Figure 9). The results show that errors for both targets over the common region did not ever exceed 200 pixels from Camera B to Camera A. Both targets were less accurately reprojected from Camera B to Camera C; average errors were approximately 200px and 300px for subjects 1 and 2, respectively. Oscillations of errors can be seen and are correlated with movement around the stage by the performers. The reprojection errors do not drift over time from the keyframe used for cross-camera alignment.

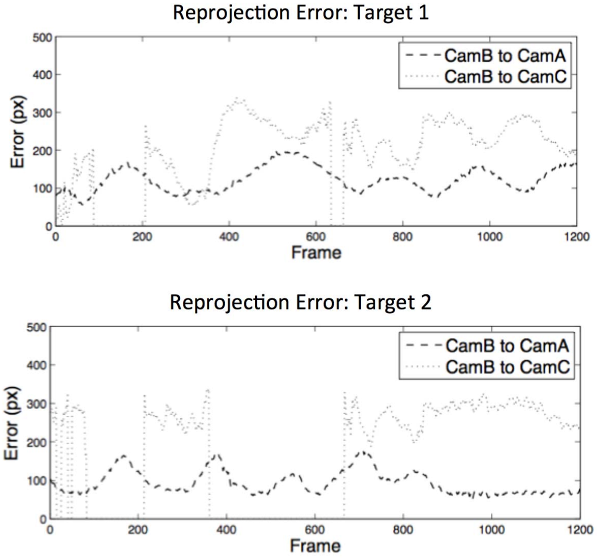

Figure 9. The error with respect to ground truth (in pixels) across two views for object 1 (top) and object 2 (bottom). Zero values are used when objects leave the camera FOV.

In addition to using truth target data to test reprojection error, we implemented and ran a modified version of the MIL tracker, capable of position updates from an externally supplied homography, on the test dataset (Figure 10). The error of the tracker wrt truth remains under 200 pixels during the 1000 frames tested and all reprojected viewing windows remained under 200 pixels of error with respect to truth in each camera. Because of the discrepancy in viewing angles, errors within one view do not necessarily cause an error in the reprojected view.

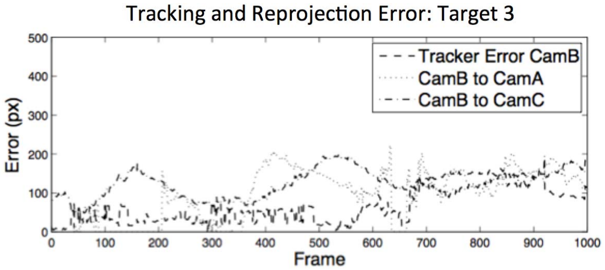

Figure 10. MIL tracker error for camera B vs. truth, along with reprojection errors of the tracks' positions in two other views.

7. Conclusion

In this paper we presented a method to correlate spatially and temporally overlapping videos collected opportunistically from multiple observers of an event, find the projective geometric relationship among overlapping viewpoints, and provide an end user with multiple views of a subject under track (Figure 11). We have demonstrated the system's effectiveness on a real dataset collected from YouTube, and reported results for the accuracy of visual grouping, temporal alignment, cross-camera reprojection and tracking. Future enhancements to this work include solving for spatial overlap, temporal overlap, audio alignment and geometric configuration within a single framework; extending from pair-wise association to global network optimization; and using track correlations to more precisely estimate temporal alignment.

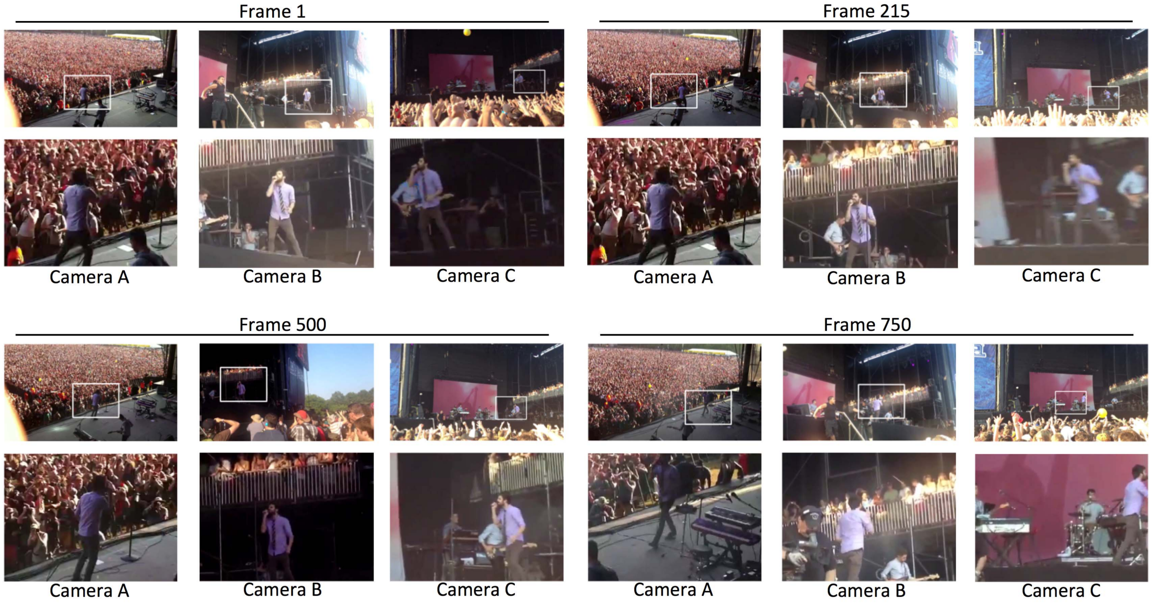

Figure 11. Above are four frames from three overlapping cameras within our testing dataset. The target's location is being tracked in Camera B and two additional viewpoints (Cameras A and C) are found to contain the target. During this 1000 frame test sequence the target remains in view within each stabilized window.

References

[1] B. Babenko, M. Yang, and S Belongie. Visual Tracking with Online Multiple Instance Learning. In CVPR, pages 983–990, 2009.

[2] T. Pollard, and M. Antone. Detecting and Tracking All Moving Objects in Wide-Area Aerial Video. In WCNWASA, pages 15-22, 2012.

[3] C. Stauffer, and K. Tieu. Automated Multi-Camera Planar Tracking Correspondence Modeling. In CVPR, 2003.

[4] S. Agarwal, N. Snavely, I. Simon, S Seitz, and R. Szeliski, Building Rome in a day. In ICCV, Pages 72-79, 2009.

[5] M. Song, D. Tao, and S. Maybank. Sparse Camera Networks for Visual Surveillance – A Comprehensive Survey. In The Computing Research Repository, January 2013.

[6] Z. Zhang, A. Scanlon, W. Yin, L. Yu, and P. Venetianer. Video Surveillance using a Multi-Camera Tracking and Fusion System. In Proc. M2SFA2, Marseille, France, 2008.

[7] A. Oliva, A. Torralba. Building the gist of a scene: the role of global image features in recognition. Progress of Brain Research, 2006.

[8] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In ICCV 2003, pp. 1470-1477.

[9] M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. The International Journal of Robotics Research 30.9 (2011), pp .1100-1123.

[10] A. Barton-Sweeney, D. Lymberopoulos, and A. Savvides. Sensor Localization and Camera Calibration in Distributed Camera Sensor Networks. In Proc. IEEE BROADNETS 2006.

[11] C. T. Aslan, K. Bernardin, R. Stiefelhagen. Automatic Calibration of Camera Networks based on Local Motion Features. Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications, 2008.

[12] D. Devarajan, R. Radke. Distributed metric calibration of large camera networks. In Proc. 1st Workshop on Broadband Advanced Sensor Networks, 2004.

[13] V. Nozick and H. Saito. Real-Time Free Viewpoint from Multiple Moving Cameras. In Proc. ACIVS 2007, pp. 72-83.

[14] M. Grochulla, T. Thormahlen, and H.P. Seidel. Using Spatially Distributed Patterns for Multiple View Camera Calibration. In Proc. MIRAGE 2011, pp. 110-121.

[15] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, H.-P. Seidel. Markerless Motion Capture with Unsynchronized Moving Cameras. In CVPR 2009, pp. 224-231.

[16] M. Taj and A. Cavallaro. Multi-Camera Track Before Detect. In Proceedings of the International Conference on Distributed Smart Cameras, 2009, pp. 1-6.

[17] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking Across Multiple Cameras With Disjoint Views. In Proc. ICCV 2003.

[18] Y. Sheikh, X. Li, and M. Shah. Trajectory Association across Non-overlapping Moving Cameras in Planar Scenes. In Proc. CVPR 2007.

[19] D. G. Lowe. "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

[20] P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade Object Detection with Deformable Part Models. In CVPR, 2010.