Sequence Alignment with the Hilbert-Schmidt Independence Criterion

Jordan Campbell*
University of Otago
j.campbell@windowslive.com

J.P. Lewis*†
Weta Digital
noisebrain@gmail.com

Yeongho Seol
Weta Digital
seolyeongho@gmail.com

ABSTRACT

We present a technique for establishing the temporal alignment between two videos of a single scene, by measuring the statistical dependence between the videos using the Hilbert-Schmidt independence criterion. Unlike previous approaches our technique does not require any feature correspondences between views, nor does it even require the two views to have any scene points in common. We show that our technique can handle arbitrary camera configurations, and can tolerate small camera motions. We demonstrate results on a number of test sequences, including cluttered outdoor scenes and those with significant occlusions.

CCS CONCEPTS

• **Computing methodologies → Computer vision**;

KEYWORDS

Video synchronization, Temporal alignment, Statistical dependence, Hilbert-Schmidt Independence Criterion

ACM Reference Format:
Jordan Campbell, J.P. Lewis, and Yeongho Seol. 2018. Sequence Alignment with the Hilbert-Schmidt Independence Criterion. In *CVMP '18: European Conference on Visual Media Production (CVMP '18), December 13–14, 2018, London, United Kingdom.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3278471.3278475

1 INTRODUCTION

Recording a scene from multiple cameras is a common task in vision, and it is generally required that the separate videos be synchronised before subsequent processing. This is usually done using specialised equipment in controlled settings.

Specialized equipment is not ideal, however, and not only because of the cost of such equipment. In outdoor settings (Fig. 1) it can be awkward to carry along and setup the various timecode boxes and cables. Motion capture (including vision-based motion capture) is increasingly common in movie production, and it may be required to change locations or camera positions between shots (in this paper a "shot" refers to a short sequence, typically a few seconds in length, delimited by discontinuous changes in camera view). In this setting, even a few minutes of setup time can be a significant cost, considering that professional actors and a sizeable crew (makeup artists, stunt people, etc.) are standing by.

When synchronized cameras are not available, videos can be manually synchronized by aligning a "high frequency" event such as a foot-fall or the traditional manually triggered clapperboard. However, this requires that the same event be visible in all cameras, which requires extra care and is not always possible. For example, a clapper visible in a camera viewing the scene from in front of a group of actors may not be visible from cameras viewing the same group from the side or rear. Videos can also be aligned by aligning their audio tracks, however this may also be challenging in outdoor settings due to wind or other factors.

In cases where the camera views are somewhat similar, the temporal synchronisation problem can be approached with computer vision techniques. If it is possible to detect common feature trajectories in each video, then the desired offset can be sought as a variant of a curve matching problem. Identifying corresponding features is difficult in itself, however, and is not feasible if the camera views are quite different. More fundamentally, we show that it is not necessary – it is possible to find dependencies between the statistics of two sets of non-corresponding features from arbitrary views of a common subject.

In this paper we introduce a technique that can easily synchronise the separate videos of a single shot after the recordings are obtained. It can directly handle the case of pairs of cameras having radically different view directions (as shown in the Figures) with few or even no features in common, and does not require additional intermediate cameras to propagate the synchronization. Our approach assumes that separate cameras will maintain a fixed (if unknown) temporal offset during the recording, which is true of modern digital equipment at least over a short period.

Our approach is based on the observation that when the sequences are in the correct temporal alignment, separate views of the same motion will not be independent. By measuring the statistical dependence we are asking whether it is likely that statistics from each view are generated by the same underlying object, i.e. whether they are a function of the same underlying distribution. However, such a relationship is likely to be complex and not something that can be discovered with linear correlation or other simple methods. We employ the Hilbert Schmidt independence criterion (Section 4) as a measure of the (lack of) statistical independence at various candidate temporal offsets.

The resulting method is both simple to implement and robust. It works with relatively low-quality "features", and does not require

*Authors contributed equally.
†Currently at SEED, Electronic Arts

CVMP '18, December 13–14, 2018, London, United Kingdom

ACM ISBN 978-1-4503-6058-6/18/12…$15.00
https://doi.org/10.1145/3278471.3278475

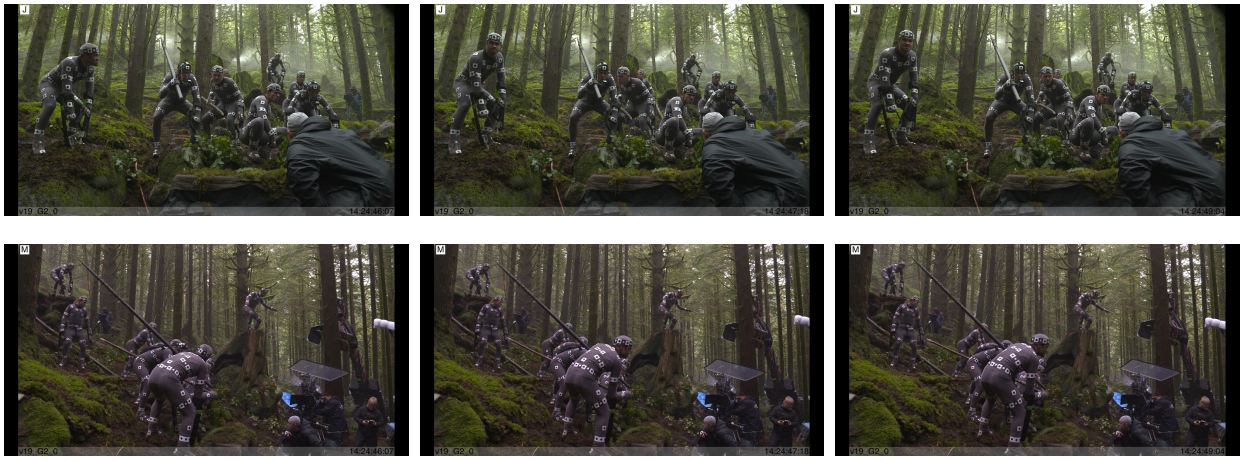

Figure 1: Outdoor motion capture sequence from the movie *Rise of the Planet of the Apes* for cameras J (top row) and M (bottom row). The largest motion in the scene is from the actor at the far left in the top row, who lifts his hands from the ground and then places them back down again. This actor appears in the foreground at the centre in the bottom row. From left to right the frames shown are 1, 36, and 70 respectively.

the same features to be visible across the various camera views, nor even features that are in common across the video from a single camera. Although our method is targeted at movie production, we anticipate that it (and the underlying approach) may be used in other problems.

2 RELATED WORK

Our survey of related work will be brief, since existing methods generally require fairly similar camera views having shared features, and do not address the situation shown in the figures here in which the cameras have different or even opposite view directions.

Caspi et al. [Caspi et al. 2006] use features tracked in each view to estimate the spatial and temporal alignment parameters. Their algorithm iteratively estimates the spatial and temporal parameters by minimising the Euclidean distance between pairs of trajectories. In each iteration the candidate matches are used to estimate the fundamental matrix between the two views. This may require background segmentation in order to identify a single moving object from which a set of feature trajectories can be extracted. Elhayek et al. [Elhayek et al. 2012] extend the approach of [Caspi et al. 2006] to deal with multiple cameras and arbitrary time shifts. They use a RANSAC based trajectory matching approach that allows them to use arbitrarily tracked features rather than those specifically belonging to a prominent subject.

Whitehead et al. [Whitehead et al. 2005] present a method that operates in 2D projective space. They first estimate the scene geometry using static image features, which are tracked independently in each camera to build a set of feature trajectories. They then use the epipolar geometry to establish correspondences between views. Correspondences are determined by projecting points from each trajectory in the first view onto the epipolar lines defined in the second view. Normalised cross-correlation is used to determine correct matches and finally the proposed correspondences are used to refine the initial scene geometry estimate.

Rao et al. [Rao et al. 2003] present a technique for performing 3D view invariant dynamic time warping (DTW) to address this situation. Given two input videos of the same scene at different times they first manually specify eight matching points between the two videos. These points are then tracked and the classic DTW algorithm is used to relate points in one trajectory to points in another. The authors account for 3D variations in viewpoints by introducing a shape measure into the DTW cost function.

Tuytelaars and Van Gool [Tuytelaars and Van Gool 2004] demonstrate an approach that uses known background points to define a fixed coordinate frame. Feature trajectories are established and then projected into this 3D coordinate frame, which allows feature trajectories to be matched across views without the need for camera calibration.

Douze et al. [Douze et al. 2016] introduce a Fourier-domain approach to correlating feature descriptors from a pair of videos, providing very efficient computation. They also describe an algorithm to align a collection of videos containing some pairs with widely differing views that cannot otherwise be aligned using a correlation-based approach. The algorithm operates on a graph in which edges indicate successfully aligned pairs of videos. Although widely differing views cannot be directly aligned, it may nevertheless be possible to align them through intermediate views between them. This approach requires having a sufficient "sampling" of intermediate camera views, which comes for free in the case of popular events such as a concert that are recorded by a number of people. It is less convenient in motion picture production where additional cameras require additional crew.

Yan and Pollefeys [Yan and Pollefeys 2004] align videos using a 3D extension of the Harris corner detector [Harris and Stephens 1988]. In order to synchronise two videos they first detect space-time interest points by applying a convolution operation to each video. They then find the temporal offset which generates the largest correlation between the distributions of interest points from

the two videos. This requires that the camera views are sufficiently close that the distribution of features can be related through linear correlation. It also requires that the scene have sufficiently fast movement to generate space-time corners. This method is advantageous as it does not require correspondences across views (as is also the case with our approach).

3 FEATURES

Our method is robust with respect to the choice of the underlying features, and the results shown here do not require a state-of-the-art optical flow algorithm. Initially we tried using the leading several PCA coefficients of a relevant sub-window of the images themselves, with good results in simple cases. In this paper we show results using PCA coefficients of optical flow vector trajectories (i.e. integral curves). This is a simple choice that appears to be sufficient in a variety of situations, and we did not experiment with other types of features. Using the raw optical flow vectors would also possible, however the PCA features allowed our approach to be easily compared to the baseline mutual information measure (Section 4) using these same features.

In each video we construct the feature matrix $\mathbf{D}_c = [\mathbf{z}_1, ..., \mathbf{z}_m]$ for camera c where each $\mathbf{z}_i$ is the column vector of vectorised 2D feature locations in frame i, $i \in (1, \ldots, m)$. We initialise the tracking process by selecting a sparse set of points from a uniform grid over the first image in each video. We then use optical flow [Farnebäck 2003] to track the points through the video (see Section 5). We use a naive feature tracking implementation that updates the 2D location of each point according to the flow vector of its corresponding pixel, without requiring that the appearance of the feature remains consistent. We then apply PCA to the matrix $\mathbf{D}_c$ to get the reduced trajectory matrix for camera c

$$\mathbf{T}_c = [\mathbf{t}_1, ..., \mathbf{t}_m]^T, \tag{1}$$

where each vector $\mathbf{t}_i \in \mathbb{R}^k$ is computed as $\mathbf{t}_i = (\mathbf{U}_{1:k})^T (\mathbf{z}_i - \bar{\mathbf{z}})$, i.e., the projection of $\mathbf{z}_i$ (with mean removed) onto the leading PCA basis vectors. We take only $k = 2$ most significant components. The inputs to our algorithm are the two-dimensional projected vectors $\mathbf{x}_i = \{\mathbf{t}_i \in \mathbf{T}_1\}$ and $\mathbf{y}_i = \{\mathbf{t_i} \in \mathbf{T}_2\}$ from each video.

4 MEASURING STATISTICAL DEPENDENCE

The goal of our algorithm is to estimate the statistical dependence between two video streams. Linear correlation is commonly used as a measure of statistical dependence, including in previous work on temporal alignment. Correlation is limited however in that it only considers the second-order statistical moments. Non-Gaussian second-order statistics, as well as the third and higher-order moments and statistics are not captured. Thus, correlation is fully justified only when the signals are fully specified by their second-order moments, i.e., the Gaussian case. In practice, the use of linear correlation requires similar views in which the observed features have a simple relationship.

In contrast to correlation, mutual information and related information theoretic measures reflect *all* statistical moments. The mutual information between discrete variables X and Y is given by $I(X;Y) = H(X) + H(Y) - H(X,Y)$, where $H(X)$ is the Shannon entropy of X, $H(X) = -\sum_{x \in X} p(x) \log p(x)$. Further, mutual information (in theory) captures *any* functional relationship between variables, i.e., $I(X;Y) = I(X; f(Y))$ for any deterministic invertible function $f()$. This is a more general notion than comparing probability distributions, and so measures such as KL divergence and Wasserstein distance are not applicable here. For example, negating a random variable will in general change its probability density and thus change its KL divergence with respect to another variable, but this transformation does not affect the mutual information with the other variable.

While computing the mutual information is straightforward given the probability densities for X and Y, computing the latter is difficult. In practice histograms or Parzen window techniques are used to estimate these functions from given data. However, probability densities are notoriously difficult to estimate from limited data. The estimated entropy and mutual information can differ depending on the chosen bin size or kernel width. Kernel-density and bin-based methods also suffer from the curse of dimensionality, requiring data that rises exponentially with dimension. Fig. 8 presents the results of bin-based estimation of mutual information on our problem, showing that it does not perform well in practice.

These problems have led to the development of alternate algorithms such as [Kraskov et al. 2004]. Here we show that the Hilbert-Schmidt independence criterion [Gretton et al. 2007] can be used as a robust measure of temporal alignment between feature trajectories from two videos.

4.1 Reproducing kernel Hilbert spaces

Reproducing kernel Hilbert spaces (RKHSs) are a Hilbert space in which functions can be represented as a weighted sum of translated copies of a symmetric kernel function, $f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i)$. Functions in a RKHS are more restricted than those in a general Hilbert space, for example, knowing the value of a function in a RKHS provides some information about its value at neighboring points, unlike the case in a general Hilbert space. This is suggested intuitively from the "convolutional" form of RKHS functions; it can also be understood from the required decay of the spectrum of the kernel.

Applications of RKHSs in machine learning are often tied to the kernel trick. The partially evaluated kernel $k(\cdot, x)$ can be regarded as a function mapping data x into a (typically high- or infinite-dimensional) feature space. The fully evaluated kernel $k(x, y)$ defines the similarity between x and y, acting as an inner product in the high-dimensional space. The feature map $k(\cdot, x)$ is never explicitly computed as algorithms are expressed using $k(x, y)$.

The statistical power of the Hilbert-Schmidt independence criterion (defined below) for our problem relies on the use of a universal or *characteristic* kernel [Sriperumbudur et al. 2010]. Characteristic kernels have the property that

$$P = Q \quad \textit{iff} \quad \mathbb{E}_{X \sim P}[k(\cdot, x)] = \mathbb{E}_{Y \sim Q}[k(\cdot, y)]$$

i.e., the equality of two probability distributions can be determined by their expectations in feature space. The Gaussian kernel

$$k(x, y) = \exp\left(-\|x - y\|^2 / 2\sigma^2\right)$$

is characteristic and corresponds to an infinite-dimensional feature space. An intuition is provided here by the fact that the series

expansion of the exponential contains a particular weighted sum of all powers (in the statistical context, moments) of the data, and two distributions are equal if and only if all their moments are identical.

4.2 Hilbert-Schmidt independence criterion

The Hilbert-Schmidt independence criterion (HSIC) is defined as the difference between joint and marginal distributions, but as measured through the maximum mean discrepancy (MMD) [Gretton et al. 2012], a kernel-based measure of the difference between distributions

$$HSIC := MMD^2(P_{XY}, P_X \otimes P_Y)$$

rather than with the Kullback-Leibler divergence as in the case of mutual information. An alternate statement is that HSIC is the squared Hilbert-Schmidt norm of the cross-covariance in feature space,

$$\begin{aligned} HSIC :=& \|C_{XY}\|^2_{HS} \\ =& \left\|\mathbb{E}_{XY}[(\phi(x) - \mathbb{E}_x(\phi(x))) \otimes (\psi(y) - \mathbb{E}_y(\psi(y)))]\right\|^2_{HS} \end{aligned}$$

where $\psi(y)$ is a feature mapping defined analogously to $\phi(x)$. Under useful assumptions ([Gretton et al. 2005, Theorem 4]) $\|C_{XY}\|_{HS} = 0$ if and only if X and Y are independent. This resembles the statement that two multivariate Gaussian variables are independent when all entries of their cross-covariance matrix are zero, but it applies to arbitrary distributions.

An empirical estimate of the HSIC is given by[1]

$$\text{HSIC}(X, Y) = \frac{1}{n^2}\text{tr}(\mathbf{KHLH}) \tag{2}$$

where $\mathbf{K}$, and $\mathbf{L}$ are the Gram matrices for the kernels $k(x_i, x_j)$ and $l(y_i, y_j)$ respectively. The centering matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$$

with $\mathbf{1}$ an $n \times 1$ vector of ones. Note that $\mathbf{H}$ is idempotent.

An intuition for this expression can be found by considering the ith element of the trace,

$$\tilde{\mathbf{K}}\tilde{\mathbf{L}}_{i,i} = \left[\tilde{k}_{i,1}, \tilde{k}_{i,2}, \cdots, \tilde{k}_{i,n}\right] \cdot \left[\tilde{l}_{i,1}, \tilde{l}_{i,2}, \cdots, \tilde{l}_{i,n}\right]$$

where $\tilde{k}_{j,k}$ denotes an element of the centered kernel matrix $\tilde{\mathbf{K}} = \mathbf{HKH}$ and likewise for $\tilde{l}_{j,k}$. A particular $\tilde{k}_{ij}$ reflects the relation or similarity of x_i and x_j as measured through the kernel

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{K}}$$

where $\langle , \rangle_{\mathcal{K}}$ denotes the inner product in the RKHS corresponding to this kernel, and likewise for $l(y_i, y_j) = \langle \psi(y_i), \psi(y_j) \rangle_{\mathcal{L}}$. This inner product will be large when the relation between sample x_i and all other x_j is similar to the corresponding relation between y_i and all other y_j.

We use the Gaussian kernel with the k_{ij} entry of $\mathbf{K}$ given by

$$k(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$$

with $\sigma = 40$ chosen empirically. The kernel $l(\cdot, \cdot)$ is defined similarly.

[1]See [Gretton et al. 2005] for the derivation of this expression. The corresponding unbiased estimate is $\text{HSIC}(X, Y) := \frac{1}{n(n-1)}\text{tr}(\mathbf{KHLH})$

Unlike mutual information, HSIC does not have an interpretation in terms of information theoretic quantities (bits or nats). On the other hand, HSIC does not require density estimation and is simple and reliable to compute. Kernel distribution embedding approaches such as HSIC can also be resistant to outliers, as can be seen by considering the effect of outliers under the Gaussian kernel. The empirical estimate converges to the population HSIC value at the rate $1/\sqrt{n}$ independent of the dimensionality of the data, meaning that it at least partially sidesteps the curse of dimensionality.

5 RESULTS

We demonstrate our results on a number of different test sequences, across a number of different conditions. Our algorithm permits an arbitrary configuration of cameras, as well as arbitrary scene activity. The only requirement is that the camera fields of view overlap. We show that our algorithm can tolerate small camera motions, however it fails when presented with camera motions that are large relative to the scene activity. For each pair of videos we manually specify a sequence of frames in one camera to act as the reference sequence. We then find the selection of contiguous frames from the test camera that maximises the HSIC defined in (2). We initialise our algorithm by extracting the feature trajectories for each camera using Farneback dense optical flow [Farnebäck 2003]. The optical flow and feature tracking methods are not the state-of-the-art, however they were chosen for their convenience and ease of use. We recognise that our algorithm could be straightforwardly improved by choosing better feature tracking algorithms.

As for computing time, the optical flow tracking takes most of the time, and the rest of the process is very fast based on Eq. (2). For a pair of 100-frame videos of 480×270 resolution, the optical flow tracking takes 33.7s. The rest of the process, which finds the best offset out of 41 candidates, takes 0.13s for PCA and 4.67s for HSIC (average 113ms per candidate temporal offset). We use 2040 uniformly sampled tracking points and 2 PCA coefficients to estimate the HSIC. Our experiments are performed on an Intel Xeon CPU E5-2620 2.00GHz with 47GB RAM.

5.1 Checkerboard sequences

Our first experiment consists of pairs of static cameras, with a single moving checkerboard. These sequences were initially recorded as input to a multi-view calibration procedure. The ground truth temporal offset was given by a hardware synchronisation. A sparse sample of the feature trajectories from our initial set of points are shown overlayed in Fig. 2.

The results in Fig. 3 are for two different sets of cameras. Figs. 3(a) and 3(b) are for cameras C and E while Fig. 3(c) is for cameras F and G. The correct temporal offset was -10 and -12 frame for cameras C/E and F/G respectively. Our results show that our algorithm was correct for two of the test cases and was within one frame (0.03 seconds) of the correct result in another test. These experiments were chosen due to the ease with which tracking can be performed.

5.2 Motion capture sequences

We tested our results on a challenging scene that contains three dynamic actors, a mix of stationary and dynamic cameras and multiple occlusions. As can be seen in Fig. 4, Fig. 5 and Fig. 6, the

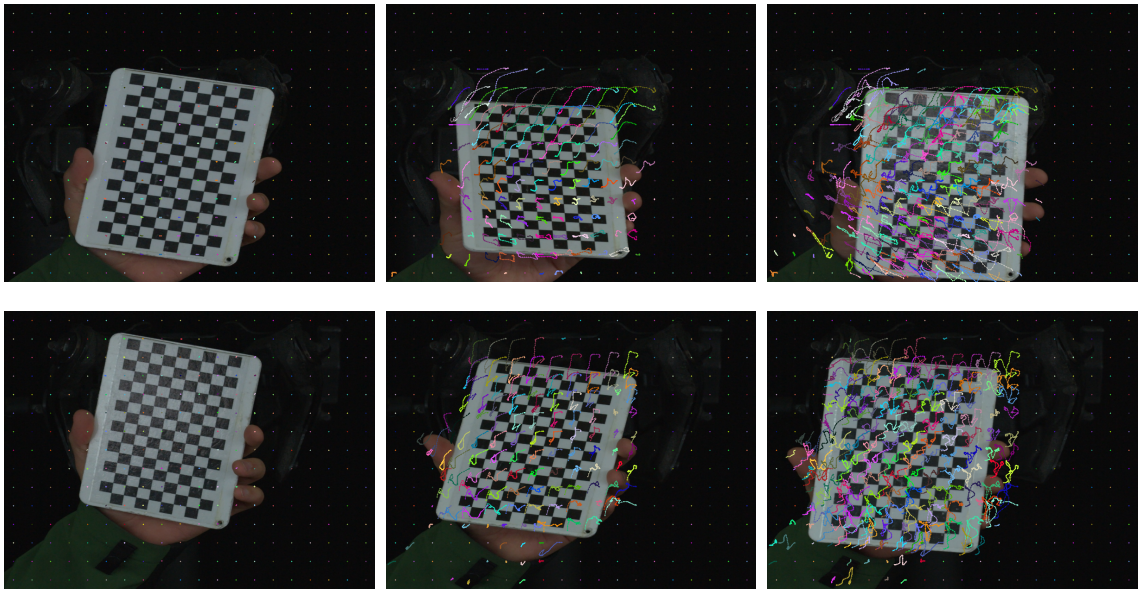

Figure 2: Three frames from the calibration sequence for camera C (top row) and camera E (bottom row). These images show only a sparse set of trajectories for ease of visualisation. The images have been cropped and scaled for display.

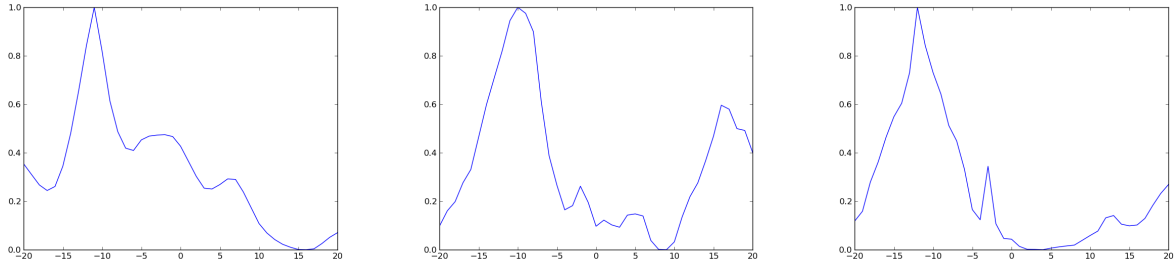

Figure 3: Three checkerboard calibration sequences. In this and the subsequent result graphs, the horizontal axis is the temporal offset and the vertical axis is the normalized HSIC value, with zero indicating statistical independence and one indicating maximum dependence. The results in (a) and (b) are for cameras C and E, for frames 1 - 101, and 100 - 200 respectively. The results in (c) are for frames 100 - 200, with cameras F and G. The ground truth offsets are -10 and -12 frames for C/E and F/G, respectively. The results in (b) and (c) are correct while the results in (a) are within one frame of the correct result.

actors are often occluded by scene objects and by the other actors. The appearance of the scene is relatively uniform and the cameras are separated by large differences in viewpoint. As a result, the feature trajectories from these scenes are less accurate typically due to the influence of multiple scene elements. In all of the following motion capture sequences, the cameras are already aligned and therefore the correct offset is zero.

The result in Fig. 7(a) is for the images shown in Fig. 4, Fig. 7(c) is for the images shown in Fig. 5, and Fig. 7(e) is for the sequence shown in Fig. 6. The results in Fig. 7(a) and (e) return the correct result while the result in Fig. 7(c) is correct to one frame (0.03 seconds). The results in Fig. 7(b) and Fig. 7(d) are for different camera pairs from the sequences in Fig. 4 and Fig. 5, respectively. They are accurate to within three frames (0.1 seconds). Finally, the results in Fig. 7(f) are from an outdoor motion capture scene for Rise of the Planet of the Apes (Fig. 1). This scene features multiple actors who have relatively small and slow motion. The cameras are widely separated and the field-of-views do not overlap entirely. As a consequence there is relatively little information shared between the cameras, however as can be seen in Fig. 7(f) our algorithm is capable of returning correct results.

The same results are shown again in Fig. 8 with the mutual information results included. As can be seen the HSIC measure outperforms the mutual information on all the sequences.

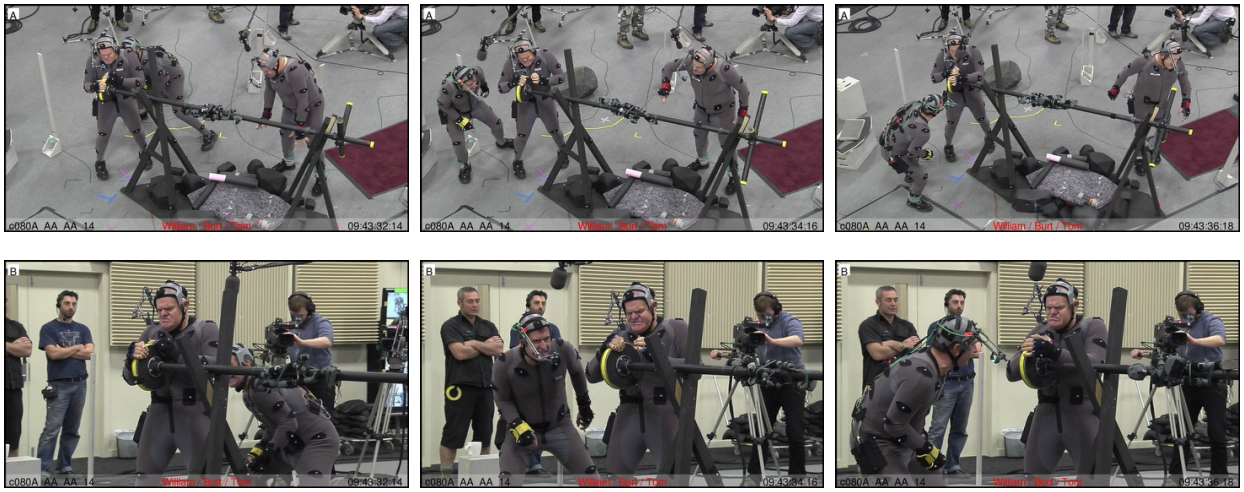

Figure 4: Motion capture sequence from *The Hobbit: An Unexpected Journey*, for cameras A (top row) and B (bottom row). In both rows we can see an actor moving from behind another actor towards the centre of the scene. From left to right the frames shown are 110, 160, and 210 respectively.

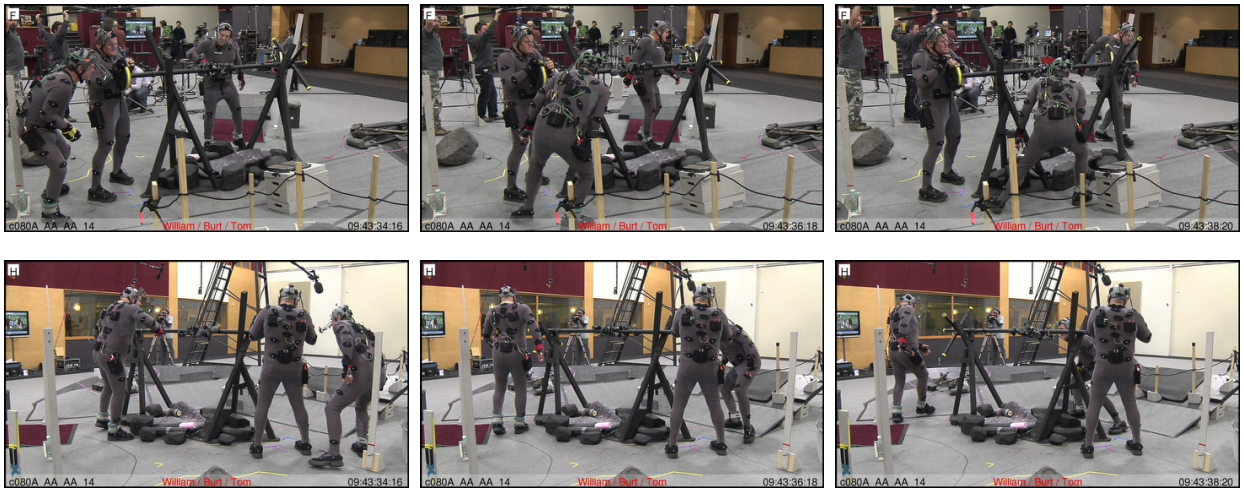

Figure 5: Motion capture sequence from *The Hobbit: An Unexpected Journey*, for cameras F (top row) and H (bottom row). In the top row we can see an actor moving towards the centre of the scene, while in the bottom row we can see the actor moving from the centre towards the right hand side, where he becomes occluded. From left to right the frames shown are 160, 210, and 260 respectively.

6 DISCUSSION AND CONCLUSION

In this work we have shown that the HSIC can be used as a robust and generally accurate measure for determining the temporal alignment between two video sequences. Our algorithm can handle arbitrary, widely differing camera views and occlusions because the HSIC captures the statistical dependence between different views of the same motion. Contrary to previous work our algorithm does not require explicit feature correspondences between views, nor does it require that the same features are even visible in both views. In addition, our method is frame accurate in many cases, and so is competitive even with frame-accurate methods that rely using very similar camera views to provide common features across the views.

Non-integer frame offsets as well as cameras that record at different frame rates (e.g. 24 vs 60fps) introduce a separate problem. Our technique does not address this resampling problem, though it may be handled to some extent with existing techniques. We only consider the case of synchronising two videos, although more videos may be handled by transitivity. Our method also does not handle synchronisation of videos where the dominant image movement is due to movement of the cameras rather than objects in the scene,

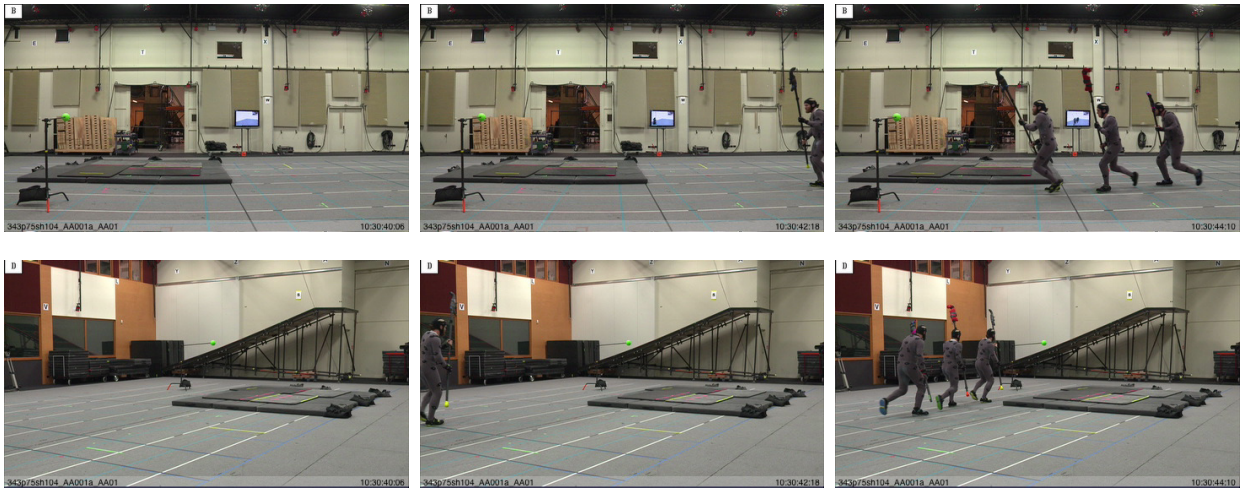

Figure 6: Motion capture sequence from *The Hobbit: The Battle of the Five Armies* for cameras B (top row) and D (bottom row). In the top row we can see the three actors as they appear from the right hand side, while in the bottom row we can see the same actors as they appear from the left hand side of the scene. From left to right the frames shown are 330, 390, and 430 respectively.

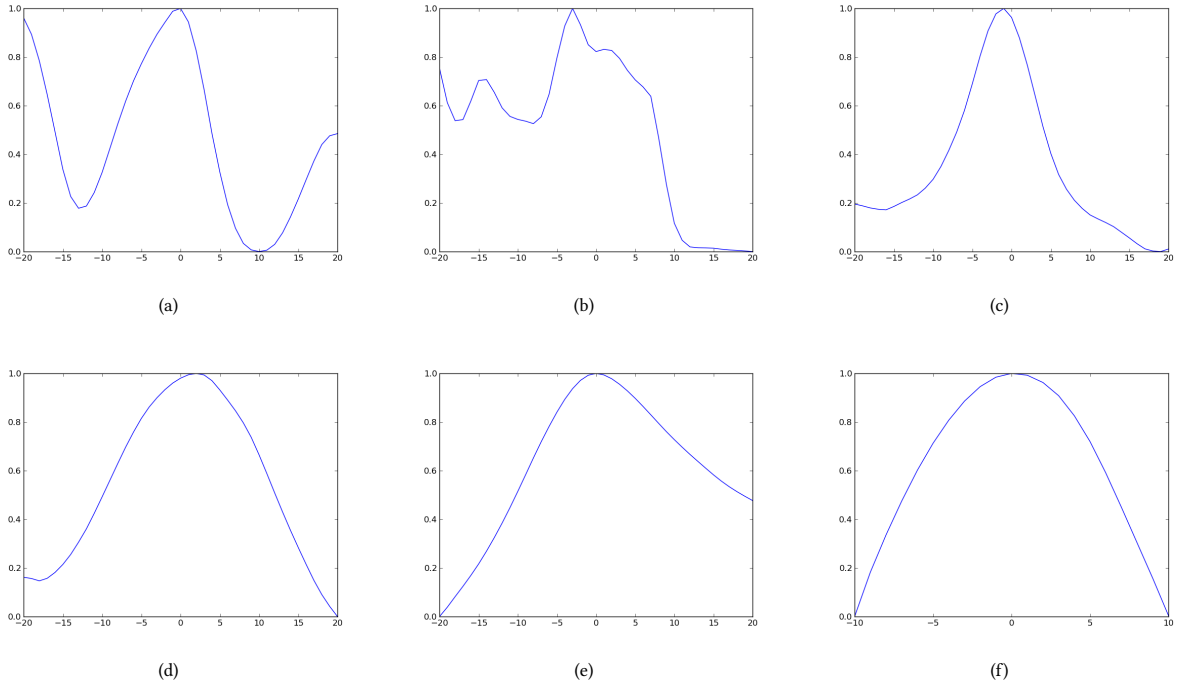

Figure 7: HSIC results for the six test sequences. In every experiment the ground truth offset is zero frames.

though a small amount of camera movement can be accommodated. Approximate segmentation of moving objects would solve this problem and allow our method to be applied, however this is in general more difficult than the simple temporal alignment problem we are addressing. Crude segmentation is feasible in some cases however, such as when the subjects have a clear colour difference from the background (Fig. 1).

We tested our algorithm on a number of challenging sequences, including scenes with multiple moving actors and multiple occlusions. Our algorithm comfortably tolerates occlusions, as well as

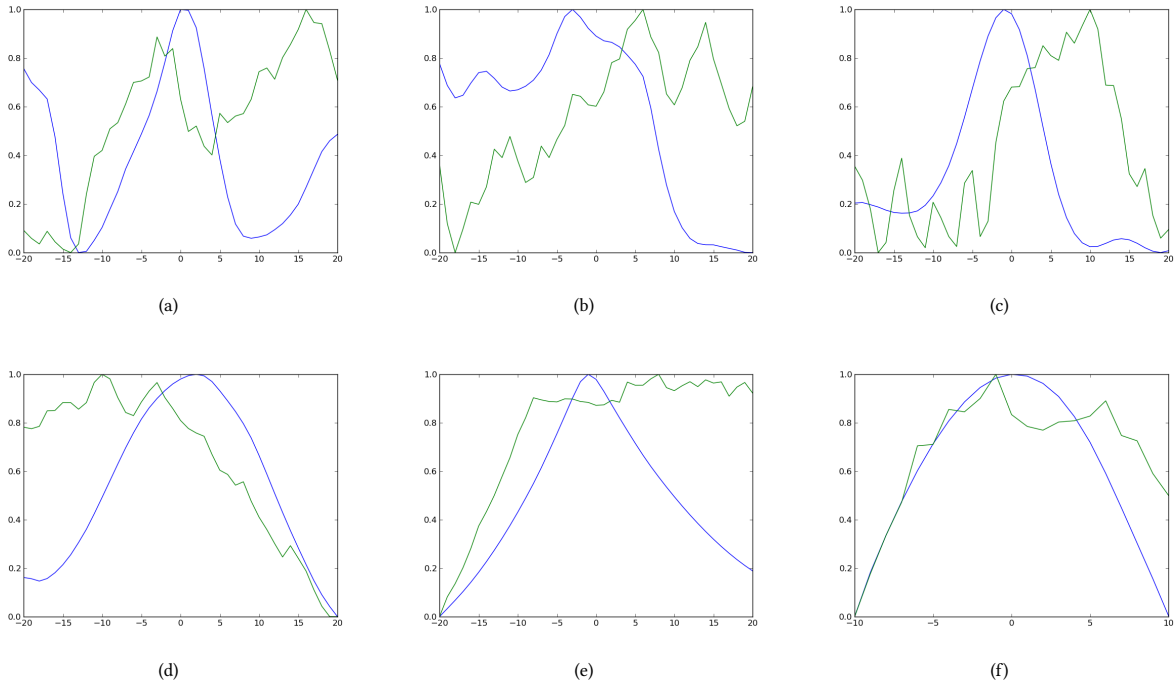

Figure 8: HSIC (blue trace) and mutual information (green trace) results for the six test sequences. In each plot the vertical axis is the information measure (HSIC or mutual information) normalized to have a maximum value of one, and the horizontal axis is the frame offset. In every experiment the ground truth offset is zero frames. In this experiment the mutual information bin width and HSIC kernel width σ were manually adjusted to provide the best results.

small camera motions. Although the method is occasionally inaccurate, it still gives a solution close to the ground truth within a few frames and makes the temporal alignment task efficient. As an extension, we are interested in testing our algorithm on different modalities, such as between video-audio or video-acceleration from an Inertial Measurement Unit (IMU).

ACKNOWLEDGMENTS

We thank Weta Digital, Joe Letteri and Dejan Momčilović for supporting this work. Jochen Tautges was a primary contributor to an earlier attempt to solve this problem, in which we used a conventional bin-based estimate of mutual information. JPL acknowledges discussions of RKHSs and related topics with Ken Anjyo and David Balduzzi. We also thank the reviewers for their helpful comments.

REFERENCES

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

Matthijs Douze, Jérôme Revaud, Jakob Verbeek, Hervé Jégou, and Cordelia Schmid. 2016. Circulant Temporal Encoding for Video Retrieval and Temporal Alignment. *Int. J. Comput. Vision* 119, 3 (Sept. 2016), 291–306.

Ahmed Elhayek, Carsten Stoll, Kwang In Kim, H-P Seidel, and Christian Theobalt. 2012. *Feature-based multi-video synchronization with subframe accuracy.* Springer. 266–275 pages.

Gunnar Farnebäck. 2003. Two-frame Motion Estimation Based on Polynomial Expansion. In *Proceedings of the 13th Scandinavian Conference on Image Analysis (SCIA'03).* 363–370.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. 2012. A Kernel Two-Sample Test. *Journal of Machine Learning Research* 13 (2012), 723–773.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. 2005. Measuring Statistical Dependence with Hilbert-Schmidt Norms. In *Algorithmic Learning Theory*, Sanjay Jain, Hans Ulrich Simon, and Etsuji Tomita (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 63–77.

Arthur Gretton, Kenji Fukumizu, Choon H Teo, Le Song, Bernhard Schölkopf, and Alex J Smola. 2007. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems.* 585–592.

Chris Harris and Mike Stephens. 1988. A combined corner and edge detector.. In *Alvey vision conference*, Vol. 15. Citeseer, 50.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Phys. Rev. E* 69 (Jun 2004), 066138. Issue 6.

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

Bharath K Sriperumbudur, Kenji Fukumizu, and Gert Lanckriet. 2010. On the relation between universality, characteristic kernels and RKHS embedding of measures. In *International Conference on Artificial Intelligence and Statistics.* 773–780.

Tinne Tuytelaars and Luc Van Gool. 2004. Synchronizing video sequences. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, Vol. 1. IEEE, I–762.

Anthony Whitehead, Robert Laganiere, and Prosenjit Bose. 2005. Temporal synchronization of video sequences in theory and in practice. In *Aplication of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, Vol. 2. IEEE, 132–137.

Jingyu Yan and Marc Pollefeys. 2004. Video synchronization via space-time interest point distribution. In *Advanced Concepts for Intelligent Vision Systems*, Vol. 1. 12–21.