

Pantomimic Gestures for Human-Robot Interaction

Michael Burke, *Student Member, IEEE*, and Joan Lasenby

Abstract—This work introduces a pantomimic gesture interface, which classifies human hand gestures using unmanned aerial vehicle (UAV) behaviour recordings as training data. We argue that pantomimic gestures are more intuitive than iconic gestures and show that a pantomimic gesture recognition strategy using micro UAV behaviour recordings can be more robust than one trained directly using hand gestures. Hand gestures are isolated by applying a maximum information criterion, with features extracted using principal component analysis (PCA) and compared using a nearest neighbour classifier. These features are biased in that they are better suited to classifying certain behaviours. We show how a Bayesian update step accounting for the geometry of training features compensates for this, resulting in fairer classification results, and introduce a weighted voting system to aid in sequence labelling.

Index Terms—pantomimic, gesture recognition, human-robot interaction, principal component analysis, time series classification

I. INTRODUCTION

An increased demand for service robots used by the general public has led to an emphasis on the design of simple and intuitive user interfaces, allowing for improved human-robot interaction. While traditional controllers such as joysticks, game-pads and other haptic interfaces are still abundant, the importance of speech and gesture in inter-human communication has led to a significant amount of work on human-robot interaction using these communication mechanisms.

Traditional approaches to gesture-based robot control have involved the use of pre-defined codebooks or dictionaries of gestures, mapped directly to desired robot behaviours [1][2]. These approaches typically require a significant amount of training on specific users, who also need to be aware of the set of commands used to select robot behaviour. Unfortunately, this prerequisite knowledge lowers the usability of gesture-based robot interfaces.

Logically, it would seem that pantomimic gestures, which attempt to mimic an object or action, would allow for the most intuitive interaction with a service robot. Rather than learn a predefined codebook of commands, a user informed that robot behaviour is selected by mimicking the desired action would be able to operate the robot with ease. Of course, difficulties in interpreting the wide variety of potential pantomimic gestures that could be used by an operator make the use of pantomimic gestures for human-robot interaction particularly challenging. The design of a comprehensive database of all possible

This work was supported by funding from the Council for Scientific and Industrial Research (CSIR), South Africa and the Cambridge Commonwealth Trust under a CSIR-Cambridge Scholarship.

M. Burke (michaelburke@ieee.org) is with the the Department of Engineering at the University of Cambridge, Cambridge, UK, CB2 1PZ and the Council for Scientific and Industrial Research, South Africa. J. Lasenby (jl221@cam.ac.uk) is with the the Department of Engineering at the University of Cambridge, Cambridge, UK, CB2 1PZ

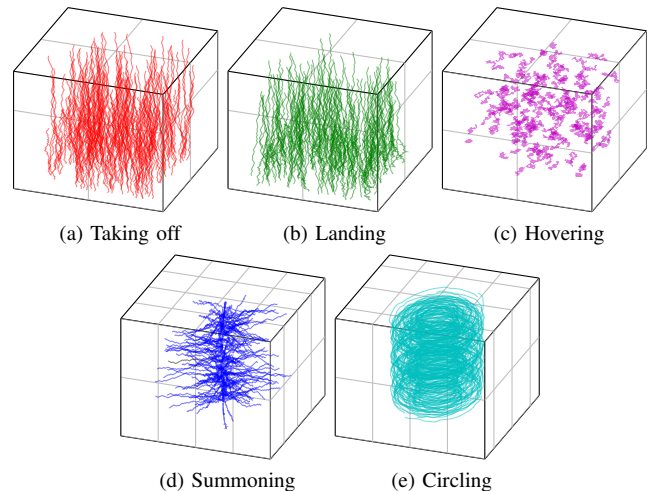


Fig. 1. Synthetic UAV behaviour trajectories are generated (1a to 1e), and combined to form a single set of labelled training data. The pantomimic gesture recognition problem addressed here requires that this training data be used to assign a human gesture to one of the five behaviour classes. For visual clarity, summoning and circling behaviours have been centred on the vertical axis. Note that the passage of time is not shown here, but this information is available as behaviour trajectories are ordered.

pantomimic gestures for each potential robot behaviour is obviously infeasible, and as a result most traditional gesture classification strategies are of little use here. However, the definition of a pantomimic gesture implies that the gesture inherently contains spatial and temporal information corresponding to that of the desired action or behaviour. If a mapping between gesture and robot behaviour could be found, with a suitable measure of correlation, this information could be used to select a likely behaviour.

This work focuses on the classification of human gestures using training data generated from recordings of available robot behaviours, in an attempt to determine the feasibility of a pantomimic gesture interface. This type of classification is particularly challenging since the training and gesture data may differ in scale, rate of occurrence and be subject to different motion constraints. To the best of our knowledge, this paper presents the first use of pantomimic gestures for human-robot interaction.

In an attempt to address some of these challenges, a sample human-robot interaction problem using an inexpensive micro unmanned aerial vehicle (UAV) has been developed. Here, users are required to control a UAV by supplying pantomimic gestures corresponding to one of five different behaviours. Gestures are recorded using a static Kinect sensor and it is assumed that users have only limited knowledge of the UAV's capabilities. This is a particularly useful illustrative example, as both noisy gesture measurements and robot behaviours are likely to be observed, resulting in a challenging use case.

Each of the UAV behaviours in the example problem can be described by a group of three dimensional time series, as visualised in Figure 1. Behaviour trajectories are described by the x , y and z positions of a UAV in Cartesian space, as they change over time. 200 samples of each behaviour were generated, with each of these behaviours re-sampled using linear interpolation to comprise 250 state measurements over the behaviour duration, corresponding to the longest sequence length in the training data. The behaviour descriptions are concatenated to form the training set for the classification problem. Our goal is to classify human hand gestures, using these behaviours as training data.

Take-off behaviours typically start at a random position on a ground plane and move upwards, while landing behaviours are the opposite, moving downwards from a random hovering position (variable height) before coming to a rest at ground level. Hover behaviours are effectively random noise, since the UAV drifts slowly about a random fixed position above ground, while summoning behaviours involve the straight line motion between two random points, at roughly the same height. Circling behaviours start at random positions above ground and move along a circular path, with a radius fixed for the duration of the behaviour, but which varies between different circling occurrences.

This work applies principal component analysis (PCA) to extract appropriate features or attributes from gesture tracks for classification. The approach groups UAV training sequences into a single matrix and linearly transforms these into a common space. The parameters used to transform the data are the principal component loadings. Appropriate 3D human gesture tracks are selected using a maximum information criterion. A candidate gesture is classified by projecting it into this space, and selecting the class with the most similar principal component loadings, after incorporating evidence relating to potential feature bias. Bayesian filtering is applied, and a Hanning window voting strategy used as a final decision rule.

The paper is organised as follows. Section II provides an overview of related work, which is followed by a description of our approach to time series feature extraction and classification in Section III. Finally, results and conclusions are provided in Sections IV and V respectively.

II. RELATED WORK

An overview of gesture recognition for the purpose of human-robot interaction is presented here. Initially, the importance of gesture in communication is discussed, motivating its use as a user interface. This is followed by a description of the state of the art in gesture recognition.

As mentioned previously, an increasing demand for service robots operating in domestic environments requires that simple and intuitive interfaces be developed. This is noted by [3], who highlight the fundamental importance of finding “natural” and easy-to-use interfaces, given that these robots are intended to interact directly with humans. Ideally, these interfaces should require little or no training and limited skills to operate.

Although relatively intuitive, users of traditional robot control interfaces such as joysticks and game-pads still require a

limited amount of training. With this in mind, it is only natural that researchers in the field of human-robot interaction look to inter-human communication mechanisms such as speech and gesture when designing intuitive user interfaces.

A number of human-robot interfaces relying on speech recognition have been developed. Unfortunately, the use of speech recognition is somewhat prohibitive, as it requires relatively controlled and quiet environments to isolate spoken commands, and restricts users to specific languages. As a result, a great deal of interest has been shown in the use of gesture for human-robot interaction.

Gesture plays a significant role in inter-human communication, sometimes acting as a primary communication language (as in the case of sign language) and often providing subsidiary and complementary information to speech. It is observed across cultures and ages, and even in individuals blind from birth [4].

A taxonomy of gesture is provided by [5], which categorises gestures into four groups; symbolic, deictic, iconic and pantomimic. Symbolic gestures are those with specific cultural meaning, and as a result have limited use in human-robot interaction. Deictic gestures are pointing gestures used to indicate objects or to convey spatial information by drawing attention to specific areas. Pointing gestures have been used previously to indicate target objects and positions to robots, and [6] have shown that a combination of head and arm pose can be used to reliably influence robot behaviour.

Iconic gestures are predefined symbols with specific meaning and most commonly used in human-robot interaction. Here, a developer will define a dictionary or codebook of gestures, each of which is mapped to a specific behaviour. This approach has been applied by [7], who used hand symbols to request various object grasping arrangements.

Pantomimic gestures are those which mimic a desired action or behaviour. A recent study on pantomimic gestures by [8] indicates that communicating with a system through gestures may be easier if an embodied approach is adopted when designing gesture vocabularies. Embodied interaction is based on the idea that human experience is formed by engaged participation in the world and that we convey meaning through this participation [9]. Intuitively then, it seems that pantomimic gestures that are an embodiment of robot actions are of most use for human-robot interaction where behaviours need to be selected, but they are rarely used, presumably due to complexity in detection and classification. As pantomimic gestures typically include more complex gesture arrangements, variability in gesturing among individuals creates greater difficulties in gesture classification.

A distinction also needs to be made between static and temporal gestures. Static gestures are typically iconic symbols held stationary for a brief period, while temporal gestures can be broken down into three phases, termed preparation, stroke and retraction, with [10] noting that most salient information about a gesture is contained in the stroke phase. As a result, if temporal gestures are to be recognised and adequately classified, key features present in the stroke phase need to be determined.

Gestures are typically described by a large amount of

multidimensional data. Reliable gesture recognition requires that salient aspects or features be extracted from this data. Two methods of dimension reduction are typically applied when selecting these salient features. The first extracts features termed most expressive, which best describe the gesture data. The second finds features termed most discriminating, or features unique to individual gestures. The latter is typically more useful when classifying gestures [11], although liable to overfitting. Before gestures can be classified, however, they need to be observed and detected using some sensing mechanism, which usually takes the form of a vision-based pose estimator.

Once user position has been located and relevant information extracted the gesture needs to be classified. If only static gestures are to be recognised, semantic features can be used to classify gestures. Triesch and Von Der Malsburg recognised 6 static hand gestures using elastic graph matching [7]. Here, hand gestures are described by labelled connected graphs with associated local image descriptors used for matching.

One approach to comparing human hand gestures is to search through a number of gestures in a training set and simply select the class most similar to the hand gesture, using an appropriate distance metric. Unfortunately, standard distance measures may fail to match trajectories sampled at different rates, or cases where one trajectory occurs faster than another.

Most temporal gesture recognition strategies remedy this by modelling the dynamics of a gesture [10]. A number of approaches have been used to model the dynamics of gestures, applying Kalman filtering, particle filters or dynamic time warping. Dynamic time warping (DTW, [12]) is a popular similarity measure, which uses dynamic programming to align two time series so as to minimise some distance measure, a computationally expensive operation. An alternative similarity measure is the longest common subsequence [13], found by searching for the longest subsequences in a time series that fall within a certain distance of one another. This allows for the comparison of time series where some parts fail to match due to noise or measurement errors.

Black and Jepsen [14] developed a probabilistic extension to dynamic time warping, applying the Condensation (Conditional Density Propagation) algorithm, for use in temporal gesture recognition. Brethes et al. [15] applied this algorithm to recognise and track a set of four distinct hand postures in a video sequence.

A number of techniques used for speech recognition have been applied to model the semantics of gesture. Finite state machines such as hidden Markov models (HMM) are frequently used to classify gestures. Using this approach, gestures are modelled as sequences of templates or model components, with varying transition probabilities. An input model is classified by determining the most likely sequence of states, given a set of observations. The Viterbi algorithm, described in detail by [16], is often used to do this. A number of gesture recognition strategies relying on HMMs have been proposed, with [17], [6] and [2] all developing HMM classifiers for human-robot interaction. In fact, HMMs have become so ubiquitous in gesture recognition that [18] note that the

majority of top performing algorithms in the recent ChaLearn gesture recognition challenge used these. Unfortunately, the behaviours in Figure 1 are not easily modelled in this way as they can differ dramatically within classes.

Recent approaches to temporal sequence matching have applied existing work in word indexing to classify time series. This has been demonstrated by [19], where 3D trajectories were broken into basic segments or building blocks, and titled with an alphabetical label. Trajectories were then indexed using these alphabet sequences, and new trajectories classified by matching sequences or words.

As an alternative approach, a number of existing machine learners largely ignore explicit temporal information and are designed to recognise classes using a set of attributes or features. Wavelet transforms have been used for feature-based time series classification [20], but ideally need to be specially crafted for individual time series, which can be time consuming. Impressive classification results were obtained by [21], using a set of meta-features that represent important events observed in each time series. More recently, a bag-of-features approach to time series classification [22] has shown promising results.

Dimension reduction techniques have also been used for feature extraction in multidimensional time series classification algorithms. Features extracted using multilinear function factorisation schemes on a single matrix containing all training data have previously been shown to provide excellent results in a number of time series classification tasks by [23]. Martin and Crowley project human motions into a weighted principal component space [24], then determine a characteristic point for the motions. Candidate time series are classified by finding motions with similar characteristic points, and using dynamic time warping to align these sequences. Principal component analysis has been applied to motion maps extracted from video sequences by [25], allowing gesture recognition from a single training example. PCA was also used by [26] to classify static hand postures in images, then combined with a finite state machine to recognise dynamic gestures. Our approach also makes use of PCA, but operates on 3D trajectories instead of video sequences.

The novelty in our approach is in the matrix stacking order we apply to the time series, which involves horizontally stacking multiple multidimensional time series from different classes, with timing information down the rows, and the use of posterior reshaping to extract feature vectors corresponding to the various dimensions in the original time series after decomposition.

Lui used higher order singular value decomposition (HO-SVD) to perform action recognition on video sequences [27]. This approach applies SVD to a number of matrices formed by unfolding a tensor in multiple ways, but differs from our work in that the decompositions are only applied to a single video sequence or third order tensor (time, horizontal and vertical pixel positions). Our decomposition uses a single decomposition on a single unfolding instance of a fourth order tensor (time, gesture category, and behaviour position) and differs primarily through the inclusion of multiple exemplars from multiple classes in the matrix that is decomposed. This

provides features that take the behaviour of other classes into account. In addition, we use a posterior reshaping of decomposed time series features, which is not present in [27].

III. METHOD

This section provides a detailed description of the proposed gesture recognition framework, covering feature extraction, gesture isolation and classification. Features are extracted from UAV behaviours, and used to classify human hand gestures, in an attempt to form a true pantomimic gesture recognition interface. The use of robot behaviours to classify human hand gestures can be viewed as a form of transductive transfer learning [28], where knowledge of a source domain is used to transfer the ability to perform a task to a different, but related, target domain. Here, knowledge of the association between behaviour labels and UAV position recordings is used to find associations between human body gestures and behaviour labels. Transfer learning is increasingly being used to avoid expensive labelling and data capture.

Our approach builds on previous work [23], which showed that matrix decomposition approaches provide equivalent performance to many state of the art classification algorithms over a wide range of time series classification problems, including character recognition on a tablet, Australian sign language classification and Japanese speaker recognition.

A. Feature extraction

We briefly describe our feature extraction approach, which relies on PCA, here. The technique can be viewed as a form of tensor factorisation for automatic feature selection, where trajectories are unfolded to form a single matrix of all behaviours. Its primary benefits are classification speed, negligible training time, and that it is almost entirely data driven, requiring no specific domain knowledge. PCA is often used for dimension reduction, and closely related to singular value decomposition. The SVD is a factorization of a matrix into the form

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*,$$

where \mathbf{U} is a unitary matrix, \mathbf{V}^* the conjugate transpose of \mathbf{V} , also a unitary matrix, and $\mathbf{\Sigma}$ is a diagonal matrix containing the singular values of \mathbf{X} , all positive and listed in decreasing order [29].

The magnitude of the singular values can be viewed as a measure of a mode's (columns of \mathbf{U}) contribution to the matrix \mathbf{X} . A low rank approximation of the matrix \mathbf{X} can be obtained by discarding the modes and basis functions (rows of \mathbf{V}) of \mathbf{X} , which correspond to singular values of small magnitude. A reduced feature set can be extracted from the data by projecting it using these basis functions: $\mathbf{Y} = \mathbf{U}_{1:m}^T \mathbf{X}$, where $\mathbf{U}_{1:m}$ is obtained by retaining the first m columns or modes of \mathbf{U} .

PCA transforms data onto a new orthogonal coordinate system so that the greatest variance of any projection of the data lies on the first coordinate (the principal component), the second largest variance along the second coordinate, and so on [30]. The principal component scores, or projections of points

along the principal components are calculated by centring each column of a matrix \mathbf{X} , then performing singular value decomposition on the shifted matrix [31]. The scores, \mathbf{P} , are given by the $\mathbf{U}\mathbf{\Sigma}$ portion of the singular value decomposition, while the loadings or coefficients are the columns of \mathbf{V} .

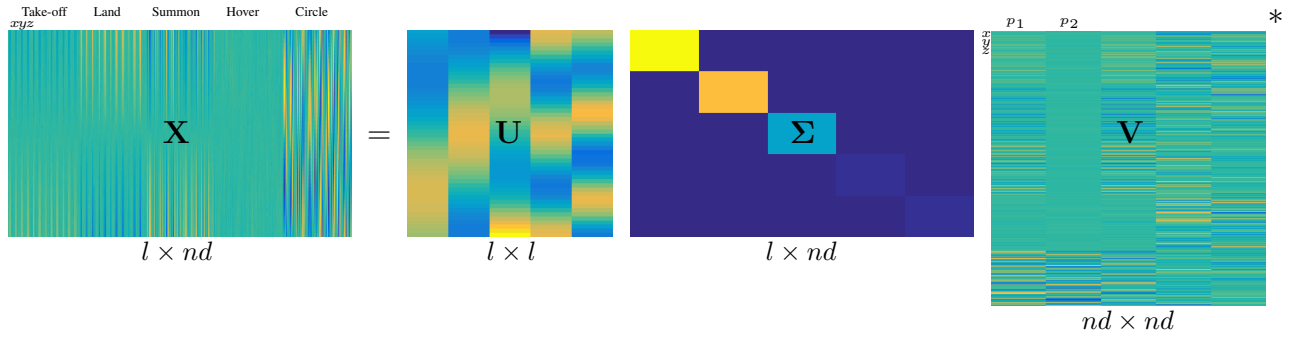
Centring occurs by shifting each column of the matrix by its mean, and is required to ensure that the first principal component lies in the direction of maximum variance. If mean centring does not occur, the maximum variance of the data could potentially lie along the mean of the data, which may not be desirable.

Dimension reduction using the SVD or PCA may not produce class features that are easily discriminable. If time series examples are normally distributed, canonical variates (CV) can be used to find a linear projection that maximises the separation between classes and hence improves classification. Canonical variates is a multi-class extension to Fischer's linear discriminant, and described in detail by [32].

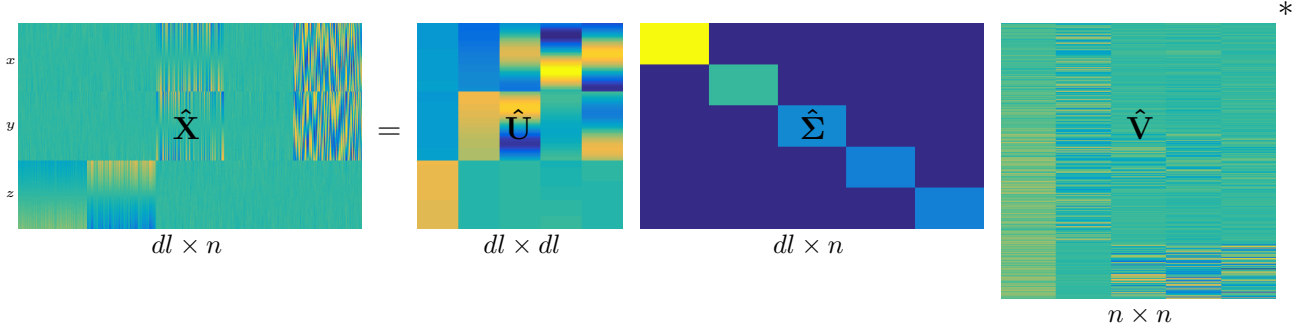
We now show how PCA can be used to extract features from time series. Let \mathbf{X} be a matrix of dimension $l \times nd$, formed by concatenating the sample trajectory matrices for all behaviour samples used for training. Here, l denotes the length of the trajectories, with d the number of dimensions in the time series, and n the number of samples. Centring the columns of this matrix and applying singular value decomposition then provides a set of $l \times l$ (\mathbf{U}), $l \times nd$ ($\mathbf{\Sigma}$) and $nd \times nd$ (\mathbf{V}) matrices as illustrated in Figure 2a. This decomposition is similar to that used directly on images by [33] for PCA-based face recognition, although we use singular value decomposition to calculate it.

It is important to highlight the differences between this stacking and that traditionally used for feature selection using PCA or canonical variates, which reshapes trajectories into 1D vectors, stacked to form a $dl \times n$ matrix, resulting in a different projection on decomposition (Figure 2b). In essence, the traditional approach treats each measurement, in every dimension, and at every time-point, as a separate feature of the time series, and then extracts the most important sub features from this set. Bundling all the dimensions into a single vector in this manner means that structural differences between time-series are potentially lost when features are extracted.


In contrast, the $nd \times nd$ loading matrix \mathbf{V} obtained using our stacking retains a spatial interpretation. Using the proposed stacking, each dimension of the multi-dimensional time series is treated as an independent trajectory and the resultant l dimensional PCA basis functions contain elements common to all these trajectory samples, regardless of dimension. This is important for spatial time series, as it allows for associations to be made between different dimensions in the original time series space. For example, the summoning trajectory in Figure 1 is very similar to a rotated take-off trajectory, but this information would be lost by applying PCA to vectorised time series combining dimensional information. By allowing this association to be made, spatial differences in the original time series are exposed in the loadings (see Figure 2a). Information for each time series sample can be recombined by grouping the loadings corresponding to each dimension in the original class example.



(a) Proposed stacking: trajectories are stacked alongside one another to form a single large matrix \mathbf{X} . A set of m, d dimensional representations (features) of the trajectories can be obtained by multiplying each trajectory by the transpose of the m -th mode or column of \mathbf{U} .



(b) Traditional stacking: trajectories are reshaped into a vector and stacked alongside one another prior to decomposition. A reduced m dimensional form of the trajectories can be obtained by multiplying trajectories by the transpose of the first m modes or columns of $\hat{\mathbf{U}}$.

Fig. 2. The intensity images in the figure (normalised key: ) show the trajectories of Figure 1 as they undergo PCA using the SVD for both the traditional and proposed stackings (Only a few modes are shown for visual clarity, although the matrix dimensions correspond to the full set of modes). Visual inspection of the loadings in \mathbf{V} for the proposed stacking show that these have retained spatial information present in the original data (eg. Column 1 of \mathbf{V} shows visible correlation with \mathbf{X}). In contrast, no such connection can be observed between the loadings of $\hat{\mathbf{V}}$ and the original data when the traditional stacking is used for feature extraction because spatial differences in the original time series are discarded, which results in potentially less separable features.

In short, PCA using the proposed stacking finds a set of projections that aligns all the input trajectories as best as possible. Projections (the PCA loadings) for a particular class of trajectories are likely to be similar, since their time series probably tended to cluster together in Cartesian space. Further, since there were likely to be spatial differences across classes in the original space, the loadings for each dimension in the time series are also likely to differ across classes, resulting in more separable features. This is illustrated in Figure 3c, where the first principal component's loadings are graphed, after reshaping to form a three dimensional feature vector. This feature distribution, coloured according to the corresponding behaviours, shows clear separation between classes.

Figures 3a and 3b show features for each behaviour in Figure 1, projected into three dimensions using canonical variates and the traditional PCA decomposition. The assumption of uni-modally distributed time series made by canonical variates was incorrect, and so it has failed to separate class features. Similarly, the standard PCA decomposition has resulted in a projection with poorly separated class features because the associated matrix stacking lacks physical or spatial meaning.

Only the first mode or column of the loading matrix is shown in Figure 3c, but in practise we could select sufficient modes to account for a large portion of the variance in the training data for classification, using the ranked singular values in Σ to automatically select the most dominant modes, or use a measure of separability to select an appropriate number of dimensions.

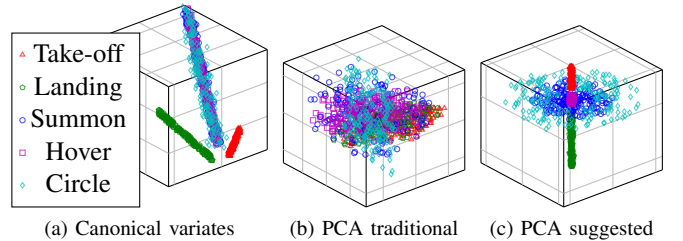


Fig. 3. A number of dimension reduction techniques could be used for feature selection, but it is clear that the separation between UAV behaviour classes is greatest for the principal component loadings associated with the suggested matrix stacking order. Figs. 3a and 3b are obtained by projecting the data into 3 dimensions, while Figure 3c is obtained by projecting the data into the first dimension of the suggested PCA, with a posterior reshaping to form an $n \times 3$ matrix.

Many attribute-based learners operate using some form of spatial or geometric segmentation, using boundaries to discriminate between classes. Intuitively then, class features that cluster together and are easily distinguished would be suited to classification using these boundary-oriented methods. A measure of the amount of overlap or interaction between features of different classes should then provide a reasonable indicator as to the quality of the features. This has been argued by [34], where the ratio of the number of points in a dataset with a nearest neighbour sharing the same class to the total number of points in the set is proposed as a measure of geometric separability. This measure can be viewed as leave-one-out analysis using a single nearest neighbour classifier,

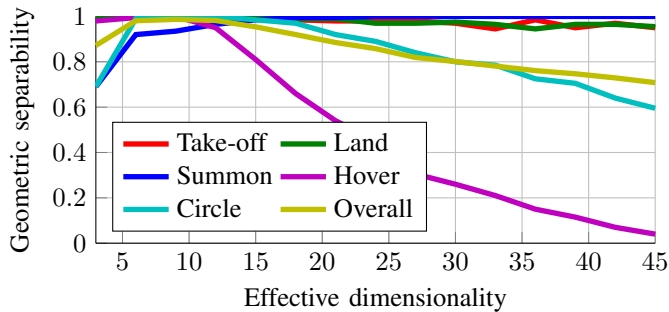


Fig. 4. The figure shows the geometric separability for each behaviour, as a function of the effective dimensionality of the proposed PCA decomposition.

which will fail more frequently if class features are similar. Figure 4 shows the geometric separability of features obtained using the proposed PCA decomposition. The nearest neighbour was determined by selecting the neighbouring sample with the minimum standardised Euclidean distance between feature vectors. Ideally, the separability of features should be as high as possible, with feature dimensionality low for faster computation.

Figure 4 suggests that the features are most separable when four modes are used, corresponding to an effective dimensionality of twelve, due to the posterior reshaping in our decomposition. However, this typically only holds when the expected feature distributions in the test set are distributed similarly to those used for training. In our transfer learning case, where UAV behaviours are used to recognise human hand gestures, this is unlikely to be true for all modes, and it is only reasonable to expect similar feature distributions in dominant modes corresponding to general information describing a behaviour. Indeed, experimental results in Section IV seem to confirm this, with only a single mode proving useful for classification.

Note that the suggested PCA decomposition struggles to separate circling and summoning behaviours when only a single mode is used, but the separation obtained is still greater than that of the various other decompositions we have attempted. We attribute this to the fact that our dataset is quite comprehensive, containing a wide variety of behaviour examples, each of which is thus likely to have a number of neighbours undergoing similar transformations in the PCA projection.

Unlike gesture recognition schemes that operate on variable length gesture sequences (DTW, HMMs, etc.), matrix factorisation approaches to time series recognition are applied to fixed length sequences, and deal with different sequence lengths in somewhat of a brute force manner, by requiring a large amount of training data covering all possible variations in sequences. This could be a problem if little training data is available, but training data is easy to generate in the case of pantomimic gesture recognition using UAV behaviours and as a result this limitation is of no concern.

B. Body part isolation

Recall that our goal is to classify gestures imitating UAV behaviours, using the UAV training data described in Figure 1. The Microsoft Xbox 360 Kinect skeleton tracker [35] is used to record gestures. The Kinect provides 3D positional data in real time, but individual gesture sequences still need to be isolated from these positions if they are to be classified. This occurs by simply buffering the positional information over an empirically selected time period. In our case we choose this time period to be about 0.5 seconds, roughly equivalent to about 15 data samples. Our experimentation showed that this sequence length was long enough to contain sufficient gesture information for classification, but short enough to allow online classification, in that the delay between gesture and recognition is not noticeable.

Selecting which human body joint to use for behaviour selection is potentially challenging, as users may decide to mimic UAV behaviours with left or right hands, or sometimes even their entire bodies. We remedy this by isolating gesture tracks for each tracked joint, and select the joint with the track conveying the greatest amount of information.

Let \mathbf{x}_{ij} be the i -th sample of the 3D position of the j -th joint. The maximum information is provided by the track with the highest approximate entropy [36], that is,

$$j = \arg \max_j \left(- \sum_{i=1}^l p(\mathbf{x}_{ij} - \boldsymbol{\mu}_j) \log_2 p(\mathbf{x}_{ij} - \boldsymbol{\mu}_j) \right),$$

where $p(\mathbf{x}_{ij} - \boldsymbol{\mu}_j)$ is the probability approximation obtained from a histogram of quantised, mean shifted positions in the track, and l the trajectory length. $\boldsymbol{\mu}_j$ is the mean position of each track. Human hands typically have far greater reach than other body parts, resulting in more disorder when they move, and so tend to be selected most often using this measure. We can also restrict the joint selection to only use the hands by simply maximising the entropy measure over a limited joint set. In this case the measure allows for gestures from both left and right handed users.

Direct comparison with the UAV training data is difficult since this is captured at a different scale to arm movements. This can be solved by scaling the x, y, z hand positions appropriately.

$$x_s = \frac{r_{\max}}{l_a} x, \quad y_s = \frac{r_{\max}}{l_a} y, \quad z_s = \frac{h_{\max}}{l_a} z.$$

Here, l_a is the average arm length calculated from the Kinect skeleton tracker, h_{\max} the maximum height the UAV operates at, and r_{\max} the maximum radius the UAV operates within.

C. Gesture classification

Once gesture sequences are isolated they can be interpolated to a fixed length (250 samples here, in line with the UAV behaviour lengths) and a classifier can be applied, after projecting the gesture trajectory, $\hat{\mathbf{X}}$, (an $l \times d$ dimensional matrix, with l the trajectory length and d the dimensionality) into the UAV feature space,

$$\mathbf{Y} = \mathbf{U}_{1:m}^T \hat{\mathbf{X}},$$

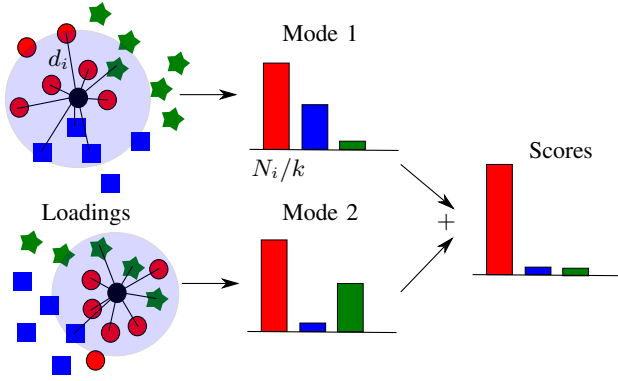


Fig. 5. The fraction of k -nearest neighbours belonging to class i provides a class distribution for each principal component mode. These distributions are normalised, and combined by summation, with optional singular value weighting. The candidate trajectory is then assigned to the class with the greatest score in the final distribution.

where $\mathbf{U}_{1:m}$ is a low rank approximation of \mathbf{U} , obtained by retaining only those m modes or columns accounting for a given portion of the variance in the training data. This results in an $m \times d$ loading matrix, with each row representing the projection into the loading space for a particular mode. In our case, we vote for a corresponding class by determining the k -nearest neighbours in standardised Euclidean space, as illustrated in Figure 5. An improved approach could use a generative model to classify observations, but the geometry of the feature distributions encountered here is difficult to model (the concentric circle feature layouts for summoning and circling behaviours for example). In addition, nearest neighbour classifiers have strong consistency results [37] and are typically good indicators of the performance of more intricate classification schemes.

In applications where training and testing data varies significantly, modes of less importance may not be similar and so intuitively should carry less weight in a classification scheme. As a result we prefer to average the contributions of each mode after weighting these by the singular values corresponding to each mode, to allow for greater emphasis on more descriptive behaviour modes, rather than performing a single md dimensional nearest neighbour search.

The distance measure used for the nearest neighbour class voting should also be chosen carefully. Figure 3c showed that classes occupy different regions in the principal component loading space. As a result, the distance summation is unlikely to correctly classify classes if the training data is spread fairly widely in some dimensions, but clustered in others. This can be remedied through the use of a standardised Euclidean distance measure. Here, the difference between each coordinate in the candidate class point and the training data is scaled by the standard deviation along the relevant dimension in the training data, and the Euclidean distance of these scaled differences used. Given the concentric circle feature distribution, an improved distance metric could be applied in polar coordinates, but this requires prior knowledge of the feature distribution, and we prefer to avoid any requirement for prior knowledge in order to retain a generic classification process.

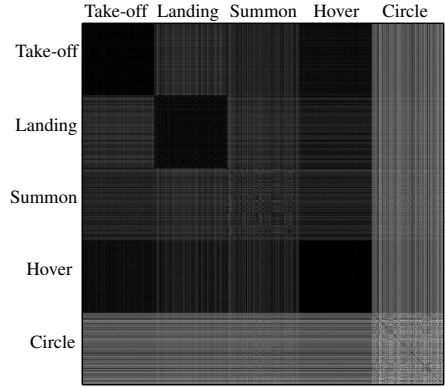


Fig. 6. A distance matrix showing the distance between features in the UAV behaviour training for the PCA features (mode 1). Ideally, features should be as close to one another as possible (dark) for a single behaviour, but farther away from other behaviours (light).

1) *Feature bias compensation*: The features selected using the PCA projection may be better for classifying certain behaviours than others, because of the geometry of the feature space. This can be observed in Figure 6, which shows an intensity map of the standardised Euclidean distances between the features of each trajectory in the UAV behaviour training set for the first mode. Here, a hover behaviour is more easily distinguished using these features than a circle behaviour. In an ideal case, with all behaviours classified fairly, the intensity map should contain five dark blocks on the diagonal, and light squares elsewhere.

We now show how this bias can be compensated for using a Bayesian update framework. Let \mathbf{f}_m be a measured feature vector extracted from a gesture (obtained from \mathbf{Y} after posterior reshaping) and B_i the i -th UAV behaviour, selected using training data \mathbf{D} . The distribution obtained from the classifier can be written as

$$P(B_i|\mathbf{f}_m, \mathbf{D}) = \frac{N_i}{k},$$

where N_i refers to the number of neighbours belonging to the i -th class and k the total number of neighbours used.

Using Bayes' rule, and given the independence between the training data and observed feature, $p(\mathbf{f}_m|\mathbf{D}) = p(\mathbf{f}_m)$, this distribution can be written as

$$\begin{aligned} P(B_i|\mathbf{f}_m, \mathbf{D}) &= \frac{P(\mathbf{f}_m, \mathbf{D}|B_i)P(B_i)}{P(\mathbf{f}_m, \mathbf{D})} \\ &= \frac{P(\mathbf{f}_m|B_i)P(B_i)}{P(\mathbf{f}_m)} \frac{P(\mathbf{D}|B_i)}{P(\mathbf{D})} \\ &= P(B_i|\mathbf{f}_m) \frac{P(B_i|\mathbf{D})}{P(B_i)}. \end{aligned}$$

Solving for the posterior probability, $P(B_i|\mathbf{f}_m)$, we obtain

$$P(B_i|\mathbf{f}_m) = \frac{P(B_i|\mathbf{f}_m, \mathbf{D})P(B_i)}{P(B_i|\mathbf{D})}.$$

Assuming that all behaviours are equally likely to be selected, the prior, $P(B_i)$, is just a constant scaling factor. The evidence term, $P(B_i|\mathbf{D})$, allows us to compensate for any bias incurred due to the geometry of features in the training set. We estimate

this by classifying each feature in the dataset, using leave-one-out analysis, and determining the frequency with which each class is selected:

$$P(B_i|\mathbf{D}) \approx \frac{N_i^c}{N},$$

where N_i^c refers to the number of times the i -th class was selected and N the total number of features in the training set. Classification decisions made using the posterior probability, $P(B_i|\mathbf{f}_m)$, which incorporates this evidence, compensate for potential behaviour bias.

2) *Behaviour selection*: Thus far, we have largely ignored the sequential nature of the gesture recognition task. In practice, the gesture recognition process would operate online using a sliding window and be repeated with each new observation, so it makes sense to incorporate previous posterior densities into the decision process. Assuming gestures are Markovian, let $P(B_{i_t}|B_{j_{t-1}})$ be the probability of transitioning between behaviours j and i over time step t . Our target density, $P(B_{i_t}|\mathbf{f}_{m_{1:t}})$, is the probability of a specific behaviour occurring at time t given a history of feature measurements $\mathbf{f}_{m_{1:t}}$.

This is easily determined using recursive Bayesian estimation,

$$P(B_{i_t}|\mathbf{f}_{m_{1:t-1}}) = \sum_{j=1}^{N_b} P(B_{i_t}|B_{j_{t-1}})P(B_{j_{t-1}}|\mathbf{f}_{m_{1:t-1}})$$

$$P(B_{i_t}|\mathbf{f}_{m_{1:t}}) = \eta P(B_{i_t}|\mathbf{f}_{m_t})P(B_{i_t}|\mathbf{f}_{m_{1:t-1}}),$$

where η is a normalising constant (Here we have used the fact that $P(B_{i_t})$ is the same for all behaviours and $P(\mathbf{f}_{m_t})$ remains constant over i and folded these terms into the normalisation constant) and N_b the number of behaviours. Finally, given this target distribution, we select the behaviour with the largest probability,

$$B_t = \arg \max_{i_t} P(B_{i_t}|\mathbf{f}_{m_{1:t}}).$$

Unfortunately, performing gesture recognition online is particularly challenging, due to misclassification seen during the preparation and retraction phases of a gesture. For example, a landing gesture inherently contains a take-off gesture as preparation, and a take-off gesture in retraction looks like a landing gesture. As a result, it is particularly important that the start and end phases of a gesture be determined, a process known as gesture spotting. DTW has been extended to isolate patterns from continuous real world data [38], while the crossing points of behaviour probability distributions were used for spotting by [39] and [40], with the latter using conditional random fields to discriminate between vocabulary gestures and non-sign patterns prior to gesture recognition. Rudimentary techniques accomplishing spotting using gesture speed and the distance to a standing state can also be found in the work of [41] and [42], but an in depth treatment of this problem is beyond the scope of this work. Henceforth, we assume that the start and end points of a gesture sequence are known, and focus on classifying sequences as a whole, in order to facilitate comparisons.

Given this information, and a sequence of classification decisions, we can vote on a most likely behaviour for a gesture.

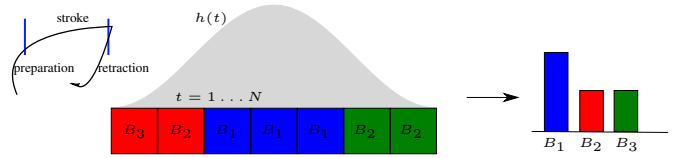


Fig. 7. The classification decisions obtained when sliding the window along the input gesture sequence are weighted by a Hanning window, then combined to vote for the most likely behaviour describing the entire gesture sequence. As an example, consider the summoning gesture depicted in the figure. As the sliding window moves along the sequence, a sequence of take-off classifications would be made in the preparation phase, followed by summoning decisions during the gesture stroke, before finally concluding with a set of landing decisions in retraction. The Hanning window weighting adds emphasis to the stroke phase of the gesture when the final classification decision is made.

A direct majority vote was used by [39], but this may not be ideal. Recall that gestures can typically be divided into preparation, stroke and retraction phases. Intuitively then, classification decisions made during the middle of a gesture should correspond to the stroke phase and carry more importance than decisions made at the beginning and end of gestures.

This intuition can be applied by voting for a gesture class by weighting each classification decision using a Hanning window,

$$B = \arg \max_i \left(\sum_t g(i, B_t) h(t) \right), \text{ where}$$

$$h(t) = 0.5 \left(1 - \cos \left(2\pi \frac{t}{N} \right) \right), \quad 0 \leq t \leq N, \text{ and}$$

$$g(i, B_t) = \begin{cases} 1 & \text{if } i = B_t, \\ 0 & \text{otherwise.} \end{cases}$$

This process is illustrated more clearly in Figure 7.

IV. PANTOMIMIC GESTURE RECOGNITION

This section provides experimental results for the pantomimic gesture recognition problem. Results were obtained by determining the classification accuracy using a dataset of 237 gesture sequences, performed by 5 different participants. Data was obtained by requesting that each participant perform 50 gestures, each corresponding to one of the 5 UAV behaviours, with these chosen at random. Kinect tracking failures were removed, reducing the dataset from 250 to 237 gestures, and the start and end points of gestures were manually labelled. No information other than requested behaviours was provided, so that the participants would perform gestures that they found most intuitive and felt best represented the requested behaviour, in line with our goal of examining the feasibility of pantomimic gesture recognition. Figure 8 shows the aligned test and training data.

Study participants were shown a photo of the micro UAV that the gestures they were performing was intended to control, and told that the UAV could take-off and land like a helicopter, hover in one place, fly towards a person (summon) and fly in circles around a person. Participants were told that their task was to perform an appropriate gesture to request one of these actions when instructed to do so by testing software. The testing software showed users a stick representation of

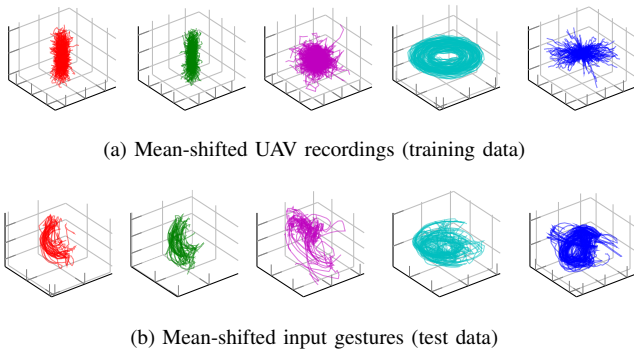


Fig. 8. The aligned test data used for experiments is shown in Figure 8b. Differences between the test set and the UAV training data (Figure 8a) primarily result from preparation and retraction stages of gesture sequences in the human gestures, the effects of which are limited through the use of Hanning window weighting when performing classification.

TABLE I
JOINT SELECTION RATE FOR VARIOUS BEHAVIOURS

Joint	Take-off	Land	Summon	Hover	Circle
Head	0.33	0.23	0.16	0.20	0.12
Left hand	0.11	0.12	0.18	0.06	0.21
Right hand	0.30	0.28	0.48	0.30	0.28
Left elbow	0.06	0.04	0.02	0.01	0.03
Right elbow	0.11	0.26	0.14	0.33	0.28
Left shoulder	0.05	0.05	0.01	0.08	0.06
Right shoulder	0.01	0.01	0.00	0.00	0.01
Torso	0.02	0.01	0.00	0.01	0.00

themselves along with randomly generated behaviour requests, allowing 30 seconds for each gesture to be completed. The participants did not see examples of the UAV performing these actions, and were left with complete freedom to interpret the verbal descriptions of the behaviours in their own manner. As a result, a diverse and varied set of gestures were obtained. For example, one participant chose to represent a hover behaviour by performing small flapping motions with outstretched arms, another held arms outstretched in a gliding motion, while others simply held out stationary hands in a stopping motion.

A. Joint selection

Table I shows the rates at which upper body joints were selected using the maximum information criterion for the various UAV behaviours in the sample problem. The majority of our test subjects were right-handed, which explains the frequent selection of the right hand and elbow as the primary joint. When performing take-off and landing gestures, participants had a tendency to use their entire bodies to mimic these behaviours, starting with rapid head and shoulder movements, which led into an arm gesture. As a result, the head is selected as the preferred joint with relatively high frequency. Circling behaviours tended to be performed using both hands out stretched, and so a larger set of joints are selected for this behaviour. A similar pattern is seen for hovering behaviours, although this can probably be attributed to the fact that hover gestures tended to contain little motion, resulting in less of a difference in entropy measures for the various joints.

TABLE II
CLASSIFICATION ACCURACIES (%) FOR VARIOUS BEHAVIOURS

Method	Take-off	Land	Summon	Hover	Circle	Overall
UAV PCA+B	71.05	61.70	64.18	91.11	70.00	70.89
UAV PCA	71.05	61.70	65.67	91.11	62.50	70.04
UAV CV+B	71.05	63.83	38.81	91.11	20.00	55.70
UAV CV	71.05	63.83	38.81	91.11	17.50	55.27
UAV DTW	68.42	63.83	68.66	71.11	2.50	56.96
UAV HMM	0.00	0.00	88.06	55.56	2.50	35.86
User PCA	65.79	46.81	19.40	31.11	82.50	45.15
User CV	76.32	61.70	5.97	64.44	22.50	42.19
User DTW	65.79	59.57	0.00	82.22	0.00	37.97
User HMM	5.26	65.96	5.97	0.00	60.00	25.74

B. Classification accuracies

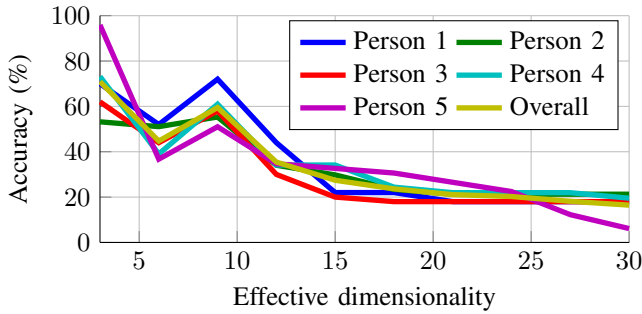
Table II shows the results of the gesture sequence recognition using the suggested PCA classifier, dynamic time warping and a nearest neighbour canonical variates classifier. Note that no confidence intervals are provided, as there is no cross validation or averaging of results here, our training and test datasets are completely independent.

The PCA-based classifier is compared to one operating on canonical variate features, a nearest neighbour DTW classifier searching through mean-shifted training data and a maximum likelihood classifier using HMMs. The hidden Markov models were trained for each class using 12 hidden states, operating on data binned into 8 states, with states selected by performing k-means clustering on mean-shifted training data. Data was mean-shifted to allow fair comparison, since classification on the original data is extremely difficult as it exhibits a great deal of variability across training samples.

Results obtained by training the classifiers using human hand gestures as training data (User) are also provided for comparison. Here, each participant's gestures were classified by using recordings of the other participants in the test dataset as training data, a more traditional gesture recognition approach. Notation +B refers to the use of the proposed Bayes' evidence term.

It is clear that the pantomimic approaches using the UAV behaviour recordings as training information (UAV) provided superior performance when compared to gesture recognition schemes trained using human hand gestures. We attribute this to the fact that the UAV behaviours contain information more likely to be present in user specific gestures, while user specific gestures are so variable that it is difficult to find similarities between participants when attempting to classify using this data. At first glance, the fact that one person's gestures are more similar to UAV behaviour recordings than they are to those of another participant may seem counter-intuitive, but readers should bear in mind that the users are attempting to mimic the UAV behaviours, and not the gesturing styles of one another. Results obtained when using human gestures as training data could potentially be improved through the inclusion of additional training data, but this is difficult to collect, while UAV training data is easily generated.

The classification accuracies for individual behaviours are useful to compare the effects of the Bayesian evidence term. As expected, the inclusion of the Bayesian evidence term boosted the circling classification rates, providing a fairer



(a) Accuracy of participants: UAV data

Fig. 9. The accuracy obtained for various participants and behaviours decreases with the effective dimensionality of the proposed PCA decomposition on UAV training data.

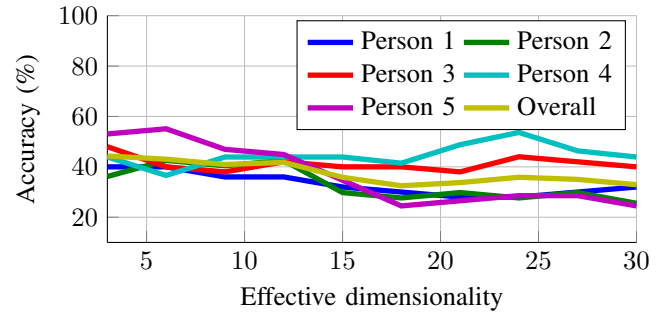
classifier. The PCA features appear best suited to classification across all behaviours. Dynamic time warping performed well for most gestures, but was unable to detect circling motions adequately and proved computationally expensive.

The poor performance of the hidden Markov models is to be expected, due to the large amount of variability in the summoning and circling behaviours. In attempting to model this variability, the HMMs trained using this data are able to explain additional behaviours, resulting in a large number of misclassifications. This could be avoided by clustering the behaviours in these classes and training multiple models for each class, but this is hardly ideal, and the additional training time complexity makes it infeasible.

Referring back to Figure 6, it seems that the take-off and landing behaviours should be relatively easily classified using the pantomimic data, but results obtained do not seem to corroborate this, with lower than expected classification accuracies here. This can be explained by the mechanics of the human arm. A UAV take-off or landing consists of almost entirely vertical motion, but a human hand mimicking this moves in a circular arc, pivoting about the shoulder, and as a result contains a larger lateral motion component, which shifts the projected feature away from the training features. This effect occurs across all behaviours, but is most noticeable here. Despite this, the results obtained here are promising, given the variations observed across participants for various behaviours.

Figure 9 shows the classification accuracy obtained with the PCA decomposition obtained from UAV data, as a function of the effective dimensionality. In contrast to the separability curves of Figure 4, which suggest that four modes should be used for classification, it is clear that the best performance is obtained when only a single mode is used. As expected, it appears that only the features of the most dominant PCA mode, which corresponds to general information describing a UAV behaviour, follow similar distributions to those features selected when applying this decomposition to human gestures.

Figure 10 shows the classification accuracy obtained with the PCA decomposition using user gesture recordings, as a function of the effective dimensionality. In general, poor performance is exhibited across all modes, and it is clear that PCA does not produce particularly separable behaviours when



(a) Accuracy of participants: user data

Fig. 10. Poor classification accuracy is obtained across participants and behaviours regardless of the effective dimensionality of the proposed PCA decomposition when it is applied to user training data.

TABLE III
ACCURACY AS A FUNCTION OF JOINT SUBSET

Right hand 74.68%	Left hand 46.84%	Both hands 73.00%	Head + hands 72.15%
Head + arms 72.57%	Full body 70.89%		

performed on human gesture recordings.

Ideally, we require features from the human gestures that are both descriptive and discriminative. The latter is required to aid in classification, while the former implies that the features should be observed across participants consistently. The human gesture data we have obtained, although limited, simply appears too variable to select both consistent and discriminative features with the methods we have attempted. In contrast, the use of UAV training data, which is cleaner, cheap and easy to generate, allows us to leverage prior knowledge about the general structure of these gestures, because they are pantomimic. Selecting PCA modes describing a large portion of the variance allows us to maintain this underlying information, and we are fortunate that the decomposition also results in fairly separable classes.

It should be noted that the use of a fixed length sliding window when classifying is only applicable if input gestures are all roughly the same length, and that an adaptive window size would be required for gesture sequences with highly variable lengths. Incorporating an adaptive window size through the inclusion of gesture spotting prior to gesture classification would remedy this problem.

While the accuracies provided above are not yet good enough for practical application, the results obtained still show the promise of using robot behaviours to train gesture recognition systems. Table III shows that an immediate increase in accuracy can be obtained by simply constraining the subset of joints used by the maximum information joint selection process. Additional increases in accuracy could be obtained by requiring that users only perform gestures using their hands, which would simplify the classification task greatly.

Table IV shows the computational complexity of the k -nearest-neighbour classifiers for comparison with dynamic time warping and the forward algorithm used to determine HMM likelihoods. Here, n denotes the number of samples in

TABLE IV
COMPUTATIONAL COMPLEXITIES AND RUNTIMES

	Complexity	Runtime
Dynamic time warping	$O(ndlw)$	1.759 s
PCA KNN	$O(mdl + n(md + \log n))$	0.001 s
Canonical variates KNN	$O(lp + n(p + \log n))$	0.001 s
HMM	$O(T^2l)$	0.040 s

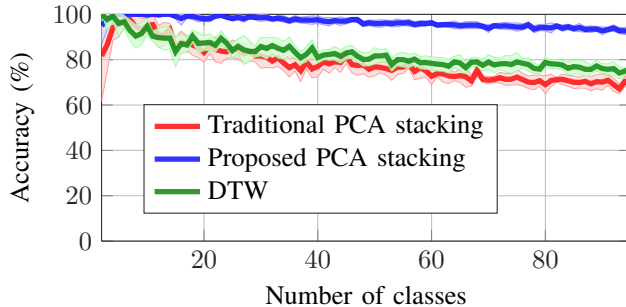


Fig. 11. The figure shows that the proposed matrix stacking produces PCA features with excellent classification scalability on the Auslan dataset, particularly when compared to dynamic time warping and classification with PCA features obtained using the traditional stacking.

the training set, d the number of dimensions describing each trajectory and l the length of each trajectory. The region used to constrain the dynamic time warping search is represented by w . The numbers of modes and nearest neighbours used by the PCA classifier using the suggested stacking are denoted by m and k respectively, while p refers to the number of dimensions used in the canonical variate projection. T refers to the number of states in the HMM. The canonical variates and PCA classifiers require far less computation as they operate on a substantially reduced feature set. Average run-times for each iteration of the pantomimic gesture recognition using UAV training data are also provided.

C. Australian sign language recognition

The pantomimic gesture recognition experiments of the previous section were conducted on only a small number of behaviour classes. The potential scalability of the proposed time series classification approach to problems with a larger number of classes is evaluated by computing the average classification accuracy obtained using an Australian sign language dataset [21]. Auslan is a high dimensional dataset of sign language comprising 95 gestures with 27 instances of each sign. Individual signs are described by a 22 dimensional time series matrix, consisting of the signer's hand positions, orientations and finger motions.

Figure 11 shows the average classification accuracy obtained when 9-fold cross validation is performed on an increasing number of signs from the dataset. It is clear that the proposed matrix stacking and time series classification approach (using a single mode, equivalent to 22 features) is far more scalable than traditional PCA features (22 were used for a fair comparison) and dynamic time warping.

Table V shows the average accuracy obtained on the full Auslan database for a variety of methods. Here, 5-fold cross

TABLE V
AUSLAN AVERAGE CLASSIFICATION ACCURACY

Approach	Accuracy (%)
Metafeatures with voting [21]	97.9 ± 0.2
Proposed PCA stacking ($k = 4, md = 66$)	95.1 ± 0.8
CV + KNN ($k = 10, p = 30$)	92.2 ± 1.2
HMM [21]	87.1 ± 0.6
Traditional PCA stacking ($k = 10, d = 66$)	78.9 ± 1.8
DTW	73.7 ± 1.7

validation was used to facilitate comparison with results reported in the literature, and parameters were tuned to provide the best results for each method. The table shows that classification with features obtained using the proposed stacking provides results comparable with hand selected features and a voted combination of back-end learners.

It should be noted that this is an easier classification task than the pantomimic gesture recognition one, as input and training data are from identical domains. Although worse results would be expected if an increasing number of behaviours were included in the pantomimic case, the experiment on the Auslan data does show that the proposed decomposition approach has the potential to scale nicely.

The pantomimic gesture recognition scenario introduced here is not intended for tele-operation. Instead, the use case is one where high level autonomy is already built into a platform, and only a subset of behaviours need to be selected, with all other operation handled autonomously. As a result, it is not necessary to map a gesture onto every possible manoeuvre as the number of behaviours available for selection is likely to remain low. Behaviours encountered are likely to be problem specific, so it is difficult to provide an estimate of the exact number of gestures the proposed pantomimic approach would extend to. However, since the proposed decomposition exposes spatial differences present in the original task space, our approach should scale as long as there are spatial differences in the behaviour recordings.

V. CONCLUSIONS

This paper has applied an extremely fast, yet simple, classification method based on principal component loadings. The approach relies on a decomposition that preserves spatial orientation aspects of multidimensional time series and results in a set of features with apparent class separability. A Bayesian update framework that compensates for potential classification bias has been introduced. A voting window was applied to improve decisions when an entire gesture sequence is available, but this is of limited use in an online case, where gestures need to be recognised as they are occurring. A clearer idea of the algorithm's online performance can be obtained by viewing the video accompanying this paper, which highlights issues resulting from misclassification during the transition phases of gestures. The addition of gesture spotting is required to differentiate between stroke and transition phases if this is to be remedied.

The applicability of this classification framework to a pantomimic gesture recognition problem has been discussed. Here, UAV behaviour descriptions have been used to train a

PCA loading classifier, and Kinect body tracking to record gesture sequences. This classification problem is extremely difficult, as gestures and behaviour recordings may appear quite different. We have proposed the use of a maximum entropy measure to select the joint conveying the most information, which allows us to deal with both left and right-handed users. Results showed that the technique can be used to determine desired behaviours when users intuitively mimic these behaviours using their bodies. We have argued that a pantomimic gesture recognition system is potentially more intuitive than one using iconic gestures as users have free choice over their gestures, and have provided a mechanism by which human hand gestures can be mapped to robot trajectories.

It is difficult to compare pantomimic and iconic gestures directly, as they correspond to different use cases. Pantomimic gestures would be useful in cases where no domain knowledge is available, for example in assistant robots deployed in public areas, while iconic gestures are certain to be more suitable when only a limited set of pre-determined gestures need to be recognised, if only because they are easier to detect. A comprehensive study on the utility of iconic and pantomimic gestures is left for future work.

Encouraging results showed that the use of UAV behaviour training data provided more robustness to a larger variety of gesture types than a gesture recognition scheme trained using human hand gestures. The latter should provide better classification results if used for specific individuals and gestures, but is clearly unsuited to less constrained problems. The results provided are promising, as the classification task addressed here is quite challenging, since participants were allowed complete freedom over their choice of gestures, which tended to be extremely varied.

REFERENCES

- [1] M. Sigalas, H. Baltzakis, and P. Trahanias, "Gesture recognition based on arm tracking for human-robot interaction," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conf. on*, October 2010, pp. 5424–5429.
- [2] H. Yang, A. Park, and S. Lee, "Gesture spotting and recognition for human-robot interaction," *Robotics, IEEE Trans. on*, vol. 23, no. 2, pp. 256–270, April 2007.
- [3] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Autonomous Robots*, vol. 9, pp. 151–173, 2000.
- [4] S. Goldin-Meadow, "The role of gesture in communication and thinking," *Trends in Cognitive Sciences*, vol. 3, no. 11, pp. 419–429, 1999.
- [5] B. Rime and L. Schiaratura, Eds., *Fundamentals of Nonverbal Behavior*. Cambridge, U.K.: Cambridge Univ. Press, 1991.
- [6] K. Nickel and R. Stiefelhagen, "Visual recognition of pointing gestures for human-robot interaction," *Image and Vision Computing*, vol. 25, no. 12, pp. 1875–1884, 2007.
- [7] J. Triesch and C. Von Der Malsburg, "A gesture interface for human-robot-interaction," in *Automatic Face and Gesture Recognition, 1998. Proc. Third IEEE International Conf. on*, April 1998, pp. 546–551.
- [8] S. A. Grandhi, G. Joue, and I. Mittelberg, "Understanding naturalness and intuitiveness in gesture production: Insights for touchless gestural interfaces," in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 821–824.
- [9] P. Dourish, "Seeking a foundation for context-aware computing," *Human-Computer Interaction*, vol. 16, no. 2-4, pp. 229–241, 2001.
- [10] Y. Wu and T. Huang, "Vision-based gesture recognition: A review," in *Gesture-Based Communication in Human-Computer Interaction*, ser. Lecture Notes in Computer Science, A. Braffort, R. Gherbi, S. Gibet, D. Teil, and J. Richardson, Eds. Springer Berlin / Heidelberg, 1999, vol. 1739, pp. 103–115.
- [11] Y. Cui, D. L. Swets, and J. J. Weng, "Learning-based hand sign recognition using SHOSLIF-M," in *Proc. 5th Int'l Conf. Computer Vision*, 1995, pp. 631–636.
- [12] H. Sakoe, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 43–49, 1978.
- [13] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Data Engineering, 2002. Proc. 18th International Conf. on*, 2002, pp. 673–684.
- [14] M. J. Black and A. D. Jepson, "Recognizing temporal trajectories using the condensation algorithm," in *Face and Gesture Recognition*, 1998, pp. 16–21.
- [15] L. Brethes, P. Menezes, F. Lerasle, and J. Hayet, "Face tracking and hand gesture recognition for human-robot interaction," in *Robotics and Automation, 2004. Proc. ICRA '04. 2004 IEEE International Conference on*, vol. 2, May 2004, pp. 1901–1906.
- [16] G. D. Forney Jr., "The Viterbi algorithm," *Proc. of the IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.
- [17] S. Lee, "Automatic gesture recognition for intelligent human-robot interaction," in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conf. on*, April 2006, pp. 645–650.
- [18] I. Guyon, V. Athitsos, P. Jangyodsuk, H. J. Escalante, and B. Hamner, "Results and analysis of the ChaLearn gesture challenge 2012," in *Advances in Depth Image Analysis and Applications*, ser. Lecture Notes in Computer Science, X. Jiang, O. R. P. Bellon, D. Goldgof, and T. Oishi, Eds. Springer Berlin Heidelberg, 2013, vol. 7854, pp. 186–204.
- [19] J. Yang, Y. F. Li, and K. Wang, "Invariant trajectory indexing for real time 3D motion recognition," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conf. on*, Sept. 2011, pp. 3440–3445.
- [20] H. Zhang, T. Ho, M. Lin, and X. Liang, "Feature extraction for time series classification using discriminating wavelet coefficients," in *Advances in Neural Networks - ISNN 2006*, ser. Lecture Notes in Computer Science, J. Wang, Z. Yi, J. M. Zurada, B. Lu, and H. Yin, Eds. Springer Berlin Heidelberg, 2006, vol. 3971, pp. 1394–1399.
- [21] M. W. Kadous and C. Sammut, "Classification of multivariate time series and structured data using constructive induction," *Mach. Learn.*, vol. 58, no. 2-3, pp. 179–216, Feb. 2005.
- [22] M. G. Baydogan, G. Runger, and E. Tuv, "A bag-of-features framework to classify time series," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 11, pp. 2796–2802, 2013.
- [23] M. Burke and J. Lasenby, "Multilinear function factorisation for time series feature extraction," in *Digital Signal Processing (DSP), 18th International Conf. on*, June 2013.
- [24] K. Forbes and E. Fiume, "An efficient search algorithm for motion data using weighted PCA," in *Symposium on Computer Animation*, D. Terzopoulos, V. B. Zordan, K. Anjyo, and P. Faloutsos, Eds. ACM, 2005, pp. 67–76.
- [25] H. J. Escalante, I. Guyon, V. Athitsos, P. Jangyodsuk, and J. Wan, "Principal motion components for gesture recognition using a single example," *CoRR*, vol. abs/1310.4822, 2013.
- [26] J. Martin and J. L. Crowley, "An appearance-based approach to gesture-recognition," in *Image Analysis and Processing*, ser. Lecture Notes in Computer Science, A. Bimbo, Ed. Springer Berlin Heidelberg, 1997, vol. 1311, pp. 340–347.
- [27] Y. M. Lui, "Human gesture recognition on product manifolds," *Journal of Machine Learning Research*, vol. 12, pp. 3297–3321, November 2012.
- [28] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [29] G. W. Stewart, "On the early history of the singular value decomposition," *SIAM Review*, vol. 35, no. 4, pp. 551–566, 1993.
- [30] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer, Oct. 2002.
- [31] M. Wall, A. Rechtsteiner, and L. Rocha, "Singular value decomposition and principal component analysis," in *A Practical Approach to Microarray Data Analysis*, D. P. Berrar, W. Dubitzky, and M. Granzow, Eds. Springer US, 2003, pp. 91–109.
- [32] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.
- [33] J. Yang, D. Zhang, A. F. Frangi, and J. Y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recogni-

tion,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 1, pp. 131–137, Jan. 2004.

- [34] C. Thornton, “Separability is a learner’s best friend,” in *Proc. of the Fourth Neural Computation and Psychology Workshop: Connectionist Representations*. Springer-Verlag, 1997, pp. 40–47.
- [35] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images,” in *Computer Vision and Pattern Recognition*, June 2011.
- [36] R. Moddemeijer, “On estimation of entropy and mutual information of continuous distributions,” *Signal Processing*, vol. 16, no. 3, pp. 233–246, 1989.
- [37] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.
- [38] R. Oka, “Spotting method for classification of real world data,” *The Computer Journal*, vol. 41, no. 8, pp. 559–565, 1998.
- [39] J. Song and D. Kim, “Simultaneous gesture segmentation and recognition based on forward spotting accumulative HMMs,” in *Pattern Recognition, 2006. ICPR 2006. 18th International Conf. on*, vol. 1, 2006, pp. 1231–1235.
- [40] H.-D. Yang, S. Sclaroff, and S.-W. Lee, “Sign language spotting with a threshold model based on conditional random fields,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 7, pp. 1264–1277, July 2009.
- [41] A. D. Wilson, A. F. Bobick, and J. Cassell, “Recovering the temporal structure of natural gesture,” in *Automatic Face and Gesture Recognition, 1996., Proc. of the Second International Conf. on*, 1996, pp. 66–71.
- [42] R. Bryll, F. Quek, and A. Esposito, “Automatic hand hold detection in natural conversation,” in *IEEE Workshop on Cues in Communication*, December 2001.



Michael Burke is a PhD graduand in the Signal Processing Group, Department of Engineering, University of Cambridge, Cambridge and a researcher at Mobile Intelligent Autonomous Systems, Modelling and Digital Sciences, Council for Scientific and Industrial Research, South Africa. His research interests are in computer vision and pattern recognition, with applications in mobile robotics and motion capture.



Joan Lasenby is currently a Reader in the Signal Processing Group, Department of Engineering, University of Cambridge, and a Fellow of Trinity College, Cambridge. Her research interests include: computer vision and 3D reconstruction; engineering applications of Geometric Algebra; medical signal and image processing.