Improvements in Feature Vector Selection and Parameter Optimisation for Continuous Gesture Recognition

Ashley Gritzman, Tomislav Batev, Adam Pantanowitz
Biomedical Engineering Research Group, School of Electrical and Information Engineering
University of the Witwatersrand
Johannesburg, South Africa
a.gritzman@eie.wits.ac.za, toma@batev.net, adam.pantanowitz@wits.ac.za

***Abstract*—Gesture recognition has attracted significant interest due to diverse potential applications, including: hand writing recognition, robot control and human-computer interfaces. This paper identifies and addresses three shortcomings in current approaches to feature vector selection and parameter optimisation for continuous gesture recognition. First, in selecting the final feature vector, researchers typically analyse only a small subset of possible feature combinations; however, the limited subset is likely to omit the optimum feature vector. Second, selection of the final feature vector is based on performance in isolated recognition; however, the final feature vector may not perform adequately in continuous recognition. No protocol currently exists to evaluate and select the final feature vector in continuous recognition mode, thus a novel scoring system is developed. Finally, optimisation of the number of states in the Hidden Markov Models (HMMs) and the number of clusters (k-means clustering) is performed independently, ignoring any possible interdependency. To investigate and address these shortcomings, a gesture recognition system geared towards sign language interpretation is designed. The system is tested on a 9-word gesture vocabulary, and subsequent analysis confirms the above conjectures: first, the optimum feature vector cannot be intuitively predicted and must be determined through rigorous analysis; second, selecting the final feature vector in continuous mode improved the accuracy score by 5.85 % and the perfect sentence recognition by 47.2 %; finally, optimising the number of states and number of clusters simultaneously improved the accuracy score by 3.0 % and the perfect sentence recognition by 11.1 %.**

Keywords—gesture recognition; feature vector selection; Hidden Markov Models; parameter optimisation; Kinect Sensor

I. INTRODUCTION

Gesture recognition is the process whereby a meaningful spatiotemporal pattern made by the user is identified by the receiver [1, 2]. Gesture recognition systems have a number of diverse potential applications including: hand writing recognition; robot control; sign language interpretation; virtual reality gaming; and, advanced human computer interfaces [1].

There are two categories of gesture recognition *viz.* isolated recognition and continuous recognition [2]. *Isolated recognition* is based on the assumption that each gesture can be individually extracted [3]. *Continuous recognition* has the additional challenge of recognising a gesture from a continuous motion [3]. Continuous recognition is desirable to facilitate natural gesturing in real-time. The two main challenges of continuous gesture recognition are spatiotemporal variability and segmentation ambiguity [4]. *Spatiotemporal variability* refers to the dynamic variation in gesture shape and duration, even for the same gesturer [5]. The intermediate motion between two gestures is termed a *transitional motion* [5]. *Segmentation ambiguity* is concerned with differentiating between meaningful gestures and transitional motions [5].

The aim of this paper is to identify and address shortcomings in feature vector selection and parameter optimisation for continuous gesture recognition. These shortcomings are general to all gesture recognition applications; however, the shortcomings are considered in the context of sign language interpretation to provide a paradigm in which they can be addressed. The sign language interpretation system is named and referred to as *Signect* (sign language + Kinect Sensor). In the context of sign language interpretation, 'isolated recognition' refers to recognition of isolated words; whereas, 'continuous recognition' refers to recognition of words in a continuous sentence.

This paper is organised as follows: Section II identifies three shortcomings in continuous recognition systems and Section III reviews the necessary Hidden Markov Model (HMM) theory; Section IV details the design of a gesture recognition system to facilitate investigation of the shortcomings; Section V defines the gesture protocol used in Section VI to analyse and address the shortcomings. Signect is implemented in C# using the statistical algorithms provided by the Accord.NET Framework [6].

II. IDENTIFICATION OF SHORTCOMINGS

Research on gesture recognition is typically focused on two major topics: determining a distinguishing feature vector; and, exploring various artificial intelligence classification techniques. This section identifies two shortcomings in feature vector selection, and one shortcoming in parameter optimisation for an HMM-based gesture recognition system.

A. Feature Vector Selection

Vogler and Metaxas [3] present a system to perform recognition of American Sign Language. The feature set is comprised of 2D and 3D features pertaining to wrist position, orientation and velocity (Cartesian and polar coordinates). Vogler and Metaxas [3] analyse the performance of 10 different feature vectors in recognising isolated gestures, and select the final feature vector accordingly; selection of the final feature vector does not consider performance in continuous

recognition. The system achieved a context independent word accuracy of 87.71 % on a 53 word vocabulary. Elmezain et al. [2] investigate the effects of combined location, orientation and velocity features in real-time hand gesture recognition. They consider 6 individual features and analyse the performance of 11 different feature vectors in recognising isolated gestures; again, performance in continuous recognition is not considered when selecting the final feature vector. The system is only tested on isolated gestures despite claims to operate in real-time. The feature vector consisting of all 6 individual features achieved the best isolated recognition results with an accuracy of 98.33 % on a gesture vocabulary of alphabetic (A-Z) and numeric (0-9) characters.

The approach of Vogler et al. [3] and Elmezain et al. [2] is indicative of the typical approach to feature vector selection. First, a set of candidate feature vectors is selected by randomly combining location, orientation and velocity features in Cartesian and/or polar coordinates. The set of candidate feature vectors is generally a very small subset of the total number of feature vectors [2, 3, 8] – the total number of feature vectors is given by 2^n, where n is the number of individual features. Second, the performance of each candidate feature vector is evaluated by computing the word accuracy in recognising *isolated* gestures [2, 3, 8]. The word accuracy is simply the percentage of words correctly classified. The feature vector which achieves the highest word accuracy is selected as the final feature.

This feature vector selection process has two main shortcomings:

1) The final feature vector is selected from a very small, somewhat random, subset of the possible feature combinations. The subset of candidate feature vectors is often less than 5 % of the total number of feature vectors; therefore, it is unlikely that the final feature vector selected will correspond to the optimum feature vector.

2) Since the final feature vector is selected in *isolated* recognition, the performance of the final feature vector can only be assured for gesture recognition systems that operate in isolated recognition mode – no assurance can be given as to the performance of the feature vector in continuous recognition mode. This is particularly problematic as real-world gesture recognition applications operate in real-time on continuous data streams, comprising both gestures and transitional motions.

B. Parameter Optimisation

The most popular classification techniques in gesture recognition are: template matching; artificial neural networks; and, *Hidden Markov Models (HMMs)* [7]. HMMs have become the preferred gesture classification technique due to their inherent ability to model order-constrained time series [3], and have achieved significant results in numerous gesture recognition systems [2, 3, 4, 8]. The parameter optimisation for HMM-based gesture recognition systems is generally focused on either the number of states in the HMMs, or the number of clusters in the vector quantiser [2, 8]. If the optimisation does address both parameters, optimisation of the number of states and number of clusters is performed independently, ignoring any possible interdependency.

III. HMM Theory

An HMM is a double stochastic process as governed by an underlying Markov chain with a finite number of states [8]. At every time instant, the process is in one of the states, and generates an observation symbol according to the random function corresponding to the current state [8]. A second random function governs the transitions between states.

Quantitatively, an HMM is characterised by five elements [9]:

N Number of states; $S = \{s_1, s_2, \dots, s_N\}$

M Number of distinct observation symbols per state; $V = \{v_1, v_2, v_3, \dots, v_M\}$

A State transition probability matrix $(N \times N)$; element a_{ij} is the probability of moving from state S_i to state S_j

B Observation symbol probability matrix $(N \times M)$; element $b_j(k)$ is the probability of emitting symbol v_k in state S_j.

Π Initial state probability matrix $(1 \times N)$; element π_i is the probability that state S_i is the initial state.

During the training phase, the Baum-Welch algorithm is used to adjust the parameters of an HMM to produce a given observation sequence [9]. During the evaluation phase, the Forward-Backward procedure is used to score how well a given HMM matches a given observation [9].

There are various HMM topologies suited to different applications. The most prominent topology in gesture recognition is the left-right topology (forward topology) due to its inherent ability to model order-constrained time series [3]. Figure 1 shows a four state left-right HMM with transition and output probabilities.

IV. System Design

Figure 3 is an overview of Signect showing the flow of data in training mode and recognition mode. Signect attempts to address the challenge of segmentation ambiguity by using a sliding window, velocity threshold, and likelihood threshold model. The challenge of spatiotemporal variability is addressed through k-means clustering and the HMM classifier. Details of the Signect system design are discussed in this section.

A. Data Acquisition

The Microsoft Kinect Sensor is used to acquire the raw gesture data in the form of a 640×480 RGB image stream, and a 320×240 depth stream [10]. The sampling frequency of the Kinect Sensor is 30 frames per second (fps) [10]. The Microsoft Kinect SDK is used to process the raw data streams, and includes a skeletal tracking algorithm which determines the 3D joint coordinates of 20 joints on the human body. The skeletal tracking algorithm is based on a randomised decision forest classifier, and is trained using hundreds of thousands of training images [11]. Gestures in the Signect gesture vocabulary are defined by movements of the shoulder, elbow, and wrist joints (see Section V).

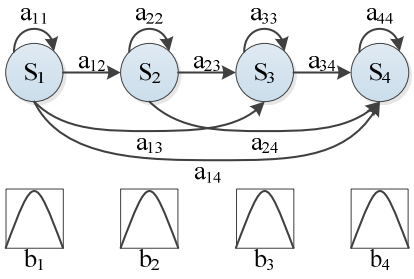

Figure 1. Four state left-right HMM

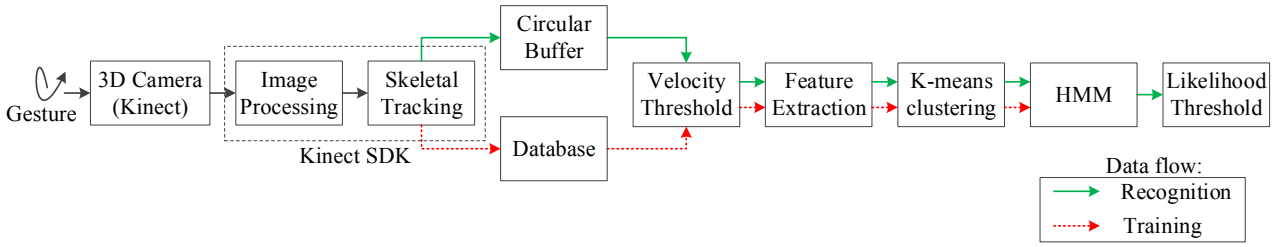

Figure 3. Overview of gesture recognition system showing flow of data in training mode and recognition mode

B. Data Storage

Several samples of each word in the gesture vocabulary are stored in the database to be used in training the classifier. A gesture is stored as a sequence of (x, y, z) joint coordinates. In recognition mode, a sliding window is used to sequentially analyse the data in 40 frame segments (1.33 s). The size of the window is chosen to be equal to the duration of the shortest gesture. This ensures that at some stage during a continuous motion, the window will be completely filled with gesture data, and will not contain any transitional motion data. In addition, a maximum of one complete gesture can be stored in the buffer at one time. Figure 2 illustrates the operation of the sliding window in gesture segmentation: at $t = t_1$ the window contains transitional data and gesture data; at $t = t_2$ the window is filled entirely with data from Gesture 2, and contains no transitional motion data.

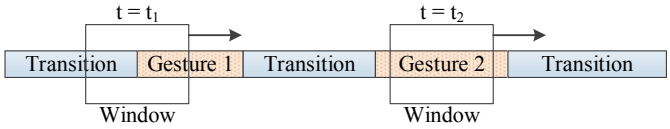

Figure 2. Operation of sliding window in gesture segmentation

C. Velocity Threshold

Recording a gesture from an initial start position to a final stop position results in an aggregation of data points at the initial and final positions. The aggregation of points must be removed as it does not contain distinctive gesture information.

In training mode, the velocity threshold $(0.1\ m/s)$ is used to remove the aggregation of data points from the beginning and end of a gesture sample. In recognition mode, the velocity threshold is used to determine whether the data in the sliding window contains sufficient movement to constitute a gesture.

The velocity threshold addresses the challenge of segmentation ambiguity as it enables simple differentiation between a resting position and a gesture. Data in the sliding window is considered to constitute a potential gesture when 20 % of the data in the window is above the velocity threshold $(0.1\ m/s)$. This eliminates erroneous classification of a potential gesture due to an abrupt jerk or twitch.

D. Feature Extraction

A distinguishing feature vector which adequately describes a gesture vocabulary is critical to the performance of the gesture recognition system [1, 3]. The Signect feature set is based on the three basic features of location, orientation and velocity using either polar or Cartesian coordinates [2]. Location and orientation features are selected relative to a reference joint to ensure that the features are invariant to the absolute position and orientation of the user within the workspace of the Kinect Sensor. Velocity features are invariant by nature. Gestures in the Signect gesture vocabulary are characterised by the movements of three joints on either arm (see Figure 4). The following set of individual features applies to both right and left arms.

1) Position of wrist relative to shoulder (posWrelS) $(x_{ws},\ y_{ws}, z_{ws})$
2) Distance from shoulder to wrist (distStoW) (r_{ws})
3) Distance and direction from right wrist to left wrist (distDirecRWtoLW) $(r_{rw-lw}, \theta_{rw-lw}, \phi_{rw-lw})$
4) Direction from shoulder to wrist (direcStoW) (θ_{ws}, ϕ_{ws})
5) Direction from elbow to wrist (direcEtoW) (θ_{we}, ϕ_{we})
6) Direction from shoulder to elbow (direcStoE) (θ_{es}, ϕ_{es})
7) Angle of elbow (angE) (θ_e)
8) Velocity magnitude of wrist (velMagW) $(|V|)$
9) Velocity direction of wrist (velDirecW) (V_θ, V_ϕ)

A feature vector is comprised of one or more individual features from the feature set. The number of possible feature combinations that can be obtained from a feature set is 2^n, where n is the number of features in the feature set. The Signect feature set is comprised of 9 features, thus 512 different feature vectors can be constructed. To ensure that all features have equal weighting, each feature in the feature vector is normalised to a range of 0 to 1. The final feature vector is determined during the feature vector selection process (see Section VI.A).

E. Vector Quantisation

The feature vector is quantised to obtain discrete observation symbols to apply to the HMM classifier [2]. The k-means clustering algorithm is used to classify the feature vectors into k clusters in the feature space [2]. The k-means clustering algorithm is based on the minimum distance between the centroid of each cluster and the points in the feature space [8]. The algorithm iteratively reclusters and recalculates the centroid of each cluster until convergence is reached [2]. It is important to note that the k-means clustering algorithm randomly selects the initial centroid means; thus, the random number generator must be seeded to obtain consistent results. The k-means clustering algorithm reduces the spatiotemporal variability of a gesture, as different gestures have different trajectories in the cluster space, while the same gestures show very similar trajectories [2]. The final number of clusters is determined in the system optimisation (see Section VI.B).

F. Classification

An HMM-based classifier is selected considering the established performance of HMMs in gesture recognition systems [2, 3, 4, 7, 8]. The theory of HMMs is reviewed in Section III. A discrete HMM is used in the left-right topology, as a continuous probability HMM requires a greater number of training samples [3]. The number of states is determined in the

TABLE 1. SIGNECT GESTURE VOCABULARY

Subject	Verb	Object
I	Do	Sport
You	Like	Exercise
We	Hate	Work

system optimisation and depends on the frame rate and the complexity of the gestures involved [3] (see Section VI.B).

In training mode, the Baum-Welch algorithm is used to train a unique HMM for each word in the gesture vocabulary [9]. In recognition mode, once quantisation has been completed on the feature vectors extracted from the sliding window, the Viterbi algorithm is used to determine the HMM that is most likely to generate the given observation sequence [9]. The unknown gesture is provisionally matched to the most likely HMM pending the outcome of the likelihood threshold.

G. Likelihood Threshold Model

The recognition error due to segmentation ambiguity caused by transitional motions can be further reduced by incorporating a likelihood threshold model. A likelihood threshold model provides a confirmation mechanism for the provisionally matched gesture patterns. An absolute threshold is not practical, as the likelihood values for a positive classification vary considerably depending on number of states, number of clusters, and the spatiotemporal variability of the training data. Lee and Kim [5] construct an artificial threshold model that consists of state copies of all trained gesture models in the system. The topology of the model is a fully connected ergodic model. The artificial threshold model can match any described gestures; however, a gesture is better described by the dedicated model because of the temporal order of a left-right HMM [5]. Thus, the artificial threshold model provides a confirmation mechanism for provisionally matched gesture patterns, and minimises the effects of transitional motions.

V. GESTURE PROTOCOL

The Signect gesture vocabulary is defined by movements of the following upper limb joints: shoulder, elbow, and wrist (see Figure 4). The gesture vocabulary is defined by selecting gestures that intuitively correspond to their meanings. The gestures make extensive use of the space surrounding the upper body. The vocabulary is comprised of three subjects, three verbs and three objects (see Table 1). The Signect gesture vocabulary can be used to construct 27 sentences of the form subject-verb-object. A detailed description of the Signect Gesture vocabulary can be found at `http://dept.ee.wits.ac.za/~gritzman/` .

To facilitate comparison between system performance in isolated recognition and continuous recognition, the Signect gesture vocabulary is used to record an isolated gesture dataset and a continuous gesture dataset. The isolated gesture dataset is obtained from 8 subjects (6 male, 2 female), and contains 80 samples for each of the 9 words (720 in total). The continuous gesture dataset is obtained from 2 subjects, and contains 9 or 10 samples for each of the 27 three-word sentences (260 in total). The *Signect Gesture Dataset* can be downloaded at `http://dept.ee.wits.ac.za/~gritzman/` and is free for research purposes.

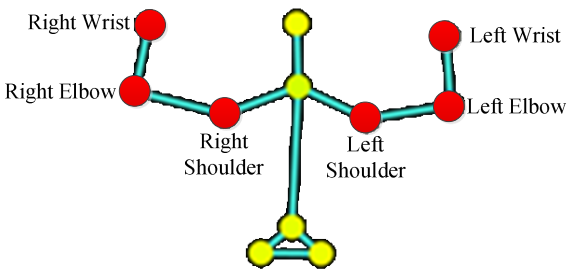

Figure 4. Joint positions used in formation of gestures

VI. FEATURE VECTOR SELECTION AND PARAMETER OPTIMISATION

The Signect gesture recognition system provides a platform to analyse three shortcomings that have been identified in feature vector selection and parameter optimisation for continuous gesture recognition. The Signect Gesture Dataset is used to quantify the improvements that can be obtained by addressing these shortcomings.

A. Feature Selection

In Section II.A, two shortcomings in feature vector selection are identified and described in detail. This section details the experimental approach to investigate these shortcomings, and the insights arising from subsequent analysis.

The first shortcoming concerns the limited number of candidate feature vectors analysed during feature vector selection. To analyse this shortcoming, it is necessary to evaluate the performance of every possible feature vector to determine whether a limited subset of feature vectors is adequate.

The second shortcoming concerns the selection of the final feature vector based on performance in recognising isolated gestures, even if the system is to operate in continuous recognition mode. To analyse this shortcoming, it is necessary to compare the performance of the *Final Feature Vector* selected in *isolated* recognition (FFV_{iso}) and the *Final Feature Vector* selected in *continuous* recognition (FFV_{cont}).

Figure 5 details the experimental procedure to analyse the shortcomings in feature vector selection for continuous gesture recognition. $\{FV_1, FV_2, \ldots, FV_N\}$ represents the set of candidate feature vectors to be analysed, and consists of every possible combination of features. FFV_{iso} is the final feature vector selected by comparing the candidate feature vectors in recognising isolated gestures. FFV_{cont} is the final feature vector selected by comparing the candidate feature vectors in recognising continuous gestures. Finally, FFV_{iso} and FFV_{cont} are both validated in isolated and continuous recognition.

The Signect Gesture Dataset (720 isolated word gestures; 260 continuous sentence gestures) is provisioned as shown in Table 2. In isolated mode, the system is trained with 60 % of the isolated gesture samples, and is tested with the remaining 40 %. In continuous mode, the system is trained with all of the isolated gesture samples, and is tested with all of the continuous gesture samples.

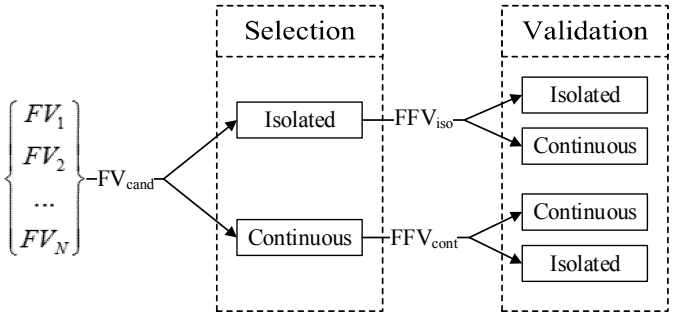

Figure 5: Experimental procedure to analyse shortcomings in feature vector selection for continuous recognition

TABLE 2. TRAINING AND TESTING DATA FOR FEATURE VECTOR SELECTION IN ISOLATED MODE AND CONTINUOUS MODE

Recognition Mode	Train Data	Test Data	
	Isolated	*Isolated*	*Continuous*
Isolated	450	270	-
Continuous	720	-	260

Selection and validation of the feature vector in continuous recognition mode gives rise to a new challenge: in isolated recognition, classification of a gesture sample is either correct or incorrect; however, in continuous mode, a sequence of gestures may be classified with partial success. In the case of a sign language interpretation, the system may correctly identify certain words in a sentence, while incorrectly identifying other words. Thus, the challenge in continuous feature vector selection and validation is to find a metric to quantify the accuracy of the classifier.

It is possible to quantify the accuracy of a continuous recognition system by evaluating each sentence as either correct or incorrect; i.e. a sentence is only correct if every word in the sentence is correctly recognised, in the correct order, and no erroneous words are included – this is termed 'perfect sentence recognition'. However, a sentence may be imperfect without affecting the meaning of the sentence. For example, consider the following two cases in which a gesturer forms the sentence "I do sport" – in case 1, the system output is "I do do sport"; in case 2, the system output is "I hate work". In case 1, an erroneous word ("do") has been included, however, the erroneous word has not affected the meaning of the sentence. In case 2, the system incorrectly classifies 2 words ("hate work", instead of "do sport"). The sentence meaning has been dramatically altered, and it is not possible to recover the original meaning from the recognition output. According to a binary evaluation metric, in both cases the output sentence is incorrect. The binary evaluation metric is unduly harsh on case 1 which contains a minor error, in comparison to case 2 which contains a major error. No protocol currently exists to quantify the partial recognition accuracy of a continuous recognition system; thus, a novel scoring system is developed and introduced. The scoring system is designed with sign language interpretation in mind, however, the concepts can be generalised to other gesture recognition applications.

The scoring system is developed to quantitatively compare the sentence input by the gesturer to the sentence output by the classifier. An *input word* is a word-gesture performed by the gesturer; an *output word* is a word identified by the classifier. The output sentence is allocated an initial score of 100, and penalties are imposed according to the rules described in Table 3. The penalties are assigned according to the effect of the error on the meaning of the sentence. In the above scenario, the input sentence is "I do sport". In case 1, the system output is "I do do sport" which contains a repetition error – score: 95. In case 2, the system output is "I hate work" which contains inclusion, omission and repetition errors; only the first word ("I") is correctly identified – score: 20.

The Signect feature set comprises 9 individual features which can be combined to form 512 different feature vectors. Each candidate feature vector is assigned a *Feature Vector Number (FN)* from 1 to 511 based on a binary representation of which individual features are included. For example, FN136 (binary `010001000`) includes feature *2* (posWrelS) and feature *5* (direcEtoW) from the feature set (Section IV.D). The system is initially set up with *35* clusters and *7* states (to be optimised later). Each feature vector is analysed in isolated recognition using binary word accuracy, and in continuous recognition using the novel scoring system and the perfect sentence recognition metric ('number perfect'). Table 4 shows results for selection and validation of the feature vector in isolated recognition and continuous recognition.

In isolated feature vector selection, four feature vectors (FN: 268, 270, 286, 398) achieved an accuracy of *99.63 %*, misclassifying only one isolated gesture sample. The four feature vectors all include the individual features 1 (posWrelS), 6 (direcStoE) and 7 (angE). There seems no intuitive way of predicting that these three individual features contain the most significant gesture information. This result suggests that a small subset of hand-picked candidate feature vectors is likely to omit the optimum feature vector; a rigorous analysis of all possible feature vectors is necessary to find the optimum feature vector.

In continuous feature vector selection, FN404 is selected as FFV_{cont} as it achieved the highest score of 84.76 (138 perfect sentences). FN404 contains individual features 1 (posWrelS), 2 (distStoW), 5 (direcEtoW) and 7 (angE). Again, there seems no intuitive way of predicting the superior performance of this feature vector, highlighting the need for rigorous feature vector analysis.

To determine whether FFV_{iso} is suitable for continuous recognition systems (as proposed in the literature), it is necessary to compare the results of FFV_{iso} and FFV_{cont} in continuous recognition. In isolated feature vector selection, four feature vectors are equal candidates for FFV_{iso} (FN: 268, 270, 286, 398), therefore the average of these four feature vectors is considered.

In continuous recognition, FFV_{iso} obtained an average rank of 130 (of 512). In continuous recognition, FFV_{iso} obtained an average score of 80.07 with an average number of 93.75 perfect sentences (of 260). FFV_{cont} obtained significantly better results in continuous recognition compared to FFV_{iso}. FFV_{cont} achieved a score of 84.76 (increase of 5.85 %), and 138 perfect sentences (increase of 47.2 %).

FFV_{iso} distinguishes effectively between isolated gestures;

TABLE 3. PENALTY SYSTEM USED IN SCORING OUTPUT SENTENCE

Error Type	Error in output sentence	Penalty
Incorrect inclusion	Erroneous output word included	-30 (-30 max)
Omission	Input word omitted	-20 for each word (-60 max)
Repetition	Input word erroneously repeated	-5 for each word (-15 max)
Ordering	Output words correct except order incorrect	-10 (-10 max)

TABLE 4. RESULTS FOR FEATURE VECTOR SELECTION AND VALIDATION IN ISOLATED RECOGNITION AND CONTINUOUS RECOGNITION

FN		Isolated		Continuous		
Dec.	*Binary*	*Rank*	*% Acc.*	*Rank*	*Score*	*# Perf.*
268	`100001100`	1	99.63	146	79.24	76
270	`100001110`	1	99.63	180	78.69	80
286	`100011110`	1	99.63	6	83.96	135
398	`110001110`	1	99.63	187	78.39	84
404	`110010100`	49	98.15	1	84.76	138
400	`110010000`	29	98.52	2	84.61	116
450	`111000010`	22	98.89	3	84.22	135

TABLE 5. RESULTS OF SIMULTANEOUS OPTIMISATION

Rank	FN	Max Score	Cluster	States
1	450	86.741	42	9
2	400	85.685	34	3
3	404	85.278	26	4

however, the distinguishing power of FFV_{iso} is substantially reduced by the segmentation ambiguity caused by transitional motions between gestures. These results indicate conclusively that the final feature vector cannot be selected in isolated recognition, if the system is to operate in continuous recognition mode. The converse is also true: in isolated recognition FFV_{cont} obtained a rank of 49 (of 512); thus, it is not appropriate to select the final feature vector in continuous recognition, if the system is to operate in isolated recognition mode.

B. *Optimisation of Number of Clusters and States*

Section II.B identifies a shortcoming in the optimisation of the number of states (HMMs) and number of clusters (k-means clustering) – optimisation is focused on either the number of states, or the number of clusters. If the optimisation does address both parameters, optimisation of the number of states and number of clusters is performed independently. In this section, the number of states and number of clusters are optimised simultaneously, and the resulting insights are discussed.

The top three ranked feature vectors from continuous feature vector selection are optimised (FN: 404, 400, 450). The number of states ranges from 1 to 13, and the number of clusters ranges from 10 to 60. The results of the simultaneous optimisation are shown in Table 5. It is interesting to note that the ranking of the top three feature vectors is reversed. All feature vectors show a score increase of above 0.5 points, with FN450 showing the maximum increase of 2.52 points (3.0 %) from 84.22 to 86.74. Optimising FN450 increased the number of perfect sentences by 11.1 % from 135 to 150. The optimum number of clusters and states does not show any distinct pattern and is unique to each feature vector.

Figure 6 shows a surface plot of the optimisation of FN450. As expected, a low number of states or a low number of clusters yields a low score; i.e. clustering or HMM classifier is too general. The maximum score is attained at 42 clusters and 9 states. As the number of clusters and number of states continues to increase, the score begins to decrease due to over fitting (the excessive number of states and clusters causes the HMMs to fit too closely to the training data, resulting in poor classification of unseen data [2]). It is clear from Figure 6 that the number of states in relation to the number of clusters significantly impacts recognition accuracy; evaluating either independently (or not at all) will not yield optimum system performance – simultaneous optimisation is essential.

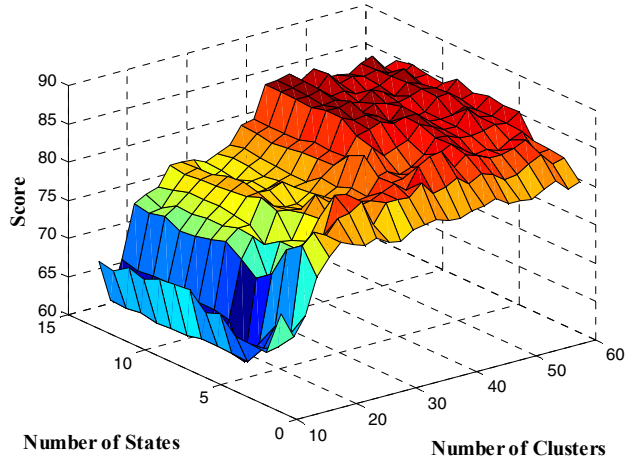

Figure 6. Simultaneous optimisation number of states (HMMs) and number of clusters (k-means clustering) for FN450

VII. CONCLUSION

This paper identifies and addresses three shortcomings in current approaches to feature vector selection and parameter optimisation for continuous gesture recognition. The shortcomings (SC) and resulting insights (IS) are summarised below.

SC1 The number of candidate feature vectors analysed is only a small subset of the possible feature combinations.

IS1 The optimum feature vector cannot be intuitively predicted; rigorous analysis of possible feature combinations is essential.

SC2 The final feature vector is selected based on the performance in recognising isolated gestures, even if the system is to operate in continuous recognition mode.

IS2 Selecting the final feature vector in continuous recognition is essential; the accuracy score improved by 5.85 % (80.07 to 84.76) and the perfect sentence recognition improved by 47.2 % (93.75 to 138).

SC3 Optimisation is focused on either number of states or number of clusters – if both are considered, optimisation is performed independently.

IS3 The number of states in relation to the number of clusters significantly impacts recognition accuracy; simultaneous optimisation improved the score by 3.0 % and the perfect sentence recognition by 11.1 %.

REFERENCES

[1] O. Portillo-Rodriguez, O. Sandoval-Gonzalez, C. Avizzano, E. Ruffaldi, D. Vercelli, M. Bergamasco, "Development of a 3D Real Time Gesture Recognition Methodology for Virtual Environment Control", Proceedings of the 17th IEEE Symposium on Robot and Human Interactive Communication, 2008, pp. 279-284.

[2] M. Elmezain, A. Al-Hamadi, B. Michaelis, "Improving Hand Gesture Recognition Using 3D Combined Features", Second International Conference on Machine Vision, 2009, pp. 128-132.

[3] C. Vogler, D. Metaxas, "ASL Recognition Based on a Coupling between HMMs and 3D Motion Analysis", Sixth International Conference on Computer Vision, 1998, pp. 363-369.

[4] C. Keskin, A. Erkan, L. Akarun, "Real Time Hand Tracking and 3D Gesture Recognition for Interactive Interfaces Using HMM", In Proceedings of the Joint International Conference ICANN/ICONIP, 2003, pp. 26-29.

[5] H. Lee, J. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 10, 1999, pp. 961-973.

[6] C. Souza, "The Accord.NET Framework", https://code.google.com/p/accord/, last accessed: 12-03-2013.

[7] J. Cracknell, A. Cairns, P. Gregor, C. Ramsay, I. Ricketts , "Gesture Recognition: An Assessment of the Performance of Recurrent Neural Networks Versus Competing Techniques", IEEE Colloquium on Applications of Neural Networks to Signal Processing, 1994, pp. 8/1-8/3.

[8] H. Yoon, J. Soh, Y. Bae, H. Yang, "Hand Gesture Recognition using Combined Features of Location, Angle and Velocity", Pattern Recognition, vol. 34, no. 7, 2001, pp. 1491-1501.

[9] L. Rabiner, "A Tutorial on HMMs and Selected Applications in Speech Recognition", Proceeding of the IEEE, vol. 77, no. 2, 1989, pp. 257-286.

[10] Microsoft Corporation, "Kinect for Windows SDK Beta", Microsoft Research, 2011.

[11] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, "Real-time Human Pose Recognition in Parts from Single Depth Images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 201