A vision based motion interface for mobile phones

Mark Barnard, Jari Hannuksela, Pekka Sangi and Janne Heikkilä

Machine Vision Group
Department of Electrical and Information Engineering
University of Oulu, Finland
barnard@ee.oulu.fi

Abstract. In this paper we present an interface system for the control of mobile devices based on motion and using existing camera technology. In this system the user can control the phone's functions by performing a series of motions with the camera and each command is defined by a unique series of these motions. A sequence of motion features is produced using the phone's camera and these characterise the translation motion of the phone. These sequences of motion features are classified using *Hidden Markov Models* (HMMs). In order to improve the robustness of the system the results of this classification are then filtered using a likelihood ratio and the entropy of the sequence to reject possibly incorrect sequences. When tested on 570 previously unseen motion sequences the system incorrectly classified only 5 sequences.

1 Introduction

Control of mobile phones is a difficult and challenging problem, given the small size of the device. Traditional input devices, such as keyboards, or screen based interfaces, such as a mouse, are not generally practical. With these constraints the input of complex commands to the device is difficult. Ideally an input system should not require any additional hardware to be fitted to existing phone and should instead be capable of using the phone's existing hardware.

Given the prevalence of mobile phones fitted with cameras and some processing capability, we propose a solution using the phone's camera and implemented in software that could be retro-fitted to most existing phones. In this system the user can operate the phone through a series of hand movements whilst holding the phone. During these movements the phone's camera is used to record the ego-motion of the camera. This motion is modelled and the parameters of this motion model are extracted as features. So these motion features characterise the motions of the user's hand. The motion feature sequences are modelled with *Hidden Markov Models* (HMMs). An HMM is a statistical model that is capable of representing temporal relations in sequences of data. Each HMM produces a likelihood of a particular command given the input feature sequence.

In order to improve the initial classification we use a two-level filtering of the result. The first level of filtering is based on the likelihood ratio between the most likely command and the second most likely command. This ratio can be seen as a confidence measure of the classification result. If this ratio is below a certain predefined threshold then the confidence in the result is low and the sequence is rejected.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

A second level of filtering is employed to reject unintentional or accidental sequences, such as when the input system is activated without the user's knowledge or the user loses control of the phone for some reason. It is important that these unintended commands are not recognised and executed as real commands.

An alternative solution for estimating the device motion could be based on special motion sensors such as accelerometers. However, there is a need to install external sensors. Today, computer vision is a more natural choice, because current mobile phones are often equipped with cameras that can provide visual input for estimating motion.

Prior work on vision based user interfaces for mobile phones has mainly focused on browsing and navigation on the display. This is also a rather new application field and there are few solutions available so far. Möhring et al. [9] presented a tracking system for estimating 3-D camera pose using special colour coded markers. The frame rate, including marker detection, barcode reading and rendering is about 5 fps. Rohs [13] used a block matching based technique to measure the relative x, y, and rotational motion. The frame rate of the algorithm was about 5 fps. Drab et al. [3] use a method called projection shift analysis to measure the x- and y- motion of the device. Recently, Haro et al. [6] proposed a feature based tracking system in order to determine scroll direction and magnitude. Their method uses motion magnitude for zooming. Unlike the other methods presented here, our motion estimation method allows the estimation the of the camera rotation as well as the lateral motion. Our solution is also efficient and provide frame rate of 10 fps.

We wish to emphasise the point here that our goal is to create a general solution to this problem rather than tailoring the solution for a particular user. So in our system after the initial training, no parameters need to be adjusted for each user. This goal is reflected by the fact that the data used for training is collected from completely different users from the data used to test the system.

2 Feature extraction

In this section, our method for estimating the global motion of a mobile phone is briefly reviewed [5]. Global motion refers to the apparent dominant 2-D motion between frames, which can be approximated by some parameterised flow field model [10]. Our method for estimating such models has two main phases. In the first phase, the motion of the selected features and related uncertainty is analysed, and in the second phase, outliers are removed and the results are used for obtaining parametric global motion estimates.

2.1 Feature motion analysis

In the feature motion analysis phase, good features from the scene is first selected. Various criteria for the feature selection have been proposed, and they typically analyse the richness of texture within a small image window [16]. In our approach, the spatial gradient is evaluated in horizontal and vertical directions for each image region $\mathcal{B}_i$ in the obtained image. Each image region $\mathcal{B}_i$ contains M by M pixels (e.g. $M = 6$). The sum of gradient values $G(\mathcal{B}_i)$ is used as a criterion. In order to distribute features over the image, the central image region is divided into N smaller rectangular subregions

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

(e.g. $N = 16$), and one feature block which maximizes the measure is selected from each region (see Fig. 1). For our computational platform, such a straightforward scheme is well suited. It must be noted that a subregion may not contain any good features, or it might contain only straight edges. The estimation of feature motion may therefore suffer from the aperture problem. In order to alleviate this problem, uncertainty in feature motion estimates is analysed and used in global motion estimation.

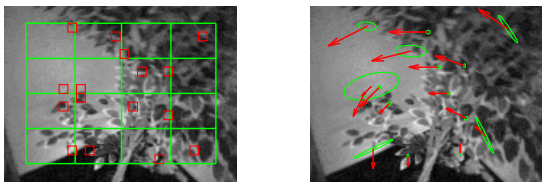

Fig. 1. Motion estimation principle. Left: feature selection, right: local motion features.

In order to determine displacements of selected feature blocks, exhaustive evaluation of a sum of squared differences (SSD) block matching measure, $D(\boldsymbol{v}; \mathcal{B}_i)$, is performed over a suitable range of integer displacements in x- and y-directions (e.g. -12...+12). The displacement $\boldsymbol{v}$ that minimizes the criterion is used as the displacement estimate $\boldsymbol{v}_i$. Uncertainty of this estimate is analysed by performing gradient-based thresholding for the motion profile. Moments of the thresholding results provide a 2×2 covariance matrix $\boldsymbol{C}_i$ (error ellipses), which represents the local motion uncertainty. Computation of $\boldsymbol{C}_i$ is based on an analysis where we select motion candidates, which possibly have the true motion in their vicinity. For selection, we use the block gradient measures computed in the previous step for thresholding the SSD values of motion candidates. The rule for determining the set of proper candidates, V_i, is of the form

$$V_i = \{\boldsymbol{v} \mid D(\boldsymbol{v}; \mathcal{B}_i) \leq k_1 G(\mathcal{B}_i) + k_2 \}, \tag{1}$$

where k_1 and k_2 are some fixed parameters [14]. Once V_i has been determined, the confidence of the feature motion estimate is summarised as a covariance matrix

$$\boldsymbol{C}_i = \frac{1}{\#V_i} \sum_{\boldsymbol{v} \in V_i} \left((\boldsymbol{v} - \boldsymbol{v}_{c,i})(\boldsymbol{v} - \boldsymbol{v}_{c,i})^T \right) + \frac{1}{12} \boldsymbol{I}, \tag{2}$$

where $\boldsymbol{v}_{c,i}$ denotes the centroid of V_i and $\boldsymbol{I}$ is a 2×2 identity matrix. The resulting covariance matrices are illustrated as green ellipses in Fig. 1.

2.2 Global motion analysis

As a result of the feature motion analysis phase, we have a set of feature motions represented by triplets $F_i = (\boldsymbol{p}_i, \boldsymbol{d}_i, \boldsymbol{C}_i)$, $i = 1, ..., N$. In the second phase, global motion is estimated using this information. We use a four parameter similarity motion model which represents the displacement $\boldsymbol{d}$ of a feature located at $\boldsymbol{p} = [x, y]^T$ using

$$\boldsymbol{d} = \boldsymbol{d}(\boldsymbol{\theta}, \boldsymbol{p}) = \boldsymbol{H}[\boldsymbol{p}]\boldsymbol{\theta} = \begin{bmatrix} 1 & 0 & x & y \\ 0 & 1 & y & -x \end{bmatrix} \boldsymbol{\theta}, \tag{3}$$

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4]^T$ is a vector of model parameters and $\boldsymbol{H}[\boldsymbol{p}]$ is an observation matrix. Here, θ_1 and θ_2 are related to common translational motion, and θ_3 and θ_4 encode information about 2-D rotation ϕ and scaling s, $\theta_3 = s\cos\phi - 1$ and $\theta_4 = s\sin\phi$.

Outlier analysis The purpose of outlier analysis is to discard those feature motion measurements that might be harmful. For example, feature motion estimates might be erroneous due to image noise, or there might be several independent motions in the scene. We assume that the majority of features are associated with the global motion we want to estimate. In order to select inlier features, a method similar to the RANSAC [4] is used. In our approach, pairs of features F_i are chosen for instantiating motion model hypotheses, which are then voted for by other features. The hypothesis that gets the most support is considered to be close to the dominant global motion and is used for selecting the inlier features. Motion hypothesis $\boldsymbol{\theta}_{k,l}$ for a feature pair (F_k, F_l) is generated by solving a system of equations, which is based on (3). As the number of features can be relatively small, we generate and evaluate hypotheses for all feature combinations.

Let us denote with $\nu_i(\boldsymbol{\theta}_{k,l})$ the number of votes that a feature F_i gives to the hypothesis $\boldsymbol{\theta}_{k,l}$. The covariance matrix $\boldsymbol{C}_i$ associated with the feature F_i provides information about the feature motion uncertainty in different directions. It is therefore reasonable to base the calculation of $\nu_i(\boldsymbol{\theta}_{k,l})$ on the Mahalanobis distance between the estimated displacement, $\boldsymbol{d}_i$, and hypothesized displacement $\boldsymbol{d}(\boldsymbol{\theta}_{k,l}, \boldsymbol{p}_i)$ given by (3). In order to simplify calculations, we use the squared Mahalanobis distance

$$d(F_i, \boldsymbol{\theta}_{k,l}) = \boldsymbol{d}_{i,k,l}^T \boldsymbol{C}_i^{-1} \boldsymbol{d}_{i,k,l}, \tag{4}$$

where $\boldsymbol{d}_{i,k,l} = \boldsymbol{d}_i - \boldsymbol{d}(\boldsymbol{\theta}_{k,l}, \boldsymbol{p}_i)$. Using this distance, the number of votes is computed using

$$\nu_i(\boldsymbol{\theta}_{k,l}) = \begin{cases} T_v - d(F_i, \boldsymbol{\theta}_{k,l}) & \text{if } d(F_i, \boldsymbol{\theta}_{k,l}) < T_v \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

where T_v is a user-defined threshold (in our experiments, $T_v = 4.0$). The hypothesis $\boldsymbol{\theta}_{k,l}$ that maximizes the sum $\sum_{i=1}^{N} \nu_i(\boldsymbol{\theta}_{k,l})$ is used for selecting features F_i', which are passed to the global motion estimation step. These inlier features are those that give some support for the best hypothesis, that is, $\nu_i(\boldsymbol{\theta}_{k,l})$ is non-zero for them.

Global motion estimation In order to compute interframe motion, we assume that the motion of any inlier feature $F_i' = (\boldsymbol{p}_i, \boldsymbol{d}_i, \boldsymbol{C}_i)$ is a realization of the random vector

$$\boldsymbol{d}_i = \boldsymbol{H}[\boldsymbol{p}_i]\boldsymbol{\theta} + \boldsymbol{\eta}_i, \tag{6}$$

where $\boldsymbol{\theta}$ is the true motion, and $\boldsymbol{\eta}_i$ is the observation noise with $E(\boldsymbol{\eta}_i) = \boldsymbol{0}$ and $E(\boldsymbol{\eta}_i \boldsymbol{\eta}_i^T) = \boldsymbol{C}_i$. Observations of feature motions are assumed to be independent. The best linear unbiased estimator for the motion is

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{H}^T \boldsymbol{W} \boldsymbol{H})^{-1} \boldsymbol{H}^T \boldsymbol{W} \boldsymbol{d}, \tag{7}$$

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

where $\boldsymbol{d}$ is the vector of motion observations composed of $\boldsymbol{d}_i$, $\boldsymbol{W}$ is the inverse of the block-diagonal matrix composed of $\boldsymbol{C}_i$, and the observation matrix $\boldsymbol{H}$ is composed of $\boldsymbol{H}[\boldsymbol{p}_i]$. Uncertainties associated with feature motions can be represented with the covariance matrix

$$\boldsymbol{C}_{\hat{\theta}} = (\boldsymbol{H}^T \boldsymbol{W} \boldsymbol{H})^{-1}. \tag{8}$$

Using a matrix decomposition (7) and (8) are evaluated efficiently.

The motion sequences obtained are used as a feature for recognising device movements. The following section will describe how this motion information can be combined with pattern recognition techniques for more advanced interaction purposes like recognising gestures.

3 Sequence Classification

Here we will present a more formal definition of the classification problem. We have a set of classes $\{Cl_1, Cl_2, \ldots Cl_N\}$ and a sequence of data $X = \{x_1, x_2, \ldots x_T\}$. In our case X is a set of motion trajectory coordinates given by the motion estimator $\hat{\boldsymbol{\theta}}$ described in Section 2. Statistical modelling is the process of creating a probability density function $p(X|Cl_k)$ in the feature space of X. This function is approximated by a set of models $\{M_1, M_2, \ldots M_N\}$ corresponding to each of the classes. The task of sequence classification based on statistical modelling is to assign the data sequence X to one particular class Cl_k. Bayes rule is used to select the most probable class by,

$$k^* = \arg\max_k p(X|M_k)P(M_k). \tag{9}$$

where k is the class, M_k is the model trained for class k and $P(M_k)$ is the prior probability of the class k which is uniform in our case.

3.1 Hidden Markov Models

In order to perform the classification we must select an appropriate method of modelling the data sequences. The most common method currently used to model sequences of data are *Hidden Markov Models* (HMMs) [12]. An HMM is a statistical model capable of representing temporal relations in sequences of data. The data is characterised as a parametric stochastic process and the parameters of this process are automatically estimated from the data. The data sequence is factorised over time by a series of hidden states and emissions from these states. The transition between states is probabilistic and depends only on the previous state. In our case the continuous emission probability from each state is modelled using *Gaussian Mixture Models* (GMMs) [12]. The topology of an HMM can be specified in a transition probability matrix A and an initial state probability vector π. HMM training can be carried out using the *Expectation-Maximisation* (EM) algorithm [2] and sequence decoding using the Viterbi algorithm [17].

HMMs have been used successfully in many different sequence recognition applications. In speech recognition HMMs are the most common method of modelling, where they are used to model phonemes, words and also sentences [12] [8]. In the field of handwriting recognition HMMs have also been applied with successful results [7]. In

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

this case HMMs are used to model sequences of pen strokes that form letters and also sequences of letters that form words. HMMs have also been used in the task of video annotation, for example tennis stroke recognition [11] and also the segmentation of soccer into play and non-play sequences [1].

4 System overview

The system we propose here uses HMMs, described in Section 3, to model the motion features described in Section 2. These models are then used to classify motion sequences that are input from the user. This initial result is then filtered in order to reject any possibly incorrect commands before they can be executed. The methods used for filtering the results are described in the next section. An overview of the system is shown in Figure 2. In order to ensure a minimum number of incorrect classifications we filter the results of the classification for each sequence. This is done using two criteria: the first is the log-likelihood ratio of the most likely class and the second most likely class and the second is the entropy of the sequence.

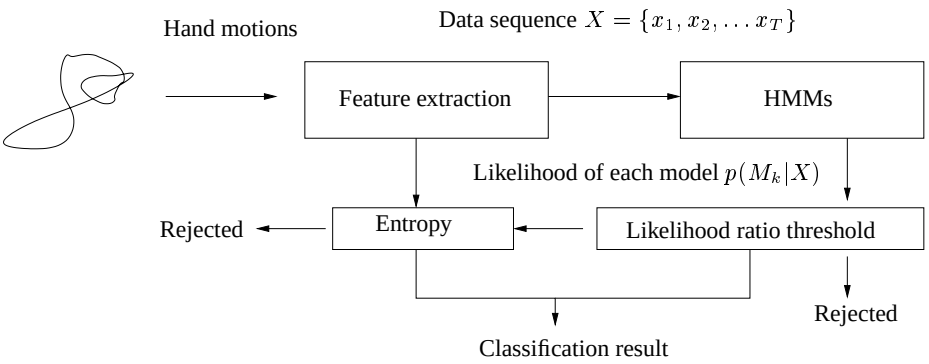

Fig. 2. An overview of the proposed system.

Likelihood ratio Recall from Section 3 that we have a set of classes $\{Cl_1, Cl_2, \ldots Cl_N\}$ and a sequence of data $X = \{x_1, x_2, \ldots x_T\}$ the class with the highest likelihood given X is denoted by Cl_a and the class with the second highest likelihood given X is denoted by Cl_b. The log-likelihood ratio δ for a particular sequence is given by

$$\delta = log(p(Cl_a|X)) - log(p(Cl_b|X)), \tag{10}$$

where $p(Cl|X)$ is the likelihood of the class Cl given the data sequence X.

In our experiments we use two threshold values for δ. The first , δ_{hard}, is a hard decision and any sequence with a log-likelihood ratio below this threshold is rejected. The second, δ_{soft} is a soft decision, for any sequence where $\delta_{hard} < \delta < \delta_{soft}$ the entropy of the sequence is used as an addition indicator of the quality of the classification.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

Entropy The Information (or Shannon) Entropy is a measure of the randomness of a probability distribution of a random variable y and is given by [15]

$$H = -K \sum_{s=1}^{S} P(y_s) log_2 P(y_s), \tag{11}$$

where $P(y_s)$ is the probability of y, S is the number of samples and K is a constant. In our case we take the first derivative of the motion trajectory X, this gives us the velocity of the motion. This continuous velocity sequence is quantised into a histogram and we calculate the entropy of the entries in this histogram, so the random variable y is the size of the histogram bins.

A sequence with a larger entropy has a larger degree of randomness. This measure of entropy is used as a method of filtering the final classification result. Our hypothesis here is that well formed signs will have a more constant velocity, and so a lower entropy, than random or poorly formed signs. In our experiments we have included a number of sequences where the user has either deliberately made a bad sign or has just moved the phone at random. These sequences are used to test the case where the system may be unintentionally turned on by the user or the user loses control of the phone whilst making a sign. The mean of the entropy of "bad" sequences in the validation set is 0.88 with standard deviation of 0.11, while the mean and standard deviation of the "good" sequences is 0.58 and 0.13 respectively.

Those sequences with a log likelihood ratio, δ, satisfying $\delta_{hard} < \delta < \delta_{soft}$ are classified according to their entropy. Any sequences with an entropy higher than a predetermined threshold are rejected as potentially incorrect. This entropy threshold H_{th} is set on the validation set as described in the next section.

5 Experiments

In order to validate the technique described here a hypothetical control system of mobile phone functions was devised. In this system a series of control commands was proposed. These commands are composed of seven simple elements based on seven different motions. These seven elements are shown in Table 1. Using these basic elements alone the system would be limited to seven different commands, so in order to provide a greater number of commands more complex commands are constructed from these motion elements. These complex commands are used for our recognition experiments and are shown in Table 2. Although we have used 11 complex commands this could easily extended to a larger number of commands.

Data and experimental procedure The experimental data was collected from 35 subjects. Each subject was asked to draw each of the commands in Table 2 five times using a standard camera equipped mobile phone, a Nokia N90. The majority of subjects had no previous experience in performing this task. There was considerable variability of the sequences both between subjects and also between different attempts from the same subject, this can be seen in Figure 3. In addition to these subjects 30 random sequences were collected. These sequences were produced by moving the camera in a random

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

Element	Type of motion
←	Left horizontal
→	Right horizontal
↑	Up vertical
↓	Down vertical
↙	Left diagonal
↘	Right diagonal
↺	Anti-clockwise circle

Table 1. Seven basic motion elements

Name	Command
Com1	↙↘↺
Com2	↙↘
Com3	↺
Com4	↑→
Com5	↑←
Com6	↓→
Com7	↓←
Com8	↑→↺
Com9	↑←↺
Com10	↓→↺
Com11	↓←↺

Table 2. Eleven complex commands constructed from the seven basic motions

way. These random or "bad" sequences were included in the data to test the system's performance with input cause by accidental activation of the camera or the user losing control of the phone whilst making a sign.

The subjects were randomly divided into training, validation and test sets. There were 20 subjects in the training set, 5 subjects in the validation set and 10 subjects in the test set. Additionally 10 "bad" sequences were added to the validation set and 20 to the test set. Giving a total of 1100, 285 and 570 sequences in the training, validation and testing sets respectively.

It must be emphasised again that there was no overlap of subjects between these three sets. The training set was used to train the parameters of the HMMs. The validation set was used set the hyper-paramters of the individual models, such as the number of Gaussians in the GMMs, that model the state distributions of the HMMs, and the number of states in the HMMs. The validation set was also used to set the hard threshold, δ_{hard} and the soft threshold δ_{soft}. In addition the entropy threshold H_{th} was set by using the validation set. The values of these parameters was; Number of states = 11, Number of Gaussians = 10, $\delta_{hard} = 5$, $\delta_{soft} = 30$ and $H_{th} = 0.72$.

Results and discussion The results of running the system on the 570 test sequences are shown in Table 3. It can be seen from these results that only 5 sequences are incorrectly classified, while 27 sequences that would have produced an incorrect result were rejected by the system. These rejected sequences included all of the 20 deliberately bad sequences. It is particularly interesting that 13 of these bad sequences were rejected using the entropy criteria. This result confirms that the higher entropy of the bad sequences observed in the validation set can be generalised to the bad sequences in the test set.

6 Conclusions

We have presented here a system that combines motion features and statistical sequence modelling to classify the hand movements of a user in order to control a mobile phone. In addition to this we introduced two methods of filtering the result of this classification, likelihood ratio and entropy, making the system more robust to the presence of random

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

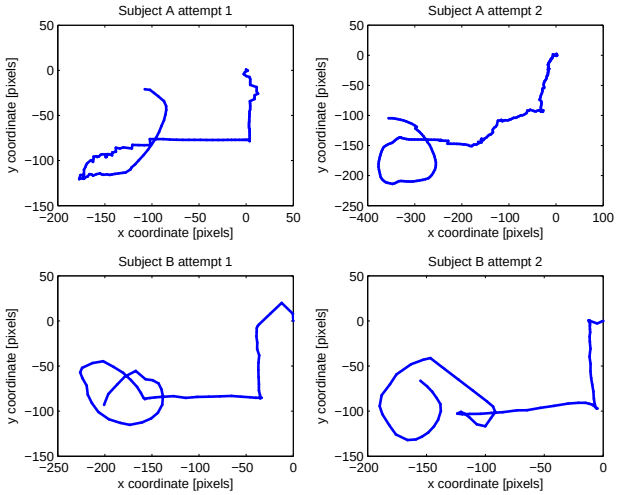

Fig. 3. The sequence ↓←◌́ performed by two different users. Each row shows two attempts of the same sign from a single user. This shows both the intra-user and inter-user variability of the data.

Command	Correct	Correct rejected	Incorrect rejected	Incorrect
Com1	50	0	0	0
Com2	47	2	0	1
Com3	46	4	0	0
Com4	49	0	1	0
Com5	50	0	0	0
Com6	48	0	2	0
Com7	50	0	0	0
Com8	47	0	1	2
Com9	46	1	1	2
Com10	48	0	2	0
Com11	50	0	0	0
Bad seq	0	20	0	0
Total	531	27	7	5

Table 3. Results of testing on 570 sequence of which 20 were intentionally bad. It should be noted that of 570 sequences only 5 were finally incorrectly classified.

or bad sequences. It is clear from the results shown in Section 5 that this method of using entropy as an indicator of badly formed sequences is able to filter out all such sequences from the final result.

The robustness of the system means that no adjustment of parameters is necessary for a new user. Aside from its accuracy and robustness another important feature of this system is that it can be installed on an existing phone equiped with a camera. While

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

other motion based systems may produce comparable results they require specific hardware to be fitted to the phone, whereas our system is implemented entirely in software.

References

1. M. Barnard, J.M. Odobez, and S. Bengio. Multi-modal audio-visual event recognition for football analysis. In *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, Toulouse, France, 2003.
2. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood for incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977.
3. S. A. Drab and N. M. Artner. Motion detection as interaction technique for games & applications on mobile devices. In *Proc. of Pervasive Mobile Interaction Devices*, 2005.
4. M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of ACM: Graphics and Image Processing*, 24:381–395, 1981.
5. J. Hannuksela, P. Sangi, and J. Heikkilä. A vision-based approach for controlling user interfaces of mobile devices. In *Proc. of IEEE Workshop on Vision for Human-Computer Interaction*, San Diego, CA, June 2005.
6. A. Haro, K. Mori, T. Capin, and S. Wilkinson. Mobile camera-based user interaction. In *ICCV 2005 Workshop on HCI*, pages 79–89, Beijing, China, 2005.
7. J. Hu, M. K. Brown, and W. Turin. Hmm based on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, 1996.
8. Frederick Jelinek. *Statistical methods for speech recognition*. MIT Press, Cambridge, MA, USA, 1997.
9. M. Möhring, C. Lessig, and O. Bimber. Optical tracking and video see-through ar on consumer cell phones. In *Proc. of Workshop on Virtual and Augmented Reality of the GI-Fachgruppe AR/VR*, pages 193–204, 2004.
10. J.M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, Dec. 1995.
11. M. Petkovic, W. Jonker, and Z. Zivkovic. Recognizing Strokes in Tennis Videos Using Hidden Markov Models. In *Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing*, Marbella, Spain, 2001.
12. Lawrence R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
13. M. Rohs. Real-world interaction with camera-phones. In *2nd International Symposium on Ubiquitous Computing Systems*, pages 39–48, 2004.
14. P. Sangi, J. Heikkilä, and Silvén O. Motion analysis using frame differences with spatial gradient measures. In *Proc. International Conference on Pattern Recognition, vol. 4*, pages 733–736, 2004.
15. C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.
16. J. Shi and C. Tomasi. Good features to track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.
17. A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions Information Theory*, IT-13:260–269, 1967.