Motion-Based Handwriting Recognition for Mobile Interaction

Jari Hannuksela, Pekka Sangi and Janne Heikkilä
Machine Vision Group, Infotech Oulu
Department of Electrical and Information Engineering
P.O. Box 4500, FIN-90014 University of Oulu, Finland
{jari.hannuksela,pekka.sangi,janne.heikkila}@ee.oulu.fi

Abstract

This paper presents a new interaction technique for camera-enabled mobile devices. The handheld device can be used for writing just by moving the device. In our method, interframe dominant motion is estimated from images, and the discrete cosine transform is used for computing discriminating features from motion trajectories. The k-nearest neighbor rule is applied for classification. A real-time implementation of the method was developed for a mobile phone. In experiments, recognition rates ranging from 92 % to 98 % were achieved, which testifies to the practicality of our approach.

1. Introduction

User interfaces of hand-held mobile devices are typically based on keypads or touch-sensitive panels operated with a pen. Interaction using such approaches can sometimes be cumbersome. For example, the number of keys on small keypads is limited, and several presses may be required for desired outcome. In addition, with touch-sensitive panels, both hands are needed for operation. For these reasons, it is interesting to consider other modalities for interaction.

A natural way for interacting with a mobile device is to move it and use information about motion, obtained by some means, to control the device. In devices equipped with a camera, successive images obtained from a camera can be used as a source for motion data. Such solutions have been proposed recently [2, 3], and they mainly focus on browsing and navigation on the display. However, the motion input combined with pattern recognition techniques can be used for more advanced interaction purposes like recognizing handwriting, gestures and signs.

We present a new user interaction technique where the handheld device can be used for writing letters and digits just by moving the device. In order to make writing easier and faster only single isolated strokes are considered. The character models are similar to the Graffiti$^{\mathrm{TM}}$like alphabet used in the Palm$^{\mathrm{TM}}$devices. Single stroke characters also simplify the recognition task and make it more reliable than using ordinary characters. A possible shortcoming is that users have to learn a special way of writing characters [5].

A large number of different methods for on-line handwriting recognition has been proposed in the literature [7]. Dynamic time warping (DTW) is among the most popular solutions for recognizing handwritten characters [10]. However, it is a computationally quite an expensive approach for a mobile phone, where the computational resources are limited. Fourier descriptors (FD) have also been used for recognizing shapes of closed boundaries [9]. FDs can be made invariant to certain geometric transformations, which is useful in some applications. In this paper, we present a novel method for recognizing characters based on discrete cosine transform (DCT). This solution is closely related to FDs but instead of using complex numbers we can compute the features with real numbers. Another benefit is that we can also classify shapes that are not closed curves. This is important in character recognition where strokes have different start and end points. The rest of the paper is organized as follows. In Section 2, we present our handwriting recognition system. Experiments are reported in Section 3. Section 4 summarizes the contributions of the paper.

2. The recognition system

The handwriting recognition is carried out in four steps. First, interframe motions of the device are estimated and collected, when the user writes the character. Each handwritten character sequence is stored as a sequence of (x, y)-coordinates. The obtained motion trajectories are then resampled. After that, a set of DCT based features are computed for motion trajectories. Finally, the character samples are classified according to the k nearest neighbors (k-NN) rule.

2.1. Motion estimation

In this section, our method for estimating the global motion of the device is briefly reviewed. A more detailed description can be found in [2]. Global motion refers here to the apparent dominant 2-D motion between frames, which can be approximated by some parameterized flow field model [4, 6]. Our method for estimating such models has two main phases. In the first phase, the motion of the selected features and related uncertainty is analyzed, and in the second phase, the results are used for obtaining parametric global motion estimates.

In order to distribute features over the image, the central image region is divided into smaller rectangular subregions, and one feature block is selected from each region (see Fig. 1). For each candidate block, the spatial gradient is evaluated in horizontal and vertical directions, and a gradient measure is evaluated using this information. The block which maximizes the measure is selected.

For a selected block, exhaustive evaluation of a sum of squared differences (SSD) block matching measure is performed for some range of displacement candidates which provides a surface of SSD values; this we call the *motion profile*. The candidate that minimizes the SSD measure is taken as the displacement estimate. Uncertainty of this estimate is analyzed by performing gradient-based thresholding for the motion profile. Moments of the thresholding results provide a 2×2 covariance matrix (error ellipses), which represents the local motion uncertainty (Fig. 1).

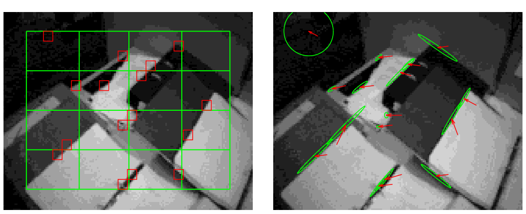

Figure 1. Motion estimation principle. Left: feature selection, right: local motion features.

After feature motion estimation, a voting-based outlier analysis is performed in order to determine a maximal subset of features which represent some common motion. The principle is based on random sampling consensus, that is, local motion features are used for generating hypotheses about global dominant motion, and all features vote for generated hypotheses. Local motion uncertainty information is taken into account in this phase.

Once inlier features have been selected, a parametric global motion model fitting is performed. We use a four-parameter similarity motion model which represents displacement $\boldsymbol{v}$ at image coordinate $\boldsymbol{p}$ according to

$$\boldsymbol{v} = \begin{bmatrix} 1 & 0 & x & y \\ 0 & 1 & y & -x \end{bmatrix} \boldsymbol{\theta} \qquad (1)$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4]^T$ is a vector of model parameters. In model fitting, uncertainty covariances associated with the inlier block motions are taken into account and used for computing a weighted estimate. The results of global motion estimation are passed to the filtering stage, where a Kalman filter is used for computing smoothed motion estimates. Finally, the motion information is integrated in order to obtain a sequence of (x,y)- coordinate points that represents the handwritten character sample.

2.2. Resampling

The motion trajectory is first preprosessed in order to prepare handwriting data for feature extraction and classification. In our system, only two steps need to be carried out. Firstly, those points whose (x,y)- coordinates are exactly the same as those of the preceding points are removed. Secondly, the number of points varies depending on speed and personal style of the writing, and the captured points are not equidistant in space. Therefore, we resample data so that each coordinate sequence has N equidistant points. This enables motion trajectories of variable lengths to be compared in the frequency domain.

2.3. Feature extraction

Once the handwriting sequence of (x,y)- coordinates has been resampled, we compute discriminative features using the discrete cosine transform (DCT). The DCT is widely used in compression applications because the energy of the input signal after transformation is packed into a few coefficients [8]. Actually, it is a close approximation to the Karhunen-Loeve transform [9]. According to our knowledge, the DCT has not been used for handwritten character recognition before. It is a reasonable choice for our system, because motion trajectories are already filtered in the motion estimation step and high frequencies are suppressed. Therefore it is natural to use only a few DCT coefficients (for example 9 in our experiments), that represent low frequencies, to describe the shape of handwritten characters. Furthermore, there are many fast algorithms for implementation.

Let $\boldsymbol{C_N}$ be an N by N matrix containing the basis vectors of the DCT. These basis vectors are defined by

$$c_{kn} = \alpha(k) cos\left(\frac{(2n+1)k\pi}{2N}\right), \qquad (2)$$

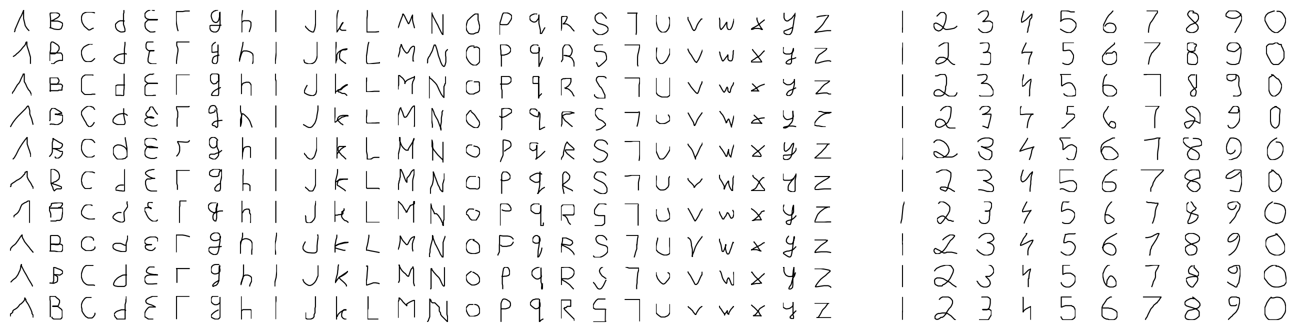

Figure 2. Examples of letter and digit samples.

where $n = 0, 1, ..., N-1$ and

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}}, & k = 0 \\ \sqrt{\frac{2}{N}}, & k = 1, 2, ..., N-1 \end{cases}.$$

We compute N by 2 point 2-D DCT using the following equation

$$\boldsymbol{Q} = \boldsymbol{C_N} \boldsymbol{P} \boldsymbol{C_2}^T, \quad (3)$$

where

$$\boldsymbol{P} = \begin{bmatrix} x_1 & x_2 & ... & x_N \\ y_1 & y_2 & ... & y_N \end{bmatrix}^T.$$

contains the coordinates of the resampled motion trajectory. In our experiments $N = 32$. The feature vector used is defined as

$$\left[\frac{|q_{10}|}{a}, \frac{q_{20}}{a}, \frac{q_{21}}{a}, \frac{|q_{30}|}{a}, \frac{|q_{31}|}{a}, \frac{q_{40}}{a}, \frac{q_{41}}{a}, \frac{|q_{50}|}{a}, \frac{|q_{51}|}{a}\right],$$

where q_{ij} is the element of $\boldsymbol{Q}$ at row i and column j, and

$$a = \sqrt{q_{10}^2 + q_{11}^2}.$$

Translation independence is achieved by ignoring the coefficients q_{00} and q_{01}, which represent the average values of the signals. In order to make features independent of the character size, the other coefficients are normalized by factor a. In addition, we take the absolute value of every odd coefficient, which provides invariance to the starting and end points of the trajectories, that is, to the drawing direction of the stroke.

2.4. Classifier

The classifier used in our system is based on the k Nearest Neighbors (k-NN) rule. This decision rule has been extensively used in pattern recognition systems because of its good performance and simple algorithm. In k-NN, unknown samples are classified by counting the labels of k-closest training samples (prototypes) according to some similarity measure such as Euclidean distance [1]. This rule has nice properties: 1) the recognition error rate approaches twice the Bayesian error rate as the number of prototypes and the value of k becomes large, 2) the classifier can still be designed even if training samples are few and 3) it can be implemented when classes overlap with each other [1].

We use the easiest implementation of the k-NN rule when Euclidean distance between the sample and each prototype in the training set is computed, and then the sample is classified into the majority class of its k- nearest neighbors. This exhaustive search is suitable for our method due to the small number of training samples.

3. Experiments

Our recognition system was evaluated on a Nokia 7610 smartphone. The platform is based on Series 60 with Symbian 6.1 OS. It contains a 123 MHz ARM-based 32-bit RISC CPU without a floating-point unit, a VGA camera and 3.4 Mb shared memory. The global motion estimation method [2] implemented earlier for this platform was utilized to collect experimental data. The frame rate of this implementation is 10 fps.

A total of 10 test subjects took part in our experiments. None of them had previously used our system. Each subject was asked to write 10 digits and 26 letters twice. Thus, two sets of digits and letters were obtained. Fig. 2 show a set of digits and letters that the subjects wrote in the experiments. The average duration to write digits and letters was around 1.5 seconds. However, it should be noted that the writing speed improved considerably during the test.

3.1. Experiment #1

To study the recognition accuracy of our system, we first selected randomly 10 training samples for each class. The other 10 samples for each class were used for evaluating the recognition rate. The random selection was repeated 20 times for both digits and letters in order to get statistics. For k-NN classification, only values $k = 1$ and $k = 3$ were used due to the small amount of training data available. Table 1 shows mean recognition rates and associated standard deviations obtained in the experiment. Performance is quite good. Most classification errors occur with similar prototypes such as 'G' and 'Y', and with more distinctive prototypes, the recognition rate can be improved. Another observation is that random selection of training samples does not guarantee that the training set contains samples from each invidual, which can decrease the recognition rate. This was the motivation for our second experiment.

Table 1. Results with experiment #1.

	Digits [%]		Letters [%]	
	Mean	Std	Mean	Std
k = 1	94.5	2.3	93.3	1.4
k = 3	93.9	2.6	92.0	2.0

3.2. Experiment #2

In this experiment, we selected the training set so that it contained one sample from each test subject for all digits/letters. In this way, we can deal with the variations of the writing style of different inviduals. The total number of samples in the training set and the test set was 100 in the experiment with digits. In the letter case, the total number of samples was 260 for both sets. The results shown in Table 2 indicate that recognition accuracies are above the average values given in Table 1. This shows that the recognition accuracy can be increased during the normal use of a device by adaption. For instance, the user can add new character samples to the training set to personalize the recognition system to his or her writing style.

Table 2. Results with experiment #2.

	Digits [%]	Letters [%]
k = 1	97.0	94.0
k = 3	98.0	93.0

4. Conclusions

We have proposed a novel interaction technique for camera-enabled mobile devices. The handheld device can be used for handwriting just by moving the device. In our method, interframe dominant motion is estimated from images, and the discrete cosine transform is used for computing discriminating features from motion trajectories. The k-nearest neighbor rule is applied for classification. A real-time implementation of the method was developed for a mobile phone. In experiments, recognition rates ranging from 92 % to 98% were achieved. The recognition accuracy can still be improved using a personal training set. However, we can already say that our solution provides a viable alternative for mobile intercation. The usability of our technique could be improved by minimizing the magnitude of the motion needed to write the characters, and allowing more flexibility in moving the device, which is one topic for future work. The use of motion information could also be extended to other purposes like recognizing gestures and signs.

Acknowledgments

The financial support of the Academy of Finland (project no. 110751), the National Technology Agency of Finland and the Nokia Foundation is gratefully acknowledged.

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[2] 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.

[3] 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.

[4] H. Haußecker and H. Spies. Motion. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook of Computer Vision and Applications, Volume 2*, chapter 13. Academic Press, San Diego, 1999.

[5] I. S. MacKenzie and R. W. Soukoreff. Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*, 17:147–198, 2002.

[6] J. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. 6(4):348–365, Dec. 1995.

[7] R. Plamondon and S. N. Srirari. On-line and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):63–84, January 2000.

[8] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 1999.

[9] O. D. Trier, A. K. Jain, and T. Taxt. Feature extraction methods for character recognition - a survey. *Pattern Recognition*, 29(4):641–662, 1996.

[10] V. Vuori, M. Aksela, J. Laaksonen, E. Oja, and J. Kangas. Adaptive character recognizer for a hand-held device: implementation and evaluation setup. In *Proc. of the 7th International Workshop on Frontiers in Handwriting Recognition*, pages 13–22, 2000.