A Fast HMM Algorithm Based on Stroke Lengths for On-line Recognition of Handwritten Music Scores

Youichi Mitobe

TOSYS Corporation

1108-5 Higashiyama Aza, Watauchi, Wakaho,

Nagano, 381–0913 JAPAN

Hidetoshi Miyao and Minoru Maruyama

Department of Information Engineering, Faculty of Engineering, Shinshu University

4-17-1 Wakasato, Nagano, 380–8553 JAPAN

{miyao, maruyama}@cs.shinshu-u.ac.jp

Abstract

The Hidden Markov Model (HMM) has been successfully applied to various kinds of on-line recognition problems including, speech recognition, handwritten character recognition, etc. In this paper, we propose an on-line method to recognize handwritten music scores. To speed up the recognition process and improve usability of the system, the following methods are explained: (1) The target HMMs are restricted based on the length of a handwritten stroke, and (2) Probability calculations of HMMs are successively made as a stroke is being written. As a result, recognition rates of 85.78% and average recognition times of 5.19ms/stroke were obtained for 6,999 test strokes of handwritten music symbols, respectively. The proposed HMM recognition rate is 2.4% higher than that achieved with the traditional method, and the processing time was 73% of that required by the traditional method.

Key words: HMM, Handwritten Music Score Recognition, On-line Symbol Recognition

1. Introduction

Current composers and arrangers record music by handwriting music symbols on sheets of paper. However, it would be desirable to convert them into computer-readable data so that they could be easily edited and allow other functions, such as printing, automatic performance, division into parts, and modulation.

In order to allow automatic conversions, many off-line handwritten music recognition systems have been proposed [1, 2, 3, 4]. However, their recognition rates are insufficient. Therefore, they are impractical because it takes too long to correct the errors. On-line systems that can recognize handwritten music could improve the recognition rates since various features, such as pen pressure, pen angle, and pen stroke, which are not easily obtained with off-line systems, could be utilized [5].

Hidden Markov Model (HMM) methods have been applied to speech recognition [6]. They have been also applied to the recognition of on-line handwritten alphabet characters [7, 8, 9], hiragana [10], and kanji [11]. The system copes with shape distortions in handwritten characters, and good recognition rates can be obtained. However, these methods have the following disadvantages:

- As the number of classifications increases, so does the processing time. This is because the HMM must calculate probabilities for every possible class.

- In general, the probabilities are not calculated while a stroke/character is being written, but they are done after the stroke/character has been written. Therefore, the probability calculations have to be done during the time between the end point of one stroke or character and the start of next another (this situation means that the pen is lifted from the pad). In other words, the user has to wait without writing symbols until the calculations are done.

The issues described above would affect the system usability.

The following steps can be used to increase the speed of the recognition process and improve the system usability:

- Target HMMs are restricted based on the length of a handwritten stroke.

IEEE
COMPUTER
SOCIETY

- Probability calculations of HMMs are successively made as a stroke is being written.

This method is applied to recognize a stroke for each music symbol.

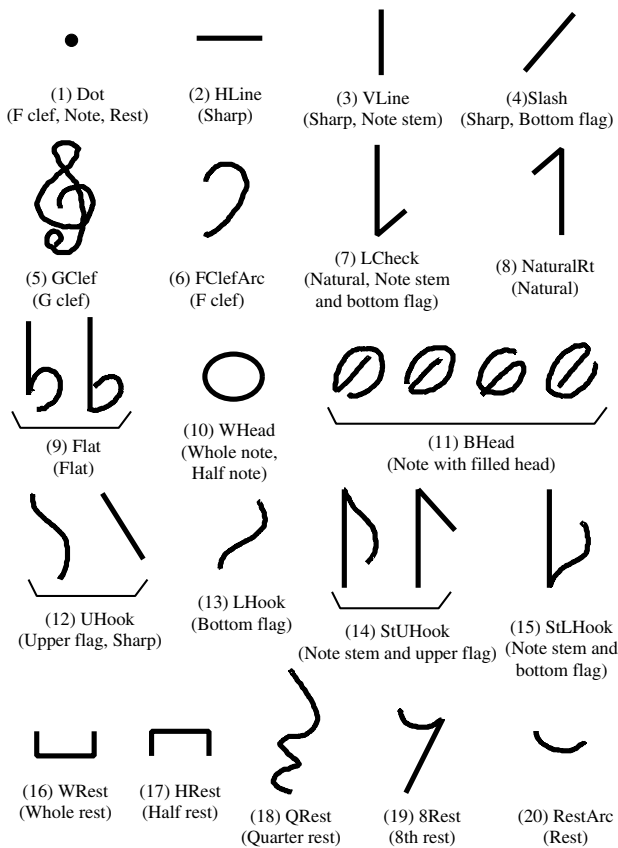## 2. Input of strokes of each music symbol

Figure 1. Individual strokes for music symbols.

Unlike alphabets or Japanese characters, a certain type of music symbols vary widely in shape. In particular, the combinations of shapes in music notes are huge since the number of note heads, dots, and flags included in a note is not limited. It is impossible to prepare HMMs for all kinds of music symbols. Therefore, the system proposed here deals with the individual strokes of a music symbol, which are then combined to make one.

The strokes that can be recognized by the proposed system are shown in Fig.1. The shapes are similar to those used for handwritten music symbols, with the exception of the filled-note head (Fig.1 (11)). We suppose that each stroke in Fig.1 is written with one stroke.

With this approach, the following music symbols can be generated: G clef, F clef, sharp, flat, natural, musical notes (except for the beamed notes), and rest symbols. The names of music symbols to which the stroke can belong are listed below the stroke names in Fig.1. The strokes can be combined in Miyao's method [5].

In this paper, we only discuss about the recognition of each stroke.

3. Hidden Markov Model

3.1. Input feature for HMM

The input stroke is divided with a spatially equal interval, and the divided segments are described by the 8-direction Freeman Chain Code, as shown in Fig.2 (a). Here, the length of a stroke is assumed to be the number of divided segments. For example, the stroke shown in Fig.2 (b) is approximately represented by the symbol sequence $\{1, 2, 1\}$, and its stroke length is 3. The sequence is fed to HMMs as input.
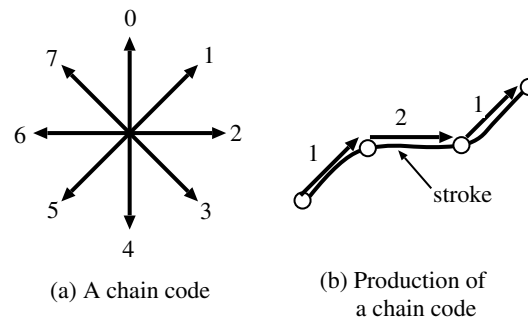

(a) A chain code (b) Production of a chain code

Figure 2. 8-direction Freeman Chain Code.

3.2. Structure of an HMM

An HMM consists of some states and arcs, which mean transitions between the states, as shown in Fig.3. An HMM is represented by the following parameters, $\lambda = (A, B, \pi)$:

$A = \{a_{ij}\} \quad 1 \le i, j \le N$,

a_{ij}: state-transition probability distributions from states S_i to S_j,

$B = \{b_{ij}(k)\} \quad 1 \le i, j \le N \quad k = 0, 1, \cdots, M - 1$,

b_{ij}: output probability of symbol k when the states transit from S_i to S_j,

$\pi = \{\pi_i\} \quad 1 \le i \le N$,

π_i: probability that the initial state is S_i,

where N means the number of states and M means the number of symbol types. Assuming that the parameter λ

is given and the symbol sequence $O = O_1 O_2 \cdots O_T$ is observed, $P(O|\lambda)$, which is the output probability of O for the HMM: λ, can be estimated.

In this system, a left-to-right type HMM, shown in Fig.3, is used with 5-states (i.e. $N = 5$). The states S_1, S_2, S_3, and S_4 have a self-loop transition, and the state S_5 does not. The number of symbol types, M, is 8. This value coincides with the number of the chain code directions. The initial state and the final state are S_1 and S_5, respectively.
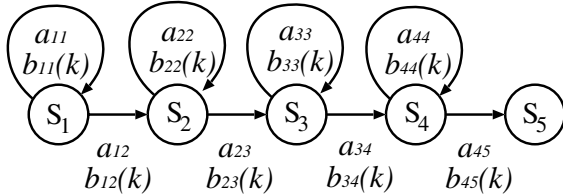

Figure 3. Left-to-right type of the Hidden Markov Model.

3.3. Learning HMM

If HMM: $\lambda = (A, B, \pi)$ is given and the symbol sequence O is observed, the following parameter, $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$, can be re-estimated (Baum-Welch algorithm [6]):
$$P(O|\bar{\lambda}) \geq P(O|\lambda).$$
For a set of multiple symbol sequences, $\boldsymbol{O} = \{O^{(1)}, O^{(2)}, \cdots, O^{(K)}\}$, the following probability is also calculated:
$$P(\boldsymbol{O}|\lambda) = \prod_{k=1}^{K} P(O^{(k)}|\lambda),$$
where the parameter λ is estimated so that the above probability can have the maximum value. HMMs for all classes are constructed using this algorithm. Therefore, in this system, 20 HMMs are built corresponding to the 20 strokes shown in Fig.1.

3.4. Recognition process

Using the learned HMMs, strokes are recognized. For a symbol sequence $O = O_1 O_2 \cdots O_T$, which is produced from an unlearned input stroke, the probability $P(O|\lambda)$ is estimated for all HMMs, in which case the HMM with the maximum probability can be selected. The class corresponding to the selected HMM is adopted as the recognition result.

The above probabilities are calculated effectively by the Viterbi algorithm[6].

4. Improvement of the recognition process

Let $L_n(c)$ and $L_x(c)$ be the minimum and maximum lengths of the symbol sequences for the class c, respectively. The lengths can be obtained from training data, which belong to the class c. In this case, assume that the length of the symbol sequence for the class c of unlearned data is also within the range between $L_n(c)$ and $L_x(c)$. This assumption indicates that target HMMs can be restricted based on the length of a handwritten stroke.

The length of the stroke is not known until the symbol has been written. To apply our idea even to the on-line recognition, the following method is used:

Initialization:
```
for c ← 1 to C {
    flag[c] ← SLEEP
}
```

Processing for symbol sequence:
```
t ← 1
while(O[t] ← input symbol, and O[t] exists){
    for c ← 1 to C {
        if(flag[c] ≠ DEAD){
            if(t < Ln[c]) Do nothing # the situation (1) in Fig.4
            else if(t = Ln[c]){ # the situation (2) in Fig.4
                The symbol sequence O={O[1],O[2],···,O[t]} is
                given to HMM[c], and the probability is calculated
                by the Viterbi algorithm [6].
                flag[c] ← ALIVE
            }
            else if(t ≤ Lx[c]) { # the situation (3) in Fig.4
                O[t] is added to HMM[c], and the probability is
                calculated.
            }
            else{ # the situation (4) in Fig.4
                flag[c] ← DEAD
            }
        }
    }
    t ← t+1
}
```
where the above variables are defined as follows:
t: time(stroke length)
O[t]: t-th input symbol(t-th chain code)
c: class
C: total number of classes
HMM[c]: HMM for class c
[Ln[c], Lx[c]]: range of the stroke lengths for class c
flag[c]: current situation of HMM[c]; the situation is
defined as follows:
  SLEEP: untreated
  ALIVE: in progress

DEAD: excluded from the process

The transition of the situation is demonstrated in Fig.4.

After the processes described above are completed, the probabilities for all HMMs that have the state 'ALIVE' are compared. As a result, the HMM with the maximum probability is selected, and the class corresponding to the selected HMM is adopted as a recognition result.
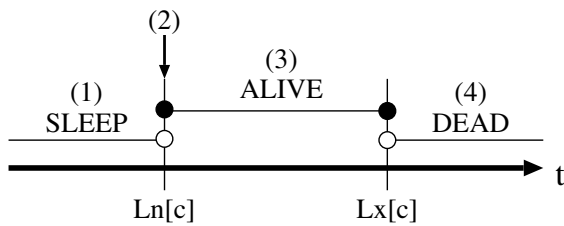

Figure 4. State of variable flag[c].

5. Experimental results and discussions

In the experiments, a PC (Pentium 3 CPU: 1.0GHz; 256MB Memory) and an A6 size tablet were used. To build the HMMs, 6,858 strokes (there are about 350 strokes for each class), written by 6 users, were used. Furthermore, the ranges of the stroke lengths for each class were determined using these strokes. The obtained ranges are shown in Table 1 as the minimum length L_n and the maximum length L_x. We apply both the proposed method and a method based on ordinary HMMs to 6,999 strokes written by 5 other users. In Table 1, we show the average processing time of the both methods.

From these results, the proposed method has reduced the average processing time for each stroke from 7.10ms in the ordinary HMM to 5.19ms. In particular, the processing times for the strokes Dot, HLine, and GClef were 13%, 43%, and 19% of those required by the ordinary HMM method, respectively. It is considered that each of these strokes has a length that is far different from the other strokes.

For the recognition process for the short strokes Dot and HLine, the calculations of probabilities were eliminated for the HMM[c] that corresponds to the longer stroke (in the strict sense, Ln[c] is longer than the length of the input stroke Dot or HLine in this case). On the other hand, for the recognition process for the long stroke GClef, the calculations of probabilities for the HMM[c] that corresponds to the shorter stroke (in this case, Lx[c] is shorter than the length of stroke GClef) were only calculated until the length of the input stroke was equal to Lx[c]. These reductions of the calculations contribute to a shorter processing time.

The recognition rates for the test strokes are shown in Table 2, where the 'Top 3 Recognition Rate' means the cumulative recognition rate from the best 3 candidates. From these results, the recognition rate for the proposed method is about 2.4% higher than that for the ordinary HMM method. It is considered that the restriction of the target HMMs based on the stroke lengths also contributes to improve the recognition rate. However, the rate was 85.78%, which is not high enough. The reasons for the low recognition rate are as follows:

- The input feature was poor. More features, such as pen pressure and pen angle, are needed to obtain a good recognition rate.

- Parameter settings, such as the number of states and the initial states of a_{ij} and b_{ij} for each HMM, were not satisfactorily adjusted.

- The number of the training data was not enough to get well-built HMMs.

On the other hand, the top 3 recognition rate was 97.06%, which is high enough because the recognition errors could be automatically corrected by using the three candidates in the combination process of the strokes.

Table 2. Recognition rate.

	Ordinary HMM	Proposed HMM
Recognition Rate	83.42%	85.78%
Top 3 Recognition Rate	95.56%	97.06%

6. Conclusion

For an HMM-based system for the recognition of handwritten music scores, the following steps are proposed to increase the speed of the recognition process and improve the system usability:

- Target HMMs are restricted based on the length of a handwritten stroke.

- Probability calculations of HMMs are successively made as a stroke is being written.

When this method was applied to recognize 6,999 test strokes, recognition rates of 85.78%, top 3 recognition rates of 97.06%, and average processing times of 5.19ms/stroke were obtained. The recognition rate of this method is 2.4% higher than that of the traditional method, and the processing time is 73% of that required by the traditional method. In conclusion, the method can reduce the processing time

Table 1. Average processing time for each stroke.

Stroke Name	#samples	Length Range		Ordinary HMM A [ms]	Proposed HMM B [ms]	Ratio B/A
		L_n	L_x			
Dot	365	2	22	0.60	0.08	0.13
HLine	351	5	33	2.11	0.91	0.43
RestArc	349	14	46	3.07	2.55	0.83
HRest	338	27	54	4.00	3.94	0.99
Slash	356	9	71	2.92	2.28	0.78
LHook	348	30	86	5.88	5.76	0.98
NaturalRt	351	23	90	5.90	5.74	0.97
WRest	352	27	97	5.11	4.95	0.97
UHook	346	12	99	4.54	4.48	0.99
8Rest	352	25	101	5.83	5.52	0.95
VLine	349	10	104	4.79	4.74	0.99
WHead	350	33	108	6.78	6.61	0.97
BHead	351	49	121	8.13	7.96	0.98
Flat	351	44	125	8.47	7.56	0.89
QRest	352	59	127	9.10	8.07	0.89
StLHook	349	41	164	9.38	7.34	0.78
LCheck	349	25	176	7.32	5.88	0.80
StUHook	349	42	185	9.21	7.09	0.77
FClefArc	349	32	203	10.13	6.91	0.68
GClef	342	163	472	28.65	5.40	0.19
Total	6,999	Average		7.10	5.19	0.73

and improve the recognition rates. In particular, the recognition time decreases drastically for a stroke, the length of which is far different from the stroke lengths of the other classes.

With this method, the processing time from the end of one stroke to the beginning of the next can be shortened since the probabilities for HMMs are calculated as the stroke is being written. In other words, a stroke can be recognized as soon as the pen is lifted from the pad.

In future studies, the recognized strokes will be combined as a music symbol used by a method [5], and usability of the system will be evaluated.

References

[1] A. Bulis, R. Almog, M. Gerner and U. Shimony, "Computerized Recognition of Hand-Written Musical Notes", *Proc. of International Computer Music Conference*, pp.110–112, 1992.

[2] K. Ng, D. Cooper, E. Stefani, R. Boyle and N. Bailey, "Embracing the Composer: Optical Recognition of Handwritten Manuscripts", *Proc. of International Computer Music Conference*, pp.500–503, 1999.

[3] G. Watkins, "A Fuzzy Syntactic Approach to Recognising Hand-Written Music", *Proc. of International Computer Music Conference*, pp.297–302, 1994.

[4] O. Yadid-Pecht, M. Gerner, L Dvir, E Brutman and U. Shimony, "Recognition of Handwritten Musical Notes by a Modified Neocognitron", *Machine Vision and Applications*, 9 (2), pp.65–72, 1996.

[5] H. Miyao and M. Maruyama, "An Online Handwritten Music Score Recognition System", *Proc. of International Conference on Pattern Recognition (ICPR2004)*, (to appear), 2004.

[6] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proc. of the IEEE*, 77 (2), pp.257–286, 1989.

[7] W. G. Aref, P. Vallabhaneni, and D. Barbará, "On Training Hidden Markov Models for Recognizing Handwritten Text", *Proc. of the International Workshop on Frontiers in Handwriting Recognition (IWFHR-4)*, pp.275–284, 1994.

[8] J. Hu, M. K. Brown, and W. Turin, "HMM-Based On-Line Handwriting Recognition", *IEEE Transactions*

on Pattern Analysis and Machine Intelligence, 18 (10), pp.1039–1045, 1996.

[9] S. Gunter and H. Bunke, "A New Combination Scheme for HMM-based Classifiers and Its Application to Handwriting Recognition", *Proc. of the International Conference on Pattern Recognition (ICPR2002)*, (II), pp.332–337, 2002.

[10] T. Nonaka and S. Ozawa, "Online Handwritten HIRA-GANA Recognition Using Hidden Markov Models", *IEICE Transactions on Information and Systems*, J74-D-II (12), pp.1810–1813, 1991. (in Japanese)

[11] J. Tokuno, N. Inami, S. Matsuda, M. Nakai, H. Shimodaira, and S. Sagayama, "Context-dependent Sub-stroke Model for HMM-based On-line Handwriting Recognition", *Proc. of the International Workshop on Frontiers in Handwriting Recognition (IWFHR-8)*, pp.78–83, 2002.

IEEE
COMPUTER
SOCIETY