IJDAR (2010) 13:19–31
DOI 10.1007/s10032-009-0100-1

ORIGINAL PAPER

Optical recognition of music symbols

A comparative study

A. Rebelo · G. Capela · Jaime S. Cardoso

Received: 13 January 2009 / Revised: 12 June 2009 / Accepted: 17 October 2009 / Published online: 17 November 2009

Abstract Many musical works produced in the past are still currently available only as original manuscripts or as photocopies. The preservation of these works requires their digitalization and transformation into a machine-readable format. However, and despite the many research activities on optical music recognition (OMR), the results for handwritten musical scores are far from ideal. Each of the proposed methods lays the emphasis on different properties and therefore makes it difficult to evaluate the efficiency of a proposed method. We present in this article a comparative study of several recognition algorithms of music symbols. After a review of the most common procedures used in this context, their respective performances are compared using both real and synthetic scores. The database of scores was augmented with replicas of the existing patterns, transformed according to an elastic deformation technique. Such transformations aim to introduce invariances in the prediction with respect to the known variability in the symbols, particularly relevant on handwritten works. The following study and the adopted databases can constitute a reference scheme for any researcher who wants to confront a new OMR algorithm face to well-known ones.

Keywords Music · Performance evaluation · Symbol recognition · Document image processing · Off-line recognition

1 Introduction

Music, from Greek *μφυσικήτέχνη–musiké (téchne)*, which means the art of the muses, can be defined as an organized sequence of sounds and silences so as to produce aesthetic pleasure in the listener. There are evidences, by pictographs, that music is known and practiced since prehistory. Over the years, music has expanded in many several music styles and for many different purposes, like educational or therapy. The centrality of music in the cultural heritage of any society and the importance of cultural diversity, as necessary for humankind as biodiversity is for nature, makes policies to promote and protect cultural diversity an integral part of sustainable development.[1]

Portugal, like many other countries, has a notorious lack in music publishing from virtually all eras of its musical history. In spite of most of the original manuscripts of music known before the twentieth century being kept in the national library in Lisbon, there is not any repository of musical information from the last century. Although there are recent efforts to catalogue and to preserve in digital form the Portuguese music from the twentieth century—notably the Music Information Center[2] and the section on musical heritage from the Institute

This work was partially funded by Fundação para a Ciência e a Tecnologia (FCT), Portugal through project PTDC/EIA/71225/2006.

A. Rebelo (✉)
INESC Porto, Faculdade de Ciências, Universidade do Porto, Porto, Portugal
e-mail: arebelo@inescporto.pt

G. Capela · J. S. Cardoso
INESC Porto, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal

G. Capela
e-mail: artur.capela@fe.up.pt

J. S. Cardoso
e-mail: jaime.cardoso@inescporto.pt

[1] http://www.unesco.org/bpi/eng/unescopress/2001/01-112e.shtml.

[2] http://www.mic.pt.

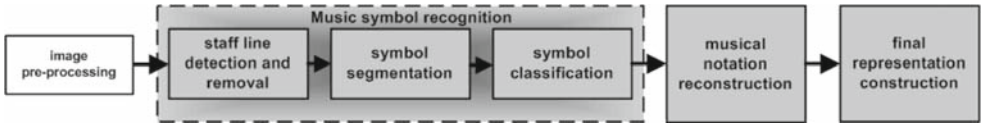


Fig. 1 Typical architecture of an OMR processing system

of the Arts website[3]—most of the music predating computer notation software was never published and still exists in the form of manuscripts or photocopies spread out all over the country in discreet places. The risk of irreversibly losing this rich cultural heritage is a reality that must be taken seriously and dealt with accordingly.

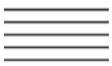
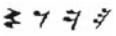
Digitization has been commonly used as a possible tool for preservation, offering easy duplications, distribution and digital processing. However, transforming the paper-based music scores and manuscripts into a machine-readable symbolic format (allowing operations such as search, retrieval and analysis) requires an optical music recognition (OMR) system. Unfortunately, the actual state of the art of handwritten music recognition is far from providing a satisfactory solution.

After the image preprocessing (application of several techniques, e.g. binarization, noise removal, blurring, deskewing, amongst others, to make the recognition process more robust and efficient), an OMR system can be broadly divided into three principal modules (see Fig. 1):

1. Recognition of musical symbols from a music sheet;
2. reconstruction of the musical information to build a logical description of musical notation;
3. construction of a musical notation model for its representation as a symbolic description of the musical sheet.

The first module is typically further divided into three stages: staff lines detection and removal to obtain an image containing only the musical symbols (staff lines spacing and thickness also provide the basic scale for relative size comparisons); symbols primitives segmentation and recognition. For reference, Table 1 introduces the set of symbols with relevance in this work. The second and third modules (musical notation reconstruction and final representation construction) are intrinsically intertwined. In the module of musical notation reconstruction, the symbols primitives are merged to form musical symbols. Usually, in this step, graphical and syntactic rules are used to introduce context information to validate and solve ambiguities from the previous module of music symbol recognition. Detected symbols are interpreted and assigned a musical meaning. In the third module of final

Table 1 Music notation

Symbols	Description
	**Staff:** An arrangement of parallel lines, together with the spaces between them.
	**Treble, Alto and Bass clef:** The first symbols that appear at the beginning of every music staff and tell us which note is found on each line or space.
	**Sharp, Flat and Natural:** The signs that are placed before the note to designate changes in sounding pitch.
	**Beams:** Used to connect notes in note-groups; demonstrate the metrical and the rhythmic divisions.
	**Accent and Staccatissimo:** Symbols for special or exaggerated stress upon any beat, or portion of a beat.
	**Crochet, Quaver and Minim:** The Crochet (closed notehead) and Minim (open notehead) symbols indicate a pitch and the relative time duration of the musical sound. Flags (Quaver) are employed to indicate the relative time values of the notes with closed noteheads.
	**Quarter, Eighth, Sixteenth and thirty-second rests:** Indicate the exact duration of silence in the music; each note value has its corresponding rest sign; the written position of a rest between two barlines is determined by its location in the meter.
	**Ties and Slurs:** Ties are a notational device used to prolong the time value of a written note into the following beat. The tie appears to be identical to slur, however, while tie almost touches the notehead centre, the slur is set somewhat above or below the notehead. Ties are normally employed to join the time value of two notes of identical pitch; Slurs affect note-groups as entities indicating that the two notes are to be played in one physical stroke, without a break between them.

[3] http://patrimonio.dgartes.pt.

representation construction, a format of musical description is created with the information previously produced.

In this paper, we focus on the step of music symbol recognition. More specifically, we will motivate the adoption of specific algorithms for staff line detection and removal and symbol detection, and we will present a comparative study of methods for the music symbol classification. We also investigate the interest of elastic deformation to extend the set of symbols used to design the classifiers, in an attempt to favour robustness to typical distortions in musical symbols.

2 Related works

The investigation in the OMR field began with Pruslin and Prerau [5]. However, it was only in the 1980s decade, when the equipment of digitalization became accessible that work in this area has expanded [2,5,10,28]. Over the years, several OMR software packages have appeared in the market, but none with a satisfactory performance in terms of precision and robustness. The complexity of the OMR task caused by the bidimensional structure of the musical notation, by the presence of the staff lines and by the existence of several combined symbols organized around the note heads have been hampering the progress in this area. Until now, even the most advanced recognition systems (Notescan in Nightingale, Midiscan in Finale, Photoscore in Sibelius, Smartscore, Sharpeye, etc.) cannot identify all musical notations. Besides that, classic OMR is more focused in regular, printed music sheets; so, a good performance is usually obtained only when processing this type of scores.

2.1 State of the art on staff line detection and removal

The problem of staff line detection is often considered simultaneously with the goal of its removal, although exceptions exist [5,27,30,34]. The importance of these operations lies on the need to isolate the musical symbols for a more efficient and correct detection of each symbol present on the score. When working with printed scores, the staff line detection and removal is completed with high performance; handwritten scores, on the other hand, still represent a challenge. These scores tend to be rather irregular and determined by the authors' own writing style. The handwritten staff lines are rarely straight or horizontal, rarely parallel to each other. Moreover, most of these works are old, and therefore there is a sharp decay in the quality of the paper and ink. Another interesting setting for the comparative study detailed latter is the common modern case where the music notation is handwritten on paper with preprinted staff lines.

The simplest approach for staff line detection consists on finding local maxima in the horizontal projection of the black pixels of the image [4,5,16,26,31,35]. Assuming straight and horizontal lines, these local maxima represent line positions. Several horizontal projections can be made with different image rotation angles, keeping the image in which the local maxima is largest. This eliminates the assumption that the lines are always horizontal. In [9], we have critically overviewed the state of the art in staff line detection and proposed a new staff line detection algorithm, where the staff line is the result of a global optimization problem. The performance was experimentally supported on two test sets adopted for the qualitative evaluation of the proposed method: the test set of 32 synthetic scores from [13], where several known deformations were applied, and a set of 50 real handwritten scores, with ground truth obtained manually.

2.2 State of the art on symbols primitives segmentation and recognition

The process of segmenting the objects from the music score, and the related operation of symbol classification, has long deserved attention from the research community [5,33,35]. Major problems result from the difficulty in obtaining individual meaningful objects. This is typically due to the printing and digitalization, as well as the paper degradation over time. In addition, distortions caused by staff lines, broken and overlapping symbols, differences in sizes and shapes or zones of high density of symbols, contribute to the complexity of the operation. It is also a fact that few research works have been done around handwritten scores [15].

The most usual approach for symbol segmentation consists in extracting elementary graphic symbols, note heads, rests, dots, etc., that can be composed to build musical notation. Usually, the primitives segmentation step is made along with the classification task [5,33,35]; however, exceptions exist [4,16]. Mahoney [5] builds a set of candidates to one or more symbols types and then uses descriptors to select the matching candidates. Carter [5] uses a line adjacency graph (LAG) to extract symbols. The objects resulting from this operation are classified according to the bounding box size, the number and organization of their constituent sections. Other authors [4,16] have chosen to apply projections to detect symbols primitives. The recognition is done using features extracted from the projection profiles. In [16], the k-nearest neighbour rule is used in the classification phase, while neural networks is the classifier selected in [4].

Randriamahefa [31] proposed a structural method based on the construction of graphs for each symbol. These are isolated by using a region-growing method and thinning. Template matching is adopted in [5,26,33,35]. In [33], a fuzzy model supported on a robust symbol detection and template matching was developed. This method is set to deal with uncertainty, flexibility and fuzziness at symbol level. The segmentation process is addressed in two steps: individual analysis of musical symbols and fuzzy model. In the first

step, the vertical segments are detected by a region-growing method and template matching. Then, beams are detected by a region-growing algorithm and a modified Hough transform. The remaining symbols are extracted again by template matching. From this first step results three recognition hypotheses: the fuzzy model is then used to make a consistent decision. The proposed process incorporates graphical and syntactic rules. Besides, it enables the application of learning procedures when potential errors occur, in an effort to gain robustness.

Other techniques for extracting and classifying musical symbols include rule-based systems to represent the musical information, a collection of processing modules that communicate by a common working memory [5] and pixel tracking with template matching [35]. Toyama [35] checks for coherence in the primitive symbols detected by estimating overlapping positions. This evaluation is done with music writing rules. Coüasnon [10,11] proposed a recognition process entirely controlled by grammar which formalizes the musical knowledge. In [32], the segmentation process involves three stages: line and curves detection by LAGs, accidentals, rests and clefs detection by a character profile method and note heads recognition by template matching. The contextual recognition is done by graph grammars. In [30], the segmentation task is based in hidden Markov models.

Despite the wide variety of suggested recognition techniques, a comparative evaluation of their merit has not yet been attempted. This work tries to address this void by presenting a quantitative comparison of different methods for the classification phase. The experimental work is conducted over a large data set of scores, consisting both of real handwritten music scores and synthetic scores to which known deformations can be applied. In Sect. 3, the algorithms selected for staff line detection, removal and symbol detection are presented and motivated. The algorithms for symbol classification under comparison are presented in Sect. 4. The data set adopted in the experiments is detailed in Sect. 5. In Sect. 6, we present the experimental results obtained with the algorithms under evaluation in this comparative study. Finally, conclusions are drawn and future work is outlined in Sect. 7.

3 Preliminary operations

The stage of recognition of musical primitives is often preceded with the elimination of the staff lines and the segmentation of the musical primitives. The reasons for doing that lie on the need to isolate the musical symbols for a more efficient and correct detection of each symbol present on the score. As these operations are not the focus of our study, we start by describing the algorithms that were selected for these operations. Most of the algorithms described in the following subsections require an estimate of the staff space height, *staffspaceheight* and staff line height, *stafflineheight*. Robust estimators are already in common use and were therefore adopted: the technique starts by computing the vertical run-lengths representation of the image. If a bit-mapped page of music is converted to vertical run-length coding, the most common black-run represents the staff line height and the most common white-run represents the staff space height [16].

Fig. 2 An illustrative example of the staff line removal **a** music score with staff lines, **b** music score without staff lines

3.1 Staff line detection

In [8,9], we presented a new and robust staff line detection algorithm based on a stable path approach. In a brief explanation, the proposed paradigm uses the image as a graph and considers a staff line as a connected path from the left margin to the right margin of the music score. As staff lines are almost the only extensive black objects on the music score, the path to look for is the shortest path between the two margins if paths (almost) entirely through black pixels are favoured.

When compared with state-of-the-art solutions, the proposed algorithm performed significantly better on the conducted experiments. The performance of our approach was almost independent of the intensity of the deformations present on the scores. Moreover, differently from many other algorithms, it does not require as input the number of lines per staff.

3.2 Staff line removal

Although the presence of staff lines is helpful for providing a vertical coordinate system for musical primitives, it is also an obstacle. Staff lines usually connect several independent symbols, can fill empty symbol regions and completely cover other symbol features. For that reason, almost all OMR systems eliminate staff lines before the recognition phase, as illustrated in Fig. 2.

The adopted staff line removal algorithm is a modified version of the line track height algorithm presented on [31]. In [9], we conducted a series of experiments, comparing existing versions of staff line removal algorithms with modified versions of them, making use of the stable path algorithm at

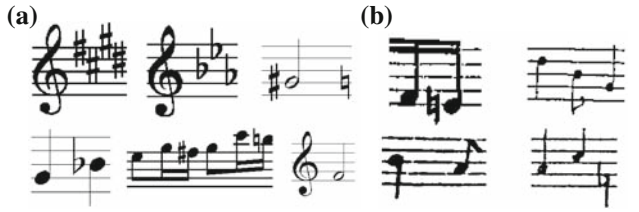


Fig. 3 Variability between different publishings in printed **a** and handwritten, **b** scores. For instance, for the same clef symbol, we have different thickness and for the same beam symbol, we have different shapes. Also, for the same editor, we can have different authors with different typewriting producing, therefore, an nonuniformity on writing

the staff line detection step. The algorithms like line track height, line track chord and Roach/Tatem from [13] were adapted for the tests. The original version of the algorithms were considered as available in [12], making use of the Dalitz algorithm in the detection phase; the modified versions use instead the stable path algorithm for detecting lines. It was experimentally confirmed that the line track height algorithm with the stable path consistently outperformed the other algorithms. Subsequently, we have improved on the line track height algorithms [7], by focusing special attention on deformations—staff lines may have discontinuities, be curved or inclined—that may occur in the music scores. The position of the staff lines obtained by a staff line detection algorithm may pass slightly above or under the real staff lines positions. Therefore, if we are in the presence of a white pixel when the staff lines are tracked, we search vertically for the closest black pixel. If that distance is lower than a specified tolerance, we move the reference position of the staff line to the position of the black pixel found.

3.3 Symbol segmentation

The next module in the processing pipeline is the segmentation of musical symbols, which we based on already existent algorithms [4,16]. The variability of the symbols (in size and shape), found both on handwritten music scores—see Fig. 3b—and in printed scores, when we have scores from different editors—see Fig. 3a—is one of the sources for the complexity of the operation.

This segmentation process consists in localizing and isolating the symbols in order to identify them. In this work, the symbols we want to recognize can be split into four different types:

1. The symbols that are featured by a vertical segment with height greater than a threshold: notes (e.g. ♩), notes with flags (e.g. ♪) and open notes (e.g. 𝅗𝅥).
2. The symbols that link the notes: beams (e.g. ♬).
3. the remaining symbols connected to staff lines: clefs, rests (e.g. 𝄾), accidentals (e.g. ♭, ♯, ♮) and time signature (e.g. 𝄴).
4. The symbols above and under staff lines: notes, relations (e.g. ⌒) and accents (e.g. >).

The segmentation of these types of symbols was based on a hierarchical decomposition of a music image. A music sheet is first analysed and split by staffs, as yielded by the staff lines removal step. Subsequently, the series connected components were identified. To extract only the symbols with appropriate size, a series selection of the connected components detected in the previous step was carried out. The thresholds used for the height and width of the symbols were experimentally chosen. These values take into account the features of the music symbols. As a bounding box of a connected component can contain multiple connected components, care was taken in order to avoid duplicate detections or miss to detect any connected component. In the end, we are ready to find and series extract all the music symbols.

3.3.1 Beam detection

Beams are one of the symbols with harder detection process. Its shape and size are very variable, and they can connect to each other and to other symbols in multiple different arrangements. They are also prone to present inconsistencies in the thickness and in the link with stems—see Fig. 2a. Thus, we propose a solution that just checks the presence of a segment of adequate height, which connects the extremities of notes.

3.3.2 Notes, notes with flags and notes open detection

In this work, we address the segmentation of the stems and note heads as a single primitive symbol. We defined the geometric features of the notes we want to extract as the objects with a height bigger than a threshold, experimentally selected as 2 x *staffspaceheight*, and a width limited by two values, also experimentally chosen as *staffspaceheight*/2 and 3× *staffspaceheight*. To make this task easier the detected beams were removed before the application of this algorithm.

3.3.3 Accidentals, rests, accents and time signature detection

Generally, these symbols have similar values for width and height. The procedure used to extract them was based on the combination of X–Y projection profiles technique [16]. On the one hand, we have symbols that have vertical sequence of black pixels, for instance, sharps, naturals and rests. On the other hand, we need to take into account the symbols topological position, because in this case, we are trying to detect accents and time signature.

3.3.4 Clefs and relations detection

These symbols have their own attributes, like a large width for the relations and a big height for the clefs. In neither of them, we have the presence of stems. With these properties in mind, the projection profiles procedure was used with specific heuristics. For clefs, a width between *staffspaceheight* and $4 \times$ *staffspaceheight* and a height between $2 \times$ *staffspaceheight* and $2 \times$ *numberstaffspace* $\times$ *staffspaceheight* $+$ *numberstaffline* $\times$ *stafflineheight* yielded the best experimental results.[4] These values take into account the fact that clef symbols are the largest of all the signs, beginning below the staff and extending above it. On the other hand, for the relations symbols (ties and slurs), the rules for extracting them were based in a large width.

4 Recognition process

The different approaches to musical symbol classification compared in this work can be categorized as follows:

- Hidden Markov models.
- K-nearest neighbour.
- Neural networks.
- Support vector machines.

For the neural network, k-nearest neighbour and support vector machines methods, each image of a symbol was initially resized to 20×20 pixels and then converted to a vector of 400 binary values; under the hidden Markov model, the images were normalized with a height and width of 150 and 30 pixels, respectively. These approaches follow standard practices in the state-of-the-art algorithms in the OMR field [30].

We randomly split the available data set into training and test sets, with 60% and 40% of the data, respectively. No special constraint was imposed on the distribution of the categories of symbols over the two sets; we only guaranteed that at least one example of each category was present in the training set. The best parameterization of each model was found based on a fourfold cross-validation scheme conducted on the training set. Finally, the error of the model was estimated on the test set. To take into account the variability in writing style, we considered the use of elastic deformation techniques to simulate distortions that can happen in real scores. The data were expanded with variations of the original examples, in order to introduce robustness in the model design. The elastic deformation technique was applied in the training data only.

[4] The *numberstaffline* is the number of lines per staff, as yielded by the staff line detection algorithm; the *numberstaffspace* = *numberstaffline* - *1* is the number of spaces between the staff lines.

4.1 Hidden Markov models

Hidden Markov models (HMMs) have almost never been used in OMR except in some isolated experiences [23,25,30]. The application of this technique to musical symbol classification had its origins on optical character recognition. One of the reasons for the use of HMMs lies in its capability to perform segmentation and recognition at the same time.

A HMM is *a doubly stochastic process* that generates sequence symbols, with an underlying stochastic process that is hidden and can only be detected through another process whose realizations are observable [6]. The hidden process consists of a set of states connected to each other by a transition probability. Transitions probabilities from a state i to another state j are given by $A = \{a_{ij}\}$, where $a_{ij} = P\left[q_{t+1} = S_j | q_t = S_i\right], \quad 1 \leq i, j \leq N$. The observed process consists of a set of outputs or observations. Each observation is contained in a state with some probability density function. The set of observations probabilities is given by $B = b_j(k)$, where $b_j(k) = P\left[o_t = x_k | q_t = S_j\right], \quad 1 \leq k \leq M, j = 1, 2, \ldots, N$. $b_j(k)$ represents the probability of the observation x_k in state S_j, o_j denotes the observation in time t and q_t represents the state in time t. HMM can now be concisely formulated as $\lambda = (A; B; \pi)$, where π is a set of initial probabilities of states [38].

The extraction of features was performed over the images of the symbols, normalized with a height and width of 150 and 30 pixels, respectively. A 2-pixel sliding window mechanism over the symbol image was used to produce the sequence of observations. In doing so, dependent observations are replaced by observations depending on the horizontal position of the window. The extracted features are based on the work of Pugin [30]:

1. the number of distinct connected components of black pixels in the window;
2. the area of the largest black connected component normalized by the area of the window;
3. the area of the smallest white connected component normalized by the area of the window;
4. the position (x and y) of the gravity centre of all black pixels in the window, normalized between 0 and 1.

A left–right, model discriminant HMM was adopted to construct a model for each class [1]. The learning of the parameters of the models ($\lambda = (A; B; \pi)$) was accomplished with the Baum–Welch algorithm. The goal of classification is to decide which class the unknown sequence belongs to, based on the model obtained in the training phase. These symbols were classified on the basis of the maximum of the likelihood ratio obtained by the Viterbi algorithm.

4.2 K-nearest neighbour

The k-nearest neighbour algorithm is amongst the simplest of all machine learning algorithms [14]. This algorithm belongs to a set of techniques called Instance-based Learning. It starts by extending the local region around a data point until the k^{th} nearest neighbour is found. An object is classified by a majority vote scheme, with the object being assigned to the class most common amongst its k-nearest neighbours. The training lies only in the estimation of the best k. Although the distance function could also be learnt during the training, in this work we adopted the most often used Euclidean distance.

4.3 Neural networks

Artificial neural networks, or neural networks for short, were originally inspired on the central nervous system and on the neurons, which constitute one of its most significant information processing elements [18]. With time, they have evolved quite independently from the biological roots, giving rise to more practical implementations, based on statistics and signal processing. In our days, the principles and algorithms of neural networks have found several applications in diverse fields including pattern recognition and signal processing.

In this work, a specific architecture of neural networks was exclusively used, namely the multi-layer perceptron (MLP), one type of a feed-forward network [18]. A MLP is a layered structure consisting of nodes or units (called neurons) and one-way connections or links between the nodes of successive layers. The training of the networks was carried out under Matlab 7 R14 and was done using back-propagation together with the Levenberg–Marquardt algorithm. We use a network with K outputs, one corresponding to each class, and target values of 1 for the correct class and 0 otherwise.

4.4 Support vector machines

Support vector machines (SVMs), pioneered by Vapnik [36] follow the main idea of constructing an hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized.

Formally, given the training set $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$ with input data $\mathbf{x}_i \in \mathbf{R}^p$ and corresponding binary class labels $d_i \in \{-1, 1\}$, the maximum-margin hyperplane is defined by $g(\mathbf{x}) = \mathbf{w}^t\varphi(\mathbf{x}) + b$ where $\varphi(\mathbf{x})$ denotes a fixed-feature space transformation and b a bias parameter; $\mathbf{x}$ is assigned to class 1 if $g(\mathbf{x}) > 0$ or to -1 if $g(\mathbf{x}) < 0$. The maximization of the margin is equivalent to solving

$$\min_{w,b,C,\xi_i} \frac{1}{2}\mathbf{w}^t\mathbf{w} + C\sum_{i=1}^{N}\xi_i$$
$$\text{s.t} \quad y_i[\mathbf{w}^t\varphi(\mathbf{x}) + b] \geq 1 - \xi_i, \quad i = 1, \ldots, N \quad \xi_i \geq 0 \tag{1}$$

where parameter $C > 0$ controls the trade-off between the classification errors and the margin. The slack variables ξ_i, $i = 1, \ldots, N$ are introduced to penalize incorrectly classified data points.

The dual of the formulation Eq. 1 leads to a dependence on the data only through inner products $\phi(\mathbf{x}_i)^t\phi(\mathbf{x}_j)$. Mercer's theorem allows us to express those inner products as a continuous, symmetric, positive semi-definite kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ defined in the input space.

There are three common types of inner-product kernels for SVMs: polynomial learning machine, radial-basis function network and tangent hyperbolic. In this work, a radial-basis function network was used, given by:

$$k(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma||\mathbf{x} - \mathbf{x}_i||^2), \gamma \geq 0 \tag{2}$$

The binary classifier just described can be extended to multiclass scenarios. Of the multiple extensions available in the literature, we used the one against one methodology.

5 Data sets

The data set adopted for the quantitative comparison of the different recognition methods consists of both real handwritten scores of five different composers—see Fig. 4—and synthetic scores, to which distortions were applied—see Fig. 5. The real scores consist on a set of 50 handwritten scores from 5 Portuguese musicians, with ground truth obtained manually. Images were previously binarized with the Otsu threshold algorithm [17], as implemented in the Gamera project.[5] The synthetic data set includes 18 ideal scores from different writers to which known deformations have been applied; this set consists on the fraction of the data set available from [13] written on the standard notation. As the standard notation is the object of study in this work, variants of types of specific notation, such as drums and percussion notation, were not considered. Likewise, since tablature is something specific to certain instruments, it was also not addressed. The deformations applied to the perfect scores were only those with significant impact on the symbols: rotation and curvature; see [13] for more details. In total, 288 images were generated from 18 perfect scores.

The full set of training patterns extracted from the database of scores was augmented with replicas of the existing patterns, transformed according to the elastic deformation

[5] http://gamera.sourceforge.net.

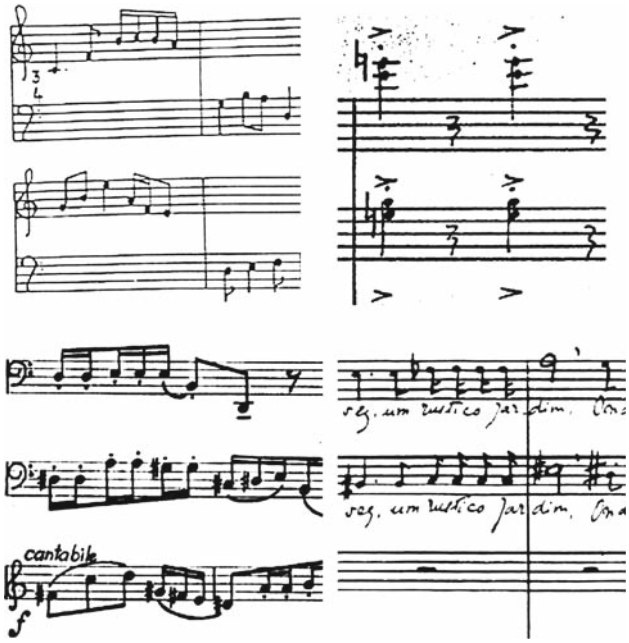


Fig. 4 Some examples of real scores used in this work

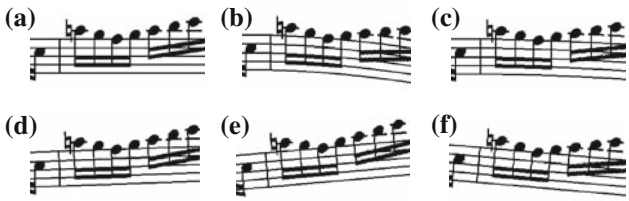


Fig. 5 Some examples of deformations applied to perfect scores: **a** original; **b** curvature with $amplitude = 0.1$; **c** curvature with $amplitude = 0.04$; **d** rotation with $angle = 2$; **e** rotation with $angle = 5$; **f** rotation with $angle = -5$

technique detailed next. Such transformations try to introduce robustness in the prediction with respect to the known variability in the symbols.

The number of handwritten/printed music symbols per class used in the training phase of the classification models is represented in Table 2. The symbols are grouped according to their shape; the rests symbols were divided into two groups—RestI and RestII. Besides that, we included the unknown class to classify those symbols that do not fit into any of the other classes. In total, we have 3222 handwritten music symbols and 2,521 printed music symbols.[6]

5.1 Elastic deformation

The research in deformable template fields applied on handwritten digits and printed characters recognition is well established (e.g. [20,24,29,37]).

[6] The database is available upon request to the authors.

Lam [24], one of the first works in this area, proposed a method of recognition in two stages. The images are first recognized by a tree classifier; those that cannot be satisfactory assigned to a class are passed to a matching algorithm, which deforms the image to match with a template. In [29], a grammar-like model for applying deformations in primitive strokes was developed, while Wakahara [37] proposed a shape-matching approach to recognize numbers manuscripts. The method uses successive local affine transformation (LAT) operations to gradually deform the image. The aim is to yield the best match to an input binary image. LAT on each point at one location is optimized using locations of other points by means of least-squares data fitting using Gaussian window functions. In document degradation models, Baird [3] done an overview in techniques that parameterized models of image defects that occur during printing, photocopying and scanning processes. In this same line, Kanungo [21,22] also proposed a statistical methodology of these deterioration processes in order to validate local degradation models.

The deformation technique used in this work to deform the musical symbols is based in Jain [19,20]. In this approach, the image is mapped on a unit square $S = [0, 1] \times [0, 1]$. The points in this square are mapped by the function $(x, y) \rightarrow (x, y) + D(x, y)$. The space of displacement functions are given by

$$\mathbf{e}^x_{mn}(x, y) = (2\sin(\pi nx)\cos(\pi my), 0) \quad (3)$$

$$\mathbf{e}^y_{mn}(x, y) = (0, 2\sin(\pi ny)\cos(\pi mx)) \quad (4)$$

Specifically, the deformation function is chosen as follows:

$$D(x, y) = \sum_{m=1}^{M}\sum_{n=1}^{N} \frac{\xi^x_{mn}\mathbf{e}^x_{mn} + \xi^y_{mn}\mathbf{e}^y_{mn}}{\lambda_{mn}} \quad (5)$$

where $\underline{\xi} = \left\{(\xi^x_{mn}, \xi^y_{mn}), m, n = 1, 2, \ldots\right\}$ are the projections of the deformation function on the orthogonal basis. Because $D(x, y)$ can represent complex deformations by choosing different coefficients of ξ_{mn} and different values of M and N, it is important to impose a probability density on $D(x, y)$. We assume that the ξ_{mn}'s are independent of each other and the x and y directions are independent, identically distributed Gaussian distributions with mean zero and variance σ^2. Figure 6 shows examples for several deformations using different higher-order terms. Note that the deformation is stronger when M, N and σ are increased.

6 Results

We randomly split the available data set into training and test sets. The splitting of the data into training and test was repeated ten times in order to obtain more stable results for accuracy by averaging and also to assess the variability of

Table 2 Distribution of the full set of handwritten and printed music symbols over the set of classes

	Symbol	Class	Total number
Handwritten Music Symbols		Accent	189
		BassClef	26
		Beam	438
		Flat	230
		Natural	317
		Note	466
		NoteFlag	122
		NoteOpen	208
		RestI	135
		RestII	401
		Sharp	345
		Staccatissimo	21
		TrebleClef	99
		Unknown	404

	Symbol	Class	Total number
Printed Music Symbols		AltoClef	201
		TieSlur	67
		Beam	291
		Flat	155
		Natural	127
		Note	304
		NoteFlag	120
		NoteOpen	309
		RestI	63
		RestII	321
		Sharp	13
		Time	122
		TrebleClef	305
		Unknown	404

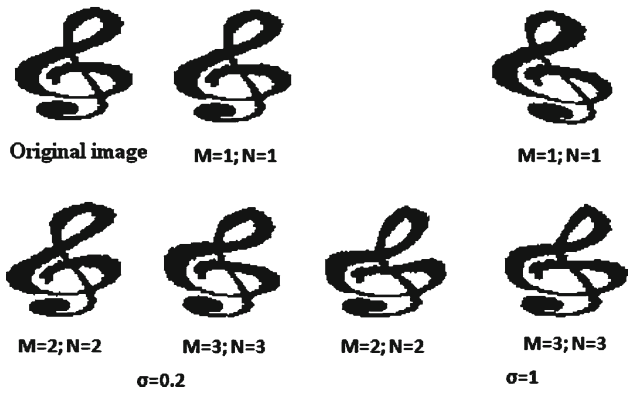


Fig. 6 Example of deformations on a musical symbol

this measure. A confidence interval was computed for the mean of these values as

$$\bar{X} - t^* \frac{S}{\sqrt{N}} \leq \mu \leq \bar{X} + t^* \frac{S}{\sqrt{N}} \tag{6}$$

where t^* is the upper $(1-C)/2$ critical value for the t distribution with $N-1$ degrees of freedom, $\bar{X}$ is the sample mean, S is the sample standard deviation and N is the sample size. The variance of a population represented by a sample is given by

$$\frac{(n-1)S^2}{\chi^2_{[1-(\alpha/2)]}} \leq \sigma^2 \leq \frac{(n-1)S^2}{\chi^2_{(\alpha/2)}} \tag{7}$$

where $\chi^2_{(\alpha/2)}$ is the tabled critical two-tailed value in the chi-square distribution below which a proportion equal to $[1-(\alpha/2)]$ of the cases falls.

From the results obtained for the handwritten music symbols—see Table 3—we can conclude that the classifier with the best performance was the support vector machine with a 99% confidence interval for the expected performance [95%; 96%]. Interestingly, the performance of the simplest model—the nearest neighbour classifier—was clearly better than the performance of the HMM and the neural network model and close to the performance of the SVMs. Finally, although the neural network performed slightly better than the HMM, it exhibited strong difficulties with some classes, presenting very low accuracy values (BassClef and NoteOpen).

The results obtained for the printed music symbols—see Table 4—further support the superiority of the SVM model, with a 99% confidence interval for the expected performance [97%; 99%]. As expected, all models presented the best performance when processing printed musical scores.

Next, we investigated the potential of the elastic deformation to improve the performance of the classification models. The deformations as given by Eq. 5 with $M = 1, 2, 3$ and $N = 1, 2, 3$ were applied in the training data.

The results in Tables 5 and 6 lead us to conclude that the application of the elastic deformation to the music symbols does not improve the performance of the classifiers. Only in two handwritten music symbols, very similar in shape,

Table 3 Accuracy obtained for the handwritten music symbols for the classifiers trained without elastically deformed symbols

	Neural network (%)	Nearest neighbour (%)	Support vector machines (%)	Hidden Markov model (%)
Accent	85	99	99	91
BassClef	13	78	77	56
Beam	85	98	95	90
Flat	84	99	98	87
Natural	93	99	98	91
Note	82	97	96	73
NoteFlag	51	86	89	64
NoteOpen	3	75	40	22
RestI	78	100	97	90
RestII	96	100	100	92
Sharp	85	98	98	84
Staccatissimo	58	100	100	100
TrebleClef	40	92	90	94
Unknown	52	71	89	38
99% CI for the expected performance in percentage: average (standard deviation)	[81 (0.7); 84 (2.6)]	[93 (0.3); 95 (1.2)]	[95 (0.2); 96 (0.6)]	[77 (1.2); 81 (4.3)]

Table 4 Accuracy obtained for the printed music symbols for the classifiers trained without elastically deformed symbols

	Neural network (%)	Nearest neighbour (%)	Support vector machines (%)	Hidden Markov model (%)
AltoClef	94	99	98	83
Beam	92	100	100	98
Flat	97	100	99	96
Natural	94	100	100	95
Note	90	99	99	91
NoteFlag	70	92	96	65
NoteOpen	88	98	97	85
TieSlur	55	94	87	81
RestI	85	100	100	83
RestII	75	100	100	69
Sharp	97	100	100	99
Time	40	100	100	27
TrebleClef	93	100	100	58
Unknown	65	79	93	74
99% CI for the expected performance in percentage: average (standard deviation)	[88 (0.4); 89 (1.5)]	[96 (0.3); 97 (1.0)]	[97 (0.2); 99 (2.1)]	[83 (0.8); 86 (2.9)]

accuracy did improve with elastically deformed symbols—see Table 7.

It is important to state that the features used in the SVM, nearest neighbour and neural network were raw pixels. This choice, grounded in standard practices in the literature, influences the performance of the classifiers: a slight change on the boundary of a symbol can modify the image scaling and as result many of the pixel values may change.

Table 5 Accuracy obtained for the handwritten music symbols for the classifiers trained with elastically deformed symbols

	Neural network (%)	Nearest neighbour (%)	Support vector machines (%)	Hidden Markov model (%)
Accent	83	100	100	87
BassClef	0	95	73	44
Beam	85	96	96	87
Flat	82	99	99	71
Natural	92	99	98	84
Note	86	97	97	64
NoteFlag	20	83	91	34
NoteOpen	2	53	43	11
RestI	59	99	97	99
RestII	93	100	100	75
Sharp	85	99	99	82
Staccatissimo	30	100	100	100
TrebleClef	33	91	90	63
Unknown	34	64	84	34
99% CI for the expected performance in percentage: average (standard deviation)	[77 (0.9); 80 (3.3)]	[92 (0.3); 93 (0.9)]	[94 (0.2); 96 (1.5)]	[69 (1.0); 72 (3.7)]

Table 6 Accuracy obtained for the printed music symbols for the classifiers trained with elastically deformed symbols

	Neural network (%)	Nearest neighbour (%)	Support vector machines (%)	Hidden Markov model (%)
AltoClef	86	99	97	97
Beam	95	100	100	99
Flat	95	100	99	82
Natural	92	100	98	95
Note	81	100	98	89
NoteFlag	43	94	97	39
NoteOpen	89	97	98	78
TieSlur	17	91	89	67
RestI	87	100	100	100
RestII	33	100	97	91
Sharp	97	100	100	100
Time	0	100	100	27
TrebleClef	89	100	100	91
Unknown	42	76	93	38
99% CI for the expected performance in percentage: average (standard deviation)	[79 (1.1); 83 (4.0)]	[95 (0.4); 97 (1.4)]	[97 (0.3); 99 (2.3)]	[81 (1.0); 84 (3.6)]

7 Conclusions

In this paper, we conducted a comparative study of classification methods for musical primitives. We examined four classification methods, namely support vector machines, neural networks, nearest neighbour and hidden Markov models, on two data sets of music scores containing both real handwritten and synthetic scores. The operations preceding the recognition phase, which include the detection and removal of staff lines and segmentation of the musical primitives, were implemented with state-of-the-art algorithms. The staff line detection and removal was based on our recently proposed stable path approach. A key advantage of this algorithm is to approach the problem as the result of

Table 7 Accuracy on the natural and sharp symbols

	Nearest neighbour (%)		Support vector machines (%)	
	With elastic deformation	Without elastic deformation	With elastic deformation	Without elastic deformation
Natural	100	99	98	98
Sharp	99	98	99	98

optimizing a global function. The segmentation method was based on a hierarchical decomposition of the music image. The enormous variability in the music symbols observed in handwritten scores, the inconsistency in size and shape of the symbols greatly complicates the segmentation process. The hierarchical approach allows dealing with such difficulties.

The SVMs, NNs and kNN received raw pixels as input features (a 400 feature vector, resulting from a 20 x 20 pixel image); the HMM received higher level features, like information about the connects components in a 30 x 150 pixel window. These options tried to reflect standard practices in the literature. The performance of any classifier depends crucially on the choice of features. Therefore, results must be interpreted in light of these design options. The SVMs attained the best performance (in line with the current results reported in the literature, where SVMs are systematically the top classifier) with a performance above 95% in the handwritten data set and above 97% in the typeset data set. The simple kNN also achieved a very competitive performance, better than the NNs and the HMMs. The less satisfying performance of the HMMs deserves additional exploration in the future: the choice of the input features, the number of states, the distribution assumed for the observed variable are design option that may be hampering the performance of the model.

Concerning the use of elastic deformations to increase the training set, it was interesting to observe that the performance did not improve. Our aim was to increase the size of the training set, creating controlled distorted symbols to prepare the classifier for the typical variations of symbols in handwritten music sheets. We would expect the classifiers designed with this extended data would be more robust, with improved performance. The results, in opposition to our initial thoughts, may have multiple explanations, which require further investigation: the distortions created were not the most suited for the recognition task, the initial data set was already quite diverse in terms of symbol variety, or the raw representation adopted for features is not the most appropriate for introducing this kind of variation. The fact that the handwritten scores were authored by only five different authors may also help explaining the results: it is possible that the writing was not that diverse to bring benefit to the design with elastic deformation. The enrichment of the data set with scores from more authors may help clarifying this issue.

The various approaches in OMR to musical symbols segmentation and classification are still below the expectations for handwritten musical scores. It is our intention, in future work, to incorporate the prior knowledge of the musical rules in the recognition of symbols. The new proposed methodology should also be naturally adaptable to manuscript images and to different musical notations.

References

1. Arica, N., Yarman-Vural, F.: An overview of character recognition focused on off-line handwriting. IEEE Trans. Syst., Man, Cybern., Part C: Applica. Rev. **31**(2), 216–233 (2001). doi:10.1109/5326.941845
2. Bainbridge, D.: An extensible optical music recognition system. In: Nineteenth Australasian Computer Science Conference, pp. 308–317 (1997)
3. Baird, H.: Document image defect models and their uses. pp. 62–67 (1993). doi:10.1109/ICDAR.1993.395781
4. Bellini, P., Bruno, I., Nesi, P.: Optical music sheet segmentation. In: Proceedings of the 1st International Conference on Web Delivering of Music, pp. 183–190 (2001)
5. Blostein, D., Baird, H.S.: A critical survey of music image analysis. In: Baird Bunke, Y. (ed.) Structured Document Image Analysis, pp. 405–434. Springer, Heidelberg (1992)
6. Bojovic, M., Savic, M.D.: Training of hidden Markov models for cursive handwritten word recognition. In: ICPR '00: Proceedings of the International Conference on Pattern Recognition, p. 1973. IEEE Computer Society, Washington, DC, USA (2000)
7. Capela, A., Rebelo, A., Cardoso, J.S., Guedes, C.: Staff line detection and removal with stable paths. In: Proceedings of the International Conference on Signal Processing and Multimedia Applications (SIGMAP 2008), pp. 263–270 (2008). http://www.inescporto.pt/~jsc/publications/conferences/2008ACapelaSIGMAP.pdf
8. Cardoso, J.S., Capela, A., Rebelo, A., Guedes, C.: A connected path approach for staff detection on a music score. In: Proceedings of the International Conference on Image Processing (ICIP 2008), pp. 1005–1008 (2008)
9. Cardoso, J.S., Capela, A., Rebelo, A., Guedes, C., da Costa, J.P.: Staff detection with stable paths. IEEE Trans. Pattern Anal. Mach. Intell. **31**(6), 1134–1139 (2009). doi:10.1109/TPAMI.2009.34
10. Coüasnon, B.: Segmentation et reconnaissance de documents guidées par la connaissance a priori: application aux partitions musicales. Ph.D. thesis, Université de Rennes (1996)
11. Coüasnon, B., Camillerapp, J.: Using grammars to segment and recognize music scores. In: Proceedings of DAS-94: International Association for Pattern Recognition Workshop on Document Analysis Systems, pp. 15–27. Kaiserslautern (1993)

12. Dalitz, C., Droettboom, M., Czerwinski, B., Fujigana, I.: Staff removal toolkit for gamera (2005–2007). http://music-staves.sourceforge.net
13. Dalitz, C., Droettboom, M., Czerwinski, B., Fujigana, I.: A comparative study of staff removal algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **30**, 753–766 (2008)
14. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification (2nd Edn.). Wiley, New York (2000)
15. Fornés, A., Lladós, J., Sánchez, G.: Primitive segmentation in old handwritten music scores. In: Liu, W., Lladós, J. (eds.) GREC, Lecture Notes in Computer Science, vol. 3926, pp. 279–290. Springer (2005). http://dblp.uni-trier.de/db/conf/grec/grec2005.html#FornesLS05
16. Fujinaga, I.: Staff detection and removal. In: George, S. (ed.) Visual Perception of Music Notation: On-Line and Off-Line Recognition, pp. 1–39. Idea Group Inc, Hershey (2004)
17. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: In: Digital Image processing using MATLAB, Pearson/Prentice-Hall, Upper Saddle River (2004)
18. Haykin, S.: Neural Networks: A Comprehensive Foundation (2nd edn.). Prentice Hall (1998). http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0132733501
19. Jain, A.K., Zhong, Y., Lakshmanan, S.: Object matching using deformable templates. IEEE Trans. Pattern Anal. Mach. Intell. **18**(3), 267–278 (1996). doi:10.1109/34.485555
20. Jain, A.K., Zongker, D.: Representation and recognition of handwritten digits using deformable templates. IEEE Trans. Pattern Anal. Mach. Intell. **19**(12), 1386–1391(1997). doi:10.1109/34.643899
21. Kanungo, T.: Document degradation models and a methodology for degradation model validation. Ph.D. thesis, Seattle, WA, USA (1996)
22. Kanungo, T., Haralick, R., Baird, H., Stuezle, W., Madigan, D.: A statistical, nonparametric methodology for document degradation model validation.. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1209–1223 (2000). doi:10.1109/34.888707
23. Kopec, G.E., Parc, P.A.C., Maltzcarnegie, D.A.: Markov source model for printed music decoding. J Electron Imaging, pp. 7–14 (1996)
24. Lam, L., Suen, C.Y.: Structural classification and relaxation matching of totally unconstrained handwritten zip-code numbers. Pattern Recognit. **21**(1), 19–32 (1988). doi:10.1016/0031-3203(88)90068-4
25. Mitobe, Y., Miyao, H., Maruyama, M.: A fast HMM algorithm based on stroke lengths for on-line recognition of handwritten music scores. In: IWFHR '04: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition, pp. 521–526. IEEE Computer Society, Washington (2004). doi:10.1109/IWFHR.2004.2
26. Miyao, H., Nakano, Y.: Note symbol extraction for printed piano scores using neural networks. IEICE Trans. Inf. Syst. **E79–D**, 548–554 (1996)
27. Miyao, H., Okamoto, M.: Stave extraction for printed music scores using DP matching. J Adv. Comput. Intell. Intell. Inform. **8**, 208–215 (2007)
28. Ng, K.: Optical music analysis for printed music score and handwritten music manuscript. In: George, S. (ed.) Visual Perception of Music Notation: On-Line and Off-Line Recognition, pp. 108–127. Idea Group Inc, Hershey (2004)
29. Nishida, H.: A structural model of shape deformation. Pattern Recognit. **28**(10), 1611–1620 (1995)
30. Pugin, L.: Optical music recognition of early typographic prints using hidden Markov models. In: ISMIR, pp. 53–56 (2006)
31. Randriamahefa, R., Cocquerez, J., Fluhr, C., Pepin, F., Philipp, S.: Printed music recognition. In: Proceedings of the Second International Conference on Document Analysis and Recognition, pp. 898–901 (1993). doi:10.1109/ICDAR.1993.395592
32. Reed, K.T., Parker, J.R.: Automatic computer recognition of printed music. Proc. 13th Int. Conf. Pattern Recognit. **3**, 803–807 (1996). doi:10.1109/ICPR.1996.547279
33. Rossant, F., Bloch, I.: Robust and adaptive omr system including fuzzy modeling, fusion of musical rules, and possible error detection. EURASIP J. Adv. Signal Process. **2007**(1), 160–160 (2007). doi:10.1155/2007/81541
34. Szwoch, M.: A robust detector for distorted music staves. In: Computer Analysis of Images and Patterns. pp. 701–708. Springer, Heidelberg (2005)
35. Toyama, F., Shoji, K., Miyamichi, J.: Symbol recognition of printed piano scores with touching symbols. pp. 480–483 (2006). doi:10.1109/ICPR.2006.1099
36. Vapnik, V.N.: Statistical Learning Theory. Wiley (1998). http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0471030031
37. Wakahara, T.: Shape matching using LAT and its application to handwritten numeral recognition. IEEE Trans. Pattern Anal. Mach. Intell. **16**(6), 618–629 (1994). doi:10.1109/34.295906
38. Wang, Y.K., Fan, K.C., Juang, Y.T., Chen, T.H.: Using hidden Markov model for chinese business card recognition. In: ICIP (1), pp. 1106–1109 (2001)