IJDAR
DOI 10.1007/s10032-011-0171-7

ORIGINAL PAPER

A robust probabilistic Braille recognition system

M. Yousefi · M. Famouri · B. Nasihatkon · Z. Azimifar · P. Fieguth

Received: 22 November 2010 / Revised: 8 July 2011 / Accepted: 11 July 2011

Abstract As a structured document, Braille is the most common means of reading and study for visually handicapped people. The need for converting Braille documents into a computer-readable format has motivated research into the implementation of Braille recognition systems. The main theme of this research is to propose robust probabilistic approaches to different steps of Braille Recognition. The method is meant to be very general in terms of being independent of those parameters of the Braille document such as skewness, scale, and spacing of the page, lines, and characters. For a given Braille document, a statistical method is proposed for estimating the scaling, spacing, and skewness parameters, whereby the detected dots of the Braille document are modeled using a parameterized probability density function. Skewness, scaling, and line spacing are estimated as a solution of a maximum-likelihood (ML) problem, which is solved using expectation maximization. Based on those parameters, each line of the Braille document is extracted, and each of three rows of individual lines is separated based on the vertical projection of the Braille dots. Finally, a scale-independent automatic document gridding procedure is proposed for dot localization and character detection based on a hidden Markov model.

Keywords Structured document · Braille · Line spacing · Scale independence · Hidden Markov Model · Expectation maximization

M. Yousefi · M. Famouri · B. Nasihatkon · Z. Azimifar (✉)
School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran
e-mail: azimifar@cse.shirazu.ac.ir

P. Fieguth
Systems Design Engineering, University of Waterloo, Waterloo, Canada

1 Introduction

Braille is the most popular convention of reading and writing among visually impaired people. Like other documents, Braille documents need to be converted to a computer-readable form so that they can take advantage of the benefits of digital documents simple maintenance, duplication, translation, text-to-speech conversion, etc.

A large body of research has been devoted to Braille processing, loosely categorized into five domains:

- Paper type and the Braille sheet quality, such as an integrated system of organic field transistors and plastic actuators, applied to a Braille sheet display [1,2];
- Electrical displays, such as methods designed for reading Braille characters from a touch screen mobile device [3,4], a tactile stimulator array [5], or pneumatic refreshable Braille display [6];
- Mechanical devices or sensors to read Braille pages [7];
- Document-to-Braille conversion, both for word-processed documents [8] and graphics [9];
- Braille recognition, extracting Braille dots and converting them to text [10].

The last group, with Braille-to-text recognition, is the topic of this paper. Some research [11–13] is based on the use of specific, specialized equipment in which the image is illuminated from a special direction and then captured by a camera.

More recent approaches are based on scanned images. Ritchings et al. [14] proposed a method addressing both single and two-sided Braille documents. However, besides some assumptions regarding document scaling, no treatment for skewness was mentioned. Antonacopoulos et al. [15] carried out a more sophisticated work, solving the scaling problem

Published online: 18 August 2011

using a set of heuristic approaches (like histograms) and solving the skewness problem using the *Hough transform*. Tai et al. [16] proposed to take advantages of the *Radon Transform* to solve the skewness problem. They also used a set of histogram-based heuristics to detect each row and column of the documents.

Since the histogram of a Braille image is expected to have three modes *(recto, verso and background)*, Al-Saleh et al [17] trained a *beta distribution* for documents and found an appropriate threshold to segment the document and then used a grid-based algorithm for detecting recto and verso dots. They did not, however, consider any document skewness.

It is worth noting that Braille documents differ from other regularly structured documents. In particular, Braille belongs to the class of simple and well-structured documents, making it possible to apply more general and high-performance methods specialized to the Braille environment. Based on this fact, this paper presents an efficient and general method for the entire Braille recognition process. The method is general in the sense that it does not make prior assumptions about such Braille document parameters as skewness, scale and spacing of the page, lines and characters. Such parameters are estimated by means of fitting some parametric statistical models. The models also consider the effect of noise and outliers on the document.

This article is comprised of two major parts. The first part, described in Sect. 2, is line detection, building upon the authors' previous work [18], but generalizing that work by adding a noise model in the estimation of line spacing and skewness parameters. The second part, described in Sect. 3, develops a hidden Markov model (HMM) for the detection of Braille characters via the detection of sets of dots. Individual line and character tests are undertaken in the respective sections; however, in Sect. 4 a comprehensive test is undertaken on longer documents, with the paper concluded in Sect. 5.

2 Line detection

Braille documents include a number of lines, each comprising of a series of characters. As Fig. 1 shows, each character consists of 6 points arranged in a 3×2 mask. Each point can be either *on* (raised) or *off* (flat). Some examples of a Braille character are illustrated in Fig. 2.

2.1 Preprocessing

Here, without lose of generality, we study one-sided Braille documents. First, a thresholding algorithm is applied to the image. The black pixels appearing after thresholding are either shadows of the Braille dots or undesired artifacts.

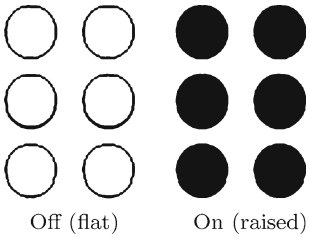

Fig. 1 The binary dot state in Braille characters

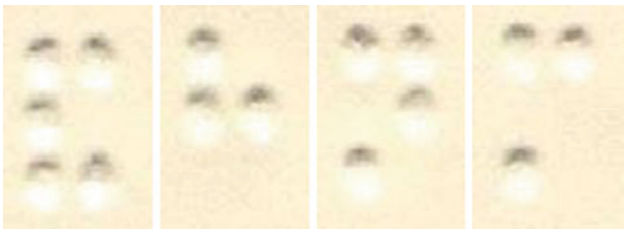

Fig. 2 Examples of Braille characters

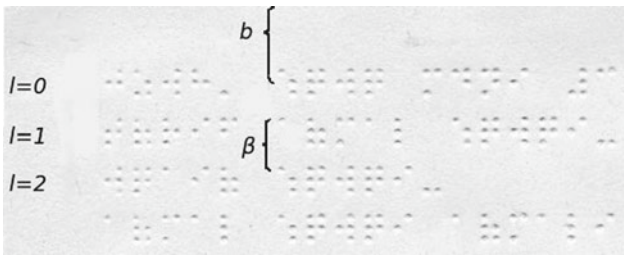

Fig. 3 Vertical Braille document parameters, showing the line spacing β and the vertical offset b

Afterward, a *connected component* algorithm is run on the thresholded image, such that each connected component represents a potential Braille dot. Let us denote these components by $\{C_1, C_2, \ldots, C_N\}$ and regard the center of the ith connected component (averaged location of pixels in the connected component) by $\mathbf{x}_i = [x_i\ y_i]^T$ and the collection of all centers by $\mathbf{X} = \{\mathbf{x}_i | 1 \leq i \leq N\}$, where N shows the total number of connected components. Note that each component C_i represents either a dot or noise.

2.2 Modeling without noise and skewness assumption

Let us assume that the document is artifact free and ignore document skewness at this point to focus on modeling and estimating document scale and line-spacing parameters. The scaling properties of the document are modeled using two parameters b and β. As shown in Fig. 3, parameter b is the offset (middle) of the first line of the document from the top of the page and β shows the distance between two consecutive lines.

In this research, training data are assumed to be samples drawn from a probability density function (PDF). Since no skewness is assumed, projections of the data belonging to a *single Braille line* to the vertical axis can be modeled by a trimodal distribution, *e.g.*, a mixture of three Gaussians each accounting for a single *row* of the Braille line (see Fig. 4a). However, to avoid some implementation issues, including model complexity and slow convergence, all three rows of the Braille line are modeled by a single Gaussian (see Fig. 4b). Of course, it seems to work unexpectedly well since the Braille alphabets have considerable bias toward using the upper two rows for must letter except for punctuation marks; therefore, the Gaussian would have a tail on the third row. However, the experiments show that sufficient accuracy is obtained using a Gaussian, Thus, the vertical coordinates of all connected components (*i.e.*, $\{y_1, \ldots, y_N\}$) can be considered as samples drawn from a one-dimensional Gaussian mixture model (GMM):

$$p_y(y) = \frac{1}{L} \sum_{l=0}^{L-1} \mathcal{N}(y\,; b + l\beta, \sigma) \tag{1}$$

where line count l is as depicted in Fig. 3, and L counts the total number of lines in the document. Note that all Gaussian functions of this mixture share a common standard deviation σ and magnitude $\frac{1}{L}$.

2.3 Modeling without noise but with skewness assumption

Adding the skewness in the above formulation is straightforward. Suppose the document is rotated by an angle ϕ. Then, the points $\mathbf{x}_i$ can be considered as unskewed image points $\mathbf{z}_i$ rotated by ϕ. Therefore, we have

$$\mathbf{z}_i = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \mathbf{x}_i = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{w}^T \end{bmatrix} \mathbf{x}_i \tag{2}$$

Thus, the vertical (second) element of the unskewed point $\mathbf{z}_i$ can be obtained by

$$(\mathbf{z}_i)_2 = \mathbf{w}^T \mathbf{x}_i = w_1\, x_i + w_2\, y_i \tag{3}$$

where w_1 and w_2 are elements of $\mathbf{w}$. Now, vertical elements of unskewed points can be considered as samples from $p_y(y)$ defined in (1). By substituting (3) into (1), we obtain the two-dimensional distribution

$$g(x, y) = \frac{1}{L} \sum_{l=0}^{L-1} \mathcal{N}(w_1 x + w_2 y; b + l\beta, \sigma) \tag{4}$$

which, accounting for page dimensions, becomes

$$p_{\mathbf{x}}(\mathbf{x}) = p_{\mathbf{x}}(x, y) = \begin{cases} c\, g(x, y) & B_l \le w_2 x - w_1 y \le B_r \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

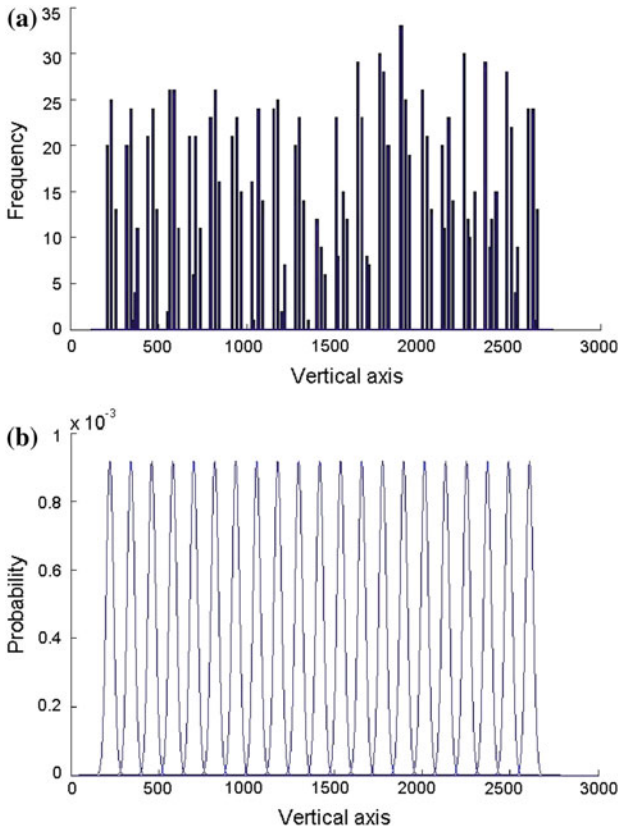

Fig. 4 **a** Histogram of the vertical projection, **b** A GMM is fitted to the vertically projected data in which each Gaussian belongs to a single line

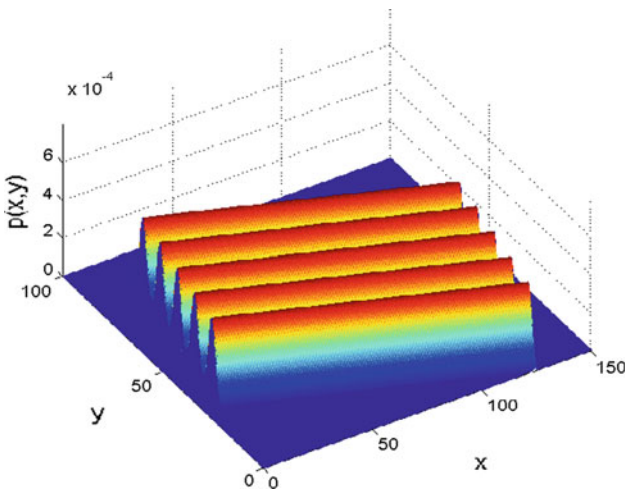

Fig. 5 An example of a 2-D PDF modeling dot positions, modeling the Braille skewness as well as the scaling properties, for $L = 5$

where B_l and B_r show page's left and right borders, respectively. Thus, $p_{\mathbf{x}}(\mathbf{x})$ is a Gaussian mixture (across rows) if projected on axis $[w_1, w_2]^T$, and uniform (along rows) if projected on $[w_2, -w_1]^T$. An example of $p_{\mathbf{x}}(\mathbf{x})$ with five lines ($L = 5$) is depicted in Fig. 5.

2.4 Modeling with noise and skewness assumption

Some of the connected components in the document will clearly not belong to the Braille dots, stemming from any number of noise or other undesired artifacts. Because of the many factors that can result in an artifact, no exact structure or model can be presumed. On the contrary, Braille dots have a well-defined structure, allowing a good discrimination between Braille dots and other artifacts. Here, three features are extracted from each connected component:

- The sample covariance matrix and eigendecomposition of the points of each connected component are calculated, with eigenvalues λ_1, λ_2. The first features is chosen to be

$$F_1 = \frac{\max(\lambda_1, \lambda_2)}{\min(\lambda_1, \lambda_2)}$$

- The second feature is $\max(\lambda_1, \lambda_2)$ divided by averaging over the same quantities among all connected components. This division is necessary to make the algorithm independent of the document scaling. Here, possible large artifacts in the image may cause problem, since they can act as an outlier misleading the average value by which those quantities are divided. Therefore, to calculate the average, we use *median* instead of *mean* to make features robust to such outliers.
- The third feature is extracted from the original grayscale image. Note that as the scanner makes shadows on one side of each Braille dot, it makes brightness on the other side, as is obvious in Fig. 2. Here, the average brightness of the pixels in a small box located at a certain distance bellow each shadow (below the center of the connected component) is used as a feature. The size of the box and the distance from the center of each connected component are proportional to the averaged quantity used in the second feature. This is because that quantity can be assumed to be proportional to the document scale. It is expected that, for the Braille dots, the pixels in this box are brighter due to that lighting property.

Note that the features used here are not necessarily the only features or the best features for this purpose and other features should be used, e.g., for generalization of the method to deal with the two-sided Braille documents. We employ these features to train a classifier. It should be noted that the Braille dots and the artifacts are not discriminated at this stage. Instead, the classifier assigns a probability to each connected component which is an estimation of the probability of that connected component to be a Braille dot and not a noise. To this end, we need a document or a set of documents whose connected components are labeled as *Braille dot* or *noise* to serve as our training data. One of the documents was selected randomly and software found all of its connected components. Then, a person labeled all of connected components that belong to the Braille document. In the provision of the training data, some considerations should be taken into account according to the nature of the features we are using. For example, in our implementation, since the features, specifically the third feature, depend on the direction of the light and therefore the scanning device, the training documents should be captured from the same type of scanner.

A Bayesian classifier is implemented to accomplish this goal. Let C be a typical connected component and $\mathbf{F}$ be the feature vector extracted from it. Here, both the class conditional distributions $p(\mathbf{F}|C \in \text{dot})$ and $p(\mathbf{F}|C \in \text{noise})$ are assumed to be Gaussian. This can be problematic for the later (noise) distribution since they do not possess a determined structure. Therefore, in our training document, different kinds of artifacts should be examined. If needed, some of these artifacts can be added manually. Since we have no idea about the amount of noise in the document, the prior probability of each component is chosen to be $P(C \in \text{dot}) = P(C \in \text{noise}) = \frac{1}{2}$. For a new Braille document and for each connected component (Say C_i), the trained Bayesian classifier considers a probability

$$\pi_i = P(C_i \in \text{dot}|\mathbf{F}_i) = \frac{p(\mathbf{F}_i|C_i \in \text{dot})\,P(C_i \in \text{dot})}{p(\mathbf{F}_i|C_i \in \text{dot})\,P(C_i \in \text{dot}) + p(\mathbf{F}_i|C_i \in \text{noise})\,P(C_i \in \text{noise})} \tag{6}$$

As noted before, no discrimination between noise and Braille dots is made at this stage. Instead, line detection is applied using all the connected components and only the probability π_i is incorporated in our formulations.

To estimate the document parameters introduced in (4), we consider the locations of the centers of connected component to be samples drawn from an special distribution. If the connected component belongs to Braille dots, then its center is assumed a sample drawn from the distribution of dots which was introduced in (5):

$$p_{\mathbf{x}}(\mathbf{x}_i|C_i \in \text{dots}) = \begin{cases} c\,g(x, y) & B_l \le w_2 x - w_1 y \le B_r \\ 0 & \text{Otherwise} \end{cases} \tag{7}$$

here, C_i is the ith connected component and $\mathbf{x}_i$ shows its center. If the connected component is a noise, since the noise can occur everywhere in the document, and our only knowledge is that the location of noise is not outside the page, we assume that the noise is uniformly distributed among the page:

$$p_{\mathbf{x}}(\mathbf{x}_i|C_i \in \text{noise}) = \begin{cases} \frac{1}{S} & \mathbf{x}_i \text{ lies inside the page borders} \\ 0 & \mathbf{x}_i \text{ lies outside the page borders} \end{cases} \tag{8}$$

where S is the area of the page. Here, for each connected component C_i, in addition to the location of its center $\mathbf{x}_i$, we have another quantity π_i that shows the probability of the connected component to be a Braille dot. Therefore, the overall distribution for a connected component is obtained as:

$$p_{\mathbf{x},\pi}(\mathbf{x}_i, \pi_i) = \pi_i \; p_{\mathbf{x}}(\mathbf{x}_i | C_i \in \text{dot}) + (1 - \pi_i) \; p_{\mathbf{x}}(\mathbf{x}_i | C_i \in \text{noise}) \tag{9}$$

2.5 Model parameter estimation

To simplify the parameter estimation, at this point, it is assumed that the number of lines L is given. Therefore, the problem reduces to estimation of the parameters $(b, \beta, \mathbf{w}, \sigma)$ using the probabilistic model (9). A method to estimate the number of lines is given in Sect. 2.6. Here, sample data are assumed to be independent and the problem is solved by maximizing the log-likelihood function. But before the calculating the log-likelihood, we need to perform some simplifications. First of all, notice that if the density function (8) is to be evaluated at a sample $\mathbf{x}_i$, it is always equal to $\frac{1}{S}$, since the sample never lies outside the page. Here, we make a further assumption that no sample $\mathbf{x}_i$ is evaluated to 0 in the function (7); thus, we take $B_r - B_l$ equal to the width of the page. Notice that these are only approximations made to simplify our formulations. It is possible that sample belonging to noise be evaluated to 0 in (7). Even our selection of the value of $B_r - B_l$ does not guarantee that no sample evaluates to zero in (7). However, we empirically observed that these approximations do not have a major impact on the results of our line detection algorithm. According to these assumptions, the log-likelihood function can be written as:

$$\begin{aligned} &\log \; p(\mathbf{x}_1, \pi_1, \mathbf{x}_2, \pi_2, \ldots, \mathbf{x}_N, \pi_N) \\ &= \log \prod_{i=1}^{N} p_{\mathbf{x},\pi}(\mathbf{x}_i, \pi_i) \\ &= \sum_{i=1}^{N} \log \left(\pi_i \left(\sum_{l=0}^{L-1} c \frac{1}{L} \mathcal{N}(\mathbf{w}^T \mathbf{x}_i; b + l\beta, \sigma) \right) \right. \\ &\qquad \left. + (1 - \pi_i) \frac{1}{S} \right) \end{aligned} \tag{10}$$

Differentiating from the above likelihood function with respect to its parameters and making them equal to zero leads to a set of complex nonlinear equations, which are hard to solve. Thus, here, the iterative method *EM* is applied to maximize the likelihood [19,20]. In this case, latent (hidden) variables are $= (\omega_1, \omega_2, \ldots, \omega_N)$, where $\omega_i \in \{0, 1, \ldots, L-1, L\}$ is a random variable showing the number of Braille lines $(0, 1, \ldots, L-1)$ to which sample $\mathbf{x}_i$ belongs. If sample $\mathbf{x}_i$ belongs to noise, then $\omega_i = L$. Following, we describe the steps of the EM algorithm:

2.5.1 Expectation step

In the expectation step, we should find the probability distribution of each ω_i given our observations and the current (at time t) estimations of the parameters: $\boldsymbol{\theta}^{(t)} = \{b^{(t)}, \beta^{(t)}, \mathbf{w}^{(t)}, \sigma^{(t)}\}$.

$$\begin{aligned} P_{il} &= P(\omega_i = l | \mathbf{x}_i, \pi_i, \boldsymbol{\theta}^{(t)}) \\ &= \frac{P(\omega_i = l, \mathbf{x}_i | \pi_i, \boldsymbol{\theta}^{(t)})}{P(\mathbf{x}_i | \pi_i, \boldsymbol{\theta}^{(t)})} \\ &= \frac{P(\omega_i = l, \mathbf{x}_i | \pi_i, \boldsymbol{\theta}^{(t)})}{\sum_{l=0}^{L} P(\omega_i = l, \mathbf{x}_i | \pi_i, \boldsymbol{\theta})} \\ &= \frac{P(\mathbf{x}_i | \omega_i = l, \pi_i, \boldsymbol{\theta}) P(\omega_i = l | \pi_i, \boldsymbol{\theta})}{\sum_{l=0}^{L} P(\mathbf{x}_i | \omega_i = l, \pi_i, \boldsymbol{\theta}) P(\omega_i = l | \pi_i, \boldsymbol{\theta})} \\ &= \frac{P(\mathbf{x}_i | \omega_i = l, \boldsymbol{\theta}) P(\omega_i = l | \pi_i, \boldsymbol{\theta})}{\sum_{l=0}^{L} P(\mathbf{x}_i | \omega_i = l, \boldsymbol{\theta}) P(\omega_i = l | \pi_i, \boldsymbol{\theta})} \end{aligned} \tag{11}$$

where

$$P(\omega_i = l | \pi_i, \boldsymbol{\theta}) = \begin{cases} (\frac{1}{L})\pi_i & \text{if } 0 \le l \le L-1, \\ (1 - \pi_i) & \text{if } l = L \end{cases} \tag{12}$$

and

$$P(\mathbf{x}_i | \omega_i = l, \boldsymbol{\theta}) = \begin{cases} c\mathcal{N}(\mathbf{w}^T \mathbf{x}_i; b + l\beta, \sigma) & \text{if } 0 \le l \le L-1, \\ \frac{1}{S} & \text{if } l = L \end{cases} \tag{13}$$

Note that in the formulation of (11), we have made use of the conditional independence of $\mathbf{x}_i$ and π_i given ω_i (if we know ω_i, then $\mathbf{x}_i$ should be a Braille dot if $\omega_i < L$ and an artifact if $\omega_i = L$).

2.5.2 Maximization step

In this step, we should form $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$ and find the next estimation of the parameters $\boldsymbol{\theta}^{(t+1)} = \{b^{(t+1)}, \beta^{(t+1)}, \mathbf{w}^{(t+1)}, \sigma^{(t+1)}\}$ by:

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \; Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) \tag{14}$$

where Q is obtained by:

$$\begin{aligned} &Q(\theta, \theta^{(t)}) \\ &= E_{\Omega} \{\log P(X, \Pi, \Omega | \theta)\} \\ &= E_{\Omega} \{\log P(x_1, \ldots, x_N, \omega_1, \ldots, \omega_N, \pi_1, \ldots, \pi_N | \theta)\} \end{aligned}$$

$$
\begin{aligned}
&= E_{\Omega}\left\{\log\prod_{i=1}^{N} P(x_i,\omega_i,\pi_i|\theta)\right\}\\
&= \sum_{i=1}^{N} E_{\Omega}\left\{\log P(x_i,\omega_i,\pi_i|\theta)\right\}\\
&= \sum_{i=1}^{N} E_{\omega_i}\left\{\log P(x_i,\omega_i,\pi_i|\theta)\right\}\\
&= \sum_{i=1}^{N}\sum_{l=0}^{L}\log P(x_i,\omega_i=l,\pi_i|\theta)P_{il}\\
&= \sum_{i=1}^{N}\sum_{l=0}^{L}\log(P(x_i|\omega_i=l,\pi_i,\theta)P(\omega_i=l,\pi_i|\theta))P_{il}\\
&= \sum_{i=1}^{N}\sum_{l=0}^{L}\log(P(x_i|\omega_i=l,\pi_i,\theta)\\
&\quad\times P(\omega_i=l|\pi_i,\theta)P(\pi_i|\theta))P_{il}\\
&= \sum_{i=1}^{N}\sum_{l=0}^{L-1}(\log P(x_i|\omega_i=l,\pi_i,\theta)\\
&\quad+\log P(\omega_i=l|\pi_i,\theta)+\log P(\pi_i|\theta))P_{il}\ +\ C_1\\
&= \sum_{i=1}^{N}\sum_{l=0}^{L-1}\log P(x_i|\omega_i=l,\pi_i,\theta)P_{il}\ +\ C_2\\
&= \sum_{i=1}^{N}\sum_{l=0}^{L-1}\log P(x_i|\omega_i=l,\theta)P_{il}\ +\ C_2\\
&= \sum_{i=1}^{N}\sum_{l=0}^{L-1}\log(c\mathcal{N}(\mathbf{w}^T\mathbf{x}_i;b+l\beta,\sigma))P_{il}\ +\ C_2\\
&= \sum_{i=1}^{N}\sum_{l=0}^{L-1}\left(-\log\sigma-\frac{1}{2}\frac{(\mathbf{w}^T\mathbf{x}_i-b-l\beta)^2}{\sigma^2}\right)P_{il}\ +\ C_3
\end{aligned}
\tag{15}
$$

where C_1, C_2 and C_3 are constants with respect to our parameters $\boldsymbol{\theta}=\{b,\boldsymbol{\beta},\mathbf{w},\sigma\}$ and thus, have no effect to the optimization problem.

In order to maximize $Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(t)})$, partial derivatives of Q with respect to each of the parameters $\boldsymbol{\theta}=\{b,\beta,\mathbf{w},\sigma\}$ should be calculated and made equal to zero:

$$
\begin{aligned}
&\frac{\partial}{\partial b}Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(t)})=0\Rightarrow\left(\sum_{i=1}^{N}\sum_{l=0}^{L-1}P_{il}\right)b+\left(\sum_{i=1}^{N}\sum_{l=0}^{L-1}lP_{il}\right)\beta\\
&=\left(\sum_{i=1}^{N}\sum_{l=0}^{L-1}P_{il}\mathbf{x}_i\right)^T\mathbf{w}
\end{aligned}
\tag{16}
$$

$$
\begin{aligned}
&\frac{\partial}{\partial\beta}Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(t)})=0\Rightarrow\left(\sum_{i=1}^{N}\sum_{l=0}^{L-1}l\,P_{il}\right)b+\left(\sum_{i=1}^{N}\sum_{l=0}^{L-1}l^2P_{il}\right)\beta\\
&=\left(\sum_{i=1}^{N}\sum_{l=0}^{L-1}lP_{il}\mathbf{x}_i\right)^T\mathbf{w}
\end{aligned}
\tag{17}
$$

To find the optimal value for parameter $\mathbf{w}=[w_1,w_2]^T$, we should consider the constraint:

$$
{w_1}^2+{w_2}^2=\mathbf{w}^T\mathbf{w}=1. \tag{18}
$$

Therefore, the Lagrange multiplier λ should be applied in differentiation:

$$
\begin{aligned}
&\nabla_{\mathbf{w}}\left(Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(t)})+\lambda(\mathbf{w}^T\mathbf{w}-1)\right)=\mathbf{0}\Rightarrow\\
&\times\left(\sum_{i=1}^{N}\sum_{l=0}^{L-1}P_{il}\,\mathbf{x}_i{\mathbf{x}_i}^T\right)\mathbf{w}-\left(\sum_{i=1}^{N}\sum_{l=0}^{L-1}P_{il}\,\mathbf{x}_i\right)b\\
&-\left(\sum_{i=1}^{N}\sum_{l=0}^{L-1}lP_{il}\mathbf{x}_i\right)\beta=\lambda'\mathbf{w}
\end{aligned}
\tag{19}
$$

where $\lambda'=2\sigma^2\lambda$. The fifth equation is the constraint (18), and the sixth equation is obtained by differentiating with respect to σ:

$$
\begin{aligned}
&\frac{\partial}{\partial\sigma}Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(t)})=0\Rightarrow\\
&\sigma^2=\frac{\sum_{i=1}^{N}\sum_{l=0}^{L-1}P_{il}\left(\mathbf{w}^T\mathbf{x}_i-(b+\beta l)\right)^2}{\sum_{i=1}^{N}\sum_{l=0}^{L-1}P_{il}}
\end{aligned}
\tag{20}
$$

Equations (16) and (17) can be combined as:

$$
\mathbf{A}\begin{bmatrix}b\\ \beta\end{bmatrix}=\mathbf{B}^T\mathbf{w} \tag{21}
$$

where matrices $\mathbf{A}$ and $\mathbf{B}$ are defined as:

$$
\mathbf{A}=\begin{bmatrix}\sum_{i=1}^{N}\sum_{l=0}^{L-1}P_{il} & \sum_{i=1}^{N}\sum_{l=0}^{L-1}lP_{il}\\ \sum_{i=1}^{N}\sum_{l=0}^{L-1}lP_{il} & \sum_{i=1}^{N}\sum_{l=0}^{L-1}l^2P_{il}\end{bmatrix}
$$

$$
\mathbf{B}=\begin{bmatrix}\sum_{i=1}^{N}\sum_{l=0}^{L-1}P_{il}\mathbf{x}_i & \sum_{i=1}^{N}\sum_{l=0}^{L-1}lP_{il}\mathbf{x}_i\end{bmatrix}
$$

Similarly, (19) can be written in matrix form:

$$
\mathbf{C}\mathbf{w}-\mathbf{B}\begin{bmatrix}b\\ \beta\end{bmatrix}=\lambda'\mathbf{w} \tag{22}
$$

where $\mathbf{B}$ is as defined above and $\mathbf{C}=\sum_{i=1}^{N}\mathbf{x}_i{\mathbf{x}_i}^T$. Considering (21) and (22) together, we have:

$$
\left(\mathbf{C}-\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T\right)\mathbf{w}=\lambda'\mathbf{w} \tag{23}
$$

Interestingly, (23) is simply an eigenvalue problem and $\mathbf{w}$ can be obtained by solving it. Although solving the eigenvalue problem results in two answers for $\mathbf{w}$, these two eigenvectors

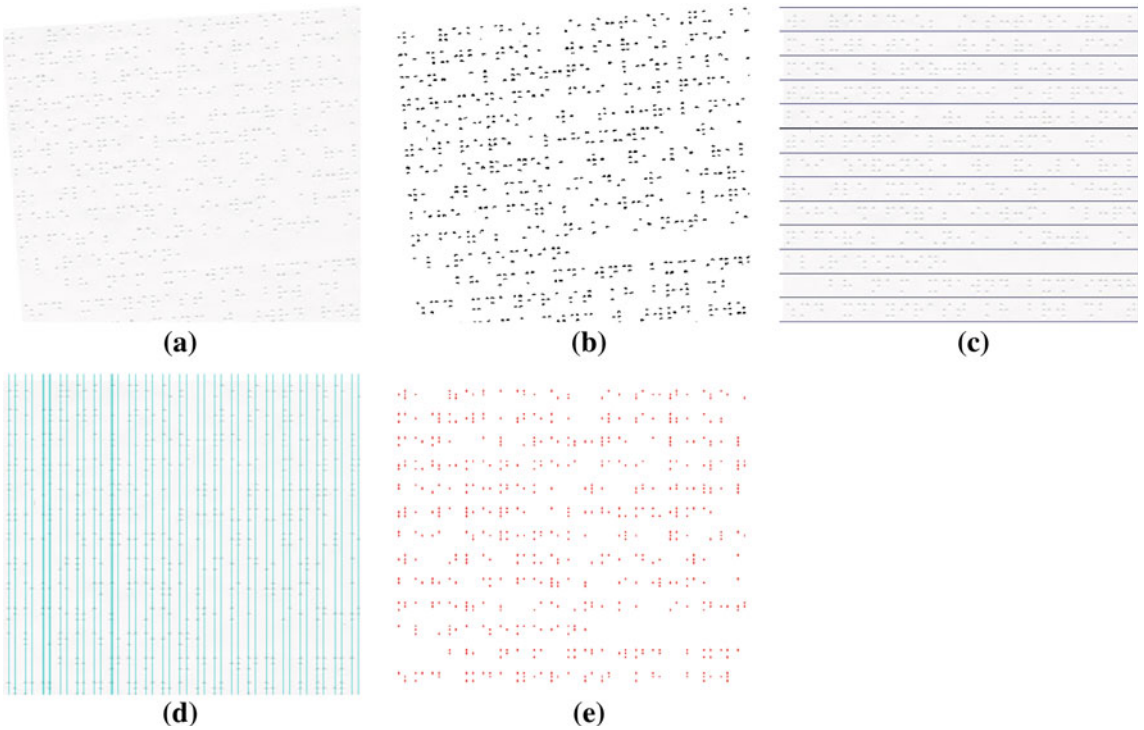

Fig. 6 **a** A portion of a Braille document with corresponding skewness of 5°, **b** the corresponding binary image, **c** the skewness is corrected and lines are separated, **d** the line parameters are detected, **e** the final estimation of Braille dots

are perpendicular to each other. Hence, we can choose one of them assuming skewness of the document does not exceed 45 degrees ($-\frac{\pi}{4} < \phi < \frac{\pi}{4}$). Once parameter $\mathbf{w}$ is found, b and β are calculated by solving (21), and then, parameter σ can be found using (20).

2.6 Line detection results

Once the parameters $\boldsymbol{\theta} = \{b, \beta, \mathbf{w}, \sigma\}$ are estimated, skewness of the document is corrected and individual lines can be separated. Figures 6 represents some results of the proposed algorithm. Figure 6a shows fragments of the input image. This image was created by synthetically rotating a regular Braille document. Although the rotation of the documents with software does not seem standard for evaluation and it is really not standard, the results show the robustness of our system against slight changes in the scanner's angle of illumination.

In Fig. 6c for the input image, skewness is corrected and lines are separated using the parameters estimated by the proposed algorithm (horizontal lines are in the form of $y_l = b + l\beta - \frac{\beta}{2}$, $(l = 0, 1, \ldots, L-1)$. Initial values for the parameters are naively selected as: $b = 0$, $\beta = (\text{Image Height})/L$, $\sigma = \frac{\beta}{4}$, $w_1 = 0$, and $w_2 = 1$.

In the proposed approach, number of lines of the document (L) is assumed to be known. The problem can be solved by executing the parameter estimation algorithm with different possible values of L and selecting that L, which maximizes the log-likelihood. In Fig. 7 the log-likelihood value is plotted for $L = 1, 2, \ldots, 40$ after applying the algorithm to a Braille document with 22 lines. Obviously, an outstanding peak is evident at the true value of L. Therefore, the actual number of lines can be estimated efficiently using a good initial estimation of L and testing a range of values around it. An important observation here is that we can have a fairly exact estimation of the skewness even by presuming a non-exact value for the number of lines. Note that knowing the skewness parameters can considerably improve our initial estimation of the number of lines.

2.7 Noise removal

In Sect. 2.4, we derived a Bayesian classifier to give us the posterior probability of each connected component as being a Braille dot or noise given its extracted features. At this point, however, we have extra information that is the location of each connected component along with the distribution of the connected components. Therefore, its is better to separate noise component from the Braille dots using $P(C_i \in \text{dot}|\mathbf{F}_i, \mathbf{x}_i, \pi_i, \boldsymbol{\theta})$ rather than $P(C_i \in \text{dot}|\mathbf{F}_i)$. This probability can be calculated Eq. (24).

$$
\begin{aligned}
&P(C_i \in \text{dot}|\mathbf{F}_i, \mathbf{x}_i, \pi_i, \boldsymbol{\theta}) \\
&= \frac{P(C_i \in \text{dot}, \mathbf{x}_i|\mathbf{F}_i, \pi_i, \boldsymbol{\theta})}{P(\mathbf{x}_i|\mathbf{F}_i, \pi_i, \boldsymbol{\theta})} \\
&= \frac{P(C_i \in \text{dot}, \mathbf{x}_i|\mathbf{F}_i, \pi_i, \boldsymbol{\theta})}{P(C_i \in \text{dot}, \mathbf{x}_i|\mathbf{F}_i, \pi_i, \boldsymbol{\theta}) + P(C_i \in \text{noise}, \mathbf{x}_i|\mathbf{F}_i, \pi_i, \boldsymbol{\theta})} \\
&= \frac{p(\mathbf{x}_i|C_i \in \text{dot}, \mathbf{F}_i, \pi_i, \boldsymbol{\theta}) P(C_i \in \text{dot}|\mathbf{F}_i, \pi_i, \boldsymbol{\theta})}{p(\mathbf{x}_i|C_i \in \text{dot}, \mathbf{F}_i, \pi_i, \boldsymbol{\theta}) P(C_i \in \text{dot}|\mathbf{F}_i, \pi_i, \boldsymbol{\theta}) + p(\mathbf{x}_i|C_i \in \text{noise}, \mathbf{F}_i, \pi_i, \boldsymbol{\theta}) P(C_i \in \text{noise}|\mathbf{F}_i, \pi_i, \boldsymbol{\theta})} \\
&= \frac{p(\mathbf{x}_i|C_i \in \text{dot}, \boldsymbol{\theta}) P(C_i \in \text{dot}|\mathbf{F}_i, \pi_i)}{p(\mathbf{x}_i|C_i \in \text{dot}, \boldsymbol{\theta}) P(C_i \in \text{dot}|\mathbf{F}_i, \pi_i) + p(\mathbf{x}_i|C_i \in \text{noise}, \boldsymbol{\theta}) P(C_i \in \text{noise}|\mathbf{F}_i, \pi_i)} \\
&= \frac{p(\mathbf{x}_i|C_i \in \text{dot}, \boldsymbol{\theta})\, \pi_i}{p(\mathbf{x}_i|C_i \in \text{dot}, \boldsymbol{\theta})\, \pi_i + p(\mathbf{x}_i|C_i \in \text{noise}, \boldsymbol{\theta})\, (1 - \pi_i)}
\end{aligned}
\tag{24}
$$

We simply remove each connected component C_i whose probability $P(C_i \in \text{dot}|\mathbf{F}_i, \mathbf{x}_i, \pi_i, \boldsymbol{\theta}) < 0.5$ as noise. This way the location of each connected component is also taken into account for noise detection.

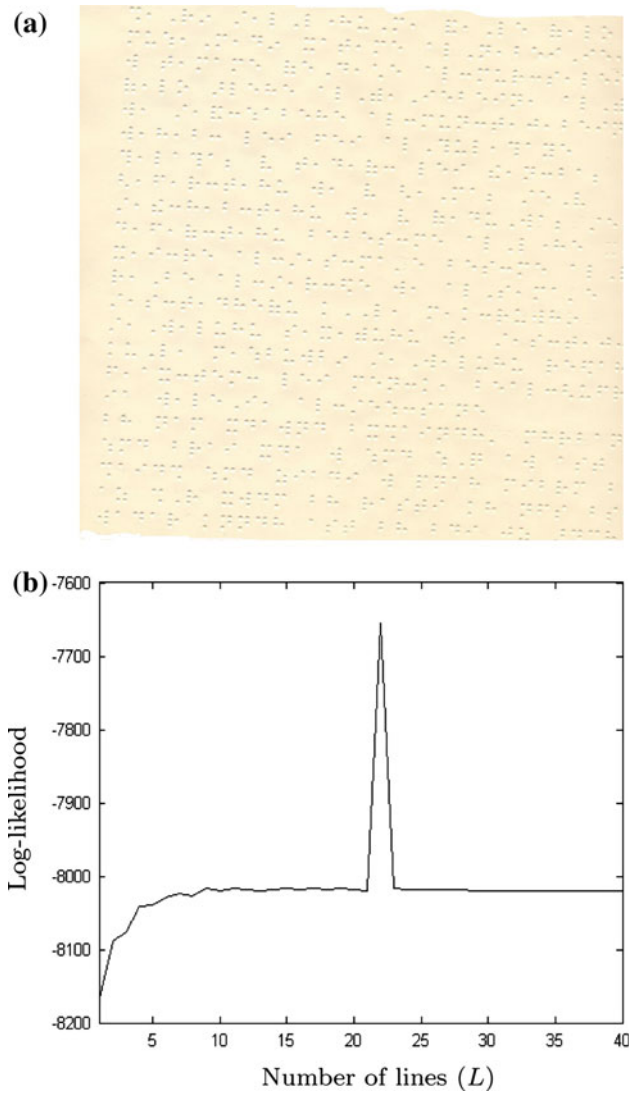

Fig. 7 **b** panel shows log-likelihood values of the trained models for different assumptions of L (*i.e.,* $L = 1, 2, \ldots, 40$). A peak stands out at $L = 22$, the true value of L, number of lines in the braille documents shown in **a** panel

3 Character detection

3.1 Row separation

In this part, it is assumed that the skewness and the true number of lines are estimated; this estimation can be done by any other algorithms like Radon transform and Hough transform; of course, the results of Radon transform and Hough transform are not accurate enough. These methods have about 2°–3° error in skewness estimation which is not satisfying for the task of finding the true number of lines. Since each Braille character is arranged as a 3×2 matrix (see Fig. 1) of dots, we can say that each Braille line is comprised of three rows. Assume that the skewness is corrected and lines are separated in a typical Braille document. Now, if the Braille dots are projected to the vertical axis, the distribution of the projected dots of each single line of the Braille document is in fact something like a mixture of three Gaussians, each representing one row of that Braille line. In the previous stage, however, these projections is modeled as a single Gaussian for each line to avoid model complexity and consecutively a slower convergence and trapping in local minima.

To separate the three rows of each line, we need to estimate the vertical distance between the rows. To do this, first, the Braille dots are projected to the vertical axis and the standard deviation of these projected dots is calculated for each Braille line. Let us denote the average value of the standard deviation of Braille lines by $\bar{\sigma}$ and the vertical distance between two rows of a single Braille line by τ. As mentioned before, the vertical projection of each Braille line can be represented by a mixture of three Gaussians each with a relatively small variance. The distance between the center of two consecutive Gaussians is τ. Here, to simplify the process, an approximation is used and it is assumed that the vertical projection of each line is represented by a mixture of three *Dirac delta* functions, *i.e.*, $\frac{1}{3}\sum_{i=-1}^{1} \delta(y - \mu_l - i\tau)$ for the line number l, where $\mu_l = b + l\beta$ is the center of the line number l. The standard deviation of the above mixture is $\sqrt{2/3}\tau$. Equating this quantity to $\bar{\sigma}$ yields $\tau = \sqrt{3/2}\bar{\sigma}$. Using τ along with the centers of the lines μ_l, the rows of each Braille line can be easily separated.

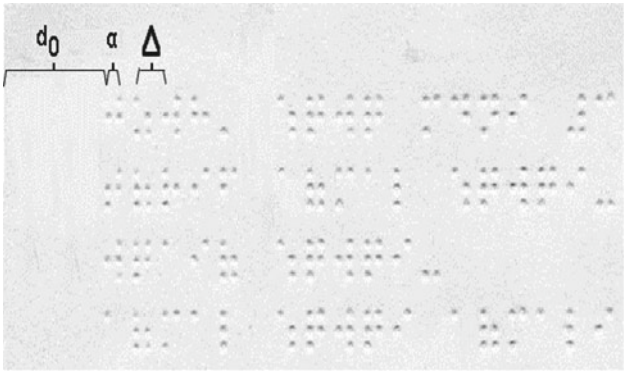

Fig. 8 Braille horizontal parameters

3.2 Estimating character parameters using HMM

By now, the vertical parameters of the Braille document are calculated. These parameters are the bias of the center of the first line from the top of the page (b), the distance between lines of the document (β), and the distance between the rows of each line (τ). To recognize Braille characters, we need to have the horizontal Braille parameters, as well. These parameters are the horizontal offset where the first character in a line stands (denoted by d_0 here), the distance between two consecutive characters (Δ), and the distance between the two columns of a single character (α). These parameters are shown in Fig. 8. Here, a first-order Hidden Markov Model (HMM) is used to estimate these parameters.

We use the term *potential dot* to refer to possible dots in all 6 locations of a Braille character, either *on* or *off*. Those potential dots, which are *on* (really exist), in the document are simply referred as *dots*. After separating rows of every line of the document, for each row, we have a list of the horizontal locations of its dots. Let us denote the horizontal location of the tth dot of the rth row of the document by x_t^r and that of an arbitrary row by x_t. These are our *observations*. Although our observation sequences have a spatial nature, here, they are indexed by t to make our formulations consistent with those commonly used in HMMs. Further, we say the observation at *time* t to refer to the tth observation for each row. Our train data, therefore, are comprised of $3L$ sequences of observations, where L is the number of Braille lines.

At each time, in addition to an observation, we have a hidden state s_t^r that shows the number of potential dot to which x_t^r belongs (for an arbitrary row it is shown by s_t). Therefore, if the tth dot of the rth row belongs to the first (second) column of the ith character of the line, then s_t^r is $2i-1$ ($2i$). Our task is to estimate the aforementioned horizontal Braille parameters and the hidden states from the sequences of observations.

To build the model, here, an approximation is made which is to assume that the existence of a real dot in each potential dot position of the document is independent of the existence of other dots in other positions. It is also assumed that the probability of each potential dot being at state *on* is $\frac{1}{2}$. This stems from the fact that we make no assumption about the type of the Braille document and have to choose the distribution with maximum entropy. Of course, more accurate probabilities can be obtained for certain classes of Braille documents (*e.g.*, document of a certain language) by applying different kinds of statistical analyses.

With aforementioned assumptions, we show that our model represents a *time-invariant first-order Hidden Markov process*. We start by the probabilities of state transitions. In a certain row, given the fact that the dot at time $t-1$ belongs to the jth potential dot, the probability of the tth dot to belong to ith potential dot is:

$$P(S_t = i | S_{t-1} = j) = \begin{cases} (\frac{1}{2})^{(i-j)} & \text{if } i < j, \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

Note that for $i < j$, it is equal to the probability of all potential dots $i+1, i+2, \ldots, j-1$ are *off* and the jth potential dot is *on*. With this argument, it is obvious that the value of s_t is conditionally independent of states $\{s_{t'} | t' < t-1\}$ given the value of s_t. It is also obvious that the state transitions are homogeneous (time-invariant). Furthermore, the initial state probability is:

$$P(s_1 = i) = \left(\frac{1}{2}\right)^i. \tag{26}$$

Ideally, each observation is the horizontal location of the potential dot to which the Braille dot belongs. However, in practice, each observation is supposed to have a small deviation from its true position. Here, this deviation is shown by a random Gaussian error with a variance of σ. Note that, σ is also an unknown parameter of our model, which is determined in the training process. Hence, the observation distribution is:

$$p(x_t | s_t = i) = \mathcal{N}(x_t; \mathrm{d}(i), \sigma) \tag{27}$$

where $\mathrm{d}(i)$ is the horizontal location of the ith potential dot and is easily obtained as $\mathrm{d}(i) = d_0 + a_i\,\Delta + b_i\,\alpha$, where $a_i = \lfloor \frac{i-1}{2} \rfloor$ and $b_i = (i+1)\%2$. It is also clear that with our assumptions, each observation is conditionally independent of other states and observations given their corresponding state values. All these facts show that our model represents a time-invariant first-order Hidden Markov process. By training this model with our sequences of observations as train data, we can obtain our desired parameters. Here, again, the *expectation-maximization* algorithm is used to train the model.

3.2.1 Making the Q function

Let us denote $\mathbf{X}^r$ and $\mathbf{S}^r$ the observation sequence and the state sequence of the row r and by $\mathbf{X}_t$ and $\mathbf{S}_t$ the total set of observations and the total set of states, respectively. The model parameters are shown by $\boldsymbol{\theta} = \{d_0, \alpha, \Delta, \sigma\}$, and the parameters at the previous stage of the algorithm are denoted by $\boldsymbol{\theta}'$. The Q function can then be obtained as:

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}') &= E_{\mathbf{S}_t}\{\log P(\mathbf{X}_t, \mathbf{S}_t|\boldsymbol{\theta})|\mathbf{X}_t\} \\ &= E_{\mathbf{S}_t}\left\{\log \prod_r P(\mathbf{X}^r, \mathbf{S}^r|\boldsymbol{\theta})|\mathbf{X}_t\right\} \\ &= E_{\mathbf{S}_t}\left\{\sum_r \log P(\mathbf{X}^r, \mathbf{S}^r|\boldsymbol{\theta})|\mathbf{X}_t\right\} \\ &= \sum_r E_{\mathbf{S}_t}\left\{\log P(\mathbf{X}^r, \mathbf{S}^r|\boldsymbol{\theta})|\mathbf{X}_t\right\} \\ &= \sum_r E_{\mathbf{S}^r}\left\{\log P(\mathbf{X}^r, \mathbf{S}^r|\boldsymbol{\theta})|\mathbf{X}^r\right\} \end{aligned} \tag{28}$$

Representing the sequence of observations and the sequence of states of a typical row by $\mathbf{X}$ and $\mathbf{S}$, the term inside the summation above is written in Eq. (29).

$$\begin{aligned} &E_{\mathbf{S}}\{\log p(\mathbf{X}, \mathbf{S}|\boldsymbol{\theta})|\mathbf{X}\} \\ &= E_{\mathbf{S}}\{\log p(x_1, \ldots, x_T, s_1, \ldots, s_T|\boldsymbol{\theta})|\mathbf{X}\} \\ &= E_{\mathbf{S}}\left\{\log\left(\prod_{t=1}^{T} p(x_t|s_t, \boldsymbol{\theta}) \prod_{t=2}^{T} P(s_t|s_{t-1}, \boldsymbol{\theta}) P(s_1|\boldsymbol{\theta})\right)|\mathbf{X}\right\} \\ &= E_{\mathbf{S}}\left\{\left(\sum_{t=1}^{T} \log p(x_t|s_t, \boldsymbol{\theta})\right. \right. \\ &\quad \left.\left. + \sum_{t=2}^{T} \log P(s_t|s_{t-1}, \boldsymbol{\theta}) + \log P(s_1|\boldsymbol{\theta})\right)|\mathbf{X}\right\} \\ &= \sum_{t=1}^{T} E_{\mathbf{S}}\{\log p(x_t|s_t, \boldsymbol{\theta})|\mathbf{X}\} \\ &\quad + \sum_{t=2}^{T} E_{\mathbf{S}}\{\log P(s_t|s_{t-1}, \boldsymbol{\theta})|\mathbf{X}\} + E_{\mathbf{S}}\{\log P(s_1|\boldsymbol{\theta})|\mathbf{X}\} \\ &= \sum_{t=1}^{T} E_{s_t}\{\log p(x_t|s_t, \boldsymbol{\theta})|\mathbf{X}\} \\ &\quad + \sum_{t=2}^{T} E_{s_t, s_{t-1}}\{\log P(s_t|s_{t-1}, \boldsymbol{\theta})|\mathbf{X}\} \\ &\quad + E_{s_1}\{\log P(s_1|\boldsymbol{\theta})|\mathbf{X}\} \\ &= \sum_{t=1}^{T} E_{s_t}\{\log p(x_t|s_t, \boldsymbol{\theta})|\mathbf{X}\} + c_1 \\ &= \sum_{t=1}^{T}\sum_{i=1}^{\infty} \log p(x_t|s_t = i, \boldsymbol{\theta})\ \ P(s_t = i|\mathbf{X}, \boldsymbol{\theta}') + c_1 \\ &= \sum_{t=1}^{T}\sum_{i=1}^{\infty} \log \mathcal{N}(x_t; \mathrm{d}(i), \sigma)\ \ P(s_t = i|\mathbf{X}, \boldsymbol{\theta}') + c_1 \\ &= \sum_{t=1}^{T}\sum_{i=1}^{\infty} \left(-\log\sigma - \frac{1}{2}\frac{(x_t - \mathrm{d}(i))^2}{\sigma^2}\right) \\ &\quad \times P(s_t = i|\mathbf{X}, \boldsymbol{\theta}') + c_2 \end{aligned} \tag{29}$$

where c_1 and c_2 are constants with respect to our parameters. Substituting (29) into (28) gives Eq. (30):

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}') &= \sum_{r=1}^{3L}\sum_{t=1}^{T}\sum_{i=1}^{\infty} \left(-\log\sigma - \frac{1}{2}\frac{(x_t^r - \mathrm{d}(i))^2}{\sigma^2}\right) \\ &\quad \times P(s_t^r = i|\mathbf{X}^r, \boldsymbol{\theta}') \ + c_3 \\ &= \sum_{r=1}^{3L}\sum_{t=1}^{T}\sum_{i=1}^{\infty} \left(-\log\sigma - \frac{1}{2}\frac{(x_t^r - \mathrm{d}(i))^2}{\sigma^2}\right) P_{rti} + c_3 \end{aligned} \tag{30}$$

where $P_{rti} = P(s_t^r = i|\mathbf{X}^r, \boldsymbol{\theta}')$ and c_3 is constant with respect to the parameters.

3.2.2 Expectation step

(This section can be eliminated since the derivations are common in all types of HMMs) In the expectation step, $P_{rti} = P(s_t^r = i|\mathbf{X}^r, \boldsymbol{\theta}')$ should be obtained. Since P_{rti} can be obtained independently for each row, here, for simplicity, it is obtained for a typical row and the super index r is dropped from the next formulations:

$$\begin{aligned} P(s_t = i|\mathbf{X}, \boldsymbol{\theta}') &= \frac{P(s_t = i, \mathbf{X}|\boldsymbol{\theta}')}{P(\mathbf{X}|\boldsymbol{\theta}')} \\ &= \frac{P(s_t = i, \mathbf{X}|\boldsymbol{\theta}')}{\sum_{i=1}^{\infty} P(s_t = i, \mathbf{X}|\boldsymbol{\theta}')} \end{aligned} \tag{31}$$

Now, $P(s_t = i, \mathbf{X}|\boldsymbol{\theta}')$ can be obtained Eq. (32): where $P(x_1, \ldots, x_t, s_t = k|\boldsymbol{\theta}')$ and $p(x_{t+1}, \ldots, x_T|s_t = k, \boldsymbol{\theta}')$ are calculated in Eqs. (33) and (34).

3.2.3 Maximization step

In this step, we take the derivative of $Q(\boldsymbol{\theta}|\boldsymbol{\theta}')$ in (30) with respect to each of the parameters and set it equal

$$\begin{aligned} P(s_t = i, \mathbf{X}|\boldsymbol{\theta}') &= P(s_t = i, x_1, \ldots, x_T|\boldsymbol{\theta}') \\ &= P(x_1, \ldots, x_t, s_t = k|\boldsymbol{\theta}') \\ &\quad \times P(x_{t+1}, \ldots, x_T|x_1, \ldots, x_t, s_t = k, \boldsymbol{\theta}') \\ &= P(x_1, \ldots, x_t, s_t = k|\boldsymbol{\theta}') \\ &\quad \times P(x_{t+1}, \ldots, x_T|s_t = k, \boldsymbol{\theta}') \end{aligned} \tag{32}$$

$$
\begin{aligned}
P(x_1,\ldots,x_t,s_t=i|\boldsymbol{\theta}') &= P(x_1,\ldots,x_{t-1},s_t=i|\boldsymbol{\theta}')\\
&\quad\times p(x_t|x_1,\ldots,x_{t-1},s_t=i,\boldsymbol{\theta}')\\
&= \sum_{j=1}^{\infty}\left\{P(s_t=i,s_{t-1}=j,x_1,\ldots,x_{t-1}|\boldsymbol{\theta}')\right\}\\
&\quad\times p(x_t|s_t=i,\boldsymbol{\theta}')\\
&= \sum_{j=1}^{\infty}\left\{P(s_t=i|s_{t-1}=j,x_1,\ldots,x_{t-1},\boldsymbol{\theta}')\right.\\
&\quad\left.\times\ P(x_1,\ldots,x_{t-1},s_{t-1}=j|\boldsymbol{\theta}')\right\}\\
&\quad\times p(x_t|s_t=i,\boldsymbol{\theta}') \qquad (33)
\end{aligned}
$$

$$
\begin{aligned}
&p(x_{t+1},\ldots,x_T|s_t=i,\boldsymbol{\theta}')\\
&= \sum_{j=1}^{\infty}\left\{P(x_{t+1},\ldots,x_T,s_{t+1}=j|s_t=i,\boldsymbol{\theta}')\right\}\\
&= \sum_{j=1}^{\infty}p(x_{t+1},\ldots,x_T|s_t=i,s_{t+1}=j,\boldsymbol{\theta}')\\
&\quad\times P(s_{t+1}=j|s_t=i,\boldsymbol{\theta}')\\
&= \sum_{j=1}^{\infty}\left\{p(x_{t+2},\ldots,x_T|s_{t+1}=j,\boldsymbol{\theta}')\right.\\
&\quad\left.\times\ p(x_{t+1}|s_{t+1}=j,\boldsymbol{\theta}')\right\}P(s_{t+1}=j|s_t=i,\boldsymbol{\theta}') \qquad (34)
\end{aligned}
$$

to zero to obtain its maximum value. Differentiating with respect to d_0, Δ and α, respectively, yield:

$$\sum_r\sum_t\sum_i(d_t-(d_0+a_i\Delta+b_i\alpha))P_{rti}=0 \qquad (35)$$

$$\sum_r\sum_t\sum_i(d_t-(d_0+a_i\Delta+b_i\alpha))P_{rti}a_i=0 \qquad (36)$$

$$\sum_r\sum_t\sum_i(d_t-(d_0+a_i\Delta+b_i\alpha))P_{rti}b_i=0 \qquad (37)$$

that form a set of linear equations solution of which gives d_0, α and Δ. Also, by differentiating w.r.t σ, we get:

$$\sum_r\sum_t\sum_i\left(-1+\frac{(d_t-\mathrm{d}(i))^2}{\sigma_2}\right)P_{rti}=0 \qquad (38)$$

And then, we can calculate σ by:

$$\sigma^2=\frac{\sum_r\sum_t\sum_i(d_t-\mathrm{d}(i))^2P_{irt}}{\sum_r\sum_t\sum_i P_{rti}} \qquad (39)$$

3.2.4 Initial parameters

Choosing a good value for the parameters is crucial for the convergence of the algorithm. To obtain d_0, first, we choose the minimum value of x_1^r over r is chosen as the initial value of d_0. To obtain the initial value of α, first, for each line, we calculate $d_1^r=\min_i x_{i+1}^r-x_i^r$. Then, the *median* of these values over r is chosen as the initial value of α. Similarly, the initial value of Δ is chosen as the median of the values d_2^r over r where $d_2^r=\min_i x_{i+2}^r-x_i^r$.

4 Experimental results

Our algorithm consists of two main parts; thus, we present the results in two separate steps. At first step, we run first part of our algorithms on two types of documents containing Braille documents and normal documents. We applied first part of our algorithms to detect skewness of each document. Although we proposed this algorithm for Braille (as a well-structured document), it is observed that it performs very accurately on normal text documents (see Fig. 9). In our experiments, we used 10 normal documents and 62 Braille documents with −25°–25° of rotation (−25°, −15°, 0°, 15° and 25°) with different page scales. These Braille documents are scanned by the HP G4010 scanner with resolution 72 dpi and rotated by image rotation algorithm. For all of these documents, our algorithm found exact number of lines and the correct skewness. Of course, we know the true degree of rotation as the rotation is performed by software. We assume that less than one degree errors are acceptable since, in this case, the dots of each line are assigned to correct row. Next, we applied second part of our algorithm to the results of Braille documents achieved in the previous step to detect their line parameters. As reported in Tables 1 and 2, the results indicate

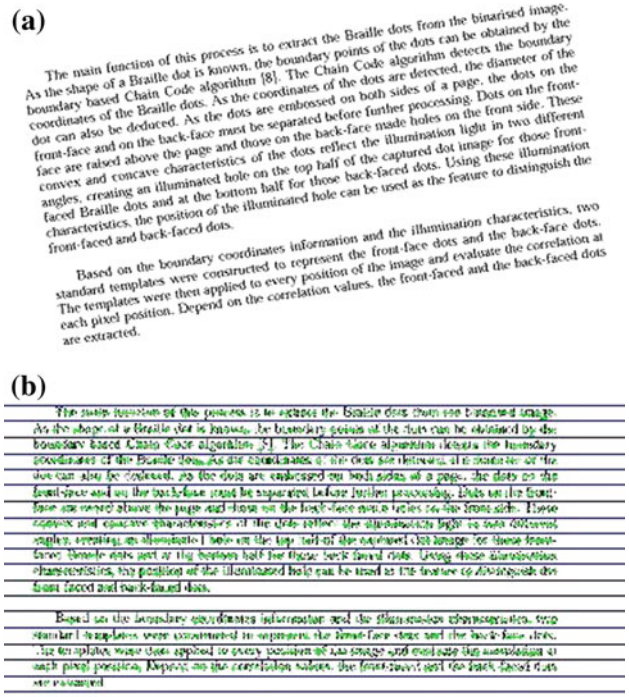

Fig. 9 **a** Fragment of a text document with a skewness of 10°, **b** the skewness is corrected and lines are separated

Table 1 Result of algorithm on text and Braille documents

Type of documents	Number of test documents	Correctly detected skewness	Correctly detected number of lines	Correctly estimated line parameters
Normal text	10	10	10	—
Braille document	62	62 (100%)	60 (96.8%)	58 of 60 Correctly Line detected(96.7%)

Values in parenthesis show the algorithm accuracy for Braille documents

Table 2 Results of running the proposed algorithm on ten randomly selected Braille documents

Test documents	Number of lines	Characters per line	Number of characters	Number of mistakes in character detection
Doc. 15	22	29	638	3
Doc. 17	22	31	682	0
Doc. 23	22	27	594	0
Doc. 34	21	28	588	1
Doc. 43	22	27	594	1
Doc. 45	22	27	594	1
Doc. 48	21	26	546	0
Doc. 52	21	26	546	1
Doc. 53	21	26	546	1
Doc. 57	22	27	594	1
Total	216	274	5922	9

The documents were selected from those for which true skewness and line detection are available

that, in 95% of documents, the parameters are calculated correctly, and we can convert any Braille character to its corresponding character in a regular text. These experiments indicate superiority of the proposed in comparison with the existing Braille character detectors, among which the best result was reported to be 87% accuracy [10]. Also, a noisy document is used to show how the proposed method works on noisy documents. Because the scanned documents did not contain any clear noise, a document corrupted by the ink spots and salt and pepper noises was added to artificially create a noisy document. Also, we added some dirty spots to the document. Figure 10 shows this noisy document and the obtained estimations. The artificial noise here is only used to show the robustness of the method against certain types of noise and artifacts. However, it is not necessarily representative of the typical noise on Braille documents which is usually concentrated on the Braille lines due to finger smudging during reading and also big blobs (comparable in size to the braille dots) in the cases that recycled paper is used.

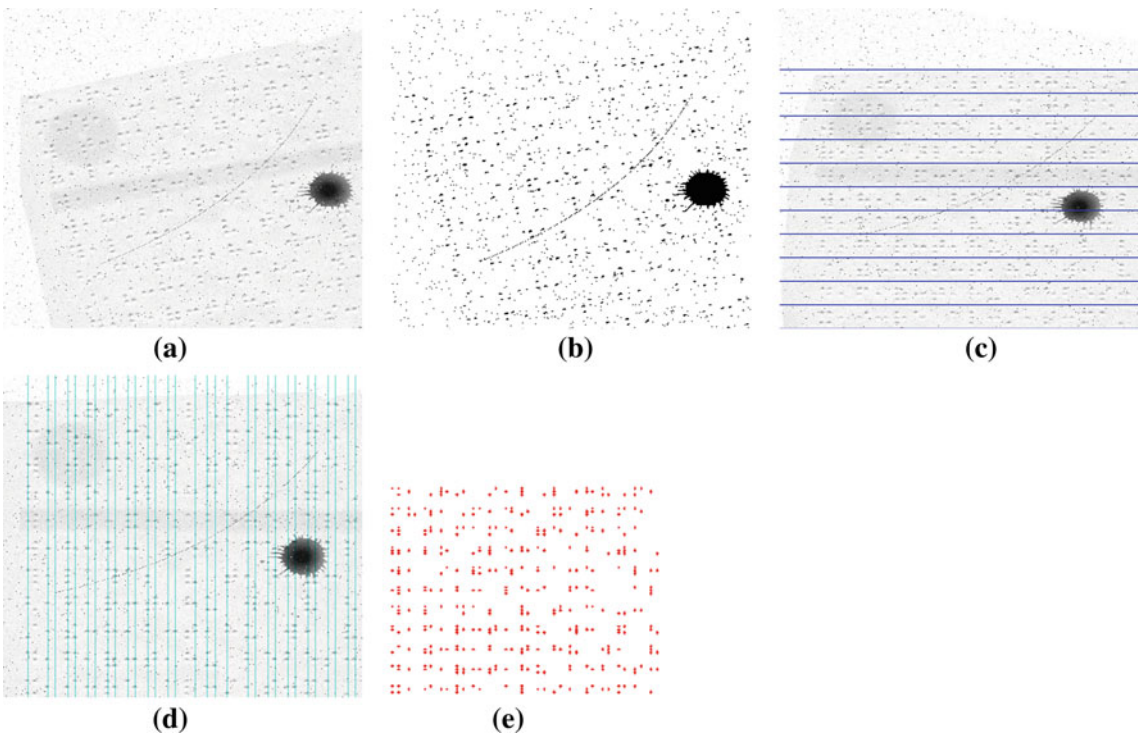

Fig. 10 **a** A portion of a noisy Braille document with corresponding skewness of 15°, **b** the corresponding binary image, **c** the skewness is corrected and lines are separated, **d** the line parameters are detected, **e** the final estimation of Braille dots

5 Conclusion and further work

A set of probabilistic approaches to robust recognition of the Braille documents are proposed. The detection process can be divided into two major stages. In the first stage, the vertical parameters of the Braille document (namely bias of the first line from the top of the page, the vertical distance between the Braille lines and the vertical distance between the rows of each line) is detected as well as the skewness of the Braille document. The detection algorithm was to fit a special probability density function to the Braille data by maximizing the likelihood using the EM algorithm. To make the algorithm robust against undesired artifacts, a noise model was added to the formulations of the above density functions.

The second stage was devoted to estimate the horizontal Braille parameters such as the horizontal bias of the beginning of each line from the left edge of the document, the distance between two consecutive Braille characters, and the distance between the two columns of each Braille character. To do this, the horizontal location of the Braille dots for each of the three rows of all the Braille lines is extracted as sequences of data. These sequences were modeled as samples drown from a first-order Hidden Markov process. The parameters were determined by training the HMM model using these sequences of data. These horizontal and vertical Braille parameters are adequate to detect all Braille characters in a typical document.

It is worthwhile to notice that the well-structured nature of the Braille documents distinguishes it from many other categories of documents. By considering this fact, here, it was sought to find a methods that is very general in making very few presumptions about the Braille document. For example, the method is designed to be scale free and robust against different types of noise and artifacts. It is also tried to avoid heuristic and ad hoc treatments as much as possible to obtain a general, purely probabilistic approach to Braille detection.

The algorithm is successfully executed on an ordinary personal computer in a small amount of time. It is about six seconds to find all the connected components, after finding the connected components it takes about seven seconds to find the skewness and the number of lines and finally about 1 second to find the line parameters. These results are obtained by a two-processor system with 2.5-GHz frequency and 4-GB RAM. Also the algorithm is implemented by MATLAB.

The approach may be generalized to treat two-sided Braille document. This is possible by modifying the classification phase of the current algorithm by classifying dots as *side1, side2, and noise* rather than *dot and noise*.

References

1. Kato, Y., Sekitani, T., Takamiya, M., Doi, M., Asaka, K., Sakurai, T., Someya, T.: Sheet-type Braille displays by integrating organic field-effect transistors and polymeric actuators. IEEE Trans. Electron Devices **54**(2), 202–209 (2007). doi:10.1109/TED.2006
2. Takamiya, M., Sekitani, T., Kato, Y., Kawaguchi, H., Someya, T., Sakurai, T.: An organic FET SRAM with back gate to increase static noise margin and its application to Braille sheet display. IEEE J. Solid-State Circuits **42**(1), 93–100 (2007). doi:10.1109/JSSC.2006
3. Rantala, J., Raisamo, R., Lylykangas, J., Surakka, V., Raisamo, J., Salminen, K., Pakkanen, T., Hippula, A.: Methods for presenting Braille characters on a mobile device with a touchscreen and tactile feedback. IEEE Trans. Haptics **2**(1), 28–39 (2009). doi:10.1109/TOH.2009
4. Lee, J.S., Lucyszyn, S.: A micromachined refreshable Braille cell. J. Microelectromech. Syst. **14**(4), 673–682 (2005). doi:10.1109/JMEMS.2005
5. Sribunruangrit, N., Marque, C.K., Lenay, C., Hanneton, S., Gapenne, O., Vanhoutte, C.: Speed-accuracy tradeoff during performance of a tracking task without visual feedback. IEEE Trans. Neural Syst. Rehabil. Eng. **12**(1), 131–139 (2004). doi:10.1109/TNSRE.2004
6. Yobas, L., Durand, D.M., Skebe, G.G., Lisy, F.J., Huff, M.A.: A novel integrable microvalve for refreshable Braille display system. J. Microelectromech. Syst. **12**(3), 252–263 (2003). doi:10.1109/JMEMS.2003
7. Tanaka, M., Miyata, K., Chonan, S.: A wearable Braille sensor system with a post processing. IEEE/ASME Trans. Mechatron. **12**(4), 430–438 (2007). doi:10.1109/TMECH.2007
8. Blenkhorn, P., Evans, G.: Automated Braille production from word-processed documents. IEEE Trans. Neural Syst. Rehabil. Eng. **9**(1), 81–85 (2001)
9. Krufka, S.E., Barner, K.E., Aysal, T.C.: Visual to tactile conversion of vector graphics. IEEE Trans. Neural Syst. Rehabil. Eng. **15**(2), 310–321 (2007)
10. Namba, M., Zhang, Z.: Cellular neural network for associative memory and its application to Braille image recognition. Neural Networks, 2006. IJCNN '06. International Joint Conference on 16–21 July, pp. 2409–2414 (2006)
11. Franois, G., Calders, P.: The reproduction of Braille originals by means of optical pattern recognition. Proceedings 5th International Workshop on Computer Braille Production, Heverlee (1985)
12. Dubus, J.P., Benjelloun, M., Devlaminck, V., Wauquier, F., Altmayer, P.: Image processing techniques to perform an autonomous system to translate relief Braille into black-ink. Proceedings IEEE Engineering in Medicine and Biology Society 10th Annual International Conference (1988)
13. Hentzschel, T.W.: An optical Braille reading system. M.Sc. Dissertation, Faculty of Technology, University of Manchester (1992)
14. Ritchings, R.T., Antonacopoulos, A., Drakopoulos, D.: Analysis of scanned Braille documents. Proceedings International Association for Pattern Recognition Workshop on Document Analysis Systems (1994)
15. Antonacopoulos, A., Bridson, D.: A robust Braille recognition system. Proceedings of the IAPR International Workshop on Document Analysis Systems, Italy (2004)
16. Tai, Z., Cheng, S., Verma, P.: Braille Document Parameters Estimation for Optical Character Recognition. Springer, Berlin (2008)
17. Al-Saleh, A., El-Zaart, A., AlSalman, A.: Dot Detection of Optical Braille Images for Braille Cells Recognition. Springer, Berlin (2008)
18. Yoosefi Babadi, M., Nasihatkon, B., Azimifar, Z., Fieguth, P.: Probabilistic estimation of Braille document parameters. IEEE International Conference on Image Processing, 2001–2004 (2009)

19. Dempster, A., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Royal Stat. Soc. **39**, 1–38 (1977)

20. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Berlin (2006)