Using Gaussian Kernels to Remove Uneven Shading from a Document Image

Xiaohua Zhang (member)†, Yuelan Xin ‡‡, Heming Huang ‡‡, Ning Xie ‡‡‡

Abstract Segmentation of document images into text or drawings is an important process, which is often related to binarization of a document image to perform character recognition and document analysis. This process is easier to do using a document image with a uniform background and illuminated under well-conditioned lighting. However, when a document image is very unevenly shaded, binarization becomes very difficult or even impossible. An effective method is to remove the shading prior to binarization. In this paper, we propose a novel method for estimating an unevenly shaded surface in an image obtained under poor illumination. A one-dimensional Gaussian kernel model is applied in both the horizontal and vertical directions to estimate the background surface, allowing uneven shading to be removed from the document image. Thereafter, the image can easily be binarized. Results of experiments conducted on many document images demonstrate that our method yields better results than other methods.

Key words: Binarization, Document Image, Uneven Shading, Gaussian Kernel, Surface Estimation.

1. Introduction

Document images are often scanned or taken from media such as papers and books. Document analysis gathers information from the image by means of a preprocessing stage to segment the text or drawings in the image. This segmentation task, referred to as binarization, is often accomplished by using one of the well-known thresholding techniques\(^1\)\(^2\). When the intensity histogram of a document image is bimodal, it is feasible to use a global threshold to classify the pixels into foreground and background. However, if the histogram is not bimodal, a global threshold is not effective for the task. For example, the image in Fig. 1(a) contains severely uneven shading caused by poor illumination. Since most of the text is almost indistinguishable from the background, the histogram is not distinctly bimodal. If the image is binarized using Otsu’s technique\(^3\), much of the background will be incorrectly segmented into dark components, as illustrated in Fig. 1(b).

Since a global threshold cannot be used to binarize an image for which the histogram is not bimodal, various local adaptive thresholding techniques\(^3\)\(^4\) have been proposed. These techniques binarize a document image by computing a local threshold based on the mean and standard deviation of the pixel intensities within a neighborhood. Among these techniques, Sauvola’s method\(^4\) is probably the most commonly used; however, despite achieving good performance, it requires that the parameter is appropriately tuned. Fig. 1(c) shows the result of applying Sauvola’s technique\(^4\), which is clearly better than that obtained by Otsu’s method. However, some background components are still segmented into the “dark” foreground owing to the sharp variations in the uneven shading.

It is clear that the above techniques do not work well enough when segmenting an image captured under poor illumination; the uneven shading hinders the segmentation. An effective scheme is to remove the uneven shading prior to binarization. In the past few decades, shading detection\(^5\)\(^6\) has been proposed by exploring edge information from color image sequences or video streams. Removing shadows from images\(^7\) has also been well studied; however, this requires first deriving an illumination invariant shadow-free image or camera color calibration. Guo et al. reported a method\(^8\) by finding the same illumination paired regions and building a graph of segments using a shadow model. A graph-cut algorithm is used to solve the labeling of
shadow and non-shadow regions. A comprehensive survey and review on shadow removal can be found in references\(^9\)\(^{10}\). Nevertheless, none of the above-mentioned techniques is able to remove severely uneven shading such as that shown in Fig. 1(a).

A number of thresholding techniques, aimed specifically at binarization of document images with uneven shading, have been proposed. Recently a new method\(^{11}\) was reported, which transforms an image into curvelet coefficients. After adjusting the histogram and denoising, an inverse curvelet transform is used; however, removal of the uneven shading is not that clean, as shown in Fig. 1(e). Gatos et al.\(^{12}\) computed the shading surface directly, using the thresholding result obtained by Sauvola’s technique. Motivated by the work of Gatos et al., Lu et al. reported an approach for binarizing poorly illuminated document images\(^{13}\)\(^{14}\)\(^{15}\). Their method uses a single-polynomial model to estimate variations in the shading surface. However, when an image contains uneven shading with severe variations, such as that in Fig. 1(a), a single polynomial function cannot effectively estimate the shading as illustrated in Fig. 1(d).

A piecewise polynomial function\(^{16}\)\(^{17}\) was proposed for estimating uneven shading. In this method, the image is divided into several partially overlapping blocks to reduce the discontinuities between the blocks. The shading surface in each block is then fitted using a single two-dimensional (2D) polynomial function. Obviously, the block size depends on the complexity of the shading. In most cases, this approach provides a good estimate. However, since the surfaces are independently estimated for each block, the estimated surface map may appear blocky, which has a significantly negative effect on the final binarization result.

In this paper, we propose a novel method for estimating unevenly shaded surfaces in a document image obtained under poor illumination. A linear combination of Gaussian kernels is used to estimate the uneven shading. The estimation is based on a robust mixed \(L_2\)- and \(L_1\)-constrained regularized minimization of weighted least-squares, which automatically excludes most “dark” pixels to improve the accuracy of the estimation. For a 2D image, only one-dimensional (1D) Gaussian kernels are used. The estimation first proceeds horizontally and then vertically, although this can be reversed. The estimated shading surface is used to construct a shading surface map, which is later used to normalize the original image by removing shading. An improved Sauvola’s technique employing edge information is applied in the final thresholding.

The contributions of this paper are as follows:
- We introduce a linear combination of 1D Gaussian kernels to model the uneven shading surface using a two-pass algorithm. A robust algorithm is utilized to estimate the combination coefficients.
- The estimated combination coefficients are used to construct a shading surface map and normalize the original image by removing shading.
- We enhance Sauvola’s technique by utilizing edge information to improve the thresholding accuracy.

The remainder of this paper is organized as follows. Section 2 describes some fundamental issues and gives an overview of the proposed algorithm. Section 3 explains our proposed method in detail. In Section 4, experimental results are reported to demonstrate the performance of the proposed method. Finally, our conclusions are presented in Section 5.

2. Overview

Given a document image captured under poor illumination conditions, a large part of the image is generally background, while a fraction of the image is occupied by “inked” pixels. Since our ultimate aim is to segment text or drawings from the background, in the following description, we only consider cases in which the input image is grayscale and the medium (such as paper) is of almost uniform color. We focus on removing the complicated uneven shading contained in the document image. To this end, the shading surface must first be estimated using an appropriate model. Several kinds of kernels can be used as basis functions to create a surface model. If the given data are periodic, trigonometric functions (such as sine or cosine) can be used as basis functions. However, shading in a document image is usually not periodic, which means that trigonometric functions are not suitable\(^{18}\)\(^{19}\). A single polynomial function model\(^{15}\) and piecewise-polynomial model\(^{17}\) have been reported, both of which are unable to estimate an uneven shading surface well. Given the above consideration, a Gaussian kernel is a better choice as a radial basis function\(^{20}\)\(^{21}\)\(^{22}\), the behavior of which depends on the sample data and thus, is typically used for building function approximation. The uneven shading surface can be modeled as

\[
f(x, y) = \sum_{j=1}^{m} \sum_{i=1}^{n} a_{ij} G((x, y), (x_j, y_i)),
\]

where \(m\) and \(n\) are, respectively, the height and width of the given image, \(a_{ij}\) is the combination coefficient,
and \(G((x,y),(x_j,y_i))\) is a 2D Gaussian kernel centered at \((x_j,y_i)\). Based on the observation that any point on the surface depends on \(m \times n\) Gaussian kernels, it is obvious that computing all points on the shading surface is time-consuming. To solve this problem, the 2D Gaussian kernel can be rewritten as \(G(x,x_j)G(y,y_i)\) in a separable way as the multiplication of two 1D Gaussian functions. Then, Eq. (1) becomes

\[
f(x,y) = \sum_{i=1}^{m} a_i G(y,y_i) \left( \sum_{j=1}^{n} a_j G(x,x_j) \right),
\]

which leads to a two-pass scheme that estimates the combination coefficients first by row and then by column as a refinement process. In the two-pass scheme, any point on the shading surface only depends on \(m + n\) Gaussian kernels, and the computational complexity is reduced from \(O(mn)\) to \(O(m + n)\) for each point.

The overall procedure for the proposed method is given below.

1. Estimate uneven shading surface in a given document image using a linear combination of 1D Gaussian kernels. The estimation is done in a two-pass manner with a robust algorithm utilized to estimate the combination coefficients for 1D data.
2. Construct a map of the shading surface by employing the estimated combination coefficients.
3. Normalize the document image by removing the available shading surface.
4. Threshold the normalized image using the enhanced Sauvola’s binarization technique that utilizes edge information and pixel statistics.

A flowchart of the overall procedure for the proposed algorithm is shown in Fig. 2. A box with rounded corners indicates a process, dotted arrows show control flow, a rectangular box represents data such as an image, a solid arrow depicts data flow, and a small circle with a cross inside denotes that two data inputs are used. We use the document image with uneven shading shown in Fig. 1(a) as a running example to explain the overall process of the proposed method.

3. Proposed Method

This section describes our method for estimating an uneven shading surface, constructing a surface map, removing the uneven shading, and binarizing a document image.

3.1 Gaussian Kernel Model

Gaussian kernels are often used to approximate data with continuous functions. Suppose there are \(n\) pixel sample locations \(\{x_i\}_{i=1}^{n}\) located at equal intervals in one dimension. The corresponding pixel intensities are \(\{f_i\}_{i=1}^{n}\), which may contain “inked” pixels as text, and “light” pixels as background. These pixels are shaded differently. Since only the shading surface

Fig. 1 Experimental results: (a) original document image with severe and uneven shading; (b), (c), (d), (e), and (f) results obtained using different methods.
needs to be estimated, the “inked” pixels are considered outliers. A linear Gaussian kernel model is defined as a linear combination of 1D Gaussian kernel functions

\[ f_a(x) = \sum_{j=1}^{n} a_j G(x, x_j), \]  

where \( a_j \) is the \( j \)-th parameter in vector \( a = (a_1, a_2, \ldots, a_n)^T \). The basis function is a 1D Gaussian function called a Gaussian kernel:

\[ G(x, c) = \exp(-\frac{\|x - c\|^2}{2\sigma^2}), \]

where \( \| \cdot \| \) denotes the \( L_2 \)-norm. Parameters \( \sigma \) and \( c \) denote the bandwidth and the center of the Gaussian kernel, respectively. The bandwidth parameter \( \sigma \) is sensitive to the estimation performance. If it is too small, over-fitting generally occurs and the model even fits “inked” pixels; if the bandwidth is too large, under-fitting occurs and the model is not sufficient to fit the background shading in the training samples. Since the bandwidth depends on the interval of consecutive sample location, to solve the over- and under-fitting problems, all pixel coordinates are normalized in the range \([-\sqrt{2}, \sqrt{2}] \). We found that this normalization allows the bandwidth to be set around 0.1; it was fixed at 0.1 in all our experiments. Parameter \( c \) is the location of a pixel. Model fitting attempts to minimize the least squares between the outputs of the model \( f_a(x_i) \) and the training data \( \{f_i\}_{i=1}^{n} \) in order to learn parameter \( a \):

\[ E(a) = \frac{1}{2} \sum_{i=1}^{n} \tilde{e}_i^2 = \frac{1}{2} \sum_{i=1}^{n} (f_a(x_i) - f_i)^2. \]

Coefficient 1/2 can be canceled for the derivatives. The above is a typical least-squares question for computing coefficient \( a \). However, the least-squares method has a weakness in that it is sensitive to noise,\(^{22}\), that is, the outliers in the case of shading surface estimation. To reduce the influence of noise, a robust learning algorithm can be used to minimize the Huber loss,\(^{24}\), defined as

\[ E(a) = \sum_{i=1}^{n} \rho(e_i). \]

The Huber loss function mixes the \( L_2 \)- and \( L_1 \)-losses, and is defined as

\[ \rho(e) = \begin{cases} \frac{e^2}{2} & |e| \leq \eta \\ \eta |e| - \frac{\eta^2}{2} & |e| > \eta \end{cases}, \]

where parameter \( \eta \) controls the weight of the data. The weight expresses how much each datum contributes to the model. When the absolute of the residue \( |e| \) is less than threshold \( \eta \), the training data are less noisy, and so the \( L_2 \)-norm is used. Conversely, when \( |e| \) is larger than threshold \( \eta \), the training data contain more noise, and so the \( L_1 \)-norm is used. The reason for this is that the \( L_2 \)-norm tends to amplify noise. By placing a second-order function as an upper bound on the Huber loss,\(^{24,25}\), the cost function in Eq. (6) can be rewritten as the weighted minimization problem

\[ \tilde{E}_H(a) = \frac{1}{2} \sum_{i=1}^{n} \tilde{w}_i \tilde{e}_i^2 + C, \]

where \( C \) is a constant independent of \( a \), and weight \( \tilde{w}_i \) is defined as

\[ \tilde{w}_i = \begin{cases} 1.0 & \tilde{e}_i \leq \eta \\ \eta/\tilde{e}_i & \text{otherwise} \end{cases}, \]

where \( \tilde{e}_i = f_a(x_i) - f_i \) is the residual between the model output and the training data when using the current solution \( a \). Equation (9) shows that if a pixel value \( f_i \) is larger than or close to the model output \( f_a(x_i) \), it may be a “light” or shaded background pixel, and is thus, assigned a large weight 1.0; otherwise, it is probably an “inked” pixel and considered an outlier, and is accordingly assigned a smaller weight \( \eta/\tilde{e}_i \). If parameter \( \eta \) is too small, the “light” or shaded background pixels with value less than the model output are assigned small weights almost causing these pixels to be rejected. If \( \eta \) is too large, the weights of the data are virtually the same and thus, the “inked” outliers cannot be excluded. Since all pixels in the image are normalized in the range \([0, 1.0]\), the threshold parameter \( \eta \) can be selected in the range \([0.01, 0.1]\); \( \eta = 0.01 \) was used in all our experiments.
The above represents an iterative weighted least-squares learning algorithm, which is explained below:

**Algorithm 1:** Estimating coefficients of model

**Input:** sample data \{(x_i, f_i) | i = 1, 2, \cdots, n\}

**Output:** coefficient \( \mathbf{a} = (a_1, a_2, \cdots, a_n)^\top \)

1. Initialize coefficient parameter \( \mathbf{a} \), using the general least-squares regression given as Eq. (5).
2. Compute the weight matrix \( \mathbf{W} \) using the current solution \( \mathbf{a}_k \) as \( \mathbf{W} \leftarrow \text{diag}(w_1, w_2, \cdots, w_n) \), where \( w_i \) is defined in Eq. (9) and diag(\( \cdot \)) forms a \( n \times n \) matrix with \( w_1, w_2, \cdots, w_n \) as diagonal elements.
3. Compute the next solution \( \mathbf{a}_{k+1} \) of weighted minimum least squares.
4. Repeat steps 2) and 3) until the solution converges or the number of iterations equals a predetermined cutoff.

The polynomial model\(^{(3)(1)(16)(17)}\) estimates the shaded surface using a threshold to explicitly exclude pixels with low values. The Gaussian kernel model implicitly computes the weights from the residues, where an outlier is assigned a smaller weight, making the algorithm more robust. **Fig. 3(b)** shows the results of several fitting methods. The black curve indicates the intensity data of a row depicted in **Fig. 3(a)** as a white line. The blue curve shows the result of fitting this with a single polynomial; note that this does not fit the raw data well. The green and pink curves, respectively, show the results of fitting with piecewise polynomials and Gaussian kernels, as proposed in this paper. It can be seen that although fitting with piecewise polynomials is better than fitting with a single polynomial, it is worse than fitting with Gaussian kernels. Scrutiny of this graph shows that the fittings act like smoothers. In particular, the linear combination of Gaussian kernels behaves like a Gaussian kernel smoother. However, our model tries to fit the background surface by excluding outliers, that is, the “inked” pixels. Since the minimization criteria are different for each fitting method as mentioned above, comparison with respect to errors is difficult. However, the final binarization results (described in Section 4) prove that the Gaussian kernel model has superior performance to the others.

### 3.2 Constructing a Shading Surface Map

To estimate uneven shading, training data must be input into the Gaussian kernel model. Before inputting the intensity data into the model, the document image is processed using a filter to find the maximum value in a small patch (such as 3 × 3 or 5 × 5 pixels). Note that we assume that the media background is lighter than the text, and the estimated shading surface is considered to be “above” the text. We used a fast maximum filtering algorithm\(^{(26)}\) in the experiments.

The shading surface is estimated by means of a two-pass method using a 1D Gaussian kernel model. In the first pass (horizontal), a linear kernel model is computed for each row. The initial value of the parameter vector \( \mathbf{a} \) for the first row is computed by general least squares without weighting. For every other row, the initial value is simply set to the solution obtained in the previous row. In most cases, only a few iterations are needed for convergence. The second pass uses the estimates from the rows to fit the models for the columns. The process is the same as that for fitting the rows. Only the initial value of the first column is computed using general least squares; the initial value for each subsequent column is the solution to the previous column.

Denoting the surface map by \( S(x, y) \), the map is constructed as follows:

**Algorithm 2:** Surface map construction

**Input:** original document image

**Output:** shading surface map \( S(x, y) \)

1. Filter the given image using a maximum filter.
2. For each row, estimate the linear combination coefficients using **Algorithm 1**.
3. Compute model output for every row using Eq. (3) and its coefficients estimated in step 2.
4. For each column, estimate the linear combination coefficients using **Algorithm 1**. Note that the input data are the computed results from step 3.
5. Compute model output for every column using Eq. (3) and the coefficients estimated in step 4.

This constructs the final map \( S(x, y) \).

Steps 2 and 3 apply to rows and steps 4 and 5 apply to columns. As an example, given the document image illustrated in **Fig. 1(a)**, **Fig. 3(c)** shows the surface map estimated in the horizontal direction. Based on observation, it is known that the uneven shading surface is almost completely estimated after steps 2 and 3. However, there still exist some mis-estimates, which are refined in steps 4 and 5 as shown in **Fig. 3(d)**. It can be seen that this figure approximately estimates the shading surface.

Since the intensity of a pixel often correlates with
that of its neighbors, it is not necessary to use all the pixels to compute combination coefficients. A sparse sampling is sufficient if the sampling rate satisfies the sampling theorem. For example, one can sample pixels in the horizontal and vertical directions. These samples are used to compute parameters in both the horizontal and vertical directions. The estimated curves are used to compute the values of the pixels in the sampled rows and columns. For all pixels other than the sampled ones, the surface values are interpolated using the estimated data.

### 3.3 Removing a Shading Surface

After a map of the uneven shading surface has been constructed, the next step is to remove the shading from the document image. This process normalizes the original image and its background becomes approximately uniform. Using the estimated surface map \( S(x,y) \) as the discrimination criterion, all pixels in document image \( f(x,y) \) are divided into one of two classes: those above the surface \( U(x,y) = \{(x,y) : f(x,y) \geq S(x,y) \} \), and those below the surface \( L(x,y) = \{(x,y) : f(x,y) < S(x,y) \} \), where \( (x,y,f) \in \mathbb{R}^3 \).

Note that the values in the surface map are in the range \([0,1]\); \( \max \) represents the maximum value in the surface map \( S(x,y) \). To remove uneven shading, the document image is normalized by a linear transformation of all pixels \( f(x,y) \) as follows:

\[
N(x,y) = \begin{cases} 
\max + \frac{1-\max}{\max} \cdot f - S & f > S \\
\frac{f \cdot \max}{\max} & f \leq S
\end{cases}
\]

where, for conciseness, \( f \) and \( S \) represent \( f(x,y) \) and \( S(x,y) \), respectively. For example, Fig. 3(e) shows the normalized result in which the uneven shading surface has been removed from the original document image.
shown in Fig. 1(a). It can be seen that most of the shading has successfully been removed, and the background is approximately uniform. Invisible characters in the low contrast areas are now visible and the binarization is easier.

3.4 Binarizing a Document Image

After the uneven shading has been removed, it is appropriate to apply a threshold method. However, since the normalized image may still have a small amount of shading, to improve the accuracy of binarization, we improve Sauvola’s technique by including edge information when thresholding the normalized image. First, the edge of the image is computed using a 3 × 3 window, centered on a pixel. To gather edge information in all directions, the edge is expressed by its gradient magnitude, which is the sum of the norm of the gradients in four directions, namely, vertical, horizontal, and two oblique directions in this window.

As an example, the edge image computed from Fig. 3(e) is shown in Fig. 3(f). The above computation can enhance the contrast of the stroke edges while weakening non-stroke edges. The edge image is then binarized, as shown in Fig. 3(g), and is used to detect the widths of the strokes using a method introduced by Lu et al.15). This width is used as the size of the window in Sauvola’s method, as applied to the normalized image. A binarization method was obtained by extending Sauvola’s method16). As in Sauvola’s method, a local threshold value \( t_p \) for each pixel is computed using the means and standard deviation in the window. The number of pixels \( E_p \) on the edges in the window are counted from the edge image; another threshold \( E_{\text{min}} \), which is the size of the window, is also given. A pixel in the normalized image is set to 0, that is, it is considered to be on a stroke, only if both of the following conditions are satisfied: 1) the pixel value is less than threshold \( t_p \); and 2) in the current window, the number of pixels \( E_p \) on the edges is larger than threshold \( E_{\text{min}} \).

The threshold result is shown in Fig. 3(h). Although the shading has mostly been removed, the binarized result may still contain some noise, especially in concavities and convexities along the text or the drawing strokes. Both small components, such as areas of isolated salt-and-pepper noise, and large components have been cleaned. The concavities and convexities can be filled or deleted using simple pixel operations. This sort of postprocessing can be performed using the various methods proposed in the literature17,27). The binarized result is better than those obtained before removing the uneven shading surface.

4. Experimental Results

We conducted experiments on many document images captured under poor illumination, thereby obtaining very complicated and uneven shading. Images in Figs. 1(a), 4(a), 5(a) and 6(a) were acquired by an IPX-11MST camera, under varying poor illumination patterns cast on a printed document. The images, of size 512 × 512, contain uneven shading, with most of the regions having low contrast. The appearances of these images are very different with a normal one obtained under uniform illumination. The proposed algorithm, written in the C language, was implemented on a Notebook computer with Windows 8.1.

Fig. 1 compares our results with those of other methods. Fig. 1(a) illustrates the original image with severely uneven shading. Fig. 1(b) is the result obtained by applying Otsu’s method directly to the original image. Fig. 1(c) is generated using Sauvola’s method. The binarized result of using a single-polynomial model to estimate the uneven shading is shown in Fig. 1(d). Fig. 1(e) shows the result using the method of Wen et al.11), where some pixels in the background are incorrectly classified as foreground. Fig. 1(f) is the result obtained using the proposed method. It can be seen that the proposed method produces a better result than all the other methods tested.

To test the validity of the proposed method further, we experimented on images containing different shading patterns. Several of these are shown in Figs. 4(a), 5(a), and 6(a) where the complexities of uneven shading are given in ascending order from Figs. 4(a) to 6(a). A large number of regions in these images have very low contrast. The shading in image Fig. 4(a) varies gradually from dark to light, while the shadings in Figs. 5(a) and 6(a) vary unevenly. Otsu’s method failed to binarize them as shown in Figs. 4(b), 5(b), and 6(b). Although Sauvola’s technique worked well on Fig. 4(a) as shown in Fig. 4(c) due to the gradual variation in shading, it could not correctly binarize the image in Fig. 6(a) with complicated uneven shading, as shown in Fig. 6(c), while a moderate result, shown in Fig.5(c) was obtained from Fig. 5(a). The results in Figs. 4(d), 5(d), and 6(d), produced by Wen’s method11), still contain a great deal of noise. Moreover, pixels on the boundary, especially at the bottom of the images, have disappeared. As expected, the single-polynomial model performed well on Fig. 4(a)
as shown in Fig. 4(e), since the shading varies gradually. However, it could not estimate uneven variation in shading surfaces as shown in Figs. 5(a) and 6(a). As illustrated in Figs. 5(e) and 6(e), the binarized results are very poor. The results in Figs. 4(f), 5(f), and 6(f) were obtained using the piecewise-polynomial model. To estimate the shading surface, the block sizes were tuned to obtain the best outputs possible. However, observation shows that some noise remained and the strokes of some characters such as “w” and “m” were squeezed together. Figs. 4(g), 5(g), and 6(g) show the estimated shading surfaces using the Gaussian kernel model. Both the gradually varying shading shown in Fig. 4(a), and the uneven shading shown in Figs. 5(a) and 6(a) were well estimated. Figs. 4(h), 5(h), and 6(h) show the normalized images, with the final binarized results depicted in Figs. 4(i), 5(i), and 6(i). By visual observation, these are cleaner than those obtained by the other methods. Note that the other methods depend heavily on the shading patterns. On the other hand, for various patterns, the proposed method achieved almost the same results. We considered the successes to be due to the fact that most of the uneven shadings were correctly removed in the process, although some weak shadings remained.

Fig. 7 compares the estimated shading surface maps produced from Fig. 6(a) by different models. Fig. 7(a) was created using a single-polynomial model, which failed to estimate the complicated uneven shading variation, while Fig. 7(b) was generated by the piecewise-polynomial model. The map looks blocky due to the fact that the surface was estimated from different blocks.
Fig. 5 Experimental results: (a) original image with uneven shading fluctuating in the horizontal direction; (b), (c), (d), (e), and (f) results using the methods of Otsu, Sauvola, Wen, and Lu, and the piecewise-polynomial method, respectively; (g), (h), and (i) estimated shading surface map, normalized image, and final binarized result using the proposed method.

and then merged together. Fig. 7(c) was obtained by the proposed Gaussian kernel model, and is superior to the results of the other two models. The key to the success of the proposed method is that it applies flexibility in dealing with changes in shading patterns. Inclusion of a stiffness property in the polynomial model, means that it does not fit well to changes in shading.

Two parameters are used: $\sigma$ in Eq. (4) and $\eta$ in Eq. (9). In our experiments, we used different parameter values to estimate the shading surface from Fig. 6(a). The results after shading removal and binarization are shown in Fig. 8. Fig. 8(a) was created with $\sigma = 0.01$ and $\eta = 0.01$, while Fig. 8(b) was generated with $\sigma = 1.00$ and $\eta = 0.1$. Based on observation, some characters are not well segmented; this is due to the fact that the shading surface was not correctly estimated due to the use of improper parameter values given above. $\sigma = 0.1$ and $\eta = 0.01$ were used for all figures except for Fig. 8.

As shown by the above experimental results, the greater part of the shading is removed or weakened by the proposed method, thereby enabling better and easier binarization. Since no ground truth data are available, it is difficult to evaluate our algorithm quantitatively at the pixel level in terms of an F-measure. As an alternative, we evaluated the optical character recognition (OCR) performance (that is, how many words and characters are recognized correctly). The ABBYY FineReader 11\textsuperscript{28)} OCR software was used to recognize the characters from the binarized images. We per-
formed OCR by applying FineReader 11 on the different binarized results obtained from the images in Figs. 1(a), 4(a), 5(a), and 6(a). The recognition results are shown in Table 1.

The original image shown in Fig. 1(a) includes 90 words, 432 letters, while each image in Figs. 4(a), 5(a), and 6(a) contains 93 words and 417 letters. Note that the numbers of words and letters in Fig. 1(a) differ slightly from those in the other images.

Since most of the area was classified as background, Otsu’s method had the worst recognition rates for words and letters. Sauvola’s method had relatively good recognition rates for Figs. 4(a) and 5(a), but it had lower rates for Fig. 1(a) and 6(a). Wen’s method achieved moderate recognition rates for all images, while the single-polynomial model obtained very low recognition rates for both words and letters. The piecewise-polynomial model achieved relatively high recognition rates. However, our proposed method achieved the highest recognition rates for both words and letters. There are eight punctuation marks in each image: 7 ~ 8 were recognized from the results generated by the proposed method; note that these punctuation recognition rates are not given in the table.

The proposed method can remove even more complicated disordered uneven shading patterns such as those illustrated in Fig. 9(a) and 9(c). These images were acquired by casting disordered shading onto printed documents. Fig. 9(a) varies in intensity from dark to bright in a wider range. Fig. 9(c) contains relatively small areas with severely uneven shading while the edges of the shading are much sharper. These complicated un-
even shading surfaces were well estimated as shown in Fig. 9(b) and 9(d) using the proposed method.

Note that the proposed method is not limited to document images with uneven shading. It also performs well on a normal document image that does not contain uneven shading. In this case, the uniform background is estimated as a “shading” surface.

5. Conclusion

This paper presented a novel method for estimating complicated, uneven shading in document images, resulting from poor illumination conditions. The method uses a linear combination of 1D Gaussian kernels to model the shading surface and construct a map thereof. The shading map is then used to normalize the document image; this is equivalent to removing uneven shading, with the resulting background becoming almost uniform. Finally, using information about the stroke edges, an improved version of Sauvola’s method is applied to the normalized image to achieve binarization. Visual and quantitative comparison of our results with those obtained by other techniques, shows that our method has superior performance compared with the other techniques. We are currently considering how to adapt this method to remove even more complex and sharp-edged shading.

Acknowledgment

The authors would like to thank JiangTao Wen from Yanshan University, China, who kindly permitted us to use his original images in Figs. 1(a), 4(a), 5(a), and 6(a). The authors would like to acknowledge the anonymous reviewers for their valuable comments. This work is part of the research supported by the Chunhui project of the Education Ministry of China under Grant Nos. Z2012100 and Z2014020, and the National Nature Science Foundation of China under Grant No. 61462072.

References

6) A. Prati, R. Cucchiara, I. Mikic and M. M. Trivedi: “Analy-

**Xiaohua Zhang** received a B.S. degree in computational mathematics from JiangXi University, China, in 1984, an M.S. degree in computer science and engineering from Jinlin University of Technology, China, in 1999, and a Dr. Eng. degree in information science and engineering from the Tokyo Institute of Technology, Japan, in 2000. After working at NEC Engineering System Inc. from 2000 to 2003, he joined the faculty of the Hiroshima Institute of Technology as an associate professor. His research interests include computer graphics, image processing, computer vision, pattern recognition, and machine learning.

**Yuelan Xin** received a B.S. degree in physics from Shaanxi Normal University, China, in 1997, an M.S. degree in computer science and engineering from Shaanxi Normal University, China, in 2009. She is currently an associate professor at Qinghai Normal University. Her research interests include image processing, pattern recognition.

**Heming Huang** received a B.S. degree in mathematics from Shaanxi Normal University, China, in 1992, an M.S. degree in computer applications from LanZhou University, P.R. China, in 2004, and a Dr. Eng. degree in pattern recognition and artificial intelligence from Southeast University, China, in 2014. He is currently a professor at Qinghai Normal University. His research interests include artificial intelligence.