Philippine Journal of Science
148 (1): 193-211, March 2019
ISSN 0031-7683
Date Received: 26 Sept 2018

Three-step Approach to Edge Detection of Texts

Kristine Rey O. Recio* and **Renier G. Mendoza**

Institute of Mathematics, University of the Philippines Diliman,
Quezon City, Metro Manila 1101 Philippines

We proposed a three-step image segmentation approach to determine the edges of images containing old texts. In general, texts from old books and articles tend to be very noisy. Thus, we first employed a suitable denoising method to obtain a smooth approximation I_s of a given image $\tilde{I}$. Then, the fuzzy edge map $\tilde{E}$ was obtained using the gradient of I_s. This gradient map gave an estimate of the edges of the texts. For the second step, the method of $k-means++$ with two clusters was employed to separate the edges from rest of the image. Because a smooth approximation of the image was used, the edges obtained are "thick." And so, in the last step of the our method, the binary image generated from the previous step was post-processed using a thinning algorithm. We implemented our method to images containing *Baybayin* texts from the National Museum of the Philippines.

Keywords: edge detection, image denoising, method of $k-$means++, thinning algorithm

INTRODUCTION

Image segmentation is a process of dividing the image into regions of meaningful parts such that the image will be simplified and easily analyzable. It is considered as one of the most important processes of image processing. The division of images depends on the image segmentation approach, namely discontinuity detection and similarity detection. The discontinuity approach partitions the image depending on the discontinuities that act as boundaries of the regions. On the other hand, the similarity approach partitions the image depending on the similarity of the data; thus, set of pixels with similar attributes makes a region.

Edge detection is one of the essential implementations of image segmentation under the discontinuity detection approach. It segments the image into regions of discontinuity where there is an abrupt change in values of the gray levels, which mainly gives the contour of the object in an image. Primarily, edge detection is a useful prerequisite for object identification and image segmentation registration.

Image noise is a deterioration during formation, recording, or transmission. This deterioration can be caused by deterministic or random phenomena. Deterministic phenomenon is related to image acquisition (*e.g.*, scanning, natural cause, blur created by motion). Random phenomenon corresponds to noise from signal transmission. Given a real image, it is usually impossible to identify the kind (or level) of noise present (Aubert and Kornprobst 2006).The effectiveness of an edge detection method depends on its capability of detecting meaningful edges, especially for noisy images.

Roberts edge detection (Roberts 1965) performs a simple 2-D spatial gradient measurement on an image. It uses Roberts cross operator, which approximates the gradient of an image through discrete differentiation. This is achieved by computing the sum of the squares of the differences between diagonally adjacent pixels. Sobel edge detection and

*Corresponding author: korecio1@up.edu.ph

Prewitt edge detection are similar in process of determining the edges of an image. The two methods use small integer-valued 3×3 direction masks – one for horizontal and one for vertical. These masks for corresponding edge detection techniques are known as Sobel filter (Sobel and Feldman 1973) and Prewitt filter (Prewitt 1970), respectively. According to Muthukrishnan and Radha (2011), Prewitt detection is slightly simpler to implement computationally than the Sobel detection, but it tends to produce somewhat noisier results. Canny edge detection (Canny 1986) is one of the standard edge detection techniques. This method uses Gaussian filter (Gonzales and Woods 2002) to smoothen the image. Then, it finds the intensity gradient of the image. Edge magnitude and direction are then obtained. It also applies non-maximal suppression to the gradient magnitude. A non-maximal suppression is a post-processing technique that is applied to an image to "thin" the edges. The Laplacian of Gaussian (LoG) (Marr and Hildredth 1980) method uses the sum of the second derivative of the Gaussian function. It smoothes the image and computes the Laplacian that yields a double-edge image. LoG locates the edges by finding zero crossings between the double edges. These methods have their own advantages and disadvantages, thus having their own drawbacks depending on the problem given (Maini and Aggarwal 2009). These methods are few of the well-established edge-detection algorithms and have built-in command in MATLAB. In Batra *et al.* (2016), a comparative analysis of these algorithms have been studied. For other survey papers, one may refer to studies of Bin and Yeganeh (2012), Roushdy (2006), and Basu (2002).

Developing edge detection algorithm is a very active research area. For example, in a recent paper (Azeroual and Afdel 2017), the use of a bilateral filtering method and the Faber Schauder wavelet transform are explored. The work (Mosleh *et al.* 2012) takes advantage of the idea that the bandlet transform effectively represents the geometry of an image to detect edges in images. The paper (Fürtinger 2012) employs a multi-phase segmentation algorithm using a regularization technique based on the Ambrosio-Tortorelli functional. In Li *et al.* (2011), cellular neural network and linear matrix inequality are studied. The use of an evolutionary algorithm, Ant Colony Optimization, is investigated by Lu and Chen (2008). Different methods (Melin *et al.* 2014, Verma *et al.* 2013, Zhang and Liu 2013, Giron *et al.* 2012, Russo and Lazzari 2005) have been studied to detect edges in images.

Figure 1 shows the application of the mentioned methods above to a 256×256 sample image. This image contains old *baybayin* texts. Observe that the image is noisy and portions of the texts are blurred. The above-mentioned methods are implemented using their built-in functions in MATLAB R2017a with their default thresholds and parameters. It can be seen that the Canny (Figure 1c) and LoG (Figure 1f) give very noisy results *i.e.*, the noises are detected as edges. Although the Roberts variant (Figure 1d) gives a less noisy result, it fails to detect some of the edges. The Sobel (Figure 1b) and Prewitt (Figure 1e) variants generated results with less noise and more defined edges among the five methods presented. However, noise is still present and some of the edges are not properly determined. The goal of this paper is to formulate a method that can effectively detect edges of old texts in a noisy image. We apply this method to images containing old texts. These images are degraded naturally and are expected to be noisy. It is impossible to identify the level of noise present in the given image containing old texts because the reference image does not exist. We deal with this problem by using a suitable image smoothing technique.

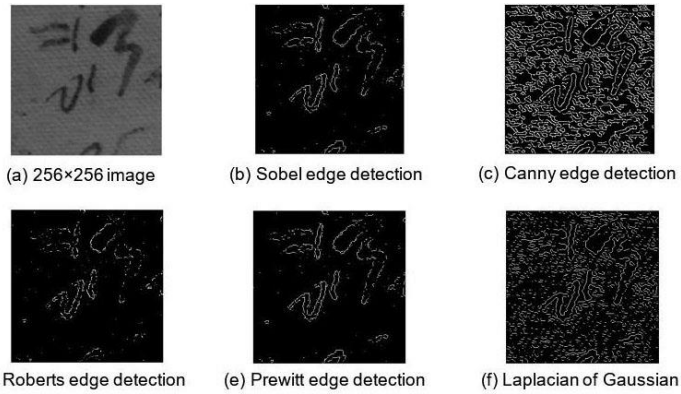


Figure 1. Edge detection methods applied to sample images 1a.

In this study, we explore a three-step approach to edge detection of texts of noisy images. The first step is to remove the noise from the image *i.e.*, to obtain a smooth approximation I_s of the image I. The smoothing step includes minimization of a cost functional. We study the solvability of this minimization problem. We also show how the problem can be solved numerically. Furthermore, the convergence analysis of the numerical solution to the theoretical solution is also presented. In the next section, we show in greater detail the method that we use to obtain I_s. We then obtain the fuzzy edge map $\tilde{E} = |\nabla I_s|$. The norm of the gradient of I_s is used to show the locations where there are abrupt changes of values in an image. The gradient of an image gives us a representation of directional changes of the intensity or color in an image. Indeed, getting the gradient is an essential method in image processing, as well as in edge detection methods. Several edge detection methods are gradient-based. Sobel, Prewitt, and Canny edge detection methods are example of this (Saif *et al.* 2016). From this, we obtain a fuzzy edge map $\tilde{E}$ which illustrates the brightness of some area of the image where the edges are located. Although we can already gather the data about the edges from $\tilde{E}$, it is a fuzzy map and what we wanted is a binary map of the raw image $\tilde{I}$. Thus, we used a clustering technique to obtain the binary image $\boldsymbol{I}$.

Clustering methods are used to group data with similar attributes according to certain distance measures. These methods can be divided into hierarchical, partitioning, density-based, model-based, grid-based, and soft-computing methods (Rokach and Maimon 2005). Intuitively, we cluster the image into two sets according to the level or intensity of each data, whether it belongs to edge set or background set. This process is accomplished by a partitioning method called $k-means++$ *algorithm* (Arthur and Vassilvitskii 2007), a modified version of a well-known algorithm for $k-$means formulation which is also known as Lloyd's algorithm (Lloyd 1982). The $k-$means++ algorithm will be further examined in the next section.

The resulting image $\boldsymbol{I}$ contains thick edges, which are due to the smoothing of the image. Thus, we apply the third step which is thinning. This is done to get a thin-line representation of the edges of the texts and so, to attain the desired result. Thinning algorithms have been widely used for data compression on character patterns. There are various methodologies on implementing the thinning algorithm and a comprehensive survey was published in 1992 to analyze the wide range of such algorithms (Lam *et al.* 1982). The thinning algorithm to be used is further explored in the succeeding section.

In our implementations, we use sample images containing *Baybayin* texts. These images are obtained from the Philippine National Museum. *Baybayin* is an ancient script of the Philippines used in the pre-Spanish colonial period. We chose these sample images because of the importance of the *Baybayin* texts in the Philippine history. Restoration of these texts is timely because the house of the representatives of the Philippines is set to approve the *National Writing System Act*. This bill seeks to declare *Baybayin* as the national writing system of the Philippines (Morallo 2018). For comparison purposes, we also apply our method to an image with Latin texts. We show how our method can effectively solve the edge-detection of texts. We compare our obtained results with the above-mentioned edge-detection methods.

THE THREE-STEP APPROACH

In this section, we discuss the method that we used to detect edges of old texts. The proposed method can be summarized in Algorithm 1 and is shown in Figure 2.

Suppose an image $\tilde{I} \in L^2(\Omega)$ is given. Noise is expected to be present in old texts. For the first step of our method, we first denoise the given image. In the succeeding subsection, we analyze the method that we used to obtain a smooth approximation of $\tilde{I}$. Next, we acquired the binary edge map $\boldsymbol{I}$ using the $k-$means++ method. This method partitions the data into a number of clusters given. In our case, $k-$means++ algorithm clusters the fuzzy image into two sets: the edge set and the background set. However, the obtained edge set contains too much data in a sense that the edges of $\boldsymbol{I}$ are relatively thick. Thus, we find the need to compress the data to obtain thin edges. We refer to this image with thin edges as $\bar{I}$. We used a thinning algorithm to accomplish this.

Figure 2. Flow of the three-step approach where $\tilde{I}$ is the given image, I_s is the smooth approximation, $\tilde{E}$ is the fuzzy edge map, $\boldsymbol{I}$ is the binary edge map, and $\hat{I}$ is the post-process binary edge map.

Algorithm 1 The Three-Step Approach

Input: $\tilde{I}, \alpha$

Output: $\bar{I}$

begin

1. Obtain a smooth approximation I_s of $\tilde{I}$. Solve $\tilde{E} = |\nabla I_s|$.
2. Cluster $\tilde{E}$ into two regions to obtain a binary image $\boldsymbol{I}$ of the edge of texts.
3. Implement thinning to $\boldsymbol{I}$ to acquire $\bar{I}$.

End

Smooth Approximation using Regularization

As seen in Figure 1, the first step of the algorithm is to find a smooth approximation of the given image $\tilde{I}$. In the sample image shown in Figure 1a, noise is visibly present. To obtain a smooth approximation of $\tilde{I}$, the noise should be minimized. The process of removing the noise from a given image is called denoising. One can find many denoising algorithms in literature. For a review, one may refer to Buades *et al.* (2005). Examples of smoothing methods are the Gaussian smoothing model (Lindenbaum *et al.* 1994), anisotropic filtering model (Perona and Malik 1990, Catté *et al.* 1992), neighborhood filtering model (Yaroslavsky 1985, Yaroslavsky and Eden 1996, Smith and Brady 1997), and regularization methods (Rudin *et al.* 1992; Osher *et al.* 2004; Tadmor *et al.* 2004). In this work, we consider a regularization technique.

One of the most famous regularization technique is the one by Rudin, Osher, and Fatemi in 1992 (Rudin *et al.* 1992). They use the functional:

$$J_{ROF}(I) = \frac{1}{2v}\int (I-\tilde{I})^2 dV + \frac{1}{2}\int_\Omega |\nabla I|\, dV = \frac{1}{2v}\left\|I-\tilde{I}\right\|_2^2 + \frac{1}{2}\left\|\nabla I\right\|_1, \tag{1}$$

where $v > 0$ is the weight parameter. Observe that the first term guarantees that the minimizer of J_{ROF} is as close as possible to $\tilde{I}$ while the second term penalizes large gradient magnitude. This way, we can get a minimizer of J_{ROF} that approximates and denoises $\tilde{I}$. Observe that if I is sufficiently regular, the term $\|\nabla I\|_1$ is referred to as the *total variation (TV)* of I. The use of J_{ROF} is called *TV-regularization*. The necessary optimality condition of J_1 is a nonlinear Lagrange equation (Chambolle & Lions 1997). Thus, the minimizer of J_{ROF} is not found in the usual Sobolev space but in $L^1(\Omega)$, in particular, in the space:

$$BV(\Omega) := \left\{ u \in L^1(\Omega) \,\middle|\, \sup\left\{ \int_\Omega u\nabla\cdot\phi\, dV \,\middle|\, \phi \in C_0^1(\Omega): \|\phi\|_{L\hat{}\infty(\Omega)} \le 1 \right\} < \infty \right\}.$$

Solving J_{ROF} in $BV(\Omega)$ requires dealing with a nonlinear partial differential equation, which is computationally expensive.

Another regularization technique that is geared towards image segmentation is the work by Mumford and Shah (1989). The Mumford-Shah functional is given by:

$$J_{MS}(I,\psi) = \frac{\alpha}{2}\int_\Omega \left|I-\tilde{I}\right|^2 dV + \frac{\kappa}{2}\int_{\Omega\setminus\Gamma} |\nabla I|^2 dV + v|\,\Gamma\,|, \tag{2}$$

where α, κ and v are weight parameters. The domain $\Omega = \Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_n \cup \Gamma$ such that Ω_i, $i = 1, \cdots, n$ are disjoint connected open subsets of Ω and $\Gamma = \Omega \cap (\bigcup_{i=1}^n \partial\Omega_i)$. The boundaries $\partial\Omega_i$ are assumed to be piecewise-

smooth. The purpose of the first two terms of J_{MS} is similar to those of J_{ROF}. The second term forces the minimizer to be smooth in Ω except along the discontinuities of I. The third term $|\Gamma|$ denotes the $(n-1)$ dimensional Hausdorff measure. This term penalizes the length of the boundaries of the image's subregions. Applications of J_{MS} can be found in (Rondi and Santosa 2001, Aubert and Kornprobst 2006). Minimization of the Mumford-Shah functional can be tricky because of the geometric nature of the third term. Existence of the minimizer of J_{MS} is not even guaranteed if there are no assumptions on the regularity of Γ (Aubert and Kornprobst 2006). Hence, obtaining the necessary optimality condition of J_{MS} becomes challenging without assumptions on the regularity of Γ. The above functionals (Eq 1) and (Eq 2) can be effective tools in image segmentation. However, numerical implementations of these algorithms are not straightforward and can be computationally costly. If the goal is to detect edges of old texts in a whole book (or even collection of books), this can be computationally expensive. In this work, we consider the optimization problem:

$$J(I) := \frac{1}{2}\int_{\Omega} \left|I - \tilde{I}\right|^2 dV + \frac{\alpha}{2}\int_{\Omega} |\nabla I|^2 \, dV, \tag{3}$$

where $\alpha > 0$ is the weight parameter of the regularization term. This functional was analyzed by Tikhonov and Arsenin (1977). If we wish to minimize Eq. 3, we wish to make the smooth approximation to be as close as possible to $\tilde{I}$. Similar to functionals (Eq 1) and (Eq 2), the first term of (Eq 3) takes care of this. Note that the minimizer of (Eq 3) if $\alpha = 0$ is $\tilde{I}$. Moreover, the second term of the (Eq 3) penalizes L^2 −norm of the gradient of I. Therefore, the minimizer of (Eq 3) not only approximates $\tilde{I}$ but also has minimal gradient norm. The functional is well-studied and has been shown to have a unique minimum in $H^1(\Omega)$ (Aubert and Kornprobst 2006).

We compute the Gateaux derivative (Aubert and Kornprobst 2006) of J in (Eq 3) in an arbitrary direction $v \in C^{\infty}(\overline{\Omega})$. Thus, using integration by parts, we have:

$$\begin{aligned}\frac{\partial J}{\partial I}(I; v) &= \frac{d}{ds}\left(\frac{1}{2}\int_{\Omega}(I + sv - \tilde{I})^2 + \frac{\alpha}{2}\int_{\Omega}|\nabla(I+sv)|^2 dV\right)_{s=0} \quad = \int_{\Omega}(I-\tilde{I})v dV + \alpha\int_{\Omega}\nabla I \cdot \nabla v dV \\ &= \int_{\Omega}(I-\tilde{I})v\, dV + \alpha\int_{\partial\Omega} v\frac{\partial I}{\partial n}dS - \int_{\Omega} v\Delta I\, dV.\end{aligned} \tag{4}$$

Setting the above derivative to $\mathbf{O}$ results to the weak necessary optimality condition in minimizing Eq. 3 (Luenberger 1969). Thus, the minimizer of Eq. 3 satisfies:

$$\int_{\Omega}(I-\tilde{I})\, v\, dV + \alpha\int_{\partial\Omega} v\frac{\partial I}{\partial n}dS - \alpha\int_{\Omega} v\,\Delta I\, dV = 0\,.$$

Suppose that I is sufficiently smooth, we apply the fundamental lemma of calculus of variations (Adams 1975) to the above equation and obtain:

$$\begin{cases} -\alpha\,\Delta I + I = \tilde{I}, & \text{in } \Omega \\ \frac{\partial I}{\partial n} = 0, & \text{in } \partial\Omega \end{cases} \tag{5}$$

Hence, the minimizer of J must satisfy Eq. 5. However, the strong smoothing properties of Laplacian (Aubert and Kornprobst 2006) in Eq. 5 results to a minimizer that is blurry. Because of this smoothing property, the edges in an image appear faded. Using the magnitude of the gradient of the minimizer of J to identify the edges results to "thick" edges. Relying solely on the minimizer of J will not be enough to properly identify the edges of old texts. This is the reason we are adding two more steps after solving for the minimizer of J. We will justify later why we chose the additional two steps and how the new three-step approach can be an effective tool in edge-detection of old texts.

To obtain the minimizer I_s of Eq. 3, we solve Eq. 5 to find the minimum of the functional in Eq. 3. Hence,

$$I_s = \text{argmin}_{I\in L^2(\Omega)}\, J[I]. \tag{6}$$

We do an analysis of the functional J. We show that Eq. 6 has a unique weak solution in a suitable space. Once we prove the existence and uniqueness of the solution, we establish a consistent discretization of the problem in order to solve for the solution numerically. To accomplish this objective, we rely on the Finite Element Method (FEM). For discussion of FEM, one may refer to studies of Solin (2005) and Brenner and Scott (2008). We also show the well-posedness and the unique solvability of the problem defining the discrete solution. Having shown these, we then prove the convergence of the numerical solution to the ideal solution. Furthermore, we show in detail how to obtain the solution numerically. We now present the variational formulation that we'll solve to obtain the minimum of J.

Theorem 2.1 *Let* $\Omega = (0,1)^2$, $\tilde{I} \in L^2(\Omega)$, *and* $\alpha > 0$. *Let* $K = \{v \in H^1(\Omega) : \frac{\partial v}{\partial n} = 0 \text{ in } \partial\Omega\}$, *then the weak formulation of the necessary optimality condition of (Eq 6) is:*

$$\int_\Omega \alpha \nabla I \cdot \nabla v + Iv \; dV = \int_\Omega \tilde{I} v \, dV, \text{ for all } v \in K. \tag{7}$$

This formulation admits a unique solution in K.

Furthermore, I *is the solution of Eq. 7 if and only if* I *is the unique minimum of* J *in Eq. 3.*

Proof: Using Eq. 4 and the boundary condition of Eq. 5, we have:

$$\frac{\partial J}{\partial I}[I; v] = \int_\Omega (I - \tilde{I})v + \alpha \nabla I \cdot \nabla v \, dV.$$

We equate the above equation to 0 and obtain the weak formulation of the necessary optimality condition for the minimization of Eq. 3:

$$A[I, v] := b[v], \; \forall v \in K \tag{8}$$

where $A: H^1(\Omega) \times H^1(\Omega) \to \mathbb{R}$ is a bilinear form given by:

$$A[I, v] := \int_\Omega \alpha \nabla I \cdot \nabla v + Iv \, dV,$$

and $b: H^1(\Omega) \to \mathbb{R}$ is a linear functional given by:

$$b[v] := \int_\Omega \tilde{I} v \, dV.$$

To show that Eq. 7 has a unique solution, we apply the Lax-Milgram Lemma (Cioranescu *et al.* 2012). First, we show that A and b are bounded. Observe that:

$$|b[v]| \leq \int_\Omega |\tilde{I} v| \, dx \leq \|\tilde{I}\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \leq \|\tilde{I}\|_{L^2(\Omega)} \|v\|_{H^1(\Omega)}$$

and:

$$\begin{aligned} |A[I, v]| &\leq \alpha \int_\Omega |\nabla I \cdot \nabla v| \, dV + \int_\Omega |Iv| \, dV \\ &\leq \alpha \|\nabla I\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)} + \|I\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \\ &\leq (\alpha + 1) \|I\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)}. \end{aligned}$$

Thus, A and b are bounded. Now, we are left to show the coercivity of A and we compute:

$$\begin{aligned} A[I,I] &= \int_\Omega \alpha \nabla I \cdot \nabla I + I^2 \, dV \\ &\geq \min\{\alpha, 1\} \left(\|\nabla I\|^2_{L^2(\Omega)} + \|I\|^2_{L^2(\Omega)} \right) \\ &= \min\{\alpha, 1\} \|I\|^2_{H^1(\Omega).} \end{aligned}$$

Thus, the coercivity follows. By the Lax-Milgram Lemma, Eq. 7 has a unique solution in K.

We now show that the solution of the weak formulation in Eq. 7 is the minimizer of J in (Eq 3). Suppose I is the minimizer of J. Let $v \in K$. Then for any $t \in \mathbb{R}$, $I + tv \in K$.

Define $z: \mathbb{R} \to \mathbb{R}$ by:

$$z(t) = J(I + tv) - J(I).$$

Observe that $z(t) \geq 0$ because I is the minimizer of J. Moreover, z attains its minimum when $t = 0$. Using Eq. 3, we can simplify z:

$$\begin{aligned} z(t) &= \frac{1}{2} \int_\Omega \left| I + tv - \tilde{I} \right|^2 dV + \frac{\alpha}{2} \int_\Omega |\nabla (I + tv)|^2 \, dV - \frac{1}{2} \int_\Omega \left| I - \tilde{I} \right|^2 dV - \frac{\alpha}{2} \int_\Omega |\nabla I|^2 \, dV \\ &= t \int_\Omega \alpha \nabla I \cdot \nabla v + \left(I - \tilde{I} \right) v \; dV + \frac{t^2}{2} \int_\Omega \alpha |\nabla v|^2 + |v|^2 \, dV. \end{aligned}$$

Note that:

$$z'(t) = \int_\Omega \alpha \nabla I \cdot \nabla v + \left(I - \tilde{I} \right) v \; dV + t \int_\Omega \alpha |\nabla v|^2 + |v|^2 \, dV.$$

Because 0 is a minimizer of z, we have:

$$0 = z'(0) = \int_\Omega \alpha \nabla I \cdot \nabla v + \left(I - \tilde{I} \right) v \; dV.$$

The above equation is equivalent to Eq. 7.

Conversely, let I be the the solution of Eq. 7. Let $w \in H^1(\Omega)$ and $v := w - I \in H^1(\Omega)$. Then, using Eq. 7:

$$\begin{aligned} J(w) - J(I) &= \frac{1}{2} \int_\Omega \left| w - \tilde{I} \right|^2 dV + \frac{\alpha}{2} \int_\Omega |\nabla (w)|^2 \, dV - \frac{1}{2} \int_\Omega \left| I - \tilde{I} \right|^2 dV - \frac{\alpha}{2} \int_\Omega |\nabla I|^2 \, dV \\ &= \int_\Omega \alpha \nabla I \cdot \nabla v + \left(I - \tilde{I} \right) v \; dV + \frac{1}{2} \int_\Omega \alpha |\nabla v|^2 + |v|^2 \, dV \\ &= \frac{1}{2} \int_\Omega \alpha |\nabla v|^2 + |v|^2 \, dV \\ &\geq 0. \end{aligned}$$

Therefore, $J(w) \geq J(I)$ and because w is arbitrary, I is the minimum of J. QED

We have shown that the given problem has a unique solution I_s. However, we are working in digital grayscale images, specifically grayscale quadratic images of resolution $N \times N$. So, we introduce a consistent discretization of the continuous approach that we presented earlier. We start by introducing the following spaces and notations [see Schumaker (2007) and Brenner and Scott (2008)].

Definition 2.1 *Let $\Omega^{1D} = (0,1)$ and $\Omega_i^{1D} = (x_{i-1}, x_i)$ be a grid on Ω^{1D} with nodes $x = ih$ and stepsize $h = \frac{1}{N}$ for $i = 1, \cdots, N$. Spline bases can be defined as:*

$$S_h^{(m)}(\Omega^{1D}) := \{s \in P^m([x_{i-1}, x_i]) : s \in C^{m-1}(\Omega^{1D}), i = 1,2,\cdots,N\} \text{ for } m = 0,1,\cdots$$

where $P^m([x_{i-1}, x_i])$ is the space of polynomials of degree m in $[x_{i-1}, x_i]$.

Definition 2.2 *Let π_m denote the canonical spline of order m which is given by:*

$$\pi_m(x) = (\pi_{m-1} * \pi_0)(x)$$

where π_0 is the characteristic function of the interval $[0,1]$. We use:

$$\hat{s}_{i+m+1}^{(m)}(x) := \pi\left(\frac{x - x_i}{h}\right), \qquad i = -m, \cdots, N-1$$

as basis function for the $S_h^{(m)}(\Omega^{1D})$.

The defined spline approximation space is in 1D, but we are working on a two-dimensional object. So, for $\Omega = (0,1)^2$, we see $S_h^{(m)}(\Omega)$ as tensor products of the spline bases. The space $S_h^{(m)}(\Omega)$ is dense in $H^m(\Omega)$. Furthermore, $\left\{s_i^{(m)}\right\}_{i=1}^{(N+m)^2}$ is a basis of $S_h^{(m)}(\Omega)$ (Schumaker 2007).

Let $\tilde{I}_h$ be the spline representation of $\tilde{I}$. Since we work with digital grayscale images of size $N \times N$, the pixelwise averaged approximation $\tilde{I}_h$ is in $S_h^{(0)}$, where $S_h^{(0)}$ is the space containing cellwise constant functions. Therefore, we can denote the intensity values of $\tilde{I}_h$ in lexicographic ordering by $\left\{\tilde{I}_{h,i}\right\}_{i=1}^{N^2}$. Then, we have:

$$\tilde{I}_h = \sum_{i=1}^{N^2} \tilde{I}_{h,i} s_i^{(0)} \tag{9}$$

where $\left\{s_i^{(0)}\right\}_{i=1}^{N^2}$ is a basis of $S_h^{(0)}$. Then, we consider the following theorem to establish a finite element discretization of the weak formulation (Eq. 8) and its unique solvability. The succeeding theorem follows from the previous theorem.

Theorem 2.2 *Let $\tilde{I}_h$ be the pixelwise data approximation of $\tilde{I}$ and let:*

$$A_h[I_{s,h}, v_h] = b_h[v], \forall v_h \in S_h^{(1)}(\Omega), \tag{10}$$

where $A: H^1(\Omega) \times H^1(\Omega) \to \mathbb{R}$ is a bilinear form given by:

$$A_h[u, v] := \int_\Omega \alpha \nabla u \cdot \nabla v + uv \, dV, \tag{11}$$

and $b: H^1(\Omega) \to \mathbb{R}$ is a linear functional given by:

$$b_h[v] := \int_\Omega \tilde{I}_h v \, dV, \tag{12}$$

be the finite element discretization of Eq. 8. Then, Eq. 10 has a unique solution $\tilde{I}_{s,h} \in S_h^{(1)}(\Omega)$.
Proof: The proof is similar to the proof of the first part of Theorem 2.1. QED

After proving the existence and uniqueness of solutions $I_s \in H^1(\Omega)$ and $I_{s,h} \in S_h^{(1)}(\Omega)$, we now show in the next theorem the convergence of $I_{s,h}$ to I_s as $h \to 0$ in $H^1(\Omega)$.

Theorem 2.3 *Let $I_{s,h}$ be the unique solution of Eq. 10 and I_s be uniquely defined by the weak formulation Eq. 8. Furthermore, suppose that:*

$$\lim_{h \to 0} \left\| \tilde{I}_h - \tilde{I} \right\|_{L^2(\Omega)} = 0. \tag{13}$$

Then,

$$\lim_{h \to 0} \left\| I_{s,h} - I_s \right\|_{H^1(\Omega)} = 0. \tag{14}$$

Proof: From Eq. 8 and Eq. 10,

$$A(I_s - I_{s,h}, v) = b(v) - b_h(v) \tag{15}$$

Thus,

$$\begin{aligned} |A(I_s - I_{s,h}, v)| &= \left| \int_\Omega v(\tilde{I}_h - \tilde{I})\ dV \right| \\ &\leq \|\tilde{I}_h - \tilde{I}\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \text{ (by Cauchy-Schwarz Inequality)} \\ &\leq \|\tilde{I}_h - \tilde{I}\|_{L^2(\Omega)} \|v\|_{H^1(\Omega)}. \text{ (by the definition of } \|\cdot\|_{H^1(\Omega)}) \end{aligned}$$

Furthermore, since A is coercive, $\exists c > 0$ such that:

$$A(I_s - I_{s,h}, I_s - I_{s,h}) \geq c\|I_s - I_{s,h}\|^2_{H^1(\Omega)}$$

Then combining this with Eq. 15, we get:

$$c\|I_s - I_{s,h}\|^2_{H^1(\Omega)} \leq \|\tilde{I}_h - \tilde{I}\|_{L^2(\Omega)} \|I_s - I_{s,h}\|_{H^1(\Omega)}$$

Therefore,

$$\|I_s - I_{s,h}\|_{H^1(\Omega)} \leq \frac{1}{c}\|\tilde{I}_h - \tilde{I}\|_{L^2(\Omega)}.$$

Hence, given the assumption (Eq 13) then (Eq 14) is established. QED

Remark. *Observe that to show the convergence (Eq. 14), the assumption (Eq. 13) was necessary. This assumption means that as $h \to 0$, the discretized data $\tilde{I}_h$ approximates the given raw data $\tilde{I}$ more accurately.*

Now that we've shown the solvability and convergence of a finite element approach for our problem in Eq. 10, we can now compute explicitly the linear system that will give $I_{s,h}$. Let $\left\{s_i^{(1)}\right\}_{i=1}^{(N+1)^2}$ to be the basis of $S_h^{(1)}(\Omega)$. We can observe that solving for $\tilde{I}_{s,h}$ means computing for $I_{s,h,i}$ such that:

$$I_{s,h} = \sum_{i=1}^{(N+1)^2} I_{s,h,i} s_i^{(1)}$$

Let $I_s \in \mathbb{R}^{(N+1)^2}$ denote the vector of coefficient values of $I_{s,h}$, *i.e.*, $I_s = \left\{I_{s,h,i}\right\}_{i=1}^{(N+1)^2}$. Then from the form of A_h given in (Eq 11), we define:

$$a[u_h, v_h] := \int_\Omega \nabla u_h \cdot \nabla v_h\, dx.$$

Now, we introduce the notation for the bending matrix $\boldsymbol{A}$ associated to A_h as:

$$\boldsymbol{A} := \left\{a\left[s_i^{(1)}, s_j^{(1)}\right]\right\}_{i,j=1}^{(N+1)^2} \in \mathbb{R}^{(N+1)^2 \times (N+1)^2}.$$

Similarly, the Gram matrix $\boldsymbol{G}$ for A_h is denoted by:

$$\boldsymbol{G} := \left\{c\left[s_i^{(1)}, s_j^{(1)}\right]\right\}_{i,j=1}^{(N+1)^2} \in \mathbb{R}^{(N+1)^2 \times (N+1)^2}$$

where

$$c[u_h, v_h] := \int_\Omega u_h v_h\, dx \quad \text{for } u_h, v_h \in S_h^{(1)}(\Omega).$$

Furthermore, let $\tilde{I} \in \mathbb{R}^{N^2}$ denote the vector intensity values of $\tilde{I}_h$ as given by Eq. 9, *i.e.*, $\tilde{I} := \{\tilde{I}_{h,i}\}_{i=1}^{N^2}$ and for b_h in Eq. 12, set $v = s_k^{(1)}$. Then by using the spline representations of $\tilde{I}_h$:

$$b_h[s_k^{(1)}] = \int_\Omega \sum_{i=1}^{N^2} \tilde{I}_{h,i} s_i^{(0)} s_k^{(1)}\, dx = \sum_{i=1}^{N^2} \int_\Omega \tilde{I}_{h,i} s_i^{(0)} s_k^{(1)}\, dx\,. \tag{16}$$

Observe that the right-hand side of Eq. 10 contains the inner product of $\tilde{I}_h \in S_h^{(0)}(\Omega)$ and an element of $S_h^{(1)}(\Omega)$. To make the system consistent, we define the projection matrix $\boldsymbol{P_1}$ that maps linear splines to constants as:

$$\boldsymbol{P_1} := \left\{ \int_\Omega s_i^{(0)} s_j^{(1)} \right\}_{i=1,\cdots,N^2, j=1,\cdots,(N+1)^2} \in \mathbb{R}^{N^2 \times (N+1)^2}$$

Then Eq. 16 is the kth component of the vector $\boldsymbol{P_1^\top}(\tilde{I})$. Thus, we attain the following expression for Eq. 10:

$$(\alpha \boldsymbol{A} + \boldsymbol{G}) I_s = \boldsymbol{P_1^\top}(\tilde{I}). \tag{17}$$

From this, we can now solve for I_s by constructing the matrices $\boldsymbol{A}$, $\boldsymbol{G}$, and $\boldsymbol{P_1}$.

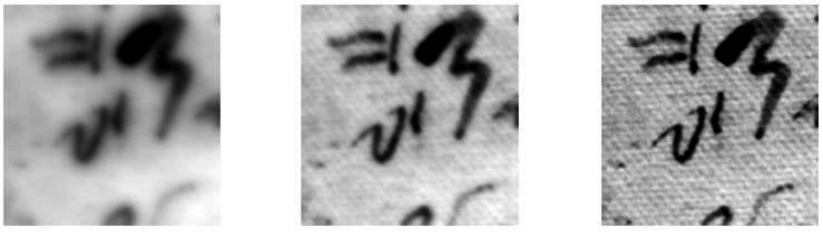
So far, we have established how the numerical solution of the minimization problem is done. Now we present some properties of the parameter α.

Theorem 2.4 *Let $\tilde{I} \in L^2(\Omega)$ and $\alpha > 0$. Let I_α be the solution of Eq. 7. Define $\tilde{I}_{ave}$ to be the average of the data $\tilde{I}$ i.e., $\tilde{I}_{ave} = \frac{1}{|\Omega|}\int_\Omega \tilde{I}\, dV$. Then we have:*

$$1. \int_\Omega |I(x,\alpha)|^2\, dV \le \int_\Omega \left|\tilde{I}\right|^2 dV$$
$$2. \int_\Omega I_\alpha\, dV = \int_\Omega \tilde{I}\, dV$$
$$3. \lim_{\alpha \to +\infty} \int_\Omega \left|I_\alpha - \tilde{I}_{ave}\right| dV = 0.$$

A generalized theorem with proof can be found in Chapter 3.2 of (Aubert and Kornprobst 2006). The first result of the theorem shows how the L^2 −norm of I_α is bounded for any arbitrary $\alpha > 0$. Observe how the bound is independent of the choice of α. The second result implies that the mean of I_α is constant, independent of the choice of α. Finally, the third result tells us that I_α converges in the $L^1(\Omega)$ −strong topology to the average of the initial data. This property also tells us that α can be regarded as the scale parameter. As α becomes large, the obtained image I_α becomes more simplified (smoothed) version of $\tilde{I}$. These properties can help us choose the appropriate smoothing parameter α.

Figure 3 shows the smooth approximation of the given image in Figure 1a with different values of α. Notice that the approximation gets smoother as the value of α becomes larger. Moreover, the image becomes less noisy but more blurred. However, the edges become less defined. On the other hand, taking smaller α makes the edges of the text more defined. However, making α too small will retain some of the noise. Hence, the parameter α should be chosen appropriately. The choice of α depends on how noisy the given image is.



(a) $\alpha = 2 \times 10^{-2}$ (b) $\alpha = 2 \times 10^{-3}$ (c) $\alpha = 2 \times 10^{-4}$

Figure 3. Smooth approximation of the given images in Figure 1a.

Given I_s, we obtain the fuzzy edge map $\tilde{E} = |\nabla I_s|$. Getting the gradient of the image is a fundamental step in an edge detection problem since it shows the locations in an image where there are abrupt changes in brightness of an image, as we can observe in Figure 4. As we can see, for larger α, the edges tend to be less defined and more fuzzy because I_s is "too smooth". For smaller α, the noise in the image is visible.

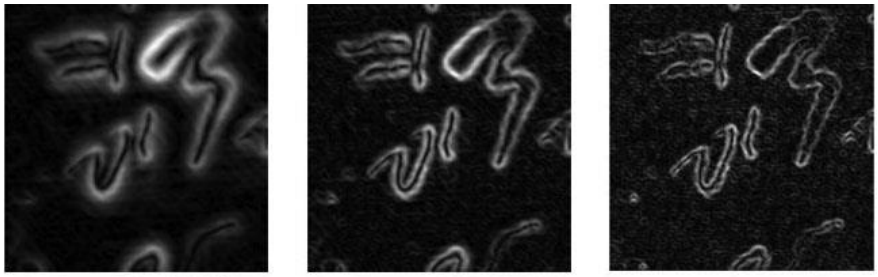


Figure 4. Corresponding fuzzy edge maps of obtained smooth approximation in Figure 3.

Binarization using K-means++

Because $\tilde{E}$ is the norm of the gradient of I_s, it has values ranging from 0 to 1. To separate the edges of the text from the image, we have to make $\tilde{E}$ binary. After obtaining the fuzzy edge map $\tilde{E}$, we work on getting the binary edge map $\boldsymbol{I}$ using $k-$means++ (Arthur and Vassilvitskii 2007) , a method used for $k-$means clustering. The method of $k-$means is a simple and fast clustering algorithm. However, the approximation can be arbitrarily bad with respect to the potential function compared to the optimal clustering since the initial arbitrary k centers are chosen uniformly at random from the data points. In 2007, Arthur and Vassilvitskii presented the $k-$means++ method, which suggested a specific way of choosing centers for the $k-$means algorithm. They have shown that by doing the modification of establishing a simple, randomized seeding technique in $k-$means, both its speed and accuracy can be improved.

Let $D(x)$ denote the shortest distance from a data point to the closest center we have chosen. The binarization technique is summarized in the following algorithm:

Algorithm 2 $k-$means++ Algorithm

Input: $\tilde{E}, k$

Output: $C_i's, \quad i = 1, \cdots, k$

Begin

1. Choose one center c_i uniformly at random from $\tilde{E}$.
2. Take a new center c_i, choosing $x \in \tilde{E}$ with probability
$$P = \frac{D(x)^2}{\sum_{x \in \tilde{E}} D(x)^2}.$$
3. Go to 2 until $\boldsymbol{k}$ centers have been taken altogether and denote $C = \{c_1, \ldots, c_k\}$.
4. Set the cluster C_i to be the set of points in $\tilde{E}$ that are closer to c_i than they are to c_j for all $j \neq i$, for each $i \in \{1, \cdots, k\}$.
5. Set c_i to be the center of mass of all points in C_i:
$$c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x, \text{ for each } i \in \{1, \cdots, k\}.$$
6. Do 4 and 5 until C no longer changes.

End

Note that the first three steps are the suggested way of choosing centers from the data points, while the rest are the steps in the standard $k-$means algorithm that comes after the arbitrary choosing of centers. We apply this algorithm

to the fuzzy edge map $\tilde{E}$. Letting $k = 2$, this method heuristically clusters the data points into two – the edge and the non-edge clusters. We implement this approach to the fuzzy edge maps in Figure 4. This is shown in Figure 5. Observe that the images are now binary. We can notice in Figure 5b that the edges are more defined. Although the binarization in Figure 5b successfully detects the edges of the text, the fuzziness of $\tilde{E}$ makes the edges "thick." For this reason, we add the following step.

(a) $\alpha = 2 \times 10^{-2}$ (b) $\alpha = 2 \times 10^{-3}$ (c) $\alpha = 2 \times 10^{-4}$

Figure 5. k – means++ applied to fuzzy edge maps of Figure 4.

Application of Thinning Algorithm

We denote the obtained binary image as $\boldsymbol{I}$. In order to remove most of the noise from the image, the value of the parameter α cannot be too small. However, this results to a smoother and more blurred fuzzy edge map. Hence, the computed binary image I is expected to be "thick." This is apparent as shown in Figure 5. Hence, we ought to apply a post-process technique to make the edges sufficiently thin. Thinning algorithms have been used to a wide variety of shapes to compress the data into their thin representations. These algorithms were found to be useful in different components of biomedical field and other sectors *e.g.*, in the works of O'Callaghan and Loveday (1973), Preston *et al.* (1979), and Dill *et al.* (1987). This is because they offer different purposes such as reducing the amount of data to be processed and giving a line-like patterns of shapes that can easily be analyzed. Such goals were attained by continuously removing pixels until the only remaining pixels are the skeleton of the figure. Thus, the effectiveness of the thinning algorithms depends on the conditions on whether the pixel is going to be deleted or not. In our method, the goal is accomplished by a parallel thinning algorithm introduced by Guo and Hall (1989). One can refer to Algorithm 3 with the help of the following definitions and the illustration in Figure 6.

x_1	x_2	x_3
x_8	p	x_4
x_7	x_6	x_5

Figure 6. 3×3 –neighborhood of pixel p.

Definition 2.4 *Given the x_i's neighborhood of a pixel p as illustrated in Figure 6, define the connection number of p, $C(p)$, as:*

$$C(p) = \sum_{i=1}^{4} {\sim}x_{2i} \wedge (x_{2i+1} \vee x_{2i+2}),$$

where $x_9 = x_1$ and $x_{10} = x_2$.

Definition 2.5 *Let $N_1(p) = \sum_{i=1}^{4} x_{2i-1} \vee x_{2i}$ and $N_2(p) = \sum_{i=1}^{4} x_{2i} \vee x_{2i+1}$ where $x_9 = x_1$. Then:*

$$N(p) = \min\{N_1(p), N_2(p)\}.$$

The first condition, $C(p) = 1$, is a prerequisite to preserve connectivity when p is deleted. This also avoids deletion of pixels inside the medial curves. On the other hand, $N_1(p)$ and $N_2(p)$ count the number of two adjacent pixels that

contain one or two 1s. Thus, the second condition – $2 \leq N(p) \leq 3$ – permits the endpoints to be preserved but also helps in deleting redundant pixels in the middle of the curves. The last condition can interchange in every iteration *i.e.*, the first one is implemented in odd iterations then the second one in even iterations. These two conditions tend to identify pixels at north and east part boundary and south and west part boundary of the objects, respectively. For every iteration, every pixel $p = 1$ in $\boldsymbol{I}$ that satisfies the said conditions is deleted *i.e.*, 1 is changed to 0. Lastly, the thinning stops when there is no longer a pixel that satisfies the conditions. Some examples of 3×3-neighborhood that satisfy the conditions are illustrated in Figure 7.

Algorithm 3 Thinning Algorithm (Guo and Hall)

Input: $\boldsymbol{I}$

Output: $\bar{I}$

begin

1. **for** every pixel $p = 1,\ p \in \boldsymbol{I}$
2. **if** $C(p) = 1$ and $2 \leq N(p) \leq 3$ and, $[\big((x_2 \vee x_3 \vee\sim x_5) \wedge x_4\big) = 0$ or $\big((x_6 \vee x_7 \vee\sim x_1) \wedge x_8\big) = 0]$
3. Delete p. (Change 1 to 0)
4. Iterate until no further deletion occurs.

End

1	1	1
0	p	0
0	0	0

1	0	0
1	p	0
0	1	1

0	0	0
1	p	1
0	1	1

Figure 7. Examples of 3×3 −neighborhood where p is deleted.

We implement the thinning algorithm to $\boldsymbol{I}$. We show the result in Figure 8. For the given binary image $\boldsymbol{I}$, it took 9 iterations – including last two conditions simultaneously in each iteration – until the algorithm terminated *i.e.*, no further deletion was made. The thinning step can be omitted if the user is already satisfied with $\boldsymbol{I}$. However, if one intends to detect the edges of texts in a book, thinning algorithm significantly reduces the size of data. Note that because of the thinning algorithm, the matrix representation of the obtained image becomes binary and sparse. This reduces the size of the data and gives users the advantage of storing more images. This comes in handy if one intends to apply the algorithm to store old texts in a book.

Figure 8. Thinning algorithm applied to image (5b) given a number of iterations.

RESULTS AND DISCUSSION

We present the results of our method applied to different sample images. For comparison, we also implement other edge detection methods presented in the introduction. We implement the corresponding MATLAB R2017a built-in functions for these algorithms using their default parameter values.

In Figure 9, we illustrate the process of our method with $\alpha = 2 \times 10^{-3}$ to a 256×256 sample image. These results are gathered and compiled from the previous discussion of the method. We can now compare our result $\bar{I}$ to the images in Figure 1. Comparisons are done purely by observation. Figure 9 shows the application of the three-step approach to the given image in Figure 9a. The first step is shown in Figure 9b and 9c. Observe that part Figure 9b has less noise. Figure 9c gives the fuzzy edge map of $\tilde{I}$. The second step is illustrated in Figure 9d. One can see that the image is already binary. However, because the fuzzy edge map is blurry, the resulting binary image appears thick. This is expected since we are minimizing a functional that penalizes sharp gradients. The third step of the algorithm is shown in Figure 9e. Notice how the thickness of the edges is significantly reduced. The noise in Figure 9e is much less compared to the original images.

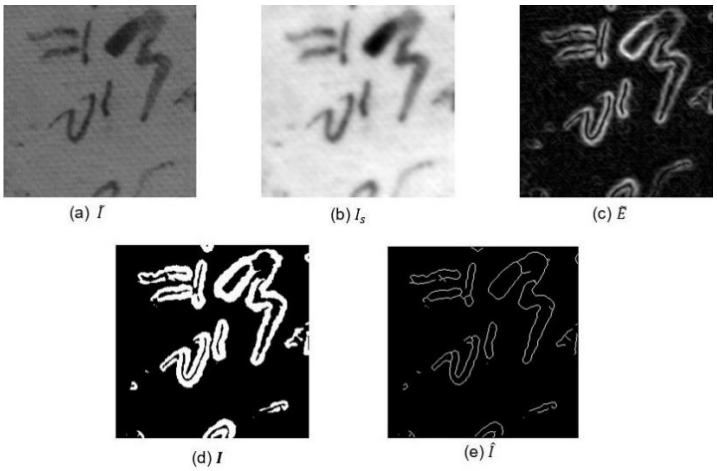


Figure 9. Application of the three-step approach to a 256×256 image.

We compare the result shown in Figure 9e with the obtained results using other algorithms in Figure 1. Figures 1c and 1f show high level of noise. Although results obtained in Figures 1b, 1d, and 1e are better, noise is still visible and some of the edges are not properly identified. One can see how our method (Figure 9e) can effectively detect the edges of the texts and significantly reduce the noise level of the given image.

It is shown in Figures 10 and 11 the application of the methods to other sample images. The texts in sample image 10a are more defined compared to previous example, thus we expect the methods to work well. However, the Sobel (Figure 10e), Roberts (Figure 10c), and Prewitt (Figure 10f) variants did not recognize some of the edges. On the other hand, the Canny (Figure 10d) and LoG (Figure 10g) variants presented defined edges of the texts but they also detected noises of the image. Our method was able to reduce the noise level of these sample images while effectively identifying most of the edges of the given texts. In another example, the sample image Figure 11a does not have defined edges as Figure 10a. Some of the edges are faded, which makes edge detection method to fail on identifying them as edges. We can see that the Roberts (Figure 11e) variant failed to detect several edges compared to those of Sobel (Figure 11c), Prewitt (Figure 11f), and our method (Figure 11b). Meanwhile, the Canny (Figure 11d) and Laplacian (Figure 11g) variants produced results with high level of noise.

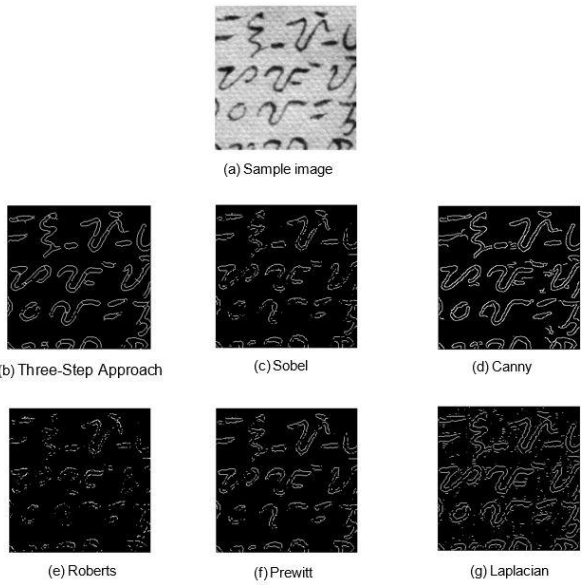


Figure 10. The three-step approach, with $\alpha = 4 \times 10^{-4}$, and other edge detection methods applied to sample image of size 256×256.

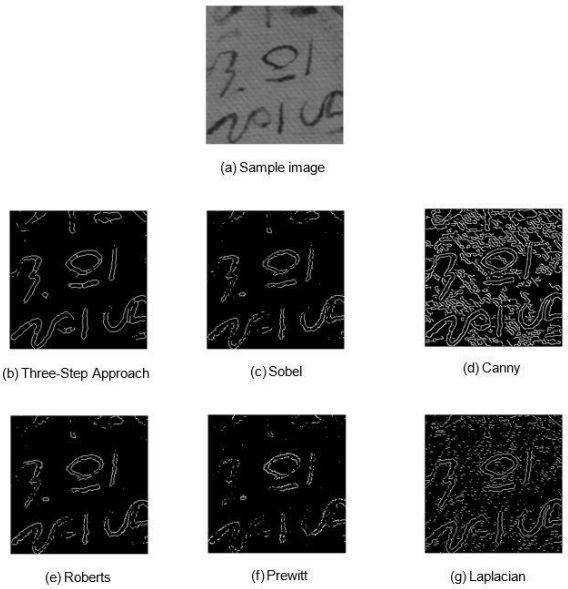


Figure 11. The three-step approach, with $\alpha = 9 \times 10^{-4}$, and other edge detection methods applied to sample image of size 256×256.

The parameter α varies in every given sample image. Recall that from the first step of our method, larger value of α makes I_s smoother. This helps to reduce the noise level of the images. However, this also tends to make edges blurry since it smoothens the whole image. We denote α_1, α_2 and α_3 to be the smoothing parameter of the first (Figure 1a), second (Figure 10a), and third (Figure 11a) sample images, respectively. We observe that α_2 has the smallest value among the examples. This is because the second sample image has defined edges compared to others. Hence, it does not need much smoothing than the others since the method can already distinguished the edges from noises and background. Meanwhile, α_1 and α_3 are larger than α_2 since the two sample images (Figure 1a and Figure 11a) contain edges that are difficult to distinguish from some noises since some of their edges are slightly defined. So, they need more smoothing than the second image to lessen the noises around the images and avoid classifying noise as edges. We would like to emphasize how the selection of α is crucial. It depends on how noisy the given image is. This is the only parameter of our method. The success of this three-step approach hinges on the appropriate choice of α.

Although we applied our method to *Baybayin* texts, it can also be used to detect edges of texts of any font. Figure 12 shows the results of edge detection methods applied to a sample image from the paper of (Mosleh *et al.* 2012, page 5, Figure 3a). The Canny (Figure 12d) variant yielded defined edges of the texts but noise is still prevalent. The Sobel (Figure 12c) and Prewitt (Figure 12f) variants produced almost similar results with less noise and less significant edges. On the other hand, the Roberts (Figure 12e) and Laplacian (Figure 12g) variants failed to detect a large amount of edges of the texts. Moreover, the Laplacian (Figure 12g) variant produced noisier result than that of Roberts (Figure12e). Meanwhile, the Three-step Approach (Figure 12b) generated image with the least noise among others and detected most of the edges of the texts. However, notice that the corners of the texts are not as sharp of the corners in the sample image. This is because of the smoothing used to deal with the noise. Also, there are branch-like lines connected to the edges of some texts that is caused by the thinning algorithm. Nonetheless, the three-step method (Figure 12a) presented easily recognizable texts by distinguishing the edges of the texts from the noise.

Figure 12. The three-step approach, with $\alpha = 5 \times 10^{-3}$, and other edge detection methods applied to sample image (a) from (Mosleh *et al.* 2012).

CONCLUSION

We presented a method in detecting edges of texts for noisy images. The method is applied to old images containing *Baybayin* texts. The first step is to generate the smooth approximation I_s of the raw image $\tilde{I}$ by minimizing a regularized error functional. We computed the numerical solution of the minimization problem using the Finite Element Method. The existence and uniqueness of solution, together with the analysis of convergence, are presented. Furthermore, the explicit linear system formulation is shown. The smoothness of the resulting image is dependent on the value of α. Larger value of α produces smoother image. The parameter α depends on the given image and can be adjusted depending on the noise present on the data. Given I_s, we obtain the fuzzy-edge map $\tilde{E}$ from the gradient of I_s. Next, the image is heuristically clustered into two using $k-$means++ algorithm. This was done to separate the edges of the texts. Lastly, we apply a post-processing method called thinning algorithm to acquire the final image $\bar{I}$. This reduces the data size of the image and make the edges more defined. The three-step approach is found to be effective on detecting the edges and reducing the noise level of the image with the appropriate choice of the parameter α. As shown in results, the method fared well with other existing methods. One can look into the use of the peak signal-to-noise ratio (PSNR) (Huynh-Thu *et al.* 2008) in choosing α. PSNR is widely-used in determining the efficiency of an image restoration algorithm. However, this requires the original image as a reference. This means that PSNR can't be applied to an image containing old texts. There are recent papers that estimate PSNR without the reference image (Moreno *et al.* 2013, Ryu *et al.* 2012). There might be a modification to no-reference PSNR before implementing it to our method, since the noise present in an image containing old texts is not signal-based. This can be a good subject for further research.

ACKNOWLEDGMENTS

The authors acknowledge the support of the Department of Science and Technology – Science Education Institute (DOST-SEI) through the Accelerated Science and Technology Human Resources Development Program (ASTHRDP) Scholarship.

REFERENCES

ADAMS R. 1975. Sobolev spaces. New York: Academic Press.

ARTHUR D, VASSILVITSKII S. 2007. K-means++: The advantages of careful seeding. In: SODA '07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete algorithms; 07–09 Jan 2007; New Orleans, LA. 1: 1027–35.

AUBERT G, KORNPROBST P. 2006. Mathematical problems in image processing, vol. 147 of Applied Mathematical Sciences, 2nd ed. New York: Springer.

AZEROUAL A, AFDEL K. 2017. Fast Image Edge Detection based on Faber Schauder Wavelet & Otsu Threshold. Heliyon 3(12): e00485

BATRA B, SINGH S, SHARMA J, ARORA S. 2016. Computational analysis of edge detection operators. International Journal of Applied Research 2(11): 257–262.

BASU M. 2002. Gaussian-Based Edge Detection Methods – A Survey. IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews 32(3): 252–260.

BIN L, YEGANEH MS. 2012. Comparison for Image Edge Detection Algorithms, IOSR Journal of Computer Engineering 2(6): 1–4.

BRENNER S, SCOTT R. 2008. The mathematical theory of finite element methods. Springer.

BUADES A, COLL B, MOREL JM. 2005. A review of image denoising algorithms with a new one. Multiscale Modeling & Simulation 4(2): 490–530.

CANNY JF. 1986. A computational approach to edge detection. IEEE Transaction on Pattern Analysis and Machine Intelligence 8: 679–714.

CATTÉ F, LIONS P, MOREL J, COLL T. 1992. Image selective smoothing and edge detection by nonlinear diffusion. SIAM Journal on Numerical Analysis 29(1): 182–193.

CHAMBOLLE A, LIONS PL. 1997. Image recovery via total variation minimization and related problems. Numerische Mathematik 76(2): 167–188.

CIORANESCU D, DONATO P, ROQUE M. 2012. Introduction to classical and variational partial differential equations. Quezon City (Philippines): The University of the Philippines Press.

DILL AR, LEVINE MD, NOBLE PB. 1987. Multiple resolution skeletons. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI 9(4): 495–504.

FÜRTINGER S. 2012 An Approach to Computing Binary Edge Maps for the Purpose of Registering Intensity Modulated Images [Ph.D. Thesis]. Graz (Austria): Karl-Franzens University of Graz.

GIRON E, FRERY A, CRIBARI-NETO F. 2012. Nonparametric Edge Detection n Speckled Imagery. Mathematics and Computers in Simulation 82: 2182–98.

GONZALEZ R, WOODS R. 2002. Digital Image Processing, 2nd ed. Pearson Education.

GUO Z, HALL RW. 1989. Parallel thinning with two-subiteration algorithms. Communications of the ACM 32(3): 359–373.

HUYNH-THU Q, GHANBAR M. 2008. Scope of Validity of PSNR in Image/Video Quality Assessment. Electronic Letters 44(13).

RYU J-W, LEE S-O, SIM D-G, HAN J-K. 2012. No-reference Peak Signal to Noise Ratio Estimation Based on Generalized Gaussian Modelling of Transform Coefficient Distributions. Optical Engineering 51(2): 027401.

LAM L, LEE SW, SUEN CY. 1992. Thinning methodologies – A comprehensive survey. IEEE Trans. Pattern Anal. Mach. Intell. 14(9): 869–885.

LI H, LIAO X, LI C, HUANG H, LI C. 2011. Edge Detection of Noisy Image based on a Cellular Neural Networks. Commun Nonlinear Sci Numer Simulat 16: 3746–59.

LINDENBAUM M, FISCHER M, BRUCKSTEIN A. 1994. On gabor's contribution to image enhancement. Pattern Recognition 27(1): 1–8.

LLOYD S. 1982. Least squares quantization in pcm. IEEE Transactions on Information Theory 28(2): 129–137.

LU D-S, CHEN C-C. 2008. Edge-Detection Improvement by Ant Colony Optimization, Pattern Recognition Letters 29: 416–425.

LUENBERGER D. 1969. Optimization by vector space methods. John Wiley & Sons.

MAINI R, AGGARWAL H. 2009. Study and comparison of various image edge detection techniques. International Journal of Image Processing (IJIP) 3(1): 1–11.

MARR D, HILDRETH E. 1980. Theory of edge detection. Proceedings of the Royal Society B: Biological Sciences 207:187–217.

MORENO J, JAIME B, SAUCEDO S. 2013. Towards No-Reference of Peak Signal to Noise Ratio based on Chromatic Induction Model. International Journal of Advanced Computer Science and Applications 4(1): 123–130.

MELIN P, GONZALES C, CASTRO J, MENDOZA O, CASTILLO O. 2014. Edge-Detection Method for Image Processing Based on Generalized Type-2 Fuzzy Logic. IEEE Transactions on Fuzzy System 22(6): 1515–25.

MORALLO A. 2018 Apr 23. House panel approves use of baybayin as country's national writing system. The Philippines Star. Retrieved from https://www.philstar.com/headlines/2018/04/23/1808717/house-panel-approves-use-baybayin-countrys-national-writing-system

MOSLEH A, BOUGUILA N, HAMZA AB. 2012. Image Text Detection using a Bandlet-Based Edge Detector and Stroke Width Transform. Proceedings of the British Machine Vision Conference: 63.1–63.12.

MUMFORD D, SHAH J. 1989. Optimal approximations by piecewise smooth functions and associated variational problems. Communications on Pure and Applied Mathematics 42(5): 577–685.

MUTHUKRISHNAN R, RADHA M. 2011. Edge detection techniques for image segmentation. International Journal of Computer Science & Information Technology (IJIP) 3(6): 259–267.

O'CALLAGHAN J, LOVEDAY J. 1973. Quantitative measurement of soil cracking patterns. Pattern Recognition 5(2): 83–98.

OSHER O, BURGER M, GOLDFARB D, XU J, YIN, W. 2004. Using geometry and iterated refinement for inverse problems (1): Total variation based image restoration. In: Department of Mathematics, UCLA, LA, CA 90095, CAM Report. p. 4–13.

PERONA P, MALIK J. 1990. Scale Space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7): 629–639.

PRESTON K, DUFF MJB, LEVIALDI S, NORGREN PE, TORIWAKI J. 1979. Basics of cellular logic with some applications in medical image processing. Proceedings of the IEEE 67(5): 826–856.

PREWITT JMS. 1970. Object enhancement and extraction. In: Picture Processing and Psychopictorics. Lipkin BS, Rosenfield A eds. New York: Academic Press. p. 75–149.

ROBERTS L. 1965. Machine Perception of Three Dimensional Solids, Optical and Electro-optical Information Processing. MIT Press.

ROKACH L, MAIMON O. 2005. Clustering methods. In: Data Mining and Knowledge Discovery Handbook. Maimon O, Rokach L eds. Boston: Springer. p. 321–352.

RONDI L, SANTOSA F. 2001. Enhanced electrical impedance tomography via the mumford-shah functional. ESAIM: Control, Optimization and Calculus of Variations 6: 517–538.

ROUSHDY M. 2006. Comparative Study of Edge Detection Algorithms Applying on the Grayscale Noisy Images Using Morphological Filter. GVIP Journal 6(4): 17–23.

RUDIN LI, OSHER S, FATEMI E. 1992. Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena 60(1): 259–268.

RUSSO F, LAZZARI A. 2005. Color Edge Detection in Presence of Gaussian Noise Using Nonlinear Prefiltering. IEEE Transactions on Instrumentation and Measurement 54(1): 352–358.

SAIF JAM, HAMMAD MH, ALQUBATI IAA. 2016. Gradient based image edge detection. IACSIT International Journal of Engineering and Technology 8(3):153–156.

SCHUMAKER L. 2007. Spline functions: basic theory, 3rd ed. Cambridge: Cambridge University Press.

SMITH SM, BRADY JM. 1997. Susan—A new approach to low level image processing. International Journal of Computer Vision 23(1): 45–78.

SOBEL I, FELDMAN G. 1973. A 3x3 isotropic gradient operator for image processing. In: Pattern Classification and Scene Analysis. Duda R, Hart P eds. New York: John Wiley & Sons. p. 271–272.

SOLIN P. 2005. Elliptic partial differential equations of second order. John Wiley & Sons.

TADMOR E, NEZZAR S, VESE L. 2004. A multiscale image representation using hierarcal (BV, L2) decompositions. Multiscale Modeling & Simulation 2(4): 554–579.

TIKHONOV A, ARSENIN V. 1977. Solutions of ill-posed problems. Washington: V. H. Winstons & Sons.

VERMA OP, HANMANDLU M, SULTANIA AK, PARIHAR AS. 2013. A Novel Fuzzy System for Edge Detection in Noisy Image using Bacterial Foraging, Multidimensional Systems and Signal Processing 24: 181–198.

YAROSLAVSKY L. 1985. Digital picture processing: an introduction. Springer Verlag.

YAROSLAVSKY L, EDEN M. 1996. Fundamentals of digital optics. Birkhauser.

ZHANG X, LIU C. 2013. An Ideal Image Edge Detection Scheme. Multidimensional Systems and Signal Processing 25(4): 659–681.