

SKCS-A Separable Kernel Family with Compact Support to improve visual segmentation of handwritten data

Ezzedine Ben Braiek* Mohamed Cheriet⁺ and Vincent Doré⁺

* *Department of Electrical Engineering, CEREP, ESSTT: 5, Av. Taha Hussein, 1008 Tunis, Tunisia*

+ *Department of GPA, LIVIA, ETS: 1100 Notre-Dame West, Montreal, Quebec, Canada H3C 1K3*

Received 22 December 2004; accepted 9 February 2005

Abstract

Extraction of pertinent data from noisy gray level document images with various and complex backgrounds such as mail envelopes, bank checks, business forms, etc... remains a challenging problem in character recognition applications. It depends on the quality of the character segmentation process. Over the last few decades, mathematical tools have been developed for this purpose. Several authors show that the Gaussian kernel is unique and offers many beneficial properties. In their recent work Remaki and Cheriet proposed a new kernel family with compact supports (KCS) in scale space that achieved good performance in extracting data information with regard to the Gaussian kernel. In this paper, we focus in further improving the KCS efficiency by proposing a new separable version of kernel family namely (SKCS). This new kernel has also a compact support and preserves the most important properties of the Gaussian kernel in order to perform image segmentation efficiently and to make the recognizer task particularly easier. A practical comparison is established between results obtained by using the KCS and the SKCS operators. Our comparison is based on the information loss and the gain in time processing. Experiments, on real life data, for extracting handwritten data, from noisy gray level images, show promising performance of the SKCS kernel, especially in reducing drastically the processing time with regard to the KCS.

Key Words: Kernel with Compact Support, Separable Kernel, Multi-scale representation, Image segmentation, Handwritten data extraction.

1 Introduction

Most handwriting recognition systems can only read clearly written characters (black characters on a uniform white background or vice versa). However handwritten data contains characters of different sizes, with variable distances and its corresponding image intensity changes occur over a wide range of scales. Usually this data is decorated with uniformly distributed graphical symbols in their background. Examples can be found in mail pieces, bank checks, business forms ...etc. When such a gray-scale image is processed, it is not simple for a computer to distinguish the pertinent data from the back ground symbols; consequently, there is a great information loss resulting in broken and touched characters with the presence of noise and unnecessary information. Over the last decades, handwriting segmentation has become the subject of intensive research. Various mathematical tools and large number of approaches have been developed. All these works aim to improve visual shapes of handwriting characters and to perform intelligent preprocessing

Correspondence to: Ezzedine.Benbraiek@esstt.rnu.tn, cheriet@livia.etsmtl.ca

Recommended for acceptance by Josep Lladós

ELCVIA ISSN: 1577-5097

Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

procedures in order to extract the pertinent information and to improve data quality [11,12], before applying the recognition algorithm [14]. For dealing with this problem, scale-space representation has been developed by the computer vision community (in particular by Witkin [3], Koenderink [2], Yuille and Poggio [25] and Lindeberg [8]), because of its efficiency in describing the real-world structures. It has been shown that the concept of scale is of crucial importance when we aim to analyze and interpret the structure of images by automatic methods; as it is the case of our application.

In multi-scale representation, the scale-space is usually generated by the Gaussian kernel where a convolution product is computed at each scale. It has been shown by Badaud et al. [7] that the Gaussian function is the only kernel in a broad class of functions which satisfies adequate scale-space conditions and offers many beneficial properties. Unfortunately, the Gaussian kernel has two practical limitations: information loss caused by the unavoidable Gaussian truncation and the prohibitive processing time due to the mask size. In a recent paper, Remaki and Cheriet [1] have proposed a new kernel family, with compact support in scale-space; which derived from the Gaussian; they called KCS (Kernel with Compact Support). This kernel is able to recover the information loss while reducing processing time.

In this paper, we propose a separable version of the KCS. This new kernel denoted SKCS (Separable Kernel with Compact Support) has the capability to achieve accurate information extraction and the advantage to reduce the processing time. Experiments, on real life data, for extracting handwritten data, from noisy gray level images are presented. A practical comparison is established between results obtained by using the KCS and the SKCS operators. Our comparison is based on the information loss and the gain in time processing.

2 Scale-space based approach

To improve performances of handwriting recognition systems visual shapes are of outstanding importance. In gray-scale images, intensity occurs over a wide range of scales. When such image is binarized there is a significant information loss, resulting in broken characters, touched characters, with the presence of noise and undesirable information. To overcome this problem, we adopt the scale-space technique [5].

The main idea of scale-space representation is to embed the original image in a one-parameter family of derived images, in which the fine scale details are successively suppressed. It consists of using a convolution with a chosen kernel in the smoothing operation.

2.1 The Laplacian of Gaussian Operator: a brief review

The use of LoG convolution masks was first suggested by Marr and Heldreth [4]. Since the second derivatives of such a kernel are considered as a decision criteria in scale-space (to obtain visual segmentation of handwritten data), we start recalling the Laplacian of Gaussian operator [23].

If $g_{\mathbf{s}}(x,y)$ is the 2D symmetrical Gaussian distribution with standard deviation \mathbf{s} :

$$g_{\mathbf{s}}(x,y) = \frac{1}{2\pi\mathbf{s}^2} e^{-\left(\frac{x^2+y^2}{2\mathbf{s}^2}\right)} \quad (1)$$

Then, the Laplacian of Gaussian operator (LoG) is:

$$LoG_{\mathbf{s}} = \nabla^2 g_{\mathbf{s}}(x,y) = -\frac{1}{\pi\mathbf{s}^4} \left(1 - \frac{x^2+y^2}{2\mathbf{s}^2}\right) e^{-\left(\frac{x^2+y^2}{2\mathbf{s}^2}\right)} \quad (2)$$

The LoG operator has been reported informally in [13, 4] and more details concerning this operator are given in [5, 9] and [20]. The different parameters of the operator are shown in figure 1 where w represents the distance between the zero-crossings.

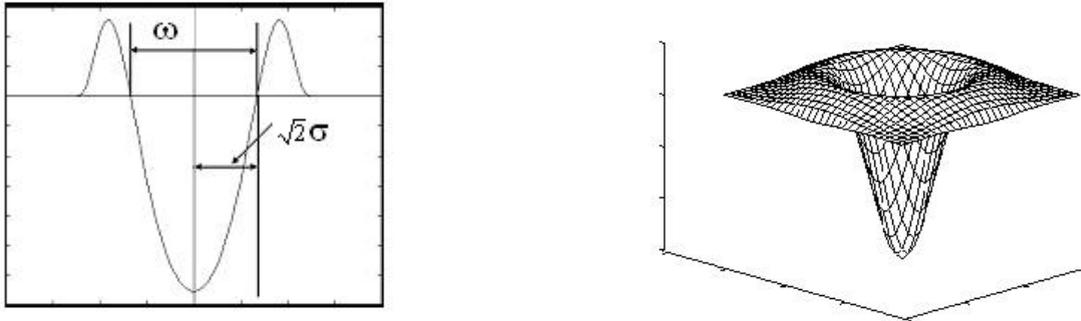


Figure 1: The 1D and 2D dimensional profile of LoG

2.2 Scale-space algorithm

As we can see, the size of the mask depends on the value of \mathbf{s} . For the implementation of the operator (the LoG kernel in this case), the general scale-space algorithm may be presented as follows:

- a- For each pixel of the input Gray-level image, apply the LoG kernel with a step scale \mathbf{s} . Begin with a step scale $\mathbf{s} = \mathbf{s}_{max}$ sufficiently high to extract a minimum amount of information, free of noise.
- b- Threshold the output image by turning off pixels with positive or null values and turning on those with negative values.
- c- Turn on the neighbours of pixels, output image of the step (b).
- d- Decrease the step scale \mathbf{s} .
- e- Apply the LoG kernel to the original image, with the step scale \mathbf{s} , only on pixels being turned on in the previous step.
- f- Threshold the image as in step (b). The output image constitutes the desired result at the scale-space \mathbf{s} .
- g- Repeat the process, on the output image obtained at step (f), from step (c) to step (g), until $\mathbf{s} = \mathbf{s}_{min}$. Obtain the desired fine image.

2.3 Limitations of the Gaussian kernel and The KCS as a solution

The algorithm has been implemented with the objective to perform image segmentation efficiently. The problem with natural images is that they are noisy and vary in contrast. By observing the algorithm behaviour on different data qualities; we have noticed a sensitive degradation on the segmented images leading to broken characters when contrast is sufficiently low and variable [1]. The computation accuracy depends on the mask size. Wide masks give more precise computation, but they increase the cost of processing time. Smaller mask sizes decrease the processing time but the accuracy is sometimes severely diminished, which induces information loss. To cope with these problems, without contradicting the uniqueness of the Gaussian kernel and by preserving the most important and useful Gaussian properties, required to perform image segmentation properly, Remaki and Cheriet [1] proposed a new kernel with Compact Support (KCS). This kernel, derived from the Gaussian, has been introduced in order to recover the information loss and to reduce the prohibitive time processing when Gaussian Kernel is used in scale-space representation [2, 3]. With the KCS there is no need to cut off the kernel while the processing time is controlled because the mask is the support of the kernel itself. The KCS is obtained by performing a topological transformation of the \mathcal{R}^2 space into a unit ball through a change of variables. This

transformation packs all the information in the unit ball. With the new variables, the Gaussian is extended over all the \mathfrak{R}^2 space by taking zero values outside the unit sphere. Therefore, the mask dimension is considerably reduced, from $11.32\mathbf{s}$ for the Gaussian to only $2\mathbf{s}$ for the KCS. The KCS construction, formulation, and analysis are presented with details in [1].

The generalized KCS may be expressed as follows:

$$\mathbf{r}_{\mathbf{s},\mathbf{g}}(x,y) = \begin{cases} \frac{1}{C_{\mathbf{g}}\mathbf{s}^2} e^{\mathbf{g}\left(\frac{\mathbf{s}^2}{x^2+y^2-\mathbf{s}^2}+1\right)} & \text{if } x^2+y^2 < \mathbf{s}^2 \\ 0 & \text{elsewhere} \end{cases} \quad (3)$$

Where: \mathbf{s} and \mathbf{g} are, respectively, the standard deviation and the width parameter.

The parameter \mathbf{g} controls the distance between the zero-crossing of the KCS to the origin of the axis; it doesn't affect the nature of $\mathbf{r}_{\mathbf{s}}$ functions; they remain kernels with compact support type. Furthermore, if $\gamma \geq 2$, the desired characteristics are guaranteed, which means that the first and second derivative curves exhibit the same behavior as the Gaussian. Figure 2 depicts the 1D and 2D profiles of the KCS.

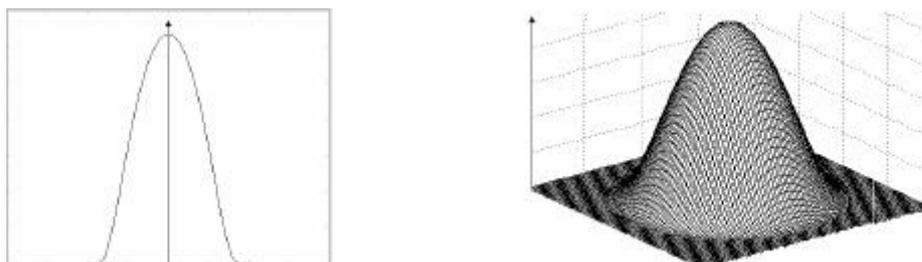


Figure 2: One and two-dimensional profiles of the KCS Kernel

Data segmentation is determined by the zero-crossings of the second derivative operator [6] (Laplacian of KCS (LoKCS)) and it is defined by the convolution of the image with LoKCS which is given by:

$$\text{LoKCS} = \nabla^2 \mathbf{r}_{\mathbf{s},\mathbf{g}}(x,y) = \begin{cases} \frac{2\mathbf{g}}{\mathbf{p}} \left[\frac{\left((x^2+y^2)^2 + \mathbf{g}\mathbf{s}^2(x^2+y^2) - \mathbf{s}^4 \right)}{\left(x^2+y^2 - \mathbf{s}^2 \right)^4} \right] e^{\left(\frac{\mathbf{s}^2}{x^2+y^2 - \mathbf{s}^2} + \mathbf{g} \right)}, & \text{if } x^2+y^2 < \mathbf{s}^2 \\ 0, & \text{elsewhere} \end{cases} \quad (4)$$

The cross-sections of the LoKCS function are shown on figure 3.

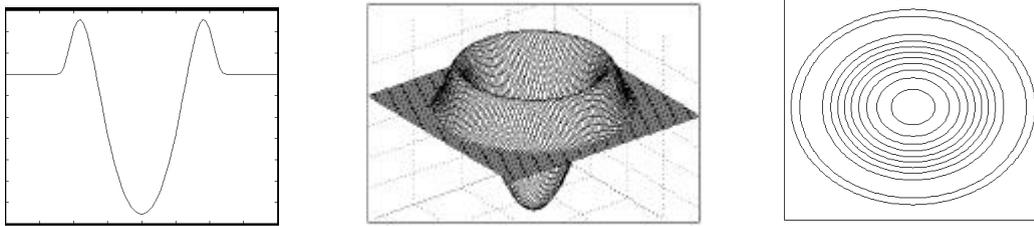


Figure 3: One and two-dimensional profiles and the contour peaks of the Laplacien of KCS (LoKCS)

2.4 The KCS main features and their improvement

Authors have shown that, thanks to their compact support property, the KCS give very satisfactory segmentation results, with an effective response to both practical limitation problems when using the Gaussian kernel, namely the information loss caused by the truncation of the Gaussian kernel and the prohibitive processing time due to the mask size.

Results have been confirmed by qualitative and quantitative comparisons of results obtained by both the Gaussian and the KCS kernels.

The question raised now is how to increase the KCS performances. In scale-space formulation, a convolution product with the kernel is computed at each scale, as mentioned above in the algorithm (see 2.2). These successive convolutions require substantial processing time. Considerable savings of time may be achieved by using separable convolution masks. For this purpose, we propose a Separable version of the KCS, denoted SKCS (Separable Kernel with Compact Support).

3 Separable version of the KCS

3.1 Introduction

The KCS formula is shown in (3). As we can see, this formula is not separable. However, the product of 1D KCS following x and y, leads to a new 2D operator [10]. To design the new proposed kernel, we define an implementation based on the decomposition of the Laplacian of the KCS shape, as the sum of two 1D filters.

3.2 Building the SKCS

The separable version of the KCS is defined as the product of one-dimensional KCS kernels [17]. To express the new version, we must first recall the one-dimensional KCS formula.

$$r_{\mathbf{s}, \mathbf{g}}(x) = \begin{cases} \frac{1}{C_{\mathbf{g}} \mathbf{s}^2} e^{\mathbf{g} \left(\frac{\mathbf{s}^2}{x^2 - \mathbf{s}^2} + 1 \right)} & \text{if } x^2 < \mathbf{s}^2 \\ 0 & \text{elsewhere} \end{cases} \quad (5)$$

- where: $C_{\mathbf{g}}$: is the normalization constant
- \mathbf{s} : is the standard deviation KCS
- \mathbf{g} : is the width parameter.

Let the one parameter 2D SKCS function be:

$$SKCS(x, y) = \Phi(x, y) = \mathbf{r}(x) \cdot \mathbf{r}(y)$$

So,

$$\Phi_{\mathbf{g}}(x, y) = \frac{e^{2\mathbf{g}}}{C_{\mathbf{g}}^2} e^{\left(\frac{\mathbf{g}}{x^2-1}\right)} \cdot e^{\left(\frac{\mathbf{g}}{y^2-1}\right)} \tag{6}$$

And,

$$\Phi_{\mathbf{s}, \mathbf{g}}(x, y) = \frac{1}{\mathbf{s}^2} \Phi_{\mathbf{g}}\left(\frac{x}{\mathbf{s}}, \frac{y}{\mathbf{s}}\right)$$

This leads to the next generalized SKCS formula:

$$\Phi_{\mathbf{s}, \mathbf{g}}(x, y) = \begin{cases} I^2 e^{\left(\frac{\mathbf{g}\mathbf{s}^2}{x^2-\mathbf{s}^2}\right)} e^{\left(\frac{\mathbf{g}\mathbf{s}^2}{y^2-\mathbf{s}^2}\right)} & \text{if } x^2 < \mathbf{s}^2 \text{ and } y^2 < \mathbf{s}^2 \\ 0 & \text{elsewhere} \end{cases} \tag{7}$$

With
$$I = \frac{e^{\mathbf{g}}}{C_{\mathbf{g}}\mathbf{s}}$$

Where $C_{\mathbf{g}} = \int_{-1}^1 e^{\left(\frac{\mathbf{g}}{x^2-1}\right)} dx$ is the normalization constant.

Then, if N is the number of points in the discretization process

$$C_{\mathbf{g}} = \Delta x \sum_{i=0}^N e^{\left(\frac{\mathbf{g}}{x_i^2-1}\right)} \quad \text{Where } x_i = (-1 + i\Delta x) \quad \text{and } \Delta x = \frac{2}{N} .$$

In our experiments, we have kept the parameter \mathbf{g} constant, while varying \mathbf{s} . $C_{\mathbf{g}}$ is then computed just once in the process. However, if we need varying the parameter \mathbf{g} during the process; we propose as in [1] a high-speed algorithm to compute $C_{\mathbf{g}}$ for every \mathbf{g} from $C_{\mathbf{g}_0}$ and $C_{\mathbf{g}_1}$ for the given values \mathbf{g}_0 and \mathbf{g}_1 .

Proposition: If $h(\mathbf{g})$ is the solution of the following second-order differential equation:

$$h''(\mathbf{g}) - \left(\frac{1}{4} + \frac{1}{2\mathbf{g}}\right)h(\mathbf{g}) = 0, \quad \mathbf{g} \in [\mathbf{g}_0, +\infty[$$

Then $C_{\mathbf{g}} = h(\mathbf{g})e^{\mathbf{g}/2}$ (Proof: see [1]).

By using the second-order Taylor expansion, we discretize the upper equation:

$$\begin{cases} h_{i+1} = (\Delta g^2 (\frac{1}{4} + \frac{1}{2g_i}) + 2)h_i + h_{i-1} \\ h_0 = C_{g_0}, \quad h_1 = C_{g_1} \end{cases} \quad (8)$$

Then, $C_{g_i} = h_i e^{\frac{1}{g_i}}$ where Δg is the g -step and $g_i = g_0 + i\Delta g$. So, C_g is obtained for all values of g with few operations.

The next figure depicts the 1D and 2D profiles of the SKCS.

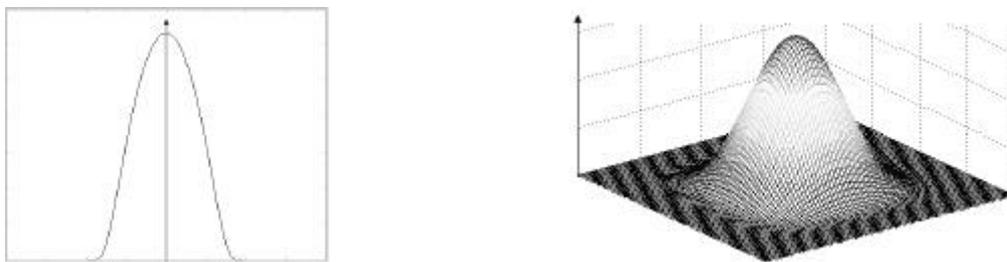


Figure 4: One and two-dimensional profiles of the SKCS Kernel

As the KCS, the SKCS exhibits the same behaviour of the Gaussian. However, the SKCS Laplacian presents, for $g = 1$, a new maximum at the origin (see figure 5). But it can be easily shown (following the same arguments as [1]) that if $g > 2$, the first and second SKCS derivatives exhibit the same behaviour as the Gaussian's e.g. the convolution with the SKCS doesn't introduce new extreme.

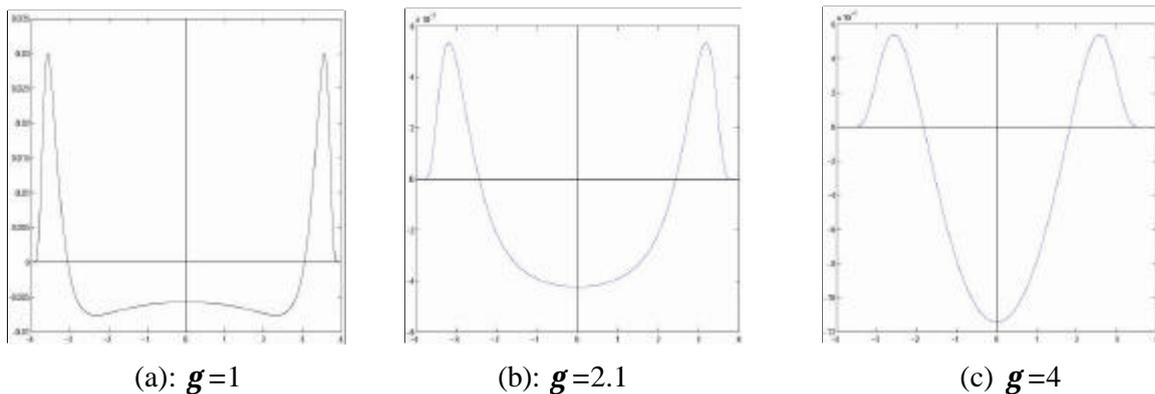


Figure 5: the influence of the parameter g on the width of LoSKCS peak, for $s = 4$.

The parameter g of the KCS controls the distance between the zeros-crossing of the KCS Laplacian to the origin of the axes. For the SKCS, g controls the distance a from the origin of the axes to the zeros-crossing-line of the SKCS Laplacian along one of the axes. Hence, for some applications, two parameters g_x, g_y can be used, which control, respectively, the distance a_x and a_y (see figure 6). As for the one

dimensional KCS, if $\mathbf{g} = (\frac{n^2}{2} - \frac{3}{2n^2} + 1)$ then for all \mathbf{s} , $\mathbf{a} \approx \mathbf{s} / n$ where n is a real. (Proof: see [1]). This proves that for a fixed n , \mathbf{g} is fixed and doesn't depend on the value of \mathbf{s} .

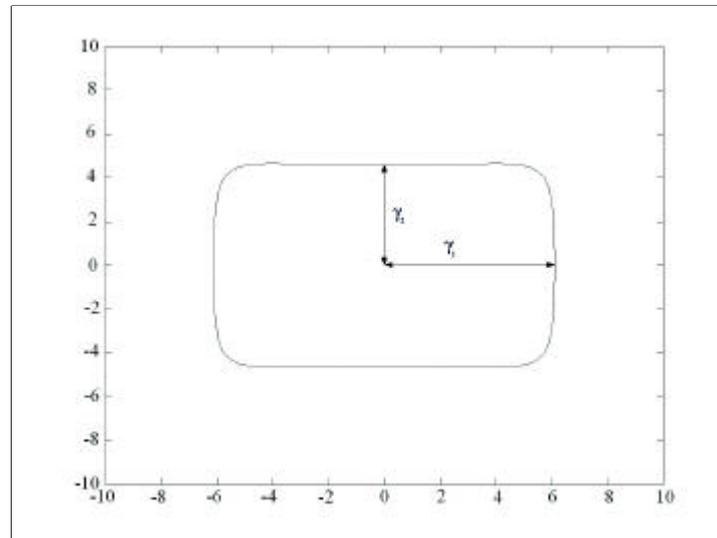


Figure 6: Support of the SKCS for $\mathbf{s} = 10$ with two parameters $\mathbf{g}_1 = 2.1$, $\mathbf{g}_2 = 4$

3.3 Discussion of some properties of the SKCS

Since one of our concerns when building the SKCS was to preserve the main properties of the Gaussian in order to perform image segmentation efficiently, it is important to verify that the SKCS satisfies the following properties:

3.3.1 Recovering the original signal when the scale parameter is close to zero.

It has been shown in [1] that the one dimensional KCS kernel \mathbf{r}_g is in the space $D(\mathfrak{R})$. As the SKCS kernel \mathbf{j}_g is the product of two one-dimensional KCS kernel, it is defined in $D(\mathfrak{R}^2)$.

Theorem [13]: Let $f \in L^p(\mathfrak{R}^2)$, where $1 \leq p \leq +\infty$, then $\lim_{s \rightarrow 0} \int_{\mathfrak{R}^2} |(\mathbf{j}_{g,s} * f - f)|^p d\mathbf{m} = 0$

The extension f of an image by 0 for all \mathfrak{R}^2 is $L^p(\mathfrak{R}^2)$ for $1 \leq p \leq +\infty$. This ensures the recovering of the original image when \mathbf{s} tend to zero [15].

3.3.2 Continuity and regularization

By its construction from the one dimensional KCS, the SKCS respects the same properties of continuity and regularization: e.g.:

- The scale-space representation is continuous with respect to the scale parameter \mathbf{s} , which means that $\mathbf{j}_{g,s} * f$ is continuous with \mathbf{s} .
- As the SKCS is in the space of mollifiers (smoothing functions), the convoluted image with it is in C^∞ .

3.3.3 Heisenberg Uncertainty Principle (H.u.p.)

As the one dimensional SKCS is exactly the one dimensional KCS, we have here again the same property for these kernels: when g increases, the H.u.p. decreases rapidly and becomes close to $(1/4\pi)$, which is the optimal value reached by the Gaussian kernel [16].

3.4 Important remarks

The 1D KCS and SKCS kernels have nearly the same behaviour (see figure 2 and 4). But the fact that the SKCS is separable means that its support is of a square form. This shape implies that all the pixels which are on the frontier will not be equidistant from the kernel center. So, the information on the diagonals may be privileged. Only, we have noticed that, however all the points which belong to the kernel support and not to the ball of the same center as the kernel of radius S , constitute at most 3% of the total kernel weight. In other words, in the convolution process with the image, the information excited by this part of the support will have a little weight on the final sum. On the other hand, the excitatory region of the support (region of important weights), which is close to the center of the mollifier (smoothing kernel), is almost circular and similar to the KCS (figure 7).

We have plotted the image difference KCS-SKCS, with the same values of γ and S (figure 7(c)). We can see that the weights of the SKCS are more important at the corners near the summits of the square. The most important weights of the KCS are located in a lozenge situated at the kernel centre.

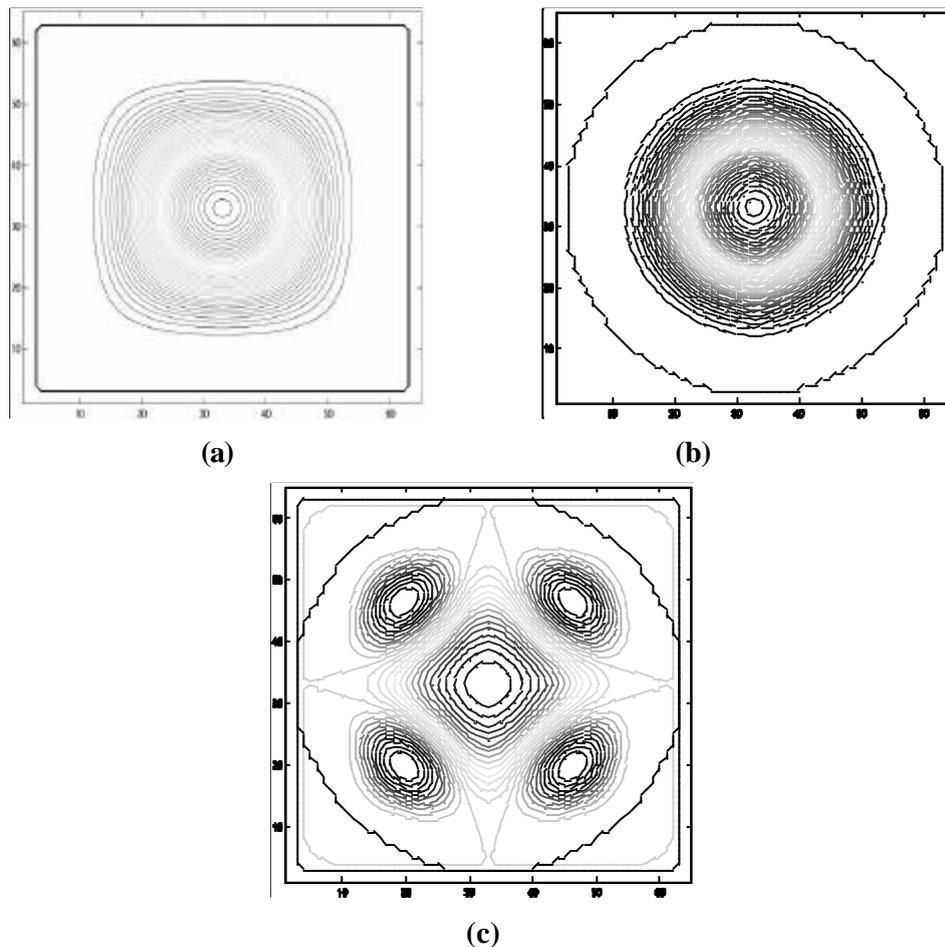


Figure 7: Peaks contour plot of the SKCS (a), KCS (b) and their difference (KCS-SKCS) (c).

3.5 The SKCS Laplacien

Here, we are interested in computing the Laplacian of the SKCS (LoSKCS), we proceed as before:

$$LoSKCS = \nabla^2 \Phi_{\mathbf{s}, \mathbf{g}}(x, y)$$

By re-writing the 2D SKCS formula:

$$\Phi_{\mathbf{s}, \mathbf{g}}(x, y) = \begin{cases} \mathbf{1}^2 \cdot e^{h(x)} \cdot e^{h(y)} & \text{if } x^2 < \mathbf{s}^2 \text{ and } y^2 < \mathbf{s}^2 \\ 0 & \text{elsewhere} \end{cases} \quad (9)$$

With

$$h(s) = \frac{\mathbf{g}\mathbf{s}^2}{(\mathbf{s}^2 - \mathbf{s}^2)} \quad (10)$$

We then have:

$$LoSKCS = \nabla^2 \Phi_{\mathbf{s}, \mathbf{g}}(x, y) = \begin{cases} K \cdot [A(x) + A(y)] \cdot e^{[h(x)]} \cdot e^{[h(y)]} & \text{if } x^2 < \mathbf{s}^2 \text{ and } y^2 < \mathbf{s}^2 \\ 0 & \text{elsewhere} \end{cases} \quad (11)$$

With

$$K = \frac{2\mathbf{g} \cdot e^{\mathbf{g}}}{C\mathbf{g}} \quad (12)$$

$$A(s) = h'(s)^2 - h''(s) \quad (13)$$

$$h'(s) = \frac{2 \cdot \mathbf{g} \cdot \mathbf{s}^2 \cdot s}{(\mathbf{s}^2 - \mathbf{s}^2)^2} \quad (14)$$

$$h''(s) = \frac{-2\mathbf{g}\mathbf{s}^2[(\mathbf{s}^2 - \mathbf{s}^2)(3\mathbf{s}^2 + \mathbf{s}^2)]}{(\mathbf{s}^2 - \mathbf{s}^2)^4} \quad (15)$$

Hence, the LoSKCS operator is written as a sum of two one-dimensional filters (one column vector and one row vector). Its equation can be rewritten as the sum of two filters as follows:

$$\nabla^2 SKCS(x, y) = H(x, y) + H(y, x) \quad (16)$$

Where

$$H(x, y) = \mathbf{m} \cdot A(x) \cdot e^{[h(x)]} \cdot e^{[h(y)]} \quad (17)$$

With

$$\mathbf{m} = \frac{e^{\mathbf{g}}}{\mathbf{s}^2} \quad (18)$$

We can still notice that the support of the LoSKCS is of square form. If g is small, the shape of the mollifier (smoothing kernel) shows 4 humps (2 horizontals and 2 verticals) instead of the regular hump of the LoKCS around the center of the kernel (figure 8), the contour peaks of the LoSKCS are circles in the middle but quickly become round of square forms. The behavior of the LoSKCS along the diagonals is then different from the LoKCS.

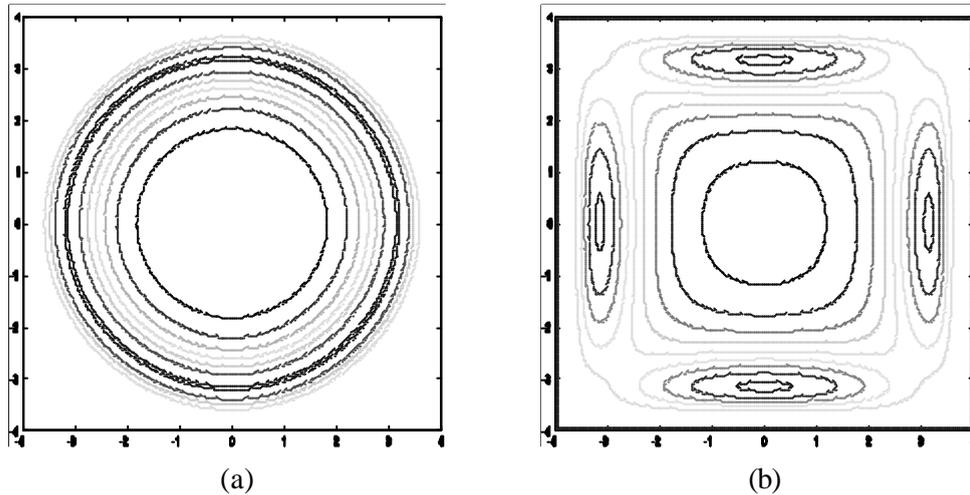


Figure 8: Peaks contour plot of LoKCS (a) and of LoSKCS (b) for $g = 2.1$

It is important to mention that, when g increases, the shape of the LoSKCS becomes closer to the shape of the LoKCS (figure 9). In order to get a kernel nearly behaving as the KCS, we advise the reader to use a g value larger than 4.

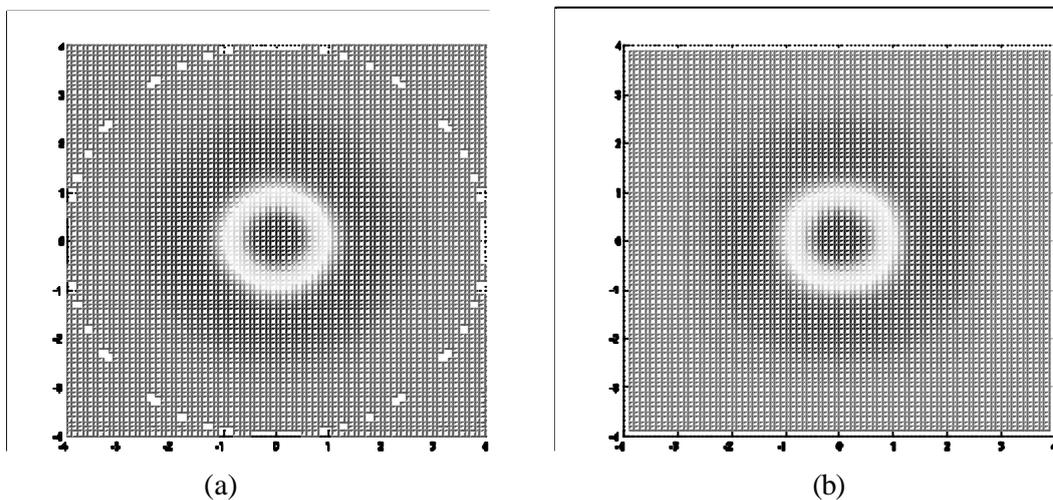


Figure 9: LoKCS (a) and LoSKCS (b) shapes for $g = 10$

4 Experimental results

For the practical aspect, in order to investigate the SKCS impacts on extracting handwritten data from degraded and noisy images, experiments have been conducted on real data, from CEDAR, at SUNY Buffalo, data base, that contain Zip codes scanned in US Postal offices on live mail and our local data base. The filters implementation is performed using a PIV-700MHz computer, under Matlab software. The

performance of the proposed method is judged by visual inspection using criteria of image shape, such as thickness, intensity and connectivity of the strokes of character shapes with a weight given to each.

To achieve this purpose, we have used the algorithm described in 2.2 and replaced the LoG operator by the LoKCS operator and the LoSKCS, respectively.

4.1 Algorithm parameters

For both kernels, KCS and SKCS, we have chosen the same width parameter ω and the same mask size contained in the interval $]-s, +s[$, then the mask size is fixed $= 2s$. For the multi-scale representation, we have decreased the s value from $s_{\max} = 4$ (which allows noise immunity) to $s_{\min} = 2$ (which allows accurate shapes) and a scale step of 0.5. In terms of mask dimension, it corresponds to a mask variation from (8x8) to (4x4).

4.2 algorithm behavior

In order to compare the performance of the KCS and the SKCS operators, our tests focused exclusively on qualitative comparison. To perform this comparison, the methodology we followed is based on the capacity of each kernel to resist to progressively degraded input images without adapting their parameters to the degradation. To perform this task, we have fixed appropriate parameters with respect to a certain number of images for both the LoKCS and LoSKCS operators. We then used these parameters to process a large image data base. We focused the discussion on two sets of images.

- a. In the first set (see figure 10), we produce the original image and the results achieved by the KCS and the SKCS operators.

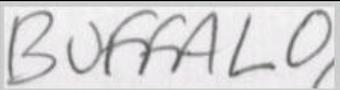
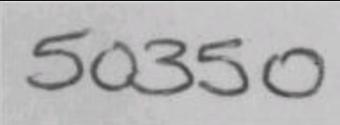
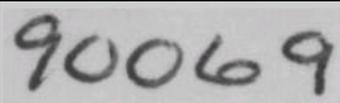
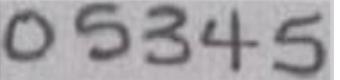
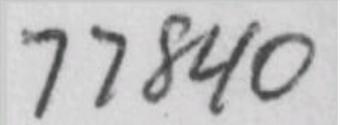
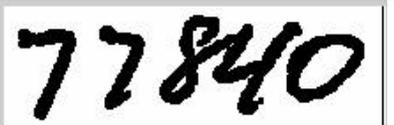
		
		
		
		
		
(a): Original Image	(b): Segmentation using the KCS	(c): Segmentation using the SKCS

Figure 10: Segmentation of gray level images

By observing the behaviour of the algorithms on different data qualities, we can confirm, in this case, that both operators give good results with accurate segmentation and high visual quality even with the presence of noise. This effective segmentation shows the robustness of the two operators in extracting visual shapes of handwritten data from noisy images and will bring an increase to the overall recognition system rate.

b. In the second set (see figure 11); we use the same parameters as in (4.1.a), and we produce degraded images for the two operators.

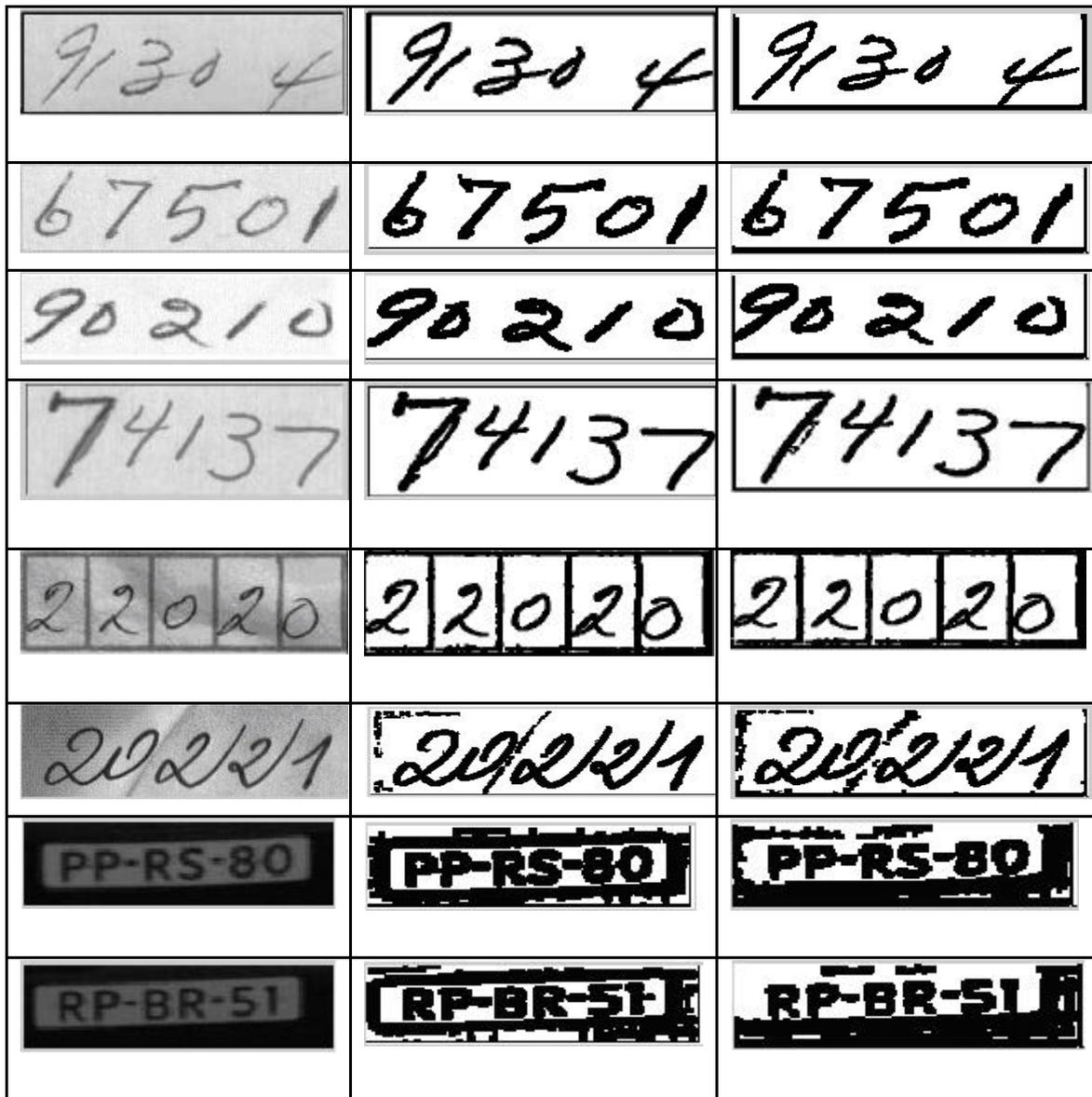


Figure 11: Segmentation of degraded images

If we focus on the segmented images in figure 11, we can see that the two operators yield satisfactory results, but we observe certain degradation which appears on the image segmented by the LoSKCS, especially in the extremities and in the non homogenous zones of the character. This phenomenon is due to the width of LoSKCS excitatory region (see comments on 3.4 section). It can be explained by the fact that the LoSKCS has a tendency to smooth more pixels than the LoKCS in certain regions of the image. We think that this degradation doesn't affect seriously the segmentation and the results, obtained by the SKCS, are sufficiently improved, to make the recogniser system task particularly easier.

4.3 Improvement of the SKCS segmentation

The methodology, we followed to compare the performance of the KCS and SKCS kernels, is based on the capacity of each kernel to segment properly the image. For this purpose, we have set the same algorithm parameters for the two kernels, without adapting them to the image degradation. To improve segmentation data quality, we have to adapt the SKCS algorithm parameters, or to adjust them to their optimal values, as shown in figure12.

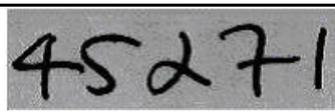
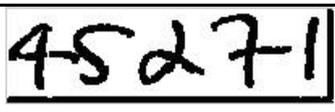
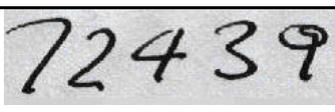
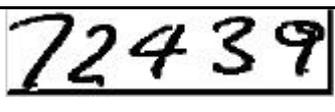
		
		
Original image	$g=2.4$	$g=4.8$

Figure 12: Improvement of the image segmentation by varying the width parameter g

We observe that the thickness of the characters augment when g increases. Hence, an appropriate choice of g improves the segmentation data quality. More experiences are currently underway, on a large testing set, in order to adapt the SKCS algorithm parameters and to test its robustness.

4.4 Processing time evaluation

In this set (see figure 13); we use the same parameters as in (4.2.a), and we produce different images for the two operators in order to evaluate the processing time of the KCS and the SKCS operators.

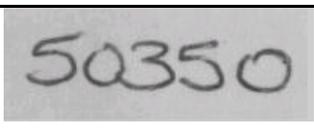
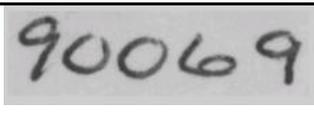
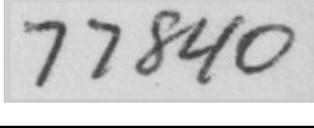
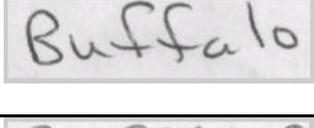
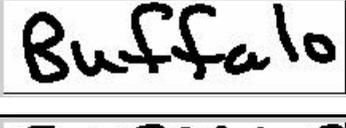
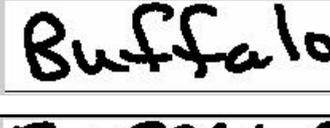
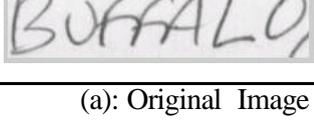
		
		
		
		
		
(a): Original Image	(b): Segmentation by using the KCS	(c): Segmentation by using the SKCS

Figure 13: Some sample images and the segmented images obtained by applying the KCS and the SKCS operators.

In figure 14, we summarize the results of the different images to illustrate the processing time, in seconds, corresponding respectively to the KCS and the SKCS operators.

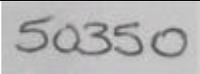
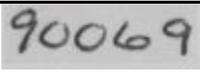
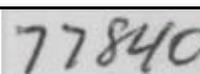
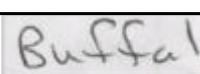
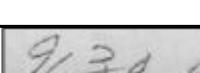
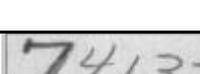
	55.04	17.69
	54.88	17.56
	62.64	20.20
	50.65	16.32
	61.08	19.89
	64.02	21.82
	51.23	16.58
Original Image	Processing time for KCS	Processing time for SKCS

Figure 14: KCS and SKCS processing time (in seconds).

We notice that the processing time required for the LoSKCS operator is distinctly less than that required for the LoKCS operator. Indeed, for the separable version of the KCS, each of the convolutions is realized as successive one-dimensional row and column convolutions. Thus reducing the total number of operations required at each pixel from M^2 to $2M$, where the size of the filters is $M \times M$ pixels.

Conclusion

In this paper, we have presented a new Separable version of Kernels with Compact Support. The new kernels (SKCS) are used to segment visual data in variable contrasted images with the presence of noise. We have shown that the obtained kernels preserve the most important properties of the Gaussian kernel in order to perform image segmentation efficiently. A thorough study of some important properties is provided and a comparative analysis based on the mask sizes of the KCS and the SKCS kernel has been discussed. An application for extracting handwritten data from noisy images is presented including a qualitative comparison between the results obtained by the KCS and the SKCS. It has been shown that, with the two operators, satisfactory segmentation with high visual quality is obtained. When applying the same parameters to both kernels, a little degradation appears when using the SKCS. This degradation is due to the SKCS shape behaviour. It corresponds only to a small part loss of information (3%) and may not have serious influence on future treatments. However, considerable savings in processing time are achieved by using the separable KCS kernel (SKCS). This advantage renders use of large LoSKCS operators practical and opens new application fields to the SKCS.

References

- [1] L. Remaki and M. Cheriet "KCS-New Kernel Family with Compact Support in Scale Space: Formulation and Impact," *IEEE Trans. on image processing*. Vol.9. No6. pp. 970-981 June 2000.
- [2] J. J. Koenderink, "The structure of images," *Biological Cybernetics*. vol. 53, pp.363-370, 1984.
- [3] A.P. Witkin, "Scale-space filtering," In *Proc. 8th Int. Joint Conf. Art. Intell.*, pp. 1019-1022, Karlsruhe, West Germany, Aug. 1983.
- [4] D. Marr and E. Hildreth, "Theory of edge detection," In *Proc. Of the Royal Society of London, Series B*, vol. 207, pp.187-217, 1980.
- [5] M.Cheriet, R. Thibault and R. Sabourin, "A multi-resolution based approach for handwriting segmentation in gray level images," In *IEEE Int. Conf. On IP*, Austin, pp.159- 168, 1994.
- [6] M.Cheriet, "Extraction of handwritten data from noisy gray-level images using multi-scale approach" *Int. Journal Pattern Recognition Artif. Intell.*, vol.13, no.5. pp. 665-685, 1999.
- [7] J. Babaud, A. P. Witkin, M. Baudin, R. O. Duda "Uniqueness of the Gaussian Kernel for Scale-Space Filtering" *IEEE Trans. On PAMI*, vol.8, no.1, pp. 26-33, January 1986.
- [8] T. Lindeberg, "Scale-Space Theory in Computer Vision," Norwell, MA: Kluwer, 1994, p. 423.
- [9] A. Huertas, G. Medioni, "Detection of Intensity Changes with Sub-pixel Accuracy Using Laplacian-Gaussian Masks" *IEEE Trans. PAMI*, vol.8, no.5, pp. 651-664, 1986.
- [10] E. Ben Braiek, L. Remaki and M. Cheriet, "Extraction of Handwritten Data from Noisy Gray Level images using a separable version of the KCS Kernel Family" *ICISP'01*, pp. 125-131, Agadir 2001.
- [11] S. Impedovo, "Frontiers in handwriting recognition" *NATO-ASI Series*, vol. F124, Springer Verlag, pp. 7-43, 1994.
- [12] E. Lecolinet and O. Baret, "Cursive word recognition *NATO-ASI Series*, vol. F124, Springer Verlag, pp. 235-264, 1994.
- [13] H. Brezis, *Analyse fonctionnelle et applications*. Paris, France : Masson, 1986.
- [14] M. Cheriet, J.N. Said, C.Y. Suen, A Recursive Thresholding Technique for Image Segmentation, *IEEE Transactions on Image Processing*, vol.7, no.6, pp.918-922, 1998.
- [15] E. Godlewski and P. A. Raviart, "Hyperbolic systems of conservation laws," in *mathematics and applications*. Paris, France: Ellipses-Edition marketing, 1991.
- [16] H. Reinhard, *Éléments de mathématiques de signal*, Paris, France:Dunod 1995.
- [17] E. Ben Braiek and M. Cheriet, "SKCS-New Kernel Family with compact support" *IEEE conf. On "ICIP'04"*, Singapore, October 2004.
- [18] P. J. Burt, "Fast filter transform for image processing" *Comput. Vision. Graph. Image Processing*, vol.16, pp.20-51, 1981.
- [19] A.L. Yuille and T.A. Poggio, Scaling theorems for zero-crossings. *IEEE Trans. Pattern Analysis and Machine Intell.*, Vol. 8, pp. 15-25, 1986.
- [20] F. Bergholm, Edge focusing, *IEEE Trans. on PAMI*, 9(6), pp. 726-741, Nov. 1987.