Convergence of Level-Wise Convolution Differential Estimators

Damien Gonzalez[1], Rémy Malgouyres[1], Henri-Alex Esbelin[1], and Chafik Samir[2]

[1] Clermont-Université, LIMOS UMR 6158 CNRS, Complexe des Cézeaux, 63172 Aubière, France
{damien.gonzalez,remy.malgouyres}@u-clermont1.fr, Alex.Esbelin@univ-bpclermont.fr
[2] Clermont-Université, ISIT UMR 6284 CNRS, Complexe des Cézeaux, 63172 Aubière, France
chafik.samir@u-clermont1.fr

Abstract. Differentials estimation of discrete signals is almost mandatory in digital segmentation. In our previous work, we introduced the fast level-wise convolution (LWC) and its complexity of $O(2n.log2(m))$. We present convergence proofs of two LWC compatible kernel families. The first one is the pseudo-binomial family, and the second one the pseudo-Gaussian family. In the experimental part, we compare our method to the Digital Straight Segment tangent estimator. Tests are done on different digitized objects at different discretization step using the DGtal library.

Keywords: Differential estimator, discrete differential operator, fast convolution, sparse differential operator, FFT.

Introduction

Digital segmentation algorithms such as active contour models often use signal parameters as energy. Estimation of differentials is almost mandatory for most of them as they use regularization terms like the snake algorithm [7]. Previous works are divided into two categories: the non convolutional and the convolutional methods.

1. Non convolutional methods. The Digital Straight Segment (DSS) tangent estimator [8,2] extracts maximal DSS and computes their tangents. One of the advantages of this method is its ability to detect corners, its convergence rate is $O(\frac{1}{3})$. The Taylor polynomial approximation [12] fits the values of a digital function by a polynomial. It introduces a roughness parameter to relax the function values within an interval. It has a bounded maximal error of $O(h^{\frac{1}{1+k}})$ for the k^{th} derivative and a resolution h, its convergence rate is $O(\frac{1}{1+k})$. The Global min-curvature estimator (GMC) is a curvature estimator along digital contours. It first estimates the uncertainty of tangents using a tangential cover, and then minimizes the global curvature.

R. Gonzalez-Diaz, M.-J. Jimenez, B. Medrano (Eds.): DGCI 2013, LNCS 7749, pp. 335–346, 2013.

2. Convolutional methods. The binomial convolution Method (BC) [11,3] approximates differentials with finite differences after applying a digital version of the scale space [9] function smoothing, using integer-only binomial coefficients as a convolution mask. It is noise resistant, has a convergence rate of $O(h^{(\frac{2}{3})^k})$ and a complexity of $O(n.m)$ with n the size of the image and m the one of the convolution mask.

In our previous work [4] we defined a differential estimator based on a Level-wise Convolution Kernel, thus obtaining a LWC method. Compared to BC, it has a better complexity and speed and appears to provide similar error results in experiments. In this paper, we use two techniques to generate level-wise kernels from a given smoothing kernel. We then focus on theorems and proofs of convergence for two kernel families and experimental convergence confirmation as well. More precisely, under some relaxed assumptions of absence of noise and for floating point methods, we prove that LWC converges in $O(h^2)$ with both a level-wise pseudo-Gaussian kernel and a level-wise binomial kernel. In the presence of a uniform noise in $O(h^\alpha)$, we prove a result similar to [3] for the faster LWC method. We have not been able to prove such a result for a pseudo-Gaussian kernel or a level-wise pseudo-gaussian kernel. At last, to further improve comparison with other methods we also show results comparing the precision and speed of our LWC with respect to DSS. The scope of this paper being limited to first order derivative estimators we don't compare here our method with GMC.

Section 1 gives definitions of the LWC and two compatible kernels. One based on the Gaussian function and one based on the binomial coefficients. In the following two sections we present mathematical proofs of convergence for the LWC using the two kernels. First the level wise pseudo-binomial kernel (LWB_n) and then the level-wise pseudo-Gaussian kernel LWG. In the experimental results section, we show comparisons with DSS in terms of precision and runtime.

1 Level-Wise Convolution

When dealing with derivatives estimation, one of the most classical methods is to use the finite differences $(f(x+h) - f(x))/h$. Although effective in continuous geometry, it cannot be applied as such to discrete images because derivative values would be limited to integers. A solution is to average each pixel of the image with its neighbour, a process called smoothing. This mathematical operation is known as the convolution product of a function in the integers interval $f : [0, n] \to [0, n]$, the image to be convolved and a function $H : [0, n] \to [0, n]$, the averaging kernel. Gaussian function as a kernel is the standard in this field as described by Lindeberg in the scale space theory [9]. The resulting image can then serve to compute differentials, using finite differences with a convolution by a differential operator Δ as the kernel. Figure 1, is an example of the convolution of a digital function $1, 2, 2, 4, 5$ by a binomial coefficients kernel $1, 2, 1$. To preserve the image scale, each value has to be divided by the mass (or weight) of the kernel (the sum of all its values) in this case $W = 2^n = 4$. After we convolve the smoothed image by a differential operator to obtain the derivatives (first

order). There are three kinds of first order derivative operator, the centered one, as in the example $f(1) - f(-1)$, the backward difference $(f(0) - f(-1))$ and the forward difference $(f(1) - f(0))$. The choice of the derivative operator depends on symmetric properties of the smoothing kernel. For a centered kernel (odd size) we use the centered operator. For an even size kernel we can convolve it with left or right shift (we use the opposite smoothing kernel shift). Out of the range of the discretization there is no information on the function values. The convolution loses precision when those pixels are required. This is known as the border problem and for the example of Figure 1, unknown values are set to 0. In the experimental part we only use functions for which we know those values. The major drawback of this method in the discrete paradigm is its complexity of $O(n.m)$, with n being the size of the image and m the size of the kernel.

Definition 1. *The discrete convolution product (noted $*$) is a transform of two discrete functions $F : \mathbb{Z} \to \mathbb{Z}$ and $H : \mathbb{Z} \to \mathbb{Z}$. At least one is of finite support.*

$$(F * H)(x) = \sum_{i \in \mathbb{Z}} f(x - i).H(i)$$

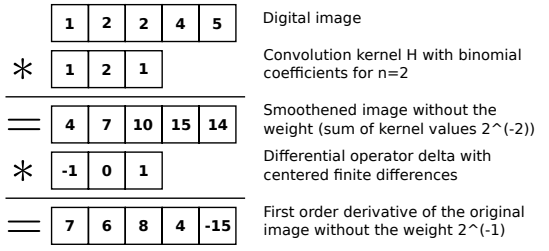
H is said to be a smoothing kernel when $\sum_{i \in \mathbb{Z}} H(i) = 1$.

Definition 1 shows the discrete convolution product of an image f with a kernel H for the pixel x of f.

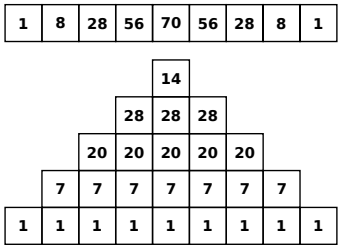
$$(F * H)(n) = \sum_{j=0}^{k} \sum_{i=-k+j}^{k-j} f(n + i).(H(-k + j) - H(-k + j - 1)) \qquad (1)$$

$$(F * H)(n + 1) = (F * H)(n) - H(-k).F(n - k - 1) + H(k).F(n + k) \qquad (2)$$

Looking at the right part of Figure 1 the kernel can be viewed in a multilevel way. The values are the same for the whole level and by adding them all we



(a) Derivative estimation example



(b) Level view of the kernel

Fig. 1. (a). First the detailed process of convolving a digital image with a binomial coefficients kernel, second the convolution of the smoothed image with the central differential operator of central finite differences. (b). Level view of the binomial kernel for $\binom{n}{k}$ with $n = 8$ and k the line index.

obtain the original kernel as shown in the upper right part. The discrete convolution product can be rewritten to apply one level of the kernel at a time, as shown in Equation (1), the convolution of the function f with the kernel H only for the pixel n. The first loop $\sum_{j=0}^{k}$ iterates through all levels and the second one $\sum_{i=-k+j}^{k-j}$ through the whole current level. The complexity has changed to $O(\sum_{i=0}^{\lfloor m/2 \rfloor} 2i+1)$ since $m = 2k+1$, but Equation (1) only concerns one pixel and since each level of the kernel has the same value, we only need to convolve for the first pixel as indicated in Equation (2) and Figure 2.

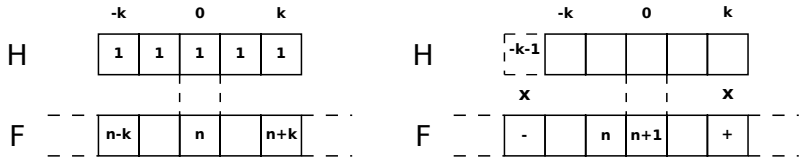


Fig. 2. Convolution of image f with kernel H of size $m = 2k+1$ centered in 0

The left part is the convolution of the pixel n and the right part is the convolution of pixel $n+1$ using the previous result. Since each level of the kernel has the same value Figure 1, we only need to subtract the product of $H(-k)$ and $F(n-k-1)$ and to add the product $H(k)$ and $F(n+k)$ to the convolution of pixel $F(n)$ to get the result.

1.1 Complexity

The resulting complexity depends on the number of levels of the kernel and we only use kernels with log_2 bounded number of levels. We have a $O(2n.log_2(m))$ complexity which is smaller than the binomials convolution of $O(n.m)$ in [11,3] and theoretically slightly better to the latest complexity of the FTT of $O(\frac{39}{4}N.log_2(N))$ in [5,10,6]. This complexity is only for the FFT, and in order to compute a convolution using Fourier transform, several steps are required. The first step, is to apply the transform to the image to be in Fourier space. The second is to multiply the image by the Gaussian kernel. The third is to apply an inverse transform to the result of the previous multiplication to return to the original space. Using other bounds for the number of levels than the logarithm can increase or decrease the complexity and it could be interesting to have a kernel with a fixed number of levels. In higher dimension the complexity will be $O(n^{dim}.log_2(m))$ with dim being the dimension.

1.2 Kernel Compatibility and Extension to Higher Dimensions

For a kernel to be compatible with this method (central difference), it must have an even size to avoid data shift. It has a log_2 bounded number of levels in order to have the complexity described in Subsection 1.1. The convolution should work in higher dimensions using the tensor product of one dimension kernels. The use of n dimension kernels is possible if they are separable in 1 dimension elements.

2 Kernels Presentation

We introduce two discrete kernels of low complexity. They are level-wise versions of two known kernels, the binomial kernel and the Gaussian kernel. There are two methods to generate level-wise kernels. The first one consists in averaging kernel's coefficients per level (pseudo-binomial). The second uses a logarithmic function (pseudo-Gaussian). The fastest discrete converging method for the first three order derivative is the binomial coefficient kernel. And the fastest in terms of computational times is the Gaussian kernel. Kernels are symmetrical: $H(x) = H(-x)$ and they decrease from center to periphery: $H(x) >= H(y) for |x| < |y|$. First let us introduce classical binomial and Gaussian kernels.

Definition 2. *The binomial kernel is a discrete approximation of the Gaussian function. Its coefficients are obtained using the binomials* $\binom{n}{k}$*, with* n *the width of the kernel and* k *its index.* $B_n : \mathbb{Z} \to \mathbb{Z}$.

$$B_n(k) = \binom{2n}{n-k}$$

Let us recall that the classical Gaussian kernel used in scale space is $g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$. We introduce now the smoothing kernel obtained by discretizing this one.

Definition 3. *The Gaussian kernel's coefficients are the ones of the centered Gaussian function.* $G_\lambda \mathbb{Z} \to \mathbb{R}$.

$$G_\lambda(i) = \alpha 2^{-\lambda i^2} \quad where\ \alpha = \sum_{i \in \mathbb{Z}} 2^{-\lambda i^2}$$

The parameter α is chosen to get 1 as total weight. The parameter λ determines the length of the mask.

2.1 Pseudo-gaussian Kernel

To create this kernel, we start from the continuous Gauss formula. We create a rough kernel with its level number bounded by the $\log_2$ function $LWG_{\lambda\gamma}$ Definition 4. It is always centered in 0 with α representing the weight such as the integral of the kernel is equal to 1 in order not to scale the image after convolution. Parameter γ controls the number and length of levels.

Definition 4. *The level-wise pseudo-Gaussian (LWG).* $LWG_{\lambda,\gamma} : \mathbb{Z} \to \mathbb{R}$.

$$LWG_{\lambda,\gamma}(i) = \alpha 2^{-\lambda \gamma^{2 \cdot \lfloor \frac{log_2(|i|)}{log_2 \gamma} \rfloor}}$$

2.2 Building Complexity

Since we can not predict at which value of i the level change will occur, we have to compute the values for the whole kernel size $O(m)$ with m being the size of the kernel. The building process can be speeded up by using the left arithmetic shift to compute powers of 2.

2.3 Pseudo-binomial Kernel

We used the binomial coefficients as a basis for this kernel since it is a good discretization of the Gaussian function G. It is a level-wise kernel and the values of the levels are the sum of the binomial coefficients between two boundaries controlled by the floor function. In Definition 5, we have the pseudo-binomial kernel LWB_n. The integers parameters are: m is the size of the kernel and n the number of levels. Let us denote $s(i)$ is the signature of the i. Figure 3 illustrates the relation between the binomial kernel and BLW. On top B_n is represented by the different levels which size are increasing by power of two. BLW level values are the average of the values in the corresponding level.

Definition 5. *The level-wise pseudo-binomial (LWB_n). $LWB_n : \mathbb{Z} \to \mathbb{Z}$.*

$$LWB_n(0) = \frac{1}{2^{2^{\alpha+1}-2}} \binom{2^{\alpha+1}-2}{2^{\alpha}-1} \quad for \quad n \neq 0$$

$$LWB_n(i) = \frac{1}{2^{2^{\alpha+1}-2+\lfloor log_2(|i|) \rfloor}} \left(\sum_{k=2^{\lfloor log_2(|i|) \rfloor}}^{k=2^{1+\lfloor log_2(|i|) \rfloor}-1} \binom{2^{\alpha+1}-2}{2^{\alpha}-1+s(i)k} \right)$$

$$where \quad \begin{cases} s(i) = +1 & if \quad n > 0 \\ s(i) = -1 & if \quad n < 0 \end{cases} \qquad and \quad m = 2^{\alpha+1} - 2$$

2.4 Building Complexity

To minimize the complexity we use the Pascal triangle building method to compute the binomial coefficients. For the memory management, we use the upper bound $\frac{4^n}{8n\sqrt{\pi n}}$ to allocate our triangle's line.

3 Convergence

In this section, we prove convergence results for discretization of functions $f : \mathbb{R} \to \mathbb{R}$ having a bounded third continuous derivative $f^{(3)}$. At the end, we compare the convergence rate of our method to the ones in literature.

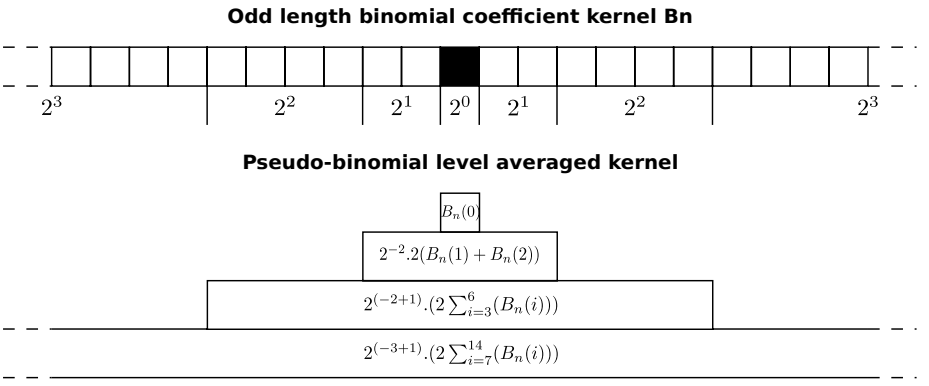


Fig. 3. Top. Odd length binomial kernel B_n with the center marked by the black square. Bottom. Pseudo-binomial kernel LWB_n. The lower line represent power of 2 coefficients determining the level size.

3.1 Convergence and Error Estimation for Unnoisy Images

We prove here a convergence result for discretization by real numbers of functions having a bounded third continuous derivative. Such a result is convenient for implementations using floating numbers. It is valid for all the kernels we mentioned in the previous section.

For a discretization step h, let $\Gamma : \mathbb{Z} \to \mathbb{R}$ be a real discretization of $f : \mathbb{R} \to \mathbb{R}$ and f' the first order derivative of f.

$$h\Gamma(i) = f(ih)$$

It is convenient for implementations using floating numbers. The following theorem is worthwhile for all the kernels we mentioned in the previous section.

Theorem 1. *Let H be any non negative symmetric smoothing kernel and $\Delta * H$ be the associated derivating kernel, where Δ is the central difference operator defined by $\Delta(-1) = \frac{1}{2}$ and $\Delta(+1) = -\frac{1}{2}$ and $\Delta(i) = 0$ for other values of i. Let $x_0 = i_0 h$ with $i_0 \in \mathbb{Z}$. Suppose that $\sum_{j \geq 1} j^2 H(j)$ exists. Suppose moreover that $H(i)$ is decreasing for non negative i.*

(1) (local convergence): $\lim_{h \to 0}((\Delta * H) * \Gamma)(x_0) = f'(x_0)$

(2) (rate of local convergence):

$$((\Delta * H) * \Gamma)(x_0) - f'(x_0) \sim_{h \to 0} \frac{1 + 3\sum_{j \geq 1} j^2 H(j)}{6} f^{(3)}(x_0) h^2$$

(3) (uniform convergence):

$$Sup\,\{|(\Delta * H) * \Gamma)(x) - f'(x)|\,; x \in h\mathbb{Z}\} \leq \frac{1 + 3\sum_{j\geq 1} j^2 H(j)}{6} \|f^{(3)}\|_\infty\ |h|^2$$

Proof. Let $a_i = (\Delta * H)(i)$. We have to evaluate $\dfrac{1}{h}\left(\sum_{i\in\mathbb{Z}} a_i f(x+ih)\right) - f'(x)$

For each i in $\mathbb{Z}$, there are x'_i between $Min\{x, x+(i-1)h\}$ and $Max\{x, x+(i-1)h\}$ such that

$$f(x+ih) = f(x) + ihf'(x) + \frac{(ih)^2}{2} f^{(2)}(x) + \frac{(ih)^3}{6} f^{(3)}(x'_i)$$

Now it is easy to check that $\sum_{i\in\mathbb{Z}} a_i = 0$ and $\sum_{i\in\mathbb{Z}} i a_i = 1$ and $\sum_{i\in\mathbb{Z}} i^2 a_i = 0$ and $\sum_{i\in\mathbb{Z}} i^3 a_i$ exists ; hence we have:

$$\frac{1}{h}\left(\sum_{k\in\mathbb{Z}} a_i f(x+ih)\right) - f'(x) = \frac{h^2}{6}\left(\sum_{i\in\mathbb{Z}} i^3 a_i f^{(3)}(x_i)\right)$$

But now, as the $i^3 a_i$ are non negative, we have $\sum_{i\in\mathbb{Z}} |i^3 a_i| = \sum_{i\in\mathbb{Z}} i^3 a_i = 2\sum_{i\geq 1} i^3 a_i = \sum_{i\geq 1} i^3 (H(i-1) - H(i+1)) = 1 + 3\sum_{j\geq 1} j^2 H(j)$ and the two first results are coming immediately when h tends to 0.

Noticing that $\left(\sum_{i\in\mathbb{Z}} i^3 a_i f^{(3)}(x_i)\right) \leq \|f^{(3)}\|_\infty \sum_{i\in\mathbb{Z}} |i^3 a_i|$, the reader would immediately consider that the proof is complete.

3.2 Error Estimation for Noisy Images

We consider here the following model for noisy images: let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be the real image; suppose that the sequence $\Gamma : \mathbb{Z} \longrightarrow \mathbb{Z}$ is a noisy discretization with a uniformly bounded error $|h\Gamma(i) - f(hi)| \leq Kh^\alpha$, where $\alpha \in]\frac{1}{2}, 1]$, $K \in R_+^*$ and $h \in R_+^*$.

Theorem 2. *Suppose that f is a C^3 function and $f^{(3)}$ is bounded. If $m = \left\lfloor h^{2(\alpha-3)/5} \right\rfloor$, then we have $|(LWB_{2m} * \Gamma)(n) - f'(nh)| \in O(h^{(4\alpha-2)/5})$ for sufficiently large h.*

Proof. We prove the following inequality in a standard way:

$$|LWB_{2m} * \Gamma(n) - f'(nh)| \leq \frac{2 + 3m^2}{6} h^2 \|\phi^{(3)}\|_\infty + \frac{2Kh^{\alpha-1}}{4^m}\binom{2m}{m}$$

From the well known Stirling's formula, $n \sim \sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n$ we get for $m \to +\infty$

$$\frac{1}{4^m}\binom{2m}{m} \sim \frac{1}{4^m}\frac{\sqrt{4\pi m}\left(\frac{2m}{\mathrm{e}}\right)^{2m}}{\left(\sqrt{2\pi m}\left(\frac{m}{\mathrm{e}}\right)^{m}\right)^2} = \frac{\sqrt{4\pi m}}{2\pi m} = \frac{1}{\sqrt{\pi m}}$$

Hence choosing $m = \lfloor h^{\frac{2(\alpha-3)}{3}} \rfloor$ provides the result.

Notice that this result is significantly worse than the corresponding result for Binomial smoothing kernel (non level-wise) which has a convergence proof for C2 functions. However this is a preliminary result that could probably be strengthen. It seems possible to generalize the results to C2 functions in the case of the pseudo-binomial kernel. Otherwise, we fail to prove such a result for pseudo-Gaussian (even non level-wise) smoothing kernel. Ulterior work may show counter examples for the pseudo-Gaussian kernel in terms of convergence or convergence rate for C2 functions.

3.3 State of the Art

Maximal error of differential estimation mainly depends on the image. This is the reason why we include the image in our convergence rate in Figure 4. We can not bound the error for an arbitrary family of functions since our upper bound depends on the norm of the derivatives, but we can bound the error for a set of functions. Trigonometrical functions have the following bound $||f'''(x)||_\infty = 1$. For polynomial functions, the bound will depend on the degree of the function.

BC	DSS	Taylor P.	LWC
1/3	2/3	1/2	$\lvert f'''(x)\rvert.\frac{h^2}{6}\sum(i^3 ai)$

Fig. 4. Convergence rates of different estimators for the first order derivative. Datas were taken from [12].

4 Experiments

We have created level-wise version of two kernel families. The Gaussian kernel used in the scale space theory (SCT), and the binomial kernel used in discrete geometry. With level-wise convolution we have reduced the convolution complexity. Figure 9 and Figure 6 show experiment results of computational time verifying theoretical complexity. Figure 7 and Figure 8 show experimental convergence rates for the two families of kernel on a ball and on an ellipsis.

The tests of the two kernels have been done on 2D digitized images generated by the DGtal library [1]. The convolution is a 1D kernel applied on the contour of the images, so computational time does not suffer from the 1D tests presented in [4]. Since our convergence proofs implies that objects are $C3$, we choose the ones that have this property. The y axis is the Euclidian norm of the

Discretization step	0.1	0.01	0.001
Pseudo-Gaussian γ	5	20	80
Pseudo-Gaussian λ	1	1	1
Pseudo-binomial α	4	8	12

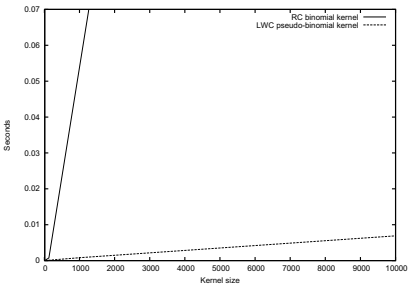
Fig. 5. Parameters of kernels families used in the experimental part

difference between the expected value and the estimators value. The x axis is the logarithm of the inverse of the discretization step. The LWG used in this experiments LWG_exp Definition 6 is a derivated form of the one presented in Definition 4.

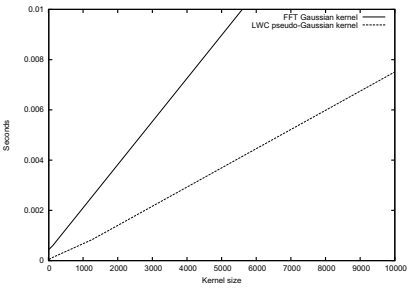
Definition 6. *Experiments version of the LWG kernel,* $LWG\_exp_{\lambda,\gamma} : \mathbb{Z} \to \mathbb{R}$. $\log_2 e$ *is a result from the change of* e *in* 2.

$$LWG\_exp_{\lambda,\gamma} = \alpha 2^{-\lambda log_2 e . \gamma^{2 . \lfloor \frac{log_2(|i|)}{log_2 \gamma} \rfloor}}$$

Kernels parameters were chosen empirically, but they are the same for each function as shown in Figure 5.



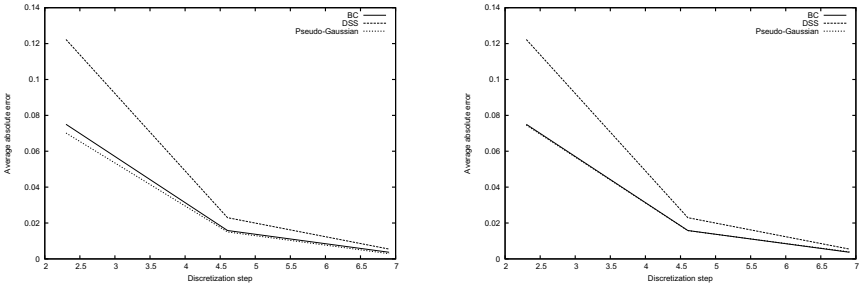
(a) BC vs LWC pseudo-binomial



(b) FFT vs LWC pseudo-Gaussian

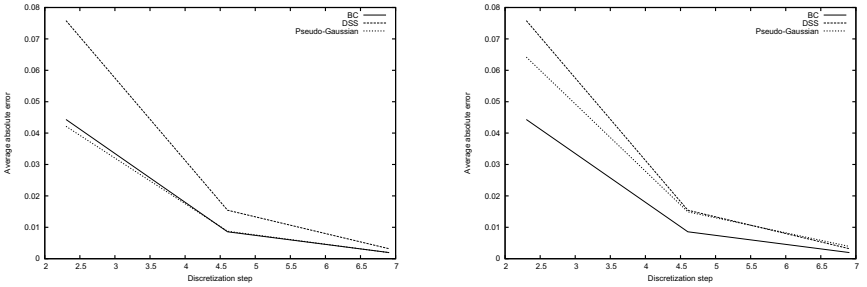
Fig. 6. Computational time comparison of convolution of an image of size 1000 by kernels of the same size. (a). Binomial kernel using regular convolution and pseudo-binomial using LWC. (b). Gaussian kernel using FFT and pseudo-Gaussian kernel using LWC.

The computational time results of Figure 6 were obtained with a digitized ball of radius 1. Both methods are not optimized for computational time result. DSS is implemented using integers and our method uses floating-point numbers. The difference is significant enough, and this regardless of implementations to conclude that our method is faster.



(a) Pseudo-binomial kernel family (b) Pseudo-Gaussian kernel family

Fig. 7. Experimental convergence rates on a digitized ball of radius 1. (a). Comparison results for the pseudo-binomial kernel family. (b). Comparison results for the pseudo-Gaussian kernel family



(a) Pseudo-binomial kernel family (b) Pseudo-Gaussian kernel family

Fig. 8. Experimental convergence rates on a digitized ellipsis of large radius 1. (a). Comparison results for the pseudo-binomial kernel family. (b). Comparison results for the pseudo-Gaussian kernel family.

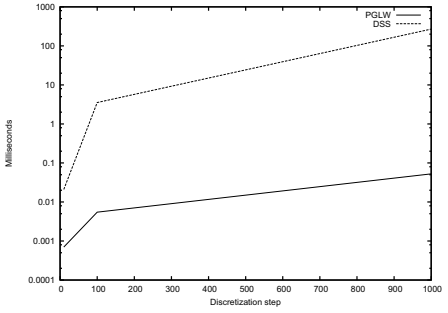


Fig. 9. Computational times in milliseconds on a logarithmic scale as a function of the resolution

5 Conclusion

We have presented a uniform convergence proof for LWC for two compatible kernel families. We have experimentally confirmed convergence as well. The experimental part shows the precision even with empiric parameters on selected differentiable objects. Also LWC is significantly faster than DSS (as implemented in the DGtal library). It is important to note that both implementations of DSS and LWC that we used may not be optimized. In our future work we will focus on parameters selection and the adaptivity of our kernel using multi-pass convolutions for higher order differentials. Noise resistance proof for pseudo-Gaussian kernels will be further investigated. We will also do some tests in higher dimension and higher derivative order. The last goal will be the GPU implementation allowing the multi-pass approach to improve the runtime.

Acknowledgement. The research leading to these results has received funding from the KIDICO project of the French *Agence Nationale de la Recherche* (Grant Agreement ANR-2010-BLAN-0205-02).

References

1. DGtal: Digital geometry tools and algorithms library, `http://liris.cnrs.fr/dgtal`
2. De Vieilleville, F., Lachaud, J.O.: Comparison and improvement of tangent estimators on digital curves. Pattern Recognition 42(8), 1693–1707 (2009)
3. Esbelin, H., Malgouyres, R., Cartade, C.: Convergence of binomial-based derivative estimation for 2 noisy discretized curves. Theoretical Computer Science 412(36), 4805 (2011)
4. Gonzalez, D., Malgouyres, R., Esbelin, H.A., Samir, C.: Fast Level-Wise Convolution. In: Barneva, R.P., Brimkov, V.E., Aggarwal, J.K. (eds.) IWCIA 2012. LNCS, vol. 7655, pp. 223–233. Springer, Heidelberg (2012), `http://dx.doi.org/10.1007/978-3-642-34732-0_17`
5. Heideman, M., Johnson, D., Burrus, C.: Gauss and the history of the fast fourier transform. IEEE ASSP Magazine 1(4), 14–21 (1984)
6. Johnson, S., Frigo, M.: A modified split-radix fft with fewer arithmetic operations. IEEE Transactions on Signal Processing 55(1), 111–119 (2007)
7. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision 1(4), 321–331 (1988)
8. Lachaud, J., Vialard, A., de Vieilleville, F.: Fast, accurate and convergent tangent estimation on digital contours. Image and Vision Computing 25(10), 1572–1587 (2007)
9. Lindeberg, T.: Scale-space theory in computer vision. Springer (1994)
10. Lundy, T., Van Buskirk, J.: A new matrix approach to real ffts and convolutions of length 2 k. Computing 80(1), 23–45 (2007)
11. Malgouyres, R., Brunet, F., Fourey, S.: Binomial Convolutions and Derivatives Estimation from Noisy Discretizations. In: Coeurjolly, D., Sivignon, I., Tougne, L., Dupont, F. (eds.) DGCI 2008. LNCS, vol. 4992, pp. 370–379. Springer, Heidelberg (2008)
12. Provot, L., Gérard, Y.: Estimation of the Derivatives of a Digital Function with a Convergent Bounded Error. In: Debled-Rennesson, I., Domenjoud, E., Kerautret, B., Even, P. (eds.) DGCI 2011. LNCS, vol. 6607, pp. 284–295. Springer, Heidelberg (2011)