An application-case for derivative learning: optimization in colour image filtering

Samuel Morillas,
Almanzor Sapena,
and José A. Conejero
Instituto Universitario de Matemática Pura y Aplicada
Universidad Politécnica de Valencia
Valencia, Spain
Email: smorillas;alsapie;aconejero@mat.upv.es

José Camacho
Departamento de Teoría de la Señal, Telemática y Comunicaciones
Universidad de Granada
Granada, Spain
Email: josecamacho@ugr.es

***Abstract*—Related to the notion of derivative of a function, its application to function optimization is an interesting and illustrative problem for Engineering students. In the present work, we develop an application of the derivative concept to optimize the filtering of a colour image. This implies to optimize the value of the filter parameter to maximize performance. We propose to maximize the quality of the filtered image represented by the *Peak Signal to Noise Ratio* (PSNR), which is a function of the filter parameter. The optimal value for the parameter is obtained by means of an algorithm based on the approximation of the derivative of the PSNR function so that finally the optimum filtered image is obtained.**

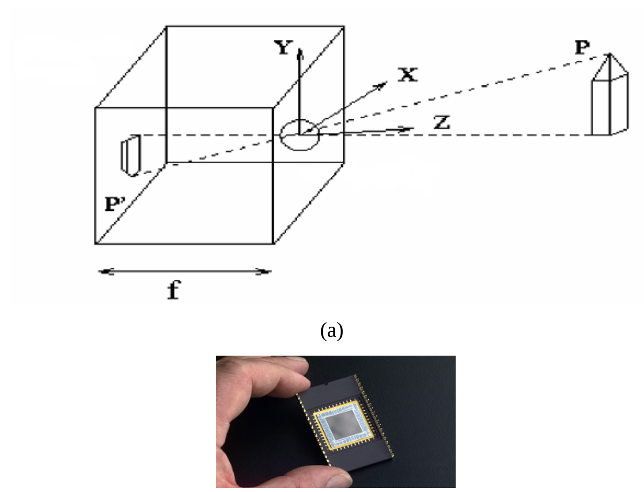
I. Introduction

Applications of calculus of maxima of one variable functions, presented to Engineering students, usually consists on finding the zeroes of a derivative which is expressed in terms of elementary functions. In some cases the numerical point of view of the calculus of this derivative is also considered. However, it is quite difficult to show a real and interesting application where the derivative plays a crucial role. In this paper we will see that the problem of optimizing an image by using colour image filtering can be reduced to the calculus of the maximum of a certain function called the *Peak Signal to Noise Ratio*.

In Section II we will show some preliminaries about image acquisition and representation. Some basic notions about image filtering are also presented. The utility of fuzzy logic is introduced for filtering design. Section III describes the optimization algorithm based on the approximation of derivatives. Section IV shows simulation results, and conclusions are provided in Section V.

II. Preliminaries

An image is a $2D$ representation of the objects in a $3D$ scene which is obtained by projection of the $3D$ objects on an image plane. In digital cameras, this projection is obtained according to the pinhole camera model [1]-[3] (see Figure 1(a)): the objects, represented in a $3D$ space, are projected through a center of projection on the $2D$ image plane. The center of projection corresponds to the camera optics and the image plane is physically represented as a Charge Coupled Device (CCD) sensor which captures the input radiation. The CCD sensor (see Figure 1(b)) comprises an array of $M \times N$ single light sensors. This implies that the acquired digital image is also an array of $M \times N$ single elements called *pixels* (from picture elements). In digital colour images, each pixel represents a single colour of the image. Colours are commonly represented in computers using the Red-Green-Blue (RGB) colour space. This colour space follows an additive model so that any colour is obtained by appropriately mixing the three primaries: Red, Green and Blue. Thus, each colour image pixel is associated with a tern of RGB values which represents the appropriate quantity of each primary colour that should be mixed to obtain the colour stored in that pixel.



(a)

(b)

Fig. 1. (a) Scheme of the pinhole camera model: The object P, represented in the (X, Y, Z) space, is projected through the center of projection (camera optics) on the image plane (CCD) as the object P'. (b) A CCD sensor.

Digital image acquisition process can be affected by many different factors able to degrade the quality of the image. For instance, deficient illumination conditions or CCD sensor malfunctions may introduce irregularities in the image also known as (Gaussian) *noise*. Other factors able to affect the

(a) (b)

Fig. 2. (a) Ideal noise-free image, (b) image corrupted with (Gaussian) noise.

image quality are, among others, transmission errors or storage faults. Figure 2 shows an ideal noise-free image and the same image contaminated with noise.

The presence of these irregularities, or noise, is not desired mainly for two reasons: (i) the perceptual image quality is lower, which is critical from the user standpoint, and (ii) the presence of noise is an important drawback for many tools of computer image analysis. Therefore, many techniques to reduce image noise have been developed in the recent years [2], [3]. Classical techniques to approach this problem are based on a linear approach. The Arithmetic Mean Filter (AMF) and the Gaussian Filter (GF) use an average operation among each pixel and its neighbours pixels [2], [3]. These methods are able to reduce image noise but they blur edges and details in the image too much. To solve this, a series of nonlinear methods were developed [2], [3],[4]-[8]. In general, these techniques are based on detecting image edges and details and smooth them less than the rest of image regions. In particular, the techniques in [5]-[8] propose to average each image pixel with only its neighbour pixels which are similar to it. Since it is difficult to differentiate between similar and non-similar pixels in a crisp way, it is more appropriate to assign a degree of similarity. The Fuzzy Bilateral Filter (FBF) [8] that we use in this work, employs fuzzy logic to assign this degree of similarity. According to fuzzy logic, the degree of similarity between two pixels is a value in $[0, 1]$ that represents in which degree two pixels are similar, where 1 means total similarity (equality) and 0 total dissimilarity. Since Lofti A. Zadeh introduced the theory of fuzzy sets in 1965, it has become a well-known area of study in the last decades. Fuzzy logic constitutes a generalization of the classical set theory which represents a gradual transition between the classical notions of outside and inside of a set. Fuzzy logic has been successfully employed in many engineering problems and many related research topics as, for instance, fuzzy topology and fuzzy metrics [9]-[13], are still active.

In most adaptive nonlinear filters, and also in the case of the FBF [8] that we use here, the smoothing capability can be tuned by modifying the values of their parameters. Thus, appropriate tuning is needed to achieve optimal performance. In this paper, we illustrate how derivatives can be used to find an appropriate setting of the λ parameter in the FBF filter. For a detail descrption of this filter, we refer the interested reader to [8]. For the purpose of this work, we just focus on the fact that the performance of the filtering process (output image quality) can be seen as a function of λ. Therefore, we propose to optimize the image quality measure represented by the *Peak Signal to Noise Ratio* (PSNR), which is a function of this parameter. The optimal value for the parameter is obtained by means of an algorithm based on the approximation of the derivative of the PSNR function so that finally the optimum filtered image is obtained.

III. Optimization Algorithm Description

The quality of a filtered image can be measured with several functions. In this case we will use the *Peak Signal to Noise Ratio* (PSNR) which, since the quality of the filtered image depends on λ, we define as a function of λ as follows:

$$PSNR(\lambda) = 20 \log \left(\frac{255}{\sqrt{\frac{1}{NMQ} \sum_{i \in \mathbf{F}} ||\mathbf{F}^o_{\mathbf{i}} - \mathbf{F}^\lambda_{\mathbf{i}}||^2}} \right) \quad (1)$$

where M, N are the image dimensions, $||\cdot||^2$ denotes the square of the Euclidean norm, $\mathbf{F}^o$ is the original noise-free image, and $\mathbf{F}^\lambda$ is the filtered image obtained after applying the FBF, respectively.

As mentioned above, we seek for the value of λ that achieve the optimum PSNR performance. To find it, we apply the following algorithm: First, we take an initial λ_0 for λ as a rough approximation to the optimum. For the current approximation to the optimum, λ_n, we approximate the derivative of $PSNR(\lambda_n)$ as

$$D(\lambda_n) = \frac{PSNR(\lambda_n + \delta) - PSNR(\lambda_n - \delta)}{2 \cdot \delta}, \quad (2)$$

where, $\delta > 0$. Now, we check whether the actual approximation to the optimum is already good enough: if $|D(\lambda_n)| < \epsilon > 0$, then the derivative at λ_n is small enough to conclude that it is very close to the optimum, and the method stops. Otherwise, we find the next approximation to the optimum λ_{n+1} from the previous λ_n as

$$\lambda_{n+1} = \lambda_n + \alpha D(\lambda_n), \quad (3)$$

where $\alpha > 0$ is called the learning parameter whose importance will be commented in Section 3, and the above procedure is repeated. Notice that, according to Eq. (5), the sign of $D(\lambda_n)$ indicates the direction in which the optimum is located and the *speed* in which the algorithm advances towards it is proportional to $|D(\lambda_n)|$.

IV. Simulation Examples

To illustrate the use of the derivative in the identification of the optimum λ for the FBF, the corrupted image of Lenna in Figure 2 (b) will be used. Figure 3 presents the PSNR computed for the image and the complete range of λ values.

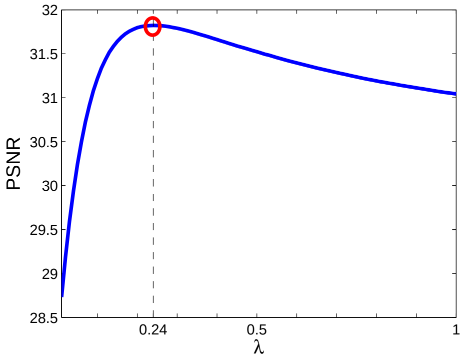


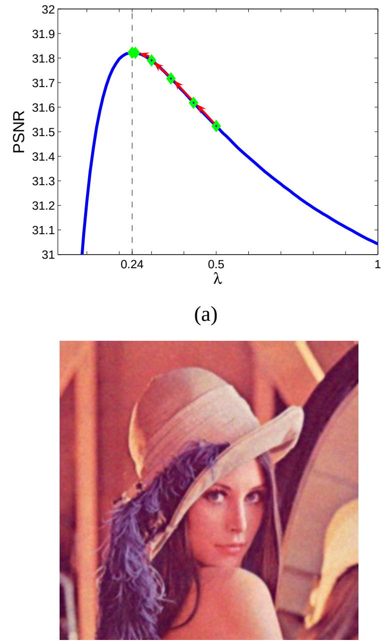
Fig. 3. PSNR curve for the complete range of λ values in the corrupted image of Lenna in Figure 2 (b).

This figure shows that the optimum filtering performance is attained for $\lambda = 0.24$. To compute this PSNR curve, it was necessary to filter the Lenna image one hundred times - from $\lambda = 0.01$ to $\lambda = 1$ in steps of 0.01. This is a valid but computational overwhelming method to obtain the optimum parameter. Furthermore, we only obtain the optimum with a certain accuracy which depends on the step used. To improve the accuracy we need to reduce the step, which in turn means to increase the number of times the image needs to be filtered. For instance, to reduce the accuracy in one order, it is necessary to increase by 10 the number of times the complete image is filtered.

An alternative to the extensive computation used to obtain Figure 3 is the derivative-based -or gradient-based- optimization treated in this paper. Let us illustrate the use of this method with an example. The optimization will be started from the central value $\lambda = 0.5$. The optimization parameters are $\delta = 0.01$, $\epsilon = 0.1$ and $\alpha = 0.05$. The evolution of the optimization is depicted in Figure 4(a). We arrive to the optimum filtering the image only 12 times - the number of approximations of the derivative (6) times the 2 applications of the filter per approximation (2 according to Eq. (2)). Furthermore, by reducing ϵ, the accuracy of the optimum can be improved with only a few more runs. The optimum filtered image attained is shown in Figure 4(b).

The convenient choice of the optimization parameters is of paramount importance to ensure convergence to the optimum. We have already commented that ϵ controls the accuracy achieved in the identification of the optimum. In this paper, we are considering the noise-free optimization case, so that we assume that the function to optimize -the PSNR curve- is free of measurement noise. If this is not the case, ϵ should be high enough so that it exceeds the noise level in the derivative estimation.

The three parameters, δ, α and ϵ, have interrelated effects. For the sake of easy understanding, the influence of the choice of δ and α will be discussed separately, varying the parameters one at a time while maintaining the other fixed. Let us run the optimization for δ values equally distanced in the logarithmic scale over [0.001, 1] and for fixed values of



(a)

(b) PSNR = 31.80

Fig. 4. Optimization for initial point $\lambda = 0.5$ and $\delta = 0.01$, $\alpha = 0.05$ and $\epsilon = 0.1$: a) optimization steps, b) optimum filtered image.

$\alpha = 0.05$ and $\epsilon = 0.1$. The results are presented in Table I. The lower the value of δ, the better the derivative estimation. This is again assuming the PSNR function to optimize is free of measurement noise - it is well known that derivative estimations are highly affected by noise. Since the derivative is more accurately estimated for low δ values, the optimization arrives to higher values of PSNR. Also, we can see that the value of δ is related to the number of points or λ values in the optimization where the derivative is estimated. The higher the value of δ, the lower the number of points. Notice that if δ is too high, the optimization algorithm may not converge to the optimum. The changes observed in the optimization method due to the use of different values of δ can be easily explained in simple geometric terms. If the current value of λ is far from the optimum, high δ values result in derivative estimations of high magnitude. Therefore, the optimization is speeded up and the number of points evaluated are reduced. If otherwise λ is close to the optimum, the optimum may be included in the interval $[\lambda - \delta, \lambda + \delta]$ where the derivative is estimated. If this is the case, the magnitude of the derivative is underestimated. The higher the value of δ, the higher the interval where this occurs and the further to the optimum we may converge.

Let us run the optimization for α values equally distanced in the logarithmic scale over [0.005, 5] and for fixed values of $\delta = 0.01$ and $\epsilon = 0.1$. The results are presented in Table II. We can see that the value of parameter α is especially important for the optimization. The optimization may easily diverge if the parameter is made too high. Nonetheless, for very low values of α, the number of points where the derivative has to be estimated is much increased. For instance, for $\alpha =$

δ	# points	Attained PSNR
$1 \cdot 10^{-3}$	8	31.822
$3.26 \cdot 10^{-3}$	6	31.822
$1 \cdot 10^{-2}$	6	31.822
$3.26 \cdot 10^{-2}$	6	31.822
$1 \cdot 10^{-1}$	5	31.815
$3.26 \cdot 10^{-1}$	4	31.634
1	∞	Divergence

TABLE I
RESULTS FOR DIFFERENT VALUES OF δ AND $\alpha = 0.05$ AND $\epsilon = 0.1$ WITH STARTING POINT $\lambda = 0.5$.

α	# points	Attained PSNR
$5 \cdot 10^{-3}$	55	31.822
$1.58 \cdot 10^{-2}$	18	31.822
$5 \cdot 10^{-2}$	6	31.822
$1.58 \cdot 10^{-1}$	29	31.822
$5 \cdot 10^{-1}$	∞	Divergence
1.58	∞	Divergence
5	∞	Divergence

TABLE II
RESULTS FOR DIFFERENT VALUES OF α AND $\delta = 0.01$ AND $\epsilon = 0.1$ WITH STARTING POINT $\lambda = 0.5$.

$5 \cdot 10^{-3}$, 55 derivative estimations were needed to arrive to the optimum. This means that the Lenna image had to be filtered 110 times, 10 times more than for computing Figure 3. The best α value, so that the algorithm converge to the optimum in the lowest number of points, depends very much on the initial point - initial λ value. As a general rule, α should be set high enough for a fast exploration while not too high to avoid divergence.

V. CONCLUSION

The use of real applications of Mathematics motivates the learning process of students in several ways. In particular, most mathematical notions can be explained by use of different models. In this work we have presented a simulation model which will allow the students to relate the notion of derivative of a function and one of its applications which is optimization problems. By the simulation process described in this paper, the students will approximate the optimal value of the parameter in a function and, moreover, they will really see how appropriate is the result of this algorithm since the visualization of the output images by using a filter technique will allow the students to search for better results. This application can be used in a first semester of Calculus either as a guided practice or as an accessible research project for advanced students.

ACKNOWLEDGMENT

The authors acknowledge the support of Universidad Politécnica de Valencia under grant Primeros Proyectos de Investigación 2008/3202.

REFERENCES

[1] *http://en.wikipedia.org/wiki/Pinhole_camera_model*
[2] R.C. González, R.E. Woods, Digital Image Processing, Prentice Hall, 2002.
[3] K.N. Plataniotis, A.N. Venetsanopoulos, Color Image processing and applications, Springer-Verlag, 2000.
[4] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 5 (1990) 629-639.
[5] C. Tomasi, R. Manduchi, Bilateral filter for gray and color images, Proc. IEEE International Conference Computer Vision, 1998, 839-846.
[6] M. Elad, On the origin of bilateral filter and ways to improve, IEEE Transactions on Image Processing 11 10 (2002) 1141-1151.
[7] R. Garnett, T. Huegerich, C. Chui, W. He, A universal noise removal algorithm with an impulse detector, IEEE Transactions on Image Processing 14 11 (2005) 1747-1754.
[8] S. Morillas, V. Gregori, A. Sapena, Fuzzy Bilateral Filtering for Color Images, ICIAR'06, Lecture Notes in Computer Science 4141 (2006) 138-145.
[9] A. George, P. Veeramani, On Some results in fuzzy metric spaces, Fuzzy Sets and Systems 64 3 (1994) 395-399.
[10] A. George, P. Veeramani, Some theorems in fuzzy metric spaces, J. Fuzzy Math. 3 (1995) 933-940.
[11] V. Gregori, S. Romaguera, Some properties of fuzzy metric spaces, Fuzzy Sets and Systems 115 3 (2000) 477-483.
[12] V. Gregori, S. Romaguera, Characterizing completable fuzzy metric spaces, Fuzzy Sets and Systems 144 3 (2004) 411-420.
[13] A. Sapena, A contribution to the study of fuzzy metric spaces, Appl. Gen. Topology 2 1(2001) 63-76.