



Repositorio Institucional de la Universidad Autónoma de Madrid

<https://repositorio.uam.es>

Esta es la **versión de autor** de la comunicación de congreso publicada en:
This is an **author produced version** of a paper published in:

Computer Vision and Image Understanding 115.11 (2011): 1536-1551

DOI: <http://dx.doi.org/10.1016/j.cviu.2011.07.005>

Copyright: © 2011 Elsevier B.V. All rights reserved

El acceso a la versión del editor puede requerir la suscripción del recurso
Access to the published version may require subscription

Edge-Preserving Color Image Denoising Through Tensor Voting[☆]

Rodrigo Moreno^{a,*}, Miguel Angel Garcia^c, Domenec Puig^b, Carme Julià^b

^a*Center for Medical Image Science and Visualization and Department of Medical and Health Sciences, Linköping University, Campus US, SE-581 85, Linköping, Sweden*

^b*Intelligent Robotics and Computer Vision Group at the Department of Computer Science and Mathematics, Rovira i Virgili University, Av. Països Catalans 26, 43007 Tarragona, Spain*

^c*Department of Informatics Engineering, Autonomous University of Madrid, Francisco Tomas y Valiente 11, 28049 Madrid, Spain*

Abstract

This paper presents a new method for edge-preserving color image denoising based on the tensor voting framework, a robust perceptual grouping technique used to extract salient information from noisy data. The tensor voting framework is adapted to encode color information through tensors in order to propagate them in a neighborhood by using a specific voting process. This voting process is specifically designed for edge-preserving color image denoising by taking into account perceptual color differences, region uniformity and edginess according to a set of intuitive perceptual criteria. Perceptual color differences are estimated by means of an optimized version of the CIEDE2000 formula, while uniformity and edginess are estimated by means of saliency maps obtained from the tensor voting process. Measurements of removed noise, edge preservation and undesirable introduced artifacts, additionally to visual inspection, show that the proposed method has a better performance than the state-of-the-art image denoising algorithms for images contaminated with CCD camera noise.

Key words: Image denoising, edge preservation, perceptual grouping, tensor voting, CIELAB, CIEDE2000

[☆]This paper is an extended version of the work “On Adapting the Tensor Voting Framework to Robust Color Image Denoising” presented in the 13th International Conference on Computer Analysis of Images and Patterns, September 2009.

*Corresponding author: Tel.: +46-13-28 6765 Fax: +46-13-10 1902

Email addresses: rodrigo.moreno@liu.se (Rodrigo Moreno), miguelangel.garcia@uam.es (Miguel Angel Garcia), domenec.puig@urv.cat (Domenec Puig), carne.julia@urv.cat (Carme Julià)

1. Introduction

Color image denoising is an important task in computer vision and image processing, as images acquired through color image sensors are usually contaminated by noise. Color image denoising algorithms can be directly used for image restoration and other higher-level tasks as a pre-processing step. The main goal of color image denoising is to suppress noise from color images while preserving their features, such as meaningful edges or texture details, as much as possible. A color image denoising algorithm is called *edge-preserving* when it is able to accomplish this goal. Liu *et al.* [1] have identified the following general features that an effective, edge-preserving color image denoising algorithm must fulfill: noise must be completely removed from flat regions; edges, texture details and global contrast must be preserved; and no artifacts must appear in the result.

Designing effective, edge-preserving color image denoising algorithms is a difficult task that can be evidenced by the fact that the majority of denoising algorithms introduce undesirable blurring and/or artifacts in the filtered images. The main reason for this difficulty is that, without any other assumptions, no color image denoising algorithm can utterly comply with all the aforementioned features listed in [1]. This is mainly due to two reasons: the complete reconstruction of the original image from one contaminated by noise is not possible in general, and some of those features are nearly contradictory. For example, distinguishing between noise and texture is an open problem.

Two main approaches have been followed in color image denoising: spatial domain and transform-domain filtering. The first approach filters the input image by using the color information of every pixel and its neighbors. The major problem of these filters is their tendency to blur the images. The second approach transforms the input image to a different space, typically to the wavelet domain, filters the transformed image and applies the inverse transformation to the result. Despite its good edge preservation properties, the major criticism to transform-based denoising algorithms is the introduction of undesirable artifacts. Section 2 presents a brief review of both approaches.

In the last years, effective approaches based on perceptual grouping have been proposed in the image processing field, such as image segmentation and edge detection (e.g. [2–4]). Perceptual grouping is defined as the ability of the human visual system both to extract significant relations from input data without any previous knowledge of the content and to group these data into

meaningful higher level structures, even in presence of missing or noisy data ([5, 6]). Among the techniques based on perceptual grouping, the tensor voting framework (TVF) appears to be one of the most appropriate for edge-preserving color image denoising, since it was designed as a generic framework that can be adapted to a variety of applications well beyond the ones which it was originally applied to. The TVF was proposed more than a decade ago by Guy and Medioni ([2, 7]) as a robust technique inspired in perceptual grouping for extracting salient information from noisy spatial data. Their approach is able to recover the shape of surfaces, edges and junctions present in a set of points in N-dimensional Euclidean spaces, in particular, in 2D and 3D. This method has also been found appropriate for extracting salient information in other contexts, such as for epipolar geometry estimation [8], denoising of random dot patterns [9], and video analysis (e.g., [10, 11]). Recently, we have proposed an efficient scheme for reducing the complexity of tensor voting to $O(1)$ [12].

The performance of the TVF strongly depends on two critical processes: the proper definition of the information encoding process and the voting process. On the one hand, in [7], tensors encode the most likely directions of surface normals at every given 2D or 3D point. This allows the method to solve the surface reconstruction problem, that is, to extract surfaces, edges and junctions from a set of noisy points. However, when the aim is to extract information not related to surfaces, edges or junctions, the input information must be either modeled in terms of the surface reconstruction problem or encoded into tensors through a different encoding process. This second alternative is likely to be more advantageous as the new encoding process can be specifically tailored to the problem requirements. On the other hand, the canonical voting fields proposed in [7] to propagate the encoded information were designed to estimate surface likeliness based on the hypothesis that normal vectors tend to have smooth changes on surfaces. These voting fields designed for surface reconstruction should not be directly used for other applications before assessing whether they are appropriate or not, since the assumptions on which they are based may no longer be valid in a context not related to surface reconstruction. This is the case of image denoising where, even in ideal conditions (i.e., without noise), color can change abruptly. This suggests that the use of the canonical voting fields may not be the best option in this scope.

This paper proposes a new solution to the problem of edge-preserving color image denoising based on an adaptation of the TVF in order to properly handle color information. First, an encoding process specifically designed to encode color, uniformity and *edginess* into tensors

is presented. Second, a voting process specifically tailored to the edge-preserving color image denoising problem is also introduced. This voting process is based on the nature of the encoded information and on a set of criteria inspired by the perceptual process of image denoising.

This paper is organized as follows. Section 2 describes previous related work. Section 3 presents the criteria taken into account in the design of the algorithm proposed in this paper. Sections 4 and 5 detail the adaptation of the TVF to edge-preserving color image denoising. Section 7 shows a comparative analysis of the proposed method against some of the state-of-the-art, edge-preserving, color image denoising algorithms by using the quality metrics described in section 6. Finally, section 8 discusses the obtained results and makes some final remarks.

2. Previous Related Work

As mentioned above, two main color image denoising approaches have been followed: spatial domain filtering and transform-domain filtering. Classical filters, such as mean, median or Gaussian filters [13], bilateral filtering [14], non local means [15], anisotropic diffusion [16] and Bayesian inference [17], among many others, follow the spatial domain filtering approach. Classical filters are simple, efficient and easy to implement. However, they frequently blur the filtered images and/or eliminate important details. The bilateral filter extends the concept of Gaussian filtering by adding a Gaussian weighting function that depends on the difference between pixel intensities. This filter is also efficient and easy to implement. However, it is unable to filter very noisy images. Non-local means (NLM) extends bilateral filtering by taking into account differences between pixel neighborhoods instead of pixel intensities. NLM is effective for image denoising and it is considered to belong to the state-of-the-art. However, it tends to generate undesirable quantization effects in edgeless regions. Filters based on anisotropic diffusion give more weight to neighbors located in the directions where edges are not present. Anisotropic diffusion usually models the filtering problem by means of partial differential equations (PDEs) (e.g. [16, 18]), although the use of graph theory has also been proposed [19]. Anisotropic diffusion has been a successful approach, with many methods based on it belonging to the state-of-the-art (e.g., [18, 19]). Techniques based on anisotropic diffusion are able to suppress noise effectively. However, they also tend to create artifacts at edges and have problems with very noisy images. Bayesian-based approaches are usually highly time consuming and face similar problems to anisotropic diffusion (e.g., [17, 20]). A different successful strategy that follows

the spatial domain filtering approach uses conditional random fields to detect and remove noise [1]. However, its main drawback is that it is highly time consuming. A different spatial domain approach applies evolutionary computation [21]. However, its scope of use is limited, since it requires a training stage.

The most popular technique within the transform-domain filtering approach is based on wavelets [22]. Basically, small coefficients of the wavelet transform of the input image are removed before applying the inverse transformation, since they are usually due to noise. Many adaptations of this principle have been proposed in the literature. For example, Gaussian scale mixtures [23], hidden Markov models [24] or optimal color space projection [25]. In spite of their good edge preservation properties—some of these methods are considered to belong to the state-of-the-art—the major criticism to wavelet-based denoising algorithms is the introduction of undesirable artifacts in the images. Other approaches that filter images in a transform-domain include Wiener filters [13], low pass filters using the Fast Fourier Transform [13] or methods based on *blind image separation*, which tries to separate two original signals (noise and signal in image denoising) from their addition (e.g., [26]). However, these approaches have been outperformed by other strategies. More recently, Yu *et al.* [27] intended to take advantage of both transform-domain and spatial domain approaches for image denoising. However, they found that their method, which is based on wavelet-based filtering and the bilateral filter, is not satisfactory to deal with real noise.

Perceptual grouping has previously been applied to color image denoising, especially in the spatial domain. For example, Ben-Shahar and Zucker [28] detect and remove color noise by using the perceptual grouping principle of good continuation [29], in taking advantage of the fact that color hue changes smoothly in most natural images. The TVF, which is also based on perceptual grouping (by using the principles of good continuation, proximity and similarity), has also been used for denoising. The application of the TVF to image processing is further discussed below in this section.

Previous studies have applied the TVF to color information mainly following two strategies. A first strategy applies the TVF to the color components directly. For example, in [30], color images are segmented by encoding the position and RGB color of every pixel into tensors of five dimensions before applying the TVF. Although this strategy uses all the color information available in the input image, it has shown limitations on noisy images. A second strategy converts

color information to a simplified representation before applying the TVF. In this direction, Mas-sad *et al.* [31] extract salient edges and junctions from gray-scale images by applying the TVF to local edge orientation encoded through 2D tensors. Jia and Tang [32] reconstruct damaged images by using the TVF on $(n \times n + 1)$ -dimensional tensors constructed from the gray-scale value of the $n \times n$ neighbors of every pixel plus the maximum value of them. Tai *et al.* [33] use the color gradient and local statistics in order to increase the resolution of images. More recently, Lim *et al.* [34] extract text from color images by applying the TVF on 3D tensors created from the pixel's position (row and column) and a single value calculated from its HSI color components. The results of those schemes based on the second strategy have shown that the TVF can be successful with color information. However, these schemes may discard important information since only a part of the available color information is used.

To our knowledge, only Tai *et al.* [35] have used the TVF in the specific area of image denoising. First, they classify every pixel as an *edge* or a *region* pixel by using local statistics. Second, they apply the TVF to the *edge* pixels in order to extract edges. Third, they define a neighborhood for each *region* pixel by using multiscale analysis and by excluding those neighbors that are separated from the *region* pixel by one of the edges extracted after the second step. Finally, the color of every pixel is calculated as the weighted mean of colors in its neighborhood. This approach has two important drawbacks: first, it depends of an initial classification of pixels that is not conducted in a robust way. Thus, the performance of the algorithm is likely to decrease in very noisy images where the number of pixels initially classified as edges increases. Second, using the weighted mean can lead to loss of texture and to quantization artifacts.

These drawbacks are mainly the consequence of using a robust technique, such as the classical TVF, in a single step of the whole process, with the other steps being based on non-robust techniques. Thus, the complete process could be improved by replacing those non-robust techniques by robust ones. The classical TVF is such a robust technique. However, the canonical voting fields used in the classical TVF are not appropriate for color information as stated above. In this context, instead of using different robust techniques, this paper explores the alternative of extending the classical TVF to the image denoising problem in the spatial domain so that it can be at the core of the denoising process, avoiding in such a way the drawbacks of the method presented in [35]. This alternative has the additional advantage that the appropriateness of the TVF for image denoising can be fully established, since the results cannot be attributed to any

other supporting technique.

Previous works have proposed two different encodings of color through tensors. On the one hand, in [36], the three channels of the HSL model are represented through a single 2D tensor in which hue, saturation and luminance are encoded through the orientation, shape and size of the tensor respectively. Unfortunately, this representation cannot be used for tensor voting since tensors defined in such a way cannot be added, which is essential for this method. On the other hand, some works have used third order tensors to encode color patches (e.g. [37–41]). However, this encoding faces three difficulties: the encoding tensors are non-positive semidefinite in general, the perceptual interpretation of them is not clear, and the summation operation on them is not defined. Thus, since tensor voting requires positive semidefinite tensors in addition to a summation operation, and its voting step has been devised to propagate perceptual information, they cannot be used for tensor voting either. Hence, a new encoding of color is proposed in this paper (cf. Subsection 4.1).

3. Perceptual Criteria for Propagating Local Information in Color Image Denoising

The first step necessary to design a voting process for adapting the TVF to a specific application is to have a reasonable description of how local information is propagated to the neighbors in that application. The canonical voting fields presented in [7] were designed to propagate the encoded information by using the plausible hypothesis that, in general, normal vectors tend to smoothly change over surfaces. However, those canonical voting fields are not the right option for color information, since color does not follow this property. This fact makes necessary the definition of a new voting process more appropriate for color image denoising. This voting process should be inspired by the perceptual process carried out by the human visual system in order to perform image denoising. However, this is a difficult task, since that process depends on many factors, such as background and surrounding color, viewing distance, texture, amount and type of noise present in the image, presence of edges or local contrast, among many others ([42, 43]).

In this article, a set of intuitive criteria inspired by the human perceptual process of image denoising is used in order to obtain good results while keeping a reasonable complexity of the proposed algorithm. These criteria are based on the perceptual grouping laws of similarity and proximity, and constitute the foundation of the voting process proposed in this paper (cf. Section 4). Three main perceptual features are involved in the definition of these criteria, namely: per-

ceptual color difference, uniformity and edginess. Perceptual color difference aims at measuring how similar every pair of colors appears to a human. Uniformity measures how variable color appears in a specific noiseless region. Edginess aims at measuring the likelihood of finding edges or texture in a specific region. Edginess in edge-preserving image denoising is not only related to the presence of edges, but also to the presence of texture, since texture should also be preserved.

The following criteria related to perceptual color differences are taken into account. On the one hand, in the absence of other clues, a small perceptual color difference between neighboring pixels should be mainly attributed to noise. In this case, the hypothesis that both pixels should have the same color becomes plausible. In this situation, edginess is usually low, while the uniformity of the region where the pixels are located is usually high. On the other hand, in the absence of other clues, a big perceptual color difference between neighboring pixels cannot be attributed exclusively to noise, since edges or texture can also generate this type of differences. Thus, other clues are necessary to decide whether to filter these pixels or not. However, if the region is noiseless, edginess is usually high, while uniformity is usually low. A special case is due to impulse noise that appears in pixels with a high perceptual color difference with respect to all their neighbors. Pixels with impulse noise should absorb the color, uniformity and edginess of their neighbors. Impulse noise must be taken into account since it may appear in imaging sensors [44].

The following criteria related to uniformity are taken into account. On the one hand, a perceptual color difference between neighboring pixels should be mainly attributed to noise when there is evidence that both pixels belong to the same uniform region. In this case, the hypothesis that both pixels should have the same color becomes plausible, disregarding the perceptual color difference between them. In this situation, edginess is usually low. On the other hand, other clues are necessary to decide whether to filter or not the pixels of regions with low uniformity, since this feature is also common in regions with edges or texture. However, if the region is noiseless, its edginess and perceptual color differences are usually high.

The following criteria related to edginess are taken into account. On the one hand, other clues are necessary to decide whether to filter or not the pixels of regions with high edginess, since, although this is a specific feature of regions with edges or texture, eventually, it can also be found in noisy regions. However, if the region is noiseless, its uniformity will usually be low. On the other hand, regions with low edginess should be treated similarly to uniform regions. Hence,

their pixels should be filtered. In this case, uniformity will also be high.

Other general criteria are also important. First, the influence of pixels on their neighbors should depend on the distance. Thus, closer neighbors should be more affected by a pixel than farther ones. Second, if uniformity and/or edginess at a pixel in a color channel appear too different from those in the other color channels, that difference could be caused by noise and, although not always being the case, the presence of noise in this pixel becomes more likely.

4. Tensor Voting Framework for Edge-Preserving Color Image Denoising

The input of the proposed method is the set of pixels of a noisy color image. Thus, positional and color information is available for every input pixel. Positional information is used to determine the neighborhood of every pixel, while color information is used to define the tensors in the encoding step. It is well known that color can be represented by a variety of color models whose selection depends on the particular application, since each model has a specific scope of use. CIELAB [45] is the most appropriate color model for the proposed color image denoising approach, since some criteria described in the previous section are based on the estimation of perceptual color differences and CIELAB was designed to measure this kind of differences [45]. Furthermore, CIELAB is the color model that most accurately represents the color perception process performed by the human visual system.

Before applying the proposed method, every CIELAB channel is normalized in the range $[0, \pi/2]$. As an example, normalization factors of $\frac{\pi}{200}$ for channel L and $\frac{\pi}{2 \times 255}$ for channels a and b are appropriate for outdoor scenarios, since the CIELAB channels are in the ranges $0 \leq L \leq 100$, $-127 \leq a \leq 128$ and $-127 \leq b \leq 128$ respectively, when color is converted from RGB for these types of scenarios¹. An additional shift of 127 for channels a and b is also necessary before normalization in order to avoid negative values. Obviously, this normalization process must be adjusted for other types of scenarios.

The next subsections describe the details of the proposed edge-preserving color image denoising method.

¹by using the standard illuminant D65 (which emulates the illuminant effect of the sun) and a two degrees observer (i.e., the subtended angle in the observer's retina by an individual color stimulus is two degrees), which are more appropriate in applications where individual color stimuli are small [42].

4.1. Encoding of Color Information

The encoding of color information must be in compliance with the perceptual criteria mentioned in Section 3. Thus, not only color, but also uniformity and edginess must be encoded. This objective can be carried out by means of tensors. However, as mentioned above, most of the methodologies previously proposed in the literature for encoding color through tensors are not suitable for tensor voting. Thus, an adequate encoding scheme must encode color, uniformity and edginess while being compatible with tensor voting. The following encoding procedure succeeds in dealing with these issues.

In the first step of the method, the color information of every pixel is encoded through three second-order 2D tensors, one for each normalized CIELAB color channel. These tensors are represented by 2×2 symmetric positive semidefinite matrices that can be graphically represented by means of 2D ellipses. There are two extreme cases for the proposed tensors: *stick tensors*, which are *stick*-shaped ellipses with a single eigenvalue, λ_1 , different from zero, and *ball tensors*, which are circumference-shaped (a circumference is indeed a 2D *ball*) ellipses whose λ_1 and λ_2 eigenvalues are equal to each other.

Three perceptual measures are encoded in the tensors associated with every input pixel, namely: the most likely normalized color of the pixel (in the specific channel), a metric of local uniformity, and an estimation of edginess at the pixel's location. Figure 1 shows the graphical interpretation of a tensor for channel L . The most likely normalized color is encoded as the angle α between the x axis, which represents the lowest possible color value in the corresponding channel, and the eigenvector corresponding to the largest eigenvalue. For example, in channel L , a tensor with $\alpha = 0$ encodes black, while a tensor with $\alpha = \frac{\pi}{2}$ encodes white. Other values of α allow the tensors in channel L to encode other possible luminance levels in the range from black to white. In addition, local uniformity and edginess are encoded by means of the normalized $\hat{s}_1 = (\lambda_1 - \lambda_2)/\lambda_1$ and $\hat{s}_2 = \lambda_2/\lambda_1$ saliencies respectively. Thus, a pixel located at a completely uniform region is represented by means of three *stick tensors*, one for every color channel. In contrast, a pixel located at an ideal edge is represented by means of three *ball tensors*, one for every color channel. The trace of the tensors cannot be used as a measurement of either local uniformity or edginess, since the voting process is dense, that is, there are voting tensors at every possible location in the image. Thus, it is possible to find similar values of the trace at both edges and uniform regions. This behavior is also common for the TVF in dense datasets. For example,

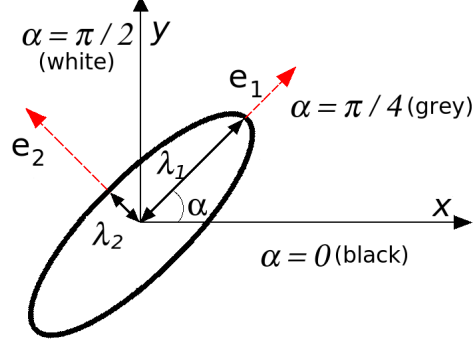


Figure 1: Encoding process for channel L . Color, uniformity and edginess are encoded by means of α and the normalized $\hat{s}_1 = (\lambda_1 - \lambda_2)/\lambda_1$ and $\hat{s}_2 = \lambda_2/\lambda_1$ saliencies respectively.

[31] describes some of the problems faced by tensor voting in dense datasets.

Before applying the voting process, it is necessary to initialize the tensors associated with every pixel. The colors of the noisy image can be easily encoded by means of the angle α between the x axis and the principal eigenvector as described above. However, since metrics of uniformity and edginess are usually not available at the beginning of the process, normalized saliency \hat{s}_1 is initialized to one and normalized saliency \hat{s}_2 is initialized to zero. Hence, the initial color information is encoded through *stick tensors* oriented along the directions that represent the original color of the pixel given in the normalized CIELAB channels. This initialization is carried out by:

$$\mathbf{T}_c(p) = \vec{t}_c(p) \vec{t}_c(p)^T, \quad (1)$$

with $\vec{t}_c(p) = [\cos(C_c(p)) \quad \sin(C_c(p))]^T$, where $\mathbf{T}_c(p)$ is the tensor of the c -th color channel (L , a and b) at pixel p and $C_c(p)$ is the normalized value of the c -th color channel at p .

Splitting color information into as many tensors as color channels is advantageous since the processing necessary to be applied to every channel may be different. For example, the use of 3-dimensional second-order tensors is inappropriate, since filtering one channel can corrupt the information of the other channels, while noise only present in some of the channels can be easily detected and eliminated through the proposed scheme. However, since the color channels are not independent in general, information from a channel must be used to process the others. Both facts are taken into account in the design of the voting process.

In addition, it is important to remark that the proposed scheme is also preferable to alter-

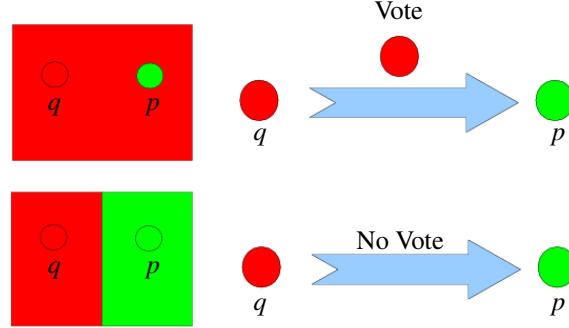


Figure 2: A *stick* vote, which can be interpreted as a vote for a specific color, requires context information. Two cases in which pixel q should cast (top) and should not cast (bottom) a *stick* vote to pixel p .

natives based on higher-dimensional second-order tensors of dimension larger than three. On the one hand, the proposed scheme leads to more efficient methods, since it only requires the inexpensive eigendecomposition of 2×2 matrices, while those alternatives require eigendecompositions whose complexity rapidly increases rapidly with dimensionality. On the other hand, the proposed scheme requires less memory per pixel (9 different values) than the alternatives (e.g., 10, 15 and 21 values for 4-, 5- and 6-dimensional tensors, respectively).

4.2. Voting Process

In the second step of the TVF, the tensors associated with every pixel are propagated to their neighbors through a convolution-like process. This step is independently applied to the tensors of every channel (L , a and b). A difference with the surface reconstruction problem is that it is not possible to apply the canonical voting fields for the image denoising problem, since a pixel cannot appropriately propagate its information to its neighbors without taking into account the local relations between the information at that pixel and at its neighbors. This fact is evidenced in the majority of criteria presented in Section 3. Hence, specially designed tensorial functions, referred to as *propagation functions*, must be used instead of the canonical voting fields presented in [7]. These propagation functions must take into account not only the information encoded in the tensors but also the local relations between neighbors. This can be appreciated in Figure 2. Although the colors of pixel q and p in the figure are the same, in the top example, pixel q should contribute to the suppression of noise at p by casting a “red” vote (see below), whereas in bottom example, information at q should not be used for filtering p , since p is already noiseless.

Two propagation functions are proposed for applying the TVF to color information: a *stick* and a *ball* propagation function. A vote is the result of applying a propagation function from q (the voter) to p (the voted). Hence, the application of the first function leads to *stick* votes, while the application of the second function produces *ball* votes. A *stick* vote can be seen as a *stick*-shaped tensor with a strength modulated by scalar factors, whereas the *ball tensor* can be seen as a circumference-shaped tensor also weighted by appropriate scalar factors. The *stick* vote is used to propagate a specific color (see Figure 2), while the *ball* propagation function is used to increase edginess by voting for all possible colors. The proposed voting process at every pixel is carried out by adding all the tensors propagated towards it from its neighbors by applying the proposed propagation functions. Thus, the total vote received at a pixel p for each color channel c , $TV_c(p)$, is given by:

$$TV_c(p) = \sum_{q \in \text{neigh}(p)} S_c(p, q) + B_c(p, q), \quad (2)$$

where $S_c(p, q)$ and $B_c(p, q)$ are the *stick* and *ball* propagation functions respectively. Section 5 presents these *stick* and *ball* propagation functions specifically tailored to color image denoising.

After applying the voting process, it is necessary to obtain eigenvectors and eigenvalues of $TV_L(p)$, $TV_a(p)$ and $TV_b(p)$ at every pixel p in order to analyze its local perceptual information. The TVF defines a standard way to interpret the voting results: uniformity increases with the normalized \hat{s}_1 saliency and the likelihood that a point belongs to an edge increases as the normalized \hat{s}_2 saliency becomes greater than the normalized \hat{s}_1 saliency. Additionally, the most likely normalized color at a pixel is given for each color channel by the angle between the first eigenvector of the corresponding tensor and the x axis. These three angles are then used to correct the color of every pixel with the most likely one, reducing in such a way the noise of the image. These three angles can always be calculated, even if all neighbors of the pixel only cast ball votes towards it. This is guaranteed by allowing self-votes, which are always *stick* votes.

The main difference between the TVF and the proposed method is that, in addition to edginess, while the former encodes and propagates orientations, the latter encodes and propagates color. Thus, while edginess and edge orientation can be directly extracted from the resulting tensors after applying the TVF, color cannot. Hence, additional steps are necessary to estimate color for image denoising, as it is the case for the strategy by Tai *et al.* [35]. In contrast, edginess and color in the proposed method can be directly extracted from the resulting tensors, while edge orientation cannot. However, this is not a problem for image denoising, since edge orientation is

not required as an outcome.

Many methods use edge orientation to prevent filtering across edges and allow the latter along them. They usually use this information to modulate the importance of every neighboring pixel in the filtering process. This anisotropic performance is also achieved by the proposed method through the voting process. *Stick* votes, which are mainly responsible for filtering, are discouraged when an edge is likely present between the voter and the votee, as described in the following section.

5. Propagation Functions for Image Denoising

This section presents the propagation functions specifically designed to color image denoising by taking into account the criteria described in Section 3.

5.1. Modeling of Variables Involved in the Voting Process

Four variables associated with local relations between neighbors are taken into account in the proposed propagation functions: the distance between pixels, the perceptual color difference, the joint uniformity measurement and the likelihood of a pixel being impulse noise. First, the Euclidean distance is used to measure the distance between pixels. Second, the perceptual color difference between pixels p and q , $\Delta E(p, q)$, is calculated through CIEDE2000 [46] (cf. Subsection 5.3). It is also necessary to estimate the perceptual color difference in a specific channel c , $\Delta E_c(p, q)$. This value is also calculated by means of CIEDE2000 by setting to zero the difference in all channels different from c . Third, since the uniformity of a region (cf. Section 3) that contains two pixels (the voter and the voted) cannot be calculated directly from their tensors (the normalized \hat{s}_1 saliency at p cannot be used directly, since it can only encode a local measurement of the uniformity at the region surrounding a pixel), a highly related metric is used instead: the joint uniformity measurement, $U_c(p, q)$, which is the product of the normalized \hat{s}_1 saliencies of both pixels, that is, the product of the local uniformity measurements. By definition, $U_c(p, q)$ varies in the range between zero (completely non-uniform) and one (completely uniform). Thus, let $\hat{s}_{1c}(p)$ be the normalized s_1 saliency at p in channel c (L , a and b). The joint uniformity of p and q in channel c , $U_c(p, q)$, can be estimated by:

$$U_c(p, q) = \hat{s}_{1c}(p) \hat{s}_{1c}(q). \quad (3)$$

Finally, the likelihood of a pixel being impulse noise, $\eta_c(p)$, can be estimated as the difference of normalized \hat{s}_2 saliencies between the pixel and its neighbors for those pixels located at local maxima of the normalized \hat{s}_2 saliency. Thus, let $s_{2c}(p)$ be the normalized \hat{s}_2 saliency at p in channel c (L , a and b) and, $\mu_{s_{2c}}(p)$ be the mean of the normalized \hat{s}_2 saliencies in the 8-neighborhood of pixel p in channel c . The likelihood of p being impulse noise in channel c , $\eta_c(p)$, is given by:

$$\eta_c(p) = \begin{cases} s_{2c}(p) - \mu_{s_{2c}}(p), & \text{if } p \text{ is at a local} \\ & \text{maximum of } s_{2c}(\cdot) \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

5.2. Design of the Stick and Ball Propagation Functions

The proposed *stick* propagation function, $S_c(p, q)$, which allows a pixel q to cast a *stick* vote to a neighboring pixel p for channel c is given by:

$$S_c(p, q) = GS(p, q) \overline{\eta}_c(q) S V'_c(p, q) ST_c(q), \quad (5)$$

with $ST_c(q)$, $GS(p, q)$, $\overline{\eta}_c(q)$ and $S V'_c(p, q)$ being defined as follows.

First, the tensor $ST_c(q)$ in (5) must encode the most likely normalized color at q , since *stick* votes cast by a pixel q are used to propagate its most likely color to its neighbors. Thus, $ST_c(q)$ is defined as the tensorized eigenvector corresponding to the largest eigenvalue of the voter pixel, that is $ST_c(q) = \vec{e}_{1c}(q) \vec{e}_{1c}(q)^T$, with $\vec{e}_{1c}(q)$ being the eigenvector with the largest eigenvalue of the tensor associated with channel c at q .

Second, the three scalar factors in (5), each ranging between zero and one, are defined as follows. The first factor, $GS(p, q)$, models the influence of the distance between p and q in the vote strength. The *stick* vote strength cast by closer neighboring pixels must be greater than by farther ones. Thus, $GS(p, q)$ is defined as $GS(p, q) = G_{\sigma_s}(\|p - q\|)$, where $G_{\sigma_s}(\cdot)$ is a decaying Gaussian function with zero mean and a user-defined standard deviation σ_s . The second factor $\overline{\eta}_c(q)$, defined as $\overline{\eta}_c(q) = 1 - \eta_c(q)$, is introduced in order to prevent a pixel q previously classified as impulse noise from propagating its information. This factor makes the vote to be zero when q is completely noisy and leaves the vote unaffected when q has not been classified as impulse noise. The third factor, $S V'_c$, takes into account the influence of the perceptual color difference, the uniformity and the noisiness of the voted pixel. This factor is given by:

$$S V'_c(p, q) = \overline{\eta}_c(p) S V_c(p, q) + \eta_c(p), \quad (6)$$

where:

$$SV_c(p, q) = \frac{G_{\sigma_d}(\Delta E(p, q)) + U_c(p, q)}{2}, \quad (7)$$

and $\overline{\eta}_c(p) = 1 - \eta_c(p)$.

$SV_c(p, q)$ models the fact that a pixel q must cast a stronger *stick* vote to p either if both pixels belong to the same uniform region or if the perceptual color difference between them is small. The joint uniformity measurement, $U_c(p, q)$, is used to determine if both pixels belong to the same region or not. A Gaussian function with zero mean and a user-defined standard deviation σ_d , $G_{\sigma_d}(\cdot)$, which decays with $\Delta E(p, q)$ is used to determine if the perceptual color difference is small or not. The value of σ_d must be chosen in accordance with the fact that the threshold of $\Delta E(p, q)$ for a Just Noticeable Color Difference (JNCD) between p and q is theoretically a single CIEDE2000 unit. However, values in the range between one and five are usually accepted in some applications [47]. In practice, results can be smoothed by increasing σ_d . A normalizing factor of two is used in order to make $SV_c(p, q)$ to vary from zero to one. The term $\eta_c(p)$ included in (6) makes noisy voted pixels, p , to adopt the color of their voting neighbors, q , disregarding local uniformity measurements and perceptual color differences between p and q . The term $\overline{\eta}_c(p)$, also included in (6), makes SV'_c to vary from zero to one. As expected, the effect of $\eta_c(p)$ and $\overline{\eta}_c(p)$ on the strength of the *stick* vote received at a noiseless pixel p is null.

In turn, the *ball* propagation function, $B_c(p, q)$, which allows a pixel q to cast a *ball* vote to a neighboring pixel p for channel c is given by:

$$B_c(p, q) = GS(p, q) \overline{\eta}_c(q) BV_c(p, q) BT(q), \quad (8)$$

with $BT_c(q)$, $GS(p, q)$, $\overline{\eta}_c(q)$ and $BV_c(p, q)$ being defined as follows.

First, the *ball tensor*, represented by the identity matrix, \mathbf{I} , is the only possible tensor for $BT(q)$, since it is the only tensor that complies with two main design restrictions: a *ball* vote must be equivalent to casting *stick* votes for all possible colors using the hypothesis that all of them are equally likely and, the normalized \hat{s}_1 saliency must be zero when only *ball* votes are received at a pixel. The first restriction is based on the fact that the best way to increase edginess at a pixel is to cast *stick* votes for all possible colors. The second restriction avoids undesirable color biases. Thus, $BT(q) = \mathbf{I}$.

Second, the strength of the *ball* vote is modulated by three scalar factors, each varying between zero and one, described as follows. The first and second factors, $GS(p, q)$ and $\overline{\eta}_c(q)$, are

the same as the ones introduced in (5) for the *stick* propagation function. They are included for similar reasons to those given in the definition of the *stick* propagation function. The third scalar term in (8), $BV_c(p, q)$, is given by:

$$BV_c(p, q) = \frac{\overline{G_{\sigma_d}}(\Delta E(p, q)) + \overline{G_{\sigma_d}}(\Delta E_c(p, q)) + \overline{U_c}(p, q)}{3}, \quad (9)$$

where $\overline{G_{\sigma_d}}(\cdot) = 1 - G_{\sigma_d}(\cdot)$ and $\overline{U_c}(p, q) = 1 - U_c(p, q)$.

$BV_c(p, q)$ models the fact that a pixel q must reinforce the edginess at the voted pixel p either if there is a big perceptual color difference between p and q , or if p and q are not in a uniform region. This behavior is modeled by means of the terms $\overline{G_{\sigma_d}}(\Delta E(p, q))$ and $\overline{U_c}(p, q)$. The additional term $\overline{G_{\sigma_d}}(\Delta E_c(p, q))$ is introduced in order to increase the edginess of pixels in which the only noisy channel is c . Thus, those pixels p with noise in a single color channel receive stronger *ball* votes in that channel. This decreases the strength of the *stick* votes cast by those noisy pixels in next iterations. In addition, this also allows the method to eliminate noise from those pixels in the following iterations more easily, since $\eta_c(p)$ tends to increase. The normalizing factor of three in (9) allows the *ball* propagation function to cast *ball* votes with a strength between zero and one.

It is important to remark that $\Delta E_c(p, q)$ is not included in the *stick* propagation function, since a small perceptual color difference in a specific channel not always indicates a uniform region, making inconvenient the propagation of color. Similarly, the terms $\eta_c(p)$ and $\overline{\eta_c}(p)$ do not appear in the *ball* propagation function, since q is not propagating a single color, making thus these terms unnecessary.

It is not difficult to show that the proposed propagation functions comply with the criteria described in Section 3. Perceptual color differences, uniformity measures, dependency on spatial distance and impulse noise measurements explicitly appear in (5) and (8). Although the edginess of a region does not explicitly appear in the above propagation functions, it is indirectly taken into account in the *ball* propagation function, since it can be modeled as the complement of the joint uniformity, that is, $\overline{U_c}(p, q)$. Table 1 shows the dependency of both types of votes on the four variables described in Subsection 5.1. As already mentioned, the *stick* votes generated through (5) can be interpreted as votes for specific colors, whereas *ball* votes generated through (8) are *ball* tensors whose strength depends on the variables described in Subsection 5.1 as shown in Table 1.

Table 1: Dependency of votes on the increase of different variables. (*) The dependency is reduced with the amount of impulse noise at p .

Increase of	<i>Stick</i> votes	<i>Ball</i> votes	Reason
Distance	Decrease	Decrease	Only close neighbors are allowed to vote.
Percept. color diff.	Decrease (*)	Increase (*)	Large color differences are common in edges.
Joint uniformity	Increase (*)	Decrease (*)	Color is propagated in uniform regions, while edginess is not.
Impulse noise at q	Decrease (*)	Decrease (*)	Pixels with impulse noise cannot propagate information.
Impulse noise at p	Increase	Increase	Pixels with impulse noise should accept all votes.

It should also be highlighted that, unlike the canonical voting fields, it is not necessary to rotate the results of the proposed propagation functions as they do not depend on the orientation of the tensors, making the process less computationally expensive than the TVF applied to surface reconstruction.

The proposed propagation functions require to apply the voting process (Section 4.2) twice. The first application produces an initial estimation of the normalized \hat{s}_1 and \hat{s}_2 saliencies, as they are necessary to calculate $U_c(p, q)$ and $\eta_c(p)$ in (3) and (4). For this first estimation, only perceptual color differences and spatial distances are taken into account, since no more information is available. Thus, $U_c(p, q)$ and $\eta_c(p)$ are set to zero in this first application. At the second application, the tensors at every pixel are initialized with the tensors obtained after the first application. Therefore, $T_c(p) = TV_c(p)$ instead of (1). In this second application, the propagation functions can be applied in their full definition, since all necessary data are available.

As mentioned above, the spatial influence of the neighbors is controlled by means of the Gaussian function $G_{\sigma_s}(\cdot)$ in (5) and (8). In practice, this function defines a neighborhood of radius $3\sigma_s$ around every pixel, since votes cast from farther pixels are too small to take them into account in the voting process. Thus, for example, a σ_s of 1.3 defines a disk-shaped neighborhood of about 50 pixels.

Basically, the neighborhood's size should be in accordance with the smallest structure in the image. Preliminary experiments yielded that $\sigma_s = 1.3$ is an appropriate value for the images

used in the experiments of Section 7. However, similarly to most image denoising methods, this parameter could change for other datasets.

5.3. Parameters of the CIEDE2000 formula

The CIEDE2000 formula, which estimates the perceptual color difference between two pixels p and q , $\Delta E(p, q)$, is defined as [46]:

$$\Delta E(p, q) = \sqrt{\left[\frac{\Delta L'}{k_L S_L}\right]^2 + \left[\frac{\Delta C'}{k_C S_C}\right]^2 + \left[\frac{\Delta H'}{k_H S_H}\right]^2 + R_T \left[\frac{\Delta C'}{k_C S_C} \frac{\Delta H'}{k_H S_H}\right]}, \quad (10)$$

where k_L , k_C and k_H are the parameters of CIEDE2000, and $\Delta L'$, $\Delta C'$, $\Delta H'$, S_L , S_C , S_H and R_T are functions of L , the CIELAB luminance, $C = \sqrt{a^2 + b^2}$, the CIELAB chroma and $h = \arctan \frac{b}{a}$, the CIELAB hue of both colors at p and q . The three CIEDE2000 parameters are used to weight the differences in CIELAB luminance, chroma and hue respectively. They can be adjusted to make the CIEDE2000 formula more suitable to every specific application by taking into account factors such as noise, background luminance or texture, since the estimation of the perceptual difference between two colors given by CIEDE2000 is only accurate for the restricted environments of the experiments that led to its design [48]. These parameters must be greater than or equal to one. If no more information is available, it is recommended to set these parameters to one [46]. However, some studies (e.g., [49, 50]) have shown that the performance of the CIEDE2000 formula in estimating perceptual color differences can be improved by appropriately setting those parameters. These studies have also found that the background luminance and the amount of noise are important factors to determine the appropriate value of the CIEDE2000 parameters. Although these studies propose equations to approximate the values of k_L , k_C and k_H , they are not applicable in all cases, since they were obtained in restricted scenarios. Therefore, it is necessary to propose a set of equations to estimate the most appropriate values for k_L , k_C and k_H for the specific application of edge-preserving image denoising.

Based on the formulation given in [50], the following equations for the CIEDE2000 parameters are proposed:

$$k_L = F_{B_L} F_{\eta_L}, \quad k_C = F_{B_C} F_{\eta_C}, \quad k_H = F_{B_h} F_{\eta_h}, \quad (11)$$

where F_{B_m} are factors that take into account the influence of the background color on the calculation of color differences in every color component m (L , C and h), and F_{η_m} are factors that take

into account the influence of noise on the calculation of color differences in every component m . These factors are defined as follows.

On the one hand, F_{B_m} takes into account the fact that, as the background luminance decreases, big color differences in chromatic channels become less perceptually visible. This effect only appears in regions with a CIELAB background luminance, L_B , below fifty². Chou and Liu [50] have found that the effect of the background on the estimation of color differences is approximately a linear function of the mean background luminance Y_B , which is the second color component of the mean background color expressed in the XYZ color model. Additionally, only parameters k_C and k_H are affected by background luminance, since the latter only affects the chromatic channels. Hence, $F_{B_L} = 1$. Thus, the effect of the background on the calculation of perceptual color differences in the color components C and h , F_{B_C} and F_{B_h} , can be estimated through the following equations:

$$F_{B_C} = F_{B_h} = 1 + R_B (1 - Y_B), \quad (12)$$

where R_B is a constant parameter. A value of three for R_B has shown the best performance in our experiments. Thus, factors F_{B_C} and F_{B_h} are in the range from one to four, since Y_B varies between zero and one. A factor of one means that the background color does not have any influence on the calculation of color differences, while a factor of four means that color differences found in the corresponding color component must be divided by four as the human visual system acuity is reduced four times in those conditions of background color.

On the other hand, F_{η_m} takes into account the fact that, as noise increases, big color differences become less perceptually visible. For this reason, it is necessary to estimate the amount of noise and texture present in the image in order to study their influence on the calculation of perceptual color differences. This objective can be achieved by assessing the variability in the image. There are three main sources of variability in an image, namely: edges, texture and noise. However, it is only necessary to determine the variability due to noise and texture since there is no evidence that k_L , k_C and k_H depend on the variability due to edges. Assuming uniformly distributed noise and texture in the noisy image, the variability due to edges can be estimated

²There is no effect of L_B on CIEDE2000 for $L_B = 50$, since the experiments that led to the definition of the CIEDE2000 formula assumed that value. Moreover, to our knowledge, there is no evidence that greater values of L_B increase the acuity of the human visual system.

through the following steps. First, the noisy image is convolved with a Gaussian of big variance (e.g., $\sigma^2 = 25$) and zero mean in order to eliminate the variability due to noise and texture. Second, the median absolute deviation (*MAD*) is calculated on both the noisy and the Gaussian filtered image. Finally, the *MAD* of the Gaussian filtered image gives an estimation of the variability due to edges, while the difference between the *MAD* of the noisy image and the *MAD* of the Gaussian filtered one gives an estimation of the variability due to noise and texture. This process is applied to the CIELAB luminance, chroma and hue components independently. Calculations of *MAD* require the evaluation of perceptual differences in luminance, chroma and hue. The absolute difference can be used to calculate perceptual luminance and chroma differences. However, that function is not appropriate for calculating perceptual hue differences, since they are more perceptually visible for colors with high chroma. The equation defined in CIEDE2000 to calculate this type of differences must be used instead. Thus, the perceptual hue difference, ΔH , is calculated as $\Delta H = 2 \sqrt{C_1 C_2} \sin\left(\frac{\Delta h}{2}\right)$, where C_1 and C_2 are the chroma of both colors and Δh is the absolute hue difference.

Three parameters, J_L , J_C and J_h , are introduced in order to control the degree of preservation of texture on each color component, given that it is important to keep not only edges but also texture while noise is suppressed. In natural images, J_C , and J_h may be set to zero, since texture in these images is mainly retained by the luminance component [28]. The definition of J_L depends on the perceptual amount of perceived texture in the image without noise. Thus, big values of J_L should be used in highly textured images as this ensures the preservation of texture. Thus, its choice involves a trade-off between preserving texture and filtering small details. In our experiments, good results were obtained for values of J_L in the range from five to ten CIEDE2000 units, depending on the amount of texture in the image.

Since a good feature of a denoising algorithm is the suppression of noise as soon as possible, the CIEDE2000 parameters have been weighted by the amount of variability due to noise with a tolerance of a single CIEDE2000 unit. Thus, the equations proposed to model the influence of noise on the calculation of perceptual color differences are:

$$F_{\eta_m} = \begin{cases} \eta_m, & \text{if } \eta_m > 1 \\ 1, & \text{otherwise,} \end{cases} \quad (13)$$

with:

$$\eta_m = (I)_m - MAD(G)_m - J_m, \quad (14)$$

where I is the noisy image, G the image filtered with the Gaussian function and $MAD(\cdot)_c$ is the MAD calculated on component m (L , C or h). Image G has also been used for estimating local values of Y_B in (12) by converting CIELAB into XYZ.

The F_{η_m} factors are in the range from one to infinite. However, in practice, values larger than fifteen are uncommon. A factor of one means that the noise level is too low to influence the estimation of color differences. A factor of five, for example, means that color differences found in the corresponding color component must be divided by five, as the human visual system acuity is reduced five times due to the noise level. Additionally, these equations make F_{η_m} to converge to one in a few iterations of the algorithm, making their calculation unnecessary once they have converged, since the images will not have enough noise to influence the estimation of color differences.

Our experiments have shown that the equations presented in this subsection lead to a good performance of the algorithm when denoising outdoor natural images. However, those equations should be adjusted for other types of scenarios.

6. Quality Measurement

In accordance with the features that must fulfill an edge-preserving image denoising algorithm (cf. Section 1), metrics to measure the amount of removed noise, the edge preservation feature and the presence of artifacts after filtering are necessary to assess the performance of image denoising algorithms. Thus, in addition to visual inspection, three metrics have been utilized in our experiments: the peak signal to noise ratio (PSNR), the PSNR weighted by the gradient (PSNRG), and the PSNR in false edges and/or undesirable artifacts (PSNRA). The PSNR is a well-known quality measure, while PSNRG and PSNRA are new metrics proposed in this paper.

The PSNR metric was selected since it is the most widely used metric to compare denoising algorithms, given that it is easy to implement and gives a good estimation of the amount of removed noise. The PSNR for a color image, I , encoded in RGB with respect to a reference image, R , is calculated as:

$$\text{PSNR} = 10 \log_{10} \left[\frac{3 \times 255^2}{\text{MSE}_R + \text{MSE}_G + \text{MSE}_B} \right], \quad (15)$$

with

$$\text{MSE}_c = \frac{1}{mn} \sum_{i=0}^n \sum_{j=0}^m (I_c(i, j) - R_c(i, j))^2. \quad (16)$$

In addition, we propose the PSNRG metric to measure the edge preservation feature. The PSNRG is defined as the PSNR weighted by the gradient of the reference image, $\|\nabla R_c(i, j)\|$. The PSNRG is a metric that gives more weight to pixels located nearby edges and discards those pixels located in flat regions. The PSNRG is defined as:

$$\text{PSNRG} = 10 \log_{10} \left[\frac{3 \times 255^2}{\text{MSEG}_R + \text{MSEG}_G + \text{MSEG}_B} \right], \quad (17)$$

with

$$\text{MSEG}_c = \frac{\sum_{i=0}^n \sum_{j=0}^m \|\nabla R_c(i, j)\| (I_c(i, j) - R_c(i, j))^2}{\sum_{i=0}^n \sum_{j=0}^m \|\nabla R_c(i, j)\|}. \quad (18)$$

Finally, we propose the PSNRA metric to measure the introduction of false edges and/or undesirable artifacts in uniform regions. In a first step, color differences at every pixel are weighted by the difference of gradients between the reference and the filtered image. This process is only applied to those pixels where the gradient of the filtered image is greater than the gradient of the reference image. In addition, only those pixels with the greatest differences of gradients are taken into account in order to eliminate a possible influence of pixels near edges whose gradient could become slightly greater than the gradient in the reference image. Finally, the PSNRA is calculated in a similar way to the PSNR for the selected pixels. Let A be the set of pixels where the difference between the gradient of I and R is greater than the 90th percentile of the differences of gradients. The PSNRA is defined as:

$$\text{PSNRA} = 10 \log_{10} \left[\frac{3 \times 255^2}{\text{MSEA}_R + \text{MSEA}_G + \text{MSEA}_B} \right], \quad (19)$$

with

$$\text{MSEA}_c = \frac{\sum_{i=0}^n \sum_{j=0}^m \Delta Gr(i, j) (I_c(i, j) - R_c(i, j))^2}{\sum_{i=0}^n \sum_{j=0}^m \Delta Gr(i, j)}, \quad (20)$$

$$\Delta Gr(i, j) = \delta(i, j) (\|\nabla I(i, j)\| - \|\nabla R(i, j)\|), \quad (21)$$

and

$$\delta(i, j) = \begin{cases} 1, & \text{if } \|\nabla I(i, j)\| > \|\nabla R(i, j)\| \\ & \text{and pixel } I(i, j) \in A \\ 0, & \text{otherwise.} \end{cases} \quad (22)$$

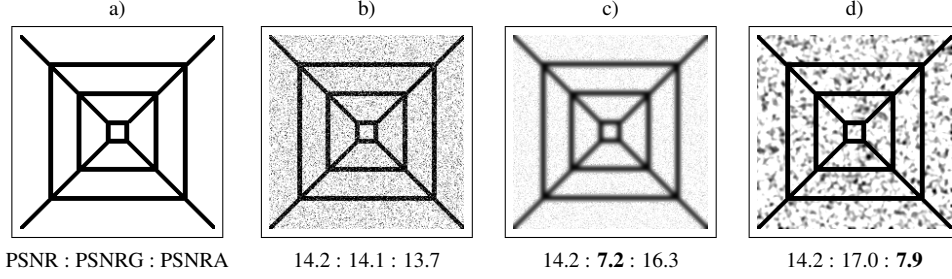


Figure 3: Visual assessment of PSNRG and PSNRA. a) Synthetic image. b-d) Noisy versions of the same image. The noisy images b)-d) have the same PSNR but different PSNRG and PSNRA.

The utility of the new proposed metrics, PSNRG and PSNRA, can be visually assessed in the synthetic example of Figure 3. This figure shows three different noisy versions of the same image that have the same PSNR but very different PSNRG and PSNRA. PSNRG is drastically reduced in Figure 3 c) due to edge blurring, while PSNRA is drastically reduced in Figure 3 d) due to the high amount of introduced artifacts.

Finally, visual inspection is also necessary in order to give a subjective assessment of the results, since the metrics mentioned above were not designed to measure perceptual fidelity, that is, it is possible to find cases in which images with high PSNR (and therefore with low noise level) could be perceived noisier than other images with lower PSNR. Furthermore, the so-called *Method Noise*, defined in [15], can also be used to visually assess the performance of image denoising. The *Method Noise* is an image defined as the difference between the input and the output of the denoising method. Visual inspection of the *Method Noise* can be used for performance evaluation, especially for additive white Gaussian noise, since the *Method Noise* yielded by an ideal algorithm must not contain structure.

7. Experimental Results

The proposed technique, referred to as the tensor voting denoiser (TVD), has been compared to the methods proposed by Kervrann and Boulanger [51] (an improved version of the non-local means method [15]), Tschumperlé [18] (based on partial differential equations), and Portilla *et al.* [23] (based on wavelets), since they represent the state-of-the-art in color image denoising. These methods will be referred to as NLM [51], PDE [18] and GSM [23] respectively. The default parameters of NLM and PDE have been used. The GSM algorithm has been applied with

$\sigma = 20$, since the best overall performance of this algorithm was attained with this standard deviation. The GSM algorithm has been applied to the three *RGB* channels independently, since this algorithm was designed for gray-level images. The TVD has been run with standard deviations $\sigma_s = 1.3$ for the G_{σ_s} Gaussian, $\sigma_d = 1.0$ for the G_{σ_d} Gaussian and $J_L = 7.0$ (cf. Subsection 5.3).

For the experiments, one hundred outdoor images with a size of either 481x321 or 321x481 pixels have been taken from the Berkeley segmentation data set [52]. These images have been contaminated with two types of noise: additive white Gaussian noise (AWGN) and CCD camera noise [1]. Although AWGN is unable to accurately model the noise generated by real cameras [44], it has been included in the comparisons since most methods of the state-of-the-art have only been tested with it. In addition, it is interesting to assess the behavior of those state-of-the-art algorithms with respect to CCD camera noise in order to validate their performance in real conditions. This type of noise, mainly generated by the CCD sensor, is strongly modified in the CCD camera imaging pipeline, which includes necessary processes for converting raw data into images, such as demosaicing and Gamma correction [53]. Ten iterations in cascade (each iteration uses as input the output of the previous one) have been run for every algorithm, input image, noise type and noise level. The output image with the highest PSNR obtained from those iterations was chosen. The chosen images have been used to compare the performance of all the algorithms. No pre or post-processing stages have been applied to the images in order to evaluate the ability of the algorithms to remove noise without any help. Thus, the reported measurements are only due to the performance of the algorithms and not to any additional stage.

Regarding computational cost, NLM was the fastest of all tested algorithms when run on an Intel Core 2 Quad Q6600 with a 4GB RAM (about three seconds per iteration), followed by PDE (around five seconds per iteration), TVD (around twenty seconds per iteration, since every application of the voting process takes ten seconds approximately) and GSM (more than two minutes per iteration). The number of iterations required by every tested algorithm depends on the type and amount of noise and on the image itself. Table 2 shows the evolution of the iterations with the amount of both AWGN and CCD noise for every tested method. This table can be used to have an estimation of the expected time of processing of an image by multiplying the time per iteration by the expected number of iterations. In addition, the evolution of the PSNR with the number of iterations for both types of noise is shown in Figure 4. It can be seen that NLM is the method with less variation after the maximum is reached. However, it is also shown that PDE,

Table 2: Average number of iterations required by the tested methods to attain the maximum PSNR for different standard deviations of AWGN noise (SDN) and amounts of CCD noise (AN).

AWGN					CCD noise				
SDN	NLM	PDE	GSM	TVD	AN (%)	NLM	PDE	GSM	TVD
5.0	1.00	1.00	1.00	1.00	2.5	1.40	1.29	1.00	1.33
10.0	1.02	1.12	1.00	1.01	5.0	2.71	2.93	1.20	2.47
15.0	1.08	1.91	1.01	1.03	7.5	3.90	4.67	2.23	3.98
20.0	1.16	2.73	1.05	1.21	10.0	4.97	6.12	3.50	5.15
25.0	1.44	3.79	1.35	1.75	12.5	6.04	7.40	4.88	6.31
30.0	1.62	4.53	1.85	2.12	15.0	7.02	8.40	5.98	7.40

GSM and TVD are able to reach higher values in CCD noise. The following subsections show the comparisons with AWGN and CCD camera noise.

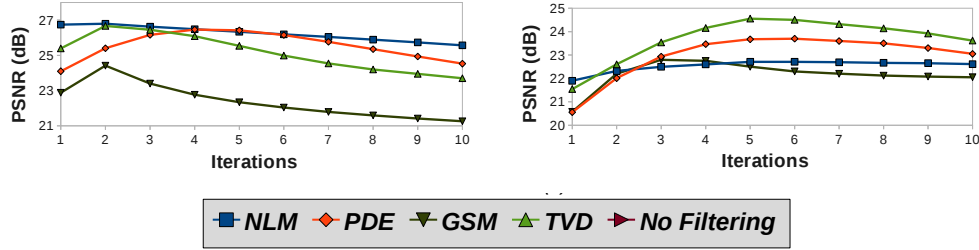


Figure 4: Evolution of PSNR for ten iterations. Left: evolution for AWGN (standard deviation = 30). Right: evolution for CCD noise (amount of noise = 10%).

7.1. Experiments with AWGN

Figure 5 shows the plots of PSNR, PSNRG and PSNRA vs. standard deviation of AWGN for NLM, PDE, GSM and TVD. According to the PSNR curve, NLM, PDE and TVD have almost the same performance for standard deviations of noise greater or equal than 10. NLM has the best performance for smaller amounts of noise followed by PDE. GSM only has a good performance for standard deviations of noise between 20 and 25.

According to the PSNRG and PSNRA curves, NLM, PDE and TVD have almost the same performance for standard deviations of noise greater than 10. That means that NLM, PDE and TVD have the same ability for preserving edges and not introducing artifacts in these scenarios.

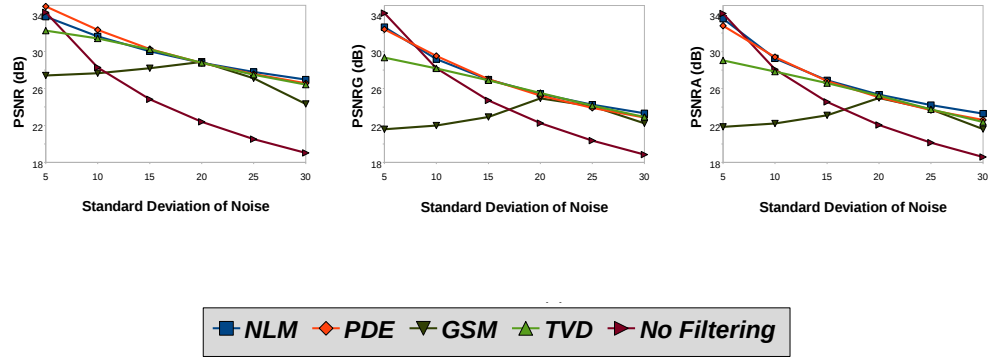


Figure 5: PSNR, PSNRG and PSNRA vs. standard deviation of AWGN for NLM, PDE, GSM and TVD.

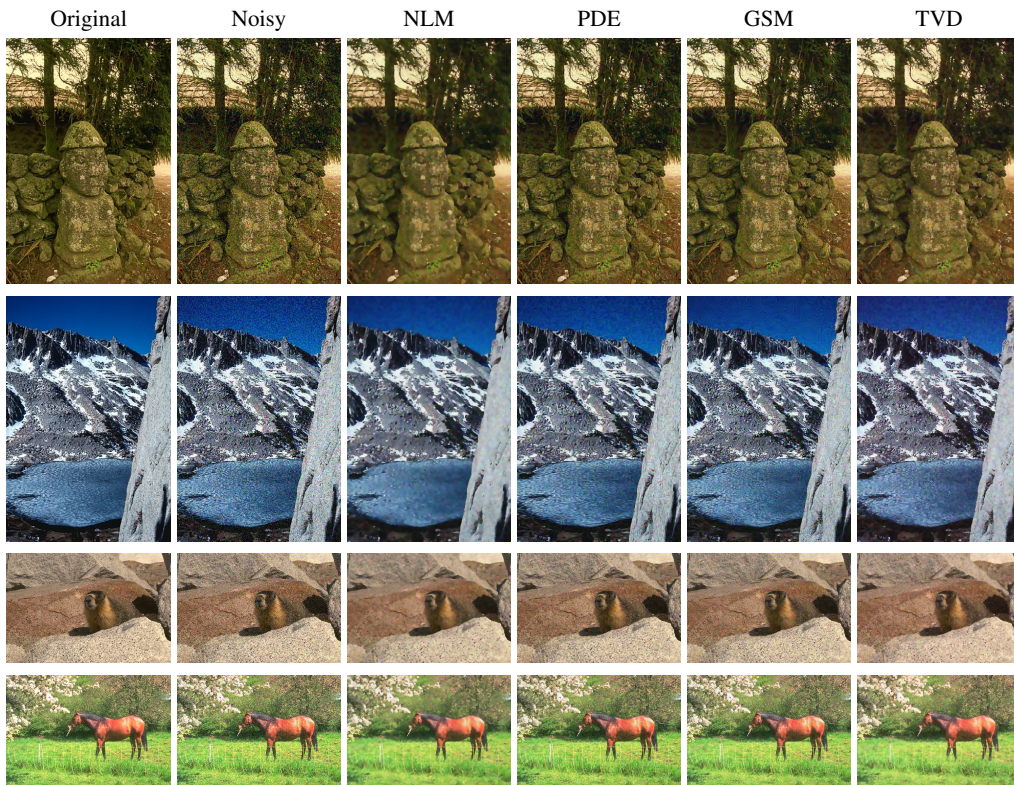


Figure 6: Denoising results for AWGN. The first column shows the original images. The second column shows the noisy images (standard deviation = 30). Columns three to six show the denoised images after applying NLM, PDE, GSM and TVD respectively.

NLM and PDE have the best performance for standard deviations of noise smaller than 15. GSM is only competitive for standard deviations of noise greater than 15. In summary, TVD has a competitive performance in PSNR, PSNRG and PSNRA with respect to AWGN, especially for highly noisy scenarios. Figure 6 shows some denoising results obtained with the four tested algorithms³. It can be seen that the denoised images are similar to each other. However, NLM and TVD introduce fewer artifacts.

An additional observation from the PSNR, PSNRG and PSNRA curves is that it is difficult for all the tested methods to remove noise near edges and to avoid the introduction of undesired artifacts. For example, the improvement in PSNR of the NLM for a standard deviation of noise of 30 is 7.94 dB, while it only attains 4.47 dB and 4.70 dB for PSNRG and PSNRA respectively.

Figure 7 shows the *Method Noise* for the images of Figure 6. It can be seen that NLM and TVD have the best performance, since their *Method Noise* images reveal little structure, which is not always the case for PDE and GSM.

7.2. Experiments with CCD Camera Noise

In order to obtain PSNR, PSNRG and PSNRA curves for real noise, it is necessary to generate synthetic noisy images from noiseless ones. Synthetic CCD camera noise was generated in the experiments as proposed by Liu *et al.* [1]. This methodology to generate synthetic CCD camera noise was chosen since it takes into account most of the sources of real noise for CCD cameras from a variety of brands.

Figure 8 shows the plots of PSNR, PSNRG and PSNRA vs. amount of CCD camera noise for NLM, PDE, GSM and TVD. Additionally, Table 3 shows the improvement in decibels for the tested algorithms, taking the PSNR of the noisy images as the baseline. The best performances are shown in bold.

According to the PSNR curve, NLM, PDE and TVD have almost the same performance for a noise of 2.5%. TVD has the best performance for larger amounts of noise followed by PDE. NLM and GSM have similar performances for amounts of noise greater than 5%. According to the PSNRG curve, NLM, PDE and the TVD have almost the same performance for amounts of noise greater than 2.5%. That means that NLM, PDE and TVD have the same ability to preserve edges in these scenarios. NLM and PDE have a better performance in PSNRG for an amount

³Full-resolution images are available at <http://deim.urv.cat/~rivi/tvd.html>

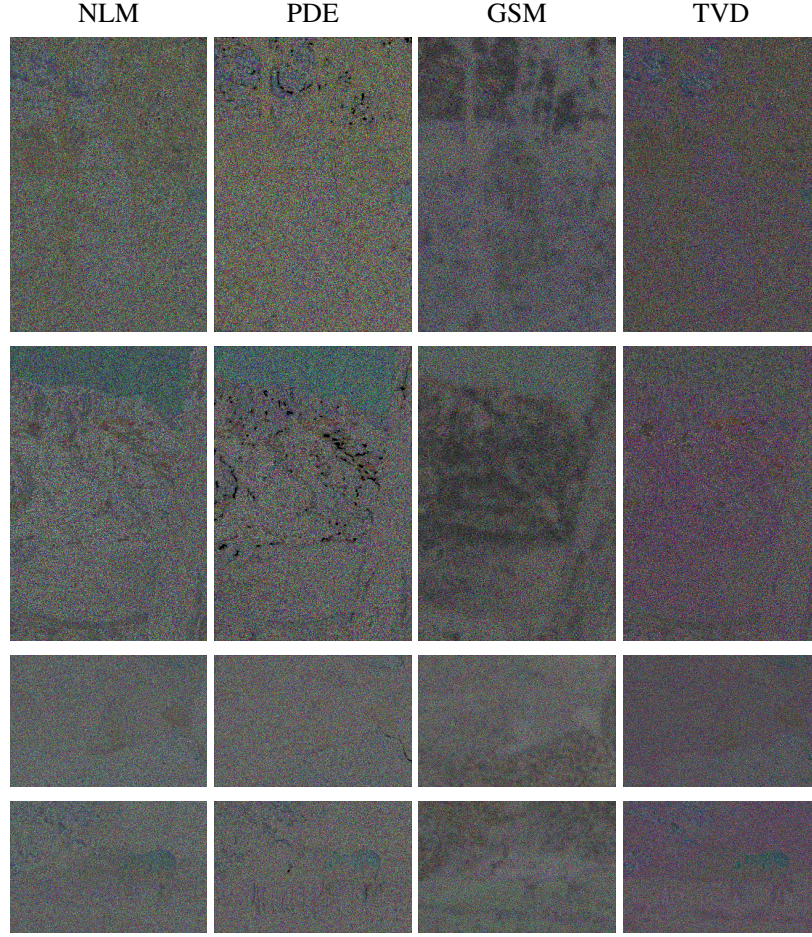


Figure 7: The *Method Noise* of NLM, PDE, GSM and TVD respectively for the images in Figure 6.

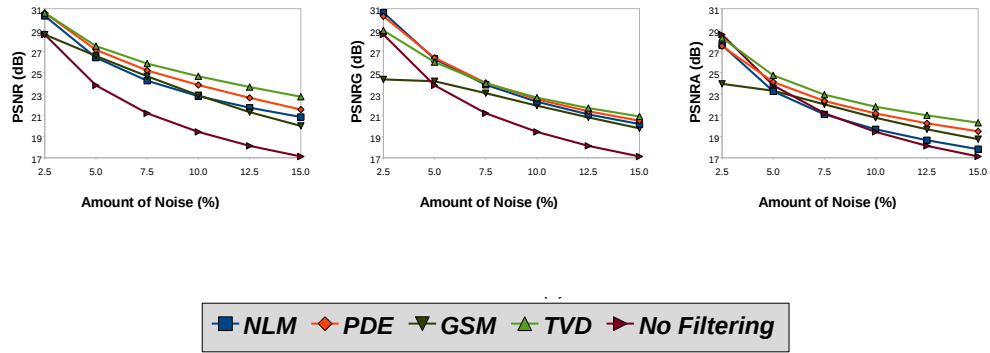


Figure 8: PSNR, PSNRG and PSNRA vs. amount of CCD camera noise generated as in Liu et al. [1] for NLM, PDE, GSM and TVD.

Table 3: PSNR, PSNRG and PSNRA improvements for CCD camera noise

Algorithm	amount of noise = 5%			amount of noise = 10%		
	PSNR	PSNRG	PSNRA	PSNR	PSNRG	PSNRA
NLM	2.61	2.48	-0.56	3.36	2.83	0.24
PDE	3.33	2.59	0.32	4.40	3.07	1.73
GSM	2.77	0.39	-0.51	3.44	2.47	1.33
TVD	3.69	2.18	0.95	5.22	3.22	2.37

of noise of 2.5%. GSM has the worst performance for all amounts of noise. GSM has a poor performance for a noise of 2.5% since the PSNRG is below the baseline curve. This means that GSM is unable to preserve edges in this case. According to the PSNRA metric, TVD has the best performance for all amounts of noise. NLM and GSM have a poor performance in PSNRA as their curves appear near or even below the baseline curve for amounts of noise below 10%. This means that NLM and GSM cannot avoid introducing artifacts for amounts of noise below 10%. In general, NLM has a worse performance than GSM, since GSM improves its performance for high amounts of noise.

Table 3 shows that TVD outperforms the other algorithms in almost all the cases according to the PSNR, PSNRG and PSNRA metrics. A negative value indicates that the filtered images are worse than the noisy ones. That condition appears in some cases for NLM and GSM. An interesting observation is that, in general, the algorithms for CCD camera noise yield lower values of PSNRG than PSNR, and lower values of PSNRA than PSNRG for the same amounts of noise. This means that it is more difficult for the tested algorithms to preserve edges than to reduce noise, and even much more difficult to avoid introducing artifacts than to preserve edges.

Figure 9 shows some close-ups of the denoising results obtained with the four tested algorithms⁴. It can be seen that NLM generates undesirable quantization artifacts and colored spots. In addition, PDE generates cross-shaped artifacts. GSM partially removes noise. TVD produces better results since it generates fewer artifacts than the other algorithms.

⁴Full-resolution images are available at <http://deim.urv.cat/~rivi/tvd.html>

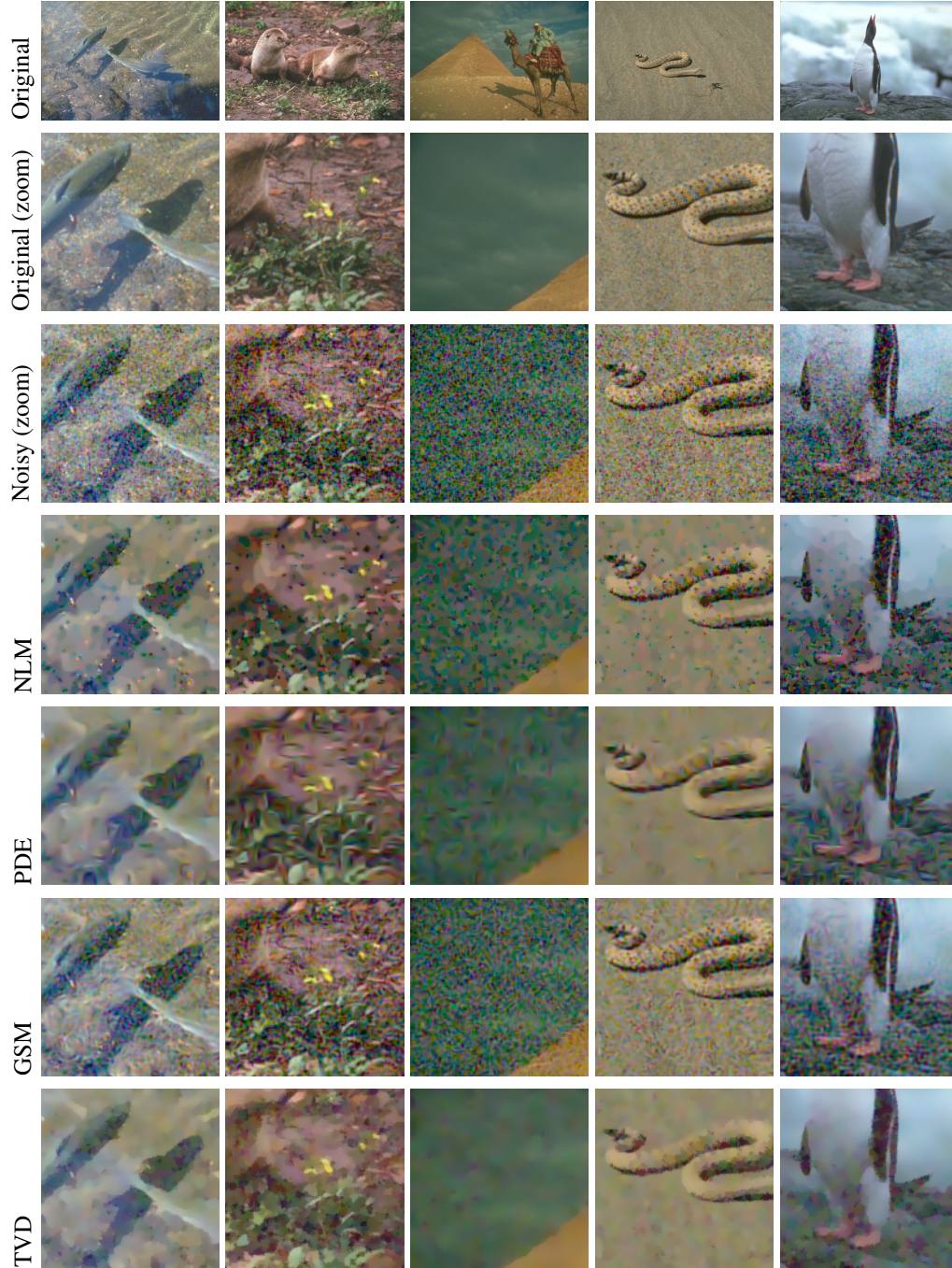


Figure 9: Denoising results. The first row shows the original images. The second row shows the noisy images (10% of CCD camera noise). Rows three to six show close-ups of the denoised images after applying NLM, PDE, GSM and TVD respectively.

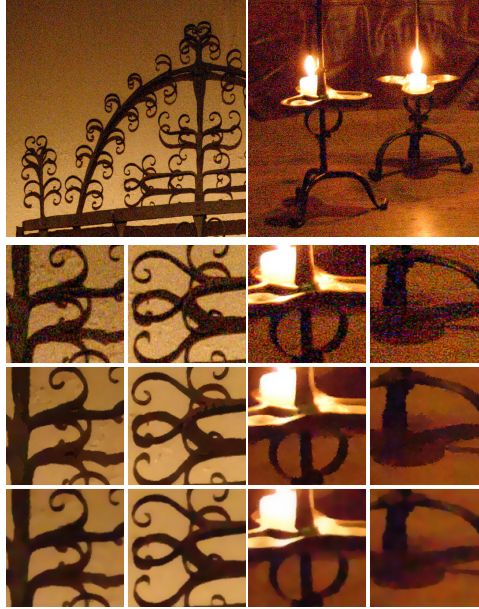


Figure 10: Comparison with [1]. The first row shows real noisy images taken from [1]. The second row shows some close-ups of the same images. The third and fourth rows show the denoising results of [1] and the TVD respectively. High-resolution images of the first, second and third rows were directly taken from the electronic version of [1].

7.3. Additional Experiments

Since no source code is available from the authors of the methods presented in [1] and [35], those techniques have not been able to be tested upon the same set of images as the other techniques. However, in order to compare their performance with TVD, the latter has been applied to the same high-resolution noisy images used in [1] and [35] respectively. Figures 10 and 11 show the qualitative comparison between the resulting images processed by TVD, and the images reported in the respective papers. Figure 10 shows that TVD has a better performance than the Liu et al.’s [1] method in real noisy images, since TVD generates fewer quantization artifacts and fewer saw-shaped artifacts in edges. Figure 11 shows that TVD outperforms the method presented in [35] on salt and pepper noisy images, since TVD also generates fewer artifacts in edges.

7.4. Parameter Selection

Figure 12 can be used to visualize the effect of the parameters of TVD on the results. First, images a)-c) show that increasing σ_s could lead to loss of small structures surrounded by bigger

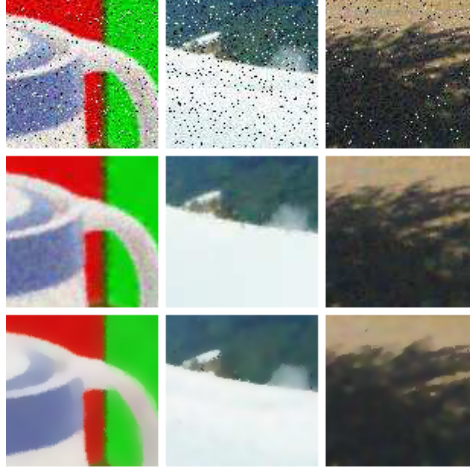


Figure 11: Comparison with [35]. The first row shows images contaminated with salt and pepper noise. The second row shows the denoising results reported in [35]. The third row shows the results obtained with TVD. High-resolution images of the first and second row were directly taken from the electronic version of [35].

ones (e.g., the glitter in the mandrill’s nose or the wrinkles in its cheeks). However, homogeneous regions are less affected by such an increase (e.g., the whiskers). On the other hand, small values of σ_s will make it necessary more iterations in very noisy scenarios. Second, images d)-f) show that big values of σ_d generate blurring (e.g., Lenna’s hat). In addition, small values of σ_d will make it necessary more iterations in very noisy scenarios. Finally, images g)-i) show that small details of texture are blurred with $J_L = 0$. However, a small J_L can be useful to increase the smoothing effect of the filter. It is interesting to remark that images h) and i) are exactly the same. This effect is mainly caused by the discontinuity in (13). Thus, every image has a limit value of J_L beyond which this parameter will have no effect on the denoising process, since F_{η_m} will become one for every pixel in the image. However, every image has a different limit value of J_L , since it depends on the amount of perceived textured in the image. As an example, according to our experiments, the limit value of J_L for the aerial image in Figure 12 is around 6.5.

8. Concluding Remarks

A new method to denoise color images while preserving edges and texture has been proposed. This method is based on an adaptation of the tensor voting framework originally proposed in [2] for surface reconstruction. The proposed adaptation of the TVF takes into account a set of criteria

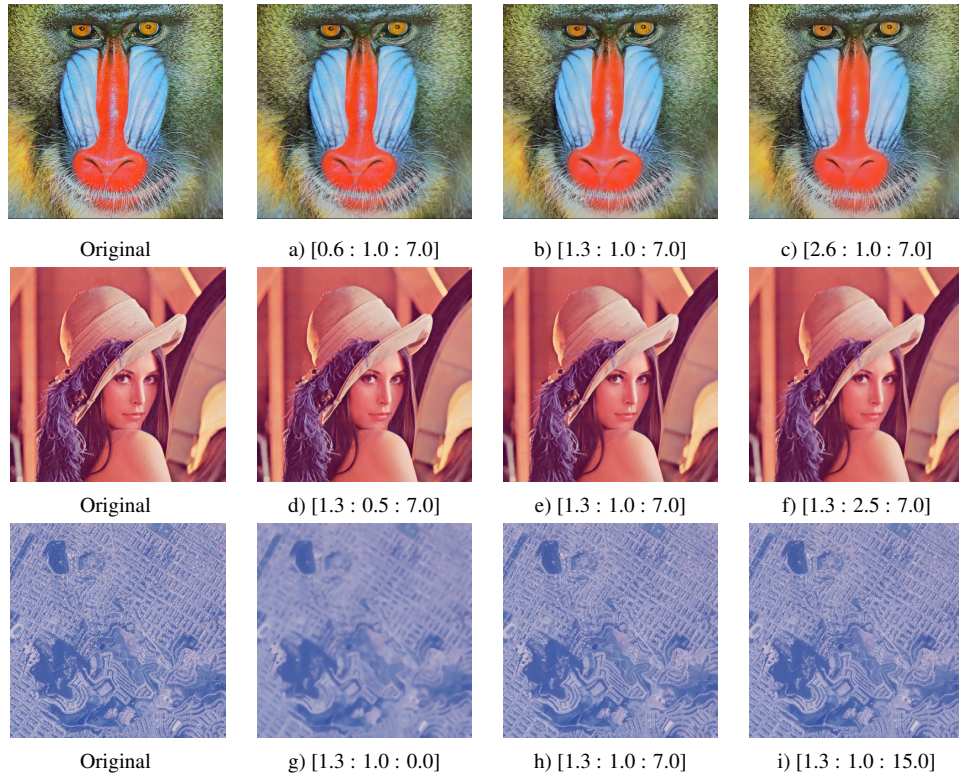


Figure 12: Effect of different values of the parameters of TVD. Two iterations of TVD have been run in order to enhance the effect of those parameters. The used parameters are shown below every image as the triplet $[\sigma_s : \sigma_d : J_L]$.

inspired by the perceptual process of image denoising. An optimized version of CIEDE2000 has been proposed to measure color differences in noisy environments by modifying its parameters k_L , k_C and k_H .

The results presented in this paper show that the use of a specific voting process makes the TVF a powerful tool for image denoising. Measurements of removed noise, edge preservation and undesirable introduced artifacts, additionally to visual inspection have been used to compare the performance of the TVD against some of the state-of-the-art color image denoising algorithms. The two new metrics, PSNRG and PSNRA, have been proposed to measure the edge preservation and undesirable artifact rejection features of the algorithms respectively, while PSNR has been used to measure the amount of removed noise. These experiments show that TVD has a similar performance to the other tested algorithms with respect to AWGN noise, and better than them regarding CCD camera noise, especially in cases with large amounts of noise. This means that, TVD is more appropriate than the other tested methods for real applications, which is the final objective of any image denoiser. The performance of TVD also shows that perception-based techniques are an effective alternative to traditional approaches for color image denoising. Thus, this result could encourage future work on adapting other perceptual techniques to this field. In addition, the new metrics PSNRG and PSNRA have been utilized to show that it is more difficult for denoising algorithms to preserve edges and to avoid the introduction of artifacts than to reduce noise.

Future work will include a validation of the proposed methodology from a psychophysical perspective in other scenarios (e.g. indoor images), as well as the utilization of the TVF for region segmentation of noisy color images. In addition, correlations between the human visual system and the new proposed PSNRG and PSNRA metrics will be studied. Such correlations are expected since the human visual system is more prone to focus on edges and artifacts than on flat regions. Finally, the execution time of TVD is expected to be significantly reduced, since the reported execution times are based on a non-optimized implementation of the method.

Acknowledgments

This research has been partially supported by the Spanish Ministry of Science and Technology under project DPI2007-66556-C03-03, by the Commissioner for Universities and Research of the Department of Innovation, Universities and Companies of the Catalanian Government and

by the European Social Fund.

References

- [1] C. Liu, R. Szeliski, S. B. Kang, C. Zitnick, W. Freeman, Automatic estimation and removal of noise from a single image, *IEEE Trans. Pattern Anal. Mach. Intell.* 30 (2) (2008) 299–314.
- [2] G. Guy, G. Medioni, Inference of surfaces, 3D curves and junctions from sparse, noisy 3D data, *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (11) (1997) 1265–1277.
- [3] S. Mahamud, L. R. Williams, K. K. Thornber, K. Xu, Segmentation of multiple salient closed contours from real images, *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (4) (2003) 433–444.
- [4] J. S. Stahl, S. Wang, Globally optimal grouping for symmetric closed boundaries by combining boundary and region information, *IEEE Trans. Pattern Anal. Mach. Intell.* 30 (3) (2008) 395–411.
- [5] D. G. Lowe, *Perceptual Organization and Visual Recognition*, Kluwer Academic Publishers, 1985.
- [6] L. Loss, G. Bebis, M. Nicolescu, A. Skurikhin, An iterative multi-scale tensor voting scheme for perceptual grouping of natural shapes in cluttered backgrounds, *Comput. Vis. Image Underst.* 113 (1) (2009) 126–149.
- [7] G. Medioni, M. S. Lee, C. K. Tang, *A Computational Framework for Feature Extraction and Segmentation*, Elsevier Science, 2000.
- [8] C. K. Tang, G. Medioni, M. S. Lee, N-Dimensional tensor voting and application to epipolar geometry estimation, *IEEE Trans. Pattern Anal. Mach. Intell.* 23 (8) (2001) 829–844.
- [9] X.-F. Shao, L.-W. Ye, M.-J. Cai, Y. Wang, Simultaneous image denoising and curve extraction by tensor voting, in: *Proc. Int. Symp. Comput. Sci. and Comput. Tech.*, 2008, pp. I: 536–538.
- [10] M. Nicolescu, G. Medioni, A voting-based computational framework for visual motion analysis and interpretation, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (5) (2005) 739–752.
- [11] C. Min, G. Medioni, Inferring segmented dense motion layers using 5D tensor voting, *IEEE Trans. Pattern Anal. Mach. Intell.* 30 (9) (2008) 1589–1602.
- [12] R. Moreno, M. Garcia, D. Puig, L. Pizarro, B. Burgeth, J. Weickert, On improving the efficiency of tensor voting, *IEEE Trans. Pattern Anal. Mach. Intell.* Early Access.
- [13] R. C. Gonzalez, R. E. Woods, *Digital image processing*, 3rd Edition, Prentice Hall, Inc., 2007.
- [14] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in: *Proc. IEEE Int. Conf. Comput. Vis.*, 1998, pp. 839–846.
- [15] A. Buades, B. Coll, J. M. Morel, A non-local algorithm for image denoising, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. II:60–65.
- [16] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Anal. Mach. Intell.* 12 (7) (1990) 629–639.
- [17] S. Roth, M. J. Black, Fields of experts: a framework for learning image priors, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. II:860–867.
- [18] D. Tschumperlé, Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's, *Int. J. Comput. Vis.* 68 (1) (2006) 65–82.
- [19] O. Lezoray, A. Elmoataz, S. Boughleux, Graph regularization for color image processing, *Comput. Vis. Image Underst.* 107 (1-2) (2007) 38–55.

- [20] S. C. Zhu, D. Mumford, Prior learning and Gibbs reaction-diffusion, *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (11) (1997) 1236–1250.
- [21] R. Lukac, K. N. Plataniotis, A. N. Venetsanopoulos, Color image denoising using evolutionary computation, *Int. J. Imaging Syst. Technol.* 15 (5) (2006) 236–251.
- [22] D. L. Donoho, I. M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, *Biometrika* 81 (3) (1994) 425–455.
- [23] J. Portilla, V. Strela, M. Wainwright, E. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, *IEEE Trans. Image Process.* 12 (11) (2003) 1338–1351.
- [24] M. M. Ichir, A. Mohammad-Djafari, Hidden markov models for wavelet-based blind source separation, *IEEE Trans. Image Process.* 15 (7) (2006) 1887–1899.
- [25] N.-X. Lian, V. Zagorodnov, Y.-P. Tan, Edge-preserving image denoising via optimal color space projection, *IEEE Trans. Image Process.* 15 (9) (2006) 2575–2587.
- [26] A. Tonazzini, L. Bedini, E. Salerno, A markov model for blind image separation by a mean-field EM algorithm, *IEEE Trans. Image Process.* 15 (2) (2006) 473–482.
- [27] H. Yu, L. Zhao, H. Wang, Image denoising using trivariate shrinkage filter in the wavelet domain and joint bilateral filter in the spatial domain, *IEEE Trans. Image Process.* 18 (10) (2009) 2364–2369.
- [28] O. Ben-Shahar, S. W. Zucker, Hue geometry and horizontal connections., *Neural Netw.* 17 (5-6) (2004) 753–771.
- [29] V. Bruce, P. R. Green, M. A. Georgeson, *Visual Perception: physiology, psychology and ecology*, 4th Edition, Psychology Press, 2003.
- [30] E. Kang, G. Medioni, Color image segmentation based on tensor voting, in: *Third Workshop Percept. Organ. Comput. Vis.*, 2001.
- [31] A. Massad, B. M., B. Mertsching, Application of the tensor voting technique for perceptual grouping to grey-level images, in: *Proc. DAGM Symp. Pattern Recognit., Lect. Notes Comput. Sci.* 2449, 2002, pp. 306–313.
- [32] J. Jia, C. K. Tang, Inference of segmented color and texture description by tensor voting, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (6) (2004) 771–786.
- [33] Y.-W. Tai, W.-S. Tong, C.-K. Tang, Perceptually-inspired and edge-directed color image super-resolution, in: *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. II:1948–1955.
- [34] J. Lim, J. Park, G. Medioni, Text segmentation in color images using tensor voting, *Image and Vis. Comput.* 25 (5) (2007) 671–685.
- [35] Y.-W. Tai, W.-S. Tong, C.-K. Tang, Simultaneous image denoising and compression by multiscale 2D tensor voting, in: *Proc. Int. Conf. Pattern Recognit.*, 2006, pp. III:818–821.
- [36] L. Rittner, F. C. Flores, R. A. Lotufo, A tensorial framework for color images, *Pattern Recognit. Lett.* 31 (4) (2010) 277–296.
- [37] J. Wen, X. Gao, Y. Yuan, D. Tao, J. Li, Incremental tensor biased discriminant analysis: A new color-based visual tracking method, *Neurocomput.* 73 (4-6) (2010) 827–839.
- [38] S. Bourennane, C. Fossati, A. Cailly, Improvement of classification for hyperspectral images based on tensor modeling, *IEEE Geosci. Remote Sens. Lett.* 7 (4) (2010) 801–805.
- [39] D. Muti, S. Bourennane, Survey on tensor signal algebraic filtering, *Signal Process.* 87 (2) (2007) 237–249.
- [40] D. Tao, S. Maybank, W. Hu, X. Li, Stable third-order tensor representation for colour image classification, in: *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, 2005, pp. 641–644.

- [41] X. He, D. Cai, H. Liu, J. Han, Image clustering with tensor representation, in: Proc. ACM Int. Conf. Multimed., 2005, pp. 132–140.
- [42] M. D. Fairchild, Color Appearance Models, 2nd Edition, Wiley-IS&T, 2005.
- [43] R. W. G. Hunt, The Reproduction of Colour, sixth Edition, Wiley-IS&T, 2004.
- [44] R. D. Gow, D. Renshaw, K. Findlater, L. Grant, S. J. McLeod, J. Hart, R. L. Nicol, A comprehensive tool for modeling CMOS image-sensor-noise performance, IEEE Trans. Electron Devices 54 (6) (2007) 1321–1329.
- [45] G. Wyszecki, W. Stiles, Color Science: Concepts and Methods, Quantitative Data and Formulae, 2nd Edition, John Wiley and Sons, 2000.
- [46] M. R. Luo, G. Cui, B. Rigg, The development of the CIE 2000 colour-difference formula: CIEDE2000, Color Res. Appl. 26 (5) (2001) 340–350.
- [47] R. Berns, The science of digitizing paintings for color-accurate image archives: A review, J. Imaging Sci. Technol. 45 (4) (2001) 305–325.
- [48] R. Kuehni, CIEDE2000, milestone or final answer?, Color Res. Appl. 27 (2) (2002) 126–128.
- [49] R. Huertas, M. Melgosa, E. Hita, Influence of random-dot textures on perception of suprathreshold color differences., J. Opt. Soc. Am. A 23 (9) (2006) 2067–2076.
- [50] C.-H. Chou, K.-C. Liu, A fidelity metric for assessing visual quality of color images, in: Proc. Int. Conf. Comput. Commun. Netw., 2007, pp. 1154–1159.
- [51] C. Kervrann, J. Boulanger, Local adaptivity to variable smoothness for exemplar-based image regularization and representation, Int. J. Comp. Vis. 79 (1) (2008) 45–69.
- [52] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proc. IEEE Int. Conf. Comput. Vis., 2001, pp. II:416–423.
- [53] Y. Tsai, V. Ramesh, T. Kanade, Statistical calibration of CCD imaging process, in: Proc. IEEE Int. Conf. Comput. Vis., 2001, pp. I:480–487.

Biographies

Rodrigo Moreno

Rodrigo Moreno received B.S. and M.S. degrees in computer science from the University of Los Andes (Colombia) in 1995 and 1997 respectively; a B.S. degree in electrical engineering from the National University of Colombia in 1995; a M.S. degree in organizations management from the University of Quebec (Canada) in 2006; and a Diploma in Advanced Studies and his Ph.D. degree from the Polytechnic University of Catalonia (Spain) in 2007 and 2010 respectively. Since 2010, he is a Post-Doctoral Researcher at the Center for Medical Image Science and Visualization at Linköping University (Sweden). His research interests include the application of perception-based techniques and tensorial methods to image analysis, computer vision, and biomedical images.

Miguel Angel Garcia

Miguel A. Garcia received the B.S., M.S., and Ph.D. degrees in computer science from Polytechnic University of Catalonia, Barcelona, Spain, in 1989, 1991, and 1996, respectively. He joined the Department of Software at Polytechnic University of Catalonia in 1996 as an Assistant Professor. From 1997 to 2006, he was with the Department of Computer Science and Mathematics at Rovira i Virgili University, Tarragona, Spain, where he was the Head of Intelligent Robotics and Computer Vision Group. In 2006, he joined the Department of Informatics Engineering at Autonomous University of Madrid, Spain, and in 2010 the Department of Electronic and Communications Technology, at Autonomous University of Madrid, Spain, where he is currently Associate Professor and member of the Video Processing and Understanding Lab. His research interests include image processing, computer vision, 3-D modeling, and mobile robotics.

Domenec Puig

He received his M.S. and Ph.D. degrees in computer science from Polytechnic University of Catalonia in 1992 and 2004 respectively. In 1992, he joined the Department of Computer Science and Mathematics at Rovira i Virgili University, where he is currently Associate Professor. Since July 2006, he is the Head of the Intelligent Robotics and Computer Vision group at the same university. His research interests include perceptual models for image analysis, texture analysis, scene analysis, and mobile robotics.

Carme Julià

She received her B.S. in Mathematics in 2002 from Polytechnic University of Catalonia, and her M.S. and Ph.D. in computer science from Autonomous University of Barcelona in 2004 and 2008 respectively. Since September 2008 she is a member of the Intelligent Robotics and Computer Vision Group at Rovira i Virgili University. Her research interests include image processing, perceptual models for image analysis, photometric stereo, and structure from motion.