Texture Classification through Multiscale Orientation Histogram Analysis

Miguel Alemán-Flores and Luis Álvarez-León

Departamento de Informática y Sistemas
Universidad de Las Palmas de Gran Canaria
35017 Las Palmas, Spain
{maleman,lalvarez}@dis.ulpgc.es

Abstract. This work presents a new approach to texture classification, in which orientation histograms and multiscale analysis have been combined to achieve a reliable method. From the outputs of a set of filters, the orientation and magnitude of the gradient in every point of a texture are estimated. By combining the orientations and relative magnitudes of the gradient, we build an orientation histogram for each texture. We have used Fourier analysis to measure the similarity between the histograms of different textures, considering the effects of a change in the size or orientation of the image to make our method invariant under these phenomena. Since different textures may generate very similar histograms, we have analyzed the evolution of these histograms at different scales, extracting a scale factor for each couple of compared textures to adjust the filters which are applied to them when the multiscale analysis is carried out.

1 Introduction

The visual identification of an object is not only provided by its shape. The texture in the inner region may be helpful to a large extent when we try to characterize materials, components, agglomerations, etc. Sonka et al. [1] define a texture as something consisting of mutually related elements. A texture consists of texture primitives or texture elements, sometimes called *texels* and, due to its wide variability, it is not simple to give a precise definition. An important problem when dealing with textures is that texture description is scale dependent. We may deal with the problem of texture classification from many different points of view, but we must take into account that the scale of the textures we are comparing is a crucial factor when measuring their similarity. The distribution of orientations has been previuosly used in [2] for the discrimination between city and suburb photos according to the presence or absence of dominant orientations. Other works have shown the results of different filters used in texture classification when applied to a certain texture benchmark set [3] and the evaluation of dissimilarity measures for color and texture [4]. A complete analysis on texture-related problems and applications, considering aspects like texture classification, segmentation or synthesis is shown in [5].

$F_0 : 0$			$F_1 : \pi/4$			$F_2 : \pi/2$			$F_3 : 3\pi/4$		
1	1	−2	1	−2	−4	−2	−4	−2	−4	−2	1
2	2	−4	1	2	−2	1	2	1	−2	2	1
1	1	−2	2	1	1	1	2	1	1	1	2

$F_4 : \pi$			$F_5 : 5\pi/4$			$F_6 : 3\pi/2$			$F_7 : 7\pi/4$		
−2	1	1	1	1	2	1	2	1	2	1	1
−4	2	2	−2	2	1	1	2	1	1	2	−2
−2	1	1	−4	−2	1	−2	−4	−2	1	−2	−4

The purpose of this work is presenting an approach to texture classification based on the description given by the estimation of the orientation in the points within a textured region. Therefore, we must first extract a value for the orientation and magnitude of the gradient in every point within the region we are examining. From these estimations, we build an orientation histogram for each texture, which represents the distribution of the orientations.

Since an orientation histogram does not univocally characterize a texture, the same representation could have been extracted from different patterns. However, the study of texture histograms at many different scales allows comparing the evolution and interaction of the gradients, so that textures which are initially considered as very similar start to evolve in a quite different manner.

The paper has been structured as follows: Section 2 shows how the orientation of the edges can be estimated from the outputs of a set of filters. In Sect. 3, these estimations are used to build orientation histograms which describe the textures in terms of quantitative edge orientation distribution and the representation of the textures through these orientation histograms allows classifying them. The multiscale analysis of the textures is introduced in Sect. 4 to generate better texture classifications by comparing them at different scales. Section 5 shows how darkening, lightening and inverting the images affect the classification, which proves robust under these transformations. Finally, Sect. 6 shows some conclusions about this work.

2 Edge Orientation Estimation

From Newton filters [6][7], we have developed a set of filters which preserve their convenient properties, but which also avoid some of the undesirable phenomena by providing them with rotational invariance and non-null weights in all positions. The weights of the eight filters and the orientations they react to are shown in Table 1.

The output of these filters is independent of the particular gray value of the image border, i.e. F_k is invariant under a gray level translation as $I \rightarrow I + C$, where I contains the gray values of the image and C is any constant. This property is very important because the relevant information is provided by

the difference between neighbors, rather than the magnitude of the image gray values.

Instead of considering the maximum of the eight filters to assign a value to the orientation, a more accurate estimation can be obtained by considering the whole pattern provided by the eight filters. The output of these eight filters for a $\pi/2$ oriented edge of magnitude 1 is $(0, 5, 8, 5, 0, -4, -4, -4)$. If we increase or decrease the orientation in a multiple of $\pi/4$, the output is only cyclically shifted, but the values and their order are not altered. Figure 1 shows the outputs of the eight filters for a circle. When the real orientation does not correspond to one of these directions, we can estimate it by interpolating the higher value with its two neighbors, which provides an accurate estimation of edge orientation. A quadratic function y is built to interpolate these three values and its maximum is used as the estimated orientation, as shown in (1) and (2), where i is the index of the filter with the highest output, F_i, positions $i-1$ and $i+1$ are calculated modulo 8 and $x_{\max}$ is the estimation of the orientation for the current point:

$$
\begin{aligned}
y = {} & \frac{8F_{i+1} - 16F_i + 8F_{i-1}}{\pi^2} \left(x - \frac{\pi(i-1)}{4} \right)^2 \\
& + \frac{8F_i - 2F_{i+1} - 6F_{i-1})}{\pi} \left(x - \frac{\pi(i-1)}{4} \right) + F_{i-1} \ .
\end{aligned}
\tag{1}
$$

$$
x_{\max} = \left(\frac{4F_i - 3F_{i-1} - F_{i+1}}{2 \left[2F_i - F_{i-1} - F_{i+1} \right]} + i - 1 \right) \frac{\pi}{4} \ .
\tag{2}
$$

By correlating the ideal pattern with the real one, we can determine how perfect the border is. As the change from one side of the border to the other one may be larger than 1, it is necessary to normalize the output. Thus, the main advantage of this kind of filters is not the location of edges, but their classification according to their orientation and the invariance under rotations and illumination changes.

Similar filters have been proposed by Prewitt, Sobel, Robinson or Kirsch [1], but they cause the duplication of edges, are independent of the central value or may produce the maximum output for imperfect edges. Furthermore, they do not constitute a set of linearly independent filters, as modified Newton filters do, and the information they provide is not complete.

3   Orientation-Based Texture Classification

With the modified Newton filters we are able to estimate accurately the orientation of the gradient in a certain pixel. Similarly, the values for the eight main orientations can be used to estimate the magnitude of this gradient. The interpolation of the outputs of the filters will provide us with a value for the direction of the gradient in every point as well as an estimation of its magnitude.
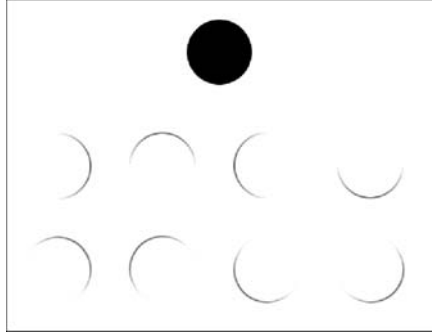
Fig. 1. Positive outputs of the modified Newton filters for the circle above (the higher the output value, the darker its representation)
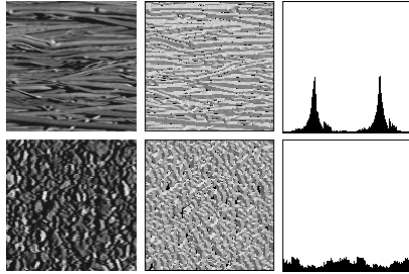

Fig. 2. Texture examples and corresponding orientation maps and orientation histograms

With these estimations, we build a histogram of the orientations by adding the magnitude of the gradient in the points where the edges present the same orientation. This histogram describes how the orientations are distributed in that region, allowing us to determine the most significant ones, the proportion they represent and their relation in terms of orientation distance and concentration. Figure 2 shows an example of two textures and their corresponding orientation histograms, which can be used to compare these textures with others.

Using the interpolation polynomial in (1), the maximum value $o_{j,k}$, which estimates the orientation in point (j, k), is given by:

$$o_{j,k} = round\left(\frac{\left(\frac{4F_i - 3F_{i-1} - F_{i+1}}{2[2F_i - F_{i-1} - F_{i+1}]}\frac{\pi}{4} + \frac{\pi(i-1)}{4}\right)L}{2\pi}\right) \ .$$ (3)

It has been rounded to adjust it to a discrete signal consisting of L equidistant values. For the magnitude, we use this orientation and substitute it in the polynomial:

$$(\nabla I)_{j,k} = \frac{8F_{i+1} - 16F_i + 8F_{i-1})}{\pi^2} \left(o_{j,k} - \frac{\pi(i-1)}{4}\right)^2$$

$$+\frac{8F_i - 2F_{i+1} - 6F_{i-1}}{\pi} \left(o_{j,k} - \frac{\pi(i-1)}{4}\right) + F_{i-1} \quad . \tag{4}$$

The values of the histogram h_i are given by the following expression, where $(\nabla I)_{j,k}$ and $o_{j,k}$ are the magnitude and the orientation extracted for point (j,k). For normalization purposes, the global weight of all positions in the histogram is set to 1, thus dividing each resulting component of the histogram by the sum of all of them:

$$h_i = \sum_{\substack{j,k \\ o_{j,k}=i}} (\nabla I)_{j,k} \qquad \text{and} \qquad h'_i = \frac{h_i}{\sum\limits_{j=0}^{L-1} h_j} \quad . \tag{5}$$

In order to relate two textures, an energy function is built, in which the Fourier coefficients of both histograms are compared. We must achieve rotational invariance, in the sense that the result must not be affected if the textures are rotated. A change in the orientation of a texture will only cause a cyclical shift in the histogram. For this reason, the Fourier coefficients are modified as follows: let f_n and g_n be the orientation histograms of length L corresponding to the same texture but shifted a positions, i.e. the texture has been rotated an angle $\theta = 2\pi a/L$, and let f_k and g_k be the kth Fourier coefficients of these histograms, then $f_k = g_k e^{-i\frac{2\pi ka}{L}}$. Thus, a measure of how similar the coefficients of both textures are is given by:

$$E(a) = \sum_{k=1}^{\frac{L}{2}} \left(f_k - g_k e^{-i\frac{2\pi ka}{L}}\right) \left(f_k - g_k e^{-i\frac{2\pi ka}{L}}\right)^* \quad . \tag{6}$$

In addition, the fact that the number of discrete orientations used for the histograms is constant and the normalization of the weights make the lengths of the signals and the total weight equal in both textures. Consequently, a change in the size of the region where the texture is analyzed will not cause the generation of a different distribution. Due to the fact that the higher frequencies are more affected by noise than the lower ones, a monotonic decreasing weighting function $w(.)$ can be used to emphasize the discrimination, thus obtaining the following expression, in which the first terms have a more important contribution than the last ones. The minimization of this function will provide the shift for which both histograms present the best matching, i.e. the rotation which makes both textures as similar as possible. The energy for that value is a measure of how similar they are:

$$E(a) = \sum_{k=1}^{\frac{L}{2}} w\left(\frac{2k}{L}\right) \left(f_k - g_k e^{-i\frac{2\pi ka}{L}}\right) \left(f_k - g_k e^{-i\frac{2\pi ka}{L}}\right)^* \quad . \tag{7}$$

We have used different linear, quadratic and exponential weighting functions and the best results were obtained when $w(x) = e^{-10x}$. To test this technique, we have applied it to a set of textures contained in a database, which is shown in Fig. 3. This database has been made publicly available for research purposes by Columbia and Utrecht Universities, Columbia-Utrecht Reflectance and Texture Database [8]. We work with grayscale images and thus, a single histogram is used to represent the orientations of the edges in light intensity. Using the techniques described in the previous sections, a certain texture is compared with all those in the database and the most similar ones are selected. The similarity between two textures is given by the energy obtained when comparing their orientation histograms.

In Fig. 4 and Table 2, we show some results of the application of the technique explained above. From the image database containing 60 textures of different natures, but visually difficult to classify, one is selected, and the 5 best comparisons are shown. Of course, as the selected image belongs to the set, the best match corresponds to itself, and the energy factor is 0.

Table 2. Lowest energy values for textures 51 and 11

order	txt. number	wtd. energy	order	txt. number	wtd. energy
1	51	0.00	1	11	0.00
2	40	16.03	2	30	0.59
3	38	49.21	3	49	0.61
4	56	118.39	4	10	1.20
5	57	157.10	5	26	1.62

As mentioned above, the orientation histograms extracted from the textures describe how the different orientations are quantitatively distributed across the region which is studied, but they do not provide any information about the spatial neighborhood of the pixels with a certain orientation. Thus, a completely noisy image, in which all orientations are equally but disorderly present in the image would generate a similar histogram than a circle, where the orientation is gradually increased along its outline. This forces us to search for a certain technique which complements the information provided by this kind of histograms in order to enhance the recognition capabilities.

A multiscale analysis of the images will provide us with a series of images which represent the evolution of each texture at different scales. In this evolution, the orientations will be differently affected by the others, depending on their spatial proximity. This will allow us to distinguish among textures where orientations are originally distributed in a similar way, but which are actually different.
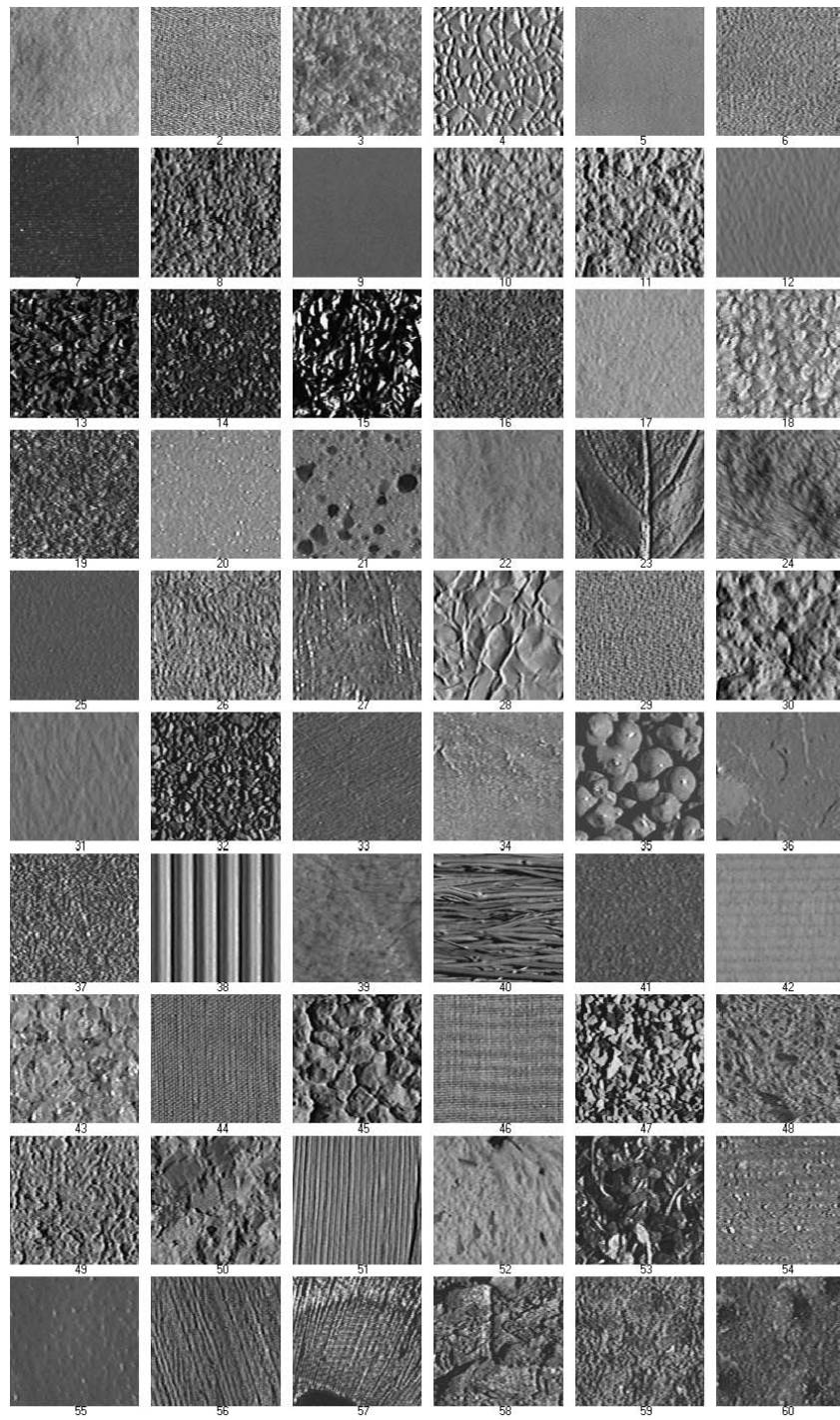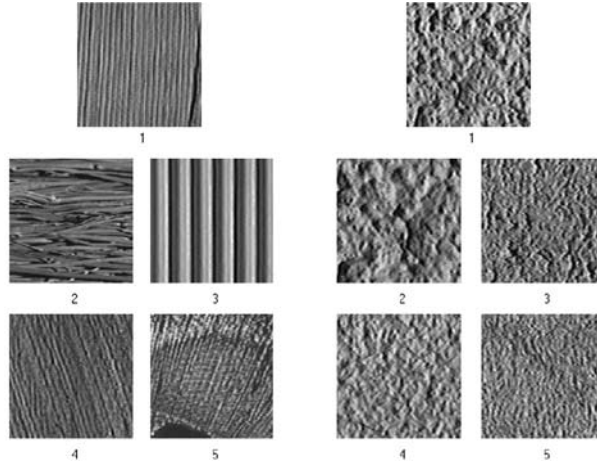
Fig. 3. Database textures

Fig. 4. Results of searching for similar textures for textures 51 and 11 using (7)

4 Texture Classification through Multiscale Analysis

The interpretation of the information we perceive from the environment depends on the scale we use to process it. At the same time, the information provided by each scale is useful and the study of the same scene at different scales makes it possible to perceive a wider range of realities. Furthermore, elements which are not distinguishable at a certain scale may be clearly distinct at a different one and the rough and detailed information extracted from an image may help us decide when comparing textures. The multiscale analysis approach has been successfully used in the literature for texture enhancement and segmentation (see [9] and [10] for more details).

A multiscale analysis can be determined by a set of transformations $\{T_t\}_{t \geq 0}$, where t represents the scale. Let I be an image, i.e. $I : \Omega \longrightarrow \Re$, where Ω is the domain where the image is defined. In what follows, we will consider for simplicity in the exposition that $\Omega = \Re^n$ and $I \in H^2(\Omega)$ (Sobolev space, see [11] for more details). That is, I and ∇I have finite L^2 norm. $I_t = T_t(I)$ is a new image which corresponds to I at a scale t. For a given image I, to which the multiscale analysis is applied, we can extract a histogram $\{h_i^t\}_{i=0,..,L-1}$ which determines the distribution of the orientations of I at the scale t. In this case, the normalization of the values within a histogram is performed with respect to the initial addition, and not with respect to the addition at that scale. In order to compare the histograms of two images, the scale must be first adjusted.

4.1 Gaussian Multiscale Analysis

As said before, a multiscale analysis generates, for a given image, a series of images which show the evolution of the input signal when a certain process is

applied. We will use a Gaussian filter, whose properties are described in [12] and [13]. In one dimension, we use the following Gaussian kernel, where the scale t is related to the standard deviation σ according to the expression $2t = \sigma^2$:

$$K_t(x) = \tfrac{1}{\sqrt{4\pi t}} e^{-\frac{x^2}{4t}} \implies T_t(f)(x) = \int_{\Re} \tfrac{1}{\sqrt{4\pi t}} e^{-\frac{(x-y)^2}{4t}} f(y) dy \ . \tag{8}$$

Afterwards, we quantize it as follows:

$$(K_t)_n = \tfrac{1}{\sqrt{4\pi t}} e^{-\frac{n^2}{4t}} \implies (x * K_t)_m = \sum_{n=-\infty}^{\infty} x_n \tfrac{1}{\sqrt{4\pi t}} e^{-\frac{(m-n)^2}{4t}} \ . \tag{9}$$

At this point, it is important to consider the relationship between the Gaussian filtering and the heat equation, given by $\partial u / \partial t = \partial^2 u / \partial x^2$, where $u(t,x)$ is the solution of the equation. Given a signal f, the result of convolving f with the Gaussian filter K_t is equivalent to the solution of the heat equation using f as the initial data $u(t,s) = K_t * f(x)$.

Considering this relationship, a discrete version of the heat equation can be used to accelerate the approximation of the Gaussian filtering (see [14] for more details), which results in a recursive scheme in three steps for each direction, as shown below, where I_0 is the original image:

$$\begin{aligned} I_j^{n+\frac{1}{3}} &= I_j^n + v I_{j-1}^{n+\frac{1}{3}} & \forall j \in Z \ . \\ I_j^{n+\frac{2}{3}} &= I_j^{n+\frac{1}{3}} + v I_{j+1}^{n+\frac{2}{3}} & \forall j \in Z \ . \\ I_j^{n+1} &= \tfrac{v}{\lambda} I_j^{n+\frac{2}{3}} & \forall j \in Z \ . \end{aligned} \tag{10}$$

This process will be performed by rows and by columns in order to obtain a discrete expression for a two-dimensional Gaussian filtering. Making use of the features of the Gaussian kernels, the result of applying a Gaussian filter with an initial scale t can be used to obtain a Gaussian filtering of the initial image for a different scale without needing to start again from the input. We will discretize the scale considering $\sigma_n = n\sigma_0$ for a given σ_0. Taking into account the relation $\sigma^2 = 2t$, the step size Δt to go from σ_n to σ_{n+1} is given by:

$$\Delta t = \frac{((n+1)\sigma_0)^2}{2} - \frac{(n\sigma_0)^2}{2} = \left(n + \frac{1}{2}\right)(\sigma_0)^2 \ . \tag{11}$$

If we use $niter$ iterations of the recursive scheme in (10) to compute $I_{\sigma_{n+1}}$ from I_{σ_n}, the discretization scheme for the heat equation is given by:

$$\delta_t^{n+1} = \frac{\left(n + \frac{1}{2}\right)(\sigma_0)^2}{niter} \ . \tag{12}$$

4.2 Scale Estimation

We must take into account that, for a certain texture, the use of different resolutions forces us to apply Gaussian functions with different standard deviations, thus requiring an adaptation stage. To do that, we first extract the evolution of the addition of the squares of the gradients at different scales, and then we use these factors to compare the textures. Even if the quantitative distribution of the orientations may be alike for different textures, the spatial distribution will cause a divergence in the evolution and interaction, so the factors will differ.

One of the properties of the Gaussian filtering is the relationship between the resolution of two images and the effects of this kind of filters. In fact, the result of applying a Gaussian filter with standard deviation σ to an image with resolution factor x is equivalent to applying a Gaussian filter with standard deviation $k\sigma$ to the same image acquired with a resolution factor kx.

Lemma 1. *Let $I_0(x, y)$, $I_0'(x, y)$ be such that there exists a constant k satisfying that $I_0'(x, y) = I_0(kx, ky) \ \forall (x, y) \in \Omega$, then $I_t'(x, y) = I_{k^2 t}(kx, ky)$.*

Proof. The result follows from the uniqueness of the solution of the heat equation taking into account that the function $I_{k^2 t}(kx, ky)$ is a solution of the heat equation for the initial datum $I_0'(x, y)$.

Given two textures, I_0 and I_0', we will estimate the scale factor k using the normalized evolution of the norm of the gradient, that is, we will use:

$$\phi(I_0, \Omega, t) = \frac{\sqrt{\int_\Omega |\nabla I_t|^2}}{\sqrt{\int_\Omega |\nabla I_o|^2}} \ . \tag{13}$$

It is well known (see for instance [12]) that $\phi(I_0, \Omega, t)$ is a decreasing function with respect to t and $Lim_{t\to\infty}\phi(I_0, \Omega, t) = 0$. On the other hand, from the previous lemma, we deduce that if $I_0'(x, y) = I_0(kx, ky) \ \forall (x, y) \in \Omega$ then:

$$\phi(I_0, \Omega, t) = \phi(I_0', k\Omega, k^2 t) = \phi(I_0', \Omega, k^2 t) \ . \tag{14}$$

So in order to estimate a scale factor k between two textures I_0 and I_0', we will compare the functions $\phi(I_0, \Omega, t)$ and $\phi(I_0', \Omega, t)$. Let $r_n^1 = \phi(I_0, \Omega, (\sigma_n)^2 / 2)$ and $r_n^2 = \phi(I_0', \Omega, (\sigma_n)^2 / 2)$ be the ratios obtained for two textures at the scale $\sigma_n = n\sigma_0$, the best adjusting coefficient k to fit the series of r_n^2 to that of r_n^1, both consisting of N terms, can be obtained as follows: First, we fit a value $0 < h < 1$ and we interpolate the values in the series r_n^1 and r_n^2 to obtain two new series σ_n^1 and σ_n^2 which estimate the scales for which the ratios $(1, 1 - h, 1 - 2h, 1 - 3h, ..., 1 - (N - 1)h)$ are obtained. In other words, we estimate the scale where $\phi(I, \Omega, (\sigma_n^1)^2 / 2) = 1 - nh$. We point out that if $nh < 1$, σ_n^1 and σ_n^2 are well-defined because $\phi(I, \Omega, t)$ is a decreasing function with respect to t and

$Lim_{t\to\infty}\phi(I_0, \Omega, t) = 0$. With these values, we minimize the following error to obtain the scale factor k:

$$e(k) = \frac{1}{N}\sum_{i=0}^{N-1}\left(\sigma_i^1 - k\sigma_i^2\right)^2 \quad .$$ (15)

$$\frac{de(k)}{dk} = 0 \implies \sum_{i=0}^{N-1}\left(\sigma_i^1 - k\sigma_i^2\right)\sigma_i^2 = 0 \implies k = \frac{\sum_{i=0}^{N-1}\left(\sigma_i^1\sigma_i^2\right)}{\sum_{i=0}^{N-1}\left(\sigma_i^2\right)^2} \quad .$$ (16)

4.3 Multiscale Texture Orientation Histogram Comparison

We can study how the energy obtained when comparing the orientation histograms evolves as we apply a Gaussian filtering to the textures. We use the adjusting factor k, as in (16), to relate the scales to be compared. In practice, to estimate k, we take $h = 0.1$ and $N = 8$, $n = 0, 1, 2, ...7$. Finally, we obtain the energies for the comparison of the histograms at N different scales, given by (17), where $n = \{0, .., N-1\}$ and σ_N is the minimum of σ_N^1 and $k\sigma_N^2$.

$$\sigma_n = \frac{n}{N}\sigma_N \qquad \text{and} \qquad \sigma_n' = \frac{nk}{N}\sigma_N \quad .$$ (17)

Figure 5 shows the results of comparing two images of the database corresponding to the same texture, acquired with a different resolution. As observed, not only the initial energy is low, but also the subsequent energies, obtained when comparing the images at the corresponding scales, decrease as we increase the scale. On the other hand, Fig. 6 shows the comparison of two images of different textures and the energies, far from decreasing, increase from the initial value. Finally, Fig. 7 shows two images of the same texture acquired at different distances, and Fig. 8 shows the corresponding ratios and energies.

5 Robustness of Texture Classification under Darkening, Lightening and Inversion

The following examples show how darkening, lightening and inverting a pattern affect the results when calculating the energy which measures the similarity between two textures. This will allow us to test the robustness of our method when some kinds of transformations are performed in the input signal.

As observed in Fig. 9 and Table 3 (left), when a texture is darkened, the resulting energy is very low. These low results indicate that the textures are in fact almost identical. Only small differences in the new light intensity due to the representation limitations cause a negligible value. The use of integer values in intensity representation forces us to round the values once they have been reduced, generating small differences in gradient values.
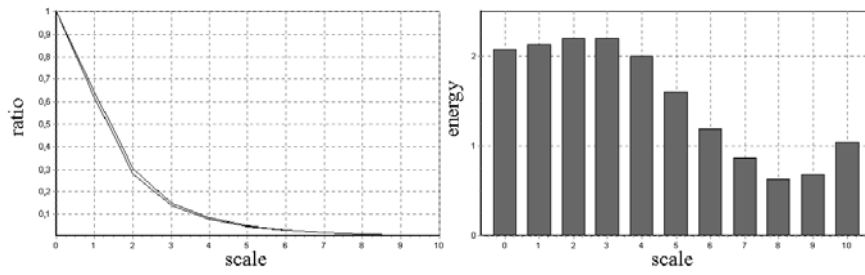
Fig. 5. Ratios (left) and energies (right) obtained when applying a Gaussian filter to textures 32 and 14 using (13) and (7) (horizontal axis represents the evolution of the scale)
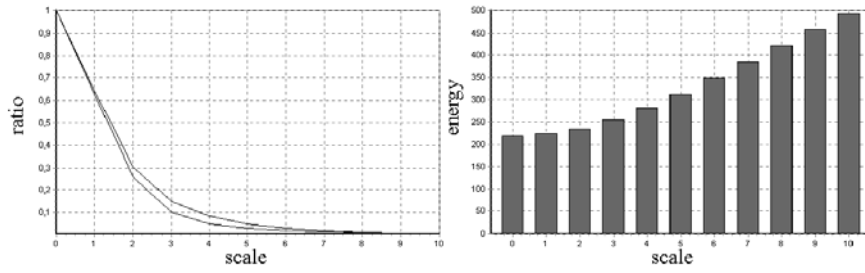

Fig. 6. Ratios (left) and energies (right) obtained when applying a Gaussian filter to textures 32 and 51 using (13) and (7) (horizontal axis represents the evolution of the scale)

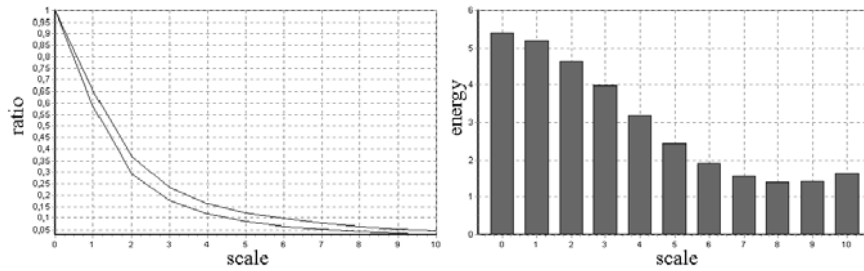**Fig. 7.** Example of two images of the same texture acquired at different distances

Fig. 8. Ratios (left) and energies (right) obtained when applying a Gaussian filter to textures in Fig. 7 using (13) and (7) (horizontal axis represents the evolution of the scale)
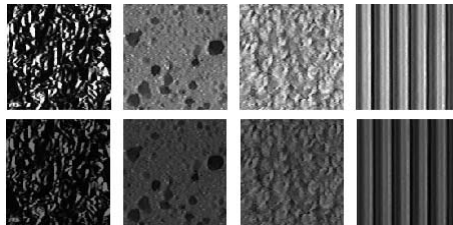

Fig. 9. Textures 15, 18, 21 and 38 before and after darkening

Figure 10 and Table 3 (middle) show similar consequences when the textures are lightened. In this case, the overflow in light intensity values for the most bright points due to the increase they undergo, which forces us to truncate the values which exceed the maximum, causes a higher difference. However, it is still much lower than those observed when similar but different textures are compared, and thus, they can be neglected.

Finally, Fig. 11 and Table 3 (right) show the results when a texture has been inverted. In most cases, the values obtained are very low and the patterns can be considered as the same texture. Nevertheless, some cases present certain problems due to the asymmetry of the filters used for edge orientation estimation. As the values which result are not low enough to clarify the similarity of the textures, thus presenting a certain ambiguity, the multiscale analysis described in the previous sections is applied and the results dispel the doubts, since they are very low when a texture is compared with its inverted version.
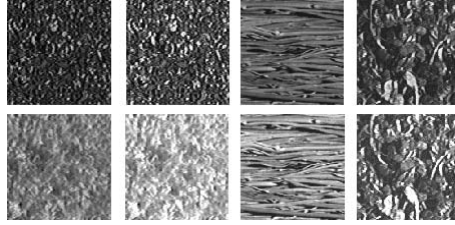
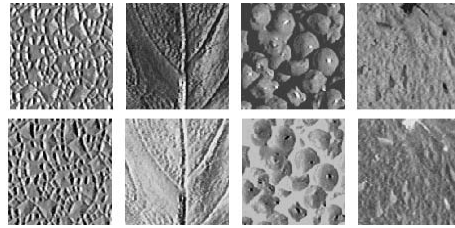**Fig. 10.** Textures 3, 14, 40 and 53 before and after lightening

Fig. 11. Textures 4, 23, 35 and 52 before and after inversion

Table 3. Comparison with darkened, lightened and inverted textures

comparison	wtd. energy	comparison	wtd. energy	comparison	wtd. energy
15-drk.15	0.0083	03-lgt.03	0.0374	04-inv.04	0.2567
18-drk.18	0.0050	14-lgt.14	0.0117	23-inv.23	0.4859
21-drk.21	0.0129	40-lgt.40	0.0052	35-inv.35	0.6009
38-drk.38	0.0145	53-lgt.53	0.0757	52-inv.52	0.4927

6   Conclusion

In this work, we have presented a new approach to texture classification. By using the modified Newton filters, we have obtained an estimation of the orientation of the edges in every point of the textures. The extraction of orientation histograms to describe the distribution of the orientations across a textured region permits us to perform an initial comparison of the textures according to the quantitative and relative distribution of the different orientations.

The comparison of the Fourier coefficients, certain normalization processes and the use of weighting functions allow a satisfactory classification in many cases, including size and rotational invariance. However, due to the ambiguities that are generated by the non-injectivity of the generation of these histograms, a further study has been carried out, by comparing the evolution of the histograms at different scales.

This multiscale analysis of the histograms has produced quite good results, since the visual similarity or difference between two textures is much more reliably detected by the evolution of the energies resulting when comparing the histograms at different scales, which have been previously adjusted. We have extracted the scale factor which must be used when comparing two textures to perform the comparison appropriately.

The quite promising numerical results obtained in the tests which have been implemented confirm the usefulness of the multiple comparison of the images, since they endow us with a much more robust discrimination criterion. Furthermore, we have tested how our method reacts when the textures are darkened, lightened or when their grayscale levels are inverted, obtaining very satisfactory results.

References

1. Sonka, M., Hlavac, V., Boyle, R.: Image processing, analysis, and machine vision. PWS-ITP (1999)
2. Gorkani M.M., Picard, R. W.: Texture orientation for sorting photos "at a glance". In the proceedings of IEEE conference on pattern recognition **I** (1995) 459-464
3. Randen, T., Husøy, J. H.: Filtering for texture classification: a comparative study. IEEE Transactions on Pattern Analysis and Machine Intelligence **21:4** (1999) 291-310
4. Puzicha, J., Buhmann, J. M., Rubner, Y., Tomasi, C.: Empirical evaluation of dissimilarity measures for color and texture. In proceedings of the IEEE International Conference on Computer Vision (ICCV'99) (1999) 1165-1173
5. Tuceryan, M., Jain, A. K.: Texture analysis. In C.H. Chen, L.F. Pau, P.S.P. Wang, editors, The Handbook of Pattern Recognition and Computer Vision (2nd edition), World Scientific Publishing Co. (1998) 207-248
6. Alemán-Flores, M., Álvarez-León, L., Moreno-Díaz jr., R.: Modified Newton filters for edge orientation estimation, shape representation and motion analysis. Cuadernos del Instituto Universitario de Ciencias y Tecnologías Cibernéticas, Universidad de Las Palmas de Gran Canaria **17** (2001) 1-30

7. Moreno-Díaz jr., R.: Computación paralela y distribuida: relación estructura-función en retinas. Tesis Doctoral (1993)
8. Columbia University and Utrecht University. Columbia-Utrecht Reflectance and Texture Database. http://www.cs.columbia.edu/CAVE/curet/.index.html
9. Paragios, N., Deriche, R.: Geodesic active regions and level set methods for supervised texture segmentation. International Journal of Computer Vision **46:3** (2002) 223
10. Weickert, J.: Multiscale texture enhancement. R. Sara (Eds.), Computer analysis of images and patterns, Lecture Notes in Computer Science Springer Berlin **970** (1995) 230-237
11. Brezis, H.: Analyse fonctionelle. Masson (1983)
12. Evans, L.: Partial Differential Equations. American Mathematical Society (1998)
13. Lindeberg, T.: Scale space theory in computer vision. Kluwer Academic Publishers (1994)
14. Alvarez, L., Mazorra, L.: Signal and image restoration using shock filters and anisotropic diffusion. SIAM J. on Numerical Analysis **31:2** (1994) 590-605