Texture Segmentation Using Local Phase Differences in Gabor Filtered Images

Anne M. LANDRAUD and Suk OH YUM

Laboratoire L3I, Faculté des Sciences, Av. Marillac, 17000-La Rochelle, France

Abstract. The present study concerns the phase information in the complex image resulting from the application of frequency-and-orientation-selective Gabor filters. The image phase is made up of a global linear part and a local phase. A strong variation of the phase occurs in the image when a boundary between two different regions is crossed. Two methods using local phase information for segmenting textures have been recently advanced. In the first one, the local phase is obtained by unwrapping the phase of the filtered image and by computing the global linear component. The second method derives the local phase from the filtered image without a phase unwrapping. Both methods are rapid but cannot be implemented with filters other than horizontal and vertical ones. Another method, which uses the phase-gradient information and allows filtering in any direction, achieves a more accurate segmentation. This new method does not require phase unwrapping since the phase gradient is obtained from the convolution of the input image with the filter gradient. We propose a method, both robust and rapid, which is a combination of the first and the third approaches, allowing use of filters with any orientation.

1 The phase components in a Gabor filtered image

The transformation which is supposed to be locally performed by cortical visual cells is the following. The Fourier transform $I_0(u,v)$ of the input image $i_0(x,y)$ is filtered by a function $G(u,v)$ giving the simple product $I(u,v)$. The filter $G(u,v)$, chosen here, is the Fourier transform of a complex Gabor function. The inverse Fourier transform of $I(u,v)$ is the output image $i(x,y)$ obtained by convolving $i_0(x,y)$ and the point spread function $g(x,y)$ of the filter in the spatial domain. The output image is a complex-valued function. This fact allows us to take into account two important parameters: the modulus and the phase. The complex output image can be written:

$$i(x,y) = i_0(x,y) * g(x,y) = Re(x,y) + jIm(x,y), \tag{1}$$

where $*$ indicates convolution and $j = \sqrt{-1}$, or:

$$i(x,y) = \left|i_0(x,y)\right| \exp[\varphi(x,y)]. \tag{2}$$

where $\phi(x)$ is the phase.The phase of the image representation (1) can be found by:

$$\varphi_{p.v.}(x,y) = \arctan[Im(x,y)/Re(x,y)]. \tag{3}$$

The phase $\varphi_{p.v.}(x,y)$, obtained by Eq.(3), is always expressed, modulo π, with a value comprised between $-\pi/2$ and $\pi/2$. It is the so-called principal value of the phase. Thus, there are discontinuities in $\pm\pi$. In order to estimate the phase values from $-\pi$ to

$+\pi$, using $\varphi_{p.v.}(x,y)$, we take into account the numerator and denominator signs in formula (4). The four possibilities, corresponding to the four quadrants, are:

$$\arctan(a/b) = \varphi_{p.v.}(x,y) \qquad \text{if } a \geq 0,\ b>0 \qquad (4\text{ a})$$
$$\arctan(a/b) = \pi - \varphi_{p.v.}(x,y) \qquad \text{if } a \geq 0,\ b<0 \qquad (4\text{ b})$$
$$\arctan(a/b) = \pi + \varphi_{p.v.}(x,y) \qquad \text{if } a<0,\ b<0 \qquad (4\text{ c})$$
$$\arctan(a/b) = 2\pi - \varphi_{p.v.}(x,y) \qquad \text{if } a<0,\ b>0. \qquad (4\text{ d})$$

On the other hand, the "true value" of the phase $\varphi(x,y)$, as represented in Eq.(2), is comprised between $-\infty$ and $+\infty$. We have to change it into a continuous function by a procedure which is called the phase unwrapping. A solution of the unwrapping problem is generally to add or to subtract 2π from the part of the function which lies just after a discontinuity:

$$\varphi(x,y) = \varphi_{p.v.}(x,y) + 2\pi n(x,y) \qquad (5)$$

where n(x,y) is given integer values.

This algorithm requires only the knowledge of the principal value of the phase but it does not allow fluctuations greater than π between different samples. Thus, if the phase difference between two samples is less than $-\pi$ (greater than $+\pi$), we just add 2π (subtract 2π) to the phase of the first sample [1]. From the point of view of information volume, the phase unwrapping is strictly equivalent to the determination of the n(x,y) factor which is necessary to get a continuous phase as in Eq.(5).

The position of a point (x,y) can be inferred from the phase of the filtered image at that point. The knowledge of the phase also allows us to estimate the distance which separates a point from another one and the amount of shift between two or several points. It has been shown [1] that the general form of the unwrapped phase is made of a global linear phase component and a local phase component. The former is a periodic one, while the latter corresponds to noise acting on the periodical part. For texture analysis, the local phase component bears an important information about the place where two different regions meet and it plays a very important role for segmenting images of textures [1-3]. The local phase is theoretically computed by subtracting the linear component from the unwrapped phase. However this would assume a solution exists to the "zero-point" problems - discussed below.

2. Other methods of texture-image segmentation using the information contained in the phase

When the phase is computed from Eq.(3), it is assumed that the numerator and denominator values are not simultaneously equal to zero. A point such that Re(x,y)=Im(x,y)=0, is called a zero point. It has been shown that images of the real world contain a great number of zero points which confound the application of the phase-unwrapping procedure.

The first stage of an algorithm proposed by Nicoulin [1] consists in locating the zero points in the filtered image by integrating the phase gradient over a closed contour having a minimum length, made of four neighbouring samples which are on the corners of a square: (x,y), (x+1,y), (x+1,y+1) and (x,y+1). The zero points with the corresponding discontinuities $\pm 2\pi$ are then shifted to the edges of the image. The second stage consists in unwrapping the phase to get $\varphi(x,y)$, while the local phase $\Delta\varphi(x,y)$ is obtained during the third stage by:

$$\Delta\varphi(x,y) = \varphi(x,y) - \varphi_{lin}(x,y), \qquad (6)$$

where the linear component $\varphi_{lin}(x,y)$ is mathematically represented by an inclined plane, the orientation of the maximum gradient of $\varphi_{lin}(x,y)$ being the same as the orientation of the filter. This method allows rapid computation of the unwrapped phase. However, the algorithm cannot be applied if the filtered image includes too many zero points. Moreover, the linear component cannot be computed with other filters than horizontal or vertical ones.

Another method, proposed by Du Buf [2], avoids the problem of zero points because it computes the local phase directly from the output image (2) without unwrapping the phase. The local phase can be computed even in the presence of a great number of zero points. This method is also very fast since it requires only one image convolution, but the oriented filters ($0<\theta<\pi/2$, $\pi/2<\theta<\pi$) cannot be used.

Bovik et al. [4] use information concerning both modulus and phase for segmenting textures. They compute a function of the phase gradient directly from the input image previously convolved with the filter gradient. This method avoids the problem of zero points. Another advantage is that the phase gradient can be obtained with oriented filters. But, at least five image convolutions have to be computed for each channel, i.e. for a frequency-orientation pair. Other researchers [5-7] have recently investigated the phase information in computer vision.

3. A new method for segmenting texture images by using the phase derivative with oriented filters

It is well known that the phase value varies abruptly at places where different textures meet. Therefore, if we use an adequate filtering process, we expect a stable phase everywhere except for the boundaries of textures. In principle, the phase gradient is more sensitive to a local variation than the local phase itself and the derivative of the phase will take an absolute value much greater on a boundary than in the other places. A straightforward method would be to estimate the derivative by means of simple differences between the phases of two neighbouring points. But, with that technique, we risk of getting a very bad approximation of the phase derivative for two reasons: (1) we have to know the unwrapped phase as a continuous function - that is not a simple problem in the presence of many zero points - and (2) an awkward noise is associated with that operation. In this section, we propose a more precise method based on some mathematical developments. Let us recall the classical formula:

$$\frac{d}{dx}\left[\operatorname{arctg}(x)\right]=\frac{1}{1+x^2}. \tag{7}$$

We suppose that a derivative of the phase exists everywhere. Under this assumption it can be shown by simple algebra that the phase derivative with respect to x at any point (x, y) is:

$$\varphi_x(x,y)=\frac{\partial}{\partial x}\left[\varphi(x,y)\right]=\frac{\operatorname{Im}\left[i^*(x,y)i_x(x,y)\right]}{\left|i(x,y)\right|^2}, \tag{8}$$

where $\left|i(x,y)\right|$ and $i^*(x,y)$ are the modulus and the conjugate complex quantity, respectively, of the complex filtered output image i(x, y) and $i_x(x,y)$is the derivative with respect to x of the complex output image:

$$i_x(x,y)=\frac{\partial}{\partial x}\operatorname{Re}(x,y)+j\frac{\partial}{\partial x}\operatorname{Im}(x,y). \tag{9}$$

The phase derivative with respect to y is computed in a similar manner. Notice that the above formulas, which describe the phase derivatives, are computed along two directions: the horizontal and the vertical ones. For any other direction, corresponding to an angle θ at which the filter is applied, we can readily deduce the derivative:

$$D_\theta[\varphi(x,y)] = \cos\theta\frac{\partial}{\partial x}\varphi(x,y) + \sin\theta\frac{\partial}{\partial y}\varphi(x,y) = \varphi_x(x,y)\cos\theta + \varphi_y(x,y)\sin\theta$$
$$= \frac{1}{|i(x,y)|^2}\left\{\mathrm{Im}\left[i^*(x,y)i_x(x,y)\right]\cos\theta + \mathrm{Im}\left[i^*(x,y)i_y(x,y)\right]\sin\theta\right\}. \quad (10)$$

The feature $D_\theta[\varphi(x,y)]$ thus provides information for extracting contours of texture images. However as $D_\theta[\varphi(x,y)]$ is arbitrar negative or positive, we have used its absolute value. This measure allows us to clearly distinguish the points where that value is high from the other points where it is practically constant. Before computing the final derived phase, we perform another stage which consists in subtracting the mean of the derived phase. Indeed, Nicoulin [1] has shown that the significant phase information intended to segment an image is contained in the local phase $\varphi_{loc}(x,y)$. This is obtained by subtracting the linear phase $\varphi_{lin}(x,y)$ from $\varphi(x,y)$:

$$\varphi_{loc}(x,y) = \varphi(x,y) - \varphi_{lin}(x,y), \quad (11)$$

where:

$$\varphi_{lin}(x,y) = f_x x + f_y y, \quad (12)$$

f_x and f_y being two constants to be determined.

The derivative of the local phase is then:

$$\frac{\partial}{\partial x}\varphi_{loc}(x,y) = \frac{\partial}{\partial x}\varphi(x,y) - \frac{\partial}{\partial x}\varphi_{lin}'(x,y) = \varphi_x(x,y) - f_x. \quad (13)$$

The values of f_x and f_y are chosen so that the greatest oscillations of the local phase be located principally near the contour, in other words so that we obtain high values on the contours and approximately zero values elsewhere. We estimate f_x as the mean of $\varphi_x(x,y)$.

When we apply a multichannel Gabor filtering, we have to integrate the texture characteristics which come from each channel. To do so, we determine the maximum value of the feature-vector components at each pixel, as we list in more details below:

. $D_k(x,y)$ is the value of the derived phase at point (x,y), i.e. the considered feature, which is obtained by applying the kth filter. k = 1, ..., P, with P=M x N.

. C_k is the mean of the values $D_k(x,y)$ which were computed in all points of the kth filter output.

. The P filters are applied and the quantity $D_k(x,y)-C_k$ is computed for k=1,...,P.

. The characteristic value attached to (x,y) is found as $\max_k\left[D_k(x,y) - C_k\right]$.

Then, a classical method for detecting contours is achieved by using an empirically determined threshold. For this purpose we use the following family of separable filters [8, 9]:

$$F_{\rho_{m_i},\theta_{m_j}}(\rho,\theta) = F_{\rho_{m_i}}(\rho)F_{\theta_{m_j}}(\theta), \quad (14)$$

ρ being the radial frequency, θ the orientation, ρ_{m_i} the ith preferential frequency and θ_{m_j} the jth preferential orientation of the filter (i,j), $0 \leq i \leq M-1$, $0 \leq j \leq N-1$, M and N being the considered numbers of frequencies and orientations, respectively, and P=MxN. $F_{\rho_{mi}}(\rho)$ is the expression in polar coordinates of the Fourier transform (where $K_i = (2\pi\sigma_i)^{-1}$):

$$F_{u_{m_i}}(u)=\exp\left[-\frac{(u-u_{m_i})^2}{2K_i^2}\right] \tag{15}$$

of the complex Gabor function:

$$f_{u_{m_i}}(x)=\frac{1}{\sigma_i\sqrt{2\pi}}\exp\left(-\frac{x^2}{2\sigma_i^2}\right)\exp\left(j2\pi u_{m_i}x\right). \tag{16}$$

The second function of Eq.(14) is the gaussian directional filter:

$$F_{\theta_{m_j}}(\theta)=\exp\left[-\left(\theta-\theta_{m_j}\right)^2/\left(2\sigma_\theta^2\right)\right]. \tag{17}$$

Our algorithm was tested on both natural and artificial textures. Fig.1 shows an example of segmentation in the case of an image made of two parts of the same texture which were lightly shifted. Other examples can be seen in Fig.2, 3 and 4. In some places, especially in Fig.2, the phase value is close to the background and we get broken edges. If we lessen the threshold value, the noise is increased. A solution might be to calculate the phase more accurately while increasing computation time.

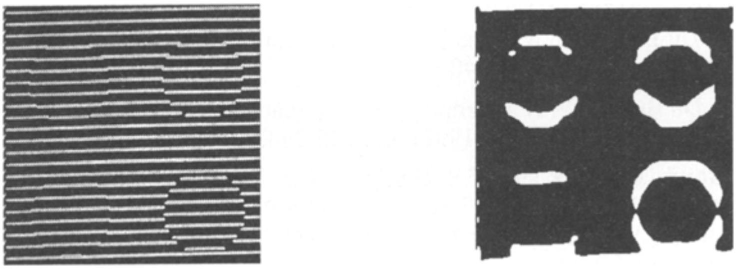

Fig. 1. An example of texture image and the result of the segmentation process using phase information. The frequency of the filter is 22 u_{m_0} and the orientation: 90°.

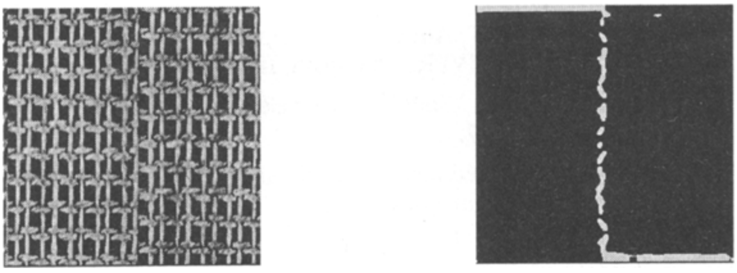

Fig. 2. Another example of artificial texture image containing circles with different shiftings and the result. The frequency of the filter is 10 u_{m_0} and the orientation: 90°.

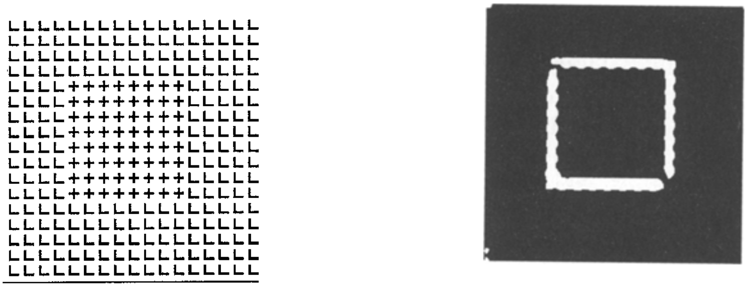

Fig. 3. An example ofsynthetic image made of "L" and "+" and the result of the segmentation process. The frequency of the filter is 24 u_{m_0} and the orientation: 135°.

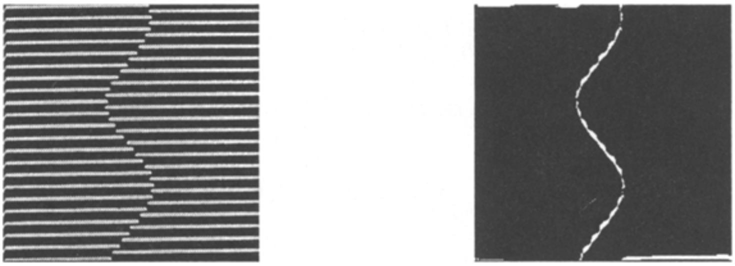

Fig. 4. An artificial texture image showing an illusory sinuous contour and the result of the segmentation process. The frequency of the filter is 21 u_{m_0} and the orientation: 90°.

References

1. A. Nicoulin, "Analyse d'images par spectre local de phase", Presses Polytechniques et Universitaires Romandes (Mars 1990).
2. J.M.H. Du Buf, "Gabor phase in texture discrimination", Signal Processing, Vol. 21 (janvier 1990), pp. 221-240.
3. J.M.H. Du Buf and P. Heitkämper, "Texture features based on Gabor phase", Signal Processing, Vol. 23 (1991), pp. 225-244.
4. A.C. Bovik, M. Clark and W.S. Geisler, "Multichannel texture analysis using localized spatial filters", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12 (janvier 1990), pp. 53-73.
5. K. Langley, T.J. Atherton, R.G. Wilson and M.H.E. Lacombe, "Vertical and horizontal disparities from phase", Proc. of 1st E.C.C.V., Antibes, New York, Springer-Verlag, 1990, pp. 315-325.
6. K. Langley, D.J. Fleet and T. Atherton, "Multiple motions from instantaneous frequency", Proc. of IEEE CVPR, Champain, Il., 1992, pp. 846-849.
7. D.J. Fleet and A.D. Jepson, "Stability of phase information", I.E.E.E. P.A.M.I., vol. 15, 1993,pp. 1253-1267.
8. A.M. Landraud, "Vision-based model of artificial texture perception", SPI/SPSE Symposium on Electronic Imaging Science & Technology, San Jose, Ca, USA (February 24 - March 1, 1991).
9. S. Oh Yum, A.M. Landraud and G. Stamon, "Une méthode de classification des textures utilisant l'information dérivée de la phase dans un système de filtrage multicanaux", Colloque AGI'94, Poitiers (1994), pp. 349-352.