By contrast, if the edges in the different channels point randomly in all directions, all diagonal terms will be nonzero and equal. In this way, it is possible to distinguish random changes by noise from coherent edges. The trace of the matrix $\boldsymbol{S}$

$$\operatorname{trace}(\boldsymbol{S}) = \sum_{d=1}^{D} S_{dd} = \sum_{d=1}^{D} \sum_{p=1}^{P} \left(\frac{\partial g_p}{\partial x_d} \right)^2 \qquad (9.184)$$

gives a measure of the edge strength that we have already defined in Eq. (9.179). It is independent of the orientation of the edge because the trace of a symmetric matrix is invariant to a rotation of the coordinate system.

In conclusion, the matrix $\boldsymbol{S}$ is the key for edge detection in multichannel signals. Note that an arbitrary number of channels can be processed and that the number of computations increases only linearly with the number of channels. The analysis is, however, of order $O(D^2)$ in the dimension of the signal.

9.8 Tensor representation of simple neighborhoods

9.8.1 Simple neighborhoods

The mathematical description of a local neighborhood by continuous functions has two significant advantages. First, it is much easier to formulate the concepts and to study their properties analytically. As long as the corresponding discrete image satisfies the sampling theorem, all the results derived from continuous functions remain valid because the sampled image is an exact representation of the continuous gray-value function. Second, we can now distinguish between errors inherent to the chosen approach and those that are only introduced by the discretization.

A simple local neighborhood is characterized by the fact that the gray value only changes in one direction. In all other directions it is constant. Because the gray values are constant along lines and form oriented structures this property of a neighborhood is denoted as *local orientation* [13] or *linear symmetry* [14]. Only more recently, the term *simple neighborhood* has been coined by Granlund and Knutsson [9].

If we orient the coordinate system along the principal directions, the gray values become a 1-D function of only one coordinate. Generally, we will denote the direction of local orientation with a unit vector $\bar{\boldsymbol{r}}$ perpendicular to the lines of constant gray values. Then, a simple neighborhood is mathematically represented by

$$g(\boldsymbol{x}) = g(\boldsymbol{x}^T \bar{\boldsymbol{r}}) \qquad (9.185)$$

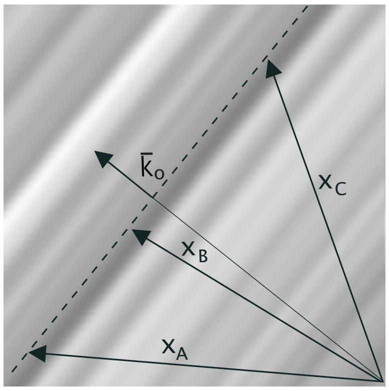

Figure 9.22: *Illustration of a linear symmetric or simple neighborhood. The gray values depend only on a coordinate given by a unit vector* $\bar{\boldsymbol{r}}$*.*

Equation Eq. (9.185) is also valid for image data with more than two dimensions. The projection of the vector $\boldsymbol{x}$ onto the unit vector $\bar{\boldsymbol{r}}$ makes the gray values depend only on a scalar quantity, the coordinate in the direction of $\bar{\boldsymbol{r}}$ (Fig. 9.22). The essential relation now is that the gradient is parallel to the direction $\bar{\boldsymbol{r}}$ into which the gray values change:

$$\nabla g(\boldsymbol{x}^T\bar{\boldsymbol{r}}) = \begin{bmatrix} \dfrac{\partial g(\boldsymbol{x}^T\bar{\boldsymbol{r}})}{\partial x_1} \\ \ldots \\ \dfrac{\partial g(\boldsymbol{x}^T\bar{\boldsymbol{r}})}{\partial x_W} \end{bmatrix} = \begin{bmatrix} \bar{r}_1 g'(\boldsymbol{x}^T\bar{\boldsymbol{r}}) \\ \ldots \\ \bar{r}_D g'(\boldsymbol{x}^T\bar{\boldsymbol{r}}) \end{bmatrix} = \bar{\boldsymbol{r}} g'(\boldsymbol{x}^T\bar{\boldsymbol{r}}) \qquad (9.186)$$

The term g' denotes the derivative of g with respect to the scalar variable $\boldsymbol{x}^T\bar{\boldsymbol{r}}$. In the hyperplane perpendicular to the gradient, the values remain locally constant.

A simple neighborhood has a special form in Fourier space. Let us first assume that the whole image is described by Eq. (9.185), that is, $\bar{\boldsymbol{r}}$ does not depend on the position. Then, from the very fact that a simple neighborhood is constant in all directions except $\bar{\boldsymbol{r}}$, we infer that the Fourier transform must be confined to a line. The direction of the line is given by $\bar{\boldsymbol{r}}$:

$$g(\boldsymbol{x}^T\bar{\boldsymbol{r}}) \quad \Longleftrightarrow \quad \hat{g}(k)\delta(\boldsymbol{k} - \bar{\boldsymbol{r}}(\boldsymbol{k}^T\bar{\boldsymbol{r}})) \qquad (9.187)$$

where k denotes the coordinate in the Fourier domain in the direction of $\bar{\boldsymbol{r}}$. The argument in the δ function is only zero when $\boldsymbol{k}$ is parallel to $\bar{\boldsymbol{r}}$.

In a second step, a window function $w(\boldsymbol{x} - \boldsymbol{x}_0)$ is used to restrict the area to a local neighborhood around a point $\boldsymbol{x}_0$. Thus $g(\boldsymbol{x}^T\bar{\boldsymbol{r}})$ in

Eq. (9.187) is multiplied by the window function $w(\boldsymbol{x}-\boldsymbol{x}_0)$ in the spatial domain. The size and shape of the neighborhood is determined by the window function. Multiplication in the space domain corresponds to a convolution in the Fourier domain (Section 8.6.3). Thus,

$$w(\boldsymbol{x}-\boldsymbol{x}_0)g(\boldsymbol{x}^T\bar{\boldsymbol{r}}) \quad \Longleftrightarrow \quad \hat{w}(\boldsymbol{k}) * \hat{g}(k)\delta(\boldsymbol{k}-\bar{\boldsymbol{r}}(\boldsymbol{k}^T\bar{\boldsymbol{r}})) \qquad (9.188)$$

where $\hat{w}(\boldsymbol{k})$ is the Fourier transform of the window function.

The limitation to a local neighborhood thus blurs the line in Fourier space to a “sausage-like” shape. Because of the reciprocity of scales between the two domains, its thickness is inversely proportional to the size of the window. From this elementary relation, we can already conclude qualitatively that the accuracy of the orientation estimate is directly related to the ratio of the window size to the wavelength of the smallest structures in the window.

9.8.2 Direction versus orientation

For an appropriate representation of simple neighborhoods, it is first important to distinguish *orientation* from *direction*. The direction is defined over the full angle range of 2π (360°). Two vectors that point in opposite directions, that is, differ by 180°, are different. The gradient vector, for example, always points into the direction into which the gray values are increasing. With respect to a bright object on a dark background, this means that the gradient at the edge is pointing towards the object. In contrast, to describe the direction of a local neighborhood, an angle range of 360° makes no sense. We cannot distinguish between patterns that are rotated by 180°. If a pattern is rotated by 180°, it still has the same direction. Thus, the direction of a simple neighborhood is different from the direction of a gradient. While for the edge of an object gradients pointing in opposite directions are conflicting and inconsistent, for the direction of a simple neighborhood this is consistent information.

In order to distinguish the two types of “directions,” we will speak of *orientation* in all cases where an angle range of only 180° is required. Orientation is still, of course, a *cyclic* quantity. Increasing the orientation beyond 180° flips it back to 0°. Therefore, an appropriate representation of orientation requires an angle doubling.

In his pioneering paper on a general picture processing operator Granlund [13] introduced a vectorial representation of the local orientation. The magnitude of the *orientation vector* is set to the certainty with which the orientation could be determined and its direction to the doubled orientation angle. This vector representation of orientation has two significant advantages.

First, it is more suitable for further processing than a separate representation of the orientation by two scalar quantities. Take, for example, averaging. Vectors are summed up by chaining them together, and the resulting sum vector is the vector from the starting point of the first vector to the end point of the last vector. The weight of an individual vector in the vector sum is given by its length. In this way, the certainty of the orientation measurement is adequately taken into account.

9.8.3 First-order tensor representation; structure tensor

The vectorial representation discussed in Section 9.8.2 is incomplete. Although it is suitable for representing the orientation of simple neighborhoods, it cannot distinguish between neighborhoods with constant values and isotropic orientation distribution (e. g., uncorrelated noise). Both cases result in an orientation vector with zero magnitude.

Therefore, it is obvious that an adequate representation of gray-value changes in a local neighborhood must be more complex. Such a representation should be able to determine a unique orientation and to distinguish constant neighborhoods from neighborhoods without local orientation.

A suitable representation can be introduced by a optimization strategy to determine the orientation of a *simple neighborhood* in a slightly more general way as performed by Kass and Witkin [15]. The optimum orientation is defined as the orientation that shows the least deviations from the directions of the gradient. A suitable measure for the deviation must treat gradients pointing in opposite directions equally. The squared scalar product between the gradient vector and the unit vector representing the local orientation $\bar{\boldsymbol{r}}$ meets this criterion

$$(\nabla g^T \bar{\boldsymbol{r}})^2 = |\nabla g|^2 \cos^2 \left(\angle(\nabla g, \bar{\boldsymbol{r}}) \right) \tag{9.189}$$

This quantity is proportional to the cosine squared of the angle between the gradient vector and the orientation vector and is thus maximal when ∇g and $\bar{\boldsymbol{r}}$ are parallel or antiparallel, and zero if they are perpendicular to each other. Therefore, the following integral is maximized in a D-dimensional local neighborhood:

$$\int w(\boldsymbol{x} - \boldsymbol{x}') \left(\nabla g(\boldsymbol{x}')^T \bar{\boldsymbol{r}} \right)^2 \, \mathrm{d}^D x' \tag{9.190}$$

where the window function w determines the size and shape of the neighborhood around a point $\boldsymbol{x}$ in which the orientation is averaged. The maximization problem must be solved for each point $\boldsymbol{x}$. Equation Eq. (9.190) can be rewritten in the following way:

$$\bar{\boldsymbol{r}}^T \boldsymbol{J} \bar{\boldsymbol{r}} \to \max \tag{9.191}$$

with

$$\boldsymbol{J} = \int_{-\infty}^{\infty} w(\boldsymbol{x} - \boldsymbol{x}') \left(\nabla g(\boldsymbol{x}') \nabla g(\boldsymbol{x}')^T \right) \, \mathrm{d}^D x'$$

The components of this symmetric D × D tensor are

$$J_{pq}(\boldsymbol{x}) = \int\limits_{-\infty}^{\infty} w(\boldsymbol{x}-\boldsymbol{x}') \left(\frac{\partial g(\boldsymbol{x}')}{\partial x'_p} \frac{\partial g(\boldsymbol{x}')}{\partial x'_q} \right) \mathrm{d}^D x' \qquad (9.192)$$

At this point it is easy to extend the tensor for multichannel signals. It is only needed to sum the tensor components for all channels. The weighting function might be different for each channel in order to consider the significance and spatial resolution of a certain channel. With all this, Eq. (9.192) extends to

$$J_{r,s}(\boldsymbol{x}) = \sum_{p=1}^{P} \int\limits_{-\infty}^{\infty} w_p(\boldsymbol{x}-\boldsymbol{x}') \left(\frac{\partial g_p(\boldsymbol{x}')}{\partial x'_r} \frac{\partial g_p(\boldsymbol{x}')}{\partial x'_s} \right) \mathrm{d}^D x' \qquad (9.193)$$

These equations indicate that a tensor is an adequate first-order representation of a local neighborhood. The term first-order has a double meaning. First, only first-order derivatives are involved. Second, only simple neighborhoods can be described in the sense that we can analyze in which direction(s) the gray values change. More complex structures such as structures with multiple orientations cannot be distinguished.

The complexity of Eqs. (9.191) and (9.192) somewhat obscures their simple meaning. The tensor is symmetric. By a rotation of the coordinate system, it can be brought into a diagonal form. Then, Eq. (9.191) reduces to

$$J = [\bar{r}'_1, \bar{r}'_2, \ldots, \bar{r}'_D] \begin{bmatrix} J_{1'1'} & 0 & \ldots & 0 \\ 0 & J_{2'2'} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & J_{D'D'} \end{bmatrix} \begin{bmatrix} \bar{r}'_1 \\ \bar{r}'_2 \\ \ldots \\ \bar{r}'_D \end{bmatrix} \quad \rightarrow \quad \max$$

or

$$J = \sum_{d'=1}^{D} J_{d'd'} (\bar{r}'_{d'})^2$$

Without loss of generality, we assume that $J_{1'1'} \geq J_{d'd'}\ \forall d' \neq 1$. Then, it is obvious that the unit vector $\bar{\boldsymbol{r}}' = [1\ 0\ \ldots\ 0]^T$ maximizes the foregoing expression. The maximum value is $J_{1'1'}$. In conclusion, this approach not only yields a tensor representation for the local neighborhood but also shows the way to determine the orientation. Essentially, we have to solve an *eigenvalue problem*. The eigenvalues λ_d and eigenvectors $\boldsymbol{k}_d$ of a D × D matrix are defined by

$$\boldsymbol{J}\boldsymbol{k}_d = \lambda_d \boldsymbol{k}_d \qquad (9.194)$$

An eigenvector $\boldsymbol{k}_d$ of $\boldsymbol{J}$ is thus a vector that is not turned in direction by multiplication with the matrix $\boldsymbol{J}$, but is only multiplied by a scalar factor, the eigenvalue λ_w. This implies that the structure tensor becomes diagonal in a coordinate system that is spanned by the eigenvectors. For our further discussion it is important to keep in mind that the eigenvalues are all real and nonnegative and form an orthogonal basis [16, 17, 18].

9.8.4 Classification of local neighborhoods

The power of the tensor representation becomes apparent if we classify the eigenvalues of the structure tensor. The classifying criterion is the number of eigenvalues that are zero. If an eigenvalue is zero, this means that the gray values in the direction of the corresponding eigenvector do not change. The number of zero eigenvalues is also closely related to the rank of a matrix. The *rank* of a matrix is defined as the dimension of the subspace for which $\boldsymbol{J}\boldsymbol{k} \neq \boldsymbol{0}$. The space for which $\boldsymbol{J}\boldsymbol{k} = \boldsymbol{0}$ is denoted as the *null space*. The dimension of the null space is the dimension of the matrix minus the rank of the matrix and equal to the number of zero eigenvalues. We will perform an analysis of the eigenvalues for two and three dimensions. In two dimensions, we can distinguish the following cases:

$\lambda_1 = \lambda_2 = 0$, *rank 0 tensor.* Both eigenvalues are zero. The mean square magnitude of the gradient $(\lambda_1 + \lambda_2)$ is zero. The local neighborhood has constant values. It belongs to an object with a homogeneous feature;

$\lambda_1 > 0, \lambda_2 = 0$, *rank 1 tensor.* One eigenvalue is zero. The values do not change in the direction of the corresponding eigenvector. The local neighborhood is a simple neighborhood with ideal orientation. This could either be the *edge* of an object or an *oriented texture*;

$\lambda_1 > 0, \lambda_2 > 0$, *rank 2 tensor.* Both eigenvalues are unequal to zero. The gray values change in all directions as at the *corner* of an object or a texture with a distributed orientation. In the special case of $\lambda_1 = \lambda_2$, we speak of an isotropic gray-value structure as it changes equally in all directions.

The classification of the eigenvalues in three dimensions is similar to the 2-D case:

$\lambda_1 = \lambda_2 = \lambda_3 = 0$, *rank 0 tensor.* The gray values do not change in any direction; constant neighborhood.

$\lambda_1 > 0, \lambda_2 = \lambda_3 = 0$, *rank 1 tensor.* The gray values change only in one direction. This direction is given by the eigenvector to the nonzero eigenvalue. The neighborhood includes a boundary between two objects (*surface*) or a *layered texture.* In a space-time image, this means a constant motion of a spatially oriented pattern ("planar wave");

$\lambda_1 > 0, \lambda_2 > 0, \lambda_3 = 0$, *rank 2 tensor.* The gray values change in two directions and are constant in a third. The eigenvector to the zero eigenvalue gives the direction of the constant gray values. This happens at the edge of a three-dimensional object in a volumetric image, or if a pattern with distributed spatial orientation moves with constant speed; and

$\lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0$, *rank 3 tensor.* The gray values change in all three directions as at the corner of an object or a region with isotropic noise.

In practice, it will not be checked whether the eigenvalues are zero but below a critical threshold that is determined by the noise level in the image.

9.8.5 Computation of the structure tensor

The structure tensor (Section 9.8.3) can be computed straightforwardly as a combination of *linear convolution* and *nonlinear point operations*. The partial derivatives in Eq. (9.192) are approximated by discrete derivative operators. The integration weighted with the window function is replaced by a convolution with a smoothing filter that has the shape of the window function. If we denote the discrete partial derivative operator with respect to the coordinate p by the operator $\mathcal{D}_p$ and the (isotropic) smoothing operator by $\mathcal{B}$, the local structure of a gray-value image can be computed with the structure tensor operator

$$\mathcal{J}_{pq} = \mathcal{B}(\mathcal{D}_p \cdot \mathcal{D}_q) \tag{9.195}$$

The equation is written in the operator notation introduced in Section 9.1.3. Pixelwise multiplication is denoted by a centered dot "·" to distinguish it from successive application of convolution operators. Equation (9.195) expresses in words that the $\mathcal{J}_{pq}$ component of the tensor is computed by convolving the image independently with $\mathcal{D}_p$ and $\mathcal{D}_q$, multiplying the two images pixelwise, and smoothing the resulting image with $\mathcal{B}$. For the inertia tensor method, a similar tensor operator can be formulated

$$\mathcal{J}'_{pp} = \sum_{q \neq p} \mathcal{B}(\mathcal{D}_q \cdot \mathcal{D}_q), \quad \mathcal{J}'_{pq} = -\mathcal{B}(\mathcal{D}_p \cdot \mathcal{D}_q) \tag{9.196}$$

These operators are valid in images of any dimension $D \geq 2$. In a D-dimensional image, the structure tensor has $D(D+1)/2$ independent components, hence 3 in 2-D and 6 in 3-D images. These components are best stored in a multichannel image with $D(D+1)/2$ channels.

The smoothing operations consume the largest number of operations. Therefore, a fast implementation must, in the first place, apply a fast smoothing algorithm. A fast algorithm can be established based on the general observation that higher-order features always show a lower resolution than the features from which they are computed. This means that the structure tensor can be stored on a coarser grid and thus in a smaller image. It is convenient and appropriate to reduce the scale by a factor of two by storing only every second pixel in every second row.

These procedures lead us in a natural way to multigrid data structures that are discussed in detail in Chapter 8.10. Multistep averaging is discussed in detail in Section 9.5.5.

Storing higher-order features on coarser scales has another significant advantage. Any subsequent processing is sped up simply by the fact that many fewer pixels have to be processed. A linear scale reduction by a factor of two results in a reduction in the number of pixels and the number of computations by a factor of 4 in two and 8 in three dimensions.

The accuracy of the orientation angle strongly depends on the implementation of the derivative filters. It is critical to use a derivative filter that has been optimized for a minimum error in the direction of the gradient. Such filters are discussed in Section 9.7.1.

9.8.6 Orientation vector

With the simple convolution and point operations discussed in the previous section, we computed the components of the structure tensor. In this section, we solve the eigenvalue problem to determine the orientation vector. In two dimensions, we can readily solve the eigenvalue problem. The orientation angle can be determined by rotating the inertia tensor into the principal axes coordinate system. As shown, for example, by Jähne [1], the orientation angle is given by

$$\tan 2\phi = \frac{2J_{12}}{J_{22} - J_{11}} \tag{9.197}$$

Without defining any prerequisites, we have obtained the anticipated angle doubling for orientation as discussed in Section 9.8.2 at the beginning of this chapter. Because $\tan 2\phi$ is gained from a quotient, we can regard the dividend as the y and the divisor as the x component of a vector and can form the *orientation vector* $\boldsymbol{o}$, as introduced by Granlund [13]

$$\boldsymbol{o} = \begin{bmatrix} J_{22} - J_{11} \\ 2J_{12} \end{bmatrix} \tag{9.198}$$

The argument of this vector gives the orientation angle and the magnitude a certainty measure for local orientation.

The result of Eq. (9.198) is remarkable in that the computation of the components of the orientation vector from the components of the orientation tensor requires just one subtraction and one multiplication by two. As these components of the orientation vector are all we need for further processing steps, we do not need the orientation angle or the magnitude of the vector. Thus, the solution of the eigenvalue problem in two dimensions is trivial.

9.8.7 Coherency

The orientation vector reduces local structure to local orientation. From three independent components of the symmetric tensor still only two are used. When we fail to observe an orientated structure in a neighborhood, we do not know whether any gray-value variations or distributed orientations are encountered. This information is included in the not yet used component of the tensor $J_{11}+J_{22}$, which gives the mean square magnitude of the gradient. Consequently, a well-equipped structure operator needs to include also the third component. A suitable linear combination is

$$\boldsymbol{s} = \begin{bmatrix} J_{11} + J_{22} \\ J_{11} - J_{22} \\ 2J_{12} \end{bmatrix} \tag{9.199}$$

This structure operator contains the two components of the orientation vector and, as an additional component, the mean square magnitude of the gradient, which is a rotation-invariant parameter. Comparing the latter with the magnitude of the orientation vector, a constant gray-value area and an isotropic gray-value structure without preferred orientation can be distinguished. In the first case, both squared quantities are zero; in the second, only the magnitude of the orientation vector. In the case of a perfectly oriented pattern, both quantities are equal. Thus their ratio seems to be a good *coherency measure* c_c for local orientation

$$c_c = \frac{(J_{22} - J_{11})^2 + 4J_{12}^2}{(J_{11} + J_{22})^2} = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2 \tag{9.200}$$

The coherency c_c ranges from 0 to 1. For ideal local orientation ($\lambda_2 = 0, \lambda_1 > 0$) it is one, for an isotropic gray-value structure ($\lambda_1 = \lambda_2 > 0$) it is zero.

9.8.8 Color coding of the two-dimensional structure tensor

A symmetric 2-D tensor has three independent pieces of information (Eq. (9.199)), which fit well to the three degrees of freedom available to represent color, for example, luminance, hue, and saturation. First, the squared magnitude of the gradient is mapped onto the intensity. Second, the coherency measure Equation (9.200) is used as the saturation. The angle of the orientation vector is represented as the hue.

In practice, a slight modification of this color representation is useful. The squared magnitude of the gradient shows variations too large to be displayed in the narrow dynamic range of a display screen with only 256 luminance levels. Therefore, a suitable normalization is required. The basic idea of this normalization is to compare the squared magnitude of the gradient with the noise level. Once the gradient is well above the noise level it is regarded as a significant piece of information. This train of thoughts suggests the following normalization for the intensity I:

$$I = \frac{J_{11} + J_{22}}{(J_{11} + J_{22}) + \gamma \sigma_n^2} \tag{9.201}$$

where σ_n is an estimate of the standard deviation of the noise level. This normalization provides a rapid transition of the luminance from one, when the magnitude of the gradient is larger than σ_n, to zero when the gradient is smaller than σ_n. The factor γ is used to optimize the display.

A demonstration of the structure tensor technique is given by the heurisko image processing workspace `orient.ws` in `/software/09`.

9.9 References

[1] Jähne, B., (1997). *Digital Image Processing—Concepts, Algorithms, and Scientific Applications,* 4th edition. New York: Springer.

[2] Zachmanoglou, E. C. and Thoe, D. W., (1986). *Introduction to Partial Differential Equations with Applications.* New York: Dover Publications.

[3] Oppenheim, A. V. and Schafer, R. W., (1989). *Discrete-Time Signal Processing. Prentice-Hall Signal Processing Series.* Englewood Cliffs, NJ: Prentice-Hall.

[4] Jähne, B., (1997). *Handbook of Digital Image Processing for Scientific Applications.* Boca Raton, FL: CRC Press.

[5] Lim, J. S., (1990). *Two-dimensional Signal and Image Processing.* Englewood Cliffs, NJ: Prentice-Hall.

[6] Poularikas, A. D. (ed.), (1996). *The Transforms and Applications Handbook.* Boca Raton, FL: CRC Press.

[7] Pitas, I. and Venetsanopoulos, A. N., (1990). *Nonlinear Digital Filters. Principles and Applications.* Norwell, MA: Kluwer Academic Publishers.

[8] Knutsson, H. and Westin, C.-F., (1993). Normalized and differential convolution. In *Proceedings CVPR'93, New York City, NY*, pp. 515–523, IEEE. Washington, DC: IEEE Computer Society Press.