Fast Anisotropic Gauss Filtering

Jan-Mark Geusebroek, Arnold W. M. Smeulders, *Member, IEEE*, and Joost van de Weijer

***Abstract*—We derive the decomposition of the anisotropic Gaussian in a one-dimensional (1-D) Gauss filter in the x-direction followed by a 1-D filter in a nonorthogonal direction φ. So also the anisotropic Gaussian can be decomposed by dimension. This appears to be extremely efficient from a computing perspective. An implementation scheme for normal convolution and for recursive filtering is proposed. Also directed derivative filters are demonstrated.**

For the recursive implementation, filtering an 512 $\times$ 512 image is performed within 40 msec on a current state of the art PC, gaining over 3 times in performance for a typical filter, independent of the standard deviations and orientation of the filter. Accuracy of the filters is still reasonable when compared to truncation error or recursive approximation error.

The anisotropic Gaussian filtering method allows fast calculation of edge and ridge maps, with high spatial and angular accuracy. For tracking applications, the normal anisotropic convolution scheme is more advantageous, with applications in the detection of dashed lines in engineering drawings. The recursive implementation is more attractive in feature detection applications, for instance in affine invariant edge and ridge detection in computer vision. The proposed computational filtering method enables the practical applicability of orientation scale-space analysis.

***Index Terms*—Directional filter, feature detection, Gauss filter, Gaussian derivatives, orientation scale-space, tracking.**

I. INTRODUCTION

ONE OF THE most fundamental tasks in computer vision is the detection of edges and lines in images. The detection of these directional structures is often based on the local differential structure of the image. Canny's edge detector examines the magnitude of the first order image derivatives [1]. A well-founded approach for line detection is given by Steger [2], where line structures are detected by examining the eigenvectors of the Hessian matrix, the Hessian being given by the local second order derivatives. Robust measurement of image derivatives is obtained by convolution with Gaussian derivative filters, a well known result from scale-space theory [3], [4].

The difficulty of edge and line detection is emphasized when the structures run close together or cross each other, as is the case in engineering drawings or two-dimensional (2-D) projections of complex three-dimensional (3-D) scenes. In these cases, isotropic filtering strategies as used in e.g., [5], [1], [6], [2] are not sufficient. Isotropic smoothing causes parallel lines to be blurred into one single line. Crossing lines are not well detected by isotropic filters [7], due to the marginal orientation selectivity of the Gaussian filter. In these cases, one would often like to have a detection method which ignores the distorting data aside the edge or line, while accumulating evidence of the edge or line data along its orientation. Hence, taking advantage of the anisotropic nature of lines and edges. This implies a sampling of orientations by anisotropic filtering. For a linear orientation scale-space, the anisotropic Gaussian is the best suited causal filter [8].

Orientation analysis is often approached by steerable filters. Freeman and Adelson [9] put forward the conditions under which a filter can be tuned to a specific orientation by making a linear combination of basis filters. Their analysis included orientation tuning of the xy-separable first order isotropic Gaussian derivative filter. According to their framework, no exact basis exists for rotating an anisotropic Gaussian. Van Ginkel *et al.* proposed a deconvolution scheme for improving the angular resolution of the Gaussian isotropic filter. Under a linearity assumption on the input image, a steerable filter with good angular resolution is obtained. The method involves a Fourier based deconvolution technique, which is of high computational complexity. Perona [7] derived a scheme for generating a finite basis which approximates an anisotropic Gaussian. The scheme allowed both steering and scaling of the anisotropic Gaussian. However, the number of basis filters is large, and the basis filters are nonseparable, requiring high computational performance.

In this paper, we show the decomposition of the anisotropic Gaussian in two Gaussian line filters in non orthogonal directions (Section II). Choosing the x-axis to decompose the filter along turns out to be extremely efficient from a computing perspective. Hence, fast algorithms [10]–[13] can be used to calculate the orientation smoothed image and its derivatives. We combine the decomposition with the recursive algorithms proposed in [12], [13], yielding a constant calculation time with respect to the Gaussian scales and orientation (Section III). We give timing results and compare accuracy with 2-D convolution in Section IV.

II. SEPARATION OF ANISOTROPIC GAUSSIAN

The general case of an oriented anisotropic Gaussian filter in two dimensions is given by (Fig. 1)

$$g_\theta(u, v; \sigma_u, \sigma_v, \theta) = \frac{1}{\sqrt{2\pi}\,\sigma_u} \exp\left\{-\frac{1}{2}\frac{u^2}{\sigma_u^2}\right\} * \frac{1}{\sqrt{2\pi}\,\sigma_v} \exp\left\{-\frac{1}{2}\frac{v^2}{\sigma_v^2}\right\} \quad (1)$$

where "$*$" denotes convolution, and where

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (2)$$

Manuscript received July 29, 2002; revised January 20, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Isabelle Bloch.

The authors are with the Intelligent Sensory Information Systems, Department of Computer Science, Faculty of Science, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands (e-mail: mark@science.uva.nl).

Digital Object Identifier 10.1109/TIP.2003.812429

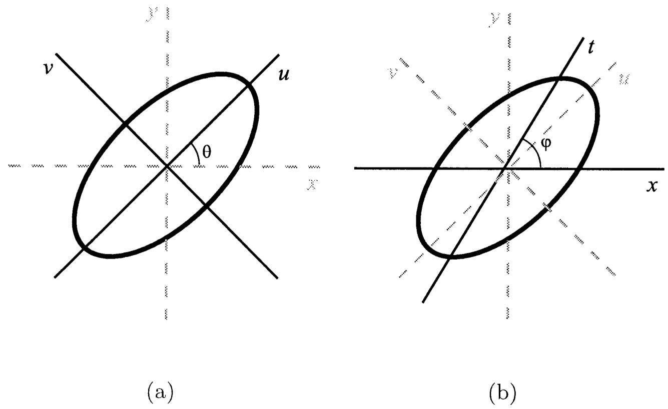

Fig. 1. Ellipse and its axes systems. An example of an anisotropic Gaussian with aspect ratio 1 : 2 and orientation $\theta = \pi/4$. (a) Principal axes uv-aligned Gaussian. (b) uv-aligned Gaussian in a nonorthogonal xt-axes system. Axis t is rotated over $\varphi \approx \pi/3$ with respect to the x-axis.

the u-axis being in the direction of θ, and the v-axis being orthogonal to θ.

As we are interested in a convenient basis from a computational perspective, separation in u and v is uninteresting. What is needed is the decomposition into a filter in the x-direction and a filter along another direction. Hence, we aim at separating the anisotropic Gaussian filter into

$$g_\theta(x, y; \sigma_u, \sigma_v, \theta) = \frac{1}{\sqrt{2\pi}\,\sigma_x} \exp\left\{-\frac{1}{2}\frac{x^2}{\sigma_x^2}\right\} * \frac{1}{\sqrt{2\pi}\,\sigma_\varphi} \exp\left\{-\frac{1}{2}\frac{t^2}{\sigma_\varphi^2}\right\} \quad (3)$$

representing the Gaussian filter along the x-direction, followed by filtering along a line $t = x\cos\varphi + y\sin\varphi$. The impulse response of Eq. (3) is given by

$$g_\theta(x, y; \sigma_u, \sigma_v, \theta) = \frac{1}{2\pi\sigma_x\sigma_\varphi} \cdot \exp\left\{-\frac{1}{2}\left(\frac{(x - y/\tan\varphi)^2}{\sigma_x^2} + \frac{(y/\sin\varphi)^2}{\sigma_\varphi^2}\right)\right\} \quad (4)$$

which should be equal to the impulse response of (1) to yield the proposed decomposition

$$g_\theta(u, v; \sigma_u, \sigma_v, \theta) = \frac{1}{2\pi\sigma_u\sigma_v} \exp\left\{-\frac{1}{2} \cdot \left(\frac{(x\cos\theta + y\sin\theta)^2}{\sigma_u^2} + \frac{(-x\sin\theta + y\cos\theta)^2}{\sigma_v^2}\right)\right\}. \quad (5)$$

Expanding the quadratic terms yields the system of equations

$$\frac{x^2}{\sigma_x^2} = x^2\frac{\cos^2\theta}{\sigma_u^2} + x^2\frac{\sin^2\theta}{\sigma_v^2} \quad (6)$$

$$\frac{y^2}{\sigma_x^2\tan^2\varphi} + \frac{y^2}{\sigma_\varphi^2\sin^2\varphi} = y^2\frac{\cos^2\theta}{\sigma_v^2} + y^2\frac{\sin^2\theta}{\sigma_u^2} \quad (7)$$

$$-\frac{2xy}{\sigma_x^2\tan\varphi} = 2xy\cos\theta\sin\theta\left(\frac{1}{\sigma_u^2} - \frac{1}{\sigma_v^2}\right). \quad (8)$$

Solving the equations yields the decomposition of the anisotropic Gaussian into a Gaussian along the x-axis, with standard deviation

$$\sigma_x = \frac{\sigma_u\sigma_v}{\sqrt{\sigma_v^2\cos^2\theta + \sigma_u^2\sin^2\theta}} \quad (9)$$

and a Gaussian along the line t: $y - x\tan\varphi = 0$, with standard deviation

$$\sigma_\varphi = \frac{1}{\sin\varphi}\sqrt{\sigma_v^2\cos^2\theta + \sigma_u^2\sin^2\theta} \quad (10)$$

and intercept

$$\tan\varphi = \frac{\sigma_v^2\cos^2\theta + \sigma_u^2\sin^2\theta}{(\sigma_u^2 - \sigma_v^2)\cos\theta\sin\theta}. \quad (11)$$

Note that the $1/\sin\varphi$ term in (10) vanishes in (4).

So we have achieved our goal namely that a Gauss filter at arbitrary orientation is decomposed into a 1-D Gauss filter with standard deviation σ_x and another 1-D Gauss filter at orientation φ and standard deviation σ_φ. For the isotropic case $\sigma_u = \sigma_v = \sigma$, it is verified easily that $\sigma_x = \sigma$, $\sigma_\varphi = \sigma$. Further, for $\theta = 0$, trivially $\sigma_x = \sigma_u$, $\sigma_\varphi = \sigma_v$, and $\tan\varphi = 0$, and for $\theta = \pi/2$, $\sigma_x = \sigma_v$, $\sigma_\varphi = \sigma_u$, and $\tan\varphi = 0$. An arbitrary example orientation of $\theta = \pi/4$ and $\sigma_v = \sigma$, $\sigma_u = 2\sigma$, results in $\sigma_x = (2/5)\sqrt{10}\sigma$, $\sigma_\varphi = \sqrt{3.4}\sigma$, and $\tan\varphi = (5/3)$ ($\varphi \approx \pi/3$), see Fig. 1(b).

III. Implementation

Implementation of (3) boils down to first applying a 1-D Gaussian convolution in the x-direction. The resulting image is then convolved with a 1-D Gaussian in the φ-direction yielding the anisotropic smoothed image. The latter step implies interpolation, which can be achieved by linear interpolation between two neighboring x-pixels on the crossing between the image x-line of interest and the t-axis [see Fig. 1(b)]. In this section, we consider two implementations of the anisotropic Gaussian, based on a common convolution operation, and based on a recursive filter [12], respectively.

Convolution Filter

Due to the filter symmetry, the x-filter can be applied by adding pixel i left from the filter center with pixel i right from the filter center, and multiplying the summed pixels with filter weight i, or

$$g_x[x, y] = w_0 f[x, y] + \sum_{i=1}^{\lfloor N/2 \rfloor} w_i\left(f[x - i, y] + f[x + i, y]\right). \quad (12)$$

Here, $f[x, y]$ is the input image, w_i is the filter kernel for half the sampled Gaussian from 0 to $\lfloor N/2 \rfloor$, and $g_x[x, y]$ is the filtered result image.

Filtering along the line t with intercept $\mu = \tan\varphi$ is achieved by a sheared filter

$$g_\theta[x, y] = w_0 g_x[x, y] + \sum_{j=1}^{\lfloor M/2 \rfloor} w_j\left(g_x[x - j/\mu, y - j] + g_x[x + j/\mu, y + j]\right). \quad (13)$$

Notice that the $y \pm j$ coordinate falls exactly on an image line, whereas the $x \pm j/\mu$ coordinate may fall between two pixels. Hence, the value of the source pixel may be obtained by interpolating between the two pixels at the line of interest. To achieve our goal of fast anisotropic filtering, we consider linear interpolation between the neighboring pixels at $x \pm j/\mu$ with interpolation coefficient a. The filter equation then becomes

$$\begin{aligned} g_\theta[x, y] = {} & w_0 g_x[x, y] + \sum_{j=1}^{\lfloor M/2 \rfloor} w_j \{a(g_x[\lfloor x - j/\mu \rfloor, y - j] \\ & + g_x[\lfloor x + j/\mu \rfloor, y + j]) \\ & + (1-a)(g_x[\lfloor x - j/\mu \rfloor - 1, y - j] \\ & + g_x[\lfloor x + j/\mu \rfloor + 1, y + j])\}. \end{aligned} \tag{14}$$

The multiplication of $w_j a$ and $w_j(1-a)$ can be taken out of the loop to reduce the computational complexity of the filter. Hence, before filtering of the image, two tables of pre-calculated filter coefficients are combined with the interpolation factors a and $1-a$, respectively. During filtering, pixels are weighted with these values, and accumulated to result in the filtered and interpolated output value.

Recursive Filter

Rather than applying convolution operators, (3) may be implemented by recursive filters. Van Vliet *et al.* [12], [13] define a scheme for 1-D Gaussian filtering with infinite support. The recursive filter requires only seven multiplications per pixel, an improvement over [11]. The complexity is independent of the Gaussian standard deviation σ. In [13] it is shown that the recursive filter is faster than its normal counterpart for $\sigma > 1$. When using the recursive filter, filtering along the x-line is given by the forward and backward filter pair

$$\begin{aligned} g_x^f[x, y] &= f[x, y] - a_1 g_x^f[x-1, y] - a_2 g_x^f[x-2, y] \\ &\quad - a_3 g_x^f[x-3, y] \\ g_x^b[x, y] &= a_0^2 g_x^f[x, y] - a_1 g_x^b[x+1, y] - a_2 g_x^b[x+2, y] \\ &\quad - a_3 g_x^b[x+3, y]. \end{aligned} \tag{15}$$

Here, a_i represent the filter coefficients as given by [12], [13], and $g_x^b[x, y]$ is the x-filtered result image. The computational complexity of the recursive filter is 7 multiplications per pixel.

Filtering along the line t with intercept $\mu = \tan\varphi$ is achieved by a sheared recursive filter

$$\begin{aligned} g_\theta^f[x + y/\mu, y] &= g_\theta^f[t] \\ &= g_x^b[x + y/\mu, y] - a_1 g_\theta^f[t-1] \\ &\quad - a_2 g_\theta^f[t-2] - a_3 g_\theta^f[t-3] \\ g_\theta[x + y/\mu, y] &= g_\theta^b[t] \\ &= a_0^2 g_\theta^f[x + y/\mu, y] - a_1 g_\theta^b[t+1] \\ &\quad - a_2 g_\theta^b[t+2] - a_3 g_\theta^b[t+3]. \end{aligned} \tag{16}$$

Note that (x, y) are constraint to lie on the line t, hence may point to positions "between" pixels. Since interpolation of the recursive filter values is not possible, the filter history $g_\theta^f[t]$ and $g_\theta^b[t]$ has to be buffered, such that all t values are at the buffer "grid." The input values, $g_x^b[x + y/\mu, y]$ for the forward filter and $g_\theta^f[x + y/\mu, y]$ for the backward filter, are linearly interpolated from the two input pixels on the left and right of the exact location. The results $g_\theta^f[x, y]$ and $g_\theta[x, y]$ are interpolated to the output pixel grid by combining with the previous result. Since all pixels are at the exact line position, interpolation can be performed linearly between the current value and the previous value.

Computational complexity of the proposed implementations and a few common methods for Gaussian convolution is shown in Table I. The table indicates computational complexity for several solutions of anisotropic Gaussian filtering. For anisotropic Gaussian filtering oriented along the xy-axes, resulting in a fixed orientation. In this case, no interpolation is necessary. For filtering along a uv-axes system, bilinear interpolation results in a 4 times higher complexity than the xy-aligned filtering. Improvement is obtained when using the proposed decomposition along a xt-axes system. In the latter case, no interpolation is needed for the x-filter step, resulting in $N/2$ multiplications, whereas linear interpolation is necessary for the t-filter step (unit steps along y-axis, interpolation at x-axis), resulting in M multiplications. Hence, improvement due to interpolation is over 50% compared to the uv-separated filter, with identical outcome.

IV. RESULTS

Performance of the filter with respect to computation speed is shown in Table II. The analysis was carried out on a Pentium III at 800 MHz for a 512 × 512 image. The maximum calculation time for the proposed xt-separable recursive implementation was 40 msec. Small variations in the computation time for the xt-separable recursive implementation is due to the varying direction of the t-axis as function of σ_u, σ_v. The variation causes the processing of different pixels with respect to the filter origin, hence are influenced by the processor cache performance. The use of recursive filters is already beneficial for $\sigma_u > 1$ or $\sigma_v > 1$. The results correspond to the predictions in Table I. The xt-recursive filter is almost two times faster than the uv separable recursive filter. For a common value of $\sigma_u = 5$ and $\sigma_v = 1$, the xt-recursive implementation is 3.25 times faster than the standard method of uv-separable convolution filtering. Even for the xt-separable convolution filter, calculation is up to two times faster than uv-separable filtering. Normal convolution filtering is advantageous when considering locally steered filtering, as in tracking applications, for example Fig. 2. The recursive filtering is, given its computation speed, more attractive when smoothing or differentiating the whole image array, as in feature detection, shown in Fig. 3.

The approximation of the 2-D Gaussian kernel of (1) by separable filters is not perfect due to interpolation of source values along the line $t = y + \tan\varphi\, x$. We evaluated the error for the xt-separable convolution filter in comparison to the full 2-D spatial convolution. The results are given in Table III. Interpolation can be considered as a smoothing step with a small rectangular kernel. Hence, the effective filter is slightly larger than

TABLE I
COMPLEXITY PER PIXEL OF VARIOUS ALGORITHMS FOR GAUSSIAN SMOOTHING. FILTER SIZE IS DENOTED BY $N \times M$, DEPENDING ON THE GAUSSIAN STANDARD DEVIATION σ

Filter type	*Separability*	*Complexity*	
		Multiplications	*Additions*
convolution	xy^1	$\lfloor N/2 \rfloor + \lfloor M/2 \rfloor + 2$	$N + M - 2$
	uv^2	$2(N + M - 1)$	$2(N + M - 2)$
	xt^2	$\lfloor N/2 \rfloor + M + 1$	$N + 2M - 3$
recursive	xy^1	14	6
	uv^2	44	36
	xt^2	21	16
2D convolution	n.a.	NM	$NM - 1$
FFT convolution[3]	n.a.	$\log WH$	$\log WH$

[1] Restricted to Gaussian filters oriented along the x- and y-axis only, thus $\theta = 0°$ or $\theta = 90°$.

[2] Unrestricted θ.

[3] The complexity of a FFT based convolution depends on the image size $W \times H$.

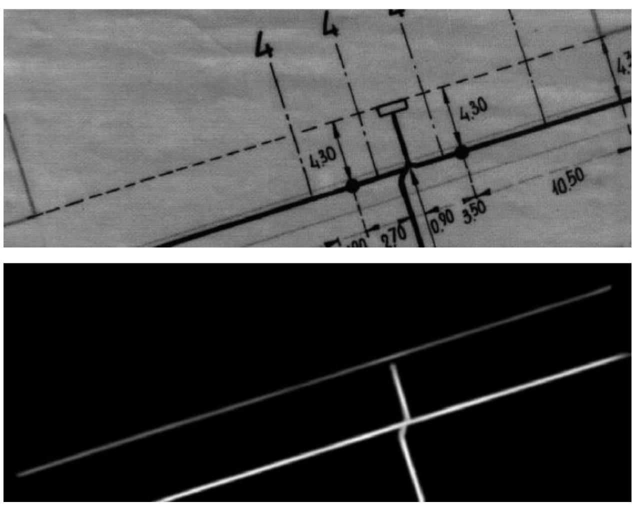

Fig. 2. Example of line detection by local anisotropic Gaussian filtering. Lines are tracked by steering the filter in the line direction. Hence, line evidence will be integrated by the large Gaussian standard deviation along the line, while maintaining spatial acuity perpendicular to the line. Original from an engineering drawing, courtesy of PNEM, The Netherlands.

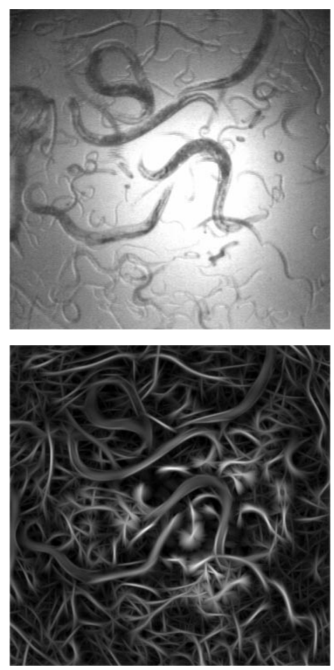

Fig. 3. Example of the detection of *C. Elegans* worms by applying recursive anisotropic Gauss filters. The original image is filtered at different orientations and scales, and the maximum response per pixel over all filters is accumulated. At each pixel, the local orientation and best fitting ellipse is available to be further processed for worm segmentation. Computation time was within 10 s for 5° angular resolution and three different aspect ratios (image size 512 × 512 pixels). Original courtesy of Janssen Pharmaceuticals, Beerse, Belgium.

the theoretical size of the anisotropic Gaussian filter. As a result, the error is large for small σ_u, σ_v, as can be concluded from the table. For the convolution filters and $\sigma_u, \sigma_v \geq 3$, the interpolation error is of the same magnitude as the truncation error for a 3σ sized filter (last four rows in the table). The interpolation error is smaller for the xt-filter than for the uv-filter. For the latter, bilinear interpolation have to be performed, corresponding to a larger interpolation filter than the linear interpolation for the xt-separable filter. For the recursive filter, the interpolation error of the forward filter accumulates in the backward filter, causing a larger error. Especially the small filters are less accurate, as pointed out in [12], [13]. Note that the error due to interpolation is negligible compared to the error made by the recursive approximation of the Gaussian filter. For the uv-separated recursive filter, the bilinear interpolation caused the error accumulation to have such a drastic effect that the result was far from Gaussian (data not shown). In conclusion, accuracy for

TABLE II
PERFORMANCE OF VARIOUS ANISOTROPIC GAUSSIAN FILTER IMPLEMENTATIONS

		Standard		*This paper*							
σ_u	σ_v	*2D convolution*[1]	*FFT convolution*	*1D convolution*[1] *uv*[3]			*1D recursive*[2]				
				uv[3]	*xt*		*uv*		*xt*		
2.0	1.0	310	760	72	57	(1.3)	76	(0.9)	40	(1.8)	
3.0	**1.0**	**640**	**760**	**90**	**62**	**(1.5)**	**75**	**(1.2)**	**40**	**(2.3)**	
5.0	**1.0**	**1950**	**760**	**130**	**72**	**(1.8)**	**75**	**(1.7)**	**40**	**(3.3)**	
7.0	**2.0**	**2390**	**760**	**190**	**91**	**(2.1)**	**75**	**(2.5)**	**40**	**(4.8)**	
7.0	4.0	3000	760	230	105	(2.2)	76	(3.0)	39	(5.9)	
10.0	3.0	4880	760	265	115	(2.3)	77	(3.4)	39	(6.8)	
10.0	5.0	5650	760	302	128	(2.4)	74	(4.1)	40	(7.6)	
10.0	7.0	6570	760	346	147	(2.4)	73	(4.7)	40	(8.7)	

All timings in [msec], averaged over 100 trials, improvement factors given between brackets. Image size 512×512 pixels. Filter direction $\theta = 45°$. Results for typical filter sizes in bold.

[1]Filter sizes truncated at 3σ.

[2]Approximation to Gauss, see Table V.

[3]Considered as reference for speed improvement factors.

the xt-separated convolution filter is better than bilinear interpolation combined with uv-separated filtering. For recursive filtering, error is larger due to the recursive approximation of the Gauss filter. For numerous applications the computation speed is of more importance than the precision of the result.

TABLE III
ACCURACY OF VARIOUS ANISOTROPIC GAUSSIAN FILTER IMPLEMENTATIONS. THE MAXIMUM ERROR OVER ALL FILTER ORIENTATIONS IS SHOWN. ERROR MEASURED AS ROOT OF THE SUM SQUARED DIFFERENCES WITH THE TRUE GAUSSIAN KERNEL

σ_u	σ_v	*convolution uv*	*convolution xt*	*recursive xt*
2.0	1.0	0.0160	0.0131	0.0536
3.0	1.0	0.0126	0.0114	0.0324
5.0	2.0	0.0018	0.0017	0.0062
7.0	2.0	0.0015	0.0014	0.0050
7.0	4.0	0.0003	0.0003	0.0012
10.0	3.0	0.0005	0.0004	0.0017
10.0	5.0	0.0001	0.0001	0.0008
10.0	7.0	0.0001	0.0001	0.0007

V. CONCLUSION

We derived the decomposition of the anisotropic Gaussian in a 1-D Gauss filter in the x-direction followed by a 1-D filter in a nonorthogonal direction φ. The decomposition is shown to be extremely efficient from a computing perspective. An implementation scheme for normal convolution and for recursive filtering is proposed. Also directed derivative filters are demonstrated.

We proposed a scheme for both anisotropic convolution filtering and anisotropic recursive filtering. Convolution filtering is advantageous when considering locally steered filtering, as is the case in tracking applications [14], [15]. Recursive filtering is more attractive when smoothing or differentiating the whole image array, for example in feature detection [1], [2], [4]. Error due to interpolation is negligible compared to the error made by the recursive approximation of the Gaussian filter, and compared to the truncation error for convolution filters. The use of fast recursive filters [12], [13] result in an calculation time of 40 ms for a 512×512 input image on a current state-of-the-art PC.

Differentiation opposite to or along the filter direction is achieved by convolution with a rotated sample difference filters. For practical applicability of orientation scale-space analysis, we believe the exact approximation of Gaussian derivatives is of less importance than the ability to compute results in limited time.

Although the decomposition of (1) is possible in higher dimensions, the method is less beneficial for 3-D filtering applications. Only one of the axes can be chosen to be aligned with the organization of the pixels in memory. For the other directions, traversing in arbitrary directions through the pixel data is required. Hence, computational gain is only marginal for higher dimensional smoothing.

The proposed anisotropic Gaussian filtering method allows fast calculation of edge and ridge maps, with high spatial and angular accuracy, improving computation speed typically by a factor 3. The anisotropic filters can be applied in cases where edge and ridge data is distorted. Invariant feature extraction from a 2-D affine projection of a 3-D scene can be achieved by tuning the anisotropic Gaussian filter, an important achievement for computer vision. When structures are inherently interrupted, as is the case for dashed line detection, anisotropic Gaussian filter may accumulate evidence along the line while maintaining spatial acuity perpendicular to the line. Orientation scale-space analysis can best be based on anisotropic Gaussian filters [16]. The proposed filtering method enables the practical applicability of orientation scale-space analysis.

References

[1] F. J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, no. 6, pp. 679–698, 1986.

[2] C. Steger, "An unbiased detector of curvilinear structures," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 113–125, 1998.

[3] J. J. Koenderink, "The structure of images," *Biol. Cybern.*, pp. 363–370, 1984.

[4] T. Lindeberg, *Scale-Space Theory in Computer Vision*. Norwell, MA: Kluwer, 1994.

[5] J. Bigün, G. H. Granlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optic flow," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 775–790, 1991.

[6] S. Kalitzin, B. ter Haar Romeny, and M. Viergever, "Invertible orientation bundles on 2d scalar images," in *Scale-Space Theories in Computer Vision*: Springer-Verlag, 1997, pp. 77–88.

[7] P. Perona, "Steerable-scalable kernels for edge detection and junction analysis," *Image Vis. Comput.*, vol. 10, pp. 663–672, 1992.

[8] J. J. Koenderink and A. J. van Doorn, "Receptive field families," *Biol. Cybern.*, vol. 63, pp. 291–297, 1990.

[9] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 891–906, 1991.

[10] R. Deriche, "Separable recursive filtering for efficient multi-scale edge detection," in *Proc. Int. Workshop on Machine Vision and Machine Intelligence*, 1987, pp. 18–23.

[11] ——, "Fast algorithms for low-level vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 78–87, 1990.

[12] L. J. van Vliet, I. T. Young, and P. W. Verbeek, "Recursive Gaussian derivative filters," in *Proc. ICPR '98*, 1998, pp. 509–514.

[13] I. T. Young and L. J. van Vliet, "Recursive implementation of the Gaussian filter," *Signal Process.*, vol. 44, pp. 139–151, 1995.

[14] E. P. Simoncelli, "Distributed representation and analysis of visual motion," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Mass. Inst. Technol., Cambridge, MA, 1993.

[15] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger, "Probability distributions of optical flow," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 1991, pp. 310–315.

[16] M. van Ginkel, P. W. Verbeek, and L. J. van Vliet, "Improved orientation selectivity for orientation estimation," in *Proc. 10th Scandinavian Conf. Image Analysis*, M. Frydrych, J. Parkkinen, and A. Visa, Eds., 1997, pp. 533–537.

Jan-Mark Geusebroek received the Ph.D. degree in computer sciences from the University of Amsterdam, Amsterdam, The Netherlands, in 2000.

He is Assistant Professor in the Intelligent Sensory Information Systems (ISIS) Group, University of Amsterdam. His research interest are in front-end vision, especially color and texture vision. His current research concentrates on computational theories for cognitive vision, based on invariant representations and visual attention.

Arnold W. M. Smeulders (S'80–M'80) is Full Professor in multimedia information analysis with a special interest in content-based image retrieval systems as well as systems for the analysis of video. In addition, he is director of the Informatics Research Institute at the University of Amsterdam, Amsterdam, The Netherlands. He heads the ISIS Research Group of 25 persons which concentrate on theory, practice, and implementation of image retrieval and computer vision. The group has an extensive record in co-operations with Dutch industry in the area of multimedia and video analysis.

Dr. Smeulders was an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He received a Fulbright Grant at Yale University in 1987, a visiting professorship at the City University Hong Kong in 1996, and ETL Tsukuba, Japan, in 1998. He is elected fellow of International Association of Pattern Recognition. His current research interest is in image retrieval, especially perceptual similarity, material recognition, and the connection between pictures and language.

Joost van de Weijer received the M.Sc. degree in applied physics at the Delft University of Technology, Delft, The Netherlands, in 1998. Since 1999, he is a Ph.D. student in the ISIS Group at the University of Amsterdam.

His current research interests include filter theory, color image filtering and photometric invariance.

Mr. van de Weijer is a member of the IEEE Signal Processing Society.