APPROXIMATE SEPARABLE 3D ANISOTROPIC GAUSS FILTER

Oliver Wirjadi

Fraunhofer ITWM
Gottlieb-Daimler-Strasse
67663 Kaiserslautern, Germany
wirjadi@itwm.fraunhofer.de

Thomas Breuel

University of Kaiserslautern
Department of Computer Science
Erwin-Schrödinger-Strasse
67663 Kaiserslautern, Germany

ABSTRACT

Anisotropic Gaussian filters are useful for adaptive smoothing and feature extraction. In our application, micro - tomographic images of fibers were smoothed by anisotropic Gaussians. In this case, this is more natural than using their isotropic counterparts. But filtering in large 3D data is very time consuming. We extend the work of Geusebroek et al. on fast Gauss filtering to three dimensions [1, 2]. We propose an approximate separable filtering scheme which consists of three 1D convolutions. Initial experiments suggest that this filter can outperform an FFT based implementation when the kernel size is small compared to the size of the 3D images.

1. INTRODUCTION

Image smoothing is usually performed using isotropic, i.e. rotationally invariant, filters. Anisotropic filtering has, for example, been applied to medical imaging [3] and line drawings [2]. In our case, the task is to smooth micro - tomographic images of fibrous materials in order to pre-process these 3D datasets.

We follow the approach in [1], who derived an exact separable anisotropic Gauss filter in two dimensions. Their result shows that a 2D Gaussian, oriented at an arbitrary angle θ, may be constructed by convolving two one dimensional Gaussians: The first one along the x-axis, the second one along a line t who's direction is dependent on the variances and direction of the kernel. Other authors have also proposed approximations to separable filters in 2D [4, 5].

In the following, we will first derive an approximation to a separable anisotropic Gaussian in three dimensions, cf. Sec. 2, which is analyzed in Sec. 3. Section 4 describes our preliminary image-domain implementation. Performance measurements and comparison to an FFT based convolution are given in Sec. 5.

2. DERIVATION

Our goal is to perform noise smoothing on fibrous structures, for which prolate ellipsoids are a reasonable model. We propose to use a Gaussian smoothing kernel with one major and two minor directions, thus defining its covariance matrix Σ as

$$\Sigma = \begin{pmatrix} \sigma_M^2 & 0 & 0 \\ 0 & \sigma_m^2 & 0 \\ 0 & 0 & \sigma_m^2 \end{pmatrix}. \qquad (1)$$

This defines a Gaussian that is elongated along the x-axis and has equal extent in the two remaining orthogonal directions. As a consequence of the equal variance in y- and z-direction, we can orient this kernel arbitrarily in 3D Euclidean space by using just two rotations, φ_y and φ_z, about the y and z axis, respectively. This yields the following coordinate transformation.

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \mathbf{R}_z(\varphi_z)\mathbf{R}_y(\varphi_y) \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \qquad (2)$$

$$\begin{pmatrix} \cos\varphi_y \cos\varphi_z & \sin\varphi_z & -\sin\varphi_y \cos\varphi_z \\ -\cos\varphi_y \sin\varphi_z & \cos\varphi_z & \sin\varphi_y \sin\varphi_z \\ \sin\varphi_y & 0 & \cos\varphi_y \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Combining (1) and (2), we can write the kernel g as

$$g(x, y, z; \sigma_m^2, \sigma_M^2, \varphi_y, \varphi_z) \quad = \quad N_{u,v,w}(0, \Sigma), \qquad (3)$$

where N denotes the density function of the normal distribution with mean and covariance parameters. For our derivation, we need the following two results.

Lemma 1 (Separable Filters [2]) *A convolution filter h is separable, i.e. can be constructed from a sequence of convolutions in lower dimensions, if and only if its Fourier transform factorizes:*

$$h(x, y, z) = h_x(x) * h_y(y) * h_z(z) \Leftrightarrow$$
$$H_{\omega_x}(\omega_x) H_{\omega_y}(\omega_y) H_{\omega_z}(\omega_z) = H(\omega_x, \omega_y, \omega_z),$$

where H denotes the Fourier transform of h.

Lemma 2 (Fourier Transform of Gaussian [6]) *The Fourier transform of a Gaussian is proportional to a Gaussian function,*

$$\mathcal{F}(N(\mathbf{a},\mathbf{A})) \propto N(j\mathbf{A}^{-1}a, \mathbf{A}^{-1}).$$

Next, as in [2], we will use Lemma 2 to transform g and rearrange terms in the Fourier domain such that the term factorizes. From Lemma 1 we will then get a separable Gaussian. Taking the Fourier transform of (3) yields

$$\begin{aligned}
&\mathcal{F}(N_{u,v,w}(0,\Sigma)) = N_{\omega_u,\omega_v,\omega_w}(0,\Sigma^{-1}) \\
&= \exp\left(-\frac{1}{2}(\omega_u,\omega_v,\omega_w)\Sigma(\omega_u,\omega_v,\omega_w)^T\right) \\
&= \exp\left(-\frac{1}{2}(\sigma_M^2\omega_u^2+\sigma_m^2\omega_v^2+\sigma_m^2\omega_w^2)\right) \qquad (4)
\end{aligned}$$

The Gaussian in (4) factorizes in the u, v and w directions, but we would like to obtain a separation along the original coordinate system directions. By expanding (4) and grouping terms we get the following.

$$\begin{aligned}
&\mathcal{F}(N_{u,v,w}(0,\Sigma)) = \\
&\exp\left(-\frac{1}{2}a_{11}\omega_x^2-\frac{1}{2}a_{22}\omega_y^2-\frac{1}{2}a_{33}\omega_z^2\right) \\
&\times\exp\left(-a_{12}\omega_x\omega_y-a_{13}\omega_x\omega_z-a_{23}\omega_y\omega_z\right), \qquad (5)
\end{aligned}$$

where

$$\begin{aligned}
a_{11} &= \sigma_M^2\cos^2\varphi_y\cos^2\varphi_z+\sigma_m^2\cos^2\varphi_y\sin^2\varphi_z+ \\
&\quad\ \sigma_m^2\sin^2\varphi_y \\
a_{22} &= \sigma_M^2\sin^2\varphi_z+\sigma_m^2\cos^2\varphi_z \\
a_{33} &= \sigma_M^2\sin^2\varphi_y\cos^2\varphi_z+\sigma_m^2\sin^2\varphi_y\sin^2\varphi_z+ \\
&\quad\ \sigma_m^2\cos^2\varphi_y \\
a_{12} &= (\sigma_M^2-\sigma_m^2)\cos\varphi_y\cos\varphi_z\sin\varphi_z \\
a_{13} &= (\sigma_m^2-\sigma_M^2)\cos\varphi_y\sin\varphi_y\cos^2\varphi_z \\
a_{23} &= (\sigma_m^2-\sigma_M^2)\sin\varphi_y\cos\varphi_z\sin\varphi_z \qquad (6)
\end{aligned}$$

All cross terms vanish from (5) when $\sigma_m^2=\sigma_M^2$, implying that the function is separable along the original coordinate axes in the isotropic case. Our goal is to rearrange (5) in such a way that we get a filter that factorizes into terms representing Gaussian kernels along lines in 3D space. Quadratic expansion yields two Gaussians in x and z direction, plus a remaining term, which we *approximate* by a 1D Gaussian.

$$\begin{aligned}
&\mathcal{F}(N_{u,v,w}(0,\Sigma)) = \\
&\exp\left(-\frac{1}{2}a_{11}\omega_x^2-\frac{1}{2}a_{22}\omega_y^2-\frac{1}{2}a_{33}\omega_z^2\right) \\
&\times\exp\left(-a_{12}\omega_x\omega_y-a_{13}\omega_x\omega_z-a_{23}\omega_y\omega_z\right) \\
&\times\exp\left(\frac{a_{12}^2}{a_{22}}\omega_x^2-\frac{a_{12}^2}{a_{22}}\omega_x^2+\frac{a_{23}^2}{a_{22}}\omega_z^2-\frac{a_{23}^2}{a_{22}}\omega_z^2\right) \\
&= \exp\left(-\frac{1}{2}(a_{11}-\frac{a_{12}^2}{a_{22}})\omega_x^2\right) \\
&\times\exp\left(-\frac{1}{2}(a_{33}-\frac{a_{23}^2}{a_{22}})\omega_z^2\right) \\
&\times\exp\left(-\frac{1}{2}(\frac{a_{12}^2}{a_{22}}\omega_x^2+a_{22}\omega_y^2+\frac{a_{23}^2}{a_{22}}\omega_z^2)\right) \\
&\times\exp\left(-\frac{1}{2}(2a_{12}\omega_x\omega_y+2a_{13}\omega_x\omega_z+2a_{23}\omega_y\omega_z)\right) \\
&\approx \exp\left(-\frac{1}{2}(a_{11}-\frac{a_{12}^2}{a_{22}})\omega_x^2\right) \\
&\times\exp\left(-\frac{1}{2}(a_{33}-\frac{a_{23}^2}{a_{22}})\omega_z^2\right) \\
&\times\exp\left(-\frac{a_{22}}{2}(\omega_y+\frac{a_{12}}{a_{22}}\omega_x+\frac{a_{23}}{a_{22}}\omega_z)^2\right) \\
&=: \tilde{\mathcal{F}}(N) \qquad (7)
\end{aligned}$$

Eq. (7) represents two convolutions along the x and z axes, with their support determined by the variables' coefficients. The third exponential represents convolution with a Gaussian along a line l that is not aligned with the xyz coordinate axes. The parametric form of that line is given by $\mathbf{x}=t\,[1,-a_{22}/(a_{12}+a_{23}),1]^T$, $t\in\mathbb{R}$. An analysis of this approximation will be given in the following section.

3. ANALYSIS

Next, we give an expression for the relative error introduced by the approximation made above.

$$\log(\tilde{\mathcal{F}}(N))-\log(\mathcal{F}(N))=\omega_x\omega_z(a_{13}-\frac{a_{12}a_{23}}{a_{22}}) \qquad (8)$$

The error term in (8) shows that the above derived separable filter differs from the true anisotropic 3D Gaussian by an exponential in the xz plane which is convolved with the true function. When substituting the coefficients a_{12}, a_{23} and a_{13} into (8) we see three cases where $\log(\tilde{\mathcal{F}}(N))-\log(\mathcal{F}(N))=0$: (1) $\sigma_m^2=\sigma_M^2$; (2) $\varphi_y=k\pi/2$, $k\in\mathbb{N}$, independent of the rotation about the z axis; and (3) $\varphi_z=\pi/2+k\pi$, $k\in\mathbb{N}$, independent of the rotation about the y axis. We analyzed the error power, i.e. $|\mathcal{F}(N)-\tilde{\mathcal{F}}(N)|^2$, depending on the filter parameters and show results in Fig.

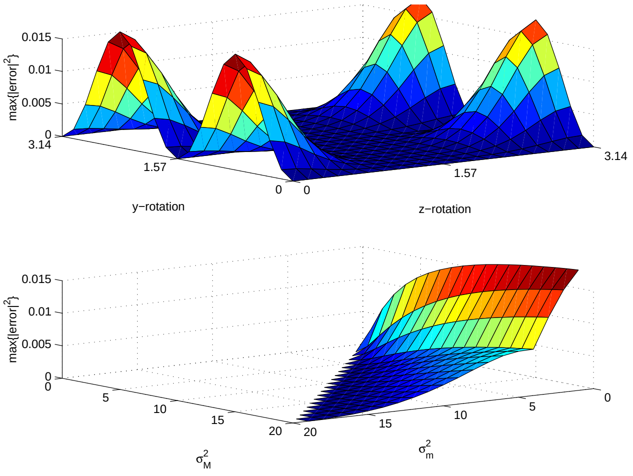

Fig. 1: Maximum error power depending on rotation angles (upper plot) and variances (lower plot).

1. The maximum error in $(\omega_x, \omega_y, \omega_z)$ is plotted. For the rotation error plot, parameters $\sigma_m^2 = 5$ and $\sigma_M^2 = 10$ were fixed. For the variance error plot, parameters $\varphi_y = \pi/4$ and $\varphi_z = 3\pi/4$ were set. A range about $\pi/2$ wide in φ_z direction with very low error, as well as the tendency for the error to decrease with increasing variance parameters can be seen.

For efficient implementation of this separable filter, the support of the 1D Gaussians should not be too large. To analyze this we once again use Lemma 2 to transform (7) back to image space.

$$\begin{aligned}&\tilde{g}(x,y,z;\sigma_M^2,\sigma_m^2,\varphi_y,\varphi_z) = \mathcal{F}^{-1}(\tilde{\mathcal{F}}(N))\\&\propto \exp\left\{-\frac{1}{2}\frac{x^2}{a_{11}-\frac{a_{12}^2}{a_{22}}}\right\} * \exp\left\{-\frac{1}{2}\frac{z^2}{a_{33}-\frac{a_{23}^2}{a_{22}}}\right\}\\&* \exp\left\{-\frac{1}{2}\frac{(y+\frac{a_{12}}{a_{22}}x+\frac{a_{23}}{a_{22}}z)^2}{a_{22}}\right\}\end{aligned} \qquad (9)$$

The standard deviations of the 1D Gaussians are thus given by the square roots of the denominators in (9). Similar to [2] we can find expressions for the standard deviations in the special case $\sigma_M = 2\sigma_m$, $\varphi_y = \varphi_z = \pi/4$: $\sigma_x = \sigma_z = \sqrt{11/8}\sigma_m$ and $\sigma_l = \sqrt{a_{22}} = \sqrt{3}\sigma_m$. This shows that in this selected case the standard deviations of the 1D Gaussians of the proposed approximate filter grow only linearly with the selected σ_m. The covariance matrix of $\tilde{g}$ becomes diagonal for $\varphi_z = 0$, thus the peaks in Fig. 1.

From a computational perspective, the advantage of separable filters is the following: A non-separable 3D convolution requires one multiplication and one addition for each voxel within the window. Suppose the mask size is $M \times M \times M$, then we need $\mathrm{O}(M^3)$ multiplications and additions. In the separable case, we need to convolve along lines only. This means that we need $3M$ additions and multiplications for the first line and equivalently for the two remaining lines. This leads to only $\mathrm{O}(M)$ multiplications and additions per voxel for the separable case.

4. IMPLEMENTATION

Eq. (9) was implemented in C++. For good performance, it is important that the locality of the image buffers in memory matches the convolution directions in (9). The 1D convolutions are performed using sliding windows. For the line t, the third part of the filter, we use Bresenham's line drawing algorithm to get a line along which 1D convolutions are performed. Line extraction is only necessary once since all subsequent lines can be obtained by incrementing the coordinates of that initial line. In all three dimensions, we use reflective edge treatment. At the time of writing, the implementation was not fully optimized, yet, with much room for improvement in data structures and image voxel access, which is most time consuming.

The kernel support is set to three standard deviations in each direction. Parallelization of the algorithm was not considered, but would be possible: The image may be divided into a number of sections and the line convolutions may then be calculated for each section separately. Termination of each stage (x, y and l) is required before proceeding to the next phase. For the results presented in the next section, parallelization is not relevant since the FFT is also parallelizable and improvements should cancel one another so that the relative results remain valid. Geusebroek et al. also implemented a recursive IIR filter [7] which we did not consider for this work.

5. RESULTS

We compared runtimes of our filter with the runtimes of an FFT based convolution. For that, we used the fftw library in single precision mode [8]. Runtimes of the non-separable convolution were so far higher than those of either the separable or FFT based method that we did not include them in the plot below.

Fig. 2 shows the runtimes of our method and the FFT based convolution for a range of image and kernel sizes. In the figure, mask sizes $25 \times 13 \times 13$ and $13 \times 7 \times 7$ correspond to filter parameters $2\sigma_m^2 = \sigma_M^2 = 4$ and $2\sigma_m^2 = \sigma_M^2 = 2$, i.e. truncation of the Gaussian at 3 standard deviations. Plotted are 10% trimmed mean values from 50 trial runs each. The runtimes of the FFT vary significantly because of more efficient FFT computation being possible for certain image sizes. For the FFT runtimes, we measured forward and inverse transform of the image plus multiplication in

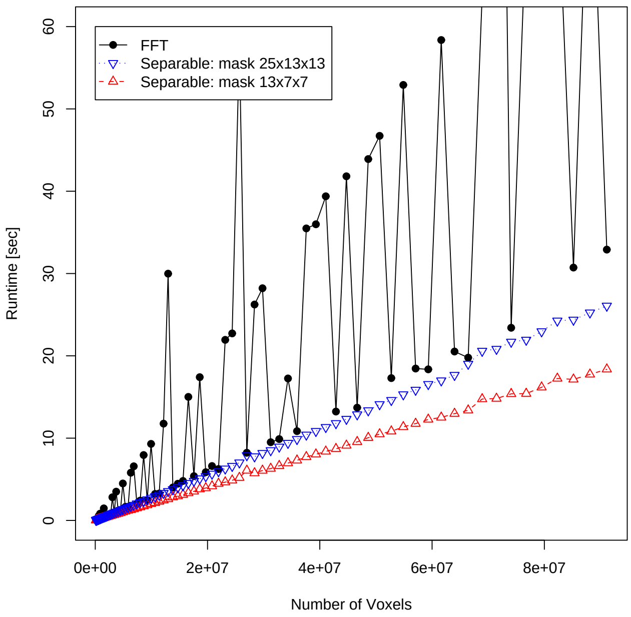

Fig. 2: Runtimes on an Intel Xeon 2.8 GHz processor: FFT and our separable filter with two mask sizes for image sizes 50^3, 55^3, $\ldots$, 450^3 voxels. Orientation: $\varphi_y = \frac{\pi}{3}$, $\varphi_z = \frac{2\pi}{3}$.

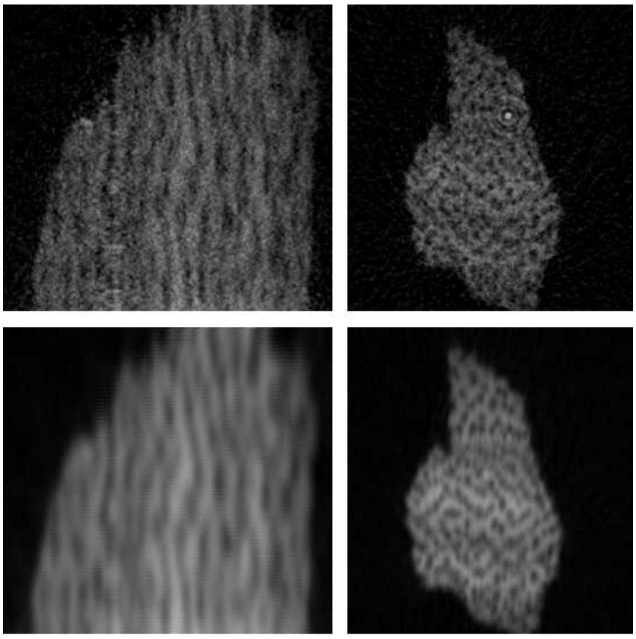

Fig. 3: Orthogonal cross sections through a $286 \times 251 \times 266$ voxel wood dataset. Upper row: original dataset, lower row: output of our method. Computation took 4.5 seconds using our separable implementation and 19 seconds using FFT.

the Fourier domain, only, because the Fourier transform of the kernel can be pre-computed.

As we mentioned in the previous section, our implementation is not fully optimized yet. Nevertheless, Fig. 2 shows that the separable convolution scheme has the potential to outperform the FFT implementation when filter size is small compared to the number of voxels.

For the example application in Fig. 3, we manually selected variance and orientation parameters to demonstrate fiber smoothing. Local orientation estimation should be used in the future to automatically select kernel orientations.

6. CONCLUSIONS AND ACKNOWLEDGMENT

We proposed an approximation to a separable anisotropic Gaussian in three dimensions and our initial experiments showed that this has the potential to outperform runtimes of other convolution schemes for small filter masks.

The first author would like to thank his advisor in Fraunhofer ITWM's Models and Algorithms in Image Processing group, Katja Schladitz, and Claudia Lautensack. We appreciate Christoph Lampert's review of this paper.

7. REFERENCES

[1] J.-M. Geusebroek, A.W.M. Smeulders, and J. van de Weijer, "Fast anisotropic gauss filtering," in *Proc. ECCV*, 2003, pp. 99–112.

[2] J.-M. Geusebroek, A.W.M. Smeulders, and J. van de Weijer, "Fast anisotropic gauss filtering," *IEEE Trans. Image Proc.*, vol. 23, no. 8, pp. 938–943, 2003.

[3] G.Z. Yang, P. Burger, D.N. Firmin, and S.R. Underwood, "Structure adaptive anisotropic image filtering," *Image and Vision Computing*, vol. 14, pp. 135–145, 1996.

[4] V. Lakshmanan, "A separable filter for directional smoothing," *IEEE Geosc. and Remote Sensing Letters*, vol. 1, no. 3, pp. 192–195, 2004.

[5] S. Treitel and J.L. Shanks, "The design of multistage separable planar filters," *IEEE Trans. Geosc. Electron.*, vol. GE-9, no. 1, 1971.

[6] E.W. Weisstein, "Fourier transform–gaussian," MathWorld–A Wolfram Web Resource.

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

[8] M. Frigo and S. G. Johnson, "Fftw: An adaptive software architecture for the fft," in *Proc. ICASSP*, 1998, vol. 3, pp. 1381–1384.