Multiresolution Moment Filters: Theory and Applications

Michael Sühling*, Muthuvel Arigovindan, Patrick Hunziker, and Michael Unser, *Fellow, IEEE*

M. Sühling, M. Arigovindan, and M. Unser are with the Biomedical Imaging Group, Swiss Federal Institute of Technology Lausanne (EPFL), CH-1015 Lausanne, Switzerland. E-mail: michael.suehling@epfl.ch, muthuvel.arigovindan@epfl.ch, michael.unser@epfl.ch.

P. Hunziker is with the University Hospital Basel, Petersgraben 5, CH-4031 Basel, Switzerland. E-mail: PHunziker@uhbs.ch.

This work was supported in part by the Swiss National Science Foundation under grant 3200-059517.99 and the Swiss Heart Foundation.

Abstract

We introduce local weighted geometric moments that are computed from an image within a sliding window at multiple scales. When the window function satisfies a two-scale relation, we prove that lower order moments can be computed efficiently at dyadic scales by using a multiresolution wavelet-like algorithm. We show that B-splines are well suited window functions because, in addition to being refinable, they are positive, symmetric, separable, and very nearly isotropic (Gaussian shape).

We present three applications of these multi-scale local moments. The first is a feature extraction method for detecting and characterizing elongated structures in images. The second is a noise reduction method which can be viewed as a multi-scale extension of Savitzky-Golay filtering. The third is a multi-scale optical flow algorithm that uses a local affine model for the motion field, extending the Lucas-Kanade optical flow method. The results obtained in all cases are promising.

Index Terms

Local Moments, Multiresolution, Weighted Least-Squares, Savitzky-Golay, Optical Flow.

I. Introduction

Global geometric moments and their invariants are widely used in many areas of image analysis, including pattern recognition [1], image reconstruction [2], and shape identification [3]. In addition to geometric moments, which are also known as regular or ordinary moments, a number of other moments has been proposed. The notion of complex moments was introduced in [4] for deriving moment invariants. Teague [5] suggested the use of orthogonal moments and introduced complex valued Zernike moments that are defined on a unit disk. A second class of orthogonal moments is given by Legendre moments which make use of Legendre polynomials. The usefulness of Legendre and Zernike moments has been demonstrated, in particular, for image reconstruction [2], [6] and pattern classification [7]. The pseudo-Zernike formulation proposed in [8] further improved these characteristics. A detailed discussion of moment-based image analysis can be found in the monograph [9].

Some authors have applied geometric moments in a local fashion for image and texture segmentation [10], [11] and direction-based interpolation [12]. The idea there was to compute moments locally over some square region of interest which is moved over the image; the window functions may be overlapping or not depending on the application. An efficient method to compute local moments inside sliding squared windows with constant weights has recently been proposed in [13].

In this paper, we are extending the notion of local geometric moments by introducing two refinements:

weighting and multiresolution. The idea of weighting is motivated by the observation that the square window that has been used so far is rather anisotropic. Indeed, if the goal is to design a ”rotation-invariant” algorithm, it makes good sense to apply an isotropic window with a radial weighting that decreases away from the center. Multiresolution is a feature that is highly desirable for designing image processing algorithms that have some degree of adaptability. The down-side, of course, is that these multi-scale refinements can be computationally very expensive, especially when the size of the window is large. The framework of wavelets [14] is a computational efficient approach to multiresolution and has proven to be successful in many applications such as image denoising [15], [16], feature enhancement [17] and shape analysis [18]. In this paper, we use wavelet-related concepts and propose a fast multiresolution wavelet-like algorithm to compute multi-scale local geometric moments of different orders with a dyadic scale progression. In particular, we will consider B-spline window functions, which become wider and more and more Gaussian-like—also meaning isotropic—as the degree of the spline increases.

We believe that these multi-scale local geometric moments could be useful tools for devising new algorithms based on what we call a ”sliding window” formulation of a problem. The basic assumption for such an approach is that the spatially-varying feature (or parameter) that one is estimating is approximately constant within the window. The unknown parameter is then estimated from the available information in the window (which often requires the evaluation of moments). Finally, the output value is attributed to the spatial location corresponding to the center of the window. This is a simple, yet powerful paradigm that can be made most effective by working at the appropriate scale (multiresolution strategy). We will illustrate these ideas in Section III by presenting three such local-moment-based algorithms:

- a new method for local shape analysis and feature extraction,
- a multi-scale noise reduction method based on Savitzky-Golay filters [19],
- a multiresolution extension of the Lucas-Kanade optical flow algorithm [20], which uses a more refined local-affine model for the motion.

These methods are fast thanks to the wavelet-like implementation. The experimental results obtained in all cases are encouraging.

II. THEORY

In this section, we will define weighted local geometric moments and their associated multiresolution moment filters. We also show how these moments can be computed efficiently in a multiresolution framework.

A. Weighted Local Geometric Moments

Global geometric moments of order $p \in \mathbb{N}_0$ and location $x_0 \in \mathbb{R}$ of a continuously-defined function f are defined as [1]

$$M_p(x_0) = \int_{\mathbb{R}} (x - x_0)^p \, f(x) \, dx. \tag{1}$$

For localization, we introduce a positive and symmetric window function w with compact support Ω. We then define weighted local geometric moments of order p, scale $j \in \mathbb{Z}$ and location x_0 as

$$m_p^{(j)}(x_0) = \int_{\mathbb{R}} (x - x_0)^p \, w\left(\frac{x - x_0}{2^j}\right) f(x) \, dx. \tag{2}$$

Note that the window function is dilated by a factor 2^j and is centered at x_0. For a given window function w, we call

$$w_p(x) = x^p \, w(x) \tag{3}$$

the moment filter mask of order p. Then the local weighted geometric moments can be rewritten in the form of a convolution as

$$\begin{aligned} m_p^{(j)}(x_0) &= 2^{jp} \int_{\mathbb{R}} w_p\left(\frac{x - x_0}{2^j}\right) f(x) \, dx & (4) \\ &= 2^{jp} \left(w_p^{(j)\,T} * f \right)(x_0), & (5) \end{aligned}$$

where the multiresolution moment filters $w_p^{(j)\,T}(x) = w_p(-x/2^j)$ are time reversed and dilated versions of the basic moment filter mask (3). The normalization factor 2^{jp} in (5) is included to simplify the formulation of the multiresolution algorithm presented next.

B. Two-Scale Equation

Computing local moments at coarser scales becomes more and more time consuming due to the increasing size of the window function. However, multiresolution pyramids of local moments can be computed efficiently, provided that the window function satisfies a two-scale equation, a concept that is closely related to the framework of wavelets [14].

Theorem 1 (Two-Scale Equation): Let w be a function that satisfies the two-scale equation

$$w\left(\frac{x}{2}\right) = \sum_l h(l) w(x - l), \tag{6}$$

for some given filter h. Then, w_p satisfies the multi-channel two-scale equation

$$2^p w_p\left(\frac{x}{2}\right) = \sum_{k=0}^{p} \left(h_{p,k} * w_k\right)(x) = \sum_{k=0}^{p} \sum_{l} h_{p,k}(l) w_k(x-l), \quad (7)$$

with filters $h_{p,k}$, $k = 0, \ldots, p$, given by

$$h_{p,k}(l) = \binom{p}{k} l^{p-k} h(l). \quad (8)$$

In particular, we have that $h_{0,0} = h$. The proof of this theorem is given in Appendix I.

C. Efficient Multi-Scale Implementation

Theorem 1 can be used to derive fast algorithms for computing local moments $m_p^{(j)}$ for scales $j = j_0, \ldots, j_1$ and orders $p = 0, \ldots, P$. To initialize the procedure, the inner products on the finest scale j_0 are computed by using (5). Due to Theorem 1, the coefficients on the subsequent coarser scales can be determined recursively.

Corollary 1: Let $m_p^{(j)}(n)$, $0 \leq p \leq P$, be local moments at scale j and positions $n \in \mathbb{Z}$. Then the moments at the next coarser scale $(j+1)$ can be computed by

$$m_p^{(j+1)}(n) = \sum_{k=0}^{p} \sum_{l} h_{p,k}^{(j)}(l)\, m_k^{(j)}(n + 2^j l) \quad (9)$$

with filter masks $h_{p,k}^{(j)}$ given by

$$h_{p,k}^{(j)} = 2^{j(p-k)} h_{p,k}. \quad (10)$$

For a proof of this corollary, we refer to Appendix II. Equation (10) means that the two-scale filters $h_{p,k}$ have to be multiplied by $2^{(p-k)}$ at each scale j prior to convolution. The filters $h_{p,k}^{(j)}$ need not to be stored separately since they are obtained by simply updating the basic filters $h_{p,k}$ at each scale. Equation (9) is a multi-channel extension of the "à trous" algorithm, which is frequently used for computing overcomplete wavelet transforms [21].

The method is easily modified for computing local moments in a sub-sampled, wavelet-like pyramid. The recursion equation (9) then simplifies to a Mallat-like algorithm (cf. [21]):

$$m_p^{(j+1)}(n) = \sum_{k=0}^{p} \left(h_{p,k}^{(j)\,T} * m_k^{(j)}\right)(2n), \quad (11)$$

where $h_{p,k}^{(j)\,T}(l) = h_{p,k}^{(j)}(-l)$ denotes the time reversed filter mask $h_{p,k}^{(j)}$. The corresponding block diagram for computing moments of order 0 to 2 in a sub-sampled fashion is shown in Fig. 1.

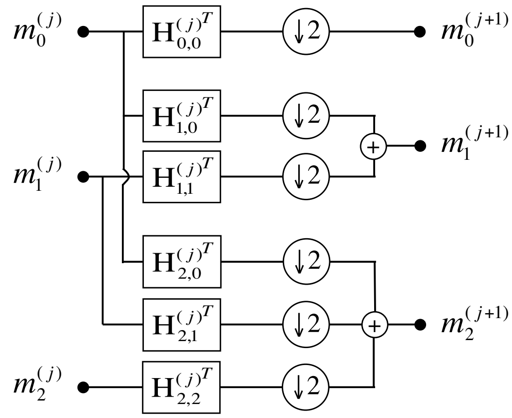


Fig. 1. Recursive computation of moments of order 0 to 2 using multi-channel two-scale filters.

To avoid boundary artifacts, the signals that are considered by the algorithm need to be extended properly at the boundary. We assume that the input signal f is extended by using a mirror boundary convention. If this signal is filtered with a symmetric filter (e.g., even moments), the output will exhibit the same symmetry. Conversely, if the signal is filtered with an anti-symmetric filter (e.g., odd moments), the output will be anti-symmetric at the boundary. Therefore, in order to implement the recursive two-scale algorithms (9) and (11), one has to alternate between the right type of boundary extension of the moments to produce an output that is consistent with the input assumptions. This is ensured by extending even and odd order moments by mirror and anti-mirror boundary conditions, respectively. From (8) it can be seen that the two-scale filters $h_{p,k}$ are symmetric or anti-symmetric, if $p - k$ is even or odd, respectively. Thus, the convolution with the properly extended moments $m_k^{(j)}$ will result in the correct boundary extension of the moments $m_p^{(j+1)}$. A summary of all possible cases is given in Table I.

The usage of the two-scale algorithm clearly pays off when computing lower order moments at coarser scales. The direct computation of the moments by (5) requires $O(2^j)$ multiplications and $O(2^{j+1})$ additions per output point at scale j. On the other hand, the computational complexity of the recursive two-scale algorithm is independent of the scale j and behaves like $O(1)$. A detailed analysis of the computational cost is given in the Appendix III-A.

D. Multiple Dimensions

The notion of multi-scale weighted moments can be extended to multiple dimensions in a straightforward way by using tensor products. In the two-dimensional (2D) case, we define moment filter masks

TABLE I

BOUNDARY EXTENSION OF MOMENTS.

Order p of $m_p^{(j+1)}$	Boundary extension of $m_p^{(j+1)}$	Order k of $m_k^{(j)}$	Boundary extension of $m_k^{(j)}$	$p-k$	Filter symmetry of $h_{p,k}$
even	mirror	even	mirror	even	symmetric
even	mirror	odd	anti-mirror	odd	anti-symmetric
odd	anti-mirror	even	mirror	odd	anti-symmetric
odd	anti-mirror	odd	anti-mirror	even	symmetric

of order $(p+q)$ as

$$w_{p,q}(x,y) = w_p(x)w_q(y). \tag{12}$$

The moments at scale j are then given by the separable convolution

$$m_{p,q}^{(j)}(x_0,y_0) = 2^{j(p+q)}\left(w_{p,q}^{(j)\,T} * f\right)(x_0,y_0), \tag{13}$$

where $w_{p,q}^{(j)\,T}(x,y) = w_p(-x/2^j)w_q(-y/2^j)$ are the associated 2D multiresolution moment filters. For an efficient computation of $m_{p,q}^{(j)}$, equations (9) and (11) are applied successively in each dimension. In the sub-sampled discrete case, this reads

$$m_{p,q}^{(j+1)}(n,m) = \sum_{k=0}^{p}\sum_{l=0}^{q}\left(h_{p,k}^{(j)\,T} h_{q,l}^{(j)\,T} * m_{k,l}^{(j)}\right)(2n,2m), \tag{14}$$

where the two-scale filters $h_{p,k}^{(j)\,T}$ and $h_{q,l}^{(j)\,T}$ are applied separately in x- and y-directions, respectively. For instance, the block diagram for the second order moment $m_{1,1}^{(j+1)}$ is illustrated in Fig. 2.

In the two-dimensional case, the direct moment computation (13) requires $O(2^{j+1})$ multiplications and $O(2^{j+2})$ additions per output point at scale j, whereas the cost of the recursive two-scale algorithm behaves as $O(1)$. For a detailed analysis of the computational complexity in two dimensions, we refer to Appendix III-B.

E. B-spline Window Function

The ideal window function should be positive, with weights decreasing away from the center, refinable, separable, and isotropic in multiple dimensions. The only choice would be a Gaussian, but it does not

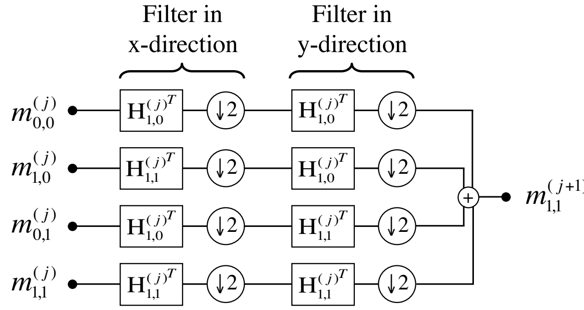


Fig. 2. Recursive computation of $m_{1,1}^{(j+1)}$ using multi-channel two-scale filters.

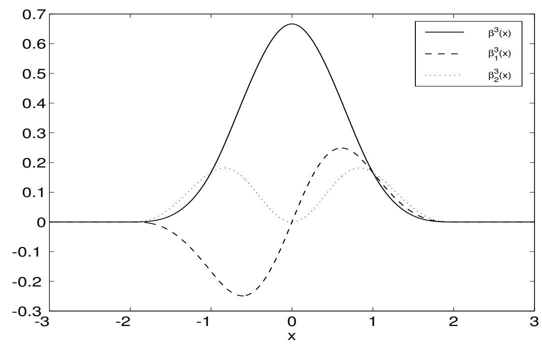


Fig. 3. Cubic B-spline β^3 and its first two moment filters β_1^3 and β_2^3.

satisfy a two-scale equation. However, B-splines β^n satisfy a two-scale equation and rapidly converge to Gaussians when their degree $n \in \mathbb{N}$ increases [22]. In fact, for a given number of filter tabs, B-splines are the smoothest scaling functions in the Sobolev sense [23]; this guaranties that they converge fastest to Gaussians in the Sobolev norm. This ensures nearly isotropy of the window in multiple dimensions. The cubic B-spline ($n = 3$), β^3, and its two first moment filters β_1^3 and β_2^3 are plotted in Fig. 3. The corresponding two-scale filters $h_{p,k}$ up to order $p = 2$ are given in Table II.

The Fourier transform of a B-spline β^n, which is the (n+1)-fold convolution of a rectangular pulse, is given by

$$\widehat{\beta}^n(\omega) = \left(\frac{\sin(\omega/2)}{\omega/2}\right)^{n+1}. \tag{15}$$

By definition, the Fourier transforms of the corresponding moment filters are given by

$$\widehat{\beta}_p^n(\omega) = i^p \frac{d^p}{d\omega^p}\widehat{\beta}^n(\omega). \tag{16}$$

B-splines of degree n are by construction in C^{n-1}, i.e., they are (n-1) times continuously differentiable; the same also holds true for the moment filters. This implies that their Fourier transforms decay at least

TABLE II

TWO-SCALE FILTERS $h_{p,k}$ UP TO ORDER $p = 2$ FOR β^3.

l	-2	-1	0	1	2
$h_{0,0}(l)$	$\frac{1}{8}$	$\frac{4}{8}$	$\frac{6}{8}$	$\frac{4}{8}$	$\frac{1}{8}$
$h_{1,0}(l)$	$-\frac{1}{4}$	$-\frac{2}{4}$	0	$\frac{2}{4}$	$\frac{1}{4}$
$h_{1,1}(l)$	$\frac{1}{8}$	$\frac{4}{8}$	$\frac{6}{8}$	$\frac{4}{8}$	$\frac{1}{8}$
$h_{2,0}(l)$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$
$h_{2,1}(l)$	$-\frac{1}{2}$	$-\frac{2}{2}$	0	$\frac{2}{2}$	$\frac{1}{2}$
$h_{2,2}(l)$	$\frac{1}{8}$	$\frac{4}{8}$	$\frac{6}{8}$	$\frac{4}{8}$	$\frac{1}{8}$

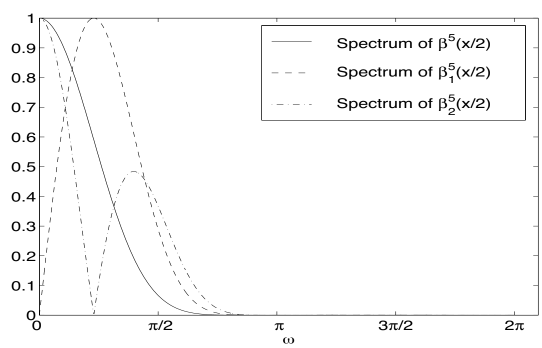


Fig. 4. Normalized spectra of moment filters $\beta^5(x/2)$, $\beta^5_1(x/2)$ and $\beta^5_2(x/2)$.

like $O(1/|\omega|^n)$ for large ω. Consequently, the Fourier transforms of the moment filters decay faster when the spline degree increases. Fig. 4 shows the normalized spectra of the B-spline $\beta^5(x/2)$ and its moment filters $\beta^5_p(x/2)$ for degree $n = 5$ and moment orders $p = 0, 1, 2$ at scale $j = 1$. It is clear from this graph that the filters are essentially bandpass, which can be used as a justification for the downsampling of moments at coarser scales.

III. APPLICATIONS

The fast algorithm presented above is applicable to a variety of image analysis problems, such as image segmentation, pattern detection, and optical flow estimation, for which local solutions over sliding windows have been proposed. These approaches can be extended by applying a multiresolution strategy which provides adaptability while also reducing computational cost. Here, we will illustrate the concept by presenting new local-moment-based algorithms for three specific tasks: (1) local shape analysis and feature extraction, (2) filtering for noise reduction, and (3) the estimation of motion fields using a local affine model.

A. Local Shape Analysis and Feature Extraction

Effective analysis of shapes is required by many computer vision applications, in particular in biomedical image analysis. One of the major issues is to determine location, orientation and size features of filamentous or spherical bright structures in an image. Examples are segmentation and characterization of biological cell images, the analysis of vessel distributions in medical images and the detection of DNA filaments in electron micrograph images. The evaluation of low order moments represents a systematic and efficient method of shape analysis. Since moments are integral-based features, they are robust against noise. Furthermore, low order moments have a direct geometrical interpretation.

1) Geometric Interpretation of Moments: The moments $m_{p,q}$ have well-defined geometric interpretations. The coordinates of the local centroid are given by

$$\bar{x} = m_{1,0}/m_{0,0} \quad \text{and} \quad \bar{y} = m_{0,1}/m_{0,0}. \tag{17}$$

The distance between the window center and the local centroid allows to detect whether the sliding window is located on the center of a bright structure or not. The so-called central moments [1] can be expressed in terms of ordinary moments $m_{p,q}$ and the coordinates of the centroid. For the second order we have

$$\mu_{2,0} = m_{2,0} - m_{0,0}\bar{x}^2, \quad \mu_{0.2} = m_{0,2} - m_{0,0}\bar{y}^2, \tag{18}$$

$$\mu_{1,1} = m_{1,1} - m_{0,0}\bar{x}\bar{y}. \tag{19}$$

These three central moments of second order are the components of the inertia matrix

$$\mathbf{J} = \begin{pmatrix} \mu_{2,0} & \mu_{1,1} \\ \mu_{1,1} & \mu_{0,2} \end{pmatrix}. \tag{20}$$

The local orientation of the analyzed object is given by the eigenvector corresponding to the minimal eigenvalue of $\mathbf{J}$. In fact, the local object is mapped onto an ellipsoid centered at $(\bar{x}, \bar{y})$. The ellipsoid axes are directed along the eigenvectors of $\mathbf{J}$ and the corresponding axes semi-lengths are the magnitudes of the respective eigenvalues λ_1 and λ_2. The orientation angle with respect to the x-axis is given by

$$\phi = \frac{1}{2}\arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right). \tag{21}$$

A measure for the eccentricity of the local ellipsoid is given by

$$\varepsilon = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2 = \frac{\left(\mu_{2,0} - \mu_{0,2}\right)^2 + 4\mu_{1,1}^2}{\left(\mu_{2,0} + \mu_{0,2}\right)^2} \tag{22}$$

and takes values between 0 and 1. It indicates whether the local object is elongated or not. The eccentricity measure is independent from the local image energy and is therefore well-suited for inter-scale comparisons.

2) Multi-Scale Detection Strategy: Brighter elongated structures or filaments can be extracted by evaluating the various moment features and putting thresholds on eigenvalues or eccentricity measures. Since the elongated structures of interest can have different sizes, we propose to detect them at multiple scales $j_0 \leq j \leq j_1$, where j_0 and j_1 are the finest and coarsest scale at which relevant structures are expected. A simple strategy, which was applied in our experiments, is described in the following: At each image pixel (n_x, n_y) we compute the local moments $m_{p,q}^{(j)}$ for $(p+q) \leq 2$. From these we derive the local orientations and eccentricities $\varepsilon^{(j)}$. To decide whether or not a local object is part of a filamentous structure, we compute the figure of merit

$$\gamma^{(j)} = \varepsilon^{(j)} e^{-(\bar{x}^2+\bar{y}^2)/(2^{(2j+1)}\sigma^2)}. \tag{23}$$

The second factor in (23) assigns more weight to cases where the local centroid $(\bar{x}, \bar{y})$ is close to the center of the local window. The parameter σ controls the range of the centroid around the window origin to be accepted. The multi-scale approach also helps us to detect cases where the local structure is located symmetrically at the periphery of the window function. To avoid these cases, the figure of merit $\gamma^{(j)}$ is set to zero, if $m_{0,0}^{(j-1)} < m_{0,0}^{(j)}$. This means that the local mean of the gray values at the next finer scale has to be greater than the local mean at the current scale.

The figure of merit (23) will be maximal at a scale that approximately matches the size of the elongated shape to detect. Therefore, we integrate the figures of merit at different scales to obtain a final estimate for the goodness of local fit by

$$\gamma = \max_{j_0 \leq j \leq j_1} \gamma^{(j)}. \tag{24}$$

3) Application: Detection of DNA Filaments: The structure of DNA molecules can be visualized by cryo-electron-microscopy (CEM) [24]. Because of the physical process involved, the resulting images have very low contrast to avoid destruction of the specimen (cf. Fig. 6). Biologists are highly interested in an automatic detection of the thin strands of DNA, but the task is challenging because of the poor signal-to-noise ratio (near 0 dB).

The proposed moment-based algorithm was tested on synthetic and real images. Fig. 5(a) and 5(b) show a synthetic circular DNA strand with two different levels of additive Gaussian noise, respectively. In this experiment, we used a B-spline window of degree 3 at scales $j = 2, 3$.

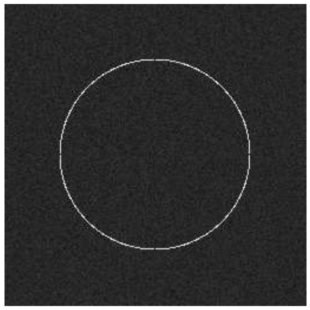
The algorithm is compared with the so-called structure tensor method [25], a standard method to estimate local orientations of image patterns. Instead of the inertia tensor (20), this method uses the structure tensor

$$\mathbf{S} = \begin{pmatrix} \langle w, I_x^2 \rangle & \langle w, I_x I_y \rangle \\ \langle w, I_x I_y \rangle & \langle w, I_y^2 \rangle \end{pmatrix}, \tag{25}$$

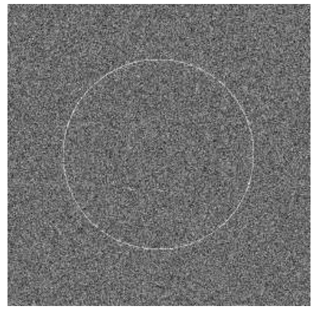
where $w(x, y)$ denotes a window function and I_x, I_y denote the partial derivatives of the image intensity $I(x, y)$. The computation of the local orientation and eccentricity measure is analog to (21) and (22), respectively. As in [25], we interpret the estimated eccentricity as a figure of merit and use a Gaussian as a window function. For the standard deviation of the Gaussian window, we used $\sigma = 1.7$ which corresponds to the effective width of the B-spline window at the finest scale $j = 2$ of the moment-based algorithm.

The estimated eccentricities of the structure tensor approach are shown in Fig. 5(c) and 5(d) for the two different noise levels, respectively. Fig. 5(e) and 5(f) show the corresponding figures of merit of the proposed moment-based algorithm. For the lower noise level both methods detect the circular structure well. However, the figure of merit of the moment-based algorithm is much thinner around the true structure since the eccentricity measure is weighted by the distance of the window center to the centroid of the local image content as described in (23). This feature is not available in the structure tensor approach. In the case of the higher noise level, the moment based algorithm still detects the elongated object fairly well (Fig. 5(f)). In contrast, the structure tensor approach degrades significantly (Fig. 5(d)). This is probably due to the fact that this method uses derivatives which are sensitive to noise, whereas the proposed approach is integral-based.

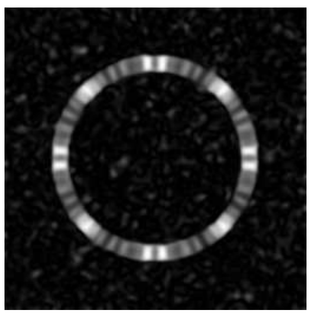
The moment-based detection algorithm was also applied to real images as shown in Fig. 6(a). Since the intensity in CEM-images may vary globally, the original images were first normalized in a pre-processing step. We used moments of order zero (local average) at scale $j = 2$ for local background subtraction. Then we computed for each pixel the figure of merit γ as described above. In particular, we used a B-spline window of degree 3 at scales $j = 2, 3$. The figures of merit were then thresholded to suppress values that correspond to non-significant structures. The final figures of merit are visualized in Fig. 6(b) in form of a needle diagram. The length of the needles is proportional to the size of the figure of merit at each pixel. The direction of the needles corresponds to the local orientation of the object. We see that



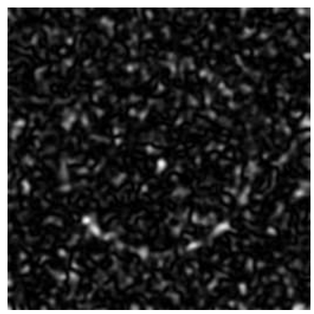
(a) Synthetic elongated structure with additive Gaussian noise (SNR = 28.14 dB)



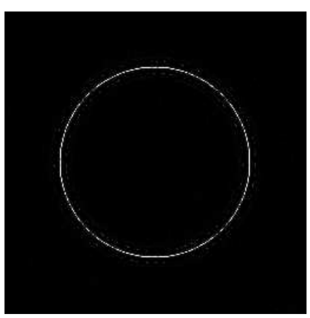
(b) Synthetic elongated structure with additive Gaussian noise (SNR = 8.15 dB)



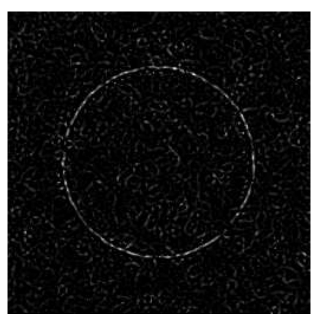
(c) Figure of merit of structure tensor-based algorithm applied to (a)



(d) Figure of merit of structure tensor-based algorithm applied to (b)

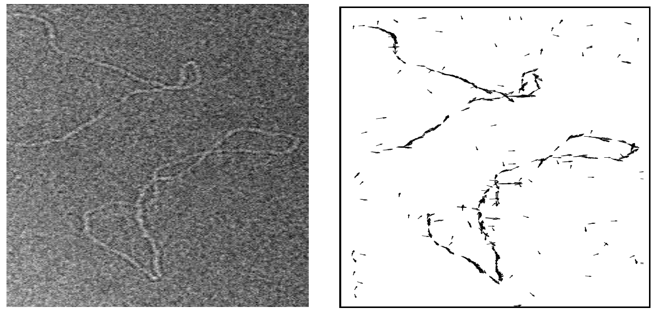


(e) Figure of merit of moment-based algorithm applied to (a)



(f) Figure of merit of moment-based algorithm applied to (b)

Fig. 5. Comparison of moment-based and inertia tensor-based detection of elongated structures for different noise levels.



(a) Original CEM-image (b) Estimated local orientation

Fig. 6. A CEM-image and detected DNA strands.

the two DNA strands contained in the image together with their local orientation were clearly detected. Failures due to the high noise content in the image are very sparse.

B. Multi-Scale, Weighted Savitzky-Golay Smoothing Filters

Savitzky-Golay filtering [19] can be thought of as a generalized moving average filter. The idea of Savitzky-Golay filtering is to find filter coefficients that preserve higher order polynomials. These filter coefficients are derived by a least-squares fitting of a polynomial of given degree within a sliding window. The smoothed points are computed by replacing each data point with the value of the fitted polynomial at the window center. For this reason, a Savitzky-Golay filter is also called a digital smoothing polynomial filter or a least-squares smoothing filter. A crucial point is the choice of the size of the window function. A small window preserves narrow features of the underlying signal, but filters less; larger windows smooth more, but lead to blurring of image details.

Originally, this approach was proposed for one-dimensional signals and used a box-shaped window function of fixed length [19]. Here, we propose a multi-dimensional extension based on a weighted least squares criterion. We also propose a new multi-scale filtering strategy whereby the final smoothed image is obtained by combining results from different scales using a hypothesis test.

1) Weighted Savitzky-Golay Filtering: Let us consider a two-dimensional polynomial of degree d

$$P_d(x,y) = \sum_{0\leq p+q\leq d} a_{p,q}\, x^p y^q, \tag{26}$$

which is specified by the $N_a = (d+1)(d+2)/2$ polynomial coefficients $a_{p,q}$. Let w denote a window function with discrete support Ω of cardinality N_Ω which is located at (m_0, n_0). To fit the polynomial locally to an image $I(m,n)$, we minimize the weighted least-squares functional

$$r^2 = \sum_{(m,n)\in\Omega} w(m,n)\Big(P_d(m,n) - I(m+m_0, n+n_0)\Big)^2. \tag{27}$$

By differentiating (27) with respect to each of the unknown polynomial coefficients $a_{p,q}$, we obtain the corresponding normal equations $\mathbf{A}^T\mathbf{W}\mathbf{A}\,\mathbf{a} = \mathbf{A}^T\mathbf{W}\,\mathbf{b}$, where

$$\left\{\mathbf{A}^T\mathbf{W}\mathbf{A}\right\}_{p_1,q_1;p_2,q_2} = \sum_{(m,n)\in\Omega} m^{p_1+p_2} n^{q_1+q_2}\, w(m,n), \tag{28}$$

$$\left\{\mathbf{A}^T\mathbf{W}\,\mathbf{b}\right\}_{p_1,q_1} = \sum_{(m,n)\in\Omega} m^{p_1} n^{q_1}\, w(m,n)\cdot I(m+m_0, n+n_0), \tag{29}$$

and

$$\{\mathbf{a}\}_{p_2,q_2} = a_{p_2,q_2}, \tag{30}$$

with $0 \le p_1 + q_1 \le d$ and $0 \le p_2 + q_2 \le d$. The index-tuples (p_1, q_1) and (p_2, q_2) denote the row and column indices of the matrix and vectors, respectively. The $N_\Omega \times N_\Omega$ diagonal matrix $\mathbf{W}$ is composed by the weights $w(m,n)$. Since $\mathbf{A}^T\mathbf{W}\mathbf{A}$ does not depend on the image data, the matrix and its inverse can be computed once and forever in advance. The right hand side vector $\mathbf{A}^T\mathbf{W}\,\mathbf{b}$ is nothing but a discrete version of the local moments (13) of order zero to d. The smoothed image point at the window center is equal to the polynomial coefficient $a_{0,0}$, which is given by the inner product of the corresponding row $\left\{(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\right\}_{0,0}$ of the matrix inverse and the right hand side $\mathbf{A}^T\mathbf{W}\,\mathbf{b}$.

2) Multi-Scale Strategy: In order to find a trade-off between the conflicting requirements of noise reduction and conservation of image details, we propose a multi-scale framework of the introduced weighted Savitzky-Golay filtering. We assume that the image is locally given by the model: polynomial signal + noise, i.e.,

$$I = P_d + \varepsilon, \tag{31}$$

where $\varepsilon \sim N(0, \sigma^2)$ corresponds to Gaussian white noise of zero mean and common variance σ^2. Thus, the residual (27) gives the expected squared deviation of the image data from the given polynomial P_d due to noise. As shown in the appendix, the normalized residual corresponds to a linear combination of $N_\Omega - N_a$ independent χ_1^2-distributed random variables, i.e.,

$$\frac{r^2}{\sigma^2} \sim \sum_{n=1}^{N_\Omega - N_a} \lambda(n)\, \chi_1^2, \tag{32}$$

where the coefficients $\lambda(n)$ are given by the $N_\Omega - N_a$ non-zero eigenvalues of the matrix

$$\mathbf{W} - \mathbf{WA}\left(\mathbf{A}^T\mathbf{WA}\right)^{-1}\mathbf{A}^T\mathbf{W}. \tag{33}$$

For a proof and the computation of the probability density function of (32) we refer to Appendix IV. Note that for uniform weights the resulting distribution (32) corresponds to a $\chi^2_{N_\Omega - N_a}$ distribution with $N_\Omega - N_a$ degrees of freedom. Asymptotically, r^2/σ^2 is normally distributed with mean

$$E\left(\frac{r^2}{\sigma^2}\right) = \sum_{n=1}^{N_\Omega - N_a} \lambda(n) \tag{34}$$

and variance

$$\mathrm{var}\left(\frac{r^2}{\sigma^2}\right) = 2\sum_{n=1}^{N_\Omega - N_a} \lambda(n)^2. \tag{35}$$

When working on real images, (32) enables us to detect image regions for which the chosen polynomial degree or window size are not adequate. More specifically, we apply a two-sided hypothesis test on r^2/σ^2 with a given significance level α. In order to avoid cases where the degree of the polynomial is too high for the given image structure and tends to fit the noise, we reject results for which the residual is below the confidence interval. This usually happens when using small windows in flat image regions. On the other hand, we also reject results for which the residual is above the confidence interval. In this case, image details like edges cannot be fitted closely by the polynomial. The aim is to use locally a window as large as possible to achieve maximum noise reduction. Consequently, we compute smoothed image versions using windows at scales $j = j_0, \ldots, j_1$. Recall that the images of moments (29) can be computed efficiently for different scales by using (9). The final smoothed image is obtained by choosing for each pixel the output value from the coarsest scale for which the normalized residual remains inside the confidence interval.

3) Numerical Results: In order to demonstrate the performance of weighting and multi-scale filtering, we have applied the algorithm to an image containing additive Gaussian white noise. Fig. 7(a) shows the original image and Fig. 7(b) shows the image after adding Gaussian white noise of standard deviation $\sigma = 20.0$ resulting in a signal-to-noise ratio of 20.40 dB. Results were computed for a B-spline window of degree 3 and a fitting polynomial of degree 2.

Fig. 8 illustrates the effect of using B-spline weighting. The left column displays the filtered outputs of B-spline-weighted Savitzky-Golay filtering at scales $j = 1, \ldots, 3$. Window sizes were 7, 15 and 31 pixels at each scale, respectively. The right column corresponds to the case of using a squared window

(a) Original image

(b) Noisy image (SNR = 20.40 dB)

Fig. 7. Original and noisy image.

with constant weights one. Here, the support was chosen to be the effective duration of the B-spline windows, resulting in window sizes of 3, 5 and 9 pixels, respectively.

Fig. 9(a) shows the final output image of the multi-scale, B-spline-weighted method. Smoothed image versions at scales $j = 1, \ldots, 3$ were combined to the final output image using a double-sided hypothesis test on the normalized residuals (32) with a significance level $\alpha = 0.01$. The signal-to-noise ratio of the final image is 27.30 dB, which is significantly larger than the signal-to-noise ratios at the single scales (26.25 dB, 24.55 dB, 21.63 dB for scales $j = 1, \ldots, 3$). Also visually, the final output image seems to be superior to the single scale outputs. Image details like edges are well preserved, whereas flat image regions are fairly smoothed.

The result is compared with two standard denoising algorithms. The first is a wavelet soft-thresholding method. The noisy image was decomposed in a 3-level wavelet transform pyramid using orthogonal Battle-Lemarié wavelets [26]. We used the same order of spline ($n = 3$) for the methods to be comparable. We also optimized the method by selecting the threshold $T = 26$, yielding the maximum signal-to-noise ratio (SNR = 24.89 dB). From Fig. 9(c) it can be seen that the wavelet-based smoothed image is clearly more blurred and suffers from typical ringing artifacts. The second comparison method is the adaptive Wiener filter [27]. This filter corresponds to a pixel-wise adaptive Wiener method based on statistics derived from a local neighborhood of each pixel. The maximum signal-to-noise ratio of 25.96 dB was obtained for a filter size of (5×5) pixels. As can be seen from Fig. 9(d), the Wiener filter preserves image details well, but smoothes less in flat image regions.

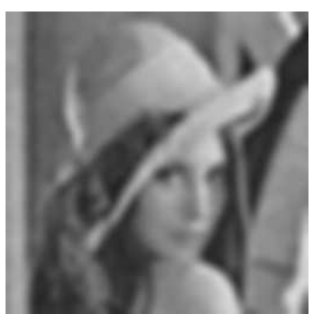
In the present approach, different scales are combined in an exclusive fashion which leads to some

(a) B-spline-weighted at scale $j = 1$ (SNR = 26.25 dB)

(b) Non-weighted at scale $j = 1$ (SNR = 22.67 dB)

(c) B-spline-weighted at scale $j = 2$ (SNR = 24.55 dB)

(d) Non-weighted at scale $j = 2$ (SNR = 25.49 dB)



(e) B-spline-weighted at scale $j = 3$ (SNR = 21.63 dB)

(f) Non-weighted at scale $j = 3$ (SNR = 23.88 dB)

Fig. 8. Images (a) to (f) demonstrate the effect of B-spline weighting at scales $j = 1, \ldots, 3$.

(a) Final multi-scale, B-spline-weighted Savitzky-Golay output (SNR = 27.30 dB)

(b) Final multi-scale, non-weighted Savitzky-Golay output (SNR = 26.42 dB)

(c) Wavelet thresholded image (SNR = 24.89 dB)

(d) Wiener filtered image (SNR = 25.96 dB)

Fig. 9. Comparison of different smoothing methods.

artifacts near edge regions. Although the proposed multi-scale denoising algorithm performs best in terms of signal-to-noise-ratio, it may be possible to improve the visual perception of the output further by using a more progressive weighted combination.

C. Optical Flow Estimation

The estimation of motion from an image sequence is a classical problem in computer vision. Among others, the optical flow technique has been proven to be a successful approach to this problem [28].

Let $I(x, y, t)$ denote the intensity of pixels at location $\mathbf{r} = (x, y)$ and time t in an image sequence. Gradient-based optical flow estimation relies on the assumption that the intensity of a particular point in

$$\mathbf{A}^T\mathbf{W}\mathbf{A} = \begin{pmatrix} \langle w, I_x^2\rangle & \langle w, I_xI_y\rangle & \langle x\,w, I_x^2\rangle & \langle y\,w, I_x^2\rangle & \langle x\,w, I_xI_y\rangle & \langle y\,w, I_xI_y\rangle \\ \langle w, I_xI_y\rangle & \langle w, I_y^2\rangle & \langle x\,w, I_xI_y\rangle & \langle y\,w, I_xI_y\rangle & \langle x\,w, I_y^2\rangle & \langle y\,w, I_y^2\rangle \\ \langle x\,w, I_x^2\rangle & \langle x\,w, I_xI_y\rangle & \langle x^2\,w, I_x^2\rangle & \langle x\,y\,w, I_x^2\rangle & \langle x^2\,w, I_xI_y\rangle & \langle x\,y\,w, I_xI_y\rangle \\ \langle y\,w, I_x^2\rangle & \langle y\,w, I_xI_y\rangle & \langle x\,y\,w, I_x^2\rangle & \langle y^2\,w, I_x^2\rangle & \langle x\,y\,w, I_xI_y\rangle & \langle y^2\,w, I_xI_y\rangle \\ \langle x\,w, I_xI_y\rangle & \langle x\,w, I_y^2\rangle & \langle x^2\,w, I_xI_y\rangle & \langle x\,y\,w, I_xI_y\rangle & \langle x^2\,w, I_y^2\rangle & \langle x\,y\,w, I_y^2\rangle \\ \langle y\,w, I_xI_y\rangle & \langle y\,w, I_y^2\rangle & \langle x\,y\,w, I_xI_y\rangle & \langle y^2\,w, I_xI_y\rangle & \langle x\,y\,w, I_y^2\rangle & \langle y^2\,w, I_y^2\rangle \end{pmatrix}$$

$$\mathbf{v} = \begin{pmatrix} u_0 \\ v_0 \\ u_x \\ u_y \\ v_x \\ v_y \end{pmatrix}, \qquad \mathbf{A}^T\mathbf{W}\,\mathbf{b} = -\begin{pmatrix} \langle w, I_xI_t\rangle \\ \langle w, I_yI_t\rangle \\ \langle x\,w, I_xI_t\rangle \\ \langle y\,w, I_xI_t\rangle \\ \langle x\,w, I_yI_t\rangle \\ \langle y\,w, I_yI_t\rangle \end{pmatrix} \tag{37}$$

---

a moving pattern does not change with time. The constant intensity assumption can be expressed as [29]

$$I_x(\mathbf{r},t)\,u(\mathbf{r},t) + I_y(\mathbf{r},t)\,v(\mathbf{r},t) + I_t(\mathbf{r},t) = 0. \tag{36}$$

I_x, I_y and I_t denote the spatial and temporal derivatives of the image intensity. The velocities u and v are, respectively, the x- and y-components of the optical flow we wish to estimate.

1) Local Affine Motion: A very popular optical flow algorithm is the Lucas-Kanade method [20], which estimates the motion locally, assuming the motion to be constant within a window of support Ω. In order to account for more complex motions, such as rotation, divergence, and shear, we extend this approach to a local affine model for the motion. If (x_0, y_0) denotes the center of the local window, this model is defined as

$$\begin{aligned} u(x,y) &= u_0 + u_x(x-x_0) + u_y(y-y_0), \\ v(x,y) &= v_0 + v_x(x-x_0) + v_y(y-y_0). \end{aligned} \tag{38}$$

The parameters u_0 and v_0 correspond to the motion at the window center and u_x, u_y, v_x, and v_y are, respectively, the first order spatial derivatives of u and v. The local motion components can be estimated by minimizing the weighted least-squares criterion

$$\int_{\mathbb{R}} w(x-x_0, y-y_0)\Big(I_x\,u + I_y\,v + I_t\Big)^2\,dx\,dy. \tag{39}$$

The symmetric window function w gives more weight to constraints at the center of the local region than to those at the periphery. By differentiating (39) with respect to each of the six unknown parameters, we obtain the so-called normal equations $\mathbf{A}^T\mathbf{W}\mathbf{A}\mathbf{v} = \mathbf{A}^T\mathbf{W}\,\mathbf{b}$ in terms of local moments of orders zero to two of the spatial and temporal derivatives of I as defined in (37).

2) Coarse-To-Fine Multi-Scale Strategy: It is obviously difficult to estimate large motions at fine scales. A way around this problem is to apply a coarse-to-fine strategy. At each scale $j_0 \leq j \leq j_1$ we compute the local moments on a grid which is sub-sampled by 2^j in each dimension. These sub-sampled, multi-scale local moments can be computed efficiently by using (11).

The motion vectors are cascaded through each resolution level as initial estimates and are then replaced if they do not already exceed a scale-dependent size. For each local estimate, we compute the confidence measure

$$1 - \sin\theta = 1 - \frac{\left\|\mathbf{W}^{1/2}\left(\mathbf{A}\mathbf{v} - \mathbf{b}\right)\right\|_{l_2}}{\left\|\mathbf{W}^{1/2}\,\mathbf{b}\right\|_{l_2}} \in [0, 1]. \tag{40}$$

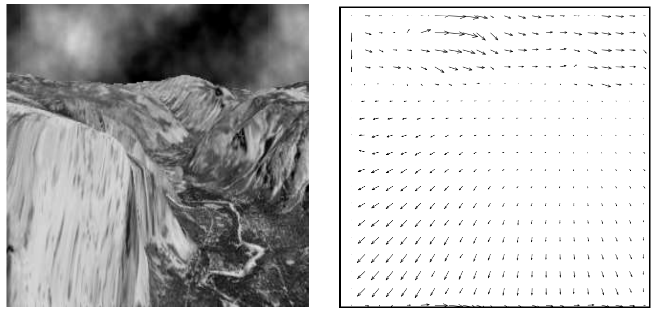
The argument θ corresponds to the angle between the vectors $\mathbf{W}^{1/2}\,\mathbf{b}$ and $\mathbf{W}^{1/2}\mathbf{A}\mathbf{v}$ and characterizes how close $\mathbf{W}^{1/2}\,\mathbf{b}$ is to the image of $\mathbf{W}^{1/2}\mathbf{A}$. A local estimate is replaced only if its confidence measure is larger than the corresponding one at the next coarser scale. Otherwise, the coarser scale estimator is kept. Furthermore, a solution of a local linear system is regarded as not admissible if the linear system is either ill-conditioned or if the length of the estimated central motion vector exceeds some scale-dependent limit. Finally, a motion estimate is set to zero if the local mean of the time derivative at the given location is below a pre-defined noise level.

The final motion estimates at the finest scale j_0 are then interpolated by B-splines to obtain a continuous representation of the motion field.

3) Numerical Results: The performance of the algorithm was tested on synthetic and real image sequences. In particular, we used the well known synthetic sequence “Yosemite”. Since the exact motion field is known, the error of the estimated motion field was computed using the angular error measure as defined in [28]. As real data we used the “Rubik Cube” sequence[1]. One frame of each sequence and its corresponding estimated motion field are shown in Fig. 10 and 11. All sequences were prefiltered with a Binomial filter of variance $\sigma^2 = 1.5$ and a B-spline window of degree 5 at scales $j = 2, \ldots, 4$ was used for moment computation.

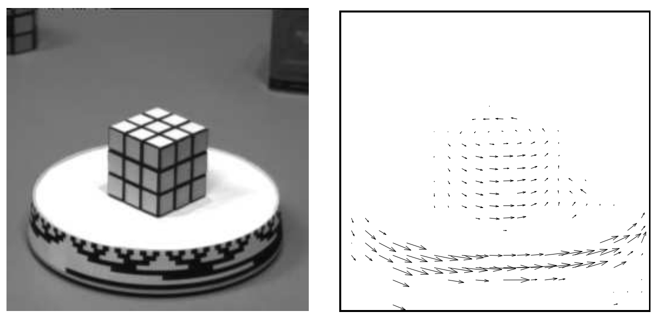
The angular error of the “Yosemite” sequence is $6.33^\circ \pm 9.98^\circ$ with a flow field density of 100%. The error of the corresponding adaptation of the Lucas-Kanade approach (same window, same multiresolution strategy, locally constant motion model) is $7.43^\circ \pm 12.72^\circ$. Barron & al. [28] report an average angular error of an optimized Horn and Schunk method [29] (spatio-temporal prefiltering, 4-point central differences

[1]All sequences were downloaded from *Barron & al.'s* FTP site at `ftp://csd.uwo.ca/pub/vision`.



(a) Frame (b) Estimated motion

Fig. 10. One frame of the Yosemite sequence and its corresponding estimated motion field.



(a) Frame (b) Estimated motion

Fig. 11. One frame of the Rubik Cube sequence and its corresponding estimated motion field.

for differentiation) of $11.26^\circ \pm 16.41^\circ$ with a flow field density of 100%. Their implementation of an improved version of the original Lucas-Kanade method (spatio-temporal prefiltering, rejecting unreliable estimates) only produced a reasonable error for a very sparse velocity field with a density of 35.1%. The rotational movement in the "Rubik Cube" sequence is also clearly recovered. The obtained results also compare favorably with all other methods evaluated in the survey of Barron & al.

IV. Conclusions

We have introduced B-spline-weighted, local geometric moments within windows of dyadic sizes. The weighting ensures isotropy in multiple dimensions and the scalability allows adaptability to local image contents. Computational efficiency was achieved by developing a Mallat-like algorithm to compute these moments at multiple scales.

Local moments provide a powerful set of features that can be used in many sliding window type algorithms. In particular, we demonstrated their usefulness on three different image analysis problems: feature extraction, noise reduction, and optical flow estimation. We proposed basic, moment-based algorithms with promising experimental results. Certain aspects of these generic algorithms can be further improved by tuning them to special applications. Besides the applications mentioned, these moments could also be useful for applications like pattern classification and image segmentation.

APPENDIX I
PROOF OF THEOREM 1

In order to prove the multi-channel two-scale equation (7), we deduce from (3) and (6) that

$$w_p\left(\frac{x}{2}\right) = \frac{1}{2^p}\sum_l h(l)x^p w(x-l).$$

Using the fact that

$$x^p = \big((x-l)+l\big)^p = \sum_{k=0}^{p}\binom{p}{k} l^{p-k}(x-l)^k$$

and applying the definition

$$(x-l)^k w(x-l) = w_k(x-l),$$

we directly obtain (7).

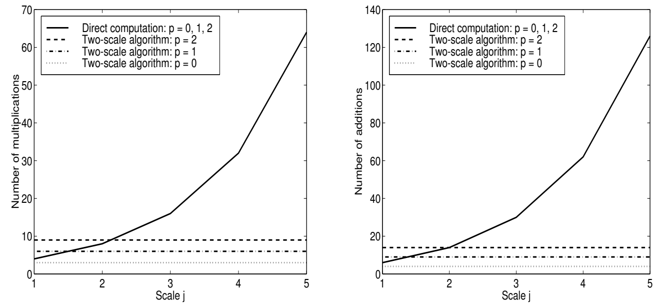
APPENDIX II
PROOF OF COROLLARY 1

By definition (5) we have that

$$\begin{aligned} m_p^{(j+1)}(n) &= 2^{(j+1)p}\int_{\mathbb{R}} w_p\left(\frac{x-n}{2^{j+1}}\right) f(x)\,dx \\ &= 2^{(j+1)p}\int_{\mathbb{R}} w_p\left(\frac{1}{2}\frac{x-n}{2^j}\right) f(x)\,dx. \end{aligned}$$

Using the two-scale equation (7) it follows that

$$\begin{aligned} & m_p^{(j+1)}(n) \\ &= 2^{(j+1)p}\int_{\mathbb{R}} \frac{1}{2^p}\sum_{k=0}^{p}\sum_l h_{p,k}(l) w_k\left(\frac{x-n}{2^j}-l\right) f(x)\,dx \\ &= 2^{jp}\sum_{k=0}^{p}\sum_l h_{p,k}(l)\int_{\mathbb{R}} w_k\left(\frac{x-(n+2^j l)}{2^j}\right) f(x)\,dx. \end{aligned}$$



(a) Number of multiplications

(b) Number of additions

Fig. 12. 1D computational complexity per output point for direct moment computation and usage of two-scale algorithm.

Applying definition (5) yields

$$\begin{aligned} m_p^{(j+1)}(n) &= 2^{jp} \sum_{k=0}^{p} \sum_{l} h_{p,k}(l) \frac{1}{2^{jk}} m_k^{(j)}(n+2^j l) \\ &= \sum_{k=0}^{p} \sum_{l} h_{p,k}(l) 2^{j(p-k)} m_k^{(j)}(n+2^j l). \end{aligned}$$

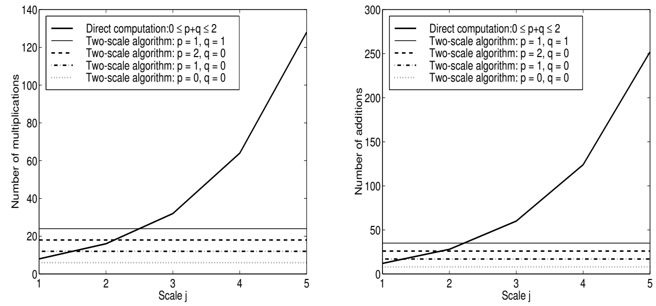
By defining $h_{p,k}^{(j)}(l) = 2^{j(p-k)} h_{p,k}(l)$, we obtain (9).

APPENDIX III
COMPUTATIONAL COMPLEXITY

In the following, we analyze the computational complexity of the recursive two-scale algorithm in the one and two-dimensional case.

A. Computational Complexity in 1D

We assume that the length of the discretized window function w is $2N+1$ and has a corresponding two-scale filter of length $2N+3$. Since the window function is symmetric, the direct calculation of (5) requires $2^j(N+1)$ multiplications and $2^{j+1}(N+1)-2$ additions per output point at scale j, independently of the order p. On the other hand, since the two-scale filters are either symmetric or anti-symmetric, the two-scale algorithms (9) and (11) require $(p+1)(N+2)$ multiplications and $2(p+1)(N+1)+p$ additions for moment order p and are independent of the scale. The computational complexities for scales $j = 1, \ldots, 5$, moment orders $p = 0, 1, 2$ and $N = 1$ are plotted in Fig. 12. Obviously, the use of



(a) Number of multiplications (b) Number of additions

Fig. 13. 2D computational complexity per output point for direct moment computation and usage of two-scale algorithm.

the two-scale algorithm starts paying off at scales $j = 1, 2, 3$ for moment orders $p = 0, 1, 2$, respectively. Since in practice usually low order moments are used, the proposed computation scheme is much more efficient at coarser scales.

B. Computational Complexity in 2D

For the two-scale algorithm (14), the number of multiplications and additions per output point at scale j are $2(p+1)(q+1)(N+2)$ and $4(p+1)(q+1)(N+1)+(p+1)(q+1)-1$, respectively; in contrast, the direct computation (13) requires $2^{j+1}(N+1)$ multiplications and $2^{j+2}(N+1)-4$ additions. The computational complexities for scales $j = 1, \ldots, 5$, moment orders $0 \leq p+q \leq 2$ and $N = 1$ are plotted in Fig. 13. The two-scale algorithm clearly pays off at coarser scales.

APPENDIX IV
COMPUTATION OF THE PROBABILITY DENSITY FUNCTION OF A WEIGHTED LEAST-SQUARES RESIDUAL

Let $\mathbf{Ax} = \widehat{\mathbf{b}}$ be an overdetermined linear system of size $n \times p$, $n > p$, and maximum rank p. We assume that the noisy observation $\widehat{\mathbf{b}}$ is given by $\widehat{\mathbf{b}} = \mathbf{b} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$ is jointly normally distributed and $\mathbf{b} = \mathbf{Ax}$.
The weighted least-squares estimator $\mathbf{x}$ is obtained by minimizing

$$r^2 = \left(\mathbf{Ax} - \widehat{\mathbf{b}}\right)^T \mathbf{W} \left(\mathbf{Ax} - \widehat{\mathbf{b}}\right),$$

where $\mathbf{W}$ is a diagonal $(n \times n)$-matrix of weights. Using the fact that $\mathbf{x} = \left(\mathbf{A}^T\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{W}\,\widehat{\mathbf{b}}$ and that $\widehat{\mathbf{b}} = \mathbf{A}\mathbf{x} + \boldsymbol{\varepsilon}$, we obtain

$$r^2 = \boldsymbol{\varepsilon}^T\mathbf{C}\,\boldsymbol{\varepsilon}, \tag{41}$$

where $\mathbf{C} = \mathbf{W} - \mathbf{W}\mathbf{A}\left(\mathbf{A}^T\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{W}$. Since $\mathbf{C}$ is symmetric, it can be decomposed as $\mathbf{C} = \mathbf{U}^T\boldsymbol{\Lambda}\mathbf{U}$, where $\mathbf{U}$ is an orthogonal matrix and $\boldsymbol{\Lambda}$ is a real diagonal matrix containing the eigenvalues of $\mathbf{C}$. Therefore, (41) can be expressed as

$$\begin{aligned} r^2 &= \left(\mathbf{U}\boldsymbol{\varepsilon}\right)^T\boldsymbol{\Lambda}\left(\mathbf{U}\boldsymbol{\varepsilon}\right) \\ &= \boldsymbol{\eta}^T\boldsymbol{\Lambda}\,\boldsymbol{\eta}, \end{aligned}$$

where $\boldsymbol{\eta} = \mathbf{U}\boldsymbol{\varepsilon}$ is also a $N(\mathbf{0}, \sigma^2\mathbf{I})$-distributed random variable due to the orthogonality of $\mathbf{U}$. Since $\mathbf{C}$ is by construction of rank $(n-p)$, we have that

$$r^2 = \sum_{k=1}^{n-p}\lambda(k)\boldsymbol{\eta}^2(k),$$

where $\lambda(k)$ denote the non-zero diagonal elements of $\boldsymbol{\Lambda}$.

Now, the $\boldsymbol{\eta}(k)/\sigma \sim N(0,1)$ are independently normally distributed so that their squares follow a χ_1^2-distribution. Consequently, the probability density function (pdf) of r^2/σ^2 is given by the convolution of χ_1^2-pdf's dilated and scaled by the factors $\lambda(k)$. Since the characteristic function (Fourier transform of the pdf) of a χ_1^2-distribution is given by

$$f_{\chi_1^2}(\omega) = \frac{1}{(1-2i\omega)^{1/2}},$$

the characteristic function of r^2/σ^2 is given by

$$f_{r^2/\sigma^2}(\omega) = \prod_{k=1}^{n-p}\frac{1}{\left(1-2i\lambda(k)\omega\right)^{1/2}}.$$

For n sufficiently large, the pdf converges to a Gaussian as a consequence of the central limit theorem.

REFERENCES

[1] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inform. Theory*, pp. 179–187, Feb. 1962.

[2] C. Teh and R. Chin, "On image analysis by the methods of moments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 4, pp. 496–513, July 1988.

[3] S. Dudani, K. Breeding, and R. McGhee, "Aircraft identification by moment invariants," *IEEE Trans. Comput.*, vol. 26, no. 1, pp. 39–46, Jan. 1977.

[4] Y. Abu-Mostafa and D. Psaltis, "Recognitive aspects of moment invariants," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 698–706, Nov. 1984.

[5] M. Teague, "Image analysis via the general theory of moments," *J. Optical Soc. Am.*, vol. 70, pp. 920–930, Aug. 1980.

[6] S. Liao and M. Pawlak, "On image analysis by moments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 3, pp. 254–266, Mar. 1996.

[7] A. Khotanzad and Y. Hong, "Invariant image recognition by zernike moments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 5, pp. 489–497, May 1990.

[8] A. Bhatia and E. Wolf, "On the circle polynomials of zernike and related orthogonal sets," *Proc. Cambridge Philosophical Society*, vol. 50, pp. 40–48, 1954.

[9] T. Reiss, *Recognizing planar objects using invariant image features. Lecture notes in computer science*. Springer, Berlin, 1993, vol. 676.

[10] B. Li and J. Shen, "Fast calculation of local moments and application to range segmentation," *Proc. International Conference on Pattern Recognition, The Haage*, vol. C, pp. 298–301, 1992.

[11] M. Tuceryan, "Moment-based texture segmentation," *Pattern Recognition Letters*, vol. 15, pp. 659–668, July 1994.

[12] Q. Gao and F. Yin, "Two-dimensional direction-based interpolation with local centered moments," *Graphical Models and Image Processing*, vol. 61, no. 6, pp. 323–339, Nov. 1999.

[13] J. Martinez and F. Thomas, "Efficient computation of local geometric moments," *IEEE Trans. Image Processing*, vol. 11, no. 9, pp. 1102–1111, 2002.

[14] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 674–693, July 1989.

[15] J.-W. Lin, A. Laine, and S. Bergmann, "Improving PET-based physiological quantification through methods of wavelet denoising," *IEEE Trans. Biomed. Eng.*, vol. 48, no. 2, pp. 202–212, Feb. 2001.

[16] R. Nowak, "Wavelet-based Rician noise removal for magnetic resonance imaging," *IEEE Trans. Image Processing*, vol. 8, no. 10, pp. 1408–1419, Oct. 1999.

[17] A. Laine, S. Schuler, J. Fan, and W. Huda, "Mammographic feature enhancement by multiscale analysis," *IEEE Trans. Med. Imag.*, vol. 13, no. 4, pp. 725–740, Dec. 1994.

[18] J.-P. Antoine, D. Barache, R. Cesar Jr., and L. da F. Costa, "Multiscale shape analysis using the continuous wavelet transform," in *Proc. IEEE International Conference on Image Processing (ICIP'96)*, vol. I, Lausanne, Switzerland, Sept. 16–19 1996, pp. 291–294.

[19] A. Savitzky and M. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[20] B. Lucas and T. Kanade, "An iterative image restoration technique with an application to stereo vision," *Proc. DARPA IU Workshop*, pp. 121–130, 1981.

[21] M. Shensa, "The discrete wavelet transform: wedding the à trous and mallat algorithms," *IEEE Trans. Signal Processing*, vol. 40, no. 10, pp. 2464–2482, Oct. 1992.

[22] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Mag.*, vol. 16, no. 6, pp. 22–38, Nov. 1999.

[23] M. Unser and T. Blu, "Wavelet theory demystified," *IEEE Trans. Signal Processing*, vol. 51, no. 2, pp. 470–483, Feb. 2003.

[24] J. Dubochet, M. Adrian, J. Chang, J. Homo, J. Lepault, A. McDowall, and P. Schultz, "Cryo-electron microscopy of vitrified specimens," *Quarterly Review of Biophysics*, vol. 21, no. 2, pp. 129–228, 1988.

[25] J. Bigün, G. Granlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 8, pp. 775–790, Aug. 1991.

[26] T. Blu and M. Unser, "The fractional spline wavelet transform: Definition and implementation," in *Proc. Twenty-Fifth IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'00)*, vol. I, Istanbul, Turkey, June 5–9 2000, pp. 512–515.

[27] J. S. Lim, *Two-Dimensional Signal and Image Processing*. Prentice Hall, 1990, pp. 536–540.

[28] J. Barron, D. Fleet, S. Beauchemin, and T. Burkitt, "Performance of optical flow techniques," *Intl. J. Computer Vision*, vol. 12, pp. 43–77, 1994.

[29] B. Horn and B. Schunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.