EFFICIENT MOMENT COMPUTATION OVER POLYGONAL DOMAINS WITH AN APPLICATION TO RAPID WEDGELET APPROXIMATION

F. FRIEDRICH, L. DEMARET, H. FÜHR AND K. WICKER

Abstract. Many algorithms in image processing rely on the computation of sums of pixel values over a large variety of subsets of the image domain. This includes the computation of image moments for pattern recognition purposes, or adaptive smoothing and regression methods, such as wedgelets.

In the first part of the paper, we present a general method which allows the fast computation of sums over a large class of polygonal domain. The approach relies on the idea of considering polygonal domains with a fixed angular resolution, combined with an efficient implementation of a discrete version of Green's theorem.

The second part deals with the application of the new methodology to a particular computational problem, namely wedgelet approximation. Our technique results in a speedup of $O(10^3)$ by comparison to preexisting implementations. A further attractive feature of our implementation is the instantaneous access to the full scale of wedgelet minimizers. We introduce a new scheme that replaces the locally constant regression underlying wedgelets by basically arbitrary local regression models. Due to the speedup obtained by the techniques explained in the first part, this scheme is computationally efficient, and at the same time much more flexible than previously suggested methods such as wedgelets or platelets.

In the final section we present numerical experiments showing the increase in speed and flexibility.

Key words. Wedgelets, platelets, image approximation, image moments, polygonal domains, discrete Green's theorem, digital lines.

AMS subject classifications. 68U10, 65K10, 26B20, 52C99.

1. Introduction. This paper presents a new technique for the rapid computation of sums over varying image domains. The algorithms are particularly useful whenever sums of greyvalues over a large number of different image domains are needed, for a fixed image. Techniques for this type of problem have been considered in the past mainly for the computation of image moments, see e.g. [1, 2, 3, 4, 5]. These papers mostly deal with specific moments (such as geometric, Legendre or Zernike moments), and often only for binary images, hence the algorithms usually depend on specific properties of the moments and cannot be directly adapted to our problem. However, a recurring theme in these papers which also plays a role here is the observation that the complexity of the algorithm may be reduced by appealing to a discrete version of Green's theorem. Recall that Green's theorem relates integration of a given function on a twodimensional (suitably regular, connected and simply connected) domain to the integration of associated functions over the boundary of the domain. In the discrete setting, one expects the reduction in dimension to result in a reduction of computational complexity.

Our approach is based on the simple observation that for *polygonal domains*, the integration step boils down to a summation over the vertices of the polygon, once certain auxiliary functions are known. The discretisation of this scheme relies on precomputed functions containing integrals over suitably chosen domains. Basically, each auxiliary function corresponds to a possible angle of an edge with the standard coordinate axis. This way, the *angular resolution* of the polygons which may be treated after the precomputation step can be prescribed in a direct and convenient way, this at linear cost, for both computational and memory requirements. As an example, discussed briefly in Remark 2.5 below, after computing the auxiliary functions for the angles $0, \pi/2$, integrals over arbitrary rectangles are computable in constant time.

Remark 2.5 illustrates that the use of Green's theorem for summation purposes is based on cancellation. To make this effect prevail in the discrete context, a careful treatment of discrete lines and polygons is necessary. The design and efficient treatment of these discrete lines turns out to be the keystone of our approach.

The second half of our paper deals with the application of our techniques to wedgelet approximations. We present an implementation of an algorithm suggested by Donoho [6], and demonstrate that our techniques result in a drastic speedup, by comparison to the only publicly available implementation as part of the package BeamLab [25]. In a sense we solve a problem that had been left open in [6]: Wedgelet approximations are piecewise constant approximations of an image, where the pieces – the *wedges* – are obtained by splitting dyadic squares along straight lines. Approximations which minimise a certain functional are of particular interest (see Section 3 below for details). Donoho proposed an algorithm for the rapid computation of the minimisers, resting on an assumption that was not further addressed in [6] – nor, to our knowledge, anywhere else in the literature. The assumption was that constant regression over arbitrary wedges could be efficiently implemented. Since constant regression depends on the image moments of orders zero, it is obvious that our techniques can be brought to bear on this problem. Hence we obtain a very efficient and transparent implementation of Donoho's algorithm, again with a conveniently prescribable angular resolution at linear cost. This implementation is available at [24].

Moreover, our implementation allows to discuss other regression models than piecewise constant functions. The observation that Donoho's model can be extended to include piecewise approximation by (e.g.) affine functions has already been made by Willett and Nowak [7]. Our discussion below includes a quite general discussion of such schemes. The reason we present this generalisation, which in itself is quite straightforward and may be considered by some as being of purely theoretical interest, is that we want to point out the potential of wedgelet methodology combined with our techniques. While with BeamLab, even piecewise constant approximation of moderate size images is rather time-consuming, the speedup due to our techniques allows to design more involved models, which can also be implemented efficiently.

2. Discrete Green's theorem and rapid summation over polygonal domains. The use of discrete versions of Green's theorem for image analysis purposes can be traced back to Tang [8]. We first recall this theorem, and its application to our central problem, for the continuous setting. Mathematically speaking, the following subsection is not necessary for the following, but it serves as motivation and template of the definitions and results presented for the discrete case.

2.1. The continuous case. We restrict the discussion to bounded domains contained in the positive quadrant $\mathcal{Q}^+ = \mathbb{R}^+ \times \mathbb{R}^+$; its boundary $\partial\mathcal{Q}^+ = (\{0\} \times \mathbb{R}^+) \cup (\mathbb{R}^+ \times \{0\})$ will also play a prominent role. Given a measurable function $F : \mathbb{R}^2 \to \mathbb{R}$, and continuously differentiable $\phi : [a, b] \to \mathbb{R}^2$ written as $\phi(t) = (\phi_1(t), \phi_2(t))$, let

$$\int_\phi F dx = \int_a^b F(\phi(t))\phi_1'(t)dt \quad ,$$

and

$$\int_\phi F dy = \int_a^b F(\phi(t))\phi_2'(t)dt \quad .$$

The definition extends to piecewise differentiable curves in the obvious way. The following version of Green's theorem is relevant to our purposes.

THEOREM 2.1. *Let $\Omega \subset \mathcal{Q}^+$ be a bounded compact domain, with piecewise continuously differentiable boundary $\partial\Omega$. Assume that $F \in L^\infty(\Omega)$ is real-valued, and define $Q : \mathcal{Q}^+ \to \mathbb{R}$ as*

$$Q(x,y) = \int_0^y F(x,t)dt \quad . \tag{2.1}$$

Then

$$\int_\Omega F(x,y)dxdy = \int_{\partial\Omega} Qdx \quad . \tag{2.2}$$

For purposes of rapid integration methods, the key feature of Green's formula is that it reduces the dimension of the integration domain by one. Let us describe how the formula simplifies further in the case of polygonal domains. Assume from now on that $\partial\Omega$ is a polygon, with nodes $z_1, \ldots, z_n, z_{n+1} = z_1 \in \mathbb{R}^2$. As Ω is simply connected, the polygon does not intersect itself. Given $z, \widetilde{z} \in \mathbb{R}^2$, we let $[z, \widetilde{z}]$ denote the straight line segment from z to $\widetilde{z}$, parameterised by $t \mapsto z + t(\widetilde{z} - z), t \in [0,1]$.

For an angle $\vartheta \in]-\pi/2, \pi/2]$, define $v_\vartheta = (\cos(\vartheta), \sin(\vartheta))$. The convention $-\pi/2 < \vartheta \leq \pi/2$ entails that v_ϑ points to the right. Given $z \in \mathcal{Q}^+$, let $I_\vartheta(z)$ denote the intersection of the halfline $z + \mathbb{R}^- v_\vartheta$ with $\partial\mathcal{Q}^+$.

We define an auxiliary function K_ϑ on $\mathcal{Q}^+$ by

$$K_\vartheta(z) = \int_{[I_\vartheta(z), z]} Qdx \quad . \tag{2.3}$$

The following lemma is then easily verified:

LEMMA 2.2. *Let $z, \widetilde{z} \in \mathcal{Q}^+$, with $z - \widetilde{z} \in \mathbb{R} \cdot v_\vartheta$. Then*

$$\int_{[z,\widetilde{z}]} Qdx = \operatorname{sign}(\langle \widetilde{z} - z, v_\vartheta \rangle) \cdot (K_\vartheta(\widetilde{z}) - K_\vartheta(z)) \quad . \tag{2.4}$$

We note that for the degenerate case $\vartheta = \pi/2$, $K_\vartheta = 0$. Plugging our definitions into Green's theorem yields

COROLLARY 2.3. *Let Ω be a simply connected domain with the boundary given by a polygon through $z_1, \ldots, z_n, z_{n+1} = z_1$. For $\vartheta \in]-\pi/2, \pi/2]$ define K_ϑ as in (2.3). Then, if ϑ_i denotes the angle between the x-axis and $z_{i+1} - z_i$, we have*

$$\int\int_\Omega F(x,y)dxdy = \sum_{i=1}^n \operatorname{sign}(\langle z_{i+1} - z_i, v_{\vartheta_i} \rangle) \cdot (K_{\vartheta_i}(z_{i+1}) - K_{\vartheta_i}(z_i)) \quad . \tag{2.5}$$

Note that the evaluation of the right-hand side involves $O(n)$ operations, *supposing the K_{ϑ_i} are known.* This simple observation makes little sense in the continuous setting, but it contains the key to the algorithm which we will derive for the discrete setting.

A geometrically intuitive understanding of the corollary can be obtained by use of the following, easily verified, observation.

LEMMA 2.4. *For $z \in \mathcal{Q}^+$ and $\vartheta \in]-\pi/2, \pi/2]$, let $\Omega_\vartheta(z)$ denote the compact set bounded by $\partial\mathcal{Q}^+$, the line segment $[I_\vartheta(z), z]$ and the vertical line through z. Then*

$$K_\vartheta(z) = \int_{\Omega_\vartheta(z)} f(x,y)dxdy \quad . \tag{2.6}$$

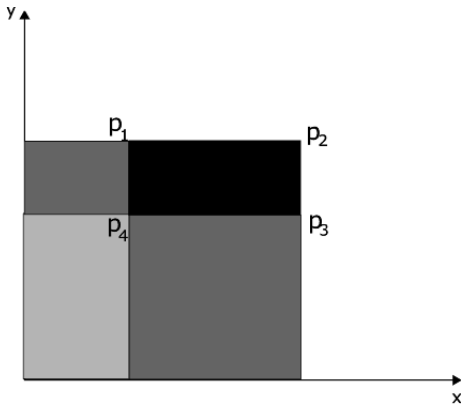

FIG. 2.1. *Left: Illustration of the cancellation principle for the black rectangle, with use of formula 2.7.*

The lemma also covers the degenerate case $\vartheta = \pi/2$, as $\Omega_\vartheta(z)$ is then a line segment, a set of measure zero.

REMARK 2.5. *Using the lemma we can recognize Green's theorem as an elegant and efficient way of handling cancellation: For illustration purposes, let us consider rectangles. Denote the four corners of the rectangular domain Ω by $p_1, \ldots, p_4$, numbered as in Figure 2.1. We claim that by the previous lemma,*

$$\int_\Omega f(x,y)dxdy = K_0(p_2) - K_0(p_1) - K_0(p_3) + K_0(p_4) \quad . \tag{2.7}$$

To see this observe that Ω is obtainable by subtracting $\Omega_0(p_2) \cup \Omega_0(p_3)$ from $\Omega_0(p_1)$. However, by subtracting the integrals accordingly, the integral over the intersection $\Omega_0(p_2) \cap \Omega_0(p_3) = \Omega_0(p_4)$ is subtracted twice, hence needs to be added again. On the other hand, observing that $K_{\pi/2} = 0$, we see that (2.7) is a special case of (2.5).

Naïvely one might expect that more complex –simply connected– domains could entail increasingly more complicated forms of cancellation. However Corollary 2.3 offers a simple and general way of handling all of them; in particular the complexity is linear in the number of the vertices, regardless of the shape. The following section is dedicated to the description of a discrete analog.

2.2. The discrete case. In this section we assume that $f : \mathcal{Q}^+ \cap \mathbb{Z}^2 \to \mathbb{R}$ is given, and describe versions of Green's theorem that allow the quick summation of function values over certain finite discrete domains $\Omega \subset \mathcal{Q}^+ \cap \mathbb{Z}^2$. In the following we restrict the presentation to *simple* domains, which are defined in 2.11. This restriction considerably simplifies the arguments and notations, while still covering the wedgelet case treated below. Note however that similar results can be formulated and proved for general simply connected domains (see [26]) , and should also be useful in the general setting.

The key to an efficient discretisation of the scheme described in the previous subsection consists in a suitable definition of straight lines, as follows:

DEFINITION 2.6. *Let $\alpha \in]-\pi/4, 3\pi/4]$ be given and let $d_x := \cos\alpha$ and $d_y := \sin\alpha$. Define*

$$\delta_\alpha := \max\{|d_x|, |d_y|\} \quad \textit{and} \quad v_\alpha^\perp =: \begin{cases} (-d_y, d_x) & \textit{if } |d_x| \geq |d_y| \\ (d_y, -d_x) & \textit{otherwise.} \end{cases}$$

The **digital line through the origin in direction** α *is then defined as*

$$L_\alpha^0 := \{p \in \mathbb{Z}^2 : -\frac{\delta_\alpha}{2} < \langle p, v_\alpha^\perp \rangle \leq \frac{\delta_\alpha}{2}\}. \tag{2.8}$$

Moreover, we define L_α^n *for* $n \in \mathbb{Z}$ *as*

$$L_\alpha^n := \{p \in \mathbb{Z}^2 : (n - \frac{1}{2})\delta_\alpha < \langle p, v_\alpha^\perp \rangle \leq (n + \frac{1}{2})\delta_\alpha\}. \tag{2.9}$$

We call n *the* **line number** *of the digital line* L_α^n.

Let in the following the mapping $\mathrm{Round} : \mathbb{R} \to \mathbb{Z}$ be defined by

$$\mathrm{Round}(x) = \max\{i \in \mathbb{Z} : i \leq x + 1/2\}.$$

The choice $\alpha \in]-\pi/4, 3\pi/4]$ leads to the following result:

LEMMA 2.7. *If a digital line* L_α^n *is a*

(a) **flat line**, *i.e.* $d_x \geq d_y$, *then with* $y_\alpha(x) := \mathrm{Round}(x \cdot d_y/d_x)$ *it holds that*

$$L_\alpha^n = (0, n) + \{(x, y_\alpha(x)) : x \in \mathbb{Z}\} \tag{2.10}$$

and if it is a

(b) **steep line**, *i.e.* $d_y > d_x$, *then with* $x_\alpha(y) := \mathrm{Round}(y \cdot d_x/d_y)$ *it holds that*

$$L_\alpha^n = (n, 0) + \{(x_\alpha(y), y) : y \in \mathbb{Z}\}. \tag{2.11}$$

Proof. We prove (a) first. If $d_x \geq d_y$ then, by $\alpha \in]-\pi/4, 3\pi/4]$ it holds that $|d_x| \geq |d_y|$, $d_x > 0$ and

$$\begin{aligned} L_\alpha^n &= \{(x, y) : (n - \frac{1}{2})d_x < -d_y \cdot x + d_x \cdot y \leq (n + \frac{1}{2})d_x\} \\ &= \{(x, y) : y \leq x \cdot \frac{d_y}{d_x} - n + \frac{1}{2} < y + 1\} \\ &= (0, n) + \{(x, y) : y \leq x \cdot \frac{d_y}{d_x} + \frac{1}{2} < y + 1\}. \end{aligned}$$

For (b) the proof can be done the same way. □

In other words, L_α^n is obtained by shifting L_α^0 by n in the vertical direction for flat lines and by shifts in the horizontal direction for steep lines. We remark that this definition of discrete lines is not new; they arise as output of the Bresenham algorithm used in computer graphics [9].

The following lemma notes the central property of digital lines: The digital lines corresponding to a fixed angle provide a partition of the image domain. In view of the observation, made in Remark 2.5, that Green's theorem – at least in the form discussed here – is primarily about cancellation, this simple fact will be crucial.

LEMMA 2.8. *Let* $\alpha \in]-\pi/4, 3\pi/4]$ *be given. The lines* $(L_\alpha^n)_{n\in\mathbb{Z}}$ *partition* $\mathbb{Z}^2$, *i.e.* $\mathbb{Z}^2 = \bigcup_{n\in\mathbb{Z}}^\bullet L_\alpha^n$.

Proof. The statement follows from (2.9), equivalently $L_\alpha^n = \{p \in \mathbb{Z}^2 : \langle p, v_\alpha^\perp \rangle \in](n - 1/2)\delta_\alpha, (n + 1/2)\delta_\alpha)]\}$, since the intervals $(](n - 1/2)\delta_\alpha, (n + 1/2)\delta_\alpha)])_{n\in\mathbb{Z}}$ form a disjoint union of $\mathbb{R}$. □

We next want to define the discrete analogs of the auxiliary functions K_ϑ. For this purpose we number the points on a digital line L_α^n, in such a way that following

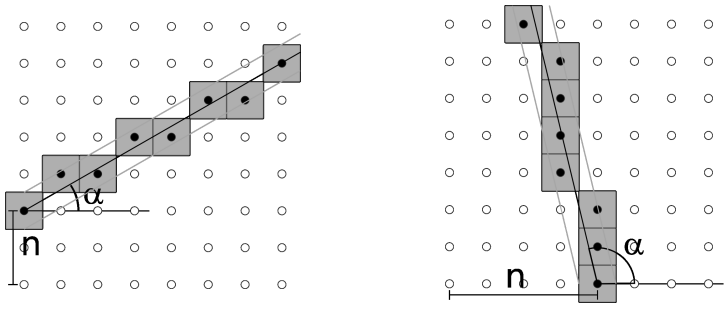

FIG. 2.2. *A flat and a steep discrete line.*

the points in ascending order, starting at the points numbered by zero, corresponds to going through L_α^n in direction (d_x, d_y), starting at a suitable element of $L_\alpha^n \cap \partial\mathcal{Q}^+$.

Let $(x_0, y_0) \in \mathbb{Z}^2$ and let for flat lines the line number given by $n = y_0 - y_\alpha(x_0)$ and for steep lines by $n = x_0 - x_\alpha(y_0)$, then $(x_0, y_0) \in L_\alpha^n$. Therefore the set of pixels on the line starting at $x = 0$ for flat lines and at $y = 0$ for steep lines and ending in $p = (x_0, y_0)$ is given by

$$[\circ, p]_\alpha := \begin{cases} (0, y_0 - y_\alpha(x_0)) + \bigcup_{x=0}^{x_0} \{(x, y_\alpha(x))\} & \text{if } d_x \geq d_y \\ (x_0 - x_\alpha(y_0), 0) + \bigcup_{y=0}^{y_0} \{(x_\alpha(y), y)\} & \text{otherwise.} \end{cases}$$

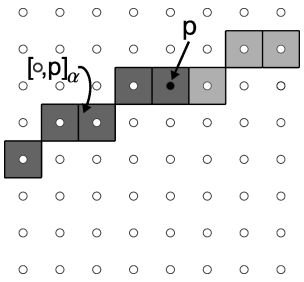

FIG. 2.3. *A line segment* $[\circ, p]_\alpha$.

Let in the following the set of pixels below some pixel $p = (x_0, y_0)$ be given by $\mathcal{C}(p) := \{(x_0, y)\}_{y=0,\ldots,y_0}$. Moreover, consider the set of points between the line segment $[\circ, p]_\alpha$ and the x-axis defined by

$$\mathcal{W}_\alpha(p) := \bigcup_{q \in [\circ, p]_\alpha} \mathcal{C}(p).$$

Now for rectangular subsets $S = \{0, \ldots, w\} \times \{0, \ldots, h\}$ $(w, h \in \mathbb{N})$ of $\mathbb{Z}^2$, let **image data** $f \in \mathbb{R}^S$, $f = (f(p))_{p \in S}$ be given. We denote the corresponding sum of values in columns below the point $p = (x_0, y_0)$ by $C(p) := \sum_{y=0}^{y_0} f((x_0, y))$, in rectangles left and below $p = (x_0, y_0)$ by $R(p) := \sum_{x=0}^{x_0} C((x, y_0))$ and below the piece of line $\mathcal{W}_\alpha(p)$ by

$$W_\alpha(p) := \sum_{q \in \mathcal{W}_\alpha(p)} f(q).$$

The following computation rules hold for columns and rectangles

$$C((x,y+1)) = C((x,y)) + f((x,y+1)) \tag{2.12}$$
$$R((x+1,y)) = R((x,y)) + C((x+1,y)). \tag{2.13}$$

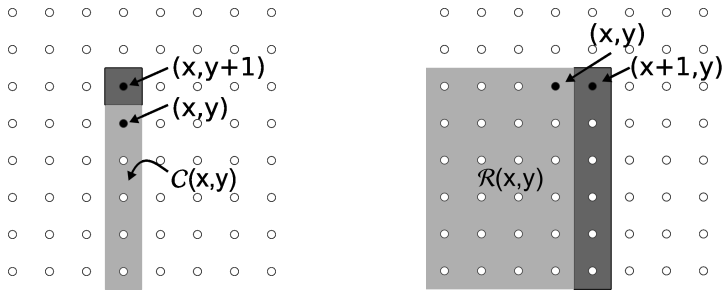

FIG. 2.4. *Summation rule for columns and rectangles.*

For the sums below a piece of line we obtain the following results. Let $p = (x,y) \in S$. For flat lines, i.e. $d_x \geq d_y$, with $p^+ = (x+1, y - y_\alpha(x) + y_\alpha(x+1))$ it holds that

$$\mathcal{W}_\alpha(p^+) = \mathcal{W}_\alpha(p) \,\dot\cup\, \mathcal{C}(p^+)$$

and therefore, if $p^+ \in S$,

$$W_\alpha(p^+) = W_\alpha(p) + C(p^+). \tag{2.14}$$

For steep lines, i.e. $d_y > d_x$, with $p^+ = (x - x_\alpha(y) + x_\alpha(y+1), y+1)$ it holds that

$$\mathcal{W}_\alpha(p^+) = \mathcal{W}_\alpha(p) \,\dot\cup\, \begin{cases} \mathcal{C}(p^+) & \text{if } x_\alpha(y) \neq x_\alpha(y^+) \\ \{p^+\} & \text{if } x_\alpha(y) = x_\alpha(y^+) \end{cases}$$

and thus, if $p^+ \in S$,

$$W_\alpha(p^+) = W_\alpha(p) + \begin{cases} C(p^+) & \text{if } x_\alpha(y) \neq x_\alpha(y^+) \\ f(p^+) & \text{if } x_\alpha(y) = x_\alpha(y^+). \end{cases} \tag{2.15}$$

Note that the computation rules for the sums C and R are special cases of the one for W_α, namely the ones with $\alpha = \pi/2$ and $\alpha = 0$.

The computation rules for C,R and W_α immediately lead to the following result:

LEMMA 2.9. *The matrices C, R and W_α can be computed in $O(|S|)$.*

Proof. For the computation of C note that $C(p) = f(p)$ for all points p with $p^y = 0$. The rest of C can be computed using $C(x,y) = C(x,y-1) + f(x,y)$ in $O(|S|)$. Each line L_α^n that intersects S has at least one point on the lower or left boundary of S. Therefore there are at most $w+h$ such lines. For each line, the left or lower intersecting point p can be computed in $O(1)$, then $W_\alpha(p) = C(p)$. Using the rules (2.14) or (2.15), the values of W_α can then for each point on each line successively be computed in $O(1)$. Since the lines partition S this operation has to be performed $|S|$ times and thus the computation of $W_\alpha(p)$ can be done in $O((w+h)+|S|) = O(|S|)$ steps. □

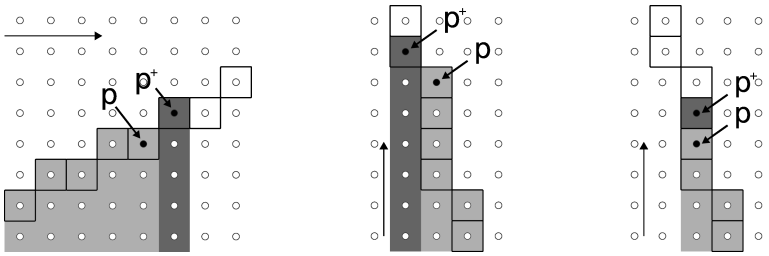

FIG. 2.5. *Summation rule for digital trapezes.*

LEMMA 2.10. *Let two points $p, q \in \mathbb{Z}^2$ be given. Then there exists $n \in \mathbb{Z}$ and $\alpha \in]-\pi/4, 3\pi/4]$ such that $p, q \in L^n_\alpha$.*

Proof. For instance we consider the angle α corresponding to $(d_x, d_y) = (q^x - p^x, q^y - p^y)$. We can restrict ourselves to the case of a flat line, the steep case being similar. Let r denote the point $(0, r^y) \in \mathbb{R}^2$, such that r lies on the continuous line through p and q. If we consider $n = \mathit{Round}(r_y)$, we have $n - r^y = y_\alpha(p^x) - p^x = y_\alpha(q^x) - q^x$, then $p, q \in L^n_\alpha$. □

Note that the discrete line L^n_α through two points p and q is in general not unique.

Let $p, q \in \mathbb{Z}^2$, $\alpha \in [-\pi/4, 3\pi/4[$ and $n \in \mathbb{Z}$ be given such that $p, q \in L^n_\alpha$. Assume that p is 'before' q on the line, i.e. assume that $p^x \leq q^x$ if $\alpha \in [-\pi/4, \pi/4[$, and that $p^y \leq q^y$ if $\alpha \in]\pi/4, 3\pi/4]$. The discrete line segment between p and q, i.e. the set of points between p and q (including p and q) on the line L^n_α, is then given by

$$[p, q]_\alpha := [\circ, q]_\alpha \setminus [\circ, p]_\alpha \cup \{p\}.$$

Accordingly, by $\overline{\mathcal{W}}_\alpha(p, q)$ we denote the **digital trapezoid** under $[p, q]_\alpha$, i.e. the set of points below this piece of line,

$$\overline{\mathcal{W}}_\alpha(p, q) = \bigcup_{r \in [p,q]_\alpha} \mathcal{C}(r).$$

For the sum over values of points below the piece $\overline{\mathcal{L}}_\alpha(p, q)$ it holds that

$$\sum_{r \in \overline{\mathcal{W}}_\alpha(p,q)} f(r) = W_\alpha(q) - W_\alpha(p) + C(p).$$

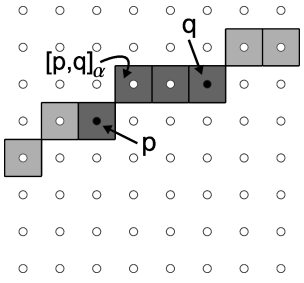

FIG. 2.6. *Segment $[p, q]_\alpha$.*

Now let two arbitrary points p, q on a line L^n_α, $\alpha \in]-\pi/4, 3\pi/4]$, be given. We denote the sum of values below the piece of line between p and q by

$$\overline{W}_\alpha(p,q) := \begin{cases} W_\alpha(q) - W_\alpha(p) + C(p) & \text{if } p^x \le q^x \text{ and } \alpha \in]-\pi/4, \pi/4] \\ W_\alpha(q) - W_\alpha(p) + C(p) & \text{if } p^y \le q^y \text{ and } \alpha \in]\pi/4, 3\pi/4] \\ W_\alpha(p) - W_\alpha(q) + C(q) & \text{otherwise.} \end{cases} \tag{2.16}$$

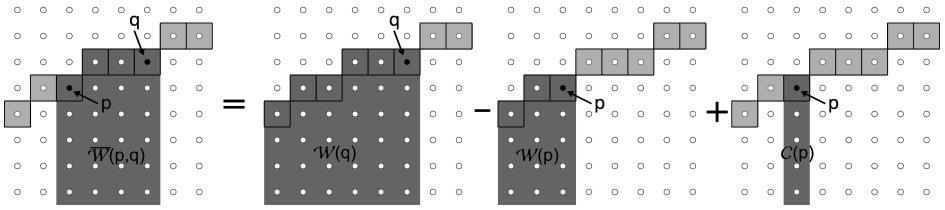

FIG. 2.7. *Summation over wedges.*

The next definition describes the class of domains to which we apply the summation techniques. In a first step, this is done by prescribing the upper boundary as a union of line segments and using the x-axis as lower boundary. More general lower boundaries can then be introduced by considering the set-theoretic difference of two such sets. See Figure 2.2 for an example.

DEFINITION 2.11. *Let $p_0 \ldots, p_n \in \mathcal{Q} \cap \mathbb{Z}^2$ and $\alpha_1, \ldots, \alpha_n \in]-\pi/4, 3\pi/4]$ be given with the following properties:*

1. *$p_0^x < p_1^x < \ldots < p_n^x$.*
2. *For each $1 \le i \le n$ there is some $k_i \in \mathbb{Z}$ such that $p_{i-1}, p_i \in L^{k_i}_{\alpha_i}$.*
3. *$[p_{i-1}, p_i]_{\alpha_i} \cap [p_i, p_{i+1}]_{\alpha_{i+1}} = \{p_i\}$ for all $1 \le i < n$.*

We then define the set $\mathcal{D}(p_i, \alpha_i; i = 0, \ldots, n)$ by

$$\mathcal{D}(p_i, \alpha_i; i = 0, \ldots, n) = \bigcup_{i=1}^{n} \overline{\mathcal{W}}_{\alpha_i}(p_{i-1}, p_i) \quad . \tag{2.17}$$

We call $\Omega \subset \mathbb{N}^2$ **simple polygonal domain of order** *n if there exist $\Omega_1 = \mathcal{D}(p_i, \alpha_i; i = 1, \ldots, m)$ and $\Omega_2 = \mathcal{D}(q_i, \vartheta_i; i = 1, \ldots, k)$ with $n = k+m$, $p_0^x = q_0^x$, $p_m^x = q_k^x$, $\Omega_2 \subset \Omega_1$ and $\Omega = \Omega_1 \setminus \Omega_2$.*

Now the following discrete Green's theorem is easily proved.

THEOREM 2.12. *Let Ω be a simple domain of order n with $\Omega = \Omega_1(p_i, \alpha_i, i = 1, \ldots, m) \setminus \Omega_2(q_i, \vartheta_i, i = 1, \ldots, k)$. Then*

$$\sum_{r \in \Omega} f(r) = \sum_{i=1}^{m} \overline{W}_{\alpha_i}(p_{i-1}, p_i) - \sum_{i=1}^{m-1} C(p_i) - \sum_{i=1}^{k} \overline{W}_{\vartheta_i}(q_{i-1}, q_i) + \sum_{i=1}^{k-1} C(q_i). \tag{2.18}$$

Proof. Firstly, since $\Omega = \Omega_1 \setminus \Omega_2$ and $\Omega_2 \subset \Omega_1$,

$$\sum_{r \in \Omega} f(r) = \sum_{r \in \Omega_1} f(r) - \sum_{r \in \Omega_2} f(r).$$

Secondly, $\Omega_1 = \bigcup_{i=1}^m \overline{\mathcal{W}}_{\alpha_i}(p_{i-1}, p_i)$. By $p_i^x < p_{i+1}^x$ only neighboring line segments intersect. It holds that

$$\overline{\mathcal{W}}_{\alpha_i}(p_{i-1}, p_i) \cap \overline{\mathcal{W}}_{\alpha_{i+1}}(p_i, p_{i+1}) = \bigcup_{r \in [p_{i-1}, p_i]_{\alpha_i}} \mathcal{C}(r) \cap \bigcup_{s \in [p_i, p_{i+1}]_{\alpha_{i+1}}} \mathcal{C}(s) = \mathcal{C}(p_i).$$

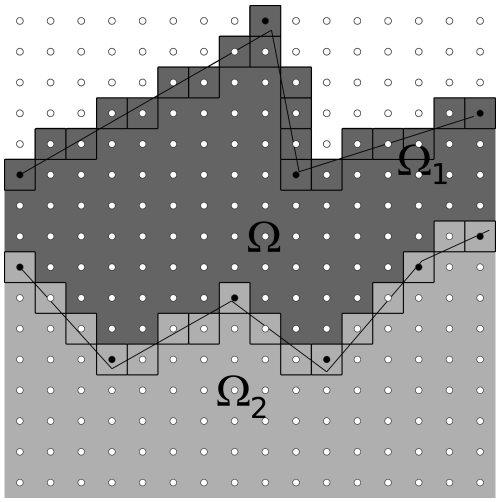

FIG. 2.8. *A simple polygonal domain.*

To see this observe that $[p_{i-1}, p_i]_{\alpha_i} \cap [p_i, p_{i+1}]_{\alpha_{i+1}} = \{p_i\}$ for all $1 \leq i < n$ and therefore $\mathcal{C}(p_i)$ is clearly contained in the intersection. If this inclusion were strict, then there would exist some $v \in \mathbb{Z}^2$ with $v^y > p_i^y$, $v^x = p_i^x$ with $v \in [p_{i-1}, p_i]_{\alpha_i} \cap [p_i, p_{i+1}]_{\alpha_{i+1}}$, a contradiction to $[p_{i-1}, p_i]_{\alpha_i} \cap [p_i, p_{i+1}]_{\alpha_{i+1}} = \{p_i\}$. Therefore

$$\sum_{r \in \Omega_1} f(r) = \sum_{i=1}^{m} \overline{W}_{\alpha_i}(p_{i-1}, p_i) - \sum_{i=1}^{m-1} C(p_i).$$

The equivalent result holds for the sum concerning Ω_2. □

2.3. Implementation. We now summarise the results of the previous section with respect to implementation. Assume that we work on greylevel images $f \in \mathbb{R}^S$ defined over rectangular domains $S = [1, \ldots, N_1] \times [1, \ldots, N_2]$. We are interested in the computation of sums $\sum_{i \in \Omega} f(i)$ over simple polygonal domain Ω. To utilise Theorem 2.12 for a fast computation, we require the set of angles of Ω to stem from a finite set $\Theta \subset [-\pi/4, 3\pi/4]$. The main required ingredients are

(i) computation and storage of the auxiliary matrices C and $\overline{W}_\vartheta$ for all angles $\vartheta \in \Theta$,
(ii) determination of vectors $(p_{i-1}, p_i, \vartheta_i)$, $1 \leq i \leq n$ defining the edges of the domain Ω,
(iii) use formula (2.18) for the computation of $\sum_{i \in \Omega} f(i)$.

According to Lemma (2.9) step (i) has complexity $O(|S| \cdot |\Theta|)$, both in running time and memory consumption. Execution of step (ii) is only required if the parameters $(p_{i-1}, p_i, \vartheta_i)$ $(1 \leq i \leq n)$ are not explicit. For example, in our applications of wedge divisions of a rectangle (cf. next section), we specify a wedge $\mathcal{W}$ by an intersection of a line $L_{n,\vartheta}$ with a rectangle R. To apply (iii) to the corresponding simple domain $\mathcal{W}$ the intersection points of R and $L_{n,\vartheta}$ on the boundary of R are required. According to Theorem 2.12, step (iii) requires no more than $2n$ additions, provided that the matrices from step (i) are known.

The following corollary summarises the findings of this section.

COROLLARY 2.13. *Let $f : [1, \ldots, N_1] \times [1, \ldots, N_2] \to \mathbb{R}$ be given. Let $\Theta \subset]-\pi/4, 3\pi/4]$ be finite. After a preprocessing step of complexity $O(|S| \cdot |\Theta|)$, both in*

time and memory, the sum

$$\sum_{i\in\Omega} f(i)$$

can be computed in $O(n)$, for any simple polygonal domain Ω of order n with angles in Θ.

We close this section by pointing out that our scheme is *exact*. Clearly, for large images and/or greyvalue ranges, overflow handling will have to be addressed, but the necessary length of the mantissa can easily be estimated.

3. Wedgelet approximations. In this section we apply the concepts described in Section 2 to the specific case of wedgelet approximation. We are concerned first with the case of locally constant functions over dyadic wedgelet partitions. These have been first proposed by Donoho [6], who derived approximation rates and showed asymptotic optimality of the wedgelet estimator, for functions of the horizon class and starlike domains. These functions are essentially characteristic functions of image subdomains with smooth boundary. It is also shown in [6] how the hierarchic structure of quadtrees allows fast computation of minimisers.

Let us fix some notations. A digital image f is a mapping from a finite rectangular domain $S \subset \mathbb{Z}^2$, into $\mathbb{R}$ and for each pixel $s \in S$, $f(s)$ denotes the associated greyscale value.

3.1. Dyadic square wedge partitions. In this subsection, we present the wedgelet partitions based on dyadic squares. For the sake of simplicity we restrict ourselves to discrete dyadic square domains of the form, $S = [1,\dots,N]\times[1,\dots,N]$ with $N = 2^p$.

We adopt the standard definition for a **partition** of the set S, as a set $\mathcal{P} = \{A_k\}_{k\in K}, A_k \subset S$, $\bigcup_{k\in K} A_k = S$ and $A_i \cap A_j = \emptyset$, if $i \neq j$. A subset $A_k \subset \mathcal{P}$ is called atom of the partition.

The set of dyadic square partitions $\mathfrak{Q}$ can be defined with the following rules.
(1) $\mathcal{P}_0 = \{S\} \in \mathfrak{Q}$.
(2) If $\mathcal{P} \in \mathfrak{Q}$, $A \in \mathcal{P}$ and $|A| > 1$ then $\mathcal{P} \setminus \{A\} \cup_i \{A_i\} \in \mathfrak{Q}$.
where $A_i, i = 1,\dots,4$ are formed by splitting A into four equally sized squares A_i. Since the image domain is a dyadic square, the atoms of a dyadic partition are also dyadic squares, of the form $[(k-1)2^i+1, k2^i]\times[(l-1)2^i+1, l2^i], i \in \{1,\dots,p\}, k,l \in \{1,\dots,2^{p-i}\}$. Let $\mathcal{S}$ be the set of dyadic squares. Then $\mathcal{Q}$ consists of all partitions $\mathcal{P}$ of S such that $\mathcal{P} \subset \mathcal{S}$.

Now we introduce our **discrete wedge splitting**. This discretisation scheme is a keystone for the development of efficient algorithms. The use of nonoverlapping lines allows an efficient implementation of discrete sums.

DEFINITION 3.1. *Let $A \in \mathcal{S}$ be a dyadic square and L^n_ϑ be a line with $(n,\vartheta) \in \mathbb{Z}^2\times]-\pi/4, 3\pi/4]$ such that $L^n_\vartheta \cap A \neq \emptyset$ and $L^{n+1}_\vartheta \cap A \neq \emptyset$. A wedge splitting of A is the partition into two* **wedge** *subdomains $\{W^1_{n,\vartheta}(A), W^2_{n,\vartheta}(A)\}$ (respectively the lower and upper wedges) defined by*

$$W^1_{n,\vartheta}(A) = \bigcup_{k\le n} L^k_\vartheta \cap A$$
$$W^2_{n,\vartheta}(A) = \bigcup_{k>n} L^k_\vartheta \cap A.$$

Note that a wedge is a convex discrete polygon, which is a special case of simple domains in the sense of 2.11, with at most 5 vertices. Indeed we can distinguish three

cases: the wedge has either 5 vertices, or is a trapeze, or is a triangle. Thus the computation of one moment requires 6 auxiliary function values in the two first cases, and 4 function values in the triangular case.

It is now possible to properly define a **dyadic wedge partition**.

DEFINITION 3.2. *A dyadic wedge partition $\mathcal{W}$ is a subdivision of a dyadic square partition $\{A_k\}_{k\in K}$ of the domain S, obtained by a discrete wedge splitting of some of its atoms A_k, replaced by a wedge split,*

$$\{W^1_{n_k,\vartheta_k}(A_k), W^2_{n_k,\vartheta_k}(A_k)\},$$

where $\vartheta_k \in]-\pi/4, 3\pi/4]$.

3.2. Wedgelet approximation. Like in [6], we are interested in piecewise constant approximations of an image f over a class of dyadic wedge partitions. We follow a variational approach, with a balancing of segmentation complexity versus approximation accuracy. Here, we measure the complexity by the number of atoms in the partition.

More precisely, let $\mathfrak{W}$ be a set of dyadic wedgelet partitions of S. For $\lambda \in \mathbb{R}$ we select a partition $\hat{\mathcal{W}}_\lambda(f)$ from $\mathfrak{W}$ which satisfies

$$\hat{\mathcal{W}}_\lambda \in \operatorname*{argmin}_{\mathcal{W}\in\mathfrak{W}} \{\|f - \hat{f}_{\mathcal{W}}\|_2^2 + \lambda|\mathcal{W}|\}, \tag{3.1}$$

where $\hat{f}_{\mathcal{W}} = \sum_{W\in\mathcal{W}} \overline{f}_W$, and

$$\overline{f}_W = \mathbb{1}_W \frac{1}{|W|} \sum_{i\in W} f(i)$$

denotes the function equal to the average of f over the atom W (which is either a pure wedge or a square), and 0 elsewhere.

Note that the pair $(\hat{\mathcal{W}}_\lambda, f_{\hat{\mathcal{W}}_\lambda})$ minimizes $(\mathcal{W}, g) \mapsto \|f - g\|_2^2 + \lambda|\mathcal{W}|$, where a pair $(\mathcal{W}, g)$ is such that g is piecewise constant over $\mathcal{W}$. It is a particular property of the chosen functional that the minimisation can be split up into a minimisation over the partition class and (local) minimisation on each segment. (Recall that $\overline{f}_W$ is the classical least square local approximation of $\|f - g\|_2^2$ over W.)

3.3. Algorithm. Let us now describe the algorithm for efficient minimisation of (3.1) for a given partition class $\mathfrak{W}$. As we manipulate huge sets $\mathfrak{W}$, finding the optimal partition among this set may lead to high computational costs. These costs are dramatically reduced by exploiting the recursive structure of the dyadic partitions. Nevertheless, still a huge number of sums ('moments') over polygonal domains have to be computed when minimizing over the class of dyadic wedge partitions. In order to handle this, we introduce an algorithm which employs the principles of Section 2 for efficient moment computation on the wedge domains.

We work now with a finite set $\Theta \subset]-\pi/4, 3\pi/4]$ of angles. This means that the condition $\vartheta \in]-\pi/4, 3\pi/4]$ is replaced by the condition $\vartheta \in \Theta$. In order to treat the horizontal lines, it is required that $0 \in \Theta$. Since dyadic partitions are obtained by successive division of squares into four squares, each dyadic partition $\mathcal{P}$ can uniquely be mapped to a quad tree T. Each node N of the tree T corresponds to a dyadic square $Q \subset S$ and the four child-nodes of N (if any) correspond to the four squares generated by a quad split of Q. Each terminal node of the tree T corresponds to a square in $\mathcal{P}$ and vice versa. Each quad tree is obtained as a pruning of the finest

quad tree whose leaves correspond to the pixels of S; in implementation the pruning is coded by marking nodes as terminal.

The minimisation of the functional consists of two steps: First, computation of the optimal wedge split for each dyadic square, i.e. for each node in the tree. Second, for given parameter $\lambda \geq 0$ the computation of the minimiser $\hat{\mathcal{W}}_\lambda$ by recursively pruning the quad tree, using the optimal wedge splits obtained in the first step. Here, it is important to note that only the second step depends on λ, and, as will be seen below, its complexity does not depend on the number of angles used in step one.

The heuristics for the second step can be explained as follows. Given $\lambda \geq 0$, there are three possible cases to consider for the optimal wedge partition $\hat{\mathcal{W}}_\lambda$.

1. $\hat{\mathcal{W}}_\lambda = \{S\}$
2. $\hat{\mathcal{W}}_\lambda = \{W_1, W_2\}$ for the optimal two wedges W_1, W_2 stored at the root node
3. $\hat{\mathcal{W}}_\lambda = \bigcup_{i=1\ldots4} \hat{\mathcal{W}}^i_\lambda$, where each $\hat{\mathcal{W}}^i_\lambda$ is the optimal wedge partition of the dyadic subsquare Q_i, and $Q_1, \ldots, Q_4$ are obtained by a quad split of S.

Hence a minimisation algorithm consist in comparing the scores induced by the three cases; in the first case the score is local approximation error plus λ, in the second case it is local approximation error plus 2λ, and in the third case it is the sum of the four scores for $\hat{\mathcal{W}}^i_\lambda$, $i = 1, \ldots, 4$. Observe that the third case requires a recursive application of the minimisation algorithm.

In more precise terms the full minimisation algorithm is described as follows:

1. Create empty tree. Initialise each node with approximation error $= \infty$.
2. Computation of optimal wedge splits:
   (a) Compute the vertical sum matrix C using 2.12 and $\overline{W}_0$ using 2.16
   (b) For each $\vartheta \in \Theta$
      i. compute the auxiliary matrix $\overline{W}_\vartheta$ using 2.16
      ii. for each dyadic square Q and each n compute the approximation error ε and associated mean values obtained by splitting Q along the line L^n_ϑ. If the induced approximation error is below the stored value at the corresponding node N, then store mean, error, angle ϑ and line number n at node N.
3. Computation of the minimiser $\hat{\mathcal{W}}_\lambda$, and the associated score, for instance using a straightforward recursive implementation of the above procedure. For a bottom-up version of the algorithm see [6]. In any case, the algorithm crucially relies on the availability of the optimal wedge splits of all dyadic squares, as provided by step 2.

4. Generalisations. In this section we show how the algorithm described in section 3 can be easily generalised to much larger classes of approximation models for which the computational complexity remains of the same order.

4.1. Wedge partitions with local regression models. The protoype model presented in section 3 was proposed first in a somewhat academic context: the goal was to get optimal approximations for some classes of piecewise constant functions with Hï£¡der regular boundaries. For natural images, however, piecewise constant models lead to undesired block artefacts, especially for the low resolutions, although these latter are often the target resolutions. Models based on richer local regression offer a first answer to this problem.

Instead of just considering optimal constant approximation on each atom of the dyadic wedge partition, we consider a finite set $\Phi = (\varphi_j)_{j \in R}$ of r linearly independent functions, $\varphi_j : S \to [0, 1]$. Defining the set Φ globally and not locally is motivated by the computation of the moments on many overlapping atoms of several partitions.

The functionalities are somewhat separated: the extraction of geometrical information is performed by the partitioning, whereas the local representation of the function is done by the regression on each atom (we recall that in our case, an atom consist merely of a wedge or of a square). There is no theoretical restriction for the choice of Φ. In practice however, discrete projections of some regular functions should be used. This requirement is connected to the the idea of piecewise smooth approximation.

If $\mathcal{W}$ is a dyadic wedge partition and $A \in \mathcal{W}$ an atom of this partition, we obtain the approximation $\hat{f}_{\Phi,\mathcal{W}}$ of the image as the sum of the projections of the local images f_A onto $\Phi_A = Span\varphi_{j,A}$. The minimisation problem reads now, for an image f and $\lambda \geq 0$,

$$\hat{\mathcal{W}}_{\lambda,\Phi}(f) = \underset{\mathcal{W}\in\mathfrak{W}}{\operatorname{argmin}}\{\|f - \hat{f}_{\Phi,\mathcal{W}}\|_2^2 + \lambda|\mathcal{W}|\}. \tag{4.1}$$

Since $\mathcal{W}$ is a partition, $\hat{f}_{\Phi,\mathcal{W}}$ can be written as the sum of the solutions of the local regression minimisation problems over each atom of $\mathcal{W}$,

$$\hat{f}_{\mathcal{W}} = \sum_{A\in\mathcal{W}} \hat{f}_A = \sum_{A\in\mathcal{W}} \sum_{j\in R} \hat{\alpha}_{A,j}\varphi_{A,j}.$$

For a given atom, this is a classical least square minimisation with some functions of the coordinates as regressing variables. The existence and uniqueness of the solution is then ensured, provided that the local family $(\varphi_{j,A})_{j\in R}$ is linearly independent. When it is not the case, the uniqueness is forced by choosing a subset $\Phi' \subset \Phi$, so that the family $(\varphi_A)_{\varphi\in\Phi'}$ is linearly independent.

Thus the local minimisation problem over A is performed by the solution of a linear system, which essentially requires the computation of the $\frac{r(r+1)}{2}$ local moments $< \varphi_i, \varphi_j >_A$, and of the r additional moments $< f, \varphi_i >_A$. Finally we also need the moment $< f, f >_A$ in order to get the corresponding error measure

$$E_A(f) = \|f_A - \hat{f}_A\|^2. \tag{4.2}$$

Solving the $r \times r$ linear system can be achieved in $O(r^3)$ flops with a classical solution method for symmetrical linear systems. In practice r should be small enough to maintain the advantage of very fast computation of the local regression parameters (typically smaller than 10), provided that the abovementioned moments are given.

Finally, for the step 2(b)ii of the algorithm described in subsection 3.3 it just remains to store the set of optimal parameters $(\alpha_j)_{j\in R}$ instead of the mean.

4.2. Locally adaptive regression models. The model described in the previous subsection allows rich classes of local approximating functions. It remains however limited by its global definition. Since a natural image can be viewed as highly non stationary signal, a further extension of our schemes consists in combining a finite number of local regression models in the same partition. In each atom an optimal model is selected according to its local relevance, i.e. it is the model for which the according reconstruction error is minimal. Formally, we consider a finite family $\mathcal{M}$ of regression models. If $\mathcal{W} \in \mathfrak{W}$ is a partition, for each atom $A \in \mathcal{W}$, let $\hat{\Phi}(A) \in \mathcal{M}$ denote an optimal local model, given by

$$\hat{f}_{\hat{\Phi}(A),A} = \min_{\Phi\in\mathcal{M}} \|f_A - \hat{f}_{\Phi,A}\|_2^2.$$

We can now define for any partition $\mathcal{W}$, the global approximation $\hat{f}_{W,\mathcal{M}}$, optimal according to the local models

$$\hat{f}_{\mathcal{W},\mathcal{M}} = \sum_{A\in\mathcal{W}} \hat{f}_{\hat{\Phi}(A),A}$$

In this case, the minimisation problem can be read

$$\hat{\mathcal{W}}_{\lambda,\mathcal{M}} = \operatorname*{argmin}_{\mathcal{W}\in\mathfrak{W}} \|f - \hat{f}_{\mathcal{W},\mathcal{M}}\|_2^2 + \lambda \sum_{A\in\mathcal{W}} |\hat{\Phi}(A)|, \tag{4.3}$$

In what regards the algorithm of subsection 3.3, it should only be remarked that in the step 2(b)ii, we now need to store for each dyadic square the index for the selected model in $\mathcal{M}$ and the error defined by

$$E_A = \min_{\Phi\in\mathcal{M}} E_{A,\Phi}.$$

Note that the step 3 is unchanged and hence the computational complexity of this step is independent on the size of the model. This ensures the real time access to every optimal solution to the locally adaptive regression problem, with varying λ, provided the optimal local models have been computed (step 2).

4.3. General penalisation functions. In functionals (3.1) or (4.1), the term $|\mathcal{W}|$ is merely the simplest example for penalising the complexity of the model. In fact a larger class of suited penalisation functions, describing the characteristics of our local approximation model can be considered to which our schemes can be applied. To fix the ideas, we treat the most general case of functional (4.3). We consider a function $g_{\mathcal{M}}$ defined for convenience on any subpartition of $\mathcal{W} \in \mathfrak{W}$. We only require that it is additive, which means that it satisfies

$$g_{\mathcal{M}}(\mathcal{W}) = g_{\mathcal{M}}(\mathcal{W}_1) + g_{\mathcal{M}}(\mathcal{W}_2), \text{ for any } (\mathcal{W}_1, \mathcal{W}_2) \text{ such that} \mathcal{W} = \mathcal{W}_1 \dot{\cup} \mathcal{W}_2.$$

An important example of such a function is the penalisation according to the coding length required for each partition according to some coding scheme (or in practical cases an estimate of this length). This might take into account not only the size of a partition, but also, for instance its depth, or some correlation structure. Such a penalisation function would enable rate-distortion optimisation, for the purpose of compression schemes of images based on partition regression representations.

5. Experiments and results. This section is devoted to some experimental results. They illustrate the efficiency of our scheme in terms of computational time, and shows the flexibility allowed for the choice of the underlying approximation assumptions. The implementation of the algorithms described in the previous sections has been made in Oberon programming language. Executable files of this implementation can be downloaded at [24].

5.1. Examples with locally adaptive models. In Section 4.2, we treated the case where the optimal coefficients are selected locally, as well as the model itself. With the notations of this section, we present here some specific instances for the set of possible models $\mathcal{M}$.

An elementary set of models is formed by the set of bivariate polynomials of degree less than d, represented by their canonical basis $(x^i y^j)_{i+j\leq d}$, and denoted with Φ_d. For d respectively equal to $0, 1, 2$, we obtain the so called *constant, linear*, and

FIG. 5.1. *Barbara,* 256×256. *Quadtree with no wedges. Left: Linear model, 14682 coefficients, PSNR: 25.96 dB. Right: Linear model, combined with a generic sine model, 14484 coefficients, PSNR: 28.97 dB.*

quadratic regression models, with respective sizes $1, 3$ and 6. Consider the quadtree-wedgelet scheme with the set of models Φ_0, Φ_1, Φ_2. These models are nested ($\Phi_0 \subset \Phi_1 \subset \Phi_2$)and thus the computation of the moments reduces to those necessary for Φ_2. If we use the additional multiplicative properties (for example $< x, y^2 >=< xy, y >$), we can even reduce to the computation of 22 moments instead of 28.

The previous models present a nice theoretical framework, leading to what could be called "discontinuous splines" over wedge partitioning. In practice however, they may suffer from numerical instabilities, especially for big images and for d bigger than 2. This drawback mainly comes from the fact that orthogonality is not conserved by local projection over atoms A, and locally, the angles between some functions can be too small for numerical purpose.

Other constructions, based on sinusoidal functions, present a much more stationary behaviour, while still adapting locally to the image contents. Such functions avoid some blocking artefacts occurring, for example, with the use of a mere linear model. Heuristically, the aim of such a model is to combine a model for texture by the locally adaptive models with a pure geometrical method (optimal wedges) to detect the significant contours between patterns in the image. Consider, for instance, the following set of parameterised models $\mathcal{M} = \Phi_{\vartheta_i, a_i} \cup \Phi_1$, where $\Phi_{\vartheta, a}$ is the local regression set composed of three functions defined by

$$\Phi_{\vartheta,a}(z) = (1, \cos(< z, v_\vartheta >), \sin(< z, v_\vartheta >)).$$

Remark that in order to keep coherence for non textured areas, we still keep additionally the mode Φ_1, also consisting of three functions.

We applied this model to the standard image Barbara, which presents many textured areas. We performed this method with the following set of parameters $(\vartheta, a) \in \{-\pi/4, 0, \pi/4, \pi/2\} \times \{1, 1/2, 1/4\}$. In Figures 5.1, 5.2 and 5.3, we compare, for a similar number of coefficients, an approximation provided by a simple linear model with an approximation provided by the set of models $\mathcal{M} = \Phi_{\vartheta_i, a_i}$. Although the comparison is not completely fair (for each piece an additional information

FIG. 5.2. *Barbara,* 256×256. *Model with* 8 *possible angles. Left: Linear model, 14730 coefficients, PSNR: 27.79 dB. Right: Linear model, combined with a generic sine model, 14511 coefficients, PSNR: 33.29 dB.*

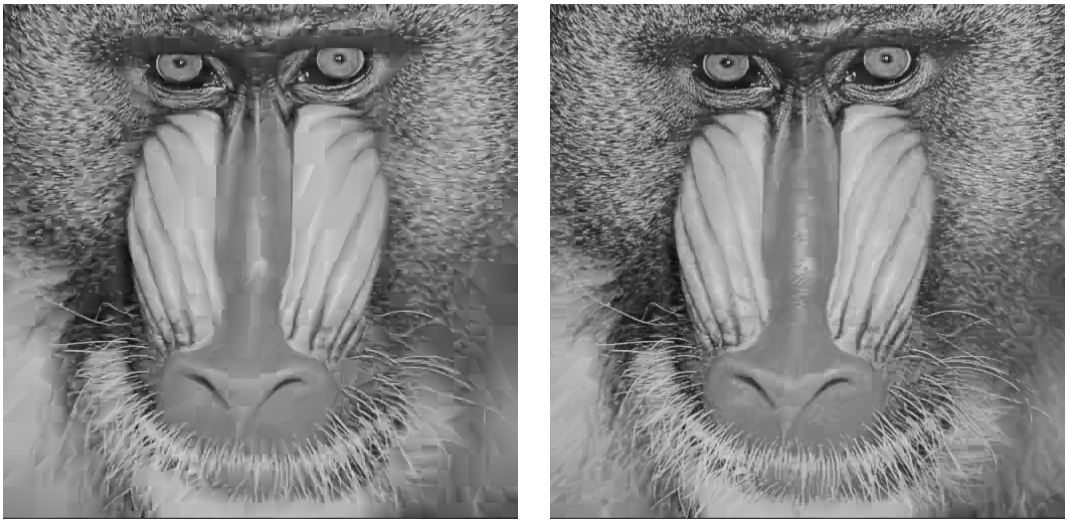

FIG. 5.3. *Baboon,* 512×512. *Model with* 8 *possible angles. Left: Linear model, 27150 coefficients, PSNR: 24.44 dB. Right: Linear model, combined with a generic sine model, 26574 coefficients, PSNR: 27.44 dB.*

indicating the model selected locally has to be added) this shows the adaptivity of the scheme: with a set of models chosen in a heuristic way, it is possible to improve dramatically the approximation, both in terms of PSNR, and in terms of visual quality (rendering of the details, precision of the textured patterns).

5.2. Runtime comparison. The very fast runtimes of our software can be shown by comparison with another library for computation of wedge segmentations, 'BeamLab'[25]. BeamLab is a package running on MatLab. It contains routines for constant wedgelet segmentations of images with exclusively dyadic dimensions. We

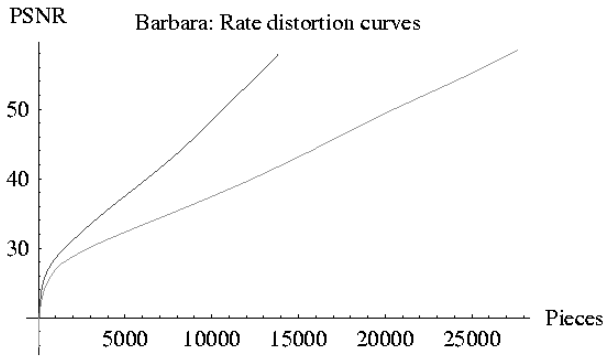

FIG. 5.4. *PSNR versus number of pieces needed for the image Barbara.*

used MatLab version 6 and BeamLab as loaded from the webpage on the 8.6.2004, see [25]. For the runtime comparison displayed in the following table, we used a 2.8 GHz Pentium IV machine. We append approximate runtimes for a wedgelet model in our approach with different number of angles. An adaptive model removes half of the angles when entering the next depth in the quad tree. The 'equivalent' model is an adaptive model with a number of 1024, 512, ..., 32 angles corresponding to image sizes 512×512, 256×256, ..., 16. This should be roughly equivalent with considering every wedgelet in the image.

Image size	16×16	32×32	64×64	128×128	256×256	512×512
BeamLab	$6.72s$	$45.34s$	$330.41s$ $\approx 5.5min$	$2676.3s$ $\approx 44.6min$	$27918s$ $\approx 7h45min$	$> 12h$
dyadic wedge equivalent	$< 0.02s$	$0.05s$	$0.2s$	$1.2s$	$8s$	$59s$
dyadic wedge 1024 angles	$1s$	$2s$	$8s$	$37s$	$140s$	$722s$
dyadic wedge 180 angles	$0.2s$	$0.3s$	$1.6s$	$6s$	$24s$	$120s$
dyadic wedge 4 angles	$< 0.01s$	$< 0.01s$	$0.04s$	$0.2s$	$1s$	$10s$
dyadic square	$< 0.01s$	$< 0.01s$	$< 0.01s$	$0.04s$	$0.5s$	$6s$

6. Conclusion. In this paper, we proposed a new efficient algorithm for rapid moment computation over polygonal domains. It provides a very efficient and flexible tool, enabling comparative experiments between many different bases on arbitrary adaptive local regression functions. A very useful property of our scheme is the possibility to compute optimal solutions for many different parameters λ very rapidly: in particular it enables computation of rate-distortion curves in very reasonable time. It also allows to easily perform tests with diverse sets of parameterised models.

Acknowledgements. We thank Piyanut Pongpiyapaiboon for help with the implementation. Felix Friedrich and Hartmut Fhr acknowledge additional funding through the European Research and Training Network HASSIP.

REFERENCES

[1] X.Y. Jiang, and H. Bunke. *Simple and Fast Computation of Moments,* Pattern Recognition **24**, 1991, no. 8, pp. 801–806.

[2] W. Philips. *A new fast algorithm for moment computation.* Pattern Recognition, **26**, 1993, no. 11, pp. 1619–1621.
[3] L. Yang and F. Albregtsen. *Fast and exact computation of cartesian geometric moments using discrete Green's theorem,* Pattern Recognition **29**, 1996, no. 7, pp. 1061–1073.
[4] H.Z. Shu, L.M. Luo, W.X. Yu and Y. Fu. *A new fast method for computing Legendre moments,* Pattern Recognition **33** (200), no. 2, pp. 341–348.
[5] J. Gu, H.Z. Shu, C. Toumoulin, L. M. Luo, *A novel algorithm for fast computation of Zernike moments*, Pattern Recognition **35**, 2002, No. 12, pp. 2905–2911.
[6] Donoho, David. *Wedgelets: nearly minimax estimation of edges,* Ann. Statist. 27, 1999, no. 3, pp. 859–897.
[7] R.M. Willett and R.D. Nowak. *Platelets: A multiscale approach for recovering edges and surfaces in photon-limited medical imaging,* IEEE Transactions on Medical Imaging **22**, 2003, pp. 332–350.
[8] G.Y. Tang. *A discrete version of Green's theorem,* IEEE Transactions on Pattern Analysis and Analytic Machine Intelligence **4**, 1982, pp. 242–249.
[9] J. Bresenham. *Algorithm for computer control of a digital plotter,* IBM Systems Journal **4**, 1965, pp. 25–30.
[10] J. Polzehl and V.G. Spokoiny. *Image Denoising: Pointwise adaptive approach,* Annals of Statistics, **31**, 2003, pp. 30–57.
[11] D. Donoho and X. Huo *Beamlets and multiscale image analysis*, Barth, Timothy J. et al.(ed.), *Multiscale and multiresolution methods. Theory and applications.* Springer Lect. Notes Comput. Sci. Eng. 20, 2002, pp. 149–196.
[12] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and regression trees.* The Wadsworth Statistics/Probability Series. Belmont, California, 1984.
[13] S. Geman and G. Geman. *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,* IEEE Trans. PAMI **12**, 1990, pp. 609–628.
[14] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods. A Mathematical Introduction.* Second Edition. Springer Verlag 2003.
[15] J. Polzehl and V.G. Spokoiny. *Adaptive weights smoothing with applications to image restoration,* J.R. Statist. Soc., Ser. B, **62**, pp. 335–354.
[16] J.-M. Morel and S. Solimini. *Variational Methods in Image Segmentation.* Birkhäuser Verlag, 1995.
[17] J. Weickert. *Anisotropic Diffusion in Image Processing.* Teubner Verlag, 1998.
[18] V. Aurich and J. Weule. *Non-linear Gaussian filters performing edge-preserving diffusion,* in *Proceedings of the 17th DAGM-Symposium, Bielefeld.* Springer Verlag, 1995, pp. 538–545.
[19] M. Lounsbery, T. DeRose and J. Warren. *Multiresolution Analysis for Surfaces of Arbitrary Topological Type,* ACM Transactions on Graphics, Vol.16,1, January 1997, pp. 34–73
[20] L. Demaret, N. Dyn, M.S. Floater and A. Iske. *Adaptive Thinning for Terrain Modelling and Image Compression,* in Advances in Multiresolution for Geometric Modelling, N.A. Dodgson, M.S.Floater, and M.A.Sabin (eds.), Springer Verlag, 2004, pp. 321–340.
[21] S. Mallat and W.-L. Hwang. *Singularity detection and processing with wavelets,* IEEE Trans. Inf. Th. **38**, 2, 1992, pp. 617–643.
[22] G. Nason and B.W. Silverman. *Wavelets for regression and other statistical problems,* Schimek, Michael G. (ed.), Smoothing and regression. Approaches, computation and application. Chichester: Wiley. Wiley Series in Probability and Statistics, 2000, pp. 159–191.
[23] R. Shukla. *Rate-Distortion Optimized Geometrical Image Processing,* Ph.D. Thesis, EPFL, Lausanne, 2004.
[24] Wedgelets website: http://www.wedgelets.de/
[25] Beamlab website: http://www-stat.stanford.edu/ beamlab/
[26] F. Friedrich. *Complexity Penalized Segmentations in 2D*, PhD-Thesis, Technical University of Munich, 2005
[27] H. F. Walker. *Implementation of the* GMRES *method using Householder transformations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 152–163.