Chapter 2
Smoothlets

Abstract In this chapter the family of functions, called smoothlets, was presented. A smoothlet is defined as a generalization of a wedgelet and a second order wedgelet. It is based on any curve beamlet, named as a curvilinear beamlet. Smoothlets, unlike the other adaptive functions, are continuous functions. Thanks to that they can adapt to edges of different blur. In more details, the smoothlet can adapt to location, scale, orientation, curvature and blur. Additionally, a sliding smoothlet was introduced. It is the smoothlet with location and size defined freely within an image. The Rate-Distortion dependency and the $M$-term approximation of smoothlets were also discussed.

Recent research in image processing is concentrated on finding efficient, sparse, representations of images. There has been defined plenty of methods that are used in image approximation. The nonadaptive methods (like ridgelets [1], curvelets [2], contourlets [3], shearlets [4], etc.), usually based on frames, are known to be fast and efficient. The overcompleteness of these methods is not a problem, since the best coefficients are only used in a representation. The adaptive methods (like wedgelets [5], beamlets [6], platelets [7], surflets [8], smoothlets [9], multiwedgelets [10], etc.), based on dictionaries, are known to be more efficient than the nonadaptive ones, since a dictionary can be defined more accurate than a frame. But, on the other hand, they are much slower due to the fact that the additional decision has to be made “how to chose the best functions” for image representation.

All adaptive methods based on dictionaries have been defined on discontinuous functions [5, 7, 8]. Only the well-defined edges could be therefore represented by such functions. In reality, an edge presented on an image can be of different level of blur. There are many reasons of that fact, for instance, it can be a motion blur, it can be caused by a scanning method inaccuracy or a light shadow falling into the scene. Some of the blurred edges are undesirable and should be sharpened in the preprocessing step, but some of them are correct and should be represented as blurred ones. To represent such blurred edges “as they are” smoothlets were defined [9]. Smoothlets are defined as continuous functions, which can adapt not only to location, scale, orientation and curvature, like second order wedgelets [11], but also to blur.
Let us note that such an approach led to the definition of a quite new model that can be used in image approximation [9]. So far, the horizon model has been considered for geometrical multiresolution adaptive image approximations. It is a simple black and white model with smooth horizon discriminating two constant areas. Smoothlets are defined to give optimal approximations of a blurred horizon model. In this model, a linear transition between two constant areas is assumed, in other words, it is a blurred version of the horizon model. Because it is a generalization of the commonly used approach, it enhances the possibilities of the approximation theory.

2.1 Preliminaries

Consider an image \( F : D \rightarrow C \) where \( D = [0, 1] \times [0, 1] \) and \( C \subset \mathbb{N} \). In practical applications \( C = \{0, \ldots, 255\} \) for grayscale images and \( C = \{0, 1\} \) for binary images. Domain \( D \) can be discretized on different levels of multiresolution. It means that one obtains \( 2^j \cdot 2^j \) elements of size \( 2^{-j} \times 2^{-j} \) for \( j \in \{0, \ldots, J\} \), \( J \in \mathbb{N} \). Let us assume that \( N = 2^J \). In that way one can consider an image of size \( N \times N \) pixels in a natural way.

Let us define subdomain

\[
D_{i_1, i_2, j} = \left[ \frac{i_1}{2^j}, \frac{i_1 + 1}{2^j} \right] \times \left[ \frac{i_2}{2^j}, \frac{i_2 + 1}{2^j} \right]
\]

for \( i_1, i_2 \in \{0, \ldots, 2^j - 1\}, j \in \{0, \ldots, J\}, J \in \mathbb{N} \). To simplify the considerations the renumerated subscripts \( i, j \) are used instead of \( i_1, i_2, j \) where \( i = i_1 + i_2 2^j \), \( i \in \{0, \ldots, 4^j - 1\} \). Subdomain \( D_{i, j} \) is thus parametrized by location \( i \) and scale \( j \). Let us note that \( D_{0,0} \) denotes the whole domain \( D \) and \( D_{i,j} \) for \( i \in \{0, \ldots, 4^j - 1\} \) denote pixels from an \( N \times N \) image.

Let us define next, a horizon as a smooth function \( h : [0, 1] \rightarrow [0, 1] \) and let us assume that \( h \in C^\alpha, \alpha > 0 \). Further, consider the characteristic function \( H : D \rightarrow \{0, 1\} \),

\[
H(x, y) = \begin{cases} 
1, & \text{for } y \leq h(x), \\
0, & \text{for } y > h(x),
\end{cases} \quad x, y \in [0, 1].
\]

Then, function \( H \) is called a horizon function if \( h \) is a horizon. Function \( H \) models the black and white image with a horizon. Let us define then a blurred horizon function as the horizon function \( H_B : D \rightarrow [0, 1] \) with a linear smooth transition between black and white areas, more precisely, between \( h \) and its translation \( h_r, h_r(x) = h(x) + r, r \in [0, 1] \). Examples of a horizon function and a blurred horizon function are presented in Fig. 2.1. In this book a blurred horizon function is considered, unlike in the literature, where a horizon function is used. Let us note, however, that the latter function is a special case of the former one. So, in this book, a wider class of functions than in the literature is taken into consideration.
2.1 Preliminaries

Fig. 2.1  
(a) A horizon function,  
(b) a blurred horizon function

Fig. 2.2  
Sample subdomains with denoted a beamlets, b curvilinear beamlets

Consider a subdomain $D_{i,j}$ for any $i \in \{0, \ldots, 4^j - 1\}$, $j \in \{0, \ldots, J\}$, $J \in \mathbb{N}$. A line segment $b_{i,j,p}$, $p \in \mathbb{R}^2$, connecting two different borders of the subdomain is called a beamlet [5]. A curvilinear segment $b_{i,j,p}$, $p \in \mathbb{R}^n$, $n \in \mathbb{N}$, connecting two borders of the subdomain is called a curvilinear beamlet [9]. In Fig. 2.2, sample subdomains with denoted sample beamlets and curvilinear beamlets are presented.

Consider an image of size $N \times N$ pixels. The set of curvilinear beamlets can be parametrized by location, scale, and curvature. So, the dictionary of curvilinear beamlets is defined as [9]

$$B = \{b_{i,j,p} : i \in \{0, \ldots, 4^j - 1\}, j \in \{0, \ldots, \log_2 N\}, p \in \mathbb{R}^n, n \in \mathbb{N}\}. \quad (2.3)$$

The most commonly used curvilinear beamlets are paraboloidal or elliptical ones. They are usually parametrized by $p = (\theta, t, d)$, where $\theta$, $t$ are the polar coordinates of the straight segment connecting the two ends of the curvilinear beamlet and $d$ is the distance between the segment’s center and the curvilinear beamlet. Let us note that, by setting $d = 0$, one obtains linear beamlets, which are parametrized by $p = (\theta, t)$. Any other classes of functions and any other parametrizations are also possible, depending on the applications.
2.2 Image Approximation by Curvilinear Beamlets

Curvilinear beamlets can be used in binary image approximation [12]. In such a case the image must consist of edges, any kind of an image with contours is allowed. The algorithm of image approximation consists of two steps.

In the first step, for each square segment $D_{i,j}, i \in \{0, \ldots, 4^j-1\}, j \in \{0, \ldots, J\}, J \in \mathbb{N}$, of the quadtree partition, the curvilinear beamlet that best approximates image $F : D_{i,j} \to \{0, 1\}$ has to be found. In the case of binary images with edges the error metric that measures the accurateness of edge approximation by a curvilinear beamlet has to be applied. The most convenient metric is the Closest Distance Metric [13], which is used in this book in the simplest form

$$CDM_0(F, F_B) = \frac{|F \cap F_B|}{|F \cup F_B|},$$

(2.4)

where $F$ denotes the original image and $F_B$ is the curvilinear image representation. $CDM_0$ measures the quotient between the number of properly detected pixels $(F \cap F_B)$ and the number of all pixels belonging either to the edge or to the curvilinear beamlet $(F \cup F_B)$. The measure is normalized and for identical images is equal to 1, whereas for quite different images it is equal to 0.

In the second step of the image approximation algorithm, a tree pruning has to be applied. The best choice is the bottom-up tree pruning algorithm due to the fact that the approximation given by that algorithm is optimal in the Rate-Distortion (R-D) sense [5] (see Appendix B for detailed explanation). Indeed, the algorithm minimizes the following R-D problem

$$R_\lambda = \min_{P \in \mathcal{P}} \{1 - CDM(F, F_B) + \lambda^2 K\},$$

(2.5)

where the minimum is taken within all possible image partitions $P$ from the quadtree partition $\mathcal{Q}P$, $K$ denotes the number of bits needed to code curvilinear beamlets and $\lambda$ is the penalization factor. In the case of the exact image representation $\lambda = 0$. In general, the larger the value of $\lambda$, the lesser the accurateness of approximation. Sample image representations by curvilinear beamlets for different values of $\lambda$ are presented in Fig. 2.3.

2.3 Smoothlet Definition

Consider a smooth function $b : [0, 1] \to [0, 1]$. The translation of $b$ is defined as $b_r(x) = b(x) + r$, for $r, x \in [0, 1]$. Given these two functions, an extruded surface can be defined, represented by the following function
2.3 Smoothlet Definition

Fig. 2.3 Image approximation by curvilinear beamlets: a image consists of 392 curvilinear beamlets, b image consists of 241 curvilinear beamlets

\[
E_{(b,r)}(x, y) = \frac{1}{r} b_r(x) - \frac{1}{r} y, \quad x, y \in [0, 1], \ r \in (0, 1].
\]

(2.6)

In other words, this function represents the surface that is obtained as the trace created by translating function \( b \) in \( \mathbb{R}^3 \). It is obvious that equation (2.6) can be rewritten in the following way:

\[
r \cdot E_{(b,r)}(x, y) = b_r(x) - y, \quad x, y \in [0, 1], \ r \in [0, 1].
\]

(2.7)

Let us note that for \( r = 0 \) one obtains \( b_r = b \) and \( y = b(x) \). In that case the extruded surface is degenerate, this is function \( b \), and is called a degenerated extruded surface [9].

Having extruded surface \( E_{(b,r)} \), let us define a smoothlet as [9]

\[
S_{(b,r)}(x, y) = \begin{cases} 
1, & \text{for } y \leq b(x), \\
E_{(b,r)}(x, y), & \text{for } b(x) < y \leq b_r(x), \\
0, & \text{for } y > b_r(x),
\end{cases}
\]

(2.8)

for \( x, y, r \in [0, 1] \). Sample smoothlets for different functions \( b \) and different values of \( r \), together with their projections on \( \mathbb{R}^2 \), are presented in Fig. 2.4.

Let us note that some special cases of smoothlets are well-known functions. Let us examine some of them [9].

Example 2.1. Assume that \( r = 0 \) and \( b \) is a linear function. One then obtains

\[
S_{(b,r)}(x, y) = \begin{cases} 
1, & \text{for } y \leq b(x), \\
0, & \text{for } y > b(x),
\end{cases}
\]

(2.9)

for \( x, y \in [0, 1] \). This is the well-known function called wedgelet [5].
Fig. 2.4 Smoothlet examples (a)–(c) and their projections (d)–(f), respectively, gray areas denote linear part; a \( y = 0.75x^2 - x + 0.6, \ r = 0.4 \), b \( y = 0.2 \sin(12x) + 0.5, \ r = 0.2 \), c \( y = -0.8x + 0.7, \ r = 0.1 \)

**Example 2.2.** Assume that \( r = 0 \) and \( b \) is a segment of a parabola, ellipse or hyperbola. One then obtains \( S_{(b,r)}(x, y) \) given by (2.9). This is the function called *second order wedgelet* [11].

**Example 2.3.** Assume that \( r = 0 \) and \( b \) is a segment of a polynomial. One then obtains \( S_{(b,r)}(x, y) \) given by (2.9). This is the function called two-dimensional *surflet* [8].

**Example 2.4.** Assume that \( r > 0 \), \( b_r \) is a linear function and \( b \) is fixed accordingly. One then obtains

\[
S_{(b,r)}(x, y) = \begin{cases} 
E_{(b,r)}(x, y), & \text{for } y \leq b_r(x), \\
0, & \text{for } y > b_r(x),
\end{cases}
\tag{2.10}
\]

for \( x, y, r \in [0, 1] \). In this way one obtains the special case of a *platelet* [7]. In fact, in the definition of the platelet any linear surface is possible instead of \( E_{(b,r)} \).

Consider a subdomain \( D_{i,j} \) for any \( i \in \{0, \ldots, 4^j - 1\}, \ j \in \{0, \ldots, J\}, \ J \in \mathbb{N} \). Let us denote \( S_{i,j,b,r} \) as the smoothlet \( S_{(b,r)} \) defined on that subdomain. Consider then an image of size \( N \times N \) pixels. In order to use smoothlets in image representation a dictionary of them has to be defined. Let us note that a smoothlet is parametrized
by location, scale, curvature and blur (in practical applications the discrete values of blur $r$ are used). So, the dictionary of smoothlets is defined as

$$S = \{S_{i,j,b,r} : i \in \{0, \ldots, 4^j - 1\}, j \in \{0, \ldots, \log_2 N\}, b \in B, r \in [0, 1]\}. \quad (2.11)$$

### 2.4 Image Approximation by Smoothlets

Smoothlets are used in image approximation by applying the following grayscale version of a smoothlet [9]

$$S^{(u,v)}_{(b,r)}(x, y) = \begin{cases} 
    u, & \text{for } y \leq b(x), \\
    E^{(u,v)}_{(b,r)}(x, y), & \text{for } b(x) < y \leq b_r(x), \\
    v, & \text{for } y > b_r(x),
\end{cases} \quad (2.12)$$

for $x, y, r \in [0, 1]$, where

$$E^{(u,v)}_{(b,r)}(x, y) = (u - v) \cdot E_{(b,r)}(x, y) + v. \quad (2.13)$$

In the case of grayscale images $u, v \in \{0, \ldots, 255\}$. Let us note that the grayscale version of the smoothlet is obtained as $S^{(u,v)}_{(b,r)} = (u - v) \cdot S_{(b,r)} + v$.

Image approximation by smoothlets consists of two steps [9]. In the first one, the full smoothlet decomposition of an image with the help of the smoothlet dictionary is performed. This means that for each square $D_{i,j}, i \in \{0, \ldots, 4^j - 1\}, j \in \{0, \ldots, J\}$, the best approximation in the MSE sense by a smoothlet is found. After the full decomposition, on all levels, the smoothlets’ coefficients are stored in the nodes of a quadtree. Then, in the second step, the bottom-up tree pruning algorithm [5] is applied to get a possibly minimal number of atoms in the approximation, ensuring the best image quality (see Appendix B for detailed explanation). Indeed, the following Lagrangian cost function is minimized:

$$R_\lambda = \min_{P \in \mathcal{QP}} \{ ||F - F_S||_2^2 + \lambda^2 K \}, \quad (2.14)$$

where $P$ is a homogenous quadtree partition of the image (elements of which are stored in the quadtree from the first step), $F$ denotes the original image, $F_S$ denotes its smoothlet representation, $K$ is the number of smoothlets used in the image representation or the number of bits used to code it, depending on the application, and $\lambda$ is the distortion rate parameter known as the Lagrangian multiplier. In the case of exact image approximation, the quality is determined and the reconstructed image is exactly like the original one. Two examples of image representation by smoothlets are presented in Fig. 2.5 with the use of different values of parameter $\lambda$. 
2.5 Sliding Smoothlets

All geometrical multiresolution adaptive methods that are based on dictionaries defined so far are related to a quadtree partition. The appropriate transform can be therefore fast and is multiresolution. But it is not shift invariant. So, it cannot be used, for instance, in object recognition because any shift of the object leads to a quite different set of coefficients. To overcome that problem, a notion of a sliding wedgelet was introduced [14]. In this section a sliding smoothlet is described, which is defined in a similar way.

A sliding smoothlet is the smoothlet with location and size fixed freely within an image. So, it is not stored in any quadtree. It rather cannot thus be used in image approximation but gives good results in edge detection. In this situation, the smoothlet transform-based algorithm can be not efficient enough because the positions of smoothlets are determined by the quadtree partition. In fact, some edges can be better approximated by smoothlets lying freely within the image domain. Such an example is presented in Fig. 2.6. As one can see, the appropriately fixed location of the smoothlet caused that the edge is more likely than the one from the quadtree partition.

2.6 Smoothlets Sparsity

In general, images obtained from different image capture devices are correlated. It means that they are represented by many coefficients, which are rather large. Geometrical multiresolution methods lead to, usually overcomplete, sparse representations. Sparse representation of an image means that the main image content (in other words, geometry of an image) is represented by a few nonzero coefficients. The rest of them represent image details. They are, usually, sufficiently small to be neglected without a noticeable quality degradation.
2.6 Smoothlets Sparsity

Fig. 2.6  a The edge detected by the smoothlet from the quadtree partition, location = (192, 64), size = 64; b the edge detected by the sliding smoothlet, location = (172, 40), size = 64

Fig. 2.7 An example of approximation of blurred horizon function by smoothlets

Sparsity is expressed by the $M$-term approximation. It is a number of significant, large in magnitude, coefficients for a given image representation. From an efficient image coding point of view another measure is commonly used—the R-D dependency. It is used to relate the minimal number of bits, denoted as rate $R$, used to code a given image with a distortion not exceeding $D$, to the distortion $D$. In this section, both these measures are applied to smoothlets’ sparsity evaluation.

Consider an image domain $D = [0, 1] \times [0, 1]$. Consider then a blurred horizon function defined on $D$. It can be approximated by a number of smoothlets on a given level of multiresolution, as presented in Fig. 2.7. In more details, an edge presented in that image can be approximated by nearly $2^j$ elements of size $2^{-j} \times 2^{-j}$, $j \in \{0, \ldots, J\}$. In this section, the use of smoothlets based on second-order beamlets is assumed, because they were used in all practical applications throughout this book. The R-D dependency of smoothlet approximation can be computed as follows.

**Rate**

In order to code a smoothlet the following number of bits is needed [9] (see Section 5.2.1 for more details on image coding by smoothlets):
• 2 bits for a node type coding and
• the following number of bits for smoothlet parameters coding:
  – 8 bits for degenerate smoothlet or
  – \((2^j + 3) + 16 + 1\) bits for smoothlet with \(d = 0\) and \(r = 0\) or
  – \((2^j + 3) + 16 + j + 1\) bits for smoothlet with \(d > 0\) and \(r = 0\) or
  – \((2^j + 3) + 16 + j\) bits for smoothlet with \(d = 0\) and \(r > 0\) or
  – \((2^j + 3) + 16 + j + j\) bits for smoothlet with \(d > 0\) and \(r > 0\).

The number \(R\) of bits needed to code a blurred horizon function at scale \(j\) is therefore evaluated as follows:

\[
R \leq 2^j \cdot 2 + 2^j((2^j + 3) + 16 + 2j) \leq k_R 2^j j, \quad k_R \in \mathbb{R}. \tag{2.15}
\]

**Distortion**

Consider a square of size \(2^{-j} \times 2^{-j}\) containing an edge. Let us assume that this edge is a \(C^\alpha\) function for \(\alpha > 0\). From the mean value theorem, it follows that the edge is totally included between two linear beamlets with distance \(2^{-2j}\) (see Fig. 2.8a) [5]. Similarly, the edge is totally included between two second order beamlets with distance \(2^{-3j}\) (see Fig. 2.8b) [9]. So, the approximation distortion of edge \(h\) by second order beamlet \(b\) is evaluated as

\[
\int_0^{2^{-j}} (b(x) - h(x))dx \leq k_1 2^{-j} 2^{-3j}, \quad k_1 \in \mathbb{R}. \tag{2.16}
\]

Consider then a blurred horizon function \(H_B\). The approximation distortion of this function by smoothlet \(S_{(b, r)}\) is computed as follows [9]:

\[
\int_0^{2^{-j}} \int_0^{2^{-j}} (S_{(b, r)}(x, y) - H_B(x, y))dydx = I_1 + I_2 + I_3, \tag{2.17}
\]

where

\[
I_1 = \int_0^{2^{-j}} \int_0^{b(x)} (S_{(b, r)}(x, y) - H_B(x, y))dydx, \tag{2.18}
\]

\[
I_2 = \int_0^{2^{-j}} \int_{b(x)}^{b_r(x)} (S_{(b, r)}(x, y) - H_B(x, y))dydx, \tag{2.19}
\]

\[
I_3 = \int_0^{2^{-j}} \int_{b_r(x)}^{2^{-j}} (S_{(b, r)}(x, y) - H_B(x, y))dydx. \tag{2.20}
\]
From the definition of functions $S_{(b,r)}$ and $H_B$, evaluation (2.16), and the direct computations, one obtains that

$$I_1 \leq 2^{-3j}, \quad I_2 \leq 2^{-j}2^{-3j}, \quad I_3 \leq 2^{-3j}.$$  \hspace{1cm} (2.21)

Then, the distortion of approximation of blurred horizon function by a smoothlet is evaluated as follows [9]:

$$\int_0^{2^{-j}} \int_0^{2^{-j}} (S_{(b,r)}(x, y) - H_B(x, y)) dy dx \leq k_2 2^{-j}2^{-3j}, \quad k_2 \in \mathbb{R}. \hspace{1cm} (2.22)$$

Let us take into account the whole blurred edge defined on $[0, 1] \times [0, 1]$, approximated by nearly $2^j$ smoothlets. One then obtains that the overall distortion $D$ on level $j$ is

$$D \leq k_D 2^{-3j}, \quad k_D \in \mathbb{R}. \hspace{1cm} (2.23)$$

**Rate-Distortion**

To compute the R-D dependency for smoothlets, let us summarize that the parameters $R$ and $D$ were evaluated by (2.15) and (2.23), respectively. So, let us recall that

$$R \sim 2^j j, \quad D \sim 2^{-3j}. \hspace{1cm} (2.24)$$

Then, let us compute $j$ from $R$ and substitute it in $D$. In that way one obtains the following R-D dependency for smoothlet coding:

$$D(R) = k_S \frac{\log R}{R^3}, \quad k_S \in \mathbb{R}. \hspace{1cm} (2.25)$$
For comparison purposes, let us recall that for wavelets \( D(R) = k_V \frac{\log R}{R} \), \( k_V \in \mathbb{R} \) [15] and for wedgelets \( D(R) = k_W \frac{\log R}{R^2} \), \( k_W \in \mathbb{R} \) [5]. However, let us note that the R-D dependencies for wavelets and wedgelets were evaluated for the horizon model. In the case of the blurred horizon model they can be even worse, especially in the case of wedgelets, which cannot cope with this model efficiently (see Fig. 1.3d).

\[ M \text{-term approximation} \]

The \( M \)-term approximation is used in the case in that there is no need to code an image efficiently (e.g., image denoising). From the above considerations, it follows that each of \( 2^j \) elements of size \( 2^{-j} \times 2^{-j} \) generates distortion \( k_D 2^{-j} 2^{-3j} \). So, a blurred horizon function, consisting of \( M \sim 2^j \) elements, generates distortion \( D \sim 2^{-3j} \). Therefore, \( D \sim M^{-3} \).

References

Geometrical Multiresolution Adaptive Transforms
Theory and Applications
Lisowska, A.
2014, XII, 107 p. 65 illus., 21 illus. in color., Hardcover
ISBN: 978-3-319-05010-2