SHAH: SHape-Adaptive Haar wavelets for image processing

Piotr Fryzlewicz * Catherine Timmermans †

May 1, 2015

Abstract

We propose the SHAH (SHape-Adaptive Haar) transform for images, which results in an orthonormal, adaptive decomposition of the image into Haar-wavelet-like components, arranged hierarchically according to decreasing importance, whose shapes reflect the features present in the image. The decomposition is as sparse as it can be for piecewise-constant images. It is performed via an stepwise bottom-up algorithm with quadratic computational complexity; however, nearly-linear variants also exist. SHAH is rapidly invertible.

We show how to use SHAH for image denoising. Having performed the SHAH transform, the coefficients are hard- or soft-thresholded, and the inverse transform taken. The SHAH image denoising algorithm compares favourably to the state of the art for piecewise-constant images. A clear asset of the methodology is its very general scope: it can be used with any images or more generally with any data that can be represented as graphs or networks.

Keywords: Adaptive transformations, greedy algorithms, multiscale, sparsity, statistical learning.

1 Introduction

The contribution of this work is twofold: firstly, we introduce a new transform for images, based on new SHape-Adaptive Haar (SHAH) wavelets from which it takes its name, and secondly, we propose a methodology for image denoising based on the SHAH transform.

*Department of Statistics, London School of Economics, Houghton Street, London WC2A 2AE, UK. e-mail: p.fryzlewicz@lse.ac.uk. Work supported by the Engineering and Physical Sciences Research Council grant no. EP/L014246/1.

†Institute of Statistics, Biostatistics and Actuarial Sciences, Université catholique de Louvain, voie du Roman Pays 20, BE-1348 Louvain-la-Neuve, Belgium. e-mail: catherine.timmermans@uclouvain.be. Financial support from the IAP research network grant P06/03 of the Belgian Government (Belgian Science Policy) is gratefully acknowledged.

The SHAH transform of an image results in its orthonormal decomposition into a ranked collection of weighted level differences between pairs of zones in the image, the "most informative" such contrasts being ranked first. It thus provides a natural decomposition of the image into a set of features ordered according to their importance for the image description. The transform identifies the edges and other prominent features of the image, and the decomposition is as sparse as it can be for piecewise constant images. The SHAH transform is performed via an stepwise bottom-up algorithm with quadratic linear complexity, but nearly linear variants also exist. It might be viewed as the selection of a particular image-driven orthonormal basis (hence the term "shape-adaptive") and the projection of the image onto the selected basis. Due to its shape-adaptivity, the transform bypasses the classical notion of dyadic wavelet scales. It can be viewed as a two-dimensional extension of the Unbalanced Haar wavelet transform of a curve (Fryzlewicz, 2007).

The SHAH transform produces sparse representations of images, especially (nearly-)piecewise-constant ones, and hence can be used in conjunction with soft- or hard-thresholding operations with the purpose of removing noise from the input image. This results in a "highly nonlinear" operation on the image, being a superposition of two nonlinear operations: SHAH and thresholding. The resulting image denoising technique is shown to perform well, in particular for piecewise-constant images. Its performance can be improved further via linear averaging.

Although this paper focuses on image analysis, it is worth emphasizing that the methodology we propose applies to more general data structures. Indeed, the SHAH transform can be applied to any data that can be encoded as a graph whose nodes are associated with a given intensity and are embedded in a normed, not necessarily two-dimensional, space.

Software implementing SHAH is available from `http://stats.lse.ac.uk/fryzlewicz/shah/shah_code.R`.

1.1 Related work

This section aims to situate our work amongst the variety of available methods.

Multiscale image representation. The SHAH transform falls into the category of "multiscale representation of images". (Nonadaptively selected) wavelet bases are a canonical example of a tool used to achieve such representations, and a survey of their use in image processing can be found in Mallat (2009b). Wavelets, although widely used and relatively well understood, suffer from inefficacies

in capturing non-horizontal or non-vertical features in an image; curvelets (Candès and Donoho, 2001) attempt to remedy this by using a more flexible family of building blocks, which are also not selected adaptively.

Adaptive image representation and processing. In contrast to wavelets or curvelets, the building blocks of the SHAH transform are selected adaptively from the data. A review of adaptive image representations can be found in Peyré (2011). The principle of adaptivity (although not the particular construction used in SHAH) is shared by a number of "-let" transforms, including bandlets of Le Pennec and Mallat (2005) (see also Mallat and Peyré 2008 for a review of related techniques), wedgelets (Donoho, 1999; Claypoole and Baraniuk, 2000), tetrolets (Krommweh, 2010), the Easy Path Wavelet Transform (Plonka, 2009), edge-adapted nonlinear multiresolution techniques (Arandiga et al., 2008) and directed trees (Narendra and Goldberg, 1980). Heijmans and Goutsias (2000) provide, through morphological wavelets, a framework for describing nonlinear lifting-based wavelets decompositions. Grouplets (Mallat, 2009a) preserve the classical notion of scale and grid subdivision present in the Haar or lifting transforms (see below for references to lifting), but equip the standard Haar transform with an "association field" that groups together points that are not necessarily neighbours. This leads, in a context different from that in SHAH, to similar Haar-like filtering operations with weights not necessarily equal to those in SHAH. We emphasise that in contrast to grouplets, SHAH does not follow the dyadic scale structure of the classical wavelet transform. Other approaches to image processing (in this case, denoising) which can be viewed as adaptive but do not use the notion of decomposition or hierarchy are, for example, adaptive weight smoothing (Polzehl and Spokoiny, 2000) and penalized regression on a graph (Kovac and Smith, 2011). A recent review of image denoising techniques can be found in Milanfar (2013).

Wavelet-like methods on graphs outside of the image context. Hammond et al. (2009) and Antoine et al. (2010) define wavelets on graphs by studying eigenvalues of the graph Laplacian; the latter takes the form of a matrix encoding the connectivity of each node and edge. Coifman and Maggioni (2006) use the powers of a diffusion operator as the scaling tool leading to multiscale analysis. Several variants of their ideas (Szlam et al., 2005; Maggioni et al., 2005) lead to different wavelet constructions. Crovella and Kolaczyk (2003) uses the n-hop distance (the minimal number of edges one has to travel to go from the central node to another) to define wavelets on the graph.

Jansen et al. (2009) use the lifting algorithm akin to that of Sweldens (1996) to construct wavelets on graphs using a bottom-up approach where wavelets between the nearest nodes get constructed first. Some authors also have defined wavelet transforms specifically designed for the dendrogram: Murtagh (2007) uses Haar bases, while Gavish et al. (2010) generalizes to unbalanced Haar. Singh et al. (2010) iteratively reduces the graph by replacing two (groups of) nodes by a single one, but unlike in SHAH, the graph structure is not used in the reduction process. The latter method is closely related to the idea behind treelets (Lee et al., 2008), defined for unordered data. We end by mentioning that SHAH can be viewed as a contiguity-constrained agglomerative clustering technique, a broad class of methods described generically in Chapter 5 of Murtagh (1985).

Relationship to Swelden's lifting transform. "Lifting" (Sweldens, 1996) is a device for designing iterative data transformations whereby (transformed) data points get "predicted" using neighbouring values and, once the prediction error has been recorded, the predicted coefficient is removed from the system to reduce its complexity. It is a non-adaptive transformation in the sense that its form does not depend on the values of the data being processed, and it is a linear transformation of the data. In its original version cited above, each iterative stage involves predicting and removing half of all available coefficients. Versions for data on more complex domains also exist, for example the "lifting one coefficient at a time" scheme of Jansen et al. (2009), which is also non-adaptive and linear.

In contrast to these, SHAH, which also uses the notion of predicting data points or their clustered regions using neighbouring values, and then successively removing them (either "one coefficient at a time", or "a small subset of coefficients at a time"), is an *adaptive* and *non-linear* transform of the data. The adaptivity and non-linearity arise as a result of SHAH choosing, in a data-dependent way, which part of the data to operate on in each stage of the transform.

To give but one example of the consequencies of these properties, we remark that image denoising via SHAH, described later in Section 3 is an operation which belongs to the class of methods described by DeVore (1998) as "highly non-linear", since it involves a non-linear operation (thresholding) performed on an adaptively (and hence non-linearly) chosen basis. This is part of the reason why linear averaging of SHAH image reconstructions can bring improvements in their quality, as described in that section.

Finally, in contrast to classical lifting, the SHAH transform is conditionally orthonormal, by which we mean "orthonormal given the selected basis". This property is important, amongst others, in the

application of SHAH to image denoising where it leads to a fast algorithm for threshold selection, and in fast computation of the inverse SHAH transform.

1.2 Organization of the paper

The paper is organised as follows. Section 2 defines the SHAH algorithm and describes some of its properties. Section 3 shows how to apply SHAH to image denoising. Section 4 concludes.

2 The SHape-Adaptive Haar transform for images

2.1 Core ideas

The SHape-Adaptive Haar (SHAH) transform encodes images in an invertible, data-driven, hierarchical and sparse way. It requires three pieces of information to describe an image: the intensities of the pixels, a notion of neighbourhood between the pixels, as well as the spatial location of the pixels in the two-dimensional space. The object describing an image in this way is termed an Intensity Network (IN). We describe below how to define it for a given image. The SHAH transform is a data-driven procedure for dimension reduction, with minimum loss of information at each step. It can be interpreted as an agglomerative-type algorithm, where pixels of an image, each initially forming a separate zone, get progressively grouped into contiguous zones according to a specific criterion. We now describe the core ideas of the SHAH transform.

Defining the IN (Intensity Network). The IN associated with an image is constructed as follows. Consider a grey level image, stored as a real-valued matrix of dimensions $N \times M$. Then, draw a network on this image. Each pixel is a node of the network, and each node is related by edges to its four nearest neighbours (in the left (or west, W), right (east, E), top (north, N) and bottom (south, S) directions, respectively). This graph structure mathematically encodes the idea of neighbourhood between the pixels of the image. More complex topologies are possible; we do not pursue them in this work but implement some in our software (more details below). Assign unique labels $l_1, l_2, \ldots, l_{NM}$ to each node. Associate an orientation with the edges so that each of them consists of an input node l_i and an output node l_j with $i < j$. (We only use the terms 'input' and 'output' to facilitate references to the oriented edge (l_i, l_j).) Store the mapping relating those labels to the Cartesian coordinates of the pixels in a codebook. Moreover, associate uniform weights to all the nodes of the network, as the

information they store (i.e. the value of the related pixel) is a priori equally important in the image description. The object comprising the pixel values (the NM real values stored in an $N \times M$ matrix) and the graph structure (the NM nodes and $2NM - N - M$ edges) embedded in the space through the codebook is termed an Intensity Network (IN). An example of an IN can be found in Figure 1.

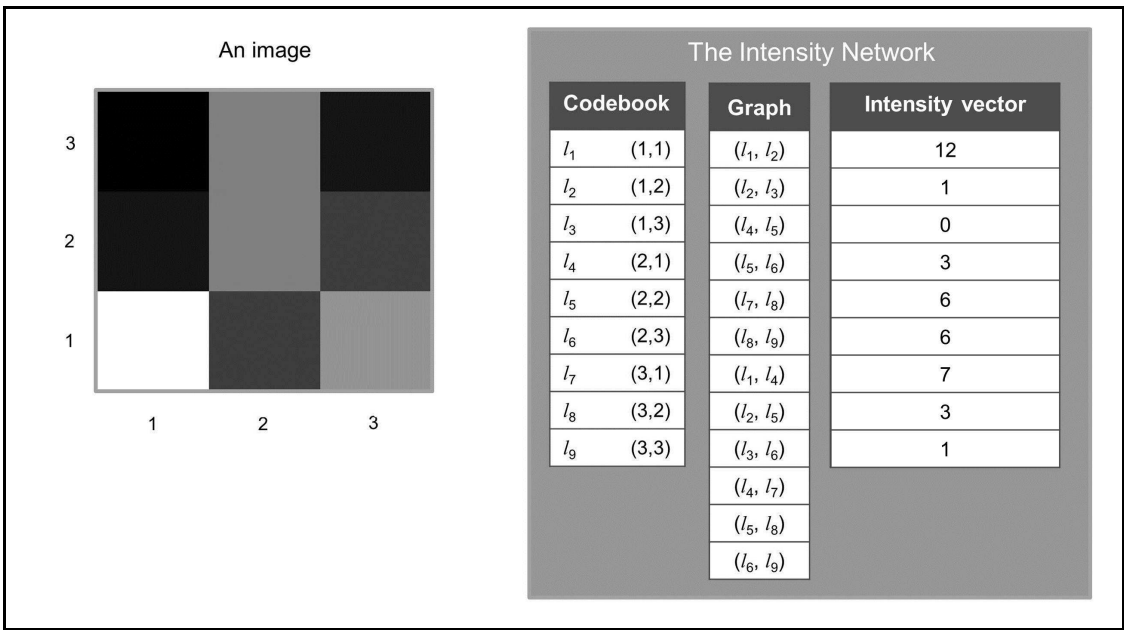


Figure 1: Right: a typical IN. Left: the image it refers to. The codebook encodes the location of the pixels (note the coordinates in this example are indexed from top to bottom, then left to right). The graph structure encodes the neighbourhood relationships between the pixels; the couples (l_i, l_j) are ordered so that $i < j$. The intensity vector encodes the values of the pixels.

Choice of image topology. Throughout this article, we work with 4-element neighbourhoods (W, N, E, S). These are, arguably, the simplest reasonable neighbourhoods, which also offer the fastest computation. More complex neighbourhood structures are clearly possible, most notably 8-element neighbourhoods (W, NW, N, NE, E, SE, S, SW). Although we do not pursue the latter in this work because of the increased computation times, we do implement the SHAH transform with 8-element neighbourhoods in the R code provided at `http://stats.lse.ac.uk/fryzlewicz/shah/shah_code.R`. One attractive feature of the SHAH algorithm is that it always proceeds in the same way once the initial edge topology has been defined. In particular, this is true of the 3-dimensional version of SHAH, also implemented in our software.

Smoothing the image. The idea of the SHAH transform is to progressively smooth the image in a data-adaptive way, while retaining as much information as possible about the current image in each

smoothing step. In practice, compute (weighted) differences between pairs of neighbour nodes along each edge. Those differences are referred to as details. Identify the smallest detail (in absolute value) and replace the values of the corresponding linked nodes by their (weighted) average. Then, reduce those two nodes to a single (linked, merged) node in the network, which is given a larger weight due to the increased number of pixels it encodes. Finally, update the graph structure of the network by removing the edge between the linked nodes. Since the detail being replaced is the smallest one, the loss of information is the smallest possible. This reduction process is iterated $NM - 1$ times, up to the point at which the image is finally reduced to a single node. Figure 2 shows an example of how the graph structure might evolve during the reduction process.

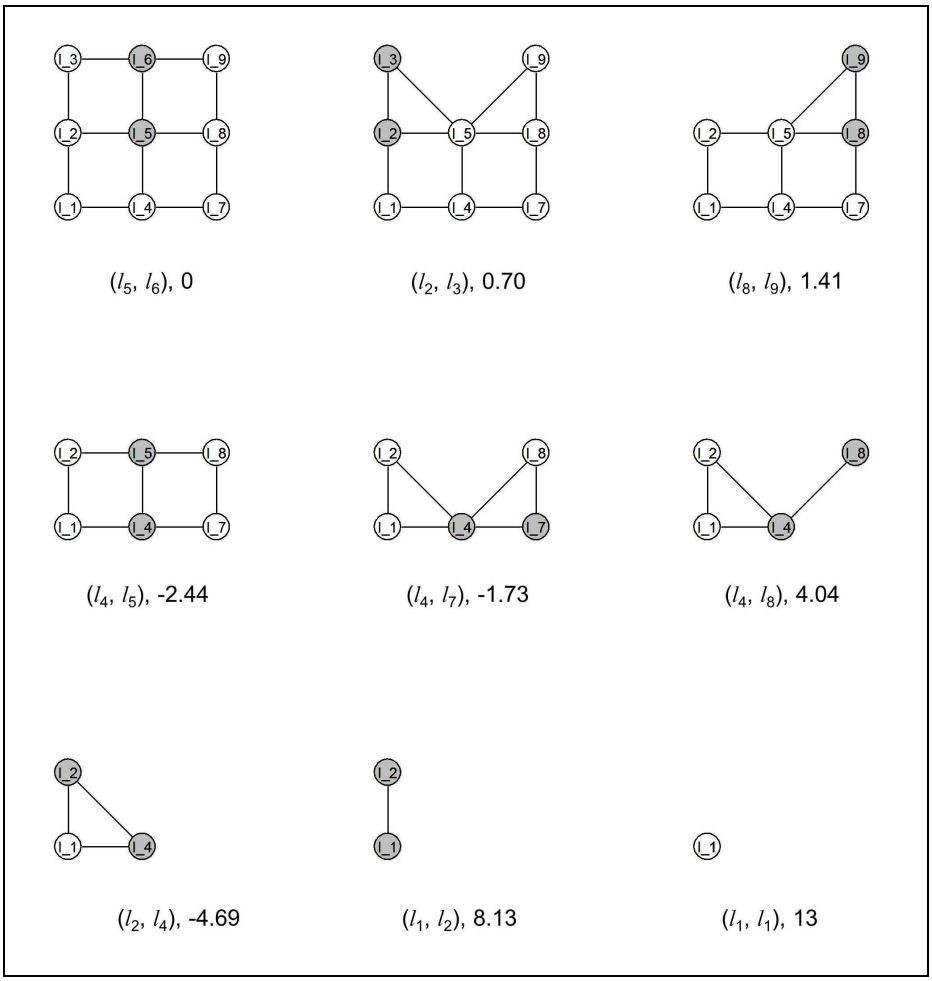


Figure 2: A schematic illustration of SHAH applied to the image from Figure 1. The network is iteratively reduced by one node at each iteration. The nodes selected for reduction are indicated in grey. The labels of the input and output nodes as well as the detail coefficients returned at iteration k are indicated below each image.

Encoding the transform. At each iteration of the algorithm, store the labels of the nodes that are removed, as well as the (weighted) difference between them. Thus, each iteration returns three values: the input node label, the output node label and the selected detail, the latter being the (weighted) value at the output node minus the (weighted) value at the input node of the edge. There are $NM-1$ iterations for reducing an $N \times M$ image to a single node associated with a unique real value for the reduced image. The complete reduction process can thus be stored in two column vectors: one of them encodes the $(NM-1)$ edges and the other encodes the $(NM-1)$ detail coefficients, which can be interpreted as intensity differences. Both of the vectors are constructed element by element, from bottom to top. In addition, the (very) top element of either vector stores, respectively, a degenerate edge linking the remaining node to itself, and the associated value of intensity. Those two vectors combined with the spatial information stored in the codebook define the SHAH transform of the image, an illustration of which can be found in Figure 3. The output of the SHAH transform will also be referred to as the SHAH signature of the input image, see Figure 4 for an example.

The SHAH transform

Codebook		Graph	Intensity differences
l_1	(1,1)	(l_1, l_1)	13
l_2	(1,2)	(l_1, l_2)	8.13
l_3	(1,3)	(l_2, l_4)	-4.69
l_4	(2,1)	(l_4, l_8)	4.04
l_5	(2,2)	(l_4, l_7)	-1.73
l_6	(2,3)	(l_4, l_5)	-2.44
l_7	(3,1)	(l_8, l_9)	1.41
l_8	(3,2)	(l_2, l_3)	0.70
l_9	(3,3)	(l_5, l_6)	0

Figure 3: SHAH of the IN from Figure 1.

Alternatively, the SHAH transform can be viewed as the projection of the image on a particular image-adapted orthonormal basis in which the basis functions are arranged hierarchically (in a multiscale way) and encode the image sparsely (see Figure 5 for an example).

Input (x_1,y_1)		Output (x_2,y_2)		d
l_1	(1,1)	l_1	(1,1)	13
l_1	(1,1)	l_2	(1,2)	8.13
l_2	(1,2)	l_4	(2,1)	-4.69
l_4	(2,1)	l_8	(3,2)	4.04
l_4	(2,1)	l_7	(3,1)	-1.73
l_4	(2,1)	l_5	(2,2)	-2.44
l_8	(3,2)	l_9	(3,3)	1.41
l_2	(1,2)	l_3	(1,3)	0.70
l_5	(2,2)	l_6	(2,3)	0

Rank

Construction

Figure 4: The signature of the image from Figure 1. The algorithm proceeds along the "Construction" arrow, as k decreases from $p-1=8$ to 0. Input and Output columns indicate, respectively, the input and output node of each edge processed. The d column contains the values of the corresponding detail coefficients.

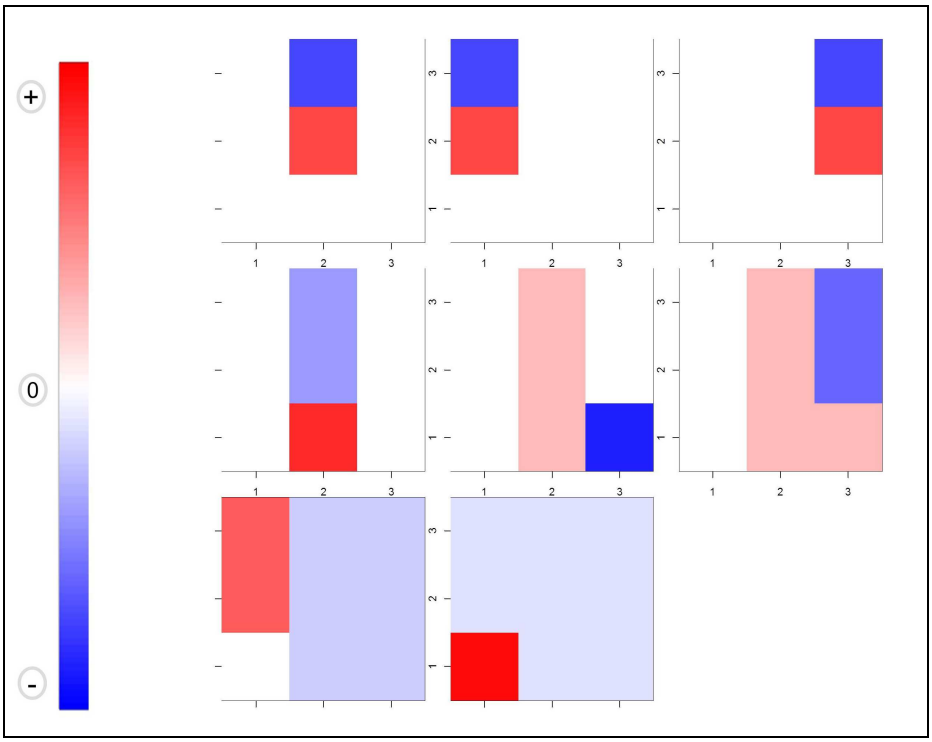


Figure 5: Basis functions $\{\Psi_k\}_{k=p-1\ldots1}$ for the image of Figure 1, obtained in the order of their construction, for rank $k=p-1$ (top left), ..., $k=2$ (bottom left), $k=1$ (bottom right), with p (the number of pixels) being equal to 9. The remaining basis function Ψ_0 is constant. The basis functions are orthonormal and, except for Ψ_0, reflect level changes between contiguous zones in the image.

Overview of the key properties. The SHAH transform is a one-to-one transformation of the input image. It provides a data-driven encoding of images, in which both the pixel intensities and the image topology are accounted for. It describes the image as a linear combination of simple, regionwise-constant basis images, hierarchically organized according to what can be viewed as the importance of the image feature they encode. If SHAH is applied to a noiseless image with edges, then the edges and the regions of constant intensity they delimit are captured in the basis elements, which leads to sparsity in the description of the image. For noisy images, the SHAH transform also attempts to concentrate, in a greedy fashion, as much energy of the image in as few coefficients as possible. The algorithm can be applied to more general geometries than a rectangular image with a grid.

2.2 The SHAH algorithm

In this section, we provide the algorithmic details of the SHAH transform. The input and output of the algorithm are defined in a formal general way. The one-dimensional version of SHAH, termed Unbalanced Haar (UH) was introduced in Fryzlewicz (2007) and applied to curve classification in Timmermans and von Sachs (2015).

Input: an image described as an Intensity Network. The IN of an image $\mathcal{I}$ is defined as a set $\{\mathcal{D}^{(p)}, \mathcal{E}^{\mathrm{IN}}, X^{(p)}\}$, where

- $\mathcal{D}^{(p)}$ is a codebook. It encodes the coordinates of the p points in the image, identified by labels $l = 1 \dots p$. Those points are the locations of the p nodes of the network.
- $\mathcal{E}^{\mathrm{IN}}$ is a graph. It is a ranked set of E oriented edges $\epsilon_l = (j, k)$, $l = 1 \dots E$, with $j, k \in \{1, \dots, p\}, j \neq k$, identifying the linked nodes. In the case when no natural orientation exists for the edges, any choice is equally convenient but an orientation is required for the transform to be invertible.
- $X^{(p)}$ is a vector of intensities. It is a real-valued vector of length p encoding the intensities of the image $\mathcal{I}$ at the successive points defined in $\mathcal{D}^{(p)}$.

A typical example is as follows.

- The image $\mathcal{I}$ is a grey level image of $N \times M$ pixels encoded as a matrix A.

- $\mathcal{D}^{(p)} = \{(j,k)\}_{j=1\ldots N, k=1\ldots M}$ with j, k defining row and column indices in A. The points are labelled $l = 1 \ldots p$, with $p = NM$.
- $X = \{X_l\}_{l=1\ldots p}$, where $X_l = a_{jk}$ is the grey level of the pixel with coordinates (j,k) associated with the label l in A.

Output: the SHAH transform of the IN. The SHAH transform of an image $\mathcal{I}$ is defined as the set $\{\mathcal{D}^{(p)}, \mathcal{E}^{\mathrm{OUT}}, d\}$, where

- $\mathcal{D}^{(p)}$ is the same codebook as in the input.
- $\mathcal{E}^{\mathrm{OUT}}$ is a graph. It is a ranked set of p oriented edges $\epsilon_l = (j,k)$, $l = 0 \ldots p-1$, with $j, k \in \{1, \ldots, p\}, j \neq k$ identifying the linked nodes. Edge ϵ_0 links to itself and $j = k$ for this edge.
- d is a vector of intensity differences. It is a real-valued vector of length p encoding the intensity differences associated with the edges successively defined in $\mathcal{E}^{\mathrm{OUT}}$. The value d_0 is an intensity instead of an intensity difference.

As an example, the output of the SHAH transform of the IN from Figure 1 is in Figure 3.

The algorithm. The algorithm, detailed below, is also illustrated in Figure 2.

The SHAH algorithm

Input

INTENSITY NETWORK= $\{\mathcal{D}^{(p)}, \mathcal{E}^{\mathrm{IN}}, X\}$

Output:

SHAH= $\{\mathcal{D}^{(p)}, \mathcal{E}^{\mathrm{OUT}}, d\}$

Notation:

Index i tracks the current iteration;

$\mathcal{E}^{(i)}$ is the set of edges in the network at iteration i: $\mathcal{E}^{(i)} = \{\epsilon_l = (j,k)\}$

$X^{(i)}$ is the value of the nodes remaining in the network at iteration i.

$j^{(i)} = \{w_l\}_{l=1\ldots p-i}$ is a set of weights associated with the $p-i$ nodes remaining in the network at iteration i.

Initialization:

$i := 1$;

$\mathcal{E}^{(1)} := \mathcal{E}^{\mathrm{IN}}$

$X^{(1)} := X$

for $j = 1 \ldots p$, $w_j := 1$.

Iteration #i:

1. Compute details $\tilde{d}_l$ along each of the edges $\epsilon_l = (j, k)$ in $\mathcal{E}^{(i)}$:

   $\tilde{d}_l := \frac{w_j}{\sqrt{w_j^2+w_k^2}} X_k - \frac{w_k}{\sqrt{w_j^2+w_k^2}} X_j$.

2. Select an edge ϵ_{l^*} with the minimum absolute value of detail:

   $l^* := \arg\min_l |\tilde{d}_l|$.

   In case of multiple equal minimum values $|\tilde{d}_l|$, select the smallest index l. Note $\epsilon_{l^*} = (j^*, k^*)$.

3. Smooth:

   $X_{j^*} := \frac{w_{j^*} X_{j^*} + w_{k^*} X_{k^*}}{\sqrt{w_{j^*}^2 + w_{k^*}^2}}$.
   $X_{k^*} := X_{j^*}$.

4. Encode the partial value of SHAH:

   $\mathcal{E}_{p-i}^{\mathrm{OUT}} := (j^*, k^*)$.

   $d_{p-i} := \tilde{d}_{l^*}$.

5. Reduce the network and prepare next iteration:

   Update $\mathcal{E}^i$ by replacing all indexes k^* by j^*.

   Discard duplicate edges in $\mathcal{E}^i$, retaining only the first occurrence of each edge.

   This defines $\mathcal{E}^{(i+1)}$.

   $w^{(i+1)} := \{w_1, \ldots, w_{j^*-1}, \sqrt{w_{j^*}^2 + w_{k^*}^2}, w_{j^*+1}, \ldots, w_{k^*-1}, w_{k^*+1} \ldots w_{p-i+1}\}$.

   $X^{(i+1)} := \{X_1, \ldots, X_{j^*-1}, X_{j^*}, X_{j^*+1}, \ldots, X_{k^*-1}, X_{k^*+1} \ldots X_{p-i+1}\}$.

   $i := i + 1$.

6. Back to Step 1, until length$(X^{(i)}) = 1$.

Final step:

$\mathcal{E}_0^{\mathrm{OUT}} := (j^*, j^*)$.

$d_0 := \frac{X^{(p)}}{\sqrt{p}}$.

Some remarks are in order. The filter taps $\underline{d}_l = (-\frac{w_k}{\sqrt{w_j^2+w_k^2}}, \frac{w_j}{\sqrt{w_j^2+w_k^2}})$ used in computing the detail coefficient d_l are always chosen so that, if the original image were constant over the region which

the detail coefficient corresponds to, the value of the detail d_l would be zero. This is a consequence of the fact that the starting values of the weights w_j at the beginning of the algorithm are equal to 1. The form of the update to the weight vector $w^{(i+1)}$ is designed to preserve this property as the algorithm progresses. The property that the detail equals zero over constant regions is a natural requirement which causes the SHAH algorithm to offer sparse representations for piecewise-constant images, in a similar vein to standard Haar wavelets which also produce zero detail coefficients in regions of constancy. This, and the requirement that $\|\underline{d}_l\|_2^2 = 1$, uniquely determines the values of the taps applied to compute the detail coefficients, up to sign flips.

The smooth weights $\underline{s}_l = (\frac{w_j}{\sqrt{w_j^2+w_k^2}}, \frac{w_k}{\sqrt{w_j^2+w_k^2}})$ are chosen so that the filters $\underline{d}_{l^*}$ and $\underline{s}_{l^*}$ are orthonormal. This implies that the SHAH transform is conditionally orthonormal, by which we mean "orthonormal given the selected basis". This property is important, amongst others, in the application of SHAH to image denoising where it leads to a fast algorithm for threshold selection, and in fast computation of the inverse SHAH transform.

The SHAH basis selection takes place iteratively, via the minimisation of $|\tilde{d}_l|$ in step 2 of the algorithm. This is a greedy procedure which ensures that each consecutive detail coefficient encodes as little variation of the image as possible, thereby attempting to concentrate as much signal as possible in the latter stages of the algorithm, in the hope of obtaining a sparse representation of the image. This is in contrast to the standard non-adaptive Haar transform for images, where no basis selection takes place, and implies, in particular, that SHAH is a non-linear transformation.

2.3 Computational complexity and variants of the algorithm

In the version described above, the computational complexity of the SHAH algorithm is quadratic in the number of pixels, i.e. is of computational order p^2. This is because at each iteration i, all the edges are examined. However, other variants of the SHAH algorithm are possible, with substantially reduced computational complexity. We outline some ideas below.

- *Examination of a fixed number of edges.* Substantial computational cost can be saved if only a pre-set number of edges (not exceeding a constant), are examined at each iteration i. The edges can be selected in a deterministic or random way. This potentially results in an algorithm of computational order p, i.e. linear in the number of pixels, depending on how the edges are selected.

- *Two- or multi-stage algorithm.* For an image of size $N \times N$, firstly divide the image into $(N/k)^2$ non-overlapping sub-images, each of size $k \times k$. Execute the algorithm on each sub-image separately (stage 1), then execute it on the resulting $N/k \times N/k$ matrix of coefficients d_0 from each sub-image (stage 2). The computational complexity is then $(N/k)^2 k^4 + (N/k)^4$, which attains its minimum when $k = N^{1/3}$, resulting in the complexity of $N^{8/3} = p^{4/3}$. The algorithm can be executed similarly in more stages than one, bringing the computational complexity arbitrarily close to linear, if the number of stages is large enough.

- *Removal of multiple nodes at once.* In the version described above, one pair of nodes is merged at each iteration (this can be viewed as the 'removal' of one of the nodes and updating of the other). An alternative might be to merge multiple pairs of nodes, corresponding to a number of smallest detail values. Merging a fixed proportion $\rho \in (0, 1)$ of the node pairs in each iteration results in an algorithm of computational order $p \log p$. Pairs of nodes can be merged simultaneously in a single iteration if, out of the set of pairs of nodes to be merged, no node belongs to more than one pair.

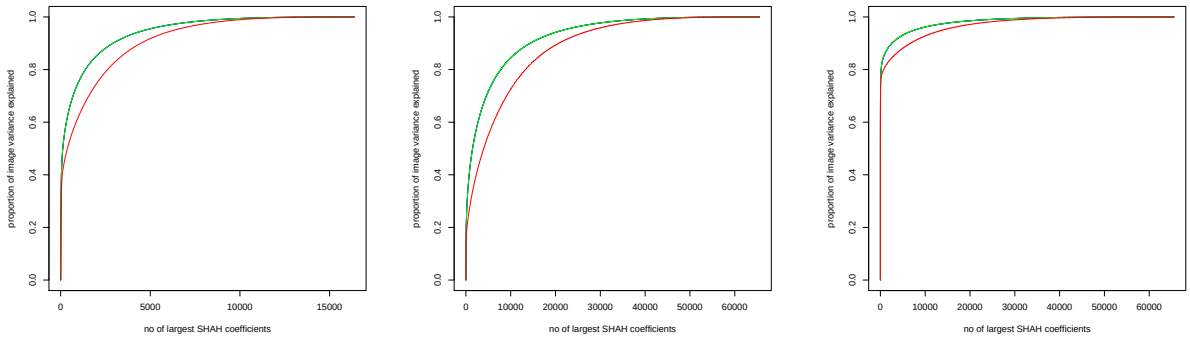
If, in addition to the output described in Section 2.2, the SHAH algorithm stores the filter coefficient $w_{j^*}/\sqrt{w_{j^*}^2 + w_{k^*}^2}$ used at each iteration i, the inverse SHAH transform is performed by simply reversing the steps of the SHAH algorithm. The computational complexity of the inverse SHAH transform is then linear in the number of pixels.

We now briefly discuss how the different variants of the algorithm compare in terms of execution times. Table 1 shows times obtained for 128×128 and 256×256 images. Computational savings will differ depending on the fixed number of edges examined in the "fixed number of edges" version, on the k parameter in the two-stage algorithm and on the ρ parameter in the "removal of multiple nodes at once" version. Clearly, the standard version, implemented in R, is unacceptably slow and one of the faster versions needs to be used in practice.

Figure 6 shows the compression capabilites of the different version of the algorithm on the noisy images from Section 3, Examples 1 and 2. The steeper the curve at the start, the larger the proportion of the variance of the image explained by the same number of the largest SHAH coefficients. The curves corresponding to the standard SHAH, the "fixed number of edges" and the "removal of multiple nodes at once" versions are practically indistinguishable. Understandably, the two stage version is a less good image compressor, because of its region constraints.

	128×128	256×256
Standard SHAH	62	999
Fixed number of edges	19	266
Two stage	5	67
Multiple nodes at once	4	18

Table 1: Execution times of various versions of the SHAH algorithm, for images of sizes 128×128 and 256×256, in seconds on a standard PC; code written in R. The "fixed number of edges" version uses $M = 1000$ edges chosen at random each time. The two-stage version uses $k = 2$. The (removal of) "multiple nodes at once" version uses $\rho = 0.01$.



(a) Noisy image from Section 3, Example 1, size 128×128.

(b) Noisy image from Section 3, Example 1, size 256×256.

(c) Noisy image from Section 3, Example 2, size 256×256.

Figure 6: Proportion of image variance (y axis) explained by each given number of the largest SHAH coefficients (x axis). Black: standard SHAH; blue: "fixed number of edges" version with $M = 1000$ edges chosen at random each time; green: "removal of multiple nodes at once" version with $\rho = 0.01$; red: two-stage version with $k = 2$. The black, blue and green lines virtually overlap.

It is also worth noting that the image from Example 2 is represented more sparsely due to its much higher signal-to-noise ratio than the image in Example 1. This is despite the fact that the noise-free image from Example 1 is piecewise-constant, and therefore it would be represented (much) more sparsely via SHAH than the noise-free image from Example 2, which is not piecewise-constant.

2.4 Properties of SHAH

In this section, we briefly summarize the key mathematical properties of SHAH. The proofs are straightforward, so we omit them.

1. *SHAH as a data-driven orthonormal decomposition of the image.* At iteration i of the SHAH algorithm, each d_{p-i} can be represented as the inner product of the original image X and an image Ψ_{p-i}, where

   - Ψ_{p-i} is selected in a data-driven way at each iteration i,

- Ψ_{p-i} has mean zero, except when $i = p$,
- Ψ_{p-i} is orthonormal to all previously selected Ψ_k, $k > p - i$.

Therefore, $\{\Psi_k\}_{k=0}^{p-1}$ is an orthonormal basis and

$$X = \sum_{k=0}^{p-1} d_k \Psi_k. \tag{1}$$

Further, due to the Parseval identity, the total energy (i.e. the sum of squares) of X equals $\sum_{k=0}^{p-1} d_k^2$. An example of the basis Ψ_k is provided in Figure 5. The orthonormality of Ψ_k is a simple consequence of the orthonormality of the detail and smooth filters used at each iteration of the algorithm. SHAH is an invertible transform.

2. *Hierarchical nature and Haar-like character of the basis* Ψ_k. Let $\text{supp}(\Psi_k)$ denote the support of Ψ_k, i.e. the domain on which it is non-zero.

   - For each $k = 1, \ldots, p-1$, $\text{supp}(\Psi_k)$ consists of two contiguous adjacent zones such that Ψ_k is constant positive on one and constant negative on the other. Ψ_0 is positive and constant on the entire domain.
   - The structure of the basis Ψ_k is hierarchical is the sense that if the supports of Ψ_l and Ψ_k overlap and $l < k$, then $\text{supp}(\Psi_k)$ must be contained either within the zone where Ψ_l is positive or the zone where it is negative.

   These properties are reminiscent of the Haar wavelet basis. However, here, the key difference is that the supports of Ψ_k are determined by the data and can have arbitrary contiguous shapes, as is apparent from the example in Figure 5. This is because the basis images Ψ_k are chosen adaptively from the data at each iteration of the algorithm.

3. *Sparsity of representation and energy concentration.*

   - For each $k = 1, \ldots, p-1$, if $\text{supp}(\Psi_k)$ is contained within a region where X is constant, then the corresponding $d_k = 0$. This is a consequence of the mean-zero property of Ψ_k.
   - Consequently, by the construction of the SHAH algorithm, for a piecewise-constant image X, the only non-zero elements of the vector $(d_0, d_1, \ldots, d_{p-1})$, besides possibly d_0, will be $d_1, \ldots d_{Z-1}$, where Z is the number of zones of contiguous identical values in X, the notion

of contiguity being defined by the linkage structure of the network. Therefore, SHAH encodes the edges of such an image in the sparsest possible way.

- For non-piecewise-constant (e.g. noisy) images, the SHAH algorithm is an attempt to achieve the same effect, i.e. to concentrate as much energy of the image X in as few initial coefficients $d_1, d_2, \ldots$ as possible, and therefore to represent its significant features sparsely.

3 Image denoising using SHAH

The SHAH wavelet transform can be used for image denoising in a similar process to any other wavelet transform, whether adaptive or not. The usual procedure in nonlinear wavelet-based image denoising is to take the wavelet transform of the image, perform a shrinkage/thresholding operation on the wavelet coefficients (in the hope of thresholding out the typically large number of coefficients that carry mostly noise, but retaining most of those carrying signal) and take the inverse wavelet transform. The statistical model we consider in this section is $X_{u,v} = f_{u,v} + \varepsilon_{u,v}$, $u, v = 1, \ldots, N$, where $X_{u,v}$ is the observed noisy image, $f_{u,v}$ is the unknown true image, and $\varepsilon_{u,v}$ is iid noise distributed as $N(0, \sigma^2)$.

At the transform stage, in the case of SHAH, we have a number of options for speeding up computation for large images, as described in Section 2.3. Empirically, we have found that the two-stage algorithm with $k = 2$ or $k = 4$ often leads to the best denoising, especially for noisier images, and this is the version we focus on here. It may come as a surprise that the two-stage algorithm is able to beat the various one-stage versions, despite its worse compression capabilities, as shown in Section 2.3. This, we believe, is due to the fact that the two-stage algorithm is "less greedy" than the one-stage versions because of its region constraints, which may be advantageous for processing noisier images, in which the one-stage algorithms may have more scope for making globally significant basis choice mistakes because of their lack of region constraints.

In the thresholding step, we pursue two strategies: apply either soft, or hard thresholding to each SHAH coefficient d_i, for $i = 1, \ldots, p-1$. This results in the following operations

$$\begin{aligned} \hat{d}_i^S &= \operatorname{sign}(d_i) \max(0, |d_i| - \lambda_S) \qquad \text{(soft thresholding)} \\ \hat{d}_i^H &= d_i \, \mathbb{I}(|d_i| > \lambda_H) \qquad \text{(hard thresholding)}, \end{aligned}$$

where λ_S and λ_H are thresholds used in soft and hard thresholding, respectively, and $\mathbb{I}()$ is the indicator

function.

Motivated by the choice of the regularisation parameter for image smoothing in Kovac and Smith (2011), we choose the threshold λ as follows (our strategy applies to both soft and hard thresholding and therefore we write, generically, λ for either λ^H or λ^S). For each candidate λ, we compute the reconstructed image $\hat{f}^{\lambda}_{u,v}$ and estimate the variance of the empirical residuals as $\tilde{\sigma}^2_{\lambda} = N^{-2}\sum_{u,v=1}^{N}(X_{u,v} - \hat{f}^{\lambda}_{u,v})^2$. By construction, $\tilde{\sigma}^2_0 = 0$ and $\tilde{\sigma}^2_{\infty}$ is the empirical variance of $X_{u,v}$, which is typically larger than σ^2. We then select the largest λ for which

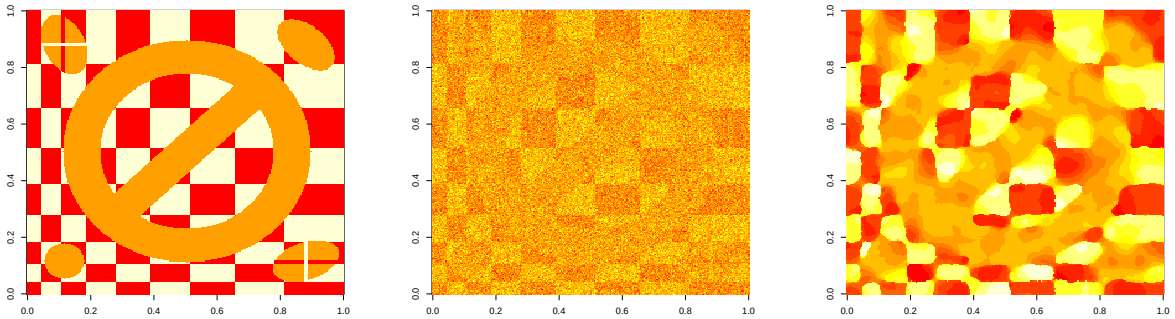
$$\tilde{\sigma}^2_{\lambda} \leq \hat{\sigma}^2, \tag{2}$$

where $\hat{\sigma}^2$ is the Median-Absolute-Deviation-based estimate of σ^2 used in Kovac and Smith (2011).

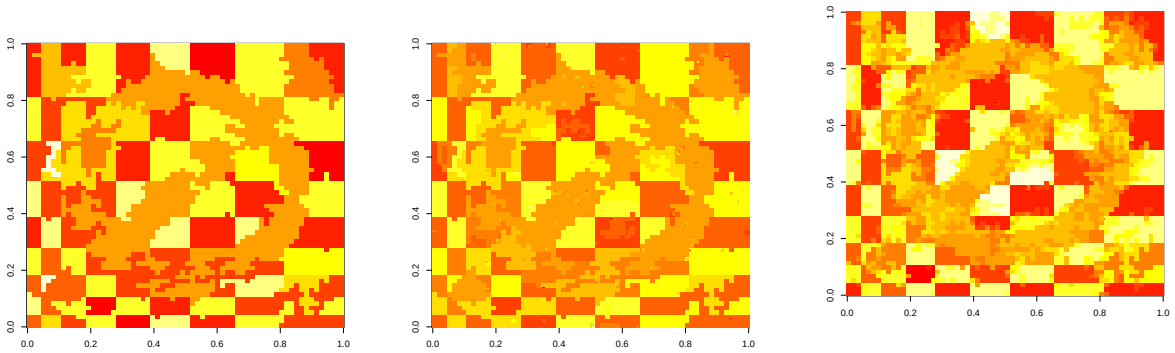
By choosing the largest possible value of λ which leads to "reasonable" residuals from the fit in the sense of (2), we ensure that the reconstructed image is "as simple as possible" in the sense of being composed of the smallest possible number of wavelet coefficients, under the constraint (2). We also note that thanks to the conditional orthonormality of SHAH (i.e., orthonormality given the selected SHAH basis), the operation of checking all possible values of λ can be performed quickly in the SHAH coefficient domain, and is implemented in the code provided in this fast way. We illustrate the potential of the above SHAH-based image denoising procedure on two examples.

Example 1. We use the cartoon medical image, of size 256×256, investigated in Polzehl and Spokoiny (2000) and Kovac and Smith (2011). The clean and noisy images are shown in the top left and top middle plots of Figure 7. This is a piecewise-constant image, for which we expect SHAH to perform well due to the piecewise-constant nature of the SHAH basis functions. The top right plot shows the reconstruction obtained by the Adaptive Weight Smoothing (AWS) technique of Polzehl and Spokoiny (2000), this was produced by the `aws` routine from the `aws` R package (version 1.9-4, dated 2014-03-05), executed with its default parameters.

We process the image via the SHAH denosing procedure described earlier, used here with $k = 4$ and both hard and soft thresholding. The execution of the code, written in R, took under 10 seconds on a standard PC. The reconstructions, shown in the bottom left and bottom middle plot of Figure 7, respectively, appear mostly satisfactory but the reconstructed circle is 'jagged' in appearance. To remedy this, significant improvements are available by performing the following procedure: (a) for $i = 1, \ldots, m$, add further iid $N(0, \sigma_1^2)$ noise to image $X_{u,v}$ to obtain $Y^{(i)}_{u,v}$, (b) perform SHAH denoising



(a) Clean image (b) Noisy image (c) Adaptive Weight Smoothing



(d) SHAH with hard thresholding (e) SHAH with soft thresholding (f) SHAH-avg with hard thresholding

Figure 7: Clean, noisy and denoised image using Adaptive Weight Smoothing and SHAH with hard and soft thresholding as well as SHAH-avg with hard thresholding.

on each image $Y^{(i)}_{u,v}$, (c) average the results over i. We call the thus-constructed procedure SHAH-avg. The averaging introduces an extra smoothing effect which tends to alleviate the jaggedness of the individual reconstructions. We note again that the SHAH denoising procedure is highly nonlinear, and it should be expected that different SHAH bases are selected for each i; therefore the individual reconstructions can be expected to differ enough for each i for the averaging effect to be helpful in removing spurious artefacts present in the individual reconstructions. Throughout this section, we demonstrate SHAH-avg with $\sigma_1 = \hat{\sigma}/2$ and $m = 10$; these parameters have not been optimised in any way.

Table 2 lists the mean-square errors of the various reconstructions, and estimates of their total variation, computed as in Kovac and Smith (2011). SHAH-avg with hard thresholding is by far the best in terms of the MSE. Apart from this method, also AWS-avg (constructed like SHAH-avg but with SHAH replaced by AWS with default parameters) and SHAH with hard thresholding lead to Total Variation values close to those of the clean image. Importantly, we note that AWS-avg does not offer a significant MSE improvement over AWS, due to the latter reconstruction already being smooth, and perhaps even overly so. SHAH-avg offers very significant MSE improvement over SHAH.

We end this example by noting that SHAH with hard thresholding retains 58 non-zero SHAH coefficients for this image, which is fewer than 0.1% of the total number of SHAH coefficients (the latter being equal to the number of pixels). This can be interpreted to mean that the reconstructed image is composed of 58 features, each of which is of the form of a difference between two consecutive regions of the image.

Example 2. We consider the `teddy` image from the R package `wavethresh`. The size is 256×256. Unlike the previous two examples, this image is not piecewise constant. The purpose of this example is to investigate how SHAH handles the task of denoising non-piecewise-constant images. The clean and noisy images are shown in the top left and top middle plots of Figure 8.

The AWS and AWS-avg reconstructions are slightly more appealing visually than those produced by SHAH and SHAH-avg (here with hard thresholding and $k = 2$), which is unsurprising given the non-piecewise-constant character of the image. However, the visual difference does not appear to be large. The MSEs for the various methods tested are in Table 3. SHAH retains 387 non-zero coefficients.

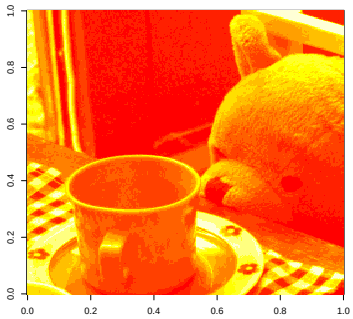
The two examples considered provide evidence for the unsurprising tendency of SHAH to perform better on piecewise-constant images than on general smooth ones. The fundamental reason for this is

	MSE	TV
Wavelet thresholding	4634	3268
Gaussian kernel estimate	2582	5416
Kovac and Smith (2011)*	2896	1696
AWS	2080	4019
AWS-avg	1955	3700
SHAH + hard thresholding	2771	3747
SHAH + soft thresholding	2634	3173
SHAH-avg + hard thresholding	1534	3921
SHAH-avg + soft thresholding	2308	3071
Clean image	0	3787

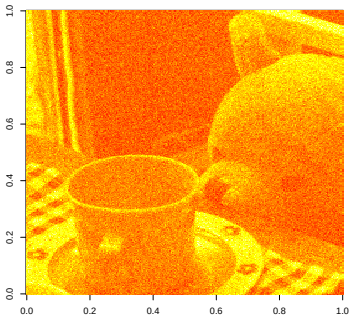
Table 2: Empirical Mean-Square Errors (MSE) and estimates of Total Variation (TV) for the various reconstruction methods of the image from Example 1. The value for the starred method is taken from Kovac and Smith (2011). AWS refers to Adaptive Weight Smoothing from Polzehl and Spokoiny (2000). AWS-avg is constructed like SHAH-avg but with SHAH replaced by AWS with default parameters. Boxed value in the MSE column is the lowest MSE across methods. Boxed values in the TV column are those within 5% of the TV for the clean image. Wavelet thresholding uses the Daubechies' Least Asymmetric filter indexed 10, combined with universal hard thresholding (default option in the R package wavethresh). Gaussian kernel estimate is an unattainable Gaussian kernel smoother in which the bandwidth was chosen by minimising the MSE with respect to the true image (execution: routine kernsm from the R package aws).

	MSE
Wavelet thresholding	2615
Gaussian kernel estimate	1007
AWS	1062
AWS-avg	837
SHAH + hard thresholding	1766
SHAH + soft thresholding	1653
SHAH-avg + hard thresholding	1249
SHAH-avg + soft thresholding	1336

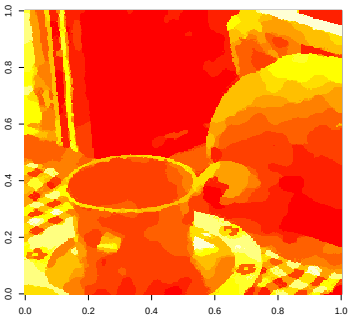
Table 3: Empirical Mean-Square Errors (MSE, divided by 10^4 and rounded) for the various reconstruction methods of the image from Example 2. AWS refers to Adaptive Weight Smoothing from Polzehl and Spokoiny (2000). AWS-avg is constructed like SHAH-avg but with SHAH replaced by AWS with default parameters. Boxed value in the MSE column is the lowest MSE across methods. Wavelet thresholding uses the Daubechies' Least Asymmetric filter indexed 10, combined with universal hard thresholding (default option in the R package wavethresh). Gaussian kernel estimate is an unattainable Gaussian kernel smoother in which the bandwidth was chosen by minimising the MSE with respect to the true image (execution: routine kernsm from the R package aws).



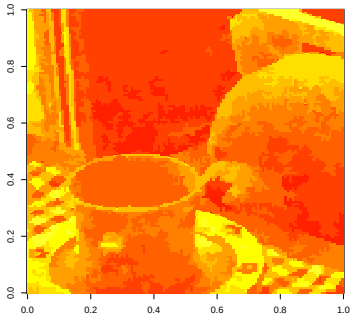
(a) Clean image



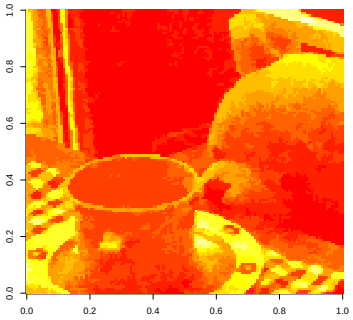
(b) Noisy image



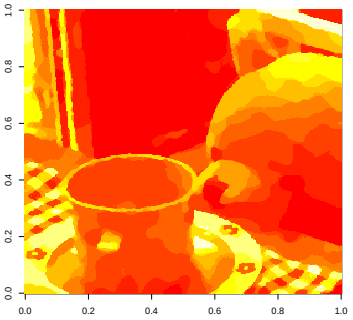
(c) Adaptive Weight Smoothing



(d) SHAH with hard thresholding



(e) SHAH-avg with hard thresholding



(f) Adaptive Weight Smoothing-avg

Figure 8: Clean, noisy and denoised image using AWS, AWS-avg, SHAH with hard thresholding and SHAH-avg with hard thresholding.

that the SHAH building blocks are themselves piecewise-constant.

4 Conclusion

In this article, we have proposed the SHAH (SHape-Adaptive Haar) transform for images, which results in an orthonormal, adaptive decomposition of the image into Haar-like components, arranged hierarchically according to decreasing importance, whose shapes reflect the features present in the image. The decomposition is extremely sparse for piecewise-constant images. It is performed via an stepwise greedy bottom-up algorithm with quadratic computational complexity; however, nearly-linear variants also exist. SHAH is rapidly invertible. We have shown how to use SHAH in conjunction with thresholding for the purpose of image denoising. SHAH is general in scope and can be used not only with images but also with any data that can be described as graphs or networks.

One interesting open question is that of the applicability of SHAH to the decomposition of colour images, for example those using the RGB colour space. In the RGB case, depending on the application, one would entertain the possibility of selecting the SHAH basis either independently for each colour band (e.g. if one wished to remove noise from each band separately), or jointly across the bands. Similar basis choice considerations would apply to multispectral or hyperspectral images. We leave this for future research.

References

J.-P. Antoine, D. Rosca, and P. Vandergheynst. Wavelet transform on manifolds: Old and new approaches. *Applied and Computational Harmonic Analysis*, 28:189–202, 2010.

F. Arandiga, A. Cohen, R. Donat, N. Dyn, and B. Matei. Approximation of piecewise smooth functions and images by edge-adapted (ENO-EA) nonlinear multiresolution techniques. *Applied Computational and Harmonic Analysis*, 24:225–250, 2008.

E. J. Candès and D. L. Donoho. Curvelets and curvilinear integrals. *Journal of Approximation Theory*, 113:59–90, 2001.

R. L. Claypoole and R. G. Baraniuk. A multiresolution wedgelet transform for image processing. In *SPIE Technical Conference on Wavelet Applications in Signal Processing VIII*, San Diego, 2000.

R. R. Coifman and M. Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21:53–94, 2006.

M. Crovella and E. Kolaczyk. Graph wavelets for spatial traffic analysis. In *Proceedings of IEEE Infocom*, 2003.

R. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.

D. L. Donoho. Wedgelets: nearly-minimax estimation of edges. *Annals of Statistics*, 27:859–897, 1999.

P. Fryzlewicz. Unbalanced Haar technique for nonparametric function estimation. *Journal of the American Statistical Association*, 102:1318–1327, 2007.

M. Gavish, B. Nadler, and R. R. Coifman. Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning. In *International Conference on Machine Learning*, pages 367–374, 2010.

D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30:129–150, 2009.

H. Heijmans and J. Goutsias. Nonlinear multiresolution signal decomposition schemes – Part II: Morphological wavelets. *IEEE Transactions on Image Processing*, 9:1897–1913, 2000.

M. Jansen, G. Nason, and B. Silverman. Multiscale methods for data on graphs and irregular multidimensional situations. *Journal of the Royal Statistical Society Series B*, 71:97–125, 2009.

A. Kovac and A. Smith. Nonparametric regression on a graph. *Journal of Computational and Graphical Statistics*, 20:432–447, 2011.

J. Krommweh. Tetrolet transform: A new adaptive Haar wavelet algorithm for sparse image representation. *Journal of Visual Communication and Image Representation*, 21:364–374, 2010.

E. Le Pennec and S. Mallat. Sparse geometrical image representation with bandelets. *IEEE Transactions on Image Processing*, 14:423–438, 2005.

A. B. Lee, B. Nadler, and L. Wasserman. Treelets – An adaptive multi-scale basis for sparse unordered data. *Annals of Applied Statistics*, 2:435–471, 2008.

M. Maggioni, J. C. Bremer, R. R. Coifman, and A. D. Szlam. Biorthogonal diffusion wavelets for multiscale representations on manifolds and graphs. In M. Papadakis, A. F. Laine, and M. A. Unser, editors, *Proc. SPIE Wavelet XI*, volume 5914, 2005.

S. Mallat. Geometrical grouplets. *Applied and Computational Harmonic Analysis*, 26:161–180, 2009a.

S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 3rd edition, 2009b.

S. Mallat and G. Peyré. Orthogonal bandlet bases for geometric images approximation. *Communications on Pure and Applied Mathematics*, 61:1173–1212, 2008.

P. Milanfar. A tour of modern image filtering. *IEEE Signal Processing Magazine*, 30:106–128, 2013.

F. Murtagh. *Multidimensional Clustering Algorithms*. Physica-Verlag, 1985.

F. Murtagh. The Haar wavelet transform of a dendrogram. *Journal of Classification*, 24:3–32, 2007.

P. Narendra and M. Goldberg. Image segmentation with directed trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:185–191, 1980.

G. Peyré. A review of adaptive image representations. *IEEE Journal of Selected Topics in Signal Processing*, 5:896–911, 2011.

G. Plonka. The easy path wavelet transform: A new adaptive wavelet transform for sparse representation of two-dimensional data. *Multiscale Modeling and Simulation*, 7:1474–1496, 2009.

J. Polzehl and V. Spokoiny. Adaptive weights smoothing with applications to image restoration. *Journal of the Royal Statistical Society Series B*, 62:335–354, 2000.

A. Singh, R. D. Nowak, and A. R. Calderbank. Detecting weak but hierarchically-structured patterns in networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS-10)*, pages 749–756, 2010.

W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3:186–200, 1996.

A. D. Szlam, M. Maggioni, R. R. Coifman, and J. C. Bremer. Diffusion-driven multiscale analysis on manifolds and graphs: top-down and bottom-up constructions. In M. Papadakis, A. F. Laine, and M. A. Unser, editors, *Proc. SPIE Wavelet XI*, volume 5914, 2005.

C. Timmermans and R. von Sachs. A novel semi-distance for measuring dissimilarities of curves with sharp local patterns. *Journal of Statistical Planning and Inference*, 160:35–50, 2015.