SHAH: SHape-Adaptive Haar wavelets for image denoising and classification

Piotr Fryzlewicz *  Catherine Timmermans †

December 11, 2014

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 iterative 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.

We also use SHAH to define the BAGIDIS semi-distance between images. It compares both the amplitudes and the locations of the SHAH components of the input images and is flexible enough to account for feature misalignment.

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, dissimilarity, statistical learning.

*Department of Statistics, London School of Economics, Houghton Street, London WC2A 2AE, UK. e-mail: p.fryzlewicz@lse.ac.uk
†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
1 Introduction

The contribution of this work is threefold: firstly, we introduce a new transform for images, based on new SHape-Adaptive Haar (SHAH) wavelets from which it takes its name; secondly, we propose a methodology for image denoising based on the SHAH transform; thirdly, we define a semi-distance between images, termed BAGIDIS, (BAses GIving DIStances), which can be seen as a two-dimensional extension of the eponymous semi-distance for curve comparison (Timmermans and von Sachs, 2013; Timmermans et al., 2013).

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 iterative 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.

The BAGIDIS semi-distance between two images uses a weighted norm to compare the SHAH transforms of the images. As the SHAH transform of each image consists of its expansion in its own image-driven basis, BAGIDIS compares not only the projections of the images onto those bases, but also the bases themselves. In doing so, BAGIDIS accounts for differences in the intensities of the features of the image and differences in their locations. It is thus capable of capturing the similarity of images that are misaligned, which is achieved by few existing distance measures. The focus of
BAGIDIS can be placed on intensity differences, location differences along one axis or the other, or any combination of those at different hierarchical levels in the description of the images. BAGIDIS can be used in association with any distance-based algorithm; in this paper, we use it in association with functional nonparametric regression and discrimination models (Ferraty and Vieu, 2006).

Although this paper focuses on image analysis, it is worth emphasizing that the methodology we propose applies to more general data structures. Indeed, both the SHAH transform and the BAGIDIS semi-distance 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 ineffectiveness in capturing non-horizontal or non-vertical features in an image: curvelets (Candes 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) and edge-adapted nonlinear multiresolution techniques (Arandiga et al., 2008). 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).

**Prediction of an external response from an image.** One may distinguish four broad families of methods for predicting a scalar value or a group membership from an image. They mostly originate from image classification and pattern recognition literature. Firstly, there are methods which compare images through landmarks, i.e. specific features present in each image in a dataset, such as e.g. facial features (eyes, nose, mouth) in face recognition problems (Beumer et al., 2005). The SHAH+BAGIDIS methodology does not fall into this class as it does not require any prior registration of features. Moreover, it does not require preliminary knowledge of what the image represents. Secondly, there are methods that compare images through suitably chosen vectors of features, such as colour histograms, measures of image coarseness, contrast, etc., computed either for the entire image (Hu et al., 2008) or localised (Frome et al., 2006), and then used in a distance measure. The key difference with SHAH is that in this class of methods, the dimensionality reduction process is non-adaptive. Thirdly, certain methods compute distances between the images directly, without relying on a preliminary dimension reduction. These include the Euclidean ($L_2$), Manhattan ($L_1$) and Hausdorff (Huttenlocher et al., 1993) distances. Finally, some methods initially use dimension reduction via e.g. principal components (Ferraty and Vieu, 2006), which SHAH+BAGIDIS bears the most similarity to, although we emphasise again that one unique feature of SHAH is that the decomposition it provides is into a set of particularly simple functions, which may make it more interpretable than these techniques. Moreover, as discussed in Timmermans and von Sachs (2013) and Timmermans et al. (2013) in the framework of curve comparison, computing a distance between principal components is a powerful technique as long as the significant patterns to be compared are perfectly aligned across a dataset. This limitation was the initial motivation for defining the BAGIDIS methodology for curves (Timmermans and von
Sachs, 2013). Another interesting robust technique is IMED (Wang et al., 2005); however, unlike BAGIDIS, it cannot be tuned to extract specific patterns in the image. The problem of misalignment of patterns in images is often tackled by registering the images (Zitová and Flusser, 2003) before computing a distance between them. However, this implies that spatial misalignment cannot be used as discriminative signal (which is possible in BAGIDIS).

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 consequences 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, 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 introduces the BAGIDIS semi-distance for images and shows how to use it in conjunction with SHAH. Section 5 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.

**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](http://stats.lse.ac.uk/fryzlewicz/shah/shah_code). 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
Figure 1: Right: a typical IN. Left: the image it refers to. The codebook encodes the location of the pixels. 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.

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 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.

**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
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.
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.

![Figure 3: SHAH of the IN from Figure 1.](image)

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.

![Table](image)

Alternatively, the SHAH transform can be viewed as the projection of the image on a particu-
lar 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).

\[
\Psi_k \quad k = p-1, \ldots, 1, \quad \text{for the image of Figure 1, obtained in the order of their construction, for rank } k = p-1 \text{ (top left), } \ldots, k = 2 \text{ (bottom left), } k = 1 \text{ (bottom right), with } p \text{ (the number of pixels) being equal to 9. The remaining basis function } \Psi_0 \text{ is constant. The basis functions are orthonormal and, except for } \Psi_0, \text{ reflect level changes between contiguous zones in the image.}
\]

**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 (2013).

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

- $D^{(p)}$ is a codebook. It encodes the coordinates of the $p$ points in the image, identified by labels $l = 1 \ldots p$. Those points are the locations of the $p$ nodes of the network.

- $E^{\text{IN}}$ is a graph. It is a ranked set of $E$ oriented edges $\epsilon_l = (j, k)$, $l = 1 \ldots E$, with $j, k \in \{1, \ldots, p\}$, $j \neq k$, 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 $I$ at the successive points defined in $D^{(p)}$.

A typical example is as follows.

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

- $D^{(p)} = \{(j, k)\}_{j=1 \ldots N, k=1 \ldots M}$ 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 $I$ is defined as the set $\{D^{(p)}, E^{\text{OUT}}, d\}$, where

- $D^{(p)}$ is the same codebook as in the input.

- $E^{\text{OUT}}$ 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 $\epsilon_0$ 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 $E^{\text{OUT}}$. 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**

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

**Output:**

$\text{SHAH} = \{D^{(i)}, E^{\text{OUT}}, d\}$

**Notation:**

- Index $i$ tracks the current iteration;
- $E^{(i)}$ is the set of edges in the network at iteration $i$: $E^{(i)} = \{\epsilon_l = (j, k)\}$
- $X^{(i)}$ is the value of the nodes remaining in the network at iteration $i$.
- $j^{(i)} = \{w_l\}_{l=1...p-i}$ is a set of weights associated with the $p - i$ nodes remaining in the network at iteration $i$.

**Initialization:**

$i := 1$;

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

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

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

**Iteration #i:**

1. Compute ‘details’ $d_l$ along each of the edges $\epsilon_l = (j, k)$ in $E^{(i)}$:

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

2. Select an edge $\epsilon^{*}$ with the minimum absolute value of detail:

   $$l^{*} := \arg \min |d_l|.$$  

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

---

13
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 SH\(A\)H:
\[
E^\text{OUT}_p := (j^*, k^*).
\]
\[
d_{p-i} := d_{i^*}.
\]

5. Reduce the network and prepare next iteration:
Update \(E^i\) by replacing all indexes \(k^*\) by \(j^*\).
Discard duplicate edges in \(E^i\), retaining only the first occurrence of each edge.
This defines \(E^{(i+1)}\).
\[
w^{(i+1)} := \{w_1, \ldots, w_{j^*-1}, \sqrt{w_{j^*}^2 + w_{k^*}^2}, w_{j^*+1}, \ldots, w_{k^*-1}, w_{k^*+1} \ldots w_{p-i+1}\}.
\]
\[
X^{(i+1)} := \{X_1, \ldots, X_{j^*-1}, X_{j^*}, X_{j^*+1}, \ldots, X_{k^*-1}, X_{k^*+1} \ldots X_{p-i+1}\}.
\]
i := i + 1.

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

**Final step:**
\[
E^\text{OUT}_0 := (j^*, j^*).
\]
\[
d_0 := \frac{X^{(p)}}{X^{(p)}}.
\]

Some remarks are in order. The filter taps \(d_l = \left(-\frac{w_k}{\sqrt{w_{j^*}^2 + w_{k^*}^2}}, \frac{w_j}{\sqrt{w_{j^*}^2 + w_{k^*}^2}}\right)\) 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 SH\(A\)H 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 \(\|d_l\|^2 = 1\), uniquely determines the values of the taps applied to compute the detail coefficients, up to sign flips.

The smooth weights \(s_l = \left(\frac{w_j}{\sqrt{w_{j^*}^2 + w_{k^*}^2}}, \frac{w_k}{\sqrt{w_{j^*}^2 + w_{k^*}^2}}\right)\) are chosen so that the filters \(d_{j^*}\) and \(s_{j^*}\) are orthonormal. This implies that the SH\(A\)H transform is conditionally orthonormal, given the selected basis. This property is important, amongst others, in the application of SH\(A\)H 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 $|d_i|$ 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 $(N/k)^2k^4 + (N/k)^4$, 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 $128 \times 128$ and $256 \times 256$ 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 $\rho$ parameter in the “removal of multiple nodes at once” version. Clearly, the standard version, implemented in pure R, is unacceptably slow and one of the faster versions needs to be used in practice.

Figure 6 shows the compression capabilities of the different version of the algorithm. 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.

### 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.

<table>
<thead>
<tr>
<th></th>
<th>$128 \times 128$</th>
<th>$256 \times 256$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard SHAH</td>
<td>62</td>
<td>999</td>
</tr>
<tr>
<td>Fixed number of edges</td>
<td>19</td>
<td>266</td>
</tr>
<tr>
<td>Two stage</td>
<td>5</td>
<td>67</td>
</tr>
<tr>
<td>Multiple nodes at once</td>
<td>4</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 1: Execution times of various versions of the SHAH algorithm, for images of sizes $128 \times 128$ and $256 \times 256$, in seconds on a standard PC; code written in pure 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 $\rho = 0.01$. 
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.

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 $\Psi_{p-i}$, where

- $\Psi_{p-i}$ is selected in a data-driven way at each iteration $i$,
- $\Psi_{p-i}$ has mean zero, except when $i = p$,
- $\Psi_{p-i}$ is orthonormal to all previously selected $\Psi_k$, $k > p - i$.

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

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

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 $\Psi_k$ is provided in Figure 5. The orthonormality of $\Psi_k$ 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** $\Psi_k$. Let $\text{supp}(\Psi_k)$ denote the support of $\Psi_k$, 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 $\Psi_k$ is constant positive on one and constant negative on the other. $\Psi_0$ is positive and constant on the entire domain.
- The structure of the basis $\Psi_k$ is hierarchical in the sense that if the supports of $\Psi_l$ and $\Psi_k$ overlap and $l < k$, then $\text{supp}(\Psi_k)$ must be contained either within the zone where $\Psi_l$ 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 $\Psi_k$ 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 $\Psi_k$ 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 $\Psi_k$.
- 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.

4. **Efficient encoding of SHAH output through the signature of the image.** We end this section by defining a particular arrangement of the information contained in the output of SHAH, which we term the signature of the image. We define this as a matrix of dimension $p \times 5$, whose $k$th row, $k = 0 \ldots p - 1$, is related to rank $k$ in the basis expansion (1). Its first two elements contain the Cartesian coordinates $(x_1, y_1)$ of the input node at rank $k$, next two contain the Cartesian coordinates
coordinates \((x_2, y_2)\) of the output node at rank \(k\). The final element is \(d_k\). An example was
given in Figure 4. This notation will be useful in Section 4.

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

\[
\hat{d}_i^S = \text{sign}(d_i) \max(0, |d_i| - \lambda_S) \quad \text{(soft thresholding)}
\]
\[
\hat{d}_i^H = d_i \mathbb{1}(|d_i| > \lambda_H) \quad \text{(hard thresholding)},
\]

where \(\lambda_S\) and \(\lambda_H\) are thresholds used in soft and hard thresholding, respectively.

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

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

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

By choosing the largest possible value of $\lambda$ 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 given the selected SHAH basis, the operation of checking all possible values of $\lambda$ 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 three examples.

**Example 1.** We use the cartoon medical image, of size $256 \times 256$, 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 denoising procedure described earlier, used here with $k = 4$ and both hard and soft thresholding. The execution of the code, written in pure 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 $Y_{u,v}^{(i)}$, (b) perform SHAH denoising on each image $Y_{u,v}^{(i)}$, (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
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.
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.** In this example, we consider a two-colour 128 × 128 image with a constant background and a constant raised circle in the middle. The clean and noisy images are shown in the top left and

<table>
<thead>
<tr>
<th>method</th>
<th>MSE</th>
<th>TV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wavelet thresholding*</td>
<td>5193</td>
<td>3462</td>
</tr>
<tr>
<td>Gaussian kernel estimate*</td>
<td>3650</td>
<td>2371</td>
</tr>
<tr>
<td>Kovac and Smith (2011)*</td>
<td>2896</td>
<td>1696</td>
</tr>
<tr>
<td>AWS</td>
<td>2080</td>
<td>4019</td>
</tr>
<tr>
<td>AWS-avg</td>
<td>1955</td>
<td><strong>3700</strong></td>
</tr>
<tr>
<td>SHAH + hard thresholding</td>
<td>2771</td>
<td><strong>3747</strong></td>
</tr>
<tr>
<td>SHAH + soft thresholding</td>
<td>2634</td>
<td>3173</td>
</tr>
<tr>
<td>SHAH-avg + hard thresholding</td>
<td>1534</td>
<td><strong>3921</strong></td>
</tr>
<tr>
<td>SHAH-avg + soft thresholding</td>
<td>2308</td>
<td>3071</td>
</tr>
<tr>
<td>Clean image</td>
<td>0</td>
<td>3787</td>
</tr>
</tbody>
</table>

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 values for the starred methods are 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.
Figure 8: Clean, noisy and denoised image using AWS, AWS-avg, SHAH with soft thresholding and SHAH-avg with soft thresholding.

top middle plots of Figure 8. The noisy image has an extremely low signal-to-noise ratio and the circular feature will be invisible to many human observers.

AWS and AWS-avg reconstructions are shown in the right-hand column of Figure 8. While both methods show the feature, they fail to indicate its piecewise-constant character, and in particular, the constant character of the background. The edge is more clearly visible in AWS than it is in AWS-avg. The MSEs of the two reconstructions are, respectively, 2000 and 2109.

SHAH with soft thresholding (with $k = 4$, shown in the bottom left plot of Figure 8) reflects the two-colour, piecewise-constant character of the image, although the shape of the reconstructed feature is not exactly right. SHAH-avg (bottom middle plot of Figure 8) shows, at least to some extent, the constant character of the background, and the shape of the feature is roughly right. The MSEs of SHAH and SHAH-avg are, respectively, 3107 and 1504. SHAH-avg exhibits the lowest MSE out of the 4 methods tested.
Figure 9: Clean, noisy and denoised image using AWS, AWS-avg, SHAH with hard thresholding and SHAH-avg with hard thresholding.

Example 3. We consider the teddy image from the R package wavethresh. The size is $256 \times 256$. 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 9.

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 (divided by $10^4$ and rounded) are: AWS 1062, AWS-avg 837, SHAH 1766, SHAH-avg 1249. SHAH retains 387 non-zero coefficients.
4 SHAH-based distance measure for images and image classification

The BAGIDIS semi-distance for curves was introduced in Timmermans and von Sachs (2013) as a data-driven and adaptive way for comparing curves with possibly misaligned sharp local patterns. Its good classification and discrimination performance was reported in Timmermans and von Sachs (2013); Timmermans et al. (2013, 2012). In this section, we extend BAGIDIS to images.

BAGIDIS for curves takes as its input the Unbalanced Haar (Fryzlewicz, 2007) decompositions of both curves. Similarly, BAGIDIS for images will operate on the SHAH transformations of both images. BAGIDIS does not require preliminary knowledge of what the images may represent. This sets it apart from many methods based on landmarks, which are designed for images containing particular features.

4.1 Definition

The BAGIDIS semi-distance provides a measure of the dissimilarity of pairs of images observed on the same grid of measurements. The compared images are first described through their Intensity Networks, with an identical codebook and an identical graph component. The SHAH transforms of both images are then computed and represented as their signatures $s^{(i)}_k = (x^{(i)}_1, y^{(i)}_1, x^{(i)}_2, y^{(i)}_2, d^{(i)}_k)$ for $i = 1, 2$. BAGIDIS compares images $I^{(1)}$ and $I^{(2)}$ hierarchically in the following way.

$$d_{BAGIDIS}(I^{(1)}, I^{(2)}) = \sum_{k=0}^{p-1} \left( \alpha_k (x^{(1)}_1 - x^{(2)}_1)^2 + \beta_k (y^{(1)}_1 - y^{(2)}_1)^2 + \gamma_k (x^{(1)}_2 - x^{(2)}_2)^2 + \delta_k (y^{(1)}_2 - y^{(2)}_2)^2 + \epsilon_k (d^{(1)}_k - d^{(2)}_k)^2 \right)^{1/2}$$

$$= \sum_{k=0}^{p-1} \left\| s^{(1)}_k - s^{(2)}_k \right\|_{2,p_k}$$

(3)

where $p_k = (\alpha_k, \beta_k, \gamma_k, \delta_k, \epsilon_k)$ is a vector of parameters in $\mathbb{R}_+^5$. Notation $\left\| \cdot \right\|_{2,p_k}$ indicates that the contribution of rank $k$ to the distance is a semi-2-norm, weighted by the parameters $p_k$. The semi-norm property arises since the parameters $\alpha_k, \beta_k, \gamma_k, \delta_k$ and $\epsilon_k$ can be 0 so that $\left\| s^{(1)}_k - s^{(2)}_k \right\|_{2,p_k} = 0$ does not necessarily imply $s^{(1)}_k = s^{(2)}_k$. However, (3) satisfies the triangle inequality and the property that $d_{BAGIDIS}(I, I) = 0$.

Such rich parameterization tends to be unmanageable in practice. The following more constrained
form might be more appropriate:

\[
\begin{align*}
\text{d}^{\text{Bagidis}}(T^{(1)}, T^{(2)}) & = \sum_{k=0}^{p-1} w_k \left( \lambda_x (x_{1k}^{(1)} - x_{1k}^{(2)})^2 + (x_{2k}^{(1)} - x_{2k}^{(2)})^2 \\
& + \lambda_y (y_{1k}^{(1)} - y_{1k}^{(2)})^2 + (y_{2k}^{(1)} - y_{2k}^{(2)})^2 \\
& + \lambda_D (d_k^{(1)} - d_k^{(2)})^2 \right)^{1/2}
\end{align*}
\]

where \( w_k \) is a non-negative weight function and \( \lambda_x, \lambda_y, \lambda_D \) are scaling parameters satisfying \( 0 \leq \lambda_x, \lambda_y, \lambda_D \leq 1 \) and \( \lambda_x + \lambda_y + \lambda_D = 1 \). The parameters \( \lambda_x, \lambda_y, \lambda_D \) are responsible for variations along the \( x \)-axis, along the \( y \)-axis, and changes in the intensity, respectively. (4) is a direct extension of BAGIDIS for curves from Timmermans and von Sachs (2013).

4.2 Discussion of the BAGIDIS semi-distance

BAGIDIS compares the signatures of the two images, which can be viewed as “paths” in the \( \mathbb{R}^5 \) space. The hierarchy induced by SHAH makes the paths comparable from one image to another. One might expect the initial parts of the paths to reflect the main (most important) features of the images, and their final parts to reflect patterns of decreasing importance and ultimately noise.

The choice of the parameters \( p_k \) is key. As is often the case in this type of problem, the selection rule should be flexible enough for BAGIDIS to be able to extract relevant information about the discriminative patterns in the images, but on the other hand constrained enough to prevent overfit. Later, we discuss the choice of \( p_k \) in some practical settings. In addition, \( p_k \) may be used to incorporate prior knowledge of the differences between the images. Finally, \( p_k \) can also be used as an exploratory tool for diagnosing what makes the images different.

Thanks to the SHAH transform underlying the BAGIDIS semi-distance, the latter takes into account the notion of neighbourhood between the pixels, which permits the identification of important regions/features in the image and ultimately their comparison in a simple way. A key property is that BAGIDIS compares both the locations and intensities of the patterns. Finally, we emphasise the fact that BAGIDIS does not require any preliminary smoothing or registration of the images.
4.3 Practical use of BAGIDIS

BAGIDIS is a semi-distance and hence can be used in a variety of procedures that use semi-distance on input, such as e.g. data visualisation, unsupervised discrimination or prediction in a supervised setting. Multidimensional scaling (Cox and Cox, 2008), Ward’s hierarchical algorithm (Lebart et al., 2004) and nonparametric functional regression (Ferraty and Vieu, 2006) are examples of algorithms that can be used with BAGIDIS in those different areas, respectively. Furthermore, the descriptive statistics and tests, specifically designed for BAGIDIS for curves (Timmermans and von Sachs, 2013), can be extended to images. Key to all these uses is a suitable selection of the parameters to be used in formula (4), which is typically achieved through an optimization strategy, and done differently depending on whether the problem falls into the ‘supervised’ or ‘unsupervised’ learning category. We do not discuss the latter in this work.

**Supervised learning.** Leave-one-out cross-validation within a training set, optimizing either e.g. the mean square error of prediction (in regression problems) or the number of misclassifications (in classification problems), can be used to select the parameters of BAGIDIS in supervised learning. Theoretical efficacy of such a scheme has been proved in the case of functional nonparametric kernel regression (Ferraty and Vieu, 2006) in Timmermans et al. (2013). To save computational time, the examples presented later in this section use a cross-validation scheme where the weights $w_k$ in (4) are chosen to be a step function taking the value of 1 up to a certain rank $k$ (to be chosen via cross-validation) and 0 thereafter. A computationally cheaper alternative, also illustrated later, might be to only select those components of the signature that significantly relate to the response, e.g. through a significant correlation with the response in the case of regression problems.

4.4 The moving pixel

We now illustrate the ability of BAGIDIS (versus competitors) to capture the relevant variations of a signal regarding the location or the intensity of a simple pattern in a noisy setting.

Three test scenarios are considered. They all involve $5 \times 5$ images whose noiseless pixels have values zero except at one randomly selected location (the “moving pixel”) whose intensity varies according to the scenario. In each case, a dataset of 90 images (70 training + 20 test) is generated and Gaussian noise with standard deviation $\sigma$ is added. Each image is related to a scalar value $Y$, determined according to the scenario. The goal is to predict $Y$. 

27
• Scenario 1: Capturing the location of a moving pixel. The noiseless intensity of the moving pixel is 10; $Y$ is the row index of the moving pixel. The maximum $\sigma$ is 4.

• Scenario 2: Capturing the location of a moving pixel with varying intensity. The noiseless intensity of the moving pixel is chosen randomly from the set $\{5, 10, 15, 20, 25\}$ with equal probabilities; $Y$ is the row index of the moving pixel. The maximum $\sigma$ is 10.

• Scenario 3: Capturing both the location and the intensity of a moving pixel. The noiseless intensity of the moving pixel and the maximum value of $\sigma$ are as in Scenario 2; $Y$ is the sum of the intensity and the row index of the moving pixel.

In each scenario, we estimate a nonparametric functional regression model (Ferraty and Vieu, 2006) on the training set, using BAGIDIS and its competitors (below). The mean square prediction error over the test images is recorded, and averaged over 10 random test/training splits.

The parameterization of the BAGIDIS semi-distance is chosen by considering the correlations $\hat{\rho}$ between each component of the signature (including the absolute value of the details) at each rank, and the response $Y$. A standard test of significance is applied on the correlations, with $H_0 : \hat{\rho} = 0$ and $H_1 : \hat{\rho} \neq 0$ with significance level $\alpha = 0.1$.

The first alternative we consider is a PCA-based semi-distance with $k = 1, 3$ or 6 components, obtained as follows:

$$d_{k}^{\text{PCA}}(I^{(1)}, I^{(2)}) = \left( \sum_{q=1}^{k} \left( \sum_{x=1}^{N} \sum_{y=1}^{M} (I^{(1)}(x, y) - I^{(2)}(x, y)) \hat{v}_{q}(x, y) \right)^2 \right)^{1/2},$$

where $\hat{v}_q, q = 1 \ldots k$ is the $q^{th}$ estimated eigenfunction and the pair $(x, y), x = 1 \ldots N, y = 1 \ldots M$, identifies the location of a pixel. We also consider the IMED distance, used in its non-parameterized form (equation (6) in Wang et al. 2005), and the Euclidean and Hausdorff distances, for which no parameterization is needed. The bandwidth of the regression estimator is chosen by cross-validation in each case.

The results are illustrated in Figure 10 (as a function of $\sigma$). The Euclidean distance and the PCA-based semi-distance perform poorly, which is unsurprising as the images are misaligned. IMED is quite competitive in Scenario 1 for low noise levels, due to its robustness to misalignment. However, it fails as soon as the intensity of the moving pixel varies (Scenarios 2, 3) as it cannot decouple the information about intensity and location (which BAGIDIS is able to achieve). The Hausdorff distance
is the most successful in Scenario 3, with BAGIDIS not far behind. However, the Hausdorff distance does poorly in Scenarios 1 and 2 as it is not able to distinguish between two images with a moving pixel of the same intensity but at different locations. Its good performance in Scenario 3 is due to the fact that the part of the response that concerns the location (with values in \( \{1, 2, 3, 4, 5\} \)) is dominated by the part of the response related to intensity (with values > 5). In summary, BAGIDIS is either the best or close to best in all experimental settings.

We end this section with a simple illustration of why BAGIDIS is flexible enough to be insensitive to image misalignment. Consider two moving-pixel images, indexed by \( i = 1, 2 \), each with the moving pixel in a different location. The SHAH coefficient vector for each will be of the form \((d_0^{(i)}, d_1^{(i)}, 0, 0, \ldots, 0)\), where \(d_0^{(1)} = d_0^{(2)}\) and \(d_1^{(1)} = d_1^{(2)}\). Therefore, by tuning the BAGIDIS semi-distance so that the other components of the signatures of these images (which encode the differing locations of the pixels) are ignored, we obtain that the BAGIDIS semi-distance between the two (misaligned) images is zero.

### 4.5 Handwritten digits

In this Section, we use SHAH to analyse (in a supervised way) grayscale images of handwritten digits ‘zero’ and ‘one’ from the publicly available MNIST database (http://yann.lecun.com/exdb/mnist/). The database contains 70000 black-and-white 28x28 images of digits. We use Brendan O’Connor’s code from https://gist.github.com/39760 to read the data into R. We restrict ourselves to the first 85 zeros and ones from the database.

**Supervised learning.** A nonparametric functional discrimination model (Ferraty and Vieu, 2006), with a k-NN algorithm, is trained on a dataset of 70 images of ones and 70 images of zeros. It is then tested on an independent set of 15 ones and 15 zeros. The number \( k \) of neighbours is selected using a local cross-validation (routine funopadi.knn.lcv from the companion website of Ferraty and Vieu 2006, adapted for BAGIDIS.) We also perform cross-validation to choose \((\lambda_x, \lambda_y, \lambda_D)\) in formula (4); they are chosen from the set \( \{0, 1/3, 2/3, 1\} \) under the constraint that they sum to 1. Each combination of these values is tested with several weight functions \( w_k \), the latter being step functions with values 1 up to a rank \( k = W \) and 0 thereafter; \( W = 2, 5, 10, 20, 50 \) and 100 are tested. This experimental design is repeated twice with different training sets and test sets. The proportion of misclassification on the test set is recorded for each configuration. Results are presented in Figure 11. The results imply that the first ranks of the signature clearly carry some discriminative information in the detail.
Figure 10: The moving pixel. Mean-square prediction error averaged over 10 repetitions, as a function of $\sigma$, in Scenarios 1, 2, 3 (respectively from top to bottom), for the competing methods.
component. However, one needs to look further down in the hierarchy (between ranks 50 and 100) to achieve perfect discrimination. This extra information lies in the components \((x_1, y_1)\) and \((x_2, y_2)\) of the signature, indicating that some information about the geometry of the image is needed. Given this, our recommendation for the choice of the parameterization would be \((W = 100, \lambda_D = 0, \lambda_x = 0.5, \lambda_y = 0.5)\).

We also tested the following faster alternative to cross-validation. We considered the formulation (3) of the semi-distance, where \(i = 0\) (\(i = 1\)) corresponded to the class of zeros (ones). We performed a t-test for the difference in the within-class means of each component of the signature vectors \(s_k^{(i)}\) of the images in the training set, for each \(k\) separately. The p-values were recorded, and only those components with extremely small p-values \((\leq 10^{-10})\) were assigned the weight of 1 in the parameterization, with all other weights being set to 0. The thus-trained model was used for discriminating the images in the test set and the number of misclassifications was recorded. The procedure was repeated 8 times with different training sets and test sets. We obtained the average misclassification rate of 0.02.

Overall, we find these results encouraging, especially given the small size of the training set and the fact that no prior information about the images or the shapes of the discriminant patterns are used in at any stage of the procedure.
5 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 iterative 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, and with the BAGIDIS semi-distance for the purpose of image classification. 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


