Image empirical mode decomposition (IEMD) is an empirical mode decomposition concept used in Hilbert–Huang transform (HHT) expanded into two dimensions for the use on images. IEMD provides a tool for image processing by its special ability to locally separate superposed spatial frequencies. The tendency is that the intrinsic mode functions (IMFs) other than the first are low-frequency images. In this study we give an overview of the state-of-the-art methods to decompose an image into a number of IMFs and a residue image with a minimum number of extrema points, together with the use of the method. Ideas and open problems are presented.

Keywords: Image processing; EMD; HHT; BEMD; IEMD; empiquency; variable sampling; noise reduction; image coding; compression; texture; quadrature filter; instantaneous frequency.

1. Introduction

Empirical mode decomposition (EMD) was introduced for the analysis of time-domain signals. The results and ideas in time-domain applications using EMD apply to two-dimensional (2D) signals, such as images, as well. IEMD decomposes the spatial frequency components into a set of IMFs where the highest spatial frequency component of each spatial position is in the first IMF and the second highest spatial frequency component of each spatial position is in the second IMF, and so on. An IMF is defined in Ref. 1 as a function in which the number of extrema points and the number of zero crossings are the same or differ by one. The upper and lower envelopes of the IMF are symmetric with respect to the local mean.

The EMD of images relies on proper interpolation in two dimensions. The problem is to fit a surface to the 2D scattered data points representing the extrema points. The interpolated surface must go through each data point. Overshot must be avoided and the second derivative must be continuous everywhere for the signal to be smooth enough for 2D-EMD.

*The content in this study was first presented at the First International Conference on the Advances of Hilbert–Huang Transform and Its Applications, in Taiwan 2006.
The residue signal should have only a few extrema points. Huang\textsuperscript{1} state that there should be no extrema points in the one-dimensional (1D) case and ideally we would like to have a similar criterion for the 2D case. This criterion is relaxed for the 2D case.

The border constraints are even more important in two dimensions than they are in the 1D-EMD case. One of the main objections for using spline interpolation in the EMD for 2D signals such as images is that the borders cause too many problems. The set of extrema points is very sparse and since the interpolation methods only interpolate between points, the borders need special care. We proposed the trick of adding extra data points at the borders to the set of extrema points.\textsuperscript{5} These extra points are placed at the corners of the image and equally spaced around the border. Without these extra points, the areas not covered by the interpolation traverse into the image in the sifting process.

This study concentrates on the IEMD as a tool for image processing and aims at displaying the advantages and shortcomings of the method. First the sifting for 2D-IMF is revisited in Sec. 2. Methods for selection of extrema points and also the use of significant extrema points as an important tool are discussed in Sec. 3. Section 4 discusses different EMD methods for 2D signals. The concept of Empiquency is presented in Sec. 5 and Image HHT in Sec. 6. Applications, such as variable sampling of the EMD, image compression, texture analysis, and noise reduction are presented in Sec. 7. The study ends with a discussion of open problems in the development of IEMD.

2. Sifting for the Two-Dimensional IMF

To find the first IMF, start with the image itself as input signal $h_{10}(m,n) = x(m,m)$. The first index is the IMF number, $l = 1, \ldots, L$, and the second index is the iteration number, $k = 1, \ldots, K$, in the sifting process; $m$ and $n$ represent the two spatial dimensions. To find the next IMF, use the residue corresponding to the previously found IMF as input signal $h_{20}(m,n) = r_1(m,m)$.

The sifting process to find the IMFs of a signal $x(m,n)$, comprises the following steps:

1. Find the positions and amplitudes of all local maxima, and find the positions and amplitudes of all local minima in the input signal.
2. Create the upper envelope by spline interpolation of the local maxima and the lower envelope by spline interpolation of the local minima. Denote the envelopes $e^+(m,n)$ and $e^-(m,n)$, respectively. Type of spline depends on application.
3. For each position $(m,n)$, calculate the mean of the upper envelope and the lower envelope:

$$
\bar{e}_{lk}(m,n) = \frac{e^+(m,n) + e^-(m,n)}{2}.
$$

The signal $\bar{e}_{lk}(m,n)$ is referred to as the envelope mean.
(4) Subtract the envelope mean signal from the previous signal:
\[ h_{lk}(m, n) = h_{l(k-1)}(m, n) + \overline{e}_{lk}(m, n). \]

This is one iteration of the sifting process. The next step is to check if the signal \( h_{lk}(m, n) \) from step 4 is an IMF or not. The process stops when the envelope mean signal is close enough to zero as proposed in Ref. 2:
\[ |\overline{e}_{lk}(m, n)| < \varepsilon \quad \forall (m, n). \]

The value of \( \varepsilon \) in the stop criterion affects the EMD in such a way that if it is not small enough, then there will not be a sufficient number of IMFs to separate all intrinsic modes in the signal. On the other hand, if the number \( \varepsilon \) were too small, the iterations will take long time. A slightly more complicated version of this stop criteria is presented in Ref. 3 along with a discussion on typical values on the threshold. Forcing the envelope mean to zero will give us the wanted symmetry of the envelope and the correct relation between the number of zero-crossings and the number of extremes that define the IMF. This way we will find the IMF without actually having to check for symmetric envelopes.

(5) Check if the mean signal is close enough to zero, based on the stop criterion. If not, repeat the process from step 1 with the resulting signal from step 4 as the input signal a sufficient number of times.

When the stop criterion is met, the IMF \( c_l(m, n) \) is defined as the last result of (4):
\[ c_l(m, n) = h_{lk}(m, n). \]

After the IMF is found, define the residue as
\[ r_l(m, n) = h_{l0}(m, n) - c_l(m, n). \]

(6) The next IMF is found by starting over from step 1, now with the residue as the input signal.
\[ h_{(l+1)0}(m, n) = r_l(m, n). \]

Steps (1) to (6) can be repeated for all the subsequent \( r_j \). The EMD is completed when the residue, ideally, does not contain any extrema points. The signal can be expressed as the sum of IMFs and the last residue:
\[ x(m, n) = r_L(m, n) + \sum_{j=1}^{L} c_j(m, n). \]

### 2.1. Example Image EMD

As an example of an Image EMD we look at the well-known Lenna image, shown in Fig. 1. The four IMFs and their corresponding residues of the image are shown in Fig. 2.

It should be noted that all these IMFs are approximately zero mean, while the DC level of the signal is contained in the residue.
3. Selection of Extrema Points

In two dimensions there are many possibilities to define extrema, each one yielding a different decomposition. In our previous work\textsuperscript{2, 4–6} we simply extract the extrema points by comparing the candidate data point with its nearest eight-connected neighbors thus only finds the stationary points of the gradient image. In Ref. 7 morphologic reconstruction based on geodesic operators is used to find the extrema points.

3.1. Stationary points

Since we are only concerned with discrete 2D signals we simply extract the extrema points by comparing the candidate data point with its nearest eight-connected
neighbors. This does not allow for saddle points to be considered as extrema points. In the case where saddle points are considered to be extrema points in the algorithm, these are both maxima and minima at the same point. This nails the saddle point to the zero mean level in the first sifting round. In the following, only stationary points and not saddle points are considered to be extrema points.

3.1.1. Example

In this example we use four different ways to select extrema and use the same thin-plate interpolation method in the sifting for the IEMD. The resulting IMFs differ but the last residue is very similar. The extrema is selected using four-connected neighborhood using simple implementation, four-connected neighborhood using morphologic methods, eight-connected neighborhood using simple implementation, and eight-connected neighborhood using morphologic methods (Fig. 3).

This example shows that the choice of method for extracting the extrema points is a subject for further research.

3.2. Significant extrema points

In the first and the second IMFs, the number of extrema points usually is very large. Not all of these are essential for the signal analysis. We can reduce the number of extrema points by selecting the significant extrema points of the IMF. This is done by setting the extrema points with low amplitude to zero, as described by

$$b(m, n) = \begin{cases} 0 & \text{if } |b(m, n)| \leq T, \\ b(m, n) & \text{if } |b(m, n)| > T. \end{cases}$$

Figure 4(a) shows the extrema points of the first IMF at their position in the image where the rest of the pixels are set to zero. Figure 4(b) shows the corresponding histogram of the image in Fig. 4(a).

If we let the coefficients with absolute amplitude lower than a suitable threshold, in this case 10, be set to zero, we get the significant extrema points with respect to the threshold, shown in Fig. 5.

Keeping only the significant extrema points reduces the number of samples needed to represent the IMF, with only a minor visual effect in the reconstructed signal. However, errors will be introduced which will show up in the SNR measure.

3.3. Extrema curve

The extension of the EMD to two dimensions implies that an extrema point in one dimension extends to a line in two dimensions. In computer vision this is called ridge detection. In topography, a ridge is defined as a separator between regions from which water flows in different directions (to different sinks). Several methods can be found in Ref. 8.
Fig. 3. IEMD created from different extrema points selected with different methods. From left to right: four-connected simple, four-connected morphologic, eight-connected simple, and eight-connected morphologic; first IMF at top and residue at the bottom.
Fig. 3. (Continued)

Fig. 4. (a) The extrema points of the first IMF at their position in the image, the rest of the pixels zeroed. (b) Histogram of the image in (a).
Fig. 5. (a) The image in Fig. 4(a) with the smallest coefficients set to zero. (b) Histogram of the image in (a).

Here we present a straightforward method to find the extrema curves by using directional derivatives.

Consider an image to be represented by the function \( f(x, y) \). The first derivatives \( \partial f/\partial x \) and \( \partial f/\partial y \) give the rate of change of gray levels in the \( x \) and \( y \) directions. The rate of change in an arbitrary direction \( \theta \) is given by the linear combination of these:

\[
\frac{\partial f}{\partial x'} = \frac{\partial f}{\partial x} \cos \theta + \frac{\partial f}{\partial y} \sin \theta.
\]

Let \( \theta \) be the direction in which the extrema point curve continues from each position \( x, y \). To find the line we want to find the points of \( f(x, y) \) where the directional derivative of the image in the direction \( \theta + \pi/2 \) equals zero.

\[
\frac{\partial f}{\partial x'} = \frac{\partial f}{\partial x} \cos(\theta + \pi/2) + \frac{\partial f}{\partial y} \sin(\theta + \pi/2) = 0.
\]

For the special case of stationary points where

\[
\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0,
\]

the extrema point extends to extrema point curves in every direction (i.e. a surface). These limits to “spiders” when the search is limited to only a few directions. This effect is clearly visualized in Fig. 8(b).

In the implementation for use on digital images we use differences instead of derivatives. To avoid detection of individual noise dots the detection of extrema point curve is taken on averaging neighborhoods to the candidate point. Let \( f^{(r)}(u, v) \) denote the average gray level of \( f \) in a neighborhood of radius \( r \) centered at \( (u, v) \). The difference of such average with the candidate point \( (x, y) \) in a general direction \( \theta \) is

\[
e^{(r)}_1(x, y) = f(x, y) - f^{(r)}(x - (r + 1) \cos \theta, y - (r + 1) \sin \theta),
\]

\[
e^{(r)}_2(x, y) = f(x, y) - f^{(r)}(x + (r + 1) \cos \theta, y + (r + 1) \sin \theta).
\]
If the sign of $e_1^{(r)}(x,y)$ and $e_2^{(r)}(x,y)$ equals the candidate point is a point on the extrema point curve. The sign also indicates maximum or minimum.

In the example presented here the averaging neighborhood is two areas of $3 \times 3$ pixels and located at opposite sides of the candidate point (see Fig. 6).

3.3.1. Example

The proposed method for extrema curve detection is tested on two IMFs of the Lenna128 image. The second IMF together with its max extrema curve image and min extrema curve image are presented in Fig. 7, and the fourth IMF together with its max extrema curve image and min extrema curve image are presented in Fig. 8.

The reason choosing these images for presenting the result of the method is that these images are smooth, and thus the extrema curves are well separated and the
effects are clearly demonstrated. Note that this is not an edge detector but a way to detect the top of a ridge or the bottom of a valley.

4. Different 2D-EMD Methods

After the first experiments on EMD in two dimensions using Delaunay triangulation combined with spline interpolation in two dimensions, a couple of different methods for EMD in two dimensions have evolved.

4.1. IEMD

In image empirical mode decomposition (IEMD) we introduced the thin-plate smoothing spline interpolation for use in the implementation of the 2D-EMD. This method gives a surface with continuous second derivative everywhere. The thin-plate smoothing spline algorithm calculates the function $f$ that minimizes the integral bending norm $I_f$:

$$I_f = \int \int \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 \right) \partial x \partial y,$$

for given scattered data in the plane. The integral is taken over the entire image, and involves the second derivatives of $f$. This method turns out to successfully decompose an image into its IMFs and a smooth residue with no or only a few extrema points. However, this implementation is extremely memory consuming and slow, because the determination of the smoothing spline involves the solution of a linear system with as many unknowns as there are data points. Still, this is the method used in the examples presented in this study.

4.2. BEMD

The radial basis functions is used to solve the interpolation problem in bidimensional empirical mode decomposition (BEMD). Still it turns out that the useful radial basis is the thin plate spline, hence IEMD and BEMD are the same in principle, but with different implementations.

4.3. DEMD

Directional empirical mode decomposition (DEMD) and its application to texture segmentation are presented in Refs. 11, 12 and 17. As a form of extending 1D-EMD to the 2D case, DEMD considers the directional frequency and envelope at each point. The decomposition is done using the 1D-EMD technique applied on first the rows and then the columns of the image, including 1D interpolation of the extrema points. When one IMF is found, the image is rotated, and the search for the IMF for the next direction is performed. How the extrema points are extracted or how the interpolation is done is not mentioned. Experimental results indicate the
effectiveness of the method for texture segmentation. Although it has been argued that the 1D-EMD cannot be used on images, this result indicates that the separable approach for 2D-EMD may be worth some thought.

4.4. Other implementations

A fast algorithm is presented in Ref. 9. In Ref. 10 a method using finite elements to construct the local mean surface of the data is presented. Also there is promising work done by others on faster methods for finding the IMFs, this work was in review at the time of writing this paper.

5. Empiquency

The EMD is a truly empirical method, not based on the Fourier frequency approach but related to the locations of extrema points and zero-crossings. Based on this we use the concept of *empiquency*,\(^4\) short for empirical mode frequency, instead of a traditional Fourier-based frequency measure to describe the signal oscillations. Let \(d\) be the distance between two neighboring extrema points. The measure of empiquency is defined as:

\[
 f_e = \frac{1}{2d}.
\]

The empiquency value \(f_e\) is assigned to every position between the respective extrema points. Each extrema point influences more than one empiquency definition; we let it take the highest value of the two empiquency values it defines.

The empiquency is accompanied by an amplitude value describing the signal at the actual position. At each extrema point the empiquency amplitude \(A_e\) is the absolute value of the extrema point. At all other positions, the empiquency amplitude takes the value of the absolute mean of the extrema points that define the empiquency for the actual position. Thus,

\[
 A_e(p) = \begin{cases} 
 |a| & \text{if } p \text{ is an extrema point,} \\
 \frac{|a| + |b|}{2} & \text{if } p \text{ is not an extrema points,}
\end{cases}
\]

where \(a\) and \(b\) are values of neighboring extrema points.

The very special properties of the IMFs are that these are locally zero mean and ideally do not have more than one extrema point between two neighboring zero-crossings. In Ref. 1 the concept of *time scale* is presented. It can be seen as the mean of all \(d\) in the signal. In this context the empiquency is the *local time scale*. Because of this relationship between zero-crossings and extrema points there is also a relation between time scale, empiquency and the concept of *sequency*. Sequency is defined as “the average number of zero-crossings per second divided by 2”\(^{13}\). For a sine or cosine the frequency and the sequency is the same, but the concept of sequency has a meaning for other signals such as Walsh functions or IMFs as well.
Maximum empiquency within a signal segment is found by examining the space between the extrema points. Figures 9 and 10 show two discrete time signals, both with the properties of an IMF but with different maximum empiquency (0.5, in a normalized scale, for the signal in Fig. 9, and 0.167 for the signal in Fig. 10). When used in compression application we assume that a signal like the one in Fig. 10 can be reconstructed with low distortion from a subsampled version, whereas a signal example like the one in Fig. 9 needs every sample to be represented without distortion due to its high maximum empiquency.

6. Image HHT

It has been known for some time how to estimate instantaneous frequency in images. The method is used for motion estimation in image sequences. We call the instantaneous frequency estimation applied on IEMD Image HHT.

As in the 1D-HHT, the EMD is used to decompose the signal into sub-signals which have a meaningful interpretation of its instantaneous frequency. We calculate the Hilbert transform in two dimensions by the use of quadrature filters. A quadrature filter suppresses all negative frequencies and produces a filtered analytic signal. By introducing a direction in the filtering and using different center frequencies we obtain a number of frequency responses for each IMF. Empiquency (in the same direction as the filter) can be used for guidance to the center frequencies to use.

6.1. Theory

We here review and cite the theory presented in Ref. 14 which shows that the magnitude of the ratio between two lognormal quadrature filters can be interpreted as the instantaneous frequency of a filtered simple signal.
An estimate of local frequency can be obtained by combining instantaneous frequency estimates over a range of scales. For each scale a narrow-band instantaneous frequency estimate is obtained using ratios of lognormal quadrature filters.

The lognormal filter $F(u)$ is a spherically separable quadrature filter with a radial frequency function that is Gaussian on a logarithmic scale:

$$F(u) = R_i(\rho)D_k(\hat{u}),$$

$$R_i(\rho) = e^{-C_B \ln^2(\rho/\rho_i)},$$

where $\rho_i$ is the center frequency

$$C_B = \frac{4}{B^2 \ln 2},$$

$$B = \frac{1}{\ln 2} \ln(\rho_u/\rho_l),$$

$\rho = ||u||$ is the norm of the frequency vector. $B$ is the 6 dB relative bandwidth in octaves, and $\rho_l$ and $\rho_u$ are the values of $\rho$ for which $R_i(\rho) = 0$.

The directional function of the quadrature filters has the following form:

$$D_k(\hat{u}) = (\hat{u} \cdot \hat{n}_k) \text{ if } u \cdot \hat{n}_k > 0,$$

$$D_k(\hat{u}) = 0 \text{ otherwise},$$

where $\hat{n}_k$ is the filter directing vector.

By combining the outputs from two or more filters which differs only in center frequency, $\rho_i$, it is possible to produce a local frequency estimate. For simple frequency neighborhoods, the contribution in the Fourier domain will be concentrated at a point at a distance $\rho$ from the origin:

$$q_i = Ae^{-C_B \ln^2(\rho/\rho_i)},$$

where $A$ is the local signal amplitude.

The ratio between the outputs from two filters differing only in their center frequencies $\rho_i$ and $\rho_j$ is

$$\frac{q_j}{q_i} = \frac{e^{-C_B \ln^2(\rho/\rho_j)}}{e^{-C_B \ln^2(\rho/\rho_i)}}.$$

Simplification of this expression leads to a power relation

$$\frac{q_j}{q_i} = \left(\frac{\rho}{\sqrt{\rho_i \rho_j}}\right)^{-2C_B \ln(\rho_j/\rho_i)}.$$

Note that a quadrature filter suppresses all negative frequencies and produces a filtered analytic signal $s_{Ai}$, where $i$ indicates filter center frequency.

We write the instantaneous frequency $\omega_i$ for $s_{Ai}$ as

$$\omega_i = \frac{\partial}{\partial x} \arg(s_{Ai}),$$
where $x$ is the projection of the spatial coordinate $\xi$ on the signal direction vector $\hat{x}$

$$x = \xi \cdot \hat{x}.$$ 

Following the proof in Ref. 14 the instantaneous frequency can be written as

$$\omega_i = \sqrt{p_i p_j} \frac{\| s_{A(i+1)} \|}{s_{A_i}} = \frac{q_{i+1}}{q_i}.$$ 

With this tool for instantaneous frequency estimation in the EMD we have the Image HHT as follows. Let the image $x$ be represented by its EMD (i.e. the IMFs and the last residue):

$$x(m, n, i) \quad i = 1, \ldots, L + 1,$$

where $x(m, n, 1)$ is the first IMF $c_1(m, n)$, $x(m, n, L)$ is the last IMF $c_L(m, n)$, and $x(m, n, L + 1)$ is the last residue $r_L(m, n)$. Each position $(m, n, i)$ in this representation holds an $N$-dimensional vector valued instantaneous frequency estimation where $N$ is the number of directions searched.

6.1.1. Example

Figure 11 shows the IEMD of the straw textures, IMF1 to IMF6, while Fig. 12 shows the sixth residue of the same.
Figure 13 shows one row of the horizontal instantaneous frequency estimation of each IMF from Fig. 11, and Fig. 14 shows the same for the sixth residue. Looking at Fig. 12 we see that the residue is a tilted plane and the same conclusion can be drawn from the plot in Fig. 14.

Fig. 13. One row (no. 61) of the IHHT of each of the first six IMFs of the straw texture.
7. Applications

7.1. Variable sampling of EMD

The IEMD is a very redundant representation of an image. With the variable sampling we represent the IEMD with fewer coefficients than the original image. This work is done mainly as a preparation for an image compression scheme.

The special property of the IMF that the empiriquency varies can be used for variable sampling. Areas with many extrema points have high empiriquency, while areas with a few or no extrema points have low empiriquency. The IMFs are smoother than the image itself; only the first IMF holds the nonsmooth parts of the image. This means that it should be possible to subsample the IMFs. Due to the different empiriquencies in the different parts of the IMF, the subsampling can be different in different parts of the IMF.

It is known that a band limited signal can be uniquely determined from its nonuniform samples, provided that the average sampling rate exceeds the Nyquist rate. The extrema points define the maximum empiriquency in the IMF. Maximum empiriquency is found by examining the space between the extrema points. In the first IMF there are areas where the two neighboring pixels both are extrema points, thus the maximum empiriquency is 0.5. Letting this define our Nyquist rate, we expect that it is not possible to subsample this IMF without distortion.

Our suggestion is to treat the IMF blockwise. This way the sampling rate for each block can be defined according to its empiriquency content. The high empiriquency blocks that cannot be subsampled according to our Nyquist rate are not modified. The remaining ones are subsampled.
In the implementation of the blocking process we choose to use overlapping blocks of size $7 \times 7$ pixels. Please notice that any a priori selection of block size degrades the adaptiveness of the method. This is to minimize the artifacts from the blocking and to further reduce the samples used to represent the IMF. The corners of the block are always represented, regardless of the chosen sampling rate.

The overlapping pixels in two neighboring blocks will be the same, but used twice, as shown in Fig. 15. This will ensure that the concatenated blocks have the same values at the edge pixels. The overlapping pixels will only belong to one of the blocks when patching and counting the total number of samples.

The sampling pattern within a block consists of every pixel, every second pixel, every third pixel, and every sixth pixel in both directions, to represent $1/1$, $1/4$, $1/9$, and $1/36$ of the pixels, respectively, as shown in Fig. 15.

To find maximum empiquency we use a separable approach, analyzing the extrema points by row and column separately and choosing the maximum empiquency.

The sampling theory relies on the use of ideal filters. The use of finite signals is in contradiction to the band limitation approach. An alternative approach is to use splines. These are piecewise polynomials with pieces that are smoothly connected together at the sample points.

Fig. 15. Variable sampling with overlapping $7 \times 7$ blocks.
Considering the fact that we are treating finite signals, the use of the spline interpolator to reconstruct the image from its samples is superior to using a windowed sinc function as interpolator. The uniformly sampled points of the block are thus connected by a surface created by the use of an interpolating cubic spline extended to two dimensions.

The method of variable sampling of overlapping blocks is implemented according to the scheme in Fig. 16. Each IMF and the residue from the EMD of the image are processed in the variable sampling block.

The details of the variable sampling block are shown in Fig. 17. The extrema point image holds only the significant extrema points with respect to some size of the zero zone. Both the image itself and the extrema point image are decomposed into overlapping subblocks. The blocks of the extrema point image are used for the determination of the maximum empiquency in the block. This steers the sampling rate of the corresponding image block. In the resulting bitstream each block is marked with a two-bit header that holds information about the sampling rate.

The details of the reconstruction block are shown in Fig. 18. The two-bit header of the stream tells us not only the sampling rate but also the length of the stream for the block. With this information the coefficients can be placed in their proper

---

**Fig. 16.** Variable sampling scheme.
place in the $7 \times 7$ pixel block. The stream only provides samples for the $6 \times 6$ block; the missing samples are found in the neighboring blocks.

Reconstruction of the IMF uses the $7 \times 7$ block for block reconstruction with interpolation. The nonoverlapping $6 \times 6$ part of the reconstructed block is used when patching the blocks together to reconstruct the image.

Reconstructing the subsampled blocks is done with cubic interpolation over a regular set of samples.

7.1.1. Example

The SNR measure is computed from the original and the reconstructed image. The average sampling rate measure is a count of the number of samples used in relation to the total number of pixels in the image. The sum of samples to represent all the IMFs and the residue is 14023 corresponding to a sampling rate of 0.86. The image is reconstructed by the sum of IMFs and residue with the distortion of 31.8 dB. The result of the variable subsampling of the example in terms of sampling rate and distortion is presented in Table 1.

The results presented Figs. 19–24 show that subsampling offers a way to keep the total numbers of samples generated by EMD approximately equal to the number of pixels of the original image. The first IMF requires more samples than the others. The last IMFs and the residue reaches a limit of one sample per block and can be represented more effective by other methods. The distortion introduced comes mainly from the selection of significant extrema points.
Table 1. The result of the variable subsampling of the example in terms of sampling rate and distortion.

<table>
<thead>
<tr>
<th></th>
<th>Sampling rate</th>
<th>Distortion (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>0.86</td>
<td>31.80</td>
</tr>
<tr>
<td>IMF1</td>
<td>0.64</td>
<td>33.04</td>
</tr>
<tr>
<td>IMF2</td>
<td>0.123</td>
<td>42.31</td>
</tr>
<tr>
<td>IMF3</td>
<td>0.035</td>
<td>40.83</td>
</tr>
<tr>
<td>IMF4</td>
<td>0.029</td>
<td>50.20</td>
</tr>
<tr>
<td>Residue 4</td>
<td>0.028</td>
<td>61.63</td>
</tr>
</tbody>
</table>

Fig. 19. Reconstructed image by the sum of IMFs and residue, 31.8 dB using an average sampling rate of 0.86.

Fig. 20. IMF1 33.0414 dB using 10500 coefficients corresponding to an average sampling rate of 0.64.

7.2. Image compression

In our image compression approach we apply quantization and entropy coding on the variable sampling components. Different approaches for image compression using IEMD have been presented in Refs. 2, 4, 5 and 16. The variable subsampling of the image renders compression to some degree. This is the basis for using
Fig. 21. IMF2 42.3124 dB using 2020 coefficients corresponding to an average sampling rate of 0.123.

Fig. 22. IMF3 40.8331 dB, 576 coefficients corresponding to an average sampling rate of 0.0352.

Fig. 23. IMF4 50.2062 dB, 474 coefficients corresponding to an average sampling rate of 0.029.
the empiquency-controlled variable sampling together DCT coding of the samples (VSDCT). We have found that the first IMF is almost as hard to compress as the image itself, whereas the rest of the IMFs and all the residues are smooth and can be effectively represented by only a small part of a full-size DCT. Although the coding result of each IMF and the residue are satisfactory, the image reconstruction by the addition of the reconstructed IMFs and reconstructed residue sum the bitrates to large numbers. Here we only decompose the image into one IMF and one residue. We use the VSDCT on the first IMF and threshold coding of the full-size DCT on the first residue. The EMD is only used to find the first IMF. The empiquency of the first IMF is used to control the choice of sampling rate for the coding of the first IMF.

The structure of the variable sampling is inherited in this coding approach. The coder is shown in Fig. 25.

The details of the VSDCT block are shown in Fig. 26. The first IMF is searched for extrema points and only the significant extrema points are kept for the definition of empiquency. Both the IMF itself and the extrema point image are decomposed into overlapping 7 × 7 pixel blocks. The value of the maximum empiquency in the extrema point image block decides the sampling rate for the uniform subsampling of the corresponding IMF block. The samples are represented by one sample alone or 6 × 6, 3 × 3, or 2 × 2 blocks of samples. These are squeezed into blocks of smaller size and DCT coded. The DCT components are then quantized, threshold, and entropy coded using Huffman code or fixlength code.

The details of the reconstruction block are as follows. The two-bit header indicates the sampling rate. With this information the components achieved from the inverse DCT transform of the stream can be placed in their proper place in the 7 × 7 pixel block. The stream only contains samples for the 6 × 6 block, the missing samples are found in the reconstructed neighboring blocks. Since the missing samples are located in the rightmost column and the lowest row of the block, the reconstruction starts with the block in the lower right corner working row-wise through
the blocks. For the blocks with no neighbors holding missing samples dummy samples are used. The $7 \times 7$ blocks of samples are interpolated and the non-overlapping $6 \times 6$ part of the reconstructed block is used to generate the output IMF. The image is then reconstructed by the adding of reconstructed IMFs and the reconstructed residue which is created by inverse DCT.

7.2.1. Example

Results of coding a detail part of Lenna 512 is shown in Figs. 27 and 28.
7.3. Noise reduction

The operation of significant extrema point selection described in Sec. 3.2 also serves as a tool for noise reduction. The insignificant extrema points are defined to be noise. It is sometimes assumed that the first IMF contains all the noise in the signal and that the noise can be removed by just skipping the first IMF. Noise usually contains all empiquencies, both higher and lower than the maximum empiquency of the signal. Image processing is often performed on digitized photos. This means that any image is noisy, due to the physics of image capturing. High-quality photos, such as the Lenna image, have a low-noise level, not visible to the human eye, with an SNR higher than 40 dB. For signals with moderate SNR, we can still assume that the noise is of lower amplitude than the significant extrema points. We can then reduce the noise by setting the extrema points with low amplitude to zero in all the IMFs.
7.4. Texture analysis

To perform good texture analysis we need a feature vector describing the texture information in the image. Let the image $x$ be represented by its EMD, i.e. the IMFs and the last residue.

$$x(m, n, i) \quad i = 1, \ldots, L + 1,$$

where $x(m, n, 1)$ is the first IMF $c_1(m, n)$, $x(m, n, L)$ is the last IMF $c_L(m, n)$, and $x(m, n, L + 1)$ is the last residue $r_L(m, n)$. Each position $(m, n, i)$ in this representation holds a vector-valued feature. This feature vector can be created in different ways. Once the feature vectors are created the texture analysis is done by using standard clustering algorithms.

7.4.1. Example: Feature vector using empiquency

The concept of empiquency is described in Sec. 5. Maximum empiquency within a signal segment is found by examining the space between the extrema points. We use it on images in a separable way, examining the image in horizontal and vertical directions separately.

In the example below, the EMD is done with extrema point search using simple extraction of the extrema points by comparing the candidate data point with its nearest four-connected neighbors. With this approach we do not consider diagonal patterns. Figure 29 shows one of the Brodatz textures, Bark. The horizontal and vertical empiquency values of the first IMF is shown Fig. 31 together with the horizontal and vertical empiquency amplitude values of the first IMF in Fig. 30. These values create the $1 \times 4$-dimensional pixel feature vectors of the first IMF. This is done for all IMFs and the residue.

[Image of Bark texture]
In this section we investigate if the use of IEMD will improve the classification performance. Features are first computed with gabor filters followed by a support vector machine (SVM) for classification, second with empiquency followed by an SVM for classification, and last with image HHT followed by an SVM for classification. A set of Brodatz like textures are used for the tests.

Input data consists of 16 texture images as shown in Fig. 9. They all have the size of 128 × 128 pixels. Each texture image was decomposed using IEMD. All IEMD sets were computed in advance for all the images. The number of IMFs varied from 5 to 8. The last one is considered a residue.

A common SVM implementation is used. It used an open library LIBSVM\textsuperscript{20} version 2.32. A radial basis function (RBF) kernel is used to allow nonlinearities, which
might enhance the classifier. Two parameters control the classification. Gamma is the inverse width of the RBF kernel. The other parameter is often called C and is a regularization parameter. Varying Gamma and C systematically is used to optimize the classifier for a given set of input features.

Classification was done by pixel. Each pixel’s feature vector was assigned to a class. All textures were used in the training. A nonoverlapping set of subimages from each texture were used for training and testing. In the first part of the tests we test the Gabor feature with EMD on one image. Due to the few images available the upper half of each image was used for training and the lower part for testing. No single pixel was common to the two sets. To obtain some more samples each image part was used to get three samples, each of size 64 × 64 pixels. In the second part of the test we test all textures against the whole training set. Here the texture images were decomposed into 8 × 8 pixel blocks and a set of 10 blocks were used for training and the rest for testing. Again no single pixel was common to the two sets.

In the first tests only one texture was tested against the training set, the aim was to look at the parameters of the Gabor filter setup and the classifier. The later test looked at the classification performance of the methods for all the textures based on the numbers of IMFs used.

This test use Gabor feature with three scales four orients on one image, tested on 16 classes. We test for how many IMFs is needed for good classification. Table 2 shows the max classification rate and corresponding values for Gamma/C. We have good classification rate when using one IMF and if we use more we have 100% classification.

The Gabor filters have a lot of parameters that can be changed. Here we use a standard set for most parameters and vary only the number of directions and scales used. Table 3 shows the result. It can be noted that using only two orientations gives worse performance.

<table>
<thead>
<tr>
<th>Number of IMFs</th>
<th>Max classification rate</th>
<th>Gamma/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>99.95</td>
<td>1/100</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>1/10</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>1/3</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>1/3</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>1/1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scale and orientation</th>
<th>Max classification rate</th>
<th>Gamma/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gabor 3, 4, 3 scales 4 orients</td>
<td>100</td>
<td>100/0.0001</td>
</tr>
<tr>
<td>Gabor 3, 4, 6 scales 8 orients</td>
<td>100</td>
<td>10/0.0001</td>
</tr>
<tr>
<td>Gabor 6, 8, 2 scales 2 orients</td>
<td>93.75</td>
<td>1/10</td>
</tr>
</tbody>
</table>

Table 2. Max classification rate and Gamma/C values.

Table 3. Gabor filter parameters.
We have used IEMD for texture analysis with two different feature extraction methods. Texture analysis was achieved using the IEMD representation holding the empiquency values or the Gabor features as a vector-valued feature. Texture classification was done by using standard SVM classifier algorithm. We present the result of combining EMD and the two different feature vectors with SVM classifier into a texture analysis method in Table 4. In general the tests with using IEMD with

<table>
<thead>
<tr>
<th>Texture No.</th>
<th>3 IMF empiquency</th>
<th>4 IMF empiquency</th>
<th>3 IMF gabor</th>
<th>EMD FFT [18]</th>
<th>2 IMF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>42.83</td>
<td>52.75</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>21.52</td>
<td>14.8</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9.58</td>
<td>11.66</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>49.83</td>
<td>52.18</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>99.26</td>
<td>100</td>
<td>100</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>56.25</td>
<td>42.68</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>57.79</td>
<td>57.14</td>
<td>100</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>83.72</td>
<td>95.85</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>29.61</td>
<td>31.58</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>88.49</td>
<td>87.65</td>
<td>100</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>23.49</td>
<td>33.06</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>26.49</td>
<td>31.58</td>
<td>100</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>99.99</td>
<td>99.12</td>
<td>100</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>99.98</td>
<td>99.98</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>10.39</td>
<td>9.66</td>
<td>100</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>28.54</td>
<td>40.51</td>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Texture No.</th>
<th>3 IMF locfreq</th>
<th>4 IMF locfreq</th>
<th>2 IMF locfreq</th>
<th>3 IMF locfreq</th>
<th>4 IMF locfreq</th>
<th>2 IMF locfreq</th>
<th>FFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>33.4</td>
<td>25.2</td>
<td>16.0</td>
<td>54.4</td>
<td>54.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>99.4</td>
<td>25.7</td>
<td>11.7</td>
<td>40.8</td>
<td>40.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>29.2</td>
<td>24.7</td>
<td>13.2</td>
<td>24.5</td>
<td>24.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>15.8</td>
<td>37.7</td>
<td>96.9</td>
<td>48.3</td>
<td>48.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.0</td>
<td>32.7</td>
<td>100.0</td>
<td>94.6</td>
<td>100.0</td>
<td>100.0</td>
<td>81.0</td>
<td></td>
</tr>
<tr>
<td>6.0</td>
<td>71.7</td>
<td>64.3</td>
<td>25.8</td>
<td>54.8</td>
<td>54.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.0</td>
<td>38.2</td>
<td>64.2</td>
<td>31.4</td>
<td>99.2</td>
<td>99.2</td>
<td>99.0</td>
<td></td>
</tr>
<tr>
<td>8.0</td>
<td>86.2</td>
<td>95.9</td>
<td>54.7</td>
<td>84.9</td>
<td>84.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.0</td>
<td>81.5</td>
<td>47.0</td>
<td>6.9</td>
<td>85.3</td>
<td>85.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.0</td>
<td>69.7</td>
<td>32.2</td>
<td>62.1</td>
<td>66.5</td>
<td>66.5</td>
<td>82.0</td>
<td></td>
</tr>
<tr>
<td>11.0</td>
<td>95.1</td>
<td>72.0</td>
<td>86.7</td>
<td>82.7</td>
<td>82.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.0</td>
<td>73.8</td>
<td>100.0</td>
<td>10.0</td>
<td>54.2</td>
<td>54.2</td>
<td>97.0</td>
<td></td>
</tr>
<tr>
<td>13.0</td>
<td>98.3</td>
<td>27.5</td>
<td>11.9</td>
<td>75.5</td>
<td>75.5</td>
<td>74.0</td>
<td></td>
</tr>
<tr>
<td>14.0</td>
<td>68.6</td>
<td>31.1</td>
<td>49.1</td>
<td>34.7</td>
<td>34.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.0</td>
<td>98.3</td>
<td>94.5</td>
<td>21.2</td>
<td>95.0</td>
<td>95.0</td>
<td>81.0</td>
<td></td>
</tr>
<tr>
<td>16.0</td>
<td>91.2</td>
<td>100.0</td>
<td>97.7</td>
<td>100.0</td>
<td>100.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Gabor was much better than IEMD with empiquency. One possible explanation might be that the number of orientations used with Gabor was three. For the empiquency feature we test only in two directions with three or four IMFs. With the empiquency setup we do not have 100% classification rate more than occasionally. In both empiquency tests only five textures have classification rate over 80%. For reference we show the max classification rate from an EMD FFT method presented in Ref. 18 for the textures that are the same in the sets.

The image HHT is introduced as a feature extraction method for texture classification. Features are first computed with Image HHT followed by an SVM for classification. A common SVM implementation is used for the classification. A set of Brodatz-like textures are used for the tests. Results presented in Table 5 show that using four orientations are better than using two orientations and minimum three IMFs should be used.

For reference we show the max classification rate from an EMD-FFT method presented in Ref. 18 for the textures that are the same in the sets.

8. Open Problems

This study is full of ideas and work yet to be done or that needs improvement before the EMD is really useful in image processing. It is presented as an inspiration for further work and the questions are free for everyone to search the answer to. The questions below are some examples to work with:

- Find a fast implementation of 2D-interpolation that is good enough for 2D-EMD!
- Define (mathematically) the criteria for a surface to be good enough for IEMD.
- Mode mixing is typical for EMD and is considered as a problem. Instead, use the mode mixing as a feature and develop unique image processing methods!
- Which is the best way to extract extrema?
- Are the different methods of extracting extrema good for different applications?
- Develop the extrema curve concept and find a way to interpolate.
- Show that DEMD is a proper way to treat images.
- Can we use the math of sequency\textsuperscript{13} for EMD?
- Selected papers and Matlab code can be found at http://go.to/imageemd.

References

2. A. Linderhed, 2D empirical mode decompositions in the spirit of image compression, in Wavelet and Independent Component Analysis Applications IX (Orlando Fl, 2002).


