PERFECT RECONSTRUCTABLE DECIMATED TWO-DIMENSIONAL EMPIRICAL MODE DECOMPOSITION FILTER BANKS

Min-sung Koh and Esteban Rodriguez-Marek

School of Computing and Engineering Sciences
Eastern Washington University
Cheney WA 99004, USA

ABSTRACT

Traditional two-dimensional empirical mode decomposition (2D-EMD) algorithms generate multiple subband signals, each having the same size of the original signal. Thus, huge amounts of data to be stored may be generated. Moreover, the computational load is massive as the decomposition levels increase. This paper introduces a method to reduce the data generated (i.e. reduce storage requirement) by incorporating decimation into the 2D-EMD, while maintaining perfect reconstruction. Furthermore, it is well established that traditional EMDs can be thought as having the structure of a single dyadic filter bank. The proposed algorithm is applicable into *any* arbitrary tree structures including octave filter banks, 2D-EMD packets when applied to a full binary tree, etc. The methodology hereby presented builds on the algorithm introduced by the authors in [8].

Index Terms— Empirical Mode Decomposition, EMD, 2D-EMD, Filter banks, Decimated, Perfect reconstruction.

1. INTRODUCTION

The original 1D-EMD proposed in [1] is a time-domain decomposition that results in the various frequency components of the signal separated into what is denominated intrinsic mode functions (IMFs). The decomposition is done without the need of traditional filtering (i.e. convolution). Instead, each IMF is obtained simply by subtracting the average signal of two envelopes (upper and lower envelopes) from the original signal. The two envelopes are obtained by performing time-domain interpolation of the maxima and minima of the signal to be decomposed. This simple process is continued either until a stop condition is fulfilled or until the last signal, called the residual, becomes a monotonic function. The flexibility displayed by the 1D-EMD in analyzing non-stationary and non-linear signals, resulted in researchers extending it into a two dimensions, such as in [2,3,4]. Various applications, such as in image compression and image fusion [5,6], etc. have been also reported. However, traditional 1D and 2D-EMDs yield IMFs (and residual) of the same size as the original signal to be decomposed. Hence, large amounts of data are generated, thereby requiring many times forbidding storage capacity and computational loads. Furthermore, it was shown in [7] that the EMD behaves, in essence, like a dyadic filter bank similar to those in wavelet transforms. Other tree structures have not been considered at all for EMDs, due to the nature of IMFs and residuals. In an effort to eliminate the drawbacks of the EMD (data expansion and fixed tree structure), the authors of this paper developed a 1D-EMD algorithm that allows for decimation, while still permitting perfect reconstruction for any arbitrary tree structure, and maintaining the desirable characteristics of traditional EMD [8]. In this paper, we extend the algorithm in [8] to design 2D-EMD filter banks. The analysis and synthesis portions of the proposed filter banks are introduced in Sections 2 and 3, respectively. Data reduction analysis is performed in Section 4 and experimental results are presented in Section 5. Conclusions follow in Section 6.

2. 2D-EMD ANALYSIS FILTER BANKS

As shown in [7], a traditional EMD can be thought as a dyadic filter bank. To make the 2D-EMD applicable to any arbitrary tree structure, only *one* IMF and *one* residual are considered at each node's subimage. In other words, borrowing the ideas from [8], only one residual and the first IMF are obtained at each decomposition step. If it is desired to grow the tree, then further decompositions of the residual and the first IMF can be done through the same process. Thus, any arbitrary tree can be generated. This, however, does not fix the size problem, as straight downsampling here would no longer yield perfect reconstruction. Hence, we call this structure "*undecimated* EMD filter banks," borrowing terminology from wavelet theory. It is clear that the data set resulting from decomposition increases with every level. To reduce the storage space required, a single stage of the analysis filter bank including the proposed decimation technique is shown in Fig. 1. Although any non-separable 2D-EMD implementation can be used, that of [4] was chosen to generate the results in this paper. Unlike traditional 2D transforms, which process columns first and rows next, using separable 1D-transforms, the algorithm in [4] does both dimensions simultaneously using any 2D inpainting interpolation algorithm. In Fig. 1, Z_r^{-1} implies a row-wise delay operation and Z_c^{-1} implies the same operation for columns. R and I denote the residual and the first IMF obtained by the 2D-EMD, respectively. The encircled ↓2×2 symbol implies decimation by two is applied both to rows and columns. Hence, the paths having $Z_r^{-1}Z_c^{-1}$ followed by ↓2x2 implies that only the odd indexed elements, both in the row and column directions, are allowed through. The outputs of these operations are denoted R_{oo} and I_{oo}. Similarly, R_{ee}/I_{ee}, R_{eo}/I_{eo}, and R_{oe}/I_{oe} imply (even, even) , (even, odd), and (odd, even) indexed elements only, respectively.

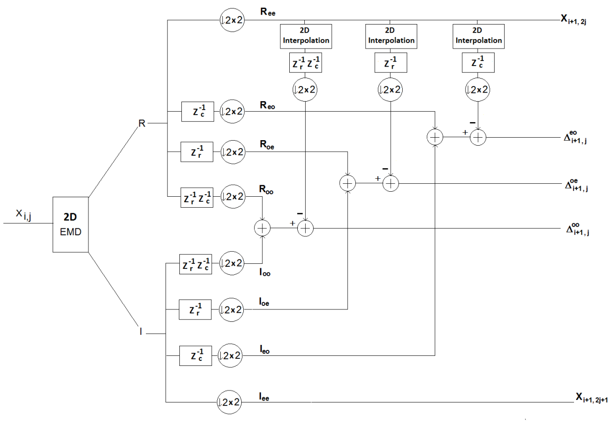

Fig. 1. One stage of analysis filter banks in decimated 2D-EMD filter banks proposed in this paper.

Mathematically, assuming $X_{i,j}$ is the $M \times N$ signal to be decomposed at node (i, j), we express these signals as

$$X_{i,j}[m,n] = R[m,n] + I[m,n] \qquad (1)$$

where $X_{i,j}, R, and\ I \in \Re^{M\times N}$ and m=0,1,…,M-1 and n=0,1,…, N-1, and

$$R_{ee}[k,r] = R[2k,2r] \quad , \text{ where } k = 0,1,\dots,\left\lceil\frac{M}{2}\right\rceil - 1 \text{ and } r = 0,1,\dots,\lceil N/2\rceil - 1 \qquad (2)$$

$$R_{eo}[k,r] = R[2k,2r+1] \quad , \text{where } k = 0,1,\dots,\left\lceil\frac{M}{2}\right\rceil - 1 \text{ and } r = 0,1,\dots,\lfloor N/2\rfloor - 1 \qquad (3)$$

$$R_{oe}[k,r] = R[2k+1,2r] \quad , \text{ where } k = 0,1,\dots,\left\lfloor\frac{M}{2}\right\rfloor - 1 \text{ and } r = 0,1,\dots,\lceil N/2\rceil - 1 \qquad (4)$$

$$R_{oo}[k,r] = R[2k+1,2r+1], \text{where } k = 0,1,\dots,\left\lfloor\frac{M}{2}\right\rfloor - 1 \text{ and } r = 0,1,\dots,\lfloor N/2\rfloor - 1 \qquad (5)$$

The first IMF signals, denoted by I_{oo}, I_{ee}, I_{eo}, and I_{oe} are obtained by using a similar set of equations. Intermediate signals denoted by $S_{ee}[k,r]$, $S_{eo}[k,r]$, $S_{oe}[k,r]$, and $S_{oo}[k,r]$, can be obtained by the summations of downsampled versions of R and I as expressed by

$$\begin{cases} S_{ee}[k,r] = R_{ee}[k,r] + I_{ee}[k,r] \\ S_{eo}[k,r] = R_{eo}[k,r] + I_{eo}[k,r] \\ S_{oe}[k,r] = R_{oe}[k,r] + I_{oe}[k,r] \\ S_{oo}[k,r] = R_{oo}[k,r] + I_{oo}[k,r] \end{cases} \qquad (6)$$

where $S_{ee}[k,r]$ is the (even, even) indexed subimage of $X_{i,j}[m,n]$ and so on. Subimages denoted by $S_{eo}[k,r]$, $S_{oe}[k,r]$, and $S_{oo}[k,r]$ can be found in Fig. 1 at the output of the summation operator fed with R_{eo} and I_{eo}, R_{oe} and I_{oe}, R_{oo} and I_{oo}, respectively. Five signals are generated from the transform from Fig. 1: R_{ee} and I_{ee} (which become a new node in the tree), and three error signals, $\Delta^{eo}_{i+1,j}$, $\Delta^{oe}_{i+1,j}$, and $\Delta^{oo}_{i+1,j}$. For further decomposition, we define the new nodes as $X_{i+1,2j}[k,r] = R_{ee}[k,r]$ and $X_{i+1,2j+1}[k,r] = I_{ee}[k,r]$, for i=0, 1, 2, ….., L and j=0, 1, 2, …., $2^L - 1$, where L is the number of decomposition levels. The error images are expressed by

$$\begin{cases} \Delta^{eo}_{i+1,j}[k,r] = S_{eo}[k,r] - \widehat{S_{eo}}[k,r] \\ \Delta^{oe}_{i+1,j}[k,r] = S_{oe}[k,r] - \widehat{S_{oe}}[k,r] \\ \Delta^{oo}_{i+1,j}[k,r] = S_{oo}[k,r] - \widehat{S_{oo}}[k,r] \end{cases} \qquad (7)$$

The estimates in equation (7) are obtained by 2D interpolations and proper downsampling, as shown in Fig. 1. Thus, the residual and the first even-indexed IMF are downsampled and three error signals are formed through merging. All error signals are stored, as they will be needed for perfect reconstruction. All outputs will have dimensions that are half the size of the original signal, i.e. each output is an $M/2\times N/2$ image. Note that when cascading the stage in Fig. 1 into multiple stages, $X_{i+1,2j}[m]$ and $X_{i+1,2j+1}[m]$ are nodes that can be further decomposed into any desired tree structure. Hence, the structure of Fig. 1 forms one stage of the analysis part of what we denominate "*decimated* 2D-EMD filter banks." Further, note that intermediate signals R_{eo}/I_{eo}, R_{oe}/R_{oe}, R_{oo}/I_{oo}, S_{eo}, S_{oe}, and S_{oo} are all *internal* signals, and need not to be stored.

In order to make decimated filter banks, the stage in Fig. 1 can be cascaded through any arbitrary tree structure. For example, the tree shown in Fig. 2, where the tree structure can be expressed by the end nodes, i.e. (3,0), (3,1), (2,1), and (1,1), where they are ordered from low to high frequency. This is similar to wavelets used to make an octave filter bank. In Fig. 2, the output denoted by $\Delta_{i+1,j}$ corresponds to all error images generated by the system in Fig. 1, and is expressed by

$$\Delta_{i+1,j} = [\Delta^{eo}_{i+1,j} \mid \Delta^{oe}_{i+1,j} \mid \Delta^{oo}_{i+1,j}] \quad \in \Re^{\frac{M}{2}\times\frac{3N}{2}} \qquad (8)$$

It can be observed, then, that the proposed decimated 2D-EMD filter bank has a similar structure to the analysis stage of the 1D-EMD introduced in [8], but it has three rather than one error signal.

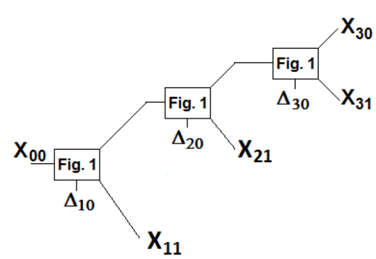

Fig. 2. A decimated EMD octave filter banks made by the proposed algorithm

Again, note that all $X_{i,j}$ in Fig. 2 correspond to R_{ee} or I_{ee}, i.e. they have only the even samples (in both dimensions) of the parent node signal, $X_{i-1,k}$. The filter bank in Fig. 2 decomposes X_{00} into four subimages, X_{11}, X_{21}, X_{30}, and X_{31}, with sizes $M/2\times N/2$, $M/4\times N/4$, $M/8\times N/8$ and $M/8\times N/8$, respectively. It is easy to infer that in general, for decomposition level l, the size of the image will be $M/2^l \times N/2^l$. If index j in $X_{i,j}$ is even (or zero), then it corresponds to R_{ee}, otherwise it is I_{ee}. In Fig. 2, each error signal also gets reduced in size with every decomposition level, l, but in this case the size is $M/2^l \times 3N/2^l$, as there are three components, each corresponding to one Δ. It seems counterintuitive, but despite the algorithm generates more signals (i.e. the error signals), it results in a significant reduction in storage capacity with respect both to the undecimated case and the original 2D-EMD. A more thorough analysis of total capacity needed by the algorithm will be presented in Section 4.

3. 2D-EMD SYNTHESIS FILTER BANKS

A single synthesis stage of the decimated 2D-EMD filter bank is shown in Fig. 3. It is basically the reverse process of the analysis stage, but the final signal recovery is achieved through the sum of the residual and IMFs at each decomposition step. Error images of $\Delta^{eo}_{i+1,j}$, $\Delta^{oe}_{i+1,j}$, and $\Delta^{oo}_{i+1,j}$ are added back to the estimated residual by going through 2D interpolation and downsampling. It is important to ensure the proper row and column indices of each error image are used, in order to recover the downsampled subimages, $S_{eo}[k,r]$, $S_{oe}[k,r]$, and $S_{oo}[k,r]$ from equation (6). The low resolution reference residual, $X_{i+1,2j}[m]$ (i.e. R_{ee}), is added back to the low resolution version of the first IMF, $X_{i+1,2j+1}[m]$ (i.e. I_{ee}), to obtain reference subimage $S_{ee}[k,r]$. All recovered subimages are upsampled (shown as ↑2x2 in Fig. 3) and shifted back to be on the correct row and column indices by using the appropriate shift operation (i.e. one of Z_c, Z_r, Z_rZ_c). Hence, all recovered subimages (i.e. $S_{ee}[k,r]$, $S_{eo}[k,r]$, $S_{oe}[k,r]$, and $S_{oo}[k,r]$) are weaved back correctly to recover the original upper image, $X_{i,j}[m,n]$. Note that estimates $\widehat{S_{eo}}[k,r]$, $\widehat{S_{oe}}[k,r]$, and $\widehat{S_{oo}}[k,r]$ stem from the reference R_{ee}. Clearly, the same 2D interpolation method used in the analysis *must* be used in the synthesis if perfect reconstruction is desired. To perform the synthesis operation of the whole system, single synthesis stages should be cascaded in the same tree structure as in Fig. 2, but operated from the end nodes of the tree to the root. Perfect reconstruction is proved by the following theorem.

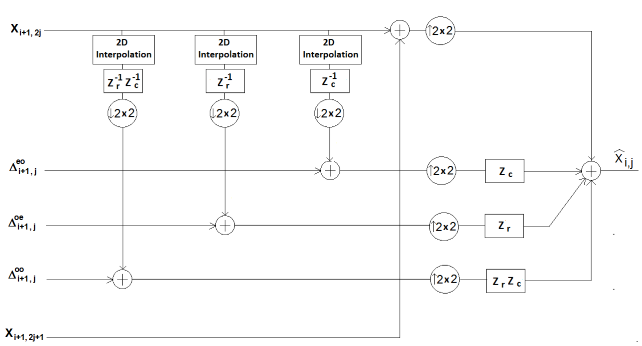

Fig. 3. One stage of synthesis part in decimated 2D-EMD filter banks proposed in this paper.

Theorem: *The decimated 2D-EMD filter bank shown in Fig. 1 and Fig. 3 form perfect reconstructable filter banks with aliasing cancelling as long as identical 2D-interpolation techniques are applied in synthesis and analysis.*

Similar proof in [8] can be extended into 2D-EMD filter banks. However, the proof is trivial when we notice that if we use the same interpolation technique in both analysis and synthesis, the combination of the systems from Figs. 1 and 3 are equivalent to the perfect reconstructable filter banks of Figure 12.9-5(b) in [9, pp. 634], having the diagonal decimation matrix M=2I.■

4. DATA REDUCTION RATIO OF THE PROPOSED ALGORITHM

The proposed algorithm produces two children per node, each at one quarter of the size of the parent. However, it also generates three error signals, each also of size $M/2 \times N/2$. Thus, the outputs of each stage add to 1.25 $M \times N$. Adding the size of all children in a traditional 2D wavelet filter bank results in $4 \times M/2 \times N/2 = M \times N$, i.e. more coefficients are required for the decimated 2D-EMD than for the wavelet transform. The amount of coefficients needed to be saved for the decimated 2D-EMD varies with the number of decomposition levels, L. More specifically, note that for a full binary tree there are 2^L nodes in the tree, with each one having a subimage of size $M/2^L \times N/2^L$. Further, every level l has 2^{l-1} pairs of R_{ee} and I_{ee}, each pair generating error data of size $M/2^l \times 3N/2^l$. Thus, we need to store

$$2^L \times \frac{M}{2^L} \times \frac{N}{2^L} + \sum_{l=1}^{L} 2^{l-1}\left(\frac{M}{2^l} \times \frac{3N}{2^l}\right) = \frac{MN}{2^L} + \frac{3MN}{2}\left(1 - \frac{1}{2^L}\right)$$
$$= MN\left(\frac{3}{2} - \frac{1}{2^{L+1}}\right) < \frac{3}{2}MN$$

coefficients. A full binary tree using the wavelet transform requires simply $M \times N$ coefficients. Thus, the decimated 2D-EMD needs at most 1.5 times more coefficients. If, say, the octave decomposition tree structure is used, it is easily proved that the number reduces to roughly 1.3.

The traditional 2D-EMD decomposition, i.e. without decimation, generates $(L+1)(M \times N)$ total data, resulting from L IMFs and one residual, each of size $M \times N$. If we use one residual and one IMF and use it to generate an L-level full binary tree structure, then a total of $2^L(M \times N)$ data is generated (for the undecimated 2D-EMD filter banks), as 2^L nodes will occur. This is an upper bound, as arbitrary tree structures may not extend to a full tree. The reduction ratio for using the decimated version, thus, is found with

$$RR = \frac{2^L(MN)}{\frac{MN}{2^L} + \frac{3MN}{2}\left(1 - \frac{1}{2^L}\right)} = \frac{2^L}{\frac{1}{2^L} + \frac{3}{2}\left(1 - \frac{1}{2^L}\right)} \qquad (9)$$

For example, for an L=5 full binary tree, equation (9) yields $RR \approx 21.6$, i.e. we need to store roughly 21.6 times less data than in the undecimated 2D-EMD case. If, say, an L=5 octave tree is used, it can be proved that RR drops to approximately 4.5, as there are less intermediate nodes in the tree generating error data.

5. EXPERIMENTAL RESULTS

The proposed algorithm has been tested for several arbitrary structures. Perfect reconstruction has been achieved, as demonstrated by PSNRs above 320 dB, implying only slight floating point errors. Fig. 4 compares the proposed algorithm with traditional 2D-EMD having a dyadic filter banks. The first column of Fig. 4 shows traditional 2D-EMD using [4] with 3 IMFs and a residual, where the first, second and third IMFs and the residual are ordered from top to bottom. The size of all IMFs and the residual is identical to the original image, 256×256 (i.e., no decimation). The second column shows the decimated versions of the first column, with a downsampling factor of 2^l both in row and column directions, where l=1, 2, 3 for each decomposition level (i.e., the order of IMF). Thus, the images sizes of the second column are $256/2^l \times 256/2^l$. The proposed algorithm's images are shown in the third column. Note that the images in the second and third columns are very similar, from which we can infer that the tree structure of traditional 2D-EMD with downsampling is similar to that of the proposed algorithm. However, as mentioned before, simple downsampling of traditional 2D-EMD is *not* perfectly reconstructable and it has no option to adapt to diverse tree structures. The proposed algorithm has many advantages for data reduction while keeping perfect reconstruction in any arbitrary tree structures.

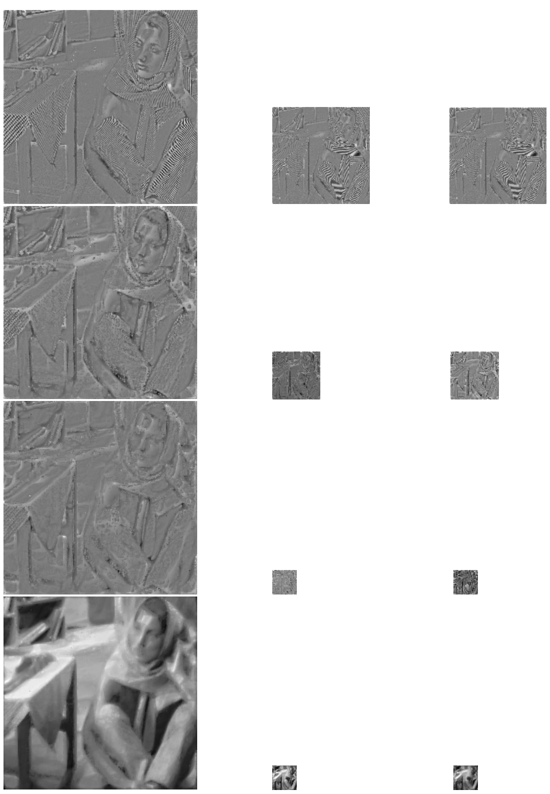

Fig. 4. Comparison the proposed algorithm with traditional 2D-EMD in Fig. 2 tree structure.

Fig. 5 shows all error images of $\Delta_{1,0}^{eo}$, $\Delta_{1,0}^{oe}$, $\Delta_{1,0}^{oo}$ and $X_{1,1}$ (i.e., I_{ee}) for node (1,0) in Fig. 2 on (left upper), (right, upper), (left, lower), and (right lower) corners of the image. As expected, all error images are very similar to each other because all images have high frequency components from the original image compared to the residual of R_{ee}. Hence, it generates somewhat redundant information, although all error information is needed for perfect reconstruction. In order to compare with traditional wavelet filter banks, we use a full binary tree structure with L=2. The images are combined into one and shown in Fig. 6, together with the correct mapping of nodes. Since all error images and $X_{i,2j+1}$ (i.e., I_{ee}) correspond to similar highpass information, $X_{i,2j+1}$ is used instead of $\Delta_{i,j}^{oo}$. From Figs. 5 and 6 it is apparent that all error image and I_{ee} correspond to combined versions of LH, HL, and HH bands in a traditional wavelet transform, i.e. they include horizontal, vertical and diagonal high frequency information.

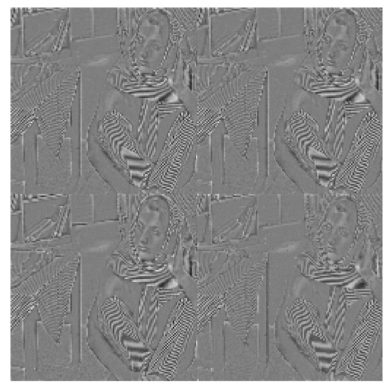

Fig. 5. All error images and *Iee* (i.e., $X_{1,1}$) images for node (1,0) in Fig. 2.

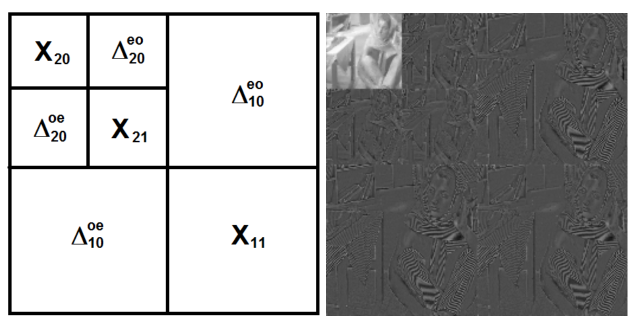

Fig. 6. Combined image using the node images of the proposed decimated 2D-EMD filter banks for the tree structures of given nodes, (2,0), (2,1), (2,2), (2,3).

6. CONCLUSIONS

This paper presents an extension of the algorithm introduced by the authors in [8] to incorporate decimation into the traditional 2D-EMD, without eliminating the perfect-reconstruction property. Moreover, the algorithm permits the generation of any arbitrary tree, while displaying good data-reduction ratios with respect to traditional 2D-EMD algorithms. The algorithm can use any 2D-EMD implementation, as long as it is not based on separable filter banks.

7. REFERENCES

[1] N.E. Huang, Z Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N-C Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and Hilbert spectrum for nonlinear and non-stationary time series analysis," in the *Proc. Royal Society London*, Ser. A 454, pp. 903–995, 1998.

[2] C. Damerval, S. Meignen, and V. Perrier, "A fast algorithm for bidimensional EMD" *IEEE Sig. Proc. Letter*, Vol. 12, no. 10, pp. 701- 704, Oct. 2005.

[3] E. Hadji, S. Diop, R. Alexandre, A.-O Boudraa, "A PDE model for 2D intrinsic mode functions", *IEEE Int. Conf. on Image Proc. (ICIP),* Jul. 2009.

[4] M. S. Koh, E. Rodriguez-Marek, and T. R. Fischer, "A new two dimensional empirical mode decomposition for images using inpainting", in the *IEEE International Conference on Signal Processing (ICSP 2010),* Beijing China, Oct. 2010.

[5] A. Linderhed, "Compression by image empirical mode decomposition", *IEEE Int. Conf. on Image Proc (ICIP).* Sept. 2005.

[6] L. Qiao, L. Peng, W. Guo, and W. Yuan, "A novel image fusion algorithm based on 2D EMD and IHS", *Proc. of Int. Conf. on Machine learning and Cybernetics*, Jul. 2008 .

[7] P. Flandrin, G. Rilling, and P. Goncalves, "Empirical mode decomposition as a filter bank", *IEEE Sig. Proc. Letters,* vol. 11, no. 2, pp. 112-114, Feb. 2004.

[8] M. S. Koh and E. Rodriguez-Marek, "Perfect reconstructable decimated one-dimensional empirical mode decomposition filter banks", submitted *into the 3rd International Conference on Signal, Image Processing and Applications(ICSIA)*, Barcelona, Spain, Aug, 2013

[9] P.P. Vaidyanathan, "*Multirate systems and filter banks*", Prentice Hall Inc, Upper Saddle River, NJ, 1993