

國立交通大學

資訊科學與工程研究所

碩士論文



中華民國 107 年 7 月

空間頻率擷取使用類梯度運算子
Spatial Frequency Extraction using Gradient-like Operator

研究 生：林以鎬

Student: Yi-Hao Lin

指導 教授：蕭子健

Advisor: Tzu-Chien Hsiao

國立交通大學
資訊科學與工程研究所
碩士論文



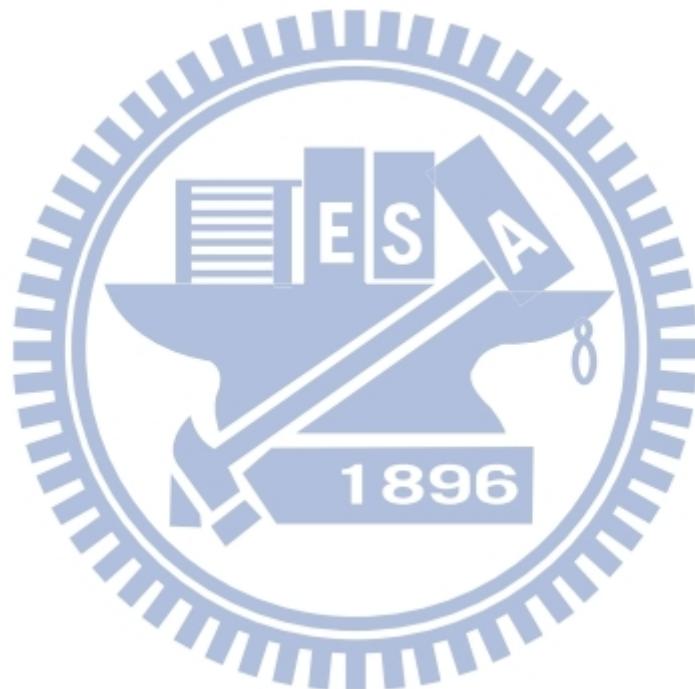
July 2018

Hsinchu, Taiwan, Republic of China

中華民國 107 年 7 月

摘要

多維度集合經驗模態拆解 (**MEEMD**) 常用於拆解影像，但其時間複雜度較高，在這裡介紹一個新方法：「空間頻率擷取使用類梯度運算子」，將運算時間減少十倍以上，且很適合平行化運算，在 **GPGPU** 版本中，將近有 500 倍的效能提升！我們的新穎方法使用類梯度運算子評估不同半徑下的空間頻率，並且將梯度運算的結果積分到空間幀，其結果很類似於 **MEEMD** 的成果。



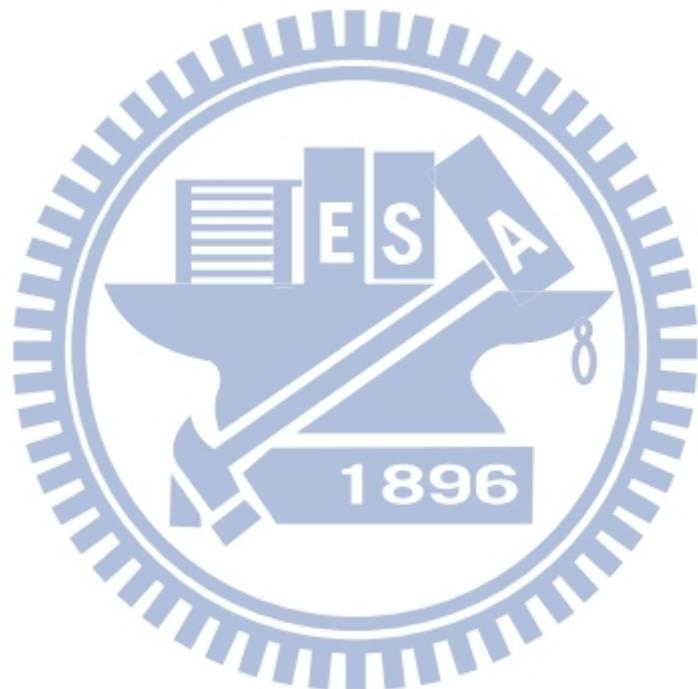
Abstract

The multi-dimensional ensemble empirical mode decomposition (MEEMD) is usually used for temporal-spatial data decomposition. One of the major issue is its high time complexity. A new gradient-liked approach to mimic similar spatial data decomposition results with more than 10x speedup. The GPGPU version of our approach can reach 500x speedup. Our novel approach use gradient-liked operator to evaluate the spatial frequency on different radius and integral the gradient result to spatial frame which is similar to BIMFs of MEEMD.



致謝

首先我要感謝我的父母，在研究所期間鼓勵我，幫助我完成研究所的學業，再來我要感謝我的靈魂朋友們，因為許多研究靈感都是在睡覺時星光體翱翔時和靈魂朋友們討論，並在回到身體時半夢半醒間獲得靈感！以及要感謝蕭子健老師花了很多時間和我討論我的研究內容。也要感謝羅佩禎老師及范倫達老師擔任口試委員，提供許多寶貴的意見。最後謝謝實驗室的夥伴們，互相鼓勵及討論研究內容。



摘要.....	i
Abstract.....	ii
致謝.....	iii
Index	iv
List of Figure.....	v
List of Table	vii
List of Abbreviation	viii
1. Introduction.....	1
1.1. Background	1
1.2. Motivation.....	4
1.3. Literature Study.....	4
1.4. Objective	4
2. Material and Method.....	5
2.1. Procedure.....	6
2.2. Choose Radius.....	7
2.3. Gradient-like Operator	7
2.3.1. Pixel Classifier	7
2.3.2. Analyze Direction	8
2.3.3. Analyze Magnitude	9
2.4. Projection	9
2.5. OpenCL implementation	10
3. Result	11
3.1. Decomposition of a square border	12
3.2. Decomposition of a different spatial frequency 2D signal.....	14
3.3. Decomposition of Lena	16
3.3. Decomposition of Squirrel	18
3.3. Decomposition of London Bridge.....	20
4. Discussion.....	22
4.1. Circular mask v.s. square mask	22
4.2. Direction error on small radius circular mask	23
4.2.1. Up sampling for fixing direction error.....	23
4.3. Performance	24
4.3.1. Down sampling for acceleration	25
4.3.2. Real-Time decomposition	25
4.4. MEEMD Emulation	26
5. Conclusion	28
6. Further work.....	29
7. Reference	30

List of Figure

FIGURE 1 EMD PROCEDURE.....	1
FIGURE 2 EEMD PROCEDURE.....	2
FIGURE 3 BEMD PROCEDURE	2
FIGURE 4 BIMFS OF LENA BY BEMD.....	2
FIGURE 5 MEEMD PROCEDURE AND EXAMPLE.....	3
FIGURE 6 BIMFS OF LENA BY MEEMD	3
FIGURE 7 SFEGO PROCEDURE AND EXAMPLE	6
FIGURE 8 SFEGO RADIUS	7
FIGURE 9 PROCEDURE OF GRADIENT-LIKED OPERATOR.....	7
FIGURE 10 PXIEL CLASSIFIER ON CIRCULAR MASK RADIUS R=2	7
FIGURE 11 SLIDING WINDOWS SEQUENCE, SLIDING WINDOW WEIGHT AND SLIDING WINDOW VALUE	8
FIGURE 12 RESULT OF DIRECTION BY GRADIENT-LIKED OPERATOR.....	8
FIGURE 13 LARGEST SLIDING WINDOW AND OPPOSITE SLIDING WINDOW	9
FIGURE 14 RESULT OF MAGNITUDE BY GRADIENT-LIKED OPERATOR.....	9
FIGURE 15 MAGNITUDE FRAME, DIRECTION FRAME AND ANGLE PHI	9
FIGURE 16 RESULT OF SPATIAL FRAME BY SFEGO	10
FIGURE 17 OPENCL COMPUTING DIAGRAM.....	10
FIGURE 18A COMBINE A SQUARE BORDER WITH SINUSOIDAL BACKGROUND	12
FIGURE 18B DECOMPOSITION OF A SQUARE BORDER	12
FIGURE 18C CORRELATION COEFFICIENT BETWEEN SFEGO AND MEEMD ON A SQUARE BORDER	12
FIGURE 18D THE FFT (BOTTOM) OF SFEGO DECOMPOSITION RESULT (TOP) OF SQUARE BORDER AND THE INVERSE FFT (MIDDLE) WHICH IS FILTER ON FREQUENCY DOMAIN	13
FIGURE 18E THE DIAGRAM OF FFT RESULT ON SQUARE BORDER.....	13
FIGURE 19A FUNCTION F IS F1 ADD F2.....	14
FIGURE 19B DECOMPOSITION OF F	14
FIGURE 19C CORRELATION COEFFICIENT BETWEEN SFEGO AND MEEMD ON F	14
FIGURE 19D THE FFT (BOTTOM) OF SFEGO DECOMPOSITION RESULT (TOP) OF F AND THE INVERSE FFT (MIDDLE) WHICH IS FILTER ON FREQUENCY DOMAIN	15
FIGURE 19E THE DIAGRAM OF FFT RESULT ON F.....	15
FIGURE 20A IMAGE OF LENA	16
FIGURE 20B DECOMPOSITION OF LENA	16
FIGURE 20C CORRELATION COEFFICIENT BETWEEN SFEGO AND MEEMD ON LENA	16
FIGURE 20D THE FFT (BOTTOM) OF SFEGO DECOMPOSITION RESULT (TOP) OF LENA AND THE INVERSE FFT (MIDDLE) WHICH IS FILTER ON FREQUENCY DOMAIN.....	17
FIGURE 20E THE DIAGRAM OF FFT RESULT ON LENA	17

FIGURE 21A IMAGE OF SQUIRREL.....	18
FIGURE 21B DECOMPOSITION OF SQUIRREL.....	18
FIGURE 21C CORRELATION COEFFICIENT BETWEEN SFEGO AND MEEMD ON SQUIRREL	18
FIGURE 21D THE FFT (BOTTOM) OF SFEGO DECOMPOSITION RESULT (TOP) OF SQUIRREL AND THE INVERSE FFT (MIDDLE) WHICH IS FILTER ON FREQUENCY DOMAIN	19
FIGURE 21E THE DIAGRAM OF FFT RESULT ON SQUIRREL.....	19
FIGURE 22A IMAGE OF LONDON BRIDGE	20
FIGURE 22B DECOMPOSITION OF LONDON	20
FIGURE 22C CORRELATION COEFFICIENT BETWEEN SFEGO AND MEEMD ON LONDON BRIDGE	20
FIGURE 22D THE FFT (BOTTOM) OF SFEGO DECOMPOSITION RESULT (TOP) OF LONDON BRIDGE AND THE INVERSE FFT (MIDDLE) WHICH IS FILTER ON FREQUENCY DOMAIN	21
FIGURE 22E THE DIAGRAM OF FFT RESULT ON LONDON BRIDGE	21
FIGURE 23 CIRCULAR MASK V.S. SQUARE MASK	22
FIGURE 24 FRAGMENT ON SMALL RADIUS DECOMPOSITION.....	23
FIGURE 25 UP SAMPLING DIAGRAM	23
FIGURE 26 UP SAMPLING RESULT OF DECOMPOSITION F AND LENA	24
FIGURE 27 DOWN SAMPLING DIAGRAM AND DOWN SAMPLING SPEEDUP TABLE	25
FIGURE 28 REAL-TIME WEBCAM DECOMPOSITION	25
FIGURE 29 5 PICTURE FOR FINDING RADIUS SEQUENCE. LENA, 2MEN, SCIURIDAE, TOWER AND BRIDGE	26
FIGURE 30 THE DECOMPOSITION AND AMPLITUDE OF LENA WITH SFEGO METHOD. FITTING IS THE SOLUTION OF LEAST SQUARE SYSTEM, AMPLITUDE IS THE ORIGINAL RESULT FROM SFEGO	27
FIGURE 31 DIAGRAM OF SFEGO COLOR	29
FIGURE 32 EXAMPLE OF SFEGO COLOR DECOMPOSITION	29
FIGURE 33 INVISIBLE SPACE SHIP ANALYSIS.....	29

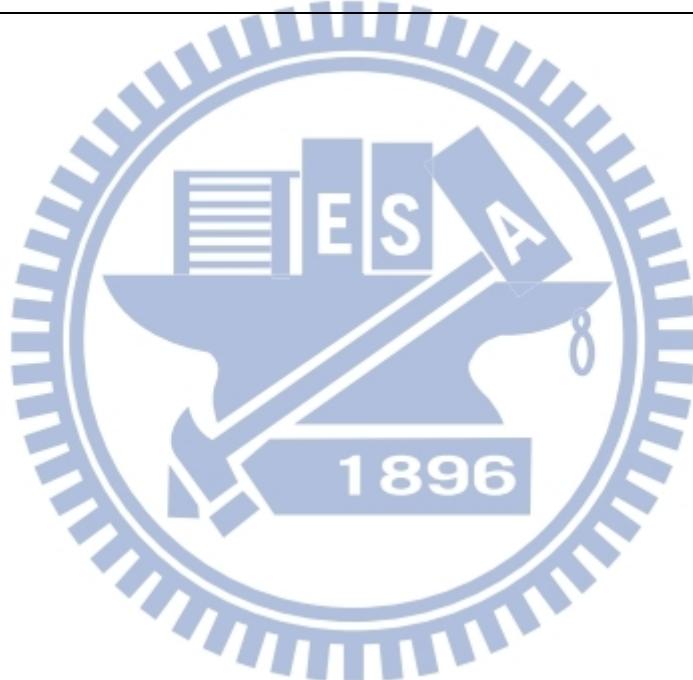
List of Table

TABLE 1 LITERATURE STUDY.....	4
TABLE 2 THE PERFORMANCE	24
TABLE 3 DOWN SAMPLING SPEEDUP TABLE.....	25
TABLE 4 THE RADIUS SEQUENCE.....	26



List of Abbreviation

EMD	Empirical Mode Decomposition
EEMD	Ensemble Empirical Mode Decomposition
BEMD	Bi-dimensional Empirical Mode Decomposition
MEEMD	Multi-dimensional Ensemble Empirical Mode Decomposition
IMF	Intrinsic Mode Function
BIMF	Bi-dimensional Intrinsic Mode Function
FABEMD	Fast and Adaptive Bi-dimensional Empirical Mode Decomposition
SFEGO	Spatial Frequency Extraction using Gradient-liked Operator
GO	Gradient-liked Operator
SF	Spatial Frame



1. Introduction

1.1. Background

Empirical Mode Decomposition (EMD) [1] is a method to decompose a signal into Intrinsic Mode Functions (IMF) that shows frequency information of each frequency band. The EMD method decomposes the signal into IMF by the following procedure. First, one repeatedly subtracts the mean value of upper and lower envelope of the signal from the original signal until the mean value of both the envelopes is close to zero that also called sifting processing. The signal after subtraction is IMF1. Second, one subtracts the IMF1 from the original signal to get the residual signal. After that, by decomposing the residual signal repeatedly by sifting processing, a sequence of IMFs (i.e. IMF1, IMF2, IMF3 ... and the last residual signal) can be generated.

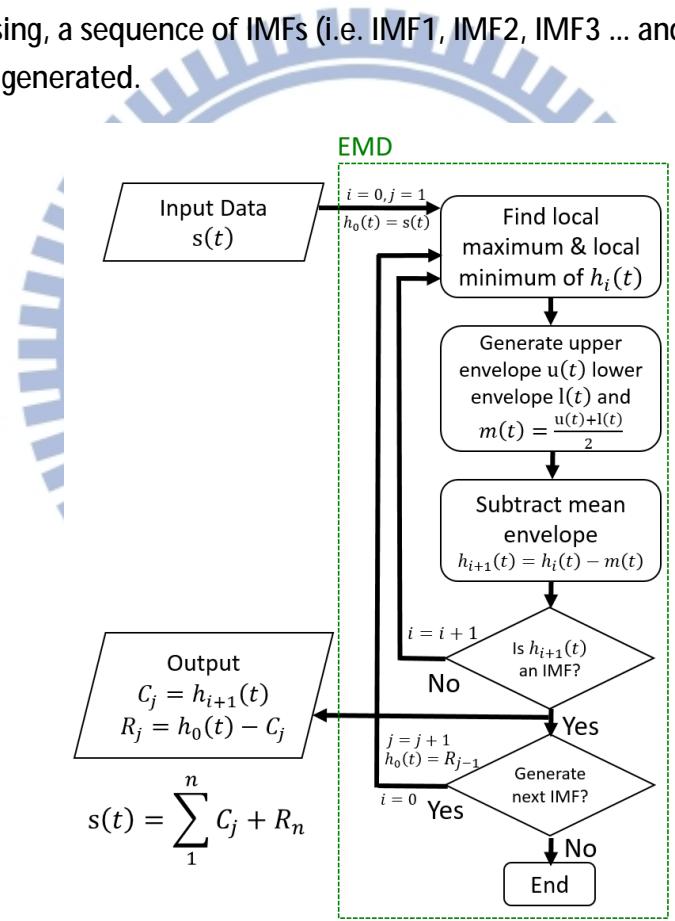


Fig. 1. EMD Procedure

One major drawback of the EMD method is mode mixing, that is, either a single IMF consists of signals of widely disparate scales or signals of a similar scale residing in different IMFs. To solve this mode mixing problem, Ensemble EMD (EEMD) [2] adds white noise which populates the whole time-frequency space uniformly to the original signal. To reduce the side effect of adding this white noise, do multiple times of EMD by adding this white noise into original signal, the average values of each

result are used for the different IMFs.

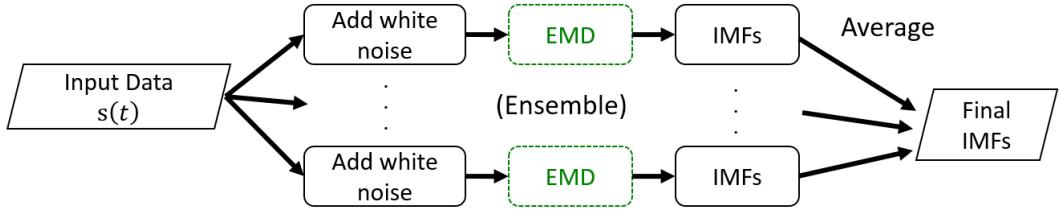


Fig. 2. EEMD Procedure

Bi-dimensional empirical mode decomposition (BEMD) [3] extends the EMD method to cope with two-dimensional data. The BEMD method finds extrema of 2D signal by morphologically reconstructs signals based on geodesic operators, and generating 2D envelope surfaces by connecting extrema points with radial basis function. The 2D IMFs also called Bi-dimensional IMFs (BIMFs) can be generated by

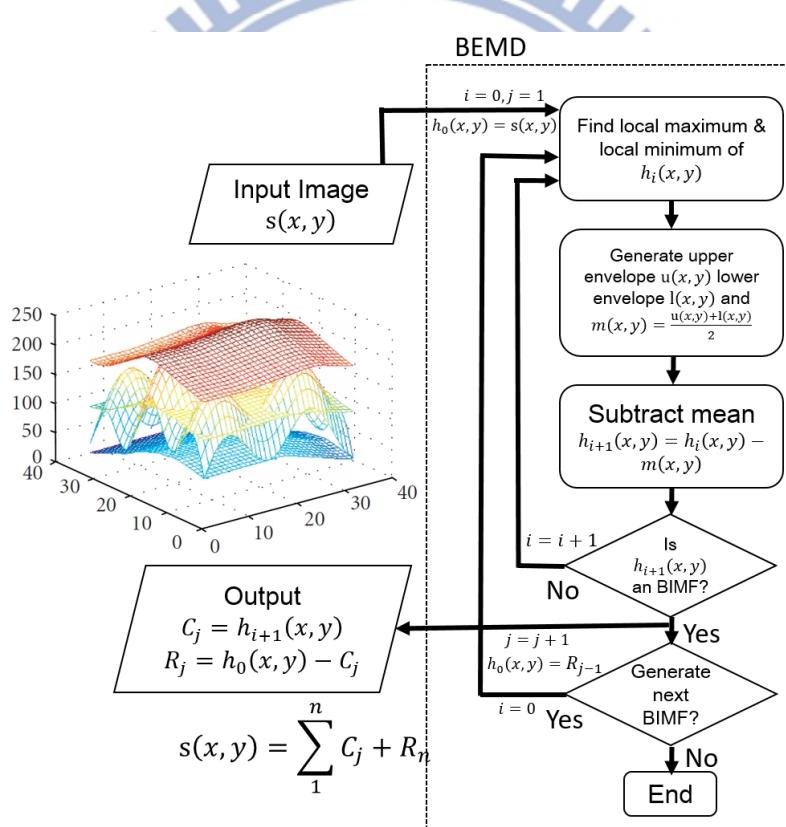


Fig. 3. BEMD Procedure

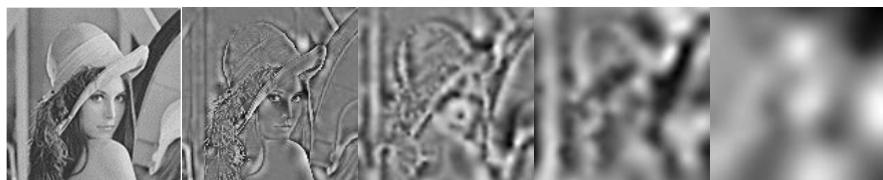


Fig. 4. BIMFs of Lena (left) by BEMD, BIMF1, BIMF2, BIMF3 and BIMF4 (right)

these 2D envelope surfaces. However, this approach could be computationally expensive in surface fitting and inherent the mode mixing problem.

Later studies proposed methods that use ensemble for avoiding mode mixing and reforming the 2D surface fitting approach into a multiple 1D EEMD approach.

Multi-dimensional EEMD (MEEMD) [4] enables the EEMD method to cope with multidimensional data and avoids the mode mixing problem in the same time. For instance, a two-dimensional EEMD can thus be constructed by applying the EEMD method to the rows and then applying to the columns of an image. Thus, a 2D IMF is

$$C_i = \sum_{k=i}^K h_{i,k} + \sum_{j=i+1}^J h_{j,i} \quad (1)$$

(in which the C_i is a 2D IMF, $h_{a,b}$ is column IMF_b of row IMF_a of the original image are the IMF number of row and column. When applying the EEMD method on the rows of the original image $h_1, h_2, h_3 \dots$ are obtained. After that, one applies the EEMD method to the columns of h_1 then get $h_{1,1}, h_{1,2}, h_{1,3} \dots$).

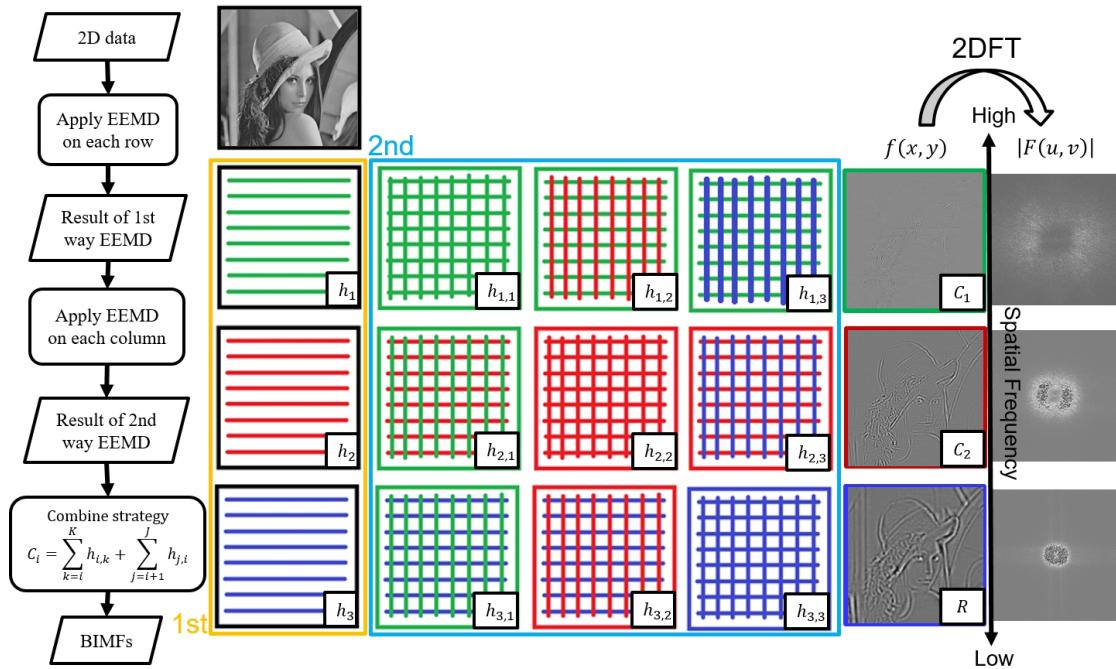


Fig. 5. MEEMD Procedure and example



Fig. 6. BIMFs of Lena (left) by MEEMD, BIMF1, BIMF2, BIMF3 and BIMF4 (right)

1.2. Motivation

The MEEMD method is an iterative progress that repeatedly decomposes a signal to generate IMFs. It applies ensemble approach by mixing white noise to the target signal numerous times to avoid mode mixing problem. However, the time complexity of performing the MEEMD method is high and the major EMD sifting processing is a data dependency procedure such that the major loop of EMD can't be parallel. Also the result of IMFs still contain noises especially for the IMFs of lower frequency band. The former fact may restrict the application of MEEMD from real-time applications.

1.3. Literature Study

After the survey on internet [5-11], we found a lot of author dedicated on the performance of the MEEMD, BEMD and EMD. The most notable is the Fast and Adaptive BEMD (FABEMD) [7] which implement the BEMD on the FPGA that provide 25fps real-time fusion of infrared and the visible image. But this method still have mode mixing problem. There is not good on the parallel of EMD due to the sifting processing in EMD is a data dependency iteration.

Year	Author	Algorithm	Execution time
2010	Pulung Waskito	EMD	7.1 times faster, L=215104, 8IMFs (CUDA)
2011	L.-W. Chang	MEEMD	62.14sec(47.1x) 666*444, 5BIMFs, En1000 (CUDA) 291.10sec(10.1x) 666*444, 5BIMFs, En1000 (OpenMP)
2012	Maciej Wielgus	FABEMD	25fps @ 640*480, 3BIMFs, FPGA, Order-Statistical Filters
2014	Y.-H. Wang	MEEMD	190.16sec @ 512*512, 6BIMFs, En100
2014	J.-X. Feng	fast MEEMD	30 times faster than MEEMD on ERSST, PCA+MEEMD
2014	Kevin P.-Y. Huang	EMD (CUDA)	37.9x and 33.7x speedup on voice and EEG datasets
2017	B.-W. Shen	Parallel EEMD	720x speedup using 200 eight-core processors (MPI+OpenMP)

Table. 1. Literature study of MEEMD, BEMD and EMD

1.4. Objective

To improve the performance and reduce the noise contains in IMFs, we propose Spatial Frequency Extraction using Gradient-like Operator (SFEGO) that use physical technique on spatial domain to calculate the gradient of the original data by different size of circular mask that large circular mask can extract lower spatial frequency trend and small circular mask extract higher spatial frequency. After the direction and differential of gradient already be done. We will integral the spatial frame by gradient result with certain circular mask size and the spatial frame may contain such wavelength of spatial frequency information.

2. Material and Method

Gradient is a concept of derivative of the function of several variables. It is usually used for estimating the slope of changes in a multi-dimensional signal. For instance, while air pressure varies in space, the gradient of the air pressure is very similar to air flow and is known as wind. Defined as,

$$\nabla f(x, y) = \frac{\partial f}{\partial x} \hat{x} + \frac{\partial f}{\partial y} \hat{y} \quad (2)$$

Traditionally, the gradient is calculated along Cartesian coordinate which only depends on adjoining data points. However, here we adopt a Gradient-like approach. Our method uses selected radius mask to extract more information. With larger radius extracts larger scale information (which leads to low spatial frequency), and smaller radius extracts high spatial frequency information. Therefore, the corresponded spatial frequency can be extracted through the choice of radius (R).

The larger circular mask extracts smoother information that is corresponding to lower spatial frequency signal varies in space. To obtain results similar to the IMF produced by the MEEMD method, we then integral (also called projection) the gradient result to spatial frame by

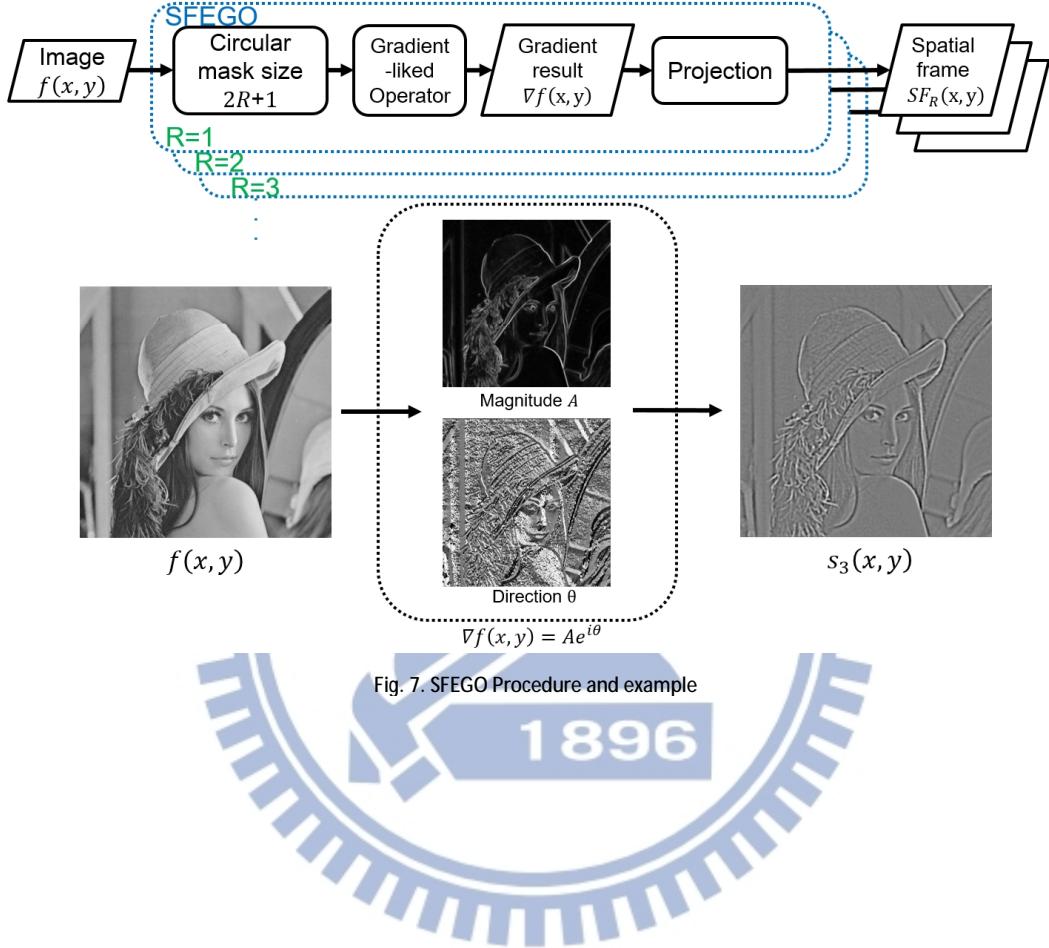
$$SF_R = \int_V \nabla^R f \, dv \quad (3)$$

After applying SFEGO on multi-dimensional signal can generate spatial frame SF_R that is BIMF in the corresponding radius R, where V is the volume within radius R. Thus, we can choose any specific radius for decomposition of the corresponded spatial frequency in multi-dimensional signal.

1896

2.1. Procedure

Our method use radius as a compute parameter for our circular mask which can extract a specific wavelength ($2R+1$) of spatial frequency. After get the gradient result for certain radius from Gradient-like Operator (GO) This gradient result contain the spatial frequency of specific wavelength. ($2R+1$) Then we can project (integral) back to spatial frame. Our approach is calculated at spatial domain.



2.2. Choose Radius

Select a radius of the circular mask which the diameter is $2R+1$ for calculating the differential result of gradient and project back on such volume.

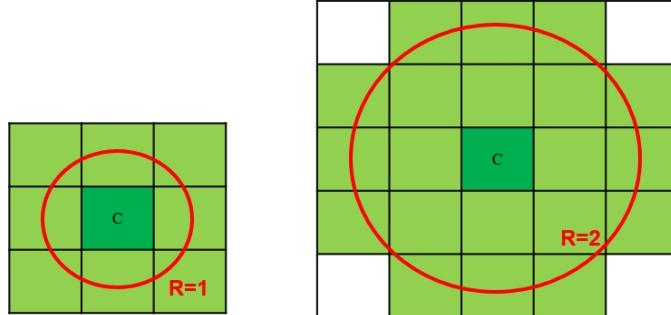


Fig. 8. SFGO Radius, Left($R=1$) and Right($R=2$)

2.3. Gradient-lied Operator

In this section, we propose a Gradient-lied Operator (GO) to calculate the gradient result for the certain circular mask radius. This gradient-lied operator may calculate the direction and magnitude for each image pixels and can be fully parallel due to this process has no data dependency on each pixels.

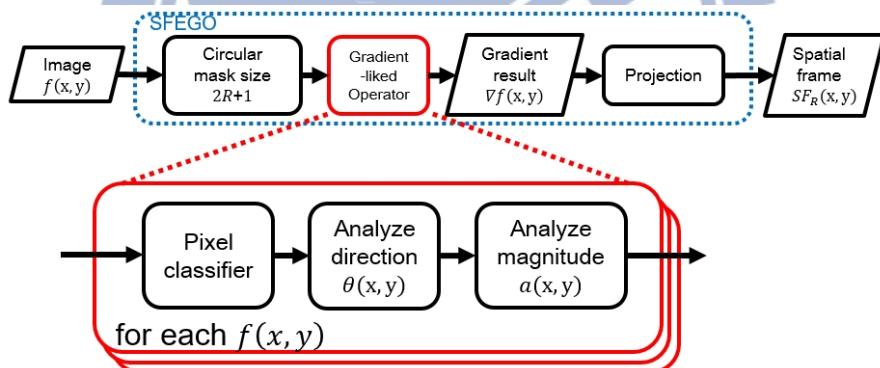


Fig. 9. Procedure of Gradient-lied Operator

2.3.1. Pixel Classifier

In this section, we will classify each pixel in the circular mask in order to evaluate the direction of gradient-lied operator. In a selected radius 2, mark the point as 1 if the value on $(x+i, y+j)$ larger than center point $C(x,y)$ and its opposite

	0	0	0 (x+i,y+j)	
0	0	0	0	0
0	0	C (x,y)	1	0
1	0	0	1	1
	1 (x-i,y-j)	1	1	

Fig. 10. Pixel Classifier on Circular Mask Radius R=2

point $(x-i, y-j)$ with initial marker 0, mark the point as -1 if the pixel is out of image range. This approach is inspired from the concept of Local Binary Pattern (LBP) [12] which is a type of feature extraction method traditionally used for classification in computer vision studies.

2.3.2. Analyze Direction

We decided to use a circular mask in our approach to obtain gradients. Although it seems to be infeasible to calculate the direction of a region solely based on few points, we come up with a method called “weight voting” to resolve this difficulty. We counted the number of markers (data points with larger value) in each sliding window sector as the weight of the direction of each sector and find the sector with a maximum weight to be the phase (direction) of the center point C.

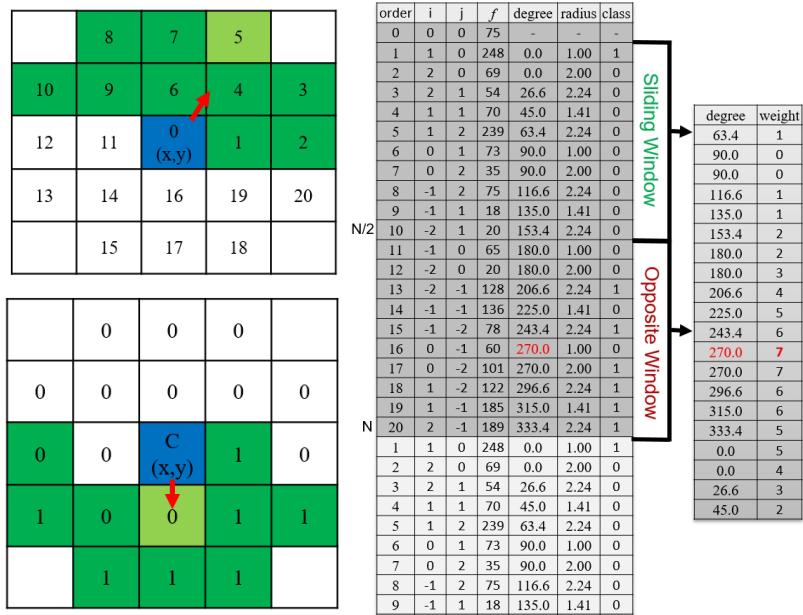


Fig. 11. Sliding Window Sequence (Left Top), Sliding Window Weight (Left Bottom) and Sliding Window Value (Right)

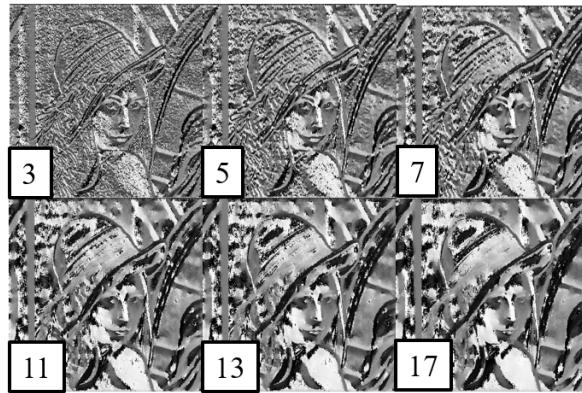


Fig. 12. Result of the direction by gradient-like operator with different radius. ($0-2\pi$ normalized to 0-255)

2.4. Analyze Magnitude

The magnitude of gradient of the center point “C” is the difference between the average value of largest sliding window and the average value of opposite sliding window inside a specific radius. Defined as

$$a(x, y) = \sum_{Largest} \frac{f(x+i, y+j)}{N_L} - \sum_{Opposite} \frac{f(x+i, y+j)}{N_O} \quad (4)$$

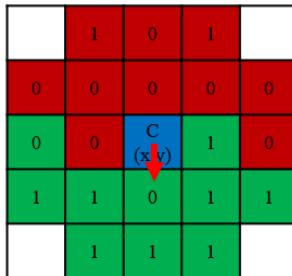


Fig. 13. Largest Sliding Window and Opposite Sliding Window

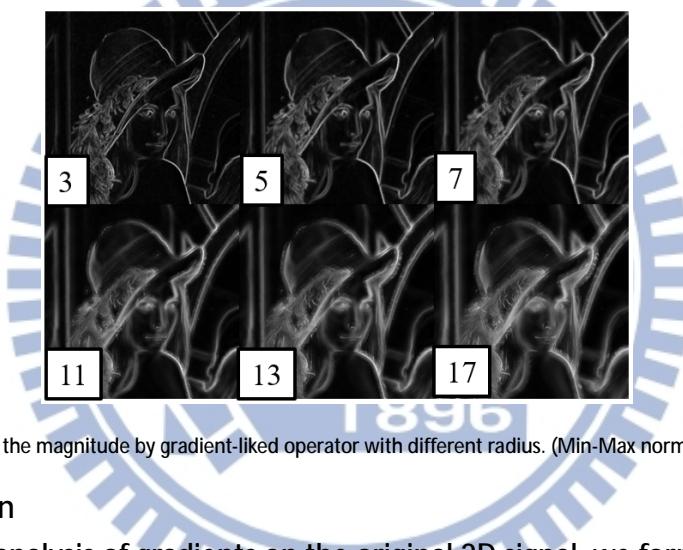


Fig. 14. Result of the magnitude by gradient-like operator with different radius. (Min-Max normalized to 0-255)

2.3. Projection

After the analysis of gradients on the original 2D signal, we formed a Spatial Frame by projecting (integrating) the gradients result from its magnitude and direction, the amount of contribution of each point $(x+i, y+j)$ within a selected radius is the direction from point $C(x, y)$ to point $(x+i, y+j)$ dot the direction of gradient of



$A = \{a(x, y)\}$



$\Theta = \{\theta(x, y)\}$

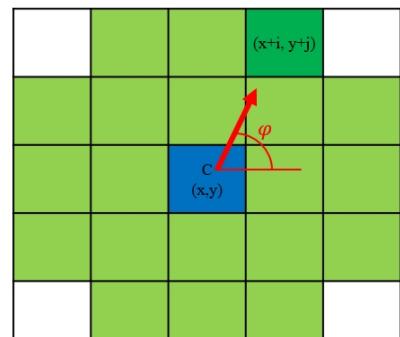


Fig. 15. Magnitude Frame (Left), Direction Frame (Middle) and angle φ . (Right)

point $(x+i, y+j)$. The sum of all the amount of contributions of center point C within the selected radius is the Spatial Frame with radius R. Defined as

$$SF_R(x, y) = \sum_{circular\ mask} \frac{a(x+i, y+j) \cdot \cos(\theta(x, y) - \varphi(x+i, y+j))}{Circular\ Mask\ Size} \quad (5)$$



Fig. 16. Result of spatial frame by SFEGO with different radius. (Min-Max normalized to 0-255)

2.4. OpenCL implementation

In section 2.4, we talk about the loop of each pixel point (x, y) can be fully parallel without data dependency which means the maximum simultaneously parallel thread can be the image size. (e.g. $512 \times 512 = 262144$) In the OpenCL heterogeneous computing. The parallel thread we create should larger than the real processor core (2048 cores on our GPU) due to the content switch between each computing thread can hide the memory access latency. In our performance test, it can reach 500x speedup.

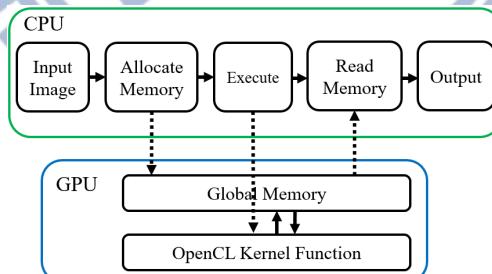


Fig. 17. OpenCL Computing diagram

3. Result

In order to achieve the same result of MEEMD with our SFEGO approach. We will use correlation coefficient to evaluate the result between MEEMD and SFEGO. Correlation coefficient is define as

$$r_{xy} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (6)$$

To evaluate the spatial frequency distribution, we also use 2D FFT to analysis the spatial frequency on frequency domain and calculate the histogram which is circular sum of magnitude of FFT on each radius. 2D Fourier Transform is defined as

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \quad (7)$$

The inverse FFT which can filter the spatial frequency on frequency domain is also a decomposition method. We use this method with following frequency band: 0~5, 5~10, 10~25, 25~50, 50~100 and 100~200.



3.1. Decomposition of a square border

In order to know the ability of decomposition a delta function from the sinusoidal background, we generate a 128x128 test image with 3 pixels wide square border and combine with the $\sin(0.3x) + \sin(0.3y)$ pattern.

In the decomposition result, MEEMD take a good place at decompose the border from the sinusoidal background. The spatial frame of radius 1 by SFEGO method contain background pattern. SFEGO have a good decomposition result at lower spatial frequency (radius=6) while the MEEMD still contain square border on BIMF2.

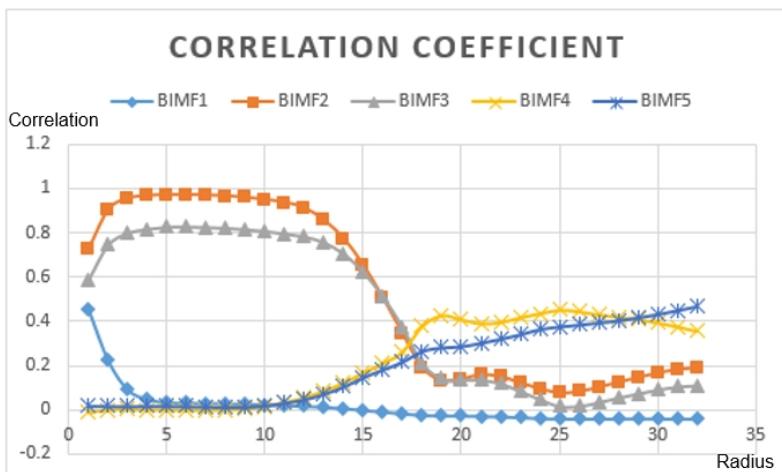
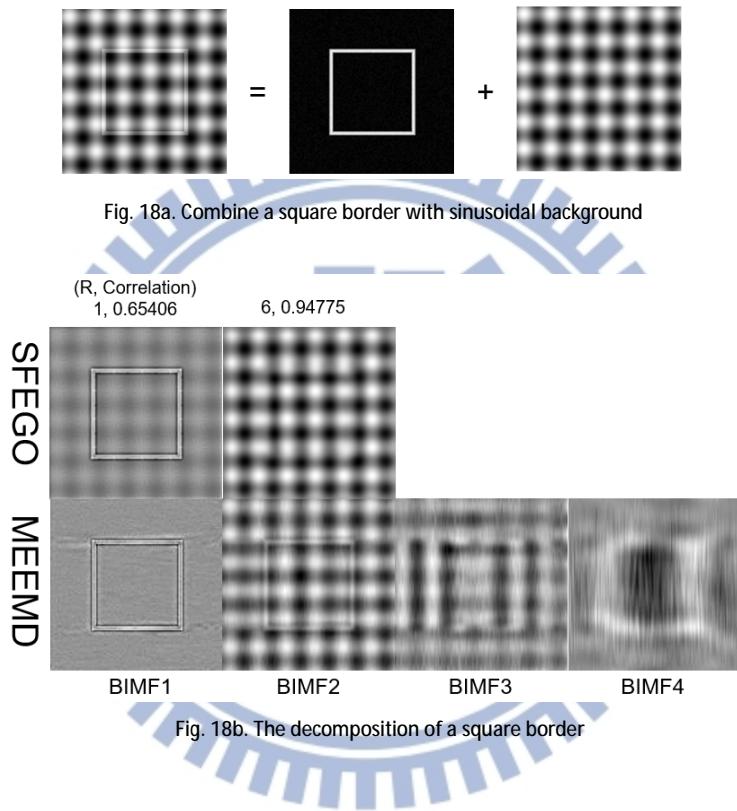


Fig. 18c. Correlation coefficient between SFEGO and MEEMD on square border

The FFT result shows that SFEGO R1 contain a small peak of low spatial frequency which is similar to the peak of SFEGO R6.

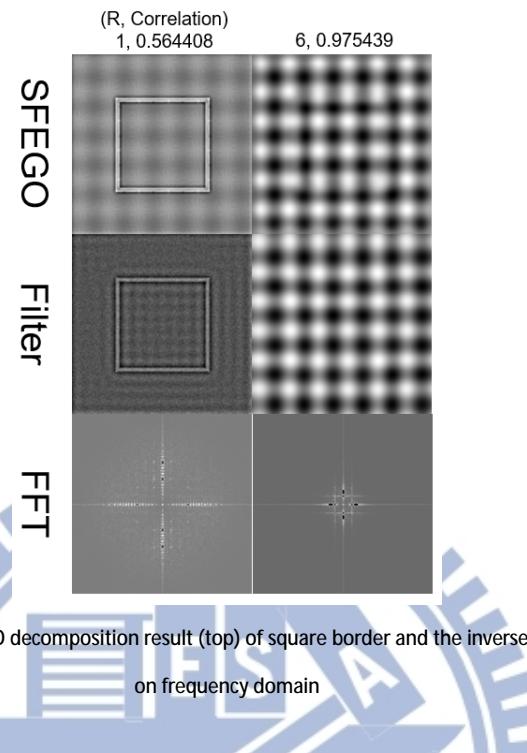


Fig. 18d. The FFT (bottom) of SFEGO decomposition result (top) of square border and the inverse FFT (middle) which is filter on frequency domain

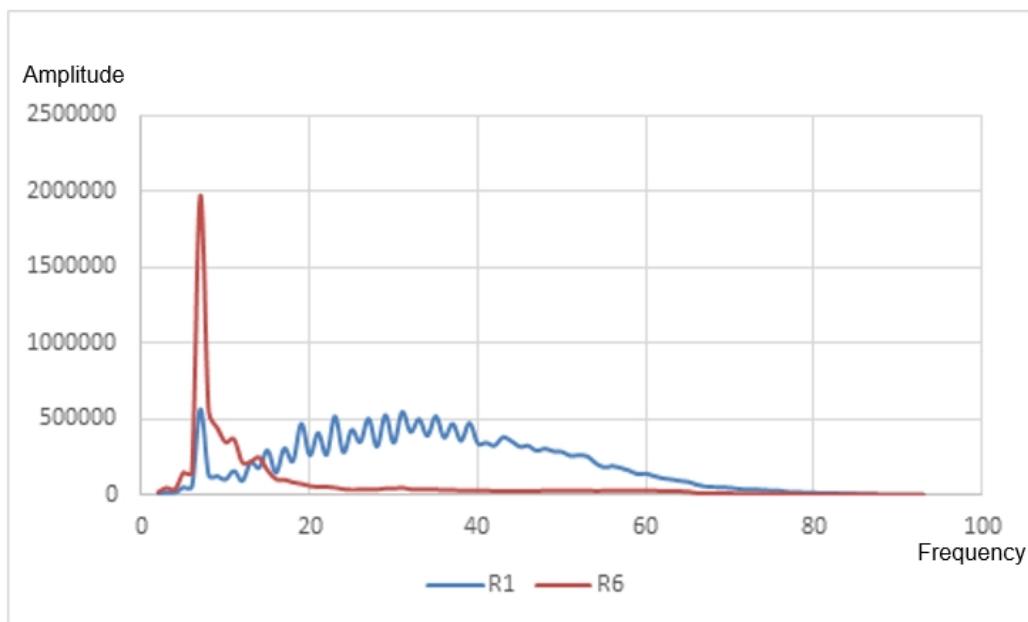


Fig. 18e. The diagram of FFT result on square border

3.2. Decomposition of a different spatial frequency 2D signal

We also tested the proposed SFEGO method on a 2D synthetic 128x128 data which is consisted of two 2D signals with different spatial frequencies and compared the obtained result to the result generated by the MEEMD method. In detail, the synthetic data F is

$$\mathbf{F} = \mathbf{F1} + \mathbf{F2} \quad (5)$$

$$\mathbf{F1(r)} = \sin(0.5r) \text{ at } (64, 64) \quad (6)$$

$$\mathbf{F2(r)} = \sin(0.2r) \text{ at } (32, 32) (32, 96) (96, 96) \quad (7)$$

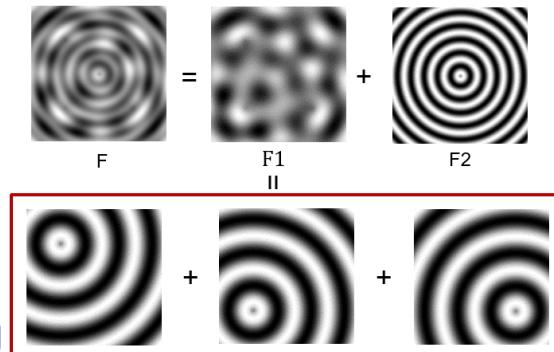


Fig. 19a. Function F (left top) is F1(middle top) add F2(right top) which F1 is the sum of three location sine wave (bottom)

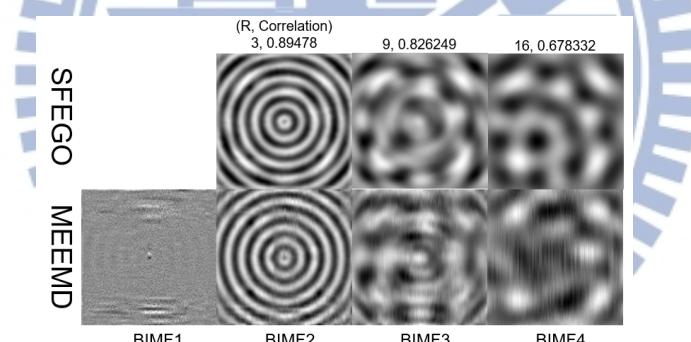


Fig. 19b. Decomposition Result of F by SFEGO (top) and MEEMD (bottom)

SFEGO approach can decomposition the F1 and F2 from F such as the BIMF2 and BIMF4 of MEEMD. In BIMF3 and higher, SFEGO approach can keep clear while the noise become notable in MEEMD.

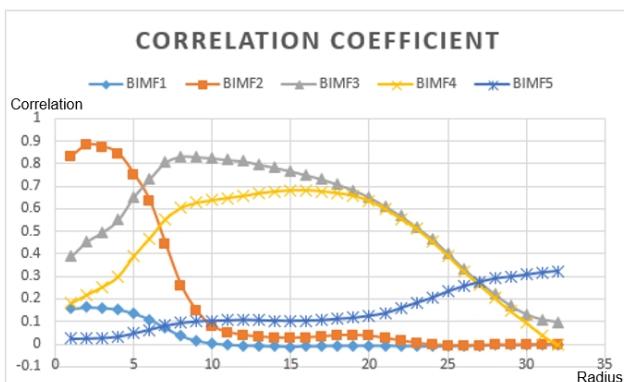


Fig. 19c. Correlation coefficient between SFEGO and MEEMD on F

The R9 and R16 has the same peak on the histogram of FFT which is the spatial frequency of F1.

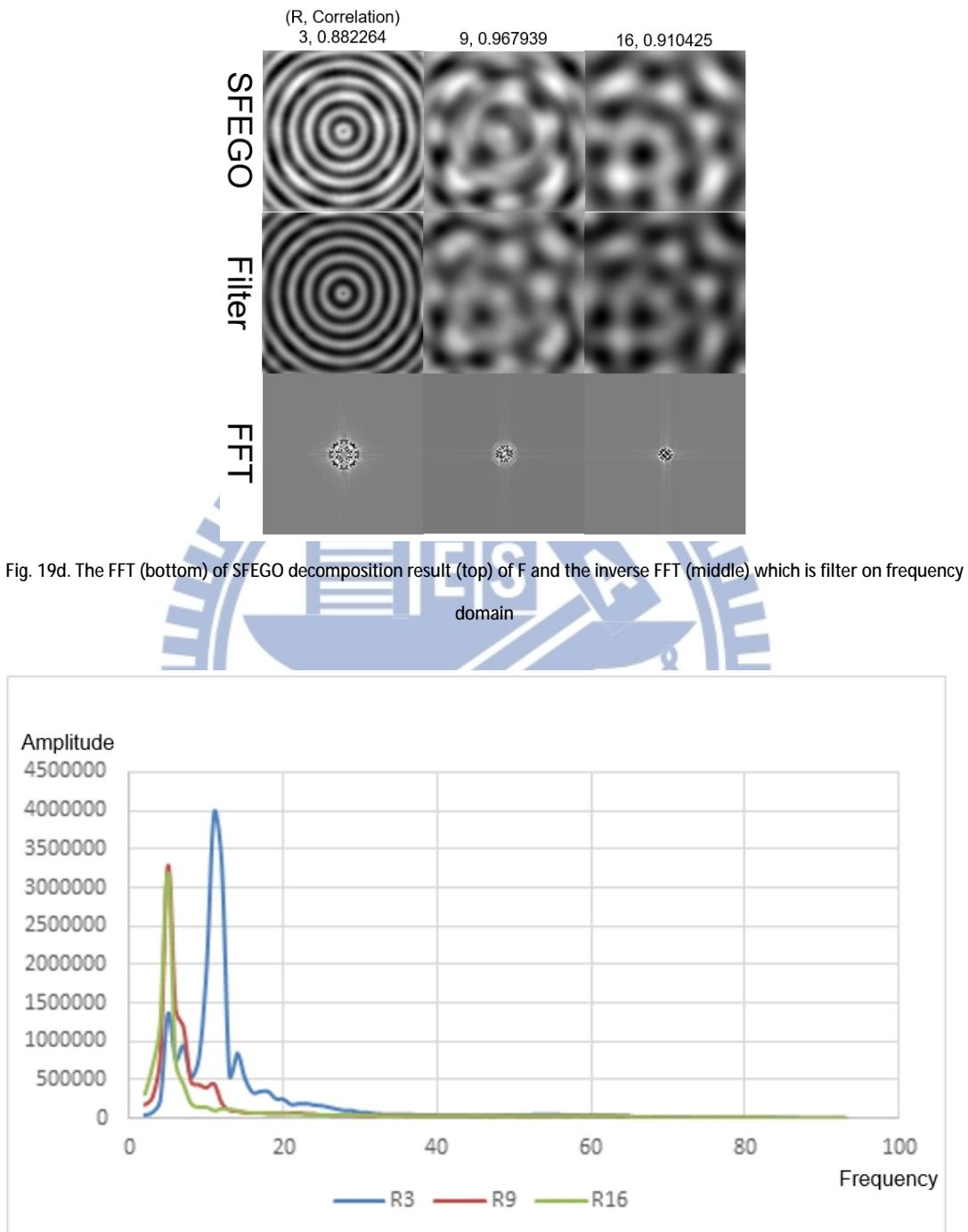


Fig. 19e. The diagram of FFT result on F

3.3. Decomposition of Lena

The example shown below is a decomposition result of a famous image “Lena” that has been typically used for demonstration purpose in the field of image processing. The matching of the BIMFs generated by the two different methods is based on maximum correlation coefficient between BIMFs generated by the MEEMD method and the spatial frame generated by the SFEGO method.



Fig. 20a. Image of Lena

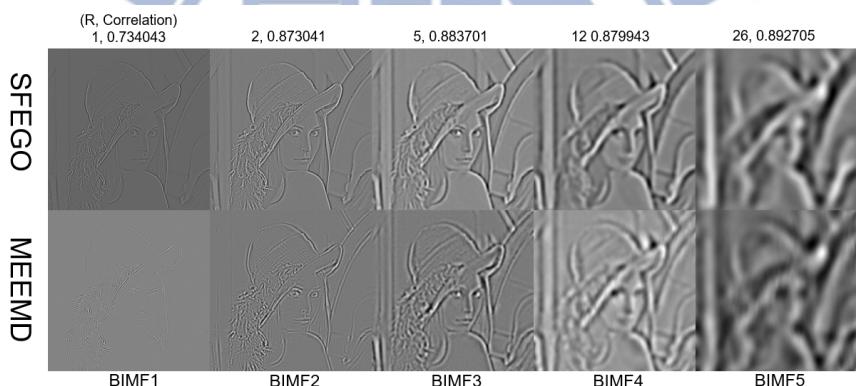


Fig. 20b. Decomposition of Lena with SFEGO and MEEMD.

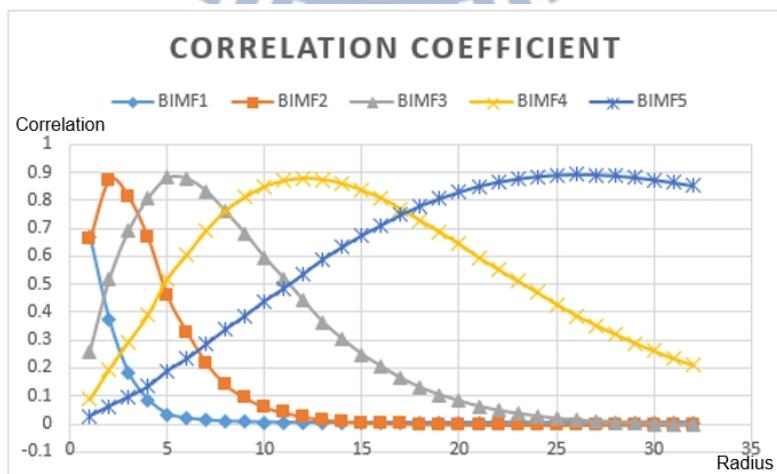


Fig. 20c. Correlation Coefficient between SFEGO and MEEMD on Lena

The diagram of FFT shows that R1, R2, R5, R12 and R26 contain different spatial frequency that is Gaussian distribution on frequency domain. The inverse FFT which filter on frequency domain contain sinusoidal pattern.

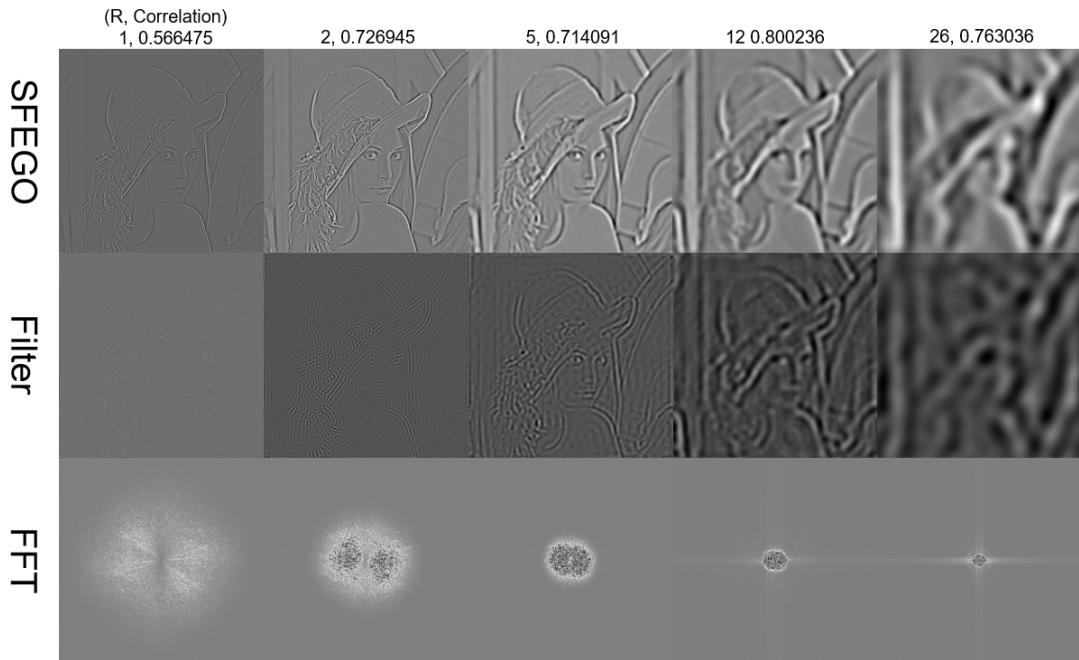


Fig. 20d. The FFT (bottom) of SFEGO decomposition result (top) of Lena and the inverse FFT (middle) which is filter on frequency domain

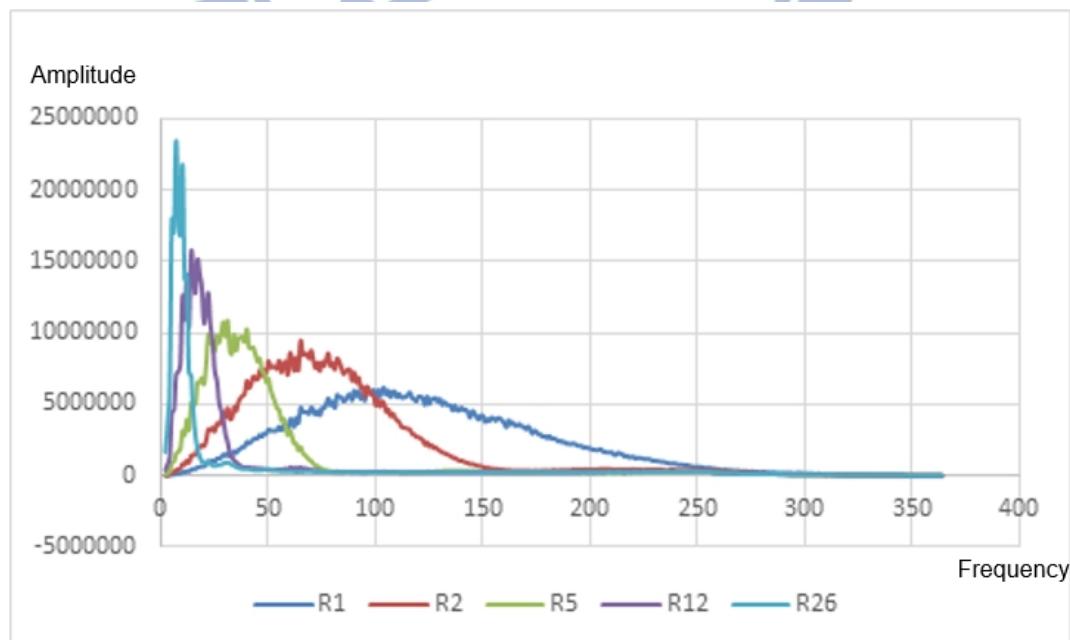


Fig. 20e. The diagram of FFT result on Lena

3.4. Decomposition of Squirrel

This example is showing the decomposition the hair structure that contain high spatial frequency. This image is download from spatial frequency domain website <https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic1.htm>

The decomposition result of SFEGO on Radius 1 contain low spatial frequency information. And the result of MEEMD contain noise which added by ensemble method on BIMF3, BIMF4 and BIMF5.



Fig. 21a. Image of squirrel

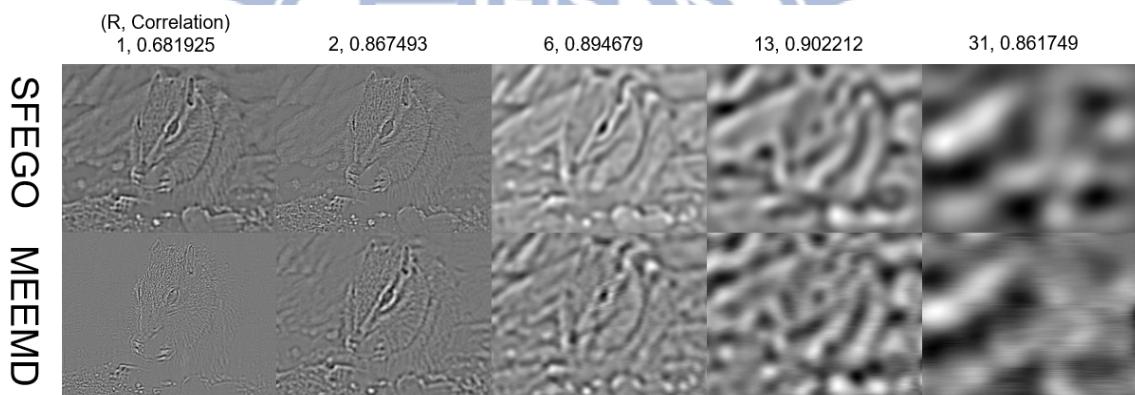


Fig. 21b. Decomposition of the squirrel

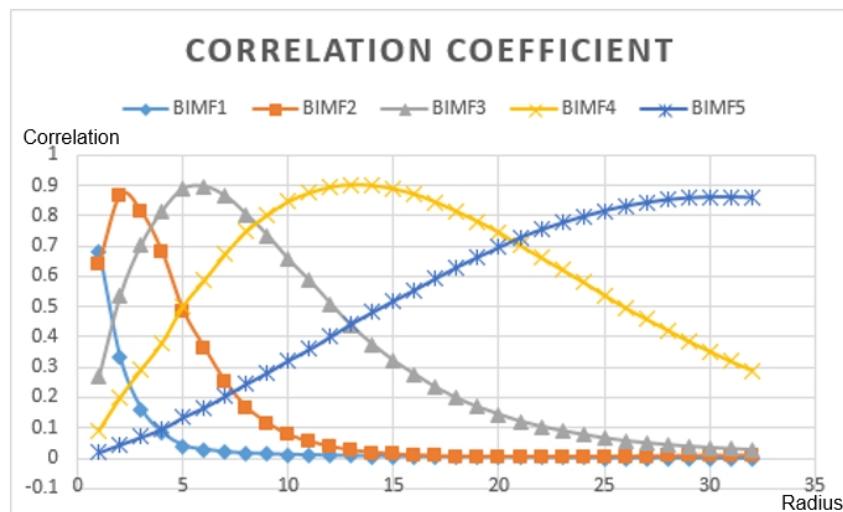


Fig. 21c. Correlation Coefficient Between SFEGO and MEEMD on Squirrel

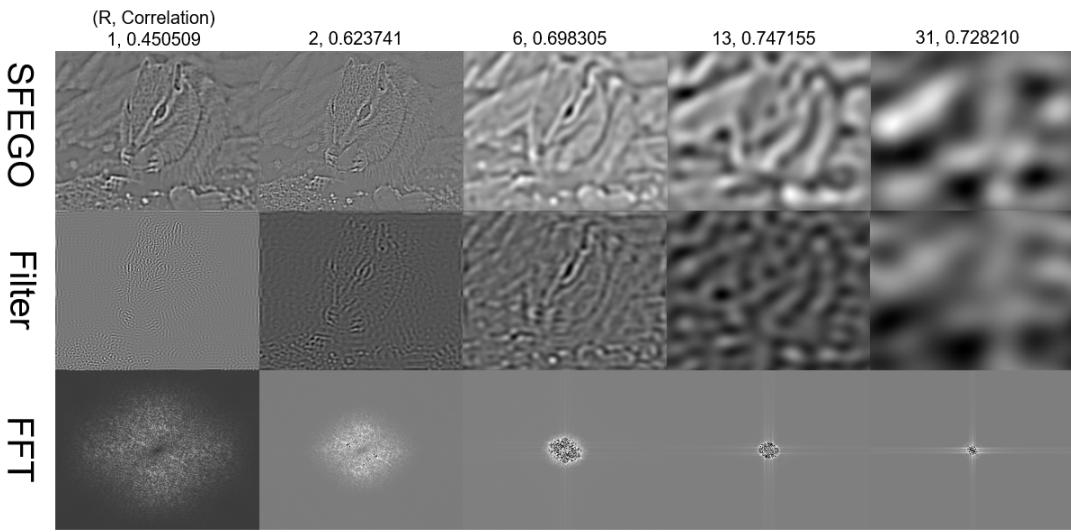


Fig. 21d. The FFT (bottom) of SFEIGO decomposition result (top) of Squirrel and the inverse FFT (middle) which is filter on frequency domain

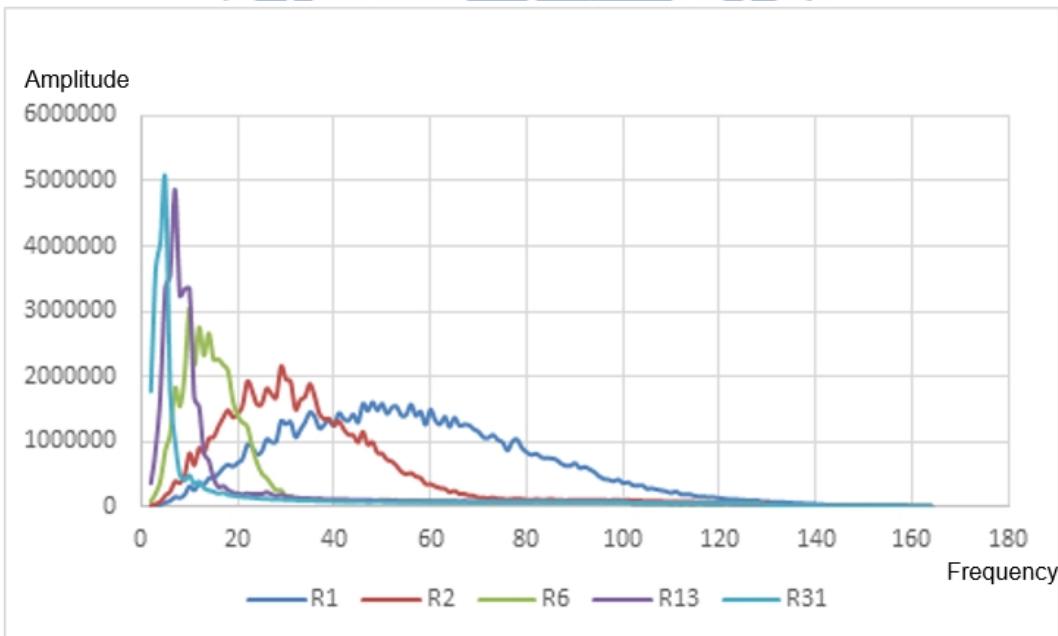


Fig. 21e. The diagram of FFT result on Squirrel

3.5. Decomposition of London Bridge

This example is showing the decomposition of building. This image is download from website of the top 100 most beautiful places.

<http://famousearths.blogspot.tw/2014/08/top-100-most-beatiful-palces.html>



Fig. 22a. London Bridge

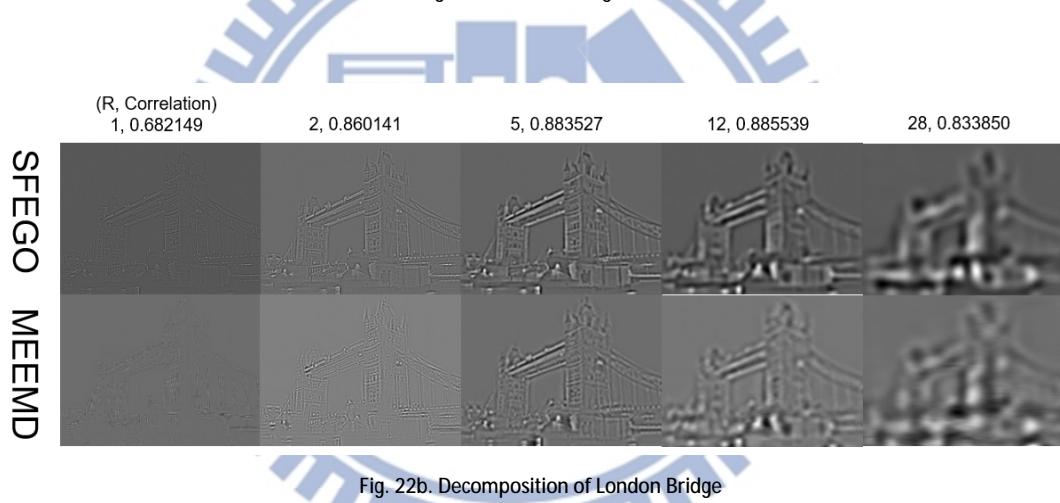


Fig. 22b. Decomposition of London Bridge

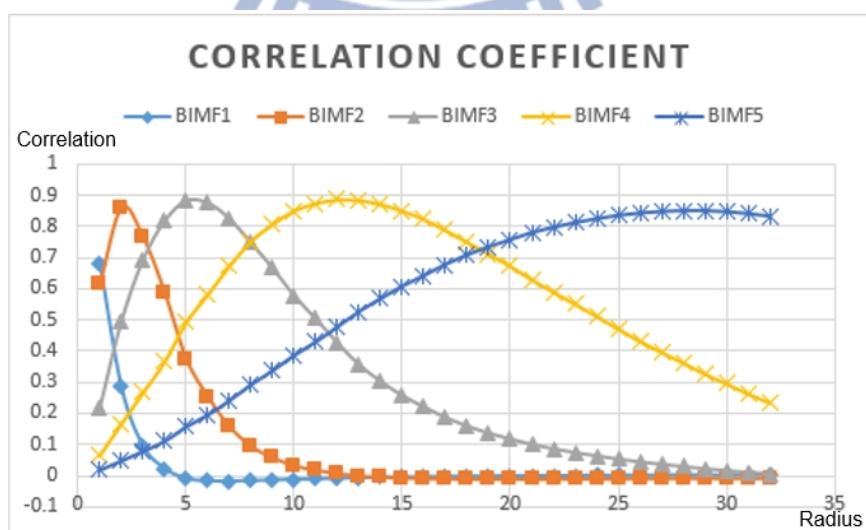


Fig. 22c. Correlation Coefficient between SFEGO and MEEMD on London Bridge

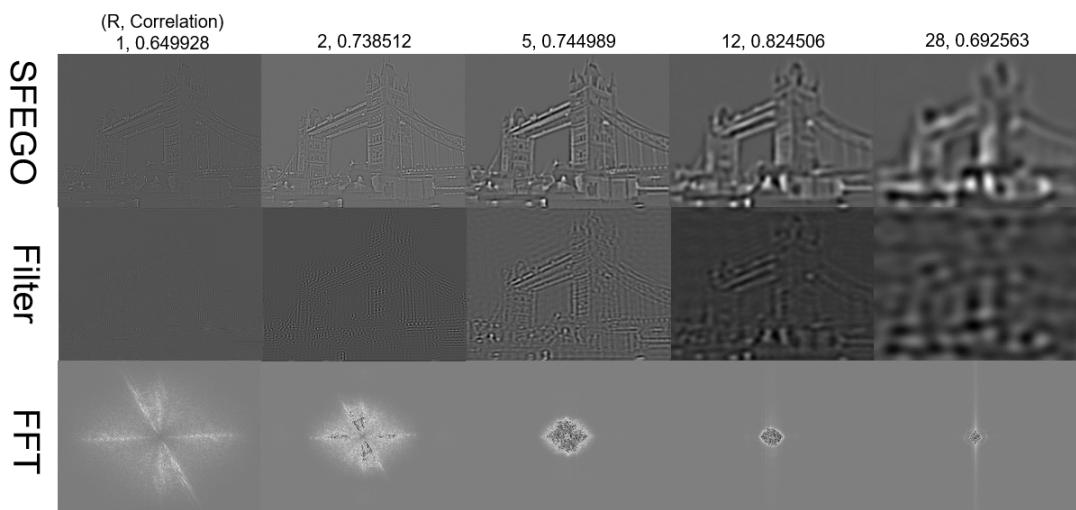


Fig. 22d. The FFT (bottom) of SFEGO decomposition result (top) of London Bridge and the inverse FFT (middle) which is filter on frequency domain

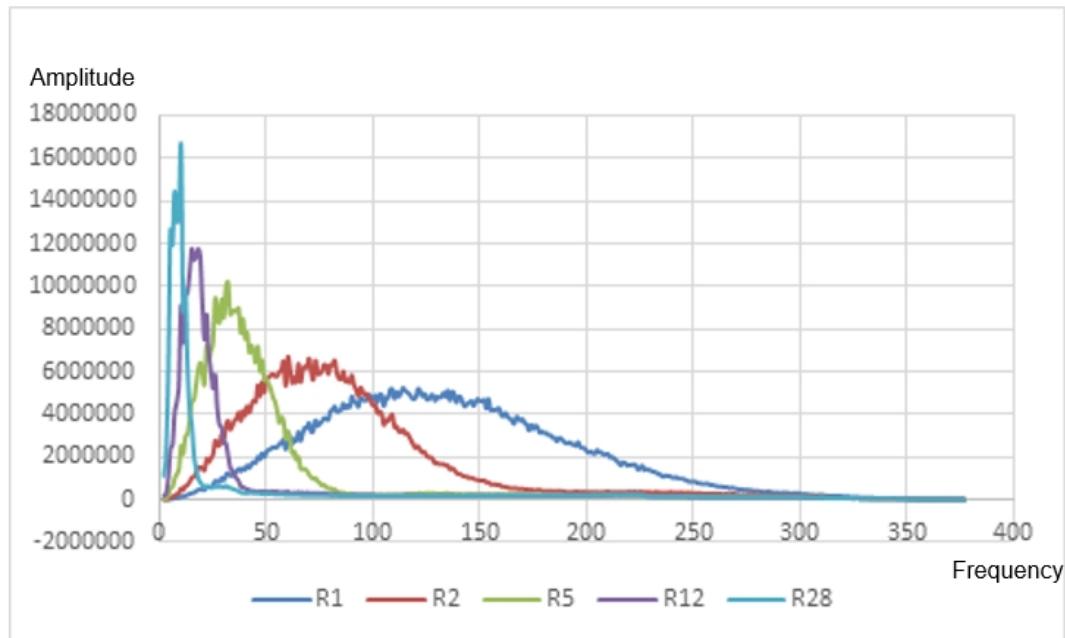


Fig. 22e. The diagram of FFT result on London Bridge

4. Discussion

We will discuss the performance of SFEGO method and compare to the MEEMD method. And the huge speedup between OpenCL version of SFEGO and C++ version. Then we will talk about the large radius SFEGO performance problem. Which can use down-sampling method with smaller radius then up-sampling to reduce the computation and get the similar result as large radius. The other hand, we can up-sampling to extract more information from original image.

4.1. Circular mask v.s. square mask

The SFEGO approach use circular mask as its computation element even other spatial filter or convolution usually use square mask. We also can test square mask as the computation element. Below is our test result. There is no different in small radius. Circular mask has better result on large radius.

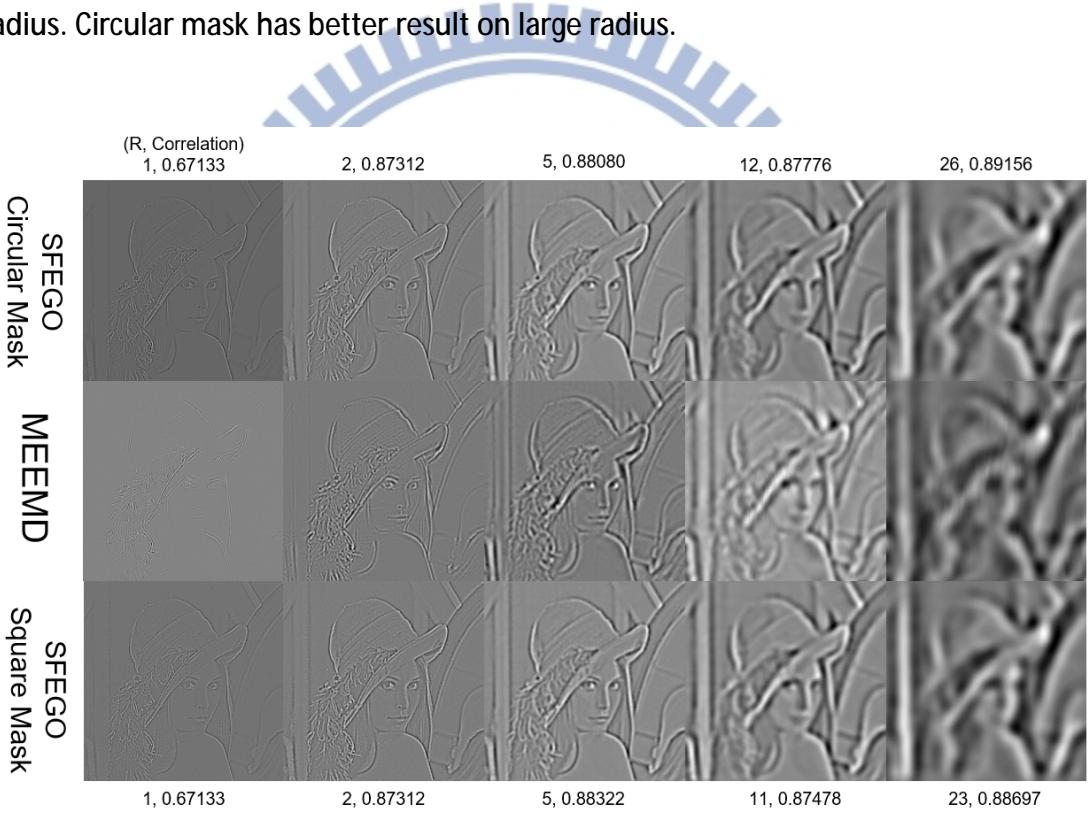


Fig. 23. Circular Mask v.s. Square Mask

4.2. Direction error on small radius circular mask

Our approach may have the fragment on the small radius which is influence by the discrete of the direction on gradient result. We use sliding window to analyze the direction of gradient which limit the direction to the sliding window size. For example, the smallest radius $R=1$ only have 8 direction which is the sliding window size. Such a discrete problem may cause the result of SFEGO contain fragment.

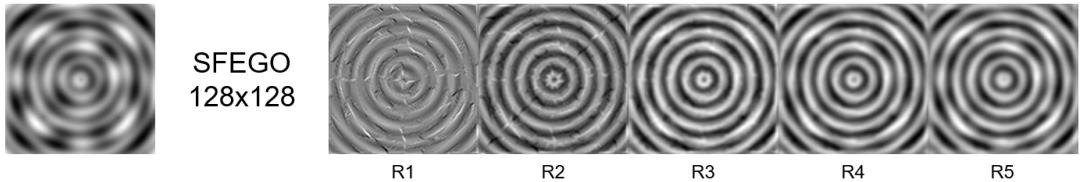


Fig. 24. Fragment on small radius ($R=1, R=2$) decomposition

4.2.1. Up sampling for fixing direction error

Up sampling can increase the detail of the decomposition. The effective radius of SFEGO is $(Radius) \div (DownSamplingRatio)$. Therefore, we can extract the spatial frequency information without direction issue that introduce in 4.1. In our test, the final correlation coefficient have a small increase after using up sampling method.



Fig. 25. Up sampling diagram

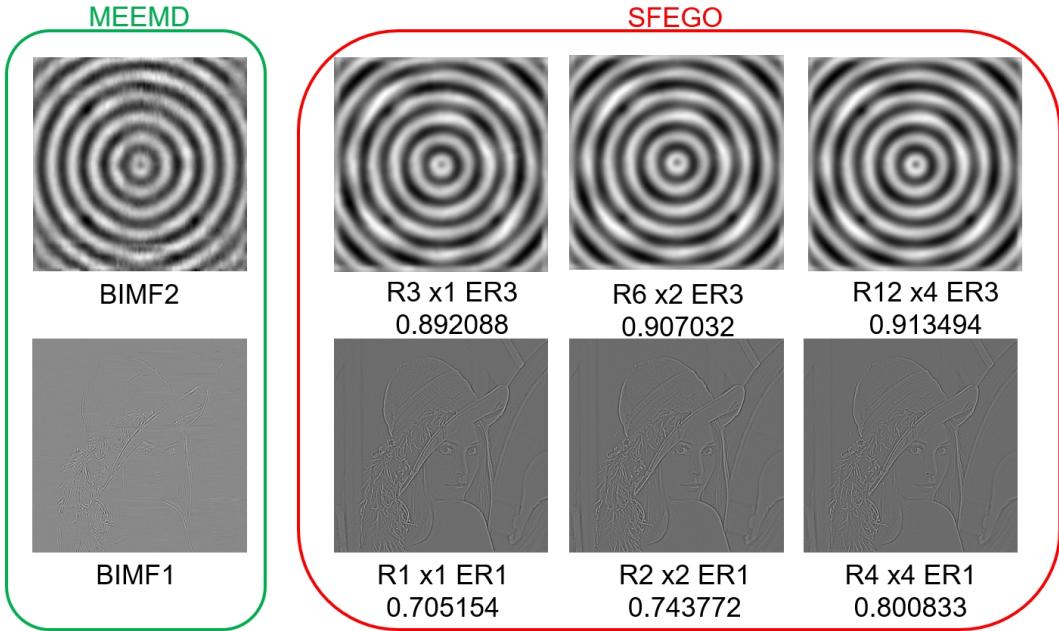


Fig. 26. Up sampling result of decomposition F (top) and Lena (bottom). ER1 and ER3 means effective radius.

4.3. Performance

SFEGO improve the performance, skip the iteration of finding IMF that direct analysis the specific spatial frequency and make similar result as MEEMD. The speedup of SFEGO on generating similar result as IMF5 of MEEMD is about 10x. The

SFEGO			MEEMD		Speedup
R	Pixels	Time(s)	IMF	Time(s)	
1	8	0.237	1	78.02	329.1983
2	20	0.613	2	175.337	286.031
3	44	1.098			
4	68	1.632			
5	100	2.504			
6	136	3.259	3	299.142	91.78951
7	184	4.371			
8	240	5.627			
9	292	6.798			
10	356	8.438			
11	428	9.627			
12	496	11.441	4	446.169	38.99738
13	592	13.441			
14	672	15.48			
15	776	16.79			
16	876	18.786			
17	980	20.828			
18	1092	23.536			
19	1200	25.801			
20	1344	28.39			
21	1480	31.516			
22	1616	34.624			
23	1756	39.043			
24	1900	40.34			
25	2076	43.568			
26	2232	47.189	5	616.073	13.05544
27	2400	50.709			
28	2584	53.938			
29	2760	58.066			
30	2948	61.916			

SFEGO			C++		OpenCL Times (sec)	Speedup
R	Pixels	Times (sec)	C++ Times (sec)	OpenCL Times (sec)		
1	8	0.274	0.015	18.33333		
2	20	0.712	0.015	46.93333		
3	44	1.288	0.016	78.9375		
4	68	1.825	0.016	114.3125		
5	100	2.848	0.019	149.1053		
6	136	3.775	0.02	187.35		
7	184	4.941	0.021	234.3333		
8	240	6.34	0.025	252.36		
9	292	7.748	0.025	308.28		
10	356	9.396	0.028	335.8214		
11	428	10.995	0.033	332.4545		
12	496	12.946	0.034	379.5882		
13	592	15.24	0.045	337.5333		
14	672	17.392	0.043	404.3488		
15	776	19.812	0.045	439.3556		
16	876	22.445999	0.054	409.7963		
17	980	25.004	0.052	480.8654		
18	1092	27.875999	0.064	435.4531		
19	1200	30.889	0.074	417.0135		
20	1344	34.154999	0.083	408.5301		
21	1480	37.726002	0.091	411.4835		
22	1616	41.153999	0.098	415.2041		
23	1756	44.216	0.096	459.0521		
24	1900	47.949001	0.09	531.5667		
25	2076	51.848	0.111	465.937		
26	2232	56.125	0.123	455.2846		
27	2400	60.174999	0.133	451.0902		
28	2584	64.321999	0.138	464.3551		
29	2760	68.949997	0.146	470.8151		
30	2948	73.912003	0.15	488.7933		

Table. 2. The performance between SFEGO and MEEMD both with single thread on Intel i7-3770 (Left) and the performance of SFEGO between C++ version and OpenCL version on Intel Pentium G2120 and AMD R9-280x (right)

heterogeneous computing version of SFEGO offload each the computation of each pixels into GPGPU thread that can reach near 500x speedup.

4.3.1. Down sampling for acceleration

SFEGO have long computation time on larger radius which the computational cost is $O(R^2)$. In order to reduce the computation, we will down sampling original image and do the smaller radius of SFEGO then up sampling the spatial frame where the effective radius is the $(Radius) \times (DownSamplingRatio)$. By this way, it can reach 50x speedup on large radius. The correlation coefficient between the normal version of SFEGO and the down-sampling version is very high. We can say that it is almost no different between two version.

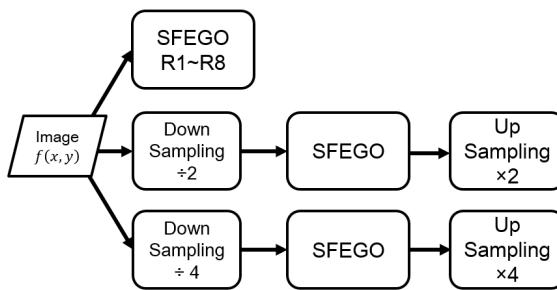


Fig. 27. Down sampling diagram (left)

Down Sampling Ratio	SFEGO Radius	Effective Radius	GPU with Original Radius	GPU with Down Sampling	Speedup	Correlation Coefficient
1	1	1	0.015	0.015	1	1
1	2	2	0.015	0.015	1	1
1	3	3	0.016	0.016	1	1
1	4	4	0.016	0.016	1	1
1	5	5	0.019	0.019	1	1
1	6	6	0.02	0.02	1	1
1	7	7	0.021	0.021	1	1
1	8	8	0.025	0.025	1	1
2	5	10	0.028	0.006	4.666667	0.987974
2	6	12	0.034	0.007	4.857143	0.993042
2	7	14	0.043	0.008	5.375	0.994158
2	8	16	0.054	0.008	6.75	0.99383
4	5	20	0.083	0.003	27.666667	0.968883
4	6	24	0.09	0.003	30	0.984733
4	7	28	0.138	0.003	46	0.989197
4	8	32	0.165	0.003	55	0.991995

Table 3. Down sampling speedup table (right)

4.3.2. Real-Time decomposition

With down sampling method, it can reach 45 BIMFs per second on our R9-280x GPU with 640x480 pixels. The average frame time for spatial frame is 22.2ms. The actual effective radius is R1, R2, R6, R12(2x6) and R26(2x13). That is 5000x speedup compare to the MEEMD method.



Fig. 28. Real-Time Webcam Decomposition

4.4. MEEMD Emulation

The SFEGO generate the Spatial Frame according to the chosen radius. While the MEEMD generate BIMFs by continuing decomposition the residual of previous BIMF. The largest correlation coefficient between SFEGO and MEEMD on the first 5 BIMFs for Lena is R1, R2, R5, R12 and R26. This radius sequence can be used later for SFEGO to emulation the MEEMD processing. In order to find the most general radius sequence, we use 5 picture find the radius sequence and use median as our general usage. Then we use curve fitting of this median radius sequence to find the best radius from BIMF6 to BIMF8. So the final radius sequence is R1, R2, R6, R27, R56, R118 and R249.



Fig. 29. 5 picture for finding radius sequence. Lena, 2men, Squirrel, tower and bridge

	Lena	2men	squirrel	tower	bridge	average	median
Diameter	$1.2881e^{0.7353x}$	$1.5408e^{0.7167x}$	$1.2334e^{0.7775x}$	$1.3744e^{0.7196x}$	$1.2512e^{0.7498x}$	$1.3346e^{0.741x}$	$1.3123e^{0.7427x}$
Radius	Lena	2men	squirrel	tower	bridge	average	median
BIMF1	1	1	1	1	1	1	1
BIMF2	2	3	2	2	2	2.2	2
BIMF3	5	6	6	5	5	5.6	6
BIMF4	12	13	13	12	12	12.4	12
BIMF5	26	27	31	25	28	27.4	27
BIMF6	53	56	65	51	56	56	56
BIMF7	110	116	142	105	119	119	118
BIMF8	230	238	310	217	251	250	249

Table. 4. The radius sequence on largest correlation coefficient

To emulation the MEEMD, we also need to evaluate the amplitude between the spatial frame of SFEGO and the BIMFs of MEEMD. The least square is introduced to solve eigen value from eigen vector and the Input. We use the SVD decomposition method to solve linear least square system which the code can be download from <http://eigen.tuxfamily.org/>. After we solve the least square system, we will get the amplitude of each spatial frame that the sum of all spatial frame is similar to the input image.

The Least Square System is defined as,

$$\mathbf{AX} = \mathbf{B} \quad (8)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{1} & \mathbf{BIMF1}_1 & \cdots & \mathbf{BIMFm}_1 \\ \mathbf{1} & \vdots & \ddots & \vdots \\ \mathbf{1} & \mathbf{BIMF1}_N & \cdots & \mathbf{BIMFm}_N \end{bmatrix} \quad (9)$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{Input}_1 \\ \vdots \\ \mathbf{Input}_N \end{bmatrix} \quad (10)$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_m \end{bmatrix} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B} \quad (11)$$

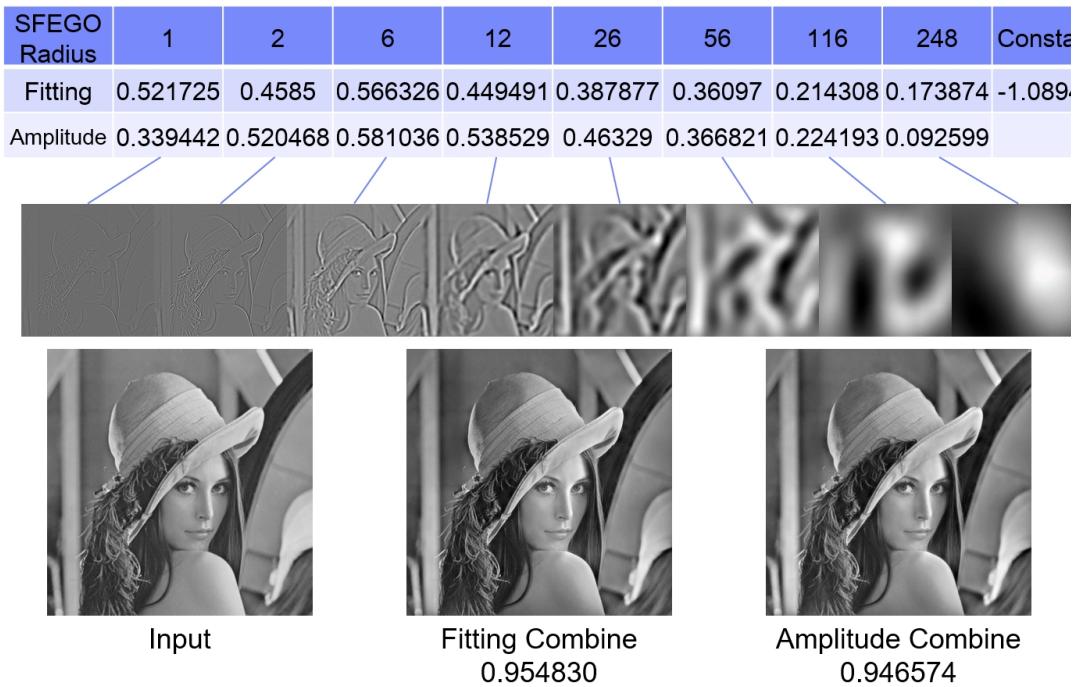


Fig. 30. The decomposition and amplitude of Lena with SFEGO method. Fitting is the solution of least square system,

Amplitude is the original result from SFEGO.

5. Conclusion

The main reason we choose the circular mask as the computational element of SFEGO is that the different size of the circular mask can extract the different spatial frequency. Although the small circular mask may contain some fragment on the direction due to the discrete of the direction computation. The result of SFEGO is very similar to MEEMD on the specific radius sequence (R1, R2, R6, R12, R27...) That is means the BIMFs from MEEMD has a fixed spatial frequency range. The performance of SFEGO can achieve real-time decomposition which is 5000x faster than MEEMD method.



6. Future Work

This Future Work is inspired by the FABEMD real-time fusion that fusing a IR image and a visible image to analysis the UFO. This approach decompose the RGB frame and generate color spatial frame which also can detect UFO. My soul mate help me to call pleiadian space ship for testing this method. We can clarity see the invisible space ship after the enhancement of color spatial frame.

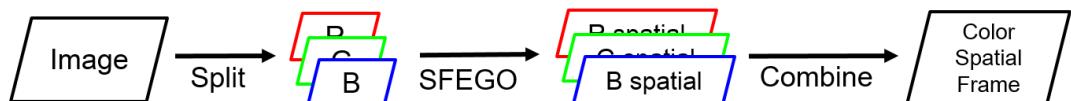


Fig. 31. Diagram of SFEGO Color

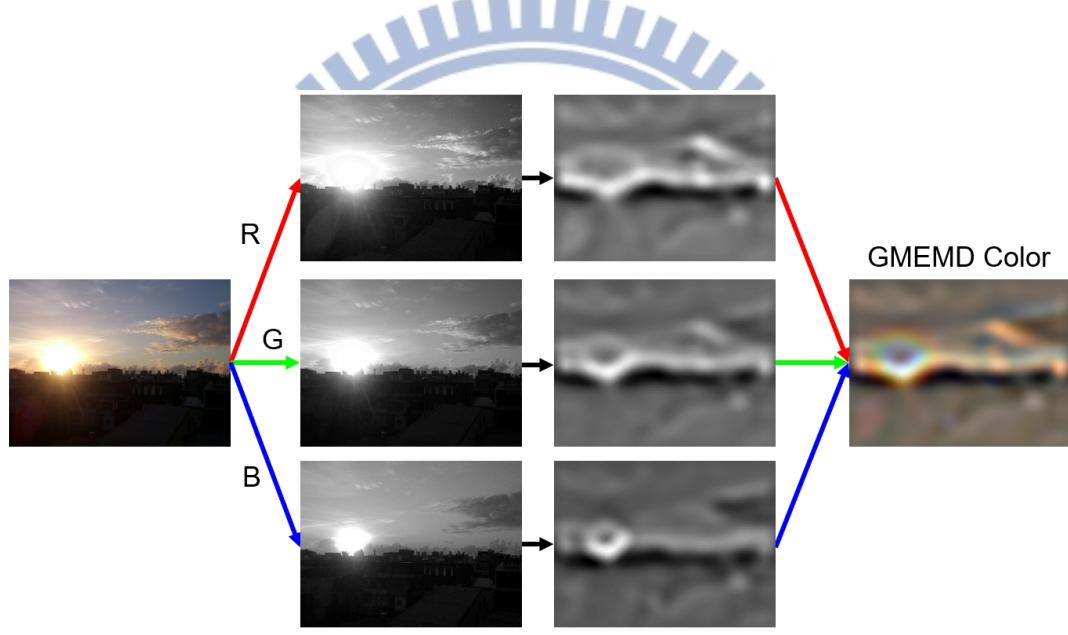


Fig. 32. Example of SFEGO Color Decomposition

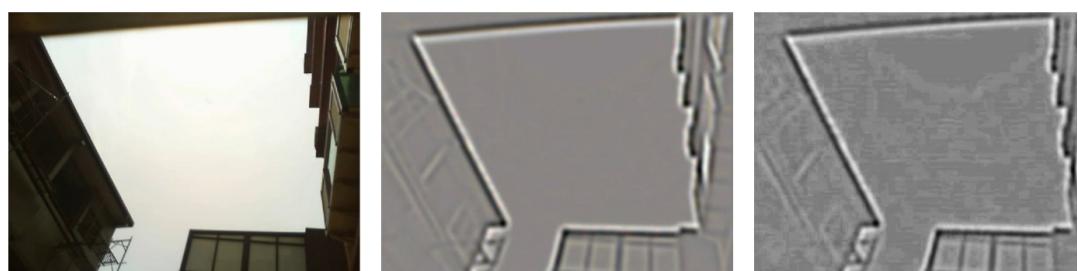
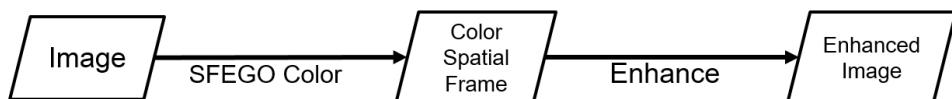


Fig. 33. Invisible Space Ship Analysis

7. Reference

- [1] Huang, Norden E., et al. "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis." *Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences*. Vol. 454. No. 1971. The Royal Society, 1998.
- [2] Wu, Zhaohua, and Norden E. Huang. "Ensemble empirical mode decomposition: a noise-assisted data analysis method." *Advances in adaptive data analysis* 1.01 (2009): 1-41.
- [3] Nunes, Jean Claude, et al. "Image analysis by bidimensional empirical mode decomposition." *Image and vision computing* 21.12 (2003): 1019-1026.
- [4] Wu, Zhaohua, Norden E. Huang, and Xianyao Chen. "The multi-dimensional ensemble empirical mode decomposition method." *Advances in Adaptive Data Analysis* 1.03 (2009): 339-372.
- [5] Waskito, Pulung, et al. "Parallelizing Hilbert-Huang transform on a GPU." *Networking and Computing (ICNC), 2010 First International Conference on*. IEEE, 2010.
- [6] Chang, Li-Wen, et al. "Parallel implementation of multi-dimensional ensemble empirical mode decomposition." *Acoustics, speech and signal processing (ICASSP), 2011 IEEE international conference on*. IEEE, 2011.
- [7] Wielgus, Maciej, et al. "Fast and adaptive bidimensional empirical mode decomposition for the real-time video fusion." *Information Fusion (FUSION), 2012 15th International Conference on*. IEEE, 2012.
- [8] Wang, Yung-Hung, et al. "On the computational complexity of the empirical mode decomposition algorithm." *Physica A: Statistical Mechanics and its Applications* 400 (2014): 159-167.
- [9] Feng, Jiaxin, Zhaohua Wu, and Guosheng Liu. "Fast multidimensional ensemble empirical mode decomposition using a data compression technique." *Journal of Climate* 27.10 (2014): 3492-3504.
- [10] Huang, Kevin PY, Charles HP Wen, and Herming Chiueh. "Flexible Parallelized Empirical Mode Decomposition in CUDA for Hilbert Huang Transform." *High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC, CSS, ICSS), 2014 IEEE Intl Conf on*. IEEE, 2014.
- [11] Shen, Bo-Wen, et al. "Parallel Implementation of the Ensemble Empirical Mode Decomposition (PEEMD) and Its Application for Earth Science Data Analysis." *Computing in Science & Engineering* (2017).
- [12] Ahonen, Timo, Abdennour Hadid, and Matti Pietikainen. "Face description with local binary patterns: Application to face recognition." *IEEE transactions on pattern analysis and machine intelligence* 28.12 (2006): 2037-2041.