

Analisis Wavelet 2D untuk Citra Photo

Seng Hansun

Program Studi Teknik Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
hansun@umn.ac.id

Diterima 4 Mei 2012
Disetujui 18 Mei 2012

Abstract—Wavelet atau gelombang pendek biasa digunakan dalam analisa sinyal, analisa numeris, maupun sistem kendali. Dalam penerapannya di analisis sinyal, wavelet dapat diterapkan untuk menganalisis citra dua dimensi dengan tujuan merekonstruksi citra dalam ukuran yang lebih kecil. Berbagai perangkat lunak juga telah mendukung analisa wavelet untuk pemrosesan sinyal ini, salah satunya wavelet toolbox dari MATLAB. Pada tulisan ini, penulis mencoba untuk merekonstruksi sebuah citra dua dimensi dengan analisis wavelet menggunakan MATLAB.

Index Terms—wavelet transform, citra photo, 2D, MATLAB

I. PENDAHULUAN

Citra adalah sebuah desain atau representasi yang dibuat dengan berbagai sarana, seperti lukisan, gambar, atau photo [1]. Berbagai penelitian terkait analisa citra dua dimensi telah umum dilakukan oleh para peneliti. Salah satu tujuannya adalah untuk merekonstruksi citra secara utuh dalam ukuran *file* yang lebih kecil, atau kompresi.

Salah satu metode yang digunakan dalam analisa citra dua dimensi adalah dengan menggunakan *wavelet transform*. Sejak keluarga *wavelet* pertama diperkenalkan oleh Alfrd Haar pada 1909, perkembangan teori *wavelet* sendiri telah berkembang dengan pesat, seperti yang dapat dilihat pada tabel di bawah.

Tabel 1. Sejarah perkembangan *wavelet* [2]

| Tahun | Perkembangan Wavelet | Keterangan |
|-------|----------------------|-------------------------------------|
| 1910 | Haar | |
| 1981 | Morlet | Konsep wavelet diperkenalkan |
| 1984 | Morlet dan Grossman | Wavelet |
| 1985 | Meyer | Orthogonal wavelet |
| 1987 | | Konferensi internasional di Prancis |
| 1988 | Mallat dan Meyer | Multiresolution |
| 1988 | Daubechies | Compact support orthogonal wavelet |
| 1989 | Mallet | Fast wavelet transform |

Penerapan *wavelet* banyak digunakan dalam analisa numeris, analisa sinyal, dan sistem kendali. Dalam penerapan di bidang analisis sinyal, *wavelet* banyak digunakan untuk kompresi audio, kompresi gambar dan video, klasifikasi tekstur, *denoising*, dan *fingerprints* [3].

Perangkat lunak yang dapat digunakan untuk membantu analisa dengan menggunakan *wavelet transform* telah banyak dikembangkan oleh peneliti maupun perusahaan. HP Laboratories Japan telah mengembangkan *software toolbox* di dalam sistem Khoros untuk membantu analisa *wavelet* [4]. Khoros *system* sendiri merupakan suatu lingkungan pengembangan *software* terintegrasi untuk pengolahan dan visualisasi informasi, berdasarkan *X Window System*. Selain itu, MATLAB sebagai salah satu *software* analisa terkemuka juga menyediakan *wavelet toolbox* untuk melakukan analisa *wavelet* di MATLAB [5].

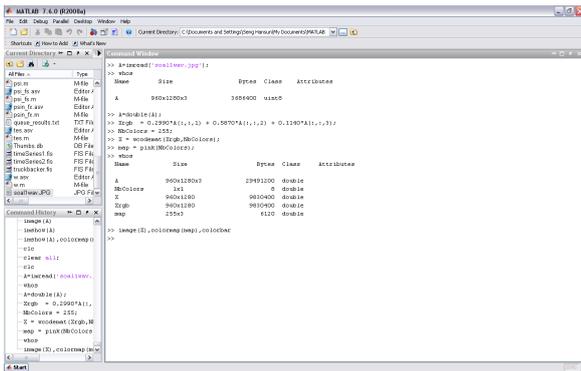
Pada tulisan ini, penulis mencoba untuk merekonstruksi citra dua dimensi, yakni photo diri, dengan menggunakan analisis *wavelet* dibantu perangkat lunak MATLAB. Terdapat dua pendekatan yang dapat dilakukan, yakni melalui *command line* dan *graphical interface* yang telah disediakan oleh MATLAB.

II. ANALISIS WAVELET 2D DENGAN COMMAND LINE

Berikut langkah-langkah yang dapat dilakukan untuk melakukan analisis *wavelet* untuk citra photo dua dimensi dengan menggunakan *command line*.

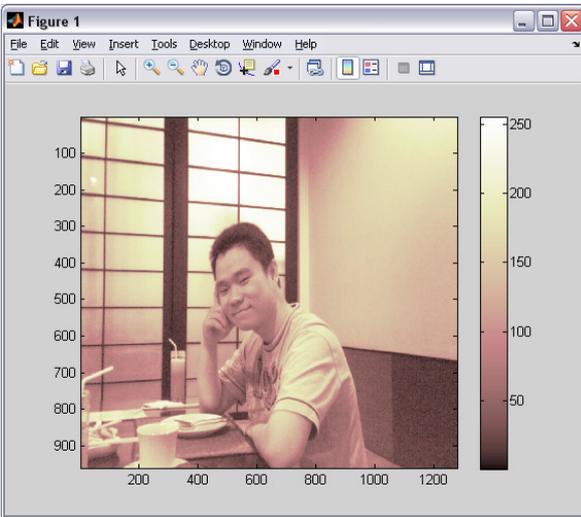
1. *Load* gambar ke dalam *Command Window* MATLAB

Gambar yang dapat ditransformasikan adalah gambar yang memiliki indeks (*indexed image*). Karena gambar yang digunakan adalah gambar asli (*truecolor image*) yang direpresentasikan oleh array `mxnx3` dari uint8, maka gambar tersebut harus di-*convert* terlebih dahulu menjadi *indexed image*. Perhatikan gambar di bawah.



Gambar 1. Perintah untuk mengubah *truecolor image* menjadi *indexed image*

Gambar *indexed* yang terbentuk terlihat seperti di bawah.



Gambar 2. *Indexed image*

2. Konversikan *indexed image* menjadi *grayscale image*

Jika *colormap image* sudah *smooth*, maka transformasi *Wavelet* dapat langsung diterapkan pada *indexed image* tersebut, sebaliknya jika tidak maka *image* tersebut harus di-convert terlebih dahulu menjadi *grayscale image*. Pada langkah konversi sebelumnya, kita telah langsung mengubah gambar asli menjadi *indexed image* dengan format *grayscale*. Perhatikan bahwa *colormap* gambar di atas telah *smooth*.

3. Lakukan *single-level Wavelet decomposition*

Untuk melakukan dekomposisi *level* tunggal dari gambar di atas dengan menggunakan *Wavelet bior3.7* (salah satu keluarga *Wavelet*), ketik perintah berikut:

```
[cA1,cH1,cV1,cD1] = dwt2(X,'bior3.7');
```

Perintah ini membangkitkan matriks koefisien dari bagian *smooth* (cA1) dan *detail horizontal* (cH1), *vertical* (cV1), dan *diagonal* (cD1) pada *level* satu.

4. Konstruksi dan tampilkan bagian *smooth* dan *detail* dari matriks koefisien

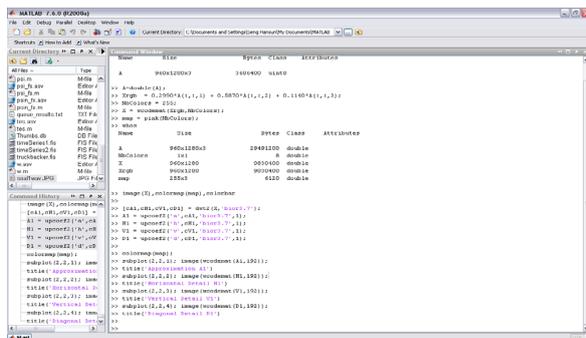
Untuk mengkonstruksi bagian *smooth* dan *detail level* satu (A1, H1, V1, D1) dari koefisien cA1, cH1, CV1, dan cD1, ketik perintah berikut:

```
A1 = upcoef2('a',cA1,'bior3.7',1);
H1 = upcoef2('h',cH1,'bior3.7',1);
V1 = upcoef2('v',cV1,'bior3.7',1);
D1 = upcoef2('d',cD1,'bior3.7',1);
```

Untuk menampilkan hasil dekomposisi *level* satu, ketik:

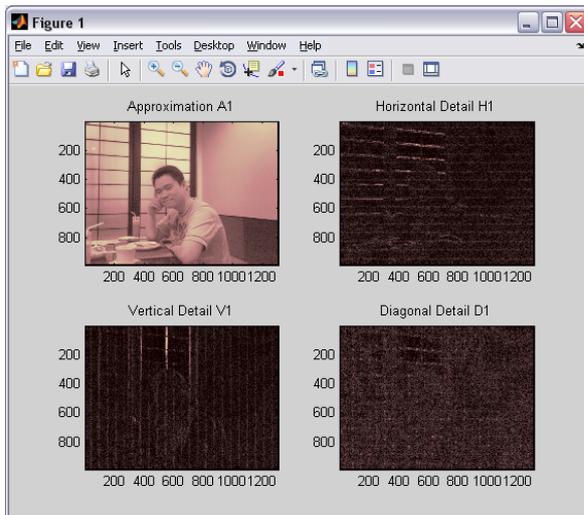
```
colormap(map);
subplot(2,2,1); image(wcodemat(A1,192));
title('Approximation A1')
subplot(2,2,2); image(wcodemat(H1,192));
title('Horizontal Detail H1')
subplot(2,2,3); image(wcodemat(V1,192));
title('Vertical Detail V1')
subplot(2,2,4); image(wcodemat(D1,192));
title('Diagonal Detail D1')
```

Perhatikan gambar di bawah.



Gambar 3. Perintah untuk menampilkan hasil dekomposisi *level* satu

Hasil dekomposisi *level* 1 yang terbentuk terlihat seperti berikut.

Gambar 4. Hasil dekomposisi *level* satu

5. Lakukan *multi-level Wavelet decomposition*

Untuk melakukan dekomposisi *level 2* dari gambar (menggunakan keluarga *Wavelet bior3.7*), ketik perintah berikut:

$$[C,S] = wavedec2(X,2,'bior3.7');$$

dimana X adalah matriks *image* asli dan 2 adalah *level* dekomposisi.

6. Ekstraksikan koefisien-koefisien bagian *smooth* dan *detail*

Untuk mengekstraksikan koefisien *smooth level 2* dari C, ketik:

$$cA2 = appcoef2(C,S,'bior3.7',2);$$

Untuk mengekstraksikan koefisien *detail level 1* dan *level 2* dari C, ketik:

$$[cH2,cV2,cD2] = detcoef2('all',C,S,2);$$

$$[cH1,cV1,cD1] = detcoef2('all',C,S,1);$$

7. Konstruksikan kembali bagian *smooth level 2* dan *detail level 1* dan *level 2*

Untuk mengkonstruksi kembali bagian *smooth level 2* dari C, ketik:

$$A2 = wrcoef2('a',C,S,'bior3.7',2);$$

Untuk mengkonstruksi kembali bagian *detail level 1* dan *level 2* dari C, ketik:

$$H1 = wrcoef2('h',C,S,'bior3.7',1);$$

$$V1 = wrcoef2('v',C,S,'bior3.7',1);$$

$$D1 = wrcoef2('d',C,S,'bior3.7',1);$$

$$H2 = wrcoef2('h',C,S,'bior3.7',2);$$

$$V2 = wrcoef2('v',C,S,'bior3.7',2);$$

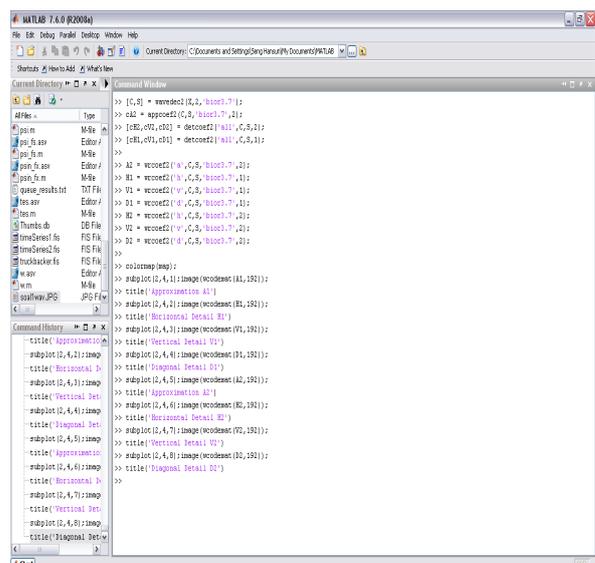
$$D2 = wrcoef2('d',C,S,'bior3.7',2);$$

8. Tampilkan hasil dari *multi-level decomposition*

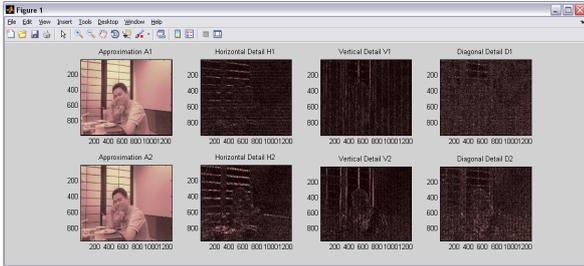
Untuk menampilkan hasil dekomposisi *level 2*, ketik:

```
colormap(map);
subplot(2,4,1);image(wcodemat(A1,192));
title('Approximation A1')
subplot(2,4,2);image(wcodemat(H1,192));
title('Horizontal Detail H1')
subplot(2,4,3);image(wcodemat(V1,192));
title('Vertical Detail V1')
subplot(2,4,4);image(wcodemat(D1,192));
title('Diagonal Detail D1')
subplot(2,4,5);image(wcodemat(A2,192));
title('Approximation A2')
subplot(2,4,6);image(wcodemat(H2,192));
title('Horizontal Detail H2')
subplot(2,4,7);image(wcodemat(V2,192));
title('Vertical Detail V2')
subplot(2,4,8);image(wcodemat(D2,192));
title('Diagonal Detail D2')
```

Perhatikan gambar di bawah.

Gambar 5. Perintah untuk menampilkan hasil dekomposisi *level* dua

Hasil dekomposisi *level* dua yang terbentuk seperti gambar berikut.

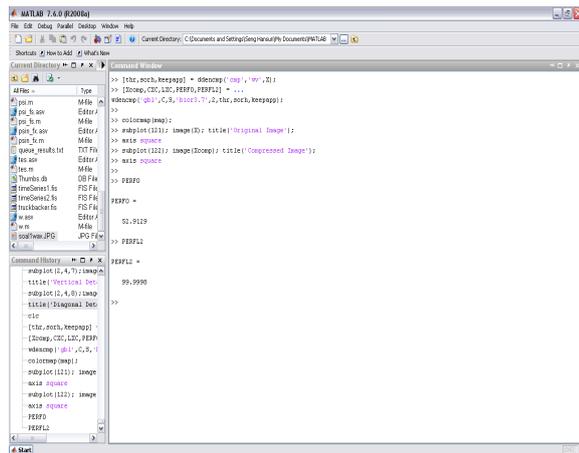


Gambar 6. Hasil dekomposisi *level* dua

9. Kompresi gambar dan tampilan

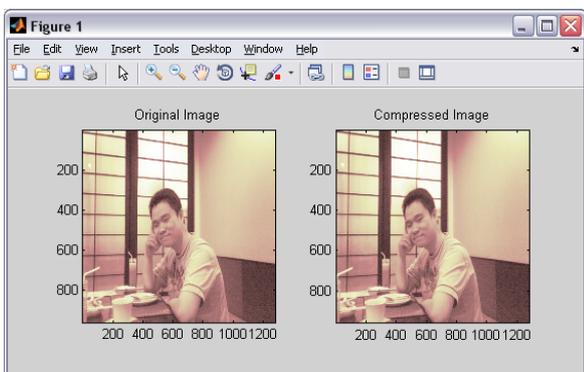
Untuk mengkompres gambar asli X, gunakan perintah *ddencmp* untuk menghitung parameter *default* dan perintah *wdencomp* untuk melakukan kompresi. Selanjutnya, tampilkan gambar asli serta gambar hasil kompresi.

Perhatikan gambar di bawah.



Gambar 7. Perintah untuk melakukan kompresi

Hasil yang diperoleh seperti berikut.



Gambar 8. Gambar hasil kompresi dengan dekomposisi *wavelet*

III. ANALISIS WAVELET 2D DENGAN GRAPHICAL INTERFACE

Selain menggunakan *command line* seperti yang telah dijelaskan di subbab sebelumnya, analisis *wavelet* dua dimensi juga dapat dilakukan dengan menggunakan *graphical user interface* (GUI) yang telah disediakan oleh MATLAB. Langkah-langkah pengerjaannya seperti yang dijelaskan berikut.

1. Jalankan 2-D Wavelet Analysis Tool

Dari MATLAB *prompt*, ketik: *wavemenu*

Wavelet Toolbox Main Menu akan muncul, seperti gambar berikut.



Gambar 9. *Wavelet toolbox main menu*

Klik menu Wavelet 2-D. Discrete Wavelet analysis tool untuk data gambar 2 dimensi akan muncul.

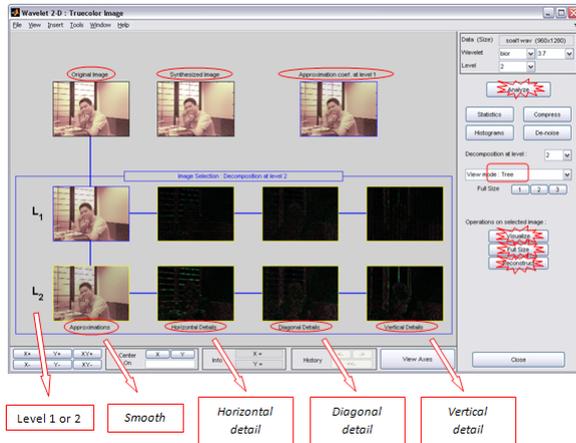
2. Load gambar

Sebelum me-load gambar, harus diperhatikan bahwa gambar yang di-load adalah gambar yang telah di-convert menjadi *indexed image* dengan format *grayscale* (*colormap 'smooth'*). Terlebih dahulu simpan gambar yang telah di-convert tadi dalam bentuk yang kompatibel dengan *Wavelet Toolbox* GUI.

Perhatikan gambar di bawah.

Visualize, menampilkan dalam ukuran penuh dengan menggunakan tombol *Full Size*, dan merekonstruksi hasil dekomposisi pada tiap posisi dengan tombol *Reconstruct*.

Perhatikan gambar berikutnya.



Gambar 14. *Tree mode*

IV. KESIMPULAN

Dari hasil percobaan yang telah dilakukan, dapat disimpulkan bahwa analisis *wavelet* berhasil dilakukan untuk merekonstruksi citra atau gambar dua dimensi. Dengan bantuan perangkat lunak MATLAB, analisis *wavelet* dua dimensi menghasilkan citra hasil dekomposisi yang memperlihatkan bagian *smooth* dan *detail* dari citra aslinya, dan dapat dilakukan kompresi dengan baik. MATLAB juga menyediakan dua cara pemanfaatan analisis *wavelet*, yakni dengan menggunakan *command line* dan *graphical user interface* yang dapat memudahkan *user* dalam melakukan analisis.

DAFTAR PUSTAKA

- [1] <http://www.merriam-webster.com/dictionary/picture>
- [2] C. Lin and Liu, "A Tutorial of the Wavelet Transform", February 23, 2010.
- [3] R.J.E. Merry, "Wavelet Theory and Applications: A Literature Study", Eindhoven University of Technology, June 2005.
- [4] D.T.L. Lee and A. Yamamoto, "Wavelet Analysis: Theory and Applications", Hewlett-Packard Journal, December 1994.
- [5] M. Misiti, Y. Misiti, G. Oppenheim, J-M. Poggi, "Wavelet Toolbox: For Use with MATLAB", MathWorks, March 1996.