Melhorando a Exatidão de Cálculo da Transformada Wavelet de Daubechies

Orientando: **Vinícius Rodrigues dos Santos**
Orientador: **Renata Hax Sander Reiser**
Coorientador: **Maurício Lima Pilla**
Colaborador: **Alice de Jesus Kozakevicius**

Universidade Federal de Pelotas
Ciência da Computação
Centro de Desenvolvimento Tecnológico

30/09/2016 - PELOTAS / RS

Contextualização

Transformadas Wavelets

- As Transformadas Wavelets (TWs) são ferramentas matemáticas para a **decomposição** hierárquica de funções.
- Utilizadas em **análise** de sinais para identificação de detalhes [1].
- Uma característica importante das TWs é a **composição** dos sinais transformados para os originais.

Aplicações

- **compressão de imagens** [2], remoção de ruído [3].
- Análise de sinais de eletroencefalogramas [4].
- Reconstrução de marcas d'água invisíveis em imagens digitais [5].
- Filtragem sinais cardíacos através de Algoritmos Adaptativos [6].
- Descrição de parâmetros estatísticos em exames de mamografia [7].

Referências de Aplicações

[1] A.E. Brito and O. Kosheleva.
Interval + image = wavelet: for image processing under interval uncertainty, wavelets are optimal.
Reliable Computing, pages 291–301, 1998.

[2] C. Christopoulos, A. Skodras, and T. Ebrahimi.
The jpeg2000 still image coding system: an overview.
Consumer Electronics, IEEE Transactions on, 46(4):1103–1127, Nov 2000.

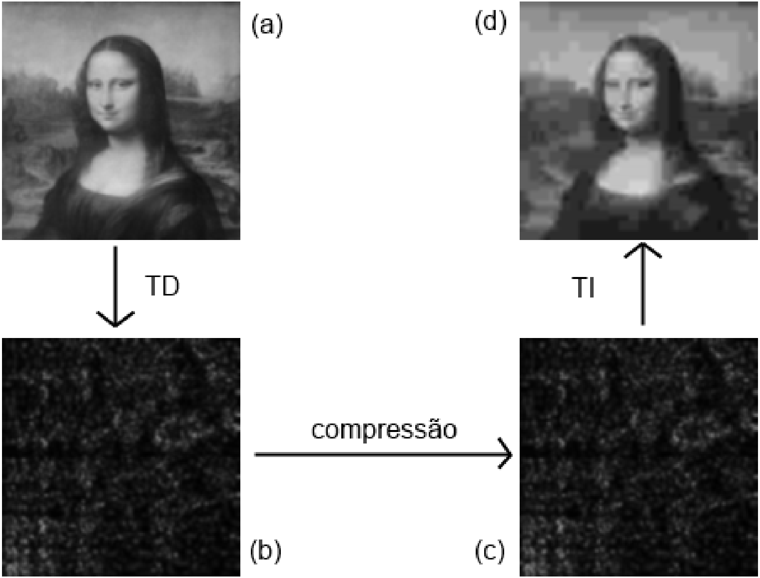
[3] D. Donoho.
De-noising by soft-thresholding.
IEEE Transactions on Information Theory, 41:613–627, 1995.

[4] T.L.T. Silveira, A.J. Kozakevicius, and C.R. Rodrigues.
Classificação de estágios de sono através da aplicação de transformada wavelet discreta sobre um único canal de eletroencefalograma.
Master's thesis, 2016.

[5] T. Minamoto and K. Aoki.
A blind digital image watermarking using interval wavelet decomposition.
Int. Journal of Signal proc., Image proc. and Pattern recognition, 3(2):59–72, 2010.

[6] G. Perin and A. J. Kozakevicius.
Filtragem wavelet de sinais cardíacos através de algoritmos adaptativos.
RITA (in portuguese), 20(3):95–111, 2013.

[7] M. Asadzadeh, E. Hashemi, and A. Kozakevicius.
On efficiency of combined daubechies wavelets and statistical parameters applied in mammography.
Applied and Computational Mathematics, 12(3):289–306, 2013.


(a)
(d)
TD
TI
compressão
(b)
(c)

Principal Contribuição

- Desenvolvimento da biblioteca **Int-DWTs** com implementações intervalares e otimizadas para Transformadas Discretas Wavelet (TDWs). [8, 9]
- Prover resultados intervalares confiáveis e automaticamente validados a aplicações que utilizam de TWDs, com foco em Computação Científica. [10, 11]

Objetivos Alcançados

- Estudo, **otimização** e implementação da **extensão intervalar** para as Transformadas Wavelet de **Haar** (TWH) e de **Daubechies** (TWD). [12]
- Implementar as seguintes abordagens:
  - **Abordagens**: não-normalizada não-padrão, não-normalizada padrão, normalizada não-padrão e normalizada padrão;
  - **Etapas**: transformação direta (decomposição) e de transformação inversa (composição).

Funcionamento Básico

Decomposição	Composição
C’[i] = (C[2i] + C[2i + 1]) / 2 C’[n/2 + i] = (C[2i] - C[2i + 1]) / 2	C’[2*i] = C[i] + C[n + i] C’[2*i + 1] = C[i] - C[n + i]

n = tamanho do vetor; i = 0... n/2 - 1;

—	(de escala)	(wavelets)
Nível	**Médias**	**Diferenças**
2	[9 7 3 5]	-
1	[8 4]	[1 -1]
0	[6]	[2]

- Vetor original: [9 7 3 5]
- Vetor transformado: [6 2 1 -1]

procedure *DecompositionStep*(C: **array** $[1..h]$ **of reals**)
  for $i \leftarrow 1$ **to** $h/2$ **do**
    $C'[i] \leftarrow (C[2i-1] + C[2i])/\underline{\sqrt{2}}$
    $C'[h/2+i] \leftarrow (C[2i-1] - C[2i])/\underline{\sqrt{2}}$
  end for
  $C \leftarrow C'$
end procedure

procedure *Decomposition*(C: **array** $[1..h]$ **of reals**)
  $C \leftarrow C/\sqrt{h}$ *(normalize input coefficients)*
  while $h > 1$ **do**
    DecompositionStep$(C[1..h])$
    $h \leftarrow h/2$
  end while
end procedure

Transformada Wavelet de Daubechies

$$h = \left(\frac{1+\sqrt{3}}{4\sqrt{2}}, \quad \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad \frac{3-\sqrt{3}}{4\sqrt{2}}, \quad \frac{1-\sqrt{3}}{4\sqrt{2}} \right) \tag{1}$$

$$g = \left(\frac{1-\sqrt{3}}{4\sqrt{2}}, \quad \frac{-3+\sqrt{3}}{4\sqrt{2}}, \quad \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad \frac{-1-\sqrt{3}}{4\sqrt{2}} \right) \tag{2}$$

$$ih = \left(\frac{3-\sqrt{3}}{4\sqrt{2}}, \quad \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad \frac{1-\sqrt{3}}{4\sqrt{2}} \right) \tag{3}$$

$$ig = \left(\frac{1-\sqrt{3}}{4\sqrt{2}}, \quad \frac{-1-\sqrt{3}}{4\sqrt{2}}, \quad \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad \frac{-1+\sqrt{3}}{4\sqrt{2}} \right) \tag{4}$$

DecompositionStep $(C[1..n])$

$i \leftarrow 1$;
$j \leftarrow 1$;
$half \leftarrow n/2$;
while $j <= n - 3$ **do**
    $C'[i] \leftarrow C[j].h1 + C[j+1].h2 + C[j+2].h3 + C[j+3].h4$;
    $C'[i + half] \leftarrow C[j].g1 + C[j+1].g2 + C[j+2].g3 + C[j+3].g4$
    $j \leftarrow j + 2$;
    $i \leftarrow i + 1$;
end
$C'[i] \leftarrow C[n-1].h1 + C[n].h2 + C[1].h3 + C[2].h4$;
$C'[i + half] \leftarrow C[n-1].g1 + C[n].g2 + C[1].g3 + C[2].g4$;
$C \leftarrow C'$;

$$h = a\left(\frac{1+\sqrt{3}}{4\sqrt{2}}\right)+b\left(\frac{3+\sqrt{3}}{4\sqrt{2}}\right)+c\left(\frac{3-\sqrt{3}}{4\sqrt{2}}\right)+d\left(\frac{1-\sqrt{3}}{4\sqrt{2}}\right) \tag{5}$$

$$g = a\left(\frac{1-\sqrt{3}}{4\sqrt{2}}\right)+b\left(\frac{\sqrt{3}-3}{4\sqrt{2}}\right)+c\left(\frac{3+\sqrt{3}}{4\sqrt{2}}\right)+d\left(\frac{-1-\sqrt{3}}{4\sqrt{2}}\right) \tag{6}$$

$$ih = a\left(\frac{3-\sqrt{3}}{4\sqrt{2}}\right)+b\left(\frac{3+\sqrt{3}}{4\sqrt{2}}\right)+c\left(\frac{1+\sqrt{3}}{4\sqrt{2}}\right)+d\left(\frac{1-\sqrt{3}}{4\sqrt{2}}\right) \tag{7}$$

$$ig = a\left(\frac{1-\sqrt{3}}{4\sqrt{2}}\right)+b\left(\frac{-1-\sqrt{3}}{4\sqrt{2}}\right)+c\left(\frac{3+\sqrt{3}}{4\sqrt{2}}\right)+d\left(\frac{\sqrt{3}-3}{4\sqrt{2}}\right) \tag{8}$$

Otimizações

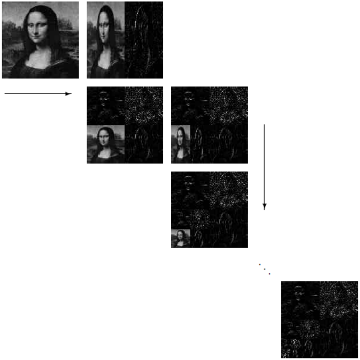
Base de Otimização

$$R = 2^{\frac{-j}{2}}$$

$$[9 \quad 7 \quad 3 \quad 5] \rightarrow [6 \quad 2 \quad 1 \quad -1] \rightarrow [6 \quad 2 \quad \frac{1}{\sqrt{2}} \quad \frac{-1}{\sqrt{2}}]$$

Entrada → Decomposição → Normalização

(a) Algoritmo Padrão



(b) Algoritmo Não-Padrão

$$\begin{aligned} h &= a\left(\frac{1+\sqrt{3}}{4\sqrt{2}}\right) + b\left(\frac{3+\sqrt{3}}{4\sqrt{2}}\right) + c\left(\frac{3-\sqrt{3}}{4\sqrt{2}}\right) + d\left(\frac{1-\sqrt{3}}{4\sqrt{2}}\right) \\ &= \frac{a\left(1+\sqrt{3}\right) + b\left(3+\sqrt{3}\right) + c\left(3-\sqrt{3}\right) + d\left(1-\sqrt{3}\right)}{4\sqrt{2}} \\ &= \frac{a + a\sqrt{3} + 3b + b\sqrt{3} + 3c - c\sqrt{3} + d - d\sqrt{3}}{4\sqrt{2}} \\ &= \frac{a + d + 3\,(b+c) + \sqrt{3}\,(a+b-c-d)}{4\sqrt{2}} \end{aligned} \tag{9}$$

$$
\begin{aligned}
g &= a\left(\frac{1-\sqrt{3}}{4\sqrt{2}}\right) + b\left(\frac{\sqrt{3}-3}{4\sqrt{2}}\right) + c\left(\frac{3+\sqrt{3}}{4\sqrt{2}}\right) + d\left(\frac{-1-\sqrt{3}}{4\sqrt{2}}\right) \\
&= \frac{a\left(1-\sqrt{3}\right) + b\left(\sqrt{3}-3\right) + c\left(3+\sqrt{3}\right) + d\left(-1-\sqrt{3}\right)}{4\sqrt{2}} \\
&= \frac{a - a\sqrt{3} + b\sqrt{3} - 3b + 3c + c\sqrt{3} - d - d\sqrt{3}}{4\sqrt{2}} \\
&= \frac{a - d + 3\left(c-b\right) + \sqrt{3}\left(b+c-a-d\right)}{4\sqrt{2}}
\end{aligned} \tag{10}
$$

$$
\begin{aligned}
ih &= a\left(\frac{3-\sqrt{3}}{4\sqrt{2}}\right)+b\left(\frac{3+\sqrt{3}}{4\sqrt{2}}\right)+c\left(\frac{1+\sqrt{3}}{4\sqrt{2}}\right)+d\left(\frac{1-\sqrt{3}}{4\sqrt{2}}\right)\\
&= \frac{a\left(3-\sqrt{3}\right)+b\left(3+\sqrt{3}\right)+c\left(1+\sqrt{3}\right)+d\left(1-\sqrt{3}\right)}{4\sqrt{2}}\\
&= \frac{3a-a\sqrt{3}+3b+b\sqrt{3}+c+c\sqrt{3}+d-d\sqrt{3}}{4\sqrt{2}}\\
&= \frac{c+d+3\,(a+b)+\sqrt{3}\,(b+c-a-d)}{4\sqrt{2}} \qquad (11)
\end{aligned}
$$

$$
\begin{aligned}
ig &= a\left(\frac{1-\sqrt{3}}{4\sqrt{2}}\right)+b\left(\frac{-1-\sqrt{3}}{4\sqrt{2}}\right)+c\left(\frac{3+\sqrt{3}}{4\sqrt{2}}\right)+d\left(\frac{\sqrt{3}-3}{4\sqrt{2}}\right) \\
&= \frac{a\left(1-\sqrt{3}\right)+b\left(-1-\sqrt{3}\right)+c\left(3+\sqrt{3}\right)+d\left(\sqrt{3}-3\right)}{4\sqrt{2}} \\
&= \frac{a-a\sqrt{3}-b-b\sqrt{3}+3c+c\sqrt{3}-3d+d\sqrt{3}}{4\sqrt{2}} \\
&= \frac{a-b+3\left(c-d\right)+\sqrt{3}\left(c+d-a-b\right)}{4\sqrt{2}} \qquad (12)
\end{aligned}
$$

Padrão de fatores de normalização (Algoritmo Padrão)

	... 3	2	1
⋮			
3	(3,3)	(3,2)	(3,1)
2	(2,3)	(2,2)	(2,1)
1	(1,3)	(1,2)	(1,1)

(a)

$$(j', j'')$$

(b)

$$2^{-\frac{(j'+j'')}{2}}$$

(c)

(a) níveis de aplicação, (b) combinação de níveis, (c) regra de normalização.

Metodologia

Estudo de caso:

- **Performance**: Relação de desempenho das execuções.
- **Accuracy**: Relação entre o maior diâmetro de intervalo de erro na computação.
- **Metrics**: Aplicação de métricas (EUC, MSE e PSNR) para medição de qualidade.

$$\mathbf{EUC}(\tilde{\mathbf{Y}}, \mathbf{Y}) = \sqrt{\sum_{j=0}^{m} \sum_{i=0}^{n} \left(\tilde{\mathbf{Y}}_{ij} - \mathbf{Y}_{ij}\right)^2} \tag{13}$$

$$\mathbf{MSE}(\tilde{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{nm} \sum_{j=0}^{m} \sum_{i=0}^{n} (\tilde{\mathbf{Y}}_{ij} - \mathbf{Y}_{ij})^2 \tag{14}$$

$$\textbf{PSNR}(\tilde{\mathbf{Y}}, \mathbf{Y}_{ij}) = 10 \cdot \log_{10} \frac{MAX^2}{\mathbf{MSE}(\tilde{\mathbf{Y}}, \mathbf{Y})} \tag{15}$$

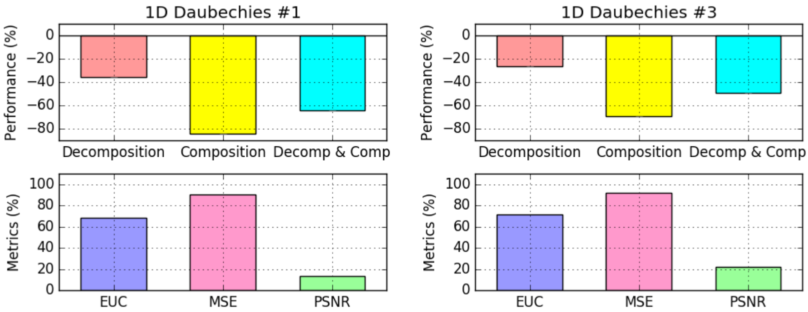
Configuração da Máquina

Processador:	Intel ® Core ™ i7 950 Processor @ 3.07GHz
Memória:	6GB DDR3 @ 1066MHz
SO:	Windows 10
Compilador:	Microsoft Visual C++ VS2013 (x64 Release)

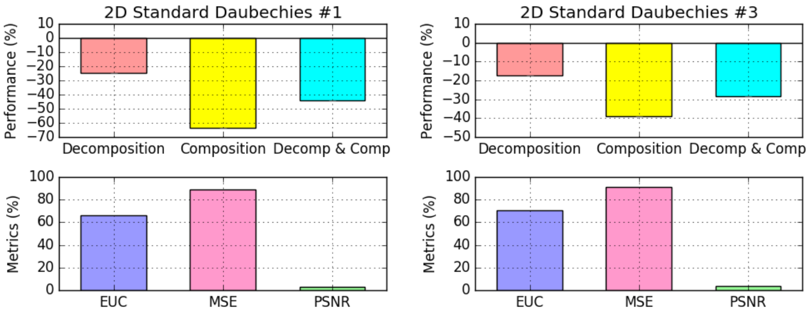
Parâmetros de Testes

- **TWD 1D e 2D**
  - **Teste #1**: 2^{20} valores aleatórios
  - ~~Teste #2~~: 2^{22} valores aleatórios
  - **Teste #3**: 2^{24} valores aleatórios

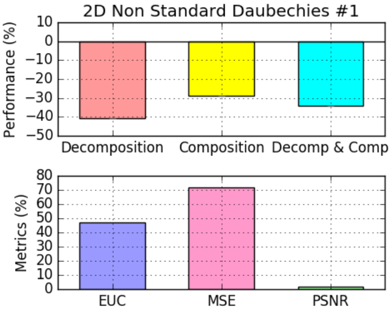
Testes #1 e #3 para a **TWD 1D**

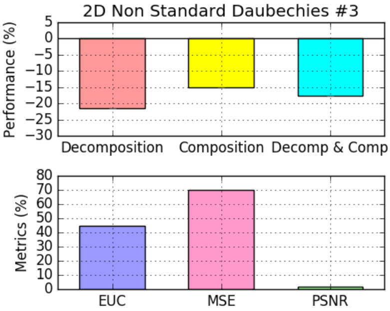


Testes #1 e #3 para a **TWD 2D Padrão**



Testes #1 e #3 para a **TWD 2D Não-Padrão**





Principais Publicações 2015-2016

- **ERAD 2015**: Estudo de Desempenho sobre a Biblioteca Int-Haar
- **SIM 2015**: Int-HWT: Increasing Performance and Exactitude of 1D and 2D Haar Wavelet Transforms
- **WEIT 2015**: Transformada de Haar Não Decimada Otimizada e sua Extensão Intervalar
- **ERAD 2016**: Transformada de Haar À Trous Otimizada e Análise de Desempenho
- **CNMAC 2016**: Otimização para Aumento de Exatidão da Transformada Wavelet de Haar À-Trous
- **TCC 2016/1**: Int-DWTs Library - Algebraic Simplifications Increasing Performance and Accuracy of Discrete Wavelet Transforms

Publicações Submetidas

- Int-HWT: Algebraic Simplifications Increasing Performance And Exactitude of the Haar Wavelet Transforms

Conclusão

- As otimizações propostas foram implementadas e validadas com sucesso.
- A TWD obteve perdas de desempenho porém com ganhos satisfatórios quanto a exatidão dos cálculos.
- A análise de complexidade algorítmica mostra a causa dos ganhos e perdas de desempenho.

Trabalhos Futuros

- O estudo, otimização e a implementação intervalar de outras Transformadas Discretas Wavelet.
- A implementação concorrente dos algoritmos da biblioteca Int-DWTs para execução em GPU.

Bibliografia I

[1] A.E. Brito and O. Kosheleva.
Interval + image = wavelet: for image processing under interval uncertainty, wavelets are optimal.
Reliable Computing, pages 291–301, 1998.

[2] C. Christopoulos, A. Skodras, and T. Ebrahimi.
The jpeg2000 still image coding system: an overview.
Consumer Electronics, IEEE Transactions on, 46(4):1103–1127, Nov 2000.

[3] D. Donoho.
De-noising by soft-thresholding.
IEEE Transactions on Information Theory, 41:613–627, 1995.

[4] T.L.T. Silveira, A.J. Kozakevicius, and C.R. Rodrigues.
Classificação de estágios de sono através da aplicação de transformada wavelet discreta sobre um único canal de eletroencefalograma.
Master's thesis, 2016.

[5] T. Minamoto and K. Aoki.
A blind digital image watermarking using interval wavelet decomposition.
Int. Journal of Signal proc., Image proc. and Pattern recognition, 3(2):59–72, 2010.

[6] G. Perin and A. J. Kozakevicius.
Filtragem wavelet de sinais cardíacos através de algoritmos adaptativos.
RITA (in portuguese), 20(3):95–111, 2013.

Bibliografia II

[7] M. Asadzadeh, E. Hashemi, and A. Kozakevicius.
On efficiency of combined daubechies wavelets and statistical parameters applied in mammography.
Applied and Computational Mathematics, 12(3):289–306, 2013.

[8] O. M. Nielsen.
Wavelets in scientific computing, 1998.
Supervisor: Per Christian Hansen, pcha@dtu.dk, DTU Compute.

[9] Ingrid Daubechies.
Orthonormal bases of compactly supported wavelets.
Communications on Pure and Applied Mathematics, 41(7):909–996, 1988.

[10] R.E. Moore.
Methods and Applications of Interval Analysis.
SIAM, Philadelphia, 1979.

[11] W. Hofschuster, W. Kramer, and M. Neher.
C-XSC and Closely Related Software Packages.
Numerical Validation in Current Hardware Architectures. Springer-Verlag, Universitat Wuppertal, 2008.

[12] E. J. Stollnitz, T. D. DeRose, and D. H. Salesin.
Wavelets for computer graphics: A primer, part 1.
IEEE Computer Graphics and Applications, 15(3):76–84, 1995.

Agradecimento e Dúvidas

Agradecimentos

- PROBIT/FAPERGS (08/2015 a 07/2016)
- Projeto PqG/2013 FAPERGS

Dúvidas

Contatos:

- vrdsantos@inf.ufpel.edu.br

Melhorando a Exatidão de Cálculo da Transformada Wavelet de Daubechies

Orientando: **Vinícius Rodrigues dos Santos**
Orientador: **Renata Hax Sander Reiser**
Coorientador: **Maurício Lima Pilla**
Colaborador: **Alice de Jesus Kozakevicius**

Universidade Federal de Pelotas
Ciência da Computação
Centro de Desenvolvimento Tecnológico

30/09/2016 - PELOTAS / RS

Métricas

Error

```
real INT_error(interval *x, int n)
{
    real r = 0.0;

    for (int i = 0; i < n; i++)
    if (INT_diameter(x[i]) > r) r = INT_diameter(x[i]);

    return r;
}
```

Distância Euclidiana

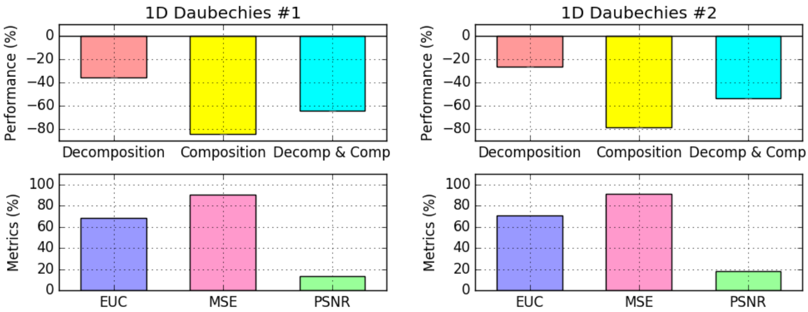
$$\mathbf{EUC}(\tilde{\mathbf{Y}}, \mathbf{Y}) = \sqrt{\sum_{j=0}^{m} \sum_{i=0}^{n} \left(\tilde{\mathbf{Y}}_{ij} - \mathbf{Y}_{ij}\right)^2}.$$

Mean Squared Error

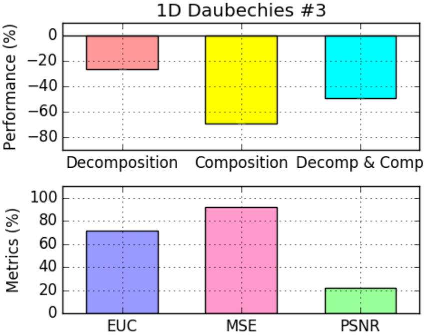
$$\mathbf{MSE}(\tilde{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{nm} \sum_{j=0}^{m} \sum_{i=0}^{n} (\tilde{\mathbf{Y}}_{ij} - \mathbf{Y}_{ij})^2$$

Peak Noise-To-Signal

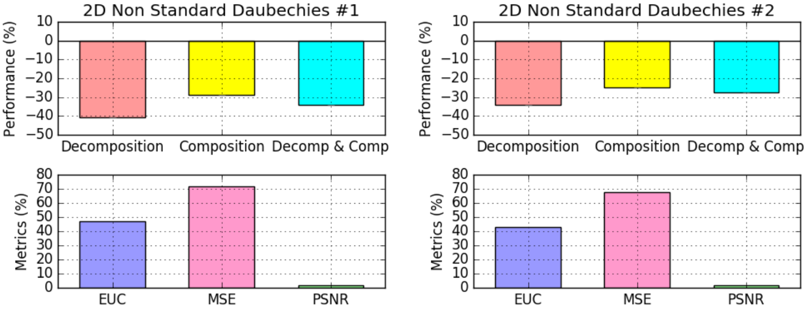
$$\textbf{PSNR}(\tilde{\mathbf{Y}}, \mathbf{Y}_{ij}) = 10 \cdot \log_{10} \frac{MAX^2}{\mathbf{MSE}(\tilde{\mathbf{Y}}, \mathbf{Y})}.$$



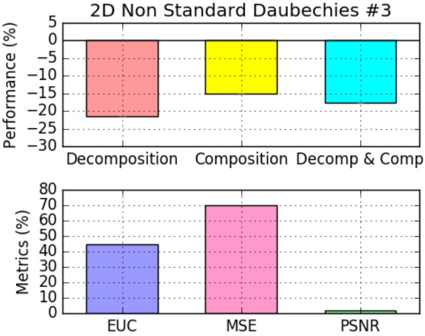
Testes #1 e #2 para a **TWD 1D** utilizando 1048576 e 4194304 valores aleatórios.



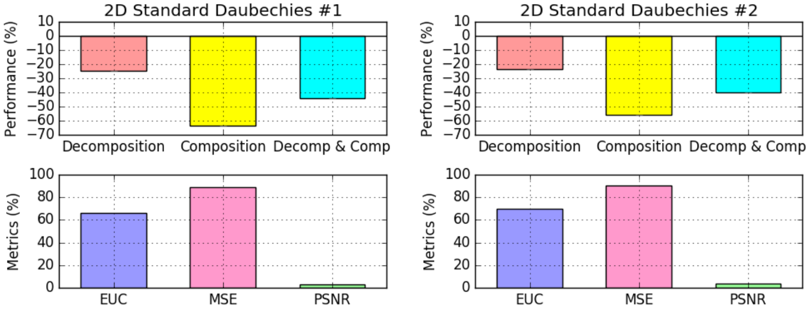
Teste #3 para a **TWD 1D** utilizando 16777216 valores aleatórios.



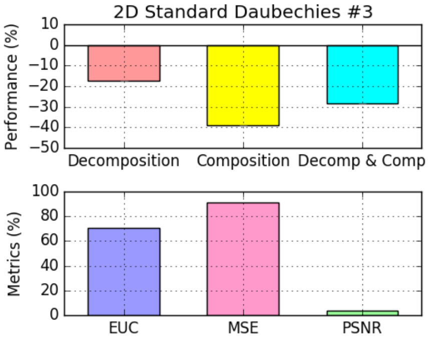
Testes #1 e #2 para a **TWD 2D Não-Padrão** utilizando 1024x1024 e 2048x2048 valores aleatórios.



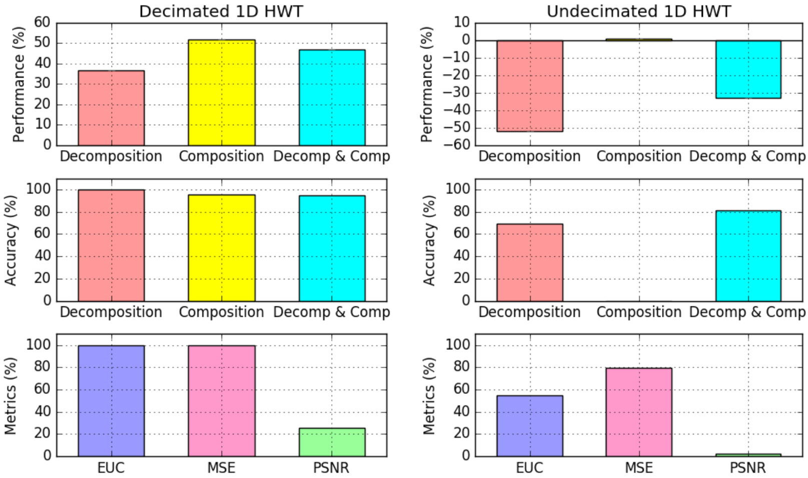
Teste #3 para a **TWD 2D Não-Padrão** utilizando 4096x4096 valores aleatórios.



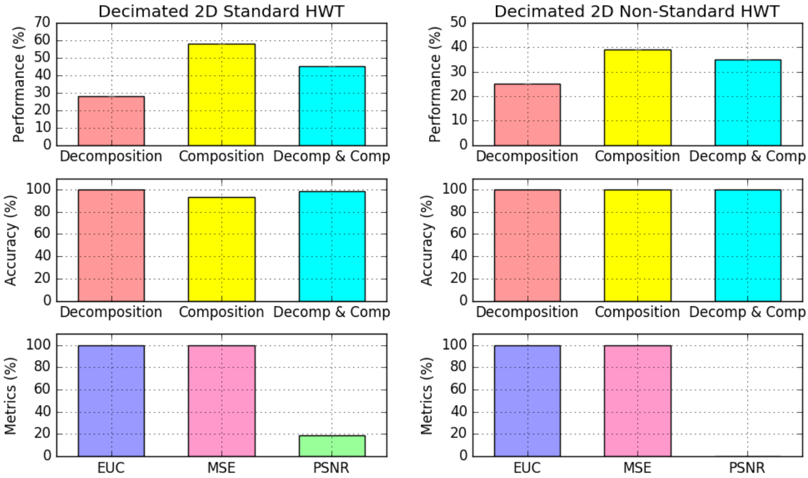
Testes #1 e #2 para a **TWD 2D Padrão** utilizando 1024x1024 e 2048x2048 valores aleatórios.



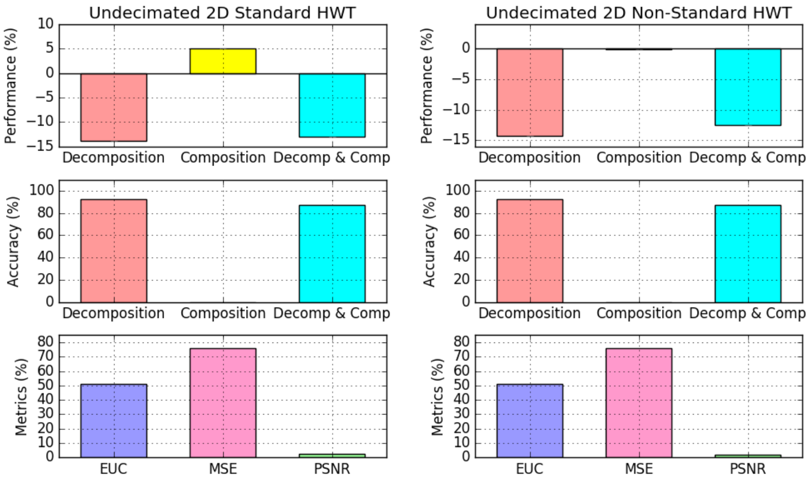
Teste #3 para a **TWD 2D Padrão** utilizando 4096x4096 valores aleatórios



Testes para a **TWH 1D** utilizando 1048576 valores aleatórios.



Testes para a **TWH 2D Padrão e Não-Padrão** utilizando 1024x1024 valores aleatórios.



Testes para a **TWH À-Trous 2D Padrão e Não-Padrão** utilizando 1024x1024 valores aleatórios.