

# Int-Haar: Extensão Intervalar da Transformada de Haar

Vinícius R. dos Santos<sup>1</sup>, Maurício Pilla<sup>1</sup>, Renata Reiser<sup>1</sup>, Alice Kozakevicius<sup>2</sup>

<sup>1</sup>Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico(CDTec), Universidade Federal de Pelotas (UFPEL)

<sup>2</sup>Universidade Federal de Santa Maria (UFSM), Departamento de Matemática  
{vrdsantos, pilla, reiser}@inf.ufpel.edu.br, alicek@ufsm.br

**Resumo.** Neste trabalho, introduz-se uma extensão intervalar da Transformada Wavelet de Haar (Haar Wavelet Transform - HWT), implementada utilizando C-XSC. Esta etapa consiste no primeiro estudo de caso para a implementação da biblioteca Int-DWTs, tendo por objetivo a obtenção de resultados intervalares para Transformadas Discretas Wavelet (Discrete Wavelet Transforms - DWTs).

## 1. Introdução

A qualidade dos resultados em Computação Científica (*Scientific Computing* - SC) depende do conhecimento e controle dos erros nos dados e processos computacionais. Este trabalho considera a Matemática Intervalar (*Interval Mathematics* - IM) e propõe uma solução baseada na aritmética de Moore [3]. Resultados intervalares carregam a segurança de sua qualidade e o grau de sua incerteza [3], pois o diâmetro do intervalo solução, além de conter as incertezas dos parâmetros de entrada, também é um indicativo da influência dos erros (intervalo de erro) dos dados de entrada e dos erros de arredondamento e truncamento, contidos no resultado final calculado.

A motivação do uso de técnicas intervalares integradas às Transformadas Discretas *Wavelets* (TDW) é prover resultados confiáveis e automaticamente validados às aplicações que necessitam dessas transformações, cujas áreas são as mais variadas, destacando-se no entanto processamento de sinais e imagens [5,6]. Assim, a principal meta deste trabalho consiste no desenvolvimento de **extensões intervalares de TDWs**, sendo inicialmente abordada a Transformada *Wavelet* de Haar (TWH). Nesta etapa, propõe-se uma **implementação da TWH utilizando C-XSC**, consistindo no primeiro estudo de caso para a implementação da biblioteca **Int-DWTs**, inserido no Projeto T-CSAPU (*Técnicas para Computação Sustentável em Ambientes Paralelos e Ubíquos*), visando a obtenção de resultados intervalares para validação automática das TDWs.

A metodologia de desenvolvimento do trabalho consiste no estudo e na análise dos algoritmos que viabilizam a computação da HWT incluindo sua versão ortonormal [4]. A partir dos algoritmos originais, desenvolveu-se uma biblioteca contendo as extensões intervalares dos mesmos, denominada *Int-Haar*.

Neste trabalho, considera-se a biblioteca C-XSC como suporte para a implementação da versão intervalar do algoritmo da HWT [4]. Esta biblioteca consiste em uma ferramenta para o desenvolvimento de algoritmos numéricos, integrando exatidão e verificação automática dos resultados, frequentemente aplicada em SC. Sendo compatível com C++, C-XSC permite a programação em alto nível de aplicações numéricas na mesma linguagem [2], agregando diversas bibliotecas suporte a SC.

Este artigo está organizado da seguinte forma: na Seção 2, a TWH unidimensional é apresentada. Após, na Seção 3, é discutida a TWH bidimensional. Os testes e resultados são apresentados na Seção 4. Finalmente, na Seção 5 as conclusões e os trabalhos futuros são apresentados.

## 2. TWH 1D

Segundo descrição em [1], dado um vetor inicial, a TWH 1D calcula médias e diferenças de cada par de elementos adjacentes deste vetor, gerando assim dois conjuntos de coeficientes: de escala (médias) e *wavelets* (diferenças). As médias calculadas podem ser usadas novamente, servindo como entrada para o cálculo do próximo nível de transformação, e assim recursivamente até obtermos um único coeficiente de escala – **processo de decomposição**.

Utilizando os conjuntos de coeficientes resultantes do processo de decomposição – também conhecido por **transformação direta** – é possível realizar-se a **transformação inversa** (composição), característica importante das transformadas *wavelet*. O algoritmo modelando o cálculo dos níveis de decomposição da TWH 1D está representado no pseudo-código na Figura 1.

De acordo com [4], no processo de composição, partindo-se do último nível da transformação, são armazenados os resultados das somas e subtrações entre os coeficientes de escala e os coeficientes *wavelet*. Recuperando-se desta forma a informação dos níveis superiores.

```
procedure DecompositionStep(C: array [1..h] of reals)
  for i ← 1 to h/2 do
    C'[i] ← (C[2i - 1] + C[2i])/ 2
    C'[h/2 + i] ← (C[2i - 1] - C[2i])/ 2
  end for
  C ← C'
end procedure
```

Figura 1: Procedimento para o cálculo dos níveis de decomposição da TWH 1D.

## 3. TWH 2D

A TWH também pode ser realizada utilizando vetores bidimensionais. De acordo com o procedimento descrito em [4], a TWH 2D pode ser aplicada por dois algoritmos distintos, os quais diferem-se em relação à ordem em que executam a decomposição pelas linhas e colunas da matriz.

A TWH 2D pode ser calculada por meio do algoritmo **padrão**, o qual executa primeiro a decomposição completa de todas as linhas da matriz e depois faz o procedimento análogo com todas as colunas. Outra opção é o algoritmo **não-padrão**, que intercala as operações ao decompor um nível de todas as linhas e de todas as colunas de cada vez. Os coeficientes de escala resultantes dessa combinação são então usados como entrada para o cálculo do próximo nível de transformação. O processo se repete até obtermos somente um único coeficiente de escala. Uma representação dos dois algoritmos supracitados é esquematizada na Figura 2.

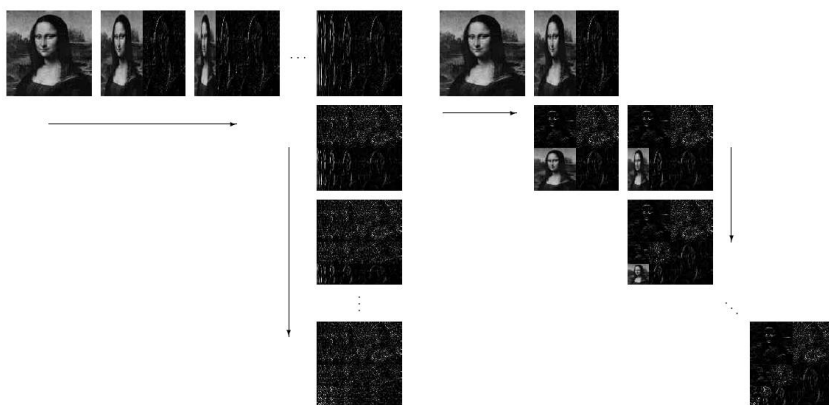


Figura 2: TWH 2D Padrão (esquerda) e Não-Padrão (direita).

Segundo [4], além dos algoritmos padrão e não-padrão, a TWH pode ser executada usando as abordagens **normalizada** e **não-normalizada**, que alteram os valores dos coeficientes da transformação. Logo, existem quatro maneiras de se calcular a TWH 2D, utilizando as combinações destas abordagens: (i) não-padrão, não-normalizada; (ii) não-padrão, normalizada; (iii) padrão, não-normalizada; (iv) padrão, normalizada.

#### 4. Testes e Resultados

A realização dos testes se baseia na aplicação das TDWs para compressão de imagens. Dessa maneira, os parâmetros intervalares para as execuções são obtidos a partir de valores pontuais contidos em pixels de imagens com escala-de-cinza de 8 bits, configurando assim intervalos pontuais, e usados como entrada para a TWH 2D intervalar.

A implementação dos procedimentos intervalares, presente na biblioteca *Int-Haar*, faz o cálculo do intervalo de erro do processo apresentando o maior comprimento de intervalo contido nos resultados da transformação. Os resultados apresentados na Figura 3 mostram os intervalos de erro para as quatro abordagens listadas anteriormente. O teste executado aplica as quatro abordagens em ambas as etapas de decomposição e composição, utilizando-se uma imagem de 1024 x 1024 pixels em escala-de-cinza 8bits como entrada para a TWH 2D.

Decomposição	-
Não-Normalizada Padrão	0.0
Não-Normalizada Não-Padrão	0.0
Normalizada Padrão	1.193712E-012
Normalizada Não-Padrão	1.108447E-012
Composição	-
Não-Normalizada Padrão	0.0
Não-Normalizada Não-Padrão	0.0
Normalizada Padrão	3.259629E-009
Normalizada Não-Padrão	1.047738E-009



Figura 3: Resultados (esquerda) e imagem utilizada no teste (direita).

De acordo com o teste executado, os erros de computação foram nulos nas execuções utilizando a abordagem não-normalizada. Durante a decomposição normalizada o algoritmo padrão apresenta um intervalo de erro de computação próximo ao algoritmo não-padrão, porém, ao final da composição da imagem, percebe-se a diferença na ordem de E-009 entre o erro acumulado pelo algoritmo padrão em relação ao não-padrão.

## 5. Considerações Finais

Os resultados intervalares obtidos até o momento incentivam a continuidade do trabalho, com futuras implementações intervalares usando a biblioteca C-XSC e o estudo de outras Transformadas Wavelet, tal como a Transformada Wavelet Ortonormal de Daubechies. Considera-se ainda a realização dos testes de validação da biblioteca Int-Haar e demais implementações da Int-DWTs, a fim de se fornecer suporte ao cálculo de erro computacional à SC. Na continuidade, busca-se a extensão da biblioteca para suporte a programação paralela e/ou distribuída, integrada tanto à plataforma OpenMP, ou ainda, avaliando o uso de paralelização massiva via placas de vídeo, como a arquitetura CUDA.

## Referências

- [1] Foster, D. (2011) “Classificação de Padrões através de Wavelets e Métodos Bayesianos” Dissertação de Mestrado, UFSM, Santa Maria, Brasil.
- [2] HöLBIG, C. (2005) “Ambiente de Alto Desempenho com Alta Exatidão para a Resolução de Problemas”. Tese (Doutorado em CC) Inf/UFRGS, POA, Brasil.
- [3] MOORE, R. (1966) “Interval Analysis”. Philadelphia, USA: pub-PH.
- [4] Stollnitz, E; DeRose, T; Salesin, D. (1995) “WAVELETS FOR COMPUTER GRAPHICS: A PRIMER, part 1” University of Washington, E.U.A.
- [5] Hari Om; Mantosh Biswas (2012) “An Improved Image Denoising Method Based on Wavelet Thresholding” CSE Department, Dhanbad, India.
- [6] Manoj Kumar; Dr. Kasabegoudar. (2012) “Wavelet Based Texture Analysis And Classification With Linear Regration Model” Collage of Engineering, Ambejogai.