

ESTUDO SOBRE DESEMPENHO DA BIBLIOTECA INT-HAAR

VINÍCIUS DOS SANTOS¹; RENATA REISER²; MAURÍCIO PILLA²; ALICE KOZAKEVICIUS³

² Universidade Federal de Pelotas – {vrdsantos, reiser, pilla}@inf.ufpel.edu.br

³ Universidade Federal de Santa Maria – alicek@ufsm.br

1. INTRODUÇÃO

Este trabalho considera a Matemática Intervalar (IM) e propõe uma solução baseada na aritmética de (MOORE, 1979), visando melhor conhecimento e controle dos erros nos dados processos associados à computação científica.

A motivação para 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 neste contexto o processamento de sinais e de imagens, veja OM; BISWAS (2012) e KUMAR et al. (2012).

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 num estudo de caso para a implementação da biblioteca Int-DWTs. Este trabalho, insere-se 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 TWH incluindo sua versão ortonormal, reportada em STOLLNITZ (1995). A partir dos algoritmos originais, desenvolveu-se uma biblioteca contendo extensões intervalares, denominada Int-Haar, introduzida em SANTOS (2013).

Durante o estudo das formulações originais para a transformada em STOLLNITZ (1995), percebeu-se a possibilidade de otimizá-las, executando simplificações algébricas a fim de se eliminar o cálculo de valores irracionais, originalmente responsáveis pelas normalizações das transformações e considerados em cada iteração de seus algoritmos usuais. Através dessas simplificações, obteve-se um ganho de precisão na formulação intervalar obtida.

Este artigo está organizado da seguinte forma: na Seção 2, são discutidas as otimizações realizadas neste trabalho como também o processo de compressão intervalar desenvolvido. Os testes e resultados são apresentados na Seção 3. Finalmente, na Seção 4 as conclusões e os trabalhos futuros são apresentados.

2. OTIMIZAÇÕES

No estudo e análise dos algoritmos originais da TWH presentes em STOLLNITZ (1995), percebe-se que para os procedimentos normalizados, todas as iterações do programa executam divisões utilizando o valor $\sqrt{2}$ como divisor. O principal problema destes procedimentos é que, como esse valor não é computável, cada nível de decomposição ou composição adiciona um certo grau de erro de cálculo aos dados. O erro gerado a cada iteração então é propagado por todos os níveis de transformação até o fim do mesmo procedimento.

Nossa solução desenvolvida para este problema, baseia-se em realizar simplificações algébricas para eliminar o cálculo de valores não computáveis, quando possível, reduzindo o erro de cálculo envolvido no processo. Dessa maneira, os novos procedimentos produzem resultados mais confiáveis em comparação com os algoritmos da literatura, reportado em STOLLNITZ (1995).

De acordo com os novos procedimentos, o processo de decomposição normalizada, tanto unidimensional quanto bidimensional, é executado da seguinte maneira: primeiramente é feita a decomposição não-normalizada dos dados, como indica a Figura 1; após essa etapa é realizada a normalização de todos os coeficientes, multiplicando-os por $2^{-j/2}$, sendo j o nível no qual o coeficiente se encontra.

A transformação bidimensional 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.

Para se otimizar a exatidão da transformação bidimensional foi necessário criar a regra R_N para o cálculo dos fatores de normalização, a qual é baseada nos padrões estudados. A regra R_N é descrita pela equação dada logo a seguir:

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

sempre que $0 \leq j', j'' \leq (\log_2 n) - 1$ e onde n indica a ordem da matriz.

| | 0 | 1 | 2 | ... |
|-----|-----------------------|-----------------------|-----------------------|-----|
| 0 | $j' = 0$ $j'' = 0$ | $j' = 0$ $j'' = 1$ | $j' = 0$ $j'' = 2$ | |
| 1 | $j' = 1$ $j'' = 0$ | $j' = 1$ $j'' = 1$ | $j' = 1$ $j'' = 2$ | |
| 2 | $j' = 2$ $j'' = 0$ | $j' = 2$ $j'' = 1$ | $j' = 2$ $j'' = 2$ | |
| ... | | | | |

(a)

| | 0 | 1 | 2 | ... |
|-----|-----------------------|-----------------------|-----------------------|-----|
| 0 | $j' = 0$ $j'' = 0$ | | | |
| 1 | | $j' = 1$ $j'' = 1$ | | |
| 2 | | | $j' = 2$ $j'' = 2$ | |
| ... | | | | |

(b)

Figura 1: Exemplo de combinações de fatores de normalização: (a) Algoritmo padrão; (b) Algoritmo não-padrão.

A principal meta do procedimento de compressão é expressar um conjunto inicial de dados utilizando a menor quantidade de pontos possível. Esta tarefa pode ser feita com ou sem perda de informação STOLLNITZ (1995).

O *Hard Thresholding* usa um valor real T como valor de corte (*threshold*), e a decisão de quais detalhes devem ser ignorados é um simples conjunto de comparações pontuais. Por isso foram desenvolvidos dois tipos de decisões para o contexto intervalar, chamados **Hard Decision** e **Soft Decision**.

O tipo de decisão *Hard Decision* é feito ao verificar-se se o dado intervalar é totalmente menor que T . Este procedimento garante que todo possível valor pontual é menor que todos os possíveis valores pontuais pertencentes a T .

A decisão do tipo *Soft Decision* é executada verificando-se que a maioria dos dados intervalares é menor que o ponto médio de T , comparando o limite esquerdo da informação e o centro do *threshold*. Este procedimento garante que a maior parte dos coeficientes pontuais estão abaixo do limite de corte.

| | | |
|---|---|--|
| <p>Data: $C[0...n]$: reais $i \leftarrow 0$; for $i < n$ do if $C[i] < \tau$ then $C[i] = 0$; end $i = i + 1$; end</p> | <p>Data: $C[0...n]$: intervalos $i \leftarrow 0$; for $i < n$ do if $\overline{C[i]} < \tau$ then $C[i] = [0 : 0]$; end $i = i + 1$; end</p> | <p>Data: $C[0...n]$: intervalos $mid\tau = (\overline{\tau} + \underline{\tau})/2$; $i \leftarrow 0$; for $i < n$ do $midC = (\overline{C[i]} + \underline{C[i]})/2$; if $midC < mid\tau$ then $C[i] = [0 : 0]$; end $i = i + 1$; end</p> |
|---|---|--|

Figura 2: Algoritmos de compressão, *Hard Thresholding* à esquerda, *Hard Decision* ao meio, e *Soft Decision* à direita.

3. RESULTADOS E DISCUSSÃO

| Métodologias | Decomposição | Composição |
|----------------------------|---------------|---------------|
| Método Padrão Novo | 3.492460E-010 | 6.705523E-008 |
| Método Padrão Original | 7.916242E-008 | 4.671514E-006 |
| Método Não-Padrão Novo | 0.000000 | 0.000000 |
| Método Não-Padrão Original | 7.171184E-008 | 1.819804E-006 |

Tabela 1: Erros de cálculo gerados pelos algoritmos da literatura e os desenvolvidos no trabalho, utilizando a abordagem normalizada.

| Métodologias | Decomposição | Composição |
|----------------------------|--------------|------------|
| Método Padrão Novo | 39,5337ms | 33,1806ms |
| Método Padrão Original | 53,4047ms | 67,9271ms |
| Método Não-Padrão Novo | 36,6334ms | 47,9163ms |
| Método Não-Padrão Original | 48,3040ms | 71,0566ms |

Tabela 2: Médias de tempos de execução geradas pelos algoritmos da literatura e os desenvolvidos no trabalho.

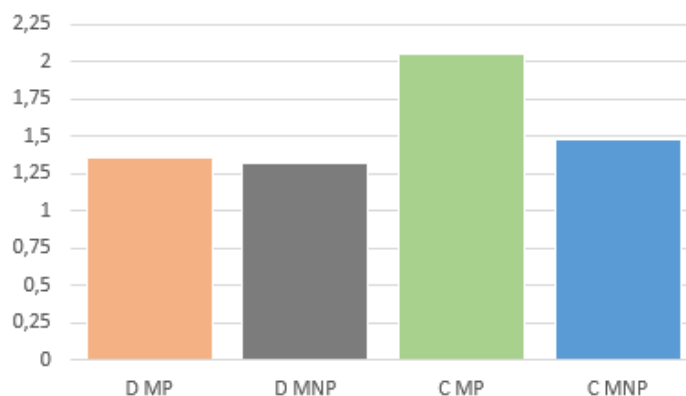


Figura 3: *Speedup* dos novos algoritmos para o método padrão (MP) e não padrão (MNP) durante a decomposição (D) e composição (C).

É importante verificar que, como é mostrado na Figura 1, os cálculos de $\sqrt{2}$ para o novo método não-padrão não precisaram ser executados, devido à simplificação algébrica implementada. Obtém-se ao final um erro de cálculo nulo, como mostrado na Tabela 1.

O tempo de execução também foi otimizado, Tabela 2. Os algoritmos originais buscam normalizar os dados preparando-os para o cálculo do próximo nível de transformação. Os novos algoritmos por sua vez executam a normalização dos dados uma única vez, ao final da decomposição ou no início da composição. Desta maneira houve um ganho de tempo considerável.

Os resultados apresentados na Tabela 2 foram alcançados ao se executar 15 vezes os procedimentos de cada método. O *speedup* dos algoritmos desenvolvidos em relação aos originais da literatura são apresentados na Figura 3. O maior desvio padrão das execuções é de aproximadamente 0,7731666.

4. CONCLUSÕES

Este trabalho apresentou a implementação da versão intervalar do algoritmo da TWH obtendo bons resultados quanto à análise do erro. As otimizações viabilizaram o aumento na exatidão dos cálculos, além de diminuir o tempo de execução.

Outras otimizações serão estudadas para se melhorar a indexação dos dados, auxiliando no acesso dos mesmos e assim diminuindo o tempo de execução da transformação. A complexidade espacial será otimizada consequentemente.

Na continuidade, busca-se a extensão da biblioteca para suporte a programação paralela, utilizando OpenMP ou o uso de paralelização massiva em GPGPUs (*General-Processing Graphics Processing Units*).

5. REFERÊNCIAS BIBLIOGRÁFICAS

DOS SANTOS, V.; PILLA, M.; REISER, R.; KOZAKEVICIUS, A. Int-Haar: Extensão intervalar da transformada de Haar. In: Anais da Escola Regional de Alto Desempenho, SBC, 2013, p. 167–170, POA.

OM, H.; BISWAS, M. An improved image denoising method based on wavelet thresholding. **Journal of Signal and Inf. Processing**, v.3, p.109–116, 2012.

KUMAR, M.; SUDHANSU, S.; KASABEGOUDAR, V. Wavelet based texture analysis and classification with linear regression model. **International Journal of Engineering Research and Applications**, v.2, p.1963–1970, 2012.

MOORE, R. *Methods and Applications of Interval Analysis*. SIAM, 1979 Philadelphia.

STOLLNITZ, E.; DeROSE, T.; SaALESIN, D. Wavelets for computer graphics: A primer, Part 1. *IEEE Computer Graphics and Applications*, 1995, 15(3):76–84.