

RELATÓRIO TÉCNICO PARCIAL

Título do Projeto: Núcleo de Excelência em Computação de Alto Desempenho e sua Aplicação em Computação Científica e Inteligência Computacional

Nome do Coordenador do Projeto: Alberto Ferreira De Souza

Nº do Processo: 48511579/2009

Instituição (por extenso): Universidade Federal do Espírito Santo

O projeto vem sendo apoiado pela FAPES desde Novembro / 2010
(mês) (ano)

Período em que se refere o Relatório: 01/11/2011 a 30/10/2012

Resumo das Atividades Executadas no Período de vigência do auxílio.

I - Principais objetivos do projeto original:

- 1 - Criar e consolidar o “Núcleo de Excelência em Computação de Alto Desempenho e sua Aplicação em Computação Científica e Inteligência Computacional” (NECAD) da UFES.
- 2 - Desenvolver novas bibliotecas numéricas de apoio ao desenvolvimento de código para a solução de problemas multi-física, simulação multi-escala, e solucionadores rápidos.
- 3 - Estender o estado da arte em Cognição Visual Artificial por meio da implementação de um veículo autônomo baseado em um automóvel comercial e do seu uso em pesquisas científicas que subsidiem a integração dos sistemas de vergência e reconstrução 3D, reconhecimento de imagens e busca visual desenvolvidos na UFES em sistemas para o mapeamento e localização simultâneos de veículos autônomos.

II - Etapas executadas e metas atingidas no período visando ao alcance dos objetivos:

O Projeto "Núcleo de Excelência em Computação de Alto Desempenho e sua Aplicação em Computação Científica e Inteligência Computacional" foi elaborado considerando os prazos previstos no Edital CNPq/FAPES Nº 012/2009. No

entanto, a FAPES precisou adiar em pouco mais de 3 meses a assinatura dos termos de outorga de todos os projetos do Edital - o Termo de Outorga referente a este Projeto foi assinado em 19/11/2010. Assim, o Cronograma do Projeto, apresentado em seu item K) PLANO DE METAS, foi deslocado para frente de 3 meses e é hoje conforme abaixo.

NOVO PLANO DE METAS DO PROJETO

ANO	Meta	Etapa/ Fase	Especificação	Indicador Físico		Período de Execução	
				Unidade Medida	Qtde	Início	Término
2010	1	Definições Preliminares					
		1	Definição de estratégias para viabilizar computacionalmente métodos de estabilização multi-escala	Relatório	1	Nov/10	Jul/11
		2	Definição de novos preconditionadores baseados nas arestas dos elementos finitos	Relatório	1	Nov/10	Jul/11
		3	Definição de estratégias para implementação do método JFNK	Relatório	1	Nov/10	Jul/11
		4	Definição de estratégias para redução da matriz Jacobiana representada por coloração de grafos	Relatório	1	Nov/10	Jul/11
		5	Definição de estratégias para redução de banda de matrizes esparsas oriundas de formulações de elementos finitos	Relatório	1	Nov/10	Jul/11
		6	Especificação da Plataforma Robótica (PR)	Relatório	1	Nov/10	Jul/11
	2	Desenvolvimento - Fase 1					
		1	Desenvolvimento e implementação de métodos de estabilização multi-escala para problemas práticos diversos	Código	1	Nov/10	Fev/12
		2	Desenvolvimento e implementação de preconditionadores baseados nas arestas dos elementos finitos	Código	1	Nov/10	Fev/12
		3	Desenvolvimento e implementação do método JFNK para formulações de elementos finitos	Código	1	Nov/10	Fev/12
		4	Desenvolvimento e implementação de estratégias para redução da matriz Jacobiana representada por coloração de grafos	Código	1	Nov/10	Fev/12
		5	Desenvolvimento e implementação de estratégias para redução de banda de matrizes esparsas oriundas de formulações de elementos finitos	Código	1	Nov/10	Fev/12
6		Desenvolvimento de uma versão do sistema de vergência e reconstrução 3D usando computação de alto desempenho	Código	1	Nov/10	Fev/12	
7	Desenvolvimento de uma versão do sistema de busca visual usando computação de alto desempenho	Código	1	Nov/10	Fev/12		

		8	Desenvolvimento de uma versão do sistema de reconhecimento de imagens usando computação de alto desempenho	código	1	Nov/10	Fev/12
	3	Acompanhamento e Documentação - Fase 1					
		1	Realização de seminários de acompanhamento e avaliação	seminário	1	Nov/10	Fev/12
		2	Elaboração patentes e/ou artigos científicos associados aos resultados alcançados com o projeto	documento	1	Nov/10	Jul/12
	4	Formação de Recursos Humanos - Fase 1					
		1	Agregação de um bolsista recém-doutor à equipe do projeto	relatório	1	Nov/10	Jul/12
		2	Agregação de dois alunos de doutorado à equipe do projeto	relatório	2	Nov/10	Jul/12
		3	Agregação de dois alunos de mestrado à equipe do projeto	relatório	2	Nov/10	Jul/12
		4	Agregação de três alunos de iniciação científica à equipe do projeto	relatório	3	Nov/10	Jul/12
2011	1	Desenvolvimento - Fase 2					
		1	Aplicação dos métodos de estabilização multi-escala desenvolvidos na Fase 1 em problemas na indústria do petróleo	código	1	Dez/11	Dez/12
		2	Desenvolvimento e implementação de condicionadores baseados nas arestas dos elementos finitos na biblioteca paralela EdgePack	código	1	Dez/11	Dez/12
		3	Desenvolvimento e implementação do método JFNK para formulações de elementos finitos na biblioteca paralela EdgePack	código	1	Dez/11	Dez/12
		4	Paralelização das estratégias desenvolvidas para redução da matriz Jacobiana representada por coloração de grafos	código	1	Dez/11	Dez/12
		5	Paralelização das estratégias desenvolvidas para redução de banda de matrizes esparsas oriundas de formulações de elementos finitos	código	1	Dez/11	Dez/12
		6	Desenvolvimento de uma versão do sistema de vergência e reconstrução 3D usando computação de alto desempenho	código	1	Dez/11	Dez/12
		7	Desenvolvimento de uma versão do sistema de busca visual usando computação de alto desempenho	código	1	Dez/11	Dez/12
		8	Desenvolvimento de uma versão do sistema de reconhecimento de imagens usando computação de alto desempenho	código	1	Dez/11	Dez/12
		9	Desenvolvimento de um sistema de exploração visual usando computação de alto desempenho	código	1	Dez/11	Dez/12

	10	Implementação da PR e integração do sistema de vergência e reconstrução 3D à mesma	versão da PR	1	Dez/11	Dez/12		
	Acompanhamento e Documentação - Fase 2							
	2	1	Realização de seminários de acompanhamento e avaliação	seminário	1	Dez/11	Out/12	
		2	Elaboração patentes e/ou artigos científicos associados aos resultados alcançados com o projeto	documento	2	Dez/11	Out/12	
	Formação de Recursos Humanos - Fase 2							
	3	1	Agregação de um bolsista recém-doutor à equipe do projeto	relatório	1	Nov/11	Jul/12	
		2	Agregação de um aluno de doutorado à equipe do projeto	relatório	1	Nov/11	Jul/12	
		3	Agregação de dois alunos de mestrado à equipe do projeto	relatório	2	Nov/11	Jul/12	
		4	Agregação de três alunos de iniciação científica à equipe do projeto	relatório	3	Nov/11	Jul/12	
2012	Desenvolvimento - Fase 3							
	1	1	Integração de todas as estratégias de implementação do método dos elementos finitos utilizando processamento de alto desempenho desenvolvidas nas fases anteriores	código	1	Nov/12	Out/13	
		2	Desenvolvimento de um sistema de localização e mapeamento simultâneos (SLAM) para a PR integrando o sistema de vergência e reconstrução 3D a outros sensores da PR	código	1	Nov/12	Out/13	
		3	Implementação da PR e integração do sistema de vergência e reconstrução 3D à mesma	versão da PR	1	Nov/12	Out/13	
	Desenvolvimento - Fase 4							
	2	1	Integração dos sistemas de exploração visual, busca visual e de reconhecimento de imagens ao sistema SLAM	código	1	Dez/12	Out/13	
	Acompanhamento e Documentação - Fase 3							
		3	1	Realização de seminários de acompanhamento e avaliação	seminário	1	Dez/12	Out/13
			2	Elaboração patentes e artigos científicos associados aos resultados alcançados com o projeto	documento	2	Dez/12	Out/13
	Formação de Recursos Humanos - Fase 3							
		4	2	Agregação de 1 aluno de doutorado à equipe do projeto	relatório	1	Dez/12	Out/13
			3	Agregação de dois alunos de mestrado à equipe do projeto	relatório	2	Dez/12	Out/13
			4	Agregação de dois alunos de iniciação científica à equipe do projeto	relatório	2	Dez/12	Out/13

Em Relatório Técnico Parcial anterior, apresentamos os resultados previstos para os primeiros ~9 meses de projeto (Meta 1, Etapas de 1 a 6), uma vez que devemos relatar as metas cumpridas apenas após concluído o prazo para o seu cumprimento e quando das prestações de contas. Este relatório se refere ao período de 01/11/2011 a 30/10/2012; assim, apresentamos nele as metas cujos períodos de execução terminaram antes de 30/10/2012 e que não foram apresentadas em Relatório Técnico anterior. Estas metas são:

1 - Meta Física 2.1/2010 - Desenvolvimento e implementação de métodos de estabilização multi-escala para problemas práticos diversos.

2 - Meta Física 2.2/2010 - Desenvolvimento e implementação de preconditionadores baseados nas arestas dos elementos finitos.

3 - Meta Física 2.3/2010 - Desenvolvimento e implementação do método JFNK para formulações de elementos finitos.

4 - Meta Física 2.4/2010 - Desenvolvimento e implementação de estratégias para redução da matriz Jacobiana representada por coloração de grafos.

5 - Meta Física 2.5/2010 - Desenvolvimento e implementação de estratégias para redução de banda de matrizes esparsas oriundas de formulações de elementos finitos.

6 - Meta Física 2.6/2010 - Desenvolvimento de uma versão do sistema de vergência e reconstrução 3D usando computação de alto desempenho.

7 - Meta Física 2.7/2010 - Desenvolvimento de uma versão do sistema de busca visual usando computação de alto desempenho.

8 - Meta Física 2.8/2010 - Desenvolvimento de uma versão do sistema de reconhecimento de imagens usando computação de alto desempenho.

9 - Meta Física 3.1/2010 - Realização de seminários de acompanhamento e avaliação.

10 - Meta Física 3.2/2010 - Elaboração de patentes e/ou artigos científicos associados aos resultados alcançados com o projeto.

11 - Meta Física 4.1/2010 - Agregação de um bolsista recém-doutor à equipe do projeto.

12 - Meta Física 4.2/2010 - Agregação de dois alunos de doutorado à equipe do projeto.

13 - Meta Física 4.3/2010 - Agregação de dois alunos de mestrado à equipe do projeto.

14 - Meta Física 4.4/2010 - Agregação de três alunos de iniciação científica à equipe do projeto.

15 - Meta Física 2.1/2011 - Realização de seminários de acompanhamento e avaliação.

16 - Meta Física 2.2/2011 - Elaboração de patentes e/ou artigos científicos associados aos resultados alcançados com o projeto.

17 - Meta Física 3.1/2011 - Agregação de um bolsista recém-doutor à equipe do projeto.

18 - Meta Física 3.2/2011 - Agregação de um aluno de doutorado à equipe do projeto.

19 - Meta Física 3.3/2011 - Agregação de dois alunos de mestrado à equipe do projeto.

20 - Meta Física 3.4/2011 - Agregação de três alunos de iniciação científica à equipe do projeto.

III - Apresentação e discussão sucinta dos principais resultados obtidos, deixando claro, os avanços teóricos, experimentais ou práticos obtido pela pesquisa.

1 - Meta Física 2.1/2010 - Desenvolvimento e implementação de métodos de estabilização multi-escala para problemas práticos diversos

O método de estabilização submalha Difusão Dinâmica (DD) foi introduzido por Arruda et al., em [ARR10] para a equação de transporte advectivo-difusivo-reactivo estacionário. Esta metodologia, baseada no método *Nonlinear Subgrid Stabilization* (NSGS) [SAN07], consiste em uma decomposição multiescala dos espaços de aproximação e do campo de velocidades em escalas resolvidas e não-resolvidas. Um operador dissipativo não-linear agindo isotropicamente em todas as escalas da discretização é adicionado à formulação de Galerkin. A quantidade de difusão artificial é determinada pela solução da equação na escala resolvida no nível de elemento, conduzindo a um método auto-adaptativo e livre de parâmetros de estabilização.

Nesta meta estendemos o método de estabilização submalha Difusão Dinâmica para problemas transientes de advecção-difusão-reação e para problemas compressíveis regidos pelas equações de Euler.

As equações diferenciais parciais que modelam problemas de transporte difusivo-convectivo-reativo têm grande aplicabilidade na engenharia e ciências em geral como, por exemplo, na simulação dos efeitos da poluição em rios, mares e atmosfera, modelagem da evolução das reservas de petróleo e gás natural no subsolo, contaminação de lençóis freáticos, e extração de águas subterrâneas.

As equações de Euler compressíveis formam um sistema acoplado de equações diferenciais parciais não lineares de primeira ordem, proveniente das leis de conservação da massa, momentum e energia. Essas equações descrevem o escoamento de fluidos na natureza considerados não viscosos. Na engenharia de petróleo as equações de Euler podem ser utilizadas na simulação de explosões acidentais em plataformas offshore e na engenharia aeroespacial essas equações são utilizadas no projeto de aeronaves e afins.

Para problemas transientes (dependentes do tempo), a metodologia numérica resulta em um sistema não linear de equações diferenciais ordinárias fortemente acoplado, com variáveis associadas à macro e micro escalas. Considerando que as escalas submalhas sejam estáticas (isto é, não variam no tempo como as escalas resolvidas), o problema de advecção-difusão-reação transiente é simplificado, resultando em soluções estáveis. Detalhes sobre o problema de advecção-difusão-reação transiente podem ser consultados no relatório técnico TEC-REP-01/2012 disponível em (http://www.lcad.inf.ufes.br/wiki/index.php/Relatórios_Técnicos).

A Figura 1.1 mostra a solução exata e a solução aproximada encontrada para um problema regido pela equação de advecção-difusão-reação. A solução obtida pelo método DD representa a região de alto gradiente sem oscilações, porém com um comportamento relativamente difusivo, como pode ser melhor observado comparando suas curvas de nível (Figura 1.1(d)) com as da solução exata (Figura 1.1(b)).

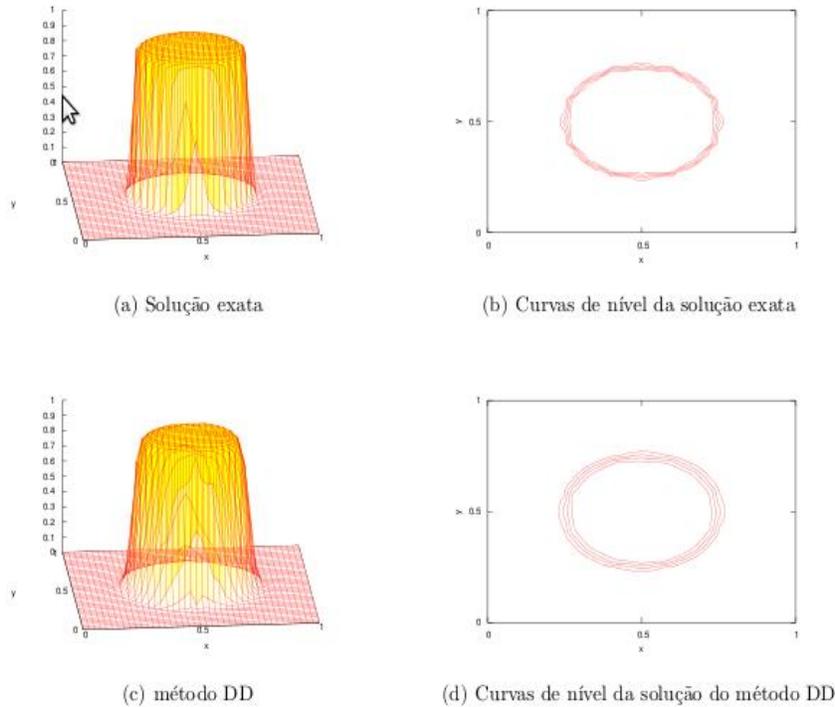


Figura 1.1 – Solução exata e solução aproximada de um problema regido pela equação de advecção-difusão-reação.

Porém, para as Equações de Euler, observamos que a hipótese de escalas submalhas estáticas não forneceu bons resultados, devido ao tipo da função bolha utilizada na micro escala. Dessa forma, desenvolvemos um mecanismo numérico para representar as variações temporais das escalas não resolvidas e os resultados foram promissores. Detalhes sobre o mecanismo numérico para representar as variações temporais das escalas não resolvidas podem ser consultados no relatório técnico TEC-REP-02/2012 disponível em (http://www.lcad.inf.ufes.br/wiki/index.php/Relatórios_Técnicos).

A Figura 1.2 mostra o esquema do conhecido problema do choque refletido regido pelas equações de Euler. Este problema consiste de três regiões de escoamento separadas por um choque oblíquo e sua reflexão ao longo de uma parede, conforme Figura 1.2.

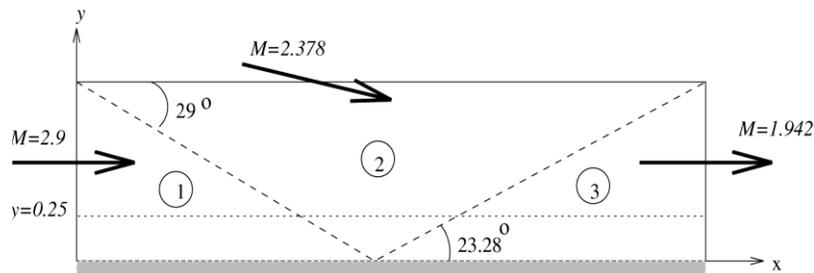


Figura 1.2 - Esquema do Problema - Choque Refletido Bidimensional.

A Figura 1.3 apresenta as isocurvas de densidade, onde observamos que a solução obtida pelo método Difusão Dinâmica é mais precisa do que àquela obtida pelo tradicional método estabilizado SUPG/CAU. A Figura 1.4 apresenta os perfis de densidade para as duas formulações avaliadas. Podemos observar que as soluções obtidas pelo método Difusão Dinâmica representam melhor os choques.

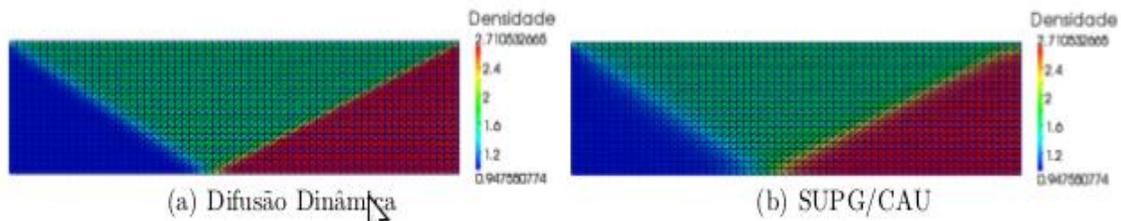


Figura 1.3 – Isocurvas de densidade – Choque refletido bidimensional

Os resultados alcançados até o momento demonstram que essa metodologia numérica representa bem regiões de alto gradiente sem oscilações para todos os problemas testados. Os principais resultados alcançados estão resumidos no artigo [MAT12].

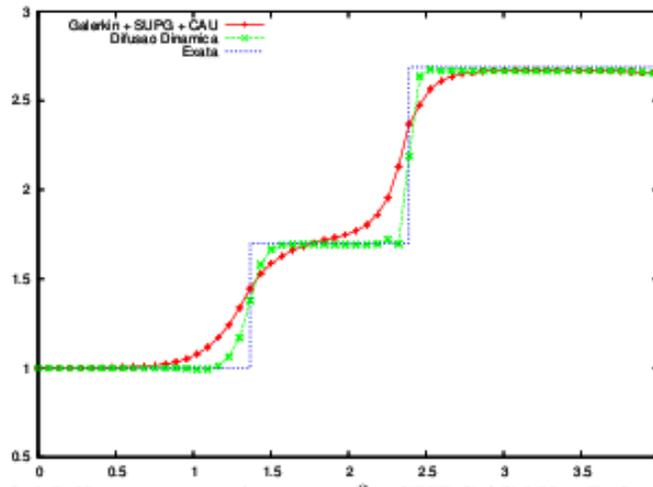


Figura 1.4 – Perfis de densidade – Choque refletido bidimensional

2 - Meta Física 2.2/2010 – Desenvolvimento e implementação de preconditionadores baseados nas arestas dos elementos finitos

O solver EdgeCFD é uma ferramenta para simulação em dinâmica dos fluidos computacional (CFD) considerando formulações estabilizadas do método dos elementos finitos e utilizando estrutura local de armazenamento de dados

baseada nas arestas dos elementos [CAT02a, ELI05, CAT09]. As estruturas de armazenamento locais, a citar, elemento-por-elemento e aresta-por-aresta, são constituídas de matrizes locais que possuem dimensão igual a quatro vezes (para problemas tridimensionais) o número de graus de liberdade por nó para a implementação baseada nos elementos e duas vezes para a implementação baseada nas arestas. Denominamos bloco diagonal as submatrizes que definem as contribuições de cada nó nele mesmo. Na estrutura de armazenamento baseada nas arestas adotada no EdgeCFD, os blocos diagonais de cada matriz de aresta são acumulados nos blocos diagonais nodais e as contribuições dos blocos não diagonais na estrutura aresta.

O procedimento descrito no parágrafo anterior possibilitou a definição do preconditionador bloco diagonal nodal já incorporado ao EdgeCFD. Porém, esse preconditionador não possui uma taxa de convergência elevada para uma gama de aplicações. O armazenamento aresta-por-aresta pode ser esquematicamente representado pela Figura 2.1, onde $Nedges$ é o número de aresta da malha de elementos finitos e Ned é o dobro do número de graus de liberdade por nó. A idéia central do processo de armazenamento é não gerar a matriz global K , mas armazenar o conjunto de matrizes locais k^s .

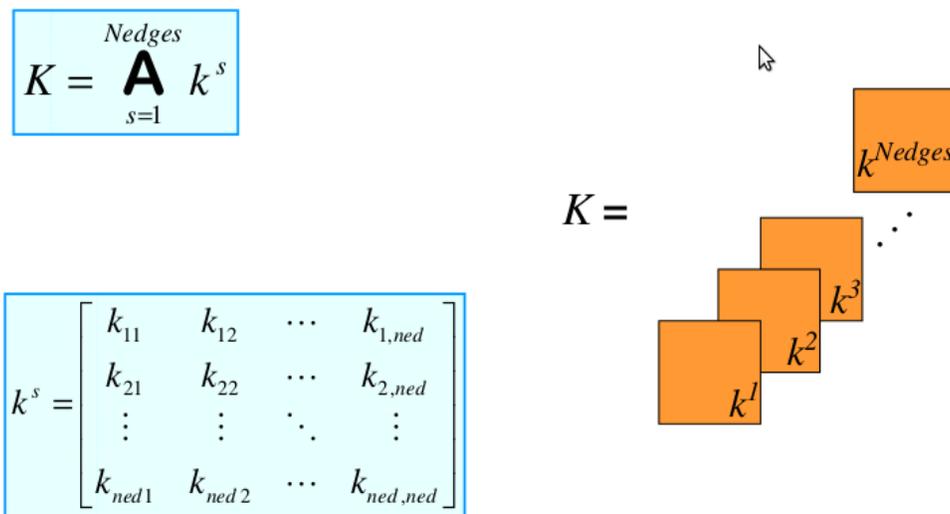


Figura 2.1 – Esquema do Armazenamento Aresta-por-Aresta.

O processo de preconditionamento prevê uma modificação no sistema linear $Ax = b$ visando melhorar a taxa de convergência do sistema linear. No momento estamos em fase de implementação do preconditionador aresta-por-aresta baseado na fatoração Gauss-Seidel local, que pode ser esquematicamente representado pela Figura 2.2.

$$(L^{-1}AU^{-1})(Ux) = (L^{-1}b)$$

$$L = \underset{s=1}{\overset{nedges}{\mathbf{A}}} \tilde{L}^s \quad U = \underset{s=1}{\overset{nedges}{\mathbf{A}}} \tilde{U}^s$$

$$\tilde{L}^s + \tilde{U}^s = A^s - \text{diag}(A^s) + 2I^s$$

$$(\tilde{L}^s + I^s) + (\tilde{U}^s + I^s) = A^s - \text{diag}(A^s)$$

Fatoração
Gauss-Seidel
Local

Figura 2.2 – Esquema do Precondicionador Aresta-por-Aresta baseado na Fatoração Gauss-Seidel.

3 - Meta Física 2.3/2010 - Desenvolvimento e implementação do método JFNK para formulações de elementos finitos

Como já observado no relatório anterior, optamos por investigar o comportamento do método de Newton inexato oriundo de formulações estabilizadas no método dos elementos finitos utilizando o solver EdgeCFD em contínuo desenvolvimento no NACAD/UFRJ. Neste contexto, o método de Newton considera uma aproximação da matriz jacobiana definida por Tezduyar, em [TEZ99], e utiliza a estrutura de armazenamento aresta por aresta. Podemos então dizer que o método a ser implementado é livre de matriz sob certo ponto de vista, uma vez que a matriz não é formalmente assemblada.

Dentro do contexto do Solver EdgeCFD implementamos (i) estratégias adaptativas do cálculo da tolerância do sistema linear a ser resolvido em cada passo não linear, denominada *termo forçante* e (ii) uma estratégia de globalização, denominada *Backtraking*, que visa calcular uma próxima aproximação da solução não-linear sem a necessidade de solucionar o sistema linear envolvido.

O termo forçante tem um papel fundamental no desempenho do método de Newton-Krylov. Um critério simples de seleção para o termo forçante pode levar o método a sucessivos estágios de *oversolving*. O principal objetivo consiste em descobrir qual nível de precisão é necessário para garantir uma rápida convergência do método de Newton sem perda de acuidade nos dados físicos de interesse. Selecionamos 3 métodos adaptativos para incluir no solver EdgeCFD: PP [PAP91], EW [EIS96] e GLT [GOM08]. Todos os métodos implementados adotam a seleção baseada na redução da norma euclidiana do resíduo não-linear. Detalhes sobre os métodos adaptativos podem ser consultados no relatório técnico TEC-REP-03/2012 disponível em (http://www.lcad.inf.ufes.br/wiki/index.php/Relatórios_Técnicos).

O método de globalização implementado, conhecido como *Backtracking*, tem por objetivo encontrar a melhor direção de busca da próxima aproximação não

linear utilizando a conhecida regra de Armijo, que propõe reduzir convenientemente a norma euclidiana do resíduo não linear. Detalhes sobre o método de globalização podem ser consultados no relatório técnico TEC-REP-03/2012 disponível em (http://www.lcad.inf.ufes.br/wiki/index.php/Relat3rios_T3cnicos).

Para testar o comportamento das estrat3gias otimizadoras implementadas consideramos um experimento t3pico de din3mica de fluidos, denominado o problema de ruptura de uma barreira (*dam break problem*). Inicialmente o fluido est3 confinado pr3ximo a face esquerda do dom3nio como mostra a Figura 3.1. O objeto de estudo 3 o movimento do fluido quando 3 considerada a ruptura da barreira.

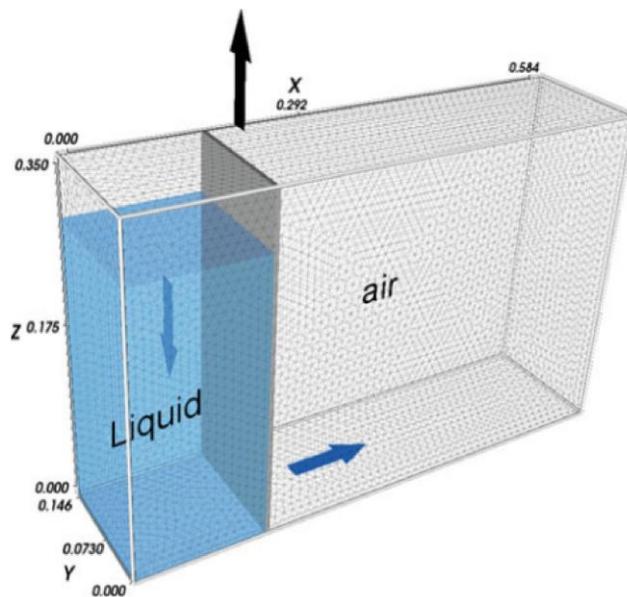


Figura 3.1 – Dom3nio do problema de ruptura de uma barreira.

O modelo matem3tico do escoamento presente nesse experimento consiste do sistema de equa33es de Navier-Stokes com acoplamento de uma equa33o de transporte que descreve a interface entre o l3quido e o ar. As equa33es de Navier-Stokes formam um sistema n3o linear incompress3vel de equa33es. No processo de solu33o pelo m3todo dos elementos finitos, em cada passo de tempo, 3 necess3rio resolver um sistema n3o linear de equa33es onde ser3o aplicadas as escolhas dos termos for3antes e o m3todo da globaliza33o. Detalhes sobre o processo de solu33o pelo m3todo dos elementos finitos podem ser consultados no relat3rio t3cnico TEC-REP-03/2012 dispon3vel em (http://www.lcad.inf.ufes.br/wiki/index.php/Relat3rios_T3cnicos).

A Figura 3.2 mostra as configura33es do ar e do l3quido em tr3s instantes de tempo. Na primeira coluna a configura33o representa os resultados quando foi considerado o termo for3ante EWK e na segunda o termo for3ante EWK com

backtracking. Podemos observar que os resultados são virtualmente iguais. Destacamos que o mesmo resultado foi observado para os termos forçantes PP e GLT.

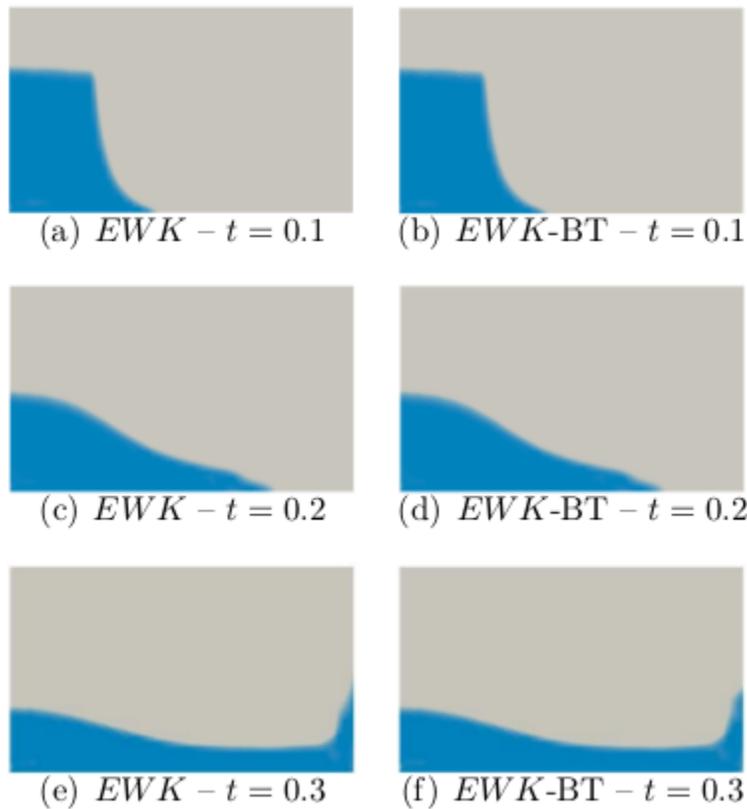


Figura 3.2 – Configurações do ar e do líquido.

A Tabela 3.1 mostra o tempo de CPU relativo para os termos forçantes EWK, GLT e PP para o método de Newton-Krylov Inexato (INK) e para o método de Newton-Krylov Inexato com Backtracking (INKB) tendo por base o procedimento padrão do método de Newton sem cálculo adaptativo de tolerâncias. Podemos observar que, em geral, os termos forçantes podem reduzir o tempo de CPU em 34% para o INK e 40% para o INKB.

Tabela 3.1 – Tempo Relativo de CPU para diferentes termos forçantes.

Forcing Term	INK	INKB
EWK	0.67	0.59
GLT	0.88	0.76
PP	0.66	0.61
Fixed 10^{-3}	1.00	

4 - Meta Física 2.4/2010 - Desenvolvimento e implementação de estratégias para redução da matriz Jacobiana representada por coloração de grafos

A matriz Jacobiana advém do cálculo das derivadas parciais de primeira ordem de uma função vetorial. Definida a Jacobiana, considera-se uma matriz de m linhas e n colunas, a primeira linha representa as derivadas parciais da primeira função em relação a n variáveis, e assim até a m -ésima linha. Esta meta do Projeto tem como um dos objetivos práticos aplicar os resultados no Método de Newton. Este método é iterativo para resolução de sistemas e, a cada iteração, avalia a Jacobiana em um ponto – i.e., é necessário acessar n vezes as colunas da matriz. Isto gera um custo computacional elevado, comprometendo a eficiência do algoritmo que necessita dessa avaliação. Para reduzir o tempo computacional dessa operação, a estratégia adotada no Projeto iniciou com a exploração da esparsidade da matriz e o conceito de ortogonalidade, ou seja, duas colunas são ditas estruturalmente ortogonais se ambas não possuem elementos não nulos na mesma linha. A partir disso, pode-se representar a matriz Jacobiana como um grafo. Existem dois tipos de grafos que representam a matriz: o grafo bipartido e o grafo de interseção de colunas [GEB05]. A definição da construção de tais grafos estão a seguir.

Seja A uma matriz $m \times n$ com linhas r_1, r_2, \dots, r_m e colunas a_1, a_2, \dots, a_n . O grafo bipartido $G_b(A)$ de A é definido como $G_b(A) = (V_1, V_2, E)$, onde $V_1 = \{ r_1, r_2, \dots, r_m \}$, $V_2 = \{ a_1, a_2, \dots, a_n \}$, e $(r_i, a_j) \in E$ sempre que $a_{ij} \neq 0$ para $1 \leq i \leq m$, $1 \leq j \leq n$. O número de vértices $|V_1| + |V_2| = m + n$ e o número de arestas $|E| = nnz(A)$, onde $nnz(A)$ é o número de $a_{ij} \neq 0$ em A .

O grafo de interseção de colunas de A é $G_c = (V, E)$, onde $V = \{ a_1, a_2, \dots, a_n \}$ e $(a_i, a_j) \in E \leftrightarrow a_i \cap a_j \neq \emptyset$. Em outras palavras, uma aresta (a_i, a_j) existe sempre que a i -ésima e a j -ésima colunas de A possuem entradas não-nulas em ao menos uma linha em comum.

A Figura 4.1 ilustra uma matriz 5×5 , os grafos de interseção de colunas e bipartidos correspondentes. Note que nesta figura são apresentadas suas colorações distintas, uma utiliza quatro cores e a outra, três cores.

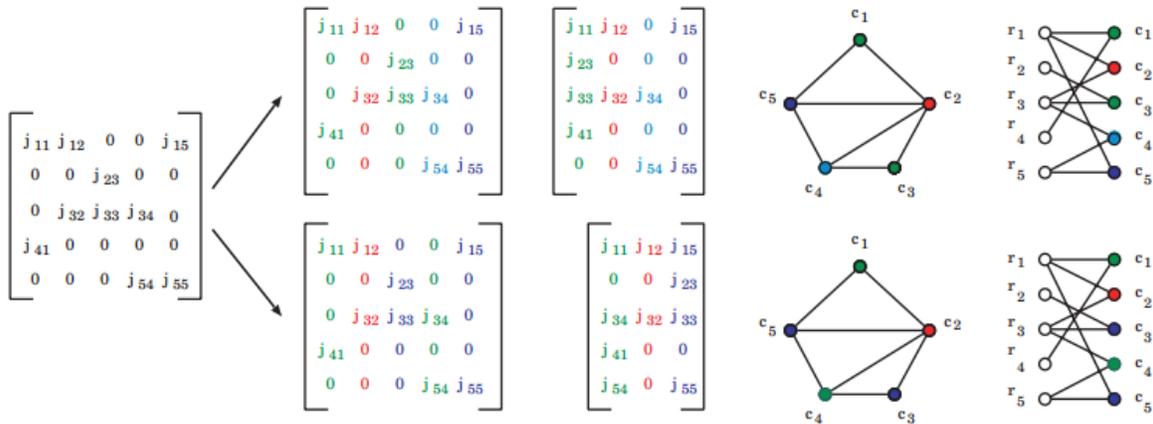


Figura 4.1: Representação da matriz 5x5 pelos grafos de interseção de colunas e bipartidos correspondentes e duas colorações possíveis. Fonte da ilustração [GEB05].

O segundo problema mencionado tem por objetivo atribuir cores distintas aos vértices do grafo que são adjacentes, utilizando o menor número de cores possível, como pode ser observado na Figura 4.1. A resolução da coloração do grafo que representa a matriz Jacobiana indica os vértices que pertencem a uma mesma partição, isto é, aqueles vértices que possuem a mesma cor. Desta forma, o número de colunas da Jacobiana é reduzido para o número de cores resultante da coloração.

Para colorir o grafo bipartido, é feita a coloração parcial de distância-2, sobre V_2 . Para melhor entendimento, a coloração distância-2 sobre os vértices de V_2 é uma atribuição de cores para estes vértices tais que a cada par de vértices da coluna, a uma distância de exatamente duas extremidades de cada um, recebe cores distintas. Mais precisamente, é um parcial do grafo bipartido uma vez que os vértice pertencentes a V_1 não recebe cores.

A coloração do grafo de interseção de colunas é feita utilizando a coloração de distância-1, isto é, vértices adjacentes recebem cores distintas.

Foram realizados testes para calcular o tempo gasto para obter a partição de colunas de algumas matrizes. Os dois tipos de representação e coloração de grafo foram considerados usando vértices de acordo com a leitura dos arquivos de dados e vértices com ordenação (decrescente ordem de graus). As matrizes testadas são quadradas e esparsas, escritas no formato *Matrix Market*.

O conjunto de matrizes estão listados na Tabela 4.1 com identificação, dimensão (n) e número de elementos não-nulos (nnz). Na quarta e quinta colunas temos o tempo de processamento total em segundos da construção dos grafos bipartido (G_b) e interseção de colunas (G_c) a partir dos arquivos das matrizes somado a coloração e os respectivos números de cores alcançadas.

Tabela 4.1: Conjunto de matrizes e resultados

Identificação	N	nnz	G _b (seg)	G _c (seg)	Nº cores G _b	Nº cores G _c
af23560.mtx	23560	484256	1.20	1.71	32	32
cage11.mtx	39082	559722	2.23	3.08	81	81
cage12.mtx	130228	2032536	9.55	13.48	96	96
e30r2000.mtx	9661	306356	0.89	1.12	65	65
e40r0100.mtx	17281	553562	1.46	1.87	95	95
lhr10.mtx	10672	232633	0.73	0.83	65	65
lhr14.mtx	14270	307858	0.94	1.09	65	65
lhr34.mtx	35152	764014	2.31	2.69	65	65
lhr71c.mtx	70304	1528092	4.58	5.28	65	65

Analisando os resultados dos testes computacionais apresentados na Tabela 4.1 podemos observar que o tempo de processamento total associado ao grafo de interseção de colunas é 31.4% maior comparado ao tempo computacional necessário utilizando o grafo bipartido. A razão disso está na construção dos grafos, pois o grafo de interseção de colunas é construído a partir do grafo bipartido. Os resultados obtidos com a coloração utilizando a ordenação dos vértices (ordem decrescente de graus) não foram apresentados na Tabela 4.1 porque o tempo computacional gasto para a ordenação foi muito alto. Com relação às colunas da matriz, a redução média obtida foi de aproximadamente 100% (comparar segunda coluna com a última coluna da Tabela 4.1).

Para simular matrizes Jacobianas construímos funções $F: R^n \rightarrow R^n$ não-lineares, a partir das matrizes disponíveis em [Dav97], cada linha da matriz dá origem a

$$f_i(x) = \prod x_j, \quad i, j = 1, \dots, n$$

uma função ressaltando que existe x_j se $a_{ij} \neq 0$.

Desta forma, as matrizes Jacobianas resultantes possuem a mesma estrutura das matrizes em [Dav97]. O método de avaliação escolhido foi Diferenças Finitas, o mesmo proposto em [GEB05]. Foi avaliada a matriz Jacobiana original em um ponto $x = (1, \dots, 1)^T$, isto é, n vezes são computadas as colunas $a_j = \frac{1}{\varepsilon} [F(x + \varepsilon e_j) - F(x)]$, onde e_j é o vetor canônico. Após a redução da matriz Jacobiana para r colunas são computadas r vezes as colunas $a_j \approx \frac{1}{\varepsilon} [F(x + \varepsilon d_j) - F(x)]$, onde d_j é o vetor que une as colunas das partições.

Na Figura 1, observando a coloração com três cores, a matriz Jacobiana foi reduzida com $d_1 = (1 \ 0 \ 0 \ 1 \ 0)^T$, $d_2 = (0 \ 1 \ 0 \ 0 \ 0)^T$ e $d_3 = (0 \ 0 \ 1 \ 0 \ 1)^T$. Os resultados computacionais, utilizando $\varepsilon = 10^{-9}$, mostram que, em média, a redução do tempo computacional para a avaliação da matriz Jacobiana em um ponto é de 77,74% do tempo de execução. É interessante observar que o desvio padrão é 0,65, indicando que esta técnica é robusta para várias dimensões das matrizes.

5 - Meta Física 2.5/2010 - Desenvolvimento e implementação de estratégias para redução de banda de matrizes esparsas oriundas de formulações de elementos finitos

O uso de técnicas de reordenamento de sistemas provocam um importante impacto na convergência dos métodos iterativos não estacionários, diminuindo o número de preenchimentos. No artigo [CAM12], resumimos um conjunto de experimentos paralelos de formulações estabilizadas de elementos finitos com adaptatividade, considerando reordenamentos clássicos como RCM (Reverse Cuthill-McKee) [CUT69, GEO71, LIU76] e QMD (Quotient Minimum Degree) [GEO89] aplicados a preconditionadores ILU, dentro do contexto das bibliotecas LibMesh [STO06] e PETSc [BAL10].

Os algoritmos clássicos de reordenamentos de sistemas encontrados na literatura podem ser reformulados para o problema de rerotulação de vértices de grafos não direcionados. Nós investigamos possibilidades de melhoria em vários algoritmos de reordenamento e para alguns deles já observamos o impacto de tais melhorias em preconditionadores ILU. Resultados parciais já encontrados estão sumarizados em [GHI11].

As métricas para avaliação da qualidade em uma dada ordenação são: (i) largura de banda – maior distância do primeiro elemento não nulo até a diagonal principal – e (ii) envelope – soma das distâncias do primeiro elemento não nulo até a diagonal principal.

O algoritmo de reordenamento RCM, tem como objetivo rotular os vértices de forma que nós adjacentes a cada vértice recebam rótulos próximos entre si. Recebendo como entrada um grafo, o método percorre-o a partir de um vértice inicial e caminha pelos vértices adjacentes em ordem crescente de graus. Como melhoria a este tradicional método de reordenamento de sistemas lineares, percebemos que o processo de caminhamento entre os vértices do grafo é equivalente a uma busca em largura nesse grafo e a ordem de visitação encontrada (invertida) corresponde a nova re-rotulação. O novo método, denominado RCM_P2 apresenta a mesma qualidade de solução com tempo de execução menor quando comparado com o RCM.

O algoritmo de reordenamento Espectral, proposto por [BAR95], consiste em encontrar o segundo menor autovalor da matriz laplaciana – conectividade algébrica – e seu autovetor associado. As ordenações em ordem crescente e em ordem decrescente do autovetor associado representam novas ordenações do grafo associado. O algoritmo Espectral clássico, escolhe como nova ordenação do grafo a ordem (crescente ou decrescente) que produz o menor envelope. Utilizamos para encontrar a conectividade algébrica do grafo associado a biblioteca Chaco [HEN95]. Com base nos experimentos realizados, observamos que não é necessário executar as ordenações crescente e

decrecente, pois os resultados alcançados com uma ou outra são muito parecidos.

O algoritmo Sloan [SLO86] atribui estados a cada nó. Um nó que ainda não recebeu nenhuma rotulação recebe o estado **pós-ativo**. Todo nó que é adjacente a um nó pós-ativo, mas não tem estado pós-ativo é definido com estado **ativo**. Os nós adjacentes a um nó ativo, mas não possuem um estado ativo ou pós-ativo, são definidos com estado **pré-ativo**. Já aqueles que não possuem estados ativo, pós-ativo ou pré-ativo, são definidos como **inativos**. A partir de dois vértices pseudo-periféricos – um nó inicial e um nó final – dados como entrada, cria-se uma fila de prioridades. Essa fila, formada apenas por nós ativos ou pré-ativos, vai sendo atualizada com informações do grafo e o nó com a maior prioridade é o próximo a ser escolhido para receber o novo rótulo. A prioridade de cada nó se relaciona com o estado desse nó e com sua distancia até o nó final. Por fim, após todos os nós serem escolhidos e re-rotulados, o processo é finalizado.

A seguir apresentamos um conjunto de experimentos com o objetivo de avaliar o desempenho dos algoritmos de reordenamento implementados para as matrizes esparsas hor__131, fidap027 e FEM_3D_thermal2 disponíveis no Repositório Matrix Market (<http://math.nist.gov/MatrixMarket/>).

Tabela 5.1 – Métricas para a Matriz hor__131

n	nnz	$a_{ij} = a_{ji}$	Envelope	Largura de Banda
434	4710	Não	51711	421

	Envelope	Largura de Banda	Tempo de Exec. (ms)
SLOAN	16790	279	5
RCM	23394	101	5
RCM-P2	23394	101	4
ESPECTRAL-C	14852	139	2+8
ESPECTRAL-D	14783	139	2+8

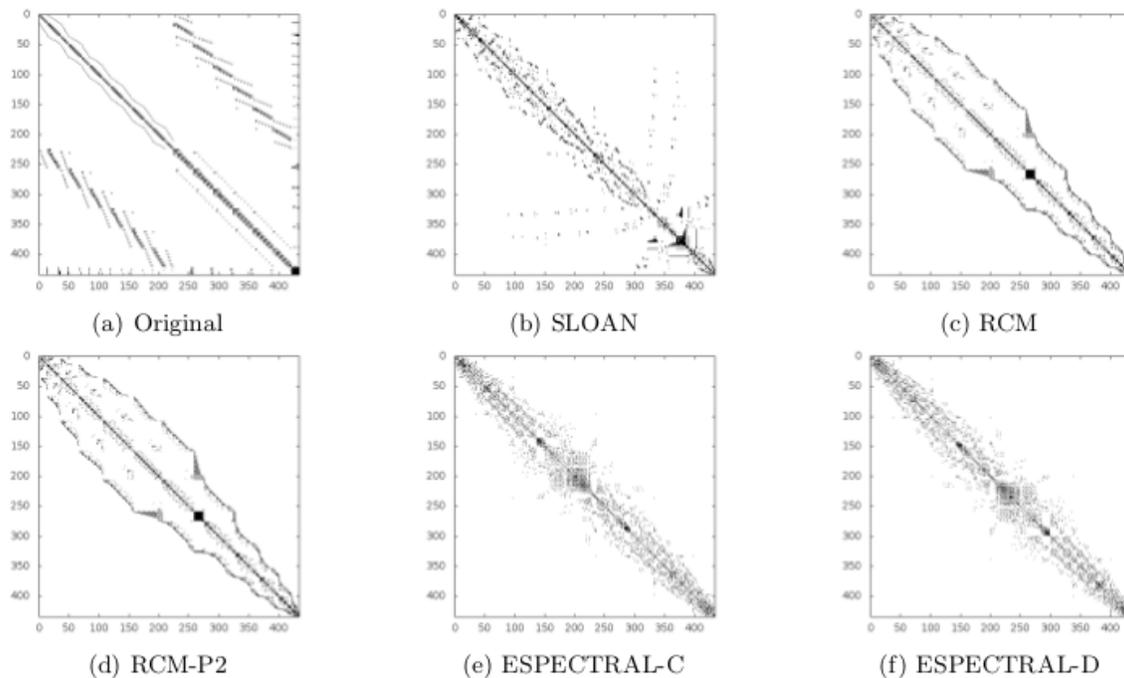


Figura 5.1 – Configurações de Esparsidade para a matriz hor__131.

A Tabela 5.1 apresenta a ordem da matriz (n), o número de elementos não nulos (nnz), a largura de banda e o envelope da matriz hor__131 sem reordenamento e após os reordenamentos apresentados acima. Na Figura 5.1 podem ser observadas as configurações de esparsidade da matriz hor__131 antes e depois de cada reordenamento. Podemos observar que todos os algoritmos reduzem envelope e largura de banda. O algoritmo RCM_P2 apresenta a maior redução em menor tempo computacional.

Tabela 5.2 - Métricas para a Matriz fidap027

n	nnz	$a_{ij} = a_{ji}$	Envelope	Largura de Banda
974	40736	Não	69912	136

	Envelope	Largura de Banda	Tempo de Exec. (ms)
SLOAN	54198	625	39
RCM	74331	262	37
RCM-P2	74331	262	34
ESPECTRAL-C	73585	189	4+35
ESPECTRAL-D	72457	189	4+35

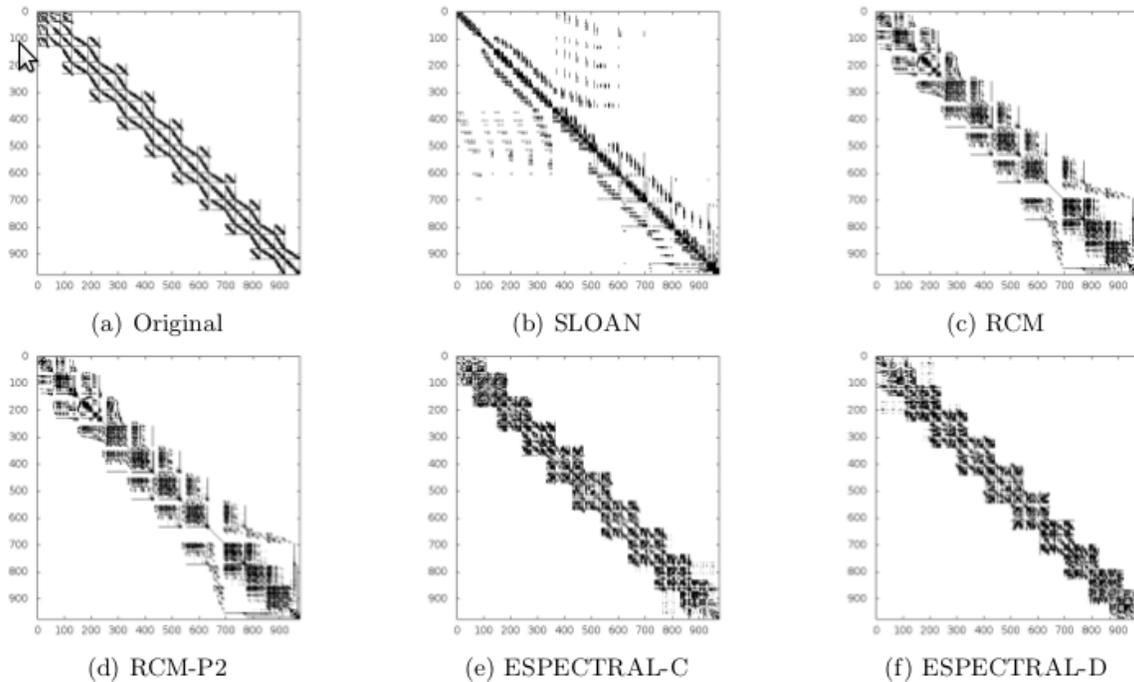


Figura 5.2 - Configurações de Esparsidade para a matriz fidap027.

Observamos na Tabela 5.2 que o algoritmo SLOAN produz a maior redução no envelope; porém, a maior redução da largura de banda se dá com o algoritmo ESPECTRAL. Contudo, observamos que, embora o RCM produza uma redução intermediária para o envelope e para a largura de banda, o tempo computacional foi menor para esse algoritmo. Na Figura 5.2 podemos observar as configurações de esparsidade obtidas.

Tabela 5.3 - Métricas para a Matriz FEM_3D_thermal2.

n	nnz	$a_{ij} = a_{ji}$	Envelope	Largura de Banda
147900	3489300	Não	241430619	117816

	Envelope	Largura de Banda	Tempo de Exec. (ms)
SLOAN	154909017	6876	29463
RCM	174631887	2819	29698
RCM-P2	174631887	2819	18197
ESPECTRAL-C	146495580	2188	2156+2892
ESPECTRAL-D	146556768	2188	2157+2892

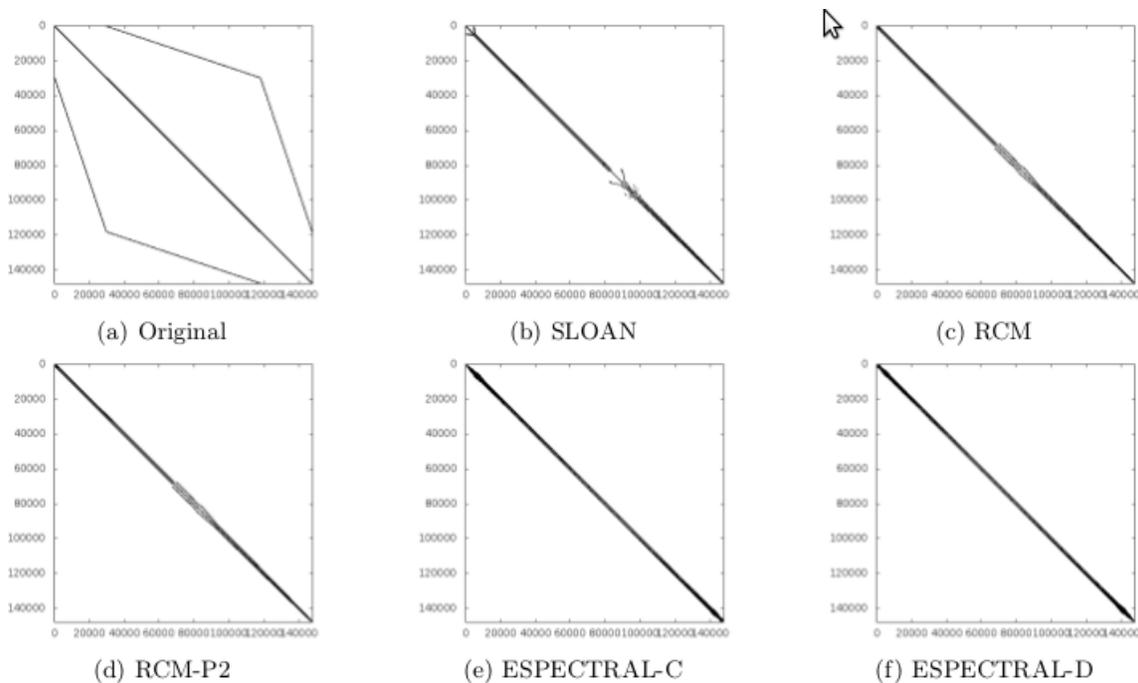


Figura 5.3 - Configurações de Esparsidade para a matriz FEM_3D_thermal2.

Para a matriz FEM_3D_thermal2 o algoritmo Espectral apresentou um comportamento bastante superior que os demais, tanto com relação a acuidade, (envelope e largura de banda) quanto com relação ao desempenho (tempo de processamento), como observado na Tabela 5.3 e na Figura 5.3.

6 - Meta Física 2.6/2010 - Desenvolvimento de uma versão do sistema de vergência e reconstrução 3D usando computação de alto desempenho

Nesta seção, apresentamos o sistema de vergência e reconstrução 3D propostos.

6.1 – Vergência

Nesta seção, apresentamos o sistema de vergência implementado com Redes Neurais Sem Peso. Iniciamos fornecemos uma introdução às Redes Neurais Sem Peso. Em seguida, descrevemos a arquitetura da rede desenvolvida para realizar a tarefa de vergência. Por fim, apresentamos os resultados obtidos com esta arquitetura em termos de tempo de vergência para as versões sequencial e paralela do sistema.

6.1.1 - Redes neurais sem peso

Diferentes de redes neurais com peso, Redes Neurais Sem Peso (RNSP) não armazenam conhecimento em suas conexões, mas em *Random Access Memories* (RAM) dentro dos nodos da rede, ou neurônios. Estes neurônios operam com valores de entrada binários e usam RAM como tabelas-verdade: as sinapses de cada neurônio coletam um vetor de bits da entrada da rede que é usado como o endereço RAM e o valor armazenado neste endereço é a saída do neurônio. O treinamento pode ser feito em um único passo e consiste basicamente em armazenar a saída desejada no endereço associado com o vetor de entrada do neurônio [ALE66].

Apesar de sua notável simplicidade, RNSP são muito efetivas como ferramentas de reconhecimento de padrões, oferecendo treinamento e teste rápidos e fácil implementação [ALE98]. No entanto, se a entrada da rede for muito grande, o tamanho da memória torna-se proibitivo, dado que deve ser igual a 2^n , onde n é o tamanho da entrada. As RNSP do tipo *Virtual Generalizing RAM* (VG-RAM) são redes neurais baseadas em RAM que somente requerem capacidade de memória para armazenar os dados relacionados ao conjunto de treinamento [LUD99]. Nos neurônios dessas redes, a memória armazena os pares entrada-saída observados durante o treinamento, ao invés da saída. Na fase de teste, a memória dos neurônios RNSP VG-RAM é pesquisada mediante a comparação entre a entrada apresentada à rede e todas as entradas dos pares entrada-saída aprendidos. A saída de cada neurônio RNSP VG-RAM é determinada pela saída do par cuja entrada é a mais próxima da entrada apresentada – a função de distância adotada pelos neurônios RNSP VG-RAM é a distância de *Hamming*. Se existir mais do que um par na mesma distância mínima da entrada apresentada, a saída do neurônio é escolhida aleatoriamente entre esses pares.

A Tabela 6.1.1 ilustra a tabela-verdade de um neurônio VG-RAM com três sinapses (X_1 , X_2 e X_3). Esta tabela-verdade contém três entradas (pares entrada-saída) que foram armazenadas durante a fase de treinamento (*entrada #1*, *entrada #2* e *entrada #3*). Durante a fase de teste, quando um vetor de entrada (*nova entrada*) é apresentado à rede, o algoritmo de teste da RNSP VG-RAM calcula a distância entre este vetor de entrada e cada entrada dos pares entrada-saída armazenados na tabela-verdade. No exemplo da Tabela 6.1.1, a distância de *Hamming* entre a *nova entrada* e a *entrada #1* é dois, porque ambos os bits X_2 e X_3 não são iguais aos bits X_2 e X_3 do vetor da *nova entrada*. A distância da *entrada #2* é um, porque X_1 é o único bit diferente. A distância da *entrada #3* é três, como o leitor pode facilmente verificar. Portanto, para este vetor de entrada, o algoritmo avalia a saída do neurônio, Y , como “saída 2”, pois é o valor de saída armazenado na *entrada #2*.

Tabela 6.1.1: Tabela-verdade de um neurônio da RNSP VG-RAM

Tabela verdade	X ₁	X ₂	X ₃	Y
entrada #1	1	1	0	saída 1
entrada #2	0	0	1	saída 2
entrada #3	0	1	0	saída 3
	↑	↑	↑	↓
nova entrada	1	0	1	saída 2

As RNSP VG-RAM já foram empregadas com sucesso por pesquisadores do Laboratório de Computação de Alto Desempenho (LCAD) do Programa de Pós-Graduação em Informática (PPGI) da UFES em diversos problemas relacionados à Cognição Visual Artificial e Aprendizado de Máquina, como controle de vergência em sistemas de visão artificial [KOM02], reconhecimento de faces [DES08b] e categorização automática de texto [DES08a, DES09, DES07, BAD08]. Esses exemplos de aplicação de RNSP VG-RAM têm em comum o modelo de memória distribuída (individual) dos seus neurônios. Contudo, a Busca Visual, outro exemplo de aplicação de RNSP VG-RAM que veremos em detalhes a seguir, emprega o modelo de memória compartilhada (ou coletiva) dos neurônios.

No caso de memória compartilhada, durante a fase de teste, cada neurônio deve calcular a distância de *Hamming* entre a nova entrada e as entradas armazenadas por todos os neurônios da rede. Neste caso, a saída é também a associada à entrada da tabela-verdade que possuir a menor distância de *Hamming* da nova entrada, mas podendo ter sido memorizada por qualquer neurônio da rede.

Vale ressaltar que o modelo de memória compartilhada implica em maior gasto de tempo para se computar a saída da rede neural. No modelo de memória distribuída tradicional, este tempo é diretamente proporcional ao número de entradas armazenadas durante a fase de treinamento, enquanto que no modelo de memória compartilhada, este passa a ser proporcional ao número de entradas de treinamento vezes o número de neurônios da rede.

6.1.2 - Arquitetura neural do sistema de vergência

Nesta seção, apresentamos a arquitetura do sistema de vergência implementado com Redes Neurais Sem Peso. Nesta direção, inicialmente fornecemos uma visão geral da arquitetura, detalhando as camadas que a compõem, como estas operam e como são conectadas entre si. Em seguida, explicamos o funcionamento do padrão de interconexão sináptico tipo log-polar e descrevemos a fase de treinamento da rede neural. Por fim, apresentamos os

métodos empregados no tratamento da saída da camada neural e controle da vergência.

6.1.2.1 - Visão geral da arquitetura neural

No sistema de vergência implementado, empregamos uma arquitetura de RNSP VG-RAM com uma camada bidimensional de neurônios, N , com $m \times n$ neurônios. A Figura 6.1.2.1.1 mostra a arquitetura neural, que tem como componente principal os neurônios N e na sua base a imagem de entrada, I , de dimensões $\xi \times \eta$. Antes de ser levada à entrada da rede neural, Φ (de dimensões $\xi \times \eta$), a imagem I sofre uma suavização gaussiana para reduzir o efeito de eventuais ruídos presentes na mesma.

Sinapses de RNSP coletam apenas um bit (0 ou 1) da entrada. Portanto, utilizamos *minchinton cells* [MIT98] para permitir entradas com valores não binários, como, por exemplo, imagens.

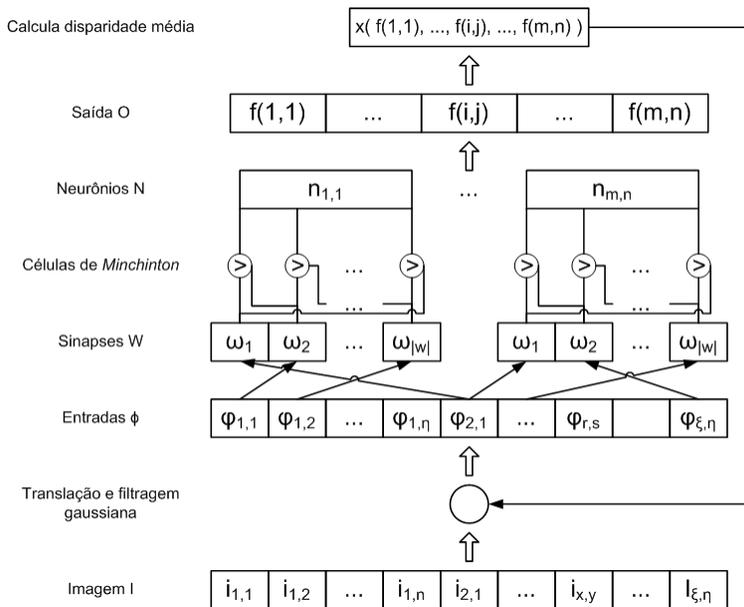


Figura 6.1.2.1.1: Arquitetura neural usada no sistema de Vergência.

6.1.2.2 - Padrão de interconexão sináptico log-polar

Cada neurônio, $n_{i,j}$, possui um conjunto de sinapses, $W = \{w_1, \dots, w_{|W|}\}$. Estas sinapses são conectadas à entrada bidimensional da rede, Φ , composta de $\xi \times \eta$ elementos de entrada, $\phi_{r,s}$, segundo um padrão de interconexão $\Omega_{i,j}(W)$. Este padrão de interconexão sináptica segue uma distribuição aleatória bidimensional com PDF (*Probabilistic Density Function*) Normal centrada no pixel ϕ_{μ_i, μ_j} , onde as coordenadas μ_i e μ_j dos elementos de Φ são determinadas pela transformação log-polar inversa das coordenadas i, j da camada neural N , como mostrado na

Figura 6.1.2.2.1, segundo a Equação 6.1.2.2.1, Equação 6.1.2.2.2, Equação 6.1.2.2.3 e Equação 6.1.2.2.4.



Figura 6.1.2.2.1: Mapeamento da posição central dos neurônios no plano da imagem de entrada. Figura retirada de [NET12]

$$d = \xi \cdot \left(2^{\left(\frac{\lfloor i-m/2 \rfloor - 1}{m/2} \right)} - \frac{1}{2} \right) \quad \text{Equação 6.1.2.2.1}$$

$$\theta = \begin{cases} \pi \cdot \left(\frac{3n}{2} - \frac{j}{n} \right) + \frac{\pi}{2n} & ; \text{se } i < \frac{m}{2} \\ \pi \cdot \left(\frac{3n}{2} + \frac{j}{n} \right) + \frac{\pi}{2n} & ; \text{se } i > \frac{m}{2} \end{cases} \quad \text{Equação 6.1.2.2.2}$$

$$\mu_i = \frac{\xi}{2} + d \cdot \cos(\theta) \quad \text{Equação 6.1.2.2.3}$$

$$\mu_j = \frac{\eta}{2} + d \cdot \sin(\theta) \quad \text{Equação 6.1.2.2.4}$$

onde m denota o número de linhas da camada neural N e n denota o número de colunas da camada neural N .

As coordenadas r e s dos elementos de Φ aos quais $n_{i,j}$ se conecta via W seguem as PDFs da Equação 6.1.2.2.5 e Equação 6.1.2.2.6, onde o desvio padrão σ da distribuição gaussiana é um parâmetro da arquitetura neural.

$$\varphi_{\mu_i, \sigma^2}(r) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(k-\mu_i)^2}{2\sigma^2}} \quad \text{Equação 6.1.2.2.5}$$

$$\varphi_{\mu_j, \sigma^2}(s) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(l-\mu_j)^2}{2\sigma^2}} \quad \text{Equação 6.1.2.2.6}$$

Este padrão de interconexão sináptica é comum em neurônios biológicos [KAN00]. Entretanto, uma diferença do padrão de interconexão sináptica adotado no sistema de vergência em relação ao empregado em outras aplicações com RNSP VG-RAM é que, embora aleatório, é o mesmo para todos os neurônios, além destes possuírem memória individual, diferente da busca visual (apresentada mais adiante no texto).

6.1.2.3 - Treinamento do sistema de vergência

Considerando a arquitetura neural descrita anteriormente, quando a rede é treinada, é como se fossem criadas camadas de neurônios treinadas com um valor de disparidade específico para cada posição monitorada na imagem de entrada, conforme ilustrado na Figura 3.1. O treinamento da RNSP para vergência é realizado para um determinado número máximo de disparidades parametrizado n , que determina quão próximo se deseja procurar pelo ponto de interesse de vergência, lembrando que a disparidade é inversamente proporcional à distância. Sendo assim, o treinamento consiste em definir o ponto de interesse de vergência na imagem direita como o centro da log-polar para em seguida deslocar a imagem direita iterativamente de $[-n;+n]$ disparidades gravando na camada neural a cada passo a disparidade em questão ($-n$).

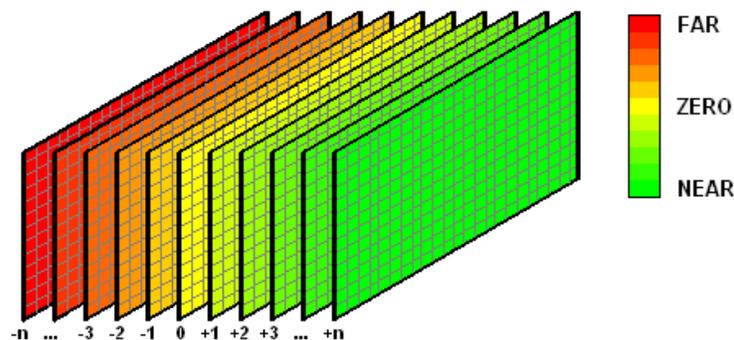


Figura 6.1.2.3.1: Treinamento do sistema de vergência. Figura retirada de [OLI05].

6.1.2.4 - Tratamento da saída da camada neural e controle de vergência

A Figura 6.1.2.4.1 mostra exemplos de situações de uso de nossa arquitetura de RNSP para vergência.

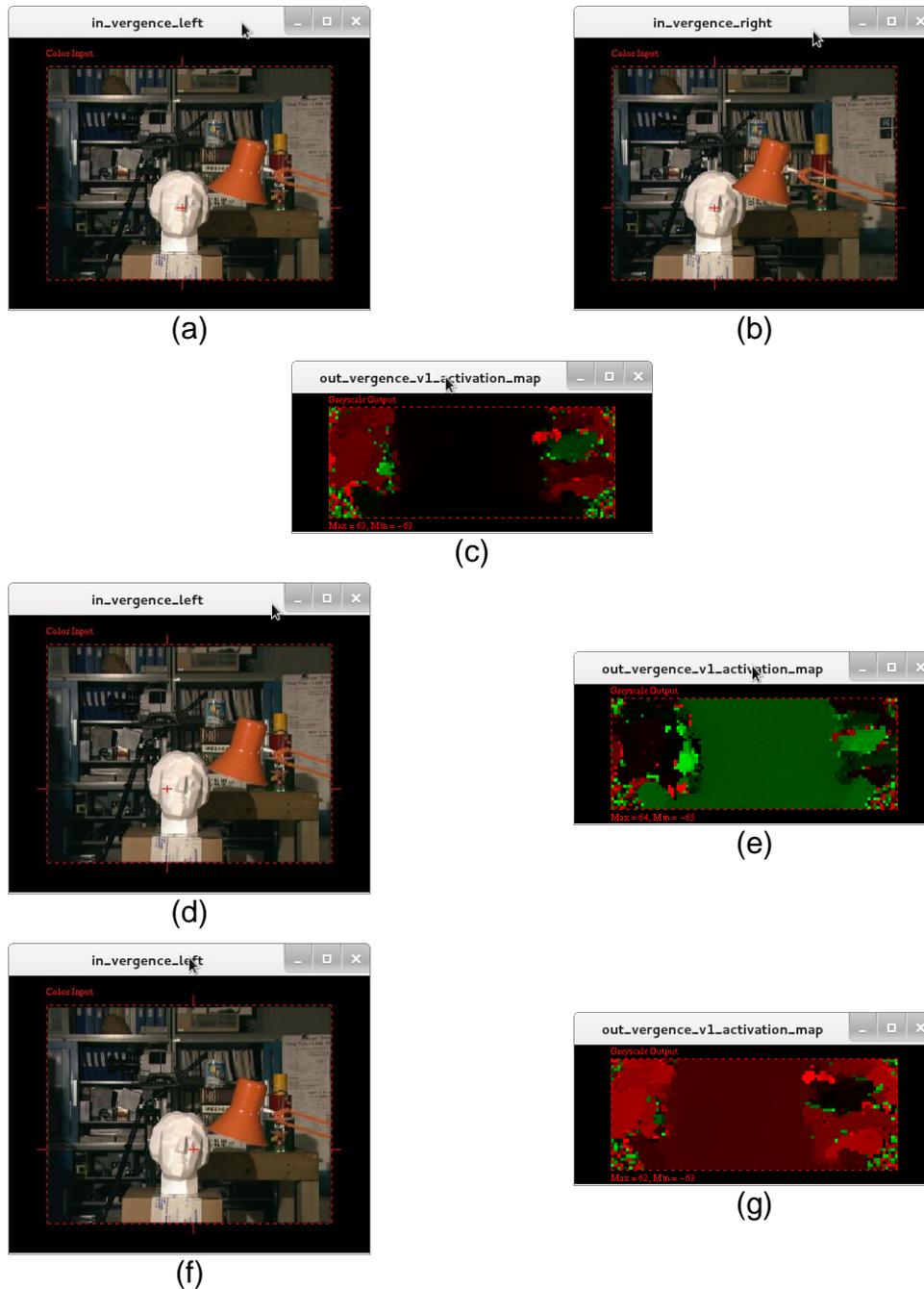


Figura 6.1.2.4.1: Funcionamento do sistema de vergência. (a) Imagem direita usada para treino. (b) Imagem esquerda após vergência final. (c) Mapa de disparidade final. (d) Imagem esquerda com ponto de atenção aquém do ponto de vergência. (e) Mapa de disparidade com ponto de atenção aquém do ponto de vergência. (f) Imagem esquerda com ponto de atenção além do ponto de vergência. (g) Mapa de disparidade com ponto de atenção além do ponto de vergência.

A Figura 6.1.2.4.1 (a), (b), (d) e (f) mostra a imagem Tsukuba (<http://vision.middlebury.edu/stereo/>) com uma cruz vermelha representando o

ponto de atenção central do mapeamento log-polar da RNSP. Figura 6.1.2.4.1 (b) demonstra uma situação de treinamento, em que a rede é treinada para vergir o ponto de atenção para o nariz da escultura visto na imagem direita. A Figura 6.1.2.4.1 (a) demonstra o ponto de atenção posicionado exatamente sobre o nariz da escultura na imagem esquerda após a vergência final. Na Figura 6.1.2.4.1 (c) apresentamos o mapa de ativação log-polar (ou mapa de disparidade codificado em cores que representam disparidades positivas em verde, negativas em vermelho e zero em preto) obtido como resposta dos neurônios após a vergência. O significado da resposta dos neurônios é a disparidade que cada um diz estar do ponto de atenção. Sendo assim, para fazer a vergência, simplesmente calculamos a média de todas as respostas (disparidades) dos neurônios. O centro da log-polar seria deslocado segundo um valor positivo ou negativo de disparidade d que tenderia a disparidade média da saída dos neurônios para zero, como mostrado na Equação 6.1.2.4.1.

$$d = \arg \min_{i,j} \frac{\sum_{i,j} f(i,j)}{m \times n} \quad \text{Equação 6.1.2.4.1}$$

Em seguida, é realizado o deslocamento que foi calculado e a saída dos neurônios é atualizada novamente. Quando esta movimentação do centro da log-polar (movimento de vergência [KAN00]), levar ao mesmo ponto, a disparidade média será próxima de zero e o ponto de vergência estará bem próximo do desejado. Caso contrário, um novo ciclo de movimento de vergência e uma nova avaliação da saída da rede é efetuada até um limite máximo, parametrizado, de vezes consecutivas.

Na Figura 6.1.2.4.1 (d) e (e) é possível visualizar o ponto de atenção à esquerda do ponto de vergência desejado (nariz da estátua) e, conseqüentemente, a região central da camada de saída dos neurônios indicando em verde uma disparidade positiva a ser tomada. Já a Figura 6.1.2.4.1 (f) e (g) apresenta um caso ilustrativo em que o ponto de atenção está à direita do ponto de vergência desejado (nariz da estátua) e, então, a região central da camada de saída dos neurônios indicando que há uma disparidade negativa (codificada em vermelho) a ser realizada até o ponto de vergência. Observe novamente na Figura 6.1.2.4.1 (c) que a região central da camada de saída dos neurônios é zero (codificada em preto) quando o ponto de atenção está sobre o ponto de vergência.

6.1.3 – Tempos de execução do sistema de vergência

Inicialmente, foi feita a implementação sequencial da arquitetura proposta utilizando o framework MAE - Máquina Associadora de Eventos (http://www.lcad.inf.ufes.br/wiki/index.php/Máquina_Associadora_de_Eventos_-_MAE). Este framework possui código fonte aberto e foi desenvolvido pelo

Grupo de Pesquisa em Ciência da Cognição do LCAD. Com o MAE, pode-se projetar tanto estruturas modulares usando RNSP com neurônios do tipo VG-RAM, quanto estruturas com arquitetura em camadas, com a definição de um processamento específico para cada camada, a exemplo de [OLI05].

Uma segunda implementação paralela foi realizada visando diminuir o tempo de processamento, utilizando computação de alto desempenho através da API de programação paralela OpenMP (<http://openmp.org/wp/>), que foi empregada no framework MAE. Por fim, implementaremos uma versão utilizando CUDA.

A Tabela 6.1.3.1 mostra os tempos de execução das versões sequencial e paralela do sistema de vergência em uma máquina Blade Dell R5500 com 2 processadores Intel Xeon 2.13 (Quad-core) com GPU CUDA enabled com 448 processadores com os seguintes parâmetros da arquitetura: (i) imagens de 320x240 pixels; (ii) tamanho da camada neural de 96x36; (iii) número de sinapses por neurônio igual a 256; (iv) número máximo de sacadas igual a 20 e; (v) número de disparidades treinadas igual a 64.

Tabela 6.1.3.1: Tempos de execução do sistema de vergência.

<i>Versão</i>	<i>Tempo (s)</i>
<i>Sequencial</i>	12,879
<i>Paralela</i>	10,623

6.2 – Reconstrução 3D

Nesta seção, apresentamos as pesquisas desenvolvidas na área de reconstrução 3D baseada em imagens estéreo. Nesta direção, inicialmente fornecemos uma introdução à reconstrução 3D estéreo. Em seguida, apresentamos as implementações dos algoritmos utilizados na solução do problema. Por fim, apresentamos os resultados alcançados.

6.2.1 – Introdução à reconstrução 3D

As imagens projetadas dentro de nossos olhos mudam todo o tempo por conta do movimento dos olhos ou do nosso corpo como um todo. Contudo, em um aparente paradoxo, percebemos o mundo retratado nas imagens capturadas pelos olhos como estável. Além disso, as imagens projetadas nas retinas humanas são bidimensionais; entretanto, o cérebro é capaz de sintetizar uma representação tridimensional estável a partir delas (o que vemos, Figura 6.2.1.1), com informações sobre cor, forma e profundidade a respeito dos objetos no ambiente ao nosso redor, eliminando os efeitos dos movimentos dos olhos e do corpo.

O sistema visual biológico viabiliza a nossa movimentação através do ambiente 3D de forma precisa. Assim, a compreensão e a modelagem de funcionalidades relevantes do sistema visual biológico, como aquelas que nos permitem ver em três dimensões, podem contribuir para o desenvolvimento de sistemas de localização e mapeamento simultâneos (*Simultaneous Localization And Mapping* – SLAM [THR05]) e de navegação de veículos autônomos.

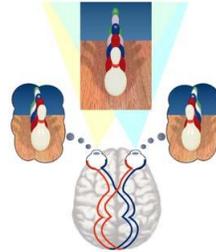


Figura 6.2.1.1: Criação, pelo cérebro, de representação 3D a partir de imagens 2D.

SLAM é talvez o problema mais fundamental da robótica autônoma. Veículos robóticos autônomos necessitam saber onde estão em sua área de atuação e como esta (sua área de atuação) está configurada para que possam navegar nela e realizar suas atividades de interesse. Para isso, estes veículos precisam usar sensores que capturem informações do ambiente em quantidade e qualidade suficiente para realizar o mapeamento relevante e se localizar nos mapas gerados. Atualmente, um dos sensores mais utilizados são lasers de varredura (*Laser Range Scan*, ou *Light Detection And Ranging* - LIDAR).

Sensores LIDAR empregam feixes de laser para medir a distância de pontos ao longo de uma linha à frente usando um mecanismo mecânico/ótico de varredura – tal mecanismo não encontra paralelo na biologia. Além disso, sensores LIDAR são fortemente afetados por chuva, entre outras condições meteorológicas naturais, o que limita sua aplicabilidade em casos que requeiram operação ao ar livre, e são facilmente detectáveis à distância (por causa do laser), o que limita sua aplicabilidade militar.

Câmeras digitais, por outro lado, são hoje capazes de capturar imagens com milhões de pontos, possuindo preço substancialmente menor que LIDARs. Contudo, para empregar câmeras digitais na solução do problema de SLAM em tempo real, é necessário processar a enorme quantidade de dados disponibilizada pelas mesmas na forma de imagens de maneira equivalente ao nosso cérebro; i.e., é necessário sintetizar representações tridimensionais estáveis a partir de imagens bidimensionais.

Para transformar as imagens bidimensionais em nuvens de pontos 3D, são necessários algoritmos que encontrem a projeção dos mesmos pontos de um objeto no mundo 3D em imagens capturadas por duas ou mais câmeras. Com a informação sobre a localização destes pontos nas diversas imagens, e com o

conhecimento sobre a geometria de cada câmera empregada e do posicionamento delas, é possível resolver o problema de percepção do mundo em 3D para pontos que sejam visualizados por duas ou mais câmeras. Técnicas que resolvem o problema de visão artificial 3D desta forma são conhecidas como técnicas de visão estéreo [YAN10, CAR09].

Para viabilizar a navegação robótica utilizando câmeras digitais, os algoritmos de visão estéreo devem executar em tempo real, isto é, em no mínimo 15 quadros por segundo (*Frames Per Second* - FPS). Neste trabalho de pesquisa, desenvolvemos duas versões paralelas do algoritmo *Constant Space Belief Propagation* (CSBP - um dos melhores algoritmos de visão estéreo conhecidos [MID11]) [YAN10]: uma em OpenMP e uma em C+CUDA.

6.2.2 – Implementações do CSBP Stereo

O algoritmo CSBP busca resolver o problema de visão estéreo por meio de *Stereo Matching* via *Belief Propagation* (BP) [FRE00, SUN03]. Este é um algoritmo baseado em troca de mensagens, muito utilizado para inferência em modelos gráficos como Redes Bayesianas e Campos Aleatórios de Markov (*Markov Random Field* – MRF).

Uma variação dos algoritmos baseados em BP é o *Hierarchical Belief Propagation* (HBP) descrito em [YAN10], este algoritmo difere de BP tradicional por trabalhar de forma hierárquica, de um nível de grão grosso para um nível de grão fino, ou seja, a cada iteração do algoritmo, a resolução espacial do modelo gráfico é reduzida até atingir um nível mínimo, onde o algoritmo começa a convergir.

Na abordagem de [YAN10] foi utilizado o algoritmo de HBP. Porém, da conclusão em [YU07], foi verificado que, na média, somente uma pequena parte dos níveis de disparidade e suas mensagens correspondentes são necessários em cada pixel da imagem para reconstruir as mensagens de BP sem muita perda. Como resultado, [YAN10] não somente aplica uma abordagem de refinamento no domínio espacial, mas também no domínio de profundidade, reduzindo gradualmente o número de níveis de disparidades, quando é feita a propagação das mensagens de um nível de grão mais grosso para um nível de grão mais fino. Tais modificações deram origem a um novo algoritmo de complexidade de espaço constante $O(1)$, denominado *Constant Space Belief Propagation* (CSBP) [YAN10], que foi alvo de estudo em nosso trabalho e é apresentado no quadro Algoritmo 1.

Algoritmo 1: Constant-Space Belief Propagation

1	Calcule o termo de dados E_D no nível mais baixo (nível $S - 1$)
2	Apenas os k_{S-1} custos de dados são selecionados e passados ao passo 5 em cada pixel. Os outros custos de dados e os níveis de disparidades correspondentes são tratados isoladamente.
3	Inicialize as mensagens com zero
4	Para $s = S - 1$ até 0 faça
5	Atualize iterativamente o vetor de mensagens em cada pixel e para cada disparidade de acordo com a Equação 1.
6	Se $s > 0$ então
7	Inicialize as mensagens para o próximo nível usando a Equação 3 e selecione os níveis de disparidade.
8	Calcule o termo de dados para o próximo nível, usando os níveis de disparidade selecionados.
9	Calcule a energia total de cada pixel para cada disparidade candidata de acordo com a Equação 2.
10	Selecione $k_{s-1} = k_s/2$ níveis de disparidade e valores de mensagem correspondente aos k_{s-1} menores valores de energia e aplique a estes o passo numero 5 para cada pixel. Os outros valores de disparidades e mensagens são tratados isoladamente e não serão levados em consideração. <i>Note que o tamanho do vetor de mensagem e o termo de dados de cada pixel será reduzido a metade nesse passo.</i>
11	Caso contrário
12	Calcule a energia total de cada pixel e cada disparidade candidata segundo a Equação 2
13	Fim Se
14	Fim Para
15	O valor de disparidade que minimiza a energia individual de cada pixel e escolhido

6.2.2.1 – Implementação paralela em OpenMP

A implementação paralela em OpenMP do algoritmo CSBP foi feita de forma incremental. Utilizou-se a ferramenta *g-prof (Gnu Profiler)* para a análise das funções mais custosas e, assim, foi possível dar prioridade à tais funções com maior tempo de processamento, porém essa abordagem de paralelização não foi suficiente para conseguir um *speed-up* significativo que garantisse uma maior vazão (*throughput*) de frames por segundo. Sendo assim, o código foi refeito como indicado no quadro Algoritmo 2.

No momento da criação das *threads* na linha 2 do Algoritmo 2 é iniciado o processo de paralelização do algoritmo. A cada tarefa principal do CSBP (representadas nas linhas 3, 5, 7, 8, 9 e 10 do Algoritmo 2) as *threads* são sincronizadas e distribuídas para execução paralela. Na tarefa da linha 3, as *threads* são responsáveis pela iteração nas linhas da imagem calculando o termo de dados E_D para cada pixel no nível mais baixo (nível $S - 1$) e seleciona os k_{S-1} custos de dados, que serão passados ao passo 5 em cada pixel. Para outras tarefas, o paralelismo segue o mesmo padrão, porém, em um nível de resolução espacial menor como detalhado no pseudocódigo do Algoritmo 2.

A cada novo par de imagens recebido da câmera é chamada uma função que implementa o Algoritmo 2 que computa um novo mapa de disparidades. Como resultado do paralelismo em OpenMP foi alcançado uma vazão de 13 FPS (frames por segundo) com uma imagem de 320x240 pixels. Contudo, uma câmera estéreo de bom desempenho captura imagens a taxas superiores a 30 FPS para resoluções como estas, e nossa versão paralela em OpenMP não foi suficiente para consumir toda entrada de dados da câmera. Por isso, uma nova abordagem de paralelização que fosse mais rápida foi implementada utilizando-se de GPUs CUDA.

Algoritmo 2: Constant-Space Belief Propagation OpenMP	
1	Inicialize as mensagens com zero
2	Criação das <i>threads</i>
3	Cada <i>thread</i> interam nas linhas da imagem calculando o termo de dados E_D para cada pixel no nível mais baixo (nível $S - 1$) e seleciona os k_{s-1} custos de dados, que serão passados ao passo 5 em cada pixel. Os outros custos de dados e os níveis de disparidades correspondentes são tratados isoladamente.
4	Para $s = S - 1$ ate 0 faça
5	Para cada pixel X de nível mais grosso, atualize de forma paralela iterando sobre o vetor de mensagens para cada grupo de linhas em cada pixel X e para cada disparidade de acordo com a Equação 1.
6	Se $s > 0$ então
7	Inicialize as mensagens para o próximo nível, paralelizando os pixels X_j das colunas da imagem, utilizando a Equação 3 e selecione os níveis de disparidade.
8	Cada <i>thread</i> calcula o termo de dados para cada pixel X da imagem para o próximo nível, usando os níveis de disparidade selecionados.
9	Cada <i>thread</i> calcula a energia total de cada pixel X_j das colunas da imagem para cada disparidade candidata de acordo com a Equação 2.
10	Cada <i>thread</i> seleciona $k_{s-1} = k_s/2$ níveis de disparidade e valores de mensagem correspondente aos k_{s-1} menores valores de energia. Posteriormente de forma seqüencial aplique o passo numero 5 para cada pixel do nível atual. Os outros valores de disparidades e mensagens são tratados isoladamente e não serão levados em consideração.
11	Caso contrário
12	Calcule a energia total de cada pixel e cada disparidade candidata segundo a Equação 2
13	Fim Se
14	Fim Para
15	O valor de disparidade que minimiza a energia individual de cada pixel e escolhido

6.2.2.2 – Implementação paralela em CUDA

Após a implementação paralela do algoritmo em OpenMP já se conhecia de antemão as tarefas que demandavam mais tempo de processamento e que, portanto, eram candidatas a se tornarem funções de *kernel* na GPU. Ao invés de construir um único *kernel* englobando todas as tarefas, tomou-se a decisão de projeto de dividir as tarefas em *kernels* separados para garantir a disponibilidade de dados na memória global da GPU entre tarefas.

Usando uma abordagem iterativa, optou-se por paralelizar cada tarefa individualmente mantendo a cópia de dados entre a memória principal e a GPU na chamada do *kernel* e então fazer a cópia de volta para não afetar o fluxo subsequente das tarefas ainda não paralelizadas na GPU.

Para minimizar a latência da transferência de dados entre CPU e GPU recomenda-se que todas as tarefas (mesmo as pequenas) sejam transformadas em *kernels*. Procedendo deste modo, o fluxo de processamento de dados na GPU se torna contínuo, já que não são realizadas transferências intermediárias entre CPU e GPU [NVI11]. Dessa forma, o algoritmo CSBP em CUDA recebe como entrada um par de imagens da câmera estéreo e as processa na GPU, devolvendo o mapa de disparidade correspondente.

Considerando a estrutura piramidal das imagens imposta pelo algoritmo HBP, a relação entre o número de disparidades e granularidade do pixel é variável em função do nível de disparidade L da pirâmide; i.e., à medida que se eleva o nível na pirâmide a granularidade do pixel aumenta e a faixa de disparidade diminui. Fazendo a correspondência do número de disparidades com o número de *threads* e do número de pixels com o número de blocos do *grid*, verifica-se que

há um *trade-off* a ser considerado na alocação de blocos e *threads* na GPU para cada nível da pirâmide de execução do algoritmo CSBP.

Nessa implementação CUDA, todos os blocos são bidimensionais para todos os *kernels* (exceto linhas 10 e 13) e foram assim dimensionados devido à necessidade de se percorrer a imagem ao longo da sua altura e largura. No caso das *threads* do *kernel* da linha 5, a escolha bidimensional deve-se ao relacionamento 4-conectado existente entre cada pixel (x,y) da imagem com seus vizinhos, fazendo com que seja necessário uma *thread* para cada pixel vizinho (x,y+1), (x,y-1), (x+1,y) e (x-1,y) para propagar as mensagens.

O algoritmo busca encontrar o mínimo de uma função de energia (disparidade), sendo necessário sincronizar esses valores entre as *threads*. Para tanto, foi utilizado uma operação atômica *atomicMin* na memória compartilhada de *threads* do mesmo bloco. A utilização dessa operação de barreira na memória compartilhada é possível e eficiente a partir da *Capability* 1.3 das placas CUDA NVIDIA.

Como resultado do paralelismo em CUDA foi alcançado uma taxa de 66 FPS, com imagens coloridas de 320x240 pixels. Apesar de uma câmera estéreo de bom desempenho capturar imagens a 48 FPS com resolução de 640x480 pixels, nossa versão paralela em CUDA tem capacidade de processamento superior a captura da câmera e, portanto, adequado ao propósito de aplicação de câmeras estéreo como um sensor tridimensional para navegação robótica em tempo real.

Algoritmo 3: Constant-Space Belief Propagation CUDA

1	Inicialize as mensagens com zero na memória da GPU (cudaMemSet)
2	Copia o par de imagens para memória da GPU (cudaMemcpy)
3.1	Uma função <i>kernel</i> com <i>grid</i> bidimensional e bloco unidimensional onde cada <i>thread</i> do bloco corresponde a um termo de dados E_D no nível mais baixo (nível $S - 1$)
3.2	Ainda nessa função <i>kernel</i> foram selecionados os k_{s-1} custos de dados e passados ao passo 5 em cada pixel. Os outros custos de dados e os níveis de disparidades correspondentes são tratados isoladamente.
4	Para $s = S - 1$ até 0 faça
5	Uma função <i>kernel</i> com <i>grid</i> bidimensional e blocos bidimensionais onde cada <i>thread</i> do bloco atualiza iterativamente o vetor de mensagens em cada pixel e para cada disparidade de acordo com a Equação 1.
6	Se $s > 0$ então
7	Uma função <i>kernel</i> com <i>grid</i> bidimensional e blocos bidimensionais onde cada <i>thread</i> do bloco corresponde a um termo de dados E_D para o próximo nível, usando os níveis de disparidade selecionados. O resultado é usado no passo 8.3.
8.1	Uma função <i>kernel</i> com <i>grid</i> bidimensional e bloco unidimensional onde somente a <i>thread</i> zero de cada bloco inicializa as mensagens para o próximo nível usando a Equação 3 e seleciona os níveis de disparidade.
8.2	Na mesma função de <i>kernel</i> todas as <i>threads</i> do bloco calculam a energia total de cada pixel para cada disparidade candidata de acordo com a Equação 2.
8.3	Ainda no mesmo <i>kernel</i> as <i>threads</i> zero de cada bloco selecionam $k_{s-1} = k_s/2$ níveis de disparidade e valores de mensagem correspondente aos k_{s-1} menores valores de energia e aplicam a estes o passo numero 5 para cada pixel. Os outros valores de disparidades e mensagens são tratados isoladamente e não serão levados em consideração.
9	Caso contrário
10	Uma função de <i>kernel</i> com <i>grid</i> e blocos unidimensionais em que as <i>threads</i> calculam a energia total de cada pixel e cada disparidade candidata segundo a Equação 2
11	Fim Se
12	Fim Para
13	Uma função de <i>kernel</i> com <i>grid</i> e blocos unidimensionais em que as <i>threads</i> selecionam o valor de disparidade que minimiza a energia individual de cada pixel
14	Copia o mapa de disparidade da memória da GPU para a memória principal do computador

6.2.3 - Resultados

A Tabela 6.2.3.1 sumariza o desempenho em termos de FPS obtido com as versões sequencial e paralelas em OpenMP e C+CUDA em função de diferentes resoluções de imagens. Na tabela, as linhas denotam a versão da implementação (sequencial, paralela em OpenMP, paralela em C+CUDA usando a GTX285 e paralela em C+CUDA usando a GTX480) e as colunas denotam a resolução da imagem (320x240, 640x480, 800x600, e 1024x768 pixels).

Tabela 6.2.3.1: Desempenho em termos de FPS da versão sequencial e das versões paralelas.

	320x240	640x480	800x600	1024x768
Sequencial	5,36	1,16	0,66	0,34
OpenMP (8 threads)	13,33	3,70	2,13	1,14
C+CUDA (GTX285)	40,18	10,52	6,48	3,73
C+CUDA (GTX480)	66,29	17,34	10,71	5,97

Como os resultados na Tabela 6.2.3.1 mostram, o desempenho absoluto da versão paralela em C+CUDA em termos de FPS foi significativamente maior que o das demais implementações paralelas e mais de 10 vezes superior que o da sequencial.

7 - Meta Física 2.7/2010 - Desenvolvimento de uma versão do sistema de busca visual usando computação de alto desempenho

Nesta seção, apresentamos a arquitetura do sistema de busca visual implementado com Redes Neurais Sem Peso. Para isso, inicialmente fornecemos uma visão geral da arquitetura, detalhando as camadas que a compõem, como estas operam e como são conectadas entre si. Em seguida, explicamos o funcionamento do padrão de interconexão sináptico tipo log-polar e descrevemos o mecanismo de operação do sistema de busca visual. Apresentamos então os métodos considerados para o tratamento da saída da camada neural, e, por fim, os resultados em termos de tempo das versões sequencial e paralela do sistema.

7.1 - Visão geral da arquitetura neural

No sistema de busca visual implementado, empregamos uma arquitetura de RNSP VG-RAM com uma camada bidimensional de neurônios, N , com $m \times n$ neurônios. A Figura 7.1.1 mostra nossa arquitetura neural, que tem como componente principal os neurônios N , e na sua base a imagem de entrada I de dimensões $\xi \times \eta$.

Antes de chegar à entrada da rede neural, Φ (de dimensões $2\xi \times 2\eta$), a imagem I sofre uma translação, um ajuste de escala e uma suavização gaussiana. A

translação é aplicada de modo que o ponto de interesse da busca visual, o ponto de fixação dos olhos, esteja posicionado no pixel central da camada de entrada Φ , onde se conectam as sinapses dos neurônios. O ajuste de escala e a suavização gaussiana são aplicados de modo que ruídos presentes na imagem na camada de entrada Φ da rede sejam minimizados. Dado que sinapses de RNSP coletam apenas um bit (0 ou 1) da entrada, para permitir seu uso com entradas que podem assumir valores não binários, usamos *minchinton cells*.

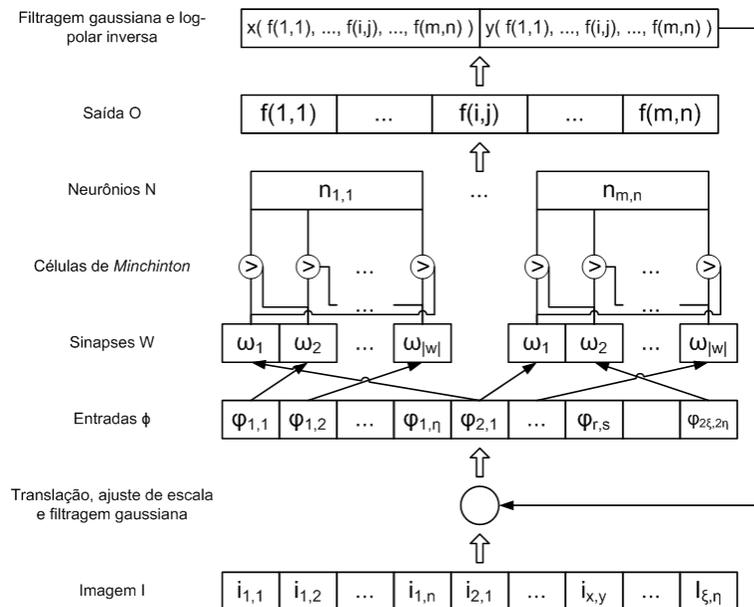


Figura 7.1.1: Arquitetura neural usada na aplicação de busca visual.

7.2 - Padrão de interconexão sináptico log-polar

Cada neurônio, $n_{i,j}$, possui um conjunto de sinapses, $W = \{w_1, \dots, w_{|W|}\}$. Estas sinapses são conectadas à entrada bidimensional da rede, Φ , composta de $2\xi \times 2\eta$ elementos de entrada, $\phi_{r,s}$, segundo um padrão de interconexão sináptico log-polar, descrito na Seção 6.2.2 da meta anterior.

Este padrão de interconexão sináptico é comum em neurônios biológicos [KAN00]. Entretanto, uma diferença do padrão de interconexão sináptico adotado na aplicação de busca visual em relação ao empregado em outras aplicações com RNSP VG-RAM é que, embora aleatório, é o mesmo para todos os neurônios, além destes possuírem memória comum, como mencionado anteriormente.

7.3 - Operação do sistema de busca visual

Dada a arquitetura neural descrita anteriormente, quando a rede é treinada, cada neurônio armazenará um padrão de bits condizente com a região da

imagem que suas sinapses monitoram e, associado a ele, uma saída cujo valor é dado pela distribuição normal truncada com média $m/2$ e desvio padrão parametrizável que denominamos padrão de ativação em forma de “barra”. No teste da rede o padrão de bits monitorado pelas sinapses de cada neurônio será comparado com os padrões de bits aprendidos por todos os neurônios – a isto denominamos memória coletiva dos neurônios. A Figura 7.3.1, abaixo, mostra exemplos de situações de treinamento e uso de nossa arquitetura de RNSP para busca visual.

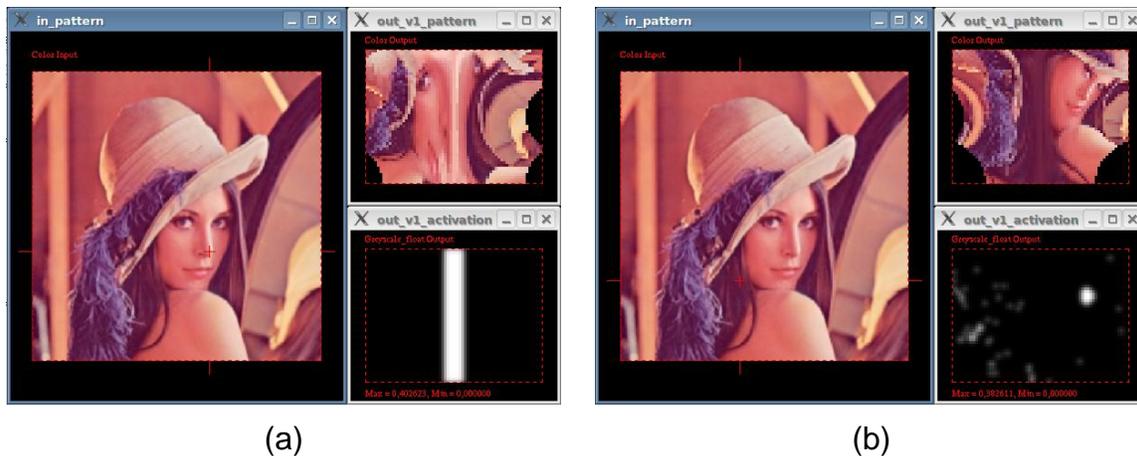


Figura 7.3.1: Treinamento e uso do sistema de busca visual. (a) Fase de Treinamento (b) Fase de Teste (Realização da Busca Visual). Figura retirada de [NET12].

A Figura 7.3.1 mostra uma situação de treinamento onde a rede é treinada para buscar o nariz da modelo na imagem. Na imagem da esquerda da Figura 7.3.1 (a), a cruz vermelha é o centro da log-polar; enquanto que, na imagem superior direita da Figura 7.3.1(a) apresentamos o mapeamento log-polar da imagem na camada de neurônios; por fim, na imagem inferior direita da Figura 7.3.1 (a), a saída da matriz de neurônios após o treinamento. Como esta última figura mostra, os neurônios com centro do campo receptivo (*receptive fields* – RFs) [KAN00] sobre ou próximos ao ponto de interesse são treinados para gerar saídas com valores altos (branco ou tons de cinza), enquanto que aqueles com RFs distantes do ponto de interesse são treinados para gerar saída zero (preto). O RF de um neurônio é dado pelos pixels monitorados pelo neurônio por meio de suas sinapses.

A Figura 7.3.1 (b) mostra uma situação de busca visual onde os neurônios da rede, treinados para buscar a imagem do nariz da modelo, geram suas saídas de acordo com a região da imagem monitorada por seus RFs. Como a imagem inferior direita da Figura 7.3.1 (b) mostra, os neurônios com RF sobre o nariz da modelo geram saída mais alta.

Na busca visual é feita após uma filtragem gaussiana da saída da camada neural e computado um vetor de deslocamento sacádico [LEE88, WAL05],

usando essa saída filtrada. Em seguida é realizado o deslocamento do centro da log-polar e a saída dos neurônios é computada novamente. Quando esta movimentação do centro da log-polar (movimento sacádico [KAN00]) levar ao mesmo ponto, ou seja, quando uma transformação sobre a saída da camada neural acusar deslocamento igual à zero, foi encontrado um ponto de interesse, caso contrário, um novo ciclo de movimento sacádico e avaliação da saída da rede é efetuado até um limite máximo, parametrizado, de vezes consecutivas.

O grau de certeza da rede quando da convergência neste ponto pode ser medido comparando-se a imagem final na saída da camada neural com aquela usada no treinamento (imagem inferior direita da Figura 7.3.1 (a)). No exemplo da Figura 7.3.1 (b), com apenas um passo de busca visual o ponto de interesse é encontrado com grau de certeza muito próximo ao máximo.

7.4 - Tratamento da saída da camada neural e movimento sacádico

Walton [WAL05] sugere, em seus trabalhos relacionados à controle visio-motor de movimentos sacádicos, que o movimento de sacada é determinado por uma soma vetorial ponderada, segundo o padrão de ativação dos neurônios do *superior culliculus*. Tal hipótese possui embasamento em experimentos com primatas envolvendo movimentos sacádicos sujeitos a múltiplos alvos e conseqüentemente a ativação de múltiplas localidades do *superior culliculus* [POR03]. Nossa arquitetura neural modela este aspecto do *superior culliculus*. Nela, o centro da log-polar seria deslocado segundo um vetor \vec{D} dado pela Equação 7.4.1, onde $\vec{V}(i, j)$ representa o vetor de deslocamento sacádico codificado pelo mapeamento da posição central do neurônio $n_{i,j}$ no plano da imagem segundo a transformada log-polar inversa, e $f(i, j)$ o valor de saída retornado por este neurônio após a filtragem gaussiana na saída.

$$\vec{D}(i, j) = \frac{\sum_{i,j} \vec{V}(i, j) \cdot f(i, j)}{\sum_{i,j} f(i, j)} \quad \text{Equação 7.4.1}$$

Muito embora seja de grande plausibilidade biológica, tal abordagem torna os movimentos sacádicos do sistema de Busca Visual que implementamos sujeitos à influência de ruídos ou à presença de regiões de pouca relevância na camada de entrada Φ .

Isto ocorre pois, a saída de algum neurônio cuja posição central do RF no plano de Φ esteja sobre um alvo indesejado pode apresentar uma ativação diferente de zero, ainda que baixa. Desta forma, o vetor sacádico $\vec{V}(i, j)$ deste neurônio $n_{i,j}$ influencia de forma indesejável esta soma vetorial e pode fazer com que o sistema de Busca Visual se desvie do ponto de interesse. Estes desvios podem ser vistos como uma conseqüência direta do uso da geometria log-polar no

modelo que implementamos se comparado à modelos de geração de movimentos sacádicos com geometria planar [WAL05].

Logo, uma forma de se evitar este tipo de perda mantendo-se a plausibilidade biológica do modelo é calcular a função inversa da log-polar nas coordenadas do neurônio com maior valor de saída, como mostrado na Equação 7.4.2.

$$\bar{D}(i, j) = \bar{V} \left(\arg \max_{i,j} (f(i, j)) \right) \quad \text{Equação 7.4.2}$$

Com este procedimento é possível realizar movimentos sacádicos mais precisos e menos sujeitos à ruídos, uma vez que os neurônios cujas sinapses sensibilizam uma região de interesse na imagem tendem a apresentar níveis de ativação muito mais elevados que os demais neurônios, logo, fazendo com que a busca convirja para uma região de máxima certeza [WAL05].

7.5 – Tempos de execução do sistema de busca visual

Inicialmente, foi feita a implementação sequencial da arquitetura proposta utilizando o framework MAE - Máquina Associadora de Eventos (http://www.lcad.inf.ufes.br/wiki/index.php/Máquina_Associadora_de_Eventos_-_MAE). Como mencionado anteriormente, este framework possui código fonte aberto e foi desenvolvido pelo Grupo de Pesquisa em Ciência da Cognição do LCAD. Com este, pode-se projetar tanto estruturas modulares usando RNSP com neurônios do tipo VG-RAM quanto estruturas com arquitetura em camadas, com a definição de um processamento específico para cada camada, a exemplo de [OLI05].

Uma segunda implementação paralela foi realizada visando diminuir o tempo de processamento, utilizando computação de alto desempenho através da API de programação paralela OpenMP (<http://openmp.org/wp/>), que foi empregada no framework MAE. Por fim, implementaremos uma versão utilizando CUDA.

A Tabela 7.5.1 mostra os tempos de execução das versões sequencial e paralela do sistema de busca visual em uma máquina Blade Dell R5500 com 2 processadores Intel Xeon 2.13 (Quad-core) com GPU CUDA enabled com 448 processadores com os seguintes parâmetros da arquitetura: (i) imagens de 320x240 pixels; (ii) tamanho da camada neural de 96x36; (iii) número de sinapses por neurônio igual a 256 e; (iv) número máximo de sacadas igual a 5.

Tabela 7.5.1: Tempos de execução do sistema de busca visual.

Versão	Tempo (s)
Sequencial	9,510
Paralela	8,018

8 - Meta Física 2.8/2010 - Desenvolvimento de uma versão do sistema de reconhecimento de imagens usando computação de alto desempenho

Nesta seção, apresentamos a arquitetura do sistema de reconhecimento de imagens implementado com Redes Neurais Sem Peso. Nesta direção, fornecemos inicialmente uma visão geral da arquitetura, detalhando as camadas que a compõem, como estas operam e como são conectadas entre si. Em seguida, apresentamos uma aplicação do sistema de reconhecimento de imagens e seu desempenho.

8.1 - Visão geral da arquitetura neural

No sistema de reconhecimento de imagens implementado, empregamos uma arquitetura de RNSP VG-RAM com uma camada bidimensional de neurônios, N , com $m \times n$ neurônios. A Figura 7.1.1 mostra a arquitetura neural, que tem como componente principal os neurônios N e na sua base a imagem de entrada I de dimensões $\xi \times \eta$.

Antes de chegar à entrada da rede neural, Φ (de dimensões $2\xi \times 2\eta$), a imagem I sofre uma translação, um ajuste de escala e uma normalização para minimizar as variações de iluminação. Para permitir seu uso com entradas que podem assumir valores não binários, usamos *minchinton cells* [MIT98].

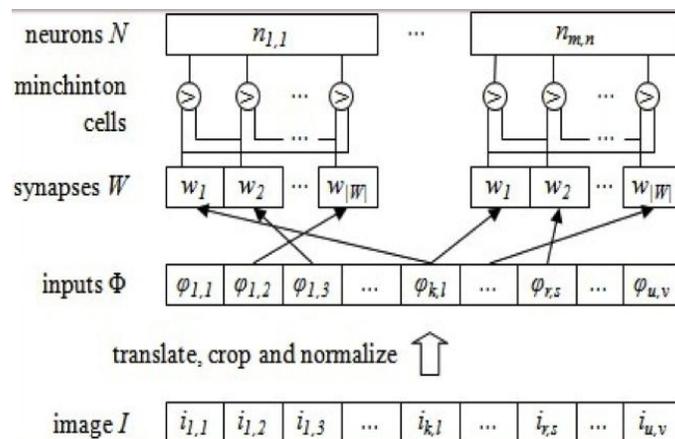


Figura 8.1.1: Arquitetura neural usada na aplicação de busca visual.

8.2 - Padrão de interconexão sináptico log-polar

Cada neurônio, $n_{i,j}$, possui um conjunto de sinapses, $W = \{w_1, \dots, w_{|W|}\}$. Estas sinapses são conectadas à entrada bidimensional da rede, Φ , composta de $2\xi \times 2\eta$ elementos de entrada, $\phi_{r,s}$, segundo um padrão de interconexão sináptico log-polar, descrito na Seção 6.2.2.

8.3 - Aplicação do sistema de reconhecimento de imagens: sistema de reconhecimento de placas de trânsito

Para avaliar o método de reconhecimento de imagens implementado, foi proposto um sistema de reconhecimento de placas de trânsito. A identificação de placas de trânsito é um importante problema para carros autônomos e sistemas de transporte inteligente, pois permite alertar o motorista (ou sistema autônomo de navegação) e comandar ou proibir determinadas ações.

Para testar o sistema desenvolvido, foi utilizada a base de dados *German Traffic Sign Recognition Benchmark* (GTSRB) disponível em [STA11]. A GTSRB contém imagens de placas de trânsito capturadas em ruas da Alemanha e metodologia para comparação dos resultados obtidos com outros sistemas de reconhecimento de placas de trânsito.

A base contém 26640 imagens de treino e 12569 imagens de teste divididas heterogeneamente em 43 classes diferentes, cada classe correspondendo a um tipo de placa de trânsito. As imagens variam em tamanho, de 15x15 até 250x250 pixels e não são necessariamente quadradas. Por serem placas reais, estão sujeitas a deteriorações físicas decorrentes do tempo e vandalismo, oclusões parciais, brilho do sol, descoloração e adesivos.

Nós submetemos os resultados da melhor configuração da rede para o sistema de avaliação do GTSRB em 15 de Setembro de 2012. Nosso sistema foi classificado em 4o. lugar, perdendo apenas para 1 método automático, conforme Figura 8.3.1.

METHOD	TOTAL
Committee of CNNs	99.46%
Human (best individual)	99.22%
Human Performance	98.84%
VGRAM WNN RGB Final 2	98.73%

Figura 8.3.1: Resultado da submissão.

8.4 – Tempos de execução do sistema de reconhecimento de placas de trânsito

Inicialmente, foi feita a implementação sequencial da arquitetura proposta utilizando o framework MAE - Máquina Associadora de Eventos (http://www.lcad.inf.ufes.br/wiki/index.php/Máquina_Associadora_de_Eventos_-_MAE). Uma segunda implementação, paralela, foi feita visando diminuir o tempo de processamento, utilizando computação de alto desempenho através da API de programação paralela OpenMP (<http://openmp.org/wp/>), que foi

empregada no framework MAE. Por fim, implementaremos uma versão utilizando CUDA.

A Tabela 8.4.1 mostra os tempos de execução (fases de treino e teste) das versões sequencial e paralela do sistema de reconhecimento de placas de trânsito proposto em uma máquina Blade Dell R5500 com 2 processadores Intel Xeon 2.13 (Quad-core) com GPU CUDA enabled com 448 processadores utilizando um subconjunto de imagens formado por 860 imagens de treino e 430 imagens de teste retiradas aleatoriamente do conjunto de teste original.

Tabela 8.4.1: Tempos de execução do sistema de reconhecimento de placas de trânsito para 860 imagens de treino e 430 de teste.

<i>Versão</i>	<i>Tempo</i>
<i>Sequencial</i>	13m53.315s
<i>Paralela</i>	11m19.959s

9 - Meta Física 3.1/2010 - Realização de seminários de acompanhamento e avaliação

Foi realizado, no dia 2 de maio de 2012, o primeiro seminário de acompanhamento e avaliação deste projeto. A agenda deste seminário é apresentada abaixo. Os slides das apresentações listadas na agenda acompanham este relatório no anexo.

1º. Seminário de Acompanhamento e Avaliação do Projeto Núcleo de Excelência em Computação de Alto Desempenho e sua Aplicação em Computação Científica e Inteligência Computacional

***Dia 2 de Maio de 2012
Centro de Treinamento da UFES***

Agenda:

09:00 - 10:00 - Apresentação da Missão "Volta da UFES" e conceitos relacionados ao seu planejamento

10:00 - 10:15 - Coffee-Break

10:15 - 11:00 - Pontos Fortes e Pontos Fracos da equipe LCAD, Desafios

e Oportunidades externas

11:00 - 12:30 - Definição dos Objetivos Estratégicos

12:30 - 14:00 - Almoço

14:00 - 15:30 - Definição das Estratégias para o alcance dos Objetivos Estratégicos

15:30 - 15:45 - Coffee-Break

15:45 - 18:00 - Definição dos Planos de Ação para a realização da Volta da UFES

10 - Meta Física 3.2/2010 – Elaboração de patentes e/ou artigos científicos associados aos resultados alcançados com o projeto

Nesta meta física, está prevista a elaboração de uma patente e/ou artigo científico associado aos resultados alcançados com o projeto. Elaboramos, submetemos e tivemos aceito para publicação o artigo científico informado abaixo. A primeira página e a página desse artigo onde consta o agradecimento à FAPES acompanham este relatório no anexo.

Moraes, J. L.; De Souza, A. F.; Badue, C. Facial Access Control Based On VG-RAM Weightless Neural Networks. In: The 2011 International Conference on Artificial Intelligence (ICAI'11), 2011, Las Vegas, USA. Proceedings of the 2011 International Conference on Artificial Intelligence (ICAI'11), 2011. p. 444-450.

Correntemente estamos discutindo a elaboração de 3 patentes relacionadas às metas 2.6-2010, 2.7-2010 e 2.8-2010 junto ao INIT (<http://portal.ufes.br/InIt>).

11 - Meta Física 4.1/2010 - Agregação de um bolsista recém-doutor à equipe do projeto

Foi agregado à equipe do projeto o bolsista recém-doutor José Jerônimo Camata, que desenvolve projeto de pesquisa sobre simulação numérica de escoamentos em ambiente de computação de alto desempenho, participando no contexto do projeto nas pesquisas relacionadas ao desenvolvimento de condicionadores baseados nas arestas dos elementos finitos e no estudo de otimizadores para os método Newton-Krylov livre de matriz.

12 - Meta Física 4.2/2010 - Agregação de dois alunos de doutorado à equipe do projeto

Foram agregados à equipe do projeto os alunos de doutorado:

1. Lucas de Paula Veronese, que desenvolve tese de doutorado sobre visão artificial tridimensional por meio de múltiplas imagens estéreo regularmente distribuídas no tempo e sua aplicação em veículos autônomos.
2. Mariella Berger, que desenvolve tese de doutorado sobre representação interna ao computador de objetos fixos e móveis baseada em visão computacional utilizando processamento de alto desempenho.

13 - Meta Física 4.3/2010 - Agregação de dois alunos de mestrado à equipe do projeto

Foram agregados à equipe do projeto os alunos de mestrado:

1. Avelino Forechi Silva, que já concluiu seu trabalho, tendo desenvolvido dissertação de mestrado sobre um sistema de navegação robótica por imagens de pontos de interesse.
2. Roberta Nunes Mattos, que já concluiu seu trabalho, tendo desenvolvido dissertação de mestrado sobre formulações estabilizadas submalhas aplicadas às equações de Euler.

14 - Meta Física 4.4/2010 - Agregação de três alunos de iniciação científica à equipe do projeto

Foram agregados à equipe do projeto os alunos de iniciação científica:

1. Judismar Arpini Júnior, que já concluiu seu trabalho, tendo desenvolvido projeto intitulado “Estudo de Técnicas de Otimização Combinatória e Teoria dos Grafos para Navegação de Veículos Autônomos”.
2. Lauro José Lyrio Júnior, que já concluiu seu trabalho, tendo desenvolvido projeto intitulado “Controle de Braço Robótico por meio de Visão Artificial”.
3. Rânik Guidolini, que já concluiu seu trabalho, tendo desenvolvido projeto intitulado “Métodos Probabilísticos de Planejamento de Caminho e de Movimento próprios para Veículos Autônomos Ackermann de Alta Velocidade”.

15 - Meta Física 2.1/2011 - Realização de seminários de acompanhamento e avaliação

Foi realizado, no dia 30 de outubro de 2012, o segundo seminário de acompanhamento e avaliação deste projeto. A agenda deste seminário é apresentada abaixo. Os slides das apresentações listadas na agenda acompanham este relatório no anexo.

2º. Seminário de Acompanhamento e Avaliação do Projeto Núcleo de Excelência em Computação de Alto Desempenho e sua Aplicação em Computação Científica e Inteligência Computacional

***Dia 30 de Outubro de 2012
Sala de Seminários do CT-VII (Aquário)***

Agenda:

13:45-14:00 – Abertura do encontro - *Alberto Ferreira De Souza*

Apresentações Técnicas

- 14:00-14:20 Aplicação dos métodos de estabilização multi-escala em problemas da indústria do petróleo
- *Lucia Catabriga*
- 14:20-14:40 Desenvolvimento e implementação de preconditionadores baseados nas arestas dos elementos finitos na biblioteca paralela EdgePack
- *Lucia Catabriga*
- 14:40-15:00 Desenvolvimento e implementação do método JFNK para formulações de elementos finitos na biblioteca paralela EdgePack
- *Lucia Catabriga*
- 15:00-15:20 Paralelização das estratégias desenvolvidas para redução da matriz Jacobiana representada por coloração de grafos
- *Marcelo Carrion*
- 15:20-15:40 Paralelização das estratégias desenvolvidas para redução de banda de matrizes esparsas oriundas de formulações de elementos finitos
- *Brenno Lugon*
- 15:40-16:00 *Coffee Break*
- 16:00-16:20 Desenvolvimento de um sistema de exploração visual usando

- computação de alto desempenho
- *Avelino Forechi*
- 16:20-16:40 Desenvolvimento de uma versão do sistema de busca visual usando computação de alto desempenho
- *Avelino Forechi*
- 16:40-17:00 Desenvolvimento de uma versão do sistema de reconhecimento de imagens usando computação de alto desempenho
- *Mariella Berger*
- 17:00-17:20 Desenvolvimento de uma versão do sistema de vergência e reconstrução 3D usando computação de alto desempenho
- *Lucas Veronese*
- 17:20-17:40 Implementação da PR e integração do sistema de vergência e reconstrução 3D à mesma
- *Alberto Ferreira De Souza*
- 17:40-18:00 – Ações futuras e Encerramento - *Alberto Ferreira De Souza*

16 - Meta Física 2.2/2011 - Elaboração de patentes e/ou artigos científicos associados aos resultados alcançados com o projeto

Nesta meta física, está prevista a elaboração de duas patentes e/ou artigos científicos associados aos resultados alcançados com o projeto. Elaboramos, submetemos e tivemos aceitos para publicação os artigos científicos informados abaixo. A primeira página e a página desses artigos onde consta o agradecimento à FAPES acompanham este relatório no anexo.

De Souza, Alberto F. ; Freitas, Fabio Daros ; COELHO DE ALMEIDA, ANDRÉ GUSTAVO. Fast learning and predicting of stock returns with virtual generalized random access memory weightless neural networks. *Concurrency and Computation*, v. 24, p. 921-933, 2012.

De Souza, A. F.; Veronese, L. P.; Lima, L. M.; Badue, C.; Catabriga, L. Evaluation of Two Parallel Finite Element Implementations of the Time-Dependent Advection Diffusion Problem: GPU versus Cluster Considering Time and Energy Consumption. In: 10th International Meeting on High-Performance Computing for Computational Science (VECPAR 2012), 2012, Kobe. *Proceedings of the 10th International Meeting on High-Performance Computing for Computational Science (VECPAR 2012)*, 2012.

17 - Meta Física 3.1/2011 - Agregação de um bolsista recém-doutor à equipe do projeto

Foi agregado à equipe do projeto o recém-doutor Edilson de Aguiar, professor Adjunto I do Departamento de Engenharias e Computação do CEUNES/UFES em São Mateus. Edilson desenvolve pesquisas nas áreas de cognição visual, visão computacional, processamento de imagem e computação gráfica.

18 - Meta Física 3.2/2011 - Agregação de um aluno de doutorado à equipe do projeto

Foi agregado à equipe do projeto o aluno de doutorado Renato Santos Pereira, que desenvolve tese de doutorado sobre estudo de técnicas otimizadoras na manipulação de matrizes esparsas com aplicações em computação científica.

19 - Meta Física 3.3/2011 - Agregação de dois alunos de mestrado à equipe do projeto

Foram agregados à equipe do projeto os alunos de mestrado:

1. Filipe Wall Mutz, que desenvolve dissertação de mestrado sobre navegação de veículos autônomos em tráfego urbano.
2. Stéfano Terzi Gasperazzo, que desenvolve dissertação de mestrado sobre estudos de algoritmos em computação natural para problemas de navegação.

20 - Meta Física 3.4/2011 - Agregação de três alunos de iniciação científica à equipe do projeto

Foram agregados à equipe do projeto os alunos de iniciação científica:

1. Jéssica Correia, que desenvolve projeto intitulado “Estudo de Otimizações em Métodos Iterativos Não Estacionários para Solução de Sistemas Lineares de Grande Porte”.
2. Lucas Catabriga, que desenvolve projeto intitulado “Fusão Probabilística de Dados de GPS, IMU, Odometria Visual, LIDAR Velodyne e Odometria das Rodas”.
3. Rafael Correia Nascimento, que desenvolve projeto intitulado “Métodos de Planejamento de Caminho e de Movimento próprios para Braços

Robóticos”.

IV - Relacione os principais fatores negativos e positivos que interferiram na execução do projeto.

O único fator negativo digno de nota foi a dificuldade que tivemos para encontrar uma empresa que fosse capaz de adaptar um automóvel para nossas pesquisas. Infelizmente não encontramos empresa nacional. A empresa Norte Americana encontrada, contudo, ofereceu o serviço de adaptação dentro dos limites dos recursos aprovados para o projeto. O automóvel já foi entregue e encontra-se em uso.

Um fator positivo importante tem sido o interesse da população em geral (notícias na imprensa sobre o projeto ligadas ao desenvolvimento de veículos autônomos podem ser encontradas em <http://lcadufes.wordpress.com/>) e dos alunos de graduação e pós-graduação, e de outros pesquisadores em particular no projeto, o que nos permitiu montar uma equipe muito competente e motivada. A tabela a seguir lista os novos alunos (ou alunos que já eram da equipe e mudaram de nível) de graduação e pós-graduação, e de pesquisadores do Projeto e suas funções.

Novos alunos (ou alunos que já eram da equipe e mudaram de nível) integrados à equipe do projeto

Nº	NOME	NÍVEL	FUNÇÃO
1	Edilson de Aguiar	Recém-doutor	Pesquisa e desenvolvimento na área de Cognição Visual
2	José Jerônimo Camata	Recém-doutor	Pesquisa e desenvolvimento na área de Computação de Alto Desempenho
3	Avelino Forechi	Doutorado	Pesquisa e desenvolvimento nas áreas de SLAM e Navegação
4	Lucas de Paula Veronese	Doutorado	Pesquisa e desenvolvimento nas áreas de SLAM e Cognição Visual
5	Mariella Berger	Doutorado	Pesquisa e desenvolvimento na área de Cognição Visual
6	Renato Santos Pereira	Doutorado	Pesquisa e desenvolvimento na área de Computação de Alto Desempenho
7	Lauro José Lyrio Júnior	Mestrado	Pesquisa e desenvolvimento nas áreas de SLAM e Cognição Visual
8	Felipe Wall Mutz	Mestrado	Pesquisa e desenvolvimento na área de Navegação
9	Tiago Alves de Oliveira	Mestrado	Pesquisa e desenvolvimento na área de Cognição Visual

10	Cayo Fontana	Mestrado	Pesquisa e desenvolvimento na área de Cognição Visual
11	Rômulo Ramos Radaelli	Mestrado	Pesquisa e desenvolvimento na área de Navegação
12	Michael André Gonçalves	Mestrado	Pesquisa e desenvolvimento na área de Navegação
13	Eduardo Max Amaro Amaral	Mestrado	Pesquisa e desenvolvimento nas áreas de SLAM e Cognição Visual
14	Stéfano Terzi Gasperazzo	Mestrado	Pesquisa e desenvolvimento na área de Navegação
15	Rânik Guidolini	Iniciação científica	Pesquisa e desenvolvimento na área de SLAM
16	Lucas Catabriga Rocha	Iniciação científica	Pesquisa e desenvolvimento na área de SLAM
17	Rafael Correia Nascimento	Iniciação científica	Pesquisa e desenvolvimento na área de Navegação
18	William de Oliveira Santos	Iniciação científica	Pesquisa e desenvolvimento na área de SLAM
19	Maurício Valli	Iniciação científica	Pesquisa e desenvolvimento nas áreas de SLAM e Cognição Visual
20	Judismar Arpini Junior	Iniciação científica	Pesquisa e desenvolvimento na área de Navegação
21	Jéssica Correia	Iniciação científica	Pesquisa e desenvolvimento na área de Computação de Alto Desempenho

V - Contatos Nacionais e Internacionais efetivamente ocorridos em função do projeto, como: convênios, pesquisadores visitantes, etc.

Graças ao projeto, estamos coordenando uma equipe nacional que participará no *DARPA Robotics Challenge* (http://www.darpa.mil/Our_Work/TTO/Programs/DARPA_Robotics_Challenge.aspx).

VI - Informe os trabalhos publicados e/ ou aceitos para publicação no período, relacionados com o projeto em pauta: livros, capítulos de livros, artigos em periódicos nacionais e internacionais, resumos em congressos, reuniões científicas e semelhantes. Anexe a primeira página dos trabalhos publicados (observar a necessidade de citação da FAPES como agência de suporte financeiro da publicação).

Berger, M.; Silva, A. F.; De Souza, A. F.; Oliveira Neto, J.; Veronese, Lucas de Paula; Badue, Claudine. Traffic Sign Recognition with VG-RAM Weightless Neural Networks. In: 12th International Conference on Intelligent Systems Design

and Applications (ISDA 2012), 2012, Kochi, India. Proceedings of the 12th International Conference on Intelligent Systems Design and Applications (ISDA 2012), 2012.

De Souza, Alberto F. ; Freitas, Fabio Daros ; COELHO DE ALMEIDA, ANDRÉ GUSTAVO. Fast learning and predicting of stock returns with virtual generalized random access memory weightless neural networks. Concurrency and Computation, v. 24, p. 921-933, 2012.

De Souza, A. F.; Veronese, L. P.; Lima, L. M.; Badue, C.; Catabriga, L. Evaluation of Two Parallel Finite Element Implementations of the Time-Dependent Advection Diffusion Problem: GPU versus Cluster Considering Time and Energy Consumption. In: 10th International Meeting on High-Performance Computing for Computational Science (VECPAR 2012), 2012, Kobe. Proceedings of the 10th International Meeting on High-Performance Computing for Computational Science (VECPAR 2012), 2012.

Veronese, L. P.; Lyrio Júnior, L. J.; Mutz, F. W.; Oliveira Neto, J.; Azevedo, V. B.; Berger, M.; De Souza, A. F.; Badue, C. Stereo Matching with VG-RAM Weightless Neural Networks. In: 12th International Conference on Intelligent Systems Design and Applications (ISDA 2012), 2012, Kochi, India. Proceedings of the 12th International Conference on Intelligent Systems Design and Applications (ISDA 2012), 2012.

Bodart, M.L.O. ; Catabriga, Lucia ; COUTINHO, A. L. G. A. . FORCING TERM EFFECTS ON INEXACT NEWTON-KRYLOV METHOD FOR SOLVING NONLINEAR EQUATIONS EMANATING FROM THE SUPG/PSPG FINITE ELEMENT FORMULATION OF NAVIER-STOKES EQUATIONS. In: 32th Iberian Latin American Congress on Computational Methods in Engineering, 2011, Ouro Preto. Proceedings of the 32th Iberian Latin American Congress on Computational Methods in Engineering. Belo Horizonte: ABMEC, 2011. v. 1. p. 1-14.

Moraes, J. L.; De Souza, A. F.; Badue, C. Facial Access Control Based On VG-RAM Weightless Neural Networks. In: The 2011 International Conference on Artificial Intelligence (ICAI'11), 2011, Las Vegas, USA. Proceedings of the 2011 International Conference on Artificial Intelligence (ICAI'11), 2011. p. 444-450.

Local e data

Nome e Assinatura do Representante

REFERÊNCIAS

- [ARR10] ARRUDA, N. C. B.; ALMEIDA, R. C.; DO CARMO, E. G. D., Dynamic diffusion formulations for advection dominated transport problems. *Mecânica Computacional* Vol XXIX, págs. 2011-2025 Eduardo Dvorkin, Marcela Goldschmit, Mario Storti (Eds.) .AMCA, Buenos Aires, Argentina, 15-18 Noviembre, 2010.
- [ALE66] ALEKSANDER, I. Self-adaptive universal logic circuits. *IEEE Electronic Letters*, v. 2, n. 8, p. 231-232, 1966.
- [ALE98] ALEKSANDER, I. RAM-Based Neural Networks From WISARD to MAGNUS: a Family of Weightless Virtual Neural Machines. In: AUSTIN, J. RAM-Based Neural Networks. [S.I.]: World Scientific, 1998. p. 18-30.
- [BAD08] BADUE, C.; PEDRONI, F.; DE SOUZA, A. F. Multi-Label Text Categorization using VG-RAM Weightless Neural Networks. *Proceedings of the 10th Brazilian Symposium on Neural Networks (SBRN'08)*, Salvador, BA, 2008. 105-110.
- [BAL10] BALAY, S.; BROWN, J.; BUSCHELMAN, K.; EIJKHOUT, V.; GROPP, W. D.; KAUSHIK, D.; KNEPLEY, M. G.; MCINNES, L. C.; SMITH, B. F.; ZHANG, H., *PETSc Users Manual*. ANL-95/11 - Revision 3.1, Argonne National Laboratory, 2010.
- [BAR95] BARNAD, S.; PHOTEN, A.; SIMON, H, A spectral algorithm for envelope reduction of sparse matrices. *Numerical Linear Algebra with Applications*, v. 3, p. 317–334, 1995.
- [CAM12] CAMATA, J. J.; ROSSA, A. L.; VALLI, A. M. P.; CATABRIGA, L.; CAREY, G. F.; COUTINHO, A. L. G. A., Reordering and incomplete preconditioning in serial and parallel adaptive mesh refinement and coarsening flow solutions. *International Journal for Numerical Methods in Fluids*, 69(4):802-823, June 2012. <http://dx.doi.org/10.1002/flid.2614>.
- [CAR09] CARVALHO, C. A.; VERONESE, L. P.; OLIVEIRA, H.; DE SOUZA, A. F., Implementation of a Biologically Inspired Stereoscopic Vision Model in C+CUDA, accepted in GPU Technology Conference, Sep 30, 2009.
- [CAT02] CATABRIGA, L.; COUTINHO, A. L. G. A., “Improving Convergence to Steady-State of Implicit SUPG Solution of Euler Equations”, *Communications in Numerical Methods in Engineering*, Vol. 18, pp. 345--353, 2002a.
- [CAT09] CATABRIGA, L.; DE SOUZA, A. F.; COUTINHO, A. L. G. A.; TEZDUYAR, T. E., “3D Edge-Based SUPG Computation of Inviscid Compressible Flows with YZ β Shock-Capturing”, *Computational Mechanics*, v. 76, p. 021208, 2009.
- [CUT69] CUTHILL, E.; MCKEE, J., Reducing the bandwidth of sparse symmetric matrices. In: *Proceedings of the 24th ACM national conference*. New York, NY, USA: ACM Press, 1969. p. 157–172.
- [DAV97] Davis T. University of Florida Sparse Matrix Collection, NA Digest, 97 (1997); <http://www.cise.ufl.edu/research/sparse/matrices>.
- [DEO74] DEO, N., “Graph Theory with Applications to Engineering and Computer Science”, Prentice-Hall, Inc. Englewood Cliffs, N.J., USA, 1974.
- [DES07] DE SOUZA, A. F. et al. Automated Free Text Classification of Economic Activities using VG-RAM Weightless Neural Networks. *Proceedings of the 7th International Conference on Intelligent Systems Design and Applications*, 2007. 782-787.
- [DES08a] DE SOUZA, A. F. et al. Improving VG-RAM WNN Multi-label Text Categorization via Label Correlation. *8th International Conference on Intelligent Systems Design and Applications*, 2008.
- [DES08b] DE SOUZA, A. F. et al. Face Recognition with VG-RAM Weightless Neural Networks. *Lecture Notes in Computer Science*, v. 5163, n. 1, p. 951-960, 2008.
- [DES09] DE SOUZA, A. F. et al. Automated Multi-label Text Categorization with VG-RAM Weightless Neural Networks. *Neurocomputing*, v. 72, p. 2209-2217, 2009.
- [EIS96] EISENSTAT, S.C.; WALKER, H.F, Choosing the forcing terms in an Inexact

- Newton method. *SIAM J. Sci. Comput.*, 17:16–32, 1996.
- [ELI05] ELIAS, R. N.; MARTINS, M. A. D.; COUTINHO, A. L. G. A., “Parallel Edge-Based Inexact Newton Solution of Steady Incompressible 3D Navier-Stokes Equations”, *Lectures Notes on Computer Sciences*, v. Unico, n. 3648, p. 1237-1245, 2005.
- [FRE00] FREEMAN, W. T.; PASZTOR, E.; CARMICHAEL, O. T., *Learning low-level vision*, *IJCV*, 40(1):25–47, 2000.
- [GEB05] GEBREMEDHIN, A.; MANNE, F.; POTHEN, A., “What color is your Jacobian? Graph coloring for computing derivatives”, *SIAM Review*, 47(4):629–705, 2005.
- [GEO71] GEORGE, J. A., *Computer implementation of the finite element method*. Tese de Doutorado, Stanford University, Stanford, CA, USA, 1971.
- [GEO89] GEORGE, A.; LIU, J., The evolution of the minimum degree ordering algorithm. *SIAM Review*, 31:1–19, 1989.
- [GHI11] GHIDETTI, K.; CATABRIGA, L.; BOERES, M. C. S.; RANGEL, M. C., A study of the influence of sparse matrices reordering algorithms for ILU(p) preconditioner on the GMRES method. In: *Fifth SIAM Workshop on Combinatorial Scientific Computing*, 2011, Darmstadt. Technical Report / Aachener Informatik-Bericht (AIB) 2011-09. Darmstadt: RWTH Aachen University, 2011. v. 1. p. 125-127.
- [GOM08] GOMES-RUGGIERO, M.; LOPES, V.; TOLEDO-BENAVIDES, J., A globally convergent inexact Newton method with a new choice for the forcing term. *Annals of Operations Research*, 157:193–205, 2008. 10.1007/s10479-007-0196-y.
- [HEN95] HENDRICKSON, B.; LELAND, R., *The chaco user's guide version 2.0*. [S.l.], 1995. Disponível em: <<http://www.sandia.gov/media/chaco.htm>>.
- [HUB95] HUBEL, D. H. *Eye, Brain and Vision*. [S.l.]: Scientific American Library, 1995.
- [KAN00] KANDEL, E. R. . S. J. H. . J. T. M. *Principles of Neural Science*. 4th Ed. ed. [S.l.]: Prentice-Hall International, Inc., 2000.
- [KOM02] KOMATI, K. S.; DE SOUZA, A. F. *Vergence Control in a Binocular Vision System using Weightless Neural Networks*. *Proceedings of the 4th International Symposium on Robotics and Automation*, 2002.
- [LEE88] LEE, C.; W.H., R.; SPARKS, D. L. *Population coding of saccadic eye movements by neurons in the superior colliculus*. *Nature*, p. 332: 357–360, 1988.
- [LIU76] LIU, W.; SHERMAN, A., *Comparative analysis of the cuthill-mckee and the reverse cuthill-mckee ordering algorithms for sparse matrices*. *SIAM J. Num. Anal.*, v. 13, p. 198–213, 1976.
- [LUD99] LUDERMIR, T. B. et al. *Weightless neural models: a review of current and past works*. *Neural Computing Surveys*, v. 2, 1999. ISSN 41-61.
- [MAT12] MATTOS, R. N.; SANTOS, I. P.; CATABRIGA, L., *Estudo numérico da formulação estabilizada submalha difusão dinâmica aplicada às equações de Euler compressíveis*. In: *10th World Congress on Computational Mechanics*, 2012, São Paulo. *Proceedings of the Computational Mechanics 2012*. São Paulo: P.M.Pimenta, E.M.B. Campello Editors, 2012. v. 1. p. 1-17.
- [MID11] *Middlebury Stereo Vision Page* – <http://vision.middlebury.edu/stereo/>, consultado em 05/08/2011
- [MIT98] MITCHELL, R. J. et al. *Comparison of Some Methods for Processing Grey Level Data in Weightless Networks*. In: *RAM-Based Neural Networks*. [S.l.]: World Scientific, 1998. p. 61-70.
- [NET12] NETO, J. D. O. *Um Sistema de Busca Visual Biologicamente Plausível Baseado em Redes Neurais Sem Peso*. 2012. Monografia (Graduação em Engenharia de Computação). Departamento de Informática, Universidade Federal do Espírito Santo, Vitória, 2012.
- [NVI11] NVIDIA, “*NVIDIA CUDA: Compute Unified Device Architecture - Programming Guide 4.0*”, 2011.
- [OLI05] OLIVEIRA, H. *Uma Modelagem Computacional de Áreas Corticais do Sistema Visual Humano Associadas à Percepção de Profundidade*. 2005. Dissertação (Mestrado em Informática). Departamento de Informática, Universidade Federal do

Espírito Santo, Vitória, 2005.

- [PAP91] PAPADRAKAKIS, M.; BALOPOULOS, V., Improved quasi-Newton methods for large non-linear problems. *J. Eng. Mech., ASCE*, 117:1201–1219, 1991.
- [POR03] PORT, N. L.; WURTZ, R. H. Sequential activity of simultaneously recorded neurons in the superior colliculus during curved saccades. *Journal of Neurophysiology*, 2003. 1887–1903.
- [SAN07] SANTOS, I. P.; ALMEIDA, R. C., “Nonlinear Subgrid Method for Advection - Difusion Problems”, *Comput. Methods Appl. Mech. Engrg.*, vol. 196, pp. 4771–4778, 2007.
- [SLO88] S.W. SLOAN, A Fortran program for profile and wavefront reductions»*Int. Jnl. Num. Meth. Engng.* 28, pp.2651-2679, 1988.
- [STA11] STALLKAMP, J.; SCHLIPSING, M.; SALMEN, J.; INGEL, C., The german traffic sign recognition benchmark: a multi-class classification competition, *The 2011 International Joint Conference on Neural Networks*, pp. 1453-1460, 2011.
- [STO06] STOGNER, R. H.; KIRK, B. S.; PETERSON, J. W.; CAREY, G. F., libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations. *Journal Engineering with Computers* 2006; 22:237–254.
- [SUN03] SUN, J.; ZHENG, N.; SHUM, H. Y., Stereo matching using belief propagation, *PAMI*, 25(7):787–800, 2003.
- [TEZ99] TEZDUYAR, T., *Finite elements in fluids: Lecture notes of the short course on finite elements in fluids*, 1999. Computational Mechanics Division, , vol. 99–77, Japan Society of Mechanical Engineers, Tokyo, Japan.
- [THR05] THRUN, S.; BURGARD, W.; FOX, D., *Probabilistic Robotics*, MIT Press, 2005.
- [WAL05] WALTON, M. M. G.; SPARKS, D. L.; GANDHI, N. J. Simulations of Saccade Curvature by Models That Place Superior Colliculus Upstream From the Local Feedback Loop. *Journal of Neurophysiology*, 2005. 2354-2358.
- [YAN10] YANG, Q.; WANG, L.; AHUJA, N., A Constant-Space Belief Propagation Algorithm for Stereo Matching, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2010.
- [YU07] YU, T.; LIN, R. S.; B. S.; TANG, B., Efficient message representations for belief propagation. In *ICCV* , pages 1–8, 2007.

Anexos Impressos
(Primeira página e página dos Trabalhos
Publicados onde consta o agradecimento
à FAPES)

Anexos Impressos
(Apresentações dos Seminários 1 e 2)