BSSLibrary – Uma biblioteca de rotinas vetorizadas para filtragem de dados em radioastronomia solar.

André Ricardo F. Martinon
DAS/INPE
martinon@das.inpe.br

Hanumant S. Sawant
DAS/INPE
sawant@das.inpe.br

Stephan Stephany
LAC/INPE
stephan@lac.inpe.br

Airam Jonatas Preto
LAC/INPE
airam@lac.inpe.br

Kleber de Mattos Dobrowoslki
DAS/INPE
kleber@das.inpe.br

Resumo

O BSSLibrary é uma biblioteca de rotinas desenvolvida para filtragem de dados espectrais de radioastronomia solar. Várias dessas rotinas foram vetorizadas de forma a otimizar seu desempenho computacional, utilizando as instruções vetoriais disponíveis nas famílias atuais de processadores com arquitetura IA-32. O BSSLibrary insere-se no software BSSData, o qual provê uma interface de tratamento de dados e visualização em radioastronomia para o Brazilian Solar Spectroscope (BSS), espectrógrafo digital decimétrico de banda larga, um radiotelescópio de alta resolução temporal e espectral, que se encontra operacional no DAS/INPE. Os resultados obtidos foram satisfatórios, quanto à qualidade da filtragem e quanto ao desempenho computacional no caso das rotinas vetorizadas.

Abstract

The BSSLibrary is a library of routines intended for the filtering of Solar Radioastronomy spectral data. Many of these routines were vectorized in order to reduce the processing time. Such routines make use of the vector instructions that are available in the current generation of IA-32 architecture processors. The BSSLibrary is part of the BSSData software, an interface for data preprocessing and visualization in Radioastronomy. This software is currently being developed for the Brazilian Solar Spectroscope (BSS), that operates in the decimetric range of wavelengths and has high temporal and spectral resolution. The BSS is already operation at the DAS/INPE. Results shows that the BSSLibrary routines yielded suitable filtering and also good computational performance, in the case of the vectorized routines.

1. Introdução

A partir de 1996 entrou em operação regular no Inpe, em São José dos Campos, um Espectrógrafo Digital Decimétrico de Banda Larga, batizado de *Brazilian Solar Spectroscope* (BSS). Com o objetivo de realizar observações solares para investigar fenômenos associados com a liberação de energia dos *flares* solares, através da análise das explosões solares decimétricas.

Em abril de 1998, iniciaram-se as observações solares sistemáticas e, desde então, muitas horas de observações foram registradas digitalmente. Foram observados muitos eventos relacionados com as explosões solares havendo a necessidade de criação de um catálogo de explosões solares. Este catálogo possui aproximadamente 340 explosões solares, devidamente classificadas de acordo com sua morfologia.

Para visualizar e analisar essas explosões contamos com dois softwares: BSSView[5] e BSSData[10]. Sendo o BSSView voltado para visualização e geração de imagens. E o BSSData para tratamento e análise dos dados. Mas atualmente, ambos os *softwares*, não oferecem todas as ferramentas necessárias, principalmente, para análise dos dados referentes as explosões solares.

Dentre estas ferramentas podemos citar a necessidade de extrair informações referentes às explosões de forma mais automatizada. Mas antes, precisamos realizar um tratamento nos dados brutos para que os algoritmos de extração de dados obtenham bons resultados. Portanto, este trabalho concentra-se no desenvolvimento de rotinas para filtragem do ruído de *background* presente nos dados. O objetivo desta filtragem é melhorar o contraste entre as regiões que possuem explosões com as regiões sem explosões.

Para solucionar esta demanda por novas ferramentas de análise de dados, o software BSSData foi reestruturado, de forma que todas as partes que o compõe fossem divididas em módulos distintos, facilitando o desenvolvimento progressivo de novas ferramentas. Para

conseguir essa modularização foram utilizadas técnicas de programação orientada à objetos. Bem como a separação, da interface do usuário das rotinas que manipulam os dados. Assim, atualmente, o software BSSData é integrado por uma interface com o usuário e uma biblioteca otimizada de rotinas, denominada BSSLibrary.

Essa primeira versão do sistema, foi desenvolvida especificamente para o sistema operacional Windows. Mas estamos estudando a possibilidade de portá-lo para o sistema operacional Linux.

A seguir descreveremos os equipamentos que compõem o *Brazilian Solar Spectroscope*. Segue-se, então, uma seção sobre a filtragem digital dos dados do BSS e alguns resultados preliminares são apresentados. Nos duas seções subsequentes, são apresentados o software BSSData e as técnicas de vetorização utilizadas para otimizá-lo, bem como, resultados preliminares empregando essas técnicas. Por fim, são apresentados os comentários finais sobre o trabalho desenvolvido.

2. Brazilian Solar Spectroscope

Um Espectrógrafo Digital Decimétrico de Banda Larga, batizado de *Brazilian Solar Spectroscope* (BSS) [13][14] está em operação regular no INPE, em São José dos Campos, desde 1996, para observações solares com alta resolução espectral e temporal. Porém, apenas em abril de 1998, foram iniciadas as observações solares sistemáticas, com o sistema completo de aquisição e visualização dos dados digitalizados.

O BSS é o único espectrógrafo solar com tais características no Hemisfério Sul. De modo que as observações solares realizadas com este instrumento aproximadamente entre 11 e 19 UT são de grande importância. Devido a sua localização (longitude ~45^{o} Oeste), entre 16 e 19 UT, o BSS é o único espectrógrafo em operação para observação solar e portanto preenche uma lacuna no monitoramento da emissão solar em rádio através de uma rede observatórios localizados em diferentes longitudes.

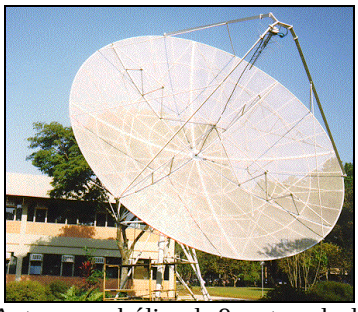


Figura 1: Antena parabólica de 9 metros de diâmetro

O BSS funciona em conjunto com uma antena parabólica de 9 metros de diâmetro (Figura 1), em cujo foco foi instalado um alimentador de banda larga composto de duas antenas log-periódicas cruzadas (Figura 2), que são restaurados através de um circuito somador [11][12][6]. Depois de somado, o sinal é introduzido em um analisador de espectros (HP8559A), depois os sinais de variação de tensão na saída do analisador seguem para os sistemas de aquisição e monitoramento dos dados. Finalmente, é feita a recepção e varredura do sinal, que é enviada para os sistemas de aquisição e monitoramento (Figura 3).

Figura 2: Alimentador log-periódico de banda larga instalado no foco da antena parabólica de 9 metros.

O objetivo das observações solares realizadas é investigar fenômenos associados com a liberação da energia dos *flares* solares, através da análise das explosões solares decimétricas, observadas principalmente acima de 1000 MHz, com altas resoluções temporal e espectral, pois as explosões decimétricas têm origem próximo às regiões de aceleração de partículas durante os *flares*.

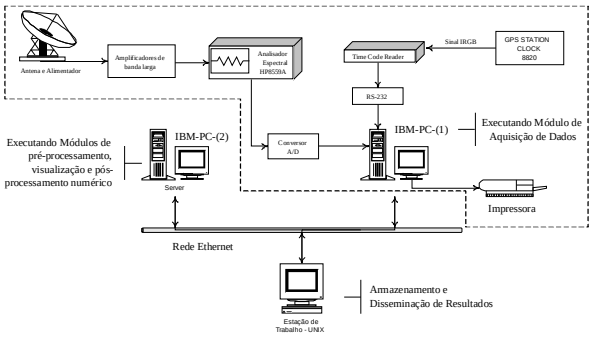


Figura 3: Diagrama de blocos representando a configuração atual do BSS

Através do BSS dados de explosões solares são digitalizados em até 100 canais de freqüência com resolução temporal de até 10 ms, gerando um fluxo de dados de até 35 Mbytes/hora. Estes dados são correntemente visualizados através de um software desenvolvido em IDL (BSSView) [5]. Neste contexto, encontra-se em fase de desenvolvimento um software para tratamento e análise de dados espectrais de

astronomia solar escrito na linguagem C/C++ e denominado BSSData [10].

Os dados adquiridos pelo BSS são armazenados em arquivos binários, em um formato próprio com a denominação ESP, que contém um cabeçalho e uma matriz de dados. Neste cabeçalho são armazenadas informações referentes à aquisição, como por exemplo: tempos inicial e final, quantidade de canais, quantidade de varreduras etc.

Em regime normal de operação, o BSS digitaliza uma matriz de 6000 varreduras e 100 canais a cada 5 minutos, gerando um arquivo de 1.14 Mbytes. Desta forma temos um fluxo de dados de 13.7 Mbytes/hora, podendo chegar a um fluxo de dados máximo de 35 Mbytes/hora.

Esses arquivos são exibidos ao usuário na forma de espectros dinâmicos, onde todos os canais de freqüência e varreduras são plotados em uma imagem bidimensional e tendo a intensidade do sinal sendo representada por cores de uma paleta de cores pré-definida. Na Figura 4 podemos ver um exemplo típico de um espectro dinâmico. Nas ordenadas estão associados os canais de freqüência enquanto nas abscissas as varreduras, ou seja, cada linha horizontal representa a variação da intensidade do sinal para uma certa freqüência em relação ao tempo.

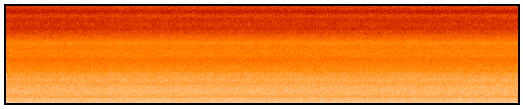


Figura 4: Exemplo de um espectro dinâmico.

Quando ocorre um evento (explosão solar), a intensidade do sinal aumenta em um ou mais canais de freqüência para um dado intervalo de tempo. Um exemplo de espectro dinâmico onde ocorre o registro de uma explosão solar pode ser visto na Figura 5.

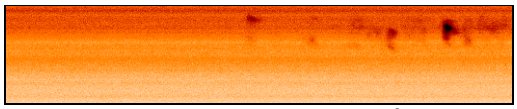


Figura 5: Exemplo de um espectro dinâmico com a ocorrência de uma explosão solar.

3. Filtragem Digital de Dados

Entre os anos de 1999 e 2002 foram catalogadas, aproximadamente, 340 explosões solares de 8 tipos diferentes. Agora existe a necessidade de analisar e extrair informações destas explosões, mas para isto deve-se desenvolver ferramentas computacionais que auxiliem neste processo.

Na análise detalhada de cada grupo de explosões solares observado deve-se considerar a variação do fluxo do *background* solar (fluxo do sol calmo) em função da freqüência e a variação temporal, além da complexidade das explosões e estruturas finas registradas superpostas ao background variável.

O sinal de *background* registrado pelo BSS não é homogêneo em freqüência, em virtude da resposta do sistema não ser exatamente idêntica para cada freqüência. Tornando esse sinal mais homogêneo as explosões mais fracas, que antes estavam misturadas ao próprio *background*, ficam mais realçadas. Isso é conseguido através de técnicas de filtragem digital de dados.

Os filtros digitais são usados geralmente para duas finalidades: (1) separação de sinais e (2) restauração de sinais. A separação de sinais é necessária quando um sinal tenha sido contaminado com interferência, ruído, ou outros sinais como no caso deste trabalho. Já a restauração de sinais é usada quando um sinal tenha sido distorcido de alguma forma.

Atualmente vários procedimentos de extração de informações de explosões solares registradas pelo BSS são feitos de forma não muito conveniente, ou seja, o usuário carece de ferramentas que o auxiliem satisfatoriamente neste processo. Desta forma, para determinar os valores usa-se apenas o espectro dinâmico e os seus respectivos perfis temporal e espectral, e todo o processo de identificação é feito visualmente através do *software* BSSData. Consequentemente tem-se uma margem de erro grande uma vez que cada pessoa pode adotar uma metodologia diferente para extrair uma mesma informação. Principalmente no caso das emissões mais fracas. Para sanar, ou ao menos diminuir essas dificuldades, optou-se pelo uso de algoritmos de PDI (Processamento Digital de Imagens) com o objetivo de obter um melhor contraste entre as áreas da imagem que apresentam informações sobre explosões e o *background*. Para posterior aplicação de algoritmos para detecção de bordas das áreas com explosões, classificando a imagem em duas áreas distintas: com e sem explosão.
De posse desta imagem seriam extraídos e quantificados os atributos de interesse, tais como obtenção dos tempos de subida e descida do sinal de uma explosão, da duração total do fenômeno, da taxa de deriva em freqüência, da banda espectral, da localização do pico, etc.

Para alcançarmos o nosso objetivo, o de extrair informações relevantes dos dados, inicialmente devemos submete-los a uma filtragem, melhorando assim o contraste das explosões em relação ao background.

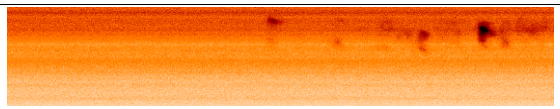
Inicialmente, estamos definindo e testando em quais categorias de explosões os filtros poderão ser aplicados de forma a obter-se o resultado esperado.

Alguns filtros já foram implementados obtendo um bom resultado. Mas ainda deve-se verificar se estamos perdendo informações com a aplicação de tais filtros, além de criar meios para se quantificar esta perda.

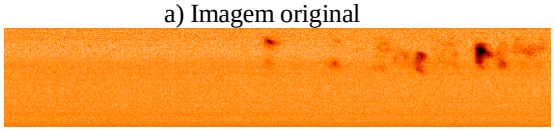
Os filtros testados, e com os melhores resultados, até o momento foram o da média, mediana e dilatação.

Antes de aplicar os filtros de média, mediana e dilatação os dados foram submetidos a um algoritmo para "remoção" de *background*. Este algoritmo funciona determinando uma amostra dos dados que possa ser representativa do sinal de *background*, segue-se então o cálculo da média dos valores de cada canal da amostra. Por fim os valores médios de cada canal são subtraídos dos dados originais. Com isso, obtemos um *background* homogêneo em todas as freqüências.

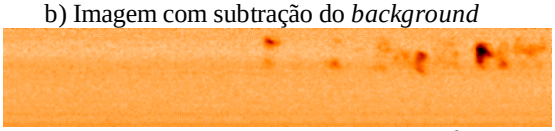
Com o *background* homogêneo aplicamos os outros filtros com o intuito de melhorar ainda mais o contraste entre as explosões e o *background*. A seguir mostramos o resultado da aplicação dos filtros de média, mediana e dilação onde fica claramente visível a eficiência do filtro de dilatação.



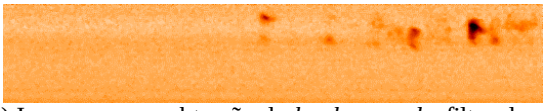
a) Imagem original



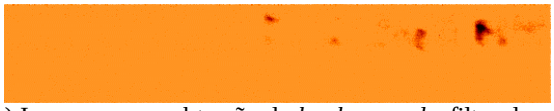
b) Imagem com subtração do *background*



c) Imagem com subtração do *background* e filtro da média 3x3.

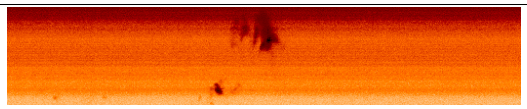


d) Imagem com subtração do *background* e filtro da mediana 3x3

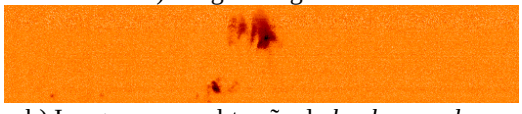


e) Imagem com subtração do *background* e filtro de dilatação 3x3.

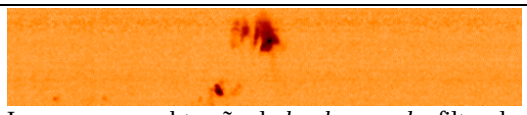
Figura 6: Primeiro teste.



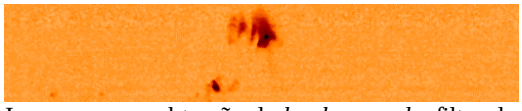
a) Imagem original.



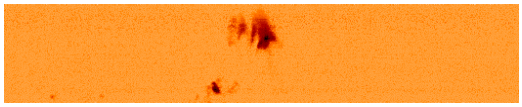
b) Imagem com subtração do *background*.



c) Imagem com subtração do *background* e filtro da média 3x3.



d) Imagem com subtração do *background* e filtro da mediana 3x3.



e) Imagem com subtração do *background* e filtro de dilatação 3x3.

Firgura 7: Segundo teste.

4. BSSData

O BSSData [10] (Figura 8) auxilia na análise dos dados, contendo ferramentas para destacar e determinar visualmente os parâmetros das explosões, manipular as cores do espectro dinâmico e organizar os dados em conjuntos distintos.

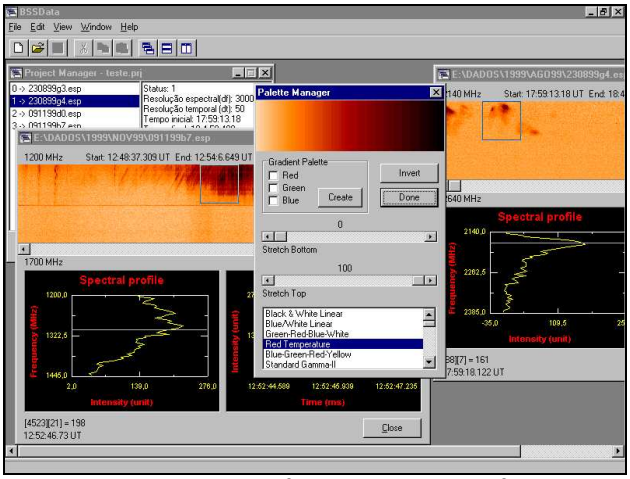


Figura 8: Exemplo de funcionamento do software BSSData

Este software integra uma interface com o usuário a uma biblioteca de rotinas, denominada BSSLibrary.

A versão preliminar da BSSLibrary foi desenvolvida em linguagem C/C++ padrão POSIX e linguagem *assembly*. E ela foi implementada na forma de uma DLL (*Dynamic Link Library*) para o sistema operacional Windows. Estuda-se a possibilidade de portá-la para o sistema operacional Linux, bem como desenvolver versões otimizadas utilizando técnicas de vetorização, descritas adiante.

Já a interface com o usuário do BSSData foi desenvolvida utilizando-se a ferramenta de programação Borland C++ Builder 6.0. Existe a possibilidade de portá-la para o sistema operacional Linux, visto que, a empresa que desenvolveu o Borland C++ Builder 6.0 recentemente colocou no mercado outra ferramenta similar para o Linux, denominada Borland Kylix 3. Com esta ferramenta poucas rotinas precisarão ser adaptadas no processo de desenvolvimento no Linux.

Como características principais dessa versão, inicialmente desenvolvida no escopo de um projeto PIBIC [10] e aperfeiçoada durante o projeto de mestrado, podemos destacar:

- O sistema está sendo desenvolvido de forma modular, permitindo maior facilidade e produtividade em futuras atualizações.
- O número de arquivos abertos simultaneamente é teoricamente ilimitado, sendo limitado apenas pelo limite máximo de memória virtual (Memória principal mais área de *swap*, menos espaço dedicado ao sistema operacional).
- Foi desenvolvida uma estrutura de classes, através de técnicas de programação orientada à objetos, de rotinas e dados de modo a permitir fácil inclusão de novas rotinas de tratamento de dados.
- A maioria das rotinas gráficas do Borland C++ Builder foi reescrita em C++ de forma a otimizar seu desempenho computacional.

A versão atual do módulo de filtragem do BSSData possui os seguintes filtros implementados:

- **Rotina para subtrair o background**: Esse *background* refere-se ao fluxo do sol calmo. Aplicando essa rotina temos como resultado um espectro dinâmico com um fundo mais homogêneo, onde as explosões ficam mais realçadas.
- **Filtro por derivadas**: Esse filtro realça as explosões dando um aspecto de relevo à imagem (pseudo 3D). A principal finalidade desse filtro é ajudar na identificação das fases de subida e descida do sinal (início e final do fenômeno de interesse).
- **Convolução**: possibilita a utilização de filtros lineares passa-baixas e passa-altas, visando a eliminação de ruídos nos dados, através da modificação de sua máscara.
- **Filtro da Mediana**: permite identificar a intensidade média do sinal, podendo-se escolher uma máscara 3x3 ou 5x5.
- **Filtros Morfológicos**: foram implementadas a dilação e a erosão, usando técnicas de morfologia matemática, podendo-se escolher os elementos estruturantes.

5. Vetorização

Para otimizar as rotinas já implementadas no software BSSData e as que serão implementadas ao BSSLibrary, utilizaremos dois conjuntos de instruções presentes na maioria dos processadores atuais. Estes conjuntos de instruções utilizam técnicas SIMD (*Single Instruction Multiple Data*), ou seja, uma única instrução é usada para executar a mesma operação em mais de uma variável.

Para otimizar as rotinas que envolvam operações somente com números inteiros, utilizaremos o conjunto de instruções MMX [7][8][9]. Já as que envolvem operações em ponto flutuante, serão otimizadas através do conjunto de instruções 3DNow! [1][2][3][4] (presente nos processadores Athlon e Duron da AMD).

A seguir esses dois conjuntos de instruções, MMX e 3DNow!, serão detalhados.

Algumas rotinas do software BSSData já foram otimizadas utilizando as instruções MMX. A seguir mostraremos os resultados dos testes realizados em relação ao tempo de processamento das rotinas.

Nestes testes, confrontamos rotinas escritas em C com rotinas similares em linguagem *assembly* utilizando instruções MMX. Esperava-se uma diminuição substancial no tempo de processamento, devido à utilização destas instruções.

Os testes foram conduzidos em um microcomputador padrão IBM/PC com processador AMD Athlon 1.2 Ghz, com 256 Mb de memória RAM, e tendo como sistema operacional o Linux (distribuição Conectiva 7.0, *kernel* 2.4.5).

Foram utilizados os compiladores GCC 2.95.3 e NASM 0.98 (compilador *assembly*). O programa foi executado em prioridade máxima, de forma a obter-se um valor preciso do tempo de processamento, valor este obtido através da instrução RDTSC (linguagem *assembly*).

Foi utilizado um vetor aleatório de 600.000 *bytes* (6000 x 100). Este tamanho foi escolhido pois reflete a realidade dos dados com os quais o software BSSData deverá trabalhar em regime normal. Exceto para a rotina de transposição foi utilizado um vetor aleatório de 640.000 *bytes* (800x800) por causa das limitações da rotina.

Comparou-se o desempenho das rotinas em C, contra rotinas utilizando MMX com o vetor alinhado (MMXa) e não alinhado (MMX). O alinhamento do vetor é uma tarefa realizada automaticamente nos programas em linguagem C, mas foi necessário realizar este procedimento manualmente no caso do MMX, pois caso contrário existe uma perda de desempenho, como pode ser visto na Tabela 1. Os valores absolutos dos testes são exibidos na Figura 9. Os números em parênteses foram normalizados utilizando os valores das rotinas em C como base.

Rotina	C	MMX	MMXa
Contraste	8.357 (1 x)	2.716 (3.08 x)	2.584 (3.23 x)
Brilho	6.439 (1 x)	2.663 (2.42 x)	2.439 (2.64 x)
MinMax (byte)	5.942 (1 x)	1.340 (4.43 x)	1.205 (4.93 x)
MinMax (word)	7.100 (1 x)	2.665 (2.66 x)	2.420 (2.93 x)
Zoom 2x	7.804 (1 x)	2.778 (2.81 x)	2.544 (3.07 x)
Zoom 4x	7.267 (1 x)	2.485 (2.92 x)	2.300 (3.16 x)
Transposta	21.397 (1 x)	2.553 (8.38 x)	2.288 (9.35 x)

Tabela 1: Comparação de desempenho (Valores em quilociclos do processador).

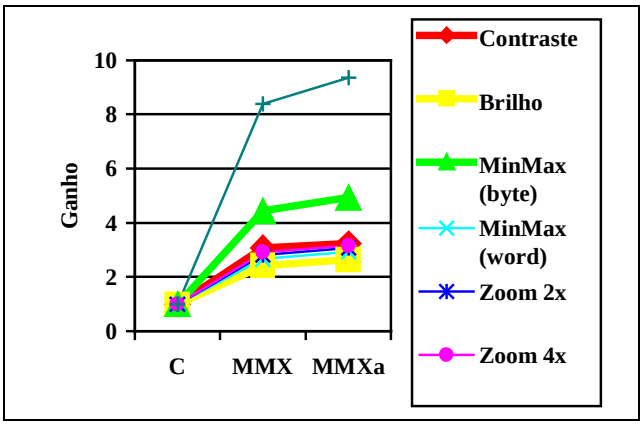


Figura 9: Gráfico comparativo de desempenho.

6. Comentários Finais

A BSSLibrary foi implementada satisfatoriamente e está sendo expandida, de forma a incorporar mais funções. Está em andamento um estudo sobre grupos morfológicos de explosões solares e a adequabilidade dos diversos filtros para cada tipo/grupo de explosão, o que permitiria aperfeiçoar os filtros existentes na biblioteca BSSLibrary.

A vetorização de algumas rotinas da biblioteca BSSLibrary melhorou significativamente o desempenho computacional das mesmas, e esperamos estender o uso dessa técnica às demais rotinas. Esperamos obter versões mais aperfeiçoadas do software BSSData e da biblioteca de rotinas BSSLibrary, assim como, versões preliminares de ambas para o sistema operacional Linux.

7. Referências

[1] AMD. *AMD Extensions to the 3DNow! and MMX instruction sets manual*. [online].<http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/22466.pdf>. March 2000.

[2] AMD. *3Dnow! Technology manual*. [online]. <http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/21928.pdf>. March 2000.

[3] AMD. *AMD Athlon processor x86 code optimization guide*. [online]. <http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/22007.pdf>. February 2002

[4] AMD. *3DNow! Instruction porting guide application note*. [online]. <http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/22621.pdf>. August 1999.

[5] Faria, C., *Dissertação de Mestrado*, INPE, 1999.

[6] Fernandes, F.C.R., *Tese de Doutorado*, INPE, 1997.

[7] Intel. *Programming with the Intel MMX technology*. [online]. <http://developer.intel.com>. May 2002.

[8] Intel. *Software development for SIMD technology*. [online]. <http://developer.intel.com>. May 2002.

[9] Intel. *IA-32 Intel architecture software developer's manual*. [online]. <http://developer.intel.com>. May 2002.

[10] Martinon, R.F., F. C. R. Fernandes, H. O. Vats, J. Â. C. F. Neri, H. S. Sawant, *Boletim da SAB Vol. 20 no. 1*, pg. 44, 2000.

[11] Sawant, H.S.; Sobral, J.H.A.; Neri, J.A.C.F.; Fernandes, F.C.R.; Cecatto, J.R.; Rosa, R.R., *Adv. Space Res.*, 13, 199, 1993.

[12] Sawant, H.S.; Sobral, J.H.A.; Fernandes, F.C.R.; Cecatto, J.R.; Day, W.R.G.; Neri, J.A.C.F.; Alonso, E.M.B., Moraes, A., *Adv. Space Res.*, 17, 391, 1996.

[13] Sawant, H.S., Subramanian, K.R., Faria, C., Stephany, S., Fernandes, F.C.R., Cecatto, J.R., Rosa, R.R., Alonso, E.M.B., Mesquita, F.P.V., Portezani, V.A., *ASP Conference Series*, 206, 341-346, 2000.

[14] Sawant, H.S., Subramanian, K.R., Sobral, J.H.A., Faria, C., Fernandes, F.C.R., Cecatto, J.R., Rosa, R.R., H. O. Vats, J. A. C. F. Neri, Alonso, E.M.B., Mesquita, F.P.V., Portezani, V.A., A. R. F. Martinon., *Solar Physics*, 2000.