

# DINCON'10

9<sup>th</sup> Brazilian Conference on Dynamics,  
Control and their Applications  
June 07-11, 2010



## UMA ABORDAGEM EM C++ PARA AMBIENTE LINUX DA TÉCNICA DE ANÁLISE DE PADRÕES GRADIENTES

*Cristiano Strieder<sup>1</sup> and Reinaldo R. Rosa<sup>2</sup>*

<sup>1</sup>INPE, São José dos Campos, Brasil, cstrieder@gmail.com

<sup>2</sup>INPE, São José dos Campos, Brasil, reinaldo@lac.inpe.br

**Abstract:** In this paper we introduce a new C++ numerical code for the so-called Gradient Pattern Analysis, which is able to characterize different morphological shapes by means of calculating the object's gradient asymmetry. The application was developed under Linux to run as a command line tool.

**Keywords:** Análise de Padrões Gradientes, C++, Linux

### 1. INTRODUÇÃO

A técnica de Análise de Padrões Gradientes foi apresentada durante a década de 90 na forma de um operador para caracterização da fragmentação da amplitude assimétrica em sistemas estendidos [1, 2]. Desde então o método vem sendo empregado em problemas que envolvem caracterização de assimetrias em diversas áreas. Em problemas de complexidade, a técnica foi empregada para investigar a complexidade estrutural de superfícies desordenadas [3] e também para caracterização de padrões complexos em sistemas espacialmente estendidos [4].

Os problemas abordados geralmente envolvem processamento de séries temporais [5] ou análise de envelopes 2D ou 3D quando é estudada sua evolução temporal [6].

Com o crescente aumento do volume de informações, característico dos problemas de computação contemporâneos, e portanto do tamanho dos dados que precisam ser analisados, é requerida uma abordagem mais robusta do método visando a utilização em conjuntos de dados mais densos. Para se adequar ao cenário atual foi desenvolvido um novo código em linguagem C++ para ambiente Linux. Esta nova ferramenta, a qual chamamos GPA++, emprega os princípios fundamentais da Análise de Padrões Gradientes e introduz novas funções plausíveis a técnica.

O método de Análise de Padrões Gradientes é capaz de caracterizar a formação e evolução de padrões através das correlações espaço-temporais entre as amplitudes das flutuações em pequena e larga escala (gradientes de flutuação de energia).

Padrões assim aparecem em diversos problemas como por exemplo na caracterização da morfologia de galáxias e também podem ser observados em sistemas estendidos como é o caso das espirais geradas pela equação complexa de Ginzburg-Landau e nas reações de Belousov-Zhabotinsky. Em todos estes casos é importante quantificar o grau de complexidade local pra poder caracterizar o padrão espacial e também estudar sua evolução temporal.

Nas sessões que seguem é apresentada a teoria relativa a técnica de Análise de Padrões Gradientes, em seguida é feita uma descrição das particularidades do novo código.

### 2. OBJETIVO

Neste trabalho é apresentada uma abordagem em linguagem C++ desenvolvida para ambiente Linux da técnica de Análise de Padrões Gradientes.

Com relação ao método de Análise de Padrões Gradientes são encontrados na literatura diversas aplicações sendo que estas utilizam soluções baseadas em licenças comerciais como *IDL - Interactive Data Language* e *Matlab*. Apesar de facilitar o desenvolvimento de soluções estes ambientes também trazem diversos problemas relacionados ao desempenho, escalabilidade, portabilidade, etc.

No meio acadêmico é frequente a utilização de propostas existentes aplicadas a novos problemas e para isso é interessante que as soluções desenvolvidas possam ser reaproveitadas. Considerando isso, neste trabalho aborda-se o método de Análise de Padrões Gradientes por meio de uma implementação em C++ desenvolvida para ambiente linux constituindo dessa forma uma solução baseada em software livre. Melhorando assim o desempenho e aumentando a portabilidade que são características importantes em ferramentas de computação científica.

Para validar os resultados encontrados através desta nova implementação são empregados exemplos clássicos encontrados na literatura. Os resultados obtidos com o novo código são apresentados e discutidos.

### 3. ANÁLISE DE PADRÕES GRADIENTES

A Análise de Padrões Gradientes ou simplesmente GPA (*Gradient Pattern Analysis*), como é mais conhecida, é uma metodologia inovadora que caracteriza a formação e evolução de padrões baseada em correlações espaço-temporais entre grandes e pequenas flutuações de amplitude da estrutura representada em um campo gradiente. Abaixo são descritos os princípios desta técnica:

#### 3.1. Fragmentação da Energia Espacial

Devemos considerar sistemas espacialmente estendidos em duas dimensões  $(x,y)$  cuja distribuição de amplitudes da energia é descrita por  $E(x,y)$ , essencialmente uma matriz de amplitudes se as duas dimensões estiverem discretizadas em  $m$  valores cada. Deve-se notar que uma sequência dinâmica de matrizes pode estar relacionada com uma evolução temporal de um envelope  $E(x,y)$  representada pelo conjunto 3D:

$$E(x,y,1), \dots, E(x,y,n), E(x,y,N) \quad (1)$$

onde o índice  $n$  é o tempo  $t$  discretizado em  $N$  instantes.

Para uma dada distribuição espacial de amplitudes,  $E(x,y,n)$ , observada no instante  $n$ , de maneira geral pode-se analisar suas estruturas pelo cálculo de medidas estatísticas na matriz de valores. Um exemplo seria o comprimento da correlação espacial, comumente definida como uma medida da importância estatística de certas regiões em comparação com a média global para todo o sistema.

Em geral, o comprimento de correlação é capaz de caracterizar somente padrões que aparecem na média ao se integrar no tempo todas as matrizes. No entanto, esta medida não é tão robusta para caracterização instantânea de padrões com alto nível de complexidade [2].

É proposto que a complexidade do padrão espacial ou da estrutura, é caracterizada pela sua falta de simetria e pela extensão da fragmentação. Afim de identificar a natureza da fragmentação espacial e sua relação com a simetria presente no sistema, são introduzidas as seguintes definições, de acordo com [2]:

1. O envelope  $E(x,y)$  não está fragmentado se tiver um único extremo simetricamente localizado sobre toda a região. Um exemplo de padrão não fragmentado é apresentado pelo envelope Gaussiano com um único pico, ver Figura 1.
2.  $E(x,y)$  está fragmentada se tiver múltiplos pontos de máximo ou mínimo. A fragmentação é simétrica se houver um extremo com relação ao qual os outros estão distribuídos simetricamente( um exemplo que possui um padrão assim é o envelope de Bessel mostrado na Figura 2). Nos outros casos, onde há uma distribuição randômica de máximos e mínimos, a fragmentação é dita assimétrica( este caso é exemplificado na Figura 3).

#### 3.2. Campo Assimétrico

De acordo com a técnica de Análise de Padrões Gradientes, para caracterizar diferentes padrões com máximos e

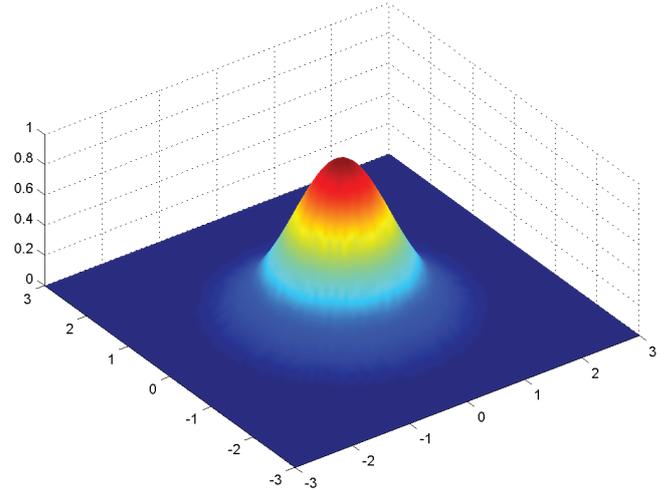


Figura 1 – Distribuição espacial de amplitudes com padrão simétrico (Função Gaussiana bidimensional).

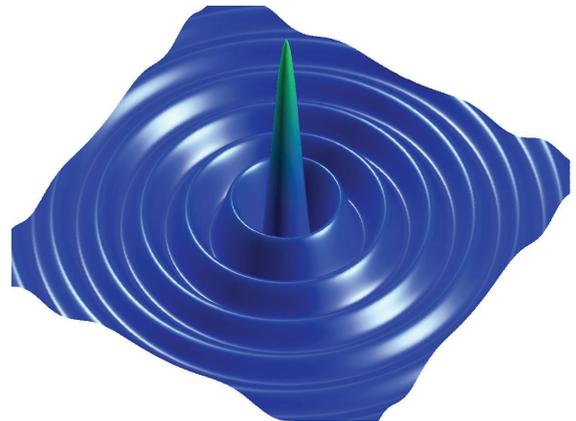


Figura 2 – Distribuição espacial de amplitudes com padrão de fragmentação simétrico (Função de Bessel bidimensional).

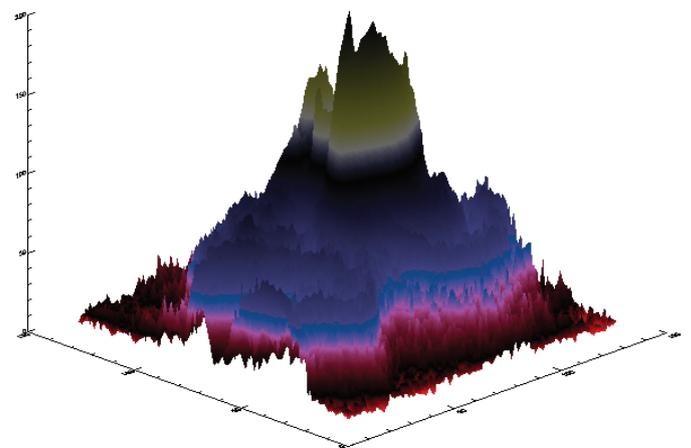


Figura 3 – Distribuição espacial de amplitudes com padrão de fragmentação assimétrico (Plasma de uma região ativa solar observada em raios X moles).

mínimos é usado o campo gradiente  $\nabla E$  da matriz  $E$ . O campo gradiente especifica quantitativamente as variações em um dado ponto e é um bom indicador quantitativo das mudanças na distribuição dos máximos e mínimos, portanto da fragmentação espacial correspondente. Pode-se representar a matriz  $E$  através do seu campo gradiente com  $V$  vetores, denotada matematicamente por  $Grad^V(E)$ , onde cada vetor está associado a um ponto em uma grade bidimensional. Dessa forma, agora podemos pensar que a matriz contém os vetores do campo gradiente em cada ponto da grade no espaço bidimensional. Nesta situação muitos dos vetores possuem a mesma magnitude com uma pequena margem de erro. Dizemos que dois destes vetores que apontam em direções opostas formam um par simétrico. Se apontarem em direções opostas formam um par assimétrico.

Ao removermos cada par de vetores simétricos do campo  $Grad^V(E)$  estaremos gerando um campo com  $N_V$  vetores resultantes, todos vetorialmente assimétricos.

### 3.3. Matrizes Elementares

Vamos fazer agora uma análise aplicada a matrizes elementares. Considerando matrizes de tamanho  $3 \times 3$ , portanto com 9 elementos. Estas matrizes podem ser divididas em dois tipos:

- Matrizes com simetria vetorial total, isto é, uma vez dado o campo gradiente  $Grad^V(E)$ , temos  $N_V = 0$ , como consequência da remoção dos pares de vetores simétricos. Na Figura 4 são mostrados três exemplos de matrizes elementares deste tipo ((a), (b) e (c)). A direita é mostrado o campo gradiente e os valores de  $V$  e  $N_V$ .
- Matrizes nas quais a simetria vetorial total não ocorre. Na Figura 5 são mostrados três exemplos ((a), (b) e (c)) de matrizes elementares deste tipo. Na coluna do meio é mostrado o campo gradiente. Percebe-se que apenas através do parâmetro  $N_V$  não é possível distinguir os exemplos (b) e (c). Isto é, apesar de nestes casos o campo gradiente ser completamente diferente, temos  $N_V = 9$  em ambos os casos.

O parâmetro  $N_V$  sozinho já é considerado como uma medida grosseira da assimetria, mesmo não distinguindo um campo com simetria de fluxo de outro totalmente assimétrico.

### 3.4. Campo de Triangulação

Dada uma distribuição Cartesiana  $P$  com  $p$  pontos, a rede constituída pelas linhas retas que unem todos os pontos  $p$  é uma triangulação. Esta triangulação é composta por uma estrutura de triângulos cujos vértices estão sempre em conexão com os pontos da distribuição. Por esta razão, o número de triângulos além de ser uma função do número de pontos  $p$ , também é função da distribuição destes pontos.

Para diferenciar campos gradientes com simetria de fluxo de outros com assimetria acentuada é utilizada aqui a triangulação de *Delaunay* [7]. A triangulação de *Delaunay* é um tipo clássico de triangulação que maximiza os ângulos inter-

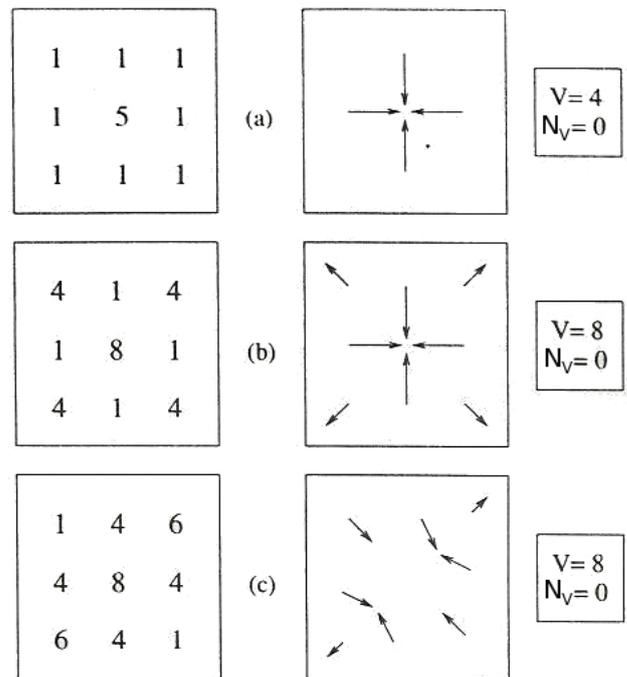


Figura 4 – Exemplos de matrizes elementares com simetria vetorial total. Nos três casos (a), (b) e (c) o número de vetores resultantes é nulo.

nos mínimos de tal forma que cada triângulo é o mais equilateral possível.

Dado um campo gradiente assimétrico  $A(E)$  com  $N_V$  vetores assimétricos. É construído um campo equivalente com  $N_V$  pontos, onde cada ponto corresponde a posição do ponto médio do do vetor. Depois de construir este campo escalar, é traçada a triangulação de Delaunay entre os pontos e desta triangulação é computado o número  $N_L$  de linhas de triangulação.

Considerando os exemplos (b) e (c) da Figura 5, é construído o campo de triangulação e obtido o número de linhas de triangulação  $N_L$ . Na Figura 6 são apresentados os resultados desta operação. O número de linhas de triangulação para o exemplo com simetria de fluxo é  $N_L = 16$ , enquanto que para o exemplo assimétrico  $N_L = 20$ . Podemos ver que através do valor de  $N_L$  podemos distinguir os dois casos, uma vez que  $N_L$  é diretamente proporcional ao nível de complexidade (ou desordem espacial) dos campos gradientes.

### 3.5. O Operador $\mathcal{F}_A$ e o coeficiente de Assimetria Gradiente $G_A$

As operações realizadas em uma matriz  $M$ , introduzidas nas seções anteriores, podem ser agrupadas em um operador que foi chamado de *Operador de Fragmentação Assimétrica*  $\mathcal{F}_A$ . O operador computacional  $\mathcal{F}_A$  opera em uma matriz  $M$  (geralmente visualizada através de um envelope  $E(x, y)$ ), transformando a matriz em um campo de triangulação  $\mathcal{T}_A$  com  $N_V$  pontos conectados por  $N_L$  linhas de triangulação. Com isso foram estabelecidas as bases para formular a definição formal para o conceito de Fragmentação Assimétrica:

Um campo de triangulação  $\mathcal{T}_A$ , com  $N_V$  pontos e  $N_L$  li-

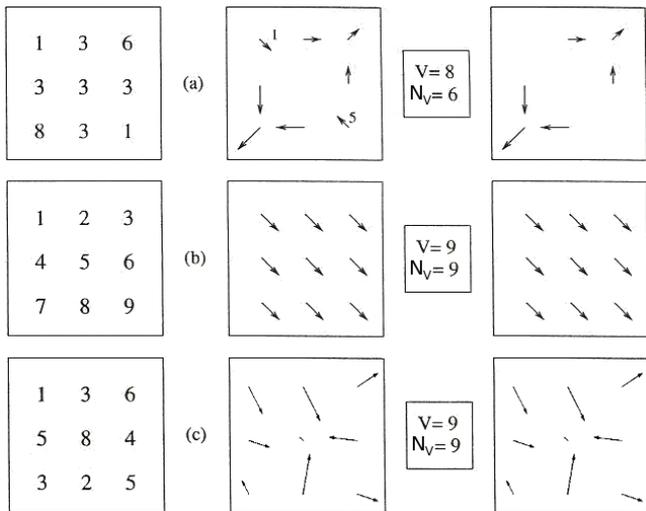


Figura 5 – Matrizes elementares que apresentam assimetria vetorial. (a) seis vetores resultantes, (b) e (c): todos os vetores assimétricos

nhas de triangulação, obtido por meio da operação  $\mathcal{F}_A[M] \mapsto G_A$ . A quantidade de fragmentação assimétrica espacial  $G_A$ , para  $M$  é dada por:

$$G_A \equiv \frac{N_L - N_V}{N_V} \quad (2)$$

#### 4. UM NOVO CÓDIGO PARA ANÁLISE DE PADRÕES GRADIENTES

Um código para este operador foi escrito originalmente em linguagem *IDL - Interactive Data Language* e outras implementações do método seguiram em ambiente Matlab. Esta nova implementação empregou a linguagem C++ utilizando a IDE Netbeans 6.8 com compilador g++ (GCC) 4.3.3 em ambiente Linux Slackware 13.0. Assim formando um ambiente robusto para o desenvolvimento e também oferecendo liberdade para que alterações incrementais da técnica GPA possam ser adicionadas sem depender de licenças comerciais para desenvolvimento.

##### 4.1. A linguagem C++

A biblioteca padrão do C++ incorpora a biblioteca padrão do C com algumas pequenas modificações para trabalhar melhor com as novas funcionalidades criadas pela linguagem. Outra grande parte da biblioteca é composta pela biblioteca padrão de gabaritos (*STL - Standard Template Library*). Ela fornece ferramentas úteis como contêineres (vetores, listas, entre outros), algoritmos (filtragem de elementos de contêiner, busca, ordenação, entre outros) e iteradores (ponteiros inteligentes genéricos para acessar tais contêineres e interligá-los aos algoritmos). Usando gabaritos é possível escrever algoritmos genéricos que funcionam para qualquer contêiner ou sequência definida por iteradores. Tendo em vista que um iterador nada mais é que um ponteiro encapsulado, é possível também utilizar os algoritmos genéricos em vetores C, utilizando-se ponteiros comuns para tal. Na implementação

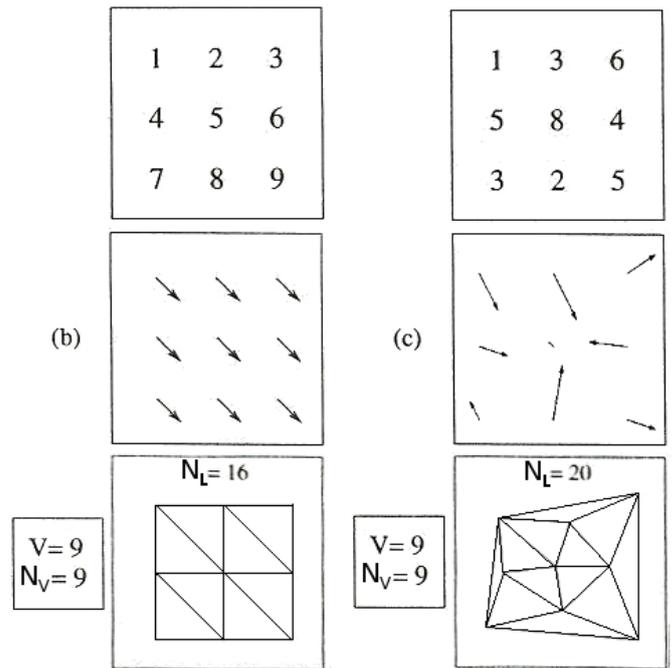


Figura 6 – Campos de triangulação das matrizes elementares

do GPA++ foi utilizada a classe vector da *STL*.

##### 4.2. Algoritmo do programa

Abaixo estão listadas em forma de linguagem natural as principais etapas do programa GPA++:

1. Início do programa
2. Recebe parâmetro: Arquivo ou diretório
3. Carrega opções do arquivo *gpa++.conf*
4. Carrega matriz do arquivo *.dat*
5. Normaliza matriz
6. Salva matriz normalizada em disco
7. Calcula campos gradientes (horizontal e vertical)
8. Salva matrizes gradientes em disco
9. Calcula matriz com o módulo dos vetores
10. Calcula a fase dos vetores e salva em arquivo
11. Prepara arquivos de entrada para Gnuplot
12. Gnuplot gera imagens dos campos gradientes
13. Conta número de vetores não nulos
14. Elimina vetores simétricos seguindo critério
15. Encontra coordenadas dos pontos médios dos vetores
16. Gera imagem do campo assimétrico
17. Prepara dados de entrada para a triangulação

18. Gera triangulação de Delaunay

19. Apresenta os resultados

20. Fim do programa

Através dos passos citados acima, esta nova implementação da técnica GPA possibilita:

- Fazer a análise de um envelope 2D na forma de uma matriz contida em um arquivo ASCII. Também é possível passar como parâmetro o caminho de um diretório que contém vários arquivos, assim é feita uma análise serial de um conjunto grande de dados.
- Normalizar a matriz de entrada pelo critério de amplitude ou pelo valor máximo.
- Utilizar diversos critérios de busca por pares simétricos.
- Especificar a tolerância no módulo e na fase que vai ser considerada para remover eventuais pares de vetores simétricos.
- Triangular os pontos da rede com linhas passando na base, no meio ou na ponta dos vetores.
- Exibir o campo de vetores.
- Exibir o campo de triangulação.

## 5. RESULTADOS

O software desenvolvido foi validado através da aplicação aos exemplos apresentados na sessão 3. Os resultados obtidos foram condizentes com o que é apresentado na literatura.

A Análise de Padrões Gradientes faz uma medida da assimetria bilateral, para verificarmos isso vamos aplicá-la no seguinte conjunto de formas geométricas:

- Retângulo
- Quadrado
- Sinal de soma
- Sinal de multiplicação
- Triângulo

Vão ser seguidas duas etapas: Primeiro são geradas formas geométricas simples que são transcritas em matrizes. GPA++ é aplicado a essa matriz. Em seguida é feita uma pequena deformação em um lado da imagem para distorcer levemente a forma geométrica. GPA++ é aplicado novamente. Com esses resultados comprova-se que a técnica de Análise de Padrões Gradientes faz uma medida da assimetria bilateral - a figura deformada apresenta assimetria bilateral que é identificada pelo GPA.



(a)

(b)

0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 0 0 0	0 0 1 1 1 1 0 0 0
0 0 1 1 1 1 0 0 0	0 0 1 1 1 1 0 0 0
0 0 1 0 0 1 0 0 0	0 0 1 0 0 1 0 0 0
0 0 1 0 0 1 0 0 0	0 0 1 0 1 0 0 0 0
0 0 1 0 0 1 0 0 0	0 0 1 0 0 1 0 0 0
0 0 1 1 1 1 0 0 0	0 0 1 1 1 1 0 0 0
0 0 1 1 1 1 0 0 0	0 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0

(c)

(d)

Figura 7 – O retângulo, mostrado na figura a direita com o seu lado direito deformado. Abaixo estão as matrizes correspondentes.

### 5.1. Dados

São geradas figuras transcritas em matrizes em tamanho  $9 \times 9$ . Para isso, inicialmente é gerada a figura de um retângulo e em seguida a mesma é apresentada com uma deformação em um dos lados. Na Figura 7 está sendo mostrado este exemplo.

Nas Figuras 8, 9, 10 e 11 são apresentadas outras formas geométricas investigadas.

O próximo passo é rodar o GPA++ sobre uma matriz que represente o dado(as formas geométricas neste caso). No caso de imagens, geralmente a Análise do Padrão Gradiente não é feita diretamente na imagem da qual queremos obter o coeficiente de fragmentação. Como sabemos, em uma ima-



(a)

(b)

0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 0	0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 0	0 1 1 1 1 1 1 1 0
0 1 1 0 0 0 1 1 0	0 1 1 0 0 0 1 0 0
0 1 1 0 0 0 1 1 0	0 1 1 0 0 1 0 0 0
0 1 1 0 0 0 1 1 0	0 1 1 0 0 0 1 0 0
0 1 1 1 1 1 1 1 0	0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 0	0 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0

(c)

(d)

Figura 8 – O quadrado, mostrado na figura a direita com o seu lado direito deformado. Abaixo estão as matrizes correspondentes.



(a)

(b)

```

0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 0
0 0 0 1 1 1 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0

```

(c)

```

0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 1 1 1 1 1 1 1 0 0
0 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0

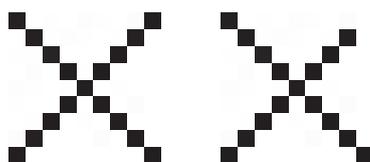
```

(d)

Figura 9 – Sinal de soma, mostrado na figura a direita com o seu lado direito deformado. Abaixo estão as matrizes correspondentes.

gem os pixels de maior brilho são descritos por valores mais altos, no domínio [0,255] para uma imagem de 8 bits. E ainda, no caso de uma imagem RGB, temos três matrizes que juntas constituem a imagem mostrada. Este não é o caso das matrizes que aparecem nos itens (b) e (c) da Figura 7, que em vez de 1 deveriam assumir valor 0 nas regiões escuras, caso se trata-se da própria matriz da imagem em tons de cinza.

Então, de alguma forma deve ser obtida uma matriz que representa o dado ( a imagem ). Aqui é aplicada uma operação de *thresholding* sobre a figura a fim de obter uma grade de valores onde o GPA++ se aplica. O código fonte da função de *thresholding* é mostrado no Programa 1. As matrizes obtidas são mostradas nos itens (c) e (d) da Figura 7.



(a)

(b)

```

1 0 0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0 1 0
0 0 1 0 0 0 1 0 0 0
0 0 0 1 0 1 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 1 0 1 0 0 0 0
0 0 1 0 0 0 1 0 0 0
0 1 0 0 0 0 0 0 1 0
1 0 0 0 0 0 0 0 0 1

```

(c)

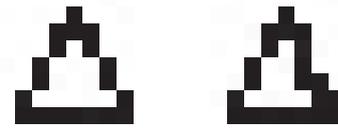
```

1 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 1 0
0 0 1 0 0 0 1 0 0 0
0 0 0 1 0 1 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 1 0 1 0 0 0 0
0 0 1 0 0 0 1 0 0 0
0 1 0 0 0 0 0 0 1 0
1 0 0 0 0 0 0 0 0 1

```

(d)

Figura 10 – Sinal de multiplicação, mostrado na figura a direita com o seu lado direito deformado. Abaixo estão as matrizes correspondentes.



(a)

(b)

```

0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 1 0 1 0 0 0 0
0 0 1 0 0 0 1 0 0 0
0 0 1 0 0 0 1 0 0 0
0 1 0 0 0 0 0 1 0 0
0 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0

```

(c)

```

0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 1 1 1 0 0 0 0
0 0 0 1 0 1 0 0 0 0
0 0 1 0 0 0 1 0 0 0
0 0 1 0 0 0 1 0 0 0
0 1 0 0 0 0 0 1 0 0
0 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0

```

(d)

Figura 11 – Triângulo, mostrado na figura a direita com o seu lado direito deformado. Abaixo estão as matrizes correspondentes.

## 5.2. Obtendo o Coeficiente de Assimetria $G_A$

Utilizando as matrizes descritas acima como entrada para o programa GPA++ obtemos os resultados apresentados na Tabela 1. Percebe-se em todos os casos que ao ser introduzida uma deformação na imagem, ou seja, quando aumenta a assimetria bilateral, esta informação é percebida pela medida do coeficiente de assimetria  $G_A$ .

## 5.3. Comportamento de $G_A$ para matrizes maiores

Foram geradas imagens em tamanho  $64 \times 64$  das figuras geométricas mostradas nas Figura 7, 8, 9, 10, 11. Em seguida foram aplicadas as mesmas operações citadas anteriormente a fim de obter o coeficiente de assimetria  $G_A$ . As imagens não são mostradas aqui pois representam as mesmas imagens, o que muda é apenas o tamanho da matriz. Os resultados obtidos são apresentados na Tabela 2.

Seria razoável esperar que, em geral, o valor de  $G_A$  aumentasse para matrizes maiores pois  $(N_L - N_V)/(N_V)$  aumenta com  $N_V$  de acordo com [2]. No entanto foram obser-

Tabela 1 – Medidas do coeficiente de assimetria obtido com GPA em matrizes de tamanho  $9 \times 9$ .

Matriz	V	$N_V$	$N_L$	$G_A$
Retângulo	44	0	0	0
Retângulo deformado	44	4	6	0.50
Quadrado	72	0	0	0
Quadrado deformado	68	10	18	0.80
Sinal de Soma	44	0	0	0
Soma deformado	40	6	9	0.50
Sinal de Multiplicação	32	0	0	0
Multiplicação deformado	31	5	7	0.40
Triângulo	46	12	22	0.83
Triângulo deformado	46	14	29	1.07

```

I = imread(png_name);

sz = size(I,1);

for i = 1 : sz
    for j = 1 : sz
        pixel = I(i,j);
        if(pixel==0)
            MATRIX(i,j) = 4;
        else
            MATRIX(i,j) = 0;
        end
    end
end

fid = fopen(dat_name, 'w');
for i = 1 : sz
    for j = 1 : sz
        data = MATRIX(i,j);
        if(j<sz)
            fprintf(fid, '%d ', data);
        else
            fprintf(fid, '%d', data);
        end
    end
    if(i<sz)
        fprintf(fid, '\n', data);
    end
end
fclose(fid);

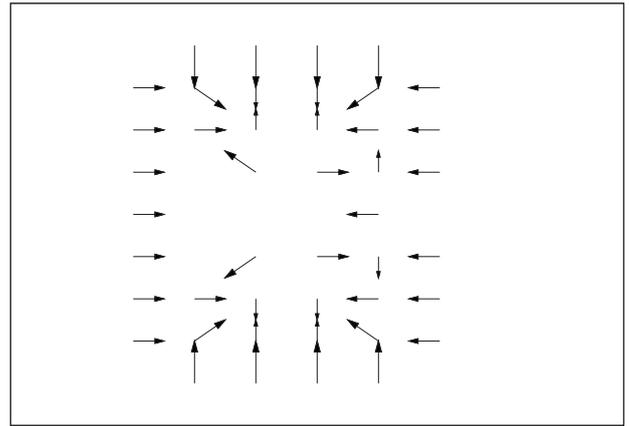
```

**Programa 1:** Função de *thresholding*.

vadas situações particulares: O coeficiente  $G_A$  do retângulo deformado  $64 \times 64$  é nulo apesar de, para a matriz  $9 \times 9$ , não ser assim. Isto ocorre pois ao descrever a região deformada em mais detalhes ( através de uma matriz maior ) a antiga irregularidade acaba mostrando certo grau de regularidade. A região que era descrita por alguns poucos vetores assimétricos (Figura 12) agora é descrita por vários pares de vetores simétricos. E, de acordo com o algoritmo da técnica

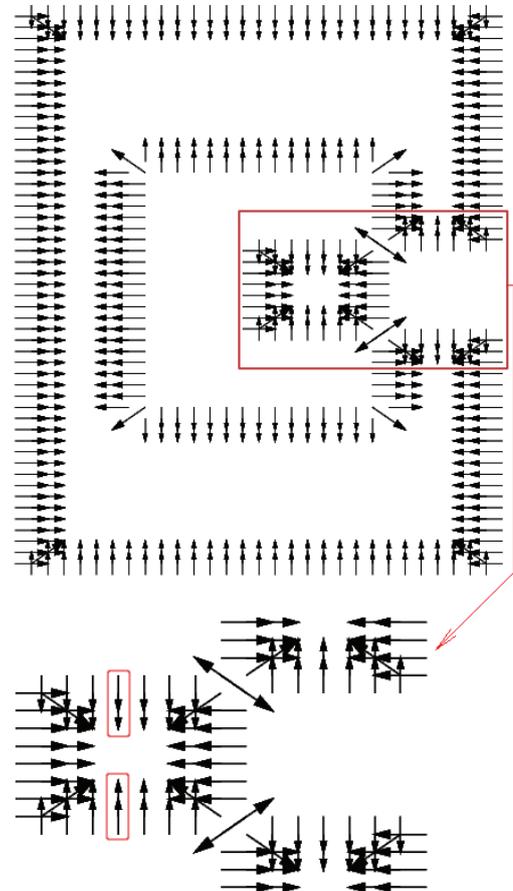
**Tabela 2 – Medidas do coeficiente de assimetria obtido com GPA em matrizes de tamanho  $64 \times 64$ .**

Matriz	V	$N_V$	$N_L$	$G_A$
Retângulo	456	0	0	0
Retângulo deformado	504	0	0	0
Quadrado	568	0	0	0
Quadrado deformado	636	8	15	0.87
Sinal de Soma	388	0	0	0
Soma deformado	398	20	45	1.25
Sinal de Multiplicação	760	0	0	0
Multiplicação deformado	735	27	72	1.66
Triângulo	624	8	14	0.75
Triângulo deformado	602	14	29	1.07



**Figura 12 – Campo de vetores do retângulo deformado em tamanho  $9 \times 9$**

de GPA, a simetria agora evidente é removida. O que acaba resultando em um coeficiente de assimetria  $G_A = 0$ . Este comportamento pode ser facilmente entendido se olharmos para o campo de vetores da matriz  $64 \times 64$  apresentado na Figura 13.



**Figura 13 – Detalhe da deformação do retângulo na matriz  $64 \times 64$ .**

## 6. DISCUSSÃO

Ao utilizar a técnica GPA existem diversas opções que podem ser fixadas. Estas considerações tem influência fundamental sobre o resultado devido a sensibilidade intrínseca do método. Por isso ao ser apresentada a medida do coeficiente de assimetria  $G_A$  é importante explicitar:

- Se a matriz foi normalizada antes de serem obtidos os campos gradientes e qual o critério de normalização foi adotado.
- Quais os valores de  $\tau$  e  $\delta$  que foram considerados.
- Se a triangulação foi feita no meio ou na ponta do vetor.
- Qual estratégia foi utilizada para busca dos pares de vetores simétricos.

A saber,  $\tau$  é a tolerância na fase dos vetores considerada ao serem comparados dois vetores. E  $\delta$  é a tolerância na fase dos mesmos.

Neste trabalho, a matriz foi normalizada pelo critério de máximo. Foi utilizada tolerância nula para  $\tau$  e  $\delta$ , ou seja,  $\tau = 0$  e  $\delta = 0$ . A triangulação foi feita no meio do vetor. Foi utilizada a estratégia clássica para busca dos pares de vetores simétricos.

### 6.1. Estratégias de remoção dos pares de vetores

A triangulação de Delaunay empregada e, consequentemente, o número de linhas de triangulação, depende da posição dos pontos da rede. A posição dos pontos da rede é determinada pela localização dos vetores remanescentes da etapa de remoção dos pares de vetores simétricos. Não só por isso, a posição destes pontos também depende da estratégia de busca dos pares simétricos. Abaixo são descritas as estratégias de busca que o código GPA++ permite utilizar:

- Considerando simetria axial (vertical): Cada vetor é comparado apenas com o seu correspondente em uma posição simétrica ao eixo que passa pelo centro verticalmente. Neste caso cada vetor é checado apenas uma vez, o que garante melhor desempenho.
- Considerando simetria central: Cada vetor é comparado com o seu correspondente que desta vez ocupa uma posição simétrica em relação ao centro. Neste caso também cada vetor é testado apenas um vez.
- Método original: Cada vetor é comparado com o seu vizinho a fim de encontrar o seu par simétrico. Desta maneira, cada vetor remanescente é comparado com todos os outros vetores remanescentes. Cada vetor pode ser usado na comparação várias vezes.
- Alternativa ao método original: Da mesma forma que no método original, neste, cada vetor é checado contra todos os outros vetores remanescentes até ser verificado o par simétrico. A diferença em relação ao método original é que desta vez o montante de vetores remanescentes tende a se concentrar no centro da matriz pois a busca se inicia elegendo um voluntário de cada extremo da matriz.

## 7. CONCLUSÃO

O novo código para o método de Análise de Padrões Gradientes apresentado neste artigo constitui-se de uma ferramenta que facilita a experimentação da técnica GPA pois não requer a instalação de ambientes como IDL ou MATLAB. Assim novas aplicações podem ser testadas de maneira rápida e direta utilizando-se este aplicativo, basta possuir o dado para análise na forma de um arquivo ASCII. Para os casos particulares, quando for conveniente alguma modificação no código fonte, essa alteração poderá ser feita sem a dependência de licenças comerciais.

O código fonte da aplicação GPA++ compreende milhares de linhas de código e por isso não é apresentado aqui. Mas o mesmo pode ser obtido através de uma solicitação ao autor.

## AGRADECIMENTOS

Agradecemos ao Instituto Nacional de Pesquisas Espaciais - INPE e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES pelo suporte financeiro.

## Referências

- [1] [http://en.wikipedia.org/wiki/Gradient\\_pattern\\_analysis](http://en.wikipedia.org/wiki/Gradient_pattern_analysis), Acessado em Maio, 2010.
- [2] R. R. Rosa, A. S. Sharma, J. A. Valdivia, "Characterization of Asymmetric Fragmentation Patterns in Spatially Extended Systems," International Journal of Modern Physics C, Vol. 10, No. 1 (1999) 147-163
- [3] R. R. Rosa and M. P. M. A. Baroni and G. T. Zaniboni and A. F. Silva and L. S. Roman and J. Pontes and M. J. A. Bolzan, "Structural complexity of disordered surfaces: Analyzing the porous silicon SFM patterns," Physica. A, Vol. 386, No. 1 (2007) 666-673
- [4] R.R. Rosa C. Rodrigues Neto, F.M. Ramos, A.S. Sharma, J.A. Valdivia, "Computational operators for dynamical complex pattern recognition," in EPS-Modelling Collective Phenomena in Complex Systems, V.22F, Pub. by European Physical Society, pp. 304-305, 1998.
- [5] R. R. Rosa, M. Karlicky, H. S. Sawant, N. L. Vijaykumar, R. A. Sych, "Gradient Pattern Analysis of Short Solar Radio Bursts," Advances in Space Research 42, pp. 844-851, 2008.
- [6] F. M. Ramos, R. R. Rosa, C. Rodrigues-Neto and A. Zanandrea, "Generalized complex entropic form for gradient pattern analysis of spatio-temporal dynamics," Physica A, vol. 283, pp. 171-174, 2000.
- [7] H. Adelsbrunner, "Handbook of Convex Geometry," Elsevier Science Publishers, 1993