



Ministério da
Ciência e Tecnologia



INPE-16593-RPA/828

AGRUPAMENTO ESPAÇO-TEMPORAL DE DESCARGAS ATMOSFÉRICAS USANDO ST-GRID

Cesar Strauss

Relatório da disciplina Princípios e Aplicações de Mineração de Dados (CAP-359)
ministrada pelo Dr. Rafael Santos.

Registro do documento original:

<<http://urlib.net/sid sid.inpe.br/mtc-m18@80/2009/10.01.19.38>>

INPE
São José dos Campos
2009

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3945-6911/6923

Fax: (012) 3945-6919

E-mail: pubtc@sid.inpe.br

CONSELHO DE EDITORAÇÃO:

Presidente:

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Membros:

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Haroldo Fraga de Campos Velho - Centro de Tecnologias Especiais (CTE)

Dr^a Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Dr. Ralf Gielow - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr. Wilson Yamaguti - Coordenação Engenharia e Tecnologia Espacial (ETE)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Jefferson Andrade Ancelmo - Serviço de Informação e Documentação (SID)

Simone A. Del-Ducca Barbedo - Serviço de Informação e Documentação (SID)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Marilúcia Santos Melo Cid - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Viveca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)



Ministério da
Ciência e Tecnologia



INPE-16593-RPA/828

AGRUPAMENTO ESPAÇO-TEMPORAL DE DESCARGAS ATMOSFÉRICAS USANDO ST-GRID

Cesar Strauss

Relatório da disciplina Princípios e Aplicações de Mineração de Dados (CAP-359)
ministrada pelo Dr. Rafael Santos.

Registro do documento original:

<<http://urlib.net/sid sid.inpe.br/mtc-m18@80/2009/10.01.19.38>>

INPE
São José dos Campos
2009

Dados Internacionais de Catalogação na Publicação (CIP)

Dantas, Nilton Souza.

D235Si Agrupamento espaço-temporal de descargas atmosféricas usando ST-GRID / Cesar Strauss. – São José dos Campos : INPE, 2009.

144 p. ; (INPE-16593-RPA/828)

Tese (Doutorado em Engenharia e Tecnologia Espaciais/Ciência e Tecnologia de Materiais e Sensores) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2009.

Orientadores : Drs. Erasmo Assumpção de Andrada e Silva, Antônio Ferreira da Silva, e Clas Persson.

1. Simulação computacional. 2. Teoria de funcional da densidade. 3. Novos materiais. 4. Estrutura Eletrônica. 5. Propriedades ópticas. I.Título.

CDU 538.915

Copyright © 2009 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, reprográfico, de microfilmagem ou outros, sem a permissão escrita do INPE, com exceção de qualquer material fornecido especificamente com o propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2009 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, microfilming, or otherwise, without written permission from INPE, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

RESUMO

Aplicou-se o método ST-GRID a dados de descargas elétricas atmosféricas com o objetivo de visualizar a evolução de sistemas de atividade elétrica ao longo do tempo e do espaço.

SPATIO-TEMPORAL CLUSTERING OF ATMOSPHERIC DISCHARGES USING ST-GRID

ABSTRACT

The ST-GRID method was applied to electrical atmospheric discharges with the goal of visualizing the evolution of systems of electrical activity.

SUMÁRIO

| | <u>Pág.</u> |
|---|-------------|
| 1 Introdução | 5 |
| 2 Material e Métodos | 5 |
| 2.1 Descrição dos dados | 5 |
| 2.2 Visualização inicial | 5 |
| 2.3 O método ST-GRID | 6 |
| 3 Conclusão | 7 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 15 |
| A ANEXO A - CÓDIGO FONTE | 16 |

1 Introdução

Núcleos convectivos são formações de nuvens cumulus nimbus cujo nascimento, evolução e dissipação é de grande interesse por parte dos meteorologistas. Segundo [Politi \(2005\)](#) e [Politi et al. \(2006\)](#), a partir da estimação da densidade de descargas elétricas nuvem-solo, esse fenômeno pode chegar a ser visualizado de maneira conveniente.

Para melhor acompanhar a evolução temporal de sistemas de atividade elétrica, proponho utilizar o método ST-GRID ([WANG et al., 2006](#)) para agrupar as descargas no espaço e no tempo em clusters. Este método foi escolhido pela simplicidade, baixo custo computacional e por determinar automaticamente o número de clusters.

2 Material e Métodos

2.1 Descrição dos dados

Os dados utilizados neste trabalho foram coletados pela rede RINDAT. Para validar o método, escolhi intervalos de 24 horas contínuas em dois dias diferentes (16/03/2007 e 01/10/2008) com cobertura nacional. Os dados estão organizados em tabelas no padrão UALF (Universal ASCII Lightning Format), que consiste em arquivos ASCII formatados em colunas de atributos de tamanho fixo separados por espaços em branco. Para esta análise, os atributos mais relevantes são latitude, longitude e o instante de ocorrência. As coordenadas e o tempo têm resolução de fração de grau e de nanosegundo respectivamente.

Algumas características desse tipo de dado são:

- a) O número de clusters não é previamente conhecido.
- b) Cada evento têm duração efêmera, ou seja, as descargas individuais não executam uma trajetória no espaço-tempo. Esse comportamento só é obtido quando observamos um grande número de descargas.

2.2 Visualização inicial

Inicialmente, obtive uma sequência de imagens, de hora em hora, onde são assinaladas as descargas individuais que ocorreram em cada intervalo. Foi utilizado o seguinte algoritmo:

- a) Leitura do arquivo de dados.
- b) Seleção das descargas na região de interesse.

- c) Segmentação das descargas no tempo e impressão do resultado.

Foi possível observar visualmente a presença de clusters em evolução nas imagens, o que incentivou a busca de métodos para identificar e acompanhar esses clusters.

2.3 O método ST-GRID

O método ST-GRID, descrito em [Wang et al. \(2006\)](#), consiste em agrupar as descargas em células numa grade e determinar os clusters a partir dos componentes conectados.

O método apresenta os seguintes parâmetros:

- a) Intervalo espacial e temporal da grade.
- b) Número mínimo de descargas em cada célula.

O método implementado consiste em:

- a) Visualizar as descargas na forma de uma sequência de imagens.
- b) Contar o número de descargas em cada célula.
- c) Determinar as células densas, que contém descargas acima do número mínimo.
- d) Determinar os componentes conectados.
- e) Visualizar a grade na forma de uma sequência de imagens.
- f) Classificar as descargas de acordo com os clusters encontrados.
- g) Visualizar as descargas classificadas na forma de uma sequência de imagens.

Considera-se que células vizinhas fazem parte do mesmo cluster. A vantagem de distribuir os eventos em células de grade é o baixo custo computacional para determinar a vizinhança de cada célula. No caso, a vizinhança é formada por 6 células, nas direções norte, sul, leste e oeste, além da célula anterior e posterior no tempo.

Um motivo para eliminar células pouco densas é tentar evitar que o ruído acabe formando uma ponte entre dois agrupamentos, levando à formação de um único cluster.

Para determinar os componentes conectados, repetiu-se iterativamente o seguinte algoritmo:

- a) Procurar uma célula densa que ainda não faz parte de nenhum cluster.
- b) Criar um novo cluster.
- c) Adicionar esta célula à lista de células a serem visitadas.
- d) Enquanto a lista não estiver vazia:
 - Retirar uma célula da lista
 - Adicionar a célula ao cluster corrente
 - Procurar células densas na vizinhança
 - Adicionar essas células à lista.

Para classificar as descargas, identifiquei a célula de grade em que a descarga ocorreu e atribui a classificação da célula à descarga.

Para implementar, escolhi utilizar a ferramenta Octave, disponível livremente, devido à facilidade de manipulação de matrizes e de visualização. O código fonte encontra-se em anexo.

Executei o método com os dois conjuntos de dados (2007 e 2008), com os seguintes parâmetros:

- Grade de 100 x 100 na dimensão espacial.
- Intervalo de 1 hora na dimensão temporal.
- Mínimo de 4 descargas em cada célula densa.

As Figuras 2.1 a 2.6 apresentam os resultados obtidos. Cada cluster recebeu uma cor diferente, determinada completamente ao acaso. Não há informação adicional ligada à cor, como distância ou tempo. Numa dada figura, quando dois agrupamentos separados recebem a mesma cor, significa que eles vão se fundir no futuro ou são fragmentos que estavam juntos no passado. As descargas que não foram classificadas em nenhum cluster receberam a cor cinza. Por outro lado, devido ao número de clusters ser relativamente alto, existe chance de haver dois clusters distintos com cores parecidas. Para tentar minimizar esse efeito, adiciona-se um pouco de branco à cada cor, também de forma aleatória.

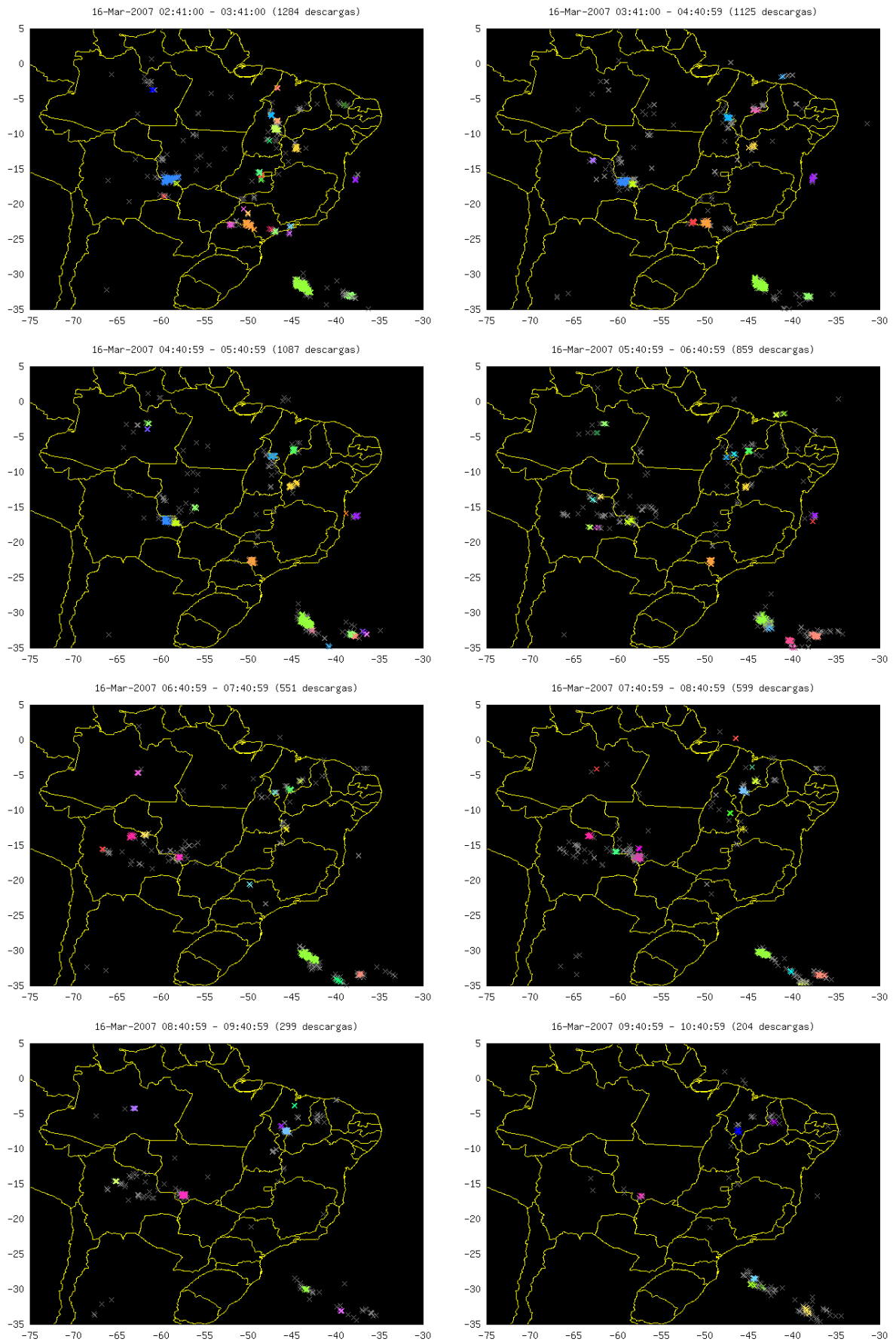


Figura 2.1 - Resultado do método. Dados de 16/03/2007, 02:41 - 10:41

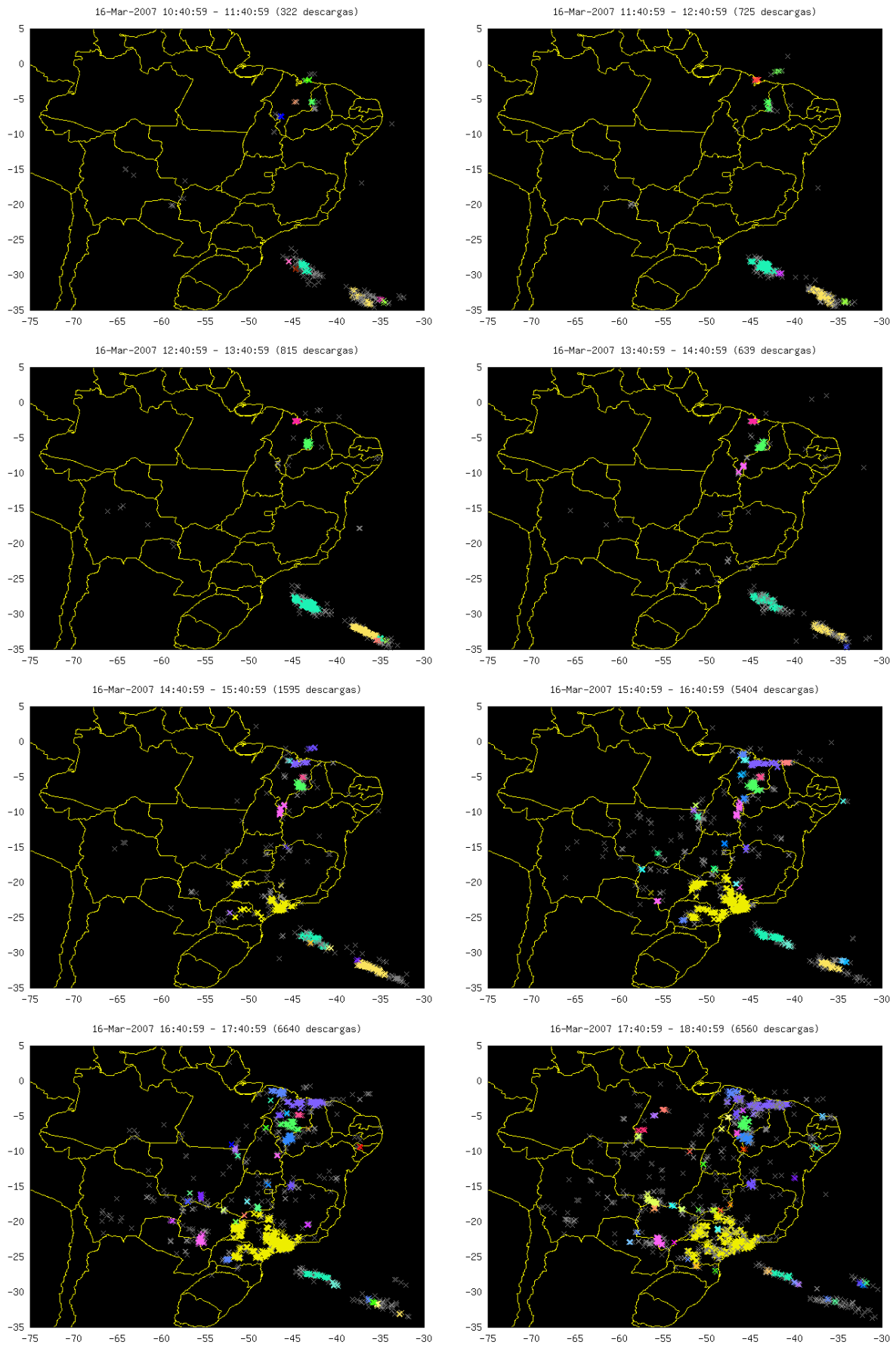


Figura 2.2 - Resultado do método. Dados de 16/03/2007, 10:41 - 18:41

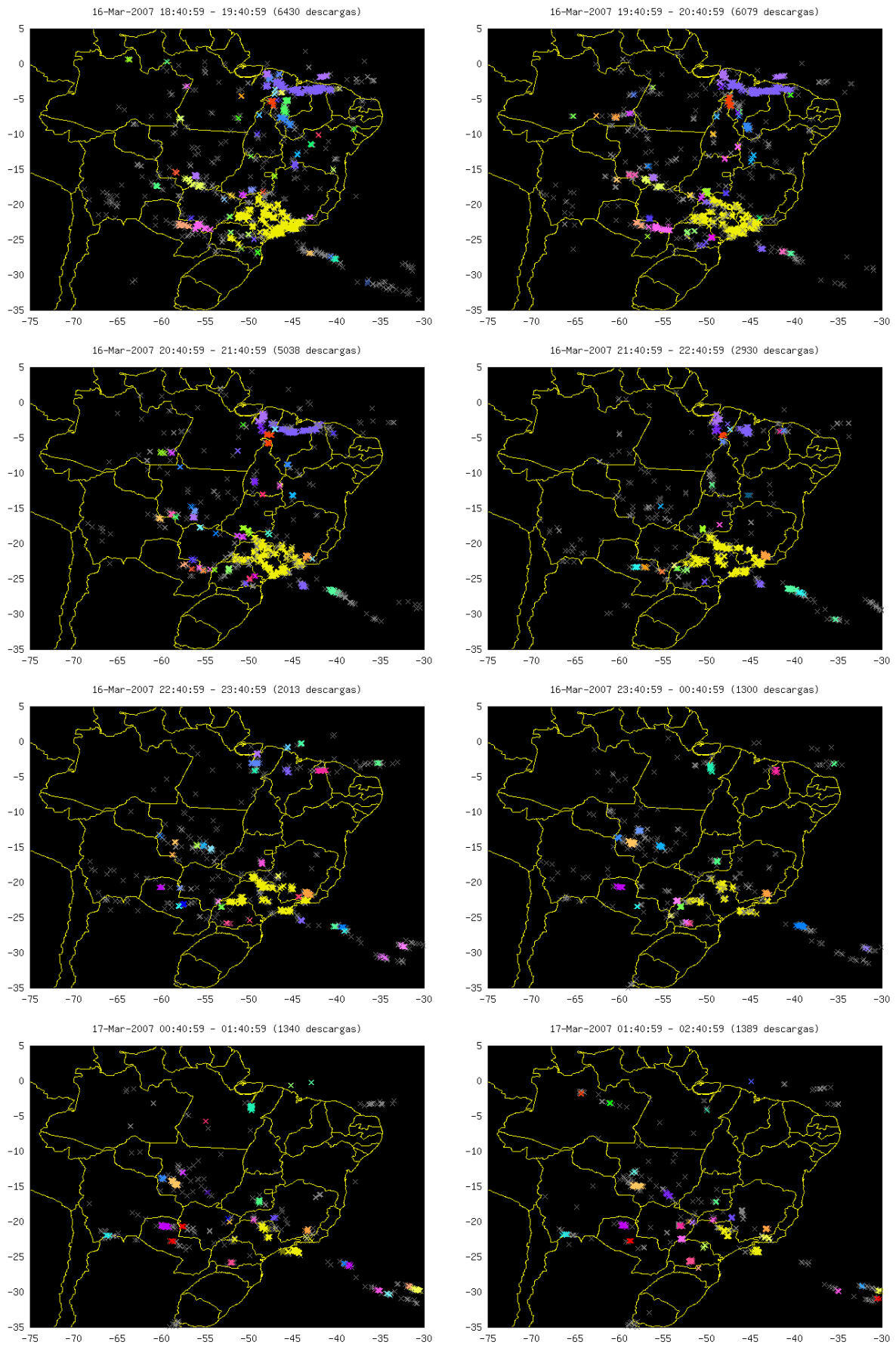


Figura 2.3 - Resultado do método. Dados de 16/03/2007, 18:41 - 02:41

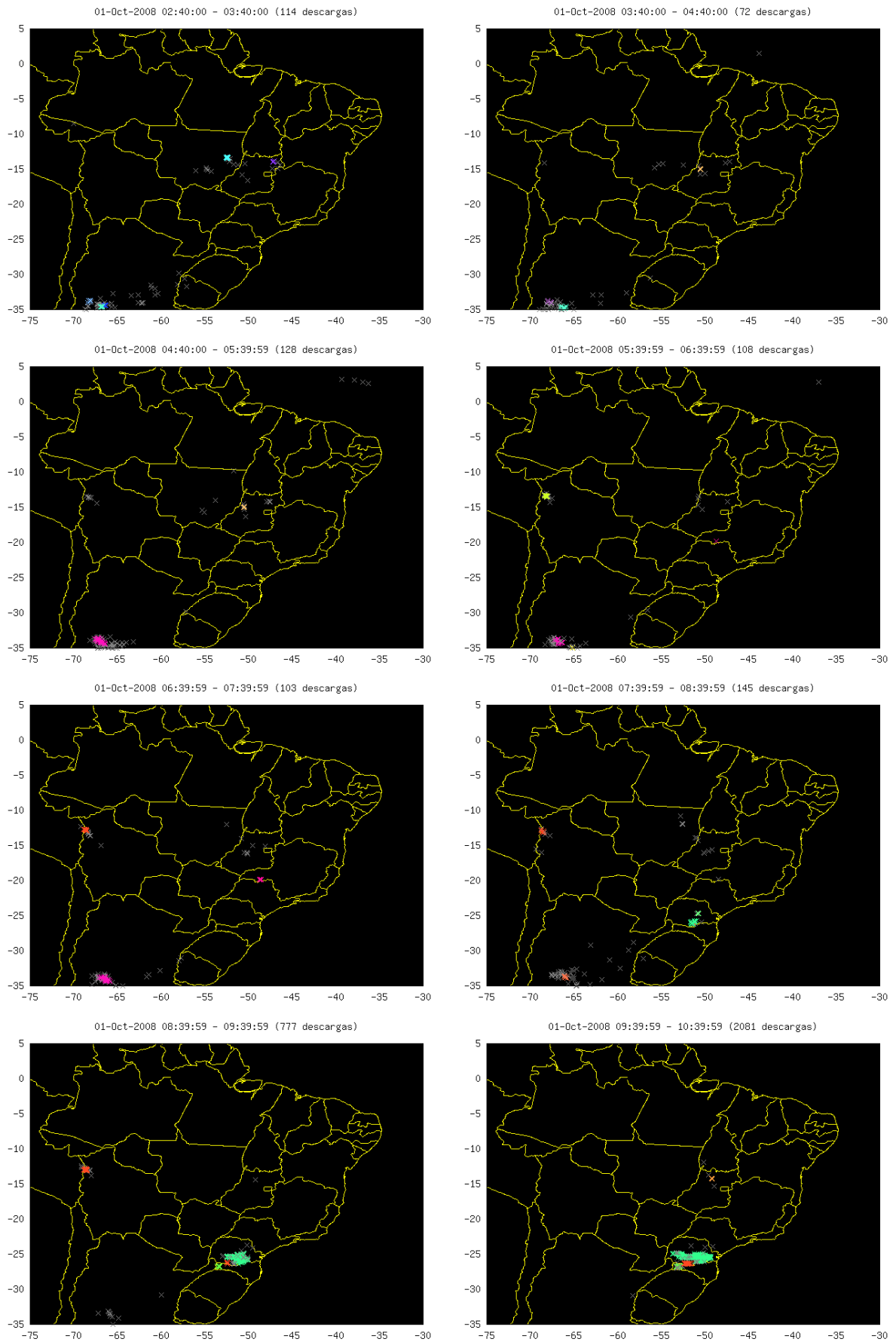


Figura 2.4 - Resultado do método. Dados de 01/10/2008, 02:41 - 10:41

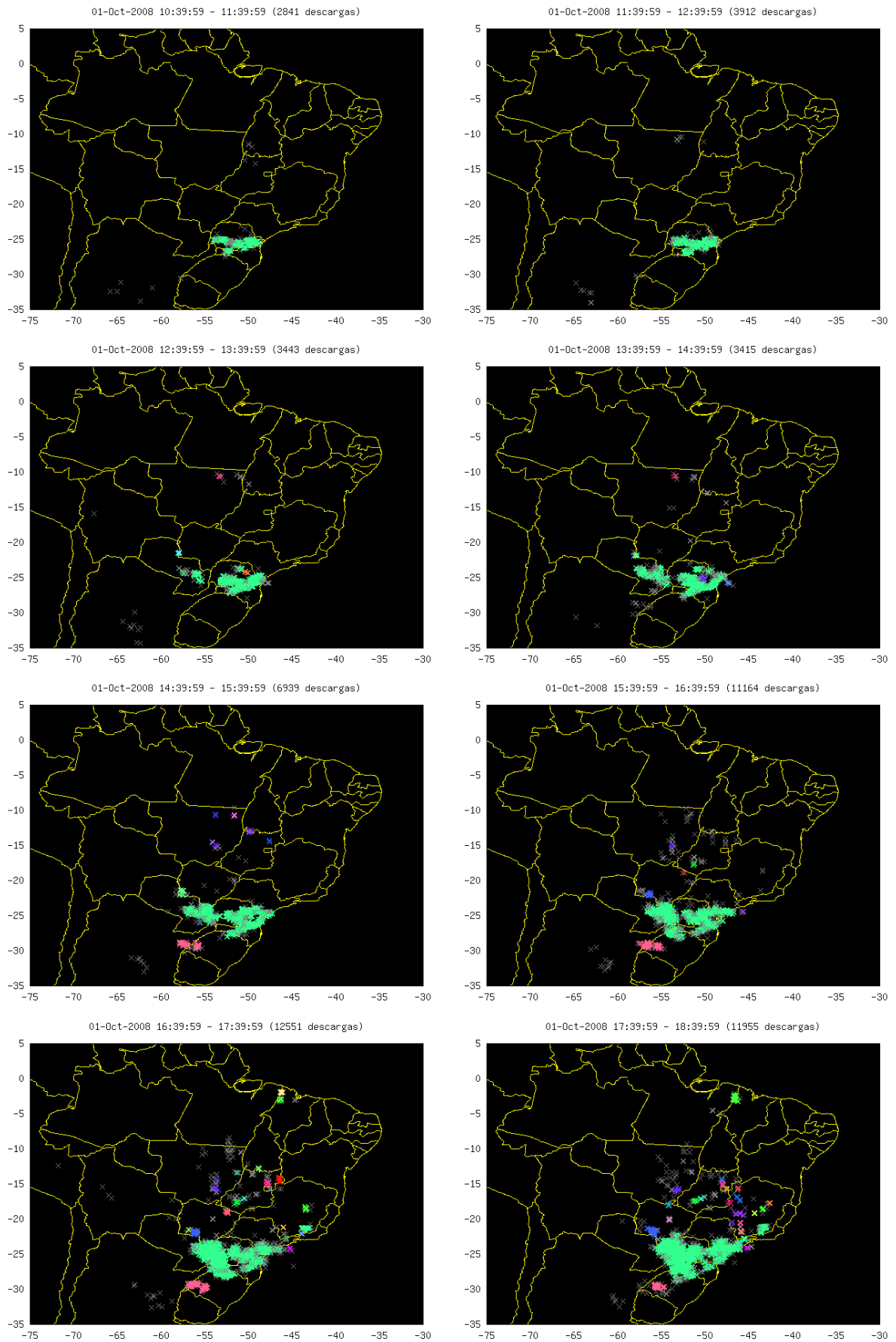


Figura 2.5 - Resultado do método. Dados de 01/10/2008, 10:41 - 18:41

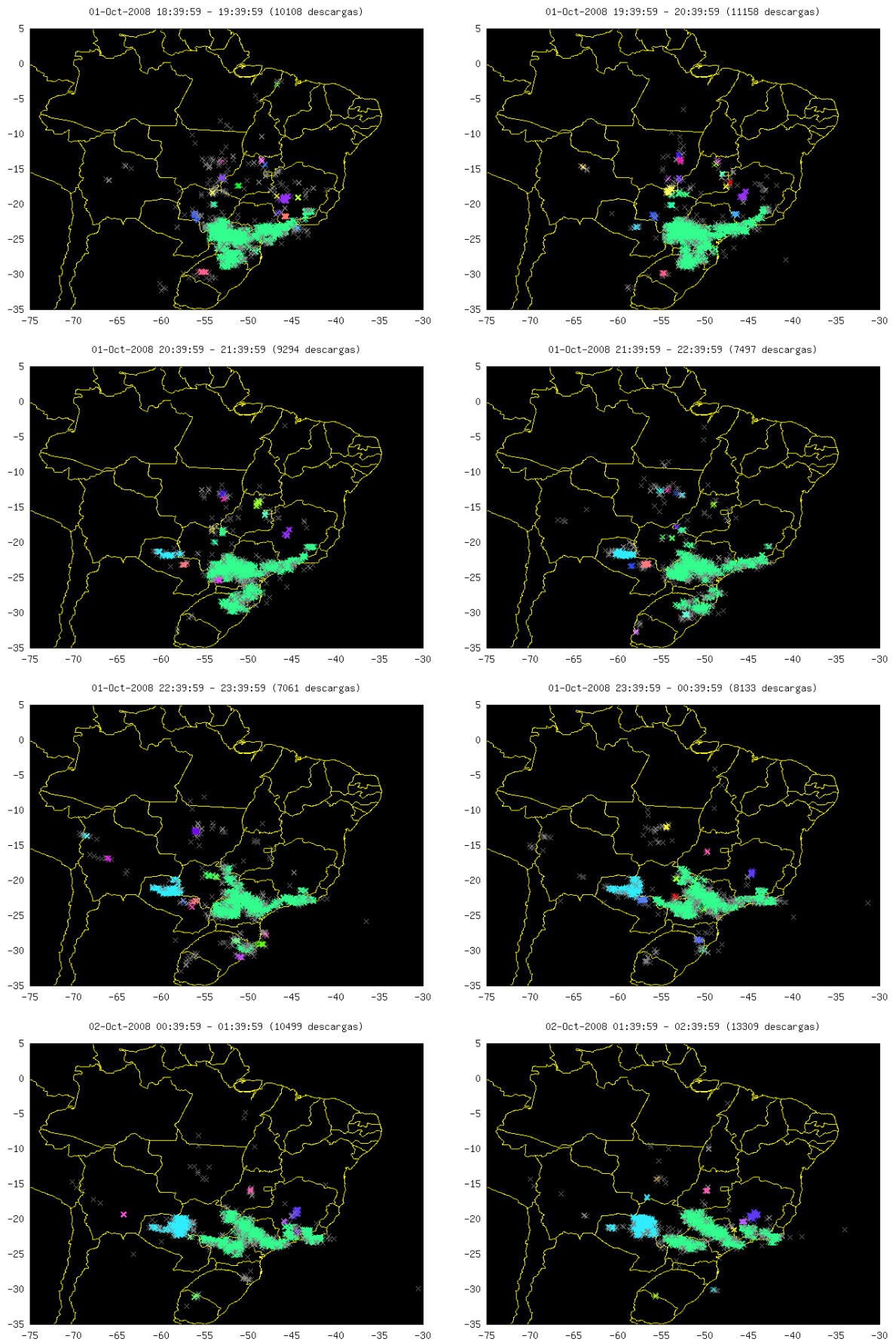


Figura 2.6 - Resultado do método. Dados de 01/10/2008, 18:41 - 02:41

3 Conclusão

O método consegue identificar os clusters principais, mas podemos observar descargas sem identificação dentro dos clusters. Isso se deve provavelmente à regiões de menor densidade dentro do cluster.

Não é necessário fornecer o número de clusters como parâmetro desse método. Por outro lado, foram identificados vários clusters muito pequenos (poucas células de grade) e de curta duração. Uma saída pode ser selecionar apenas os clusters maiores.

Mesmo se um cluster se fragmenta, o método consegue acompanhar os fragmentos e os identifica com o cluster original. O mesmo se aplica a fragmentos que acabam se fundindo. Isso acontece porque o algoritmo de vizinhança pode, partindo de um fragmento, alcançar todos os fragmentos conectados no espaço e no tempo.

REFERÊNCIAS BIBLIOGRÁFICAS

POLITI, J. **Implementação de uma metodologia para mineração de dados aplicada ao estudo de núcleos convectivos**. Dissertação (Mestrado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2005–02–03 2005. [5](#)

POLITI, J.; STEPHANY, S.; DOMINGUES, M. O.; JUNIOR, O. M. Mineração de dados meteorológicos associados a atividade convectiva empregando dados de descargas elétricas atmosféricas. **Revista Brasileira de Meteorologia**, v. 21, n. 2, p. 232–244, agosto 2006. [5](#)

WANG, M.; WANG, A.; LI, A. Mining spatial-temporal clusters from geo-databases. In: LI, X.; ZAÏANE, O. R.; LI, Z. (Ed.). **Advanced Data Mining and Applications**. [S.l.]: Springer, 2006. (Lecture Notes in Computer Science, v. 4093), p. 263–270. ISBN 3-540-37025-0. [5](#), [6](#)

A ANEXO A - CÓDIGO FONTE

```
%  
% Arquivo: stgrid.m  
%  
  
% Parâmetros do algoritmo  
  
% Definições de grade  
% Limite temporal da grade  
Lb.ti=datenum([2007 03 16 02 41 00]);  
Lb.tf=datenum([2007 03 17 02 41 00]);  
% Limites espacial da grade  
Lb.lat1=-35;  
Lb.lat2=5;  
Lb.lon1=-75;  
Lb.lon2=-30;  
% Intervalo de tempo da grade (horas)  
Lb.slice = 1;  
% Numero de celulas (espacial)  
Lb.grid.x=100;  
Lb.grid.y=100;  
% prefixo para as imagens  
Lb.fig='R2007';  
% Numero de descargas minimo  
Lb.k=4;  
% Arquivo de dados  
Lb.data='raios16032007.dat';  
  
% Parametros do segundo conjunto de dados  
Lc=Lb;  
Lc.ti=datenum([2008 10 01 02 40 00]);  
Lc.tf=datenum([2008 10 02 02 40 00]);  
Lc.fig='R2008';  
Lc.data='raios2008.dat';  
  
% Escolha do conjunto de dados  
L = Lb;  
%L = Lc;  
  
% Cria um mapa da região considerada
```

```

function [lat ,lon] = filtermap(r_lat , r_lon ,L)
    k=0;
    lat=0;
    lon=0;
    for n=1:L.thin: size(r_lat)
        e_lat=r_lat(n);
        e_lon=r_lon(n);
        if (e_lat>=L.lat1 && e_lat<=L.lat2 && e_lon>=L.lon1 && e_lon<=L.lon2)
            k = k + 1;
            lat(k)=r_lat(n);
            lon(k)=r_lon(n);
        end
    end
end

% Preenche as células da grade a partir das descargas
function M=slice(r_t , r_lat , r_lon ,L)
    colormap(gray);
    k=0;
    t=0;
    lat=0;
    lon=0;
    tslice = L.slice / 24;
    % cria a grade
    M=zeros(L.grid.y+1,L.grid.x+1,round((L.tf-L.ti)/tslice));
    slice_ti = L.ti;
    slice_tf = L.ti + tslice;
    s=1;
    % varre as descargas
    for n=1:size(r_t)
        if (r_t(n) < L.ti)
            continue;
        end
        e_lat=r_lat(n);
        e_lon=r_lon(n);
        % filtra as descargas na região de interesse
        if (e_lat>=L.lat1 && e_lat<=L.lat2 && e_lon>=L.lon1 && e_lon<=L.lon2)
            e_t = r_t(n);
            % determina a célula ocupada pela descarga
            x=round((e_lon-L.lon1)/(L.lon2-L.lon1)*L.grid.x)+1;

```

```

y=round((e_lat-L.lat1)/(L.lat2-L.lat1)*L.grid.y)+1;
% atualiza a célula
M(y,x,s)=M(y,x,s)+1;
% imprime a visualização da grade
if (e_t > slice_tf)
    clf
    hold on
    % imprime o fundo
    image ([L.lon1 L.lon2],[L.lat1 L.lat2],[1 1;1 1]);
    % desenha o mapa e plota as descargas
    plot(L.map.lon,L.map.lat,'.y',lon,lat,'xc');
    axis ([L.lon1,L.lon2,L.lat1,L.lat2])
    title (sprintf( '%s_-%s_(%d_descargas) ',
        datestr(slice_ti),datestr(slice_tf,13),k));
    hold off
    % salva em disco
    fig=sprintf( '%s-%02d.png',L.fig,s);
    print(fig)
    % avança para o próximo intervalo
    slice_ti = slice_tf;
    slice_tf = slice_tf + tslice;
    s = s + 1;
    k=0;
    t=0;
    lat=0;
    lon=0;
end
k = k + 1;
t(k) = r_t(n);
lat(k)=r_lat(n);
lon(k)=r_lon(n);
if (r_t(n) > L.tf)
    break;
end
end
end
end

% Programa principal
%
```

```

% Variáveis:
%
% Tabela de descargas
% r_t: coluna de tempo
% r_lat e r_lon: colunas das coordenadas
%
% map_lat, map_lon: desenho do mapa
%
% M: matriz da grade, células contém o número de descargas
% M4: matriz da grade, células densas assinaladas
% C: grade classificada em clusters

% Carrega os dados
if (!exist('r'))
    printf('Carregando_os_dados.\n');
    fflush(stdout);
    r=load(L.data);
    r_t=datenum(r(:,2:7))+r(:,8)*1e-9/86400;
    r_lat=r(:,9);
    r_lon=r(:,10);
end

% Carrega o mapa
if (!exist('map'))
    printf('Carregando_o_mapa.\n');
    fflush(stdout);
    load('cac_south_america_map.dat')
    L.thin = 6;
    map=cac_south_america_map;
    map_lon=map(:,3);
    map_lat=map(:,2);
    [map_lat, map_lon] = filtermap(map_lat, map_lon, L);
end

L.map.lat = map_lat;
L.map.lon = map_lon;

% Calcula a grade
if (!exist('M'))

```

```

    printf('Calculando_a_grade.\n');
    fflush(stdout);
    M=slice(r_t , r_lat , r_lon ,L);
    M4=(M<=L.k)*1;
end

% Determina os clusters
if (~exist('C'))
    printf('Calculando_clusters.\n');
    fflush(stdout);
    C=connect(M4);
end

% Imprime a grade classificada em clusters
figure;
nc = max(max(max(C)))+1;
% randomiza a tabela de cores
cm=[rand(nc,1) rand(nc,1)/2+0.5 ones(nc,1)];
cm=hsv2rgb(cm);
cm(1,:)= [0 0 0];
colormap(cm);
for i=1:size(C,3)
    clf
    hold on
    % imprime uma fatia da grade no tempo
    image([L.lon1 L.lon2],[L.lat1 L.lat2],C(:,:,i));
    plot(L.map.lon,L.map.lat,'.y');
    axis([L.lon1,L.lon2,L.lat1,L.lat2])
    ti=L.ti+(i-1)*L.slice;
    tf=L.ti+i*L.slice;
    title(sprintf('%s_-%s',datestr(ti),datestr(tf,13)));
    % salva em disco
    fig=sprintf('%s_grid_%02d.png',L.fig,i);
    print(fig)
end

% Atribui classes às desargas e imprime o resultado
figure;
atribui(r_t , r_lat , r_lon ,L,C,cm);

%
```

```

% Arquivo: connect.m
%

% Determina componentes conectados
function M = connect(M)
    k = 2;
    Ni=size(M,1);
    Nj=size(M,2);
    Ns=size(M,3);
    for s=1:Ns
        for i=1:Ni
            for j=1:Nj
                % procura uma célula sem atribuição de cluster
                if (M(i , j , s)==0)
                    % cria um novo cluster
                    k = k + 1;
                    % inicializa a lista de células conectadas
                    I(1) = i;
                    J(1) = j;
                    S(1) = s;
                    n=1;
                    while (n>0)
                        % retira um elemento da lista
                        pi = I(n);
                        pj = J(n);
                        ps = S(n);
                        I(n) = [];
                        J(n) = [];
                        S(n) = [];
                        n = n - 1;
                        % procura um vizinho denso e o adiciona à lista
                        if (M(pi , pj , ps)==0)
                            M(pi , pj , ps) = k;
                            if (pj<Nj && M(pi , pj+1 , ps)==0)
                                n = n + 1;
                                I(n) = pi;
                                J(n) = pj+1;
                                S(n) = ps;
                            end
                        if (pi<Ni && M(pi+1 , pj , ps)==0)

```



```
        n = n + 1;
        I(n) = pi+1;
        J(n) = pj;
        S(n) = ps;
    end
    if (pi>1 && M(pi-1,pj,ps)==0)
        n = n + 1;
        I(n) = pi-1;
        J(n) = pj;
        S(n) = ps;
    end
    if (pj>1 && M(pi,pj-1,ps)==0)
        n = n + 1;
        I(n) = pi;
        J(n) = pj-1;
        S(n) = ps;
    end
    if (ps<Ns && M(pi,pj,ps+1)==0)
        n = n + 1;
        I(n) = pi;
        J(n) = pj;
        S(n) = ps+1;
    end
    if (ps>1 && M(pi,pj,ps-1)==0)
        n = n + 1;
        I(n) = pi;
        J(n) = pj;
        S(n) = ps-1;
    end
end
end
end
end
end
end
end
end
end

%
% Archivo: atribui.m
%
```

```

% Atribui classes às descargas de acordo com a grade
function atribui(r_t , r_lat , r_lon ,L,C,cm)
    k=0;
    t=0;
    lat=0;
    lon=0;
    tslice = L.slice / 24;
    slice_ti = L.ti;
    slice_tf = L.ti + tslice;
    s=1;
    f=1;
    colormap(cm);
    cm(1,:)=[0.5 0.5 0.5];
    cl{f}={};
    for n=1:size(r_t)
        if (r_t(n) < L.ti)
            continue;
        endif
        e_lat=r_lat(n);
        e_lon=r_lon(n);
        % filtra a descarga de acordo com a região de interesse
        if (e_lat>=L.lat1 && e_lat<=L.lat2 && e_lon>=L.lon1 && e_lon<=L.lon2)
            e_t = r_t(n);
            % imprime a visualização das descargas classificadas anteriormente
            if (e_t > slice_tf)
                clf
                hold on
                image ([L.lon1 L.lon2],[L.lat1 L.lat2],[0 0;0 0]);
                plot(L.map.lon,L.map.lat,'.y');
                % plota as descargas de cada classe, colorindo de acordo.
                for i=1:size(cl{f},2)
                    p=plot(cl{f}{i}.lon,cl{f}{i}.lat,'x');
                    set(p,'color',cm(cl{f}{i}.ix,:));
                end
                axis ([L.lon1,L.lon2,L.lat1,L.lat2]);
                title (sprintf('%s_-%s_(%d_descargas)',
                    datestr(slice_ti),
                    datestr(slice_tf,13),k));
                fig=sprintf('%s_class_%02d.png',L.fig,s);
                print(fig)
            end
        end
    end

```

```

        slice_ti = slice_tf;
        slice_tf = slice_tf + tslice;
        s = s + 1;
        k=0;
        cl{f}={};
    endif
    k = k + 1;
    if (r_t(n) > L.tf)
        break;
    end
    % determina a célula a que pertence a descarga
    x=round((e_lon-L.lon1)/(L.lon2-L.lon1)*L.grid.x)+1;
    y=round((e_lat-L.lat1)/(L.lat2-L.lat1)*L.grid.y)+1;
    % determina a classe da descarga
    e_cl = C(y,x,s);
    % procura a lista de descargas dessa classe
    found=0;
    for ix=1:size(cl{f},2)
        if (cl{f}{ix}.ix == e_cl)
            found=1;
            break;
        end
    end
    if (!found)
        % cria uma nova lista
        ix=size(cl{f},2)+1;
        cl{f}{ix}.ix=e_cl;
        n=1;
    else
        n=size(cl{f}{ix}.lat)+1;
    end
    % adiciona a descarga à lista de descargas desta classe
    cl{f}{ix}.lat(n)=e_lat;
    cl{f}{ix}.lon(n)=e_lon;
end
end
end

```

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.