

PROPOSTA DE ALGORITMO INTEGRADO A SISTEMA INTERATIVO PARA CONTROLE DE QUALIDADE E VISUALIZAÇÃO DE DADOS DE RADARES METEOROLÓGICOS

Renan Henrique Luz Seluzniak

Programa de Pós Graduação em Métodos Numéricos em Engenharia, UFPR
Centro Politécnico da UFPR, Caixa Postal 19100, CEP:81531-980, Curitiba,
Paraná, Brasil
renan.seluz@gmail.com

Tiago Martinuzzi Buriol

Instituto Federal de Educação, Ciência e Tecnologia - IFPR
Rua Antônio Carlos Rodrigues, 453, CEP:83215-750, Paranaguá, Paraná,
Brasil
tiago.buriol@ifpr.edu.br

Cesar Beneti

Instituto Tecnológico SIMEPAR
Centro Politécnico da UFPR, Caixa Postal 19100, CEP:81531-980, Curitiba,
Paraná, Brasil
beneti@simepar.br

Sergio Scheer

Programa de Pós Graduação em Métodos Numéricos em Engenharia, UFPR
Centro Politécnico da UFPR, Caixa Postal 19100, CEP:81531-980, Curitiba,
Paraná, Brasil
renan.seluz@gmail.com

RESUMO

A qualidade dos dados provenientes de radares meteorológicos é comumente afetada por sinais indesejados, que não correspondem à precipitação, ocasionados pela presença de insetos e pássaros, acidentes topográficos e outras interferências. Nos últimos anos, a tecnologia de radares polarimétricos tem estado disponível permitindo a coleta de um número maior de variáveis e a sua utilização para a classificação de hidrometeoros. Essa técnica permite a fazer uma melhor distinção entre dados que representam chuva e dados “ruins”. No entanto, para dados históricos, obtidos a partir de radares de polarização simples, essas variáveis não estão disponíveis. Neste trabalho, foi implementado um algoritmo que utiliza variáveis de polarização simples para o controle de qualidade dos dados e, então, comparado com os resultados da classificação de hidrometeoros para validação do algoritmo e integração em sistema de visualização interativa. Foram utilizadas as bibliotecas Numpy e Pygame para processamento e visualização interativa dos dados.

PALAVRAS CHAVE: Radar Polarimétrico, Controle de Qualidade, Visualizador Interativo.

Área principal: Programação Matemática.

ABSTRACT

The quality of data from weather radar is commonly affected by unwanted signals, which do not correspond to precipitation, caused by the presence of insects and birds, topographical and other interference. In recent years, technology has been available polarimetric radar allowing collection of a larger number of variables and their use for the hydrometeor classification. This technique allows to make a better distinction between data representing rain and "bad" data. However, for historical data obtained from single-polarization radar, these variables are not available. In this paper, we implemented an algorithm that uses simple polarization variables for the quality control of the data and compared with the results of hydrometeors classification, to verify the algorithm efficiency and operational viability in historical data. The Numpy and Pygame libraries were used for processing and interactive visualization of data.

KEYWORDS: Polarimetric Weather Radar. Quality Control. Numpy.
Main area: Mathematical Programming

1. Introdução

A qualidade dos dados adquiridos a partir de radares meteorológicos é comumente afetada por “ruídos” indesejados, ocasionados pela presença de insetos, pássaros, partículas suspensas no ar, propagações anômalas de sinal e acidentes de topografia. Essas interferências resultam em ecos (sinais) que não correspondem à precipitação, o que pode comprometer o uso destes dados para diversas aplicações. Assim, algoritmos utilizados diferentes abordagens, usando técnicas baseadas em texturas (Dixon e Seed, 2014), utilizando a estrutura 3D dos dados (Steiner e Smith, 2002), inteligência artificial (Lakshmanan et al, 2003) ou correlacionando variáveis polarimétricas (Park et al, 2005, Lakshmanan et al, 2014 e Friedrich et al, 2006), têm sido propostos para “filtrar” os dados. Em geral, os algoritmos concentram-se inicialmente em remover sinais ocasionados por propagações anômalas e ecos de terreno, pois tais contaminações consistem em valores de alta refletividade.

Propagações anômalas são tipos de interferências que podem levar a uma significativa contaminação nos dados de radar por ecos de terreno, normalmente não detectados pelo radar. São causadas pela mudança na direção de propagação da energia emitida pelo radar devido a variações nas condições atmosféricas. Com isso, o raio de propagação pode atingir a superfície da terra e retornar sinais que se confundem com sinais de chuva, principalmente quando esses sinais ocorrem em regiões contendo chuva. Isso dificulta sua remoção, resultando em um grave problema para aplicações como o cálculo de estimativas de chuva, dentre outras. Assim, é relevante dispor de um algoritmo que possa ser integrado ao sistema de visualização de dados de radar, o que permite a análise interativa e um maior controle dos parâmetros dos filtros.

O Instituto Tecnológico SIMEPAR, opera um radar meteorológico polarimétrico, ou seja, de polarização dupla (Cao, Yeary e Zhang, 2012), instalado no município de Cascavel, no estado do Paraná. Em seu alcance máximo, cobre praticamente todo o estado e parte dos estados de Santa Catarina, Rio Grande do Sul, São Paulo e Mato Grosso do Sul e, ainda, parte dos países vizinhos Paraguai e Argentina. O radar atua coletando informações de grande valor agregado na previsão meteorológica e monitoramento de eventos severos, e no acompanhamento da entrada de sistemas frontais no Estado do Paraná. No entanto, frequentemente a qualidade dos produtos gerados é afetada por ecos que não correspondem à precipitação consistindo em um problema para sua utilização.

Embora existam muitos trabalhos na literatura científica que tratam deste assunto, a maioria dos trabalhos mencionados não apresenta a preocupação com o tempo de processamento do algoritmo, mas sim com a eficácia. No entanto, para a integração de algoritmos para controle de qualidade em sistemas de visualização interativos, é necessário equilibrar a eficácia com desempenho computacional.

Na busca de soluções para melhorar o controle de qualidade dos produtos gerados pelos radares, foi realizada uma modificação no algoritmo de Steiner e Smith (2002) visando a sua integração ao sistema de visualização interativo RADEX (Buriol e Beneti, 2014), em desenvolvimento no SIMEPAR. Para isso, foram realizadas simplificações que permitiram reduzir o tempo de processamento necessário sem comprometer significativamente os resultados. Essa modificação foi necessária para que o algoritmo responda de forma suficientemente rápida modo que o usuário pode ativar o filtro e instantaneamente visualizar o resultado. Além disso, o algoritmo modificado faz utiliza apenas o momento (variável) refletividade (UH), dentre as diversas variáveis disponíveis nos dados de radares polarimétricos. Isso permite que o algoritmo seja utilizado em dados históricos e em radares de polarização simples, nos quais um número menor de variáveis encontra-se disponível.

Para uma validação preliminar do algoritmo desenvolvido, os resultados obtidos foram comparados com os resultados do algoritmo de classificação de hidrometeoros (HMC), disponível como uma das variáveis polarimétricas. Neste trabalho foi possível verificar que a solução apresentada proporcionou a remoção de uma quantidade considerada satisfatória de

dados ruins. Além disso, o algoritmo apresentou desempenho adequado para a aplicação operacional de modo integrado ao sistema de visualização RADEX. O algoritmo tem sido testado tanto em dados históricos como dados de radares de polarização simples.

2. O sistema de visualização interativa RADEX

Como plataforma de testes, tem sido utilizado o sistema de visualização interativa RADEX - Radar Data Explorer, desenvolvido no SIMEPAR (Buriol e Beneti, 2014). O sistema foi concebido com foco na facilidade de uso e interatividade, provendo funcionalidades focadas na exploração visual e interativa do volume tridimensional de dados adquiridos pelo radar. Dentre as principais ferramentas do sistema RADEX, destacam-se a capacidade de gerar PPI (Plan Position Indicator) dinamicamente, em diferentes níveis de resolução, dependendo do *zoom*, para diferentes elevações e momentos (variáveis), e alternar entre as visualizações rapidamente para análise visual e comparação. Além de realizar cortes verticais, comparação de variáveis sobrepondo parcialmente um mesmo PPI, e outras ferramentas convenientes.

O sistema RADEX foi desenvolvido em Python fazendo uso, principalmente, dos pacotes NumPy e Pygame. A escolha dessa linguagem permitiu obter um equilíbrio, ponderando o esforço de programação demandado para o desenvolvimento do aplicativo e o desempenho final obtido. Em razão de Python funcionar como uma linguagem interpretada (embora possa ser compilada), o desempenho dos programas costuma ser inferior ao dos programas escritos em outras linguagens, como C/C++ e Fortran. Ainda assim, Python tem sido indicada como uma boa alternativa para aplicações científicas e desenvolvimento de pesquisas (Lin, 2012).

Algumas de suas características incluem ser uma linguagem fácil de aprender, ter uma vasta coleção de bibliotecas e módulos disponíveis, permitir o desenvolvimento rápido de programas (alta produtividade), possuir uma sintaxe limpa gerando códigos legíveis e possibilitar a compilação de códigos em linguagem de máquina. Dentre as bibliotecas disponíveis, destaca-se Numpy, para processamento numérico, que suporta operações com vetores e matrizes. As operações matemáticas em objetos *ndarray* (matrizes n-dimensionais) do Numpy são realizadas elemento a elemento em códigos pré-compilados escritos em C o que proporciona um considerável ganho de desempenho no processamento.

O uso de Numpy também proporciona algumas conveniências em relação à produtividade, reduzindo o esforço de programação, uma vez que dispõe de inúmeras funções para manipulação de vetores e matrizes. Um exemplo prático para ilustrar as vantagens do uso de Numpy é mostrado a seguir. Caso estejamos interessados em substituir todas as ocorrências do valor -999 de uma matriz pelo valor 0, bastaria escrever a linha de código abaixo:

```
matriz = numpy.where(matrix == -999 , 0 , matriz)
```

Com isso, Numpy tornou-se o padrão para representação de dados numéricos em Python, pois possibilita a implementação eficiente de algoritmos de processamento numérico em uma linguagem de alto nível. O ganho de desempenho é devido, em geral, ao uso de três técnicas: cálculos vetorizados, acesso aos dados na memória sem a necessidade de copiar e a minimização do número de operações aritméticas (Walt, 2011). Em sua essência, no entanto, um *array* do Numpy é apenas uma forma conveniente de descrever um ou mais blocos de memória do computador, de modo que os números representados possam ser facilmente manipulados.

Pygame é um módulo Python que fornece a API da biblioteca SDL (*Simple Direct Media*), desenvolvida em C, utilizada para visualização e criação de jogos, permitindo a portabilidade de jogos entre diferentes plataformas. Por ser destinada ao desenvolvimento de

jogos, também é adequado para o desenvolvimento de sistemas de visualização interativos, proporcionando um conjunto de funções bastante convenientes, por exemplo, para acesso direto ao *hardware* gráfico, *mouse* e teclado (McGugan, 2007).

A visualização dos dados do radar polarimétrico utilizando o aplicativo RADEX (figura 1), permitiu uma maior conveniência na exploração e identificação das características dos dados e dos múltiplos momentos disponíveis. Por exemplo, o uso do algoritmo de zoom do RADEX permite aproximar a imagem e analisar mais detalhadamente uma região específica dos dados sem que a estrutura original dos dados, em um sistema de coordenadas esféricas, seja alterada. Outras características desse sistema é a de oferecer controles gráficos sobre a imagem 2D (bidimensional) para visualizar a estrutura interna do volume 3D (tridimensional) de dados, por exemplo, pela geração rápida de cortes verticais.

Por meio de inspeção visual do volume de dados de refletividade, é possível verificar que ecos de terreno devido a propagações anômalas têm pequena extensão vertical e tendem a contaminar somente as varreduras de baixa elevação (geralmente até a 2ª elevação), dependendo do gradiente vertical do índice de refração (Steiner e Smith, 2002). Propagações anômalas de sinal também podem ser reconhecidas, a partir de dados de refletividade, devido a uma maior variabilidade espacial em comparação com ecos de precipitação. Já os sinais regulares de acidentes de terreno apresentam maior correlação temporal.

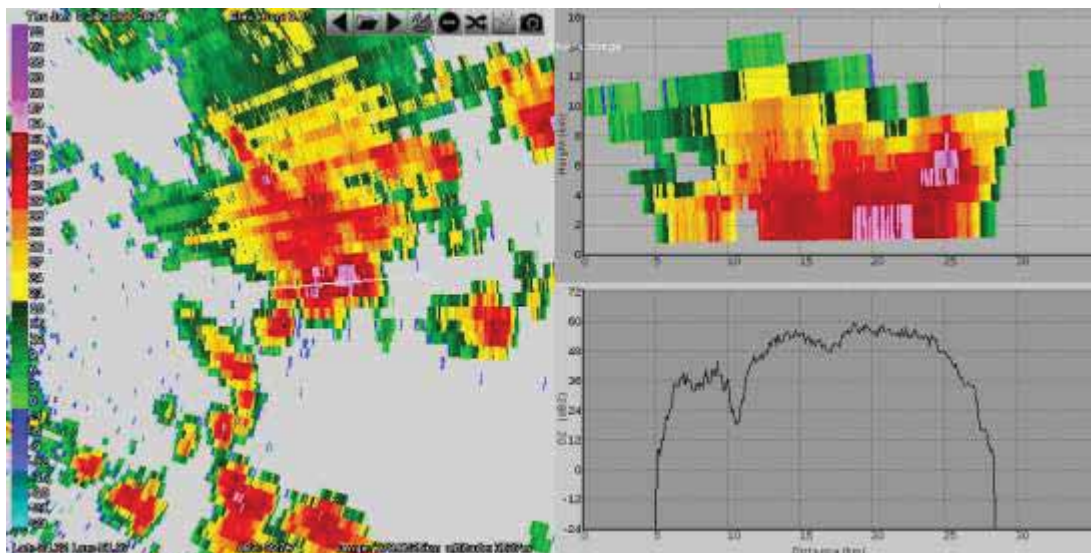


Figura 1: Exemplo de visualização de dados do radar de polarimétrico de Cascavel no RADEX. O quadro da esquerda apresenta o *zoom* em uma área do PPI. No quadro superior do lado direito é mostrado um corte vertical do volume de dados e abaixo o gráfico dos valores de refletividade (dBZ), na elevação visualizado, ao longo do corte.

3. Técnica de Filtragem

Foi implementado no sistema RADEX o algoritmo proposto neste trabalho, o qual se baseia no algoritmo de Steiner e Smith (2002), para identificação e remoção de ecos de terreno causados por propagações anômalas. O algoritmo foi projetado para ser aplicado em um único volume de dados, fazendo uso da estrutura tridimensional dos dados de refletividade. No algoritmo, são utilizados três parâmetros: a extensão vertical de ecos de radar (ECHOtop), a variação absoluta da refletividade (SPINchange) e o gradiente vertical de refletividade (vertGRAD). O algoritmo é executado no conjunto de dados dispostos em uma grade descrita em coordenadas esféricas com três graus de liberdade (r , θ , φ) (figura 2). O primeiro, r , é o range (distância do alvo até radar), θ é o azimute (ângulo em relação ao

norte), e φ é a elevação da antena, realizado para cada elemento da matriz de refletividade (UH).

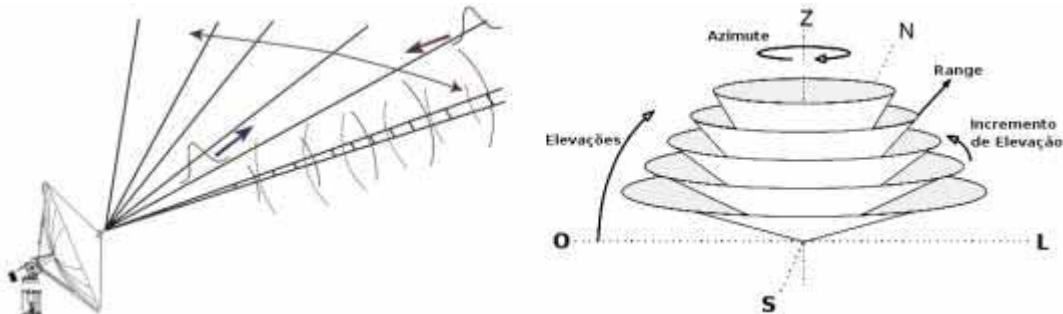


Figura 2. Esquema de varredura e a estrutura do volume de dados.

Em Steiner e Smith (2002), o parâmetro ECHOTop estima a maior altitude onde o valor de refletividade (dBZ) excede um valor específico (threshold), correspondente a um intensidade mínima (REFLthresh = 5 dBZ), já que valores abaixo deles não são significativos. Então, é realizada a checagem para verificar se o ECHOTop pertence ou não a primeira elevação, caso ele pertença, este elemento não representa chuva, sendo portanto, removido. Ao contrário, é mantido. O parâmetro SPINchange indica o número de flutuações de refletividade, expressando assim todas as possíveis mudanças de spin. Variações maiores que 2 dBZ (aumento ou redução) nos valores de refletividade, a partir de um pixel para o pixel adjacente na direção radial, resultam no que é chamado de spin. Quando há aumento de refletividade é dito que ocorreu spin para cima se houve redução, é dito spin para baixo. Valores abaixo de 2 dBZ são considerados insignificantes e, portanto, não tem efeito sobre o parâmetro spin. O parâmetro vertGRAD estima a diferença de refletividade entre os pixels das duas primeiras elevações, normalizando pela diferença dos ângulos das elevações. A unidade do vertGRAD é $dBZ \times Graus^{-1}$.

A figura 3 mostra o esquema de decisões do algoritmo desenvolvido para este trabalho. A sequência é aplicada para cada elemento da matriz polar do volume sendo denotado como Zpixel. Primeiramente, são removidos todos os valores com refletividade abaixo de um limiar (REFLthresh = 5 dBZ). Então, é realizado o teste para verificar se o valor ECHOTop está contido na primeira elevação, caso ele esteja, ele é removido. Posteriormente é realizado um teste verificando a variação na refletividade (correspondendo à mudança de spin em Steiner e Smith), o teste compara os valores de SPINchange e SPINthresh (é um filtro mais agressivo para intensidades superiores). Se SPINthresh for maior que SPINchange, conclui-se que a intensidade do SPINchange é inferior a ponto de não influenciar numa chuva, portanto, mantêm-se o Zpixel, caso contrário, é realizado o teste com vertGRAD. Se o mesmo for menor que $10 dBZ \times Graus^{-1}$, mantêm-se o Zpixel, caso contrário, é utilizado o pixel da elevação acima.



Figura 3: Fluxograma do algoritmo, verificado para cada elemento da matriz

Para realizar a validação do algoritmo apresentado, foi realizada a comparação dos resultados obtidos com os resultados do algoritmo de classificação de hidrometeoros (HMC) disponível como um dos momentos (variáveis) do radar polarimétrico. O algoritmo HMC é baseado em lógica fuzzy e é usado para identificação das partículas captadas pelo radar (Lim, Chandrasekar e Bringi, 2005). O algoritmo classifica os hidrometeoros como: *Ground clutter*; *Sea clutter*; *Biological*; *Dry snow*; *Wet snow*; *Ice crystals*; *Graupel*; *Big drops*; *Rain*; *Heavy rain* e *Hail* (figura 4).

A avaliação do resultado foi realizada por meio da comparação dos dados considerados “não chuva” com os dados identificados como *clutters* (ecos, propagações anômalas) e agentes biológicos pelo algoritmo HMC. Ou seja, a classificação de hidrometeoros processada pelo algoritmo HMC foi considerada como sendo o resultado correto. Então, foram contabilizados os valores classificados corretamente pelo algoritmo desenvolvido. A figura 4 mostra o resultado do algoritmo HMC, onde é possível ver a classificação em agentes biológicos (BI), ecos de terreno (GC), chuva (RA), dentre outros.

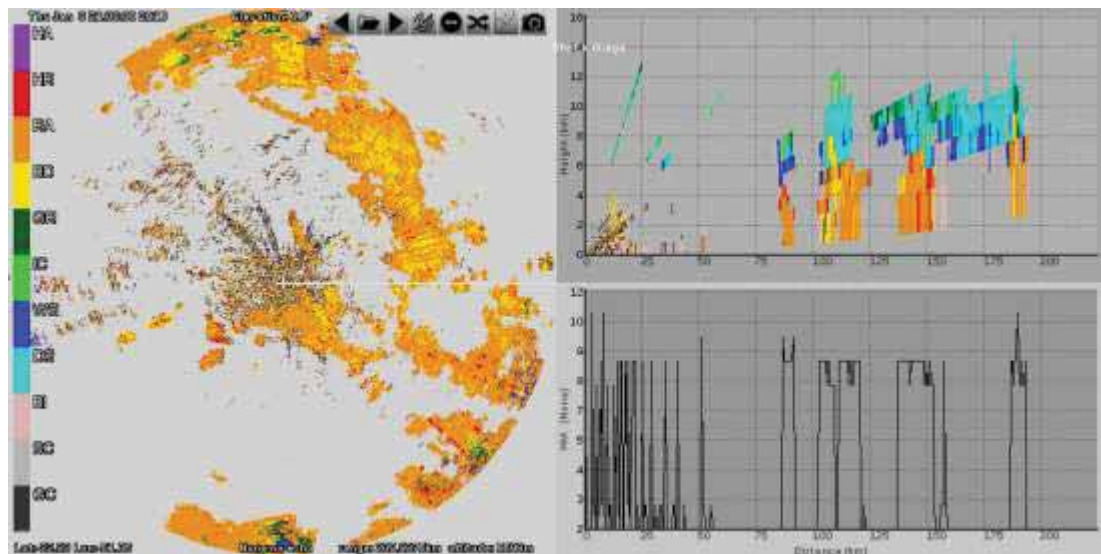


Figura 4: Classificação de hidrometeoros.

4. Resultados

A seguir são apresentados os resultados obtidos para dados em três dias e horários diferentes. Foram utilizados os dados do radar polarimétrico em que foi aplicado o filtro desenvolvido e comparado com os resultados do algoritmo HMC. O objetivo é ilustrar e o comportamento do algoritmo na remoção de ecos, ruídos, propagações anômalas (figuras 5 a 7). Com os resultados do algoritmo HMC, foi possível verificar o que representa dados meteorológicos (chuvas) e o que não representa dados meteorológicos (ecos, ruídos, propagações anômalas). Para medir a qualidade da filtragem foram contabilizadas as quantidades de pixels classificados corretamente em comparação aos resultados do HMC (tabelas 1 a 3).

A figura 5 mostra dados do dia 08/01/2015 às 21h38min.

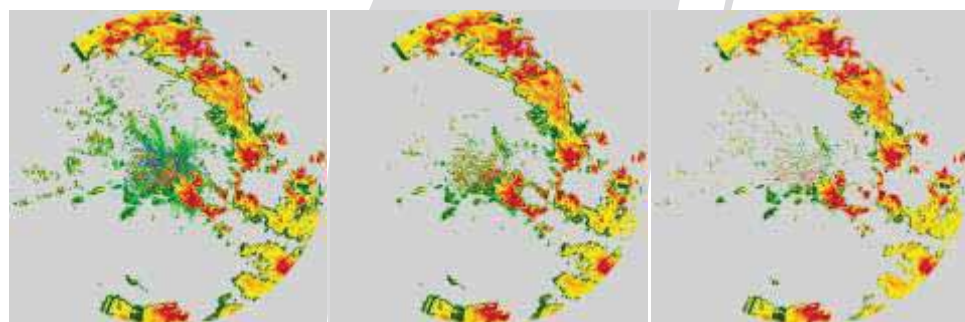


Figura 5 : Dia 08/01/2015 às 21h38min: dados originais (esquerda); imagem filtrada via algoritmo desenvolvido (centro) e; filtragem via HMC(direita).

Número de pixels com HMC	157765
Número de pixels em comum	146691
Porcentagem de pixels que representam chuva através do algoritmo	92.98%

Tabela 1: Comparações do algoritmo desenvolvido com HMC

A figura 6 mostra dados do dia 13/02/2015 às 21h00min.

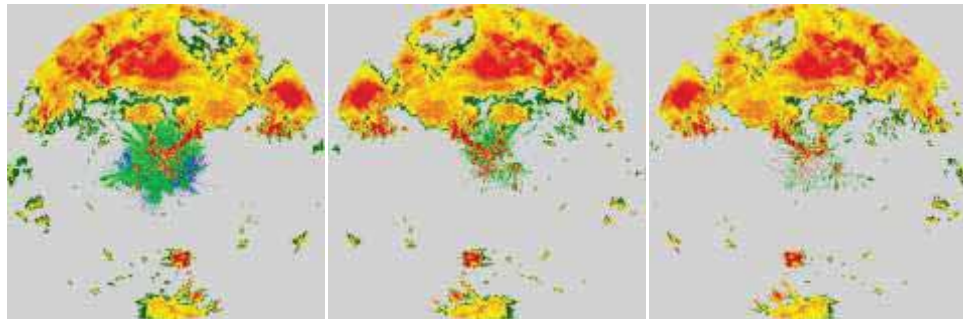


Figura 6 : Dia 13/02/2015 às 21h00min: dados originais (esquerda); imagem filtrada via algoritmo desenvolvido (centro) e; filtragem via HMC(direita).

Número de pixels com HMC	223510
Número de pixels com algoritmo	208284
Porcentagem de pixels que representam chuva através do algoritmo	93.18%

Tabela 2: Comparações do algoritmo desenvolvido com HMC

A figura 7 mostra dados do dia 28/03/2015 às 21h00min.

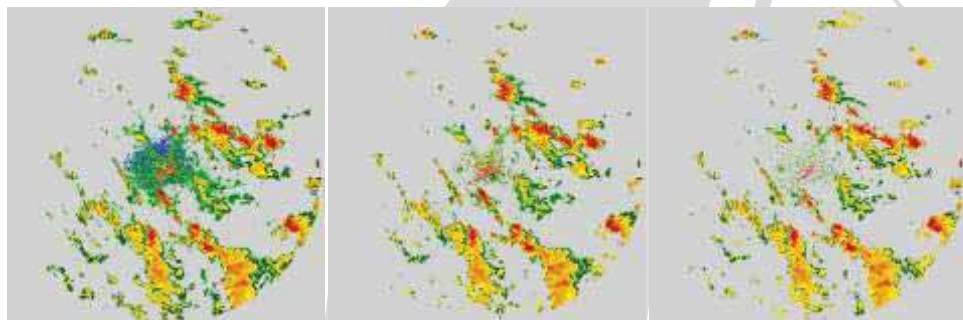


Figura7 : Dia 28/03/2015 às 21h00min: dados originais (esquerda); imagem filtrada via algoritmo desenvolvido (centro) e; filtragem via HMC(direita).

Número de pixels com HMC	136294
Número de pixels com algoritmo	121066
Porcentagem de pixels que representam chuva através do algoritmo	88.82%

Tabela 3: Comparações do algoritmo desenvolvido com HMC

Pode ser observado que a filtragem via algoritmo desenvolvido é muito próxima da filtragem via HMC. Mesmo o algoritmo desenvolvido removendo uma quantia menor dos dados que não representam chuva (ecos, ruídos, propagações anômalas), o algoritmo apresentou

resultado satisfatório e por ser desenvolvido em Numpy, o tempo de espera do usuário foi baixo em ambos os dados (demorando menos de 1.5 segundos em ambas as simulações), tornando viável o seu uso no RADEX, pois tem que ocorrer uma harmonia em tempo de processamento quanto na qualidade de filtragem dos dados. Por ser confiável para o uso do radar polarimétrico, é concluído que pode ser usado esta filtragem desenvolvida no trabalho para dados históricos.

5. Considerações Finais

Nesse trabalho, foi implementado uma modificação do algoritmo de Steiner e Smith (2002) para controle de qualidade de dados de radar meteorológico, o qual utiliza apenas a variável refletividade (UH) de polarização simples. Os resultados obtidos pela aplicação desse algoritmo foram comparados aos resultados do algoritmo de classificação de hidrometeoros (Lim, 2005), disponíveis dentre os dados gerados pelo radar polarimétrico (de polarização dupla). Na sequência foram propostas algumas modificações no primeiro algoritmo, o que proporcionou uma ligeira melhoria nos resultados obtidos, considerando os dados testados, adquiridos pelo radar de simples polarização, instalado na cidade de Teixeira Soares, PR. Para realizar o processamento e a visualização dos dados, foi utilizado o sistema RADEX, desenvolvido na linguagem de programação Python, juntamente com as bibliotecas Numpy e Pygame.

Referências

- Buriol, T. M. e Beneti, C.** (2014), Visualização Científica Interativa de Dados de Radar Meteorológico. *Anais do XXXV CNMAC*, 1-6.
- Cao, Q., Yeary, M. B. e Zhang, G.** (2012), Efficient Ways to Learn Weather Radar Polarimetry. *IEEE Transactions on Education*, 55, 58-68.
- Dixon, M., Seed, A.**, Developments in echo tracking - enhancing TITAN. ERAD, August, 2014.
- Friedrich, K., Hagen, M. e Einfalt, T.** (2006), A Quality Control Concept for Radar Reflectivity, Polarimetric Parameters, and Doppler Velocity. *Journal of Atmospheric and Oceanic Technology - J ATMOS OCEAN TECHNOL*, vol. 23, no. 7. DOI: 10.1175/JTECH1920.1
- Lakshmanan, V., Hondl, K., Stumpf, G. e Smith T.**, Quality control of wea. radar data using texture features and a neural network, in 5th Int'l Conf. on Adv. in Patt. Recogn., (Kolkata), IEEE, Dec 2003.
- Lakshmanan, V., Karstens, C., Krause, J., e Tang, L.** (2014), Quality Control of Weather Radar Data Using Polarimetric Variables. *J. Atmos. Oceanic Technol.*, 31, 1234-1249.
- Lim, S., Chandrasekar, V. e Bringi, V.N.** (2005), Hydrometeor classification system using dual-polarization radar measurements: model improvements and in-situ verification. *IEEE Trans. Geosci. Remote. Sens.* 43, 792-801.
- Lin J.W.** Why python is the next wave in earth sciences computing. *Bulletin of the American Meteorological Society*, 93(12):1823-1824, 2012a.
- McGugan W.** Beginning Game Development with Python and Pygame. Apress, 2007. ISBN 978-1-59059-872-6.
- Park, H., A. Ryzhkov, D. Zrnica, e K. Kim** (2009), The Hydrometeor Classification Algorithm for the polarimetric WSR-88D: Description and application to a mcs. *Wea. Forecasting*, 24 (3), 730-748.
- Steiner, M. e Smith, J.** (2002): Use of three-dimensional reflectivity structure for automated detection and removal of non-precipitating echoes in radar data. *J. Atmos. Ocea. Tech.*, 19, 673-686.
- Walt, S., Colbert, S. e Varoquaux, G.**, The NumPy array: a structure for efficient numerical computation. In *IEEE Computing in Science Engineering*, March/April 2011.