



RADEX - Um Sistema para Visualização Interativa e Análise de Dados de Radares Meteorológicos

Tiago Martinuzzi Buriol

tiago.buriol@ifpr.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Paraná - IFPR
Av. Victor Ferreira do Amaral nº 306, CEP 82530-230, Curitiba-PR, Brasil

Renan Henrique Luz Seluzniak

renan.seluz@gmail.com

Programa de Pós Graduação em Métodos Numéricos em Engenharia - UFPR
Centro Politécnico da UFPR, CEP 81531-980, Curitiba, Paraná, Brasil

Cesar Beneti

beneti@simepar.br

Instituto Tecnológico SIMEPAR
Centro Politécnico da UFPR, CEP 81531-980, Curitiba, Paraná, Brasil

Sergio Scheer

scheer@ufpr.br

Universidade Federal do Paraná - UFPR
Rua XV de Novembro nº 1299, CEP 80060-000, Curitiba, Paraná, Brasil

Resumo. A visualização de dados de radares meteorológicos auxilia o analista a identificar a entrada de sistemas frontais e eventos severos, dando importante suporte cognitivo à avaliação e à tomada de decisões. Mas, tradicionalmente, esses sistemas têm mantido alguns padrões de interface e funcionalidades que têm persistido por muitos anos. Com a evolução da tecnologia a quantidade de informações geradas pelos radares aumentou, em muitos casos, mais de cinco vezes, assim, também cresceu o interesse por soluções que permitam analisar e extrair mais rápida e facilmente informações relevantes a partir desses dados. Neste trabalho é apresentado o sistema RADEX para visualização interativa de dados de radares meteorológicos. Sua interface é orientada ao usuário, projetada para ser simples e intuitiva. Provê ferramentas personalizadas, com controles gráficos para definição de parâmetros em algoritmos de visualização. Os produtos de visualização são gerados "sob demanda" e armazenados de modo que possam ser acessados e alternados instantaneamente para comparação e análise visual.

Palavras-chave: Visualização Científica Interativa, Python, Radar Meteorológico

1 INTRODUÇÃO

Os radares meteorológicos surgiram durante a Segunda Guerra Mundial e hoje são amplamente utilizados para estudar e monitorar características do tempo. Esses equipamentos trabalham emitindo pulsos de energia eletromagnética, em frequência de micro-ondas, para a atmosfera. Quando os pulsos interceptam algum objeto, como gotas de chuva, por exemplo, parte da energia é "refletida" e retorna para o radar, por essa razão usa-se o termo "refletividade". Essas informações são processadas e os dados produzidos são usados para monitorar o clima e realizar previsões a curto prazo, dentre outras aplicações. De forma simplificada, um radar meteorológico funciona apontando inicialmente sua antena em uma direção que faz um pequeno ângulo vertical (ângulo de elevação), então, emite um pulso eletromagnético por uma fração de segundo e recebe o retorno da energia, em seguida o radar gira aproximadamente 1° e, então, repete o processo. Uma vez que o radar completa uma revolução (varredura) o ângulo de elevação da antena é aumentado e repete-se o processo (Rinehart, 2004). O conjunto de varreduras forma um volume de dados distribuídos em uma grade descrita em coordenadas esféricas.

Atualmente, o Instituto Tecnológico SIMEPAR opera dois radares meteorológicos. O mais antigo deles, instalado no município de Teixeira Soares no estado do Paraná, em 1996, é um radar de polarização simples, ou seja, emite somente ondas horizontais. O mais novo, instalado em 2013 no município de Cascavel no estado do Paraná, é um radar de polarização dupla, ou seja, emite ondas verticais e horizontais (Islam and Rico-Ramirez, 2013; Cao et al., 2012; Silveira, 1998). Em ambos os radares, o alcance máximo cobre praticamente todo o estado do Paraná 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. Esses radares funcionam continuamente, 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. Pelo menos a cada dez minutos é gerado um volume que pode conter milhões de pontos e cerca de vinte variáveis associadas a cada ponto da grade tridimensional. Assim, ferramentas interativas que permitam visualizar esses dados e extrair rapidamente informações relevantes a partir deles, são de grande importância (Tory and Möller, 2004). Ferramentas computacionais que permitem interagir e explorar visualmente grandes conjuntos de dados são recursos de grande valor para a análise e a compreensão de fenômenos complexos. No entanto, muitos aspectos, incluindo fatores humanos e computacionais, apresentam desafios consideráveis para o desenvolvimento de ferramentas de visualização eficientes (Tory and Möller, 2004).

Neste trabalho é apresentado o sistema RADEX de visualização interativa, para análise de dados de radares meteorológicos. O sistema foi desenvolvido em Python e utiliza um conjunto de módulos e bibliotecas especializadas, tais como NumPy, para processamento numérico, e Pygame para visualização e interação. O ideia principal do aplicativo é prover um meio fácil e rápido de explorar os dados de forma interativa e intuitiva. Foi projetado com orientação ao usuário buscando disponibilizar ferramentas baseadas em controles gráficos e possibilitam gerar produtos de visualização "sob demanda", ou seja, os produtos são gerados a medida que o usuário interage com o sistema. Uma vez gerado, um produto de visualização (*buffer* de imagem) é armazenado e pode ser acessado instantaneamente. Essa abordagem não requer o pré processamento dos dados e permite mecanismos para comparação, alternando rapidamente entre uma imagem e outra. Os mecanismos de processamento e visualização interativa dos dados proporciona uma navegação espaço-temporal oferecendo suporte cognitivo ao analista.

2 DADOS E PRODUTOS DE RADARES METEOROLÓGICOS

A tecnologia de radares meteorológicos evoluiu consideravelmente nos últimos dez anos e tem estado cada vez mais disponíveis em vários centros operacionais em todo o mundo. As informações geradas com esses equipamentos têm, em muitos casos, aumentado mais de cinco vezes, dessa forma, analisar e extrair rapidamente informações relevantes deste conjunto de dados, complexos e multidimensionais, é um desafio. Radares de dupla polarização (polari-métricos), como é o caso do radar de Cascavel (CAS), operado pelo SIMEPAR, emitem ondas horizontal e verticalmente, permitindo detectar maior quantidade de partículas presentes na atmosfera e coletar um número maior de informações, em comparação a radares de polarização simples. Em geral, radares mais antigos são de polarização simples, como é o caso do radar de Teixeira Soares (TXS).

Os radares operados pelo Instituto tecnologico SIMEPAR geram dados a partir da emissão de feixes de ondas eletromagnéticas a cada 1 grau (varredura azimutal), completando um giro de 360 graus para cada elevação da antenna (inclinação vertical). O radar TXS, por exemplo, dispõe para cada ponto em seus dados, pelo menos três variáveis (chamadas de "momentos"): Refletividade (DBZH), Velocidade Radial (VELH) e Largura Espectral (WIDTHH). Já o radar CAS possui mais de vinte momentos associados a cada ponto. Ao gerar imagens para visualização destes dados, transformações entre o sistema de coordenadas esférico para cartesiano são realizadas. Adicionalmente, uma série de rotinas envolvendo métodos numéricos computacionais, tais como interpolação, seleção e mapeamento dos dados são processados. Portanto, o desempenho computacional é um dos aspectos chave para proporcionar uma interação fluida em um aplicativo para visualização interativa.

O volume de dados gerado ao final de um ciclo completo do radar TXS corresponde a uma sequência de 360 varreduras azimutais com 14 diferentes elevações da antenna, com 800 pontos (gates ou bins) igualmente espaçados ao longo de cada feixe de 200 km. Por fim, tem-se uma grade em coordenadas polares com resolução de 360x14x800, completando 4.032.000 pontos. O radar CAS pode gerar até 2500 *gates* por feixe. Cada um destes pontos podem conter dezenas de variáveis associadas a ele. A cada 10 minutos é gerado um volume de dados descrito em coordenadas esféricas, com três graus de liberdade (r , θ , ϕ) (Figura 1). O primeiro, r , é o range (distância do alvo até radar), θ é o azimuth (ângulo em relação ao norte), e ϕ é a elevação da antenna.

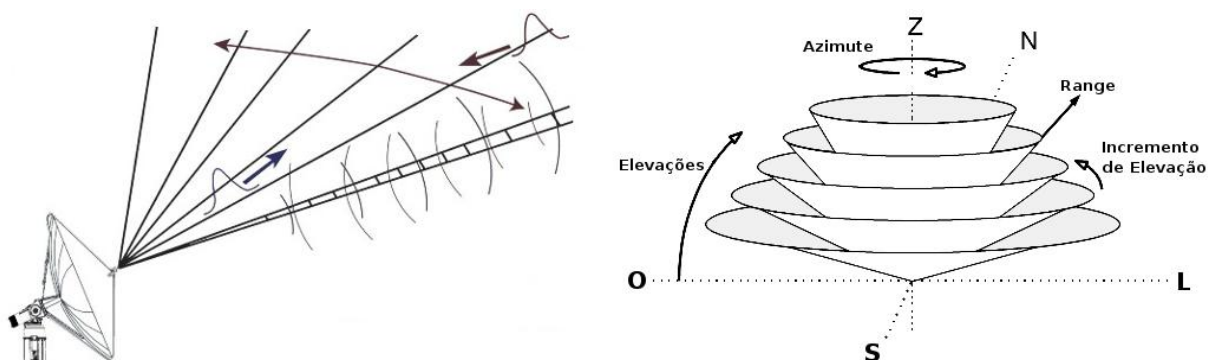


Figura 1: Esquema de varredura e a estrutura do volume de dados.

As imagens geradas a partir dos dados do radar são chamadas de produtos de visualização (ou simplesmente "produtos"). Como exemplo mais básico de um produto de radar meteoroló-

gico, tem-se o PPI (Plan Position Indicator) que é a projeção de uma única elevação (varredura de um giro completo sob um ângulo vertical fixo) sobre um mapa ou segundo uma projeção cartográfica. Em um PPI, cada valor de um momento é atribuído à uma área definida por um retângulo polar. Quanto mais próximo do radar, menor é a área desse retângulo, e portanto, a resolução dos dados é maior nas proximidades do centro do PPI. Outros produtos são o CAPPI (Constant Altitude Plan Position Indicator) que consiste em um corte horizontal no volume a uma altura constante e o RHI (Range Height Indicator) que é a visualização de um corte vertical na direção radial do radar.

Dentre as dezenas de variáveis que um radar pode medir, destaca-se a refletividade (DBZH) que é um fator entre a irradiância emitida pelo radar e a recebida por ele depois de retroespalhadas pelos hidrometeoros (gotas de chuva) presentes na atmosfera. A unidade utilizada é o dBZ, que é uma escala logarítmica da refletividade. Os valores variam entre -24 dBZ e 72 dBZ, quanto maiores forem esses valores, maiores serão os diâmetros das gotas presentes no volume medido, indicando maior intensidade de precipitação. Os diversos produtos gerados a partir dos dados são visualizados e analisados a fim de se obter informações para previsões a curto prazo, como possibilidade de tempestades e tormentas. A figura 2 mostra um sistema 2D para visualização de dados do radar e, também, uma representação simplificada da estrutura 3D dos dados como eles são originalmente adquiridos.

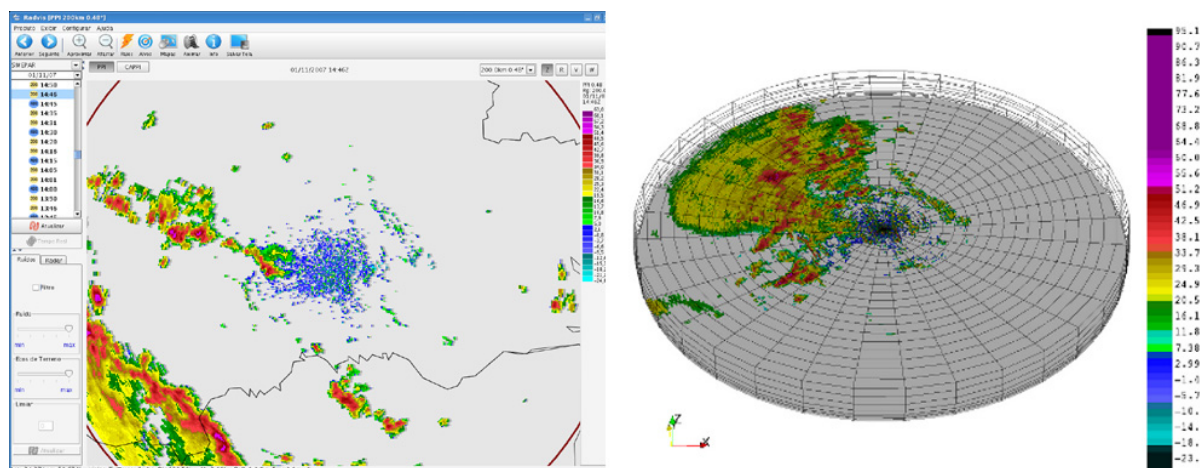


Figura 2: Uma ferramenta utilizada para visualizar produtos do radar e a estrutura original do volume de dados em uma grade em coordenadas esféricas

Em geral, a abordagem adotada é pré-processar os dados mapeando-os em uma grade cartesiana com resolução previamente definida, em seguida os produtos são gerados e armazenados como imagens e disponibilizados consulta e visualização. Diversos produtos, previamente escolhidos, podem ser gerados automaticamente em um servidor, armazenados e disponibilizados. Quanto maior for o número de produtos diferentes a serem gerados e maior a resolução do volume de dados, mais intenso é o processamento numérico necessário e, portanto, linguagens de programação e técnicas que permitam aumentar o desempenho são utilizadas. Uma vez que os produtos estão disponíveis, é comum, no cotidiano do analista, visualizar muitas imagens na mesma tela, a fim de comparar variáveis de uma mesma varredura e proceder a análise dos dados e tomada de decisões.

Algumas desvantagens desse tipo de abordagem é que ao converter os dados em uma grade cartesiana de resolução pré definida, perde-se a característica e estrutura original dos dados (em

coordenadas polares). Outra desvantagem é que o usuário, ou analista, não tem grande liberdade para explorar o interior do volume de dados. A geração de produtos personalizados, como cortes em planos de interesse específico do analista, nem sempre está facilmente disponível. Assim, um sistema de fácil utilização e ofereça ao analista maior liberdade para controlar e gerar produtos de seu interesse, processando os dados "sob demanda", a medida que o usuário solicita, seria de grande valor.

Neste trabalho tem-se desenvolvido um sistema para visualização interativa de dados de radar que munido de ferramentas projetadas para serem fáceis de usar, com resposta rápida e diferentes das abordagens convencionais. A ideia é explorar a evolução dos hardwares gráficos e em novos paradigmas de interação para criar ferramentas de visualização orientadas ao usuário, que possam oferecer um ganho no processo de análise desses dados. Para isso, é necessário gerar os produtos de visualização a partir dos dados originais, de forma que a estrutura inicial seja perdida. Com isso, os produtos devem ser obtidos a partir de solicitações que o usuário faz interagindo graficamente sobre uma representação dos dados. A abordagem adotada é armazenar os produtos a medida que vão sendo gerados, para que o usuário possa "navegar" entre as imagens alternando rapidamente entre uma e outra a fim de analisar e comparar os produtos.

3 O VISUALIZADOR INTERATIVO RADEX

O sistema RADEX foi desenvolvido em Python (www.python.org), que é uma linguagem de programação conhecida por ser versátil, possuir uma vasta biblioteca padrão, além de módulos de terceiros e, também, por ser de fácil aprendizagem. Outras de suas características incluem permitir o desenvolvimento rápido de programas (alta produtividade) (Wei-Bing L. J., 2012), possuir uma sintaxe limpa gerando códigos legíveis e possibilitar a compilação de códigos em linguagem de máquina. Assim, buscando um equilíbrio entre esforço demandado para o desenvolvimento e o desempenho final do aplicativo, optou-se por utilizar, além da linguagem Python, os pacotes Pygame (www.pygame.org) e Numpy (www.numpy.org).

Pygame é uma biblioteca multiplataforma para o desenvolvimento de jogos, que funciona em uma camada sobre a biblioteca SDL (Simple DirectMedia Layer). Esta última, é uma biblioteca multimídia livre e de código aberto, multiplataforma, escrita em C, que representa uma interface simples para acesso direto ao hardware gráfico, de áudio, e dispositivos de entrada de várias plataformas. Por ser destinada ao desenvolvimento de jogos, Pygame também é adequado para o desenvolvimento de sistemas de visualização interativos, proporcionando um conjunto de funções convenientes para o desenvolvimento de controles gráficos e mecanismos de interação (Jones, 2005; McGugan, 2007).

NumPy é o pacote básico para computação científica e processamento numérico em Python. Possui estruturas de dados que permitem trabalhar com arranjos, vetores e matrizes de N dimensões de com considerável eficiência (Coelho, 2007) e conveniências. Provê funções e operações sofisticadas, realizadas por códigos compilados escritos em C, o que garante um desempenho superior se comparado às operações com vetores originais do Python.

O uso de Numpy proporciona algumas conveniências em relação à produtividade, reduzindo consideravelmente o esforço de programação, uma vez que dispõe de inúmeras funções para manipulação de vetores multidimensionais e rotinas de álgebra linear. 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)
```

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 S. et al., 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.

Este conjunto de bibliotecas proporcionou agilidade no desenvolvimento e na experimentação de ferramentas e funcionalidades de visualização altamente interativas, focadas na facilidade de uso e na velocidade de resposta dos algoritmos às solicitações do usuário. Buscou-se com as funcionalidades implementadas proporcionar um meio efetivo de exploração e de análise visual dos dados, preservando a sua estrutura original. No sistema desenvolvido, a interação ocorre de maneira gráfica e um PPI serve como ponto de partida para a definição dos parâmetros usados nos algoritmos de visualização.

Por exemplo, a partir da tela inicial, se o usuário manter pressionado o botão esquerdo do mouse e mover o cursor sobre a representação dos dados, uma linha para o corte vertical será mostrada. No instante em que o usuário soltar o botão o algoritmo que processa os dados para gerar o corte é executado, e assim, a visualização do corte e o gráfico *ascope*, mostrando o valor na variável ao longo da linha de corte, para uma elevação, são apresentados em uma fração de segundo na parte lateral do visualizador (figura 3).

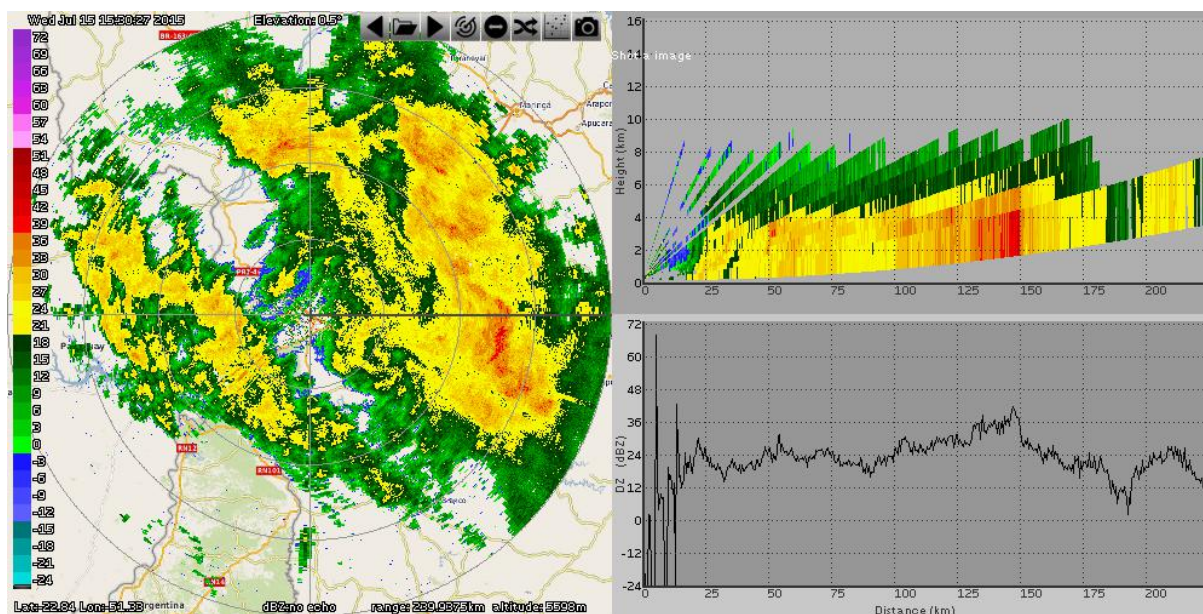


Figura 3: Tela do sistema RADEX para visualização interativa de dados de radares meteorológicos.

A estratégia adotada, chamada aqui de geração de produtos "sob demanda", consiste em manter um conjunto de dados na memória RAM da máquina e obter produtos diversos a partir

destes dados, interagindo diretamente em uma representação visual, que serve como base (por exemplo, um PPI). Dessa maneira, é possível dar *zoom* ou obter cortes verticais do volume de dados criando novas imagens a partir dos dados originais, sem haver a necessidade de converter previamente a grade polar em uma grade cartesiana. Para descrever brevemente a utilização da ferramenta, supondo que o usuário queira ver um vorte vertical, basta para isso traçar uma linha sobre o PPI usando o com o mouse (clique, arrastar e soltar). Ou então, para ampliar e transladar gradualmente a imagem, basta apontar o ponto de foco e girar a rodinha do mouse, assim que o movimento cessa, a imagem é atualizada (reprocessada) a partir dos dados originais.

Para conferir uma resposta rápida na geração do corte vertical, a abordagem utilizada foi calcular a posição espacial correspondente a cada pixel da imagem que representa o corte e, então, buscar nos dados o valor da variável para aquela posição. Essa estratégia é conveniente para a utilização do pacote Numpy, pois as operações com matrizes podem ser escritas com poucas linhas de programação e o desempenho tem sido satisfatório para o propósito do trabalho. Não é objetivo deste trabalho apresentar testes de velocidade de processamento, no entanto, é possível estimar que em um PC comum, um tempo menor que 1/5 de segundo, o que é aceitável.

A representação dos dados, como já foi mencionado, pode ser ampliada (*zoom in*) interativamente, em tempo real, utilizando a rodinha do mouse e mantendo a seta do mouse no centro área de interesse. A ampliação da imagem é apresentada imediatamente, utilizando as funções do Pygame para aplicar um fator de escala na imagem. Após a interação cessar (giro da rodinha do mouse), a imagem é atualizada, sendo reprocessada a partir dos dados originais. Com isso, a estrutura polar original dos dados é preservada. É comum, em aplicativos para visualizar este tipo de dado, a conversão da estrutura polar em uma grade cartesiana, nesse caso, ao ampliar a imagem a resolução e a estrutura original dos dados são perdidos. A figura 4 mostra a ampliação de uma região em que manteve-se a estrutura original dos dados (figura da esquerda) em comparação a uma ampliação realizada sobre uma região previamente convertida para o sistema cartesiano (figura da direita). As estruturas de matrizes e as operações algébricas disponíveis em

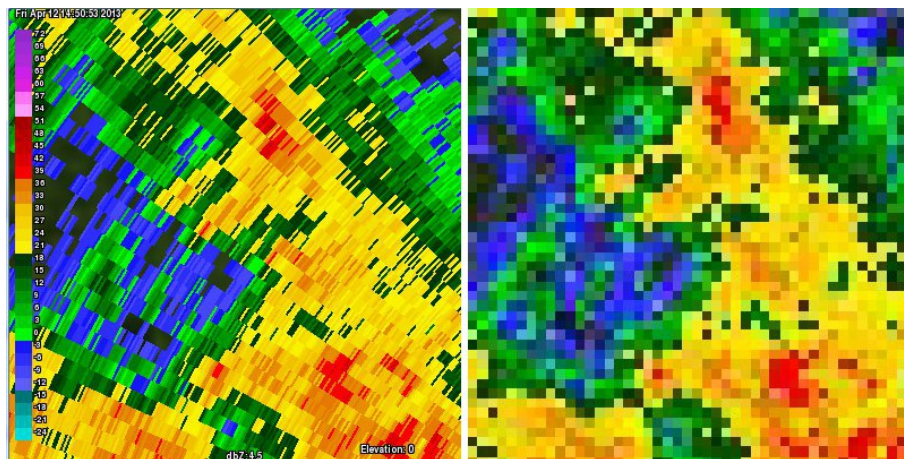


Figura 4: Visualização ampliada dos dados mantendo a estrutura polar (esquerda) e visualização ampliada dos dados previamente convertidos em estrutura cartesiana

Numpy permitem em um única linha de código, por exemplo, realizar operações sobre todos os elementos de uma matriz de forma otimizada. Para exemplificar, em um caso específico em que se deseja criar uma grade cartesiana de resolução 500x500, e converter as coordenadas (x, y) dos pontos em coordenadas polares (r, t), pode-se fazer com as 3 linhas mostrados abaixo.

```
X , Y = numpy.meshgrid(numpy.arange(1000), numpy.arange(1000))  
R = numpy.sqrt(X**2 + Y**2)  
T = numpy.arctan2(Y,X)
```

4 VISUALIZAÇÃO INTERATIVA E PRODUTOS "SOB DEMANDA"

Visando proporcionar ao usuário uma experiência de análise de dados fluida, por meio de uma interface simples e mecanismos de interação eficazes, adotou-se a abordagem de geração de produtos de visualização "sob demanda". A ideia é realizar o processamento dos dados e a geração das imagens (produtos de visualização) a medida que o usuário solicita ao sistema de forma interativa. Por exemplo, ao abrir um arquivo de dados, o usuário visualiza um PPI da variável refletividade e um corte vertical padrão. Ao selecionar outra variável, outro PPI e corte vertical são gerados e armazenados em um "banco de produtos". Nesse momento o sistema possui dois PPIs armazenados, então, o usuário poder alternar entre um e outro, usando as teclas ou botões do aplicativo, e visualizar os PPIs instantaneamente. Dessa forma, a medida que o usuário vai navegando e interagindo com os dados, o sistema vai armazenando produtos de visualização que passam a estar disponíveis instantaneamente. Ou seja, o tempo de processamento (cerca de 200 ms) não é mais necessário. Nas seções seguintes são apresentadas algumas ferramentas disponíveis no sistema RADEX.

4.1 Navegação espaço-temporal

Ao analisar dados de diferentes radares, contendo varreduras com horário diferentes, é desejável que o usuário possa avançar ou retroceder no tempo, para um mesmo radar (navegação temporal), ou alternar entre diferentes elevações de um mesmo volume de dados (navegação espacial). Para permitir esse tipo de interação, evitando o reprocessamento de dados, foi desenvolvido um mecanismo de gerenciamento da navegação utilizando a estrutura de dicionários do Python. A ideia é guardar informações chave sobre os dados processados e acessar seus produtos conforme o usuário solicitar. Por exemplo, se o usuário abrir um conjunto de dados provenientes de diferentes radares com várias datas para cada radar. Então, ao pressionar as setas direita e esquerda, no teclado, o usuário visualiza os dados desse mesmo radar na data e horário imediatamente posterior ou anterior, respectivamente. Ao pressionar um botão na tela, muda o radar. As setas "para cima" e "para baixo" alternam as elevações imediatamente superior ou inferior, respectivamente. Um dicionário Python para cada radar é usado para guardar vários tipos de informações, incluindo *sprites* do Pygame e matrizes de dados. Antes de processar os cálculos para gerar algum produto é feita uma busca nos dicionários para verificar se tal operação já foi feita, caso já tenha sido feita, o produto já processado é mostrado.

4.2 Corte vertical arbitrário

Sobre um PPI, o usuário pode traçar um segmento de reta arbitrário e rapidamente visualizar um corte vertical no volume de dados (figura 5). Isso permite que o analista explore o interior do volume de dados e possa obter informações sobre a forma de uma célula de tempestade, por exemplo, e com isso avaliar a situação para uma tomada de decisão. Adicionalmente, ao passar a seta do mouse sobre a linha do corte, é mostrada uma linha vertical (em vermelho na tela superior, à direita na figura 5) na imagem do corte, o que permite ter uma percepção acurada da posição sobre o mapa da projeção da linha vertical. Também é mostrado dinamicamente, ao desenhar a linha do corte, o tamanho correspondente (em km) do corte sobre o mapa. Na figura 5 também é possível ver o resultado do *zoom* sobre o PPI.

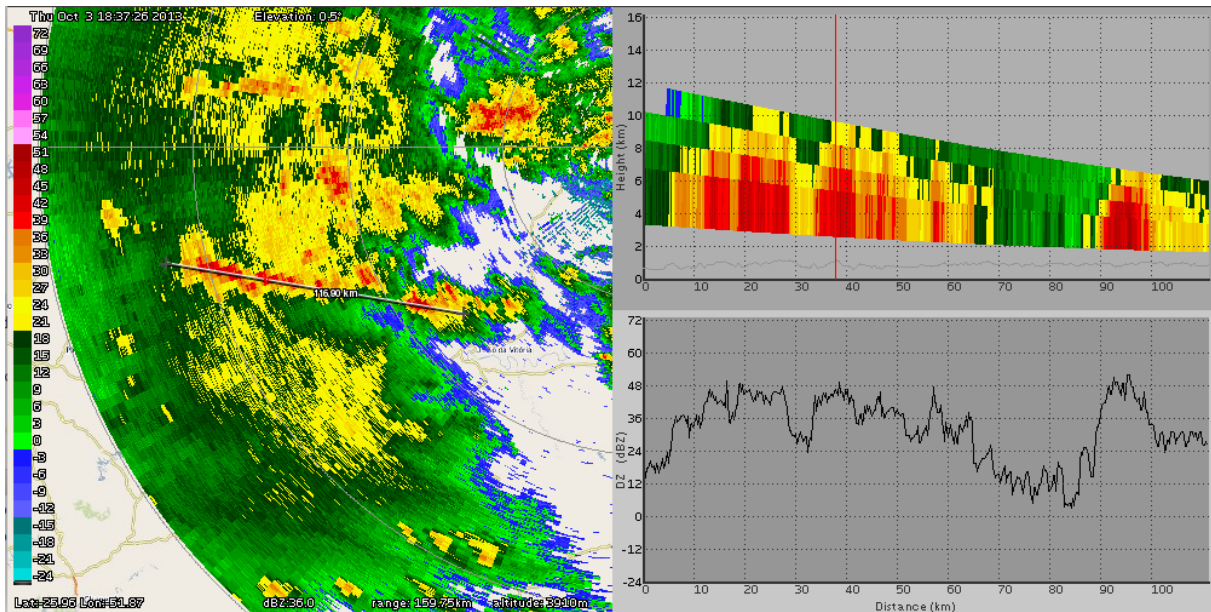


Figura 5: Resultado de um corte vertical arbitrário no volume de dados.

4.3 Gráfico de dispersão

Para visualizar correlações entre variáveis, foi adicionado uma ferramenta para gerar o gráfico da dispersão. O gráfico da dispersão trata-se de comparar duas variáveis diferentes numa região determinada pelo usuário, mostrada na figura 6. Junto com o gráfico de dispersão, é realizado o perfil vertical dos dados, onde uma média dos valores da variável é plotada a em relação à altitude no volume de dados.

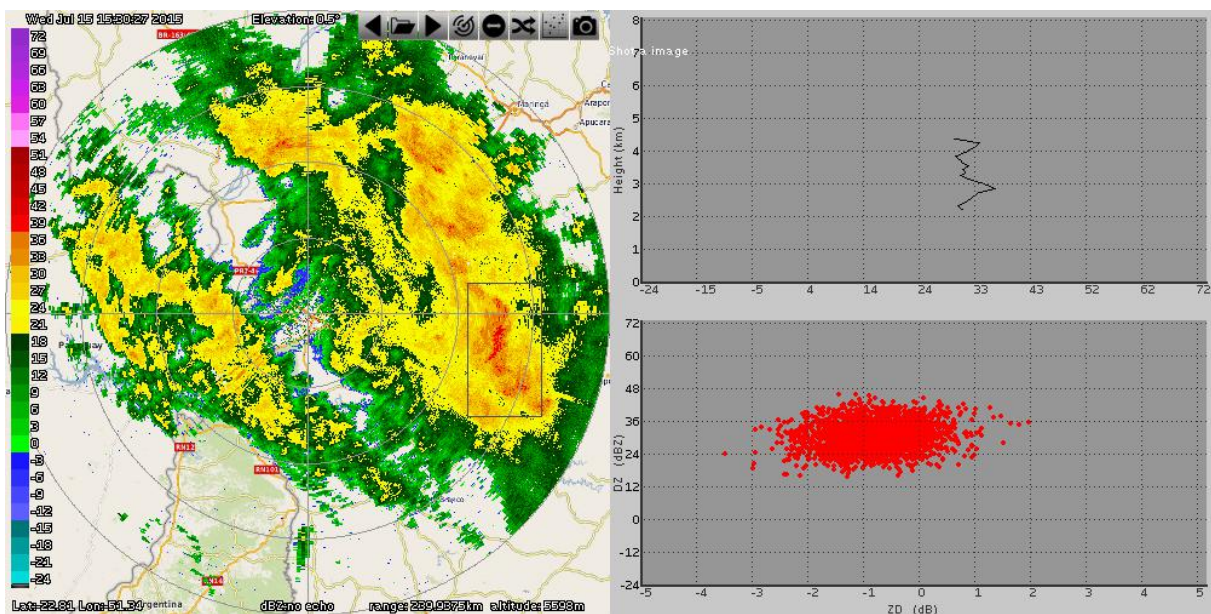


Figura 6: Comparação feita sobre DBZH e ZDR

4.4 A "função cortina"

Como alternativa à visualização lado a lado para comparação de duas variáveis (além de outras finalidades) foi desenvolvida a "função cortina" (nome dado devido à semelhança com o abrir e fechar de uma cortina). Basicamente, o efeito é obtido pela sobreposição de duas imagens, em que uma linha vertical divide a visualização, à direita é mostrada a imagem de uma variável e à esquerda de outra. A linha pode ser movimentada com o mouse lembrando o movimento de abrir e fechar uma cortina. Essa ferramenta tem se mostrado muito útil para verificar a incidência de chuva em um região específica de interesse, por exemplo, verificar se uma célula de tempestade encontra-se sobre uma rodovia, visualizar a topografia sob os dados ou comparar duas variáveis diferentes (figura 7). A função é acionada segurando o botão do mouse pressionado sobre o botão na tela correspondente à nova variável a ser visualizada e, então, movimentando o mouse.

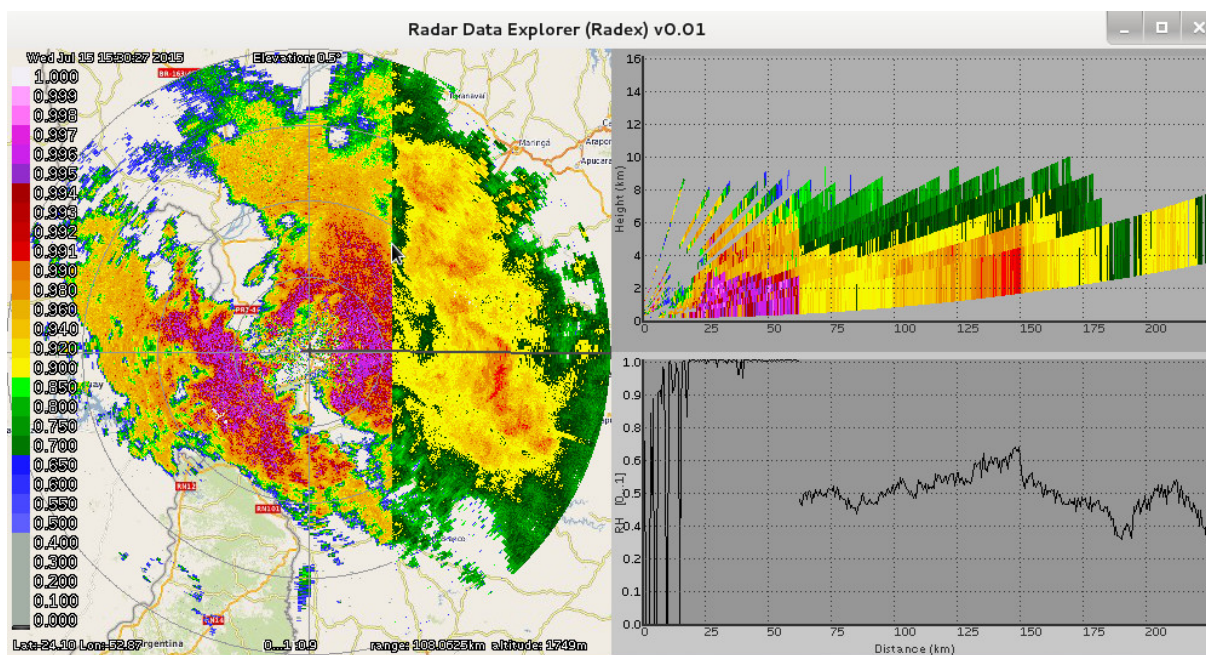


Figura 7: Usando cortina para comparar RHOHV e DBZH

4.5 Filtros para controle de qualidade

De forma integrada ao sistema de visualização foram implementados filtros para o controle de qualidade dos dados de modo que o analista possa ligar e desligar combinando com as outras formas de visualização disponíveis. Os dados adquiridos a partir de radares meteorológicos são comumente afetados 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, a partir de estudos sobre técnicas utilizadas para filtrar estes dados (Steiner and Smith, 2002; Dixon and Seed, 2014; Friedrich et al., 2006), foi desenvolvido um algoritmo a partir da modificação do algoritmo de Steiner M. e Smith J. (2002). A modificação foi no sentido de simplificação para que o processamento ocorra mais rapidamente e possa atender às questões de interatividade e usabilidade. O resultado de uma aplicação do filtro é mostrado na figura 8.

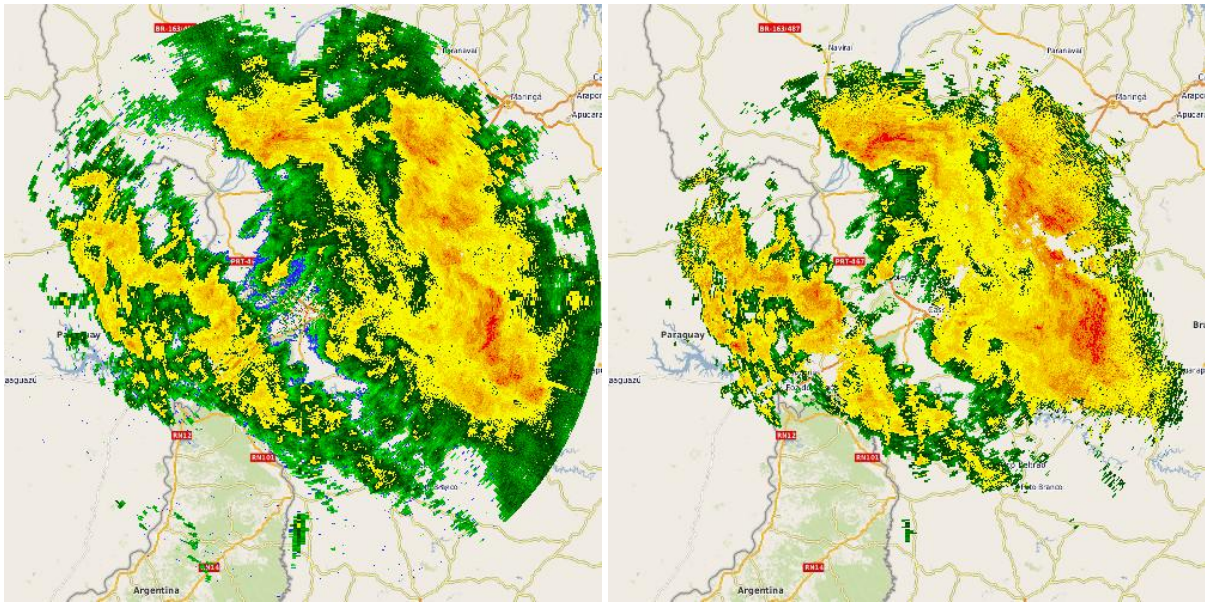


Figura 8: Imagem original (esquerda) e imagem filtrada via algoritmo desenvolvido (direita).

4.6 Cortes radiais

Outra funcionalidade disponível é a visualização interativa de cortes radiais, em que o usuário pode visualizar de forma dinâmica e interativa um conjunto de cortes radiais pré-processados. Basicamente o usuário seleciona um setor circular no PPI e os cortes verticais na direção radial são gerados. Então o usuário pode passar o mouse e alternar a visualização dinamicamente e dessa forma, explorar o interior do volume de dados (Figura 9). A visualização funciona como uma animação, controlada pelo movimento do mouse.

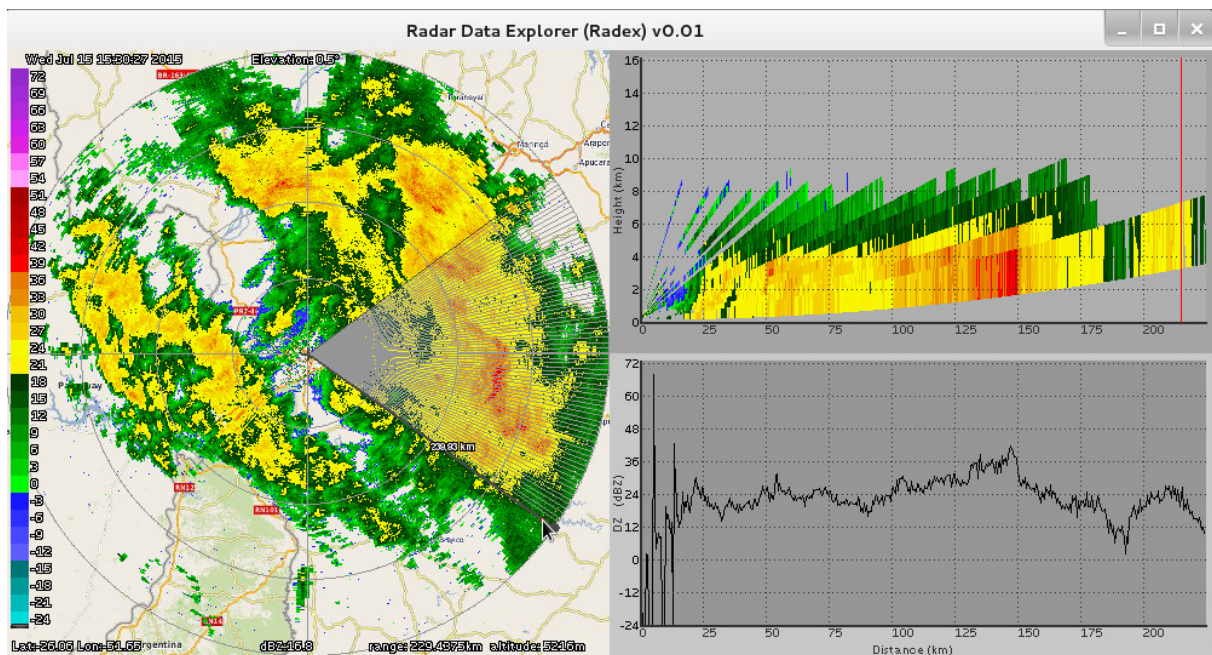


Figura 9: Seleção da área para realização do corte radial

4.7 Visualização 3D de um volume selecionado

Por fim, uma funcionalidade que ainda encontra-se em desenvolvimento é a seleção de uma região de interesse e a visualização 3D do volume por meio de técnicas de visualização científica para campos escalares. Dentre as técnicas que estão sendo implementadas estão a visualização volumétrica e a extração de isosuperfícies. Para a visualização 3D tem sido utilizado a biblioteca PyVTK a qual é permite o uso de Python para acessar funções da biblioteca VTK (www.vtk.org).

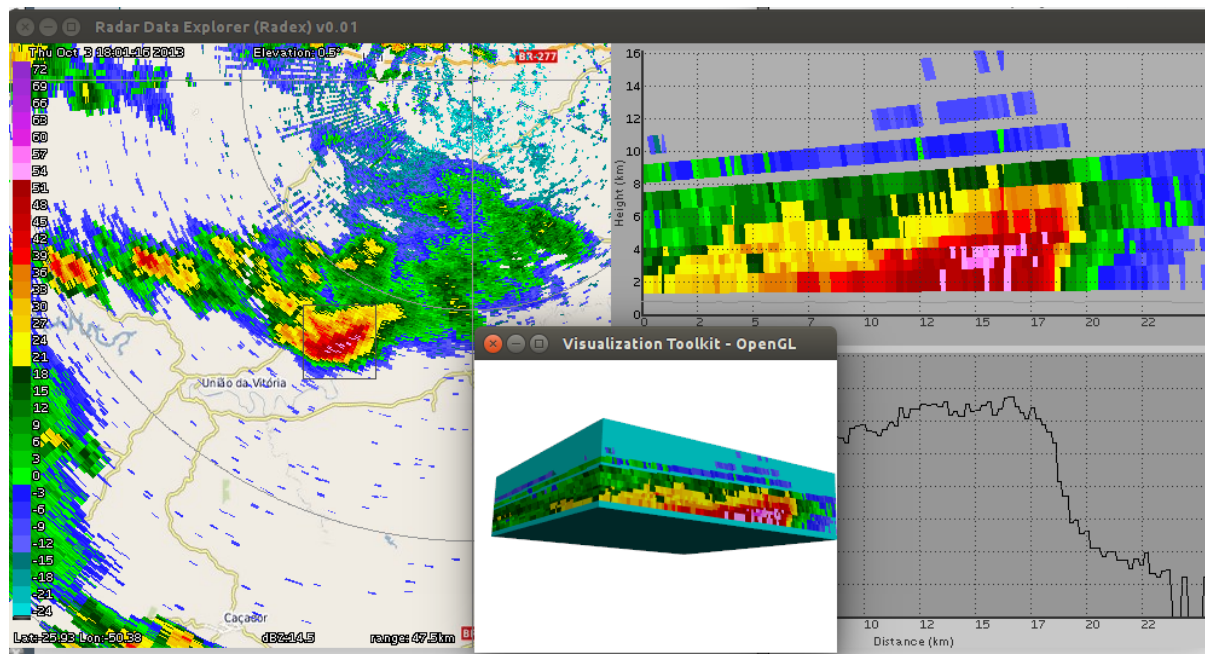


Figura 10: visualização 3D de um volume selecionado.

5 CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentado o sistema RADEX, para visualização de dados de radares meteorológicos, e suas principais funcionalidades. Muitas das ferramentas desenvolvidas oferecem funcionalidades personalizadas, não disponíveis em outros programas, comerciais ou não. Os parâmetros para algoritmos de visualização implementados são definidos por meio de interação direta com uma representação dos dados. As estratégias adotadas para a geração das representações dos dados, implementadas fazendo uso intensivo de estruturas de dados baseados em vetores e matrizes, permitiu obter uma interação fluida, com desempenho satisfatório. A experiência do desenvolvimento da ferramenta de visualização apresentada neste trabalho, mostrou que a linguagem Python pode ser uma boa escolha para desenvolvimento de sistemas gráficos que combinam processamento numérico e interatividade. Embora em computação científica linguagens como C/C++ e Fortran sejam mais utilizadas, linguagens como Python oferecem vantagens em termos de produtividade e alcançando desempenho satisfatório para muitas aplicações. As facilidades e conveniências providas pelas bibliotecas e módulos disponíveis em Python, permitem a criação de programas de forma mais rápida e com melhor performance.

REFERÊNCIAS

CILAMCE 2015

Proceedings of the XXXVI Iberian Latin-American Congress on Computational Methods in Engineering
Ney Augusto Dumont (Editor), ABMEC, Rio de Janeiro, RJ, Brazil, November 22-25, 2015

- Cao, Q., Yeary, M., & Zhang, G., 2012. Efficient ways to learn weather radar polarimetry. *Education, IEEE Transactions on*, , vol. 55, n. 1, pp. 58–68.
- Coelho, F. C., 2007. *Computação Científica com Python*. Flávio Codeço Coelho, Petrópolis, RJ, ed. 1.
- Dixon, M. & Seed, A., 2014. Developments in echo tracking - enhancing titan. *8th European Conference on Radar in Meteorology and Hydrology*, , vol. , pp. 1–14.
- Friedrich, K., Hagen, M., & Einfalt, T., 2006. A quality control concept for radar reflectivity, polarimetric parameters, and doppler velocity. *Journal of Atmospheric and Oceanic Technology*, , vol. 23, n. 7, pp. 865–887.
- Islam, T. & Rico-Ramirez, M. A., 2013. An overview of the remote sensing of precipitation with polarimetric radar. *Progress in Physical Geography*, , vol. , pp. 0309133313514993.
- Jones, R., 2005. Rapid game development in python. *OpenSource Developers' Conference*, , vol. , pp. 84–90.
- McGugan, W., 2007. *Beginning Game Development with Python and Pygame*. ed. 1.
- Rinehart, R. E., 2004. *Radar for Meteorologists*. Rinehart Publications, Columbia, ed. 4.
- Silveira, R., 1998. *Recognition of clutter in weather radars using polarization diversity information and artificial neural networks*. PhD thesis, University of Essex.
- Steiner, M. & Smith, J. A., 2002. Use of three-dimensional reflectivity structure for automated detection and removal of nonprecipitating echoes in radar data. *J. Atmos. Oceanic Technol.*, , vol. , n. 19, pp. 673–686.
- Tory, M. & Möller, T., 2004. Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics*, , vol. 10, n. 1, pp. 72–84.