

Visualização Científica Interativa de Dados de Radar Meteorológico

Tiago Martinuzzi Buriol

Instituto Federal do Paraná - IFPR

82530-230, Curitiba, PR

E-mail: tiago.buriol@ifpr.edu.br

Cesar Beneti

Instituto Tecnológico Simepar

81531-980, Curitiba, PR

E-mail: beneti@simepar.br

Resumo: A visualização é um importante recurso de apoio para investigações científicas. Por meio de técnicas de visualização científica, pesquisadores de diversas áreas podem extrair visualmente informações úteis a partir de grandes conjuntos de dados complexos. No entanto, muitos aspectos, incluindo fatores humanos e computacionais, apresentam desafios consideráveis para o desenvolvimento de ferramentas de visualização eficientes e eficazes. Questões como desempenho computacional, interface para interação e percepção visual, precisam ser considerados no projeto. Neste trabalho é apresentado o desenvolvimento de um visualizador de dados de radar meteorológico cujo foco é a facilidade de uso e de interação. O objetivo tem sido propor, desenvolver e experimentar diferentes mecanismos de exploração visual, buscando encontrar soluções que minimizem a carga cognitiva demandada pelo uso da ferramenta e maximizem a compreensão que o analista pode obter sobre os dados. Para isso tem sido utilizada a linguagem de programação Python, juntamente com os pacotes Numpy e Pygame, os quais permitem obter um equilíbrio entre produtividade e desempenho final do aplicativo. As estratégias utilizadas para a implementação dos algoritmos de processamento numérico e de visualização científica, bem como as vantagens e limitações encontradas são apresentadas e discutidas.

1. INTRODUÇÃO

Diferentes atividades científicas e profissionais, cada vez mais, fazem uso de grandes volumes de dados gerados diariamente a partir de instrumentos de medição e de simulações computacionais. Assim, ferramentas computacionais para visualização científica tornaram-se instrumentos indispensáveis em áreas que fazem uso de processamento de dados e de métodos numéricos. A visualização provê a cientistas um importante suporte cognitivo, que melhora a percepção sobre os dados e facilita a construção de um modelo conceitual do fenômeno em estudo [4].

No entanto, produzir representações gráficas capazes de revelar informações úteis contidas em grandes conjuntos de dados complexos não é uma tarefa simples [1]. A escolha das técnicas e a definição dos parâmetros de entrada para os algoritmos de visualização científica são aspectos críticos. Além disso, questões relacionadas à usabilidade e ao desempenho computacional podem impor diversas limitações. Neste sentido, ferramentas de visualização eficientes e eficazes devem prover mecanismos que permitam incorporar formas simples e fluidas de interagir com os dados e com o processo de visualização [2].

Desenvolver ferramentas de visualização requer conhecimento em diferentes áreas, tais como ciência da computação, matemática e *design* de interfaces. Algumas das dificuldades frequentemente encontradas são relativas à demanda por processamento em tempo real e às limitações na manipulação dos dados e controle dos algoritmos de visualização [6]. É desejável que as ferramentas sejam simples usar e que ofereçam mecanismos para interagir com a representação dos dados, permitindo ao cientista criar e verificar hipóteses. Mecanismos interação intuitivos proporcionam a redução da carga cognitiva demandada pelo uso da ferramenta, permitindo ao usuário focar sua atenção nos dados e no fenômeno em estudo [8].

Nesse sentido, a visualização não é apenas um *output* mas um processo cíclico, envolvendo refinamento, manipulação e investigação (fig. 1).

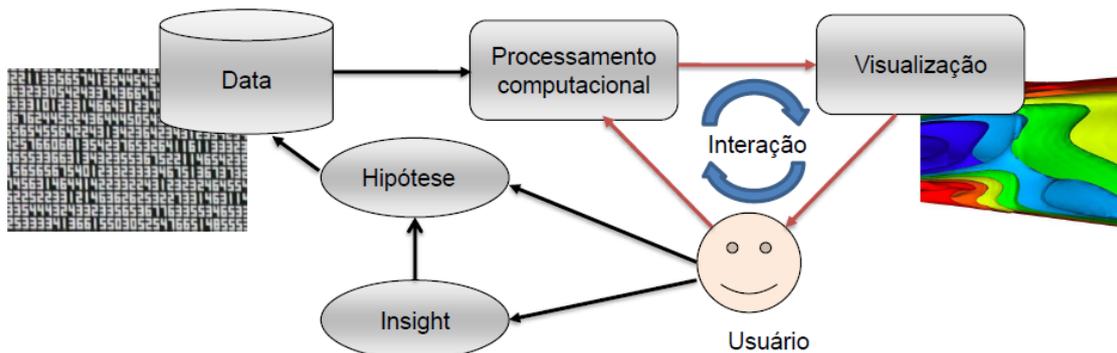


Figura 1 - Diagrama do processo interativo de visualização científica.

Nesse contexto, a usabilidade é um aspecto de grande relevância, pois ao inspecionar visualmente um conjunto de dados, é importante que o analista possa se concentrar nos dados ao invés de preocupar-se em como utilizar o sistema. Então, além do sistema prover uma interface fácil de usar, também se espera que o sistema responda rapidamente às solicitações do usuário, ou seja, o desempenho dos algoritmos numéricos e de visualização deve ser considerado.

Atualmente existem diversas bibliotecas e *toolkits* que permitem acelerar o desenvolvimento de sistemas de visualização científica, reduzindo consideravelmente o esforço de programação. Muitas destas bibliotecas dispõem de algoritmos para visualização baseados em técnicas bem conhecidas tais como mapeamento por cores, glifos, geração de cortes e contornos. No entanto, os controles para aplicação das técnicas de visualização nem sempre são fáceis de utilizar. Pelo contrário, frequentemente a dificuldade para lidar com a interface do sistema acaba tomando do usuário mais atenção do que o desejado.

Tradicionalmente, os sistemas para visualização de dados meteorológicos têm mantido alguns padrões de interface com usuário e de funcionalidades, os quais têm persistido por muitos anos. Os projetistas e programadores desses sistemas têm dado pouca importância para questões relativas à interface, especialmente com relação à usabilidade e à concepção de novas formas de explorar visualmente os dados.

Buscou-se, durante este trabalho, desenvolver um sistema de visualização, para dados de radares meteorológicos, orientado ao usuário. A ideia principal foi criar e experimentar funcionalidades diferenciadas, focadas na facilidade de uso e alta interatividade. O objetivo tem sido oferecer um conjunto controles que permitam ao usuário obter de forma simples, rápida e objetiva, diferentes produtos de visualização 2D. Com isso pretende-se minimizar a sobrecarga cognitiva demandada pela interface e proporcionar um meio eficiente de se obter um modelo conceitual da estrutura 3D dos dados.

Neste artigo, é apresentado o desenvolvimento de uma ferramenta de visualização de dados de radar meteorológico, com interface projetada para ser simples e intuitiva, permitindo ao analista explorar os dados, interagindo diretamente em uma representação destes dados. Foram desenvolvidos controles gráficos para os parâmetros de entradas dos algoritmos de visualização científica de forma que o usuário possa modificar facilmente esses parâmetros e rapidamente visualizar o resultado desejado.

2. VISUALIZAÇÃO DE DADOS 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 a partir deste conjunto de dados, complexos e multidimensionais, é um desafio.

Os dados obtidos por radares meteorológicos são distribuídos espacialmente em uma grade descrita por meio de um sistema de coordenadas esféricas. 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.

Um dos radares meteorológicos operado pelo Instituto tecnológico SIMEPAR, no estado do Paraná, gera 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 antena (inclinação vertical). O volume de dados gerado ao final de um ciclo completo corresponde a uma sequência de 360 varreduras azimutais com 14 diferentes elevações da antena, com 800 pontos 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. Cada um destes pontos podem conter dezenas de variáveis associadas a ele. A cada 10 minutos é gerado um volume de dados.

A partir desses dados são obtido “produtos”, que são imagens 2D representando uma ou mais variáveis, podendo ser a projeção de uma elevação (PPI – *Plan Position Indicator*) ou em um corte horizontal do volume (CAPPI – *Constant Altitude Plan Position Indicator*). Dentre as dezenas de variáveis que um radar pode medir, destaca-se a refletividade (Z) 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 zero e 63 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.

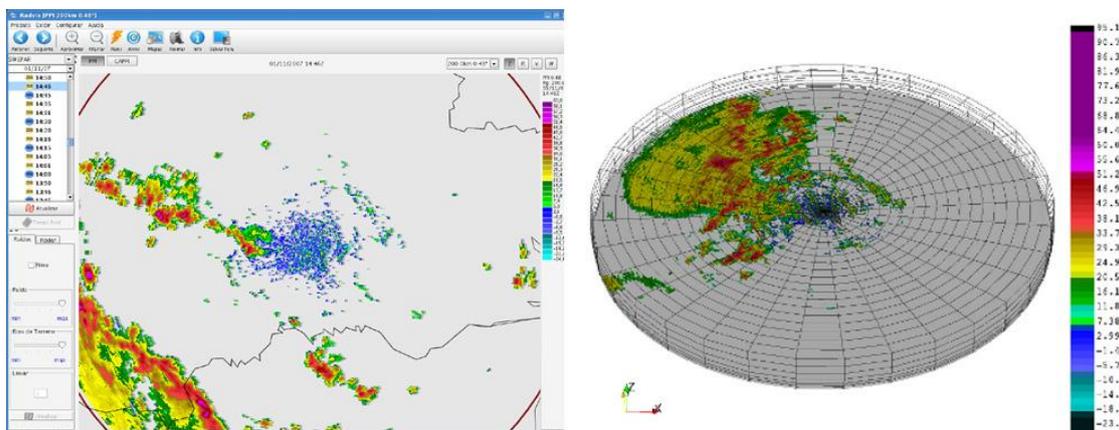


Figura 2 - Uma ferramenta utilizada para visualizar produtos do radar e a estrutura original do volume de dados em uma grade em coordenadas esféricas.

Em geral, a abordagem adotada é converter previamente os dados em uma grade cartesiana com resolução previamente definida, gerar e armazenar os produtos como imagens e disponibilizar consulta. Diversos produtos, previamente escolhidos, podem ser gerados

automaticamente em um servidor e ficarem armazenados e disponíveis. 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 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. 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 disponível facilmente. Um sistema de fácil utilização e ofereça ao analista maior liberdade para controlar e gerar produtos de seu interesse, seria de grande valor.

Neste trabalho buscou-se desenvolver ferramentas para visualização interativa de dados de radar que sejam fáceis de usar, suficientemente rápidas e diferentes das abordagens convencionais. A ideia é explorar a evolução dos *hardwares* gráficos e em novos paradigmas de interação para criar novas ferramentas de visualização orientadas ao usuário, que possam oferecer um ganho no processo de análise dos dados. Pretende-se gerar os produtos de visualização a partir dos dados originais, de forma que a estrutura original dos dados não seja perdida. Os produtos devem ser obtidos a partir de solicitações que o usuário faz interagindo graficamente sobre uma representação dos dados, a medida que os produtos vão sendo gerados, eles são armazenados e o usuário pode “navegar” alternando rapidamente entre um e outro a fim de analisá-los.

3. DESENVOLVIMENTO DE UM VISUALIZADOR INTERATIVO

Buscando um equilíbrio entre esforço demandado para o desenvolvimento e o desempenho final do aplicativo, optou-se por utilizar a linguagem Python juntamente com os pacotes Numpy e Pygame. Esta linguagem sido indicada como uma boa alternativa para aplicações científicas e desenvolvimento de pesquisas [3], [5]. Utilizando o pacote Numpy, é possível escrever algoritmos para processamento numérico com relativa facilidade e cujo desempenho tem se mostrado satisfatório para este propósito. Pygame proporciona uma maneira facilitada de desenvolver sistemas visuais interativos, oferecendo acesso direto *hardware* gráfico, além do mouse e teclado [7].

Este conjunto de bibliotecas em Python acelerou o desenvolvimento de um sistema de visualização altamente interativo, focado na facilidade de uso e na velocidade de resposta do sistema às solicitações do usuário. Buscou-se com as funcionalidades implementadas proporcionar um meio efetivo de exploração e de análise visual dos dados. 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 solta o botão o algoritmo que processa os dados para gerar o corte é executado e o corte é apresentado em uma fração de segundo.

A estratégia adotada foi manter um conjunto de dados na memória 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 do sistema, supondo que o usuário queria ver um vorte vertical, bastaria traçar uma linha (clique, arrastar e soltar) com o mouse. Ou então, para ampliar

e transladar gradualmente a imagem basta utilizando a rodinha do mouse, assim que o movimento cessa a imagem é atualizada a partir dos dados originais.

A abordagem utilizada para conferir uma resposta rápida na geração do corte vertical foi calcular a posição espacial de cada pixel da imagem que representa o corte e, então, buscar nos dados o valor da variável correspondente àquela 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 desempenho satisfatório. 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 para nosso propósito.

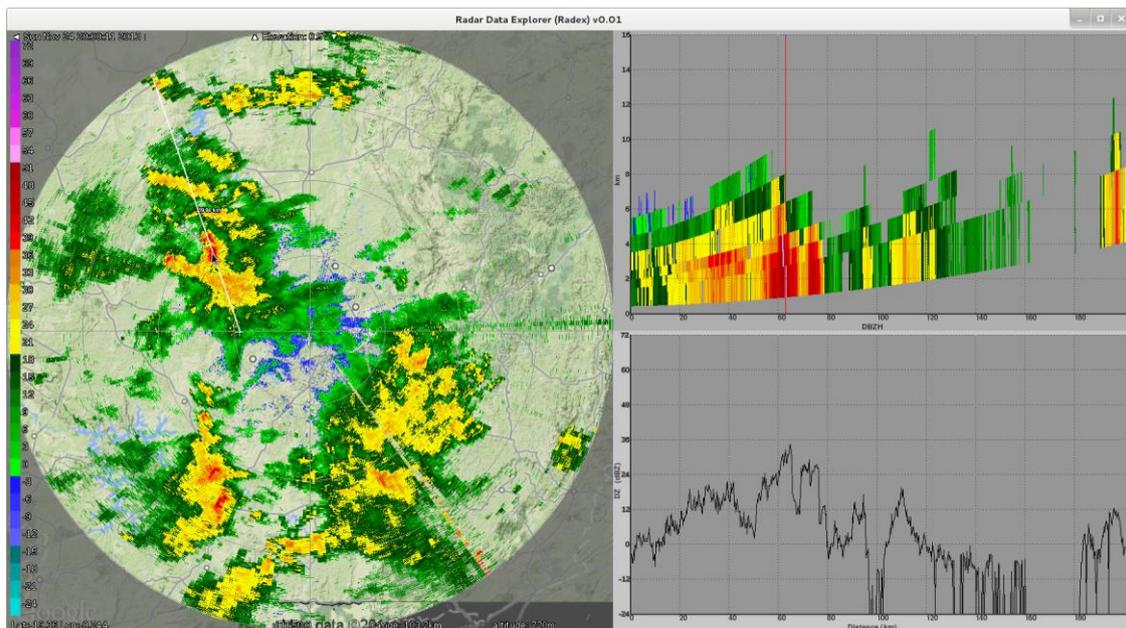


Figura 3 – Tela do visualizador de dados desenvolvido

Outra característica do aplicativo desenvolvido é que a representação dos dados pode ser ampliada (zoom in) interativamente, em tempo real, utilizando a rodinha do mouse e mantendo o cursor 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, a representação dos dados é atualizada, a partir dos dados originais, mostrando-os em sua estrutura polar original. É comum, em aplicativos para visualizar este tipo de dado, a conversão da estrutura polar em uma grade cartesiana, então, ao ampliar uma área a resolução e configuração original é perdida. 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).

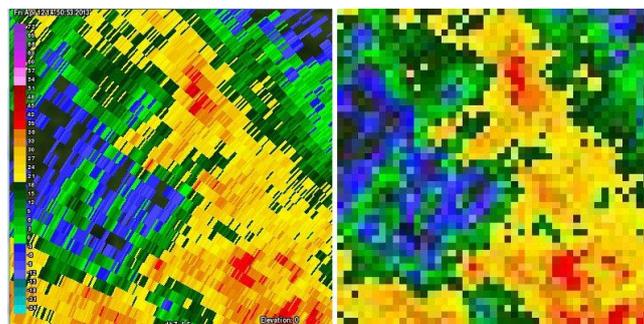


Figura 4 – Visualização ampliada dos dados mantendo a estrutura polar (esquerda) e visualização ampliada dos dados previamente convertidos em estrutura cartesiana.

As estruturas de matrizes e as operações algébricas disponível em Numpy permitem em um única linha de código, por exemplo, realizar operações sobre todos os elementos de uma matriz de forma otimizada. 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(500) , numpy.arange(500))
R = numpy.sqrt(X**2 + Y**2)
T = numpy.arctan2(Y,X)
```

4. CONSIDERAÇÕES FINAIS

Como resultado prático deste trabalho foi obtido um aplicativo para visualização de dados de radares meteorológicos que provê 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 um 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 facilitada.

REFERÊNCIAS

- [1] C. Johnson, Top scientific visualization research problems, Computer Graphics and Applications, IEEE , vol.24, no.4, pp.13,17, 2004.
- [2] D.F. Keefe, T. Isenberg, Reimagining the Scientific Visualization Interaction Paradigm, Computer , vol.46, n.5, pp.51,57, (2013).
- [3] F. C. Coelho, Computação Científica com Python. 1th. Ed. [s.l.]: flávio codeco coelho, 2007. 305 p.
- [4] H. Wright, "Introduction to Scientific Visualization", Springer London, 2007.
- [5] L.J. Wei-Bing, 2012: Why Python Is the Next Wave in Earth Sciences Computing. Bull. Amer. Meteor. Soc., 93, 1823–1824.
- [6] Q. Wen. 3D Interaction with Scientific Data: An experimental and perceptual approach. PhD thesis. Technische Universiteit Eindhoven, 2008.
- [7] R. Jones. Rapid Game Development in Python. OpenSource Developers' Conference (2005), 84-90.
- [8] M.Tory, T. Moller, T., "Human factors in visualization research," Visualization and Computer Graphics, IEEE Transactions on , vol.10, no.1, pp.72,84, Jan.-Feb. 2004