

Vistas Explodidas

Fábio Markus Miranda*
PUC-Rio

Abstract

Abstract

Resumo

Resumo

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry & Object Modeling; I.3.8 [Computer Graphics]: Applications;

Keywords: exploded view illustration, interactive visualization

1 Introdução

Um problema típico na visualização de modelos 3D complexos é que as características mais interessantes podem estar obstruídas por outras partes menos importantes. Para minimizar esse problema, várias técnicas de ilustração foram desenvolvidas ao longo dos anos, e alteram o nível de abstração visual ou então a disposição espacial de partes do modelo. Técnicas de visibilidade inteligente [Viola and Gröller 2005] levam em consideração a relevância dos objetos e suas características, e não apenas o seu posicionamento no espaço. Além disso, espera-se uma familiaridade do observador com os objetos; com base em sua experiência, o observador pode completá-lo mentalmente.

Uma das técnicas de visibilidade inteligente é a *vista explodida*. O seu conceito básico é modificar o posicionamento de partes de um modelo para facilitar a visão de características importantes. Ao contrário de técnicas como *ghosted view* e *cut away view*, a vista explodida permite a visualização de um modelo sem que se perca qualquer tipo de informação sobre ele, já que não há nenhum tipo de transparência ou remoção de partes mais externas. O usuário pode então, de forma fácil, reconstruir mentalmente o modelo e assim ter uma visão global e integrada do mesmo.

Este trabalho implementa uma técnica de vista explodida apresentada em [Li et al. 2008]. Toda a informação necessária para explodir um modelo são armazenadas em um grafo de explosão, que dita a ordem e a direção que cada parte deverá ser explodida. O objeto principal é dar uma visão geral sobre o tema, explicar o trabalho escolhido, assim como a sua implementação, discutir dificuldades e facilidades e apresentar possíveis propostas para trabalhos futuros.

Este trabalho está dividido da seguinte forma: na Seção 2 são apresentados os trabalhos relacionados. A Seção 3 apresenta uma visão geral do sistema de vista explodida, sendo que detalhes de implementação são discutidos na Seção 4. Os resultados são mostrados na Seção 5. A conclusão e possíveis trabalhos futuros estão na Seção 6.

2 Trabalhos Relacionados

O trabalho apresentado em [Viola and Gröller 2005] faz um levantamento de diversas técnicas desenvolvidas com o objetivo de facilitar a visualização de modelos complexos e, por ser bastante abran-

gente, serve como uma base para o estudo da área. Em [Elmqvist and Tsigas 2008], os autores propõem uma taxonomia para classificar essas diversas técnicas, apresentando os pontos positivos e negativos de cada uma.

Os trabalhos relacionados a visão explodida podem ser divididos em dois grupos. Uma primeira série de trabalhos busca criar vistas explodidas com base em dados volumétricos. O primeiro a explorar essa área foi [Carpendale et al. 1996], onde os autores os autores descrevem a técnica *visual access distortion*, que busca criar um caminho de visualização sem obstruções até a parte de maior interesse. Os trabalhos [Bruckner 2006] e [McGuffin et al. 2003] apresentam propostas para a visualização de dados volumétricos com vistas explodidas.

Uma outra vertente na pesquisa de vistas explodidas é baseada na explosão de modelos 3D complexos, não baseados em dados volumétricos. O trabalho [Agrawala et al. 2003] é uma referência inicial neste assunto e apresenta um sistema para geração de instruções de montagem, mas não propõe nada relacionado a vistas explodidas em específico. Esse tema só é abordado no artigo [Li et al. 2008], que busca justamente expandir as instruções de montagem e criar um sistema para a visualização de vistas explodidas, dada a semelhança entre os dois assuntos.

Outro trabalho também relacionado a vistas explodidas é [Niedermaier et al. 2003], onde os autores apresentam um sistema para a geração automática de vistas explodidas de ambientes arquitetônicos.

3 Visão Geral

O sistema implementado neste trabalho (e baseado no que foi proposto em [Li et al. 2008]) recebe como entrada um modelo 3D composto por diversas partes. Com base no modelo, é construído uma estrutura (*grafo de explosão*) que encapsula todas as informações pertinentes a sua explosão. Todo o sistema pode ser então resumido em três partes:

- Representação da explosão na forma de um grafo (Seção 3.1).
- Geração do grafo (Seção 3.2).
- Visualização do grafo (Seção 3.3).

3.1 Grafo de explosão

O grafo de explosão nada mais é do que uma estrutura que dita como cada parte deverá ser explodida. O relacionamento entre os nós irá determinar a ordem em que as partes são explodidas, sem que ocorra nenhuma violação das restrições de movimento (ou seja, uma parte atravessar outra).

Considerando uma parte p_i do modelo 3D, e sabendo que esta parte é obstruída pelo modelo p_j , então temos que esse último será um nó filho de p_i . No momento da explosão, p_j será explodido antes de p_i , garantindo que não haverá nenhuma violação nas restrições de movimento.

O grafo de explosão também será responsável por armazenar a direção em que cada parte deverá ser explodida e a distância que ela deverá percorrer.

*e-mail: fmiranda@inf.puc-rio.br

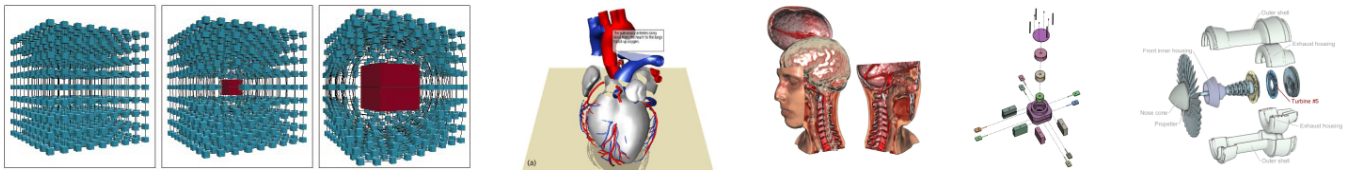


Figura 1: Trabalhos relacionados ([Carpendale et al. 1996], [McGuffin et al. 2003], [Bruckner 2006], [Agrawala et al. 2003], [Li et al. 2008]).

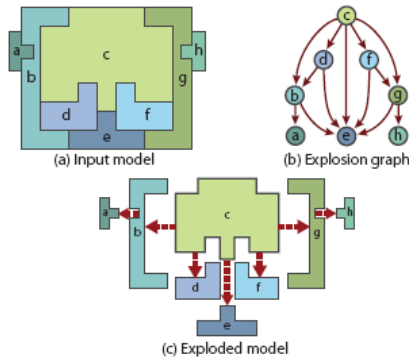


Figura 2: Grafo de explosão ([Li et al. 2008])

3.2 Geração do grafo de explosão

A geração do grafo é feita através do seguinte algoritmo iterativo, e que foi originalmente proposto em [Agrawala et al. 2003]:

```

Dados: S: conjunto com as partes ativas do modelo (ainda não
inseridas no grafo)
Dados: P: sub-conjunto de S de partes que não possuam
restrição em pelo menos uma direção
enquanto  $S \neq \emptyset$  faça
    Determina P;
    para cada  $p_i \in P$  faça
        Determina a distância mínima  $d_i$  que  $p_i$  teria que
mover para sair do bounding box das partes em
contato com  $p_i$ ;
    fim
     $p_{min} = \text{Min}_{p_i \in P}(d_i)$ ;
     $S = S - p_{min}$ ;
fim

```

Algoritmo 1: Geração do grafo de explosão

As partes do modelo são inicialmente inseridas em um conjunto S, de partes ativas, que ainda não foram inseridas no grafo. Um sub-conjunto P é então encontrado, com partes que não possuem restrição em pelo menos uma direção. Para cada parte p_i em P é determinada a distância mínima para que a parte saia do *bounding box* das partes em que contato. A parte com menor distância é removida de S e adicionada ao grafo, junto com arestas ligando ela a todas as outras partes que estão em contato (e que já estavam no grafo). O algoritmo termina quando todas as partes já tiverem sido inseridas no grafo. A maneira com que o algoritmo constrói o grafo de explosão garante que o relacionamento entre os nós respeite as restrições de movimento.

3.2.1 Partes cercadas

Um caso importante que é tratado pelo sistema é quando há "partes cercadas" (como na Figura 3).

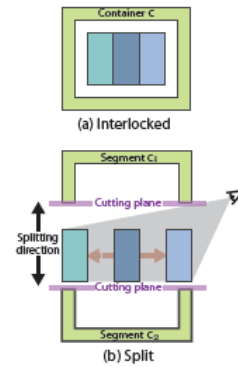


Figura 3: Partes cercadas ([Li et al. 2008])

Nesses casos, há a necessidade de se partir o recipiente (*container*) em duas partes (c_1 e c_2). O particionamento ocorrerá em um dos três possíveis eixos de explosão. Para cada eixo de explosão (x, y e z), será determinada uma distância d que corresponde ao deslocamento das partes c_1 e c_2 de tal modo que as seguintes condições sejam respeitadas:

1. c_1 e c_2 estão fora do *bounding box* das partes cercadas.
2. As partes cercadas estão visíveis.

Das três distâncias calculadas (d_x , d_y , d_z), o eixo de explosão será aquele correspondente à menor distância d .

Como se pode perceber através da condição 2, a determinação do eixo de explosão de um recipiente depende da posição da câmera. É necessário então gerar diversos grafos, cada um com uma posição de câmera diferente. O artigo de referência calcula 26 diferentes grafos, correspondentes às faces, arestas e cantos de um cubo, e faz uma troca automática de acordo com a posição da câmera do usuário na hora da visualização.

3.3 Visualização

Para visualizar o modelo explodido, o usuário deve escolher uma de suas partes, que estará visível assim que todos os seus descendentes sejam explodidos. Para que as partes não violem restrições, as explosões são feitas seguindo uma ordem topológica reversa, ou seja, os descendentes são explodidos primeiro. O colapso das partes é feita de maneira oposta: os descendentes são colapsados por último.

Um problema que pode surgir durante a visualização é a obstrução da visão de uma determinada parte. As informações de obstrução

não são guardadas no grafo de explosão, e são calculadas durante a própria visualização. Para impedir que partes de interesse sejam obstruídas, o seguinte algoritmo é utilizado:

```

Dados: G: Grafo de explosão
Dados: T: partes a serem explodidas (escolhidas pelo usuário)
/* Visita os nós em ordem topológica */
para cada  $p_i \in G$  e  $p_i \notin T$  faça
    /* Verifica se  $p_i$  obstrui algum nó ascendente */
    enquanto  $p_i$  obstrui algum  $p_j$ ,  $j < i$  faça
        Move  $p_i$  na sua direção de explosão, e também seus descendentes;
    fim
fim

```

Algoritmo 2: Geração do grafo de explosão

4 Implementação

Para a implementação do sistema, foram utilizados o *OpenScene-Graph*, para a renderização e carregamento dos modelos 3D, e *PQP* para a detecção de colisões entre as partes do modelo. A Figura 4 apresenta uma visão geral dos módulos implementados, que serão discutidos posteriormente.

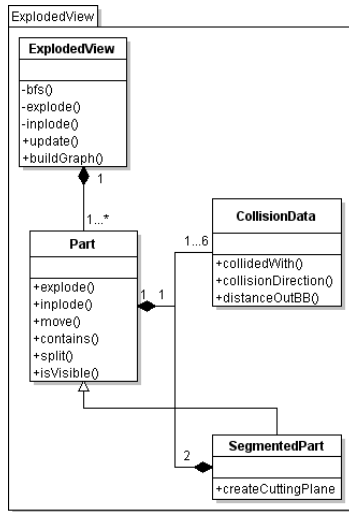


Figura 4: Visão geral da arquitetura

- **ExplodedView:** classe geral, responsável por tratar os dados de entrada (tanto o modelo 3D quanto o *input* do usuário).
- **Part:** classe que representa as partes do modelo.
- **SegmentedPart:** classe que herda de **Part** e representa os recipientes divididos.
- **CollisionData:** representa os dados de colisão.

A geração do grafo de explosão é feita logo no início da execução do programa. As partes são extraídas do modelo 3D através de um **NodeVisitor** que procura pelos nomes dos nós através de uma expressão regular. Cada nó (parte) é então associado a um novo objeto do tipo **Part**.

Para a detecção das direções obstruídas de cada parte, foi empregada duas abordagens. A primeira, explorada em [Agrawala et al. 2003], constrói um *grafo de bloqueio direcional*

Considerando que o usuário quer explodir o nó p , o sistema implementado irá executar uma busca em largura (*BFS*) a partir de p e armazenar o nível de cada nó descendente. A Figura 5 mostra os níveis montados pela busca em largura para um modelo simples e a partir da parte A.

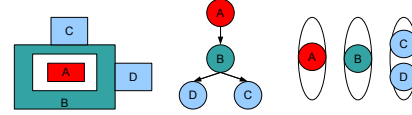


Figura 5: Exemplo de níveis da BFS

A explosão e a implosão do modelo será feita então com base nessas níveis, de forma simples:

```

Dados:  $A_i$ : níveis de um grafo,  $i < n$ 
para  $i = n..1$  faça
    | Explode nós no nível  $A_i$ ;
fim

```

Algoritmo 3: Explosão de uma parte

```

Dados:  $A_i$ : níveis de um grafo,  $i < n$ 
para  $i = 1..n$  faça
    | Implode nós no nível  $A_i$ ;
fim

```

Algoritmo 4: Implosão de uma parte

A divisão dos recipientes (*containers*) foi feita com o uso de um **ClipNode** do OSG, que irá determinar o plano de corte para a parte ser renderizada.

4.1 Características não abordadas

Algumas características do sistema proposto em [Li et al. 2008] não foram implementadas, como ...

5 Resultados

Os modelos de teste utilizados serão exibidos a seguir, destacando as características do sistema que eles procuravam explorar.

5.1 Modelo 1

5.2 Modelo 2

5.3 Modelo 3

5.4 Modelo 4

6 Conclusão e Trabalhos Futuros

Conclusão e trabalhos futuros - Falar sobre vistas explodidas na visão de informação, como nós da internet, etc. - Falar da tese que utiliza realidade aumentada.

Referências

- AGRAWALA, M., PHAN, D., HEISER, J., HAYMAKER, J., KLINGNER, J., HANRAHAN, P., AND TVERSKY, B. 2003. Designing effective step-by-step assembly instructions. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, ACM, New York, NY, USA, 828–837.
- BRUCKNER, S. 2006. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics* 12, 5, 1077–1084. Member-Groller, M. Eduard.
- CARPENDALE, M. S. T., CARPENDALE, T., COWPERTHWAIT, D. J., AND FRACCHIA, F. D. 1996. Distortion viewing techniques for 3-dimensional data. In *INFOVIS '96: Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96)*, IEEE Computer Society, Washington, DC, USA, 46.
- ELMQVIST, N., AND TSIGAS, P. 2008. A taxonomy of 3d occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 5, 1095–1109.
- LI, W., AGRAWALA, M., CURLESS, B., AND SALESIN, D. 2008. Automated generation of interactive 3d exploded view diagrams. *ACM Trans. Graph.* 27, 3, 1–7.
- MCGUFFIN, M. J., TANCAU, L., AND BALAKRISHNAN, R. 2003. Using deformations for browsing volumetric data. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, IEEE Computer Society, Washington, DC, USA, 53.
- NIEDERAUER, C., HOUSTON, M., AGRAWALA, M., AND HUMPHREYS, G. 2003. Non-invasive interactive visualization of dynamic architectural environments. In *I3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics*, ACM, New York, NY, USA, 55–58.
- VIOLA, I., AND GRÖLLER, M. E. 2005. Smart visibility in visualization. In *Proceedings of EG Workshop on Computational Aesthetics Computational Aesthetics in Graphics, Visualization and Imaging*, B. G. L. Neumann, M. Sbert and W. Purgathofer, Eds., 209–216.



Figura 6: *Modelo 1 antes e depois da explosão da parte central.*

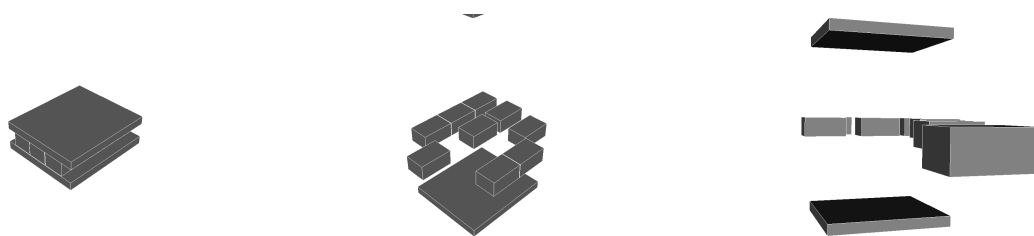


Figura 7: *Modelo 2 antes e depois da explosão da parte central.*

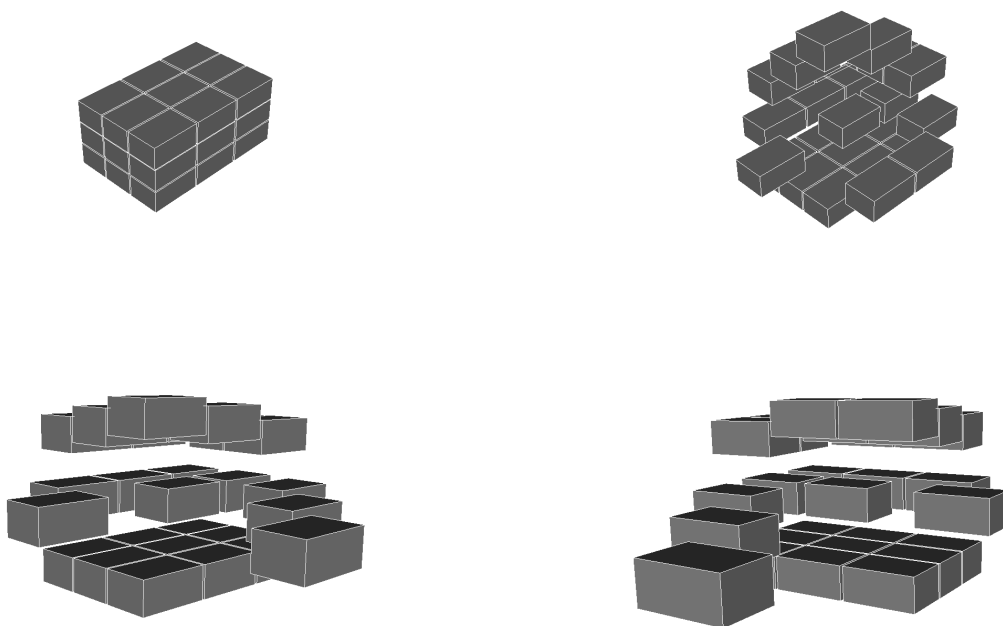


Figura 8: *Modelo 3 antes e depois da explosão da parte central.*

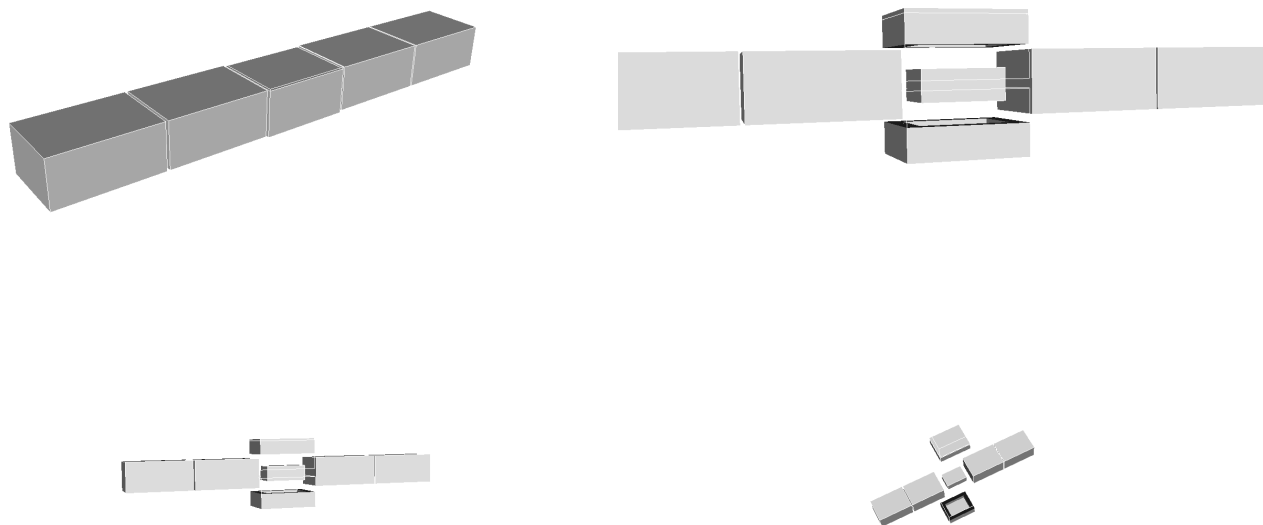


Figura 9: *Modelo 4 antes e depois da explosão da parte central.*



Figura 10: *Modelo 5 antes e depois da explosão da parte central.*