

# Vistas Explodidas: um estudo sobre o trabalho *Automated generation of interactive 3D exploded view diagrams*

Fábio Markus Miranda\*  
PUC-Rio

## Resumo

Este trabalho implementa a proposta descrita em *Automated generation of interactive 3D exploded view diagrams* para a visualização de modelos 3D complexos com a utilização da técnica de vista explodida. Com ela, é possível ter uma visão global do modelo e entender a relação entre as diversas partes que o compõe, sem perda de nenhum detalhe, como ocorre em outras técnicas.

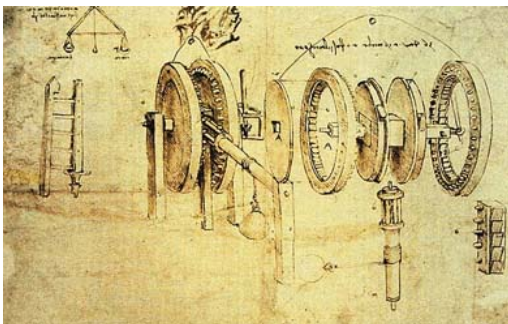
A base do trabalho é a geração de um *grafo de explosão* que irá ditar como as diferentes partes que compõe um modelo 3D se relacionam, qual a direção de explosão de cada uma e em qual ordem se dará a explosão. Uma discussão é feita ao final do trabalho, mostrando diferentes modelos de teste explodidos.

**Keywords:** exploded view, visualization

## 1 Introdução

Um problema típico na visualização de modelos complexos, como motores, peças industriais e dispositivos eletrônicos, é que as características mais interessantes podem estar obstruídas por outras partes menos importantes. 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, permitindo um maior entendimento do objeto em questão.

Uma das técnicas de visibilidade é a *vista explodida*, desenvolvida ao longo de séculos na área de ilustração técnica, e agora utilizada para a visualização de modelos 3D. 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, baseado em sua experiência, reconstruir mentalmente o modelo e assim ter uma visão global e o relacionamento entre as partes que o constituem.



**Figura 1:** Exemplo de uma ilustração de vista explodida de Leonardo da Vinci.

Este trabalho implementa uma técnica de vista explodida apresentada em [Li et al. 2008]. O trabalho busca diminuir a desordem

visual ao mesmo tempo que minimiza as distâncias de explosão, facilitando assim o entendimento das relações entre as partes do modelo. Para isso, é proposto a construção de um *grafo de explosão*, que irá ditar a ordem e a direção com que cada parte poderá ser explodida. O usuário poderá escolher quais partes deseja visualizar e o sistema irá expandir (ou colapsar) de acordo com o que foi previamente gerado no grafo.

O objeto principal aqui 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 abrangente, serve como uma base para o estudo da área. Em [Elmqvist and Tsigas 2008], os autores apresentam cinco *patterns* (como *tour planner*, *interactive exploder*, *virtual X-Ray*, etc), derivados de 25 características, com o objetivo de classificar diferentes técnicas de gerenciamento de oclusão, apresentando os pontos positivos e negativos de cada uma. Segundo o artigo, a vista explodida possui a vantagem de não remover nenhum tipo de informação do modelo e também mantém todas as suas informações de relacionamento, ao contrário das técnicas baseadas, por exemplo, em *cut outs*.

Os trabalhos relacionados a vista 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 descrevem a técnica *visual access distortion*, que busca criar um caminho de visualização sem obstruções até a parte de maior interesse. Porém, a técnica, como descrita originalmente, aumenta o tamanho das partes mais importantes do modelo, o que pode distorcer qualquer tipo de relacionamento espacial entre as partes. Os trabalhos [Bruckner 2006] e [McGuffin et al. 2003] também apresentam propostas para a visualização de dados volumétricos com vistas explodidas, mas sem nenhum tipo de deformação. Os resultados são aplicados na visualização de dados volumétricos de caráter médico.

Uma outra vertente na pesquisa de vistas explodidas é voltado para a explosão de modelos 3D complexos, não baseados em dados volumétricos. O trabalho [Agrawala et al. 2003] é uma referência inicial nesse 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. Esse último trabalho também é integrado ao trabalho [Li et al. 2007], onde é apresentado

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



**Figura 2:** Trabalhos relacionados ([Carpendale et al. 1996], [McGuffin et al. 2003], [Bruckner 2006], [Agrawala et al. 2003], [Li et al. 2008], [Niederauer et al. 2003], [Luboschik and Schumann 2007]).

um sistema de *cut away view*.

Outro trabalho também relacionado a vistas explodidas é [Niederauer et al. 2003], onde os autores apresentam um sistema para a geração automática de vistas explodidas de construções, levando em consideração as normais do modelo e a altura média dos andares. Como resultado, os autores mostram a utilização do sistema na navegação interativa em ambientes arquitetônicos e também em jogos *online*, onde o "telespectador" pode ter uma visão global dos diferentes andares de um mapa de um jogo.

Finalmente, o trabalho [Luboschik and Schumann 2007] utiliza técnicas de vista explodida não para visualizar modelos 3D, mas para visualizar dados, como a classificação taxonômica de animais.

A Figura 2 apresenta algumas imagens dos trabalhos na área de vistas explodidas.

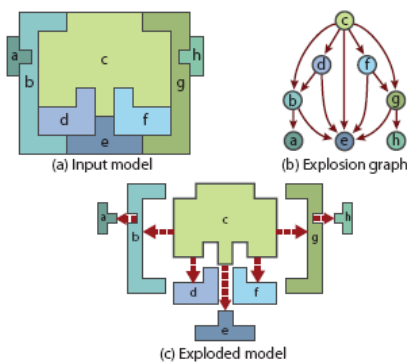
### 3 Visão Geral

O sistema proposto em [Li et al. 2008] recebe como entrada um modelo 3D composto por diversas partes, que deverão ser explodidas. 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).



**Figura 3:** Grafo de explosão ([Li et al. 2008])

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.

#### 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 facilitar a reconstrução mental do modelo por parte do usuário, as partes só poderão explodir em seis direções, correspondentes aos eixos  $x$ ,  $y$  e  $z$ .

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 4).

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 ( $x$ ,  $y$

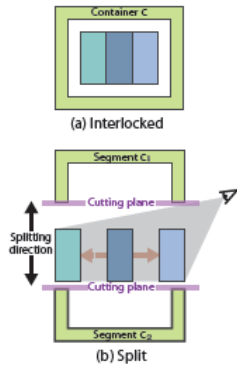


Figura 4: Partes cercadas ([Li et al. 2008])

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: Deslocamento das partes que obstruem uma parte de interesse

## 4 Implementação

Para a implementação do sistema, foi utilizado o *OpenSceneGraph* [osg], para a renderização e carregamento dos modelos 3D, e *PQP* [ppq] para a detecção de colisões entre as partes do modelo. A Figura 5 apresenta uma visão geral dos módulos implementados, que serão discutidos posteriormente.

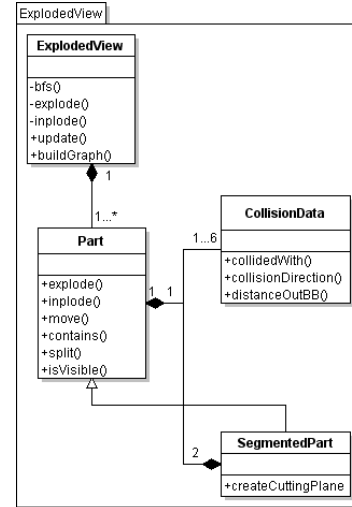


Figura 5: Visão geral da arquitetura

- **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 uma simplificação do que é proposto em [Agrawala et al. 2003], mas que não altera a geração do grafo de explosão. Na implementação, para detectar direções bloqueadas, as partes são deslocadas em um  $\Delta d$ .

A interação com o usuário é feita através de um *menu* onde é possível selecionar a parte que se deseja ter uma melhor visualização. Considerando que o usuário quer visualizar 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 6 mostra os níveis montados pela busca em largura para um modelo simples e a partir da parte A.

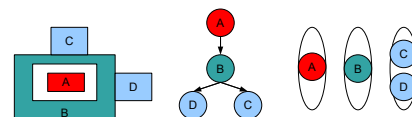


Figura 6: Exemplo de níveis da BFS

A explosão e a implosão do modelo serão feitas então com base nesses níveis, explodindo os níveis 1 até  $n$ , ou então, de forma inversa, implodindo de  $n$  até 1.

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. Quando uma parte é caracterizada como recipiente, ele será dividido por um **ClipNode** ao longo dos três eixos; as duas partes resultantes da divisão serão explodidas. O eixo com a menor distância de explosão e que permita a visualização das partes contidas será escolhido como o eixo de explosão.

O *loop* do sistema pode ser resumido da seguinte forma:

```

Dados:  $G$ : grafo de explosão
Dados:  $T$ : parte escolhida pelo usuário para observar
Dados:  $A_{i,j}$ : níveis de uma BFS
enquanto Usuário não saiu do sistema faça
    se  $T \neq \emptyset$  então
         $A = BFS(T)$ 
    fim
    /* Explode partes */
    se  $T \neq \emptyset$  e  $A \neq \emptyset$  então
        para cada  $p_{i,j} \in A_{i,j}$  faça
            se  $deslocamento(p_{i,j}) < deslocamento para$ 
                sair do bounding box então
                    Desloca  $p_{i,j}$  em  $\Delta d$  na sua direção de
                    explosão;
            fim
        fim
    fim
    /* Implode partes */
    se  $T = \emptyset$  e  $A \neq \emptyset$  então
        para cada  $p_{i,j} \in A_{i,j}$  faça
            se  $deslocamento(p_{i,j}) > 0$  então
                Desloca  $p_{i,j}$  em  $-\Delta d$  na sua direção de
                explosão;
            fim
        fim
    fim
    /* Verifica visibilidade de T */
    se Todas as partes em  $A$  foram explodidas então
        se  $T$  não está visível então
            para cada  $p_{i,j} \in A_{i,j}$  faça
                Desloca  $p_{i,j}$  em  $\Delta d$  na sua direção de
                explosão;
            fim
        fim
    fim
fim

```

**Algoritmo 3:** Algoritmo geral do sistema

Quando o usuário seleciona uma parte  $T$  a ser visualizada, o sistema irá executar a busca em largura no grafo de explosão, que retornará os níveis contendo as peças filhas do nó  $T$ . Essas serão então deslocadas em  $\Delta d$  na sua direção de explosão a cada iteração do *loop*; quanto todas as peças de um determinado nível forem explodidas, o próximo nível será explodido. Quando todas as peças já estiverem explodidas, será verificado se a peça escolhida pelo usuário está visível ou não; caso não esteja visível, todas as peças filhas de  $T$  serão deslocadas novamente em  $\Delta d$ .

#### 4.1 Características não abordadas

O sistema proposto em [Li et al. 2008] aborda algumas características que não foram implementadas, como o tratamento de gra-

fos de explosão hierárquicos, que permite a explosão de grupos de partes independentemente. Além disso, os autores também demonstram um modo onde o usuário pode, com o *mouse*, manipular diretamente a explosão das partes. Há também uma integração com outro trabalho ([Li et al. 2007]) que permite a visualização de modelos explodidos juntamente com técnicas de *cutaways*.

Além disso, ao contrário do artigo, que constrói 26 grafos, esta implementação constrói apenas um, levando em consideração apenas um ponto de vista. Como o ponto de vista só é utilizado para determinar o plano de corte de partes envolventes, a simplificação não afetou o funcionamento geral do sistema. Os cortes continuam ocorrendo, só que não necessariamente no plano que resulta em uma menor distância para sair do *bounding box*.

## 5 Resultados

Durante o desenvolvimento do sistema, alguns modelos 3D foram construídos com o objetivo de testar e expor as principais características do sistema. Dentre eles, cinco modelos cobrem de maneira consistente todos os aspectos desenvolvidos. Para manter um padrão nos testes, a peça central é sempre a peça escolhida para ser observada.

Os tempos de geração do grafo de explosão foram medidos em um computador Core 2 Duo 2.8 Ghz, 2GB Memória RAM.

O resultado dos testes e uma discussão sobre cada modelo pode ser visto a seguir.

### 5.1 Modelo 1

O primeiro modelo, Figura 7, possui apenas quatro partes, dispostos de forma simples. Quando explodido, as partes que obstruem a parte central se movem na mesma direção que a direção de explosão, como visto a direita da Figura 7.

Tempo de geração do grafo de explosão: 1s.

### 5.2 Modelo 2

O modelo número dois apresenta uma configuração mais complexa, como visto na Figura 8. O objetivo principal foi testar a direção em que duas partes (as tampas inferior e superior) iriam explodir. O mais sensato seria que elas explodissem no eixo Z, o que de fato ocorreu. Com a mudança na posição da câmera, e obstrução da parte central, as outras partes vão se movendo ao longo de sua direção de explosão (até a parte central ficar totalmente visível).

Tempo de geração do grafo de explosão: 7s.

### 5.3 Modelo 3

O modelo três é similar ao modelo anterior, mas as duas tampas foram divididas em seis pedaços cada. Novamente, o objetivo era notar o comportamento da explosão dessas partes. A Figura 9 mostra as partes se movendo ao longo de sua direção de explosão para manter a parte central sempre visível, independentemente da posição da câmera.

Tempo de geração do grafo de explosão: 32s.

### 5.4 Modelo 4

O modelo quatro apresenta uma parte cercada por outra parte, como mostra a Figura 10. O sistema então dividiu a parte envolvente em duas outras partes. A mudança na posição da câmera determina

o quanto as duas novas metades irão se mover ao longo de sua direção de explosão. Obviamente, as metades devem ter direções de explosão opostas.

Tempo de geração do grafo de explosão: 7s.

## 5.5 Modelo 5

O modelo cinco apresenta uma parte central cercada por diferentes partes, como mostra a Figura 11. A disposição das peças é mais complexa do que a exibida no modelo 1, mas o sistema consegue separá-las satisfatoriamente.

Tempo de geração do grafo de explosão: 2s.

## 5.6 Modelo 6

O último modelo adiciona uma nova parte ao modelo 5, que encobre todas as partes (Figura 12). Porém, essa parte possui uma abertura lateral. Como há uma direção livre, não há a necessidade de se dividir nenhuma parte de modelo, as partes simplesmente se deslocam ao longo da direção livre, saindo da cobertura.

Tempo de geração do grafo de explosão: 5s.

## 6 Conclusão e Trabalhos Futuros

O artigo [Li et al. 2008], com sua proposta de grafos de explosão, apresenta uma boa maneira para a visualização de modelos 3D, uma vez que essa representação é uma forma ideal e bastante intuitiva de relacionar diferentes partes e o bloqueio existente entre cada uma delas. Um ponto negativo, porém, é que o tempo de geração do grafo de explosão pode crescer muito com o número de partes, podendo chegar a 1778s para modelos com 55 partes.

As principais características foram implementadas neste trabalho; alguns pontos foram simplificados, mas isso não comprometeu o resultado final, já que o principal ponto, a explosão das partes, foi implementado satisfatoriamente, com bons resultados para os modelos de teste.

Além do estudo do artigo citado, este trabalho se mostrou muito útil para adquirir um conhecimento maior na área de técnicas de visibilidade inteligente e os problemas que cada técnica busca solucionar.

Um grande problema durante o desenvolvimento foi a falta de modelos 3D para o teste do sistema. Apesar dos modelos criados representarem bem as características e recursos do sistema, eles certamente possuem uma complexidade bem inferior aos modelos 3D reais, seja na quantidade de partes ou na disposição das mesmas. O próprio artigo cita que o sistema pode ter resultados insatisfatórios para modelos com superfícies com algum tipo de irregularidade.

Um problema que pode ser explorado é a explosão de peças ao longo de um conjunto de eixos, possibilitando que modelos mais complexos sejam explodidos. Como a reconstrução mental do modelo por parte do usuário pode ficar comprometida, outros recursos gráficos poderiam ser estudados, como linhas ligando a posição original da parte até o seu destino. Outro aspecto é a divisão do modelo 3D em partes. O artigo que este trabalho implementou já considera que o modelo de entrada está bem dividido, algo que não ocorre necessariamente.

Além disso, o uso de vista explodida para a visualização de dados e informação (como [Luboschik and Schumann 2007]), e não só para modelos 3D convencionais, parece ser promissor.

Finalmente, poucos trabalhos buscam unir diferentes técnicas de visualização inteligente e avaliar os resultados e a facilidade de

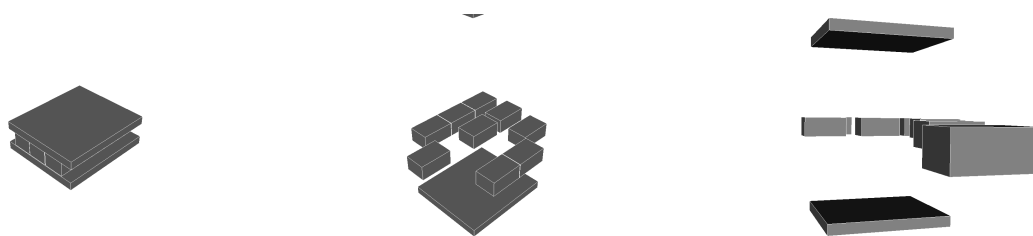
interpretação do modelo. Um caminho seria avaliar se tal união realmente é vantajosa e como ela poderia ser feita. O uso de realidade aumentada também poderia ser estudado, como forma de auxiliar a interação do usuário com as diversas partes que formam um modelo 3D.

## Referências

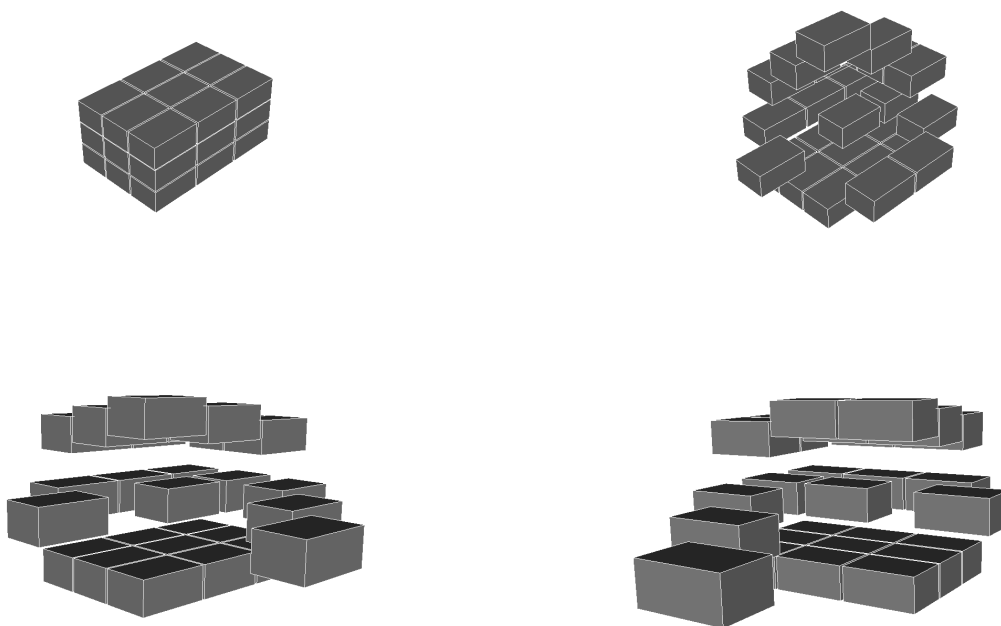
- AGRAWALA, M., PHAN, D., HEISER, J., HAYMAKER, J., KLINGNER, J., HANRAHAN, P., AND TVERSKY, B. 2003. Designing effective step-by-step assembly instructions. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, ACM, New York, NY, USA, 828–837.
- BRUCKNER, S. 2006. Exploded views for volume data. *IEEE Transactions 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., RITTER, L., AGRAWALA, M., CURLESS, B., AND SALE-SIN, D. 2007. Interactive cutaway illustrations of complex 3d models. *ACM Trans. Graph.* 26, 3 (July).
- 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.
- LUBOSCHIK, M., AND SCHUMANN, H. 2007. Explode to explain - illustrative information visualization. In *IV '07: Proceedings of the 11th International Conference Information Visualization*, IEEE Computer Society, Washington, DC, USA, 301–307.
- 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.
- Openscenegraph. <http://www.openscenegraph.org/>.
- Pqp - the proximity query package. <http://gamma.cs.unc.edu/SSV/>.
- 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.



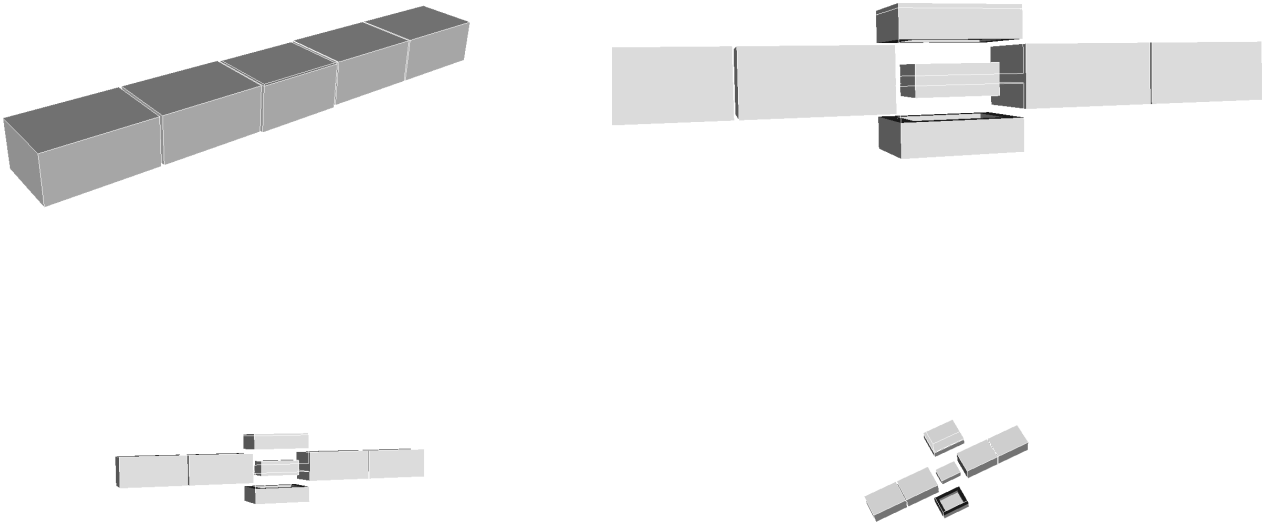
**Figura 7:** *Modelo 1 antes e depois da explosão da parte central.*



**Figura 8:** *Modelo 2 antes e depois da explosão da parte central, e alterando a posição da câmera.*



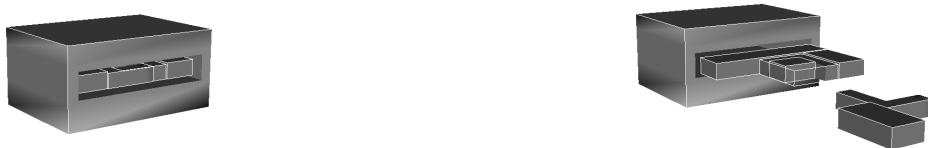
**Figura 9:** *Modelo 3 antes e depois da explosão da parte central, e alterando a posição da câmera.*



**Figura 10:** *Modelo 4 antes e depois da explosão da parte central, com uma parte envolvendo totalmente outra parte.*



**Figura 11:** *Modelo 5 antes e depois da explosão da parte central.*



**Figura 12:** *Modelo 6 antes e depois da explosão da parte central.*