

# ALGORITMO PARA DETERMINAÇÃO DE COORDENADAS ESPACIAIS DE OBJETOS COM BASE EM IMAGENS

*Vivian Dorat Betoni<sup>1</sup>, Dr. Valdemir Carrara<sup>2</sup>*

<sup>1</sup>Universidade do Vale do Paraíba, Av. Shishima Hifumi, 2911 – Urbanova  
São José dos Campos, SP, 12244-000

<sup>2</sup>Instituto Nacional de Pesquisas Espaciais – INPE / Divisão de Mecânica Espacial e Controle – DMC  
Av dos Astronautas, 1.758 – Jardim da Granja  
São José dos Campos, SP, 12227-010,  
vdb29586@yahoo.com.br <sup>(1)</sup>, val@dem.inpe.br <sup>(2)</sup>

**Resumo** - Este trabalho tem como objetivo a criação e desenvolvimento de um algoritmo que define coordenadas espaciais de objetos, com base em um processamento de coordenadas bidimensionais obtidas através de imagens (fotografias), em diversos ângulos, de uma mesma cena, gerando assim o objeto desta cena em três dimensões. O processamento de coordenadas consiste em uma minimização do erro entre as projeções das prováveis soluções das coordenadas espaciais (dadas pelo usuário) e as respectivas coordenadas reais das imagens. Para a determinação do erro e sua variação com relação aos parâmetros das transformações e sua minimização, o algoritmo realiza aplicações de transformações lineares (rotação, variação de escala e translação) como em cálculos estereoscópicos. O algoritmo será desenvolvido em linguagem C ou C++, com visualização realizada por meio de OpenGL.

**Palavras-chave:** Computação Gráfica, OpenGL, Estereoscopia.

**Área do Conhecimento:** Ciências Exatas, Ciência da Computação, Computação Gráfica

## Introdução

A criação de um modelo computacional foto-realístico tridimensional encontra várias aplicações que vão desde planejamentos arquiteturais, reconstruções arqueológicas até a criação de ambientes virtuais e efeitos especiais. Combinando técnicas de Computação Gráfica e Computação Visional torna-se possível a criação de um algoritmo que defina coordenadas espaciais de objetos através de um processamento de coordenadas bidimensionais, utilizando como base algumas imagens (fotografias) de um mesmo objeto em diversos ângulos, de uma mesma cena, gerando assim o objeto desta cena em três dimensões. O processo de modelagem é composto por duas partes:

- na primeira, utilizando-se de técnicas de modelagem fotogramétrica, busca-se a extração de informações nas imagens, a serem usadas na sua reconstrução tridimensional, de maneira interativa, mas com cálculos para detecção de possíveis erros nos pontos indicados pelo usuário e, caso ocorra, da minimização desta distância.

- na segunda parte, por meio de técnicas estereoscópicas, é feita a recuperação do grau de desvio da imagem base (principal) com as outras imagens e seus respectivos pontos de interligação, recuperando a profundidade da extensão espaçada dos pares de imagem (ou seja, a

variação de escala das câmeras), rotação e translação.

Portanto, o objetivo deste trabalho é determinar as coordenadas espaciais de um objeto, utilizando-se como base um grupo de imagens e buscando uma modelagem otimizada, com o uso de menos fotos do que utilizam as aproximações de modelagem com base em imagens atuais. A linguagem usada no momento é a C, mas contará com recursos de OpenGL para a visualização, que se dará utilizando a linguagem C++.

## Materiais e Métodos

Em 1996 Paul Debevec desenvolveu um programa que visava precisão, conveniência e realismo. Utilizando-se de sistemas de síntese gráfica baseados em geometria e novas técnicas baseadas em imagens, gerava-se uma arquitetura fotogramétrica com tratamento de detalhes recuperados por correspondência estereoscópica que requeria um pequeno conjunto de fotos, produzindo resultados de qualquer ponto de vista (Debevec, 1996). Este programa, denominado *Façade*, seria o precursor do software *Canoma* da *MetaCreations* (Viewpoint Corporation, 2000).

Neste trabalho é dada uma atenção especial ao desenvolvimento de um algoritmo que, servindo-se de pontos oferecidos pelo usuário em cada imagem bidimensional utilizada, minimiza a

distância entre os pontos fornecidos e os pontos reais com base em transformações paramétricas do posicionamento e orientação das câmeras.

#### Formulação da minimização do erro

Dado um conjunto de  $n$  pontos em um espaço vetorial (ou seja, os pontos reais das imagens) denotados por:

$$P_i = (x_i, y_i, z_i), i = 1, \dots, n, \quad (1)$$

relativos a um sistema de coordenadas retangulares  $x, y, z$ , deseja-se estimar estas coordenadas com base no processamento das projeções destes pontos numa seqüência de imagens.

Considera-se agora  $l$  outros sistemas de coordenadas  $x_k, y_k, z_k$ , com  $k = 1, 2, \dots, l$  (número de imagens) relacionados ao sistema  $x, y, z$  por meio de uma matriz de rotação  $A_k$  e de um vetor de translação  $T_k$ . Os pontos  $P_i$  são projetados nos planos  $x_k, y_k$  por meio de uma projeção em perspectiva, o que origina o conjunto de pontos:

$$Q_{ki} = (x_{ki}, y_{ki}, 1), i = 1, \dots, n, \quad (2)$$

para cada sistema de coordenadas.

Porém, cada conjunto  $Q_{ki}$  destes pontos contém um subconjunto distinto de projeções de  $P_i$ , ou seja, os pontos fornecidos (ou estimados) pelo usuário que, em geral, não constitui a totalidade dos  $n$  pontos. Isto se deve ao fato de que nem todos os pontos são observados em cada imagem. As coordenadas espaciais dos pontos  $P_i$  no sistema  $k$  são dadas por  $P_i^k$  e são obtidas a partir das equações da projeção em perspectiva:

$$P_i^k = z_{ki} Q_{ki} = z_{ki} \begin{pmatrix} x_{ki} \\ y_{ki} \\ 1 \end{pmatrix} \quad (3)$$

onde  $z_{ki}$  representa a profundidade desconhecida de  $P_i^k$ , e o índice superior  $k$  indica o sistema de coordenadas no qual o ponto é referido.

Por sua vez as posições relativas ao sistema de coordenadas das projeções ( $Q_{ki}$ ) relacionam-se com o sistema  $x, y, z$  dos pontos  $P_i$  por meio de uma matriz de rotação  $A_k$  e de uma translação de um vetor  $T_k$ :

$$P_i = A_k (P_i^k - T_k) \quad (4)$$

Procura-se reconstituir os pontos reais ( $P_i$ ) utilizando-se exclusivamente os diversos conjuntos  $Q_{ki}$ . Para isto, não só as coordenadas dos pontos  $P_i$  devem ser calculadas, mas também a rotação e a translação efetuadas em cada sistema de coordenadas, bem como as profundidades  $z_{ki}$  de cada ponto.

A solução adotada aqui consiste na minimização da distância entre uma estimativa  $S_i$  dos pontos  $P_i$ , com base nas diversas projeções destes pontos nos sistemas  $k$ , ou seja:

$$f = \min \sum_{k=1}^l \left[ \sum_{i=1}^{n_k} (P_i - S_i)^2 \right] \quad (5)$$

Este problema apresenta duas indeterminações: a localização do sistema de coordenadas dos pontos  $P_i$  e a escala dos sistemas de coordenadas.

Logo, deve-se adotar um sistema qualquer para a solução do problema. Adota-se para  $x, y, z$  um sistema paralelo ao da primeira imagem,  $x_1, y_1, z_1$ , aplicando apenas uma translação de  $h_1$  no eixo  $z$ . Com isso tem-se que  $A_1 = I$  e  $T_1 = (0, 0, h_1)^T$ . Como a minimização não depende do sistema na qual as distâncias são medidas, então é igualmente válido escrever:

$$f = \min \sum_{k=1}^l \left[ \sum_{i=1}^{n_k} (P_i^k - S_i^k)^2 \right]. \quad (6)$$

Porém se  $S_i$  for uma solução válida, então  $aS_i$  e  $aP_i$ , onde  $a$  é uma constante, também será, o que significa que a escala do problema não pode ser determinada por meio deste processo. Assim, deve-se adotar uma dada medida como sendo a unidade. Adota-se, por conveniência,  $|T_1| = h_1 = 1$ . Contudo deve-se mencionar que os vetores de translação das demais imagens não serão unitários.

Dado um ponto  $P_i$  no sistema  $xyz$ , este ponto pode ser transformado em coordenadas relativas ao sistema  $x_k y_k z_k$  com:

- aplicação de 3 rotações em direções distintas, de ângulos  $\alpha_k, \beta_k$  e  $\theta_k$ , nos eixos cartesianos  $y, x$  e  $z$ , respectivamente.
- aplicação de uma translação de um escalar  $h_k$  na direção de  $z$ .

- aplicação de uma translação de  $u_k$  e  $v_k$  nas direções  $x_k$  e  $y_k$ , respectivamente (ou seja,  $T_k = (u_k \ v_k \ h_k)^T$ ) conforme mostra a Figura 1

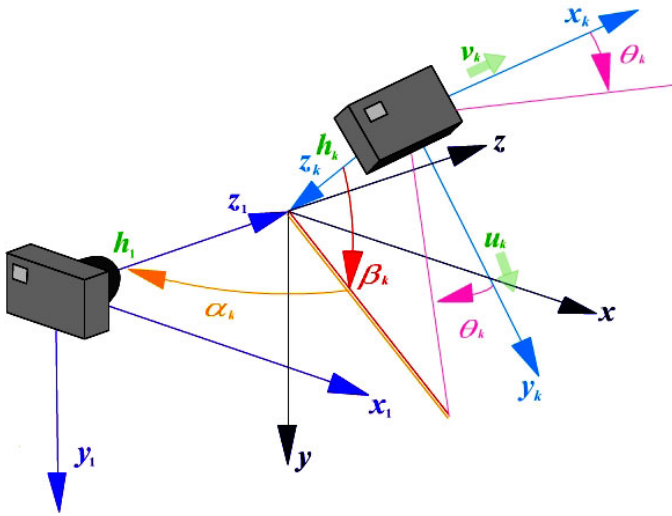


Figura 1 - Sistemas de coordenadas da câmera em duas orientações distintas

Traduzindo-se em termos de equações, tem-se:

$$P_i^k = A_k^T P_i + T_k \quad (7)$$

onde a matriz de coordenadas da rotação é dada por:

$$A_k = R_y(\alpha_k) R_x(\beta_k) R_z(\theta_k), \quad (8)$$

tal que:

$$R_y(\alpha_k) = \begin{pmatrix} \cos \alpha_k & 0 & -\sin \alpha_k \\ 0 & 1 & 0 \\ \sin \alpha_k & 0 & \cos \alpha_k \end{pmatrix}, \quad (9)$$

$$R_x(\beta_k) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \beta_k & \sin \beta_k \\ 0 & -\sin \beta_k & \cos \beta_k \end{pmatrix}, \quad (10)$$

$$R_z(\theta_k) = \begin{pmatrix} \cos \theta_k & \sin \theta_k & 0 \\ -\sin \theta_k & \cos \theta_k & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (11)$$

Como o objetivo é comparar as coordenadas dos pontos das diversas imagens entre si, deve-se então adotar um sistema de coordenadas único. Embora o sistema verdadeiro dos pontos  $P_i$  não

possa ser determinado, é possível adotar o sistema de coordenadas normalizado, onde  $h_1 = 1$ . Neste caso deve-se expressar as diversas projeções de  $P_i$  neste sistema, e, portanto, é necessário inverter a relação de transformação, ou seja:

$$P_i = A_k (P_i^k - T_k) \quad (12)$$

e substituindo-se a transformação de coordenadas e da equação da projeção em perspectiva na função de minimização tem-se:

$$f = \min \sum_{k=1}^l \left[ \sum_{i=1}^{n_k} (z_{ki} A_k Q_{ki} - A_k T_k - S_i)^2 \right], \quad (13)$$

onde a primeira somatória constitui o conjunto de imagens e a segunda é constituída pelos pontos de cada uma das imagens.

A minimização de  $f$  requer que o gradiente seja nulo na solução do problema. O número de valores (parâmetros de transformação, coordenadas e profundidades) a serem estimados no total é dado por :

$$N = 6l + 3n + \sum_{k=1}^l n_k, \quad (14)$$

que pode ser um número relativamente grande se houverem muitos pontos e muitas imagens.

Uma vez que a função de minimização é quadrática contendo apenas funções lineares ou trigonométricas, pode-se obter este gradiente efetuando-se derivadas analíticas de cada um destes parâmetros. Evita-se, desta forma, a necessidade de se utilizar funções de otimização exclusivamente numéricas. Além disso, a derivada analítica permite obter a solução ótima num único cálculo, a menos, é claro, que as demais variáveis não sejam corretas.

### Algoritmo

O método de otimização empregado inicialmente consistiu num algoritmo iterativo onde era admitida inicialmente uma solução, e depois se procurava atingir o resultado ótimo com base no cálculo consecutivo das soluções analíticas de cada uma das variáveis. Supôs-se que tal procedimento levaria à solução ótima, pois individualmente cada uma das soluções analíticas apresentava uma variação lenta com os demais parâmetros. Contudo verificou-se rapidamente não ser este o caso, pois o método somente convergia caso a solução admitida estivesse muito próxima da solução real. Em vista disso, partiu-se para

uma análise de quais são as circunstâncias que garantem a convergência, e, ao mesmo tempo, uma busca exaustiva por soluções que estejam suficientemente próximas da solução real. O principal obstáculo é obter algoritmos de busca rápida, uma vez que ela pode ser bastante lenta, principalmente se a região de convergência for pequena. Desconhece-se, ainda, qual é o tamanho desta região para cada um dos parâmetros de otimização.

O algoritmo foi desenvolvido inicialmente na linguagem C, utilizando-se o compilador lccwin, por possuir interface mais compacta, ideal para o desenvolvimento de pequenos algoritmos. É uma ferramenta padrão ANSI C (não compila programas em C++) disponível gratuitamente para uso não-comercial (Navia, J., 1995).

## Resultados e discussão

As coordenadas dos pontos no espaço são geradas aleatoriamente, portanto ainda sem a interação com o usuário. O algoritmo para geração destes pontos também gera as coordenadas bidimensionais das imagens, admitindo-se para isso rotações e translações para cada câmera, além da eliminação (também aleatoriamente) de alguns destes pontos. Um vetor adicional  $K_i^k$  informa se o ponto  $P_i^k$  é ou não observado na imagem  $k$ . O programa conta ainda com estruturas destinadas a realizar cálculos com matrizes e funções para implementar as matrizes de rotação.

Foi também implementado o cálculo da função de minimização  $f$ , bem como funções para cálculo das derivadas (analíticas) desta função com relação aos parâmetros a serem determinados, como os ângulos  $\alpha_k$ ,  $\beta_k$  e  $\theta_k$ , a translação  $h_k$ ,  $u_k$ ,  $v_k$ , e as profundidades  $z_{ki}$  dos pontos.

Por fim, implementou-se um laço onde todos os parâmetros são variados dentro de certos limites (exceto o cálculo de  $S_i$  e  $z_{ki}$ , que são feitos analiticamente), com incrementos previamente definidos, e busca-se, dentre todas as combinações, aquela que apresenta o menor valor para a função de minimização  $f$ .

No presente estágio de desenvolvimento este laço está sendo otimizado, visando reduzir o excessivo tempo de processamento demandado. Como exemplo, se o problema for composto de duas imagens e considerando-se que seriam necessários apenas 100 pontos para cada parâmetro a ser estimado, dentre  $\alpha_k$ ,  $\beta_k$ ,  $\theta_k$ ,  $h_k$ ,  $u_k$  e  $v_k$ , então seria necessário calcular a função de minimização em  $10^{12}$  pontos. Percebe-se portanto que mesmo com os computadores atuais esta tarefa é bastante demorada, o que viabiliza a

busca por soluções otimizadas e algoritmos mais velozes.

## Conclusão

No estado atual de desenvolvimento, o algoritmo apresenta uma convergência bastante lenta, o que é normal neste tipo de problema. Pretende-se otimizar o código de forma a acelerar a busca da zona de convergência do método analítico. Será importante, neste ponto, identificar o tamanho da região de convergência e quais os fatores que influenciam sua dimensão. Outros métodos de busca podem ser empregados assim que a convergência estiver garantida, como, por exemplo, algoritmos genéticos ou outros critérios de otimização.

Na presente fase de desenvolvimento, considera-se que os objetivos no que diz respeito à familiarização com as formulações e métodos matemáticos e compreensão do trabalho em si foram alcançados. O projeto encontra-se em fase de testes para avaliar a convergência da busca da solução. Os próximos aprimoramentos podem ser resumidos em:

- Melhorar os algoritmos de busca da solução ótima;
- Criação de uma interface;
- Trabalhar com imagens reais;
- Permitir a entrada dos pontos das imagens.

## Referências

- Debevec, P. E.; Modeling and Rendering Architecture from Photographs. Dissertation. UC at Berkeley. Disponível em [www.debevec.org/Thesis/debevec-phdthesis-1996.pdf](http://www.debevec.org/Thesis/debevec-phdthesis-1996.pdf), 1996. Acesso em 20 fev. 2006
- Debevec, P. E.; Taylor, C. J.; Malik, J.; Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach. Proceedings of the 23rd SIGGRAPH Conference, p. 11-20, 1996.
- Navia, Jacob; Compilador LCC-Win32. Disponível em [www.cs.virginia.edu/~lcc-win32/index.html](http://www.cs.virginia.edu/~lcc-win32/index.html) ou em [www.q-software-solutions.com/lccwin32/](http://www.q-software-solutions.com/lccwin32/). Acesso em 15.mar.2006.
- Viewpoint Corporation. [www.metacreations.com/products/canoma/](http://www.metacreations.com/products/canoma/), 2000. Acesso em 15.mar.2006.