

6 Implementação do sistema

Como resultado dos estudos deste trabalho, implementamos um protótipo de um sistema de reconstrução 3D com base nos métodos de escultura do espaço. Nosso sistema é bastante simples e de baixo custo, consistindo de um computador Pentium III, munido de uma placa GeForce II e um conjunto de quatro webcams Logitech conectadas por um hub USB, colocadas nos vértices superiores de um cubo de 1x1x1m(Figura 6.1).



Figura 6.1: Imagem do sistema

O processo de reconstrução envolve cinco etapas fundamentais: a calibração das câmeras, que pode ser efetuada uma única vez, desde que a posição e a orientação das mesmas não sejam alteradas; a calibração do fundo que tem como o objetivo permitir a extração das regiões delimitadas pelas silhuetas do objeto; a captura das imagens e a segmentação dos objetos de interesse; e finalmente, o processo de reconstrução propriamente dito, através dos métodos de escultura do espaço aqui apresentados (Figura 6.2).

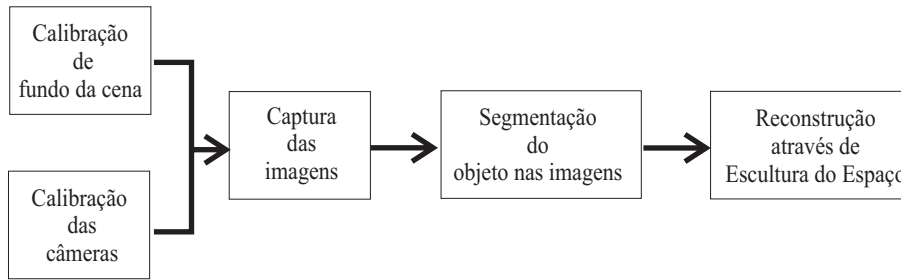


Figura 6.2: Processo de reconstrução

Um dos maiores desafios na construção deste sistema foi lidar com os problemas causados pelo uso de webcams, as quais possuem resolução muito baixa, além de introduzirem muito ruído nas imagens capturadas.

Vejamos como foram implementados cada uma das etapas associadas ao processo de reconstrução no sistema construído.

6.1 Calibração das câmeras

O método utilizado para calibrar as câmeras é baseado no processo de calibração automático proposto por Szenberg em sua tese de doutoramento [75]. O método de Szenberg é capaz de calibrar uma câmera automaticamente através do reconhecimento de um modelo descrito por um conjunto de segmentos de reta. Em nosso caso utilizamos o padrão mostrado na figura 6.3.

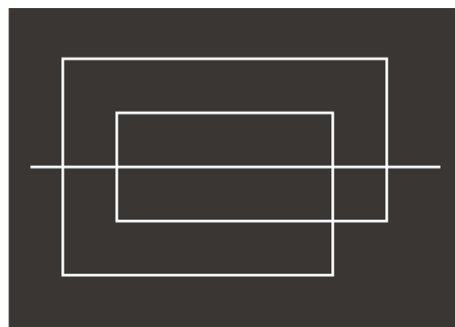


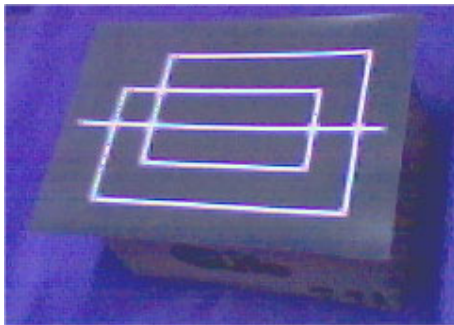
Figura 6.3: Padrão de calibração

Primeiramente, a imagem contendo o modelo é filtrada, através de um filtro *laplaciano da gaussiana*, com o objetivo de realçar os pixels pertencentes aos segmentos de reta de interesse. Em seguida, uma etapa de busca é efetuada com o objetivo de detectar segmentos de retas longos, a partir de pequenos segmentos de reta com mesma orientação, os quais são analisados

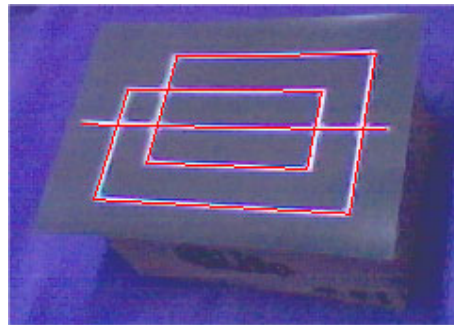
com base em uma árvore de interpretações. O resultado desta análise retorna os segmentos reconhecidos, isto é, aqueles que satisfazem as restrições utilizadas na caracterização do modelo. Posteriormente, é calculada uma transformação projetiva que mapeia os segmentos do modelo nos segmentos reconhecidos, através da qual pode-se detectar novos segmentos ou reajustar os antigos, tanto na própria imagem usada na calibração quanto em uma nova imagem obtida a partir de uma ligeira modificação na câmera. Finalmente, com base em um conjunto de pontos de calibração, obtidos a partir da interseção dos segmentos reconhecidos e ajustados, alimentamos o método clássico de calibração de câmeras de Tsai [22], que nos retorna então os parâmetros extrínsecos e intrínsecos desejados.

Uma das características mais notáveis do método de Szenberg é sua capacidade de ajustar a calibração de uma câmera que teve seus parâmetros ligeiramente alterados como por exemplo em uma seqüência de imagens de vídeo, bastando para isso, efetuar somente um passo de reajuste dos segmentos reconhecidos e uma nova iteração do método de Tsai. Em nosso caso não precisamos desta funcionalidade, porém ela pode ser utilizada com bastante sucesso em um método de reconstrução que envolva reposicionamento de câmeras.

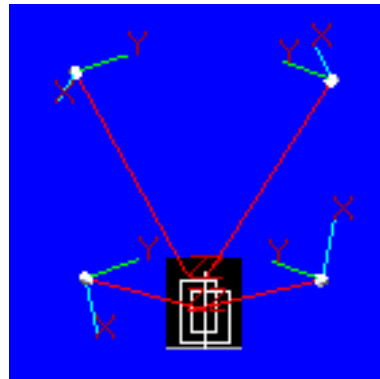
O método proposto por Szenberg também tem como vantagens a sua extrema robustez. Antes de sua utilização, experimentamos o método de calibração da biblioteca OpenCV [5], o qual não foi capaz de nos fornecer resultados satisfatórios. O nível de ruído presente nas imagens tornava impossível a detecção de todos os pontos de calibração necessários, inviabilizando assim o restante do processo. Quando, por outro lado, começamos a adotar o método de Szenberg, não mais precisamos nos preocupar com este problema já que os pontos de calibração são gerados automaticamente a partir do reconhecimento do modelo, o qual pode ser feito mesmo quando apenas um subconjunto dos segmentos que o compõem são detectados.



6.4(a): Um dos padrões de calibração



6.4(b): Padrão calibrado



6.4(c): Imagem tridimensional do posicionamento das câmeras

Figura 6.4: Resultado da calibração das câmeras do sistema

6.2

Câmera Tsai x Câmera OpenGL

Em nosso método, é necessário determinar as matrizes de transformação de câmera e de projeção, usadas pela OpenGL, a partir dos parâmetros intrínsecos e extrínsecos produzidos pelo método de calibração de Tsai. Vejamos como isso pode ser feito.

6.2.1

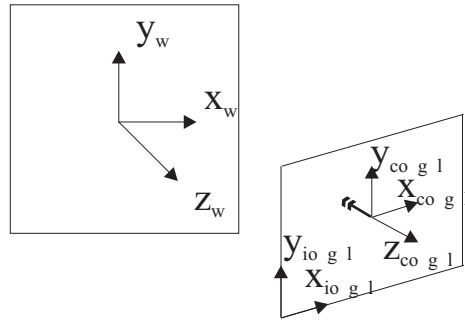
Determinando a matriz de transformação de câmera da OpenGL

Um cuidado especial deve ser tomado no cálculo das matrizes de transformação de câmera da OpenGL a partir dos parâmetros de rotação e translação fornecidos pelo método de Tsai. A relação entre a matriz de transformação de câmera da OpenGL, normalmente conhecida como *modelview matrix*, e os parâmetros extrínsecos determinados pelo método de Tsai, dependem dos sistemas de coordenadas nos quais estão definidas as coordenadas dos pontos de calibração no mundo e nas imagens.

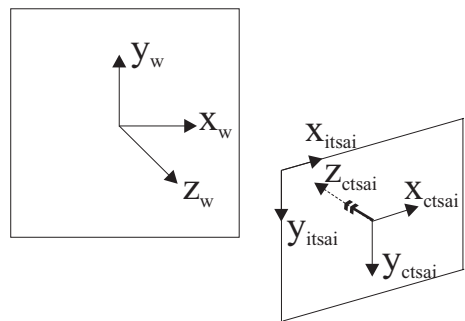
Tanto o sistema de coordenadas adotado pelo método de Tsai, quanto o adotado pela OpenGL, são dextrógiros. Por conseguinte, esperar-se-ia que os parâmetros de rotação e translação, obtidos a partir da calibração determinada pelo método de Tsai, pudessem automaticamente determinar uma *modelview* válida para o sistema de visualização da OpenGL. No entanto, a convenção adotada pela OpenGL é a de que a câmera aponta para o sentido negativo do eixo Z (Figura 6.5(a)), enquanto que, no sistema adotado por Tsai, considera-se que a câmera aponta para o sentido positivo do eixo Z . Além disso, o sistema de coordenadas da imagem, no método de Tsai é diferente do utilizado na OpenGL. Para que possa ser consistente com um sistema dextrógiro, o eixo Y no sistema de coordenadas da imagem adotada pelo método de Tsai aponta para baixo, ao invés de para cima, como na OpenGL (Figura 6.5(b)).

Por causa destas diferenças, os parâmetros de rotação r_1, r_2, \dots, r_9 e de translação t_x, t_y, t_z somente determinam uma matriz compatível com o sistema utilizado pela OpenGL, quando as coordenadas dos pontos de calibração nas imagens, forem obtidas de acordo com o sistema de coordenadas no espaço da imagem adotado pelo método de Tsai. Ainda assim, a matriz produzida a partir dos parâmetros calculados pelo método de Tsai não é exatamente uma matriz de transformação OpenGL, pois os eixos Y e Z encontram-se invertidos. A correção a ser efetuada é bastante

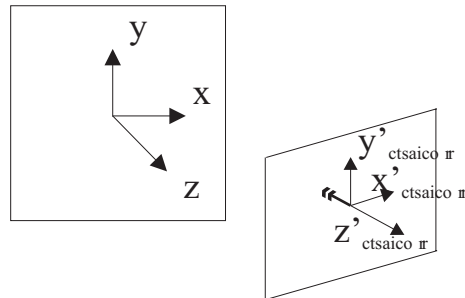
simples: basta inverter os eixos Y e Z (Figura 6.5(c)), operação esta que pode ser representada matricialmente através da expressão abaixo:



6.5(a): Sistema de coordenadas OpenGL



6.5(b): Sistema de coordenadas do método de Tsai



6.5(c): Sistema de coordenadas de Tsai corrigido

Figura 6.5: Sistemas de coordenadas utilizados na OpenGL e no método de Tsai

$$\begin{aligned}
 modelview &= M_{tsai}MC \\
 \begin{pmatrix} r_1 & -r_2 & -r_3 & t_x \\ r_4 & -r_5 & -r_6 & t_y \\ r_7 & -r_8 & -r_9 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} &= \begin{pmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 & \qquad \qquad \qquad (6.2.1)
 \end{aligned}$$

6.2.2

Determinando a matriz de projeção da OpenGL

A matriz de projeção da OpenGL pode ser montada manualmente com base nos parâmetros intrínsecos produzidos pela rotina de calibração. No entanto, como esta matriz é uma matriz normalizada, é bem mais simples utilizar as próprias rotinas da OpenGL dedicadas a sua especificação.

Em nossa implementação especificamos a matriz de projeção de forma direta, através da chamada da função *gluPerspective*, passando como argumentos os valores do *campo de visão*, da *razão de aspecto*, do *plano mais próximo* e do *plano mais distante*. O *campo de visão* e *razão de aspecto* são obtidos a partir das equações abaixo:

$$\begin{aligned} \text{campo de visão} &= \frac{2 * \arctan(\text{altura da imagem} * 1/2) * \pi}{\text{distância focal} * 180} \\ \text{razão de aspecto} &= \frac{\text{largura da imagem}}{\text{altura da imagem}} \end{aligned} \quad (6.2.2)$$

Já o *plano mais próximo* e o *plano mais distante* podem ser especificados arbitrariamente, desde que o frustum determinado contenha a cena desejada.

6.3

Calibração do fundo da cena

A calibração do fundo da cena é uma das etapas mais importantes para a realização do processo de reconstrução no sistema que propomos, pois é através dela que podemos segmentar o objeto de interesse nas imagens de entrada.

Para que possamos segmentar os objetos de interesse nas imagens, é necessário modelar o fundo da cena visto por cada uma das câmeras. Isto pode ser feito, para cada câmera, através da captura de imagens da cena sem os objetos que desejamos reconstruir. Esta abordagem, no entanto, não funciona, pois as intensidades de cada uma das componentes de cor dos pixels de uma determinada imagem são modificadas a cada instante de tempo por ruído aleatório, tornando impossível a utilização de um modelo determinístico. Por esta razão, precisamos modelar as imagens do fundo da cena estatisticamente.

O valor de uma dada componente de cor, digamos ρ em um certo pixel de uma imagem com ruído, pode ser modelado, apropriadamente, através

de uma variável aleatória com uma distribuição aproximada normal com uma média μ e variância σ^2 desconhecidas.

Como não conhecemos os parâmetros da distribuição de tal variável aleatória, precisamos estimá-los de alguma forma através de um conjunto de amostras. Em nosso caso, optamos por modelar cada uma das imagens do fundo da cena, através de matrizes onde cada elemento é um intervalo de confiança *ico* para a média μ em cada pixel. Cada um destes intervalos de confiança é calculado com base no erro entre uma estimativa pontual \bar{X} , dada pela média de um conjunto de n amostras tomadas em diferentes instantes de tempo, e a média populacional μ . Este erro, segundo o *Teorema Central do Limite*, tem sua distribuição aproximada dada por $e = (\bar{X} - \mu) \sim N(0, \sigma_{\bar{x}}^2)$ onde, $\sigma_{\bar{x}}^2 = \sigma^2/n$

Logo, a probabilidade de que tal erro e seja menor que uma determinada magnitude, dada por um múltiplo do desvio padrão amostral, é

$$P(|e| < \delta\sigma_{\bar{x}}) = P(|\bar{X} - \mu| < \delta\sigma_{\bar{x}}) = NC,$$

onde NC é o nível de confiança.

Reescrevendo esta afirmação probabilística como

$$P(\bar{X} - \delta\sigma_{\bar{x}} < \mu < \bar{X} + \delta\sigma_{\bar{x}}) = NC \tag{6.3.1}$$

podemos extrair o intervalo com o nível de confiança NC desejado

$$ico = \bar{X} \pm \delta\sigma_{\bar{x}} \tag{6.3.2}$$

Suponha agora que temos n componentes de cor para cada elemento de uma imagem, e que desejamos encontrar intervalos de confiança para cada componente ρ^k de maneira que a probabilidade de que todos eles contenham as respectivas médias populacionais seja igual a um determinado valor p . Neste caso, considerando que os erros são independentes temos que

$$\begin{aligned}
 \prod_{k=1}^n P(|e_{\rho^k}| < \delta\sigma_{\overline{x\rho^k}}) &= \prod_{k=1}^n P(|\overline{X_{\rho^k}} - \mu_{\rho^k}| < \delta\sigma_{\overline{x\rho^k}}) \\
 &= \prod_{k=1}^n P(\overline{X_{\rho^k}} - \delta\sigma_{\overline{x\rho^k}} < \mu_{\rho^k} < \overline{X_{\rho^k}} + \delta\sigma_{\overline{x\rho^k}}) \\
 &= p
 \end{aligned} \tag{6.3.3}$$

Se assumirmos que as probabilidades são as mesmas para cada componente de cor, então cada intervalo pode ser obtido de modo análogo com um nível de confiança $NC = \sqrt[n]{p}$.

No caso em que trabalhamos com as três componentes r , g e b , os intervalos de confiança que contêm simultaneamente as médias μ_r , μ_g e μ_b , em um dado pixel, com um nível de confiança de 95%, são dados respectivamente por:

$$\begin{aligned}
 icor &= \overline{Xr} \pm 2,39\sigma_{\overline{xr}} \\
 icog &= \overline{Xg} \pm 2,39\sigma_{\overline{xg}} \\
 icob &= \overline{Xb} \pm 2,39\sigma_{\overline{xb}}
 \end{aligned}$$

Devido ao ruído excessivo introduzido nas imagens obtidas através de webcams, é necessário obter um grande número de amostras temporais para que possamos obter estatísticas confiáveis para cada pixel. O ruído é notavelmente percebido em regiões de alta frequência, onde praticamente é impossível gerar informações com variabilidade bem comportada.

6.4 Segmentação com base em mapas de intervalos de confiança

A captura das imagens é realizada através de uma média temporal para que possamos eliminar parte do ruído aleatório introduzido pelo sistema de aquisição. Em nosso caso, utilizamos uma média de quatro quadros para cada imagem capturada.

Após a captura das imagens de entrada, precisamos segmentar os objetos de interesse com base nas matrizes de intervalos de confiança que modelam as imagens de fundo vistas por cada uma das câmeras.

A segmentação é bastante simples, consistindo em teste que verifica se os valores das componentes de cor de cada pixel da imagem capturada estão dentro do intervalo de confiança correspondente. Caso o teste seja bem

sucedido, isto é, se o valor da componente de cor de um pixel estiver dentro do intervalo de confiança desejado identificamo-lo como pixel de fundo, caso contrário, identificamo-lo com parte de uma região correspondente aos objetos de interesse nas imagens.

Quando aplicamos tal teste para segmentar as imagens, verificamos que as regiões de sombra e penumbra pertencentes ao fundo da cena acabavam sendo classificadas como parte dos objetos de interesse. Para evitar tais problemas criamos novas matrizes de intervalos de confiança, desta vez para componentes de cromaticidade, e adicionamos um novo teste ao processo de segmentação.

Desta forma, para cada pixel classificado como objeto de interesse, segundo o teste baseado em intervalos de confiança para as componentes de cor, verificamos as suas componentes de cromaticidade. Se estas componentes estiverem dentro dos respectivos intervalos de confiança, classificamos o pixel em questão como de fundo.

Após a classificação, aplicamos um filtro morfológico nos mapas de segmentação resultantes com o objetivo de remover pequenos elementos espúrios e fechar pequenos buracos no interior das regiões correspondentes aos objetos.

6.5 Reconstrução

A etapa de reconstrução é realizada através dos algoritmos de escultura do espaço que foram descritos neste trabalho. É necessário apenas a especificação da resolução desejada (resolução máxima no caso do algoritmo de escultura do espaço adaptativo) e a região do espaço que determina o volume de reconstrução inicial.