3
O Método Proposto

Este capítulo apresenta o método desenvolvido por esta pesquisa, que é composto por diversas técnicas integradas, formando um fluxo de execução cuja entrada é constituída por uma foto digital e um modelo computacional de edificação e cuja saída são parâmetros extrínsecos e intrínsecos de uma câmera virtual.

O objetivo central do método é prover meios para recuperar modelos de câmeras a partir de fotos com o auxílio do usuário. Para simplificar a interação, foi necessário desenvolver técnicas que processam tanto a geometria do modelo quanto as imagens de entrada.

Abaixo há uma explanação **introdutória** de cada uma das etapas do método:

1. Os dois primeiros passos são independentes entre si e, por isso, podem ser executados separadamente. Porém, ambos os processos devem ter sido concluídos para que ocorram as etapas seguintes. Esses passos iniciais são: (a) calibração de câmera; (b) carregamento e processamento da geometria.

   (a) A calibração de câmera consiste na reconstrução de parâmetros intrínsecos, que são calculados a partir de uma imagem fornecida como entrada. Se essa imagem contiver informações EXIF apropriadas, a geometria da câmera pode ser reconstruída e a calibração ocorre automaticamente. Caso tais informações não estejam presentes, o usuário deve então marcar segmentos em direções principais da cena para calibrar a imagem por pontos de fuga;

   (b) O carregamento e processamento da geometria iniciam-se quando um modelo virtual de uma edificação é fornecido como entrada. Tal modelo, produzido em ferramentas como 3D Studio Max®, é carregado através da leitura de um arquivo de dados que o representa e sofre então um processamento automático onde determinadas arestas são extraídas de sua malha. Esse processamento resulta, assim, em uma lista de arestas, chamadas aqui de arestas estruturais do

PUC-Rio - Certificação Digital Nº 0611927/CA

modelo, que respeitam um conjunto de critérios geométricos. Tal lista é então utilizada nos cálculos das etapas posteriores, de tal modo que a geometria original do modelo não mais é usada para este fim — ela é utilizada apenas para decoração da cena e para efetuar a etapa do teste de oclusão, descrita a diante.

2. Como próximo passo, o usuário fornece manualmente uma translação e uma rotação que são aplicados à câmera virtual para que o modelo — composto agora apenas pelas arestas estruturais pré-selecionadas — alinhe-se aproximadamente com imagem de entrada. O alinhamento ocorre porque a imagem de entrada é mapeada no plano de projeção da câmera calibrada, de tal forma que, ao alterar seu posicionamento, a posição relativa entre a imagem e o modelo também se altere — do ponto de vista do observador, a imagem está sempre parada no espaço da tela e o modelo se move. A partir deste posicionamento inicial, o sistema ajusta corretamente modelo e imagem através das técnicas descritas nas etapas que seguem;

3. Com a posição provida pelo usuário, os segmentos selecionados na fase de processamento da malha passam por um teste de oclusão em relação ao modelo original. Dessa forma, apenas os segmentos visíveis a partir da posição inicial fornecida são usados no processo de localização de correspondências entre modelo e imagem;

4. As arestas que passaram no teste de oclusão são então projetadas no espaço da imagem utilizando os parâmetros intrínsecos de câmera já recuperados na primeira etapa;

5. Através de uma busca (automática) por arestas na vizinhança de cada posição bidimensional obtida pela projeção, são selecionados indicadores de segmentos da imagem que correspondem aos segmentos extraídos do modelo;

6. As correspondências entre modelo e imagem são finalmente exibidas para o usuário através de marcações visuais, para que ele possa então descartá-las, aprová-las ou complementá-las. A complementação realizada pelo usuário é composta de novas marcações feitas apenas na imagem. A partir destas marcações e da posição inicial de câmera fornecida pelo usuário, é possível calcular por proximidade as arestas correspondentes do modelo. Somente em casos onde não há possibilidade de se detectar correspondências por proximidade (devido à baixa qualidade da imagem) é que se faz necessária a associação manual direta entre modelo e imagem;

PUC-Rio - Certificação Digital Nº 0611927/CA

A complementação é necessária em casos onde não é possível detectar arestas nas três direções principais da cena, o que caracteriza um requisito fundamental para a etapa de ajuste de câmera (minimização). Tal fato pode ocorrer principalmente devido à ruído ou por oclusão. Assim, o usuário complementa o conjunto de arestas detectadas marcando novas arestas em regiões em que nenhum ou poucos segmentos foram encontrados;

7. Utilizando o conjunto de associações entre modelo e imagem já aprovado pelo usuário e o posicionamento inicial de câmera, um algoritmo de minimização é utilizado para ajustar o posicionamento da câmera de modo a alinhar os segmentos da imagem e do modelo que foram previamente correlacionados;

8. Com a posição e orientação calculadas, a câmera é registrada para que seja possível navegar entre as diversas fotos que podem ser fornecidas como entrada. O processo se repete — com exceção do processamento da malha do modelo — caso uma nova foto seja fornecida como entrada.

As etapas e o fluxo de execução do método podem ser visualizados no diagrama 3.1.

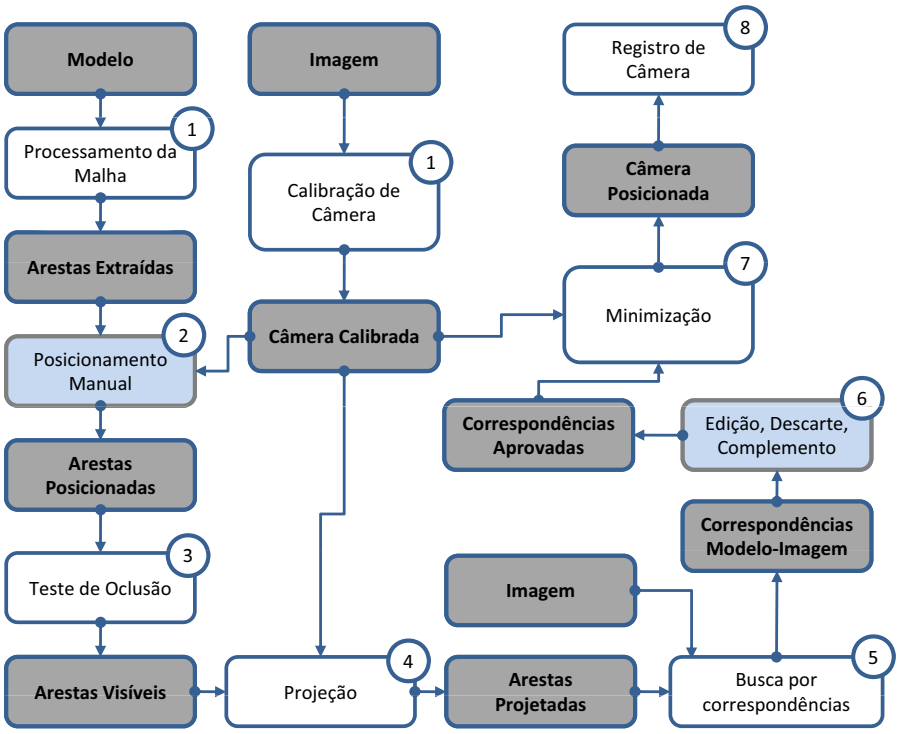
O objetivo final desta pesquisa é realizar casamentos entre modelos virtuais e imagens reais de edificações. Para tanto, a recuperação de câmera é tida como passo principal.

Como já foi exposto, o processo de identificação de características entre imagem e modelo é um passo fundamental para realizar a calibração e a recuperação dos parâmetros externos de câmera. Contudo, tal processo constitui um dos principais problemas da visão computacional e, assim, representa um grande obstáculo para o objetivo final de reconstruir câmeras virtuais a partir de fotos. Por este motivo, a abordagem aqui adotada estabelece que o usuário tem um papel importante no gerenciamento dessas associações, principalmente em casos onde, por fatores inerentes a qualidade da imagem de entrada, não é possível detectar uma quantidade suficiente de segmentos.

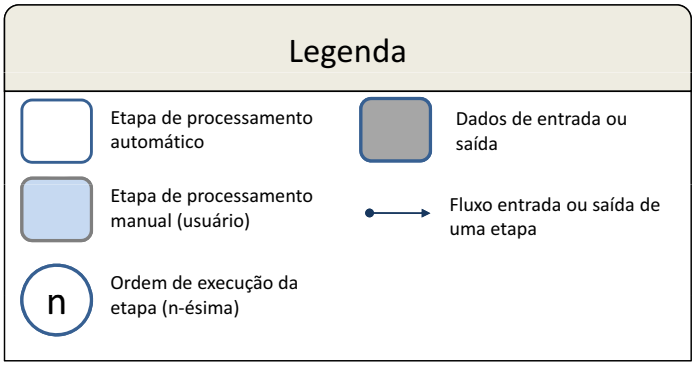
O método proposto visa auxiliar ao máximo as ações do usuário, de tal forma que, em casos simples — onde há pouco ruído nas imagens — ele possivelmente não precise fornecer marcações extras para que a aplicação recupere a câmera. Nesses casos, o usuário precisará apenas fornecer um posicionamento inicial de câmera apropriado.

As seções que seguem, se destinam a explicar cada uma das etapas de processamento do método recém-introduzidas.

PUC-Rio - Certificação Digital Nº 0611927/CA



3.1(a): Processo de funcionamento geral do método proposto. Todo o processo se inicia com um imagem e com um modelo virtual fornecidos como entrada. A numeração indica a ordem obrigatória de execução das etapas. Etapas com números iguais são independentes e não relacionadas (podem ser executadas separadamente ou ao mesmo tempo).



3.1(b): Legenda do diagrama acima.

Figura 3.1: Diagrama esquemático do funcionamento geral do método proposto.

3.1 Métodos para Calibração de Câmera

O processo de calibração tem por objetivo recuperar parâmetros intrínsecos de câmera, a partir da foto de uma cena real digitalizada, para gerar uma matriz de K de calibração. Dois métodos compõem a estratégia de calibração neste trabalho: o de calibração por pontos de fuga e o de calibração

PUC-Rio - Certificação Digital Nº 0611927/CA

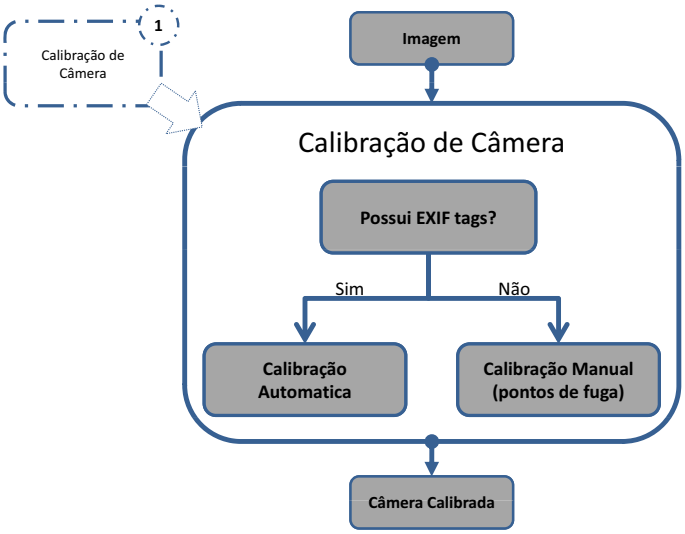


Figura 3.2: Etapa de calibração vista em mais detalhes. Caso não haja na imagem de entrada informações EXIF que permitam a calibração automática da câmera, ela ocorre manualmente utilizando-se marcações na própria imagem para se obter pontos de fuga e, então, calcular a matriz de calibração K.

por EXIF tags. Este último não integra nenhuma técnica específica da visão computacional. Ele é apenas uma forma conveniente e eficiente de recuperar o estado interno de uma câmera a partir das informações que ela própria fornece através do padrão de meta-dados EXIF. Como será visto, quando tais meta-dados estão disponíveis em um arquivo de imagem, é possível recuperar a geometria completa da câmera com grande precisão.

Este trabalho utiliza o modelo *pinhole* CCD (*Charge-Coupled Device*) de câmera, que é um modelo finito parecido com o clássico modelo *pinhole*, mas que leva em consideração que os pixels do plano de projeção podem não ser quadrados.

Como mencionado, uma câmera calibrada é aquela na qual se pode relacionar um ponto qualquer em seu plano de projeção ao respectivo raio $\mathbf{d} = \mathtt{K}^{-1}\mathrm{x}$ que parte de seu centro óptico. A matriz K nesta equação é chamada matriz de calibração e é computada a partir de parâmetros intrínsecos da câmera. A seção a seguir apresenta quais são estes parâmetros, como eles se relacionam com o modelo de camera clássico, o modelo *pinhole*, e o que representa a matriz K dentro desse modelo.

3.1.1 O Modelo de Câmera Pinhole

O modelo *pinhole* é um modelo finito de câmera na qual pontos no mundo são mapeados para um plano através de uma projeção ideal — sem

PUC-Rio - Certificação Digital Nº 0611927/CA

se considerar distorções causadas por lentes ou dispositivos eletrônicos de captura. Neste modelo, uma câmera é constituída por um centro de projeção **C**, posicionado na origem de um sistema euclidiano chamado Sistema de Coordenadas da Câmera, um plano de projeção ou plano focal $Z = f$ (onde f é a distância focal da câmera medida em unidades do mundo) e um eixo principal ou eixo óptico, que vai do centro de projeção e passa pelo centro do plano focal (ponto principal). O centro de projeção **C** é um ponto qualquer no espaço Euclidiano $\mathbb{R}^3$ que foi adotado como sendo a origem do sistema da câmera. A Figura a 3.3 ilustra este modelo.

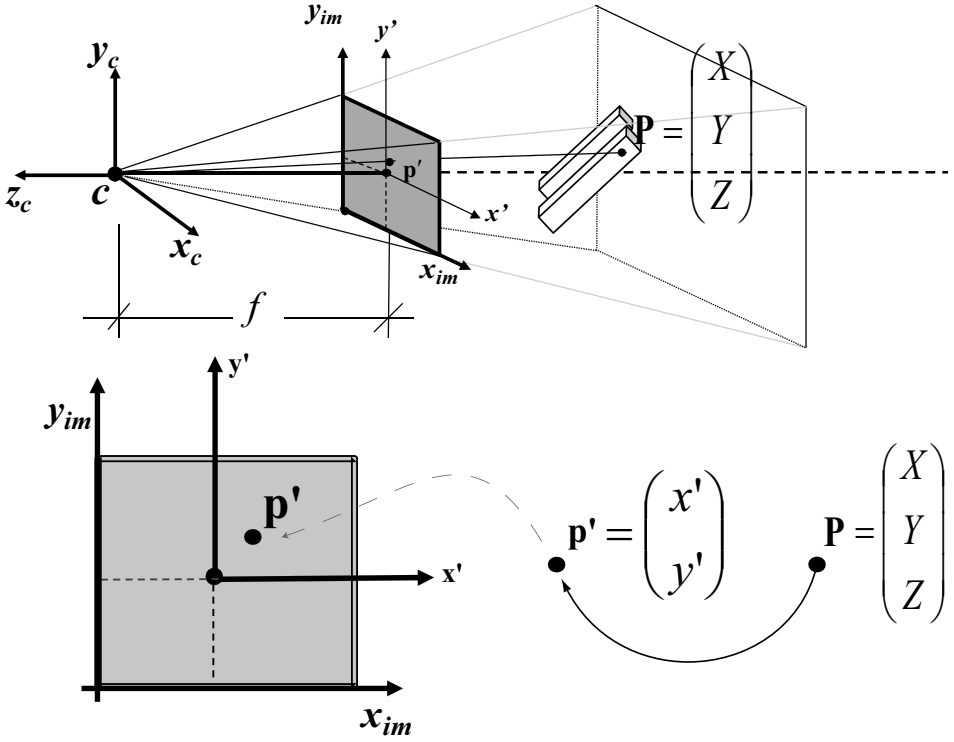


Figura 3.3: Modelo de câmera *pinhole*. Nesse modelo um ponto P do mundo é projetado para um ponto p' do plano focal através de um mapeamento simples obtido por semelhança de triângulos.

por semelhança de triângulos, é simples calcular as coordenadas de um ponto no mundo $(\mathrm{X}, \mathrm{Y}, \mathrm{Z})$ no plano de projeção da câmera:

$$(\mathrm{X}, \mathrm{Y}, \mathrm{Z}) \mapsto (f\mathrm{X}/\mathrm{Z}, f\mathrm{Y}/\mathrm{Z})^T \tag{3-1}$$

que representa um mapeamento do espaço Euclidiano $\mathbb{R}^3$ para um espaço Euclidiano $\mathbb{R}^2$.

Contudo, este mapeamento é não linear em função da profundidade de um ponto do mundo. Por isso, costuma-se representar convenientemente o

PUC-Rio - Certificação Digital Nº 0611927/CA

mesmo mapeamento através de uma transformação linear em Coordenadas Homogêneas, da seguinte forma:

$$\begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\mathrm{X} \\ f\mathrm{Y} \\ \mathrm{Z} \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} \tag{3-2}$$

As expressões 3-1 e 3-2, no entanto, não representam a forma geral de um mapeamento nesse modelo de câmera, pois assumem que a origem do sistema de coordenadas da imagem reside estritamente no ponto principal — por onde o eixo óptico corta o plano focal. No entanto, isso nem sempre é verdade, já que o sistema de coordenadas da imagem pode ter origem em qualquer outro ponto do plano de projeção da câmera (comumente no canto inferior esquerdo). Assim, o mapeamento geral de um ponto do mundo para um ponto da imagem no plano de projeção da câmera pode ser escrito como

$$(\mathrm{X}, \mathrm{Y}, \mathrm{Z}) \mapsto (f\mathrm{X}/\mathrm{Z} + P_x, f\mathrm{Y}/\mathrm{Z} + p_y)^T \tag{3-3}$$

onde (p_x, p_y) são as coordenadas do ponto principal do plano de projeção. A Figura 3.4 mostra o plano de projeção com diferentes sistemas de coordenadas para a imagem e para a câmera.

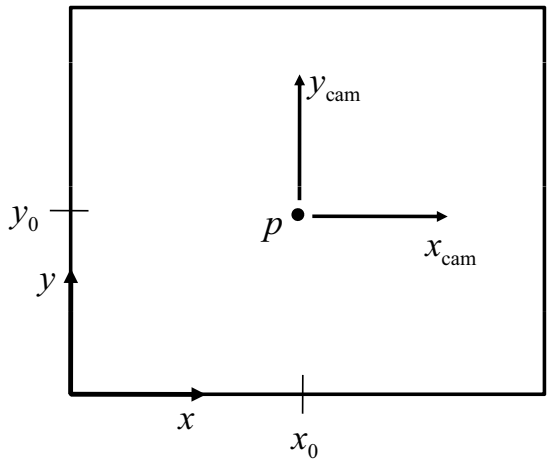


Figura 3.4: Sistema de coordenadas da imagem (x, y) e da câmera (x_{cam}, y_{cam}).

Assim, pode-se então escrever o mapeamento linear homogêneo como sendo:

PUC-Rio - Certificação Digital Nº 0611927/CA

$$\begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\mathrm{X} + p_x \\ f\mathrm{Y} + p_y \\ \mathrm{Z} \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} \tag{3-4}$$

A matriz $\mathtt{K}$ representa exatamente o mapeamento linear homogêneo mostrado na equação acima. Portanto, a matriz de calibração que representa o modelo *pinhole* de câmera é

$$\mathtt{K} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \tag{3-5}$$

Logo, quando se assume um modelo *pinhole* de câmera, têm-se apenas três parâmetros para reconstruir, que são f, p_x e p_y.

3.1.2 Câmeras de CCD

Embora o modelo *pinhole* seja usado como base para ambos os processos de calibração aqui tratados, ele não considera alguns fatores oriundos de propriedades físicas de câmeras digitais reais, tais como a possibilidade de existirem pixels que não sejam quadrados. No modelo recém exposto, o sistema de coordenadas da imagem é considerado como sendo simétrico em ambas as direções x e y do plano focal. No entanto, câmeras digitais reais são compostas por CCDs, que são micro-circuitos — responsáveis pela captura da luz — que compõem seus planos focais e que têm dimensões finitas possivelmente não simétricas. Logo, ao medir o sistema de coordenadas do plano do focal em pixels, deve-se considerar fatores de escala s_x e s_y aplicados a distância focal real da câmera, em unidades do mundo, que representam respectivamente as dimensões horizontal e vertical de um pixel também em unidades do mundo (Figura 3.5). O mapeamento linear homogêneo que leva em consideração estes fatores é escrito como

$$\mathtt{K} = \begin{bmatrix} \alpha_x & & p_x \\ & \alpha_y & p_y \\ & & 1 \end{bmatrix} \tag{3-6}$$

onde $\alpha_x = f/s_x$ e $\alpha_y = f/s_y$.

PUC-Rio - Certificação Digital Nº 0611927/CA

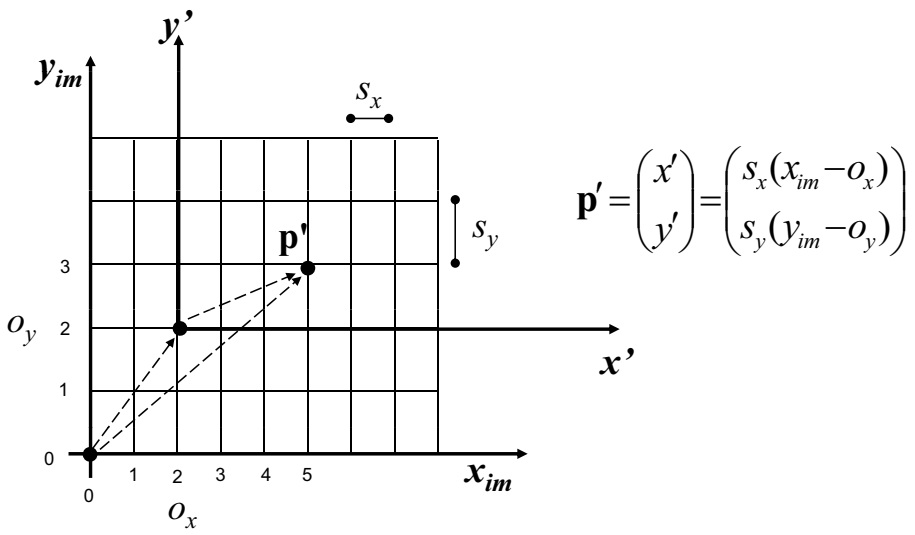


Figura 3.5: Mapeamento entre coordenadas da imagem (em pixels) e coordenadas do plano de projeção da câmera. A posição final no plano depende de um ponto adotado como origem (geralmente, mas não necessariamente, o centro do plano, o ponto principal) e do tamanho de um pixel na horizontal e na vertical.

A matriz de calibração pode ainda exibir um parâmetro s (*skew parameter*) que representa uma distorção de cisalhamento (aplicada ao eixo y do plano focal) causada por dispositivos eletrônicos. Tal distorção é, em geral, muito baixa em câmeras que possuem dispositivos de captura de boa qualidade [Hartley and Zisserman, 2003] e, por isso, este trabalho assume $s = 0$ para efeitos de simplificação.

Tendo apresentado o modelo de câmera adotado, são apresentados a seguir os métodos utilizados para calcular a matriz de calibração `K`. Este trabalho desenvolveu uma estratégia para calibração de câmera que é composta por duas técnicas distintas utilizadas de forma exclusiva. Essas técnicas são a calibração por meta-dados EXIF e a calibração por pontos de fuga. O critério para escolha entre uma dessas técnicas é simples: caso a imagem de entrada possua informações EXIF necessárias, o processo de calibração ocorre automaticamente através do uso direto destas. Em caso contrário, o usuário precisa fornecer um conjunto de segmentos de reta, corretamente posicionados no espaço da imagem, que é utilizado para calibrar a câmera através de pontos de fuga. Esse processo pode ser visualizado através dos diagramas 3.2 e 3.1(a).

Cada uma dessas técnicas exibe vantagens e desvantagens, que são listadas resumidamente abaixo:

PUC-Rio - Certificação Digital Nº 0611927/CA

Calibração por meta-dados EXIF:

– **Vantagens**

1. Funciona de modo totalmente automatizado (não necessita interação do usuário);
2. É potencialmente muito precisa, pois utiliza dados fornecidos pelo próprio fabricante da câmera real;
3. Permite recuperar completamente a geometria da câmera pois provê, além dos outros parâmetros, as dimensões de um pixel.

– **Desvantagens**

1. Pode não ser possível usá-lo em certas fotos (caso algumas tags EXIF não estejam disponíveis ou caso não haja cabeçalho EXIF algum).

Calibração por Pontos de Fuga:

– **Vantagens**

1. Pode sempre ser usado em qualquer foto cujas dimensões x, y e z apareçam (não depende de meta-dados EXIF);
2. Pode ser usado para calcular a orientação da câmera.

– **Desvantagens**

1. Necessita da interação do usuário (pela abordagem aqui usada);
2. Calibra a câmera em função de um fator de escala (recupera a distância focal em pixels);
3. Possui uma precisão muito inferior à do método de calibração usando EXIF tags, pois depende de marcações na imagem que, por ruído ou erro do usuário, não são totalmente precisas.

As seções que seguem apresentam estes dois métodos de calibração.

PUC-Rio - Certificação Digital Nº 0611927/CA

3.1.3
Calibração Usando EXIF Tags

As *tags* (ou rótulos) EXIF são um padrão para escrita e leitura de meta-dados que fornecem informações diversas em relação ao estado de uma câmera digital. Tais dados são gravados pelas próprias câmeras em seus arquivos de imagens codificados em formatos como JPEG ou TIFF. Essas *tags* são duplas de dados formadas por uma chave textual e um valor (que pode ser escalar ou textual) e representam propriedades relativas ao estado interno da câmera no momento em que a foto foi obtida e às suas características físicas, como densidade de pixels e tamanho do plano de projeção.

O site especializado da biblioteca Exiv2[1] fornece uma extensa lista das tags disponíveis em diversos formatos de arquivo digital e de marcas de câmeras. Essa biblioteca foi utilizada neste trabalho para efetuar a leitura de informações EXIF nas imagens de entrada.

Dentre as inúmeras *tags* disponíveis, as que são de especial interesse para esta pesquisa são aquelas que informam a densidade (em pixels por unidade de medida) do plano de projeção em suas duas dimensões, a distância focal da câmera no momento em que a foto foi tirada, a unidade usada para medir a densidade do plano de projeção e a quantidade total de pixels ao longo da dimensão y do plano de projeção. Tais *tags* representam parâmetros internos da câmera, que são mostrados pela Figura 3.6 e descritos na Tabela 3.1.

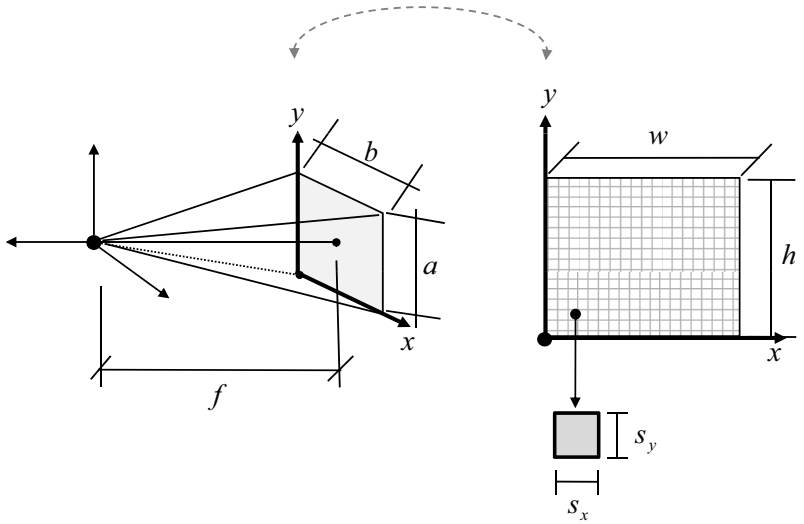


Figura 3.6: Simbologia atribuída aos parâmetros da câmera que são mapeados pelas tags EXIF. O significado de cada símbolo e suas respectivas chaves EXIF são mostradas nas tabelas 3.1 e 3.2.

[1]Para mais informações a respeito dos meta-dados EXIF e sobre a biblioteca Exiv2 consulte http://www.exiv2.org/.

PUC-Rio - Certificação Digital Nº 0611927/CA

Parâmetros de câmera e as *tags* EXIF		
Parâmetro	*Tag* EXIF	Descrição
f	[*Exif.Photo.FocalLength*]	Distância focal em milímetros
h	[*Exif.Photo.PixelYDimension*]	Altura do plano de projeção em pixels
d_x	[*Exif.Photo.FocalPlaneXResolution*]	Densidade em x do plano focal
d_y	[*Exif.Photo.FocalPlaneYResolution*]	Densidade em y do plano focal

Tabela 3.1: Parâmetros de câmera que são obtidos diretamente das *tags* EXIF (vide Figura 3.6).

A unidade utilizada medir as densidades d_x,d_y é obtida a partir de outra *tag* EXIF que é mostrada adiante.

Os demais parâmetros a, s_x e s_y, mostrados na Figura 3.6, não são obtidos diretamente através das chaves EXIF, mas são calculados a partir delas, como é mostrado a na próxima subseção. A descrição desses parâmetros é feita na tabela abaixo.

Parâmetros de câmera calculados a partir das *tags* EXIF	
Parâmetro	Descrição
a	Altura do plano focal em milímetros
s_x	Dimensão de um pixel do plano focal ao longo de seu eixo x, em milímetros
s_y	Dimensão de um pixel do plano focal ao longo de seu eixo y, em milímetros

Tabela 3.2: Descrição dos parâmetros que são calculados a partir das *tags* EXIF.

PUC-Rio - Certificação Digital Nº 0611927/CA

PUC-Rio - Certificação Digital Nº 0611927/CA

Utilizando as tags para a calibração

Tendo em mãos a densidade de pixels do plano de projeção e a distância focal em milímetros, resta apenas recuperar a unidade usada no cálculo de tal densidade para se que seja possível obter o tamanho real do plano focal, o que é feito através da chave [*Exif.Image.ResolutionUnit*]. Em geral, as câmeras utilizam a unidade de polegadas para medir a densidade de pixels, o que torna necessário uma conversão de unidade para milímetros, que é a unidade adotada como padrão neste trabalho.

É interessante ressaltar que a densidade de pixels real do plano de projeção **não** é igual à densidade de pixels por unidade na imagem, porque um pixel do plano de projeção da câmera real é representado por um único CCD e a imagem final provida pela câmera é resultado de um processamento com possível perda e compressão.

Com a densidade de pixels e a distância focal em mãos, é possível então descobrir o tamanho real de um pixel da câmera e desvendar os parâmetros s_x e s_y, bastando para isso utilizar a chave [*Exif.Photo.PixelYDimension*], que fornece a quantidade de pixels na dimensão y do plano de projeção. Assim, os parâmetros da tabela 3.2 podem ser calculados, como mostrado abaixo.

$$\begin{aligned} d_x(mm) &= d_x(pol)/25.4 \\ s_x &= 1/d_x(mm) \\ &\dots \\ s_y &= 1/d_y(mm) \\ a &= s_y \times h \end{aligned}$$

Com esses dados em mãos, pode-se reconstruir totalmente a geometria interna da câmera. De fato, pode-se até calcular seu ângulo de abertura em y ($fovy$) utilizando a altura real do plano de projeção e a distância focal, também em unidades do mundo:

$$fovy = 2 \times \arctan\left(\frac{a \times 0.5}{f}\right)$$

3.1.4 Calibração A Partir de Pontos de Fuga

O processo de calibração por pontos de fuga é realizado aqui manualmente. Tal processo envolve duas etapas básicas: 1) obtenção da posição dos

pontos de fuga a partir de marcações feitas pelo usuário nas imagens; 2) cálculo da matriz `K` de calibração e, opcionalmente, de uma matriz `R` de orientação a partir dos pontos de fuga obtidos.

Esta pesquisa analisa três métodos para se calcular a posição desses pontos de fuga a partir das marcações feitas por um usuário: i) O método simples; ii) O método usual; iii) O método robusto; que serão descritos a seguir. É importante notar que, embora as marcações feitas nas imagens sejam segmentos de reta, elas serão tratadas como retas em alguns casos para realizar certos cálculos específicos (isso sempre é possível, já que por um segmento de reta passa apenas uma reta).

1. Método simples: Usuário marca apenas duas retas por direção principal da cena e, através de interseção simples entre reta, o ponto de fuga é calculado. Este método é o menos preciso porque utiliza um conjunto mínimo de dados — que contêm erro — para recuperação do ponto de fuga. Isso faz com que a quantidade de erro por segmento provido pelo usuário se torna alta (baixa distribuição do erro).

2. Método usual: Usuário marca na imagem diversos segmentos que representam segmentos paralelos a cada uma das três direções principais do espaço do mundo e, por mínimos quadrados, o centróide das distâncias para todas as retas é calculado como sendo o ponto de fuga formado por essas retas [Alvarez, 2002] (Figura 3.7). Dadas n retas, o ponto de fuga é obtido minimizando-se o somatório:

$$min \sum_{i=1}^{n} (a_i x + b_i y + c_i)^2$$

onde $a_i x + b_i y + c_i = 0$ é a representação de cada reta. A minimização acima pode ser resolvida por um sistema linear matricial da forma $Ax = b$ utilizando decomposição por valores singulares (SVD). Este método é, em geral, o que é mais comumente utilizado para cálculo de pontos de fuga.

3. Método robusto: Funciona como o método usual, utilizando diversos segmentos por direção principal da cena, mas calcula o ponto de fuga através de uma parametrização diferente. A idéia é que as retas que formam o ponto de fuga devem realmente ter apenas este ponto em comum. Assim, para cada marcação do usuário é associada uma equação de reta que contém o ponto de fuga. Esse é então obtido minimizando-se o erro entre as retas criadas e os segmentos marcados [Hartley and Zisserman, 2003] (Figura 3.8).

PUC-Rio - Certificação Digital Nº 0611927/CA

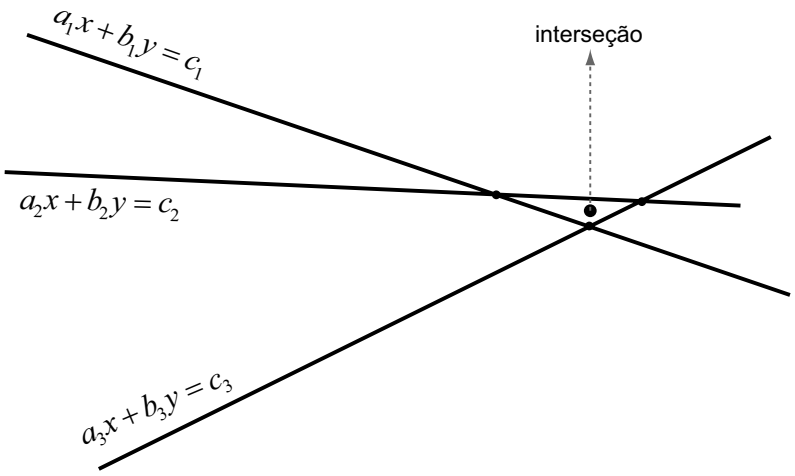
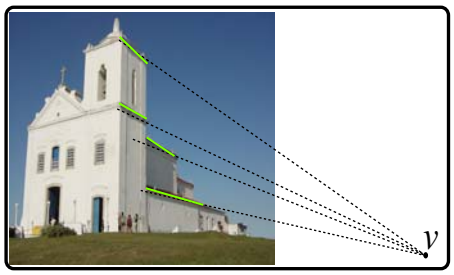
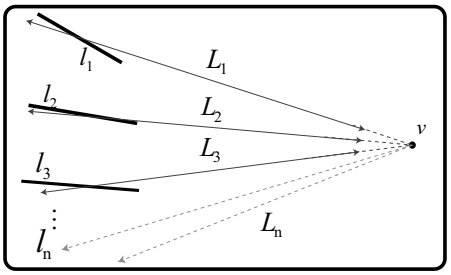


Figura 3.7: Interseção entre diversas retas. O ponto de interseção é calculado utilizando-se mínimos quadrados.

PUC-Rio - Certificação Digital Nº 0611927/CA



3.8(a): Representação dos pontos de fuga utilizando retas que se interceptam em um único ponto.



3.8(b): Ilustração do processo para cálculo de ponto de fuga utilizando n retas. Atribui-se uma reta $(L_1, L_2, L_3, \ldots, L_n)$ para cada um dos segmentos da imagem. O ponto de fuga pertence a todas as retas e sua posição é obtida minimizando-se as distâncias entre elas e esses segmentos.

Figura 3.8: Cálculo de pontos de fuga por múltiplas retas utilizando uma parametrização mais robusta.

Existem outros métodos para calcular pontos de fuga utilizando câmeras não calibradas, como o proposto por Roberto Cipolla et al em [Cipolla et al., 1999b]. Contudo, não é o objetivo desta dissertação esgotar tal assunto e, por isso, são analisados apenas os três métodos recém-apresentados.

A próxima subseção mostra como utilizar os pontos de fuga para recuperar a geometria interna da câmera.

Usando Pontos de Fuga Para Calibração

É possível calibrar a câmera a partir de uma única imagem sem o conhecimento de qualquer métrica da cena se esta exibir elementos que

forneçam restrições geométricas, tais como pontos de fuga. Para encontrar a matriz $\mathtt{K}$, é preciso identificar três pontos de fuga relativos a três direções ortogonais do mundo. Esse processo se baseia no fato de que pontos no infinito no espaço $\mathbb{P}^3$ são mapeados para pontos finitos no espaço $\mathbb{P}^2$ através de uma transformação projetiva. A imagem de um ponto no infinito não depende da translação da câmera, apenas de sua rotação. Seja $\mathtt{P}_{3\times4}$ uma matriz de transformação projetiva, tal que $\mathtt{P}_{3\times4} = [\mathtt{K}][\mathtt{R}][\mathtt{T}]$,

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \mathtt{P}_{3\times4} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 0 \end{bmatrix} = [\mathtt{K}][\mathtt{R}] \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 0 \end{bmatrix}$$

onde $\mathtt{P}_{3\times4}$ é uma matriz de projeção. Pode-se notar que a translação embutida em $\mathtt{P}$ não influencia a posição final (na imagem) de um ponto no infinito.

Assim, para três pontos de fuga em três direções ortogonais no mundo, tem-se:

$$\begin{bmatrix} w_1x_1 & w_2x_2 & w_3x_3 \\ w_1y_1 & w_2y_2 & w_3y_3 \\ w_1 & w_2 & w_3 \end{bmatrix} = \mathtt{P}_{3\times4} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \mathtt{KR} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \mathtt{KR} \qquad (3\text{-}7)$$

onde cada w_i é um fator de escala desconhecido, x_i, y_i são as coordenadas dos pontos de fuga no sistema de coordenadas da imagem e $\mathtt{R}$ é a matriz de rotação da câmera que determina sua orientação em relação à origem do mundo. Assim, fica claro que pontos de fuga fornecem restrições geométricas fortes, como pôde ser observado algebricamente pela equação acima. A seguir essa restrição será utilizada para recuperar a orientação e a matriz $\mathtt{K}$ de calibração.

Cálculo da Matriz de Calibração K

Como $\mathtt{R}$ é ortonormal:

$$\mathtt{R}^{-1} = \mathtt{R}^T$$

$$\mathtt{R}^T\mathtt{R} = \mathtt{R}\mathtt{R}^T = \mathtt{I}$$

Elevando cada lado de 3-7 ao quadrado:

PUC-Rio - Certificação Digital Nº 0611927/CA

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix} \left(\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix} \right)^T = \mathtt{KR}(\mathtt{KR})^T$$

Como $\mathtt{KR}(\mathtt{KR})^T = \mathtt{K}(\mathtt{RR}^T)\mathtt{K}^T = \mathtt{KK}^T$, tem-se:

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} w_1^2 & 0 & 0 \\ 0 & w_2^2 & 0 \\ 0 & 0 & w_3^2 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}^T = \mathtt{KK}^T \tag{3-8}$$

Se adotarmos certas premissas em relação à natureza da câmera, tais como sistema de eixos ortogonal da imagem (sem cisalhamento) e *Aspect Ratio* conhecido, o número de parâmetros desconhecidos na matriz K se reduz a 4 e temos K da seguinte forma:

$$\mathtt{K} = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

onde $\alpha_x = -f/s_x$, $\alpha_y = -f/s_y$, f é a distância focal medida em pixels e s_x, s_y são respectivamente as dimensões em x e em y de um pixel da câmera. Como a princípio não sabemos as dimensões exatas de um pixel, podemos assumir *a priori* um pixel quadrado (*Aspect Ratio* igual a 1). Isso faz $\alpha_x = \alpha_y$ e faz com que passemos a medir f em pixels. Temos então a matriz K com 3 parâmetros, da seguinte forma:

$$\mathtt{K} = \begin{bmatrix} \alpha & 0 & x_0 \\ 0 & \alpha & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Expandindo os termos à esquerda e substituindo K na equação 3-8, obtém-se:

$$\begin{bmatrix} w_1^2 x_1 & w_1^2 y_1 & w_1^2 \\ w_2^2 x_2 & w_2^2 y_2 & w_2^2 \\ w_3^2 x_3 & w_3^2 y_3 & w_3^2 \end{bmatrix} = \begin{bmatrix} \alpha^2 + x_0^2 & x_0 y_0 & x_0 \\ x_0 y_0 & \alpha^2 + x_0^2 & y_0 \\ x_0 & y_0 & 1 \end{bmatrix} \tag{3-9}$$

de onde pode-se obter 6 equações (uma para cada termo da matriz simétrica) para encontrar os parâmetros w_i, x_0 e y_0.

PUC-Rio - Certificação Digital Nº 0611927/CA

Contudo, através da interpretação geométrica dessas equações, é mais simples calcular tais parâmetros. Como é mostrado na próxima subseção, os parâmetros w_i correspondem às coordenadas baricêntricas do ortocentro do triângulo formado pelos pontos de fuga no espaço da imagem. O ortocentro, por sua vez, representa a projeção do centro óptico da câmera na imagem e também o centro x_0, y_0 da imagem. Logo, é possível calcular todos os parâmetros necessários através da interpretação geométrica dos pontos de fuga na imagem, como apresentado a seguir.

Interpretação Geométrica

As coordenadas de três pontos de fuga correspondentes a direções ortogonais no espaço do mundo, formam raios perpendiculares entre si partindo do centro de projeção. Isso pode ser verificado pela restrição de perpendicularidade entre pontos de fuga e a imagem da cônica no infinito, já que dois pontos $\mathbf{p}_1, \mathbf{p}_2$ na imagem têm raios perpendiculares se e somente se [Hartley and Zisserman, 2003]:

$$\mathbf{p}_1 \omega \mathbf{p}_2 = 0 \tag{3-10}$$

onde ω é a imagem da cônica no infinito e $\omega = (KK^T)^{-1}$.

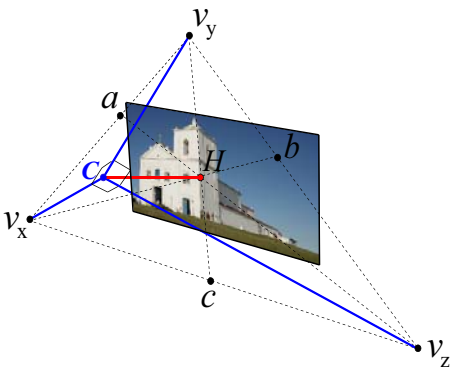
Assim, entre os pontos de fuga v_x, v_y, v_z na imagem tem-se:

$$\mathrm{v}_x \omega \mathrm{v}_y = 0$$
$$\mathrm{v}_x \omega \mathrm{v}_z = 0$$
$$\mathrm{v}_y \omega \mathrm{v}_z = 0$$

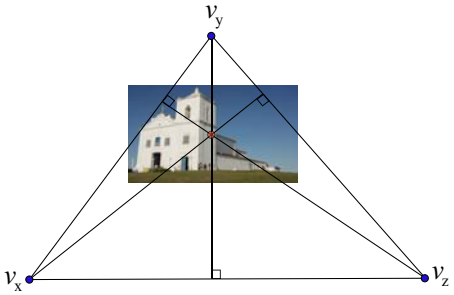
Logo, os raios que se estendem do centro óptico C até cada um dos pontos de fuga (representados como v_x, v_y, v_z) respeitam essa condição de perpendicularidade dois a dois. Este fato, somado ao fato de que os três pontos de fuga formam um triângulo cujo ortocentro é a projeção do centro óptico da câmera, permite calcular todos os parâmetros da equação 3-9. A Figura 3.9(a) mostra essa configuração geométrica. Esse fato é usado também para interpretar geometricamente e calcular a distância focal f.

Para tanto, é preciso calcular a projeção do centro óptico $\mathbf{C}$ na imagem. Essa coordenada quase sempre coincide com o centro da imagem e corresponde ao ortocentro (H) do triângulo formado pelos três pontos de fuga. A projeção do centro óptico, também conhecida como ponto principal, só não coincide com o centro do plano focal em casos onde a imagem foi recortada ou onde

PUC-Rio - Certificação Digital Nº 0611927/CA



3.9(a): O centro H da imagem em geral coincide com a projeção do centro óptico C da câmera. Os três pontos de fuga correspondentes a três direções ortogonais no mundo têm raios perpendiculares entre si dois-a-dois. O segmento vermelho corresponde a distância focal f.



3.9(b): O centro H da imagem equivale ao ortocentro do triângulo formado pelos três pontos de fuga.

Figura 3.9: Interpretação geométrica da distância focal e do centro da imagem, que coincide com o ortocentro do triângulo formado pelos pontos de fuga.

há desalinhamento do eixo óptico da câmera. No entanto, esses casos serão ignorados aqui. A Figura 3.9(a) ilustra o centro óptico da câmera e sua projeção no ortocentro H.

Para calcular o ortocentro do triângulo v_x, v_y, v_z, pode-se utilizar suas coordenadas baricêntricas $(r_x : r_y : r_z) = (\cot\alpha_z \cot\alpha_y : \cot\alpha_z \cot\alpha_x : \cot\alpha_x \cot\alpha_y)$ como mostra a equação a seguir.

$$\begin{aligned} H &= r_x \mathrm{v}_x + r_y \mathrm{v}_y + r_z \mathrm{v}_z = \\ &= (\cot\alpha_z \cot\alpha_y)\mathrm{v}_x + (\cot\alpha_z \cot\alpha_x)\mathrm{v}_y + (\cot\alpha_x \cot\alpha_y)\mathrm{v}_z \end{aligned}$$

Em [Cipolla et al., 1999a] é provado que as coordenadas baricêntricas (r_x, r_y, r_z) equivalem respectivamente aos parâmetros (w_1^2, w_2^2, w_3^2) da equação 3-9. Assim, como já mencionado, é possível calcular todos os parâmetros dessa equação através de uma abordagem geométrica. Os detalhes matemáticos para calcular tais coordenadas baricêntricas são encontrados no Apêndice A.

Uma vez obtida a projeção H do centro óptico, torna-se simples calcular a distância focal f utilizando também propriedades geométricas. Através da Figura 3.9 pode-se observar que

$$\overline{C\mathrm{v}_x}^2 = \overline{H\mathrm{v}_x}^2 + f^2 \tag{3-11}$$

$$\overline{C\mathrm{v}_y}^2 = \overline{H\mathrm{v}_y}^2 + f^2 \tag{3-12}$$

PUC-Rio - Certificação Digital Nº 0611927/CA

Somando as equações 3-11, vem:

$$\overline{C\mathrm{v}_x}^2 + \overline{C\mathrm{v}_y}^2 = \overline{H\mathrm{v}_x}^2 + \overline{H\mathrm{v}_y}^2 + 2f^2 \tag{3-13}$$

Por 3-10, o triângulo $C\mathrm{v}_x\mathrm{v}_y$ é retângulo. Logo, tem-se:

$$\overline{\mathrm{v}_x\mathrm{v}_y}^2 = \overline{C\mathrm{v}_x}^2 + \overline{C\mathrm{v}_y}^2 \tag{3-14}$$

$$\overline{\mathrm{v}_x\mathrm{x}_y}^2 = \overline{H\mathrm{v}_x}^2 + \overline{H\mathrm{v}_y}^2 + 2f^2 \tag{3-15}$$

$$\therefore$$

$$f = \frac{\sqrt{\overline{\mathrm{v}_x\mathrm{v}_y}^2 - \overline{H\mathrm{v}_x}^2 - \overline{H\mathrm{v}_y}^2}}{2} \tag{3-16}$$

Cálculo da orientação R

Embora a rotação $\mathtt{R}$ obtida pelo processo de calibração por pontos de fuga não seja diretamente utilizada nesta dissertação, esta matriz é calculada a seguir. Trata-se um processo simples que resulta em uma rotação que pode opcionalmente ser usada como parte da solução inicial ($\mathtt{R}_0$) para o algoritmo de minimização que é apresentado na Seção 3.7.1.

Da equação 3-7 conclui-se que:

$$\mathtt{R} = \mathtt{K}^{-1} \begin{bmatrix} w_1x_1 & w_2x_2 & w_3x_3 \\ w_1y_1 & w_2y_2 & w_3y_3 \\ w_1 & w_2 & w_3 \end{bmatrix} \tag{3-17}$$

substituindo a matriz $\mathtt{K}$ já calculada, obtemos a matriz $\mathtt{R}$.

Como apresentado por Zhengyou Zhang em [Zhang, 1998], é possível partir de $\mathtt{R}$ para chegar à resolução de $\mathtt{K}$ desenvolvendo algebricamente a equação 3-17 em função dos parâmetros de $\mathtt{K}$ e usando o fato de que $\mathtt{R}$ é ortonormal: os vetores das bases são unitários e são perpendiculares entre si, o que provê 6 restrições algébricas, suficientes para recuperar os parâmetros de $\mathtt{K}$ e, conseqüentemente, a rotação $\mathtt{R}$.

PUC-Rio - Certificação Digital Nº 0611927/CA

3.1.5 Adicionando informações métricas

Esta seção mostra uma maneira simples de recuperar a translação da cena a partir de uma rotação `R` — recuperada, por exemplo, através do processo de calibração por pontos de fuga recém-mostrado — e de mais dois pontos de relação entre imagem e modelo. Dessa forma pode-se recuperar todos os parâmetros da câmera com pouco conhecimento da geometria de cena. Esse método foi desenvolvido por esta pesquisa e o autor não tem conhecimento dela na literatura.

O método de calibração por pontos de fuga mostrado fornece apenas a matriz de orientação `R` e a de calibração de câmera `K`. Pela decomposição `P` = [`K`][`R`][`T`], observa-se que a `K` projeta pontos que estão escritos no sistema de coordenadas da câmera. A transformação Euclidiana [`R`][`T`] é responsável por mapear pontos do mundo para o sistema da câmera. Como não se conhece `T`, não é possível estimar a posição da câmera em relação ao mundo e, sem informações métricas adicionais, não há como estimar essa matriz.

Contudo, se certas informações métricas estão disponíveis, pode-se calcular a posição relativa da câmera em relação a origem do sistema de coordenadas no mundo. Com dois pares de associações $[\mathbb{P}^3, \mathbb{P}^2]$ entre coordenadas conhecidas do mundo e suas respectivas imagens, podemos encontrar a posição relativa da câmera. Seja $\mathbf{X}_1 = (\mathrm{X}_1, \mathrm{Y}_1, \mathrm{Z}_1, 1), \mathbf{X}_2 = (\mathrm{X}_2, \mathrm{Y}_2, \mathrm{Z}_2, 1) \in \mathbb{P}^3$ e suas respectivas coordenadas na imagem $\mathbf{x}_1 = (x_1, y_1, 1), \mathbf{x}_2 = (x_2, y_2, 1) \in \mathbb{P}^2$:

$$\begin{bmatrix} x_1 \\ y_1 \\ w_1 \end{bmatrix} = [\mathtt{K}][\mathtt{R}] \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} \tag{3-18}$$

$$\begin{bmatrix} x_2 \\ y_2 \\ w_2 \end{bmatrix} = [\mathtt{K}][\mathtt{R}] \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} \tag{3-19}$$

onde os parâmetros desconhecidos são w_1, w_2, t_x, t_y, t_z. Temos então 6 equações e 5 incógnitas que podem ser calculadas utilizando-se o método de resolução de sistemas lineares por decomposição de valores singulares (SVD). Os parâmetros w_1 e w_2 são intrínsecos do mapeamento projetivo. Note que embora o objetivo seja recuperar os parâmetros t_x, t_y, t_z, não se pode ignorar w_1 e w_2 (ex. assumir

PUC-Rio - Certificação Digital Nº 0611927/CA

$w_1 = 1, w_2 = 1$) e utilizar apenas um único ponto do mundo para o cálculo de `T` pois não há um mapeamento um-para-um de pontos do mundo e pontos $(x, y, 1)$ na imagem, já que todos os pontos $\lambda(\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 1)$ são transformados para o mesmo ponto no sistema de coordenadas da imagem.

PUC-Rio - Certificação Digital Nº 0611927/CA

3.2 Extração das Arestas Estruturais do Modelo

O processamento da geometria acontece como uma seqüência de etapas cujo passo inicial é a leitura do modelo, que é composto por uma lista de triângulos, vértices e índices que associam os triângulos aos vértices. A ordem em que estas operações ocorrem influencia diretamente o resultado final e, por isso, elas devem ser executadas na mesma ordem em que são apresentadas aqui.

Embora esta etapa de processamento seja complexa e consuma um tempo de processamento que ultrapassa o limite de interatividade com o usuário, ela pode ser feita como um pré-processamento que é armazenado para uso futuro, ou seja, pode ser feita uma única vez para cada arquivo de modelo, enquanto tal arquivo não sofrer alterações.

As operações necessárias para a extração das arestas estruturais do modelo a partir dessa lista são descritas abaixo:

Construção de Informações Topológicas:

Nesta etapa uma estrutura topológica simples é utilizada para prover informações sobre vértices adjacentes a outros vértices e triângulos adjacentes à arestas. Essas informações serão utilizadas para as etapas seguintes, quando arestas que residem entre triângulos coplanares vizinhos são eliminadas. A estrutura topológica utilizada deve ser capaz de fornecer as seguintes perguntas a respeito da topologia da malha geométrica:

- A partir de um vértice qualquer, quais são os índices dos vértices que lhe são adjacentes?;
- Dada uma face j composta pelos vértices de índices (v_1, v_2, v_3), qual o índice da face adjacente a aresta oposta a qualquer um desses vértices?

Para realizar tal tarefa foi utilizada a biblioteca Trimesh2[2] que provê todas as estruturas de dados e funcionalidades necessárias para que tais informações sejam obtidas.

Criação de Lista Unívoca de Arestas:

Em uma malha de triângulos, cada aresta pode ser interpretada duas vezes como sendo arestas diferentes. Isso ocorre porque dois vértices da mesma aresta podem ser percorridos em sentidos contrários, a partir de triângulos adjacentes. Isso pode representar um problema porque freqüentemente existe uma quantidade grande de arestas a serem processadas e, além disso, tal fato

[2] O site oficial da biblioteca Trimesh2 é http://www.cs.princeton.edu/gfx/proj/trimesh2/.

PUC-Rio - Certificação Digital Nº 0611927/CA

pode fazer com que a lista de arestas resultante do processo de simplificação tenha elementos repetidos. Como o objetivo é obter uma lista de arestas do modelo que possam ser associadas a arestas da imagem, é preciso criar uma lista unívoca para que não haja processamento repetido ou dupla associação entre arestas do modelo e da imagem (o que resulta em um processamento extra durante o processo de minimização para ajuste da posição de câmera).

Para resolver tal problema existem algumas soluções simples:

- Pode-se ordenar todas as arestas por identificador de vértice. Isso é feito armazenando cada aresta em uma lista por ordem crescente ou decrescente de seus vértices (por exemplo, cada aresta (i, j) é sempre inserida como (i, j) na lista se $i < j$ ou como (j, i) caso contrário). Esta estratégia é simples, mas requer uma passada por todas as arestas para inserção e outra para ordenação.
- Outra estratégia é utilizar um algoritmo de *Hashmap* para armazenar as arestas de forma única usando os índices dos vértices das arestas como identificadores. Para tal, é preciso gerar uma chave única para cada aresta a partir desses índices, o que é simples de ser feito. É esta a estratégia utilizada pelo método implementado nesta dissertação para gerar uma lista única de arestas. Como, em geral, o número de arestas em um modelo é alto (diretamente proporcional ao numero de triângulos), o custo-benefício de utilizar tal estrutura pode justificar seu uso, já que com um único acesso ao *hash* pode-se acessar ou armazenar uma aresta de forma unívoca.

PUC-Rio - Certificação Digital Nº 0611927/CA

Descarte de Arestas Coplanares:

Com a lista de arestas e a estrutura topológica em mãos, torna-se simples percorrer cada uma das aresta e verificar se ela é adjacente a dois triângulos coplanares. Para isso, basta verificar se as normais dos respectivos triângulos adjacentes são paralelas, utilizando uma tolerância mínima pré-estabelecida.

Calculando as normais n_1 e n_2 de dois triângulos adjacentes por uma aresta a, o critério de remoção de tal aresta é

$$(1 - abs(n_1.n_2)) <= t_{coplanar}$$

onde $t_{coplanar}$ é qualquer tolerância pequena o suficiente para o critério de normais paralelas.

Agregação de Sequências de Arestas Colineares:

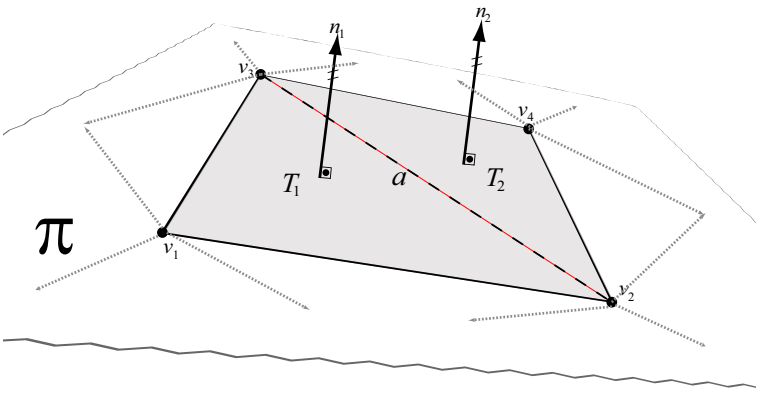


Figura 3.10: Descarte de arestas contidas em planos. Cada aresta que for adjacente a dois triângulos coplanares (aresta a pontilhada) é descartada.

Esta etapa tem por objetivo unir segmentos colineares consecutivos. Ela é necessária porque o trabalho realizado nesta dissertação optou por descartar segmentos proporcionalmente curtos em relação ao tamanho total do modelo. Tais segmentos em geral representam partes que têm pouca probabilidade de serem relacionadas com elementos da imagem e freqüentemente apenas dificultam o processo de ajuste de câmera.

Contudo, o trabalho aqui realizado não assume nenhum tipo de critério em relação a natureza da modelagem geométrica da malha do modelo. Por este motivo é preciso considerar que a malha pode ser composta por um conjunto muito grande de segmentos curtos com uma minoria de segmentos longos, o que, segundo o critério de descarte adotado, pode ocasionar na eliminação quase completa da malha.

Para resolver tal problema e ser capaz de tratar quaisquer tipos de malhas geométricas, desenvolveu-se um processo de agregação de segmentos colineares curtos. Tal processo faz com que diversos segmentos consecutivos em uma mesma borda sejam unidos e transformados em um único segmento maior.

O procedimento é feito utilizando-se a estrutura topológica, que pode prover vizinhança de vértices, e a lista unívoca de arestas (L_u) construída previamente. *Para evitar inconsistências na estrutura topológica, ela nunca é alterada, apenas a outra estrutura, a lista L_u, é modificada.*

O algoritmo funciona da seguinte forma: para cada aresta na lista L_u, obtêm-se dois vértices (v_1, v_2) dentre os quais um deles é selecionado (v_s). Verifica-se então, para cada vértice v_i adjacente a v_s, se há uma aresta (v_s, v_i) colinear a (v_1, v_2) e se esta aresta existe na lista L_u (como a estrutura topológica não é alterada, ela pode conter arestas que já foram removidas da lista). Em caso positivo, os vértices (v_1, v_2, v_i) são unidos e transformados em uma única aresta, a lista de aresta é atualizada e o procedimento continua para o vértice

PUC-Rio - Certificação Digital Nº 0611927/CA

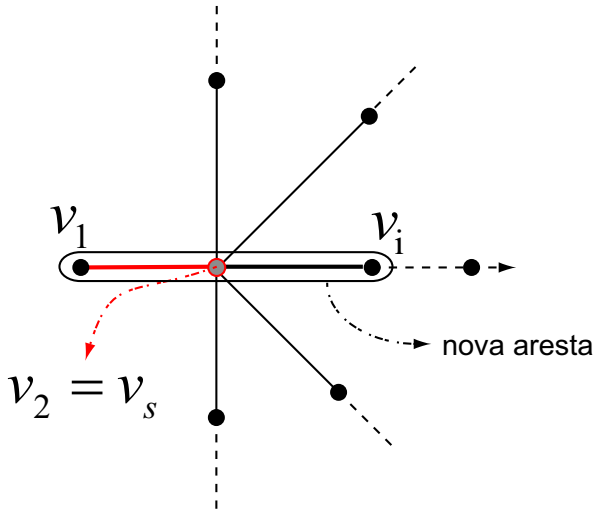


Figura 3.11: Processo de agregação de arestas colineares consecutivas.

PUC-Rio - Certificação Digital Nº 0611927/CA

v_i — o vértice v_s é descartado da respectiva aresta, restando apenas os vértices da extremidade, que passam a constituir uma nova aresta maior. Já em caso negativo, uma outra aresta é selecionada da lista e o processo se repete. O critério de colinearidade entre arestas constitui-se, por sua vez, de uma simples verificação de paralelismo utilizando-se uma tolerância arbitrária. A Figura 3.11 ilustra o processo de agregação.

Após o término desse processo, não existirão arestas colineares consecutivas, de tal modo que os segmentos proporcionalmente curtos representaram de fato arestas menos significativas do modelo e a maior aresta pode então ser utilizada como parâmetro para a eliminação dessas arestas pequenas.

Normalização da Malha:

Após a etapa de agregação é possível efetuar o processo de descarte de arestas curtas para restringir ainda mais o conjunto de arestas que serão posteriormente utilizadas no processo de associação modelo-imagem. Isto justifica-se porque estamos interessados em extrair a estrutura fundamental do modelo, ou seja, a estrutura que melhor represente a forma básica do modelo. Arestas muito curtas podem causar um aumento da complexidade do algoritmo de ajuste de câmera, já que ele é proporcional ao número de arestas usadas. A Figura 3.12 ilustra uma situação em que milhares de arestas curtas pertencem à malha de uma edificação e não têm utilidade alguma para o processo de ajuste. Caso tais arestas fossem consideradas, tornariam o tempo de execução do algoritmo muito mais alto e não trariam benefício, pois dificilmente seriam associadas a algum segmento detectado na imagem.

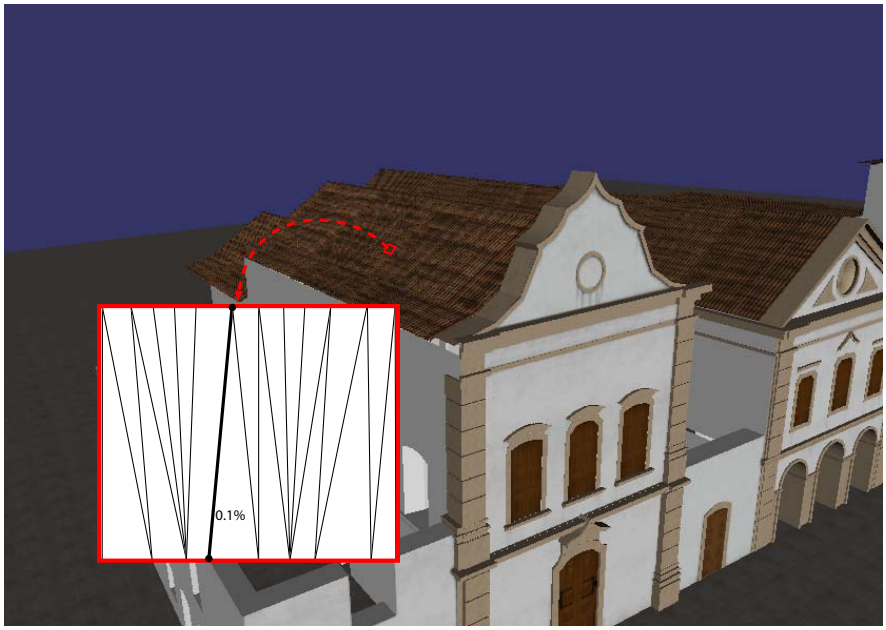


Figura 3.12: Arestas curtas que pertencem à malha geométrica de um modelo feito no 3D Studio Max® (arestas que compõem o telhado). Tal modelo contém milhares de arestas pequenas que chegam a 0.1% do tamanho da maior aresta e podem ser eliminadas durante o processo de extração.

No entanto, não é possível pré-estabelecer um tamanho mínimo para descarte de aresta, uma vez que a escala de cada modelo pode variar. Por esse motivo, as arestas são normalizadas para que tal limite seja baseado no tamanho da maior aresta do modelo, que após tal processo tem comprimento unitário. Essa estratégia de normalização elimina então as arestas que são proporcionalmente pequenas em relação ao tamanho do modelo total.

Descarte de Arestas Curtas:

Com o *threshold* para arestas curtas em mãos (que varia de 0 a 1) e o modelo normalizado, ocorre um processo automático de descarte das arestas que são consideradas proporcionalmente pequenas.

Os testes do processo de simplificação em um modelo com mais de 1.000.000 de arestas resultou em uma lista com pouco mais de 3.000 arestas (0.3%), das quais a maioria será descartada durante o teste de oclusão mostrado adiante. De fato, neste modelo pouco mais de 90 arestas visíveis foram selecionadas, em média, para a etapa de seleção das relações entre modelo e imagem. A Figura 3.13 ilustra o resultado.

As arestas resultantes deste passo compõem então o conjunto que será utilizado para o processo de localização de correspondências entre imagem e modelo. Elas são chamadas nesta dissertação de *arestas estruturais do modelo*.

PUC-Rio - Certificação Digital Nº 0611927/CA

3.13(a): Parte da malha original de um modelo exibida em *wireframe*. Esta malha contém mais de 1.000.000 de arestas no total.

3.13(b): Algumas das arestas visíveis selecionadas marcadas em amarelo por cima do modelo, utilizando-se o mesmo ponto de vista da imagem ao lado. Milhares de arestas foram descartadas.

Figura 3.13: Arestas selecionadas para uso (em amarelo) na etapa de detecção de relações entre modelo e imagem.

PUC-Rio - Certificação Digital Nº 0611927/CA

3.3 Posicionamento Inicial de Câmera

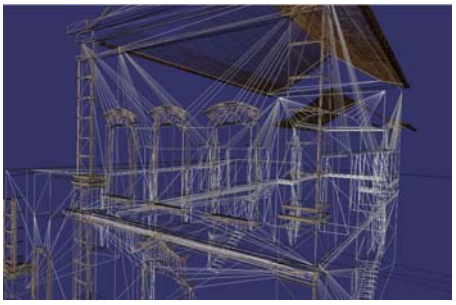
O posicionamento inicial da câmera é um processo manual cuja execução depende do usuário. Nesta etapa ele deve movimentar a câmera virtual de modo a aproximar o ajuste entre modelo e foto. Esta etapa é necessária por três razões: 1) porque o processo de minimização desenvolvido (Seção 3.7.1) precisa de valores iniciais apropriados para que os parâmetros externos de câmera possam convergir para o valor ótimo; 2) porque as arestas do modelo precisam ser projetadas próximas às arestas da imagem para que as correspondências sejam corretamente calculadas; 3) porque o teste de oclusão necessita de uma posição inicial relativa entre o modelo e a câmera para que as arestas estruturais oclusas sejam descartadas.

É a partir desta posição inicial de câmera que o algoritmo de ajuste automático recupera a rotação e translação ótimas da câmera, projetando a geometria do modelo na foto a partir dessa posição e buscando correspondências entre as arestas projetadas visíveis (que passaram no teste de oclusão) e segmentos detectados na foto. Esses processos serão mostrados em seções posteriores.

Contudo, para que seja possível projetar a geometria do modelo na foto usando a posição de câmera inicial, é preciso antes de tudo processar a malha desse modelo para extrair apenas as arestas tridimensionais que sejam candidatas a terem associação direta com um segmento na imagem.

Frequentemente, malhas geométricas geradas em aplicações de modelagem como 3D Studio Max® exibem muitos segmentos tridimensionais que não têm chance de corresponder à segmentos da imagem de entrada, seja porque

não são visíveis (estão compondo um quadrilátero e são coplanares) ou porque são muito curtos e podem ser desprezados. As figuras 3.14(a) e 3.14(b) ilustram o primeiro caso.



3.14(a): Modelo gerado no 3D-Studio Max® exibido em modo *wireframe*. Pode-se observar que paredes e forros são compostos por triângulos adjacentes que têm muitas arestas coplanares compondo uma mesma face.

3.14(b): Imagem do mesmo modelo da figura acima, desenhada em modo normal. As arestas que compõem uma mesma face são invisíveis quando o modelo é redenrizado em modo normal.

Figura 3.14: Modelo criado por ferramenta de modelagem comercial exibido de duas formas: em modo *wireframe*, onde as arestas dos triângulos que o compõem aparecem; e em modo normal, onde somente um subconjunto dessas arestas aparecem.

O processo de simplificação da geometria tem, então, o objetivo de extrair apenas as arestas estruturais da malha do modelo, como arestas de borda que têm um tamanho suficientemente grande (proporcionalmente ao modelo). Tal processo de eliminação de segmentos inválidos foi desenvolvido utilizando-se alguns métodos simples que são expostos a seguir.

3.4 Teste de Oclusão

O teste de oclusão descarta segmentos estruturais que não são visíveis quando superpostos em relação ao modelo em sua forma original. A idéia é que apenas os segmentos estruturais que têm chance de serem associados a segmentos da imagem sejam selecionados para uso.

Após a etapa inicial de pré-processamento do modelo, diversos segmentos são obtidos, mas quando o usuário fornece um posicionamento inicial de câmera, muitos deles ficam oclusos quando sobrepostos com o modelo original, como os segmentos que estão atrás do modelo em relação ao ponto de vista corrente. Como a imagem é exatamente a projeção de partes visíveis do modelo real em relação ao ponto de vista da câmera, os segmentos que não são visíveis não podem ser associados a segmentos (visíveis) da imagem.

PUC-Rio - Certificação Digital Nº 0611927/CA

A posição de câmera inicial informada pelo usuário está próxima à posição final que gerou a foto, mas não é a posição exata, o que pode fazer com que segmentos que não são visíveis — do ponto de vista que gerou a foto — não sejam descartados nesta etapa e vice-versa. Contudo, é provável que a maioria dos segmentos sejam corretamente descartados ou selecionados nesta fase, reduzindo o número total de segmentos que devem ser tratados na etapa de associação entre modelo e imagem.

O algoritmo de descarte de segmentos oclusos utiliza a funcionalidade de *Occlusion Query* do OpenGL. O procedimento é simples e é descrito abaixo:

1. renderização do modelo em sua forma original;
2. renderização de cada uma das arestas estruturais seguido de uma consulta de *Occlusion Query*
3. verificação da consulta de oclusão: se a aresta renderizada gerou pixels, ela é visível e deve ser selecionada. Caso contrário ela é descartada.

Esse procedimento irá selecionar as arestas 3D visíveis do modelo que devem ser projetadas na tela para servirem de base para o processo de associação imagem-modelo. A etapa de projeção dessas arestas é descrita a seguir.

3.5 Projeção da Geometria

Para que as arestas selecionadas nas etapas de pré-processamento da geometria do modelo e de teste de oclusão sejam associadas a arestas da imagem, é preciso projetá-las no plano focal da câmera. Tanto a imagem quanto os segmentos selecionados, são transformados para o espaço do plano de projeção da câmera. Esse processo é possível graças à calibração de câmera feita anteriormente, que recupera seu ângulo de abertura e sua distância focal.

Para projetar os segmentos extraídos do modelo, uma matriz de projeção perspectiva é criada utilizando-se os parâmetros intrínsecos da câmera calibrada. O Capítulo 3.1 mostra como pontos do mundo são projetados no plano de projeção da câmera.

O produto final desta etapa é então um conjunto de segmentos bidimensionais que devem ser associados a segmentos detectados na imagem. Para fazer tal associação é utilizado um processo de busca de segmentos da imagem nas vizinhanças da própria aresta a qual se está tentando associar. Esse processo é explicado na próxima seção.

PUC-Rio - Certificação Digital Nº 0611927/CA

3.6 Cálculo de Correspondências Modelo-Imagem

Para ajustar a posição do objeto a partir da foto é necessário medir o erro entre certas características ou *features* do modelo e da imagem (Seção 3.7.1). Assim precisamos saber fundamentalmente quais elementos da imagem correspondem a elementos do modelo, o que será chamado aqui de *problema de correspondência*[3]. Este é um dos principais problemas da visão computacional [Fischer and Bolles, 1981] e não tem solução trivial. Existe uma grande quantidade de trabalhos que se propõem a resolvê-lo e, em geral, podem ser classificados em automáticos ou semi-automáticos (que dependem da interação do usuário). Esta pesquisa propõe um método para resolver este problema de forma semi-automática, em duas etapas relacionadas: detecção de *features* na imagem e a interpretação das mesmas em relação ao modelo.

O método aqui utilizado adota o segmento de reta como *feature* da imagem em detrimento do ponto (quina). Embora não tenham sido analisadas justificativas[4] matemáticas para esta escolha, existem alguns fatores que fazem com que este tipo de *feature* seja preferido: i) esta dissertação trata exclusivamente de fotografias de edificações, que na maioria dos casos possuem uma geometria formada por objetos que exibem muitas arestas; ii) segmentos de reta podem fornecer uma quantidade maior de informação para aumentar a qualidade do processo de detecção, pois existem mecanismos de controle que permitem que esses segmentos sejam descartados sob certas condições, tais como distância mínima entre pontos colineares e limite mínimo de tamanho; iii) esta pesquisa utiliza fotos de edificações em ambientes externos, que podem conter, por exemplo, vegetação ou outros objetos complexos. Nestes tipos de ambiente, o segmento de reta se mostrou também mais eficaz. A técnica utilizada para detectar arestas é mostrada a seguir nesta seção.

Para encontrar correspondências entre modelo e imagem, esta pesquisa propõe uma técnica que utiliza o próprio modelo para auxiliar no processo de detecção de arestas. Para facilitar o entendimento do método, os dois passos citados — detecção e interpretação — serão explicados separadamente.

[3]O nome deste problema pode ser utilizado para remeter ao problema da correspondência entre imagens, mas neste trabalho será utilizado para designar o problema da associação entre imagem e a cena.

[4]Uma justificativa matemática poderia ser elaborada através de testes de precisão da minimização (ajuste de câmera) em função de um fator de ruído aplicado às marcações, utilizando cada um dos tipos de features — quinas e arestas.

PUC-Rio - Certificação Digital Nº 0611927/CA

3.6.1 Detecção de Arestas na Imagem

Para localizar segmentos de reta em imagens reais, é preciso antes pré-processá-las utilizando filtros apropriados para eliminar ruído ou realçar arestas e bordas. Existem muitos trabalhos que propõem diferentes tipos de filtros para efetuar tais processamentos, porém, um dos mais conceituados é o filtro Canny de realce de arestas, desenvolvido por J.Canny em [Canny, 1986] e utilizado nesta pesquisa. As imagens pré-processadas e realçadas são então utilizadas por algoritmos especializados em detecção de arestas que vão fornecer as coordenadas dos segmentos de reta ou as equações de reta que representam matematicamente um certo conjunto de pixels na imagem. Dois destes algoritmos são a *Transformada de Hough* (HT) e a *Transformada de Hough Probabilística* (PPHT).

A Transformada de Hough é um método popular de extração de primitivas geométricas a partir de imagens. A versão mais simples possível desse algoritmo é feito para detectar apenas retas, mas o método pode ser facilmente generalizado para detectar outros tipos de objetos geométricos como círculos ou quadrados. Existem diversas classes da HT que diferem de acordo com certos tipos específicos de informação disponíveis na imagem de entrada. Para uma imagem qualquer a Transformada de Hough Padrão (SHT - *Standard Hough Transform*) [Trucco and Verri, 1998] pode ser usada.

Em [J.Matas et al., 2004], é proposto um método avançado para detectar múltiplas primitivas geométricas chamado Transformada de Hough Probabilísticas (PPHT - *Progressive Probabilistic Hough Transform*). Esse método, que também é utilizado nesta pesquisa através da biblioteca OpenCV, fornece segmentos de reta na imagem ao invés de equações de retas como na HT padrão e é também consideravelmente mais eficiente.

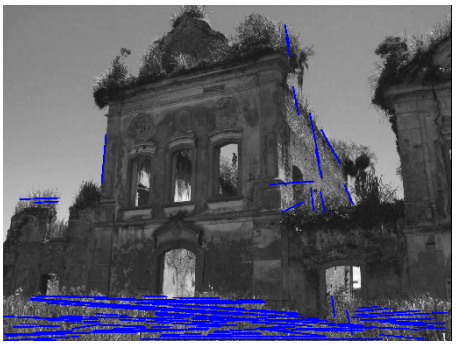
Ambos os métodos foram testados por esta pesquisa, a título de comparação. Cada um apresenta capacidades diferentes dependendo da qualidade da imagem de entrada e, por isso, não é possível afirmar definitivamente qual o melhor método para detecção de arestas. O PPHT é mais eficiente em imagens que possuem bastante ruído, como mencionado, mas é também menos preciso. A escolha de uso por algum dos dois métodos depende diretamente da natureza da imagem de entrada, porém o método PPHT teve resultados melhores quando usado junto à um conjunto de imagens de ambientes externos — o número de segmentos de qualidade detectados foi maior do que o teste feito com o HT e o tempo de execução foi consideravelmente menor em imagens de alta resolução.

PUC-Rio - Certificação Digital Nº 0611927/CA

3.6.2 Interpretação dos Segmentos Detectados

O modelo real em sua posição inicial possivelmente possui um conjunto de características que pode ser rastreado na imagem e a imagem contém um sub-conjunto das arestas visíveis do modelo. Esse fato é a base para o algoritmo de detecção de correspondências entre características aqui proposto.

A presença de ruído afeta a detecção de arestas na imagem. Um problema comum é encontrar, em regiões com muito ruído, um número grande arestas que não correspondem a partes do modelo real. A enorme quantidade de arestas detectadas nessas regiões torna difícil a seleção das que correspondem realmente a segmentos do modelo. Isso faz com que seja preciso detectar um número maior de arestas antes que seja possível ter arestas válidas, o que ainda assim não garante que isso ocorra. A Figura 3.15 ilustra o problema em uma imagem que exibe uma região com vegetação densa e gramado.



3.15(a): Imagem em tons de cinza que contém vegetação densa e gramado. Mais de 400 arestas foram detectadas (em azul), mas a maioria foi detectada em áreas de intenso ruído e apenas poucas arestas válidas foram obtidas.

3.15(b): Imagem com filtro Canny de realce de arestas.

3.15(c): Imagem original.

Figura 3.15:

No entanto, esta dissertação propõe um método que consiste em utilizar

PUC-Rio - Certificação Digital Nº 0611927/CA

o próprio modelo para restringir a área de busca por segmentos na imagem. O processo consiste em criar uma janela em volta da projeção de cada aresta visível do modelo — já posicionado pelo usuário —, que delimita uma Área de Interesse (*Region of Interest* - ROI). Cada ROI funciona como uma sub-imagem através da qual pode-se restringir a área de detecção. A aresta procurada deve estar contida nesta região de interesse e ter inclinação apropriada, tamanho mínimo e extremidades próximas às da aresta do modelo que foi projetada, seguindo a premissa de que esse modelo está aproximadamente ajustado com a foto.

As arestas do modelo que são projetadas de forma paralela aos eixos x ou y da imagem — arestas horizontais ou verticais — têm ROIs com área desejada para busca de arestas. Contudo, arestas inclinadas terão regiões de interesse maiores, como mostra a Figura 3.16. Para resolver tal problema, são calculadas duas retas dentro de cada ROI, que delimita uma sub-região efetiva válida dentro dessa região. Em arestas horizontais ou verticais, todo o ROI constitui a área efetiva de interesse. Já em outros casos, é necessário calcular duas retas delimitadoras de forma simples:

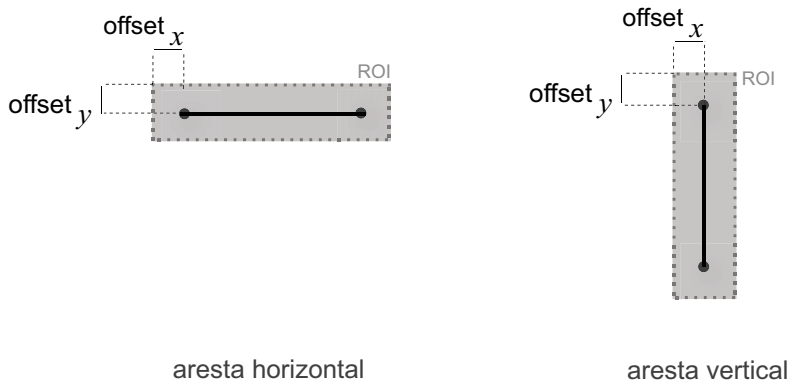
1. usando os dois pontos extremos da aresta projetada, uma equação de reta inicial é calculada;

2. tal reta dá então origem a duas outras retas, através da soma de um valor de *offset* positivo e negativo ($\pm d$).

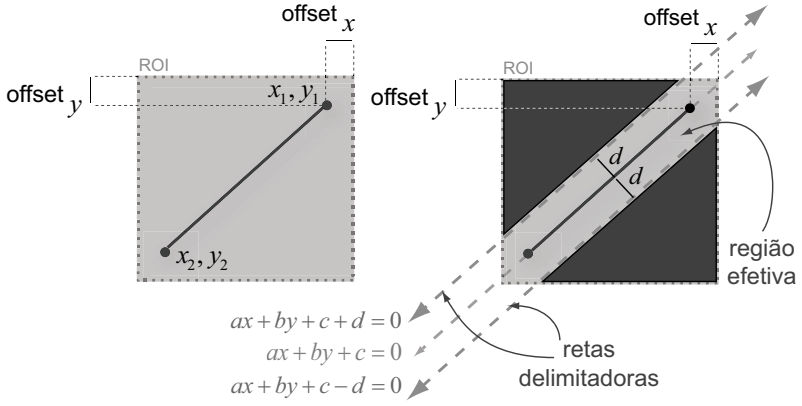
Com essas duas retas é simples verificar se uma aresta detectada está fora da região de interesse efetiva ou não — basta verificar se seus vértices estão acima da reta inferior e abaixo da reta superior. A distância de *offset* utilizada é o maior dos *offsets* em x ou em y de um ROI — que, em geral, são iguais (Figura 3.17(a)). Esses valores de *offset* poderiam ser qualquer outro valor suficientemente grande. De fato, o valor apropriado para o tamanho de todos os *offsets* de um ROI pode variar de acordo com a imagem e com o modelo. Todas as arestas detectadas que estão fora da região de interesse efetiva são descartadas.

O principal motivo para utilizar tal cálculo para delimitar uma região efetiva de interesse dentro de um ROI é simplificar o processo de detecção de arestas dentro desta região. Caso cada retângulo que delimita um ROI fosse criado genericamente de forma não alinhada com os eixos da imagem, seria preciso copiar os pixels de tal região para uma nova imagem ou calcular uma transformação entre dois sistemas de coordenadas a cada acesso de um pixel dentro da região, o que aumentaria a complexidade do processo de detecção de arestas.

PUC-Rio - Certificação Digital Nº 0611927/CA



3.16(a): Exemplos de ROI em arestas verticais e horizontais. Nestes casos as retas delimitadoras estão localizadas nas bordas do ROI, de modo que este coincide com a área de interesse efetiva.



3.16(b): Exemplo de ROI em arestas inclinadas. A aresta da esquerda possui um ROI sem delimitação de área efetiva. O ROI da direita possui delimitação por retas calculadas a partir dos extremos do segmento projetado $(x_1, y_1), (x_2, y_2)$.

Figura 3.16: Cálculo da região efetiva de interesse dentro de um ROI.

Para classificar as arestas dentro da região efetiva de um ROI desenvolveu-se nesta pesquisa uma função que mede o quão bem um segmento da imagem se ajusta a um segmento projetado do modelo. Tal função foi criada a partir de uma adaptação da equação proposta por Taylor et al em [Taylor and Kriegman, 1995], que mede o erro entre dois segmentos no espaço da imagem da seguinte forma:

$$\int_0^l h^2(s)ds = \frac{l}{3}(h_1^2 + h_1h_2 + h_2^2) \tag{3-20}$$

onde $s \in [0, l]$, l representa o tamanho do segmento medido, escrito como $l = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ e $h(s)$ é a função que retorna a menor distância entre um ponto $P(s)$ e a linha projetada (Figura 3.17(a)).

A função de classificação de arestas em um ROI foi então criada através

PUC-Rio - Certificação Digital Nº 0611927/CA

de uma adaptação dessa função de erro utilizando-se seu inverso — já que o objetivo é obter uma classificação tão boa quando menor o erro medido — e aplicando-se uma função logarítmica para suavizar o crescimento da curva de classificação. Assim, a função de classificação proposta é escrita como:

$$rank(d_1, d_2, l) = \log\left(\frac{l^2/3}{\int_0^l h^2(s)ds}\right) = \log\left(\frac{l}{(h_1^2 + h_1 h_2 + h_2^2)}\right) \qquad (3\text{-}21)$$

onde *rank* é a função que classifica o quão bem um determinado segmento da imagem se adapta a um segmento do modelo e h_1, h_2 são as distâncias entre as extremidades da aresta detectada e a reta formada pela aresta do modelo projetada.

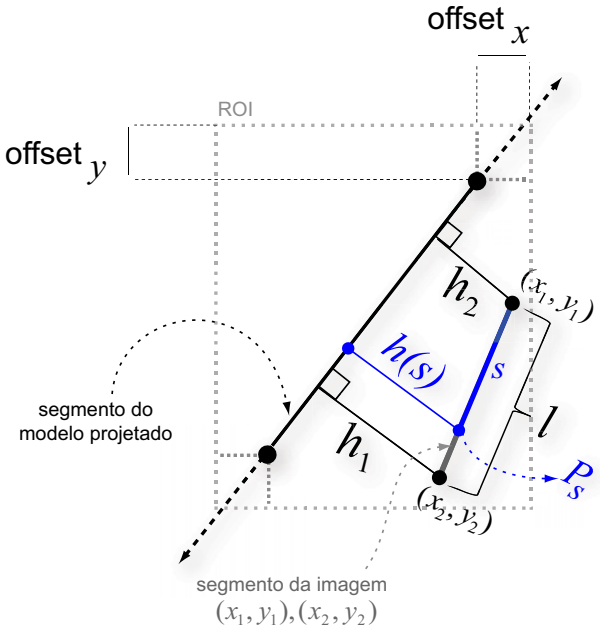
Cada aresta em um ROI recebe uma classificação segundo o critério mostrado e a aresta com o maior valor de classificação é escolhida como sendo a correspondência correta da aresta do modelo. Por este cálculo observa-se que arestas maiores têm prioridade sobre arestas menores, mas arestas menores que estão mais próximas e mais alinhadas têm prioridade sobre arestas maiores e distantes ou menos alinhadas.

Contudo, esta técnica exibe dois problemas:

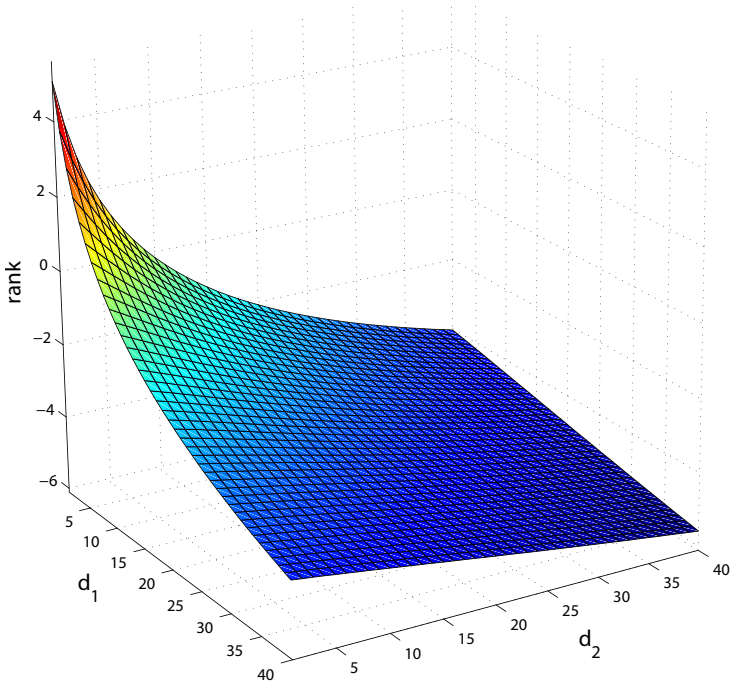
1. Ela não é capaz de detectar sempre bons segmentos para uma determinada aresta do modelo, porque o interpreta localmente: mesmo que um segmento da imagem não represente corretamente um segmento do modelo ele pode ser selecionado para a associação porque o critério de seleção mostrado, que utiliza a classificação mostrada em 3-21, não utiliza informações da topologia do modelo, ou seja, não há um tratamento que associe globalmente o modelo com a imagem. Isso significa que um segmento que deveria ser selecionado pode ser descartado dentro de um ROI em detrimento de outro que não representa partes do modelo na imagem, simplesmente porque o modelo está em uma posição inicial ruim. Logo, *a eficiência desse método é tão boa quanto for o ajuste inicial fornecido pelo usuário, dentro de uma tolerância aceitável de erro que é arbitrária e deve ser definida visualmente.*

2. Podem correr interseções entre ROIs da imagem, causadas por arestas do modelo que são projetadas em posições próximas na imagem. Isso possibilita que duas arestas diferentes do modelo elejam o mesmo segmento da imagem como sendo sua correspondência. Quase sempre isso caracteriza uma inconsistência, pois duas arestas do modelo só podem representar a mesma aresta na imagem se forem projetadas de forma totalmente so-

PUC-Rio - Certificação Digital Nº 0611927/CA



3.17(a): Classificação dos segmentos detectados dentro de uma Região de Interesse (ROI). A aresta $(x_1, y_1), (x_2, y_2)$ foi detectada na imagem dentro da região de interesse mostrada, representada pelo retângulo.



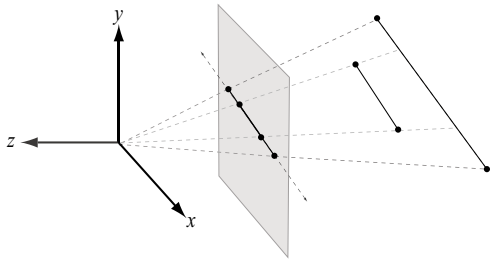
3.17(b): Gráfico da função de classificação *rank* para um segmento de tamanho $l = 100$ (pixels) em torno de uma janela de erro de 40×40 pixels. Pode-se observar que a classificação é tão boa quanto menor forem os valores de d_1 e d_2 (também em **pixels**). O valor de l apenas escala o gráfico mas não altera seu comportamento.

PUC-Rio - Certificação Digital Nº 0611927/CA

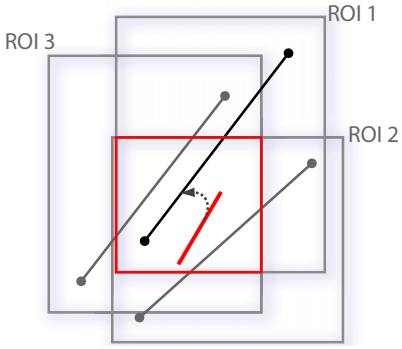
breposta. A abordagem deste trabalho adota o critério de não permitir tal configuração — já que os segmentos do modelo são tratados como retas 3D. Assim, uma técnica simples para tratamento de interseções entre ROIs é proposta a seguir.

3.6.3 Tratando interseções entre ROIs

A técnica mostrada para associação de arestas da imagem com o modelo permite que diversos ROIs contenham interseções (Figura 3.17(d)). Como mencionado, isso possibilita que várias arestas do modelo sejam associadas à mesma aresta na imagem. Embora esta situação seja consistente do ponto de vista geométrico, uma vez que muitas arestas 3D coplanares podem ser projetadas no mesmo segmento da imagem, ela não caracteriza uma situação válida aqui, porque, cada segmento do modelo é tratado como uma reta de extensão infinita. Logo, apenas um desses segmentos deve ser considerado visível. Assim, caso não se trate interseções entre ROIs, pode ocorrer de um segmento visível ser projetado juntamente com um outro segmento maior (Figura 3.17(c)). Pode ocorrer ainda de dois segmentos serem projetados muito próximos na imagem e, assim, ocorrer a interseção de seus respectivos ROIs. Este último caso consiste em uma situação inválida por que dois segmentos distintos não coplanares nunca podem representar a mesma aresta na imagem.



3.17(c): Projeção de dois segmentos do modelo que são coplanares e visíveis. Esta configuração possibilita que dois segmentos do modelo sejam associados a uma mesma aresta da imagem, o que não é desejado na abordagem proposta.



3.17(d): Interseção de ROIs. A aresta em vermelho pode ser associada a qualquer um dos ROIs mostrados. O tratamento de conflito irá associar ao melhor segmento do modelo, que no caso é o que está no ROI 1.

Figura 3.17: Tratamento de interseções entre diferentes ROIs.

Para tratar os casos de interseção entre ROIs é preciso ter algum tipo de critério de desempate entre dois ou mais segmentos do modelo que compartilham uma mesma aresta. O processo é simples: para cada ROI de

PUC-Rio - Certificação Digital Nº 0611927/CA

cada aresta visível do modelo, verifica-se se há algum outro ROI de outra aresta que o interseciona. Para todos os ROIs que têm interseção, verifica-se se há conflito de arestas, ou seja, se um dos segmentos dos ROIs conflitantes tem a mesma aresta da imagem como associação. Em caso positivo, há uma reavaliação da classe para ver qual das arestas do modelo se adapta melhor ao segmento da imagem. Por exemplo, na Figura 3.17(d), a aresta da imagem (em vermelho) é associada ao segmento do ROI 1 mostrado, pois se adapta melhor a ele de acordo com o critério de classificação usado. O teste de interseção entre ROIs é um simples teste de interseção entre retângulos.

3.7 Cálculo de Posição e Orientação da Câmera

Com um conjunto suficientemente grande de correspondências entre a imagem e a cena, obtido em etapas anteriores, o *problema da correspondência* está resolvido e pode-se então mensurar o erro entre a projeção de arestas do modelo da imagem. A partir de tais medições pode-se recuperar a posição e orientação de uma câmera calibrada minimizando uma função-objetivo que depende dos parâmetros externos da câmera e das correspondências obtidas. Essa função mede o erro total entre as arestas do modelo e da imagem em função dos parâmetros externos da câmera, que são recuperados a partir de um processo de minimização para encontrar os valores ótimos de tais parâmetros.

O método para recuperar a rotação e translação proposto nesta dissertação não é considerado um método de reconstrução completa da posição de câmera, tal como [Tsai, 1986], mas sim um método de ajuste que aceita posições de câmeras iniciais e, através de um processo de minimização, encontra a posição final utilizando as correspondências entre imagem e modelo. A função-objetivo que é minimizada para a obtenção da posição e da orientação finais da câmera é não linear, e por isso precisa respeitar certos critérios de convergência [Taylor and Kriegman, 1995]. A próxima seção explica como tal função-objetivo é construída.

3.7.1 Ajuste e Minimização

Para posicionar a câmera no mundo, é preciso uma transformação de corpo rígido, que é decomposta em uma rotação (`R`) e uma translação (`T`). A parametrização utilizada para representar tais transformações foi extraída do trabalho realizado por Taylor et al em [Taylor and Kriegman, 1994] e é exibida em maiores detalhes no Apêndice B.1.

PUC-Rio - Certificação Digital Nº 0611927/CA

Para representar a translação no espaço tridimensional são utilizados três parâmetros independentes (t_x, t_y, t_z). Já para representar uma rotação utiliza-se um ângulo e um eixo unitário. Tal ângulo de rotação pode ser escrito como a norma do vetor não unitário $\overrightarrow{e}$ composto pelos parâmetros w_x, w_y e w_z, de forma que o eixo de rotação é obtido através da normalização deste mesmo vetor. Assim, o ângulo de rotação θ_r passa a ser uma função desses três parâmetros:

$$\theta_r = \|\overrightarrow{e}\| = \sqrt{w_x^2 + w_y^2 + w_z^2}$$

e o eixo de rotação $\hat{e_r}$ é então calculado a partir da normalização de $\overrightarrow{e}$,

$$\hat{e_r} = \overrightarrow{e}/\|\overrightarrow{e}\| = \overrightarrow{e}/\theta_r$$

o que faz com que a rotação possa ser inteiramente representada com apenas três graus de liberdade.

O processo de formação da imagem pode ser modelado como sendo uma função $P : \Re^3 \rightarrow \Re^2$ que recebe como entrada uma posição e uma orientação de câmera (`R`, `T`) e uma linha tridimensional $\mathbf{r}_i$ e retorna um segmento de reta bidimensional [Taylor and Kriegman, 1995],[Debevec, 1996]. Tomando $\mathbf{u}_i$ como um segmento de reta que representa a projeção da linha tridimensional i do modelo, pode-se estimar um erro de projeção medindo a disparidade entre cada segmento detectado na imagem e cada uma dessas linhas 3D projetadas da seguinte forma:

$$O = \sum_{i=1}^{n} Error(P((w_x, w_y, w_z), (t_x, t_y, t_z), \mathbf{r}_i), \mathbf{u}_i) \qquad (3\text{-}22)$$

Esta dissertação utiliza a representação de linha tridimensional como sendo um ponto no $\mathbb{R}^6$[Taylor and Kriegman, 1995]. Nesta representação, são usados dois vetores tridimensionais, formando um par $(\hat{v}, \overrightarrow{d})$, que representam respectivamente a direção da reta no espaço e a distância e direção em relação a origem. Nesta representação, $(\hat{v}, \overrightarrow{d}) = (-\hat{v}, \overrightarrow{d})$, já que uma reta tridimensional se estende para as duas direções $(\hat{v}, -\hat{v})$. Essa representação é ilustrada pela Figura 3.18

Como a reta $\mathbf{r}_i$ é função das coordenadas dos vértices do modelo, temos uma função objetivo que envolve apenas os parâmetros externos da câmera, já que a câmera está calibrada.

Assim, o objetivo é encontrar a opção de parâmetros externos de câmera que forneça o melhor alinhamento possível entre os segmentos da imagem e do modelo projetado. Podemos encontrar esse conjunto ótimo de parâmetros minimizando O, como será descrito posteriormente, restando, para isso, apenas

PUC-Rio - Certificação Digital Nº 0611927/CA

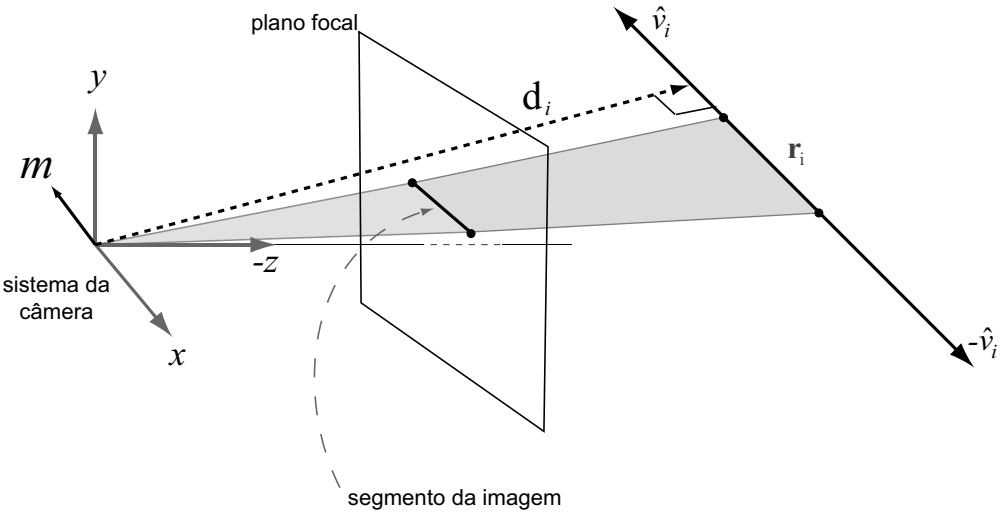


Figura 3.18: Representação de uma reta tridimensional $\mathbf{r}_i \in \mathbb{R}^6$, pelos vetores $\hat{v} \in \mathbb{R}^3, d \in \mathbb{R}^3$, escrita no sistema de coordenadas da câmera.

definirmos matematicamente a função *Error* e a função P.

Como é mostrado na Figura 3.18, a reta 3D, que é definida pelo par $\langle \mathbf{v}, \mathbf{d} \rangle$, e o centro da câmera definem um plano cuja normal é representada pelo vetor $\mathbf{m} = (m_x, m_y, m_z)^T$. A projeção da reta é então a interseção desse plano com o plano da imagem. Já que estamos utilizando a representação de câmera idêntica a do OpenGL, onde o plano de projeção se localiza em $z = -f$, temos o sistema linear

$$\begin{cases} m_x x + m_y y + m_z z &= 0 \\ z = -f \end{cases} \tag{3-23}$$

cuja solução é a equação linear da reta projetada no plano da imagem

$$m_x x + m_y y - m_z f = 0 \tag{3-24}$$

Isso significa que qualquer reta projetada pode ser escrita em função da normal $\mathbf{m}$ mostrada escrita no sistema de coordenadas da câmera.

Contudo, é preciso transformar a reta $\mathbf{r}_i$ do sistema do mundo para o sistema da câmera, o que equivale a transformar $\mathbf{d}_i$ e $\mathbf{v}_i$ para este sistema (Figura 3.19.

Então, é possível especificar a função de projeção P, como em [Taylor and Kriegman, 1995], como

$$\mathbf{m} = \mathtt{R}(\mathbf{v}_i \times (\mathbf{d}_i^w - \mathbf{t})) \tag{3-25}$$

PUC-Rio - Certificação Digital Nº 0611927/CA

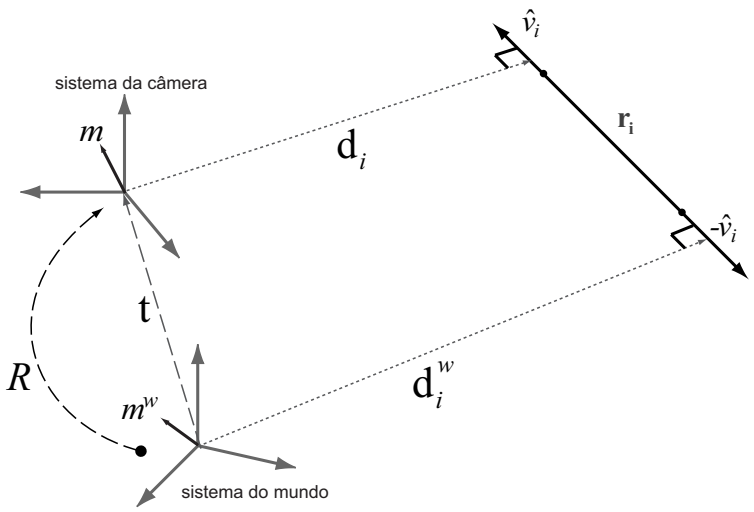


Figura 3.19: Representação da reta $\mathbf{r}_i \in \mathbb{R}^6$ e da normal $\mathbf{m}$ nos espaços da câmera e do mundo.

já que $\mathbf{m}$ indica onde o segmento projetado irá aparecer na imagem. A matriz de rotação `R` e o vetor de translação $\mathbf{t}$ compõem a transformação que leva um ponto do espaço do mundo para o espaço da câmera.

Uma discussão interessante a respeito do processo de minimização vem do fato de que seria possível considerar que a função-objetivo depende também dos parâmetros internos da câmera, o que implicaria na necessidade de obtenção de uma solução inicial para esses parâmetros que seria fornecida pelo usuário e permitiria que as etapas iniciais de calibração pudessem ser eliminadas. Nesse caso, os segmentos projetados do modelo seriam função de um mapeamento projetivo desconhecido, mas que teriam solução inicial próxima. Isso seria possível porque as coordenadas da imagem (conhecidas) são obtidas em função das coordenadas do modelo no espaço da câmera — que são constantes dada a posição e orientação iniciais fornecidas pelo usuário — e dos próprios parâmetros da mesma:

$$\begin{pmatrix} x_{im} \\ y_{im} \end{pmatrix} = \begin{pmatrix} \alpha_x \mathrm{X}/\mathrm{Z} + p_x \\ \alpha_y \mathrm{Y}/\mathrm{Z} + p_y \end{pmatrix}$$

onde $\mathrm{X}, \mathrm{Y}, \mathrm{Z}$ são coordenadas de um ponto escritas no sistema da câmera, p_x, p_y são as coordenadas do ponto principal (centro do plano de projeção), α_x, α_y são os parâmetros que representam a distância focal em pixels no modelo CCD de câmera (3.1.2) e x_{im}, y_{im} são as coordenadas projetadas no espaço da imagem, que são constantes pois são obtidas dos segmentos previamente detectados.

Contudo, prover manualmente uma solução inicial para os parâmetros

PUC-Rio - Certificação Digital Nº 0611927/CA

intrínsecos da câmera — juntamente com seus parâmetros extrínsecos — é difícil porque, além de tornarem mais complexa a minimização por atribuírem a ela mais graus de liberdade, eles influenciam diretamente a acuidade do ajuste da transformação de corpo rígido. Em outras palavras, se o usuário tiver que estimar, além de tudo, um ângulo de abertura e uma distância focal, ao invés de fornecer somente uma rotação e uma translação, ele não conseguirá ajustar de forma adequada o modelo, pois essas últimas transformações são influenciadas diretamente pela transformação projetiva. Seria difícil, assim, prover manualmente um conjunto inicial tão grande que fosse capaz de fazer convergir o processo de ajuste por minimização. É por este motivo que a calibração é feita em um passo anterior. Assim, a função-objetivo modelada assume a câmera já calibrada e ajusta apenas seus parâmetros externos. Tal função será descrita a seguir.

PUC-Rio - Certificação Digital Nº 0611927/CA

3.7.2 A função de Erro

É preciso definir matematicamente uma maneira de medir o erro entre a reta 3D projetada a partir do modelo — obtida através de uma aresta 3D — e medições feitas na imagem (segmentos detectados). Para tanto, é utilizada a função de erro proposta por [Taylor and Kriegman, 1995] e já apresentada na Seção 3.6.2 (Equação 3-20). A Figura 3.20 mostra como é feita a medição de erro utilizando a aresta projetada representada pela normal $\mathbf{m}$.

A função de erro é então escrita como:

$$\begin{aligned} Error(P((\mathrm{w_x},\mathrm{w_y},\mathrm{w_z}),(\mathrm{t_x},\mathrm{t_y},\mathrm{t_z}),\mathbf{r}_i),\mathbf{u}_i) = \int_0^l h^2(s)ds = \\ = \frac{l}{3}(h_1^2 + h_1h_2 + h_2^2) = \\ = \mathbf{m}^T(A^TBA)\mathbf{m} \end{aligned} \tag{3-26}$$

A matrizes A e B são:

$$A = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix}, B = \frac{l}{3(m_x^2 + m_y^2)} \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix}$$

Tal função leva em conta o comprimento de cada segmento como sendo um fator de peso para o ajuste porque foi originalmente desenvolvida em [Taylor and Kriegman, 1995] para ajustar segmentos marcados manualmente. Nestes casos, o usuário marca os segmentos de forma completa na imagem.

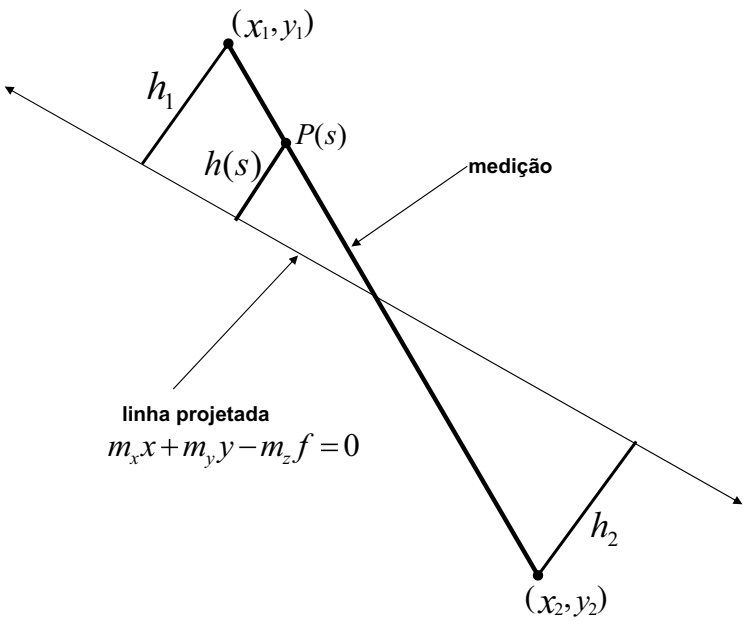


Figura 3.20: A medição do erro entre o segmento medido na imagem e o segmento que representa a linha do modelo-base projetada. Os coeficientes m_x, m_y, m_z são os coeficientes da normal (equações 3-23 e 3-24).

Contudo, isso não ocorre nesta dissertação, já que segmentos curtos, *a priori*, podem representar partes do mundo tão bem quanto segmentos longos, porque eles podem ter sido detectados automaticamente na imagem. Segmentos detectados automaticamente podem ser frações de segmentos originais que representam corretamente os segmentos da cena segundo o método de ajuste proposto. Apenas para critério de seleção de arestas em um ROI o tamanho do segmento é considerado (Seção 3.6.2). Assim, cada segmento selecionado no conjunto final de correspondências entre modelo e imagem tem peso igual e a matriz `B` usada aqui foi adaptada:

$$B = \frac{1}{(m_x^2 + m_y^2)} \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix}$$

A função-objetivo O é então definida como:

$$\sum_{i=1}^{n} Error(P(R, t, \mathbf{r}_i), \mathbf{u}_i) = \sum_{i=1}^{n} \mathbf{m}^T (A^T B A) \mathbf{m}$$

Para encontrar os parâmetros de câmera em função do erro provido por esta função, devemos minimizá-la utilizando algum método de gradiente-descente. Isso envolve calcular as derivadas segunda e primeira desta função em relação a cada um de seus parâmetros. Como esta função é não linear em função dos parâmetros de rotação, ela deve ser derivada de uma forma

PUC-Rio - Certificação Digital Nº 0611927/CA

especial utilizando o Grupo $S0(3)$ em conjunto com *manifolds* especiais, como é mostrado no Apêndice B. A função de erro pode ser derivada utilizando-se diferenças finitas através de métodos numéricos [Ruggiero and da Rocha Lopes, 1996], mas ainda sim as técnicas apresentadas em tal apêndice devem ser aplicadas.

3.8 Limitações e Vantagens

Esta seção tem como objetivo analisar resumidamente as vantagens e limitações das etapas que foram recém-apresentadas.

Limitações:

1. Este método, composto pelas várias etapas mostradas, não é automático e tem uma forte dependência das informações providas pelo usuário. Isso faz com que o usuário possa vir a ter dificuldade em ajustar algumas imagens que contêm muito ruído ou cujo objeto retratado se encontra distante ou pouco visível;

2. Por utilizar uma estratégia local de busca por relações entre modelo e imagem, o método é incapaz de ajustar corretamente a posição e a orientação da câmera quando o usuário não provê um bom posicionamento inicial. Em outras palavras o método permite que qualquer segmento próximo a um segmento projetado do modelo seja associado a ele, independentemente se tal segmento corresponde corretamente ao modelo. Isso também faz com que a câmera simplesmente não possa ser reconstruída a partir de certas fotos quando o conjunto de associações imagem-modelo é inconsistente — quando não se pode calcular um mapeamento entre os segmentos do mundo e os segmentos selecionados na imagem;

3. O processo de minimização mostrado só é capaz de ajustar corretamente a câmera casos sejam fornecidas associações em cada uma das direções principais da cenas (x, y, z) ou em outras três direções ortogonais. Se a imagem tiver apenas uma fachada visível contida, por exemplo, no plano xy, a reconstrução de câmera será imprecisa ou pode ser inviável.

Vantagens:

1. Embora a estratégia seja semi-automática e dependa da interação do usuário, ela é desenvolvida de modo que ele nunca precise informar diretamente relações entre imagem e modelo. O usuário apenas precisa

PUC-Rio - Certificação Digital Nº 0611927/CA

marcar segmentos na imagem que não puderam ser detectados automaticamente devido à grande quantidade de ruído. A associação entre imagem e modelo é feita, então, automaticamente utilizando a estratégia local apresentada;

2. Por ser semi-automático, o método provê toda a flexibilidade para que o usuário possa controlar o processo de ajuste. Isso significa que, mesmo que todas os segmentos associados ao modelo tenham sido marcado pelo usuário, ainda é possível reconstruir a câmera. Esse fato representa vantagem em relação a sistemas totalmente automatizados, quando se trata de imagens reais com grande quantidade de ruído, pois tais sistemas podem simplesmente não reconstruir a câmera, dada a complexidade de relacionar de forma totalmente automática imagem e modelo;

3. Em casos muito simples, em que a imagem possuir pouco ruído e a geometria do modelo retratado é simples, o método pode reconstruir totalmente a câmera de modo quase automatizado, de tal forma que o usuário tenha pouco esforço.

Embora existam outras possíveis falhas e desvantagens inerentes ao método proposto, estas apresentadas aqui, de forma resumida, são as principais e que influenciam diretamente seu uso. Na seção de Trabalhos Futuros (6.1) serão expostas algumas idéias para aperfeiçoar ou inovar certas técnicas que compõem este método — como melhorias no sistema de associação imagem-modelo, além de algumas outras que propõem soluções para outros fins relacionados.

O passo final do método proposto é o registro da câmera reconstruída. Tal passo consiste no armazenamento dos parâmetros de câmera obtidos para que seja possível posteriormente efetuar a navegação pelos diversos pontos de vista obtidos individualmente — representados por estas câmeras. Essas funcionalidades — registro e navegação — são descritas no próximo capítulo, em que uma aplicação exemplo é apresentada. Essa aplicação implementa o método proposto e disponibiliza diversas ferramentas para facilitar o processo de reconstrução de câmera.

PUC-Rio - Certificação Digital Nº 0611927/CA