

Revisão de Conceitos em Projeção, Homografia, Calibração de  
Câmera, Geometria Epipolar, Mapas de Profundidade e  
Varredura de Planos

Maikon Cismoski dos Santos\*

**Disciplina:** Visão Computacional  
Prof. Dr. Anderson Rocha

Campinas, Outono/Inverno de 2012

---

\*Aluno de Doutorado do Instituto de Computação, Unicamp, sob orientação do Prof. Dr. Hélio Pedrini.

# Sumário

<b>A Conhecimentos Básicos</b>	<b>1</b>
A.1 Modelo de Câmera Estenopeica . . . . .	1
A.1.1 Parâmetros Intrínsecos . . . . .	1
A.1.2 Parâmetros Extrínsecos . . . . .	4
A.2 Projeção de Pontos 2D para Raios no Espaço 3D . . . . .	4
A.3 Homografia 2D . . . . .	5
A.3.1 Estimação da Homografia 2D . . . . .	6
A.3.2 Normalização de Dados . . . . .	8
A.4 Calibração da Câmera . . . . .	10
A.4.1 Homografia entre o Plano do Padrão e o Plano da Imagem . . . . .	11
A.4.2 Estimação dos Parâmetros Intrínsecos . . . . .	12
A.4.3 Estimação dos Parâmetros Extrínsecos . . . . .	15
A.5 Geometria Epipolar . . . . .	15
A.5.1 Matriz Fundamental . . . . .	16
A.5.2 Matriz Essencial . . . . .	19
A.5.3 Triangulação . . . . .	19
A.5.4 Transferência de Pontos Usando Matrizes Fundamentais . . . . .	22
A.6 Decomposição em Valores Singulares . . . . .	23
A.6.1 Descomposição . . . . .	23
A.6.2 Relação entre Valores Singulares e Autovalores . . . . .	24
A.6.3 Solução de Mínimos Quadrados de Sistemas Homogêneos . . . . .	24
<b>B Conhecimentos Adicionais</b>	<b>26</b>
B.1 Mapa de Profundidade . . . . .	26
B.2 Light Field e Lumigraph . . . . .	34
B.3 Varredura de Planos . . . . .	38
<b>Referências</b>	<b>46</b>

## A Conhecimentos Básicos

### A.1 Modelo de Câmera Estenopeica

Esta seção descreve o método de aquisição de imagem conhecido como modelo de câmera estenopeica (câmera *pinhole*). Este modelo de câmera define um mapeamento geométrico do mundo 3D para o plano da imagem 2D, conhecido como projeção perspectiva. Esta seção está subdividida em duas partes. Na subseção A.1.1, os parâmetros intrínsecos da câmera, como distância focal e distorção das lentes, são descritos e, na subseção A.1.2, os parâmetros extrínsecos como a orientação e translação da câmera são apresentados.

#### A.1.1 Parâmetros Intrínsecos

O modelo de câmera estenopeica é ilustrado na Figura 1 e detalhado a seguir.  $C$  representa o centro de projeção, ponto em que todos os raios intersectam-se, também conhecido como *centro da câmera* ou *centro óptico*. Considerando que  $C$  está na origem, o plano  $Z = f$ , conhecido como *plano da imagem* ou *plano focal*, é posicionado em frente ao centro de projeção e  $f$  é a distância focal. A linha que passa por  $C$  e é perpendicular ao plano da imagem é chamada de *eixo principal* ou *eixo óptico* e o ponto onde o eixo principal encontra o plano da imagem é chamado de *ponto principal* (o ponto  $p$  ilustrado na Figura 1).

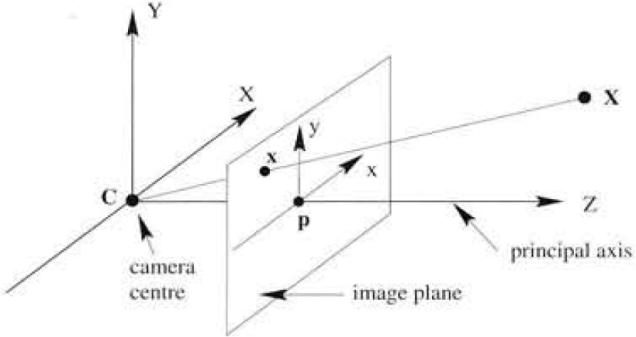


Figura 1: Modelo de câmera estenopeica. Fonte: [13].

Conforme [13], empregando o modelo de câmera estenopeica e considerando que o centro de projeção está localizado na origem do sistema de coordenadas 3D e o eixo óptico é colinear ao eixo  $Z$ , como mostra a Figura 1, um ponto no espaço com coordenadas  $(X, Y, Z)^T$  é mapeado para um ponto no plano da imagem  $(u, v)^T$  por meio da seguinte equação

$$(u, v)^T = (fX/Z, fY/Z)^T \quad (1)$$

Se os pontos no espaço 3D e os pontos no plano da imagem são representados por coordenadas

homogêneas, a equação 1 pode ser escrita em notação matricial como

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2)$$

em que  $\lambda = Z$  é o fator de escala homogêneo.

Nas equações 1 e 2, assume-se que a origem das coordenadas no plano da imagem é o ponto principal. Entretanto, a maioria dos sistemas de imagens define a origem como sendo o pixel com a coordenada mais à esquerda e mais acima, exigindo um mapeamento para converter o sistema de coordenadas no plano da imagem quando necessário. Em [13], utilizando coordenadas homogêneas, este mapeamento é expresso por

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3)$$

em que  $(p_x, p_y)^T$  são as coordenadas do ponto principal.

A equação 3 considera que as coordenadas da imagem possuem a mesma escala na direção do eixo  $x$  e  $y$ , ou seja, os pixels têm o formato quadrado 1 : 1. A razão entre a largura e a altura do pixel é denominada razão de aspecto. Entretanto, em câmeras que adotam o sensor de imagem do tipo CCD [19], há a possibilidade de que pixels não sejam quadrados [13]. Se as coordenadas da imagem são medidas em pixels e o número de pixels por unidade de distância nas coordenadas da imagem são  $\eta_x$  e  $\eta_y$  (na direção do eixos  $x$  e  $y$ , respectivamente), segundo [13], a equação 3 pode ser reescrita como

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f\eta_x & 0 & p_x & 0 \\ 0 & f\eta_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4)$$

Outro fator a ser considerado é o de inclinação da imagem, embora na maioria das câmeras esse parâmetro seja zero (não há inclinação) [13]. A Figura 2 (b) ilustra uma imagem com inclinação. Levando em consideração o parâmetro de inclinação  $\tau$ , a equação 4 pode ser escrita como

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f\eta_x & \tau & p_x & 0 \\ 0 & f\eta_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} K & | & \mathbf{0}_3 \end{bmatrix} Q \quad (5)$$

em que  $\mathbf{0}_3$  é um vetor nulo,  $Q = (X, Y, Z, 1)^T$  é um ponto no espaço 3D e  $K$  representa os

parâmetros intrínsecos da câmera, também conhecidos como matriz de calibração da câmera. Quando a razão de aspecto é  $1 : 1$ ,  $\eta_x = 1$  e  $\eta_y = 1$ . Se a imagem não é inclinada, então  $\tau = 0$ .

Até aqui, os parâmetros intrínsecos empregados no processo de formação de imagem pelo modelo de câmera estenopeica foi descrito. Entretanto, as lentes das câmera reais podem sofrer com algum tipo distorção. Para [13], a modelagem exata da lentes é uma tarefa complexa, sendo a distorção radial a mais importante a ser considerada. A distorção radial causa o mapeamento de uma reta, para uma linha com uma determinada curvatura, como mostra a Figura 3. Para [25], a relação entre a posição dos pixels com distorção radial corrigida  $(x_u, y_u)^T$  e com distorção radial  $(x_d, y_d)^T$  é definida por

$$\begin{bmatrix} x_u - p_x \\ y_u - p_y \end{bmatrix} = L(r_d) \begin{bmatrix} x_d - p_x \\ y_d - p_y \end{bmatrix} \quad (6)$$

em que  $(p_x, p_y)^T$  são as coordenadas do ponto principal e  $L(r_d) = 1 + k_1 r_d^2$ , em que  $k_1$  é a quantidade de distorção radial presente na imagem e  $r_d^2 = (x_d - p_x)^2 + (y_d - p_y)^2$ .

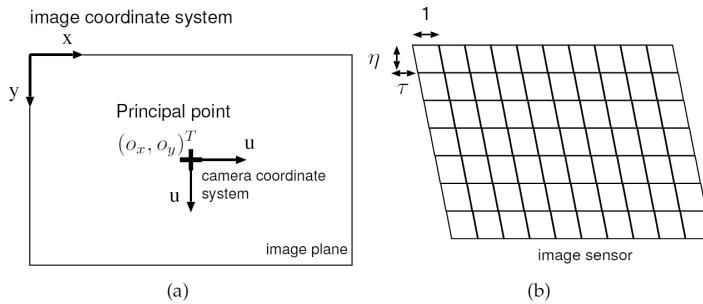


Figura 2: (a) plano de imagem com o sistema de coordenadas da câmera  $(u, v)^T$  e a imagem  $(x, y)^T$ ; (b) imagem inclinada e as coordenadas da imagem com diferentes escalas na direção dos eixos  $x$  e  $y$ . Fonte: [25].

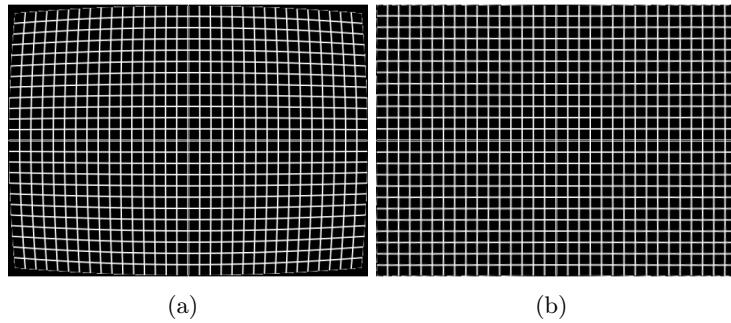


Figura 3: (a) imagem com distorção radial; (b) imagem com distorção radial corrigida. Fonte: [5].

A correção da distorção radial na imagem requer a estimativa dos parâmetros  $k_1$  e  $(p_x, p_y)^T$  da equação 6. Estes parâmetros podem ser estimados pelo cálculo da curvatura de uma linha na imagem 2D, que é uma linha reta no espaço 3D. Utilizando um padrão de calibração, as

linhas destacadas na Figura 4, no mundo real, são linhas retas, entretanto, a imagem capturada apresenta as mesmas linhas com uma determinada curvatura. Mais detalhes de como estimar os parâmetros para a correção da distorção radial são descritos em [5] e [18].

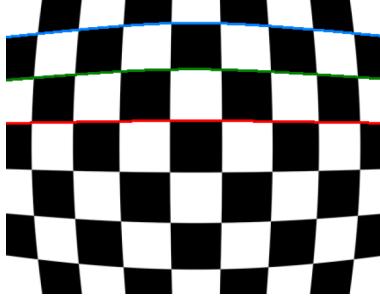


Figura 4: Estimação da distorção radial utilizando um padrão de calibração. Fonte: [18].

### A.1.2 Parâmetros Extrínsecos

Os parâmetros extrínsecos relacionam o sistema de coordenadas da câmera com o sistema de coordenadas do mundo, descrevendo a posição e orientação da câmera no mundo 3D, como mostra a Figura 5. A equação 5 considera que a câmera está na origem do sistema de coordenadas no mundo e o plano  $Z = f$  é o plano da câmera, como detalhado na subseção A.1.1. Para [13], a posição e orientação da câmera no sistema de coordenadas no mundo podem ser escritas como

$$\lambda q = \begin{bmatrix} K & | & \mathbf{0}_3 \end{bmatrix} \begin{bmatrix} R & -R\tilde{C} \\ 0_3^T & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = KR \begin{bmatrix} I & | & -\tilde{C} \end{bmatrix} Q \quad (7)$$

em que a matriz  $K$  define os parâmetros intrínsecos detalhados em A.1.1,  $R$  e  $\tilde{C}$  representam os parâmetros extrínsecos da câmeras, em que  $R$  é uma matriz de rotação  $3 \times 3$  e  $\tilde{C}$  define o centro de projeção da câmera no mundo em coordenadas não-homogêneas,  $I$  é uma matriz identidade  $3 \times 3$  e  $q = (x, y, 1)^T$  e  $Q = (X, Y, Z, 1)^T$  representam o mesmo ponto nos sistemas de coordenadas da câmera e no mundo 3D, respectivamente.

A equação 7 define o mapeamento completo da câmera estenopeica,  $P = KR[I|-\tilde{C}]$ , incluindo os parâmetros intrínsecos e extrínsecos. A matriz  $P$ , de dimensão  $3 \times 4$ , é conhecida como a *matriz de projeção da câmera*, em que um ponto  $Q$  do mundo 3D é mapeado para um ponto  $q$  no plano da imagem 2D por  $q = PQ$ .

## A.2 Projeção de Pontos 2D para Raios no Espaço 3D

Na seção A.1 foi descrito como mapear um ponto no espaço 3D para um ponto 2D no plano da imagem. Esta seção apresenta o processo inverso, conhecido na literatura como *back-projection*,

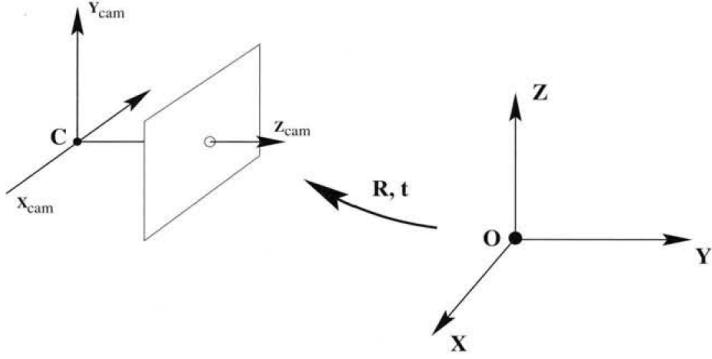


Figura 5: Transformação Euclidiana entre o sistema de coordenadas da câmera e do mundo. Fonte: [13].

em que um ponto  $q$  na imagem é projetado para um conjunto de pontos no espaço 3D, sendo que os pontos 3D pertencentes a este conjunto constituem uma raio no espaço, o qual passa pelo centro de projeção da câmera.

Para [25], o raio  $Q(\lambda)$  que passa pelo centro da câmera  $\tilde{C} = (\tilde{C}_x, \tilde{C}_y, \tilde{C}_z)^T$  e pelo ponto  $q = (x, y, 1)^T$  no plano da imagem é definido como

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{\tilde{C} + \lambda R^{-1} K^{-1} q}_{\text{Raio } Q(\lambda)} \quad (8)$$

em que  $\lambda$  é o fator escalar positivo que define a posição do ponto 3D  $(X, Y, Z)^T$  sobre o raio. Se a coordenada  $Z$  é conhecida, as coordenadas  $X$  e  $Y$  podem ser obtidas por

$$\lambda = \frac{Z - \tilde{C}_z}{z_3} \quad (9)$$

em que  $(z_1, z_2, z_3)^T = R^{-1} K^{-1} q$ .

### A.3 Homografia 2D

Homografia 2D é uma transformação projetiva planar que mapeia pontos de um plano para outro plano [13]. Este processo é ilustrado na Figura 6, em que o ponto  $x$  no plano  $\pi$  é mapeado para seu o ponto correspondente  $x'$  no plano  $\pi'$ . Este mapeamento linear de pontos pode ser escrito em coordenadas homogêneas como  $x'_i = Hx_i$ , em que  $H$  é a matriz de homografia que define o mapeamento de um conjunto de pontos correspondentes  $x_i \leftrightarrow x'_i$  entre dois planos.

Esta seção está dividida em duas partes. Na subseção A.3.1, o algoritmo Transformação Linear Direta (DLT) é apresentado e empregado na estimativa da matriz de homografia  $H$  e, na subseção A.3.2, uma versão do algoritmo DLT com normalização de dados é descrita.

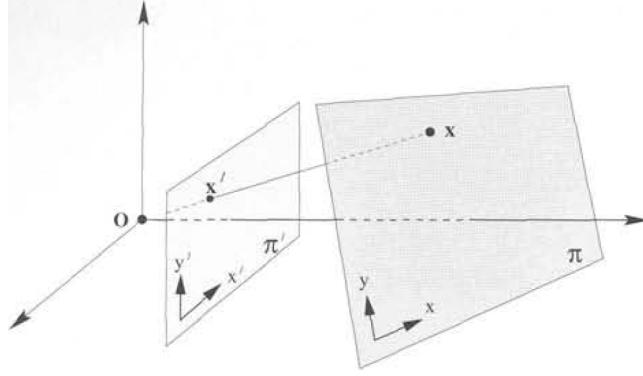


Figura 6: Mapeamento entre planos. Fonte: [13].

### A.3.1 Estimação da Homografia 2D

Dado um conjunto de pontos  $p_i = (x_i, y_i, w_i)^T$  sobre um plano  $\pi$  e outro conjunto de pontos correspondentes  $p'_i = (x'_i, y'_i, w'_i)^T$  em um plano  $\pi'$ , esta seção trata do problema de computar uma transformação projetiva  $H$  que mapeia cada ponto  $p_i$  para  $p'_i$ . Este mapeamento pode ser escrito como

$$\begin{bmatrix} x'_i \\ y'_i \\ w'_i \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix} \quad (10)$$

ou como  $p'_i = Hp_i$ , em que  $H$  é uma matriz  $3 \times 3$ , não-singular.

Note que na Equação 10, os pontos 2D são expressos em coordenadas homogêneas. Embora  $p'_i$  e  $Hp_i$  representem o mesmo ponto no plano 2D, analisando  $p'_i$  e  $Hp_i$  como vetores 3D, eles não são iguais, pois, embora tenham a mesma direção, eles podem ter uma magnitude diferente para um fator de escala homogêneo  $\lambda \neq 0$ . Assim, a Equação 10 pode ser reescrita como

$$\lambda p'_i = Hp_i \quad (11)$$

Para eliminar o fator de escala, a Equação 11 pode ser expressa em termos de um produto vetorial como

$$p'_i \times (\lambda p'_i) = p'_i \times Hp_i \quad (12)$$

que pode ser reescrita como

$$p'_i \times Hp_i = \mathbf{0}_3 \quad (13)$$

Dessa forma, uma solução linear de  $H$  pode ser derivada da equação 13. Escrevendo a  $j$ -ésima linha da matriz  $H$  como  $\mathbf{h}^{jT}$ , o produto  $Hp_i$  pode ser denotado por

$$Hp_i = \begin{bmatrix} \mathbf{h}^{1T} p_i \\ \mathbf{h}^{2T} p_i \\ \mathbf{h}^{3T} p_i \end{bmatrix}. \quad (14)$$

Utilizando a Equação 14, o produto vetorial da Equação 13 pode ser reformulado como

$$p'_i \times Hp_i = \begin{bmatrix} y'_i \mathbf{h}^{3T} p_i - w'_i \mathbf{h}^{2T} p_i \\ w'_i \mathbf{h}^{1T} p_i - x'_i \mathbf{h}^{3T} p_i \\ x'_i \mathbf{h}^{2T} p_i - y'_i \mathbf{h}^{1T} p_i \end{bmatrix} = \mathbf{0}_3 \quad (15)$$

em que  $p'_i = (x'_i, y'_i, w'_i)^T$ .

Uma vez que  $\mathbf{h}^{jT} p_i = p_i^T \mathbf{h}^j$ , a Equação 15 pode ser reescrita como

$$\begin{bmatrix} \mathbf{0}_3^T & -w'_i p_i^T & y'_i p_i^T \\ w'_i p_i^T & \mathbf{0}_3^T & -x'_i p_i^T \\ -y'_i p_i^T & x'_i p_i^T & \mathbf{0}_3^T \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix} = A_i \mathbf{h} = \mathbf{0}_9 \quad (16)$$

em que  $A_i$  é uma matriz  $3 \times 9$ , tal que suas entradas são formadas pelas coordenadas de um par de pontos correspondentes conhecidos e  $\mathbf{h}$  é um vetor coluna contendo as entradas não conhecidas de  $H$ , ou seja,  $\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^T$ .

Note que, das três linhas da matriz  $A_i$  na Equação 16, somente duas são linearmente independentes. A terceira linha é obtida pela soma de  $-x'_i$  vezes a primeira linha e  $-y'_i$  vezes a segunda. Segundo [13], pode-se assumir que  $w'_i = 1$  e a terceira linha pode ser omitida para a solução de  $H$ . Assim, a Equação 16 pode ser reescrita como

$$\begin{bmatrix} \mathbf{0}_3^T & -w'_i p_i^T & y'_i p_i^T \\ w'_i p_i^T & \mathbf{0}_3^T & -x'_i p_i^T \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix} = A_i \mathbf{h} = \mathbf{0}_9 \quad (17)$$

em que  $A_i$  é uma matriz  $2 \times 9$ .

A matriz  $H$  possui 9 entradas e 8 graus de liberdade. Baseado nos graus de liberdade de  $H$ , um limite inferior  $n$ , que determina a quantidade de pontos correspondentes  $p_i \leftrightarrow p'_i$  necessários para computar a transformação projetiva  $H$  pode ser estabelecido. Cada par de pontos correspondentes tem dois graus de liberdade, pois as coordenadas de um ponto  $p_i$  são determinadas por dois elementos  $x$  e  $y$  (o fator de escala homogêneo é arbitrário) e os dois graus de liberdade do ponto  $p_i$  devem corresponder ao ponto mapeado  $Hp_i$ . Assim, pelo menos quatro pontos correspondentes são necessários ( $n = 4$ ), sendo que, para cada par de pontos correspondentes  $p_i \leftrightarrow p'_i$ , uma matriz  $A_i$  de dimensão  $2 \times 9$  é computada. Então, a matriz  $A$  de dimensão  $2n \times 9$ , composta por cada matriz  $A_i$ , estabelece o sistema linear  $A\mathbf{h} = 0$ , cuja solução resolve  $H$ . O sistema linear  $A\mathbf{h} = 0$

pode ser escrito como

$$\left[ \begin{array}{ccccccccc} 0 & 0 & 0 & -w'_1x_1 & -w'_1y_1 & -w'_1w_1 & y'_1x_1 & y'_1y_1 & y'_1w_1 \\ w'_1x_1 & w'_1y_1 & w'_1w_1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1w_1 \\ 0 & 0 & 0 & -w'_2x_2 & -w'_2y_2 & -w'_2w_2 & y'_2x_2 & y'_2y_2 & y'_2w_2 \\ w'_2x_2 & w'_2y_2 & w'_2w_2 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 & -x'_2w_2 \\ \vdots & \vdots \\ 0 & 0 & 0 & -w'_nx_n & -w'_ny_n & -w'_nw_n & y'_nx_n & y'_ny_n & y'_nw_n \\ w'_nx_n & w'_ny_n & w'_nw_n & 0 & 0 & 0 & -x'_nx_n & -x'_ny_n & -x'_nw_n \end{array} \right] \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \mathbf{0} \quad (18)$$

Para [13], se o sistema linear apresentado na Equação 18 for construído a partir de 4 pontos correspondentes, a matriz  $A$  terá dimensão  $8 \times 9$  e se o posto de  $A$  é 8, existe um espaço nulo que provê a solução de  $\mathbf{h}$ . Quando o número de pontos correspondentes é maior do que 4, o sistema é sobredeterminado, ou seja, o número de equações é maior que o número de incógnitas e duas situações devem ser consideradas. Na primeira, assume-se que a posição dos pontos é exata e o posto da matriz  $A$  é 8, então, a solução também é determinada pelo espaço nulo de  $A$ . Na segunda hipótese, a posição dos pontos não é exata (a correspondência dos pontos foi estimada por um algoritmo, por exemplo) e a solução exata para o sistema  $A\mathbf{h} = 0$  é inexistente e uma solução aproximada deve ser calculada.

Uma solução aproximada para o sistema linear homogêneo  $A\mathbf{h} = 0$  pode ser obtida pela Decomposição em Valores Singulares (SVD), em que  $\mathbf{h}$  é o autovetor da matriz  $A^T A$  correspondente ao menor autovalor de  $A^T A$ , conforme detalhado na seção A.6.

### A.3.2 Normalização de Dados

Nesta subseção, uma versão com normalização de dados do algoritmo DTL (descrito em A.3) é apresentada. Segundo [13], a normalização das coordenadas dos pontos correspondentes é necessária para evitar instabilidade numérica, produzindo resultados mais precisos na estimativa da homografia.

Considerando os pontos correspondentes  $p_i \leftrightarrow p'_i$  e uma matriz de homografia  $H$ , a normalização das coordenadas de  $p_i$  e  $p'_i$  é sumarizada como segue [13]:

- (i) realizar uma transformação de translação nos pontos, de modo que a origem seja o centróide do plano.
- (ii) aplicar uma transformação de escala nos pontos, de forma que a distância média de um ponto  $p$  da sua origem seja  $\sqrt{2}$ .
- (iii) efetuar as transformações do passos (i) e (ii) nos pontos  $p_i$  e  $p'_i$  de maneira independente.

Assim, a normalização é caracterizada pelas transformações de translação e escala sobre os pontos  $p_i$  e  $p'_i$ , que pode ser escrita como  $\tilde{p}_i = Tp_i$  e  $\tilde{p}'_i = Tp'_i$ , em que

$$T = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \\ 0 & 0 & 1 \end{bmatrix} \text{ e } T' = \begin{bmatrix} s' & 0 & t'_x \\ 0 & s' & t'_y \\ 0 & 0 & 1 \end{bmatrix} \quad (19)$$

Os parâmetros  $s$  e  $t = (t_x, t_y)^T$  definem, respectivamente, os fatores de escala e translação para a matriz  $T$  e, de forma similar,  $s'$  e  $t' = (t'_x, t'_y)^T$  para a matriz  $T'$ . A descrição de como esses parâmetros são calculados é detalhada a seguir. Somente os parâmetros da matriz  $T$  são descritos, pois a matriz  $T'$  pode ser calculada de forma semelhante, empregando os mesmos conceitos.

Considerando os pontos  $p_i = (x_i, y_i, 1)^T$  e uma transformação  $T$  (Equação 19), a normalização de  $p_i$  é dada por

$$\tilde{p}_i = Tp_i = \begin{bmatrix} sx_i + t_x \\ sy_i + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \\ 1 \end{bmatrix} \quad (20)$$

em que  $\tilde{p}_i = (\tilde{x}, \tilde{y}, 1)^T$  é o ponto  $p_i$  normalizado.

Conforme sumarizado anteriormente, uma etapa da normalização consiste em transladar os pontos do plano de forma que a origem seja o centróide do plano (o ponto com coordenadas  $(0, 0, 1)^T$ ). Com base nisso, se antes da normalização a origem do sistema de coordenadas do plano  $\pi$ , no qual os pontos  $p_i$  pertencem, é o ponto  $\bar{p} = (\bar{x}, \bar{y}, 1)^T$  e o centróide de  $\pi$  é denotado por  $\tilde{\bar{p}} = (\tilde{\bar{x}}, \tilde{\bar{y}}, 1)^T = (0, 0, 1)^T$ , em que  $\tilde{\bar{p}}$  representa o ponto  $\bar{p}$  normalizado, a seguinte relação pode ser estabelecida

$$\begin{bmatrix} \tilde{\bar{x}} \\ \tilde{\bar{y}} \\ 1 \end{bmatrix} = \begin{bmatrix} s\bar{x} + t_x \\ s\bar{y} + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (21)$$

em que  $t_x = -s\bar{x}$  e  $t_y = -s\bar{y}$ .

Com o parâmetro  $t$  calculado, resta o cálculo de  $s$  a partir da premissa de que a distância média de um ponto  $p$  da sua origem é  $\sqrt{2}$ , que pode ser escrito como

$$\begin{aligned} \sqrt{2} &= \frac{1}{n} \sum_i \sqrt{(\tilde{x}_i - \bar{x})^2 + (\tilde{y}_i - \bar{y})^2} \\ &= \frac{1}{n} \sum_i \sqrt{\tilde{x}_i^2 + \tilde{y}_i^2} \\ &= \frac{1}{n} \sum_i \sqrt{(sx_i + t_x)^2 + (sy_i + t_y)^2} \\ &= \frac{1}{n} \sum_i \sqrt{(sx_i - s\bar{x})^2 + (sy_i - s\bar{y})^2} \\ &= \frac{s}{n} \sum_i \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \end{aligned} \quad (22)$$

Assim,  $s$  pode ser calculado a partir da Equação 22 como

$$s = \frac{\sqrt{2}}{\frac{1}{n} \sum_i \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2}} \quad (23)$$

Com  $s$  e  $t$  computados, a matriz  $T$  da Equação 19 define a normalização dos pontos  $p_i$ . A matriz  $T'$  é computada de forma similar, utilizando as Equações 21 e 23 para os pontos  $p'_i$ .

A a partir da normalização dos pontos detalhada, a versão do algoritmo DLT com normalização de dados é sumarizada a seguir:

- (i) normalizar as coordenadas dos pontos  $p_i$ , obtendo os pontos  $\tilde{p}_i$ , empregando as Equações 21 e 23.
- (ii) normalizar as coordenadas dos pontos  $p'_i$  de forma semelhante a do passo anterior, computando os pontos  $\tilde{p}'_i$  normalizados.
- (iii) aplicar o algoritmo DLT, descrito em A.3, para os pontos correspondentes  $\tilde{p}_i \leftrightarrow \tilde{p}'_i$ , computando a matriz de homografia  $\tilde{H}$ .
- (iv) obter a matriz  $H$  da partir de  $\tilde{H}$ , efetuando uma desnormalização detalhada na equação a seguir

$$p'_i = \underbrace{H}_{\text{homografia H}} p_i = (T')^{-1} \tilde{p}'_i = (T')^{-1} \tilde{H} \tilde{p}_i = \underbrace{(T')^{-1} \tilde{H} T}_{\text{homografia H}} p_i \quad (24)$$

#### A.4 Calibração da Câmera

O modelo de câmera estenopeica foi descrito na seção A.1. Esta seção descreve como os parâmetros intrínsecos e extrínsecos da câmera podem ser estimados por meio de um método de calibração da câmera. O algoritmo de calibração da câmera apresentado nesta seção é o de Zhang [46]. O processo de estimativa de todos os parâmetros da câmera também é conhecido *calibração forte* ou, alternativamente, que a câmera é fortemente calibrada.

O algoritmo utiliza padrões de calibração, tal como o padrão de tabuleiro ilustrado na Figura 7, que tem uma superfície plana, sua geometria é conhecida e possui quadrados, alternados em branco e preto. O primeiro passo da calibração da câmera consiste na captura de diversas imagens do padrão de calibração adotado, em diversas orientações e posições (pelo menos 3 imagens são necessárias). Na segunda etapa, os cantos de cada quadrado (*features*) são detectados nas imagens e a correspondência entre os pontos 2D nas imagens e os 3D no plano do tabuleiro é computada, atribuindo a origem do sistema de coordenadas do mundo para um ponto relativo ao canto de um quadrado no tabuleiro (Figura 7) e assim fazendo com que todos os pontos no tabuleiros estejam sobre o plano. Na literatura, há diversas abordagens para a extração de pontos dos padrões de calibração [36, 40]. Por fim, na terceira etapa, empregando a correspondência computada na etapa anterior, os parâmetros intrínsecos e extrínsecos são estimados, como detalhado nas subseções A.4.2 e A.4.3, respectivamente.

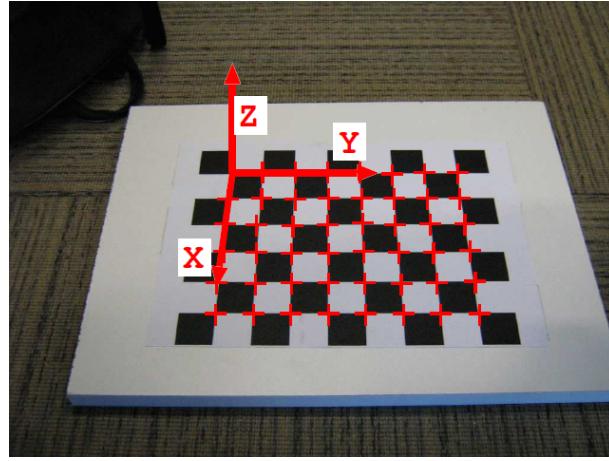


Figura 7: Padrão de tabuleiro empregado no processo de calibração da câmera. O sistema de coordenadas do mundo é atribuído para o ponto relativo ao canto do quadrado superior mais à esquerda. Os cantos de cada quadrado são destacados na cor vermelha. Fonte: [25].

#### A.4.1 Homografia entre o Plano do Padrão e o Plano da Imagem

Na seção A.1, o modelo de câmera estenopeica foi descrito e a projeção de um ponto 3D no mundo para um ponto 2D no plano da imagem é dada por

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = KR \begin{bmatrix} I & -\tilde{C} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (25)$$

que pode ser reescrita como

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (26)$$

em que  $t = -R\tilde{C}$  e  $\mathbf{r}_i$  denota a  $i$ -ésima coluna da matriz  $R$ .

Considerando que o plano do padrão de calibração (tabuleiro da Figura 7, por exemplo) está sobre  $Z = 0$  no sistema de coordenadas do mundo, a Equação 26 pode ser reformulada para

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z=0 \\ 1 \end{bmatrix} = K \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (27)$$

Assim, dado um ponto  $X$  sobre o plano do padrão de calibração e um ponto  $x$  no plano da imagem, a homografia que realiza o mapeamento  $\lambda x = HX$  é dada por

$$H = K \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & t \end{bmatrix} \quad (28)$$

Utilizando a correspondência entre os ponto 3D (*features*) do padrão de calibração e os pontos 2D das imagens inicialmente capturadas, com pelo menos 4 pontos correspondentes, a matriz de homografia  $H$  pode ser estimada conforme detalhado na seção A.3.

#### A.4.2 Estimação dos Parâmetros Intrínsecos

Com a homografia  $H$  computada, o mapeamento dos pontos entre o plano do padrão de calibração e o plano da imagem é estabelecido. Considerando que a  $i$ -ésima coluna da matriz  $H$  é denotada por  $\mathbf{h}_i$ , tem-se que  $H = [\mathbf{h}_1 \mathbf{h}_2 \mathbf{h}_3]$  e, pela Equação 28, a seguinte igualdade pode ser estabelecida

$$\begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{bmatrix} = \lambda K \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & t \end{bmatrix} \quad (29)$$

em que  $\lambda$  é o fator de escala homogêneo. A Equação 29 por ser reescrita como

$$\frac{1}{\lambda} K^{-1} \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & t \end{bmatrix} \quad (30)$$

em que

$$\begin{aligned} \mathbf{r}_1 &= \frac{1}{\lambda} K^{-1} \mathbf{h}_1 \\ \mathbf{r}_2 &= \frac{1}{\lambda} K^{-1} \mathbf{h}_2 \end{aligned}$$

Assumindo que  $\mathbf{r}_1$  e  $\mathbf{r}_2$  são ortonormais,  $\lambda = \|K^{-1}\mathbf{h}_1\| = \|K^{-1}\mathbf{h}_2\|$ , uma vez que  $\|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1$  e duas restrições podem ser estabelecidas: (i)  $\mathbf{r}_1$  e  $\mathbf{r}_2$  são perpendiculares, então  $\mathbf{r}_1 \cdot \mathbf{r}_2 = 0$ , em que " $\cdot$ " denota o produto escalar e (ii)  $\|\mathbf{r}_1\|^2 = \|\mathbf{r}_2\|^2$ , uma vez que  $\|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1$ . Com base na Equação 30, estas restrições podem ser escritas como

$$\mathbf{h}_1^T K^{-T} K^{-1} \mathbf{h}_2 = 0 \quad (31)$$

$$\mathbf{h}_1^T K^{-T} K^{-1} \mathbf{h}_1 = \mathbf{h}_2^T K^{-T} K^{-1} \mathbf{h}_2 \quad (32)$$

em que  $K^{-T} = (K^{-1})^T$  e a matriz  $K$  é desconhecida.

Por questão de correspondência entre a notação utilizada por este trabalho e a adotada por [46], a matriz  $K$  que representa os parâmetros intrínsecos da câmera, detalhados na seção A.1.1, possui a seguinte igualdade

$$K = \begin{bmatrix} f\eta_x & \tau & p_x \\ 0 & f\eta_y & p_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (33)$$

e, segundo [46],  $K^{-T}K^{-1}$  pode ser escrito como

$$B = K^{-T}K^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} \quad (34)$$

Note que a matriz  $B$  é simétrica e os termos repetidos ( $B_{12}, B_{13}, B_{23}$ ) podem ser omitidos com a definição de um vetor  $\mathbf{b}$  de 6 termos como

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T \quad (35)$$

Considerando que a  $i$ -ésima coluna da matriz  $H$  é denotada por  $\mathbf{h}_i$ , temos que  $H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3]$  e  $\mathbf{h}_i = [h_{i1} \ h_{i2} \ h_{i3}]^T$ . Fazendo referência às Equações 31 e 32,  $\mathbf{h}_i^T B \mathbf{h}_j$  pode ser escrito como

$$\begin{aligned} \mathbf{h}_i^T B \mathbf{h}_j &= [h_{i1} \ h_{i2} \ h_{i3}] \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \begin{bmatrix} h_{j1} \\ h_{j2} \\ h_{j3} \end{bmatrix} \\ &= [B_{11}h_{i1} + B_{12}h_{i2} + B_{13}h_{i3}, \ B_{12}h_{i1} + B_{22}h_{i2} + B_{23}h_{i3}, \ B_{13}h_{i1} + \\ &\quad B_{23}h_{i2} + B_{33}h_{i3}] \begin{bmatrix} h_{j1} \\ h_{j2} \\ h_{j3} \end{bmatrix} \\ &= [B_{11}(h_{i1}h_{j1}) + B_{12}(h_{i2}h_{j1} + h_{i1}h_{j2}) + B_{22}(h_{i2}h_{j2}) + \\ &\quad B_{13}(h_{i3}h_{j1} + h_{i1}h_{j3}) + B_{23}(h_{i3}h_{j2} + h_{i2}h_{j3}) + B_{33}(h_{i3}h_{j3})] \\ &= [h_{i1}h_{j1}, h_{i2}h_{j1} + h_{i1}h_{j2}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}] \begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix} \\ &= \mathbf{v}_{ij}^T \mathbf{b} \end{aligned} \quad (36)$$

A partir da Equação 36, as Equações 31 e 32 podem ser reescritas como

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0}_6 \quad (37)$$

Note que o sistema apresentado na Equação 37 é relativo a uma única homografia  $H$ , estimada a partir de apenas uma imagem, provendo duas equações lineares. Segundo [46], para a estimativa completa dos parâmetros intrínsecos, pelo menos 6 equações são necessárias para resolver  $\mathbf{b}$ , ou seja, pelo menos 3 imagens devem ser capturadas do padrão de calibração e 3 homografias devem ser estimadas. Considerando que  $n$  imagens foram capturadas, a Equação 37 pode ser reescrita como

$$V\mathbf{b} = \mathbf{0}_6 \quad (38)$$

em que  $V$  é uma matriz  $2n \times 6$ , contendo  $2n$  equações, relativas às  $n$  homografias estimadas. Este sistema linear homogêneo pode ser calculado pela Decomposição em Valores Singulares (SVD), em que  $\mathbf{b}$  corresponde ao autovetor da matriz  $V^T V$  correspondente ao menor autovalor de  $V^T V$ , conforme detalhado na seção A.6.

Com o vetor  $\mathbf{b}$  estimado, as entradas da matriz  $B$  são definidas e os parâmetros intrínsecos podem ser extraídos. Considerando  $B = \lambda K^{-T} K$ , para um fator de escala  $\lambda$  arbitrário, de acordo com [46], os parâmetros intrínsecos são obtidos a partir de  $B$  como segue

$$\begin{aligned} v_0 &= \frac{(B_{12}B_{13} - B_{11}B_{23})}{(B_{11}B_{22} - B_{12}^2)} \\ \lambda &= B_{33} - \frac{[B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]}{B_{11}} \\ \alpha &= \sqrt{\frac{\lambda}{B_{11}}} \\ \beta &= \sqrt{\frac{\lambda B_{11}}{(B_{11}B_{22} - B_{12}^2)}} \\ \gamma &= \frac{-B_{12}\alpha^2\beta}{\lambda} \\ u_0 &= \frac{\lambda v_0}{\alpha} - \frac{B_{13}\alpha^2}{\lambda} \end{aligned}$$

Note que os parâmetros intrínsecos apresentados, estão relacionados com as entradas da matriz  $K$  pela igualdade da Equação 33.

#### A.4.3 Estimação dos Parâmetros Extrínsecos

Com os parâmetros intrínsecos conhecidos (matriz  $K$ ), os parâmetros extrínsecos podem ser computados a partir da Equação 30 como segue.

$$\begin{aligned}\mathbf{r}_1 &= \frac{1}{\lambda} K^{-1} \mathbf{h}_1 \\ \mathbf{r}_2 &= \frac{1}{\lambda} K^{-1} \mathbf{h}_2 \\ \mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2 \\ \mathbf{t} &= \frac{1}{\lambda} K^{-1} \mathbf{h}_3\end{aligned}\tag{39}$$

em que  $\times$  denota o produto vetorial e  $\lambda = \|K^{-1} \mathbf{h}_1\| = \|K^{-1} \mathbf{h}_2\|$ , uma vez que  $\mathbf{r}_1$  e  $\mathbf{r}_2$  são ortonormais e  $\|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1$ .

#### A.5 Geometria Epipolar

Considere duas câmeras que capturam a mesma cena, cada uma com o seu centro de projeção, sendo estes não coincidentes. Cada par de imagem capturado por essas câmeras representa duas perspectivas diferentes de uma mesma cena estática. A geometria epipolar estabelece uma relação geométrica entre duas vistas capturadas nessas condições. Tradicionalmente, esta geometria é empregada na busca por pontos correspondentes em algoritmos de visão estéreo [13]. O processo de estimação da geometria epipolar também é conhecido como *calibração fraca* das câmeras, ou alternativamente, que as câmeras são fracamente calibradas.

A correspondência entre pontos empregando o conceito de geometria epipolar é ilustrado na Figura 8 (lado esquerdo). Dados dois pontos correspondentes,  $x$  e  $x'$ , referentes aos planos das imagens da câmera da esquerda e da direita, respectivamente, a relação entre estes pontos é dada pelo *plano epipolar*  $\pi$ , em que os centros de projeções das câmeras  $C$  e  $C'$ , o ponto  $X$  no espaço 3D e os pontos  $x$  e  $x'$  nos planos da imagem são coplanares.

No lado direito da Figura 8, observa-se que o ponto  $x$  no plano da imagem pode ser projetado para o espaço 3D por um raio formado por  $x$  e o centro de projeção  $C$ . Este raio é visualizado como uma linha  $l'$  no plano da segunda vista. Assim, o ponto  $X$  do espaço 3D que é projetado para o ponto  $x$  na primeira vista deve estar sobre este raio e também sobre a linha  $l'$  na segunda vista.

Em termos de algoritmos de visão estéreo destinados à computação de correspondências de pontos, o benefício proporcionado pela geometria epipolar é que, dado um ponto  $x$  referente à primeira vista, a busca por pontos correspondentes na segunda vista não precisa cobrir toda a imagem e restringe-se apenas a uma linha  $l'$ . Esta restrição também é conhecida na literatura como *restrição epipolar*.

As entidades geométricas relacionadas à geometria epipolar são listadas a seguir:

- *linha de base*: linha que passa pelos dois centros de projeções;
- *epipolo*: ponto de intersecção da linha de base com o plano da imagem;

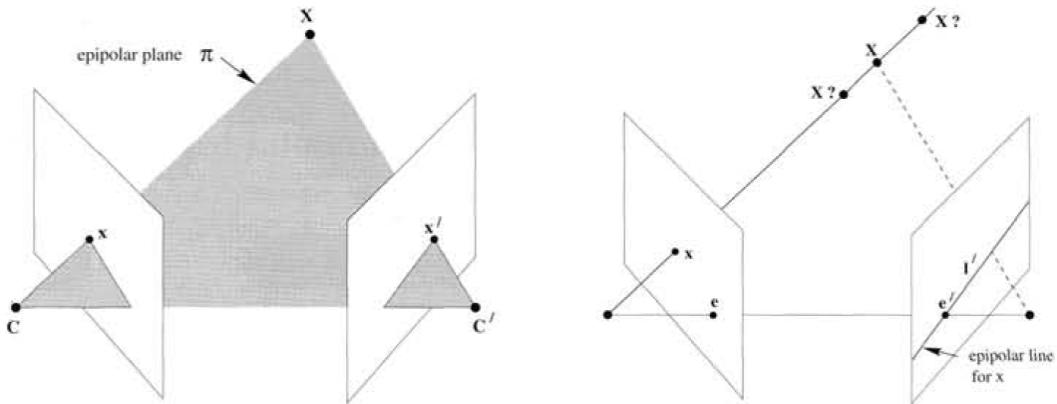


Figura 8: Correspondência entre pontos utilizando geometria epipolar. Fonte: [13].

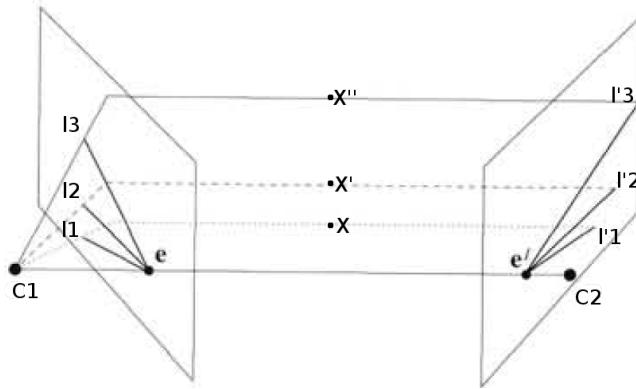


Figura 9: Planos epipolares definidos entre os centros de projeções e um conjunto de pontos no espaço 3D. Adaptada de [13]. Note que todas as linhas epipolares interceptam o epipolo.

- *plano epipolar*: plano definido pelo ponto 3D  $X$  e os centros de projeção  $C$  e  $C'$ . Note que, para cada ponto  $X$ , um plano epipolar é definido e que todas linhas epipolares interceptam o epipolo, como mostra a Figura 9. Além disso, um plano epipolar intercepta os planos das imagens da esquerda e da direita nas linhas epipolares e define a correspondência entre elas;
- *linha epipolar*: é a linha determinada pela intersecção do plano da imagem com o plano epipolar.

#### A.5.1 Matriz Fundamental

A representação algébrica da geometria epipolar pode ser obtida a partir da *matriz fundamental*  $F$ , que representa um mapeamento projetivo de pontos de uma imagem para linhas epipolares de outra [13], conforme descrição a seguir. Dado um par de imagens, cada ponto  $x$  da primeira imagem possui uma linha epipolar  $l'$  correspondente na segunda imagem e um ponto  $x'$ , que corresponde a  $x$ , está sobre a linha  $l'$ . Assim, existe um mapeamento  $x \mapsto l'$  de um ponto de uma imagem para sua linha epipolar correspondente em outra.

Com base em [13], as principais propriedades da matriz fundamental são listadas a seguir:

- $F$  é uma matriz  $3 \times 3$ , com 7 graus de liberdade e posto 2.
- *correspondência entre pontos* - se  $x$  e  $x'$  são pontos correspondentes, então  $x'^T F x = 0$ , para todos os pontos  $x \leftrightarrow x'$  correspondentes. No que  $l' = Fx$ .
- *transposição*: se  $F$  é uma matriz fundamental de um par de câmeras  $(P, P')$ , então  $F^T$  é a matriz fundamental para as câmeras  $(P', P)$ .
- *linhas epipolares*: a correspondência entre um ponto  $x$  da primeira imagem e a linha epipolar da segunda imagem é definida por  $l' = Fx$ . Similarmente,  $l = F^T x'$  define a correspondência entre um ponto  $x'$  da segunda imagem com a linha epipolar  $l$  da primeira imagem.
- *epipolo*: para qualquer ponto  $x$  da primeira imagem (exceto  $e$ ) a linha epipolar  $l' = Fx$  na segunda imagem contém  $e'$  e  $e'^T(Fx) = (e'^T F)x = 0$  para qualquer  $x$ . Assim,  $e'^T F = 0$  e  $e'$  é calculado pelo espaço nulo à esquerda de  $F$ . Similarmente,  $Fe = 0$  e  $e$  é computado pelo espaço nulo à direita de  $F$ .

**Computação da Matriz Fundamental** Dado um par de pontos correspondentes  $q \leftrightarrow q'$  em duas imagens, a matriz fundamental  $F$  é definida como

$$q' F q = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \quad (40)$$

que resulta

$$x' x f_{11} + x' y f_{12} + x' f_{13} + y' x f_{21} + y' y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0 \quad (41)$$

Considerando o vetor com 9 termos  $\mathbf{f} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^T$  construído a partir das entradas de  $F$ , a Equação 41 pode ser escrita como

$$(x' x, x' y, x', y' x, y' y, y', x, y, 1)\mathbf{f} = 0 \quad (42)$$

A Equação 42 foi definida para um par de pontos correspondentes. Considerando  $n \geq 7$  pares de pontos correspondentes  $q_i \leftrightarrow q'_i$ , um sistema linear de equações cuja a solução determina  $\mathbf{f}$  pode ser obtido como

$$A\mathbf{f} = \begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ x'_2 x_2 & x'_2 y_2 & x'_2 & y'_2 x_2 & y'_2 y_2 & y'_2 & x_2 & y_2 & 1 \\ \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = 0 \quad (43)$$

em que  $A$  é uma matriz  $n \times 9$ .

Segundo [13], se o posto de  $A$  é exatamente 8, a solução do sistema linear é gerada pelo espaço nulo de  $A$ . Se a correspondência dos pontos não é exata (a correspondência dos pontos foi estimada por um algoritmo suscetível a erro, por exemplo), a solução exata para o sistema  $A\mathbf{f} = 0$  é inexistente e uma solução aproximada deve ser calculada. Uma solução aproximada para o sistema linear homogêneo  $A\mathbf{f} = 0$  pode ser obtida pela Decomposição em Valores Singulares (SVD), em que  $\mathbf{f}$  corresponde ao autovetor da matriz  $A^T A$  relativo ao menor autovalor de  $A^T A$ , conforme detalhado na seção A.6.

**Restrição de Singularidade** Conforme as propriedades da matriz fundamental descritas anteriormente,  $F$  é singular, uma vez que tem posto 2 e os epipolos  $e'$  e  $e$  são computados pelos espaços nulos à esquerda e à direita de  $F$ , respectivamente. Lembrando que, as linhas epipolares são coincidentes com os epipolos, conforme ilustra a Figura 9. A matriz fundamental computada pela Equação 43 nem sempre é singular e pode fazer com que a linhas epipolares não sejam coincidentes [13].

Para resolver esse problema, segundo [13], uma restrição de singularidade deve ser imposta, substituindo  $F$  pela matriz  $F'$ , sendo que  $F'$  minimiza  $\|F - F'\|$  e possui a restrição de que o determinante de  $F'$  é igual 0. A matriz  $F'$  pode ser computada utilizando SVD (seção A.6) como segue. A matriz  $F$  pode ser decomposta em  $F = UDV^T$ , em que  $D = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$ , que satisfaz  $\sigma_1 \geq \sigma_2 \geq \sigma_3$  e  $F'$  é calculado por  $F' = U\text{diag}(\sigma_1, \sigma_2, 0)V^T$  que minimiza  $\|F - F'\|$ .

**Algoritmo dos 8 Pontos** O algoritmo dos 8 pontos [13] é um método empregado para computar a matriz fundamental utilizando pelo menos 8 pontos correspondentes ( $n \geq 8$ ), coordenadas normalizadas e empregando a restrição de singularidade. Dado um conjunto de pontos correspondentes  $q_i \leftrightarrow q'_i$ , o algoritmo dos 8 pontos que computa  $F$  e satisfaz  $q'_i F q_i = 0$  é sumarizado a seguir:

- (i) normalizar as coordenadas dos pontos  $q_i$  e  $q'_i$ , obtendo  $\hat{q}_i = Tq_i$  e  $\hat{q}'_i = Tq'_i$ , em que  $T$  consiste em uma transformação de normalização composta por uma translação e escala. O processo de normalização é detalhado na seção A.3.2.
- (ii) computar a matriz fundamental  $\hat{F}$  a partir dos pontos correspondentes  $\hat{q}_i \leftrightarrow \hat{q}'_i$  utilizando a Equação 43.
- (iii) calcular a matriz  $\hat{F}'$  a partir de  $\hat{F}$ , sendo que  $\hat{F}'$  respeita a restrição de singularidade, ou seja,  $\hat{F}'$  minimiza  $\|\hat{F} - \hat{F}'\|$  e possui a restrição de que  $\det(\hat{F}') = 0$ .
- (iv) obter a matriz fundamental  $F$  relativa aos pontos correspondentes  $q_i \leftrightarrow q'_i$  a partir de uma desnormalização  $F = T'^T \hat{F}' T$ , como detalhado na seção A.3.2.

### A.5.2 Matriz Essencial

Na seção A.1 foi descrito o modelo de câmera estenopeica e, pela Equação 7, a matriz de projeção da câmera  $P$  pode ser decomposta em

$$P = K \begin{bmatrix} R & | & \mathbf{t} \end{bmatrix} \quad (44)$$

em que  $K$  representa os parâmetros intrínsecos da câmera e  $R$  e  $\mathbf{t} = -R\tilde{C}$  os parâmetros extrínsecos.

A partir da Equação 44, a projeção de um ponto  $X$  do espaço 3D para um ponto  $x$  no plano da imagem é dada por  $x = PX$ . Se a matriz  $K$  é conhecida, uma normalização das coordenadas dos pontos do plano da imagem pode ser obtida como

$$\hat{x} = K^{-1}x = \begin{bmatrix} R & | & \mathbf{t} \end{bmatrix} X \quad (45)$$

em que  $\hat{x}$  é um ponto da imagem expresso em coordenadas normalizadas.

A matriz essencial  $E$ , com dimensão  $3 \times 3$ , é uma especialização da matriz fundamental para o caso das coordenadas das imagens normalizadas [13], que pode ser definida por

$$\hat{x}'^T E \hat{x} = 0 \quad (46)$$

em que  $\hat{x}$  e  $\hat{x}'$  são pontos correspondentes. Assumindo que há duas câmeras,  $P$  e  $P'$ , relacionadas aos pontos correspondentes  $x$  e  $x'$ , respectivamente, pela Equação 45, a Equação 46 pode ser reescrita como

$$x'^T K'^{-T} E K^{-1} x = 0 \quad (47)$$

em que a matriz  $K$  e  $K'$  representam os parâmetros intrínsecos das câmeras  $P$  e  $P'$ , respectivamente, e  $K'^{-T} = (K'^{-1})^T$ . Assim, comparando a Equação 47 com  $x'^T F x = 0$  da matriz fundamental, a relação entre a matriz essencial e a matriz fundamental pode ser escrita como

$$K'^{-T} E K^{-1} = F \quad (48)$$

que pode ser reformulada como

$$E = K'^T F K \quad (49)$$

### A.5.3 Triangulação

Dado um par de câmeras  $P$  e  $P'$  e um par de pontos correspondentes  $x \leftrightarrow x'$ , a triangulação consiste na computação de um ponto  $X$  do espaço 3D, em que  $X$  projeta o par de pontos correspondente nas imagens, respeitando a restrição epipolar  $x'^T F x = 0$ . O processo de triangulação é ilustrado na Figura 10. A restrição epipolar denota que existem dois raios sobre o plano epipolar que passam pelos pontos correspondentes e pelos centros de projeção de cada câmera, intersec-

tando no ponto  $X$  no espaço 3D. Além disso, o ponto  $x$  está sobre a linha epipolar  $F^T x'$  e  $x'$  sobre  $Fx$ .

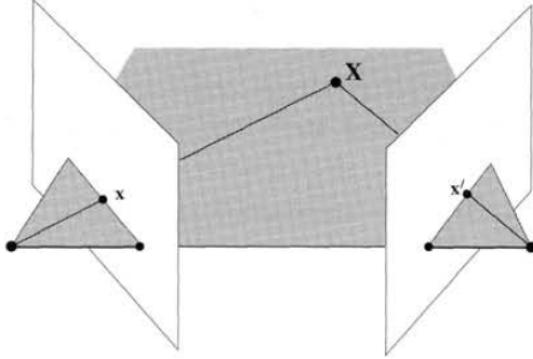


Figura 10: Triangulação. Fonte: [13].

Entretanto, as coordenadas dos pontos correspondentes  $x$  e  $x'$  podem conter algum tipo de ruído, gerado por erros de estimativa, por exemplo, fazendo com que o ponto de intersecção  $X$ , relativo aos raios que projetam  $x$  e  $x'$ , não possa ser estabelecido, ou seja,  $X$  pode não satisfazer ambas as equações  $x = PX$  e  $x' = P'X$ , como mostra a Figura 11 (a). Além disso, se o ponto de intersecção  $X$  não é determinado, os pontos  $x$  e  $x'$  não satisfazem a restrição epipolar  $x'^T F x = 0$ , como ilustra a Figura 11 (b).

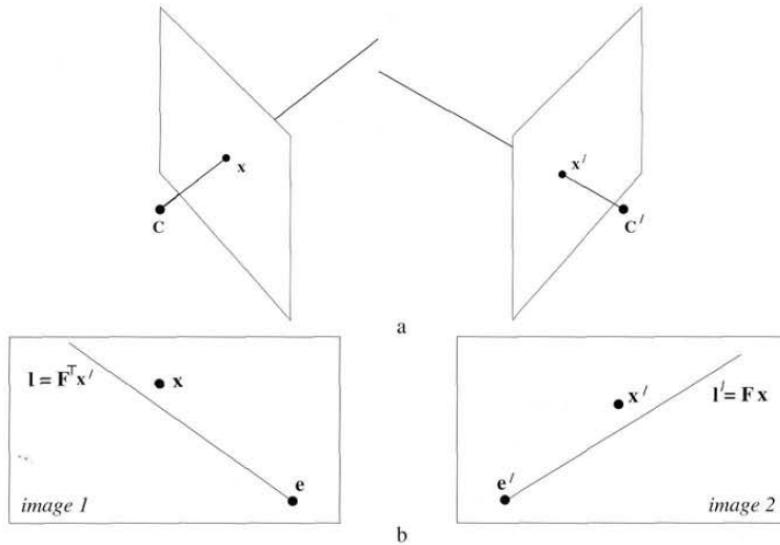


Figura 11: Triangulação errônea decorrente de erro no cálculo das coordenadas dos pontos  $x$  e  $x'$ . Fonte: [13]. (a) os raios que passam por  $x$  e  $x'$  não intersectam em um ponto  $X$  no espaço 3D; (b) os pontos  $x$  e  $x'$  não respeitam a restrição epipolar e não estão sobre as linhas epipolares.

Com base nessas premissas, uma solução aproximada para  $X$  deve ser estimada. Um método linear de triangulação que estima  $X$  é apresentado em [13] e descrito a seguir. Considerando

somente a primeira câmera, de acordo com o modelo de câmera estenopeica descrito na seção A.1, tem-se que

$$\lambda x = PX \quad (50)$$

em que  $\lambda$  é um fator de escala homogêneo e  $x = (x_1, x_2, 1)^T$ . A Equação 50 pode ser expressa em termos do produto vetorial como

$$x \times (\lambda x) = x \times (PX) \quad (51)$$

que pode ser reformulada como

$$x \times (PX) = 0 \quad (52)$$

Escrevendo a  $i$ -ésima linha da matriz  $P$  como  $\mathbf{p}_i^T$ , tem-se que

$$x \times \begin{bmatrix} \mathbf{p}_1^T X \\ \mathbf{p}_2^T X \\ \mathbf{p}_3^T X \end{bmatrix} = 0 \quad (53)$$

produzindo três equações

$$\begin{aligned} (\mathbf{p}_1^T X) - x_1(\mathbf{p}_3^T X) &= 0 \\ x_2(\mathbf{p}_3^T X) - (\mathbf{p}_2^T X) &= 0 \\ x_1(\mathbf{p}_2^T X) - x_2(\mathbf{p}_1^T X) &= 0 \end{aligned} \quad (54)$$

Note que, das três equações obtidas pelo cálculo do produto vetorial, somente duas são linearmente independentes, uma vez que a terceira linha pode ser obtida pela soma de  $-x_2$  vezes a primeira linha com  $-x_1$  vezes a segunda linha. Assim, um sistema linear  $T_1 X = 0$ , pode ser definido como

$$T_1 X = \begin{bmatrix} \mathbf{p}_1^T - x_1 \mathbf{p}_3^T \\ x_2 \mathbf{p}_3^T - \mathbf{p}_2^T \end{bmatrix} X = 0 \quad (55)$$

que pode ser reescrito como

$$\begin{bmatrix} x_1 \mathbf{p}_3^T - \mathbf{p}_1^T \\ x_2 \mathbf{p}_3^T - \mathbf{p}_2^T \end{bmatrix} X = 0 \quad (56)$$

em que a solução deste encontra  $X$  para  $x = PX$ , ou seja, para a primeira câmera. Similarmente, um sistema linear  $T_2 X = 0$  para a segunda câmera pode ser obtido, em que  $X$  satisfaz  $x' = P'X$  como

$$T_2 X = \begin{bmatrix} x'_1 \mathbf{p}'_3^T - \mathbf{p}'_1^T \\ x'_2 \mathbf{p}'_3^T - \mathbf{p}'_2^T \end{bmatrix} X = 0 \quad (57)$$

A partir das Equações 56 e 57, um sistema linear  $TX = 0$  pode ser definido a partir da

combinação das equações de ambas as câmeras,  $x = PX$  e  $x' = P'X$ , como

$$TX = \begin{bmatrix} x_1 \mathbf{p}_3^T - \mathbf{p}_1^T \\ x_2 \mathbf{p}_3^T - \mathbf{p}_2^T \\ x'_1 \mathbf{p}'_3^T - \mathbf{p}'_1^T \\ x'_2 \mathbf{p}'_3^T - \mathbf{p}'_2^T \end{bmatrix} X = 0 \quad (58)$$

tal que a solução do sistema linear deve considerar duas situações. Na primeira, se os pontos  $x$  e  $x'$  respeitam a restrição epipolar  $x'^T F x = 0$ , então existe um espaço nulo de  $T$  que provê a solução exata de  $X$ . Na segunda hipótese, a restrição epipolar não é respeitada e uma solução aproximada deve ser computada. Uma solução aproximada para o sistema linear homogêneo  $TX = 0$  pode ser obtida pela Decomposição em Valores Singulares (SVD), em que  $X$  é o autovetor da matriz  $T^T T$  correspondente ao menor autovalor de  $T^T T$ , conforme detalhado na seção A.6.

#### A.5.4 Transferência de Pontos Usando Matrizes Fundamentais

Dadas três vistas de uma mesma cena, capturadas por três câmeras, sendo que os centros de projeções das câmeras são não coincidentes, e um conjunto de pares de pontos correspondentes  $x \leftrightarrow x'$  entre a primeira e a segunda vista, a transferência de pontos consiste em determinar a posição do ponto  $x''$  correspondente aos pontos  $x$  e  $x'$ , na terceira vista.

O problema de transferência de pontos pode ser resolvido utilizando matrizes fundamentais que relacionam as três vistas, como ilustra a Figura 12. Dadas três matrizes fundamentais,  $F_{21}$ ,  $F_{31}$  e  $F_{32}$ , em que  $x'^T F_{21} x = 0$ ,  $x''^T F_{31} x = 0$  e  $x''^T F_{32} x' = 0$ , o ponto  $x''$  da terceira vista, correspondente a  $x$  da primeira vista, deve estar sobre a linha epipolar  $F_{31}x$  e, similarmente, o ponto  $x''$  da terceira vista, correspondente a  $x'$  da segunda vista, deve estar sobre a linha  $F_{32}x$ . Assim, segundo [13], o ponto  $x''$  corresponde à intersecção das duas linhas epipolares na terceira vista, de modo que

$$x'' = (F_{31}x) \times (F_{32}x') \quad (59)$$

em que  $\times$  denota o produto vetorial.

Um problema inerente à transferência de pontos empregando matrizes fundamentais é que, quando as linhas epipolares  $F_{31}x$  e  $F_{32}x$  são colineares, o ponto  $x''$  não pode ser calculado. Isso ocorre quando o ponto 3D  $X$ , que projeta  $x$ ,  $x'$  e  $x''$  está sobre o plano trifocal (plano formado pelos três centros de projeções das câmeras). O pior caso é quando os três centros de projeções são colineares, fazendo com que a transferência de pontos falhe para todos os pontos  $x''$  da terceira vista [13]. Este problema pode ser evitado por meio de um posicionamento das câmeras de forma que  $X$  não esteja sobre o plano trifocal.

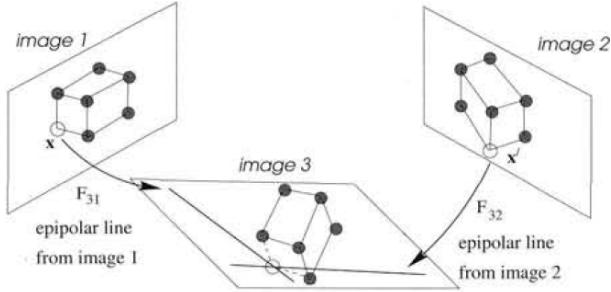


Figura 12: Transferência de pontos empregando matrizes fundamentais. Fonte: [13].

## A.6 Decomposição em Valores Singulares

Esta seção está dividida como segue. A subseção A.6.1 descreve o processo de decomposição de uma matriz em valores singulares, a subseção A.6.2 apresenta a relação entre os valores singulares e autovetores e, por fim, a subseção A.6.3 descreve a solução de mínimos quadros de sistemas homogêneos empregando a decomposição em valores singulares.

### A.6.1 Descomposição

A Decomposição em Valores Singulares (SVD) é um método de decomposição de matrizes tradicionalmente empregado para resolver sistemas de equações sobredeterminados, ou seja, em que o número de equações é maior que o número de incógnitas [13].

Dada uma matriz  $A$  com dimensão  $m \times n$ , em que  $m \geq n$ , a SVD da matriz  $A$  é uma fatoração como

$$A = UDV^T \quad (60)$$

em que  $D$  é uma matriz diagonal  $n \times n$  com entradas não-negativas e  $U$  e  $V$  são matrizes ortogonais, como dimensões  $m \times n$  e  $n \times n$ , respectivamente. Assume-se que a decomposição é computada de forma que os elementos da diagonal da matriz  $D$  estejam em ordem decrescente, ou seja

$$D = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix},$$

em que  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ .

Como  $U$  e  $V$  são matrizes ortogonais, as seguintes propriedades podem ser consideradas [12]

$$\begin{aligned} U^T U &= I_{n \times n} \\ V^T V &= I_{n \times n} \\ VV^T &= I_{n \times n} \\ \|Ux\| &= \|x\| \\ \|Vx\| &= \|x\| \end{aligned} \tag{61}$$

### A.6.2 Relação entre Valores Singulares e Autovalores

Antes de apresentar a relação entre os valores singulares e autovalores, o problema de diagonalização ortogonal é descrito. Para [12], dada uma matriz  $Q$  de dimensão  $n \times n$ ,  $Q$  é diagonalizável, se  $Q$  possui um conjunto ortonormal de  $n$  autovetores e  $Q$  é uma matriz simétrica. A diagonalização de  $Q$  pode ser escrita como

$$P^T Q P = D \tag{62}$$

em que  $P = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n]^T$  e  $\mathbf{p}_i$  é o  $i$ -ésimo autovetor de  $Q$  e  $D$  é uma matriz diagonal contendo os autovalores associados a  $\mathbf{p}_i$  em sua diagonal. A Equação 62 pode ser reescrita como

$$Q = P D P^T \tag{63}$$

Considerando que a SVD da matriz  $A$  é  $A = UDV^T$ , segue que  $A^T A = VDU^T UDV^T = VD^2V^T$ . Assim, as entradas da diagonal de  $D^2$  são os autovalores da matriz  $A^T A$  e as colunas de  $V$  são os autovetores de  $A^T A$ , ou seja, a relação entre valores singulares e autovalores é que os valores singulares de  $A$  são as raízes quadradas dos autovalores de  $A^T A$ .

### A.6.3 Solução de Mínimos Quadrados de Sistemas Homogêneos

Sistemas homogêneos, tal como  $Ax = 0$ , surgem em diferentes problemas, como na estimativa da matriz de homografia A.3 e calibração da câmera A.4.1. Considerando que o sistema linear de equações é sobredeterminado, ou seja,  $A$  possui dimensão  $m \times n$  e  $m > n$ , segundo [13], este tipo de sistema geralmente não tem uma solução exata e uma solução aproximada é desejada, buscando encontrar um vetor  $x$  que minimize  $\|Ax\|$ . O vetor  $x$  é chamado de uma solução de mínimos quadrados, no qual pode ser computado pela SVD como segue.

Inicialmente, uma restrição para  $\|x\|$  deve ser imposta, uma vez que a solução nula  $x = 0$  não é a desejada. Se  $x$  é uma solução para o sistema de equações, então  $kx$  também o é, para uma constante  $k$ . Uma restrição razoável é definir que  $\|x\| = 1$  [13]. Fatorando  $A$  como  $A = UDV^T$ , o problema é minimizar  $\|UDV^T x\|$  para  $\|x\| = 1$ . A partir das propriedades das matrizes ortonormais  $U$  e  $V$ , detalhadas na Equação 61,  $\|UDV^T x\| = \|DV^T x\|$  e  $\|V^T x\| = \|x\|$ .

Assim, com essas simplificações,  $\|DV^T x\|$  deve ser minimizado para a condição  $\|V^T x\| = 1$ ,

que pode ser reescrito como  $\|Dy\|$  restrito a  $\|y\| = 1$ , em que  $y = V^T x$ .  $Dy$  pode ser escrito como

$$Dy = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \sigma_1 y_1 \\ \sigma_2 y_2 \\ \vdots \\ \sigma_n y_n \end{bmatrix} \quad (64)$$

Considerando que  $D$  é uma matriz diagonal e suas entradas na diagonal estão em ordem decrescente ( $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ ), os termos do vetor  $y$  estão sendo multiplicados pelos valores singulares e o último elemento do vetor  $Dy$  contém o menor valor singular. Assim, como o objetivo é minimizar  $\|Dy\|$ , a solução desse problema é especificar  $y = [0, 0, \dots, 1]^T$ , no qual respeita a restrição  $\|y\| = 1$  e é relativo ao menor valor singular. Como definido anteriormente,  $y = V^T x$  e a solução do sistema linear homogêneo  $Ax = 0$  é dada por  $x = Vy$ , que pode ser escrito como

$$x = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ V_{21} & V_{2,2} & \cdots & V_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ V_{n,1} & V_{n,2} & \cdots & V_{n,n} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} V_{1n} \\ V_{2n} \\ \vdots \\ V_{n,n} \end{bmatrix} \quad (65)$$

em que  $x$  corresponde à última coluna da matriz  $V$ .

Resumindo, dado um sistema  $Ax = 0$ , em que a dimensão da matriz  $A$  é  $m \times n$ , com  $m > n$ , a solução de  $x$  é dada pela última coluna de  $V$ , em que  $A = UDV^T$ . Alternativamente, conforme subseção A.6.3, a solução de  $x$  pode ser descrita como o autovetor da matriz  $A^T A$  correspondente ao menor autovalor.

## B Conhecimentos Adicionais

Esta seção está dividida em três partes. Primeiramente, os trabalhos que utilizam mapas de profundidade para VBR são descritos na subseção B.1. Posteriormente, as abordagens de VBR baseadas nas técnicas *light field* e *lumigraph* são apresentadas na subseção B.2. Por fim, a técnica de varredura de planos é descrita na subseção B.3, bem como os trabalhos que empregam esta técnica para VBR.

### B.1 Mapa de Profundidade

A extração de informações 3D de uma cena a partir de imagens 2D pode ser obtida por uma triangulação (Apêndice A.5.3), em que a posição 3D (profundidade) relativa a um par de pontos correspondentes<sup>1</sup> entre duas imagens é calculada. O *mapa de profundidade*, também conhecido como *imagem de profundidade*, armazena os valores de profundidade estimados para cada ponto de uma imagem 2D, representando a estrutura 3D da cena. A Figura 13 (c) mostra o mapa de profundidade computado por uma triangulação dos pontos correspondentes e não ocluídos das imagens ilustradas na Figura 13 (a) e (b). As áreas mais claras do mapa de profundidade representam as superfícies dos objetos mais próximas do plano da imagem, enquanto as áreas escuras, as superfícies mais distantes.

A profundidade calculada pela triangulação requer que a correspondência de pontos entre um par de imagens seja estimada. Tradicionalmente, a busca por pontos correspondentes é realizada levando em consideração a restrição epipolar (Apêndice A.5), a qual estabelece que, dado um ponto  $x$  da primeira vista, a busca por um ponto  $x'$  correspondente na segunda vista não precisa cobrir toda a imagem e restringe-se apenas a uma linha epipolar. A busca pode ser aprimorada empregando imagens retificadas. A retificação de um par de imagens transforma cada plano das imagens de forma que as linhas epipolares se tornem colineares e paralelas horizontalmente [6], permitindo que a busca seja realizada ao longo das linhas horizontais das imagens retificadas, aumentando o desempenho computacional [24].

Na busca por pontos correspondentes, empregando um par de imagens retificadas, a medida de disparidade  $d$ , que é a diferença entre os pontos correspondentes das imagens da esquerda e direita, pode ser usada para obter o valor de profundidade  $z$ , pela relação definida em [29] como

$$d = bf \frac{1}{z} \quad (66)$$

em que  $b$  é a distância entre os centros de projeção das câmeras (*baseline*) e  $f$  é a distância focal.

Um algoritmo básico para estimativa de um mapa de profundidade entre um par de imagens retificadas é apresentado em [25, 29] e descrito a seguir. Considerando o pixel  $p_1$  da primeira imagem  $I_1$  (imagem de referência), a busca pelo ponto correspondente  $p_2$  na segunda imagem

---

<sup>1</sup>Dados dois pontos  $x$  e  $x'$ , eles são ditos correspondentes se existir um ponto  $X$  no espaço 3D, o qual é projetado para o ponto  $x$ , em uma primeira vista, e  $x'$  em uma segunda vista [13].

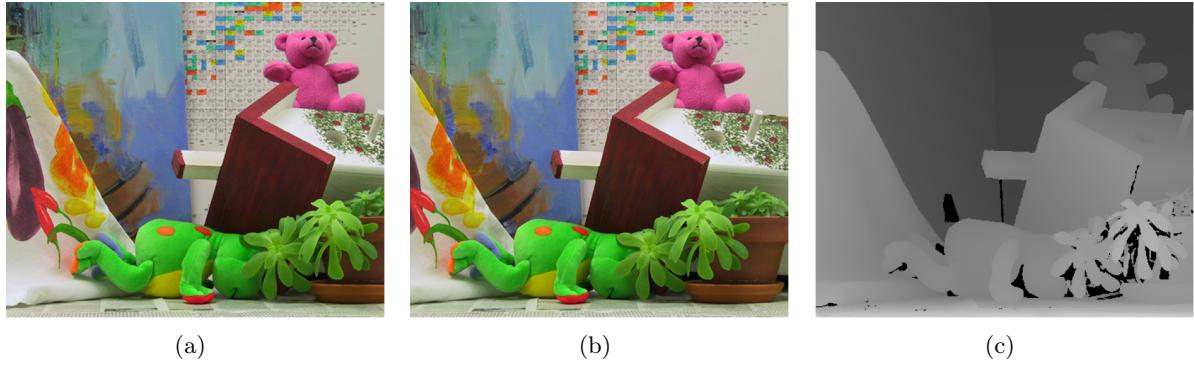


Figura 13: Mapa de profundidade (imagem (c)) gerado por uma triangulação dos pontos correspondentes das imagens (a) e (b). Fonte: [25].

$I_2$  é realizada ao longo da linha horizontal, em que a similaridade entre  $p_1$  e  $p_2$  é medida pela comparação entre blocos  $W$  que cercam os pixels, como ilustra a Figura 14. A similaridade (correlação) dos blocos pode ser medida pela soma das diferenças absolutas e a disparidade  $d$  de um pixel na posição  $(x, y)$  na imagem  $I_1$  pode ser calculada como

$$d(x, y) = \arg \min_{0 \leq \tilde{d} \leq d_{max}} \sum_{(i,j) \in W} |I_1(x + i, y + j) - I_2(x + i - \tilde{d}, y + j)| \quad (67)$$

em que  $d_{max}$  representa a disparidade máxima, limitando a busca,  $\tilde{d}$  é a disparidade candidata para um determinado par de blocos,  $i$  e  $j$  definem as coordenadas dos pixels pertencentes aos blocos e  $d(x, y)$  é a disparidade computada para o pixel na posição  $(x, y)$  da imagem  $I_1$ , com relação à menor diferença de similaridade entre os blocos comparados. A partir da disparidade  $d$  computada, a profundidade pode ser obtida pela Equação 66.

Com base em [37], os principais problemas inerentes à estimativa de mapas de profundidade são listados a seguir:

- a) *ruído*: variações na iluminação e ruídos presentes nas imagens produzem diferentes intensidades de cores entre as vistas, fazendo com que a correspondência dos pontos estimada seja incerta.
- b) *regiões sem textura*: em regiões com cor constante, a medida de similaridade calculada pode ser a mesma para todos os valores de disparidade estimados, resultando em valores de disparidade incorretos.
- c) *descontinuidades de profundidade*: no mapa de profundidade, descontinuidades abruptas dos valores de profundidade estão tipicamente associadas com as bordas dos objetos. A determinação da medida de correlação de blocos próximos às descontinuidades possibilita que um bloco contenha porções de objetos com duas profundidades diferentes (*foreground*

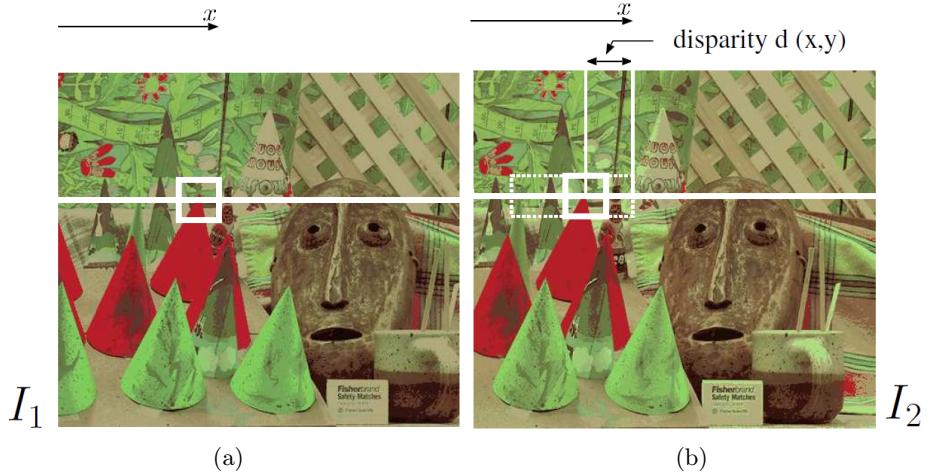


Figura 14: Estimação da disparidade pela verificação de similaridade entre blocos (regiões destacadas na cor branca) da imagem de referência  $I_1$  e da imagem  $I_2$  ao longo da linha epipolar. Fonte: [25].

e *background*), fazendo com que a medida de similaridade seja computada de forma incorreta [1, 29].

d) *occlusão*: não é possível determinar a correspondência de pontos de regiões ocluídas.

Um dos primeiros trabalhos a empregar mapas de profundidade para VBR foi apresentado por Kanade et al. [16]. A estrutura 3D da cena é extraída com a estimação de mapas de profundidade para cada vista capturada. A estimação é realizada com base na técnica descrita em [29], em que a correspondência de pontos é determinada com relação a múltiplos pares de imagens e levando em consideração a profundida inversa  $\varsigma$ , obtida com reformulação da Equação 66 como

$$\frac{d}{bf} = \frac{1}{z} = \varsigma \quad (68)$$

Na estimação de mapas de profundidade, considerando um par de câmeras, em que a distância  $b$  entre os centros de projeção é curta, os valores de profundidade não são precisos e uma distância maior é desejada [29]. Em contraste, quando a distância entre os centros de projeção é grande, a região de busca pelo ponto correspondente deve ser maior (valor de  $d_{max}$  da Equação 67) e a disparidade estimada pode ser falsa [29]. A busca de pontos correspondentes, levando em consideração  $\varsigma$  da Equação 68, permite que a estimação dos pontos correspondentes seja realizada independentemente da distância entre os centros de projeção das câmeras, pois o valor de  $\varsigma$  é constante, uma vez que cada ponto tem somente uma profundidade  $z$ .

Assim, para cada possível valor de  $\varsigma$ , a correlação de blocos entre uma imagem de referência e as demais imagens é medida na computação do mapa de profundidade, tal como descrito na Equação 67, só que a posição dos blocos na segunda imagem não é determinada pela disparidade, mas sim em função de  $\varsigma$ . Dado um ponto na imagem de referência, para encontrar a posição na

segunda imagem correspondente à profundidade inversa  $\varsigma$ , o ponto de referência, juntamente com o valor de  $\varsigma$ , é convertido para um ponto 3D (este tipo de projeção é detalhado no Apêndice A.2) e, posteriormente, projetado para as outras imagens.

Com a informação 3D da cena extraída, a cena é reconstruída e a síntese de novas vistas da câmera virtual é realizada com a renderização da malha triangular obtida na reconstrução, levando em consideração o ponto de vista desejado e aplicando as imagens das câmeras de entrada como texturas sobre os polígonos. A partir da definição do ponto de vista da câmera virtual, imagens das câmeras mais próximas a este ponto de vista são selecionadas para a reconstrução do objeto. Para cada imagem selecionada, uma malha triangular é produzida a partir do mapa de profundidade, convertendo cada região  $2 \times 2$  do mapa de profundidade em dois triângulos, como ilustra a Figura 15. As coordenadas  $(x, y, z)$  de cada ponto são computadas utilizando as coordenadas da imagem e a profundidade, tal como descrito no Apêndice A.2. As coordenadas de textura  $(u, v)$  são definidas como mostra a Figura 15, para uma imagem de dimensão  $m \times n$ . Como múltiplas imagens são selecionadas para a reconstrução do objeto, cada uma relativa a um ponto de vista diferente, a síntese de uma nova vista da câmera virtual consiste em combinar as imagens renderizadas de cada reconstrução.

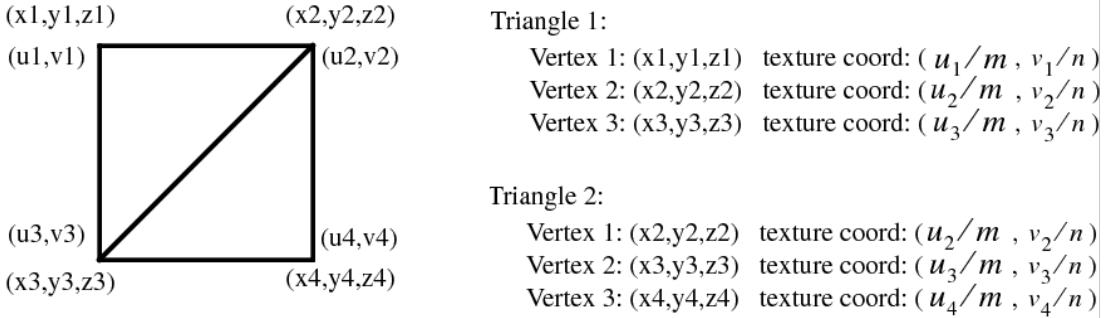


Figura 15: Definição das coordenadas de textura e vértices de triângulos a partir do mapa de profundidade. Fonte: [16].

A abordagem descrita por Kanade et al. [16], embora possibilite a captura e a renderização dinâmicas da cena e do mundo real, possui diversas desvantagens, as quais são citadas a seguir. Problemas na estimativa do mapa de profundidade, decorrentes de descontinuidades de profundidade presentes nas bordas dos objetos e regiões sem textura, onde a cor é constante, são tratados manualmente. A computação não é realizada em tempo real. Um grande número de câmeras é empregado para a captura da cena, sendo 51 câmeras arranjadas em uma estrutura com forma de uma abóbada. A qualidade das vistas renderizadas é pobre, contendo buracos e artefatos.

Zitnick et al. [47] apresentam um sistema de VBR baseado na reconstrução da cena 3D a partir de mapas de profundidade. Uma etapa de pré-processamento de dados é adotada, em que os mapas de profundidade são estimados e os dados (vídeos e informações de profundidade) são comprimidos e armazenados. Posteriormente, em tempo real, as vistas da câmera virtual são construídas a partir da renderização dos polígonos. As imagens das câmeras são usadas como

texturas para o modelo reconstruído.

A estimativa dos mapas de profundidade é baseada na segmentação de cores das imagens de entrada. O algoritmo é composto por 3 etapas principais, as quais são detalhadas a seguir. Na primeira etapa, cada imagem de entrada é independentemente segmentada. Assume-se que regiões de cores homogêneas são suscetíveis (candidatas) a ter valores de disparidade similares. Para criação dos segmentos, um filtro de suavização é inicialmente aplicado às imagens e, em seguida, os segmentos são computados pela comparação das cores de pixels vizinhos.

Após a segmentação das imagens, na segunda etapa, um valor único de disparidade é estabelecido para cada segmento  $s_{ij}$  de uma imagem  $I_i$  de entrada. Para isso, a probabilidade  $p_{ij}(d)$  de um segmento  $s_{ij}$  ter uma disparidade  $d$  é calculada como

$$p_{ij}(d) = \frac{\prod_{k \in N_i} m_{ijk}(d)}{\sum_{d'} \prod_{k \in N_i} m_{ijk}(d')} \quad (69)$$

em que  $m_{ijk}(d)$  é uma função que mede a similaridade entre os segmentos  $s_{ij}$  da imagem  $I_i$  e o segmento  $s_{kj}$  da imagem  $I_k$ , em relação à disparidade  $d$  e  $N_i$  são as imagens capturadas pelas câmeras à esquerda e à direita de uma câmera  $i$ . O denominador da Equação 69 representa a soma de todas as medidas de similaridade computadas para cada disparidade  $d'$  candidata, assegurando que  $\sum_d p_{ij}(d) = 1$ .

A função que mede a similaridade entre segmentos é baseada em um histograma computado. Para cada pixel  $x$  de um segmento  $s_{ij}$ , a projeção do seu pixel correspondente  $x'$  no segmento  $s_{kj}$  da imagem  $I_k$  é definida pela disparidade  $d$ . Então, um histograma  $h$  é computado a partir das cores dos pixels pela razão  $I_i(x)/I_k(x')$  e a medida de similaridade é obtida com a soma das três maiores frequências contíguas do histograma

$$m_{ijk}(d) = \max(h_{l-1} + h_l + h_{l+1}) \quad (70)$$

em que  $h_l$  é a  $l$ -ésima frequência no histograma.

Na terceira e na última etapas, os valores de disparidade são recalculados com base em segmentos e imagens vizinhas, levando em consideração as restrições de suavização e consistência. A restrição de suavização estabelece que, quando as cores de segmentos vizinhos são similares, os valores de disparidade dos segmentos também devem ser similares. Na segunda restrição, os valores de disparidade entre segmentos de imagens vizinhas são analisados. Se um segmento  $s_{ij}$  é projetado sobre o segmento  $s_{kj}$  em uma imagem vizinha, os valores de disparidade dos segmentos devem ser próximos.

Com a disparidade computada para cada ponto das imagens de entrada, os mapas de profundidade podem ser obtidos pela Equação 66 e a cena pode ser reconstruída. Para evitar artefatos decorrentes das descontinuidades de profundidade, as regiões de descontinuidade são detectadas com a análise dos valores de disparidade e, posteriormente, os triângulos entre as descontinuidades não são renderizados. Com base na técnica apresentada em [2], os valores de profundidade ao

longo das regiões de descontinuidades são recalculados por uma média dos valores de disparidade de pixels vizinhos e uma malha triangular é criada para a reconstrução dessas regiões. A vista final é computada pela combinação das imagens produzidas na renderização da cena e das regiões de descontinuidade.

A abordagem descrita tem como principais vantagens o uso de poucas câmeras (8 câmeras são empregadas) e as vistas renderizadas são de alta qualidade e de alta resolução ( $1024 \times 768$ ). A qualidade das vistas renderizadas está associada com a robustez do método de estimativa de mapas de profundidade usado, que trata oclusões de forma explícita e descontinuidades. Em contraste, uma desvantagem a ser destacada é o tempo de computação do método proposto, em que somente a parte de renderização é executada em tempo real.

Li et al. [21] descrevem um sistema de VBR empregando câmeras não-calibradas. O algoritmo de renderização consiste em projetar pontos correspondentes entre duas imagens de referência para a câmera virtual. Inicialmente, a correspondência de pontos entre as imagens de referência é computada pela estimativa da informação de disparidade, empregando um algoritmo baseado na segmentação de cores.

As imagens de referência são segmentadas e a disparidade inicial para cada segmento é computada. Os segmentos da primeira imagem são projetados para a segunda utilizando a restrição epipolar, em que os pontos  $x$  e  $x'$ , da primeira e segunda imagem, respectivamente, são correspondentes se,  $x'^T F x = 0$ . A matriz fundamental  $F$  é estimada (Apêndice A.5.1) empregando um conjunto de pontos putativamente correspondentes e a similaridade dos segmentos é medida pela relação entre a quantidade de pontos em cada segmento e o número de pares de pontos correspondentes calculados pela restrição epipolar. Posteriormente, os valores de disparidade são recalculados levando em consideração os segmentos vizinhos.

A posição da câmera virtual é definida entre duas câmeras reais de referência, em que a translação e a rotação relativas da primeira câmera para a câmera virtual são computadas. Como as câmeras não são calibradas, as matrizes de projeção das câmeras de referência são inicialmente extraídas. Segundo [21], a maioria das câmeras digitais possui uma distância focal entre 0,8 e 3 vezes a dimensão da imagem. A partir dessa premissa, assume-se que os parâmetros intrínsecos (Apêndice A.1.1) são idênticos para ambas as câmeras e estabelecidos por uma aproximação baseada nas dimensões das imagens como

$$K_1 = K_2 = \begin{bmatrix} (w+h)/2 & 0 & w/2 \\ 0 & (w+h)/2 & h/2 \\ 0 & 0 & 1 \end{bmatrix} \quad (71)$$

em que  $K_1$  e  $K_2$  são os parâmetros intrínsecos da primeira e segunda câmera de referência, respectivamente, e a dimensão das imagens é estabelecida por  $w \times h$ . Os parâmetros extrínsecos das câmeras de referência são extraídos a partir de uma decomposição da matriz essencial  $E$ , conforme detalhado em [13]. Como descrito no Apêndice A.5.2, a matriz essencial pode ser obtida pelos parâmetros intrínsecos definidos anteriormente, juntamente com a matriz fundamental  $F$ ,

usada para estabelecer a restrição epipolar durante a computação da disparidade.

Com o posicionamento da câmera virtual definido entre as duas câmeras reais e a informação de disparidade computada, as vistas da câmera virtual são renderizadas. Dado um par de pontos correspondentes  $x$  e  $x'$  das imagens de referência, os quais são definidos pela informação de disparidade, a projeção  $x$  e  $x'$  para o ponto  $x''$  na câmera virtual é computada pela transferência de pontos utilizando matrizes fundamentais, como detalhado no Apêndice A.5.4. A cor do pixel  $x''$  é estabelecida por uma interpolação linear, combinando as cores de  $x$  e  $x'$ .

A principal contribuição da abordagem de VBR descrita por Li et al. [21] é o método de extração dos parâmetros das câmeras, permitindo o posicionamento da câmera virtual e a renderização dos quadros utilizando câmeras não-calibradas. Entretanto, a definição dos parâmetros intrínsecos por uma aproximação pode gerar erros de projeção, afetando a qualidade das vistas renderizadas. Além disso, a computação do sistema proposto não é realizada em tempo real e a disparidade computada não leva em consideração as descontinuidades de profundidade.

Oh et al. [28] propõem uma abordagem para VBR usando mapas de profundidade para representar a informação 3D da cena e a técnica 3D *warping* [23] na renderização das vistas da câmera virtual. A técnica 3D *warping* consiste na projeção (*warping*) de uma imagem de referência para uma imagem destino, relativo ao ponto de vista desejado. Essa técnica requer, como dados de entrada, a imagem de referência, o mapa de profundidade associado com a imagem de referência e as matrizes de projeção de ambas as câmeras. A projeção definida por 3D *warping* pode ser calculada pela reconstrução dos pontos 3D relativos à imagem de referência, utilizando a informação de profundidade disponível e, posteriormente, a projeção destes pontos para a câmera virtual.

A câmera virtual é posicionada entre duas câmeras reais e a renderização de novas vistas é realizada pela projeção das duas imagens de referência para a câmera virtual, sendo que as cores dos pixels da vista renderizada é estabelecida pela combinação das duas imagens projetadas. Problemas como buracos e desocclusão, denominados artefatos, podem surgir na imagem destino [32]. Buracos aparecem devido à diferença entre a resolução de amostragem entre a imagem de referência e a destino. Desocclusão corresponde a uma superfície que não é visível pela câmera de referência, mas faz parte do campo de visão da câmera virtual.

Os artefatos são coloridos com as cores dos pixels que fazem fronteira com o artefato. Dois casos são considerados. Quando o artefato localiza-se inteiramente sobre o plano frontal (*foreground*), o artefato é preenchido (colorido) com regiões vizinhas, as quais fazem parte somente do plano frontal. No segundo caso, quando o artefato está sobre dois planos, o plano frontal e o plano de fundo (*background*), somente as regiões vizinhas ao artefato pertencente ao plano de fundo são utilizadas no preenchimento. Para determinar se um ponto está sobre o plano frontal ou plano de fundo, o valor de profundidade do ponto é analisado.

O uso de 3D *warping* para VBR permite a síntese de vistas da câmera virtual sem a necessidade da reconstrução explícita da cena 3D. Entretanto, as regiões de desocclusão podem ser grandes, devido à falta de amostras (imagens) durante a renderização e o método usado para preencher os

artefatos podem produzir vistas com baixa qualidade nessas regiões. Além disso, o sistema não é em tempo real, pois leva em consideração que o mapa de profundida é computado em uma fase de pré-processamento.

Yoon et al. [45] descrevem um método de VBR baseado na técnica *Layered Depth Image* (LDI) [34]. LDI é uma imagem com múltiplas camadas, em que as coordenadas  $(x, y)$  dos pixels armazenam múltiplos valores de cor e profundidade. O procedimento de geração de uma LDI é mostrado na Figura 16, em que são ilustradas duas câmeras de referência  $C_2$  e  $C_3$  e uma câmera LDI ( $C_1$ ) posicionada entre as câmeras de referência. Cada câmera possui a informação de cor (imagem capturada) e o mapa de profundidade estimado. A LDI é construída com relação ao ponto de vista da câmera  $C_1$  a partir da projeção (*warping*) das imagens das câmeras  $C_2$  e  $C_3$  para  $C_1$ . Durante a projeção, quando dois ou mais pixels são mapeados para a mesma coordenada  $(x, y)$ , os valores de profundidade são comparados. Se a diferença entre os valores de profundidade for maior que um determinado limiar, uma nova camada é criada e as informações de cor e profundidade são armazenadas (ponto  $b$ ); caso contrário, o pixel projetado é combinado na mesma camada (pontos  $c$  e  $d$ ).

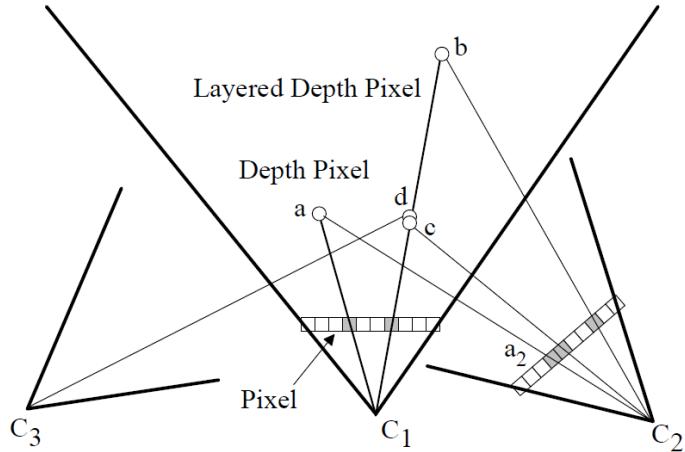


Figura 16: Construção de uma LDI a partir de múltiplas imagens. Fonte: [34].

Duas vantagens principais podem ser destacadas no uso de LDI para VBR. Uma LDI armazena a informação de cor e profundidade da cena de várias câmeras próximas, sendo que pontos em comum são combinados, reduzindo a quantidade de informação a ser processada e transmitida. Outro aspecto a ser considerado é que, em comparação com a técnica 3D *warping*, uma LDI armazena mais informação (amostras), minimizando os problemas de desoclução. Nota-se que, os pixels armazenados em camadas mais profundas e que são ocluídos em relação à câmera LDI (ponto  $b$  é ocluído por  $d$ ), podem ser usados para preencher áreas de desoclução na renderização de outros pontos de vistas. Em contraste, LDI requer a estimação de mapas de profundidade e, muitas vezes, essa estimação é um procedimento custoso, como descrito nos trabalhos acima, fazendo com que a computação não seja realizada em tempo real.

Wang et al. [39] desenvolveram um sistema VBR com tempo de computação significativo,

estimando os mapas de profundidade e renderizando as vistas da câmera virtual com uma taxa de 6,5 quadros por segundo, sem usar uma etapa de pré-processamento de dados. A renderização dos quadros e a estimativa da informação de profundidade foram implementadas empregando o modelo de programação e ambiente de desenvolvimento denominado *Compute Unified Device Architecture* (CUDA) [8], o qual permite que programas paralelos sejam escritos, usando uma extensão da linguagem C e executados em GPUs da NVIDIA.

A estimativa dos mapas de profundidade é realizada seguindo o mesmo princípio da Equação 67, mas usando blocos de tamanho variável. Assume-se que regiões com intensidades de cores similares, tem o mesmo valor de disparidade. A partir dessa premissa, blocos grandes são usados para regiões de cores constantes e blocos pequenos em regiões com diversas texturas. A medida de correlação usada é a soma das diferenças absolutas. Esse algoritmo é executado em paralelo, utilizando memória compartilhada entre múltiplas *threads*.

A renderização das vistas da câmera virtual consiste em reconstruir os pontos 3D das câmeras de referência a partir do mapa de profundidade estimado e projetar diretamente para a câmera virtual. A execução dessa tarefa também é realizada por múltiplas *threads*. Uma *thread* é alocada para cada pixel da imagem de referência. As *threads* são executadas em paralelo, sendo que cada uma realiza a tarefa de projeção de 1 pixel da imagem de referência para a imagem destino.

A abordagem descrita tem como principal vantagem o seu tempo de computação que, em comparação aos demais métodos baseados em mapas de profundidade descritos nesta subseção, não utiliza uma etapa de pré-processamento dos dados e consegue uma boa taxa de renderização. Em contraste, o método funciona apenas em placas gráficas da NVIDIA [27].

## B.2 Light Field e Lumigraph

*Light field* [20] e *lumigraph* [11] são técnicas que possibilitam a síntese de novas vistas de uma cena estática, com iluminação fixa, utilizando como dados de entrada somente múltiplas imagens, sem a necessidade da extração de informações geométricas da cena. O processo de renderização é baseado na parametrização dos raios que intersectam dois planos, o das câmeras *uv* e o focal *st*, como ilustra a Figura 17 (a). O plano *uv* é subdividido em diversas partes, criando uma representação discreta em forma de uma grade. Cada subdivisão do plano *uv* tem uma imagem (amostra) associada, a qual é capturada usando o plano *st* como plano da imagem (Figura 17 (b)). Observa-se na Figura 17 (c) que os planos *uv* e *st* são fixos e o campo de visão das câmeras durante a captura das amostras é relativo à movimentação da câmera do centro do plano *uv* em direção às bordas.

Para renderizar novas vistas, realiza-se o processo de reamostragem. Dada uma posição e direção do ponto de vista desejado, a intersecção dos raios que passam por cada pixel da nova vista com o plano focal e das câmeras é calculada. As coordenadas *uv* estabelecem a imagem a ser usada na reamostragem e as coordenadas *st* definem o pixel que será selecionado na amostra. Empregando câmeras calibradas (Apêndice A.4), em que os parâmetros intrínsecos e extrínsecos são conhecidos, a renderização das vistas pode ser obtida pela projeção dos pontos 2D da imagem

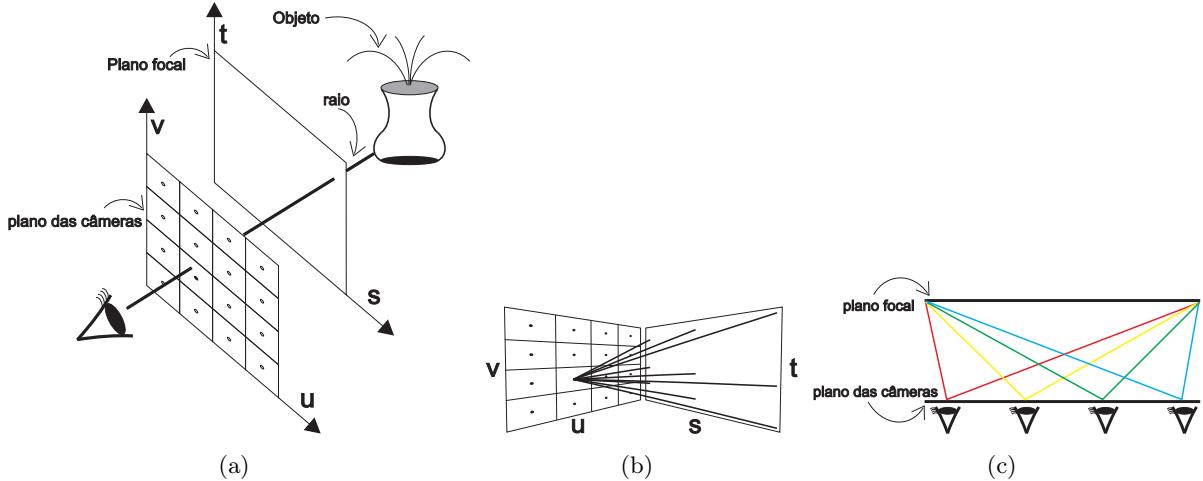


Figura 17: Parametrização dos raios que intersectam os planos das câmeras  $uv$  e o focal  $st$ .  
Fonte: [32].

para raios no espaço 3D, como descrito no Apêndice A.2.

O método de renderização descrito pode gerar vistas com artefatos, em que a cor de um pixel computada durante a reamostragem não corresponde à cor original do raio que intercepta o objeto na cena. Tradicionalmente, esse problema é tratado com o uso de uma interpolação, tal como a interpolação do vizinho mais próximo [10] ou adição de um número maior de câmeras. Os princípios de *light field* e *lumigraph* são semelhantes. A maior diferença entre as técnicas é que *lumigraph* usa uma aproximação geométrica para aprimorar o processo de renderização, diminuindo o número de amostras necessárias para compor a nova vista.

Schirmacher et al. [33] estendem o conceito de *light field* e *lumigraph* para cenas dinâmicas. O sistema de VBR apresentado adota uma arquitetura cliente-servidor com processamento distribuído na renderização das vistas da câmera virtual. O observador (cliente), ao especificar um novo ponto de vista, informa o servidor da ação desejada. O servidor, ao receber a requisição, coleta as imagens necessárias para a renderização das vistas em relação ao novo ponto de vista, renderiza e transmite o vídeo para o cliente.

As duas principais tarefas realizadas pelo servidor consistem na aquisição e renderização das imagens. Na etapa de aquisição, a cena é capturada por 6 câmeras, posicionadas em duas linhas por três colunas no plano  $st$  (Figura 17). A captura da cena é realizada por 3 computadores, distribuindo o processamento das imagens. Cada computador está conectado com um par de câmeras e é responsável por calcular o mapa de profundidade para cada par de imagens capturadas.

Para a renderização das vistas da câmera virtual, uma triangulação do plano  $st$  é inicialmente computada, semelhante à triangulação apresentada na Figura 18. Com este particionamento, as amostras que são empregadas no processo de renderização são estabelecidas (triângulos sombreados na Figura 18), levando em conta o ponto de vista da câmera virtual. Posteriormente, cada ponto das imagens selecionadas é projetado para a vista da câmera virtual. Como o mapa de

profundidade foi computado na etapa de captura, pontos 3D podem ser reconstruídos e depois projetados para a câmera virtual.

As vantagens e desvantagens do método descrito são detalhadas a seguir. O uso de mapa de profundidade contribui para o emprego de poucas câmeras no processo de renderização, uma vez que a informação 3D da cena para cada câmera é extraída. O servidor transmite para o cliente somente o vídeo renderizado de acordo com seu atual ponto de vista, reduzindo a transmissão de dados em comparação se a renderização fosse realizada no cliente. Em contraste, a computação não é realizada em tempo real, limitada principalmente pela geração dos mapas de profundidade. Além disso, ruídos nos mapas de profundidade estimados, decorrentes de oclusões e descontinuidades, não são tratados e podem comprometer a qualidade das vistas renderizadas.

Yang et al. [42] apresentam um sistema de VBR em tempo real baseado na técnica *light field*. A captura da cena é realizada por  $8 \times 8$  câmeras dispostas em um painel, o qual representa o plano das câmeras (Figura 17). A arquitetura do sistema é formada por 6 computadores responsáveis pelo gerenciamento das câmeras e o processamento distribuído da renderização das vistas, além de mais 1 computador empregado na composição das imagens finais. O sistema opera conforme descrição a seguir.

Primeiramente, o observador manipula a câmera virtual especificando um ponto de vista desejado. A partir dessa ação, o vídeo referente a este ponto de vista é requisitado para o computador central, o qual é responsável pela composição das vistas. Esta requisição é repassada para os computadores que controlam as câmeras. Cada câmera contribui para a renderização de fragmentos de imagem que são processados de forma distribuída, em que cada fragmento representa uma parte da imagem de saída. Por fim, para formar a imagem de saída, o compositor combina os fragmentos e envia a imagem resultante para o observador.

A renderização das vistas segue o princípio das técnicas *light field* e *lumigraph*, ilustradas na Figura 18. O plano das câmeras tem uma topologia fixa a partir de uma triangulação. Todos os triângulos associados com uma câmera são projetados sobre o plano focal. Estes triângulos são renderizados com relação ao ponto de vista desejado, aplicando as imagens das câmeras como textura. A renderização de cada triângulo constitui um fragmento de imagem empregado na composição da imagem final. Este método de renderização é mais eficiente em comparação à renderização feita pixel-a-pixel [11] e a renderização dos triângulos e o mapeamento das texturas podem ser implementados para executar em uma GPU.

As principais vantagens da abordagem proposta por Yang et al. [42] são a computação do sistema que é realizada em tempo real e a largura de banda para transmissão do vídeo ao observador que é limitada ao tamanho da imagem de saída. Em contraste, o processamento das vistas é centralizado e pode comprometer o desempenho com a adição de múltiplos observadores. Além disso, um grande número de câmeras e computadores é empregado para a síntese das vistas, tornando a implementação da abordagem de alto custo.

Liu et al. [22] estendem o trabalho apresentado por Yang et al. [42], agregando a compressão dos múltiplos vídeos capturados, transmissão via Internet e suporte a múltiplos observadores. Ao

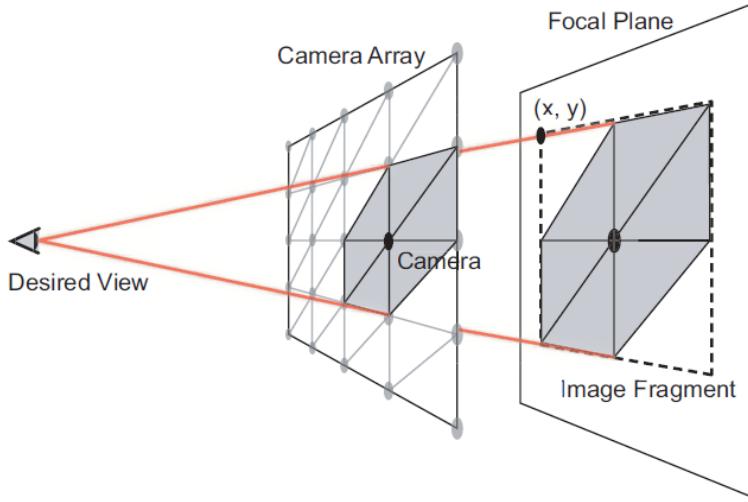


Figura 18: Triangulação do plano das câmeras empregada na renderização com *light field* e *lumigraph*. Fonte: [42].

invés de um computador central renderizar as vistas, computadores clientes realizam essa tarefa a partir de vídeos comprimidos recebidos pela rede. O sistema de VBR é em tempo real. As desvantagens dessa abordagem são o uso de diversos computadores empregados na captura e na transmissão dos dados e a grande quantidade de câmeras.

Em geral, as abordagens de VBR baseadas em *light field* e *lumigraph*, descritas na literatura que são em tempo real, empregam uma grande quantidade de câmeras e múltiplos computadores para o processamento distribuído das vistas. Outros métodos adotam algum tipo de informação geométrica, tal como a estimativa dos mapas de profundidade das imagens capturadas, reduzindo o número de câmeras, mas elevando o tempo de computação. Trabalhos recentes, tais como os apresentados por Wei et al. [41] e Jeon e Park [15], definem múltiplos planos focais (ao invés de apenas um, como ilustrado na Figura 17) na renderização das imagens, produzindo vistas com alta qualidade e com um número menor de câmeras.

A definição de múltiplos planos focais, cada um posicionado em uma profundidade diferente na cena, permite a renderização de múltiplas imagens de um mesmo ponto de vista e em diferentes profundidades. A vista final é computada pela verificação da profundidade ideal para cada ponto da imagem, analisando as múltiplas imagens computadas. Wei et al. [41] definem a profundidade dos pontos a partir da análise da similaridade de cor de cada ponto em relação a sua reprojeção em câmeras vizinhas. Jeon e Park [15] verificam a diferença entre blocos das múltiplas imagens renderizadas. A principal desvantagem dessas duas abordagens é que a computação não é realizada em tempo real, uma vez que um maior número de planos focais deve ser processado.

### B.3 Varredura de Planos

A técnica de varredura de planos foi inicialmente proposta por Collins [3], empregada na estimativa da relação geométrica entre múltiplas vistas, computando simultaneamente, a correspondência 2D entre pontos das imagens e a posição 3D desses pontos na cena (nesse trabalho, os pontos são restritos a características (*features*) extraídas das imagens).

Um dos primeiros trabalhos a empregar a técnica de varredura de planos para VBR foi apresentado por Yang et al. [44]. Utilizando 5 câmeras previamente calibradas (Apêndice A.4), em tempo real, o método pode ser utilizado para a síntese de novas vistas, a estimativa de mapas de profundidade ou a reconstrução da geometria de objetos, como detalhado a seguir.

Inicialmente, como ilustra a Figura 19, o espaço 3D é discretizado em planos paralelos e os objetos da cena estão localizados entre dois planos, o plano mais à frente e o mais afastado em relação às câmeras (*near* e *far*, respectivamente). A câmera virtual, denotada por  $cam_x$ , é posicionada entre duas câmeras reais (as câmeras reais são expressas por  $cam_1, \dots, cam_n$ ).

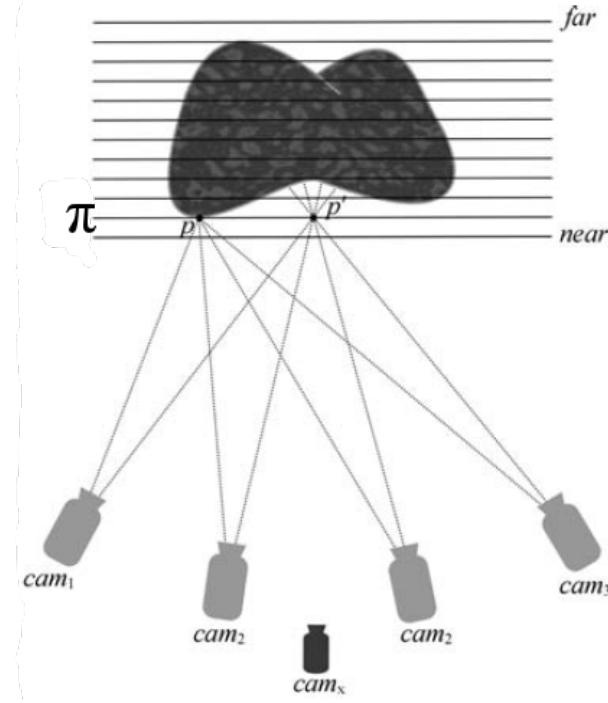


Figura 19: Discretização do espaço em planos e posicionamento das câmeras no ambiente. Fonte: [38].

A síntese dos quadros da câmera virtual é realizada pela varredura de planos, sendo que, para cada varredura, uma vista é renderizada. Durante a varredura de planos, o processamento de um plano  $\pi$  consiste em quatro etapas principais: (i) projeção das imagens das câmeras reais sobre o plano  $\pi$ , (ii) computação da cor e de uma medida denominada *consistência de cor* dos pontos projetados sobre  $\pi$ , (iii) projeção dos pontos sobre o plano  $\pi$  para o plano da câmera virtual, (iv) síntese da vista da câmera virtual a partir de uma votação.

Na primeira etapa, como mostra o lado esquerdo da Figura 20, as imagens referentes às  $n$  câmeras reais são projetadas para o plano  $\pi$ . Considerando que um ponto sobre o plano de uma câmera  $i$  é denotado por  $p_i$ , a primeira etapa consiste em projetar cada ponto  $p_i$  para um ponto  $P$  no espaço 3D, sendo que  $P$  está sobre o plano  $\pi$ . Como são  $n$  câmeras reais, cada ponto  $P$  pode ser projetado por  $n$  pontos diferentes. Este tipo de projeção é detalhada no Apêndice A.2.

Posteriormente, na segunda etapa, a cor e a medida de consistência de cor de cada ponto  $P$  sobre  $\pi$  são calculadas. A cor de  $P$  é estabelecida como sendo a cor média dos pontos  $p_i$ . A Figura 19 ilustra a projeção dos pontos  $P$  e  $P'$  em diferentes locais sobre o plano  $\pi$ . Neste caso, a cor de  $P$  e  $P'$  é computada pela média das cores relativas a quatro pontos. Nota-se que o ponto  $P$  está sobre o objeto na cena e provavelmente o valor de consistência de cor de  $P$  será maior que de  $P'$ , pois a cor de  $P'$  é formada pela média das cores dos pontos nas imagens, cujos raios de projeção coincidem em quatro lugares diferentes no objeto.

Segundo [44], assumindo que as superfícies dos objetos da cena são visíveis (não há oclusão) e exclusivamente difusas, a medida de consistência de cor de cada ponto  $P$  sobre o plano  $\pi$  pode ser computada como a soma das diferenças quadradas (SSD) entre a luminância dos pixels das imagens reais  $L_i$  e a luminância do pixel de uma imagem de referência  $L_{base}$ , conforme a equação

$$SSD = \sum_i (L_i - L_{base})^2 \quad (72)$$

em que o valor de  $L_{base}$  é definido pela imagem da câmera mais próxima da câmera virtual ( $cam_x$ ) e os valores de  $L_i$  são relativos aos pixels definidos pelos pontos  $p_i$  de cada câmera  $i$ . Assim, quanto menor for o valor de  $SSD$ , maior é a consistência de cor e também maior será a probabilidade de um ponto  $P$  estar sobre a superfície de um objeto na cena.

Na terceira etapa, os pontos sobre o plano  $\pi$  são projetados para o plano da câmera virtual, como mostra o lado direito da Figura 20. Como as coordenadas dos pontos 3D sobre o plano  $\pi$  são conhecidas e as câmeras são calibradas, essa projeção pode ser concebida pelo modelo de câmera estenopeica detalhado no Apêndice A.1.

Por fim, na quarta e na última etapas, as cores dos pixels da imagem da câmera virtual é determinada por uma votação. Quando os pontos são projetados para a câmera virtual durante a terceira etapa, os valores de consistência de cor dos pontos projetados com os atuais valores da câmera virtual são comparados e a cor de cada ponto da câmera virtual é definida pela cor mais consistente ao longo da varredura dos planos.

O método de renderização por varredura de planos detalhado acima pode ser facilmente adaptado para computar o mapa de profundidade da vista relativa à câmera virtual. Ao sintetizar uma vista, a informação de profundidade já é computada implicitamente, uma vez que um ponto  $p = (u, v)^T$  sobre o plano de uma câmera é projetado por um ponto  $P = (P_x, P_y, P_z)^T$  sobre um plano  $\pi$  no espaço 3D e, assim, a coordenada  $p_z$  define o valor profundidade do ponto  $p$ . Com o mapa de profundidade computado, a malha triangular de objetos da cena pode ser construída e as imagens das câmeras reais podem ser mapeadas como texturas sobre o modelo 3D.

A principal vantagem do método descrito é sua capacidade de computar, em tempo real, a renderização de novas vistas, a estimativa do mapa de profundidade e a reconstrução da geometria. Isso é possível porque a técnica pode ser implementada em GPU, projetando as imagens das câmeras reais para um plano no espaço por meio do mapeamento de múltiplas texturas, enquanto o processo de votação pode ser realizado empregando *fragment shader* [30].

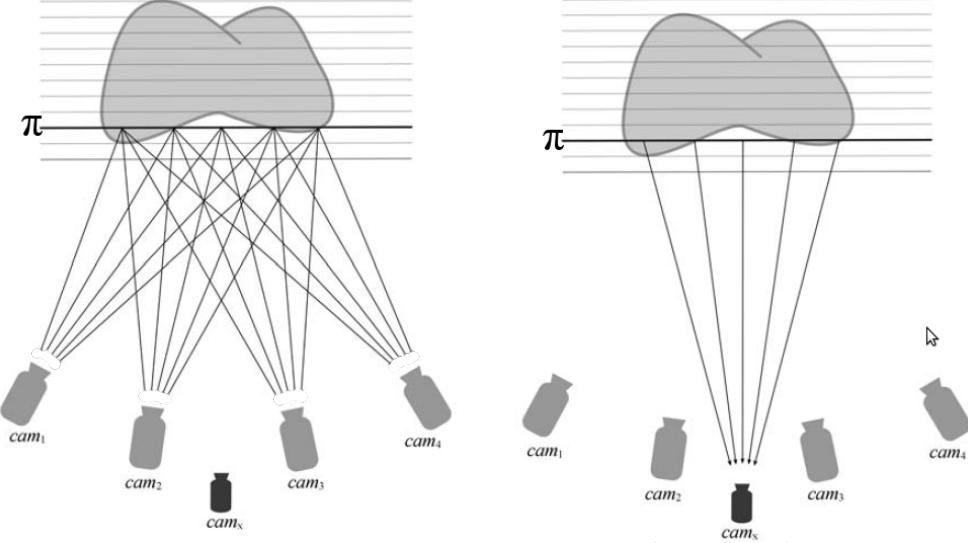


Figura 20: Renderização de vistas empregando o método de varredura de planos. Adaptada de [38].

Entre as desvantagens, pode-se citar que a qualidade da vista sintetizada pode depender do número de planos, sendo que, em cenas complexas, um número maior de planos é normalmente requerido para discretizar o espaço 3D e a distância (*baseline*) entre as câmeras reais deve ser curta para minimizar os problemas causados pela oclusão [44]. Além disso, o mapa de profundidade computado pela técnica de varredura de planos pode apresentar ruído, uma vez que oclusões não são tratadas e os valores de profundidade são computados pixel-a-pixel.

Yang e Pollefeys [43] aprimoraram o método de estimativa do mapa de profundidade baseado em varredura de planos inicialmente proposto por Yang et al. [44]. O algoritmo apresentado é destinado à visão estéreo, embora possa ser estendido para múltiplos pares de imagens. O método consiste em combinar a soma das diferenças quadradas para blocos de pixels (ao invés de ser pixel-a-pixel como em [44]) de diferentes tamanhos. As somas das diferenças quadradas para diferentes resoluções são computadas utilizando *mipmaps* [35]. As vantagens desta abordagem são que o mapa de profundidade pode ser computado em tempo real e o ruído no mapa de profundidade computado é reduzido significativamente em comparação com o trabalho proposto por [44]. Entretanto, as descontinuidades não são tratadas de forma explícita como feito em [47].

Gallup et al. [7] descrevem uma abordagem para reconstrução 3D de ambientes urbanos empregando varreduras de planos em múltiplas direções. Inicialmente, as direções de múltiplas

varreduras são identificadas baseadas na orientação das superfícies planares da cena. Posteriormente, a varredura dos planos em múltiplas direções é executada, sendo que cada varredura produz um mapa de profundidade que, quando combinados, produzem o mapa de profundidade da cena utilizado na reconstrução do ambiente urbano. O mapa de profundidade estimado por esta técnica é mais preciso se comparado com o método de varredura tradicional, que utiliza planos paralelos, como ilustra a Figura 21. Nota-se que na varredura de planos paralelos, ilustrada na Figura 21 (a), nem todos os pontos sobre o plano são correspondentes. Já na Figura 21 (b), em que a varredura é alinhada com a superfície, todos os pontos são correspondentes. A desvantagem desta técnica é que ela é restrita a cenas em que a orientação das superfícies planares pode ser determinada, tal como ruas de uma área urbana [7].

Geys et al. [9] apresentam um sistema em tempo real para teleconferência e ensino a distância em que o ponto de vista da câmera é flexível, permitindo a visualização da cena sob diferentes pontos de vista, não ficando restrita apenas aos pontos de vista definidos pelas câmeras reais. Além disso, o sistema implementa realidade aumentada, possibilitando a troca do plano de fundo (*background*) da cena e a adição de rótulos virtuais. A cena é composta por um instrutor, que representa o objeto dinâmico da cena (plano frontal). A captura da cena é realizada por três câmeras calibradas, posicionadas ao redor do instrutor. O processo de síntese das vistas da câmera virtual consiste inicialmente em estimar o mapa de profundidade. Novas vistas são obtidas pela renderização da malha triangular construída empregando a informação de profundidade computada. As imagens das câmeras reais são utilizadas como texturas do modelo 3D construído.

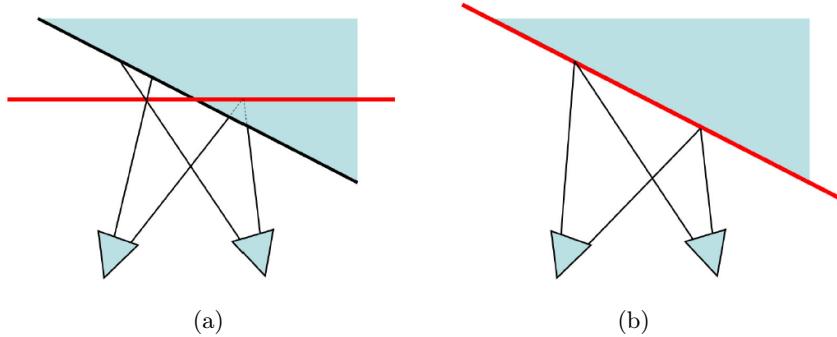


Figura 21: Varredura de planos com múltiplas direções. (a) superfície inclinada com varredura de planos paralelos; (b) varredura de planos alinhada com a superfície inclinada. Fonte: [7].

Assume-se que o plano de fundo da cena raramente é atualizado, ou seja, é praticamente estático. Assim, o mapa de profundidade é computado de forma híbrida, empregando um algoritmo mais custoso, que produza valores de profundida mais precisos para o plano de fundo e utilizando um algoritmo mais rápido para estimar os valores de profundidade do plano frontal. Inicialmente, a cena é segmentada, extraíndo o plano de fundo e o *bounding-box* dos objetos pertencentes ao plano frontal. Então, a informação de profundidade relativa ao plano de fundo é gerada pixel-a-pixel pela técnica de varredura de planos empregando GPU. A profundidade dos pontos referentes

a cada objeto dinâmico é computada na CPU, que calcula a soma das diferenças absolutas entre blocos de pixels das duas imagens, sendo que a busca por pontos correspondentes é delimitada pelo *bounding-box* que engloba os objetos dinâmicos e não para toda a imagem, acelerando o processo de computação.

As principais vantagens do trabalho apresentado por Geys et al. [9] são que o método possibilita a síntese de vistas da câmera virtual em tempo real, permite a implementação de realidade aumentada e requer poucas câmeras. Em contraste, em ambientes onde o plano de fundo é dinâmico, a técnica não pode ser empregada, pois a extração do plano de fundo é realizada pela comparação entre uma imagem capturada inicialmente, antes do processo de captura, com o quadro atual das câmeras de referência.

Nozick e Saito [26] descrevem um método para VBR em tempo real empregando a técnica de varreduras de planos no processo de síntese das vistas da câmera virtual. As principais contribuições desse trabalho são um novo método para determinar a medida de consistência de cor e o processo de captura da cena que pode ser realizado por câmeras em movimento (no caso, quatro *webcams* são empregadas). Segundo [26], a computação da consistência de cor proposta por [44] (Equação 72) restringe o posicionamento da câmera virtual, que deve estar próxima de uma câmera de referência e entre duas câmeras reais; caso contrário, a câmera virtual não pode ser representada. Além disso, artefatos podem surgir no vídeo renderizado quando há mudança na câmera de referência. Com base nessas premissas, Nozick e Saito [26] propõem um método para calcular a consistência de cor a partir da variância entre as cores dos pontos sobre os planos das câmeras e a cor média

$$v = \sum_{i=1..n} (c_i - c_m)^2 \quad (73)$$

em que  $v$  representa a variância calculada,  $n$  o número de câmeras reais,  $c_i$  a cor de um ponto projetado no plano de varredura referente à imagem  $i$  e  $c_m$  representa a cor média que pode ser calculada como

$$c_m = \frac{1}{n} \sum_{i=1..n} c_i$$

Outra contribuição do trabalho descrito em [26] é o processo de captura da cena, que pode ser feito por câmeras em movimento. Para isso, os parâmetros das câmeras devem ser atualizados constantemente a cada quadro. As câmeras são calibradas utilizando marcadores, que são padrões de calibração 2D desenhados em quadros de cor preta, como mostra a Figura 22. Os marcadores correspondem ao plano de fundo da cena. Primeiramente, uma imagem de referência deve ser capturada contendo todos os marcadores (Figura 22 (b)). Posteriormente, para cada câmera, uma relação geométrica entre o plano da imagem de referência e o plano da câmera real (Figura 22 (b)) é obtida pela estimativa de uma matriz de homografia  $H$  (Apêndice A.3). O conjunto de pontos correspondentes necessários para a estimativa de  $H$  é computado pela biblioteca ARToolKit [17]. Por fim, a homografia  $H$  é utilizada para calibrar a câmera empregando a técnica apresentada por [46] e descrita no Apêndice A.4.

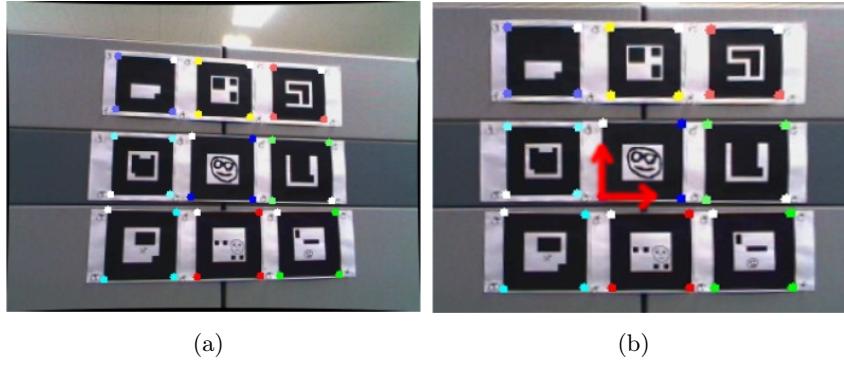


Figura 22: Marcadores empregados na calibração das câmeras. (a) imagem capturada durante a renderização da cena. (b) imagem de referência capturada antes do processo de renderização. Fonte: [7].

As principais desvantagens relacionadas ao trabalho apresentado por Nozick e Saito [26] são o uso de um plano de fundo composto por marcadores, que não é o real da cena, sendo necessária sua extração e substituição por um plano de fundo virtual para exibição dos quadros e, além disso, a técnica somente funciona se os marcadores são visualizados por todas as câmeras, restringindo a movimentação das câmeras.

Nozick e Saito [38] estendem o processo de renderização baseado na varredura de planos, de uma vista para múltiplas, como ilustrado na Figura 23. A partir de uma votação, a consistência das cores é computada, os pontos sobre os planos durante a varredura são projetados para múltiplas vistas, ao invés de uma única, como no método tradicional. Essa extensão foi proposta para obter pares de imagens capturadas da mesma cena e de diferentes pontos de vistas, utilizados por dispositivos *auto-estereoscópicos 3D*<sup>2</sup>. O método é executado em tempo real e utiliza quatro *webcams* calibradas.

Jarusirisawad et al. [14] apresentam uma abordagem para VBR utilizando varredura de planos. A principal contribuição desse trabalho é o uso de câmeras não-calibradas, em que os parâmetros intrínsecos ou extrínsecos das câmeras não são estimados previamente e uma calibração fraca das câmeras é realizada em tempo real. A calibração fraca para múltiplas câmeras empregada nesse trabalho é baseada em [31]. A calibração é realizada a partir do espaço 3D definido pelas coordenadas de duas câmeras de referência, como ilustra a Figura 24. A ideia básica do sistema de calibração é definir a correspondência entre pontos de vistas de referência e, posteriormente, transferir esta correspondência para as demais câmeras reais (*non-basis cameras*) via matrizes fundamentais (a transferência de pontos usando matrizes fundamentais é detalhada no Apêndice A.5.4).

Como os parâmetros das câmeras são desconhecidos, a posição da câmera virtual é definida

---

<sup>2</sup>Dispositivos auto-estereoscópicos 3D provêm a percepção 3D sem a necessidade de lentes especiais [4].

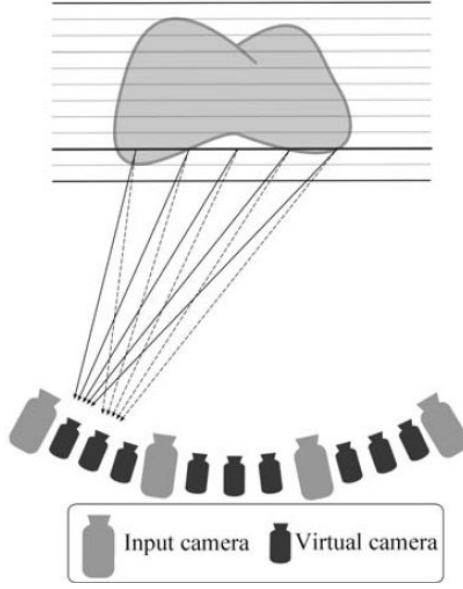


Figura 23: Renderização de múltiplas vistas empregando o método de varredura de planos. Adaptada de [38].

a partir de duas câmeras reais adjacentes pela seguinte interpolação

$$x_3 = (1 - r)x_1 + rx_2, \quad (74)$$

em que  $x_3$  é ponto do plano da câmera virtual computado,  $x_1$  e  $x_2$  são pontos relativos às câmeras reais adjacentes à esquerda e à direita da câmera virtual, respectivamente e  $r$  define a distância entre as duas câmeras reais utilizadas na interpolação, para  $0 \leq r \leq 1$ . A definição da câmera virtual é ilustrada na Figura 25.

O processo de síntese das vistas da câmera virtual é realizado pela técnica de varredura de planos, semelhante aos trabalhos descritos acima. Entretanto, com o uso de câmeras não-calibradas, as matrizes de projeção das câmeras são desconhecidas e a projeção de pontos entre os planos das câmeras e planos no espaço 3D é estabelecida por matrizes de homografia  $H$ . A estimativa de  $H$  é estabelecida pela correspondência entre pontos sobre os planos das câmeras e pontos sobre um plano  $\pi$  no espaço 3D. Os pontos correspondentes usados na estimativa são obtidos pela calibração fraca das câmeras (Figura 24).

As principais vantagens do método proposto por Jarusirisawad et al. [14] são o emprego de câmeras não-calibradas, a renderização em tempo real dos quadros da câmera virtual e as poucas câmeras necessárias para a captura da cena (no caso, 6 *webcams* são utilizadas).

Em contraste, o método tem como desvantagens a curta distância entre a posição das câmeras na cena e o posicionamento precário da câmera virtual. A proximidade das câmeras está relacionada com a redução de artefatos nas vistas renderizadas. O posicionamento da câmera virtual, estabelecido pela Equação 74 e ilustrado na Figura 25, não oferece uma transição suave

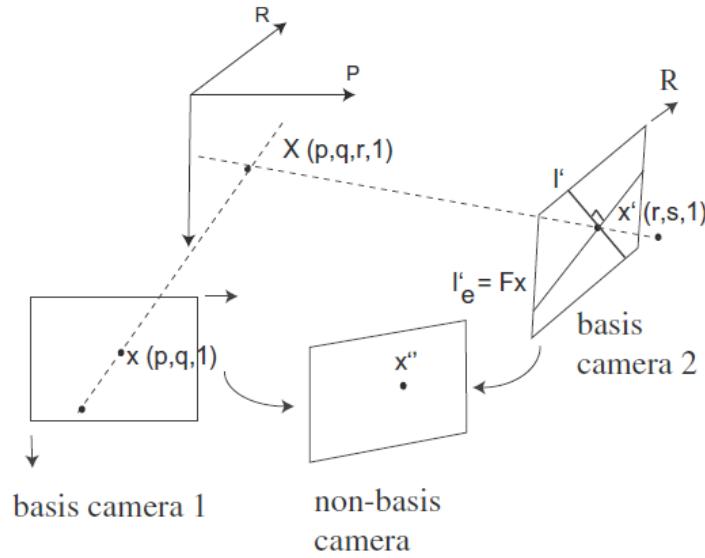


Figura 24: Calibração fraca das câmeras. Fonte: [14].

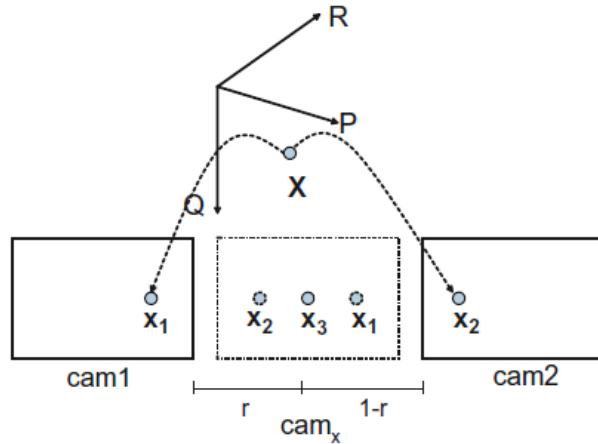


Figura 25: Definição da posição da câmera virtual a partir de duas câmeras reais adjacentes. Fonte: [14].

do ponto de vista da primeira câmera de referência para a segunda, pois não leva em consideração a mudança gradual do foco e do ponto principal da primeira para a segunda câmera, nem a interpolação da rotação entre as duas câmeras, somente a interpolação da posição dos centros de projeção das câmeras é considerada.

## Referências

- [1] BEKAERT, T., GAUTAMA, S., PHILIPS, W. e GOOSSENS, R. Dense and Reliable DSM Generation from VHR Stereo Pairs in Urban Environments. In *Proceedings 2nd International Workshop on the Future of Remote Sensing* (Antwerp, oct 2006), pp. 1–4.
- [2] CHUANG, Y.-Y., CURLESS, B., SALESIN, D. e SZELISKI, R. A Bayesian Approach to Digital Matting. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2001), vol. 2, pp. 264–271.
- [3] COLLINS, R. T. A Space-Sweep Approach to True Multi-Image Matching. In *Conference on Computer Vision and Pattern Recognition (CVPR '96)* (Washington, DC, USA, 1996), IEEE Computer Society, pp. 358–.
- [4] DODGSON, N. A. Autostereoscopic 3D Displays. *Computer* 38 (August 2005), 31–36.
- [5] FRY, J. e PUSATERI, M. A System and Method for Auto-Correction of First Order Lens Distortion. In *39th Applied Imagery Pattern Recognition Workshop (AIPR)* (oct. 2010), pp. 1–4.
- [6] FUSIELLO, A., TRUCCO, E. e VERRI, A. A Compact Algorithm for Rectification of Stereo Pairs. *Mach. Vision Appl.* 12, 1 (Julho 2000), 16–22.
- [7] GALLUP, D., FRAHM, J.-M., MORDOHAI, P., YANG, Q. e POLLEFEYS, M. Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions. In *IEEE Conference on Computer Vision and Pattern Recognition* (june 2007), pp. 1–8.
- [8] GARLAND, M., LE GRAND, S., NICKOLLS, J., ANDERSON, J., HARDWICK, J., MORTON, S., PHILLIPS, E., ZHANG, Y. e VOLKOV, V. Parallel Computing Experiences with CUDA. *IEEE Micro* 28, 4 (July-Aug. 2008), 13–27.
- [9] GEYS, I. e DE ROECK, S. The Augmented Auditorium: Fast Interpolated and Augmented View Generation. In *2nd IEE European Conference on Visual Media Production* (2005), pp. 94–103.
- [10] GONZALEZ, R. C. e WOODS, R. E. *Digital Image Processing*, 2nd ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.
- [11] GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R. e COHEN, M. F. The Lumigraph. In *23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (New York, NY, USA, 1996), pp. 43–54.
- [12] GROBE, E., ANTON, H. e GROBE, C. *Elementary Linear Algebra, Student Solutions Manual*. Wiley, 2000.

- [13] HARTLEY, R. E ZISSEMAN, A. *Multiple View Geometry in Computer Vision*, second ed. Cambridge University Press, 2004.
- [14] JARUSIRISAWAD, S., NOZICK, V. E SAITO, H. Real-Time Video-based Rendering from Uncalibrated Cameras using Plane-Sweep Algorithm. *Journal of Visual Communication and Image Representation* 21 (Julho 2010), 577–585.
- [15] JEON, Y. E PARK, H. Fast All In-Focus Light Field Rendering Using Dynamic Block-Based Focusing Technique. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video* (may 2011), pp. 1–4.
- [16] KANADE, T., NARAYANAN, P. J. E R, P. W. Virtualized Reality: Concepts and Early Results. In *IEEE Workshop on the Representation on Visual Scene* (1995).
- [17] KATO, H. E BILLINGHURST, M. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality* (Washington, DC, USA, 1999), IEEE Computer Society, pp. 85–94.
- [18] KIM, B.-K., CHUNG, S.-W., SONG, M.-K. E SONG, W.-J. Correcting Radial Lens Distortion with Advanced Outlier Elimination. In *International Conference on Audio Language and Image Processing (ICALIP)* (nov. 2010), pp. 1693–1699.
- [19] KIMMEL, R. Demosaicing: Image reconstruction from color ccd samples. *IEEE Transactions on Image Processing* 8, 9 (sep 1999), 1221 –1228.
- [20] LEVOY, M. E HANRAHAN, P. Light Field Rendering. In *23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (New York, NY, USA, 1996), pp. 31–42.
- [21] LI, W., ZHOU, J., LI, B. E SEZAN, M. I. Virtual view specification and synthesis for free viewpoint television. *IEEE Transactions on Circuits and Systems for Video Technology* 19, 4 (2009), 533–546.
- [22] LIU, Y., DAI, Q. E XU, W. A Real Time Interactive Dynamic Light Field Transmission System. In *IEEE International Conference on Multimedia and Expo* (july 2006), pp. 2173–2176.
- [23] McMILLAN, L. *An Image-Based Approach to Three-Dimensional Computer Graphics*. PhD thesis, University of North Carolina, Abril 1997.
- [24] MEDIONI, G. E KANG, S. B. *Emerging Topics in Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.

- [25] MORVAN, Y. *Acquisition, Compression and Rendering of Depth and Texture for Multi-View Video*. PhD thesis, Eindhoven University of Technology, The Netherlands, Abril 2009.
- [26] NOZICK, V. E SAITO, H. Real-Time Free Viewpoint from Multiple Moving Cameras. In *Proceedings of the 9th international conference on Advanced concepts for intelligent vision systems* (Berlin, Heidelberg, 2007), ACIVS'07, Springer-Verlag, pp. 72–83.
- [27] NVIDIA CORPORATION. CUDA GPUs, 2012. <http://developer.nvidia.com/cuda-gpus>, acesso em: 15 de fevereiro de 2012.
- [28] OH, K.-J., YEA, S. E HO, Y.-S. Hole Filling Method Using Depth Based In-Painting for View Synthesis in Free Viewpoint Television and 3-D Video. In *Picture Coding Symposium* (may 2009), pp. 1–4.
- [29] OKUTOMI, M. E KANADE, T. A Multiple-Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15 (Abril 1993), 353–363.
- [30] ROST, R. J., LICEA-KANE, B., GINSBURG, D., KESSENICH, J. M., LICHTENBELT, B., MALAN, H. E WEIBLEN, M. *OpenGL Shading Language*, 3rd ed. Addison-Wesley Professional, 2009.
- [31] SAITO, H. E KANADE, T. Shape Reconstruction in Projective Grid Space from Large Number of Images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1999), vol. 2, pp. 49–54.
- [32] SANTOS, M. C. Renderização de Cenas Tridimensionais Interativas em Computadores com Recursos Gráficos Limitados. Dissertação de Mestrado, Universidade Federal do Paraná, Curitiba, PR, Brasil, março 2009.
- [33] SCHIRMACHER, H., LI, M. E SEIDEL, H.-P. On-the-Fly Processing of Generalized Lumigraphs. In *Eurographics* (2001), pp. 165–173.
- [34] SHADE, J., GORTLER, S., WEI HE, L. E SZELISKI, R. Layered Depth Images. In *25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (New York, NY, USA, 1998), pp. 231–242.
- [35] SHREINER, D. E GROUP, T. K. O. A. W. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 3.0 and 3.1*, 7th ed. Addison-Wesley Professional, 2009.
- [36] SHU, C., BRUNTON, A. E FIALA, M. A topological approach to finding grids in calibration patterns. *Machine Vision and Applications* 21 (October 2010), 949–957.
- [37] SUN, J., ZHENG, N.-N. E SHUM, H.-Y. Stereo Matching Using Belief Propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 7 (july 2003), 787–800.

- [38] VINCENT NOZICK, H. S. On-line Free-viewpoint Video: From Single to Multiple View Rendering. *International Journal of Automation and Computing* 5, 3 (2008).
- [39] WANG, L.-H., ZHANG, J., YAO, S.-J., LI, D.-X. e ZHANG, M. GPU Based Implementation of 3DTV System. In *Sixth International Conference on Image and Graphics* (Aug. 2011), pp. 847–851.
- [40] WANG, Z., WANG, Z. e WU, Y. Recognition of Corners of Planar Checkboard Calibration Pattern Image. In *Chinese Control and Decision Conference (CCDC)* (may 2010), pp. 3224–3228.
- [41] WEI, W., ZHI, J. Z., CONG, Y. S. e DAN, Z. An efficient method for all-in-focused light field rendering. In *3rd IEEE International Conference on Computer Science and Information* (july 2010), vol. 1, pp. 399–404.
- [42] YANG, J. C., EVERETT, M., BUEHLER, C. e McMILLAN, L. A real-time distributed light field camera. In *13th Eurographics Workshop on Rendering* (Aire-la-Ville, Suíça, 2002), Eurographics Association, pp. 77–86.
- [43] YANG, R. e POLLEFEYS, M. Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware. In *IEEE Conference on Computer Vision and Pattern Recognition* (june 2003), vol. 1, pp. 211–217.
- [44] YANG, R., WELCH, G. e BISHOP, G. Real-Time Consensus-Based Scene Reconstruction Using Commodity Graphics Hardware. In *10th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2002), IEEE Computer Society.
- [45] YOON, S.-U., LEE, E.-K., KIM, S.-Y. e HO, Y.-S. A framework for representation and processing of multi-view video using the concept of layered depth image. *Journal of VLSI Signal Processing Systems* 46, 2-3 (2007), 87–102.
- [46] ZHANG, Z. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 11 (nov 2000), 1330 – 1334.
- [47] ZITNICK, C. L., KANG, S. B., UYTTENDAELE, M., WINDER, S. e SZELISKI, R. High-Quality Video View Interpolation using a Layered Representation. *ACM Transactions on Graphics* 23 (Agosto 2004), 600–608.