

2

Texturas com Relevo e Equações de Pré Warping

A noção de *warping* de imagens é fundamental para o entendimento da técnica abordada nesta dissertação. Este capítulo apresenta definições formais para vários conceitos relacionados, tais como: *imagens*, *imagens com profundidade*, *mapas de warp* e *texturas com relevo*. Além disso, é explicada neste capítulo a equação de *warping* 3D de imagens [24] proposta por McMillan, bem como é explorada uma discussão acerca da fatoração ideal para tal equação sob o ponto de vista de sua eficiência computacional. Como propósito principal, discute-se as equações de pré *warping* [26], propostas por Oliveira, que são a base para o processo de mapeamento de textura com relevo. Finalmente, este capítulo discorre sobre a ordem compatível de oclusão para imagens com profundidade. Este método, proposto em [24], especifica a ordem na qual os *pixels* de uma imagem com profundidade devem ser processados de maneira que a visibilidade correta seja alcançada para pontos de vista arbitrários.

2.1

Imagens e Mapas de Warp

Uma *imagem contínua* consiste em um suporte bidimensional onde, a cada ponto, associa-se uma informação de cor. Assim, utiliza-se como modelo matemático para representar uma imagem a função $i : U \rightarrow C$, onde $U \subset \mathbb{R}^2$ é o suporte da imagem e C é o espaço de cor [17]. Neste modelo, o contra-domínio é geralmente um espaço tridimensional $C = \mathbb{R}^3$ como, por exemplo, o espaço RGB (*Red*, *Green* e *Blue*). Além da cor, os elementos de C podem carregar informações sobre outros atributos da imagem, tais como transparência, profundidade, etc.

Para que uma imagem possa ser visualizada em um computador é

necessário que exista uma representação digital para tal entidade. Assim, é necessário discretizar tanto o domínio quanto o contra-domínio da função imagem. Denota-se por *amostragem* o processo de discretizar o suporte bidimensional e de *quantização* a discretização do espaço de cor. Dessa forma, uma *imagem digital* $i_d : U' \rightarrow C'$ é uma amostra de uma imagem contínua, onde $U' \subset U$ é um reticulado uniforme e ortogonal e $C' \subset C$ é uma quantização de C .

Um mapa de *warp* 2D $w : U \rightarrow W$, onde $W \subset \mathbb{R}^2$, é uma transformação geométrica que deforma o suporte de uma imagem conseqüentemente produzindo uma nova imagem. Usualmente, sua entrada é referenciada como *imagem fonte*, enquanto que sua saída é chamada *imagem destino*. No caso digital, o mapa de *warp* $w_d : U' \rightarrow W \subset \mathbb{R}^2$ distorce o reticulado de entrada, geralmente fazendo com que sua ortogonalidade e uniformidade sejam perdidas. Vale notar que a técnica habitualmente denominada *warping 3D de imagens* [24] é, de fato, um mapa de *warp* 2D.

2.2

Imagens com Profundidade e Texturas com Relevo

Segundo Oliveira [26], uma *imagem com profundidade* é um par $\{i_d, K\}$, onde $i_d : U' \rightarrow C'$ é uma imagem digital e K é um modelo de câmera associado a i_d . Cada elemento do espaço de cor C' de i_d é aumentado para incluir um valor escalar representando a distância, no espaço Euclideano, entre o ponto amostrado e uma entidade de referência. Se K é um modelo de câmera de projeção perspectiva, a imagem é denominada uma *imagem de projeção perspectiva com profundidade* e a entidade de referência é o centro de projeção de K . Se, entretanto, K é um modelo de câmera de projeção paralela, a imagem é chamada de *imagem de projeção paralela com profundidade* e a entidade de referência é o plano de projeção de K .

A Figura 2.1 exhibe um modelo de uma câmera de projeção perspectiva. Os vetores \vec{a} e \vec{b} formam uma base para o plano de projeção da câmera. As normas destes vetores são, respectivamente, o espaçamento horizontal e vertical da amostra no espaço Euclideano. \dot{C} é o centro de projeção (COP) da câmera e \vec{c} é um vetor do COP à origem do plano de projeção [24].

A Figura 2.2 mostra o modelo para uma câmera de projeção paralela. Os vetores \vec{a} e \vec{b} têm as mesmas definições como no caso da câmera

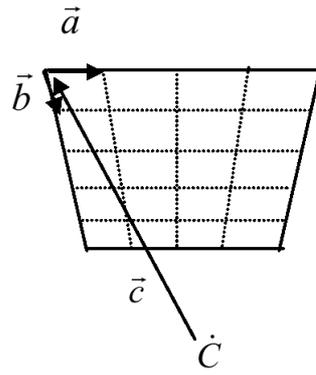


Figura 2.1: Modelo de câmera de projeção perspectiva. (Imagem obtida em [26].)

perspectiva da Figura 2.1. O vetor \vec{f} é um vetor unitário e ortogonal ao plano de projeção gerado por \vec{a} e \vec{b} . As extremidades finais de todos estes vetores estão em \dot{C} , a origem do plano de projeção da câmera.

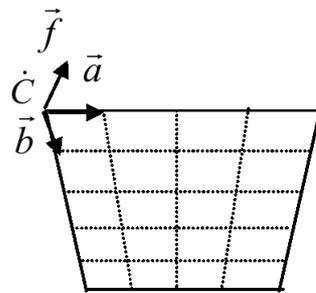


Figura 2.2: Representação do modelo de câmera de projeção paralela. (Imagem obtida em [26].)

Texturas com relevo são imagens de projeção paralela com profundidade e são utilizadas como primitivas de modelagem e renderização. As Figuras 2.3(a) e 2.3(b) ilustram a textura de cor e o mapa de profundidade associado à textura com relevo exibida na Figura 2.4.

2.3

Warping Tridimensional de Imagens

Warping tridimensional de imagens [24] é uma transformação geométrica $w : U' \rightarrow W \subset \mathbb{R}^2$ que mapeia uma imagem fonte com profundidade $\{i_s, K_s\}$ sobre uma imagem destino i_t . O conteúdo geométrico da cena é representado de forma implícita combinando-se informação de profundidade com o modelo de câmera associado à imagem fonte. Con-



Figura 2.3: Textura de cor (a) e mapa de profundidade (b) associado a uma textura com relevo. Regiões mais escuras no mapa de profundidade indicam superfícies mais distantes. (Imagem obtida em [26].)

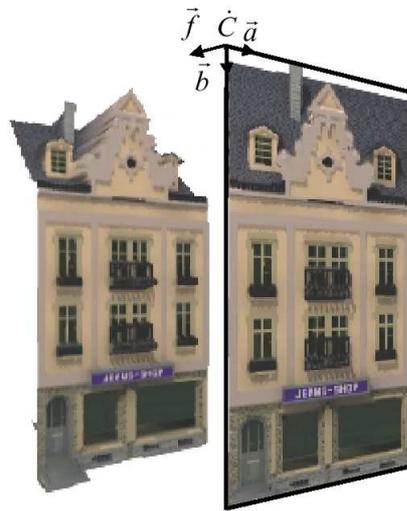


Figura 2.4: Uma textura com relevo e sua reprojeção visualizada de um ângulo oblíquo. (Imagem obtida em [26].)

seqüentemente, seja \hat{x} um ponto no espaço Euclideano cuja projeção sobre o suporte da imagem $\{i_s, K_s\}$ tenha coordenadas (u_s, v_s) . Utilizando-se o modelo de câmera perspectiva da Figura 2.1, as coordenadas de \hat{x} podem ser escritas como

$$\hat{x} = \dot{C}_s + \begin{bmatrix} a_{sx} & b_{sx} & c_{sx} \\ a_{sy} & b_{sy} & c_{sy} \\ a_{sz} & b_{sz} & c_{sz} \end{bmatrix} \begin{bmatrix} u_s \\ v_s \\ 1 \end{bmatrix} t_s(u_s, v_s) = \dot{C}_s + P_s \vec{x}_s t_s(u_s, v_s) \quad (2-1)$$

onde P_s é uma matriz 3x3 que representa o modelo de câmera associado à imagem fonte, $\vec{x}_s = \begin{bmatrix} u_s & v_s & 1 \end{bmatrix}^T$ e o escalar $t_s(u_s, v_s)$ é a razão entre as distâncias de \dot{C}_s a \hat{x} e de \dot{C}_s ao *pixel* (u_s, v_s) . Veja a Figura 2.5.

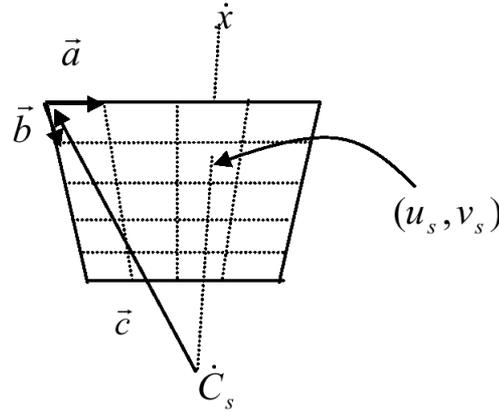


Figura 2.5: Recuperação das coordenadas de um ponto no espaço Euclidiano a partir de uma imagem de projeção perspectiva com profundidade. (*Imagem obtida em [26].*)

A projeção de \dot{x} sobre uma imagem destino qualquer pode ser obtida expressando-se as coordenadas de \dot{x} em ambos os sistemas de câmera [24](Figura 2.6):

$$\begin{aligned} \dot{C}_t + P_t \vec{x}_t t_t(u_t, v_t) &= \dot{x} = \dot{C}_s + P_s \vec{x}_s t_s(u_s, v_s) \\ \vec{x}_t t_t(u_t, v_t) &= P_t^{-1} \left[P_s \vec{x}_s t_s(u_s, v_s) + (\dot{C}_s - \dot{C}_t) \right] \\ \vec{x}_t &\doteq P_t^{-1} \left[P_s \vec{x}_s t_s(u_s, v_s) + (\dot{C}_s - \dot{C}_t) \right] \end{aligned} \quad (2-2)$$

onde \doteq é uma equivalência projetiva, isto é, o mesmo exceto por um múltiplo escalar.

Dividindo-se a Equação (2-2) por $t_s(u_s, v_s)$ e distribuindo-se P_t^{-1} tem-se:

$$\vec{x}_t \doteq P_t^{-1} P_s \vec{x}_s + P_t^{-1} (\dot{C}_s - \dot{C}_t) \delta_s(u_s, v_s) \quad (2-3)$$

onde o termo $\delta_s(u_s, v_s) = 1/t_s(u_s, v_s)$ é denominado *disparidade generalizada* do *pixel* fonte (u_s, v_s) [24]. A Equação (2-3), chamada de *equação de warping 3D de imagens* [24], provê uma interpretação mais intuitiva do processo: a imagem destino é obtida aplicando-se uma transformação perspectiva planar à imagem fonte, e um deslocamento por *pixel* proporcional ao termo $\delta_s(u_s, v_s)$ na direção da linha epipolar¹.

Vale notar que o mapeamento de textura é um caso especial de *warping 3D de imagens* no qual todos os *pixels* da imagem fonte compartilham um único valor de disparidade [24].

¹A projeção do COP de uma câmera sobre o plano de projeção de uma outra câmera.

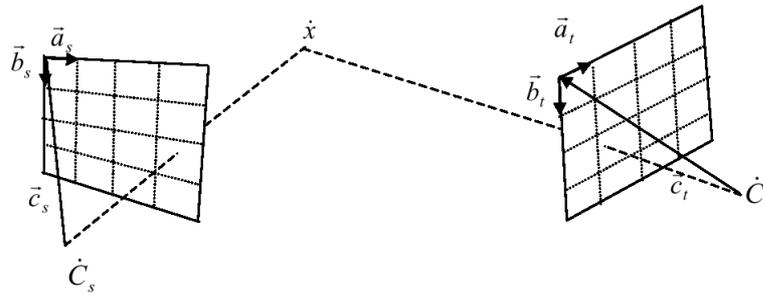


Figura 2.6: Ponto \hat{x} no espaço Euclidiano projetado sobre o suporte da imagem fonte e sobre o suporte da imagem destino. (Imagem obtida em [26].)

2.4

A Fatoração Ideal

A Equação (2-3) descreve de maneira concisa o processo de *warping*. Como discutido na Seção 2.3, esta equação é a composição de duas transformações bidimensionais: uma transformação perspectiva planar (referente ao primeiro termo) e um deslocamento por *pixel* na direção da linha epipolar. Idealmente, o mapeamento descrito por tal equação deveria ser fatorado para permitir que a transformação perspectiva planar (essencialmente, uma operação de mapeamento de textura) seja aplicada após o deslocamento na direção da linha epipolar. Tal fatoração, proposta por Oliveira [26], oferece algumas vantagens. Primeiramente, é possível beneficiar-se do mapeamento de textura implementado em *hardware* para realizar a transformação final. Além disso, a equação resultante tem uma estrutura unidimensional muito simples, o que permite que tal equação seja implementada utilizando-se somente operações ao longo de linhas e colunas, como será discutido mais adiante.

Com a finalidade de se obter esta fatoração ideal [26], é necessário encontrar um pré *warp* $p : U' \rightarrow Q \subset \mathbb{R}^2$ de modo que a composição $m \circ p : U' \rightarrow W \subset \mathbb{R}^2$, onde $m : Q \rightarrow W \subset \mathbb{R}^2$ é um mapeamento de textura padrão, seja equivalente ao *warp* 3D de imagem $w : U' \rightarrow W \subset \mathbb{R}^2$. O *warp* w mapeia amostras de uma imagem fonte com profundidade sobre o plano de projeção de uma câmera destino. m deve mapear *texels* do suporte da imagem fonte sobre *pixels* no suporte da imagem destino. Portanto, a questão fundamental é que, durante o processo de *warping*, visibilidade e possíveis distorções induzidas sejam computadas em relação ao COP da imagem destino.

Deste modo, sejam $(u_i, v_i) = (u_s + \Delta u, v_s + \Delta v)$ as coordenadas intermediárias obtidas após deslocar as coordenadas do *pixel* fonte (u_s, v_s) por $(\Delta u, \Delta v)$. A equivalência entre o mapeamento composto $m \circ p$ e w é modelada pelas Equações (2-4) e (2-5).

$$\frac{Au_i + Bv_i + C}{Iu_i + Jv_i + K} = \frac{Au_s + Bv_s + C + D\delta(u_s, v_s)}{Iu_s + Jv_s + K + L\delta(u_s, v_s)} \quad (2-4)$$

$$\frac{Eu_i + Fv_i + G}{Iu_i + Jv_i + K} = \frac{Eu_s + Fv_s + G + H\delta(u_s, v_s)}{Iu_s + Jv_s + K + L\delta(u_s, v_s)} \quad (2-5)$$

A interpretação correta destas equações é crucial para o entendimento central da técnica utilizada nesta dissertação. De modo resumido, as Equações (2-4) e (2-5) podem ser interpretadas como sendo o seguinte problema: **Encontrar o par (u_i, v_i) para cada *pixel* (u_s, v_s) da imagem fonte $\{i_s, K_s\}$ de maneira que o resultado da visualização de $\{i_s, K_s\}$ vista do COP de uma câmera destino K_t seja idêntico ao resultado visual obtido aplicando-se o *warp 3D* de $\{i_s, K_s\}$ sobre o plano de projeção de K_t .** O processo é ilustrado na Figura 2.7. Uma vez que a imagem processada está pronta para ser mapeada sobre o suporte da imagem fonte, esta pode ser mapeada sobre um polígono que possua as dimensões, posição e orientação de tal plano de suporte. Esta noção permite texturas com relevo serem utilizadas como primitivas de modelagem.

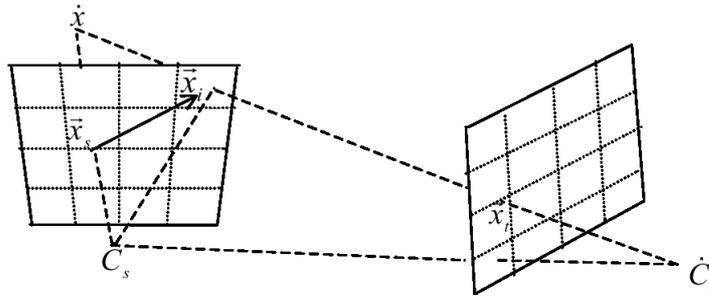


Figura 2.7: A amostra \vec{x}_s é deslocada para \vec{x}_i com o propósito de obter a visão de \dot{x} visto de \dot{C}_t . (Imagem obtida em [26].)

O *warp* desejado p é obtido resolvendo-se as Equações (2-4) e (2-5) para u_i e v_i . A resolução de tal sistema produz as seguintes equações unidimensionais:

$$u_i = \frac{u_s + q_1\delta(u_s, v_s)}{1 + q_3\delta(u_s, v_s)} \quad (2-6)$$

$$v_i = \frac{v_s + q_2\delta(u_s, v_s)}{1 + q_3\delta(u_s, v_s)} \quad (2-7)$$

onde q_1 , q_2 e q_3 são constantes para uma dada configuração das câmeras fonte e destino e, junto com $\delta(u_s, v_s)$, determinam a quantidade de deslocamento que as coordenadas de *pixels* da imagem fonte sofrerão ao longo da linha epipolar.

As equações de *pré warping* (2-6) e (2-7) resolvem visibilidade e realizam parte da transformação perspectiva. O passo de mapeamento de textura é responsável pela distorção perspectiva remanescente e por algumas transformações de escala e rotação.

2.5

Equações de Pré Warping para Texturas com Relevo

O uso de imagens de projeção paralela como primitivas de modelagem apresenta algumas vantagens. Por exemplo, é possível beneficiar-se do relacionamento perpendicular entre os raios da amostragem e o suporte de uma imagem de projeção paralela para confeccionar um algoritmo simples capaz de gerar representações baseadas em imagens. Os detalhes de tal algoritmo são apresentados posteriormente.

Dado um modelo de câmera de projeção paralela, as coordenadas de um ponto \dot{x} no espaço Euclideano são dadas por:

$$\dot{x} = \dot{C}_s + \begin{bmatrix} a_{sx} & b_{sx} & c_{sx} \\ a_{sy} & b_{sy} & c_{sy} \\ a_{sz} & b_{sz} & c_{sz} \end{bmatrix} \begin{bmatrix} u_s \\ v_s \\ displ(u_s, v_s) \end{bmatrix} = \dot{C}_s + P'_s \vec{x}'_s \quad (2-8)$$

onde, $displ(u_s, v_s)$ é o deslocamento ortogonal, ou altura, associado ao *pixel* fonte (u_s, v_s) . Expressando-se as coordenadas de \dot{x} tanto numa câmera destino de projeção perspectiva quanto numa câmera fonte de projeção paralela (Figura 2.8), a equação de *warping* 3D torna-se:

$$\begin{aligned} \dot{C}_t + P_t \vec{x}_t t_t(u_t, v_t) &= \dot{x} = \dot{C}_s + P'_s \vec{x}'_s \\ t_t(u_t, v_t) P_t \vec{x}_t &= P'_s \vec{x}'_s + (\dot{C}_s - \dot{C}_t) \\ \vec{x}_t &\doteq P_t^{-1} \left[P'_s \vec{x}'_s + (\dot{C}_s - \dot{C}_t) \right] \end{aligned} \quad (2-9)$$

Novamente, \doteq é uma equivalência projetiva. Deste modo, as equações de *warping* 3D de imagens envolvendo uma imagem fonte de projeção

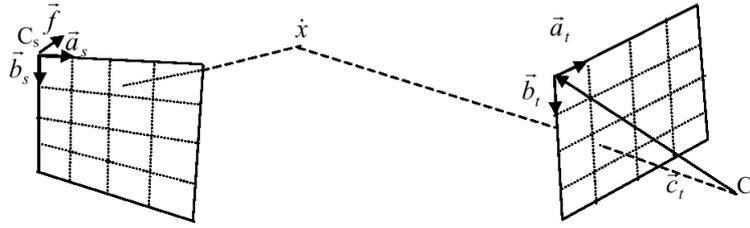


Figura 2.8: Cálculo da projeção do ponto \hat{x} sobre uma câmera destino de projeção perspectiva a partir de suas coordenadas em uma câmera fonte de projeção paralela. (Imagem obtida em [26].)

paralela e uma imagem destino de projeção perspectiva são dadas por:

$$u_t = \frac{Au_s + Bv_s + C + D' \text{displ}(u_s, v_s)}{Iu_s + Jv_s + K + L' \text{displ}(u_s, v_s)} \quad (2-10)$$

$$v_t = \frac{Eu_s + Fv_s + G + H' \text{displ}(u_s, v_s)}{Iu_s + Jv_s + K + L' \text{displ}(u_s, v_s)} \quad (2-11)$$

onde,

$$\begin{aligned} A &= \vec{a}_s \cdot (\vec{b}_t \times \vec{c}_t) & B &= \vec{b}_s \cdot (\vec{b}_t \times \vec{c}_t) & C &= (\dot{C}_s - \dot{C}_t) \cdot (\vec{b}_t \times \vec{c}_t) \\ D' &= \vec{f}_s \cdot (\vec{b}_t \times \vec{c}_t) & E &= \vec{a}_s \cdot (\vec{c}_t \times \vec{a}_t) & F &= \vec{b}_s \cdot (\vec{c}_t \times \vec{a}_t) \\ G &= (\dot{C}_s - \dot{C}_t) \cdot (\vec{c}_t \times \vec{a}_t) & H' &= \vec{f}_s \cdot (\vec{c}_t \times \vec{a}_t) & I &= \vec{a}_s \cdot (\vec{a}_t \times \vec{b}_t) \\ J &= \vec{b}_s \cdot (\vec{a}_t \times \vec{b}_t) & K &= (\dot{C}_s - \dot{C}_t) \cdot (\vec{a}_t \times \vec{b}_t) & L' &= \vec{f}_s \cdot (\vec{a}_t \times \vec{b}_t) \end{aligned}$$

As expressões que representam o mapeamento de textura são obtidas das Equações (2-10) e (2-11) anulando-se o fator $\text{displ}(u_s, v_s)$ para todos os *pixels* fonte. Portanto, o problema de encontrar o *warp* almejado pode ser modelado como:

$$\frac{Au_i + Bv_i + C}{Iu_i + Jv_i + K} = \frac{Au_s + Bv_s + C + D' \text{displ}(u_s, v_s)}{Iu_s + Jv_s + K + L' \text{displ}(u_s, v_s)} \quad (2-12)$$

$$\frac{Eu_i + Fv_i + G}{Iu_i + Jv_i + K} = \frac{Eu_s + Fv_s + G + H' \text{displ}(u_s, v_s)}{Iu_s + Jv_s + K + L' \text{displ}(u_s, v_s)} \quad (2-13)$$

O pré *warp* associado à fatoração ideal é então obtido resolvendo-se o sistema acima para u_i e v_i :

$$u_i = \frac{u_s + k_1 \text{displ}(u_s, v_s)}{1 + k_3 \text{displ}(u_s, v_s)} \quad (2-14)$$

$$v_i = \frac{v_s + k_2 \text{displ}(u_s, v_s)}{1 + k_3 \text{displ}(u_s, v_s)} \quad (2-15)$$

onde, k_1 , k_2 e k_3 são constantes para uma dada configuração das câmeras fonte e destino e, junto com $displ(u_s, v_s)$, determinam a quantidade de deslocamento nas coordenadas dos *pixels* da imagem fonte.

As Equações (2-14) e (2-15) são denominadas *equações de pré warping para texturas com relevo*. Tal fatoração prova ter muitas propriedades desejáveis. Em particular, as coordenadas de um *pixel* na imagem intermediária² podem ser calculadas de forma independente umas das outras, isto é, u_i não depende de v_s e v_i não depende de u_s . Também, quando $displ(u_s, v_s) = 0$ nenhum cálculo é requerido.

As divisões por *pixel* podem ser evitadas quantizando-se os valores de deslocamento (num passo de pré processamento) e armazenando o recíproco do denominador da Equação (2-14) em uma tabela. Estas otimizações são discutidas em maiores detalhes no capítulo seguinte.

2.5.1

Os Coeficientes das Equações de Pré Warping

A quantidade de deslocamento $(\Delta u, \Delta v)$ a ser aplicado a um *pixel* fonte não depende dos parâmetros da câmera destino exceto por seu COP [26]. Por esta razão, é possível especificar livremente os parâmetros \vec{a}_t , \vec{b}_t e \vec{c}_t que definem uma câmera destino temporária utilizada somente para propósitos de pré *warp* e que geralmente difere da câmera virtual utilizada para visualização da cena final.

Escolhendo-se apropriadamente tais parâmetros, é possível eliminar vários dos coeficientes nas Equações (2-12) e (2-13) forçando-se as correspondentes triplas de produtos terem a forma $\vec{v} \cdot (\vec{v} \times \vec{w})$ ou $\vec{w} \cdot (\vec{v} \times \vec{w})$. Tal procedimento conduz a uma drástica simplificação das expressões utilizadas para computar k_1 , k_2 e k_3 .

Por exemplo, as condições $\vec{a}_t = \alpha \vec{a}_s$, $\vec{b}_t = \beta \vec{b}_s$ e $\vec{c}_t = \gamma(\dot{C}_s - \dot{C}_t)$, para $\alpha, \beta, \gamma \in \mathbb{R}$ e não nulos, eliminam os coeficientes B , C , E , G , I e J e são trivialmente satisfeitas permitindo-se que os suportes de ambas as imagens fonte e destino coincidam, incluindo suas origens e seus vetores base (Figura 2.9).

Logo, os subscritos de todos os vetores podem ser retirados sem

²Imagem resultante do processamento horizontal da imagem fonte.

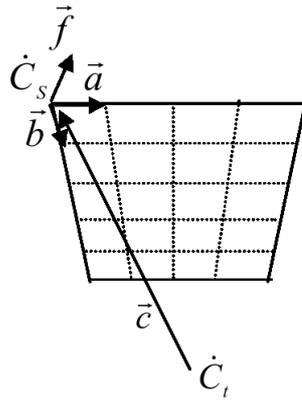


Figura 2.9: Câmeras de projeção paralela e perspectiva compartilhando o mesmo plano de projeção (origem, vetores \vec{a} e \vec{b}). (Imagem obtida em [26].)

risco de confusão e os coeficientes das Equações (2-14) e (2-15) tornam-se: $k_1 = \frac{\vec{f} \cdot (\vec{b} \times \vec{c})}{\vec{a} \cdot (\vec{b} \times \vec{c})}$, $k_2 = \frac{\vec{f} \cdot (\vec{c} \times \vec{a})}{\vec{a} \cdot (\vec{b} \times \vec{c})}$ e $k_3 = \frac{1}{(\vec{c} \cdot \vec{f})}$.

2.6

Ordem Compatível de Oclusão

Ao realizar o *warping* 3D surgirá sempre o problema de conflito de *pixels* sobre uma mesma área. Isto ocorre porque a profundidade de cada *pixel* da imagem pertence a pontos de superfícies em profundidades diferentes no mundo real. Assim sendo, um *pixel* pode deslocar-se mais do que o seu vizinho no processo de *warping*.

Além disso, também existe o problema dos espaços não preenchidos: um *pixel* p_{ij} sofre um deslocamento para uma determinada posição, mas nenhum outro *pixel* foi movido para a posição anteriormente ocupada por p_{ij} . Caso estes espaços não sejam corretamente preenchidos com alguma cor, a imagem resultante do *warping* conterà uma série de ruídos.

Para resolver estes problemas Oliveira [26] sugere que seja utilizado o algoritmo que soluciona o problema da ordem compatível de oclusão proposto em [24].

A ordem compatível de oclusão proposta por McMillan [24], equivalente ao algoritmo do pintor, especifica possíveis ordens nas quais os *pixels* de uma imagem de projeção perspectiva com profundidade devem sofrer uma transformação de *warp*, de modo que a visibilidade correta seja alcançada para pontos de vista arbitrários sem explícitas comparações de

profundidade. O Algoritmo 1 descreve o procedimento:

```

algoritmo ordemCompativelOclusao( $i_{ds}$ ,  $K_s$ ,  $i_{dt}$ ,  $K_t$ )
  1 epipolar = A projeção de  $K_t$ .COP sobre  $i_{ds}.U'$ ;
  2 Divida  $i_{ds}.U'$  em, no máximo, quatro quadrantes  $quad_i$ , uti-
    lizando o ponto epipolar como pivô;
  3 Se  $K_t$ .COP está atrás de  $K_s$ .COP então
  4   Para_todo  $quad_i$  faça
  5      $i_{dt}.C' = \{i_{dt}.C'\} \cup \{\text{warp}(i_{ds}.\text{amostras} \in quad_i) \text{ do}$ 
    ponto epipolar em direção às bordas de  $quad_i\}$ ;
  6   fim_para_todo
  7 senão
  8   Para_todo  $quad_i$  faça
  9      $i_{dt}.C' = \{i_{dt}.C'\} \cup \{\text{warp}(i_{ds}.\text{amostras} \in quad_i) \text{ das}$ 
    bordas de  $quad_i$  em direção ao ponto epipolar};
  10  fim_para_todo
  11 fim_se
fim

```

Algoritmo 1: Ordem compatível de oclusão para imagens fonte de projeção perspectiva [24]. **epipolar** é o ponto que divide o suporte da imagem fonte em quadrantes. i_{ds} e i_{dt} são, respectivamente, as imagens fonte e destino. K_s e K_t são, respectivamente, os modelos de câmeras fonte e destino.

Na linha **1** é calculado o ponto de intersecção entre a linha epipolar e o suporte da imagem i_{ds} . Então, este ponto de intersecção, referenciado no pseudo-código por **epipolar**, é utilizado como pivô na divisão do suporte da imagem i_{ds} em, no máximo, quatro quadrantes (linha **2**). Na linha **3** é verificada a posição do COP do modelo de câmera K_t em relação à posição do COP de K_s . Se o primeiro se encontra atrás do segundo, então, para cada quadrante, os *pixels* da imagem i_{ds} devem sofrer uma transformação de *warping* respeitando a seguinte ordem: do ponto **epipolar** em direção às bordas do quadrante em questão (linhas **4**, **5** e **6**). Caso o COP do modelo de câmera K_t se encontre na frente do COP de K_s , então, o *warping* é aplicado das bordas do quadrante em questão em direção ao ponto **epipolar** (linhas **8**, **9** e **10**).

Na Figura 2.10 as regiões de cor branca e cinza representam os quadrantes da imagem fonte e as setas definem a ordem na qual o processo de *warping* é realizado.

A intuição geométrica por detrás do Algoritmo 1 é ilustrada na Figura 2.11 para o caso no qual o COP da imagem destino está atrás do COP da imagem fonte: sempre que múltiplas amostras se localizam ao longo

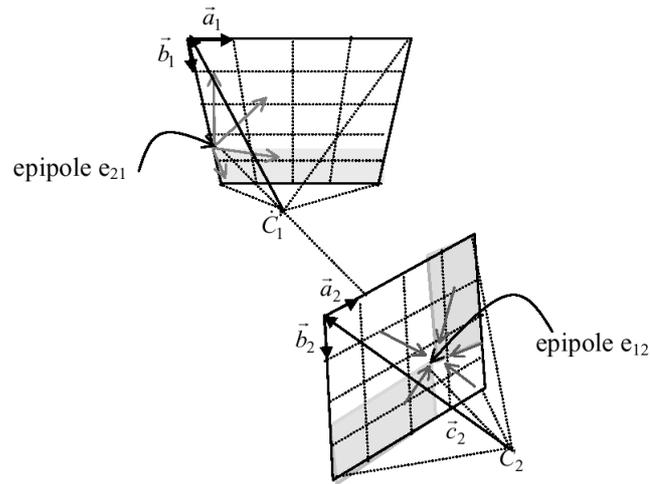


Figura 2.10: Ordem compatível de oclusão. A linha epipolar divide a imagem fonte em no máximo quatro regiões. As setas indicam a ordem na qual os *pixels* devem ser processados. (Imagem obtida em [26].)

do mesmo raio, aquela cujo correspondente *pixel* está mais distante da linha epipolar é a amostra mais próxima do COP da imagem destino e, portanto, poderia seguramente sobrescrever as amostras processadas anteriormente. Assim, os *pixels* na imagem fonte devem ser processados da epipolar em direção às bordas da imagem fonte.

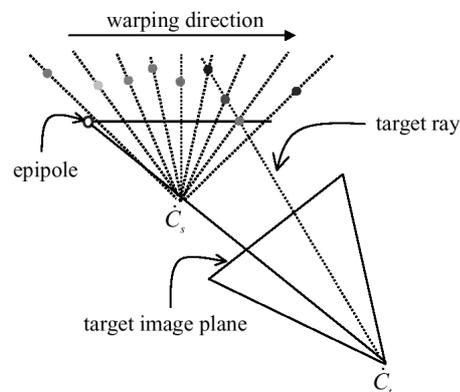


Figura 2.11: Ordem compatível de oclusão: intuição geométrica. O COP da imagem destino está atrás do COP da imagem fonte. Sempre que múltiplas amostras se localizam ao longo do mesmo raio, aquela cuja projeção no suporte da imagem fonte está mais distante da linha epipolar é a mais próxima do COP da imagem destino. (Imagem obtida em [26].)

Quando imagens de projeção paralela com profundidade são utilizadas como imagens fonte, a ordem de oclusão para estas imagens é sempre das bordas da imagem fonte em direção à epipolar [26]. Veja o Algoritmo 2.

```

algoritmo ordemCompativelOclusao( $i_{ds}, i_{dt}, K_t$ )
  1 epipolar = A projeção de  $K_t \cdot \text{COP}$  sobre  $i_{ds} \cdot U'$ ;
  2 Divida  $i_{ds} \cdot U'$  em, no máximo, quatro quadrantes  $quad_i$ , uti-
    lizando o ponto epipolar como pivô;
  3 Para_todo  $quad_i$  faça
  4    $i_{dt} \cdot C' = \{i_{dt} \cdot C'\} \cup \{\text{warp}(i_{ds} \cdot \text{amostras} \in quad_i)$  das
    bordas de  $quad_i$  em direção ao ponto epipolar};
  5 fim_para_todo
fim

```

Algoritmo 2: Ordem compatível de oclusão para imagens fonte de projeção paralela [24]. *epipolar* é o ponto que divide o suporte da imagem fonte em quadrantes. i_{ds} e i_{dt} são, respectivamente, as imagens fonte e destino. K_t é o modelo de câmera destino.

2.7

Discussão

Um mapa de *warp* 2D $w_d : U' \rightarrow W \subset \mathbb{R}^2$ é uma transformação geométrica que deforma o suporte de uma imagem conseqüentemente produzindo uma nova imagem. Utilizando-se uma textura com relevo $\{i_d, K\}$ como entrada para w_d é possível obter uma visualização correta do conteúdo de i_d visualizada de diferentes pontos de vista.

Para que este processo possa ser utilizado em aplicações interativas é essencial que exista uma maneira eficiente de implementá-lo. A partir da formulação definida pelas Equações (2-12) e (2-13) é possível fatorar o processo de *warping* em dois passos: uma transformação perspectiva planar e um deslocamento por *pixel* na direção da linha epipolar [26]. Uma das principais vantagens de tal fatoração é a realização do passo de transformação perspectiva planar através da operação de mapeamento de textura implementada em *hardware*. Além disso, a equação resultante tem uma estrutura unidimensional muito simples, o que garante a eficiência do processo.

O pré *warp* é feito no espaço da textura ao longo das linhas e colunas das imagens fonte e intermediária. Como visto na Seção 2.4, as equações de pré *warping* resolvem visibilidade e realizam parte da transformação perspectiva. A informação a respeito da posição e orientação da câmera, essencial para a transformação, é adicionada ao processo via os coeficientes k_i .

Sempre que $displ(u_s, v_s) = 0$ nenhuma transformação é necessária.

Deste modo, quando um grande número de *pixels* têm deslocamento igual a zero é possível aumentar a velocidade de processamento evitando-se transformações desnecessárias.

Porém, o mapeamento de textura com relevo não é um mapeamento um para um e, portanto, sempre surgirá o problema de conflito de *pixels* sobre uma mesma área. Para resolver este problema os *pixels* durante o *warping* são processados na ordem compatível de oclusão. Este método, proposto em [24], especifica a ordem na qual os *texels* de uma textura com relevo devem ser processados de maneira que a visibilidade correta seja alcançada para pontos de vista arbitrários. Este procedimento é descrito pelo Algoritmo 2.