

3 Rastreamento

Em Realidade Virtual (RV) e Realidade Aumentada (RA), o processo de acompanhamento das coordenadas de um objeto em tempo real chama-se **rastreamento**. Os objetos mais comuns a serem rastreados são os capacetes de RV e dispositivos manuais de interação tri-dimensional. Em muitos casos, a posição (três graus de liberdade) e a orientação (três graus de liberdade) do objeto devem ser recuperados, dando origem a sistemas de rastreamento com seis graus de liberdade.

Para um sistema de RV e RA, o rastreamento deve possuir as seguintes características:

- **Velocidade:** as aplicações devem funcionar a taxas de 30 a 40 Hz, para não atrapalharem a sensação de imersão;
- **Precisão:** as aplicações de RA, principalmente, necessitam que os objetos virtuais casem com os objetos reais, no processo de registro [3, 4];
- **Ruído:** quando não há movimento do objeto sendo rastreado, o mesmo deve estar parado no mundo virtual;
- **Robustez:** pequenos movimentos no mundo real devem gerar pequenos movimentos no mundo virtual. Os pontos fora da curva devem ser devidamente descartados;
- **Mobilidade:** o dispositivo que está sendo rastreado e o rastreador não devem atrapalhar o movimento do usuário;
- **Predição:** a fim de evitar atrasos na renderização, técnicas podem ser utilizadas para prever a posição futura do objeto sendo rastreado.

O rastreamento óptico pode oferecer vantagens sobre os métodos magnéticos ou mecânicos. Ele permite que sejam utilizados dispositivos sem fio, facilitando a mobilidade do usuário, rastreamento de múltiplos objetos, oferece pouca sensibilidade a interferências externas e pode ser utilizado com câmeras digitais convencionais, tornando-se um produto de fácil acesso. A maioria dos

sistemas ópticos utiliza objetos cobertos por materiais retroreflexivos e receptores com filtros infravermelhos. Na Figura 15 podemos observar um capacete de RV com 3 esferas cobertas com esse tipo de material que serão rastreadas para determinar a posição e orientação da cabeça do usuário.



Figura 15: HMD rastreado por esferas retroreflexivas.

A Figura 16 ilustra a visualização de um conjunto de esferas retroreflexivas utilizando-se uma câmera convencional, uma câmera com sensores infravermelhos dentro de uma sala com lâmpadas incandescentes e outra em uma sala com lâmpadas fluorescentes. O problema ilustrado é que a luz amarela contém a componente infravermelha, atrapalhando a captura da imagem.

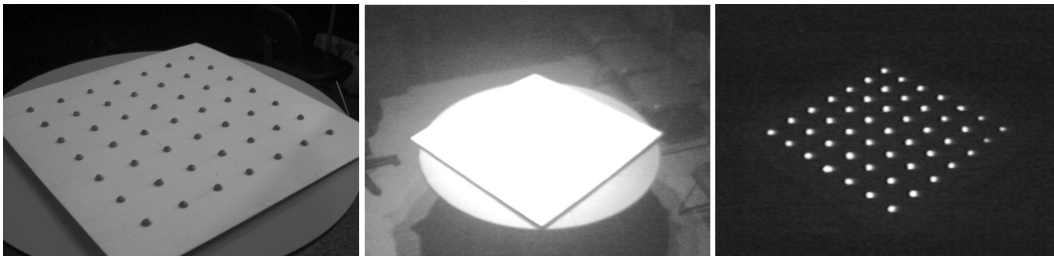


Figura 16: Esferas retroreflexivas: (i) Câmera convencional; (ii) Câmera com filtro infravermelho em ambiente com lâmpadas incandescentes; (iii) Câmera com filtro infravermelho em ambiente com lâmpadas fluorescentes.

A abordagem mais simples em rastreamento de dispositivos de interação é a utilização de somente um marcador (geralmente esférico). Essa abordagem permite que apenas 3 graus de liberdade (posição) sejam recuperados, utilizando mais de uma câmera. Outra abordagem existente é a utilização de marcadores com uma posição tri-dimensional conhecida. Utilizando propriedades projetivas invariantes, apresentadas em [31], é possível comparar a posição dos pontos da

imagem com padrões em uma base de dados. O padrão que melhor representa o padrão do modelo é escolhido. É possível determinar a posição e a orientação do dispositivo utilizando imagens de mais de uma câmera.

Na presente dissertação, o dispositivo proposto será composto por apenas uma esfera. Ela será rastreada por quatro câmeras a fim de prover redundância. Com apenas uma câmera, é possível determinar um segmento de reta que parte da sua posição e intercepta a imagem no ponto detectado. Com duas câmeras é possível determinar, com uma certa aproximação, a interseção entre dois segmentos e estabelecer a posição da esfera no espaço. Essa aproximação existe porque, devido a erros de calibração, as retas não se interceptam realmente. A heurística empregada para determinar a “interseção” foi a de se utilizar o ponto mais próximo entre as duas retas. Com mais de duas câmeras, o resultado pode ser estabelecido com mais precisão, além do sistema permitir que a visão de algumas câmeras seja bloqueada em algum momento, provendo maior mobilidade ao usuário.

3.1. Calibração da câmera

A etapa inicial no processo de rastreamento feito por câmeras digitais consiste na calibração das câmeras. Calibrar uma câmera é encontrar parâmetros (vistos mais adiante) que permitam definir uma matriz de transformação que levam os pontos do mundo tridimensional para o plano da imagem gerada por aquela câmera.

Existem diversas técnicas de calibração de câmeras descritas na literatura, sendo que o algoritmo utilizado nesta tese foi o algoritmo proposto por Tsai em [29, 30].

3.1.1. Método de Tsai

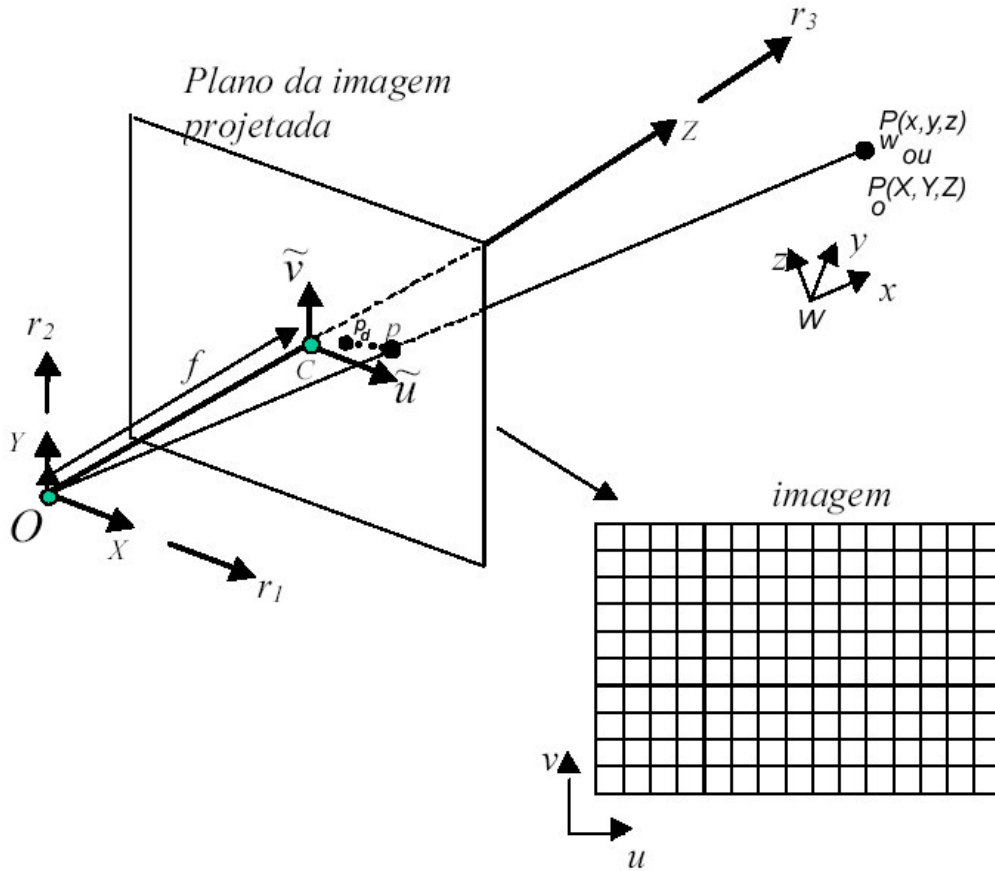


Figura 17: Modelo de câmera “pinhole”.

A Figura 17 ilustra o modelo de câmera “pinhole” utilizado para calibração. (x, y, z) é a coordenada do ponto P no sistema de coordenadas do mundo 3D. (X, Y, Z) é a coordenada do ponto no sistema de coordenadas da câmera, com origem no ponto O , que é o seu centro óptico, e com seu eixo z coincidindo com o eixo óptico. (\tilde{u}, \tilde{v}) é o sistema de coordenadas da imagem centralizado em C (interseção entre o eixo focal e o plano da imagem) e paralelo ao eixo (X, Y) . f é a distância focal, ou a distância entre o plano da imagem e o centro óptico. $p(\tilde{u}, \tilde{v})$ é a coordenada na imagem do ponto (X, Y, Z) se o modelo de câmera fosse perfeito e não houvesse distorção. $p_d(\tilde{u}_d, \tilde{v}_d)$ é o ponto real na imagem onde (X, Y, Z) é projetado. Contudo, como o sistema do computador é discretizado em *pixels*, alguns parâmetros que relacionam a coordenada do ponto na imagem com a coordenada (u, v) devem ser especificados e calibrados. Os

quatro passos da transformação total do ponto (x, y, z) em (u, v) estão ilustrados na Figura 18.

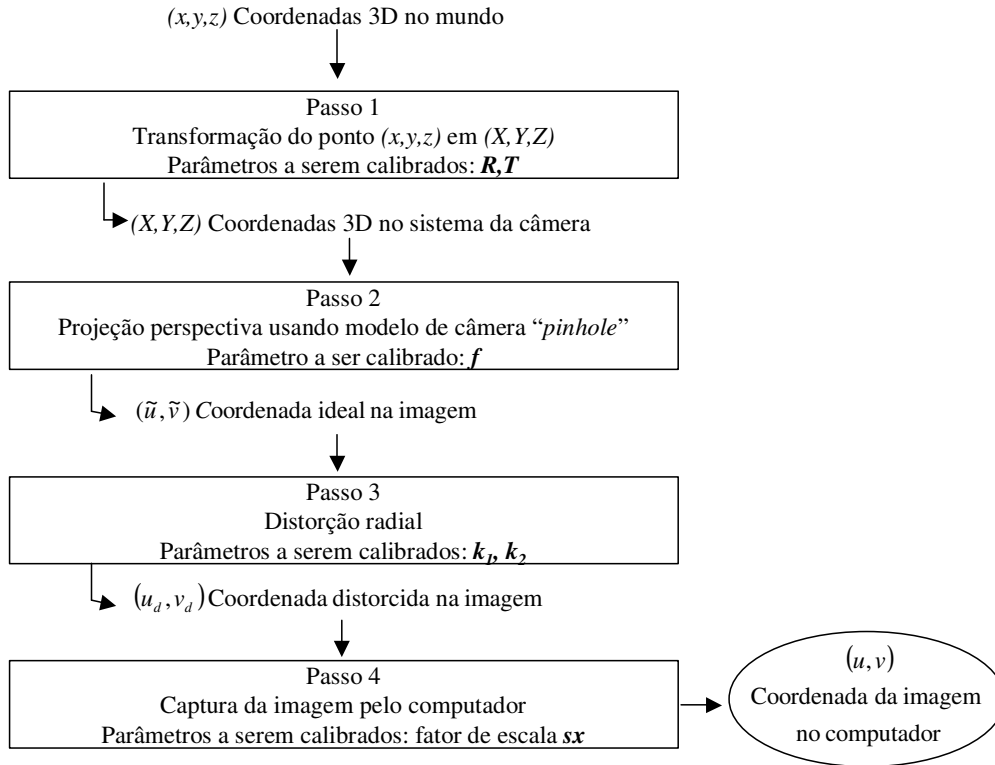


Figura 18: Os quatro passos da transformação de coordenadas do mundo em coordenadas da imagem no computador

Passo 1: Transformação do ponto em coordenadas do mundo (x, y, z) para as coordenadas do sistema da câmera (X, Y, Z)

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T, \quad (3.1)$$

onde R é uma matriz de rotação 3×3

$$R \equiv \begin{bmatrix} \vec{r}_1 \\ \vec{r}_2 \\ \vec{r}_3 \end{bmatrix} = \begin{bmatrix} r_{1x} & r_{1y} & r_{1z} \\ r_{2x} & r_{2y} & r_{2z} \\ r_{3x} & r_{3y} & r_{3z} \end{bmatrix}, \quad (3.2)$$

e T é um vetor de translação

$$T \equiv \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}. \quad (3.3)$$

Os parâmetros a serem calibrados são R e T . Convém observar que o ponto $T = (t_x, t_y, t_z)$ representa a origem dos sistema do mundo escrita em coordenadas da câmera. As matrizes R e T podem ser vistas como uma única matriz $[R|T]$

$$[R|T] = \begin{bmatrix} \vec{r}_1 & t_x \\ \vec{r}_2 & t_y \\ \vec{r}_3 & t_z \end{bmatrix} = \begin{bmatrix} r_{1x} & r_{1y} & r_{1z} & t_x \\ r_{2x} & r_{2y} & r_{2z} & t_y \\ r_{3x} & r_{3y} & r_{3z} & t_z \end{bmatrix} \quad (3.4)$$

Passo 2: Transformação das coordenadas da câmera (X, Y, Z) para as coordenadas ideais da imagem (\tilde{u}, \tilde{v}) usando projeção perspectiva

$$\tilde{u} = f \frac{X}{Z} \quad (3.5)$$

$$\tilde{v} = f \frac{Y}{Z}. \quad (3.6)$$

O parâmetro a ser calibrado é a distância focal f .

Passo 3: Distorção radial das lentes: os parâmetros a serem calibrados são os coeficientes de distorção. A modelagem das distorções da lente pode ser encontrada em [17].

Passo 4: Transformação das coordenadas reais da imagem em coordenadas da imagem no computador levando em consideração os tamanhos dos *pixels* da tela.

Os parâmetros listados acima podem ser divididos em duas categorias: *intrínsecos* e *extrínsecos*. Os parâmetros intrínsecos da câmera são sua distância focal (f), os coeficientes de distorção da lente, o fator de escala (tamanho de um *pixel* no plano de captura da imagem) e as coordenadas da origem (u_o, v_o) no plano da imagem. Os parâmetros extrínsecos são a matriz de rotação R e o vetor de translação T .

O modelo ilustrado acima pode ser simplificado se não levarmos em consideração os passos 3 e 4, admitindo, portanto, que não existe distorção radial e que o centro da imagem no computador coincide com o centro óptico.

As equações (3.4), (3.5) e (3.6) podem ser compatibilizadas a fim de produzir uma transformação direta do ponto (x, y, z) do mundo no ponto (u, v) na imagem

$$\begin{bmatrix} us \\ vs \\ s \end{bmatrix} = \begin{bmatrix} f\vec{r}_1 & ft_x \\ f\vec{r}_2 & ft_y \\ \vec{r}_3 & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (3.7)$$

onde (us, vs, s) são coordenadas homogêneas. Multiplicando, encontramos:

$$u - u_0 = f \frac{r_{1x}x + r_{1y}y + r_{1z}z + t_x}{r_{3x}x + r_{3y}y + r_{3z}z + t_z} \quad (3.8)$$

e

$$v - v_0 = f \frac{r_{2x}x + r_{2y}y + r_{2z}z + t_y}{r_{3x}x + r_{3y}y + r_{3z}z + t_z}. \quad (3.9)$$

Uma observação a ser feita é que a escala existente na imagem está embutida na distância focal f .

Na presente dissertação, o método de Tsai coplanar foi utilizado, isto é, o método que encontra os parâmetros baseado em pontos do mundo tridimensional contidos no plano $z = 0$.

A partir das equações (3.8) e (3.9), assumindo $z = 0$ para todos os pontos, temos

$$u - u_0 = \tilde{u} = f \frac{r_{1x}x + r_{1y}y + t_x}{r_{3x}x + r_{3y}y + t_z} \quad (3.10)$$

e

$$v - v_0 = \tilde{v} = f \frac{r_{2x}x + r_{2y}y + t_y}{r_{3x}x + r_{3y}y + t_z}. \quad (3.11)$$

Dividindo (3.10) por (3.11) e em seguida o numerador e o denominador da razão da direita por t_y , temos:

$$\frac{r_{1x}}{t_y} x_i \tilde{v}_i + \frac{r_{1y}}{t_y} y_i \tilde{v}_i - \frac{r_{2x}}{t_y} x_i \tilde{u}_i - \frac{r_{2y}}{t_y} y_i \tilde{u}_i + \frac{t_x}{t_y} \tilde{v}_i = \tilde{u}_i. \quad (3.12)$$

O indicador i subscrito em cada variável corresponde aos vários pontos amostrados e utilizados no método Tsai.

Obtemos com isso um sistema linear $Au = b$, onde A é uma matriz $n \times 5$ e cada linha A_i é dada por $(x_i \tilde{v}_i, y_i \tilde{v}_i, -x_i \tilde{u}_i, -y_i \tilde{u}_i, \tilde{v}_i)$, u sendo um vetor dado por

$$(U_1 \ U_2 \ U_3 \ U_4 \ U_5) = \begin{pmatrix} \frac{r_{1x}}{t_y} & \frac{r_{1y}}{t_y} & \frac{r_{2x}}{t_y} & \frac{r_{2y}}{t_y} & \frac{t_x}{t_y} \end{pmatrix} \quad (3.13)$$

e cada elemento do vetor b sendo dado por \tilde{u}_i .

Como \vec{r}_1 , \vec{r}_2 e \vec{r}_3 são ortonormais, e designando $\alpha = \frac{r_{1z}}{t_y}$ e $\beta = \frac{r_{2z}}{t_y}$, temos

que

$$\begin{cases} \alpha\beta = -U_1U_3 - U_2U_4 \\ \alpha^2 + U_1^2 + U_2^2 = \beta^2 + U_3^2 + U_4^2 \end{cases} \quad (3.14)$$

Resolvendo o sistema acima, calcula-se t_y a partir de

$$t_y^2 = \frac{U - \sqrt{U^2 - 4(U_1U_4 - U_2U_3)^2}}{2(U_1U_4 - U_2U_3)^2} \quad (3.15)$$

onde $U = U_1^2 + U_2^2 + U_3^2 + U_4^2$.

Portanto, com estes valores já definidos e utilizando a equação 3.13, é possível determinar os valores de r_{1x} , r_{1y} , r_{2x} , r_{2y} e t_x . Pelo fato dos vetores \vec{r}_1 e \vec{r}_2 terem normais iguais a um, encontram-se também os valores de r_{1z} e r_{2z} , e conseqüentemente calcula-se \vec{r}_3 , pois \vec{r}_1 , \vec{r}_2 e \vec{r}_3 são ortonormais. Finalmente, usando os valores já encontrados e as equações (3.10) e (3.11), calculamos f e t_z . Pode haver a necessidade de se ajustar os sinais que inicialmente assumimos como positivos, como, por exemplo, t_y .

Em [26], Szenberg apresenta um estudo de métodos de calibração de câmera, onde ele conclui que o método de Tsai não apresenta bons resultados quando o número de pontos de calibração são poucos. Para resolver esse problema, ele utiliza uma homografia (transformação projetiva planar), na qual é definido um mapeamento 2D para 2D, de um plano contido no mundo real para o plano da imagem, para estimar uma quantidade maior de pontos para serem utilizados no Tsai.

3.2.

Pré-processamento

Antes de se calibrar a câmera é necessário um pré-processamento na imagem a fim de se encontrar os objetos a serem rastreados mais facilmente.

A primeira etapa do pré-processamento é a aplicação de um filtro digital para eliminação de pequenos detalhes e ruído que existam na imagem. Um filtro adequado para esta tarefa é o filtro de suavização Gaussiano. Esse filtro é uma

convolução bidimensional cujo núcleo possui algumas propriedades especiais. A forma da distribuição gaussiana unidimensional é dada por:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}},$$

onde σ é o desvio padrão da distribuição e assumindo que a distribuição tem uma média igual a zero. Um exemplo desta distribuição está ilustrado na Figura 19.

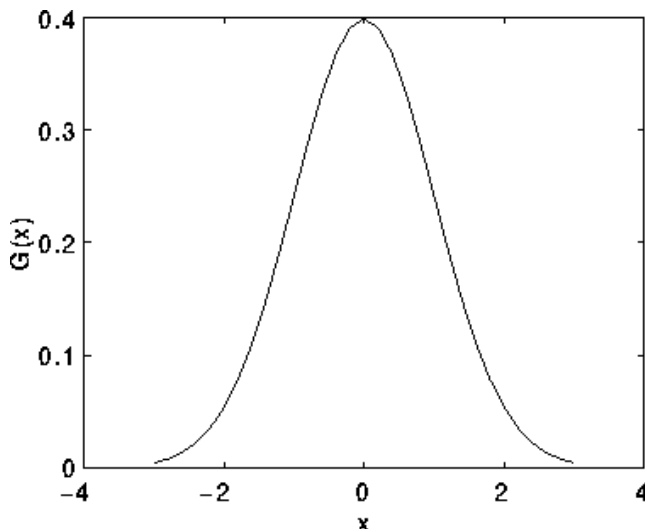


Figura 19: Distribuição gaussiana unidimensional com média 0 e $\sigma = 1$.

Em 2D, uma curva gaussiana isotrópica, isto é, circularmente simétrica, possui a forma:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

Essa distribuição é ilustrada na Figura 20.

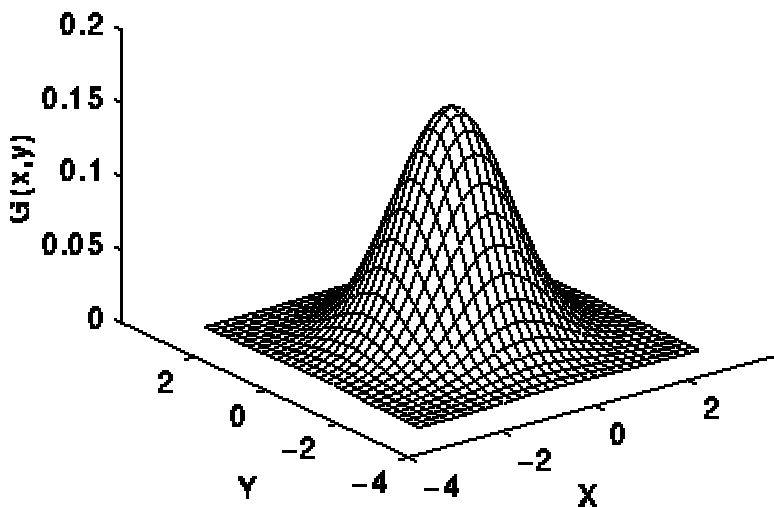


Figura 20: Distribuição gaussiana bidimensional com média (0,0) e $\sigma = 1$.

A idéia da suavização gaussiana é a utilização da distribuição 2D como uma função de “espalhamento pontual”, e isso é alcançado através da convolução. Como a imagem é armazenada com uma coleção de *pixels* discretos, é necessária a construção de uma aproximação discreta da função gaussiana antes de se aplicar a convolução. Na teoria, a distribuição gaussiana é não nula em todos os pontos, exigindo que o núcleo da convolução seja infinitamente grande, mas na prática, a distribuição é efetivamente zero a mais de três desvios padrão da média. Logo, é possível quebrar o núcleo a partir desse ponto. A Figura 21 mostra um núcleo de convolução com valores inteiros que aproxima uma distribuição gaussiana com um desvio padrão $\sigma = 1$.

	1	4	7	4	1
	4	16	26	16	4
$\frac{1}{273}$	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

Figura 21: Aproximação discreta da função gaussiana com $\sigma = 1$.

Uma vez tendo-se calculado um núcleo satisfatório, a suavização gaussiana é aplicada usando métodos de convolução conhecidos [12]. A Figura 22 mostra o resultado da aplicação da suavização gaussiana. A imagem fica mais “borrada”, porém os pontos brancos estão mais conectados ao seu centro.

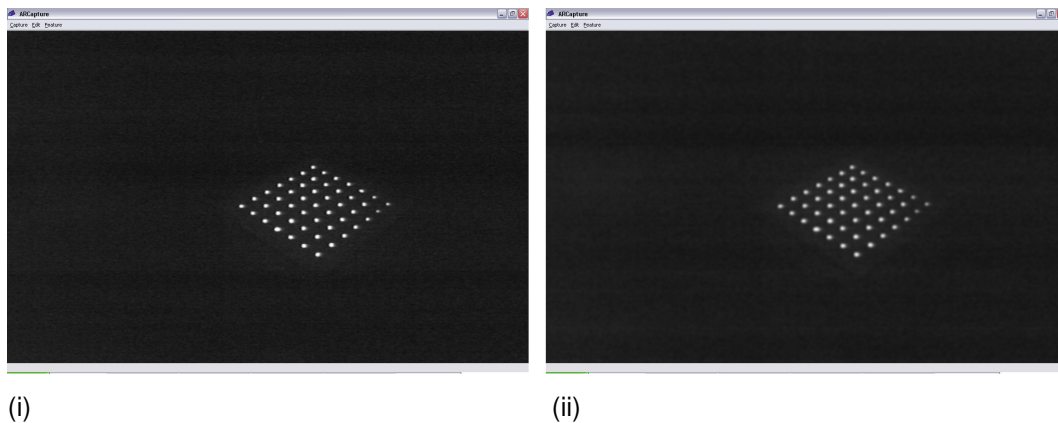


Figura 22: Filtragem gaussiana: (i) imagem original; (ii) filtro gaussiano.

A próxima etapa do pré-processamento é a transformação da imagem em tons de cinza em uma imagem binária, com uma operação denominada *segmentação* ou *threshold*. Na verdade, a operação necessária é bastante simples e chama-se *threshold* binário.

O *threshold* binário funciona da seguinte maneira

$$dst(x, y) = \begin{cases} valorMax, & se\ src(x, y) > threshold \\ 0, & caso\ contrário \end{cases}, \quad (3.16)$$

onde

$dst(x, y)$ valor da imagem destino,

$src(x, y)$ valor da imagem de origem,

$valorMax$ valor para substituição da cor do *pixel* da imagem de origem,

$threshold$ limite para substituição da cor na imagem de origem.

Como observado na Figura 23, antes de se executar o *threshold* definido pela equação (3.16), a imagem precisou ser invertida.

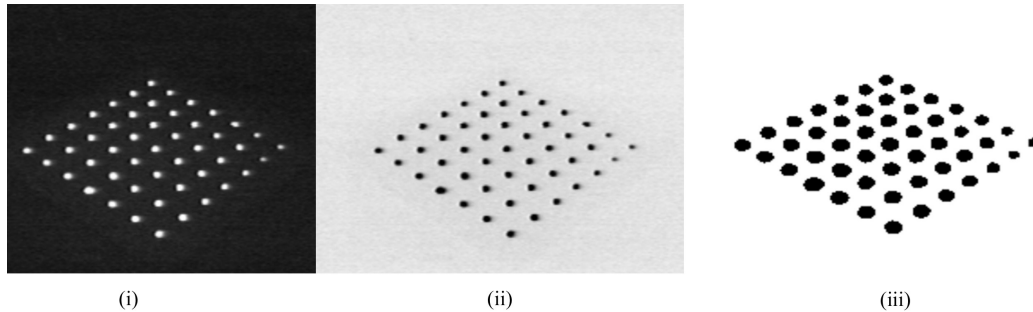


Figura 23: *Segmentação*: (i) imagem original; (ii) imagem invertida; (iii) imagem segmentada

Após a etapa de pré-processamento, a imagem está pronta para que possam ser extraídos os marcadores para calibração e para que o objeto desejado seja rastreado. No caso da implementação desta dissertação, todos os objetos envolvidos são esferas, logo, algum mecanismo de extração de elipses deve ser executado. É o que será visto na seção a seguir.

3.3. Extração de elipses

A fim de se extrair as elipses ou outros objetos de uma imagem, é necessária a separação das componentes conexas.

A identificação de componentes conexas funciona percorrendo uma imagem, *pixel* por *pixel* (de cima para baixo e da esquerda para direita, por exemplo) a fim de identificar as regiões de *pixels* conexas, isto é, as regiões onde *pixels* adjacentes compartilham os mesmos valores de intensidade V . Em uma

imagem binária, temos $V = \{1\}$, contudo em uma imagem em escala de cinza, V possuirá um conjunto de valores, por exemplo: $V = \{51, 52, 53, \dots, 77, 78, 79, 80\}$. O algoritmo de identificação de componentes conexas funciona com imagens binárias e em tons de cinza e com vários tipos de conectividade entre *pixels*.

A conectividade entre *pixels* descreve a relação entre dois ou mais *pixels*. Para que dois *pixels* estejam conectados, eles precisam validar algumas condições relativas a brilho do *pixel* e adjacência. Primeiramente, seus valores devem estar no mesmo conjunto de valores V .

Para formular o critério de adjacência, deve-se primeiro introduzir o conceito de *vizinhança*. Para um *pixel* p com coordenadas (x, y) o conjunto de *pixels* dados por:

$$N_4(p) = \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\}$$

é chamado de sua 4-vizinhança. Sua 8-vizinhança é definida como:

$$N_8(p) = N_4 \cup \{(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)\}$$

A partir desses conjuntos podemos definir o conceito de 4-conectividade e 8-conectividade. Dois *pixels* p e q , ambos tendo valor no mesmo conjunto V , são 4-conectados se q pertence ao conjunto $N_4(p)$ e 8-conectados se q pertence ao conjunto $N_8(p)$. A Figura 24 mostra duas componentes conexas baseadas na 4-conectividade.

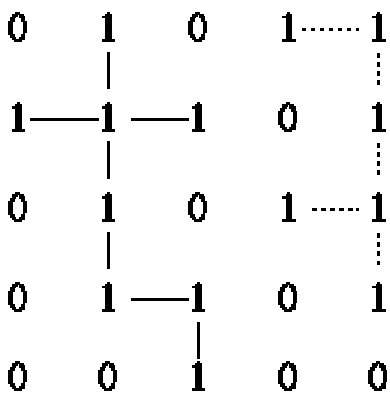


Figura 24: Duas componentes conexas baseado na 4-conectividade.

Para descrever o algoritmo de identificação de componentes conexas, admitimos que a imagem sendo analisada é binária e a conectividade utilizada é a 8.

O algoritmo de identificação de componentes conexas percorre a imagem por linha até encontrar um ponto p com valor igual a 1. Quando isso acontece, o algoritmo examina os quatro pontos vizinhos de p que já foram encontrados durante o rastreo da imagem, isto é, os vizinhos (i) à esquerda de p , (ii) acima de p , (iii e iv) as suas duas diagonais superiores. Baseado nessa informação, a identificação ocorre como a seguir:

- se todos os quatros vizinhos são 0, associe um novo identificador a p , senão
- se somente um vizinho é 1, associe o seu identificador a p , senão
- se um ou mais vizinhos são 1, associe um dos identificadores a p e anote as equivalências.

Após completar todo o rastreo da imagem, os pares de identificadores equivalentes são classificados por classes de equivalência e um identificador é associado a cada classe. Como passo final, um novo rastreo é feito na imagem, onde cada identificador é substituído pelo identificador da sua classe de equivalência.

Depois de identificadas as componentes conexas, detectam-se quais dos conjuntos de *pixels* são elipses, a fim de remover possíveis erros na imagem, como a mão do usuário, ou outro objeto que pode vir a surgir. A posição dos marcadores pode ser aproximada pelos centros das elipses encontradas. O método de extração de elipses utilizado foi o de mínimos-quadrado [22].

De posse das elipses, é necessária uma etapa da identificação dos objetos para associação da sua posição na imagem com sua posição no padrão de calibração. No desenvolvimento dessa dissertação, a identificação foi feita manualmente, pois a mesma era executada uma única vez para a mesma configuração de câmeras. Para o rastreamento do objeto, uma única esfera foi utilizada, logo não houve necessidade de identificação.

3.4.

Posicionamento da esfera rastreada

A partir das matrizes R e T obtidas pela calibração de uma câmera, é possível determinar a posição tridimensional da esfera no plano $z = 0$. Com essa

posição, calcula-se um raio que parte da posição da câmera e intercepta o plano, passando pelo centro da esfera.

A fim de determinar a posição real (x, y, z) da esfera, é necessário que haja no mínimo duas câmeras acompanhando-na. De posse do raio que parte das duas câmeras, é possível encontrar a interseção entre elas. Porém, devido a erros provenientes da extração do centro da esfera, dos pixels serem discretos e do próprio método Tsai, a interseção certamente será nula. Logo, o ponto mais próximo entre os dois segmentos de reta deve ser encontrado.

De acordo com a Figura 25, o que queremos encontrar é o ponto médio do vetor w_c .

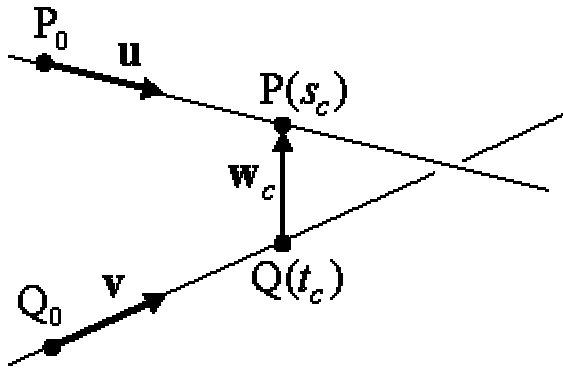


Figura 25: Duas retas não paralelas no espaço. O vetor w_c é o vetor de menor comprimento entre elas.

Podemos escrever as retas como sendo $L_1: P(s) = P_0 + s(P_1 - P_0)$ e $L_2: Q(t) = Q_0 + t(Q_1 - Q_0)$. Seja $w(s, t) = P(s) - Q(t)$ um vetor entre pontos das duas retas. O que queremos encontrar é o vetor $w(s, t)$ de menor comprimento para todo s e t .

Em qualquer espaço n -dimensional, as duas linhas, L_1 e L_2 , estão mais próximas em um único par de pontos, $P(s_c)$ e $Q(t_c)$, onde $w(s, t)$ possui tamanho mínimo. Se L_1 e L_2 não forem paralelas, então o seguimento de reta $\overline{P(s_c)Q(t_c)}$ que une o par de pontos é perpendicular a ambas as linhas ao mesmo tempo. Nenhum outro segmento entre as retas L_1 e L_2 possui esta propriedade. Logo, o vetor $w_c = w(s_c, t_c)$ é perpendicular as direções \vec{u} e \vec{v} , e isto é equivalente a satisfazer as duas equações $u \cdot w_c = 0$ e $v \cdot w_c = 0$.

As duas equações são resolvidas substituindo-se

$$w_c = P(s_c) - Q(t_c) = w_0 + s_c \vec{u} - t_c \vec{v},$$

onde $w_0 = P_0 - Q_0$, nas duas equações lineares:

$$(\vec{u} \cdot \vec{u})s_c - (\vec{u} \cdot \vec{v})t_c = -\vec{u} \cdot w_0$$

$$(\vec{v} \cdot \vec{u})s_c - (\vec{v} \cdot \vec{v})t_c = -\vec{v} \cdot w_0$$

Fazendo $a = \vec{u} \cdot \vec{u}$, $b = \vec{u} \cdot \vec{v}$, $c = \vec{v} \cdot \vec{v}$, $d = \vec{u} \cdot w_0$ e $e = \vec{v} \cdot w_0$, encontramos

$$s_c = \frac{be - cd}{ac - b^2} \quad \text{e} \quad t_c = \frac{ae - bd}{ac - b^2},$$

sempre que o denominador $ac - b^2$ for diferente de zero.

A posição do centro da esfera é igual a $\frac{w_c}{2}$. Os pontos Q_0 e P_0 são as coordenadas das câmeras obtidas pela calibração e os pontos P_1 e Q_1 são os pontos amostrados do centro da esfera.

Quando existem n câmeras capturando, o mesmo algoritmo pode ser aplicado para um combinação $\binom{n}{2}$ de segmentos de reta. Com o conjunto de pontos equidistantes obtidos, pode-se calcular o centro da esfera através da média da posição dos pontos ou descartando os pontos mais distantes.

O algoritmo descrito acima é uma heurística que tenta aproximar a interseção de quatro retas muito próximas que, supostamente, deveriam se interceptar. Essa não é a única heurística existente. Outras maneiras existem, como por exemplo, a utilização do método dos mínimos quadrados para minimizar a distância entre o ponto de interseção e as quatro retas.

3.5. Filtro de Kalman

Um filtro de Kalman [15,33] tem como idéia básica a predição e a correção de predição de um evento. A predição é feita a partir de valores passados e a correção de predição utiliza medições provenientes da observação de um fenômeno. A Figura 26 ilustra o resultado de uma predição, onde o movimento observado difere do movimento previsto.

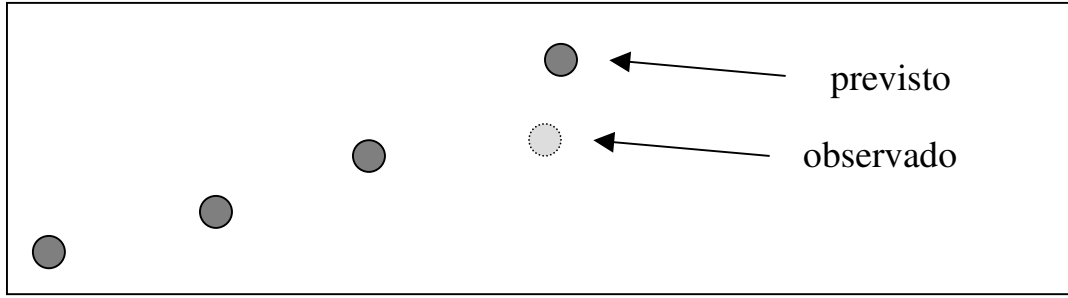


Figura 26: Idéia básica do Filtro de Kalman.

O fenômeno a ser observado e corrigido pelo filtro de Kalman é um processo $x \in \mathfrak{R}^n$ que obedece a uma lei de evolução linear, com a presença de erros aleatórios, ou seja,

$$x_k = \Phi_k x_{k-1} + \varepsilon_k,$$

$\begin{matrix} n \times 1 & n \times n & n \times 1 & n \times 1 \end{matrix}$

e uma medição $z \in \mathfrak{R}^m$ descrita por

$$z_k = H_k x_k + \mu_k, \text{ onde}$$

$\begin{matrix} m \times 1 & m \times n & n \times 1 & m \times 1 \end{matrix}$

Φ_k é a matriz de transição do passo anterior $k-1$ para o passo k , na ausência de ruído no processo,

H_k é a matriz de medição que relaciona o estado x_k à medida z_k e,

$$\begin{cases} \varepsilon_k \sim N\left(0, Q_k\right) \\ \mu_k \sim N\left(0, R_k\right) \end{cases} \text{ são distribuições normais de probabilidade com média } 0 \text{ e}$$

matriz de covariância Q_k e R_k .

O problema a ser resolvido pelo filtro de Kalman é, dada a estimativa \hat{x}_{k-1} do estado anterior, com sua respectiva matriz de covariância \hat{P}_{k-1} , procurar uma solução da forma

$$\hat{x}_k = \tilde{x}_k + K_k (z_k - H_k \tilde{x}_k), \text{ onde } \tilde{x}_k = \Phi_k \hat{x}_{k-1},$$

de tal modo que o valor esperado do erro quadrático $(E(\hat{x}_k - x_k)^2)$ seja mínimo. A matriz K_k é denominada ganho de Kalman.

A solução do filtro de Kalman foi obtida de [33]. Ela é composta de duas etapas: predição e correção. A solução da etapa de predição é

$$(1) \text{ Estimar o processo: } \tilde{x}_k = \Phi_k \hat{x}_{k-1}$$

(2) Estimar a matriz de covariância: $\tilde{P}_k = \Phi_k \hat{P}_{k-1} \Phi_k^T + Q_k$

e a da etapa de correção é

(1) Calcular o ganho de Kalman: $K_k = \tilde{P}_k H_k^T (H_k \tilde{P}_k H_k^T + R_k)^{-1}$

(2) Atualizar a estimativa através da medida z_k : $\hat{x}_k = \tilde{x}_k + K_k (z_k - H_k \tilde{x}_k)$

(3) Atualizar a matriz de covariância: $\hat{P}_k = (I - K_k H_k) \tilde{P}_k$

Dessas fórmulas, o mais importante é concluirmos qual será o comportamento do filtro diante dos valores das matrizes de covariância Q e R . Para isso devemos analisar o comportamento da matriz K . Ela pode ser definida como

$$K = \frac{\Phi_k P_{k-1} \Phi_k^T H_k^T + Q_k H_k^T}{H_k \Phi_k P_{k-1} \Phi_k^T H_k^T + H_k Q_k H_k^T + R_k}$$

Dessa equação concluímos que:

- se $Q \gg R$, então $K = 1/H$ e $\hat{x}_k = z_k$, ou seja, se o erro associado ao processo for muito maior do que o erro associado a medição, o filtro de Kalman utilizará os valores medidos como predição;
- se $R \gg Q$, então $K = 0$, $\hat{x}_k = \tilde{x}_k$, ou seja, se o erro da medição for muito maior, o filtro utilizará os valores corrigidos com base no processo.

Observamos também, que o comportamento do filtro depende muito pouco da matriz de covariância inicial P_0 .

Na presente tese, o filtro de Kalman foi utilizado para estimar o movimento sendo realizado pela esfera rastreada. Para isso, o modelo empregado foi o seguinte:

$$\begin{bmatrix} X \\ Y \\ Z \\ \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}_k = \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}_{k-1} + \varepsilon \quad \text{e} \quad \begin{bmatrix} Z_x \\ Z_y \\ Z_z \\ Z_{\dot{x}} \\ Z_{\dot{y}} \\ Z_{\dot{z}} \end{bmatrix}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}_{k-1} + \mu$$

onde (X, Y, Z) é a posição, $(\dot{X}, \dot{Y}, \dot{Z})$ a velocidade e dt é a variação em segundos, medido a cada passo. No Capítulo 4, serão dados mais detalhes sobre os valores das matrizes de covariância Q e R .

3.6. Algoritmo Proposto

O algoritmo proposto é composto de duas etapas: calibração e rastreamento, sendo que elas possuem alguns procedimentos em comum. Ambas as etapas trabalham com a imagem segmentada, logo, as quatro primeiras tarefas são iguais:

- a) Captura da imagem pela câmera;
- b) Aplicação de filtro gaussiano;
- c) Inversão da imagem;
- d) Segmentação.

A partir desse momento, a imagem segmentada será utilizada pela etapa de calibração, que é feita uma única vez para cada uma das quatro câmeras utilizadas:

- e) Detecção das elipses;
- f) Identificação dos pontos de calibração (etapa manual);
- g) Calibração da câmera;
- h) Gravação dos arquivos de calibração;

Como a etapa de calibração é feita somente uma vez, os arquivos que armazenam as informações das câmeras calibradas serão utilizados posteriormente para recuperação da posição de cada câmera. A etapa de rastreamento possui as seguintes tarefas:

- e) Para cada câmera:
 - f) Detecção e determinação do centro da esfera sendo rastreada;
 - g) Conversão do centro da esfera em coordenadas da imagem para coordenadas do mundo;
 - h) Definição do segmento de reta r_i , que passa pela posição da câmera e pelo centro da esfera obtido por essa câmera;
- i) Para cada par de segmentos de reta i e j :
 - j) Cálculo do ponto P_{ij} mais próximo entre as retas i e j ;
- k) Determinação da média das posições dos pontos P_{ij} .

A média encontrada é o centro da esfera sendo rastreada. Na próxima seção, serão vistos detalhes da implementação e os resultados obtidos.