

### 3

## O Dispositivo de Rastreamento Óptico

Neste capítulo faremos uma descrição detalhada do dispositivo de rastreamento proposto, a qual se dividirá em duas partes. A primeira parte apresentará em detalhes a estrutura e características físicas do dispositivo. A segunda parte descreverá o processo de rastreamento óptico implementado com a finalidade de recuperar a posição 3D de marcadores esféricos que são rastreados dentro do volume de trabalho definido pelo dispositivo.

### 3.1.

#### Projeto Físico do Dispositivo

O dispositivo construído como parte desta dissertação foi projetado com o objetivo de capturar e simular, em essência, a mesma arquitetura e funcionamento que apresentam os dispositivos de rastreamento óptico comerciais utilizados para o rastreamento de áreas amplas, que usam a luz infravermelha como componente elementar para seu funcionamento.

Esses dispositivos comerciais têm em sua arquitetura certas características em comum:

- Uma área física bem definida, onde fica concentrado o campo de visão das câmeras e onde se faz o rastreamento dos marcadores definidos.
- Um conjunto de câmeras, que são colocadas a uma altura moderada da área de rastreamento para não atrapalhar a movimentação que o usuário possa fazer dentro desse espaço.
- Um conjunto de marcadores, que na maioria das vezes são esféricos, e que serão os alvos no processo de rastreamento.

As características acima são refletidas no projeto criado para o dispositivo proposto (Figura 14).



Figura 14: Características comuns da arquitetura dos dispositivos comerciais e o proposto nesta dissertação. [38]

Os componentes do dispositivo proposto são descritos a seguir. Na Figura 15 é apresentado o dispositivo proposto:

- 3 câmeras web, para nossa implementação se utilizou duas câmeras do tipo CCD Logitech QuickCam Pro 4000 e uma CMOS Creative WebCam NX Pro.
- Uma caixa de madeira, pintada de cor preta, com dimensões de  $0.5 \times 0.5 \times 0.5$  m.
- Um suporte móvel de alumínio, que serve de base para as câmeras web.
- Duas lâmpadas, colocadas nos extremos laterais da caixa de madeira.
- Bolas de isopor branco utilizadas como marcadores.



Figura 15: Dispositivo proposto nesta dissertação.

Quanto à forma de conexão do dispositivo, podemos dizer que ele só precisa estar ligado a um único computador, o qual requer de 3 portas USB livres para a conexão direta e individual de cada câmera web. O resto do funcionamento fica sob o controle de uma biblioteca implementada para administrar e interpretar a emissão dos eventos, gerados a partir de adaptações do dispositivo, e enviá-los a uma aplicação.

Uma diferença com dispositivos comerciais é que nosso dispositivo não utiliza a luz infravermelha como parte elementar de seu funcionamento. O motivo pelo qual os dispositivos comerciais usam este tipo de luz é porque precisam de um meio físico que permita ressaltar os marcadores dentro do espaço de trabalho. Eles conseguem isto ao trabalhar com câmeras onde foram colocados filtros especiais sobre as lentes para permitir a detecção de fontes de luz infravermelha e marcadores revestidos de material que reflete este tipo de luz.

Para manter essa mesma característica, nosso dispositivo utiliza uma estratégia baseada em criar um contraste natural de cor entre o que seria o volume de trabalho, definido pelas dimensões da caixa de cor preta, e os marcadores esféricos que são as bolinhas de isopor branco. Esta característica procura simular o mesmo ambiente físico que se consegue quando se utiliza luz infravermelha, podendo prescindir de câmeras modificadas para capturar algum tipo de luz especial para o processo de rastreamento.

Temos também algumas características adicionais levadas em conta na construção do nosso dispositivo, em razão do uso e do tipo de aplicação ao qual está dirigido seu projeto físico. Essas características adicionais são:

- O desenho compacto permite colocar o dispositivo dentro um ambiente desktop comum.
- As dimensões do espaço de trabalho foram projetadas para suportar o rastreamento da mão do usuário, que leva colocada uma luva preta para não alterar o contraste entre a cor do fundo e a dos marcadores.
- A estrutura permite liberdade de movimento na hora do rastreamento dos marcadores, os quais poderão estar colocados na luva ou serem agrupados em padrões específicos, que também poderiam ser controlados pela mão do usuário (Figura 16).

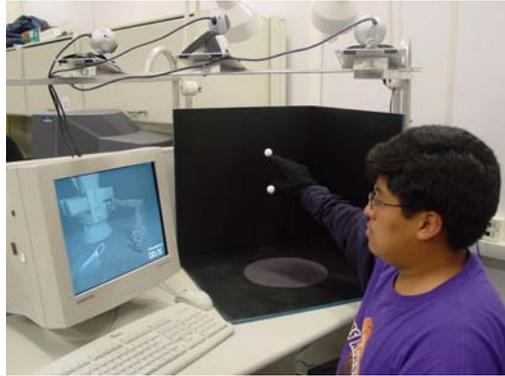


Figura 16: Espaço para rastreamento da movimentação livre da mão do usuário.

Sobre a resolução de captura das câmeras, foi definido trabalhar a uma resolução de  $320 \times 240$  pixels. Esta resolução é a metade da resolução máxima das câmeras utilizadas, mas provê uma imagem nítida e útil para as etapas de processamento das imagens de vídeo.

### 3.2. Processo de Rastreamento

A segunda parte deste capítulo apresentará a teoria relacionada e a implementação dos subprocessos que compõem o rastreamento óptico como um todo.

O processo de rastreamento óptico é composto por vários subprocessos:

- Captura e processamento da imagem de vídeo, extração de marcadores.
- Calibração das câmeras web.
- Correlação de marcadores.
- Reconstrução 3D dos marcadores.

A seguir, faremos a descrição de cada subprocesso, que contará com uma breve introdução da teoria que o envolve, sendo apoiada por imagens de uma aplicação de teste que foi implementada para visualizar os resultados obtidos depois de executar o subprocesso sobre as imagens capturadas.

### 3.2.1. Processamento da Imagem e Extração de Marcadores

O primeiro subprocesso na implementação do sistema de rastreamento é a extração de marcadores e pontos de referência a partir da análise e do processamento da imagem de vídeo capturada por cada câmera web.

Neste subprocesso se tem como primeiro passo a captura e visualização na tela do computador das imagens de vídeo que vêm de cada câmera. Na implementação podemos observar como as 3 imagens de vídeo são capturadas simultaneamente. As imagens são capturadas com uma resolução de 320×240 pixels, a qual permite manter uma taxa de visualização que em média oscila entre 29.05 e 30.00 fps.

A Figura 17 mostra nossa aplicação de teste, na qual podemos observar as 3 imagens de vídeo capturadas e uma imagem repetida. As três imagens têm diferentes perspectivas do espaço de rastreamento da caixa que compõe o dispositivo, em razão das diferentes posições e campos de visão de cada câmera web colocada no suporte de alumínio da caixa.

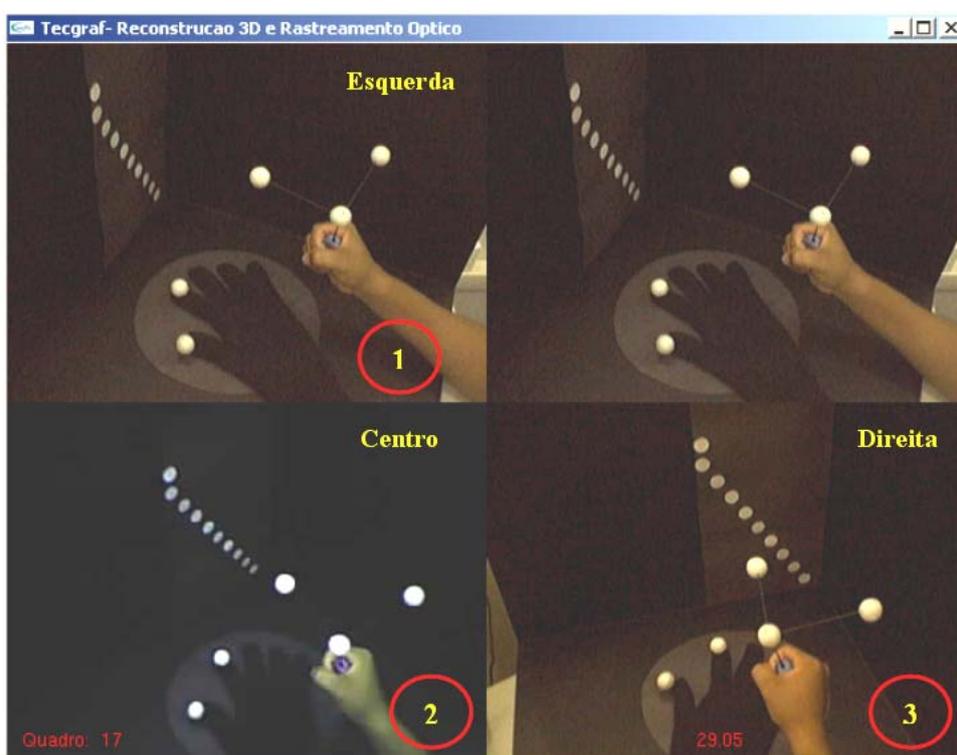


Figura 17 : Imagens de vídeo capturadas por 3 câmeras web.

Como próximo passo na extração das características a partir das imagens de vídeo, foi necessário converter os 3 canais de cor das imagens em um só canal em escala de cinzas. Esta simplificação permite diminuir a quantidade de informação referente às imagens analisadas e, ao mesmo tempo, ressaltar as características de contraste de cor que há entre o fundo e os marcadores ou pontos de referência. Nós assumimos esta característica como parte restritiva no ambiente onde será feito o rastreamento dos marcadores.

A conversão dos canais de cor para a escala de cinzas é feita pela seguinte fórmula implementada pela biblioteca OpenCV [23] para a conversão de cada valor de um pixel em cor para um valor em escala de cinza :

$$GrayScale_{Value}(x,y) = (R_{Value} \times 0.2126 + G_{Value} \times 0.7156 + B_{Value} \times 0.072169)_{Value}(x,y)$$

Depois da conversão das imagens em escala de cinzas, o passo seguinte é aplicar em cada imagem uma série de filtros que nos permitam inicialmente eliminar ruídos e suavizar os contornos e formas definidos pelos objetos na imagem de vídeo [24].

O primeiro filtro utilizado é um filtro de Gauss para suavizar as variações dos contornos dos objetos da imagem, e assim permitir uma melhor identificação dos contornos num passo posterior.

Na Figura 18, as imagens foram transformadas para imagens em escala de cinzas e processadas com um filtro de Gauss, com um kernel 2D de máscara 5x5, o qual é convoluído com a imagem de vídeo capturada. Esse filtro ajuda a diminuir o efeito de sombras e brilhos que possam estar presentes nas imagens. A máscara 5x5 criada para convoluir cada pixel da imagem para o caso discreto com um desvio padrão de  $\sigma = 1$  e  $m = 5$  é a seguinte:

$$I_G = I \otimes G$$

$$I_G = \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} e^{-\frac{h^2}{2\sigma^2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} e^{-\frac{k^2}{2\sigma^2}} I(i-h, j-k) \quad G_{5 \times 5} = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

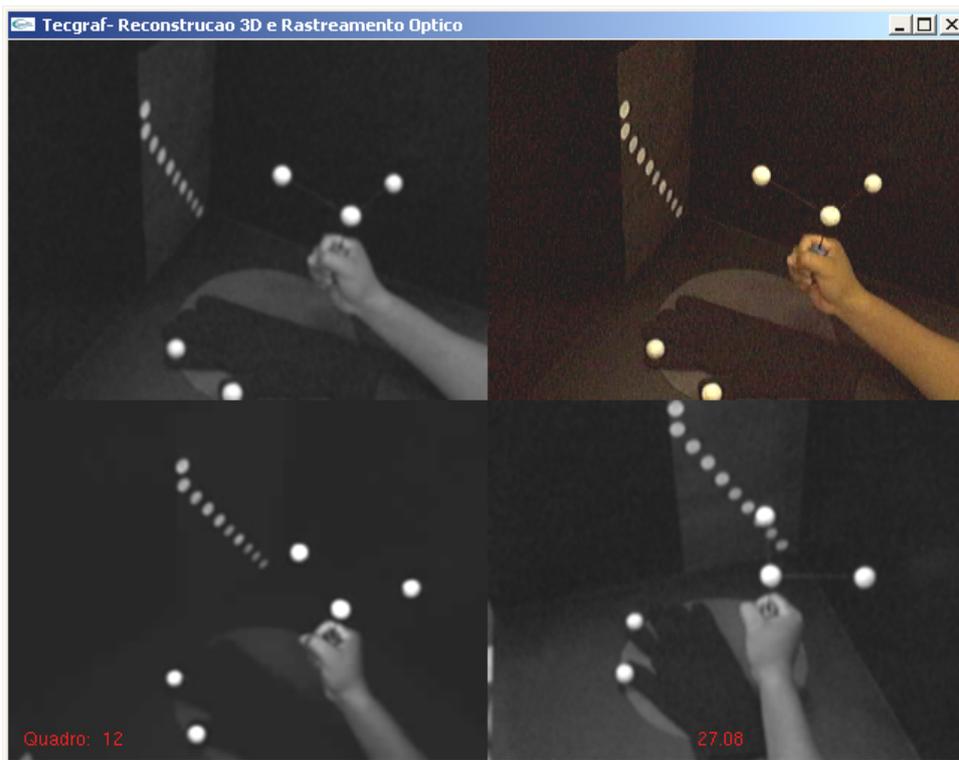


Figura 18 : Imagens de vídeo convertidas à escala de cinzas e processadas com filtro gaussiano de kernel 5x5.

Como próximo passo, foi definida a aplicação de um filtro de threshold sobre a imagem em escala de cinzas, o qual permitirá extrair as áreas de maior contraste na imagem.

Como se assumiu trabalhar num ambiente de contraste, o filtro de threshold devolverá uma imagem binária na qual as possíveis áreas dos pontos mais claros dentro da imagem serão ressaltadas. Esta restrição de trabalhar com um fundo em contraste em relação aos marcadores foi definida pela correlação que se quer criar com um ambiente de rastreamento com câmeras de detecção de luz infravermelha e marcadores revestidos de material retrorreflexivo.

A fórmula do filtro de threshold que a biblioteca OpenCV implementa tem o seguinte critério:

$$Binary(x, y)_{value} = \begin{cases} = Valor\ Maximo(255) & ;\ if\ (GrayScale_{value}(x, y)) > threshold_{value} \\ 0 & ;\ Otherwise \end{cases}$$

Dentro de nossa aplicação teste, o limiar  $threshold_{value}$  utilizado foi de 150, isto dentro da escala de cinzas de [0, 255], mas também foram testados valores dentro do intervalo (144–158), que também deram bons resultados.



Figura 19 : Imagens de vídeo convertidas em imagens binárias por filtro threshold.

Como se pode observar na Figura 19, as áreas dos objetos em contraste com o fundo estão sendo ressaltadas como áreas brancas. É aqui onde se planeja extrair os possíveis contornos dos pontos de referência ou dos marcadores a serem rastreados, e com esta informação, calcular o centro das áreas circulares que definem os marcadores.

A estratégia para achar o centro das áreas brancas, que representam os marcadores na imagem binária, consiste em aproximar o centro da área com a média aritmética das coordenadas 2D dos pontos que formam o contorno das áreas circulares brancas que representam nossos marcadores.

Quando já se definiu a média aritmética das coordenadas 2D dos pontos de cada contorno, nos eixos “ $u$ ” e “ $v$ ”, utilizamos esses valores como o centro que define a posição de nosso marcador em coordenadas da imagem  $(u, v)$ . Este mesmo processo é feito para se rastrear os marcadores esféricos que utilizaremos no processo de reconstrução. Na Figura 20, é apresentado o resultado do processo de extração dos centros das áreas, que representam marcadores e pontos de referência no espaço de rastreamento.

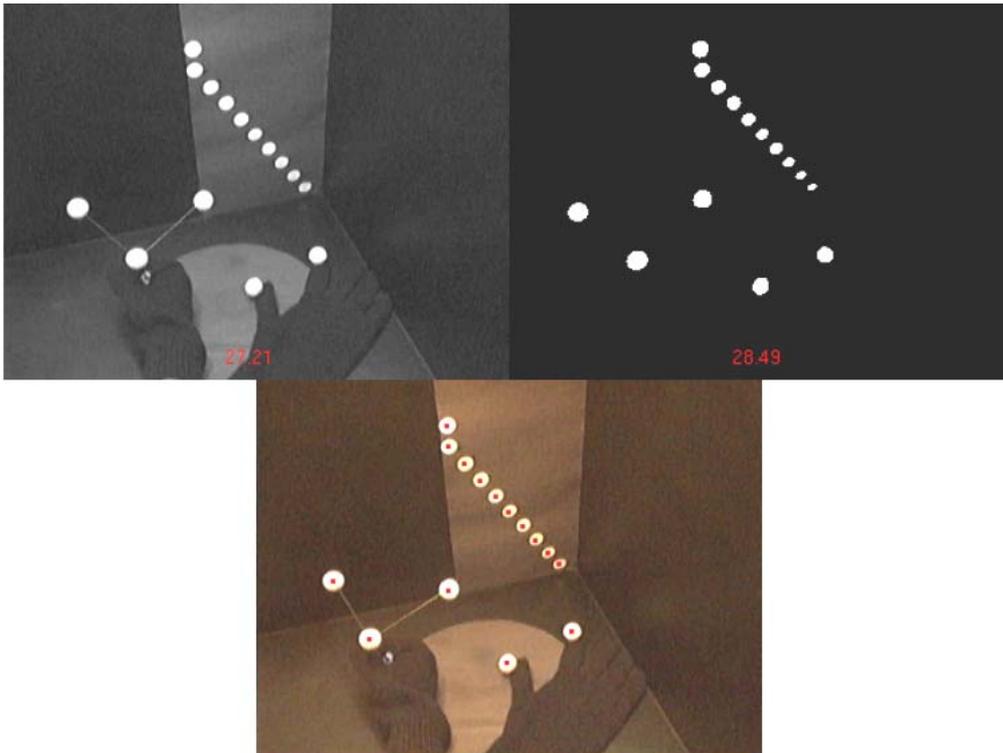


Figura 20 : Imagem de vídeo onde se visualiza o centro dos pontos do padrão e marcadores.

Com a extração das características termina esta primeira etapa do sistema de rastreamento.

Com base nas coordenadas da imagem 2D correspondentes aos pontos de referência e marcadores, pode-se continuar com a etapa de calibração, correspondência de marcadores e finalmente reconstrução.

### 3.2.2. Calibração de Câmeras

O segundo subprocesso na implementação do sistema de rastreamento óptico, que nos permitirá produzir a imersão em um cenário virtual, é a calibração de cada câmera web que utilizaremos no dispositivo proposto.

A calibração de cada câmera tem como objetivo encontrar uma medida de relacionamento entre o mundo 3D, ou seja, o espaço físico que é visto pela câmera, e o plano da imagem 2D que nós vemos na tela do computador visualizando o espaço físico capturado.

Essa medida de relacionamento é expressa como uma transformação que inclui uma matriz de rotação e um vetor de translação que nos permitam levar

coordenadas do mundo 3D físico – medidas tomadas em milímetros – para posições 3D do sistema de coordenadas internas definidas para cada câmera web – medidas em escala livre – e, em um último passo, transformar as coordenadas 3D expressas no sistema de cada câmera em posições 2D no plano da imagem que é desenhado no computador – medidas expressas em pixels. O plano da imagem está posicionado a uma distância “ $f$ ” (distância focal expressa em coordenadas do sistema da câmera).

Para a calibração da câmera existem vários tipos de algoritmos definidos na literatura. Em uma das classificações existentes, eles são divididos em algoritmos de calibração fotogramétrica e algoritmos de autocalibração [25].

Para a nossa proposta decidiu-se utilizar um algoritmo de calibração fotométrica, o método TSAI na versão não-coplanar, que se baseia na captura de um padrão 3D conhecido, do qual são conhecidas as medidas físicas de certos pontos de referência.

### 3.2.2.1. Implementação do Método TSAI Não-Coplanar

O método de calibração de câmera proposto por Roger Y. Tsai descreve como achar os parâmetros intrínsecos e extrínsecos de uma câmera que permite a transformação de pontos no espaço 3D em pixels no plano da imagem da câmera [26, 27].

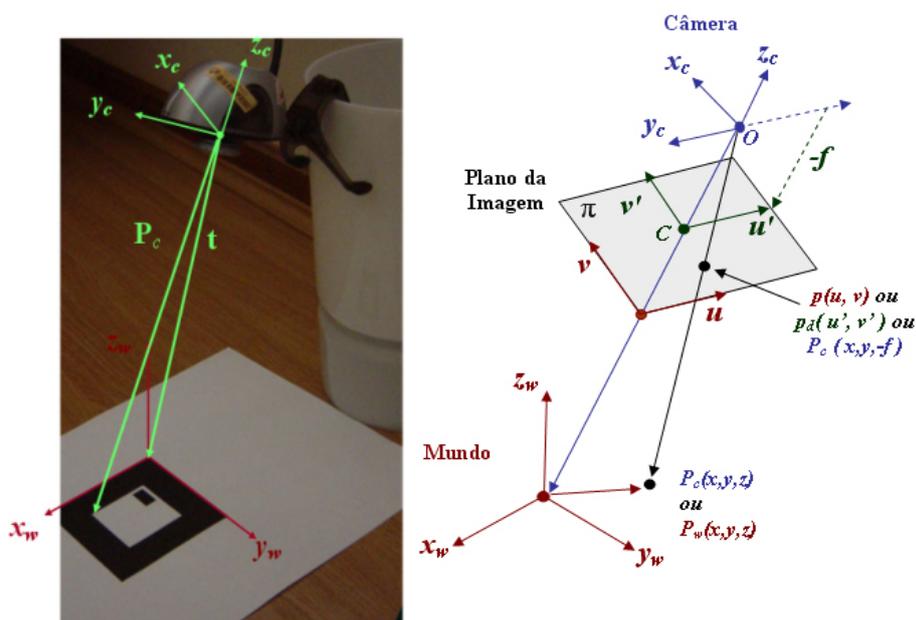


Figura 21: Modelo de captura da câmera pinhole.

Como mostrado na Figura 21, no modelo da câmera pinhole tem-se um ponto no espaço do mundo  $P_w(x, y, z)$ , o qual tem como equivalente em coordenadas do mundo da câmera o ponto  $P_c(X, Y, Z)$ . O raio que sai do centro do sistema de coordenadas da câmera intercepta o plano da imagem no ponto  $P_c(X, Y, -f)$ , isto em coordenadas da câmera, que em coordenadas no plano da imagem é  $p(u', v')$ . Define-se o ponto “O” como o centro do sistema de coordenadas da câmera e “C” como centro de coordenadas 2D do plano da imagem “ $\pi$ ”. Normalmente este centro será considerado como o ponto (0,0) da imagem o que realmente é definido por o canto inferior esquerdo, como mostrado na Figura 21, para o centro do plano  $p(u', v') = p(u - C_u, v - C_v)$ . Além disso, devido à presença da distorção radial das lentes da câmera, o ponto  $P_d(u'_d, v'_d)$  seria o ponto final que conseguimos ver na imagem desenhada no plano da imagem. Como último passo é necessário transformar essas coordenadas da imagem, com distorção, para coordenadas no frame buffer do dispositivo de captura  $p_f(u', v')$ .

No método de calibração TSAI, na versão não-coplanar, poderíamos precisar conhecer alguns parâmetros específicos sobre o hardware das câmeras a serem utilizadas, que são:

- $dx$  e  $dy$ , a distância horizontal e vertical entre os centros das células contidas no chip CCD dentro da câmera.
- $Ncx$ , o número de células contidas em uma fila no chip CCD.
- $Nfx$ , o número de pixels contidos em uma fila no buffer de memória.

Para nossa experiência trabalhou-se com dois tipos de câmera web, mostradas na Figura 22, para as quais foram encontrados os parâmetros de hardware necessários ao algoritmo de TSAI:

<b>Câmera</b>	Logitech QuickCam Pro - 4000	Creative Webcam Nx -Pro
<b>Modelo do Chip</b>	Sony ICX098AK, Sony ICX098BQ, Sharp LZ24BP	Hyundai HV7131B
<b>Célula tamanho horizontal dx</b>	0.0056 $\mu\text{m}$	0.008 $\mu\text{m}$
<b>Célula tamanho vertical dy</b>	0.0056 $\mu\text{m}$	0.008 $\mu\text{m}$
<b>Número de células Ncx</b>	659	642
<b>Número de pixels Nfx</b>	640	640

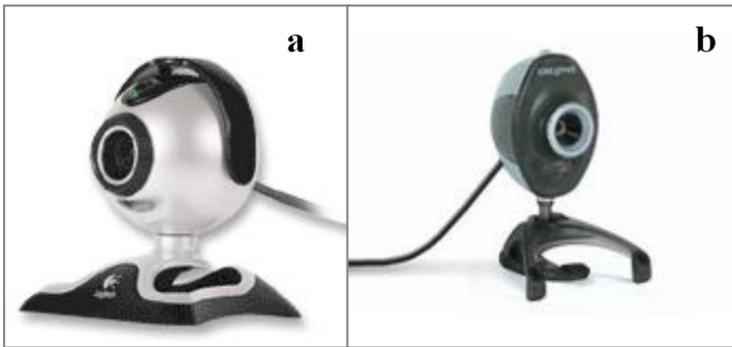


Figura 22 : Câmeras web utilizadas, (a) Logitech QuickCam Pro-4000, (b) Creative WebCam Nx-Pro.

Esses dados, na versão original do algoritmo TSAI não-coplanar, nos permitem definir alguns fatores de escala que seriam importantes no caso de se trabalhar com aplicações que requerem precisão na recuperação e no relacionamento entre as medidas físicas de nosso hardware e os resultados que o algoritmo TSAI gerará.

Por isso, nas aplicações em que a precisão nas medidas recuperadas é importante, o conhecimento desses parâmetros será justificável, mas nossa implementação, com a intenção de generalizar, pode assumir que trabalhamos com uma câmera pinhole padrão cujos parâmetros de que precisamos seriam:

- **Modelo do chip:** Nenhum.
- Nas características de um chip é descrito só o tamanho de uma célula em micrômetros, para o qual se assume que a distância entre os centros das células é igual ao tamanho dela. Então, assumimos na horizontal  $dx = 1.0 \mu m$  e na vertical  $dy = 1.0 \mu m$ , o que levará a uma proporção unitária de medidas da célula.
- O número de pixels efetivo utilizado dentro do chip foi  $N_{cx} = 640$  (células), e o tamanho do frame buffer com o qual a câmera trabalha é de  $N_{fx} = 640$  (pixels). Aqui o fator entre o tamanho físico e número de pixels capturado por cada célula de nosso chip simétrico será também unitário.

Depois de definir os dados referentes aos parâmetros internos de nosso hardware, inicia-se o processo de calibração definindo o mundo ou espaço de trabalho onde interagiremos. A Figura 23 ilustra como seriam os eixos do sistema de mundo definido pela estrutura da caixa que compõe o dispositivo, em uma configuração se utiliza fundo preto para a caixa e no outro se utiliza fundo branco.

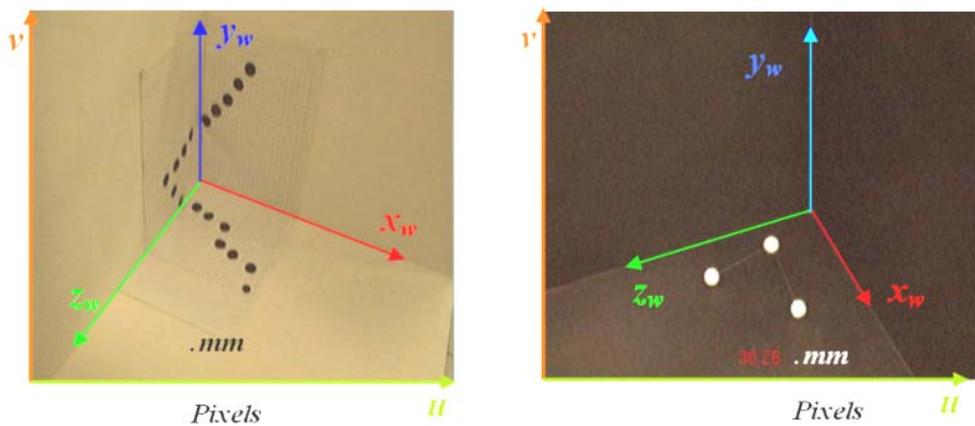


Figura 23 : Imagem dos eixos do sistema global de referência dentro da caixa de rastreamento.

Como próximo passo, temos que definir um padrão de calibração, o qual contém pontos de referência dos quais se conhece suas coordenadas no espaço 3D (mm). O que precisamos é encontrar esses mesmos pontos em coordenadas do plano da imagem (pixels) dentro da imagem de vídeo. É assim que definimos inicialmente o padrão visto na Figura 24.

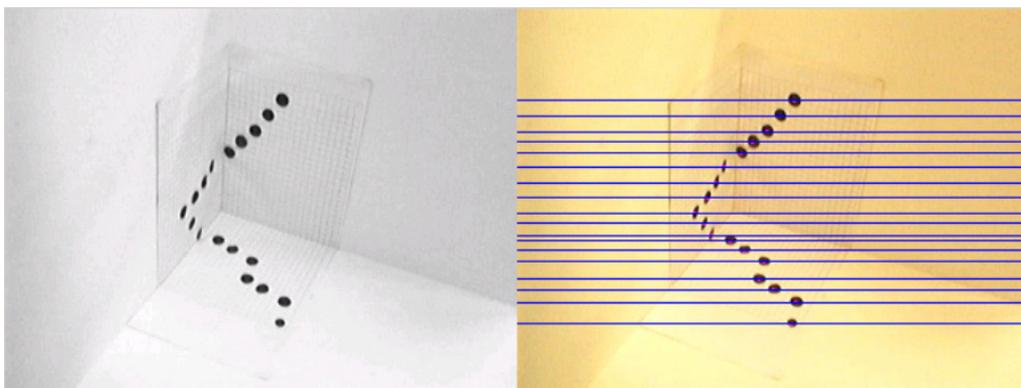


Figura 24 : Imagem do padrão utilizado inicialmente.

Este padrão tem como principal característica a distribuição dos pontos de referência, os quais estão espalhados entre os 3 planos que definem nosso espaço

3D. Esta característica é parte de uma exigência do método TSAI na versão não-coplanar.

A identificação dos pontos de referência na imagem é feita com base no algoritmo de detecção de contornos que fez parte da etapa de extração de características, o qual nos permite extrair contornos dos pontos pretos dentro da imagem. A coordenada  $(u, v)$  do centro, expressa em pixels, servirá como um dos pontos de referência que, junto com as coordenadas 3D em mm dos mesmos pontos identificados, serão os principais dados de entrada para o algoritmo de calibração TSAI.

O processo é mostrado na Figura 25. Após a detecção do centro de cada ponto preto reconhecido, desenha-se uma linha que passa pelo centro de cada ponto, que nos servirá para mostrar a correlação natural entre o ponto 3D no espaço e o ponto 2D na imagem.

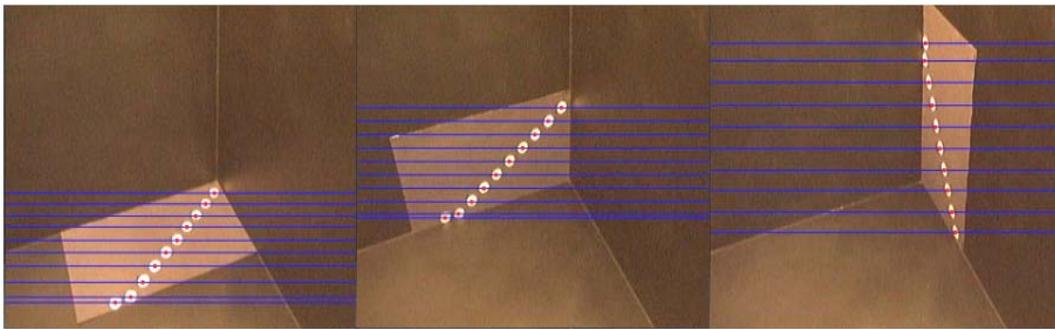


Figura 25 : Imagem do padrão com a identificação dos pontos de calibração.

A estratégia utilizada para fazer a correlação entre os pontos foi a forma crescente que têm os pontos nas coordenadas da imagem no eixo “ $v$ ”. Com esta distribuição dos pontos, que podemos ver na Figura 25, conseguimos ter os pontos perfeitamente identificados para o processo de calibração.

A variação feita entre o padrão da Figura 24 e o da Figura 25 foi a divisão do padrão inicial nos 3 planos independentes que o formavam e a captura independente de cada plano. Este tipo de adaptação permitiu capturar uma maior quantidade de pontos de referência. Como podemos notar na Figura 24, a quantidade de pontos recuperados com o padrão inicial foi de 18 pontos; já com o padrão dividido utilizado para calibração conseguimos 30 pontos, 10 em cada plano.

### 3.2.2.2. O Método TSAI Não-Coplanar

Conforme definido por Tsai [27], o processo de transformação de pontos em coordenadas 3D do mundo para coordenadas na imagem consta de 4 passos, descritos também em [28].

#### Passo 1:

$$P_c \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R \left( P_w \begin{bmatrix} x \\ y \\ z \end{bmatrix} \right) + T_{Camera} \quad (1)$$

O primeiro passo é transformar os pontos em coordenadas do mundo  $P_w(x, y, z)$ , que são os pontos de referência conhecidos em mm, para pontos expressos em coordenadas do sistema de coordenadas da câmera  $P_c(X, Y, Z)$ , o qual tem uma escala livre.

Os parâmetros a serem calculados são a matriz de Rotação “ $R$ ” e um vetor de Translação “ $T$ ”, que representam a rotação do sistema de coordenadas do mundo em relação à sua origem, até ficar com seus eixos na mesma posição e orientação do sistema de coordenadas da câmera, seguida de uma translação 3D que equivale a levar o sistema do mundo, já rotado, para a posição do centro do sistema de coordenadas da câmera (Figura 26).

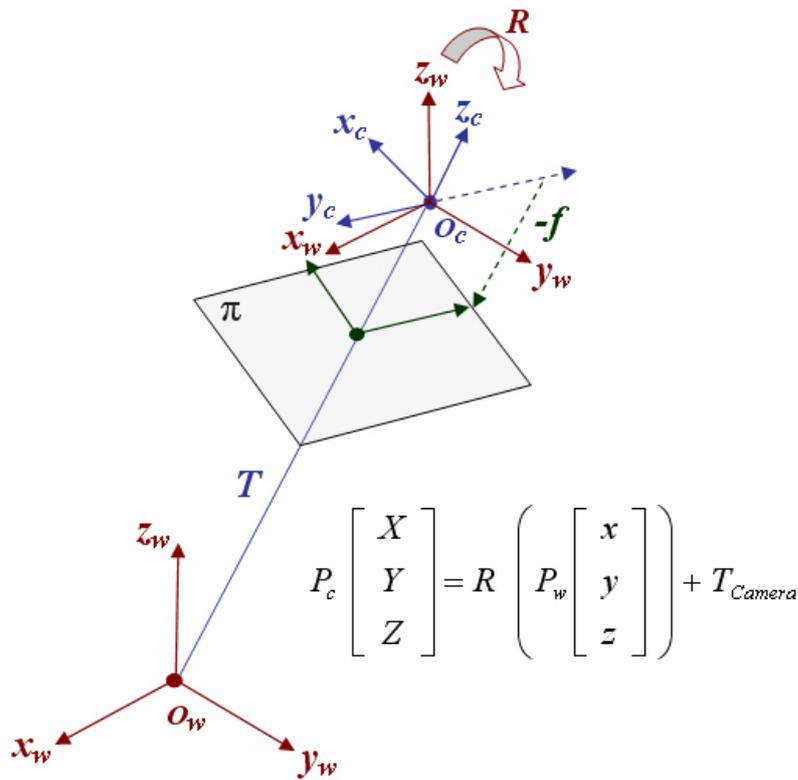


Figura 26 : Imagem que descreve a transformação entre os sistemas de coordenadas do mundo para a câmera.

Então o método TSAI não-coplanar calcula os parâmetros extrínsecos que se vêem representados pela matriz de rotação e o vetor de translação.

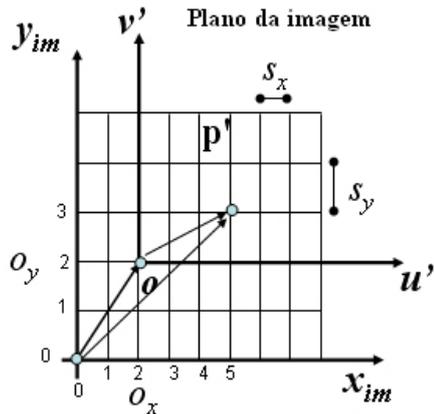
Para este processo são utilizados os pontos de referência, em coordenadas da imagem, que capturamos e que estão corretamente em correlação com seus equivalentes em coordenadas no mundo

$$\begin{aligned} (u_1, v_1) &\longrightarrow (X_1^w, Y_1^w, Z_1^w) \\ (u_2, v_2) &\longrightarrow (X_2^w, Y_2^w, Z_2^w) \\ (u_3, v_3) &\longrightarrow (X_3^w, Y_3^w, Z_3^w) \\ \\ (u_N, v_N) &\longrightarrow (X_N^w, Y_N^w, Z_N^w) \end{aligned}$$

Assim o método TSAI não-coplanar tentará montar um sistema de equações com o qual calculará os elementos que compõem a matriz de rotação e o vetor de translação, os quais são conhecidos como os parâmetros extrínsecos da câmera. Para obter as equações temos que montar a relação das coordenadas obtidas do

plano da imagem com as coordenadas da câmera, e depois das coordenadas da câmera com coordenadas do mundo. É assim que, na seqüência, é mostrado como chegamos a essa correlação direta entre coordenadas do plano da imagem e as coordenadas do mundo.

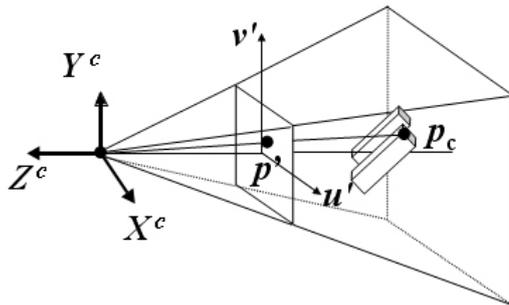
- Um ponto expresso em coordenadas do plano da imagem.



$$\frac{u'}{s_x} = x_{im} - o_x \quad \frac{v'}{s_y} = y_{im} - o_y$$

$$p' = \begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} s_x(x_{im} - o_x) \\ s_y(y_{im} - o_y) \end{pmatrix}$$

- O mesmo ponto expresso em coordenadas da câmera.



$$p_c = \begin{bmatrix} X^c \\ Y^c \\ Z^c \end{bmatrix}$$

$$p' = \begin{pmatrix} u' \\ v' \end{pmatrix} = -f \begin{pmatrix} X^c / Z^c \\ Y^c / Z^c \end{pmatrix}$$

- Então, de coordenadas da imagem para coordenadas da câmera.

$$p' = \begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} s_x(x_{im} - o_x) \\ s_y(y_{im} - o_y) \end{pmatrix} = -f \begin{pmatrix} X^c / Z^c \\ Y^c / Z^c \end{pmatrix}$$

$$x_{im} - o_x = -\frac{f}{s_x} \frac{X^c}{Z^c} = -f_x \frac{X^c}{Z^c}$$

$$y_{im} - o_y = -\frac{f}{s_y} \frac{Y^c}{Z^c} = -f_y \frac{Y^c}{Z^c}$$

(2)

- Levando em conta a equação eq.(1), podemos expressar a equação como sendo:

$$\begin{bmatrix} X^c \\ Y^c \\ Z^c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X^w \\ Y^w \\ Z^w \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}_{camera} \quad (3)$$

- É assim que, se desenvolvemos na equação eq.(3) a multiplicação e a adição sobre as coordenadas do mundo e as relacionamos com as coordenadas da câmera da equação eq.(2), temos:

$$x_{im} - o_x = -f_x \frac{r_{11}X^w + r_{12}Y^w + r_{13}Z^w + T_x}{r_{31}X^w + r_{32}Y^w + r_{33}Z^w + T_z}$$

$$y_{im} - o_y = -f_y \frac{r_{21}X^w + r_{22}Y^w + r_{23}Z^w + T_y}{r_{31}X^w + r_{32}Y^w + r_{33}Z^w + T_z}$$

- Assumindo que se conhecem os valores de  $(o_x, o_y)$  e definindo

$$x_i = x_{im} - o_x \quad y_i = y_{im} - o_y, \text{ chegamos à seguinte expressão:}$$

$$x_i = -f_x \frac{r_{11}X_i^w + r_{12}Y_i^w + r_{13}Z_i^w + T_x}{r_{31}X_i^w + r_{32}Y_i^w + r_{33}Z_i^w + T_z}$$

$$y_i = -f_y \frac{r_{21}X_i^w + r_{22}Y_i^w + r_{23}Z_i^w + T_y}{r_{31}X_i^w + r_{32}Y_i^w + r_{33}Z_i^w + T_z} \quad (4)$$

- Resolvendo a equação eq.(4) teremos:

$$r_{31}X_i^w + r_{32}Y_i^w + r_{33}Z_i^w + T_z = -\frac{f_x}{x_i} (r_{11}X_i^w + r_{12}Y_i^w + r_{13}Z_i^w + T_x) \quad (5)$$

$$r_{31}X_i^w + r_{32}Y_i^w + r_{33}Z_i^w + T_z = -\frac{f_y}{y_i} (r_{21}X_i^w + r_{22}Y_i^w + r_{23}Z_i^w + T_y)$$

$$x_i f_y (r_{21}X_i^w + r_{22}Y_i^w + r_{23}Z_i^w + T_y) = y_i f_x (r_{11}X_i^w + r_{12}Y_i^w + r_{13}Z_i^w + T_x)$$

$$f_x / f_y = \alpha \cong 1$$

$$x_i X_i^w r_{21} + x_i Y_i^w r_{22} + x_i Z_i^w r_{23} + x_i T_y = y_i X_i^w \alpha r_{11} + y_i Y_i^w \alpha r_{12} + y_i Z_i^w \alpha r_{13} + y_i \alpha T_x$$

- Como o fator  $\alpha$  representa uma proporção entre o tamanho horizontal e o vertical de nossa célula do chip de captura e essas dimensões são definidas como unitárias em nossa câmera padrão, pode-se definir este fator como sendo unitário:

$$x_i X_i^w r_{21} + x_i Y_i^w r_{22} + x_i Z_i^w r_{23} + x_i T_y = y_i X_i^w r_{11} + y_i Y_i^w r_{12} + y_i Z_i^w r_{13} + y_i T_x$$

- A equação acima pode ser representada como:

$$x_i X_i^w v_1 + x_i Y_i^w v_2 + x_i Z_i^w v_3 + x_i v_4 - y_i X_i^w v_5 - y_i Y_i^w v_6 - y_i Z_i^w v_7 - y_i v_8 = 0$$

- Onde:

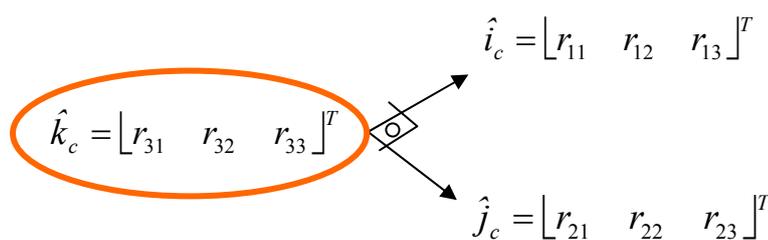
$$v_1 = r_{21} \quad v_5 = r_{11}$$

$$v_2 = r_{22} \quad v_6 = r_{12} \quad \mathbf{A}\mathbf{v} = \mathbf{0} \Rightarrow \dots \Rightarrow \bar{\mathbf{v}} \neq \mathbf{0}$$

$$v_3 = r_{23} \quad v_7 = r_{13}$$

$$v_4 = T_y \quad v_8 = T_x$$

- Resolvendo o sistema de equações acima, encontramos dois dos vetores componentes da matriz de rotação, faltando achar ainda um vetor. Utilizando o produto vetorial dos vetores encontrados, podemos calcular o terceiro vetor ortogonal com o qual se formará nossa matriz de rotação:



$$\hat{i}_c = \begin{bmatrix} r_{11} & r_{12} & r_{13} \end{bmatrix}^T$$

$$\hat{j}_c = \begin{bmatrix} r_{21} & r_{22} & r_{23} \end{bmatrix}^T$$

$$\hat{k}_c = \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix}^T$$

- Finalmente, falta encontrar a terceira componente do vetor de translação. Para isso, com os valores já encontrados, resolvemos a equação eq.(5) e achamos essa terceira componente:

$$r_{31}X_i^w + r_{32}Y_i^w + r_{33}Z_i^w + \mathbf{T}_z = -\frac{f_x}{x_i} \left( r_{11}X_i^w + r_{12}Y_i^w + r_{13}Z_i^w + T_x \right)$$

**Passo 2:**

O segundo passo é transformar os pontos em coordenadas da câmera para coordenadas da imagem (Figura 27), assumindo que estamos numa situação sem distorção, usando a projeção em perspectiva.

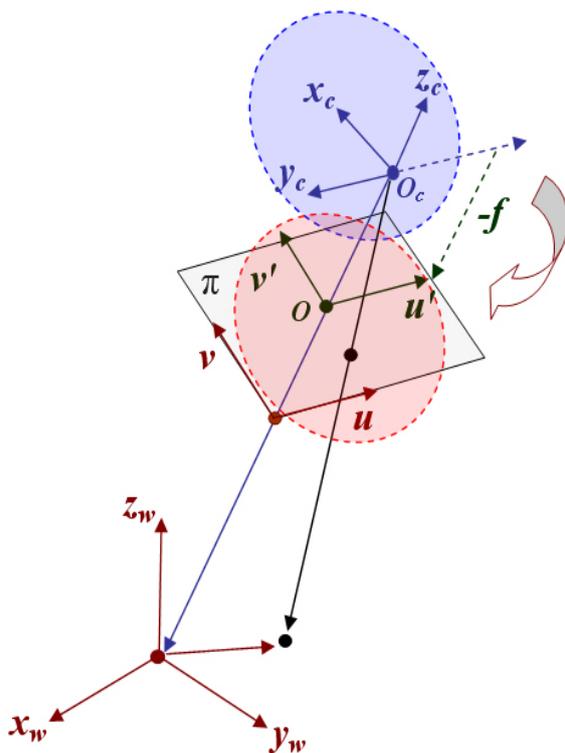


Figura 27 : Imagem que descreve a transformação entre os sistemas de coordenadas da câmera para coordenadas da imagem.

O parâmetro a ser calculado será a distância focal  $-f$  :

$$P_c(X, Y, Z) \Rightarrow p(u', v')$$

$$u' = -f \frac{X}{Z}$$

$$v' = -f \frac{Y}{Z}$$

**Passo 3:**

O terceiro passo é calcular as coordenadas dos pontos na imagem modificados pela distorção radial.

O parâmetro a ser calculado aqui são os coeficientes  $K_1$ ,  $K_2$ , mas como o segundo coeficiente tem um valor bem pequeno não será considerado no cálculo.

$$p(u', v') \Rightarrow p_d(u'_d, v'_d)$$

$$u'_d + D_x = u'$$

$$v'_d + D_y = v'$$

Onde o cálculo da distorção em  $D_x, D_y$  é:

$$D_x = u'_d (K_1 r^2 + K_2 r^4 + \dots)$$

$$D_y = v'_d (K_1 r^2 + K_2 r^4 + \dots)$$

$$r = \sqrt{u'_d{}^2 + v'_d{}^2}$$

#### Passo 4:

O quarto passo é calcular a transformação das coordenadas reais da imagem em coordenadas da imagem no computador levando em consideração os tamanhos dos pixels da tela.

Os parâmetros calculados são  $S_x$ , que é o fator de incerteza introduzido para tratar casos como o retardo entre o tempo de captura das imagens e o tempo real de exploração dentro do hardware do dispositivo de captura, e  $d'_x = d_x \frac{Ncx}{Nfx}$ , que é a escala entre o tamanho da célula do chip de captura e o número de pixels que representa.

$$p_d(u'_d, v'_d) \Rightarrow p_f(u'_f, v'_f)$$

$$u'_f = S_x d_x^{-1} u'_d + O_x$$

$$v'_f = d_y^{-1} v'_d + O_y$$

O algoritmo de calibração de TSAI deve fornecer como dados os parâmetros descritos em cada passo, mas esses dados podem ser agrupados em duas categorias:

- Parâmetros intrínsecos: distância focal  $f$ , fator de incerteza  $S_x$ , parâmetro de distorção radial  $K_1$ .

- Parâmetros extrínsecos: a matriz de rotações “ $R$ ” e o vetor de translação “ $T$ ”.

Com esses dados, no modelo da câmera pinhole, podemos criar as seguintes matrizes que definem todo o processo que leva um ponto das coordenadas do mundo 3D para coordenadas na imagem 2D:

- A Matriz de Parâmetros Intrínsecos leva um ponto em coordenadas da câmera para coordenadas na tela do computador.

$$M_{\text{int}} = \begin{bmatrix} -f \frac{S_x}{d'_x} & 0 & O_x \\ 0 & -f \frac{S_y}{d'_y} & O_y \\ 0 & 0 & 1 \end{bmatrix} \quad M_{\text{int}} = \begin{bmatrix} -f_x & 0 & O_x \\ 0 & -f_y & O_y \\ 0 & 0 & 1 \end{bmatrix}$$

- A Matriz de Parâmetros Extrínsecos leva um ponto em coordenadas do mundo para coordenadas da câmera.

$$M_{\text{ext}} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \end{bmatrix} \quad M_{\text{ext}} = [ R_{3 \times 3} \quad T_{3 \times 1} ]$$

Assim, temos o processo descrito na Figura 28:

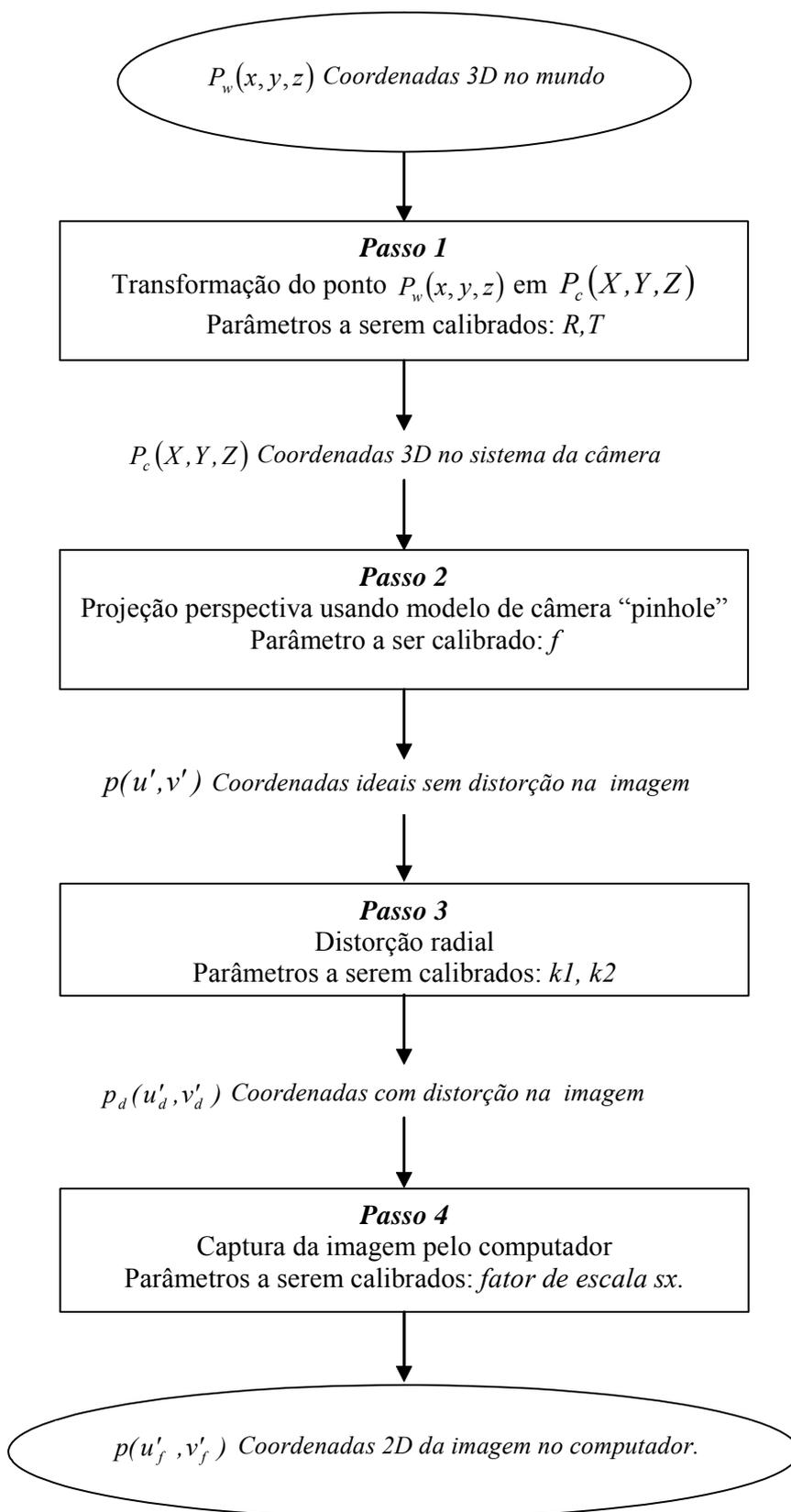


Figura 28: Descrição dos 4 passos que transformam coordenadas do mundo em coordenadas da imagem.

Estas matrizes definem o seguinte pipeline de transformação:

$$P_c \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} P_w \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

$$p \begin{bmatrix} u' \\ v' \\ w \end{bmatrix} = \begin{bmatrix} -f_x & 0 & O_x \\ 0 & -f_y & O_y \\ 0 & 0 & 1 \end{bmatrix} P_c \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

$$\begin{aligned} x_{im} &= -f_x \frac{X^c}{Z^c} + O_x \\ y_{im} &= -f_y \frac{Y^c}{Z^c} + O_y \end{aligned} \quad \begin{pmatrix} x_{im} \\ y_{im} \end{pmatrix} = \begin{pmatrix} u/w \\ v/w \end{pmatrix}$$

### 3.2.3. Correlação de Marcadores

O terceiro subprocesso na implementação do sistema de rastreamento óptico é a correlação entre os diferentes marcadores que aparecem nas imagens das câmeras web.

Um caso intuitivo (Figura 29) é quando temos só um marcador dentro das imagens das câmeras web. Nesse caso, a estratégia consistiria em só utilizar a coordenada 2D do único ponto em contraste que se consegue ver em cada imagem das câmeras. Porém até mesmo nesse caso pode acontecer o problema de ruído que cria um falso marcador nas imagens, para o qual terá que se aplicar um filtro gaussiano e um filtro threshold para suavizar as imagens e convertê-las em binárias, respectivamente.

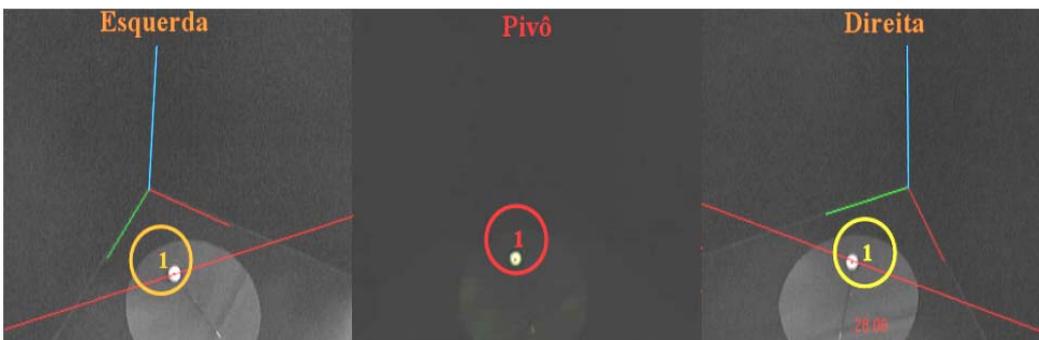


Figura 29 : Caso intuitivo.

Um caso mais complexo é quando temos mais de um marcador nas imagens das câmeras web, e com isto temos que fazer uma correlação entre “n” marcadores detectados nas câmeras. Como mostra a Figura 30, precisamos saber definir qual é o marcador 1 da imagem base ou pivô em relação às outras câmeras.



Figura 30 : Caso com mais de um marcador.

Para solucionar este problema de correlação entre os diferentes marcadores que aparecem nas imagens, utilizaremos alguns conceitos da geometria epipolar para fazer uma correlação inicial entre os marcadores detectados nas imagens.

A utilização da geometria epipolar nos dá as seguintes vantagens:

- Diminuir o espaço de busca dos marcadores dentro das imagens.
- Calcular, no melhor caso, uma correlação um a um entre os marcadores.

A estratégia empregada para implementar este método foi a seguinte:

- Primeiro, capturar os pontos de amostra para o cálculo das matrizes fundamentais entre as câmeras.
- Segundo, com a informação da amostra, calcular para as câmeras tomadas duas a duas, as matrizes fundamentais, que relacionam a geometria projetiva interna entre as câmeras usadas.
- Terceiro, com as matrizes fundamentais já calculadas, criar linhas epipolares a partir dos pontos na imagem da câmera pivô, as quais são a representação dos marcadores capturados nessa câmera.
- Quarto, as linhas epipolares criadas são projetadas sobre o plano da imagem das outras câmeras subordinadas.

- Finalmente, em cada câmera subordinada, é feito um teste para determinar que marcador, em coordenadas da imagem, está mais perto de cada linha epipolar projetada sobre o mesmo plano da imagem. Isto nos permite fazer uma correlação inicial entre os marcadores que aparecem nas imagens de vídeo das câmeras web utilizadas.

Segundo os conceitos da geometria epipolar, as linhas epipolares projetadas devem cair sobre os marcadores que aparecem nas outras câmeras web, e a precisão no cálculo das linhas epipolares dependerá somente da quantidade de pontos de amostra com os quais forem criadas as matrizes fundamentais.

### 3.2.3.1. Captura da Amostra de Pontos

Como se fez na etapa de calibração das câmeras, nesta etapa também precisamos fazer a captura de um conjunto de pontos, corretamente identificados e correlacionados entre as câmeras, que servirão de amostra para fazer o cálculo da matriz fundamental entre as câmeras.

A única informação que devemos conhecer dos pontos capturados são suas coordenadas na imagem.

A estratégia para a captura é a seguinte: captura-se a posição de um único marcador dentro do espaço de rastreamento, o que é feito com as 3 câmeras web simultaneamente (Figura 31).

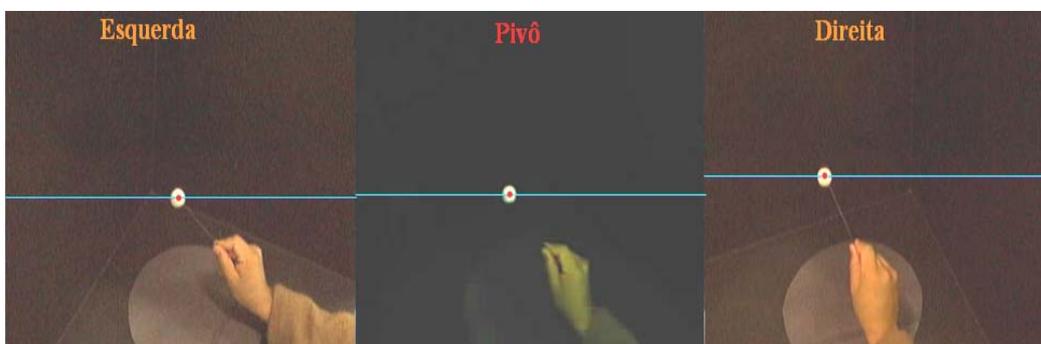


Figura 31 : Marcador para captura da amostra.

O marcador é capturado nas 3 imagens das câmeras web utilizadas e só são armazenadas as coordenadas 2D do marcador se ele é visto pelas 3 câmeras ao mesmo tempo, ou seja, em cada quadro guardamos 3 pontos 2D corretamente correlacionados em cada câmera.

Aqui temos que fazer uma consideração em relação ao problema de sincronização na captura das câmeras, para o qual se assume que o possível retardo é pequeno. Continuaremos nosso processo assumindo que a influência do retardo será mínima.

Na Figura 32, são visualizados os pontos já capturados, que foram definidos pelo movimento do marcador, em forma livre, dentro do espaço de rastreamento útil que pode ser visto pelas câmeras. Assim foram testados e capturados de 3000 a 8000 pontos, isto levou um tempo de 2 a 4 minutos. Podemos ver os pontos capturados em cada imagem:

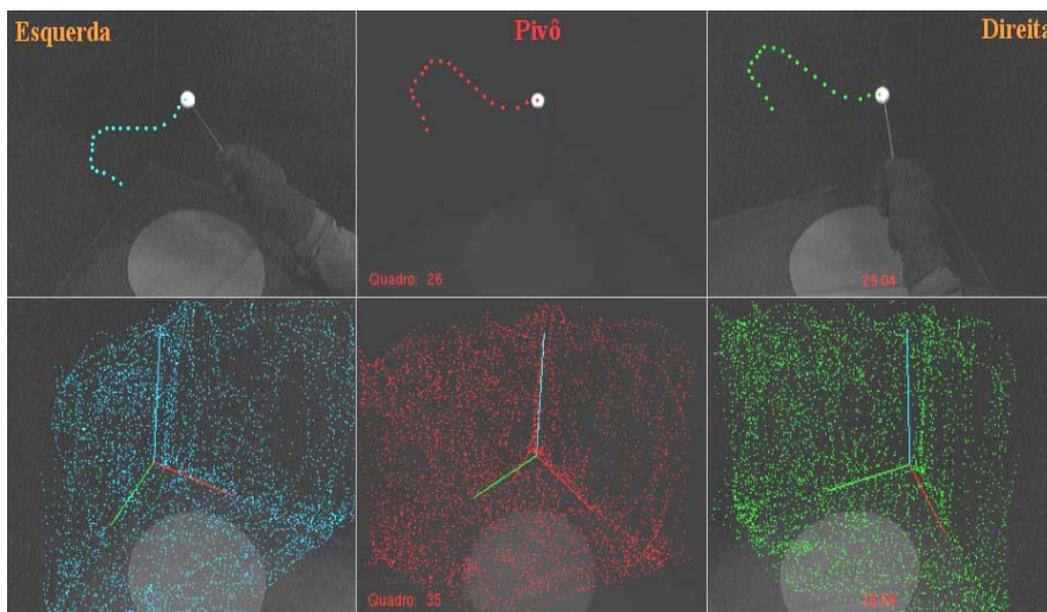


Figura 32 : Pontos capturados em coordenadas 2D em cada câmera.

Como pode-se ver na Figura 32, foi possível armazenar uma boa quantidade de pontos, que serviram de amostra para o cálculo das matrizes fundamentais. Quanto mais pontos forem utilizados, maior será a precisão da matriz fundamental.

### 3.2.3.2. A Geometria Epipolar

A geometria epipolar é a geometria projetiva intrínseca que relaciona duas vistas capturadas de diferentes pontos de referência (Figura 33). Ela é independente da estrutura da cena e só depende dos parâmetros internos das câmeras e da posição relativa entre elas [29].

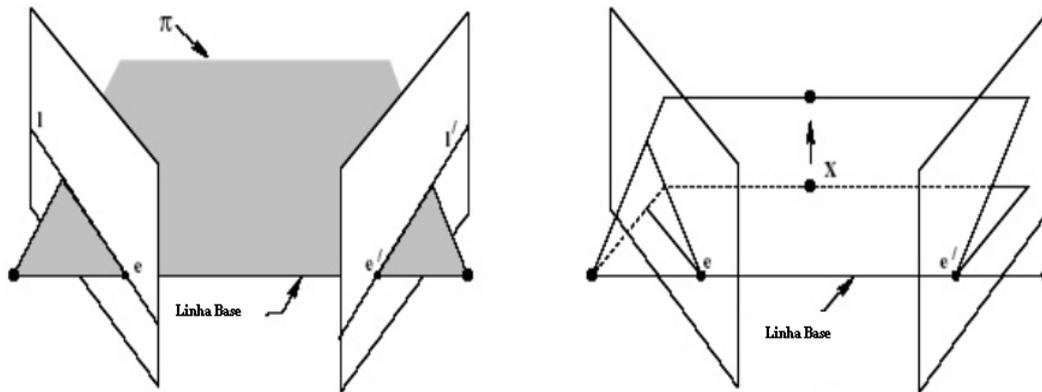


Figura 33 : Interseção entre os planos da imagem e planos gerados pela linha base.

Esta geometria é usualmente requerida devido à necessidade de encontrar uma correspondência entre pontos que aparecem nas diferentes imagens das vistas.

Assim, suponhamos que temos um ponto  $X$  no mundo, o qual está presente nas duas vistas definidas por duas câmeras. Ele aparece no plano da imagem de cada vista como  $x$  na primeira vista e na segunda como  $x'$ . Para mostrar a correspondência que existe entre esses dois pontos, a solução é projetar cada ponto  $x$  e  $x'$  até o ponto  $X$ , o que gerará um plano triangular  $\pi$ , o qual intercepta cada plano da imagem de cada câmera como uma linha  $l'$  projetada desde o ponto de interseção da linha base até o ponto da imagem ( $x$  ou  $x'$ ) (Figura 34).

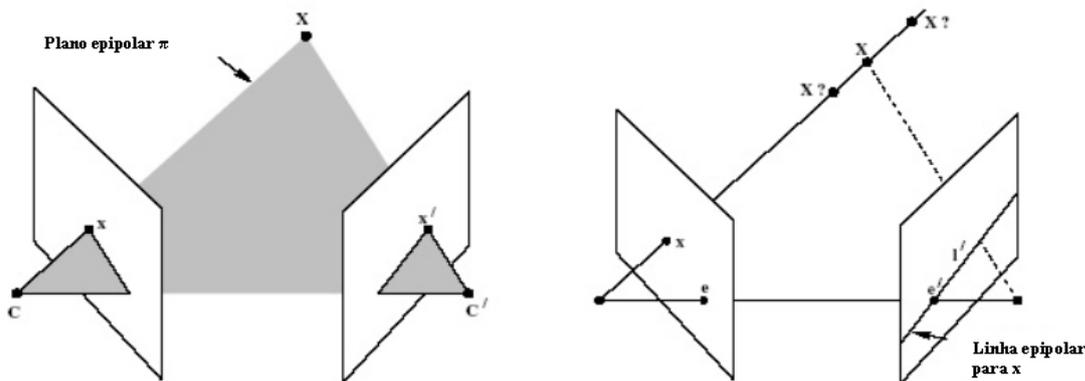


Figura 34 : Figura 34: Definição de uma linha epipolar.

É assim que, se projetamos o ponto  $x$  na imagem da esquerda e vamos descendo o raio de projeção, pelo plano  $\pi$ , até interceptar o plano da imagem da direita, obtemos uma linha  $l'$  de interseção, sobre a qual deve cair o ponto  $x'$  que está também correlacionado com o ponto  $X$  do mundo.

Na geometria epipolar, a matriz fundamental é um elemento-chave. Ela é uma matriz de ordem  $3 \times 3$  e rank 2, que encapsula a geometria intrínseca que descreve a geometria epipolar.

Essa matriz fundamental cumpre a seguinte condição:

$$x'^T F x = 0$$

Tal condição é satisfeita para todo par de pontos  $x \rightarrow x'$  nas duas imagens. Aqui podemos entender então que essa matriz permite encontrar uma correspondência entre pontos definidos em duas câmeras diferentes.

A matriz fundamental também ajuda a gerar as linhas epipolares, pois as multiplicações da matriz fundamental pelos pontos, nas diferentes imagens, representariam a projeção do ponto da imagem, seja esta a da esquerda ou da direita, sobre o plano da imagem da outra câmera com a qual se faz a correspondência. Assim se definem as seguintes equações:

$$l' = F x$$

$$l = F^T x'$$

Com esta informação sobre a geometria epipolar, foi desenvolvida nossa primeira estratégia de ação, que é encontrar as matrizes fundamentais que relacionam as vistas das câmeras web com que trabalhamos. Isto é feito combinando duas a duas câmeras e achando a matriz fundamental para cada duas

vistas, num próximo passo, utilizaremos essas matrizes fundamentais para a geração de linhas epipolares que nos permitam fazer a correspondência entre os pontos na imagem que representam nossos marcadores.

Nosso primeiro passo foi então:

- Recuperar os pontos capturados, em coordenadas da imagem, para calcular a matriz fundamental entre duas câmeras.
- Aplicar o algoritmo RANSAC [30] para encontrar as matrizes fundamentais entre as câmeras.

O algoritmo RANSAC é um algoritmo iterativo que se divide em duas fases. Na primeira, ele faz um cálculo a priori da matriz fundamental, o qual é realizado com um algoritmo linear. Além disso, ele faz uma escolha randômica de pontos, dentro da amostra capturada, com os quais é feito o cálculo da matriz.

Em uma segunda fase, o algoritmo começa a agir iterativamente, testando a precisão da matriz fundamental que achou, o que é feito calculando a distância entre as linhas epipolares criadas para os pontos com quais se faz a correspondência dentro da mesma amostra inicial [31, 32], este algoritmo está implementado dentro da biblioteca OpenCV[23] que utilizamos em nossa implementação.

Com as matrizes fundamentais encontradas, temos a principal ferramenta para aplicar nossa estratégia de utilizar as linhas epipolares como método de correlação dos marcadores entre as diferentes imagens.

Então, em um próximo passo, depois de fazer o cálculo das matrizes fundamentais entre as câmeras, pode-se começar o processo de correlação. A estratégia usada foi a seguinte:

- Primeiro, escolhe-se uma câmera como pivô de nosso sistema de captura. Em nosso dispositivo, foi escolhida a câmera do centro como pivô do sistema de 3 câmeras web (Figura 34).
- Segundo, utilizando os pontos 2D que representam os marcadores, na câmera pivô e nas matrizes fundamentais, entre nosso pivô e as outras câmeras, calculamos as linhas epipolares projetadas sobre as câmeras da direita e da esquerda.
- Terceiro, com as linhas já projetadas sobre os planos da imagem da câmera da esquerda e da direita, executamos um algoritmo de

correlação entre os pontos 2D que em cada câmera subordinada representam os marcadores e as linhas epipolares projetadas a partir do pivô.

- Finalmente, cria-se uma lista de pontos correlacionados, os quais serão enviados para a etapa de reconstrução onde se fará o descarte final das correlações que possam estar erradas.

### 3.2.3.3. Estratégia para a Correlação

A descrição da estratégia seguida para esta etapa de correlação está dividida em 4 passos:

**Passo 1:** Seleção de uma câmera pivô.

O dispositivo implementado é formado por 3 câmeras web que trabalham em conjunto no processo de correlação, mas é preciso escolher uma delas como pivô para gerar as linhas epipolares sobre as câmeras subordinadas (Figura 35).

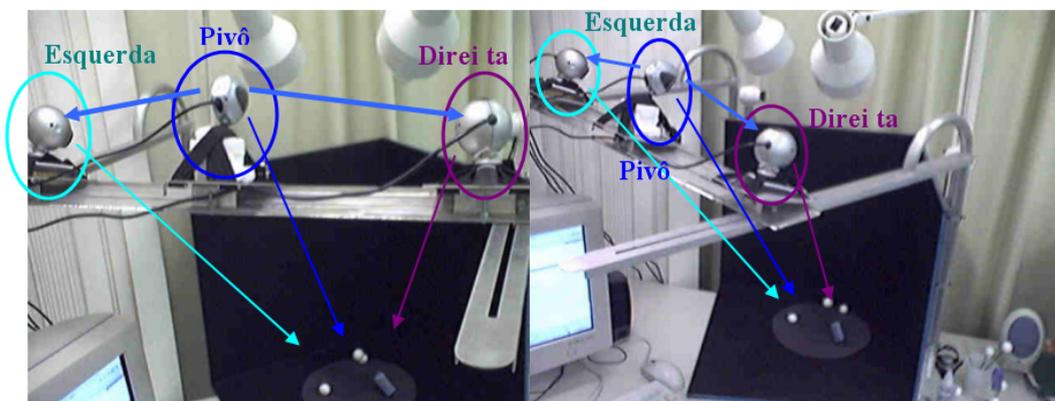


Figura 35 : Posicionamento das câmeras no dispositivo proposto.

A câmera do centro foi escolhida como pivô devido às seguintes considerações:

- Esta câmera tem um maior campo de visão de nossa caixa que representa o espaço de rastreamento efetivo de que dispomos.
- Ela está colocada ligeiramente acima do plano base das câmeras subordinadas da direita e da esquerda.

O motivo pelo qual as câmeras foram colocadas em diferentes alturas foi porque, no momento de fazer a correlação, tínhamos o problema de ter muitos casos em que mais de um marcador caía sobre uma mesma linha epipolar (Figura 36).

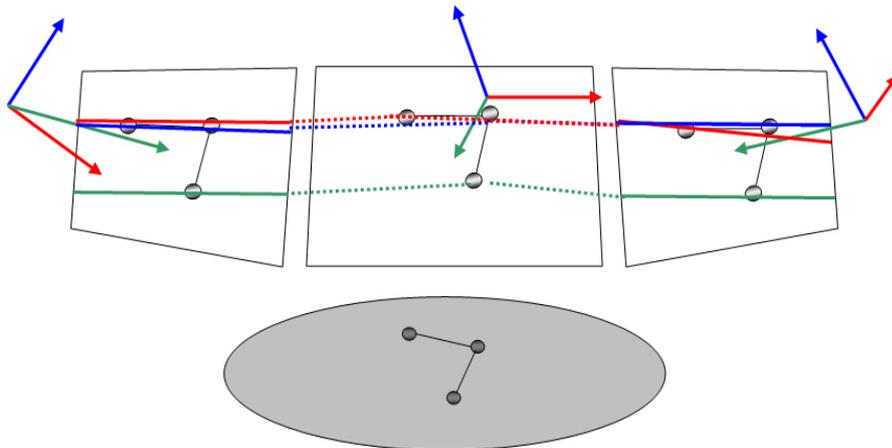


Figura 36 : Planos da imagem com as câmeras posicionadas numa mesma altura.

Como pode-se ver na Figura 36, se colocamos as câmeras com seus planos de imagem a uma mesma altura, começamos a ter problemas com a correspondência entre as linhas epipolares e os marcadores, havendo casos em que em uma mesma linha epipolar caem dois ou mais marcadores.

O que acontece nesses casos é que os marcadores caem em uma mesma linha de interseção criada entre o plano definido pela triangulação dos marcadores nas imagens e o marcador físico no mundo com o plano da imagem de cada câmera.

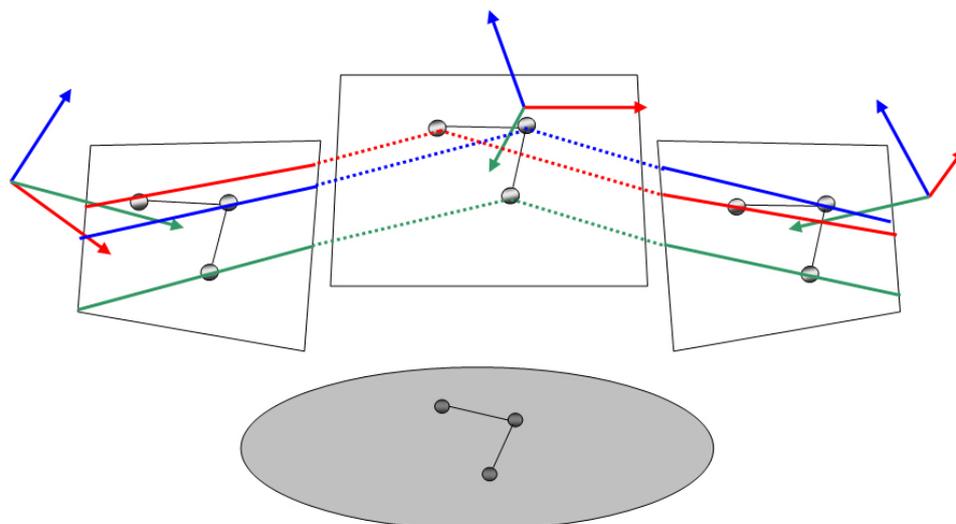


Figura 37 : Planos da imagem com as câmeras posicionadas em diferentes bases, ou seja, diferentes alturas.

Na Figura 37, mostra-se a vantagem de colocar as câmeras em diferentes alturas. Ali podemos ver que as linhas ficam mais bem definidas para a correspondência com um marcador. Isto porque os planos que definem os marcadores não ficam sobrepostos, portanto a correspondência entre linha e marcador é mais fácil de se perceber visualmente.

Ainda teremos casos em que os planos se sobrepõem, mas sua possibilidade de acontecer será menor. Na Figura 38 vemos como as câmeras web usadas no dispositivo ficam em diferentes alturas.

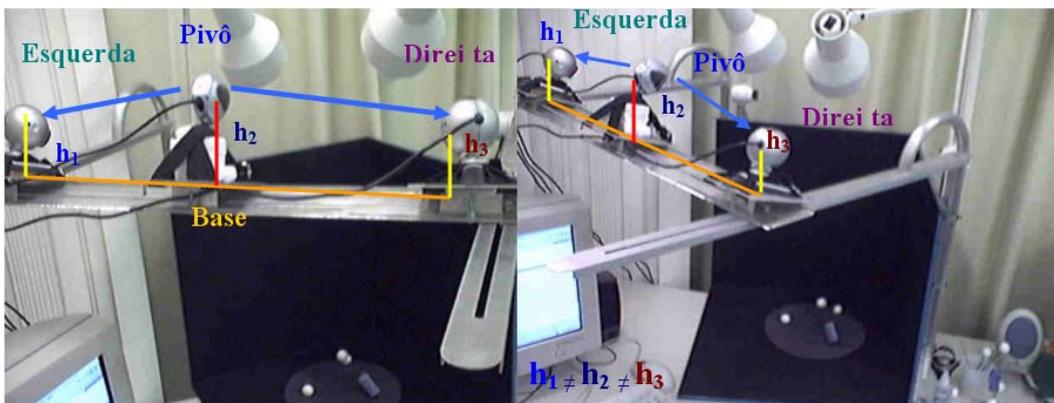


Figura 38 : O posicionamento das câmeras é em bases diferentes, ou seja, diferentes alturas.

### Passo 2: Cálculo das linhas epipolares.

Como nosso segundo passo, temos que calcular as linhas epipolares projetadas a partir dos marcadores encontrados na câmera pivô (Figura 39).

O processo de geração das linhas é o seguinte:

- Primeiro, tomam-se os pontos 2D que são as posições em coordenadas da imagem dos marcadores encontrados na imagem da câmera pivô.
- Segundo, para encontrar as linhas, tomamos os pontos e os multiplicamos pelas matrizes fundamentais encontradas, tanto para a câmera da esquerda como para a câmera da direita.

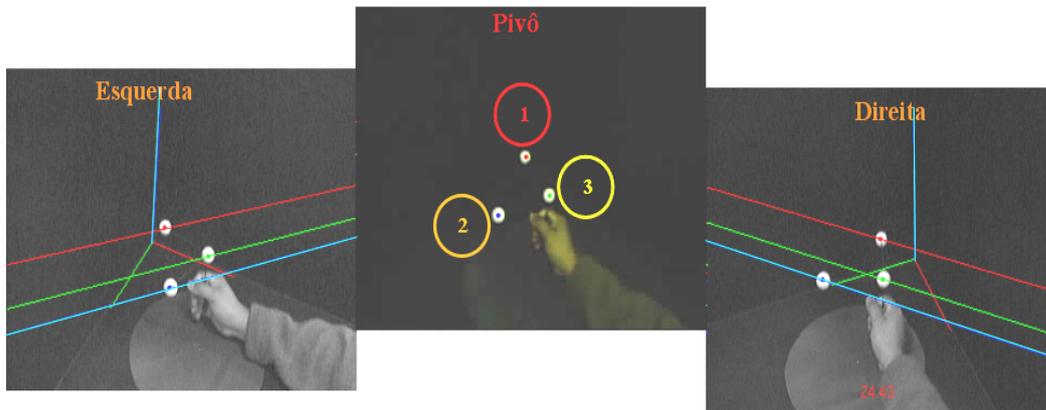


Figura 39 : Geração de linhas epipolares pela câmera pivô e projetadas sobre as câmeras dos lados.

Assim temos por definição:

$$p_{Dir}^T F_1 p_{Pivô} = 0 \quad p_{Esq}^T F_2 p_{Pivô} = 0$$

Onde  $p_{Esq}$ ,  $p_{Pivô}$  e  $p_{Dir}$  são as coordenadas da imagem do marcador encontradas entre duas câmeras, seja entre a câmera pivô e a câmera da esquerda ou a câmera da direita.

Assim, define-se que:

$$u_{Dir} = F_1 p_{Pivô} \quad u_{Esq} = F_2 p_{Pivô}$$

Onde  $u_{Esq}$  é a equação da linha epipolar projetiva que corresponde ao ponto  $p_{Pivô}$  projetado no plano da imagem da esquerda, e  $u_{Dir}$  é a equação da linha epipolar projetiva que corresponde ao ponto  $p_{Pivô}$  projetado no plano da imagem da direita.

**Passo 3:** Algoritmo de correlação entre ponto e linha.

Como terceiro passo, dentro do plano de imagem, foi implementado um algoritmo de correlação baseado em um teste entre a distância de um ponto a uma linha determinada dentro do plano.

Na aplicação prática deste algoritmo, encontraram-se alguns casos especiais, onde um caso bem difícil de definir, foi quando dois pontos 2D

(posições dos marcadores no plano da imagem) caem sobre uma mesma linha epipolar. É a partir deste caso que foi definido o seguinte algoritmo:

### **Algoritmo de Correlação**

#### ***Dados de Entrada:***

- Lista de pontos 2D correspondentes aos marcadores dentro da imagem.
- Linha epipolar correspondente a um marcador específico na imagem da câmera pivô.

#### ***Dados de Saída:***

- Lista final de índices dos pontos ou ponto que estiver mais perto da linha.

#### ***Variáveis temporais:***

- Dois raios de círculos limites, utilizados como grau de precisão para o processo de definir se uma distância entre o ponto e a reta é aceitável ou não.
- Lista de candidatos inicial.
- Lista de candidatos de alta precisão.
- Índice para guardar uma referência ao ponto com menor distância.

#### ***Seqüência do algoritmo:***

- Para cada ponto testar a distância entre ele e a linha testada.
- **Se** a distância for menor que o comprimento do primeiro raio:
  - Guardar o índice do ponto dentro de uma lista de candidatos inicial.
  - **Se** a distância obtida for menor que o comprimento do segundo raio:
    - Guardar o índice do ponto dentro de uma lista de candidatos de alta precisão.
  - **Se não então**, não guardar índice na lista de alta precisão .
- **Se não** for menor, **então** testar:
  - **Se** a distância calculada é a menor distância dentre as distâncias achadas.

- Guardar o índice do ponto que tiver a menor distância e que está fora dos raios definidos.
  - **Se não então**, ignorar a distância.
- Depois, preencher as listas e verificar os seguintes casos:
- **Se** as listas de alta precisão e a lista de candidatos inicial estiverem vazias:
  - Guardar como índice válido, dentro da lista final, o índice do ponto com a menor distância.
- **Se não**, ou seja, se as listas não estiverem vazias, **então** testar:
  - **Se** a lista de candidatos inicial tiver um candidato ou se a lista de precisão tiver um candidato ou se ambas tiverem um só candidato:
    - Guardar como índice válido, dentro da lista final, o índice da lista de alta precisão ou o índice da lista inicial.
  - **Se não então**:
    - Guardar como índices válidos, dentro da lista final, todos os índices contidos dentro da lista de candidatos inicial.
- Enviar de volta a lista final, com os índices que estiverem definidos como mais perto da linha.

O algoritmo de correlação, como resultado, fornecerá uma lista de índices candidatos que, dentro do plano da imagem, foram os que tiveram uma distância de 2 a 5 pixels entre eles e a linha testada, estes dois valores em pixels foram os limiares definidos como raios de precisão e alta precisão.

A estratégia de implementar dois círculos como áreas de restrição foi utilizada porque, como se trabalha com coordenadas discretas referentes ao plano da imagem, nossos pontos não caem exatamente sobre as linhas epipolares projetadas.

Além disso, temos o caso no qual dois pontos na imagem podem cair sobre uma mesma linha epipolar.

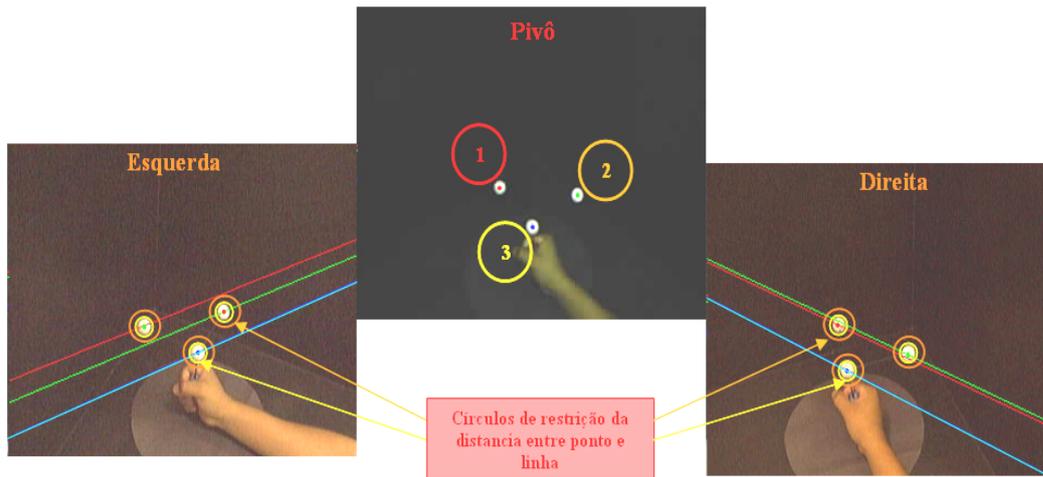


Figura 40 : Visualização da estratégia de círculos de restrição.

Como mostrado na Figura 40, a estratégia de colocar círculos restritivos, para definir se um ponto está perto ou não de uma linha, ajuda a detectar os casos especiais de sobreposição de 2 marcadores sobre uma mesma linha epipolar (imagem da direita). Para esses casos são guardados numa lista final os vértices que ficaram com a distância menor ou igual ao raio dos círculos restritivos.

#### **Passo 4:** Criação de lista de pontos correlacionados

Como passo final, será retornada uma lista de pontos correlacionados, a qual será nossa correspondência entre os marcadores que aparecem na câmera pivô e os marcadores que aparecem nas câmeras da esquerda e da direita.

Essa lista nos fornece uma correspondência um a um entre os marcadores em imagens tomadas duas a duas, ou uma correspondência entre um e vários marcadores, nos casos especiais.

O resultado pode ser visto na Tabela 1, com a correspondência dos marcadores em cada imagem, o que é mostrado na Figura 41:

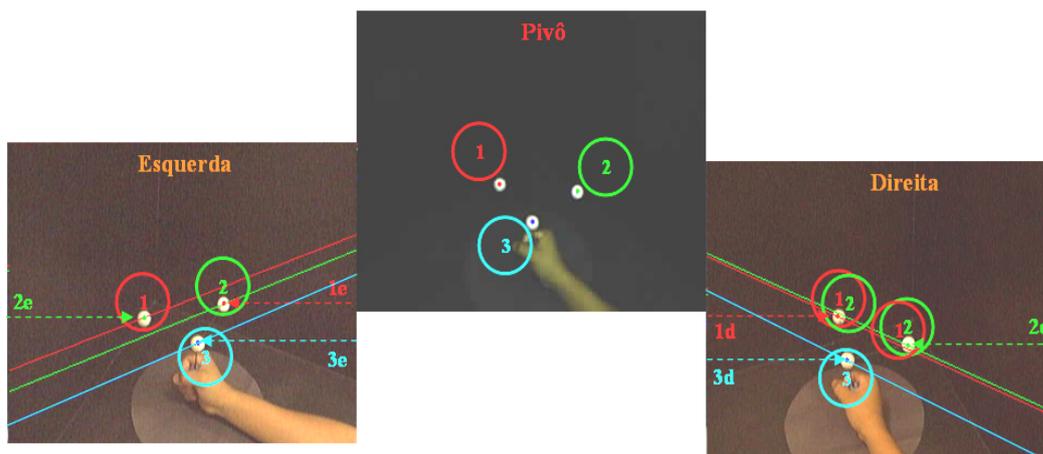


Figura 41 : Visualização dos índices dos marcadores de cada câmera e a correspondência encontrada para cada linha.

Marcadores Câmera Pivô	Marcadores Câmera Pivô	Marcadores Câmera Esquerda	Marcadores Câmera Direita
Índices para o 1º Marcador	1	2e	1d, 2d
Índices para o 2º Marcador	2	1e	1d, 2d
Índices para o 3º Marcador	3	3e	3d

Tabela 1 : Lista de índices dos marcadores correlacionados em relação à câmera pivô.

Na Figura 41, pode-se ver a flexibilidade no momento de definir a relação entre um ponto e uma linha. No caso do primeiro e do segundo marcador, na câmera da direita, temos mais de um marcador que cai sobre duas linhas epipolares que são projetadas (Tabela 1). Esse tipo de inconsistência será resolvido em uma etapa seguinte, na reconstrução 3D dos marcadores.

### Passo 5: Definição de heurísticas auxiliares

Adicionalmente ao processo de captura e correlação de marcadores foram implementadas, nesta etapa de correlação, duas heurísticas auxiliares orientadas a resolver dois casos especiais que acontecem no momento da captura e que podem gerar problemas na correlação.

Tais casos são:

- Quando na câmera escolhida como pivô não é possível capturar o número total dos marcadores com que se está trabalhando. Nossa experiência trabalha com 3 marcadores.

- Quando em qualquer câmera se perde pelo menos um marcador por causa de oclusão (Figura 42).

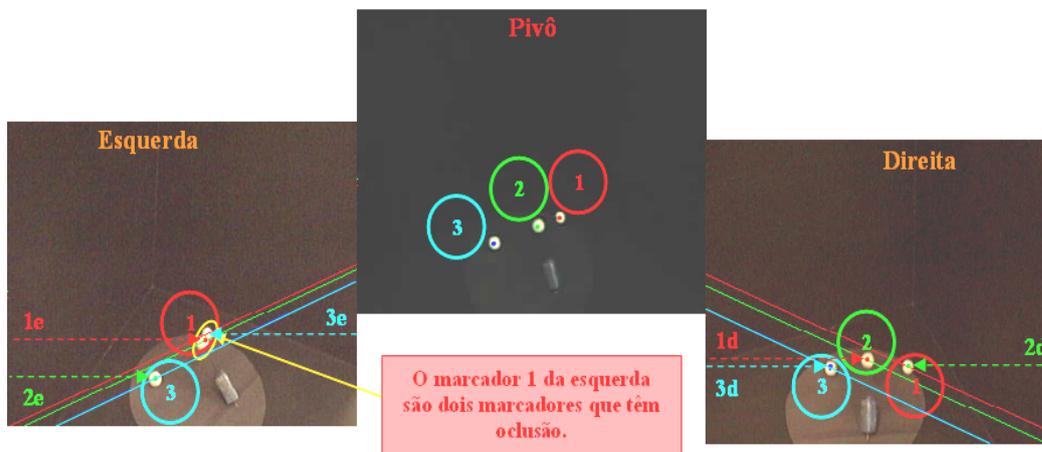


Figura 42 : Na câmera da esquerda, dois marcadores se ocultam mutuamente e são detectados como um só marcador.

Para o primeiro caso, a solução planejada foi que, antes de começar o processo de correlação, se verifique se na câmera escolhida como pivô há o número de marcadores com os quais se está trabalhando, neste caso 3 marcadores. Se houver um número menor de marcadores encontrados na imagem, é preciso trocar de câmera pivô, isto é, verificar nas outras câmeras se alguma detectou os 3 marcadores na sua imagem. Logo que se encontrar uma câmera que detectou todos os marcadores, troca-se a câmera pivô atual por essa câmera, são redefinidas a câmera da direita e da esquerda da nossa nova câmera pivô.

No segundo caso, se alguma câmera detectar, na captura, um número menor de marcadores, é porque aconteceu um caso de oclusão de um ou mais marcadores. Aqui colocaremos a seguinte restrição para enfrentar esse problema:

- Testa-se em todas as câmeras a quantidade de marcadores encontrados.
- Só são permitidos como estados aceitáveis para correlação aqueles em que no máximo uma câmera tenha uma quantidade menor de marcadores detectados do que a estabelecida inicialmente. Nosso ambiente tem 3 marcadores e o limite ficou em 2 câmeras com todos os marcadores bem identificados nas suas imagens.

Assim podemos chegar a ter a seguinte tabela, que representa o caso mostrado na Figura 42:

Marcadores Câmera Pivô	Marcadores Câmera Pivô	Marcadores Câmera Esquerda	Marcadores Câmera Direita
Índices para o 1º Marcador	1	1e	2d
Índices para o 2º Marcador	2	1e	1d
Índices para o 3º Marcador	3	2e	3d

Tabela 2 : Lista de índices com caso de oclusão de um marcador na câmera da esquerda.

Como pode-se ver na Tabela 2, com a informação resgatada das imagens nas 3 câmeras (Figura 42) ainda podemos fazer a correspondência e definir candidatos a serem o 1º, o 2º ou o 3º marcador em cada imagem em razão da ordem dos marcadores na câmera pivô.

### 3.2.4. Reconstrução 3D

Nossa quarta etapa na implementação do sistema de rastreamento óptico é a reconstrução 3D dos diferentes marcadores que aparecem nas imagens das câmeras web. A Figura 43 mostra a reconstrução de um marcador.

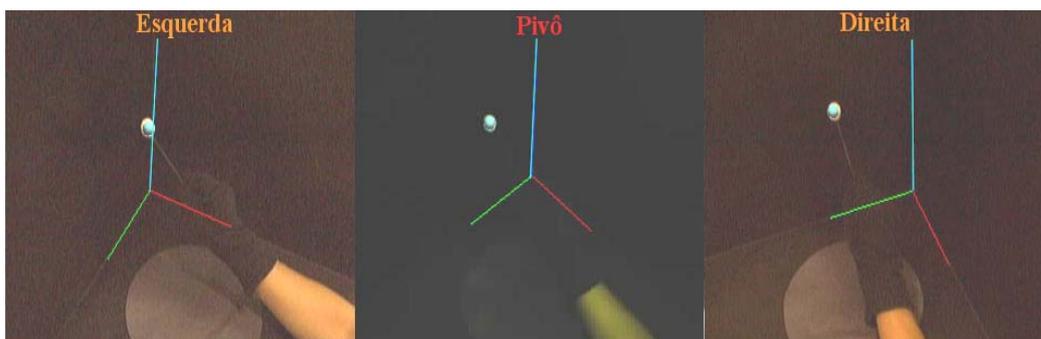


Figura 43 : Reconstrução da posição 3D de cada marcador.

Para fazer a reconstrução tomam-se como base os pontos em coordenadas da imagem que vêm da etapa de extração de características e as tabelas de correlação criadas na etapa de correspondência.

A estratégia utilizada nesta etapa foi:

- Primeiro, recuperar as coordenadas 2D dos marcadores em cada imagem, utilizando a lista de índices criada na etapa de correlação.
- Segundo, recuperar as matrizes de parâmetros intrínsecos e extrínsecos, calculados na etapa de calibração das câmeras.
- Terceiro, aplicar o algoritmo de reconstrução entre a câmera pivô e as câmeras da esquerda e da direita para calcular a posição 3D, em coordenadas da câmera pivô, de cada marcador definido.
- Quarto, criar uma lista de pontos 3D, reconstruídos em coordenadas da câmera pivô, para cada marcador.
- Finalmente, aplicar uma heurística para eliminar da lista de pontos candidatos possíveis candidatos repetidos ou que são muito semelhantes.

Na seqüência é detalhada essa estratégia de reconstrução.

### 3.2.4.1. Recuperação de Dados de Entrada

Como primeiro e segundo passos, temos a recuperação das coordenadas na imagem dos marcadores e das matrizes de parâmetros intrínsecos e extrínsecos de cada câmera, encontrados na etapa de calibração. A Figura 44 mostra as matrizes de parâmetros intrínsecos e extrínsecos.

$$P \begin{bmatrix} u' \\ v' \\ w \end{bmatrix} = \begin{bmatrix} -f_x & 0 & O_x \\ 0 & -f_y & O_y \\ 0 & 0 & 1 \end{bmatrix} P_c \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (a)$$

$$P_c \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} P_w \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (b)$$

$$\begin{bmatrix} u' \\ v' \\ w \end{bmatrix} = \begin{bmatrix} -f_x & 0 & o_x \\ 0 & -f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix} \quad (c)$$

Figura 44: (a) Matrizes de parâmetros intrínsecos, (b) parâmetros extrínsecos, (c) conversão de ponto no mundo para ponto na imagem.

A Figura 45 mostra as coordenadas 2D dos marcadores recuperadas das imagens.

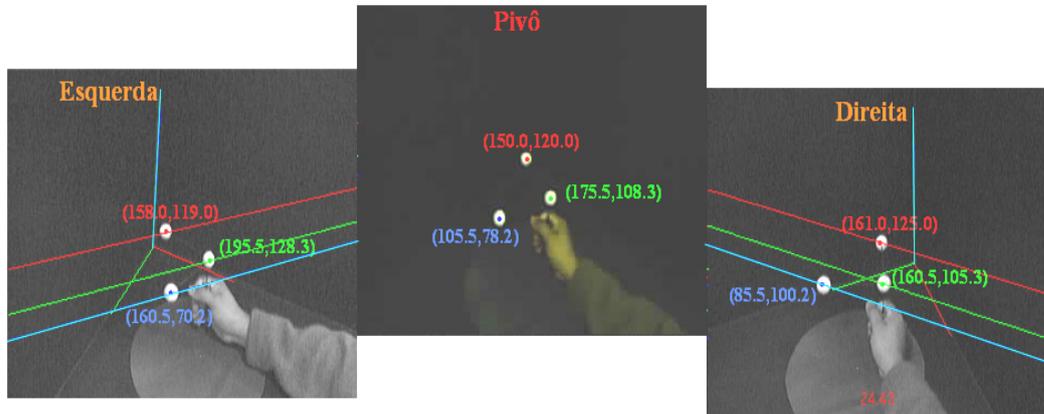


Figura 45 : Coordenadas 2D de cada marcador em cada câmera.

Estes dados são de grande importância no momento de aplicar o algoritmo de reconstrução. Além deles, é importante também a tabela de correlação dos índices, pois será com base nessa ordenação que se fará a reconstrução para cada marcador na imagem.

#### 3.2.4.2. Método de Reconstrução

Como terceiro passo na estratégia definida, temos o cálculo das coordenadas 3D de cada marcador. A reconstrução será feita em coordenadas da câmera pivô. A seguir é definido o processo geral da reconstrução. A Figura 46 mostra o processo graficamente.

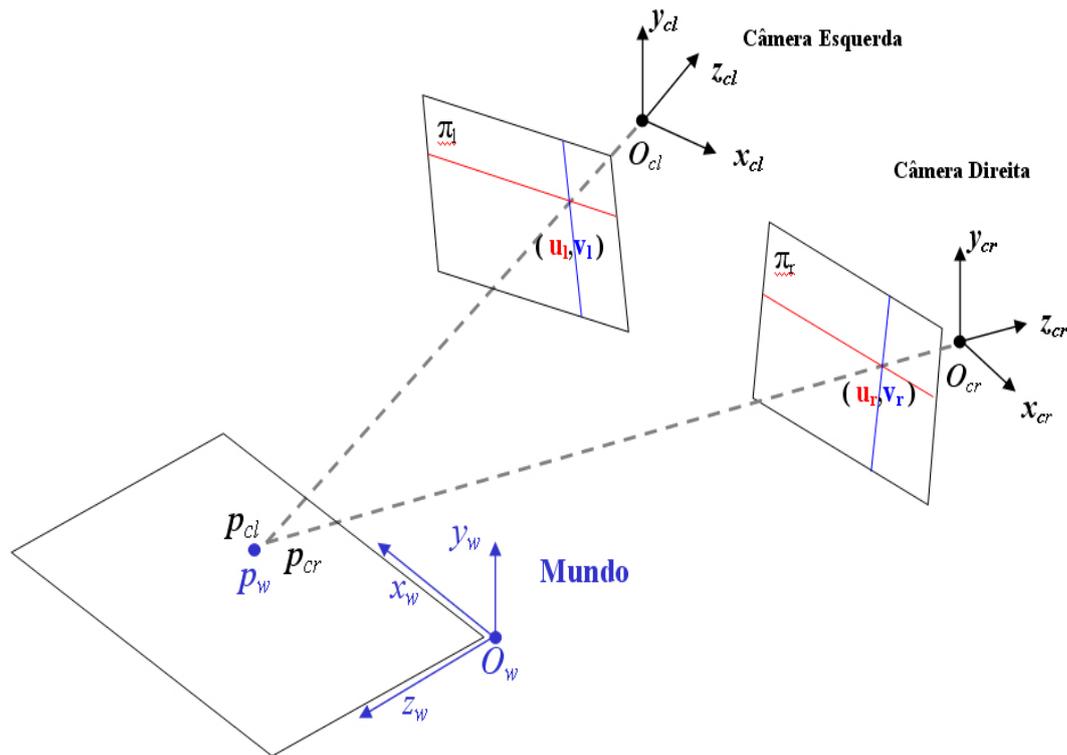


Figura 46 : Processo geral da reconstrução.

O ponto  $p_w$  em coordenadas do mundo também pode ser  $p_{cl}$  em coordenadas da câmera da esquerda ou  $p_{cr}$  em coordenadas da câmera da direita, onde:

$$p_w \xrightarrow{*[RT]} p_{cr} \xrightarrow{*[RT]} p_{cl} = \begin{bmatrix} Xc_l & Yc_l & Zc_l & 1 \end{bmatrix}^T$$

A idéia é que, como temos as conexões entre a câmera da esquerda e a câmera pivô, e entre a câmera da direita e a câmera pivô, ou seja, uma relação das câmeras tomadas duas a duas, tendo como base no processo a câmera pivô executaremos o algoritmo de reconstrução tomando as câmeras duas a duas. Como mostrado na Figura 46, para o processo são necessárias duas câmeras. No caso de nosso dispositivo, a câmera pivô sempre fará parte do processo, ou seja, teremos uma reconstrução feita entre a câmera da esquerda e a câmera pivô, e outra reconstrução entre a câmera da direita e a câmera pivô .

A seguir o método de reconstrução 3D definido em [33,34] e que utilizaremos será descrito passo a passo:

- Para a câmera da esquerda temos:

$$\begin{bmatrix} w_l & u_l & w_l & v_l & w_l \end{bmatrix}^T = P_l \begin{bmatrix} Xc_l & Yc_l & Zc_l & 1 \end{bmatrix}^T \quad (6)$$

- Para a câmera da direita temos:

$$\begin{bmatrix} w_r & u_r & w_r & v_r & w_r \end{bmatrix}^T = P_r \begin{bmatrix} Xc_l & Yc_l & Zc_l & 1 \end{bmatrix}^T \quad (7)$$

- Tem-se que  $\begin{bmatrix} u & v \end{bmatrix}^T$  são as coordenadas em pixels, nas imagens esquerda e direita, de um marcador capturado e reconhecido, e  $\begin{bmatrix} Xc_l & Yc_l & Zc_l & 1 \end{bmatrix}^T$  é a coordenada na câmera da esquerda do ponto  $\begin{bmatrix} Xw & Yw & Zw \end{bmatrix}^T$  em coordenadas do mundo.

Nas equações eq.(6) e eq.(7) podemos ver algo em comum: ambas as equações trabalham com coordenadas da câmera da esquerda. Aqui surge uma indagação: como se pode, na equação eq.(7), levar um ponto expresso em coordenadas da câmera da esquerda para coordenadas da imagem da câmera da direita? Além disso, como converter esse ponto 3D, expresso em coordenadas da câmera esquerda, em coordenadas em pixels da câmera da direita?

A resposta é a composição que devem ter as matrizes  $P_l$  e  $P_r$ , que levam um ponto em coordenadas da câmera da esquerda para coordenadas em pixels em cada imagem das câmeras. Assim podemos descrever o seguinte:

- Para levar um ponto em coordenadas da câmera da esquerda, na equação eq.(6), para coordenadas em pixels da mesma câmera, precisamos apenas da matriz de parâmetros intrínsecos  $Mi_l$ , onde a matriz  $P_l$  deve ser de ordem  $3 \times 4$ , por isso incrementa-se o vetor coluna 0 da matriz  $Mi_l$ :

$$P_l = Mi_l \begin{bmatrix} I_{3 \times 3} & | & 0_{3 \times 1} \end{bmatrix}_{3 \times 4} = \begin{bmatrix} Mi_{3 \times 3} & | & 0_{3 \times 1} \end{bmatrix}_{3 \times 4} \quad (8)$$

- Para levar um ponto, em coordenadas da câmera esquerda na equação eq.(7), para coordenadas em pixels da câmera direita, precisaremos da matriz de parâmetros intrínsecos da câmera direita  $Mi_r$ .

$$P_r = Mi_r \left[ R_{3 \times 3} \mid T_{3 \times 1} \right]_{3 \times 4} = \left[ \left[ Mi_r \mid R \right]_{3 \times 3} / \left[ Mi_r \mid T \right]_{3 \times 1} \right]_{3 \times 4} \quad (9)$$

Mas na equação eq.(7) ainda temos que converter um ponto em coordenadas da câmera da esquerda para coordenadas da câmera da direita. Para isso, é preciso calcular as matrizes  $[R \mid T]$  da equação eq.(9), que nos permitem fazer essa conversão, para o que temos nas seguintes equações, onde:

- $p_{cl}$  é um ponto em coordenadas da câmera da esquerda.
- $p_{cr}$  é um ponto em coordenadas da câmera da direita.
- $p_w$  é um ponto em coordenadas do mundo.

$$p_{cl} = R_l p_w + T_l \quad (10)$$

$$p_{cr} = R_r p_w + T_r \quad (11)$$

Aqui as matrizes  $R_l$  e  $R_r$  e os vetores  $T_l$  e  $T_r$  são os parâmetros extrínsecos calculados com o método de calibração TSAI não-coplanar. Resolvendo as equações eq.(10) e eq.(11):

- Expressamos  $p_w$  em coordenadas  $p_{cl}$  da eq. (10); assim temos:

$$R_l^T p_{cl} = R_l^T R_l p_w + R_l^T T_l$$

$$p_w = R_l^T p_{cl} - R_l^T T_l \quad (12)$$

- Trocamos na eq. (11)  $p_w$  pelo valor calculado na eq. (12) e temos:

$$p_{cr} = R_r \left( R_l^T p_{cl} - R_l^T T_l \right) + T_r$$

$$p_{cr} = \left( R_r R_l^T \right) p_{cl} + \left( -R_r R_l^T T_l + T_r \right)$$

$$p_{cr} = R p_{cl} + (-R T_l + T_r)$$

$$p_{cr} = R p_{cl} + T \quad (13)$$

- De onde também podem-se obter as fórmulas para encontrar  $R$  e  $T$  :

$$R = R_r R_l^T \quad \text{e} \quad T = -R T_l + T_r$$

A equação eq.(13) leva um ponto em coordenadas da câmera esquerda para coordenadas na câmera da direita.

Então, temos as matrizes  $[ R \quad | \quad T ]$  calculadas, as quais levam um ponto em coordenadas da câmera da direita para coordenadas da câmera esquerda. É assim que, na eq. (9), já temos todas as matrizes para calcular, e agora podemos expressar a eq. (6) e a eq. (7) da seguinte forma:

$$[ w_l u_l \quad w_l v_l \quad w_l ]^T = M_l [ I_{3 \times 3} \quad | \quad 0_{3 \times 1} ]_{3 \times 4} [ X_{c_l} \quad Y_{c_l} \quad Z_{c_l} \quad 1 ]^T \quad (14)$$

$$[ w_r u_r \quad w_r v_r \quad w_r ]^T = M_r [ R_{3 \times 3} \quad | \quad T_{3 \times 1} ]_{3 \times 4} [ X_{c_l} \quad Y_{c_l} \quad Z_{c_l} \quad 1 ]^T \quad (15)$$

- É a partir da eq. (14) e eq. (15) que armamos um sistema de equações para calcular o valor de:

$$[ X_{c_l} \quad Y_{c_l} \quad Z_{c_l} \quad 1 ]^T$$

- É assim que, da eq. (14) e eq. (15), podem-se extrair duas equações de cada igualdade:

$$\begin{bmatrix} w_r u_r \\ w_r v_r \\ w_r \end{bmatrix} = \begin{bmatrix} P_r^1 \\ P_r^2 \\ P_r^3 \end{bmatrix}_{3 \times 4} \begin{bmatrix} X_{c_l} \\ Y_{c_l} \\ Z_{c_l} \\ 1 \end{bmatrix}_{4 \times 1} \quad \begin{bmatrix} w_l u_l \\ w_l v_l \\ w_l \end{bmatrix} = \begin{bmatrix} P_l^1 \\ P_l^2 \\ P_l^3 \end{bmatrix}_{3 \times 4} \begin{bmatrix} X_{c_l} \\ Y_{c_l} \\ Z_{c_l} \\ 1 \end{bmatrix}_{4 \times 1}$$

$$B_{4 \times 4} = \begin{bmatrix} p_l^1 & -u_l p_l^3 \\ p_l^2 & -v_l p_l^3 \\ p_r^1 & -u_r p_r^3 \\ p_r^2 & -v_r p_r^3 \end{bmatrix}_{4 \times 4} \quad (16)$$

- Aqui,  $p_l^1, p_l^2, p_l^3, p_r^1, p_r^2, p_r^3$  são os vetores fila das matrizes  $P_l, P_r$ . A matriz  $B$  mostrada na eq. (16) forma o sistema  $Ax = 0$ :

$$B_{4 \times 4} \begin{bmatrix} Xc_1 & Yc_1 & Zc_1 & 1 \end{bmatrix}^T = 0_{4 \times 1} \quad (17)$$

- Uma solução para o sistema proposto pela eq. (17) é o autovetor associado com o menor autovalor recuperado da decomposição SVD. Esta solução é  $\begin{bmatrix} wXc_1 & wYc_1 & wZc_1 & w \end{bmatrix}^T$ , de onde podemos extrair o valor do marcador em coordenadas da câmera da esquerda:

$$\begin{bmatrix} Xc_1 & Yc_1 & Zc_1 & 1 \end{bmatrix}^T$$

### 3.2.4.3. Processo de Reconstrução

Conforme descrito no algoritmo de reconstrução, cada marcador é reconstruído fazendo uso de duas câmeras de cada vez e os pontos em coordenadas em pixels dos marcadores nessas câmeras.

É agora que utilizamos a lista de índices da etapa de correlação e, partir dessa lista, tomamos dois a dois os índices dos pontos indicados como pontos candidatos na imagem para serem um dos marcadores (isto entre a câmera pivô e as outras câmeras).

Utilizando a Tabela 1 correspondente à Figura 41, temos:

Marcadores Câmera Pivô	Marcadores Câmera Pivô	Marcadores Câmera Esquerda	Marcadores Câmera Direita
<i>Índices para o 1º Marcador</i>	1	2e	1d, 2d
<i>Índices para o 2º Marcador</i>	2	1e	1d, 2d
<i>Índices para o 3º Marcador</i>	3	3e	3d

Tabela 3 : Correspondência entre candidatos da câmera da esquerda e da direita com a câmera pivô.

Como mostrado na Tabela 3, para o marcador 1 na câmera pivô há dois possíveis pontos candidatos na câmera da direita e só um na esquerda. Isto significa que, na hora de aplicar a reconstrução tomando as câmeras duas a duas, geramos 1 ponto reconstruído entre a câmera pivô e da esquerda, e entre a câmera pivô e a câmera da direita há 2 candidatos reconstruídos.

É assim que, finalmente temos uma lista de pontos reconstruídos, como candidatos para cada marcador, conforme mostrado abaixo:

Marcadores Câmera Pivô	Numero Candidatos reconstruídos entre Câmera Pivô e da esquerda	Numero Candidatos reconstruídos entre Câmera Pivô e da direita	Totais candidatos por marcador
<i>Para o 1º Marcador</i>	1	2	3
<i>Para o 2º Marcador</i>	1	2	3
<i>Para o 3º Marcador</i>	1	1	2

Tabela 4 : Número de pontos reconstruídos entre as câmeras.

Como mostrado na Figura 47, entre os candidatos dos pontos reconstruídos temos candidatos com as posições corretas, mas também são gerados alguns em posição errada. Também geram-se posições de marcadores que estão tão perto, quase com as mesmas coordenadas, que na imagem é difícil distingui-los. Propomos duas soluções para esses casos:

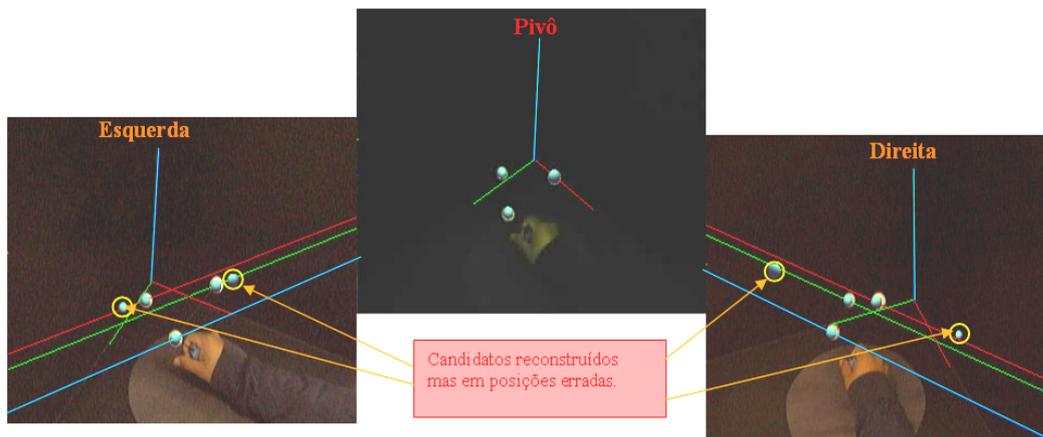


Figura 47 : Visualização de todos os marcadores reconstruídos.

1. Para os pontos que têm coordenadas 3D reconstruídas muito semelhantes, será feita uma média entre elas, deixando só uma coordenada 3D como candidato representando um marcador específico.
2. Para casos em que não é possível definir se dois pontos reconstruídos são semelhantes, será utilizada a estratégia de analisar características físicas do conjunto de marcadores, tais como o conhecimento da medida física da distância ou ângulo entre eles.

#### **Solução 1:** Descartar os candidatos semelhantes.

Visualmente podemos observar na Figura 47 que da câmera pivô e da direita os marcadores errados são facilmente eliminados com a primeira solução. Porém, devido à ambigüidade que se vê na imagem da câmera da esquerda, ainda se continua com a dúvida a respeito de quais são os marcadores certos e quais os errados. A Tabela 5 mostra o número de candidatos que sobram depois de executar a primeira solução.

Marcadores Câmera Pivô	Candidatos reconstruídos entre a câmera pivô e as câmeras da esquerda e direita
<i>Para o 1º Marcador</i>	2
<i>Para o 2º Marcador</i>	2
<i>Para o 3º Marcador</i>	1

Tabela 5 : Número de pontos 3D candidatos finais para cada marcador.

Finalmente, uma lista de pontos 3D candidatos, já sem repetições ou candidatos semelhantes, é enviada para se executar a segunda solução, na qual os pontos errados serão eliminados e se detectará o conjunto correto de marcadores como resultado final da reconstrução 3D dos marcadores.

**Solução 2:** Análise das características físicas do conjunto de marcadores.

Como segunda solução para eliminar candidatos de marcadores reconstruídos errados, definimos a identificação do conjunto de marcadores como um objeto só, ou seja, os marcadores definem uma forma específica.

No nosso caso, o conjunto de marcadores define um triângulo retângulo isósceles. Essa será a forma que os candidatos da lista final de pontos 3D devem gerar.

A forma predefinida para o conjunto de marcadores será a que nos ajudará a descartar aqueles candidatos a pontos 3D que representam posições erradas.

A estratégia definida nesta etapa é descrita a seguir:

- Primeiro, definir as características do objeto definido pelo conjunto de marcadores.
- Segundo, criar combinações com os pontos 3D candidatos, que procurem formar o objeto definido.
- Finalmente, definir e identificar uma só combinação de candidatos, os quais serão os pontos 3D definitivos, que representam os 3 marcadores que utilizamos como padrão a ser rastreado.

**Passo 1:** Definir as características físicas do conjunto de marcadores.

A seguir será definida a forma física que possui nosso conjunto de marcadores.

A primeira característica é que ele forma um triângulo unindo os 3 marcadores que o compõem. Esse triângulo possui um ângulo reto e é isósceles, ou seja, tem dois lados com igual tamanho.

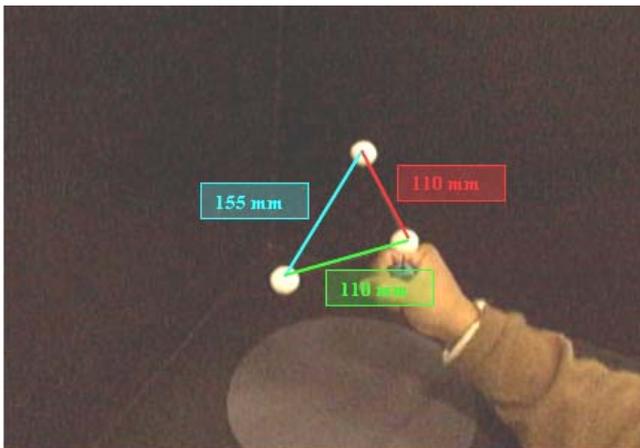


Figura 48 : Distribuição e forma definidas pelos marcadores físicos.

A característica física que usaremos serão as distâncias existentes entre os marcadores, como vemos na Figura 48. Temos 2 lados com 110 mm de tamanho e a hipotenusa do triângulo com aproximadamente 155.5 mm.

A decisão de utilizar a distância entre os marcadores e não o ângulo formado pelos lados foi porque nas experiências realizadas a margem de erro é mais difícil de controlar com medidas dos ângulos do que com as medidas das distâncias entre os marcadores.

**Passo 2:** Criação das combinações de pontos 3D candidatos.

Neste passo, definiremos como se faz a seleção entre as combinações dos pontos 3D candidatos encontrados na etapa final da reconstrução. Elas são testadas uma a uma, até que alguma combinação possa cumprir a característica física previamente definida no conjunto de marcadores.

Como mostra a Figura 49, as combinações são criadas unindo os pontos 3D que temos como candidatos. Assim vemos que as linhas vermelhas e verdes definem os lados que devem ser iguais e as linhas azuis definem a hipotenusa.

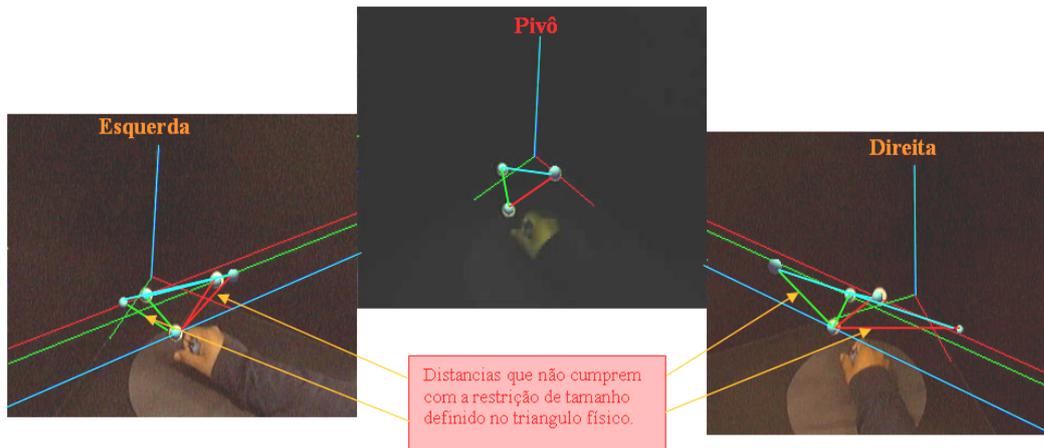


Figura 49 : Criação de combinações entre pontos candidatos.

A estratégia consiste então em criar vetores entre os pontos 3D, em coordenadas do mundo em milímetros, e testar se o tamanho dos vetores está dentro de um intervalo permitido de distâncias, em razão do tamanho dos lados do triângulo. Assim temos:

- Para os lados do triângulo que são iguais, permitimos uma distância de [100, 135] mm.
- Para o lado da hipotenusa do triângulo, permitimos uma distância de [150, 180] mm.

Com esses intervalos restritivos, podemos escolher dentre todas as possíveis combinações aquela que se aproxime melhor das distâncias físicas definidas pelos marcadores no mundo.

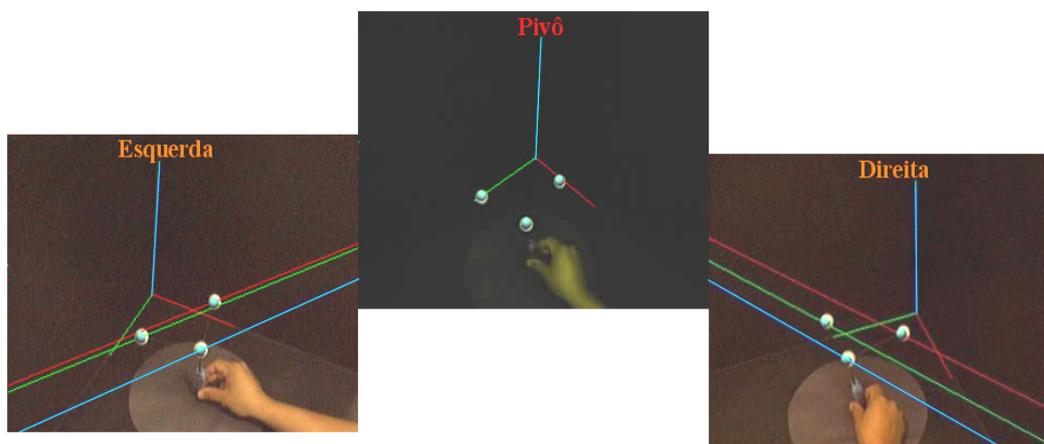


Figura 50 : Candidatos 3D escolhidos no final do processo.

O resultado final terá 3 pontos 3D entre os candidatos, que são escolhidos como correspondentes aos 3 marcadores físicos que se está rastreando, definindo nosso triângulo isósceles. Na Figura 50, os candidatos escolhidos são visualizados sobrepostos aos marcadores físicos que aparecem em cada imagem de vídeo.

Este resultado final da reconstrução 3D dos marcadores nos servirá na próxima etapa para rastrear seus movimentos como um objeto único, a partir do qual será feita a recuperação dos 6 graus de liberdade que este objeto tem dentro do espaço de trabalho.