# Calibração robusta de vídeo para realidade aumentada

Bruno Madeira
Instituto Militar de Engenharia
Rio de Janeiro, RJ
madeira@de9.ime.eb.br

Luiz Velho
Instituto Nacional de
Matemática Pura e Aplicada
Rio de Janeiro, RJ
lvelho@impa.br

Paulo Cezar Carvalho
Instituto Nacional de
Matemática Pura e Aplicada
Rio de Janeiro, RJ
pcezar@impa.br

## Resumo

*Neste artigo descrevemos um algoritmo robusto capaz de determinar os parâmetros extrínsecos assumidos por uma câmera na captação dos quadros de um vídeo, dado que os parametros intrínsecos foram previamente estimados, e que esses não variam ao longo do tempo. Apresentamos no final do artigo os resultados do uso desse algoritmo na criação de um sistema capaz de fazer realidade aumentada em um vídeo.*

## 1. Introdução

O principal problema que precisa ser resolvido para o desenvolvimento de um sistema de realidade aumentada é a determinação dos parâmetros da câmera utilizados na captação dos quadros do vídeo que se deseja combinar com imagens sintéticas. Neste artigo descrevemos um algoritmo, composto por diversos procedimentos heurísticos baseados em visão computacional, que resolve esse problema. Para isso utilizam-se correspondências entre projeções de diversos pontos da cena sobre os diversos quadros do vídeo.

A cena precisa ser rígida, ou seja, os pontos da cena não podem ter sua posição modificada, pois as restrições impostas por essa propriedade sobre suas projeções é que torna possível a determinação dos parâmetros da câmera.

Tendo em vista que mesmo vídeos de curta duração são formados por centenas de quadros é necessário que a correspondência entre as projeções seja feita de forma automática. Técnicas de processamento de imagens utilizadas no estabelecimento de correspondências de projeções em quadros de um vídeo estão fora do escopo do artigo. Um algoritmo largamente empregado para esse propósito é o Kanade-Lucas-Tomasi ( KLT ), descrito detalhadamente em [8]. O preço pago pela automatização é a possibilidade de falha nas medições das projeções dos pontos, que torna necessário o uso de técnicas robustas.

Muitas das idéias utilizadas aqui são baseadas em [3]. Existem entretando grandes diferenças no que diz respeito a estratégia de robustecimento empregada. Além disso, no nosso caso foi assumida a hipótese que os parâmetros intrínsecos da câmera utilizada são conhecidos.

## 2. Modelo de câmera

Em sistemas de realidade aumentada são necessários modelos de câmeras apropriados para estimação de parâmetros e para síntese de imagens. O principal motivo para uma modelagem diferenciada para síntese de imagens é a necessidade de solucionar problemas de oclusão entre superfícies da cena. No caso de objetos *wire-frame* pode-se utilizar em ambas as situações o modelo que será descrito.

### 2.1. Câmera na origem

Para uma projeção perspectiva cujo centro de projeção está posicionado em $(0,0,0)^T$, e cujo plano de projeção é perpendicular ao eixo-$z$, temos que a transformação associada é $T_1 : S \subset \mathbb{R}^3 \rightarrow \mathbb{R}^2$, definida por

$$T_1\{(x,y,z)^T\} = \left(d\frac{x}{z}, d\frac{y}{z}\right)^T,$$

onde $S$ é o conjunto formado pelos pontos de $\mathbb{R}^3$ que não possuem a coordenada $z = 0$, e $d$ corresponde à distância entre o centro e o plano de projeção. Essa distância é denominada distância focal.

### 2.2. Câmera em posição genérica

A transformação correspondente a uma câmera posicionada de maneira arbitrária é dada pela composição $T_1 \circ T_2 : T_2^{-1}(S) \rightarrow \mathbb{R}^2$, onde $T_2 : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ é um movimento rígido definido por

$$T_2(x) = R(x - c),$$

em que $c$ é a posição do centro de projeção, e $R$ é uma matriz de rotação, que determina a orientação da câmera.

A matriz de rotação $R$ e o vetor $c$ podem ser parametrizados por 6 números reais, que correspondem aos parâmetros extrínsecos da câmera.

### 2.3. Câmera digital

No caso de câmeras digitais, temos que a imagem é projetada sobre uma matriz de sensores, que realizam uma amostragem da mesma. Essa amostragem define um novo sistema de coordenadas para a imagem projetada. A mudança de coordenadas da imagem é definida por uma transformação afim do plano $T_3 : \mathbb{R}^2 \to \mathbb{R}^2$, da forma,

$$T_3(x) = diag(m_x, m_y) + (x_0, y_0)^T,$$

onde $m_x$ e $m_y$ correspondem ao número de sensores por unidade de comprimento na direção $x$ e $y$ respectivamente, e o par $(x_0, y_0)^T$ corresponde ao ponto principal, que é a coordenada em pixels, da projeção ortogonal do centro de projeção sobre o plano de projeção.

### 2.4. Modelo projetivo

Podemos reescrever as transformações $T_1$, $T_2$ e $T_3$ como transformações projetivas $T_1 : \mathbb{R}P^3 \to \mathbb{R}P^2$, $T_2 : \mathbb{R}P^3 \to \mathbb{R}P^3$ e $T_3 : \mathbb{R}P^2 \to \mathbb{R}P^2$, obtendo as seguintes representações matriciais:

$$T_1 = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad T_2 = \begin{pmatrix} R & -Rc \\ 0^T & 1 \end{pmatrix} \text{ e}$$

$$T_3 = \begin{pmatrix} m_x & 0 & x_0 \\ 0 & m_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Nesse caso estamos considerando que os pontos de $\mathbb{R}^3$ são identificados com pontos do $\mathbb{R}P^3$ pela transformação $(x, y, z)^T \mapsto (x, y, z, 1)^T$, e uma identificação análoga é feita entre os pontos do $\mathbb{R}^2$ e do $\mathbb{R}P^2$. Utilizaremos essa identificação em todo o resto do texto.

### 2.5. Notação $K[R|t]$

É imediata a verificação de que as transformações projetivas $T_3 \circ T_1 \circ T_2 : \mathbb{R}P^3 \to \mathbb{R}P^2$ podem ser representadas pelo produto de uma matriz $3 \times 3$ por uma matriz $3 \times 4$, como mostrado abaixo

$$T_3 \circ T_1 \circ T_2 = \begin{pmatrix} dm_x & 0 & x_0 \\ 0 & dm_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & -Rc \end{pmatrix}.$$

Nesse caso é comum utilizar a notação compacta $K[R|-Rc]$ para expressar esse produto. Nessa notação $K$ corresponde a matriz $3 \times 3$ que especifica os parâmetros intrínsecos da câmera, e $[R|-Rc]$ corresponde a matriz $4 \times 3$ que especifica os parâmetros extrínsecos. É comum também o uso da notação $K[R|t]$, cuja única diferença para a notação anterior é que a posição do centro óptico da câmera não é explicitadada, tendo em vista que o produto $-Rc$ é substituído por um vetor $t \in \mathbb{R}^3$ , que representa a translação da câmera.

No que se segue vamos assumir que a câmera utilizada para capturar o vídeo não sofre modificação em seus parâmetros intrínsecos. Além disso, vamos considerar que a matriz $K$ correspondente já foi estimada previamente. Maneiras de fazer essa estimação podem ser encontradas em [9].

## 3. Definições

Adotaremos as seguintes definições:

1. **Par de pontos homólogos**
   Dado um par de imagens $(I_1, I_2)$, dizemos que $(x_1, x_2) \in \mathbb{R}P^2 \times \mathbb{R}P^2$ é um par de pontos homólogos associados ao par de imagens $(I_1, I_2)$ se existe um ponto $X \in \mathbb{R}P^3$, da cena, que se projeta em $I_1$ no ponto $x_1$, e se projeta em $I_2$ no ponto $x_2$.
2. **Vídeo**
   Um vídeo é uma família finita de imagens $(I)_n = (I_1, ..., I_n)$, onde cada imagem $I_k$ corresponde a um quadro captado por uma câmera. Tem-se ainda que a ordem definida pela indexação dos quadros corresponde a ordem em que os quadros foram captados pela câmera.
3. **Família de pontos homólogos**
   Dado um vídeo $(I)_n = (I_1, ..., I_n)$, dizemos que a família $(x)_n = (x_1, ..., x_n)$, onde $x_i \in \mathbb{R}P^2$, é uma família de pontos homólogos associada ao vídeo $(I)_n$ se existe um ponto $X \in \mathbb{R}P^3$, da cena, tal que a projeção de $X$ em $I_j$ é $x_j$, para todo $j \in \{1, ..., n\}$.
4. **Matriz de pontos homólogos**
   Uma matriz $M$, $m \times n$, formada por elementos de $\mathbb{R}P^2$, é uma matriz de pontos homólogos associada a um vídeo $(I)_n$ se cada uma de suas linhas define uma família de pontos homólogos associada a $(I)_n$. Com essa definição temos também que a j-ésima coluna de $M$ corresponde aos pontos homólogos do quadro $I_j$.
5. **Configuração**
   Uma configuração é um par $((P)_n, \Omega)$, onde $(P)_n = (P_1, \ldots, P_n)$ é uma família de câmeras e $\Omega = \{X_1, \ldots, X_m\}$, com $X_i \in \mathbb{R}P^3$, é um conjunto de pontos da cena.
6. **Explicação para famílias de pontos homólogos**
   Estabelecida uma tolerância $\varepsilon \in \mathbb{R}^+$, definimos que

uma explicação projetiva para uma família de pontos homólogos $(x)_n = (x_1, ..., x_n)$ é uma configuração $((P)_n, \Omega)$ tal que $\forall i \in \{1, ..., n\}$, $\exists X_j \in \Omega$ que satisfaz $\|P_i X_j - x_i\| < \varepsilon$. Nesse caso dizemos também que a configuração $((P)_n, \Omega)$ explica projetivamente a família de pontos homólogos $(x)_n$.

7. **Explicação para matrizes de pontos homólogos**
Uma explicação projetiva para uma matriz de pontos homólogos $M$ é uma configuração que explica todas as famílias de pontos homólogos das linhas de $M$. Nesse caso dizemos também que a configuração explica projetivamente a matriz de pontos homólogos $M$.

## 4. Calibração em três passos

Apresentaremos agora um algoritmo que encontra uma explicação projetiva $((P)_n, \{X_1, ..., X_m\})$ para uma matriz de pontos homólogos $M$ associada a um vídeo $(I)_n$.

O algoritmo é formado pelos seguintes passos:

1. Passo 1: Utilizar as colunas de $M$ correspondentes aos pontos homólogos de uma par de quadros $I_i$ e $I_j$ para determinar $P_i$ e $P_j$.
2. Passo 2: Utilizar o par $P_i$ e $P_j$ e a matriz $M$ para determinar o conjunto $\{X_1, ..., X_m\}$.
3. Passo 3: Utilizar o conjunto $\{X_1, ..., X_m\}$ e a matriz $M$ para determinar a família de câmeras $(P)_n$.

Os passos 1 e 3 são problemas de calibração de câmeras e o passo 2 é um problema de reconstrução tridimensional. Um estudo extenso e detalhado sobre esses problemas pode ser encontrado em [4]. Para simplificarmos nossa notação, chamaremos as colunas i e j da matriz $M$, escolhidas para a execução dos passos 1 e 2, de colunas base de $M$.

Mostraremos a seguir como os três passos do algoritmo podem ser reformulados de maneira a serem resolvidos pela proposição abaixo, que estabelece a solução para o problema de encontrar $x \in S^n$ que minimiza $\|Ax\|$, onde $A : \mathbb{R}^n \to \mathbb{R}^m$ é uma transformação linear. Uma prova para a proposição pode ser encontrada em [4].

**Proposição 1.** *Seja $Udiag(\lambda_1, \lambda_2, ..., \lambda_n)V^T$, com $\lambda_1 \geqslant \lambda_2 \geqslant ... \geqslant \lambda_n \geqslant 0$, a decomposição SVD de uma matriz A, $m \times n$ , em que $m \geqslant n$. Se $v \in \mathbb{R}^n$ é o vetor correspondente a n-ésima coluna de V, tem-se que $v$ é o vetor que minimiza a função $x \mapsto \|Ax\|$, definida sobre os pontos de $\mathbb{R}^n$ que satisfazem $\|x\| = 1$.*

Denotaremos esse problema de forma compacta por $\min_{\|x\|=1} \|Ax\|$.

## 5. Passo 1: Calibração de pares de câmeras

Para determinarmos $P_i$ e $P_j$ a partir das colunas bases de $M$ pode-se utilizar o algoritmo de oito pontos, apresentado inicialmente em [6], e cujo funcionamento pode ser facilmente compreendido pela proposição abaixo, apresentada em [7], que estabelece uma restrição para as coordenadas definidas em dois referenciais do $\mathbb{R}^3$, que estão relacionados por um movimento rígido.

**Proposição 2.** *Sejam $X \in \mathbb{R}^3$ e $X' \in \mathbb{R}^3$ definidos de forma que $X' = RX + t$, onde R é uma matriz de rotação e $t \in \mathbb{R}^3$. Se $[t]_\times : \mathbb{R}^3 \to \mathbb{R}^3$ é o operador linear definido por $[t]_\times (x) = t \times x$, então vale a relação $X'^T \left([t]_\times R\right) X = 0$*

**Demonstração**
Com efeito, usando o fato de o vetor $X' \times t$ ser perpendicular tanto a $X'$ quanto a $t$, temos que $(X' \times t) \cdot X' = 0$ e $(X' \times t) \cdot t = 0$. Como conseqüência vale
$X'^T \left([t]_\times R\right) X = X' \cdot (t \times RX) = (X' \times t) \cdot RX = (X' \times t) \cdot (RX + t) = (X' \times t) \cdot X' = 0$.

### 5.1. Matriz essencial

Definindo $E = [t]_\times R$, temos pela proposição 2 que vale a expressão $X'^T E X = 0$, que relaciona as coordenadas de um ponto da cena nos referenciais associados as câmeras $[I|0]$ e $[R|t]$. Para se obter uma relação entre as coordenadas das projeções desse ponto nas imagens captadas por essas câmeras, basta observar que para todo $\lambda_1, \lambda_2 \in \mathbb{R} - \{0\}$ vale

$$X'^T E X = 0 \iff \left(\lambda_1 X'^T\right) E \left(\lambda_2 X\right) = 0.$$

Temos então que se $x \in \mathbb{R}P^2$ e $x' \in \mathbb{R}P^2$ são as coordenadas homogêneas das projeções de um ponto da cena obtidas pelas câmeras $[I|0]$ e $[R|t]$ respectivamente, vale a relação $x'^T E x = 0$, onde nesse caso tem-se que a matriz $E$, chamada de matriz essencial, fica definida a menos de um produto por um escalar.

### 5.2. Matriz fundamental

Consideremos agora que $x \in \mathbb{R}P^2$ é a projeção de um ponto $X \in \mathbb{R}P^3$ obtida pela câmera $K\,[R \mid t]$. A projeção do mesmo ponto $X$ obtida pela câmera $[R \mid t]$ é dada por $K^{-1}x$. Com esse resultado podemos generalizar a relação estabelecida pela matriz essencial para o caso em que as câmeras não possuem a matriz dos parâmetros intrínsecos iguais a $I$. Mais precisamente, dadas duas câmeras $K_1\,[I \mid 0]$ e $K_2\,[R \mid t]$, temos que se as projeções de um ponto $X$ relativas a essas câmeras forem $x$ e $x'$ respectivamente, então vale a

relação $\left(K_2^{-1}x'\right)^T \left([t]_\times R\right)\left(K_1^{-1}x\right) = 0$. Essa relação pode ser reescrita como

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

onde $F = K_2^{-T}[t]_\times R K_1^{-1}$ é uma matriz $3 \times 3$, denominada matriz fundamental.

### 5.3. Cálculo da matriz fundamental

O algoritmo de oito pontos estima a matriz fundamental $F$ que relaciona duas colunas de $M$, $(M_{1i}, \ldots, M_{mi})^T$ e $(M_{1j}, \ldots, M_{mj})^T$, pela solução do sistema linear, definido sobre as 9 componentes de $F$

$$M_{ki}^T F M_{kj} = 0,$$

para $k \in \{1, \ldots, m\}$.

Como $F$ é definida a menos de um produto por um escalar, é necessário que tenhamos $m$ no mínimo igual a 8 para que a solução do sistema fique determinada. Se $m \geqslant 9$ pode-se reformular o problema como sendo o de encontrar a matriz $F_0$ que minimiza a função objetivo

$$F \mapsto \sum_{k=1}^{m} \left(M_{ki}^T F M_{kj}\right)^2,$$

que pode ser resolvido pela proposição 1 bastando para isso ser reescrito na forma $\underset{\|\mathcal{F}\|=1}{min} \|A\mathcal{F}\|$, com $\mathcal{F} = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^T$ e $A$ definida por

$$\begin{pmatrix} u'_1u_1 & v'_1u_1 & u_1 & u'_1v_1 & v'_1v_1 & v_1 & u'_1 & v'_1 & 1 \\ u'_2u_2 & v'_2u_2 & u_2 & u'_2v_2 & v'_2v_2 & v_2 & u'_2 & v'_2 & 1 \\ u'_3u_3 & v'_3u_3 & u_3 & u'_3v_3 & v'_3v_3 & v_3 & u'_3 & v'_3 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u'_mu_m & v'_mu_m & u_m & u'_mv_m & v'_mv_m & v_m & u'_m & v'_m & 1 \end{pmatrix},$$

onde $M_{ki} = (u_k, v_k, 1)^T$ e $M_{kj} = (u'_k, v'_k, 1)^T$.

Geralmente a matriz $F_0$ encontrada não é singular, que é uma propriedade que toda matriz fundamental satisfaz. Utiliza-se, então, a matriz singular $\tilde{F}$ que minimiza $\|\tilde{F} - F_0\|$ como estimativa para a matriz fundamental. A norma considerada nesse caso é a de Frobenius, pois dessa maneira a solução é obtida facilmente pela aplicação da proposição abaixo, cuja demonstração pode ser encontrada em [10]

**Proposição 3.** *Se $Udiag(r, s, t)V^T$ é a decomposição SVD de $F_0$, com $r \geqslant s \geqslant t$, então a matriz singular $\tilde{F}$ que minimiza $\|\tilde{F} - F_0\|$, é dada por $\tilde{F} = Udiag(r, s, 0)V^T$.*

Esse método de estimação de matrizes fundamentais é mal condicionado. Tal problema pode ser resolvido por uma simples normalização das coordenadas dos pontos homólogos, como descrito em [5].

### 5.4. Determinando os parâmetros extrínsecos

Mostraremos agora como resolver o problema de encontrar os parâmetros extrínsecos do par de câmeras $P_i$ e $P_j$ dado que são conhecidas as respectivas matrizes de parâmetros intrínsecos $K_i$ e $K_j$, e a matriz fundamental $F$, que correlaciona os pontos homólogos das imagens captadas por $P_i$ e $P_j$.

Inicialmente observamos que podemos definir uma matriz essencial $E = K_i^T F K_j$ que relaciona as projeções obtidas pelas câmeras $K_i^{-1}P_i$ e $K_j^{-1}P_j$.

Podemos assumir sem perda de generalidade que $K_i^{-1}P_i = [R \mid t]$ e que $K_j^{-1}P_j = [I \mid 0]$, sendo assim a matriz $E = [t]_\times R$ é o produto da matriz anti-simétrica $[t]_\times$, pela matriz de rotação $R$. A determinação dos possíveis valores de $t$ e $R$ fica resolvida pela proposição abaixo, cuja demonstração pode ser encontrada em [4]

**Proposição 4.** *Supondo que a decomposição SVD de uma matriz essencial $E$ é igual a $Udiag(1, 1, 0)V^T$, existem duas maneiras de fatorar $E$, de forma que $E = SR$, onde $S$ é uma matriz ainti-simétrica e $R$ é uma matriz de rotação. Tem-se que $S = UZU^T$ e $R = UWV^T$ ou $R = UW^TV^T$, onde*

$$W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ e } Z = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

A proposição 4 mostra que existem duas possíveis escolhas para a matriz de rotação $R$. Para determinarmos quais são os possíveis vetores $t$, basta levar em conta os seguintes fatos:

a) $[t]_\times t = t \times t = 0$.
b) Toda matriz essencial é definida a menos de uma multiplicação por um escalar.

Usando a notação da proposição, temos pelo primeiro fato, que todo vetor $t$ deve pertencer ao núcleo de $[t]_\times = UZU^T$, ou seja, $\exists \lambda \in \mathbb{R}$ tal que $t = \lambda U(0, 0, 1)^T$. O segundo fato mostra que, na realidade, $t$ pode ser qualquer elemento da forma $\lambda U(0, 0, 1)^T$, com $\lambda \in \mathbb{R}$.

Podemos reduzir o número de soluções utilizando o fato de existirem configurações $((P_i, P_j), \{X_1, \ldots, X_m\})$ que embora expliquem projetivamente os pontos homólogos, não são fisicamente realizáveis, como apresentado na Figura 1. A solução para esse problema consiste em descartar as configurações que fazem com que a reconstrução tridimensional de pontos homólogos possua a coordenada $z$ negativa para algum dos referenciais definidos pelas câmeras. Esse processo de reconstrução será explicado na próxima seção, que descreve o passo 2 do algoritmo.

Ao serem eliminadas as configurações não realizáveis, ficam determinados de maneira única a rotação $R$, a direção,

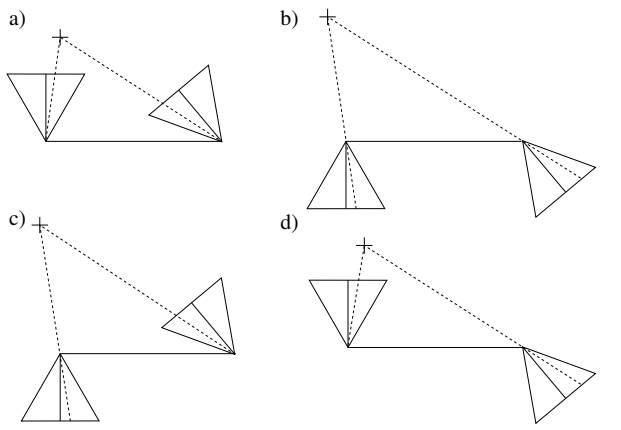


**Figura 1. Embora existam quatro configurações que explicam projetivamente o par de pontos homólogos, apenas em (a) o ponto projetado esta posicionado a frente de ambas as câmeras**

e o sentido do vetor $t$. O valor de $\|t\|$ continua sendo impossível de determinar devido a ambigüidade de escala inerente a qualquer processo de reconstrução tridimensional baseado em imagens.

## 6. Passo 2: Reconstrução tridimensional

O passo 2 do algoritmo encontra o conjunto de pontos da cena $\Omega = \{X_1, \ldots, X_m\}$ a partir dos pontos homólogos da i-ésima e j-ésima colunas da matriz de pontos homólogos $M$ e das câmeras $P_i$ e $P_j$, calculadas pelo passo 1.

Interpretando $M_{ki} = (u, v, 1)^T$ e $P_i X_k$ como vetores do $\mathbb{R}^3$, temos que $M_{ki} \times (P_i X_k) = 0$. Chamando de $P_i^n$ a n-ésima linha de $P_i$, pode-se reescrever essa expressão como o seguinte conjunto de três equações lineares em $X_k$, onde duas são linearmente independentes

$$\begin{aligned} u\left(P_i^3 X_k\right) - \left(P_i^1 X_k\right) &= 0, \\ v\left(P_i^3 X_k\right) - \left(P_i^2 X_k\right) &= 0, \\ u\left(P_i^2 X_k\right) - v\left(P_i^1 X_k\right) &= 0. \end{aligned}$$

Analogamente temos que $M_{kj} = (u\prime, v\prime, 1)^T$ pode ser utilizado para obtermos mais outras duas equações lineares em $X_k$, e linearmente independentes, bastando observar que $M_{kj} \times (P_j X_k) = 0$. Agrupando quatro dessas equações obtemos um sistema linear homogêneo da forma $AX_k = 0$ onde

$$A = \begin{pmatrix} uP_i^3 - P_i^1 \\ vP_i^3 - P_i^2 \\ u\prime P_j^3 - P_j^1 \\ v\prime P_j^3 - P_j^2 \end{pmatrix}.$$

Esse é um sitema linear de quatro equações sobre as quatro coordenadas homogêneas de $X_k$, logo é um sistema linear super-determinado, que pode ser convertido para o problema de otimização $\underset{\|X_k\|=1}{min} \|AX_k\|$, cuja solução é dada pela proposição 1.

## 7. Passo 3: Calibração de várias câmeras

O passo 3 encontra a família de câmeras $(P)_n = (P_1, \ldots, P_n)$ a partir da matriz de pontos homólogos $M$ e do conjunto de pontos $\Omega = \{X_1, \ldots, X_m\}$ determinado no passo 2.

Para resolver o problema basta observar que encontrar $P$ que satisfaz

$$\forall k \in \{1, \ldots, m\}, PX_k = (u_k, v_k, 1)^T$$

é equivalente a resolver o sistema linear super determinado $A\mathcal{P} = 0$, onde

$$A = \begin{pmatrix} X_1^T & 0^T & -u_1 X_1^T \\ 0^T & X_1^T & -v_1 X_1^T \\ X_2^T & 0^T & -u_2 X_2^T \\ 0^T & X_2^T & -v_2 X_2^T \\ \vdots & \vdots & \vdots \\ X_m^T & 0^T & -u_m X_m^T \\ 0^T & X_m^T & -v_m X_m^T \end{pmatrix}$$

e $\mathcal{P} = (P_{11}, P_{12}, P_{13}, P_{14}, P_{21}, \ldots, P_{33}, P_{34})^T$ é um vetor formado pelos doze elementos da matriz $P$ que precisam ser determinados.

Quando $\sharp\Omega \geqslant 6$ podemos utilizar a proposição 1 para resolver o problema de otimização $\underset{\|\mathcal{P}\|=1}{min} \|A\mathcal{P}\|$, cuja solução fornece uma estimativa para os elementos da matriz $P$. Esse processo pode ser aplicado repetidamente determinando cada uma das câmeras da família $(P)_n$.

## 8. Problemas da calibração em três passos

Uma implementação ingênua da calibração em três passos, descrita anteriormente, apresenta resultados ruins devido aos seguintes problemas:

1. Problema do passo 1: Podem ocorrer erros grosseiros durante a execução do passo 1 pois a matriz fundamental é estimada utilizando-se um conjunto de pontos homólogos que pode apresentar erros grosseiros, já que estamos considerando que esses são determinados automaticamente por um algoritmo de processamento de imagens que não oferece garantias sobre sua precisão ou correção.
2. Problema do passo 2: Podem ocorrer erros grosseiros durante a execução do passo 2 devido a problemas de condicionamento do processo de reconstrução, pois

é possível que o ponto da cena reconstruído seja tal que uma grande perturbação de sua posição em uma direção cause uma pequena modificação nas coordenadas das projeções obtidas pelas câmeras.

3. Problema do passo 3: O passo 3 não impõe a restrição dos parâmetros intrínsecos que são assumidos como sendo conhecidos, e que são usados no passo 1, quando se obtém a matriz essencial $E = K_i^T F K_j$, em 5.4.

Mostraremos como resolver esses problemas de maneira a tornar robusta a calibração feita em três passos. Para tal, faremos uso do algoritmo RANSAC.

## 8.1. Algoritmo RANSAC

O algoritmo RANSAC ( Random Sample Consensus ), foi proposto por Fischler e Bolles em [2], onde foi apresentado nos seguintes termos

*"Dados um modelo que precisa de um mínimo de n pontos para ter seus parâmetros livres instanciados, e um conjunto de pontos $P$, tal que o número de pontos de $P$ é maior do que $n$, isto é $\sharp(P) \geqslant n$. Selecione aleatoriamente um subconjunto $S_1$, de $n$ pontos de $P$ e instancie o modelo. Utilize o modelo instanciado $M_1$ para determinar um subconjunto $S_1^*$ de pontos de $P$, que satisfazem um critério de tolerância de erro em relação a $M_1$. O conjunto $S_1^*$ é chamado de conjunto de consenso de $S_1$.*

*Se $\sharp(S_1^*)$ for maior que um certo limiar t, que é função de uma estimativa do número de erros grosseiros em $P$. Use $S_1^*$ para computar ( possivelmente usando mínimos quadrados ) um novo modelo $M_1^*$.*

*Se $\sharp(S_1^*)$ for menor que t, selecione aleatoriamente um novo subconjunto $S_2$ e repita o processo acima. Caso depois de um número pré-determinado de iterações, nenhum conjunto de consenso com $t$ ou mais elementos tiver sido encontrado, encontre o modelo correspondente ao maior conjunto de consenso, ou termine acusando um erro."*

Apresentaremos a seguir como é possível utilizar o RANSAC para resolver os problemas dos passos 1 e 2. Utilizaremos a notação definida acima para tornar simples a identificação dos princípios do paradigma RANSAC.

## 8.2. Solução para o problema do passo 1

Podemos nesse caso considerar que o algoritmo de oito pontos fornece uma maneira de se obter uma matriz fundamental, que corresponde ao modelo $M_1$, a partir de um conjunto formado por oito pares de pontos homólogos correspondentes a $S_1$, obtidos nas colunas base de $M$.

Pode-se utilizar um critério de tolerância para definir o conjunto de consenso $S_1^*$ baseado na função objetivo do algoritmo de oito pontos, mais precisamente, dado um limiar $\eta_1 \in \mathbb{R}^+$ estabelecido empiricamente, incluimos em $S_1^*$ os pares de pontos homólogos $(x_i, x_j)$ das colunas base de $M$, se $|x_i'^T F x_j| \leqslant \eta_1$, onde $F$ é a matriz fundamental estimada usando o conjunto $S_1$. O modelo $M_1^*$ é uma matriz fundamental que pode ser obtida aplicando-se o próprio algoritmo de oito pontos sobre os pontos homólogos de $S_1^*$.

## 8.3. Solução para o problema do passo 2

Seja $Q$ o conjunto formado pelas reconstruções tridimensionais dos pares de pontos homólogos das colunas base de $M$, que fazem parte do conjunto de consenso encontrado durante a aplicação do RANSAC na estimação da matriz fundamental.

Para resolvermos o problema de condicionamento do passo 2 vamos utilizar o RANSAC durante a execução do passo 3. Para isso temos que o conjunto $\Gamma$, formado por seis pares $(X, m)$, faz o papel do modelo $S_1$, onde $X$ é um elemento de $Q$, e $m$ é a linha de $M$ correspondente a família de pontos homólogos associada a $X$. O modelo $M_1$ corresponde a uma família de câmeras $(P)_n$ obtida pela aplicação do passo 3 utilizando-se apenas os elementos de $\Gamma$. O critério de tolerância usado para definir $S_1^*$ é baseado na medida do erro de reprojeção. Mais precisamente, dado um limiar $\eta_2 \in \mathbb{R}^+$ escolhido empiricamente, inserimos em $S_1^*$ os pares $(X', m\prime)$, com $X' \in Q$, que satisfazem, $\forall j \in \{1, ..., n\}$, $\|P_j X' - m\prime_j\| \leqslant \eta_2$. O modelo $M_1^*$ corresponde a uma família de câmeras $(P^*)_n$, estimada a partir do conjunto $S_1^*$.

Dessa forma, temos que o conjunto formado pelos pontos $X'$ inseridos em $S_1^*$, e a família de câmeras $(P^*)_m$, definem uma explicação projetiva, de tolerância $\eta_2$, para uma matriz de pontos homólogos $M'$, formada por linhas de $M$.

## 8.4. Solução para o problema do passo 3

Considerando que a matriz de pontos homólogos $M$ possui $n$ colunas, temos que existem $\left(n^2 - n\right)/2$ possíveis escolhas para o par de colunas base. Sendo assim, pode-se tentar resolver o problema do passo 3 descartando-se a solução caso os parâmetros intrínsecos de alguma das câmeras encontradas seja muito diferente dos parâmetros que estamos assumindo como conhecidos. Os três passos são repetidos considerendo escolhas diferentes de colunas bases até que uma solução satisfatória seja encontrada. Mais precisamente, dado um limiar $\eta_3 \in \mathbb{R}^+$ escolhido empiricamente, recusamos a família $(P^*)_n$ caso $\|K_j - K\| \geqslant \eta_3$, para algum $j \in \{1, \ldots, n\}$, onde $K_j$ é matriz dos parâmetros

intrínsecos obtida pela fatoração de $P_j$ na forma $K_j\,[R_j|t_j]$, e $K$ é a matriz dos parâmetros intrínsecos que estamos assumindo como conhecida. Em [9] existe a explicação de como fatorar $P_j$.

## 9. Escolha das colunas base

Como temos a possibilidade de escolher $\left(n^2-n\right)/2$ pares de colunas bases para usarmos nos passos 1 e 2, faz sentido escolhermos aquele que forneça o melhor resultado. Nesse sentido, definimos que o melhor resultado é a configuração que não foi descartada por problemas de parâmetros intrínsecos no passo 3 e que explica o maior número de linhas da matriz de pontos homólogos $M$. Uma maneira bastante eficiente para determinar esse par foi obtida utilizando-se a seguinte estratégia:

1. Não se deve tentar utilizar colunas bases cuja distância média dos pontos homólogos não supere um certo limiar.
2. Se o número de pares de pontos homólogos obtido pelo RANSAC aplicado ao passo 1 for menor que o número de linhas de $M$ explicadas por uma configuração $C$, já calculada utilizando-se uma outra escolha de colunas base, deve-se abortar a execução, pois é impossível que a configuração $C$ seja melhorada. Com isso evitamos a realização do RANSAC no passo 2, que é o de maior custo computacional.
3. Devemos utilizar primeiro colunas afastadas de $M$ como colunas base, pois normalmente essas fornecem um resultado melhor que as colunas próximas. Isso faz com que os bons resultados sejam determinados antes dos ruins, e com isso aumentamos o efeito do ítem anterior.

## 10. Calibação via Levenberg-Marquadt

Seja $\left((P)_n, \{X_1, \ldots, X_m\}\right)$ uma explicação projetiva para uma matriz de pontos homólogos $M$. Podemos definir o erro de reprojeção associado a essa explicação como

$$\sum_{k=1}^{n}\sum_{i=1}^{m}\|P_k X_i - M_{ik}\|^2$$

Temos que quanto menor o erro de reprojeção melhor é a explicação. Com isso, faz sentido definirmos o problema de encontrar uma explicação projetiva ótima para uma matriz de pontos homólogos $M$. Esse problema pode ser atacado utilizando-se o algoritmo Levenberg-Marquadt, que corresponde ao processo conhecido na literatura pelo nome *Bundle Adjustment*. Uma boa referência sobre *Bundle Adjustment* pode ser encontrada em [4].

Diferente do caso geral, em que cada câmera contribui com 11 graus de liberdade para o espaço de parâmetros da função objetivo usada pelo algoritmo Levenberg-Marquadt, temos que, no nosso caso de interesse, cada câmera contribui apenas com 6 graus de liberdade, pois todas as câmeras são da forma $K\,[R|t]$, com $K$ conhecido. A translação $t$ pode ser parametrizada trivialmente. Já o problema de parametrizar $R$ é menos imediato, podendo ser resolvido pelo uso de uma representação eixo-ângulo, como descrito em [1].

## 11. Seleção de famílias de pontos homólogos

Um dos problemas existente no algoritmo de calibração em três passos é a possibilidade de alguma família de pontos homólogos ser descartada por apresentar um erro de reprojeção muito elevado em algum quadro, devido ao fato da reconstrução tridimensional realizada pelo passo 2 só levar em consideração um único par de quadros. A solução que foi adotada para esse problema foi combinar a calibração em três passos com uma calibração feita com Levenberg-Marquadt.

Inicialmente é determinada uma explicação projetiva $\left((P)_n, \Omega_1\right)$ obtida pela execução dos três passos utilizando-se um limiar $\eta_2$, definido em 8.3, relativamente alto, escolhido de maneira que uma grande quantidade de famílias de pontos homólogos seja aceita mesmo que alguns pontos com erros grosseiros possam contaminar a solução. Essa solução é então refinada por um algoritmo formado por ciclos de quatro passos que são apresentados abaixo, com o objetivo de selecionar de maneira mais criteriosa as famílias de pontos homólogos que devem ser consideradas na estimação da explicação projetiva.

1. Executam-se algumas iterações do algoritmo Levenbeg-Marquadt utilizando como estimativa inicial a explicação projetiva $\left((P)_n, \Omega_1\right)$, determinando-se uma outra explicação projetiva $\left((P')_n, \Omega_2\right)$ de menor erro de reprojeção associado.
2. Utilizam-se pares de câmeras de $(P')_n$ para determinar uma nova reconstrução $\Omega_3$ para todos os pontos homólogos de $M$. Esse processo pode ser realizado escolhendo-se pares de câmeras diferentes para reconstruir cada ponto de $\Omega_3$, de forma que, cada par utilizado seja aquele que minimiza o erro de reprojeção associado a cada ponto.
3. Descartam-se os pontos de $\Omega_3$ cujo erro de reprojeção em relação às câmeras de $(P')_n$ são maiores que um limiar $\eta_2'$, escolhido de forma mais rigorosa que que $\eta_2$, ou seja, $\eta_2' < \eta_2$. Obtém-se assim um novo conjunto de pontos $\Omega_4$.
4. Estima-se uma nova família de câmeras $(P'')_n$ a partir do conjunto de pontos $\Omega_4$ e das respectivas linhas

da matriz de pontos homólogos $M$. Com isso obtemos uma explicação projetiva $\left(\left(P''\right)_n, \Omega_4\right)$ que pode ser utilizada para alimentar um novo ciclo de refinamento.

A cada ciclo pode-se utilizar um limiar de tolerância para o erro de reprojeção cada vez menor tendo em vista que como a solução fica cada vez mais correta podemos ser cada vez mais rigorosos.

Após executarmos um determinado número de ciclos de refinamentos podemos aplicar o algoritmo Levenberg-Marquadt até sua convergência obtendo uma explicação projetiva cujo erro de reprojeção associado às famílias de pontos homólogos selecionadas é um mínimo local.

## 12. Decomposição do vídeo em fragmentos

Em um video $(I)_n$, é possível que existam quadros $I_a$ e $I_b$ que não admitam nenhum par de pontos homólogos, no caso de nenhum ponto da cena ser projetado em ambas as imagens. Além disso, algoritmos como o KLT podem não conseguir acompanhar com precisão pontos em longas seqüências de imagens. Como conseqüência, tem-se que não é possível, em geral, definir uma matriz de pontos homólogos para um vídeo completo

Usando o fato do movimento da câmera ser contínuo, pode-se realizar uma decomposição do vídeo $(I)_n$ em fragmentos, de forma que todos os fragmentos admitam uma matriz de pontos homólogos. Sendo mais preciso, estamos definindo como um fragmento, de $k+1$ quadros, de um vídeo $(I_1, ..., I_n)$, como sendo um vídeo da forma $(I_j, ..., I_{j+k})$, onde $\{j, j+1, ..., j+k\} \subset \{1, 2, ..., n\}$.

Nos experimentos realizados, os fragmentos foram determinados por uma heurística. A solução adotada foi que cada fragmento seria obtido comparando-se um quadro $I_j$ com seus sucessores até que fosse encontrado um quadro $I_{j+k}$, em que os pontos homólogos de $I_j$ e $I_{j+k}$, apresentassem uma distância média acima de um limiar $\varepsilon \in \mathbb{R}^+$, escolhido experimentalmente. Obtendo-se assim um fragmento de $k+1$ quadros $(I_j, I_{j+1}, ..., I_{j+k})$.

Para que posteriormente os fragmentos possam ser unidos, tem-se que a decomposição é feita de forma que exista a superposição de um quadro entre cada par de fragmentos adjacentes. Ou seja o vídeo $(I)_k$ é decomposto em fragmentos da forma $(I_1, ..., I_{k_1})$, $(I_{k_1}, ..., I_{k_2}), ..., \left(I_{k_{n-2}}, ..., I_{k_{n-1}}\right), \left(I_{k_{n-1}}, ..., I_{k_n}\right)$, onde cada fragmento é obtido como explicado acima.

É possível que ao tentar determinar o último fragmento, não seja possível satisfazer a restrição do limiar $\varepsilon$, devido ao encontro do final do vídeo, nesse caso descartam-se esses últimos quadros, para evitar problemas de calibração causados pela pequena modificação das coordenadas dos pontos das famílias de pontos homólogos associadas ao fragmento.

## 13. Junção de fragmentos

Consideremos que foram determinadas explicações projetivas para as matrizes de pontos homólogos dos fragmentos de um vídeo $(I)_n$. Mostraremos agora como utilizar essas explicações para determinar uma família de câmeras $(P)_n$ correspondente às câmeras que foram utilizadas para captar $(I)_n$. É preciso levar em consideração que cada explicação projetiva foi definida em um referencial próprio, e em uma escala própria. Sendo assim, vamos dividir o problema em dois

1. Alinhamento de fragmentos
2. Compatibilização de escalas

### 13.1. Alinhamento de fragmentos

Dadas duas configurações $E_1 = \left(\left(G\right)_r, \Omega\right)$ e $E_2 = \left(\left(Q\right)_s, \Psi\right)$, que explicam projetivamente as matrizes de pontos homólogos $M_1$ e $M_2$, associadas respectivamente aos fragmentos consecutivos $F_1 = (I_k, I_{k+1}, ..., I_{k+r})$, e $F_2 = (I_{k+r}, I_{k+r+1}, ..., I_{k+r+s})$ de um vídeo $(I)_n$, queremos determinar um movimento rígido que transforma $\left(Q\right)_s$ em uma família de câmeras $\left(Q'\right)_s$ tal que $G_r = Q'_1$. Diremos nesse caso que $\left(G\right)_r$ e $\left(Q'\right)_s$ estão alinhadas.

Sejam $Q_1 = K\left[R_1|t_1\right]$ e $G_r = K\left[R_2|t_2\right]$, podemos determinar a família $\left(Q'\right)_s$ aplicando a seguinte transformação aos elementos de $\left(Q\right)_s$

$$K\left[R|t\right] \mapsto K\left[\left(RR_1^T R_2\right)|RR_1^T\left(t_2 - t_1\right) + t\right].$$

Podemos usar repetidas vezes essa transformação para alinharmos todas as famílias de câmeras associadas a cada um dos fragmentos de $\left(I\right)_n$.

### 13.2. Compatibilização de escalas

O fato de duas famílias de câmeras $\left(G\right)_r$ e $\left(Q\right)_s$, associadas a fragmentos consecutivos, estarem alinhadas, não significa que elas estejam prontas para serem concatenadas de forma a gerar a família de câmeras utilizada na captação dos dois fragmentos. Isso ocorre pois geralmente $\left(G\right)_r$ e $\left(Q\right)_s$ estão calibradas em escalas diferentes.

Podemos resolver o problema de compatibilização de escalas explorando o fato que dadas duas explicações projetivas $E_1 = \left(\left(G\right)_r, \Omega\right)$ e $E_2 = \left(\left(Q\right)_s, \Psi\right)$ associadas a fragmentos consecutivos, normalmente existe um conjunto não vazio $S \subset \Omega$ cujos elementos são pontos da cena que também aparecem em $\Psi$. O fator de escala $\lambda$ pode ser obtido como resposta do seguinte problema de otimização

*Determinar* $\lambda \in \mathbb{R}^+$ *tal que aplicando-se a transformação* $K\left[R|t\right] \mapsto K\left[R|\lambda t\right]$ *sobre todas as*

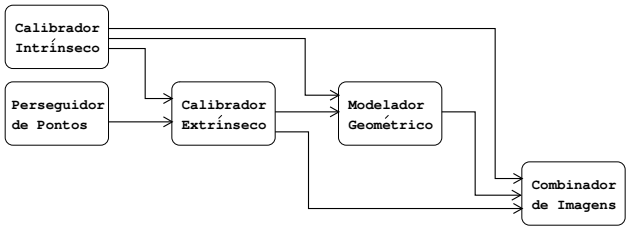


**Figura 2. Arquitetura do sistema**

*câmeras em* $(Q)_s$*, obtém-se uma nova família de câmeras que ao ser alinhada com a família* $(G)_r$ *define uma família de câmeras* $(Q')_s$ *que faz com que o erro de reprojeção associado à explicação projetiva* $((Q')_s, S)$ *seja mínimo.*

A solução desse problema não é simples, pois como as coordenadas dos elementos de $S$ são estimadas através de um processo de minimização do erro de reprojeção associado a $((G)_r, \Omega)$, é possível que algum dos pontos de $S$ apresente erros grosseiros de reprojeção quando feitas por câmeras de $(Q')_s$. Isso pode ocorrer caso grandes modificações das coordenadas de pontos de $S$, em alguma direção, não produzam alterações significativas sobre as projeções obtidas pelas câmeras de $(G)_r$. É necessário detectar e eliminar esses pontos de $S$ que podem gerar problemas ao cálculo do $\lambda$ ótimo. Isso foi resolvido estabelecendo um limiar que define o maior erro de reprojeção aceitável para que um ponto de $S$ não seja descartado.

## 14. Resultados

Foi implementado um sistema capaz de fazer realidade aumentada em um vídeo. O sistema é composto por um conjunto de módulos combinados em uma arquitetura de filtros e canais, como ilustrado na Figura 2.

O processamento realizado por cada módulo é o seguinte

1. **Calibrador Intrínseco**
   Recebe como entrada um conjunto de correspondências de pontos 3D-2D e fornece como saída uma matriz de parâmetros intrínsecos. O algoritmo utilizado para fazer isso pode ser encontrado em [9].
2. **Perseguidor de Pontos**
   Recebe como entrada um vídeo digital e fornece como saída um conjunto de famílias de pontos homólogos estimados pelo algoritmo Kanade-Lucas-Tomasi.
3. **Calibrador Extrínseco**
   Recebe como entrada uma matriz de parâmetros intrínsecos e um conjunto de famílias de pontos homólogos associados aos quadros de um vídeo, e fornece como saída os parametros extrínsecos associados a cada quadro. Esse módulo implementa o algoritmo descrito nesse artigo, utilizando para isso rotinas de álgebra linear numérica e de otimização da biblioteca *GNU Scientific Library* (GSL).
4. **Modelador Geométrico**
   Recebe como entrada um vídeo digital, os parâmetros intrínsecos da câmera que o captou, os parametros extrínsecos associados a cada quadro do vídeo, e um objeto poliedral $P$. Esse módulo apresenta uma interface gráfica que permite que um usuário modifique a posição e as dimensões de $P$ observando interativamente o efeito correspondente sobre um conjunto de quadros do vídeo. A saída do módulo é o objeto $P$ modificado.
5. **Combinador de Imagens**
   Recebe como entrada um vídeo digital, os parâmetros intrínsecos da câmera que o captou, os parametros extrínsecos associados a cada quadro do vídeo, e um objeto poliedral. A saída é o vídeo formado pela composição dos quadros do vídeo de entrada com o objeto vitual.

A Figura 3 mostra alguns quadros de um vídeo obtido como saída do sistema.

## 15. Considerações sobre desempenho

Não foi feita uma análise detalhada do desempenho do sistema que foi implementado. A grosso modo, obtivemos uma relação da ordem de dezenas de minutos para calibrar cada segundo de vídeo. Nestes testes foi utilizando um computador com processador Pentium IV de 3GHz. Tal resultado poderia ter sido melhorado se tivesse sido utilizada uma implementação de Levenber-Marquardt otimizada para o problema de calibração [4].

## 16. Conclusões

Apresentamos neste artigo um algoritmo capaz de determinar os parâmetros extrínsecos das câmeras utilizadas na captação de um vídeo. Foi descrito de forma mais detalhada que em [3] a resolução do problema de estimação de uma explicação projetiva por uma solução em três passos, sem o uso de tensores trifocais. Foram explicitados os possíveis problemas durante a execução desses três passos, tendo sido apresentadas soluções, que foram testadas no protótipo implementado.

O método ainda apresenta as seguintes deficiências, que esperamos que sejam resolvidas em trabalhos futuros:

1. Existem muitos limiares independentes que precisam ser ajustados para que o algoritmo funcione apropriadamente.

**Figura 3. Quadros selecionados de um vídeo obtido como saída do sistema**

2. Não existem garantias de que em todos os passos do algoritmo existirá um conjunto suficiente de famílias de pontos homólogos para que se possa aplicar a proposição 1 .
3. O resultado final não é uma otimização global sobre o erro de reprojeção em todos os quadros do vídeo. O que o algoritmo faz é uma otimização em cada fragmento, seguida de uma junção ótima das famílias de câmeras estimadas.
4. Um mesmo ponto tridimensional possui representações diferentes durante a execução da otimização em cada fragmento, conseqüentemente o processo de junção de fragmentos utilizado pelo algoritmo é frágil, pois ocorrem muitos erros grosseiros quando reconstruções tridimensionais de pontos de um fragmento são projetadas pelas câmeras de outro, durante a compatibilização das escalas dos fragmentos.

## Referências


[1] G. Farin and D. Hansford. *The Geometry Toolbox for Graphics and Modeling*, chapter 12, page 181. AK Peters, LTD, 1998.

[2] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM*, 24(6):381–395, 1981.

[3] S. Gibson, J. Cook, T. Howard, R. Hubbold, and D. Oram. Accurate camera calibration for off-line, video-based augmented reality. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 37, 2002.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in computer vision, second edition*. Cambrige University Press, Cambridge, United Kingdom, 2003.

[5] R. I. Hartley. In defence of the 8-point algorithm. In *ICCV*, pages 1064–1070, 1995.

[6] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[7] C. L. Sabharwal. Stereoscopic projections and 3d scene reconstruction. In *SAC '92: Proceedings of the 1992 ACM/SIGAPP symposium on Applied computing*, pages 1248–1257, New York, NY, USA, 1992. ACM Press.

[8] C. Tomasi and T. Kanade. Detection and tracking of point features. *Thecnical Report CMU-CS-91-132*, 24(6), April 1991.

[9] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, Upper Saddler River, NJ, USA, 1998.

[10] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:13–27, 1984.