

# CALIBRAÇÃO MONOCULAR COM GABARITOS 1D: UMA FERRAMENTA PRÁTICA COM O MATLAB

JOSÉ A. DE FRANÇA\*, MARCELO R. STEMMER†, MARIA B. DE M. FRANÇA\*

\**Departamento de Engenharia Elétrica - Universidade Estadual de Londrina  
Caixa Postal 6025 - 86051-990, Londrina, PR, Brasil*

†*Universidade Federal de Santa Catarina - Departamento de Automação e Sistemas  
Caixa Postal 476 - 88040-970, Florianópolis-SC, Brazil*

Emails: josealexandre@eeol.org, marcelo@das.ufsc.br, bernadete@eeol.org

**Resumo**— Recentemente, um novo método de calibração de câmeras foi proposto. Tal método usa um gabarito de uma única dimensão (um bastão com algumas esferas) para determinar os parâmetros intrínsecos da câmera. Apesar de bastante flexível, o método é muito susceptível a ruídos, inviabilizando o seu uso na prática. No presente trabalho, propõe-se uma normalização nos dados de entrada que torna o algoritmo muito mais robusto. Tal normalização não aumenta a complexidade computacional do algoritmo, mas aumenta a sua exatidão sensivelmente. Além disso, uma biblioteca de funções para o MATLAB que, dentre outras coisas, extraem automaticamente e de forma robusta os pontos do gabarito 1D, é disponibilizada livremente para tornar o método uma prática ferramenta para calibração monocular. Resultados experimentais em dados sintéticos e imagens reais mostram que a metodologia proposta obtém bons resultados quando comparada a outros métodos clássicos.

**Palavras-chave**— Calibração de câmeras, Gabarito de calibração, Objetos 1D.

**Abstract**— Recently, a new camera calibration method was proposed. That method uses a pattern of one dimension (a stick with some spheres) to determinate the intrinsic camera parameters. Although this method is very flexible, it is very susceptible to noise. In this work, we propose a normalization of the input datas that makes the algorithm much more robust. That normalization does not increase the algorithm complexity, but increases the accuracy. Besides, a MATLAB toolbox, that extracts automatically, in robust way, the points of the 1D pattern, is available. It makes the method a practical tool for monocular calibration. Practical results in synthetic datas and real images show that the method proposed gives good results when compared with others classic methods.

**Keywords**— Camera calibration, calibration apparatus, 1D objects.

## 1 Introdução

Quando a informação que se deseja extrair de uma imagem envolve algum tipo de grandeza geométrica (comprimento, ângulos, distâncias, etc), é necessário conhecer exatamente os parâmetros utilizados pela câmera para transformar um ponto no espaço 3D em um ponto no plano da imagem. Tais parâmetros podem ser conhecidos através de um processo chamado de calibração.

O processo de calibração consiste em utilizar a câmera para capturar imagens de um objeto de dimensões e forma bem conhecidos (chamado de gabarito de calibração). Em seguida, utiliza-se a relação entre pontos do gabarito e suas respectivas projeções no plano de imagem para determinar os parâmetros da câmera.

Inicialmente, os algoritmos de calibração utilizavam apenas gabaritos 3D (Tsai, 1987). Geralmente, tais gabaritos são compostos por dois planos ortogonais com um padrão bem conhecido (composto por quadrados ou círculos) em suas faces. Com o passar dos anos, foram propostos novos métodos de calibração que utilizam gabaritos 2D (Agrawal and Davis, 2003; Zhang, 2000), ou seja, um único plano com um padrão conhecido em uma das faces. Mais recentemente, alguns algoritmos de calibração propostos utilizam objetos de uma única dimensão (pontos em uma reta) para calibrar sistemas de visão que utilizem uma única câmera (Zhang, 2004) ou um conjunto binocular (de França and Stemmer, 2006). A principal vantagem

desta abordagem é que um objeto 1D pode ser “visto” por mais de uma câmera (ou conjunto binocular) ao mesmo tempo. Isto torna possível, por exemplo, calibrar de uma só vez o sistema de visão de um time de robôs que estejam posicionados em torno do gabarito.

No artigo que deu origem à teoria de calibração com gabaritos de uma dimensão, Zhang (2004) propôs um algoritmo não-linear para resolver o problema de calibração. Tal algoritmo necessita de uma estimativa inicial dos parâmetros da câmera. Esta estimativa pode ser obtida com auxílio de um método linear proposto no mesmo artigo. Infelizmente, esse método linear é bastante susceptível a ruídos normalmente presentes na projeção dos pontos do gabarito. Isto leva a uma estimativa inexata dos parâmetros da câmera, pois, se for mal inicializado, o algoritmo não-linear pode “ficar preso” em um mínimo local. Dessa forma, o método proposto por Zhang (2004) é importante para a área de calibração de câmeras apenas no aspecto teórico.

No presente trabalho, demonstra-se empiricamente que a exatidão do algoritmo de calibração linear proposto por Zhang pode ser bastante melhorada levando-se em conta algumas considerações numéricas, mais especificamente, o condicionamento do conjunto de equações lineares que deve ser resolvido. Uma simples transformação nas projeções dos pontos do gabarito antes da solução deste conjunto de equações aumenta a estabilidade do método significativamente, pois torna o problema melhor condicionado. Além disso, a complexidade computacional inserida

no algoritmo pela transformação proposta é desprezível.

Também, apresenta-se uma biblioteca de funções para o MATLAB<sup>1</sup> desenvolvida para realização dos testes com o método proposto. Em especial, a biblioteca apresentada tem a característica de não necessitar de nenhum *toolbox* especial do MATLAB e constitui uma prática ferramenta de calibração monocular.

## 2 Notação

No decorrer do texto, matrizes e vetores são representados por letras, números ou símbolos em negrito. Constantes são expressas por letras, números ou símbolos em itálico. Além disso, adotou-se a prática notação  $\mathbf{A}^{-T} = (\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1}$  para toda matriz quadrada e inversível.

Considerando o modelo de câmera *pinhole* (Faugeras and Luong, 2001), as coordenadas de um ponto 3D no sistema de coordenadas do ambiente de uma câmera são apresentadas como  $\mathbf{M} = [x, y, z]^T$  e a projeção correspondente no plano de imagem  $I$ , como  $\mathbf{m} = [u, v]^T$ . Além disso, as coordenadas homogêneas de um ponto  $\mathbf{m} = [u, v, \dots]^T$  são representadas por  $\tilde{\mathbf{m}}$ , isto é,  $\tilde{\mathbf{m}} = [u, v, \dots, 1]^T$ . De uma forma mais geral, um ponto qualquer em coordenadas homogêneas é representado por  $\tilde{\mathbf{m}} = [u, v, \dots, t]^T$ . Um índice, se houver, indica a posição do ponto em um conjunto de pontos.

Com a notação adotada, a relação entre um ponto 3D,  $\mathbf{M}$ , e sua projeção,  $\mathbf{m}$ , em uma câmera *pinhole* é dada por

$$\tilde{\mathbf{m}} \simeq \mathbf{A} [\mathbf{I}_3 \ \mathbf{0}_3] \tilde{\mathbf{M}}, \quad (1)$$

com

$$\mathbf{A} = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

onde “ $\simeq$ ” indica que os dois lados da equação podem diferir por uma constante,  $\alpha$  e  $\beta$  dão a relação entre pixels da imagem e distâncias no ambiente da câmera, respectivamente, nas direções horizontal e vertical. Já o ponto  $\mathbf{m}_0 = [u_0, v_0]^T$  são as coordenadas do ponto central da câmera. Sendo assim, há apenas quatro parâmetros intrínsecos a serem estimados durante a calibração, ou seja,  $\alpha$ ,  $\beta$ ,  $u_0$  e  $v_0$ .

## 3 Descrição do Método

O gabarito 1D utilizado na calibração deve ser algo como um bastão (ou mesmo um cordão) com diversas esferas (indexadas por  $j$ ) ao longo de sua extensão. A distância entre esferas consecutivas deve ser conhecida com exatidão. Então, no processo de calibração, o gabarito deve deslocar-se no campo visual da câmera. Enquanto isso, uma seqüência de imagens (indexada por  $i$ ) deve ser capturada. Zhang (2004) demonstrou que a calibração monocular com este tipo

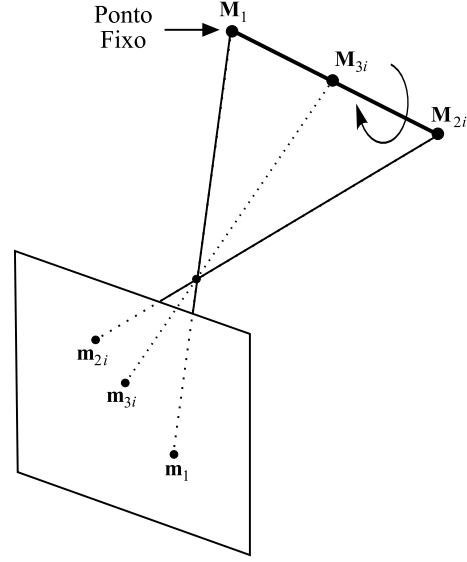


Figura 1: Esboço do gabarito 1D utilizado na calibração.

gabarito só é possível se duas condições forem satisfeitas: o gabarito conter três ou mais pontos e um desses pontos,  $\mathbf{M}_1$ , deve ser fixo, como mostra o esboço da figura 1. Dessa forma, a projeção deste ponto,  $\mathbf{m}_1$ , é a mesma em todas as imagens da seqüência.

No esquema da figura 1, o comprimento,  $L$ , do gabarito pode ser expresso em função das coordenadas dos pontos extremos  $\mathbf{M}_1$  e  $\mathbf{M}_{2i}$ , ou seja,

$$\|\mathbf{M}_{2i} - \mathbf{M}_1\| = L. \quad (3)$$

Além disso, desde que as posições relativas das esferas são conhecidas, um ponto do gabarito localizado entre  $\mathbf{M}_1$  e  $\mathbf{M}_{2i}$  pode ser expresso por

$$\mathbf{M}_{ji} = \lambda_{1j} \mathbf{M}_1 + \lambda_{2j} \mathbf{M}_{2i}, \quad (4)$$

onde  $\lambda_{1j}$  e  $\lambda_{2j}$  são conhecidos.

Da equação (1), considerando a profundidade (desconhecida) do ponto  $\mathbf{M}_{ji}$  igual a  $z_{ji}$ , tem-se

$$\mathbf{M}_1 = z_1 \mathbf{A}^{-1} \tilde{\mathbf{m}}_1, \quad (5)$$

$$\mathbf{M}_{2i} = z_{2i} \mathbf{A}^{-1} \tilde{\mathbf{m}}_{2i}, \quad (6)$$

$$\mathbf{M}_{ji} = z_{ji} \mathbf{A}^{-1} \tilde{\mathbf{m}}_{ji}. \quad (7)$$

Dessa forma, substituindo as equações anteriores em (4), obtém-se

$$z_{ji} \tilde{\mathbf{m}}_{ji} = z_1 \lambda_{1j} \tilde{\mathbf{m}}_{1i} + z_{2i} \lambda_{2j} \tilde{\mathbf{m}}_{2i} \quad (8)$$

após eliminar-se  $\mathbf{A}^{-1}$  de ambos os lados.

Agora, aplicando-se o produto cruzado em ambos os lados da equação anterior com  $\tilde{\mathbf{m}}_{ji}$ , tem-se

$$z_1 \lambda_{1j} (\tilde{\mathbf{m}}_{1i} \times \tilde{\mathbf{m}}_{ji}) + z_{2i} \lambda_{2j} (\tilde{\mathbf{m}}_{2i} \times \tilde{\mathbf{m}}_{ji}) = \mathbf{0} \quad (9)$$

que pode ser escrita como

$$z_{2i} = -z_1 \frac{\lambda_{1j} (\tilde{\mathbf{m}}_{1i} \times \tilde{\mathbf{m}}_{ji}) \cdot (\tilde{\mathbf{m}}_{2i} \times \tilde{\mathbf{m}}_{ji})}{\lambda_{2j} (\tilde{\mathbf{m}}_{2i} \times \tilde{\mathbf{m}}_{ji}) \cdot (\tilde{\mathbf{m}}_{2i} \times \tilde{\mathbf{m}}_{ji})}. \quad (10)$$

<sup>1</sup>MATLAB é marca registrada da MathWorks.

Substituindo (5) e (6) em (3) e considerando  $z_{2i}$  dado por (10), obtém-se

$$z_1 \|\mathbf{A}^{-1} \mathbf{h}_{ji}\| = L, \quad (11)$$

onde

$$\mathbf{h}_{ji} = \tilde{\mathbf{m}}_1 + \frac{\lambda_{1j}(\tilde{\mathbf{m}}_1 \times \tilde{\mathbf{m}}_{ji}) \cdot (\tilde{\mathbf{m}}_{2i} \times \tilde{\mathbf{m}}_{ji})}{\lambda_{2j}(\tilde{\mathbf{m}}_{2i} \times \tilde{\mathbf{m}}_{ji}) \cdot (\tilde{\mathbf{m}}_{2i} \times \tilde{\mathbf{m}}_{ji})} \tilde{\mathbf{m}}_{2i}. \quad (12)$$

A equação (11) é equivalente a

$$\mathbf{h}_{ji}^T \mathbf{B} \mathbf{h}_{ji} = L^2, \quad (13)$$

onde,

$$\mathbf{B} = z_1^2 \mathbf{A}^{-T} \mathbf{A}^{-1} = \begin{bmatrix} B_{11} & 0 & B_{13} \\ 0 & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}. \quad (14)$$

Levando em consideração que

$$\mathbf{h}_{ji}^T \mathbf{B} \mathbf{h}_{ji} = [a_{ji}^2, b_{ji}^2, 2a_{ji}c_{ji}, 2b_{ji}c_{ji}, c_{ji}^2]^T \mathbf{b} = \mathbf{u}_{ji} \mathbf{b}, \quad (15)$$

com  $\mathbf{h}_{ji} = [a_{ji}, b_{ji}, c_{ji}]^T$  e  $\mathbf{b} = [B_{11}, B_{22}, B_{13}, B_{23}, B_{33}]^T$ , a equação (13) pode ser reescrita como

$$\mathbf{u}_{ji}^T \mathbf{b} = L^2. \quad (16)$$

Com  $n$  imagens, tem-se  $\mathbf{U}_n = [\mathbf{u}_{j1}, \mathbf{u}_{j2}, \dots, \mathbf{u}_{jn}]^T$ . Dessa forma, considerando  $\mathbf{L}^2 = [L^2, \dots, L^2]^T$ , pode-se encontrar  $\mathbf{b}$  resolvendo-se

$$\mathbf{U}_n \mathbf{b} = \mathbf{L}^2, \quad (17)$$

ou seja,

$$\mathbf{b} = (\mathbf{U}_n^T \mathbf{U}_n)^{-1} \mathbf{U}_n^T \mathbf{L}^2. \quad (18)$$

Dado um gabarito com mais de três pontos, tem-se um número maior de equações. Contudo, para cada imagem, apenas uma equação é linearmente independente. Assim, desde que existem 5 incógnitas (os 4 parâmetros de  $\mathbf{A}$  e  $z_1$ ), são necessários, no mínimo, 5 deslocamentos do gabarito para solucionar o problema.

Uma vez que  $\mathbf{B}$  for conhecida, a matriz  $z_1 \mathbf{A}^{-1}$  pode ser obtida de  $\mathbf{B}$  através da decomposição de Cholesky (Golub and Van Loan, 1996). Por sua vez, com  $z_1$  e  $\mathbf{A}$  conhecidos, o ponto  $\mathbf{M}_1$  pode ser obtido a partir de (5) e os pontos  $\mathbf{M}_{2i}$  a partir de (10) e (6). Por último, pode-se usar (4) para obter-se  $\mathbf{M}_{ji}$ .

#### 4 Aperfeiçoamento do Método

Na visão computacional, as coordenadas das projeções são dadas em relação ao canto superior-esquerdo da imagem. Aplicando uma transformação projetiva,  $\mathbf{T}_2$ , aos pontos,  $\tilde{\mathbf{m}}$ , da imagem, tem-se  $\hat{\mathbf{m}} = \mathbf{T}_2 \tilde{\mathbf{m}}$ . Dessa forma, da equação (1), obtém-se  $\hat{\mathbf{m}} \simeq \mathbf{T}_2 \mathbf{A} [\mathbf{I} \ 0] \tilde{\mathbf{M}}$ . Esta relação implica que  $\hat{\mathbf{A}} = \mathbf{T}_2 \mathbf{A}$  representa os parâmetros intrínsecos da câmara que correspondem as projeções  $\hat{\mathbf{m}}$ .

Como discutido na seção anterior, para o caso da calibração com um gabarito 1D, tem-se a equação (11) que leva a (17). De forma semelhante, considerando os pontos  $\tilde{\mathbf{m}}$  e a matriz  $\hat{\mathbf{A}}$ , obtém-se um sistema de equações da forma  $\hat{\mathbf{U}}_n \hat{\mathbf{b}} = \mathbf{L}^2$ . Além disso, é fácil demonstrar que  $\hat{\mathbf{U}}_n = \mathbf{U}_n \mathbf{S}$ , onde  $\mathbf{S}$  é uma matriz  $4 \times 4$  que depende apenas dos elementos de  $\mathbf{T}_2$ .

Com o resultado anterior, seria de se esperar que o problema de calibração com um gabarito 1D chegasse a exatamente a mesma solução caso fossem realizados os passos a seguir.

1. Transformar os pontos fazendo  $\hat{\mathbf{m}}_{ji} = \mathbf{T}_2 \tilde{\mathbf{m}}_{ji}$ .
2. Encontrar  $\hat{\mathbf{b}}$ , resolvendo (18) e utilizando os pontos  $\hat{\mathbf{m}}_{ji}$ .
3. Fazer  $\mathbf{b} = \mathbf{S} \hat{\mathbf{b}}$ .

Contudo, isto seria verdadeiro se e somente se  $\hat{\mathbf{b}} = \mathbf{S}^{-1} \mathbf{b}$  fosse solução do problema

$$\min_{\mathbf{x}} \|(\mathbf{U}_n \mathbf{S}) \mathbf{x} - \mathbf{L}^2\|.$$

No entanto, baseado nos resultados experimentais da próxima seção, isto não é verdadeiro. Dessa forma, a solução do problema de calibração com um gabarito 1D é extremamente dependente da matriz  $\mathbf{T}_2$ . Sendo assim, é necessário encontrar a transformação  $\mathbf{T}_2$  que forneça a melhor estimativa do vetor  $\mathbf{b}$  e, portanto, da matriz  $\hat{\mathbf{A}}$ . Diversos autores já propuseram soluções para este tipo de problema (Izquierdo and Guerra, 2003; Torr, 1995). Em particular, neste trabalho utilizou-se a transformação proposta por Hartley (1997), ou seja,  $\mathbf{T}_2$  deve ser tal que o centróide do conjunto de pontos  $\mathbf{m}_{ji}$  esteja na origem e a distância média de um ponto à origem seja  $\sqrt{2}$ . Esta mudança produz melhora significativa na exatidão da matriz  $\mathbf{A}$ . Além disso, o custo computacional acrescentado ao problema devido a esta transformação é desprezível.

Uma vez que os parâmetros intrínsecos da câmara tenham sido estimados no sistema de coordenadas definido por  $\mathbf{T}_2$ , a matriz  $\mathbf{A}$  (do sistema de coordenadas original) pode ser obtida fazendo

$$\mathbf{A} = \mathbf{T}_2^{-1} \hat{\mathbf{A}}. \quad (19)$$

#### 5 Implementação do Algoritmo

A maioria das técnicas de calibração existentes na literatura utilizam um ou mais planos perpendiculares. Tais gabaritos são fáceis de ser confeccionados. Algumas vezes, basta apenas um padrão conhecido impresso em uma folha de papel. Por outro lado, um gabarito 1D composto por esferas é um pouco mais trabalhoso de ser fabricado. Contudo, tal gabarito tem características que permitem a localização automática dos seus pontos.

Neste trabalho, desenvolveu-se uma biblioteca de funções para o MATLAB que, dado uma seqüência de imagens do gabarito, efetuam a localização automática dos pontos do mesmo e realizam a calibração

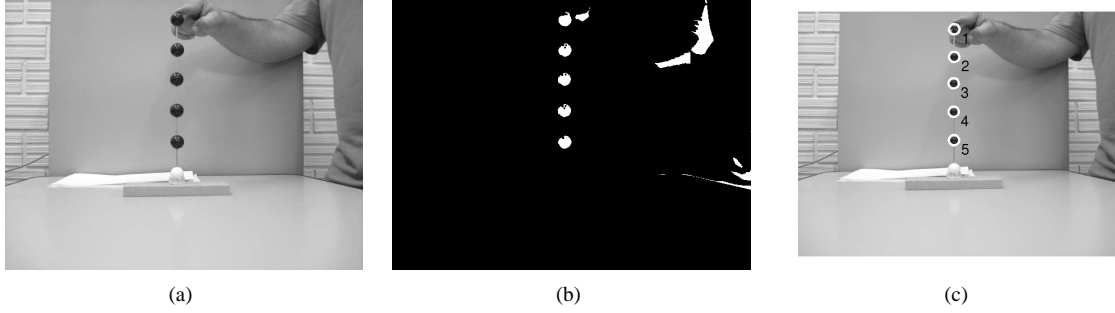


Figura 2: Sequência de imagens esboçando o processo de extração dos pontos do gabarito 1D. (a) imagem de entrada; (b) detecção dos *blobs*; (c) imagem final com pontos reconhecidos.

da câmera. Tal biblioteca está disponível livremente na internet. O leitor é convidado a acessar o website <http://research.eeol.org/publications/enia200/> para obter a biblioteca e uma descrição detalhada das funções implementadas. Contudo, nesta seção, é dado um resumo das principais operações realizadas pela biblioteca de funções.

Para facilitar a extração dos pontos do gabarito, o ideal é que a cor das esferas que compõem o mesmo sejam de uma cor que se destaque em relação ao ambiente. Neste trabalho, a cor escolhida foi preta, como apresentado na figura 2(a). Observando a figura, observa-se que a esfera fixa do gabarito está na cor branca. Isto foi feito para que o algoritmo não identifique esta esfera nesta fase, pois ela poderia ser mesclada com a base do gabarito.

Uma binarização da imagem do gabarito deve destacar bem as esferas negras, como mostra a figura 2(b). Agora, um algoritmo de *blob* gaussiano (Lindeberg, 1994, Cap. 7) pode agrupar os pixels em tons claros para segmentar as esferas do bastão<sup>2</sup>. Infelizmente, além das esferas do gabarito, outras regiões indesejadas da imagem também são segmentadas. Para contornar o problema, o algoritmo robusto baseado no RANSAC foi implementado.

Dado uma imagem com  $N$  *blobs*, com centróide em  $\mathbf{c}_j$ , escolhe-se, aleatoriamente, (Rousseeuw, 1987)

$$N_{sc} = \log(1 - p) / \log(1 - (1 - \epsilon)^2) \quad (20)$$

pares de pontos. Para cada par (indexado por  $n$ ), estima-se uma reta,  $\mathbf{l}_n$ , e calcula-se o resíduo<sup>3</sup>,  $r_{nj}^2(\mathbf{l}_n, \mathbf{c}_j)$ , de todos os  $N$  pontos. Cada resíduo é comparado a um limiar  $r_{th}^2$ . Se  $r_{nj}^2 < r_{th}^2$ , o ponto  $\mathbf{c}_j$  é considerado bom. Após  $N_{sc}$  iterações, retém-se a reta,  $\mathbf{l}_{win}$ , que ajustou-se ao maior número de bons pontos. Por último, estima-se a reta  $\mathbf{l}_i$  com apenas os bons pontos (identificados por  $r_{(win)j}^2 < r_{th}^2$ ). Tais “bons pontos” são os pontos  $\mathbf{m}_{2i}$  e  $\mathbf{m}_{ji}$  do gabarito (figura 1) e são retornados pelo código em `cb_1dExtractPts.m`, cuja saída é apresentada na figura 2(c), que implementa o algoritmo descrito nesta seção.

<sup>2</sup>Essa é a fase com maior custo computacional do processo. Por conta disso, este trecho de código foi codificado em linguagem C e é executado de dentro do ambiente do MATLAB.

<sup>3</sup>Quadrado da distancia do ponto  $\mathbf{c}_j$  a reta  $\mathbf{l}_n$ .

O método RANSAC pode obter uma boa estimação da reta formada pelo gabarito, mesmo que mais de 50% dos *blobs* segmentados não pertençam ao gabarito. Para isto, os parâmetros  $p$  (probabilidade de pelo menos uma das retas  $\mathbf{l}_n$  ser estimada com apenas pontos bons) e  $\epsilon$  (percentagem estimada de pontos ruins) devem ser bem selecionados em (20). Neste trabalho, considera-se  $p = 0,99$  e  $\epsilon = 0,7$ .

É bom observar que, quanto mais esferas negras o gabarito contiver, maior são as chances de estimar a reta  $\mathbf{l}_n$  corretamente. Além disso, se, para uma data imagem, o algoritmo retornar um número de pontos diferente do número de esfera do gabarito, tal imagem deve ser descartada.

Além dos pontos do gabarito, o código em `cb_1dExtractPts.m` retorna uma matriz da forma  $\mathbf{R} = [\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_N]$ , onde  $\mathbf{l}_i = [\mathbf{n}_i^T, q_i]^T$  é a reta na imagem que melhor ajusta-se aos pontos do gabarito. Desde que a projeção do ponto fixo,  $\mathbf{m}_1$ , do gabarito pertence a todas as retas, tem-se  $\mathbf{l}_i^T \tilde{\mathbf{m}}_1 = \mathbf{n}_i^T \mathbf{m}_1 + q_i = 0$  caso o ruído seja ignorado. Dessa forma,  $\mathbf{m}_1$  pode ser estimado minimizando  $\sum_{i=1}^N \|\mathbf{n}_i^T \mathbf{m}_1 + q_i\|^2$ , ou seja,

$$\mathbf{m}_1 = - \left( \sum_{i=1}^N \mathbf{n}_i \mathbf{n}_i^T \right)^{-1} \left( \sum_{i=1}^N q_i \mathbf{n}_i \right).$$

Uma vez que o ponto  $\mathbf{m}_1$  e os pontos  $\mathbf{m}_{2i}$  e  $\mathbf{m}_{ji}$  de todas as imagens estejam disponíveis, a calibração pode ser efetuada. Isto é feito pelo código em `cb_1dZhang.m` que normaliza os pontos como indicado na seção 4 e monta os vetores  $\mathbf{h}_{ji}$  de acordo com (15) para resolver (17) e encontrar  $\mathbf{A}$  e  $z_1$  pela decomposição de Cholesky.

Uma vez que a calibração tenha sido efetuada, o movimento do gabarito durante o processo pode ser esboçado com auxílio das equações (4), (5) e (6). Isto é feito pelo código em `cb_1dReconstruction.m` e tem uma saída como apresentado na figura 3.

## 6 Resultados Experimentais

Na avaliação do método proposto, foram utilizados dados sintéticos e dados adquiridos de imagens reais. Os resultados obtidos com cada tipo de dado são discutidos nas seções a seguir.

## 6.1 Dados Sintéticas

Os dados sintéticos foram obtidos supondo uma câmera com os seguintes parâmetros:  $\alpha = 842$ ;  $\beta = 879$ ;  $u_0 = 358$ ;  $v_0 = 207$ . A resolução da imagem é feita igual a  $640 \times 512$  pixels. Além disso, foi simulado um bastão de 30 cm de comprimento e três pontos colineares e eqüidistantes. Em tal simulação, 20 imagens sintéticas foram geradas. Entre uma imagem e outra, os ângulos horizontal e vertical variam aleatoriamente, respectivamente, dentro dos intervalos  $[-\pi/2, \pi/2]$  e  $[-\pi/2, \pi/2]$ .

Os dados obtidos na simulação foram utilizados para avaliar o desempenho do método com respeito ao nível de ruído presente nos dados. Para isto, ruído gaussiano de média zero e desvio padrão  $\sigma$  foi acrescentado aos pontos projetados nas imagens sintéticas. Tal ruído variou de 0,1 a 1,0 pixel. Para cada nível de ruído foram realizadas 250 simulações e a mediana de cada parâmetro intrínseco foi retido. Essas medianas foram comparadas com os parâmetros da câmera simulada. Em cada simulação, os parâmetros intrínsecos da câmera foi estimado pelo método se Zhang (2004) e pelo proposto no presente artigo. Os resultados da comparação são apresentados nas figuras 4 e 5. Nestas, os erros são apresentados de forma relativa (em porcentagem).

Analisando os resultados observa-se que a exatidão do método que utiliza a normalização proposta degrada-se rapidamente com o aumento do nível de ruído presente nos dados. Por outro lado, a exatidão do método de Zhang varia mais lentamente. Contudo, a exatidão do método proposto é da ordem de 10 vezes maior que a do método de Zhang. Em especial, para  $\sigma = 0,5$  (valor maior do que os encontrados na prática), tem-se um erro para  $\alpha$  e  $\beta$  em torno de 0,1 %.

## 6.2 Dados Reais

Para adquirir os dados reais, foi utilizado um gabarito com 6 esferas de plástico, apresentado na figura 2(a). A distância entre os centróides de esferas vizinhas foi medida com auxílio de um paquímetro. Ma-

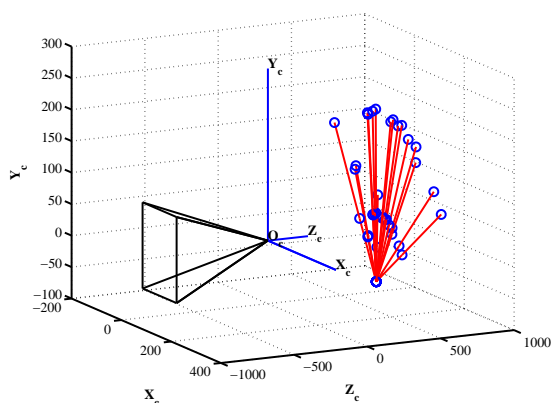


Figura 3: Reconstrução 3D do gabarito 1D realizada com a matriz  $A$  estimada.

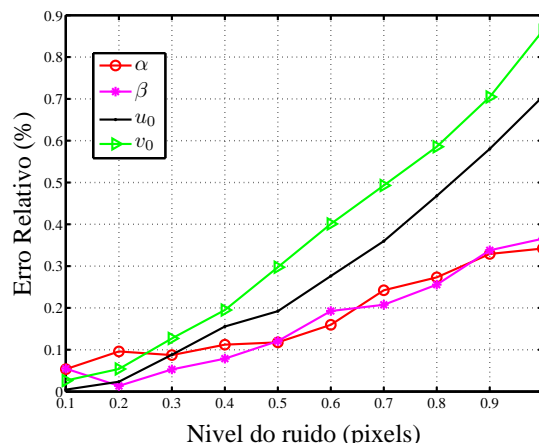


Figura 4: Erro relativo obtido na estimação dos parâmetros intrínsecos da câmera pelo método linear com dados normalizados.

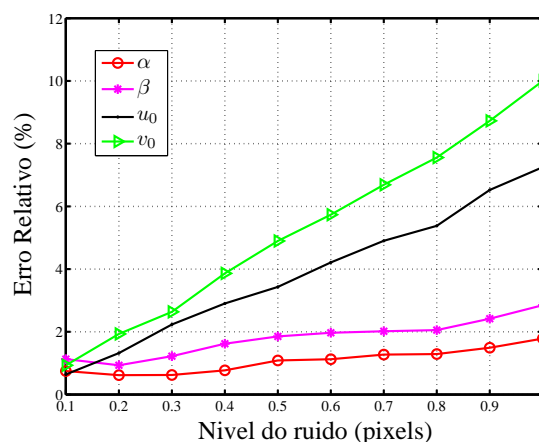


Figura 5: Erro relativo obtido na estimação dos parâmetros intrínsecos da câmera pelo método linear com dados sem normalização.

nualmente, o gabarito foi movido em diferentes posições a uma distância de aproximadamente 50 cm da câmera. Durante o movimento, 20 imagens foram adquiridas. O resultado deste experimento é apresentado tabela 1. O resultado referenciado como **NORMALIZADO** foi obtido com o método linear e a normalização proposta. O resultado **ZHANG** é a solução não-linear do problema de calibração 1D sem normalização.

A título de comparação, os parâmetros intrínsecos da câmera utilizada nos testes também foram estimados pelo método proposto por Zhang (2000). O gabarito utilizado neste método é um plano com um padrão quadriculado. Durante a calibração, foram realizados 10 deslocamentos e uma seqüência com 10 imagens foram capturadas. Em seguida, a implementação de Bouguet (2004) do algoritmo de Zhang (2000) foi utilizada para estimar os parâmetros da câmera. O resultado desta calibração está apresentado na terceira linha da tabela 1 e é referenciado por **BOUGUET**. Observe-se que, para o caso analisado, o resultado **ZHANG** está bastante diferente do obtido pelo algoritmo Bou-

Tabela 1: Parâmetros intrínsecos do conjunto binocular utilizado para gerar os dados sintéticos.

Método	Parâmetros			
	$\alpha$	$\beta$	$u_0$	$v_0$
<b>NORMALIZADO</b>	735,43	688,63	345,65	224,70
<b>ZHANG</b>	106,40	109,79	58,63	59,85
<b>BOUGUET</b>	718, 11 $\pm$ 5, 25	715, 72 $\pm$ 4, 63	324, 22 $\pm$ 2, 10	243, 45 $\pm$ 4, 57

guet (2004). Por outro lado, mesmo sendo obtido por um algoritmo linear, o resultado **NORMALIZADO** é muito próximo do **BOUGUET**, obtido por métodos não-lineares.

## 7 Conclusões e Trabalhos Futuros

Neste trabalho aperfeiçoou-se a técnica de calibração monocular com um gabarito de uma única dimensão. Tal técnica apenas requer que a câmera observe, a partir de diferentes orientações, um bastão graduado, ou seja, pontos colineares, cujas posições relativas são conhecidas. Durante a captura das imagens a câmera deve estar imóvel, pois uma das extremidades do gabarito deve permanecer em um ponto fixo durante o movimento. No entanto, tal movimento não precisa ser conhecido.

A metodologia de calibração consiste em um algoritmo linear que fornece uma estimação inicial dos parâmetros intrínsecos da câmera. Para que seja possível uma solução linear, uma substituição de variáveis é necessária. Tal substituição torna o sistema muito susceptível a ruídos. Contudo, demonstrou-se experimentalmente neste trabalho que uma simples normalização dos dados de entrada tornam o problema melhor condicionado. Tal normalização é essencial para que a técnica possa ser utilizada em situações práticas.

Com as modificações sugeridas neste trabalho, baseado nos resultados obtidos tanto em simulações quanto em imagens reais, concluí-se que um bastão com apenas três pontos colineares constitui um prático gabarito de calibração. Tal gabarito pode, inclusive, estar visível para várias câmeras ao mesmo tempo. Isto permite a calibração de vários conjuntos binoculares em bloco. Além disso, dependendo da exatidão pretendida, a solução obtida de forma linear, mas com a normalização proposta, pode dispensar a necessidade de refinamento por métodos não-lineares bem mais complexos.

O trabalho de pesquisa sobre o tema de calibração com um gabarito 1D ainda continua. Faz-se necessário provar matematicamente o efeito da normalização dos dados no resultado final e a escolha da melhor transformação  $T_2$ . Além disso, deve-se examinar a influência da variação do número de imagens e da quantidade de pontos do gabarito na solução final do problema.

## Referências

- Agrawal, M. and Davis, L. (2003). Camera calibration using spheres: A semidefinite programming approach, *IX IEEE International Conference on Computer Vision*, pp. 782–789.
- Bouguet, J. T. (2004). *Camera Calibration Toolbox for Matlab*, MRL - Intel Corp. [online] [http://www.vision.caltech.edu/bouguetj/calib\\_doc/index.html](http://www.vision.caltech.edu/bouguetj/calib_doc/index.html).
- de França, J. A. and Stemmer, M. R. (2006). Stereo calibrations with a free-moving stick, *VII Induscon - Conferência Internacional de Aplicações Industriais*, Recife, Brazil.
- Faugeras, O. and Luong, Q.-T. (2001). *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*, MIT Press, Cambridge, MA.
- Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*, The Johns University Press.
- Hartley, R. (1997). In defence of the eight point algorithm, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **19**(6): 580–593.
- Izquierdo, E. and Guerra, V. (2003). Estimating the essential matrix by efficient linear techniques, *IEEE Trans. on Circuits and Systems for Video Technology* **13**(9): 925–934.
- Lindeberg, T. (1994). *Scale-Space Theory in Computer Vision*, Kluwer Academic Publishers.
- Rousseeuw, P. J. (1987). *Robust Regression and Outlier Detection*, Wiley.
- Torr, P. H. S. (1995). *Outlier Detection and Motion Segmentation*, PhD thesis, University of Oxford, Engineering Dept, Inglaterra.
- Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf cameras and lenses, *IEEE Journal Robotics and Automation* **RA-3**(4): 323–344.
- Zhang, Z. (2000). A flexible new technique for camera calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(11): 1330–1334.
- Zhang, Z. (2004). Camera calibration with one-dimensional objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**: 892–899.