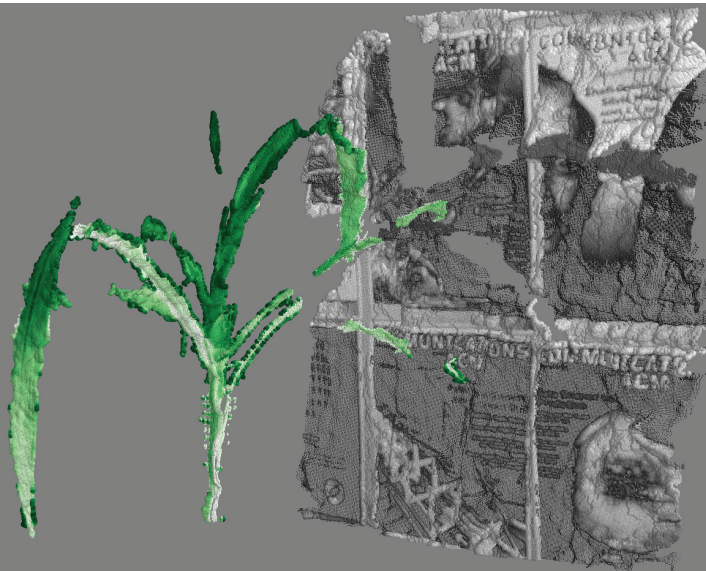


## Odometria visual com SVO e sua aplicação em reconstrução tridimensional





*Empresa Brasileira de Pesquisa Agropecuária  
Embrapa Informática Agropecuária  
Ministério da Agricultura, Pecuária e Abastecimento*

# ***Documentos 130***

## **Odometria visual com SVO e sua aplicação em reconstrução tridimensional**

*Thiago Teixeira Santos*

## **Embrapa Informática Agropecuária**

Av. André Tosello, 209 - Barão Geraldo  
Caixa Postal 6041 - 13083-886 - Campinas, SP  
Fone: (19) 3211-5700 - Fax: (19) 3211-5754  
www.embrapa.br/informatica-agropecuaria  
SAC: www.embrapa.br/fale-conosco/sac/

### **Comitê de Publicações**

Presidente: *Silvia Maria Fonseca Silveira Massruhá*

Secretária: *Carla Cristiane Osawa*

Membros: *Adhemar Zerlotini Neto, Stanley Robson de Medeiros Oliveira, Thiago Teixeira Santos, Maria Goretti Gurgel Praxedes, Adriana Farah Gonzalez, Neide Makiko Furukawa, Carla Cristiane Osawa*

Membros suplentes: *Felipe Rodrigues da Silva, José Ruy Porto de Carvalho, Eduardo Delgado Assad, Fábio César da Silva*

Supervisor editorial: *Stanley Robson de Medeiros Oliveira, Neide Makiko Furukawa*

Revisor de texto: *Adriana Farah Gonzalez*

Normalização bibliográfica: *Maria Goretti Gurgel Praxedes*

Editoração eletrônica/Arte capa: *Neide Makiko Furukawa*

Imagens capa: *Thiago Teixeira Santos*

### **1ª edição**

on-line 2014

#### **Todos os direitos reservados.**

A reprodução não autorizada desta publicação, no todo ou em parte, constitui violação dos direitos autorais (Lei nº 9.610).

#### **Dados Internacionais de Catalogação na Publicação (CIP) Embrapa Informática Agropecuária**

---

Santos, Thiago Teixeira.

Odometria visual com SVO e sua aplicação em reconstrução tridimensional / Thiago Teixeira Santos. - Campinas : Embrapa Informática Agropecuária, 2014.

28 p. : il. ; 14,8 cm x 21 cm. - (Documentos / Embrapa Informática Agropecuária, ISSN 1677-9274; 130).

1. Odometria visual. 2. Reconstrução 3D. 3. Fenotipagem de plantas. 4. Navegação robótica. I. Embrapa Informática Agropecuária. Título. II. Série.

CDD 006.6 (21.ed.)

---

© Embrapa 2014

## **Autor**

### **Thiago Teixeira Santos**

Cientista da computação, doutor em Ciências da Computação, pesquisador da Embrapa Informática Agropecuária, Campinas, SP

# Apresentação

Em visão computacional e robótica, a determinação da localização no ambiente de um equipamento móvel, a partir de imagens por ele geradas é um problema fundamental. Tal equipamento pode ser uma câmera, um smartphone ou um robô, como um Veículo Autônomo Não Tripulado (Vant) ou drone, por exemplo. Um sistema de posicionamento global, Global Positioning System (GPS) pode ser utilizado para localização em campo aberto com precisão de metros, mas outras alternativas para localização são necessárias em ambientes internos, ou quando precisão de centímetros ou milímetros é requerida.

O presente trabalho analisa um método recentemente proposto na literatura para odometria visual em tempo real, isto é, a localização espacial em tempo real obtida somente a partir de imagens geradas por uma câmera acoplada ao equipamento, e sua aplicabilidade em problemas de reconstrução tridimensional de objetos. A reconstrução tridimensional tem sido utilizada ativamente nos últimos anos na determinação da arquitetura de plantas em plataformas de fenotipagem de alto desempenho. Odometria visual também pode ser aplicada no uso de Vants na agricultura, tema que tem recebido considerável atenção recentemente.

***Kleber Xavier Sampaio de Souza***

Chefe-geral

Embrapa Informática Agropecuária

# Sumário

<b>Introdução</b> .....	9
<b>Materiais e métodos</b> .....	10
Avaliação da odometria produzida pela SVO .....	13
Computação das matrizes de projeção .....	13
Restrição epipolar .....	14
<b>Resultados e discussão</b> .....	15
Estimação Bayesiana de profundidade .....	18
Inicialização da estimativa com uma segunda imagem .....	20
Atualização Bayesiana .....	22
<b>Conclusões</b> .....	25
<b>Referências</b> .....	26



# Odometria visual com SVO e sua aplicação em reconstrução tridimensional

---

*Thiago Teixeira Santos*

## Introdução

Odometria visual é o processo de estimação da localização de uma câmera somente a partir das imagens por ela produzidas. Trata-se de um componente importante em navegação robótica<sup>1</sup> (THRUN et al., 2005) e reconstrução tridimensional de objetos (FURUKAWA; PONCE, 2010; SNAVELY et al., 2008). Em pesquisa agrícola, odometria visual pode ser utilizada na reconstrução tridimensional aplicada à fenotipagem de plantas (JAY et al., 2014; LOU et al., 2014; PAPROKI et al., 2012; SANTOS; OLIVEIRA, 2012; SANTOS et al., 2015<sup>2</sup>) e na navegação de *drones* e *rovers* agrícolas (GRENZDÖRFFER et al., 2008; ROVIRA-MÁS et al., 2005; SLAUGHTER et al., 2008; SOUSA et al., 2003). Em locais em que navegação GPS não pode ser utilizada (por exemplo, ambientes internos como laboratórios e casas de vegetação) ou em atividades nas quais a precisão do GPS é insuficiente (em reconstrução 3D), a odometria visual surge, junto

---

<sup>1</sup> Por exemplo, a odometria visual é utilizada pela National Aeronautics and Space Administration (Nasa) no robô Curiosity em suas atividades de exploração de Marte.

<sup>2</sup> Santos, T. T.; Koenigkan, L. V.; Barbedo, J. G. A.; Rodrigues, G.C. 3D Plant modeling: localization, mapping and segmentation for plant phenotyping using a single hand-held camera. First Workshop on Computer Vision Problems in Plant Phenotyping, 2014, artigo a ser publicado na LNCS pela Springer em 2015.



com sensoriamento a laser, como uma das alternativas mais promissoras. Mas diferentemente do sensoriamento a laser, odometria visual pode ser realizada com câmeras de baixo custo.

Recentemente, Forster et al. (2014) apresentaram um método para odometria visual com uma única câmera baseado em métodos diretos de estimação de movimento (BAKER; MATTHEWS, 2004; IRANI; ANANDAN, 2000). Batizado de Semi-direct Visual Odometry (SVO), o método foi capaz de operar a 55 quadros por segundo em um pequeno computador embarcado em drones<sup>3</sup>. Isso significa que o sistema foi capaz de prover odometria em tempo real (acima de 30 quadros por segundo), computada de forma autônoma no próprio drone.

O presente trabalho apresenta os resultados obtidos em testes realizados na Embrapa Informática Agropecuária com o sistema SVO, nos quais avaliou-se a precisão da odometria obtida para reconstrução 3D. O trabalho também discute um arcabouço Bayesiano pelo qual a localização de pontos 3D na superfície dos objetos de interesse pode ser iterativamente atualizada a partir de novas imagens obtidas pela câmera.

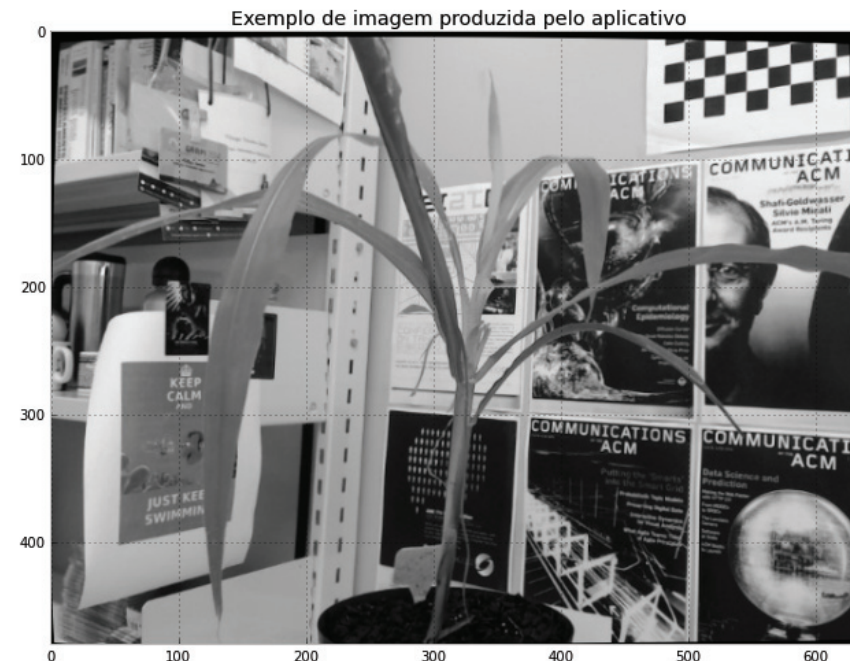
## Materiais e métodos

Foi utilizado a biblioteca SVO, versão 3ba099<sup>4</sup>, compilada localmente em um sistema Ubuntu 14.04. A câmera utilizada foi um dispositivo Logitech HD Pro C920, que produziu imagens de 640 x 480 pixels. Uma pequena área de teste foi revestida com pôsteres, que forneciam texturas necessárias para que SVO fosse capaz de estimar a movimentação da câmera (Figura 1). Um software utilitário foi desenvolvido para:

- 1) Realizar a captura das imagens a partir da câmera.
- 2) Estimar a posição da câmera a partir de chamadas à API da SVO.
- 3) Exibir em tempo real a localização da câmera em um gráfico.

<sup>3</sup> Segundo Forster et al. (2014), o processador utilizado foi um ARM Cortex A-9 com 4 núcleos a 1.6 GHz.

<sup>4</sup> GitHub. Disponível em: <[https://github.com/uzh-rpg/rpg\\_svo](https://github.com/uzh-rpg/rpg_svo)>.



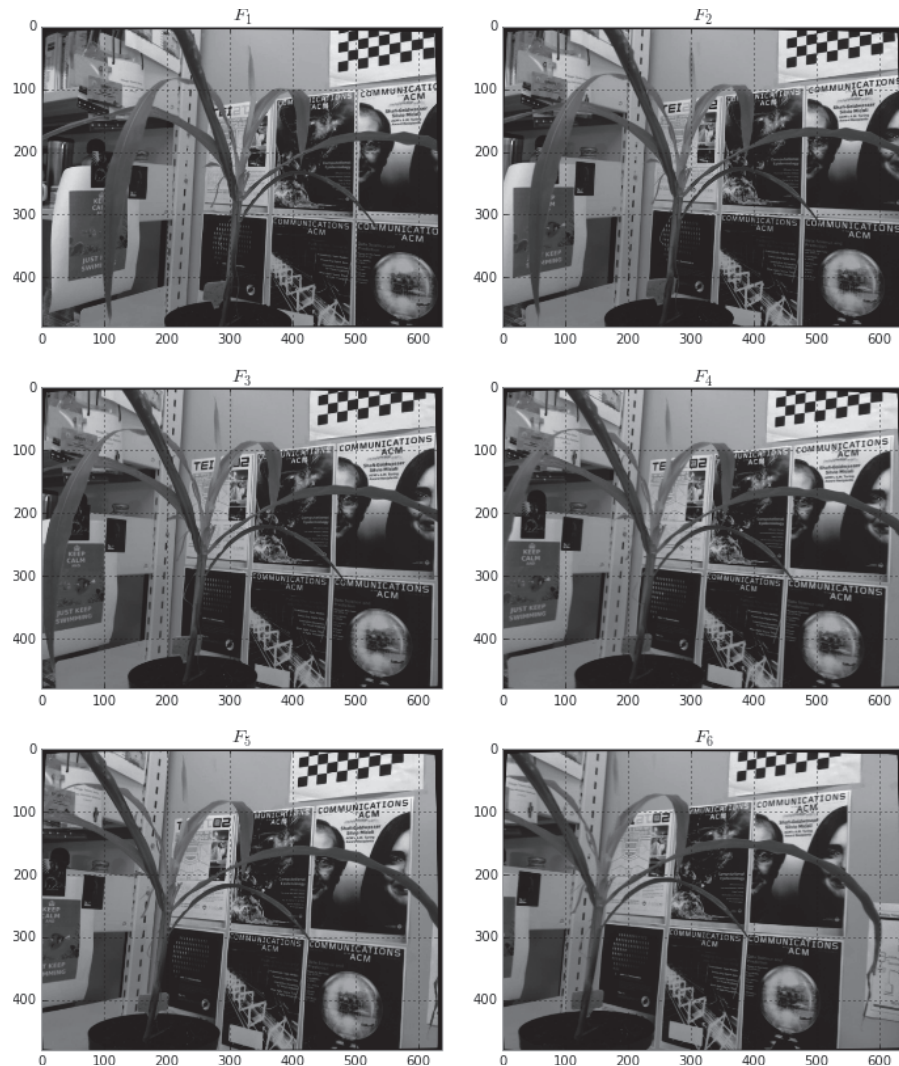
**Figura 1.** Imagem capturada pelo utilitário. Pôsteres foram utilizados para garantir a textura necessária ao funcionamento da SVO.

- 4) Armazenar, em disco, a imagem e a posição da câmera em um dado instante determinado pelo usuário.

SVO estima os parâmetros extrínsecos da câmera, isto é, sua rotação e translação no espaço. Já parâmetros intrínsecos da câmera foram obtidos previamente utilizando o método de calibração proposto por Zhang (2000) e implementado na biblioteca OpenCV<sup>5</sup>. Os parâmetros intrínsecos são representados por  $K$ , uma matriz 3 x 3.

O experimento descrito a seguir utilizou seis imagens produzidas pelo utilitário descrito acima (Figura 2), que armazenou em arquivo tanto as imagens quanto os parâmetros extrínsecos da câmera no momento da captura, computados pela SVO.

<sup>5</sup> OpenCV. Disponível em: <<http://opencv.org>>.



**Figura 2.** Seis imagens capturadas pela câmera. A localização da câmera no momento da captura foi gravada, como computada pela SVO.

## Avaliação da odometria produzida pela SVO

Tendo em vista aplicações em metrologia e fenotipagem de plantas, estamos interessados na capacidade da SVO em produzir odometria acurada para reconstrução tridimensional. A reconstrução tridimensional a partir de múltiplas imagens requer:

- 1) As matrizes de projeção  $P_k$  para cada posição  $k$  da câmera.
- 2) Pares de pontos correspondentes  $\mathbf{x}_i \leftrightarrow \mathbf{x}_j$ , em que  $\mathbf{x}_k$  é um ponto na imagem produzida pela câmera na posição  $k$  (HARTLEY; ZISSERMAN, 2004). Devemos então avaliar a odometria produzida quanto a sua capacidade de produzir matrizes  $P_k$  precisas.

### Computação das matrizes de projeção

SVO produz odometria na forma de matrizes de rotação  $R_k$  e vetores de translação  $\mathbf{t}_k$  para cada posição  $k$  da câmera. A matriz intrínseca  $K$  é a mesma para a câmera em todas as posições<sup>6</sup>. As matrizes de projeção são então obtidas por

$$P_k = K[R_k | \mathbf{t}_k] \quad (1)$$

em que  $[R_k | \mathbf{t}_k]$  é uma matriz  $3 \times 4$  na qual as 3 primeiras colunas são definidas pelas colunas de  $R_k$  e a última coluna definida pelo vetor  $\mathbf{t}_k$ . Seja  $\mathbf{C}_k$  um ponto tridimensional que representa o centro de projeção da câmera em cada posição. Tais pontos podem ser determinados a partir da equação

$$P_k \mathbf{C}_k = \mathbf{0}. \quad (2)$$

A Figura 3 exhibe a posição do centro de projeção da câmera no espaço no instante de aquisição de cada uma das seis imagens exibidas na Figura 2, como computado a partir dos valores de  $R_k$  e  $\mathbf{t}_k$ , produzidos pela SVO, e das equações 1 e 2.

<sup>6</sup> Isto significa que a câmera tem **foco fixo**. Nos experimentos, o recurso de autofocus da câmera C920 foi desligado antes da calibração e da aquisição das imagens.

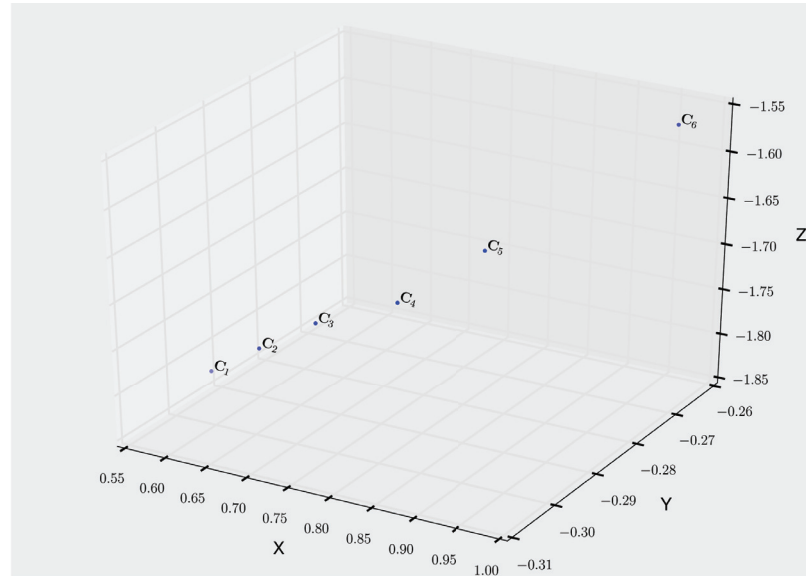


Figura 3. Posição do centro de projeção da câmera para cada uma das seis imagens.

### Restrição epipolar

Considere um par de imagens  $F_i$  e  $F_j$ , capturadas nas posições  $i$  e  $j$  e um ponto qualquer  $\mathbf{X} = (X, Y, Z, 1)^T$  no espaço 3D, observado nas duas imagens. O ponto bidimensional  $\mathbf{x}_k = (x, y, 1)^T$ , a projeção de  $\mathbf{X}$  no plano de imagem da câmera na posição  $k$ , é definido por

$$\mathbf{x}_k = \mathbb{P}_k \mathbf{X} \quad (3)$$

No problema de reconstrução 3D a partir de duas imagens,  $\mathbf{X}$  é desconhecido, mas pode ser facilmente estimado a partir das matrizes  $\mathbb{P}_i$  e  $\mathbb{P}_j$  e do par de pontos correspondentes  $\mathbf{x}_i$  e  $\mathbf{x}_j$ . Dado um ponto  $\mathbf{x}_i$  na imagem  $F_i$ , o problema passa a ser determinar a localização de seu correspondente  $\mathbf{x}_j$  na imagem  $F_j$ , o que permitirá determinar  $\mathbf{X}$  em seguida.

Determinar  $\mathbf{x}_j$  é facilitado pela **restrição epipolar** (HARTLEY; ZISSERMAN, 2004). A posição de  $\mathbf{x}_j$  está restrita à linha definida pelo produto  $\mathbb{F}\mathbf{x}_i$ , em que  $\mathbb{F}$ , a **matriz fundamental**, é obtida por

$$\mathbb{F} = [\mathbf{e}_j]_x \mathbb{P}_j \mathbb{P}_i^+ \quad (4)$$

$\mathbb{P}_i^+$  é a pseudoinversa de  $\mathbb{P}_i$  e o ponto  $\mathbf{e}_j$ , chamado de **epipolo**, é a projeção de  $\mathbf{C}_j$  na imagem  $F_i$ :

$$\mathbf{e}_j = (a, b, c)^T = \mathbb{P}_i \mathbf{C}_j \quad (5)$$

Já  $[\mathbf{e}_j]_x$  é uma matriz 3 x 3 definida por

$$[\mathbf{e}_j]_x = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix} \quad (6)$$

Temos assim uma maneira simples de avaliar a precisão da odometria produzida pelo método SVO. A odometria nos fornece os parâmetros para obtermos as matrizes de projeção  $\mathbb{P}_i$  e  $\mathbb{P}_j$ . A partir de um conjunto  $\{\mathbf{x}_i^n \leftrightarrow \mathbf{x}_j^n\}_{n=1..N}$  de referência, podemos avaliar se  $\mathbb{P}_i$  e  $\mathbb{P}_j$  são precisas analisando a distância a que cada  $\mathbf{x}_j^n$  se encontra da linha epipolar  $\mathbb{F}\mathbf{x}_i^n$ .

## Resultados e discussão

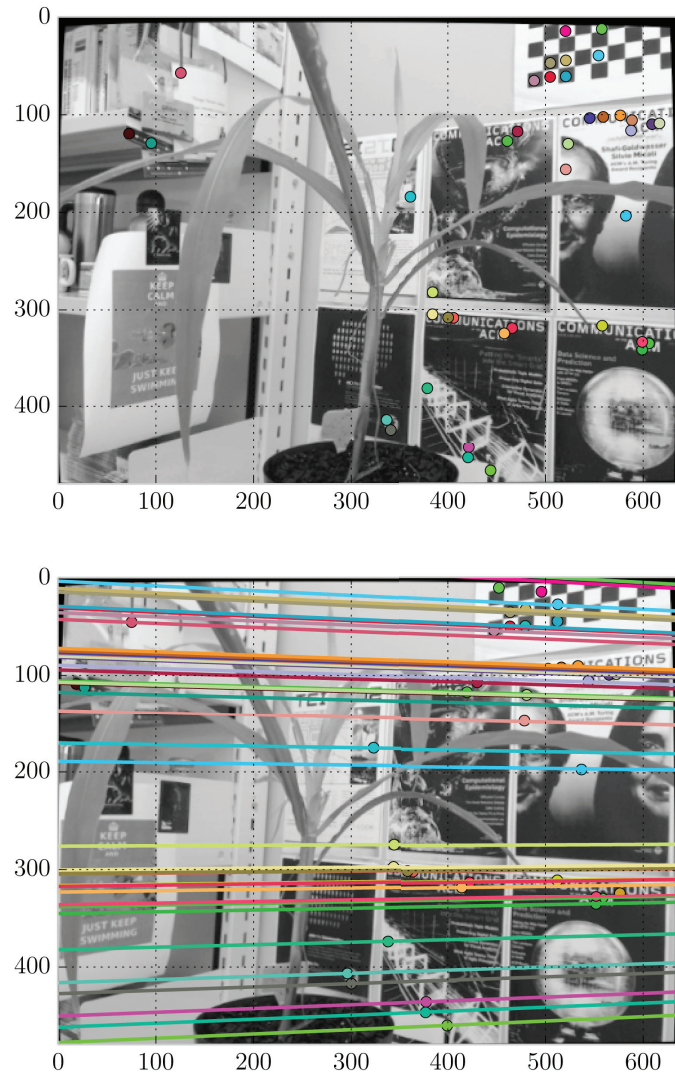
O conjunto de referência  $\{\mathbf{x}_i^n \leftrightarrow \mathbf{x}_j^n\}_{n=1..N}$  foi produzido pelo método SIFT, como proposto por Lowe (2004). SIFT possui uma fase de detecção, na qual pontos salientes são encontrados na imagem em diversas escalas e sua vizinhança caracterizada através de descritores. Segue-se então uma fase de casamento (*matching*) na qual os descritores de pontos em duas imagens diferentes são comparados<sup>7</sup>.

A Figura 4 exibe alguns pares de pontos correspondentes  $\mathbf{x}_i^n$  e  $\mathbf{x}_j^n$  e a linha epipolar  $\mathbb{L}_i^n = \mathbb{F}\mathbf{x}_i^n$ . Se a odometria for precisa, espera-se que cada  $\mathbb{L}_i^n$  intercepte o ponto  $\mathbf{x}_j^n$ .

Seja  $\mathbb{F}_{ij}$  a matriz fundamental determinada para um par de imagens  $ij$ . Para cada  $n$ , computamos as distâncias entre  $\mathbb{L}_i^n$  e  $\mathbf{x}_j^n$  e, em seguida, analisamos a mediana e o máximo encontrado para os  $N$  pares de pontos. Os dados podem ser observados na Tabela 1.

<sup>7</sup> O leitor interessado em mais detalhes deve referir a Lowe (2004) e Mikolajczyk et al. (2005).





**Figura 4.** Pontos correspondentes obtidos por SIFT e suas linhas epipolares, computadas a partir da matriz fundamental produzida com a odometria fornecida pela SVO.

**Tabela 1.** Dados sobre as distâncias computadas entre  $N$  pontos  $x_j^n$  e suas linhas epipolares correspondentes  $l_j^n = Fx_i^n$ .

	mediana	máximo	$N$
$F_{0,1}$	0.3295	209.3421	84
$F_{0,2}$	0.1414	17.9818	55
$F_{0,3}$	0.3976	13.5312	54
$F_{0,4}$	0.6139	8.4116	38
$F_{0,5}$	0.2988	198.6811	36
$F_{0,6}$	0.1865	19.9839	24
$F_{1,2}$	0.1124	10.9598	62
$F_{1,3}$	0.1459	11.1963	60
$F_{1,4}$	0.3268	7.2656	45
$F_{1,5}$	0.1452	15.1496	53
$F_{1,6}$	0.2072	21.9571	31
$F_{2,3}$	0.1677	8.2321	72
$F_{2,4}$	0.3530	19.0782	65
$F_{2,5}$	0.1631	291.1054	70
$F_{2,6}$	0.1804	26.3705	37
$F_{3,4}$	0.2036	196.8797	72
$F_{3,5}$	0.1258	160.6963	64
$F_{3,6}$	0.2048	197.0131	51
$F_{4,5}$	0.2458	20.1242	74
$F_{4,6}$	0.1394	193.6646	61
$F_{5,6}$	0.2713	132.0985	40

Os resultados mostram que a mediana é inferior a 1 pixel de distância, uma evidência que a odometria produzida é bastante precisa. Porém, os valores máximo mostram que foram encontradas distâncias grandes entre alguns pontos e suas linhas epipolares correspondentes. Examinando-se o boxplot exibido na Figura 5, é possível ver a distribuição dos valores das distâncias. Nota-se que os valores das distâncias se concentram em torno de zero e os maiores valores são *outliers*, algo esperado uma vez que a metodologia SIFT pode produzir correspondências espúrias.

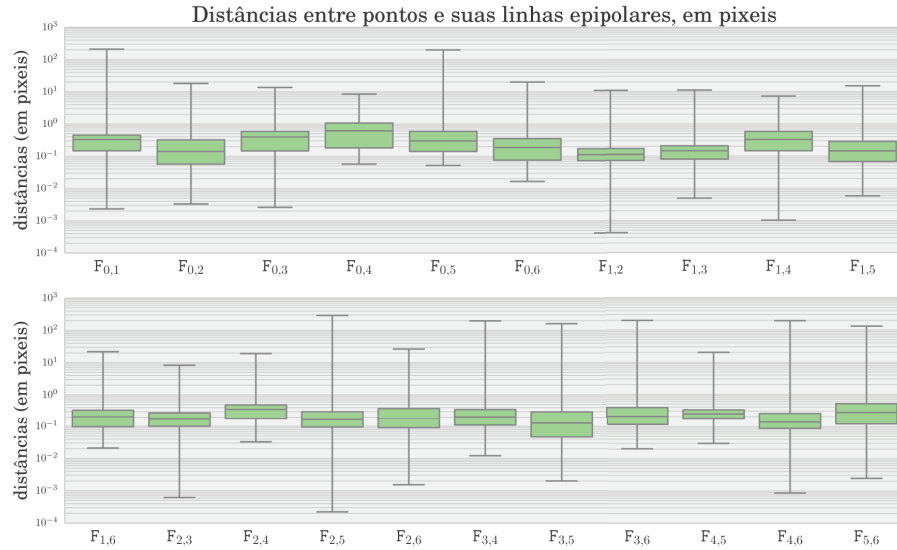


Figura 5. Boxplot das distâncias entre pontos e suas linhas epipolares correspondentes.

## Estimação Bayesiana de profundidade

Considere um ponto  $\mathbf{x}_i$  na imagem  $i$ . Tendo em mãos  $\mathbf{P}_i$  e  $\mathbf{C}_i$  (obtidos por meio da odometria fornecida pela SVO, como discutido anteriormente), a localização de um ponto  $\mathbf{X}_d$  que é projetado sobre  $\mathbf{x}_i$  e encontra-se a uma distância  $d$  do centro de projeção  $\mathbf{C}_i$  é dada por

$$\mathbf{X}_d = \mathbf{C}_i + d \cdot \vec{r}, \quad (7)$$

em que  $\vec{r}$  é um vetor unitário definido como

$$\vec{r} = \frac{(\mathbf{P}^+ \mathbf{x}_i - \mathbf{C}_i)}{\|(\mathbf{P}^+ \mathbf{x}_i - \mathbf{C}_i)\|} \quad (8)$$

Desta forma, a localização do ponto em 3D é parametrizada pela distância  $d$  e o processo de reconstrução tridimensional pode se basear na estimação de um valor adequado para  $d$ . Na Figura 6 (a), vemos um ponto  $\mathbf{x}$  na imagem  $F_1$  e, na Figura 6 (b), um gráfico 3D exibindo o raio definido pelo ponto  $\mathbf{x}$  e o centro de projeção da câmera,  $\mathbf{C}_1$ . O ponto tridimensional  $\mathbf{X}_d$  que foi projetado em  $\mathbf{x}$  deve se encontrar nesse raio, entre dois limites

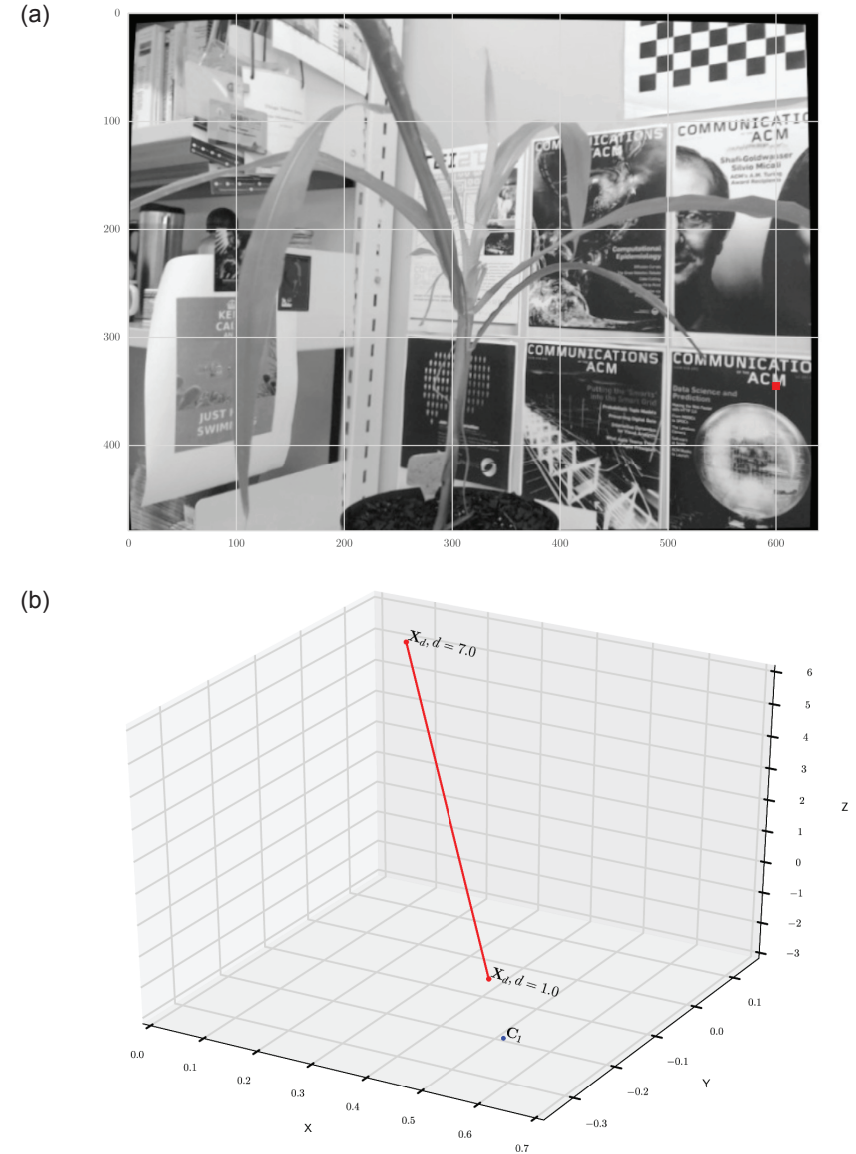


Figura 6. (a) Um ponto  $\mathbf{x}_i$  em  $F_1$  é exibido (em vermelho). (b) O raio definido por  $\mathbf{x}_i$  e pelo centro de projeção de  $F_1$  determina a posição do ponto  $\mathbf{X}$ , que deve estar entre as distâncias mínima e máxima definidas.

previamente definidos  $d_{\min}$  e  $d_{\max}$  (configurados em 1 e 7 respectivamente - assume-se aqui que não há objetos de interesse fora desse intervalo na cena).

### Inicialização da estimativa com uma segunda imagem

Dadas uma nova imagem  $F_j$  e sua matriz de projeção  $P_j$ , o ponto  $\mathbf{x}_j$  que corresponde a  $\mathbf{x}_i$  deve se encontrar na linha  $l_j = F_{ij}\mathbf{x}_i$ . Percorrendo os pontos na linha visíveis na imagem, podemos selecionar o melhor ponto possível, de acordo com algum critério de similaridade. Assumindo que não tenha havido uma mudança drástica de pose entre as imagens  $i$  e  $j$ , a correlação cruzada normalizada (*normalized cross correlation* - NCC) pode ser utilizada para estabelecer correspondências entre pontos. A NCC é definida por

$$\text{ncc}(I_i, I_j) = \frac{1}{m-1} \sum_{\mathbf{x}} \frac{I_i(\mathbf{x}) - \mu_i}{\sigma_i} \cdot \frac{I_j(\mathbf{x}) - \mu_j}{\sigma_j} \quad (9)$$

em que  $I_i$  e  $I_j$  são retalhos (*patches*), pixels vizinhos aos pontos  $\mathbf{x}_i$  e  $\mathbf{x}_j$  nas imagens  $F_i$  e  $F_j$  respectivamente, enquanto que  $\mu_k$  e  $\sigma_k$  são média e o desvio padrão computados no *patch*  $I_k$ . O código Python abaixo é uma implementação da NCC, utilizando *patches* quadrados com 11 x 11 pixels:

```
def get_patch(I, x, wid=5):
    if x[0] + wid + 1 >= width or x[0] - wid < 0:
        return zeros((2*wid+1, 2*wid+1)).flatten()

    if x[1] + wid + 1 >= height or x[1] - wid < 0:
        return zeros((2*wid+1, 2*wid+1)).flatten()

    patch = I[x[1]-wid:x[1]+wid+1, x[0]-wid:x[0]+wid+1].flatten()
    return patch

def ncc(patch_i, patch_j, threshold=0.5):
    n = len(patch_i)
    if n == 0:
        return 0.

    d_i = (patch_i - mean(patch_i)) / std(patch_i)
    d_j = (patch_j - mean(patch_j)) / std(patch_j)
```

```
ncc_value = sum(d_i * d_j) / (n-1)
if ncc_value > threshold:
    return ncc_value
return 0.0
```

A Figura 7 mostra um ponto  $\mathbf{x}_i$  na imagem  $F_1$ . Na imagem  $F_2$ , a linha epipolar é exibida em azul e, em vermelho, os valores computados para o NCC para os pontos ao longo da linha. O valor com maior valor de NCC é selecionado como  $\mathbf{x}_j$ .

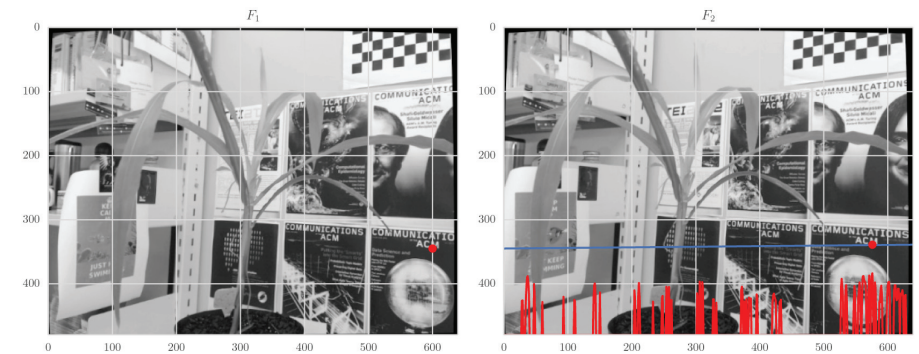


Figura 7. O valor do NCC (em vermelho) é computado para cada ponto na linha epipolar (em azul). O ponto correspondente é declarado na posição onde o NCC atinge o valor máximo.

O ponto  $\mathbf{X}$  pode ser estimado a partir do par de pontos utilizando **triangularização**, por meio de um procedimento de otimização por mínimos quadrados que encontra o ponto  $\mathbf{x}$  mais próximo do par de raios definidos por  $\mathbf{x}_i$  e  $\mathbf{C}_i$ ,  $\mathbf{x}_j$  e  $\mathbf{C}_j$ . O código Python abaixo implementa o procedimento, solucionando a minimização através de uma decomposição por valores singulares, consulte Hartley e Zisserman (2004) para detalhes:

```
def dlt_triangulation(ui, Pi, uj, Pj):
    """Implementacao do metodo apresentado por Hartley & Zisserman,
    12.2"""
    ui /= ui[2]
    xi, yi = ui[0], ui[1]
    uj /= uj[2]
    xj, yj = uj[0], uj[1]
    a0 = xi * Pi[2,:] - Pi[0,:]
    a1 = yi * Pi[2,:] - Pi[1,:]
    a2 = xj * Pj[2,:] - Pj[0,:]
```

```

a3 = yj * Pj[2,:] - Pj[1,:]
A = vstack((a0, a1, a2, a3))
U, s, VT = linalg.svd(A)
V = VT.T
X3d = V[:, -1]
return X3d/X3d[3]

```

A distância  $d$  é facilmente obtida por

$$d = \|\mathbf{X} - \mathbf{C}_i\| \quad (10)$$

A Figura 8 mostra um exemplo do resultado obtido considerando-se as imagens  $F_i$  e  $F_j$  para o ponto  $\mathbf{x}_i = (600, 345)^T$ . O valor encontrado para  $d$  foi 2.84.

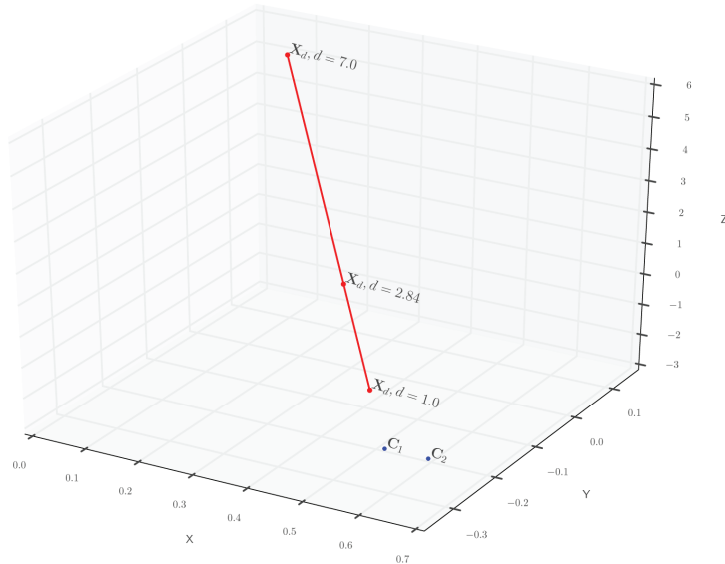


Figura 8. Estimativa  $\mathbf{X}_d$ ,  $d = 2.84$  para o pixel  $\mathbf{x}_i = (600, 345)^T$ , a partir das imagens  $F_i$  e  $F_j$ .

### Atualização Bayesiana

Devido ao ruído nas imagens, às limitações de resolução e às falhas do processo de correspondência por NCC, a distância  $d = 2.84$  não pode ser tomada como um valor absoluto, mas sim uma **estimativa** para a posição do ponto  $\mathbf{X}$ . Convém modelar probabilisticamente essa estimativa, obten-

do-se assim uma distribuição inicial (*prior*) para  $d$ . Aqui, vamos utilizar, inicialmente, distribuição normal com média em 2.84 e desvio padrão de  $\sigma = 0.1$ , obtendo-se o *prior* exibido na Figura 9. Tal escolha do desvio padrão é arbitrária, ela apenas informa ao modelo probabilístico nossa confiança no valor inicial. Conforme novas observações, são feitas a partir de novas imagens,  $\sigma$  será ajustado para valores mais precisos.

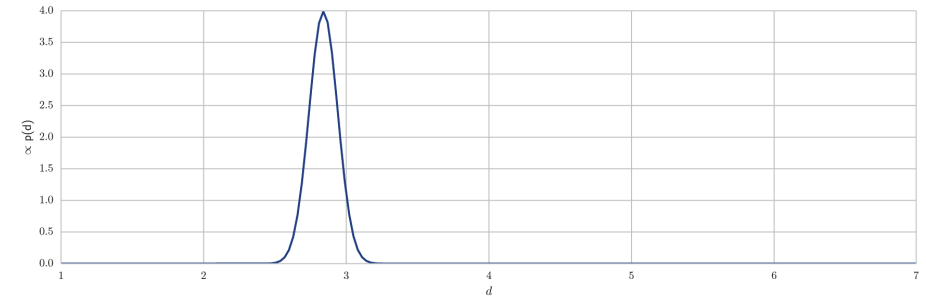


Figura 9. Distribuição (*prior*) para a distância  $d$  correspondente ao ponto  $\mathbf{x}_i = (600, 345)^T$ .

Com a chegada de uma nova imagem e sua matriz de projeção, podemos repetir o procedimento de triangularização, encontrando um novo valor para  $d$ . Considerando que  $\tilde{\mathbf{x}}_j$  seja o novo ponto correspondente encontrado para  $\mathbf{x}_i$  via NCC, a atualização obedece a regra de Bayes (GELMAN et al., 2013):

$$p(d | \tilde{\mathbf{x}}_j) \propto p(\tilde{\mathbf{x}}_j | d) p(d) \quad (11)$$

A verossimilhança  $p(\tilde{\mathbf{x}}_j | d)$  é modelada por uma distribuição normal  $N(0, \sigma)$  da distância entre  $\tilde{\mathbf{X}}$ , obtido pela triangularização entre  $\mathbf{x}_j$  e  $\tilde{\mathbf{x}}_j$ , e  $\mathbf{X}$ . Na Figura 10, podemos confrontar os valores obtidos para  $d$  e a diferença produzida na localização de  $\mathbf{X}$ . A atualização Bayesiana é ilustrada na Figura 11 enquanto que a localização de  $\mathbf{X}$  atualizada, tomando-se o valor *máximo a posteriori* para  $d = 2.69$ , é exibida na Figura 12.

Conforme novas observações chegam, o procedimento de atualização se repete e a estimativa de  $\mathbf{X}$  se torna mais acurada.



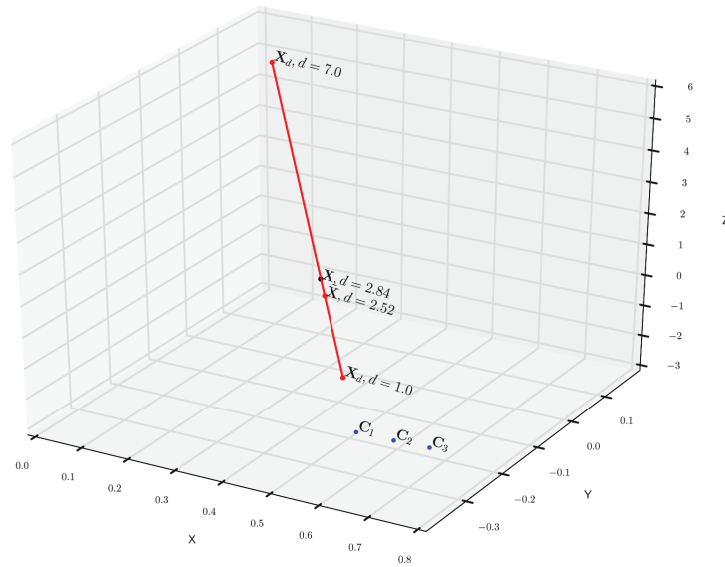


Figura 10. Incerteza na posição de  $X$ : o valor anterior (*prior*) para a distância era  $d = 2.84$ , mas a nova observação,  $\tilde{X}$ , apresenta um valor de  $\tilde{d} = 2.52$ .

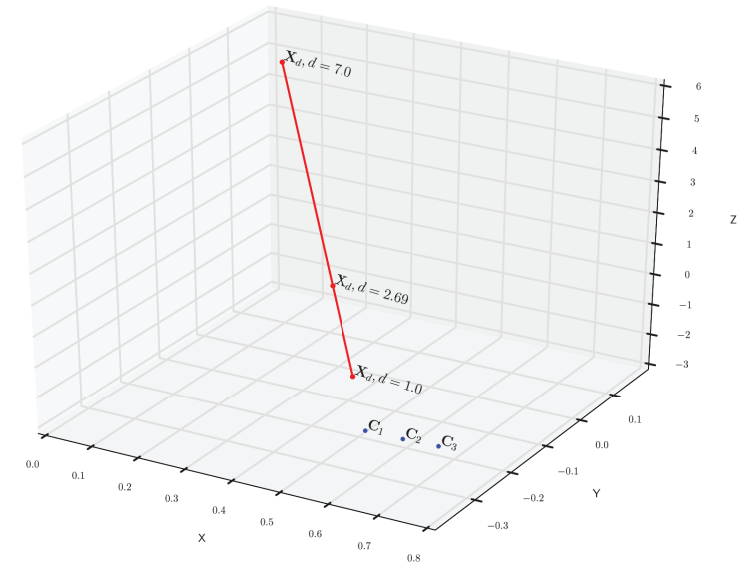


Figura 12. *Posterior*: posição atualizada para  $d = 2.69$ , incorporando assim a informação do *prior* ( $d = 2.84$ ) e da nova observação ( $d = 2.52$ , ver Figura 10).

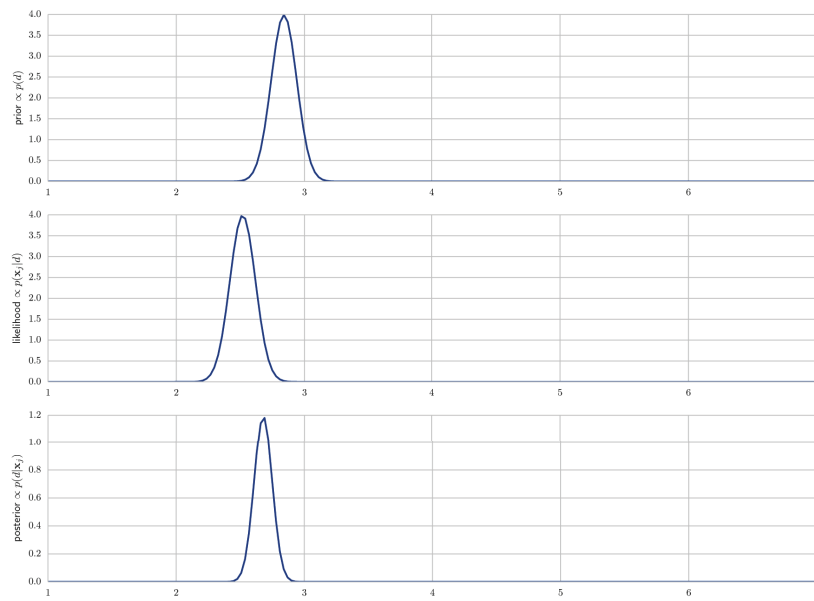


Figura 11. Acima: *prior*  $p(d)$ . Centro: verossimilhança  $p(\tilde{X}_i|d)$ . Abaixo: atualização produzida pelo produto entre *prior* e verossimilhança, o *posterior*  $p(d|\tilde{X}_i)$ .

## Conclusões

Neste trabalho, avaliou-se a biblioteca SVO, proposta por Forster et al. (2014) para odometria visual em drones, como uma alternativa capaz de produzir dados suficientemente precisos para reconstrução 3D. Utilizando restrição epipolar, concluímos que os resultados obtidos pela SVO podem ser utilizados na obtenção de matrizes de projeção suficientemente acuradas para reconstrução tridimensional. Na prática, isto significa que o método pode ser utilizado em aplicações de modelagem 3D, manipulação robótica e instrumentação auxiliada por realidade aumentada. Aplicações podem se beneficiar do baixo custo computacional da SVO, que pode operar em processadores modestos utilizados em dispositivos embarcados.

Mostrou-se também que métodos Bayesianos de estimação podem ser utilizados para integrar a informação obtida de várias imagens, atualizando estimativas da posição dos pontos tridimensionais na superfície dos objetos em cena. Implementações eficientes dessa ideia foram propostas recentemente por Vogiatzis e Hernandez (2011) e Pizzoli et al. (2014),



capazes de executar em tempo real. Tais implementações, se combinadas de forma paralela à odometria visual fornecida pela SVO, possibilitam a criação de aplicações capazes de modelar objetos em 3D em tempo real, isto é, produzir modelos 3D de forma instantânea durante a aquisição das imagens. A construção dessas aplicações, no contexto de modelagem da parte aérea de plantas, em ambientes controlados e no campo, será nosso foco em trabalhos subsequentes.

## Referências

- BAKER, S.; MATTHEWS, I. Lucas-Kanade 20 years on: a unifying framework. **International Journal of Computer Vision**, v. 56, n. 3, p. 221-255, Feb. 2004.
- FORSTER, C.; PIZZOLI, M.; SCARAMUZZA, D. SVO: fast semi-direct monocular visual odometry. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2014, Hong Kong. **Proceedings...** [Canadá]: Institute of Electrical and Electronics Engineers, 2014. p. 15-22. Icra.
- FURUKAWA, Y.; PONCE, J. Accurate, dense, and robust multiview stereopsis. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 32, n. 8, p. 1362-1376, Aug. 2010. DOI: 10.1109/TPAMI.2009.161.
- GELMAN, A.; CARLIN, J. B.; STERN, H. S.; DUNSON, D. B.; VEHTARI, A.; RUBIN, D. B. **Bayesian data analysis**. 3rd ed. Boca Raton: CRC, 2013. 675 p. (Chapman & Hall/CRC texts in statistical science, 106).
- GRENZDÖRFFER, G.; ENGEL, A.; TEICHERT, B. The photogrammetric potential of low-cost uavs in forestry and agriculture. **The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, v. 37, n. B1, p. 1207-1214, 2008.
- HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. 2nd ed. Cambridge: Cambridge University, 2004. 673 p.
- IRANI, M.; ANANDAN, P. About direct methods. In: INTERNATIONAL WORKSHOP ON VISION ALGORITHMS, 1999, Corfu. **Vision algorithms: theory and practice: proceedings**. Berlin: Springer, 2000. p. 267-277. (Lecture notes in computer science, v. 1883). Eds Bill Triggs, Andrew Zisserman, Richard Szeliski.
- JAY, S.; RABATEL, G.; GORRETTA, N. In-field crop row stereo-reconstruction for plant phenotyping. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND ASSOCIATED HIGH-TECHNOLOGIES AND EQUIPMENT FOR AGRICULTURE AND FORESTRY, 2., 2014, Madrid. **Proceedings...** [s.n.], 2014. p. 339-348. RHEA-2014.

- LOU, L.; LIU, Y.; SHENG, M.; HAN, J.; DOONAN, J. H. A cost-effective automatic 3D reconstruction pipeline for plants using multi-view images. In: ANNUAL CONFERENCE TAROS, 15., 2014, Birmingham. **Advances in Autonomous Robotics Systems: proceedings**. New York: Springer, 2014. p. 221-230. Editors: Michael Mistry, Aleš Leonardis, Mark Witkowski, Chris Melhuish. (Lecture notes in computer science, v. 8717).
- LOWE, D. G. Distinctive Image Features from Scale-Invariant Keypoints. **International Journal of Computer Vision**, v. 60, n. 1-2, p. 91-110, Nov. 2004.
- MIKOLAJCZYK, K.; TUYTELAARS, T.; SCHMID, C.; ZISSERMAN, A.; MATAS, J.; SCHAFFALITZKY, F.; KADIR, T.; VAN GOOL, L. A comparison of Affine Region Detectors. **International Journal of Computer Vision**, v. 65, n. 1-2, p. 43-72, Oct. 2005.
- PAPROKI, A.; SIRAUT, X.; BERRY, S.; FURBANK, R.; FRIPP, J. A novel mesh processing based technique for 3D plant analysis. **BMC plant biology**, v. 12, p. 63, May 2012.
- PIZZOLI, M.; FORSTER, C.; SCARAMUZZA, D. REMODE: probabilistic, monocular dense reconstruction in real time. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, Hong Kong, 2014. **Proceedings...** New York: Institute of Electrical and Electronics Engineers, 2014. p. 2609-2616. Icra 2014.
- ROVIRA-MÁS, F.; ZHANG, Q.; REID, J. Creation of three-dimensional crop maps based on aerial stereoimages. **Biosystems Engineering**, v. 90, n. 3, p. 251-259, Mar. 2005.
- SANTOS, T. T.; OLIVEIRA, A. A. de. Image-based 3D digitizing for plant architecture analysis and phenotyping. In: CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES, 25.; WORKSHOP ON INDUSTRY APPLICATIONS, 2012, Ouro Preto. **Proceedings...** [S.I.]: Brazilian Computer Society, 2012. Não paginado. SIBGRAPI.
- SLAUGHTER, D.; GILES, D.; DOWNEY, D. Autonomous robotic weed control systems: a review. **Computers and Electronics in Agriculture**, v. 61, n. 1, p. 63-78, Apr. 2008. DOI:10.1016/j.compag.2007.05.008.
- SNAVELY, N.; SEITZ, S.; SZELISKI, R. Modeling the world from Internet photo collections. **International Journal of Computer Vision**, v. 80, n. 2, p. 189-210, Nov. 2008.
- SOUSA, R. V. de; SILVA, A. R. Y. da; INAMASU, R. Y.; PORTO, A. J. V. Robô agrícola móvel (ram): uma revisão das pesquisas recentes sobre sistemas de navegação autônoma de robôs e veículos agrícolas. In: CONGRESSO BRASILEIRO DA SOCIEDADE BRASILEIRA DE INFORMÁTICA APLICADA À AGROPECUÁRIA E À AGROINDÚSTRIA, 4., 2003, Porto Seguro. **Anais...** Lavras: SBI-AGRO, 2003. 1 CD-ROM.
- THRUN, S.; BURGARD, W.; FOX, D. Probabilistic robotics. Cambridge: MIT press, 2005. 647 p.

VOGIATZIS, G.; HERNANDEZ, C. Video-based, real-time multi view stereo. **Image and Vision Computing**, v. 29, n. 7, p. 434-441, Jun. 2011.

ZHANG, Z. A flexible new technique for camera calibration. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 22, n. 11, p. 1330-1334, Nov. 2000.



---

*Informática Agropecuária*