

Universidade Estadual de Campinas UNICAMP

IA368N

Atividade 3

Sensor characterization and line fitting

Vinicius Rodrigues Sanches – RA: 208976

Introdução Teórica

A caracterização do sensor compreende conhecermos os seus parâmetros intrínsecos de erro, acurácia, precisão, faixa de trabalho e outros. Nessa atividade, uma das coisas mais importantes é determinar se o sensor utilizado possui um erro Gaussiano, o que nos possibilita aplicar uma série de técnicas de navegação robótica com ótima precisão. É de fundamental importância que quaisquer sensores utilizados na fusão de sensores apresentem o comportamento Gaussiano. Sem isso, todas as técnicas não se comportam bem, ou até mesmo se tornam impossíveis de serem aplicadas.

Objetivo

O objetivo é coletar dados do sensor laser do robô real no laboratório e determinar seus principais parâmetros, bem como se apresenta comportamento Gaussiano. Outro objetivo é introduzir uma nova técnica de extração de features o landmarks baseados em linhas, que são lidas a partir das leituras das distâncias dos lasers até uma parede ou outro obstáculo linear, representadas em coordenadas polares (uma distância r e um ângulo α).

Estabelecer as características do sensor laser do Pioneer P3-DX.

Após estabelecer conexão com o sensor laser do robô, comandi várias leituras em sequência. Ao todo foram mil leituras de distâncias das paredes com o robô parado com ângulos relativos de -110 graus à 110 graus com 20 graus de intervalo (12 feixes por leitura). Para facilitar o desenvolvimento, fiz um código que faz essas leituras e também os guarda em um arquivo CSV para poder ser utilizado como massa de dados para testes do programa. Segue o trecho de código abaixo:

```
dist = [];  
count = 0;  
while count < 1000  
    aux = [];  
    aux = Pioneer_p3dx_getLaserData(connection,'distances');  
    %if ~ismember(aux,dist,"rows")  
        dist = [dist; aux];  
    %endif  
    count ++;  
    %pause(0.250);  
end  
csvwrite('./data_vinicius.txt',dist);  
%dist = csvread('./data_vinicius.txt');  
  
rows = size(dist)(1);  
columns = size(dist)(2);  
deviation = [];  
average = [];  
variance = [];  
precision = [];  
figure(1);  
for c=1:columns  
    deviation= [deviation std(dist(1:100,c))];
```

```

average = [average mean(dist(1:100,c))];
variance = [variance var(dist(1:100,c))];
precision = [precision average(c)/((max(dist(1:rows,c))-
min(dist(1:rows,c)))+average(c))];

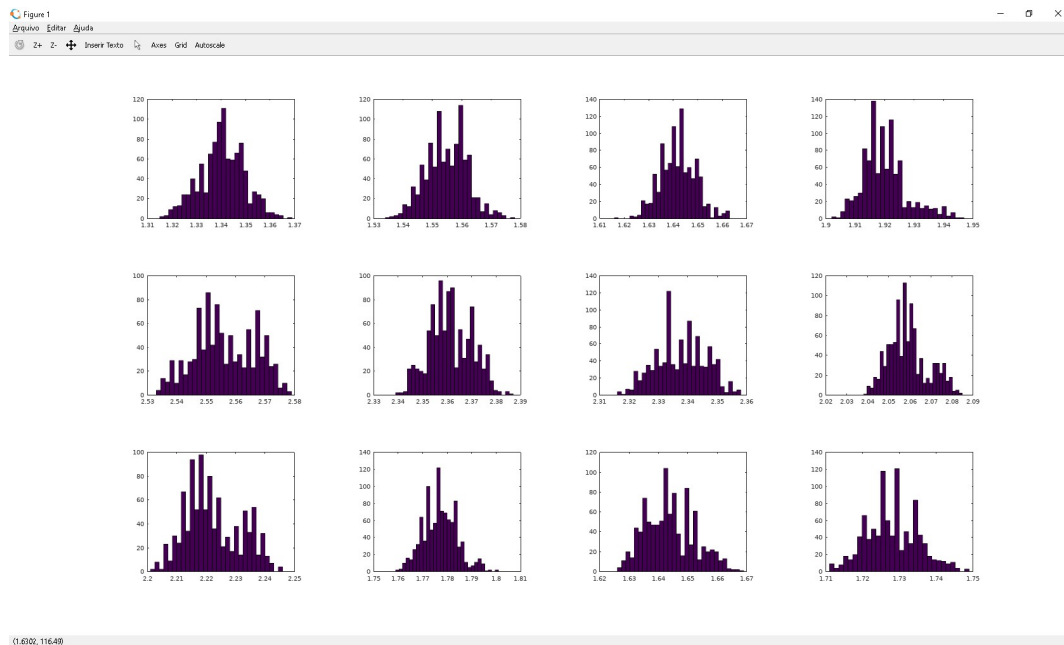
subplot(3,4,c);
hist(dist(:,c),30);

endfor
printf("Std dev: %f\n",deviation);
printf("Media: %f\n",average);
printf("Variância: %f\n",variance);
printf("Precisão: %f\n",precision);
fflush(stdout);

alpha_global = 1;
step = -0.0005;
printf("Relatório de teste de normalidade dos dados.\n");
for c=1:columns
    printf("Feixe de laser: %d\n", c);
    aux = 1;
    for alpha=0.95:step:0.0005
        printf("Alpha: %f\n",alpha);
        [H, pValue, W] = swtest(dist(1:20,c),alpha);
        if H == 1 && aux > alpha
            aux = alpha;
        end
        printf("H: %d\n",H);
        printf("pValue: %d\n",pValue);
        printf("W: %d\n",W);
        fflush(stdout);
    end
    if H == 0
        aux+=step;
    end
    if alpha_global > aux
        alpha_global = aux;
    end
end
end

```

Seguem os histogramas das leituras dos feixes de laser:



Todos os parâmetros descritivos do laser em ordem:

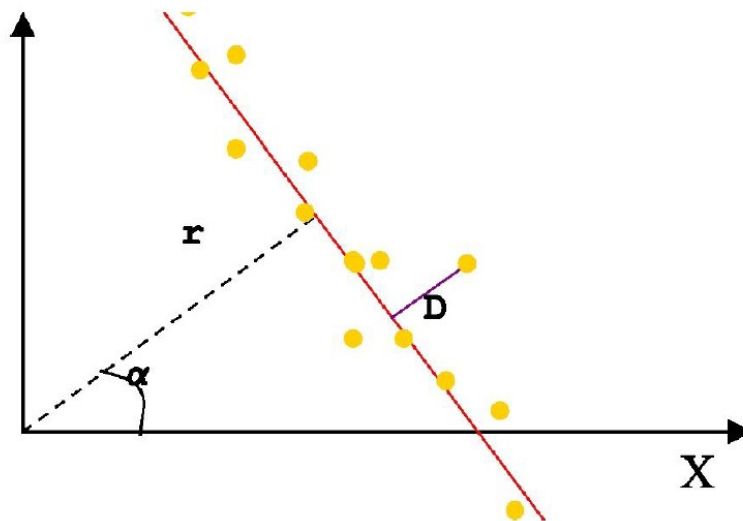
Média	Desvio	Precisão Média (*100%)
1.3406490	0.0090319	0.9612806
1.5549370	0.0069248	0.9724817
1.6418890	0.0071995	0.9721711
1.9202490	0.0074583	0.9771021
2.5562060	0.0103168	0.9823227
2.3615600	0.0086334	0.9800794
2.3376610	0.0084698	0.9823504
2.0592000	0.0093659	0.9776849
2.2224960	0.0093458	0.9801543
1.7771400	0.0068423	0.9769122
1.6439600	0.0082974	0.9745104
1.7281510	0.0070101	0.9784843

No teste global de normalidade(Shapiro-Wilk), procurei um alpha mínimo global para as primeiras 20 amostras. Tendo que ressaltar que se usar mais que isso nessa distribuição de pontos do robô real, não funciona. Esse teste não reage bem a muitos pontos, a não ser que a distribuição dos mesmos seja muito “perfeita”, o que não é o caso desse nosso experimento. O valor global de significância do alpha para as primeiras 20 amostras ficaram em torno de 0.013500.

Ajuste e extração de linhas para localização robótica

Vamos agora verificar a realização das operações de extração de linha. Seguindo a distribuição Gaussiana escolhemos utilizar o sistema de coordenadas polares (r, α) que segue a fórmula de transformação de coordenadas cartesianas:

$$x \cos \alpha + y \sin \alpha = r$$



Usamos para tal operação o algoritmo de Split-and-Merge.

Algorithm 1: Split-and-Merge

Data: Set S consisting of all N points, a distance threshold $d > 0$

Result: L , a list of sets of points each resembling a line

$L \leftarrow (S), i \leftarrow 1;$

while $i \leq \text{len}(L)$ **do**

 fit a line (r, α) to the set L_i ;

 detect the point $P \in L_i$ with the maximum distance D to the line (r, α) ;

if $D < d$ **then**

$i \leftarrow i + 1$

else

 split L_i at P into S_1 and S_2 ;

$L_i \leftarrow S_1; L_{i+1} \leftarrow S_2$;

end

end

Merge collinear sets in L ;

De acordo com as fórmulas abaixo, alteramos o arquivo fitLine.m para gerar o r e o α da linha em questão.

$$S(r, \alpha) := \sum_i \underbrace{(r - x^i \cos \alpha - y^i \sin \alpha)^2}_{=D((\alpha, r), (x^i, y^i))}$$

$$\alpha = \frac{\text{atan2}(\text{nom}, \text{denom})}{2}$$

$$\text{nom} := -2 \sum_i (x^i - x_c)(y^i - y_c)$$

$$\text{denom} := \sum_i (y^i - y_c)^2 - (x^i - x_c)^2$$

Sendo assim, o código foi alterado para implementar essas funções:

```
N = max(size(XY));
```

```
xc = sum(XY(1,:))/N;
```

```

yc = sum(XY(2,:))/N;

nom = -2 * sum((XY(1, :) - xc).*(XY(2, :) - yc));
denom = sum((XY(2, :) - yc).*(XY(2, :) - yc) - (XY(1, :) - xc).*(XY(1, :) - xc));
alpha = atan2(nom,denom)/2;
% compute parameter r (see exercise pages)
r = xc*cos(alpha)+yc*sin(alpha);

```

Os testes com o programa de testes **code/Ex3_LineExtraction/test/testLineFitting.m** e **code/Ex3_LineExtraction/test/testLineExtraction.m** passaram com sucesso.

Conclusões

Pude observar que com relação aos parâmetros dos sensores, nem sempre fica fácil de provar sua normalidade somente utilizando o teste de Shapiro-Wilk.

Outro ponto interessante foi com relação ao algoritmo Split-and-Merge utilizando um sistema de coordenadas polares para extrair retas dos pontos fornecidos pelo laser. Se torna um sistema bem leve, pois só precisa de dois parâmetros para cada reta extraída.

Source code on GitHub

<https://github.com/viniciusrsanches/IA368N>

Bibliografia

Siegwart, Roland.

Introduction to autonomous mobile robots. - 2nd ed. / Roland Siegwart, Illah R. Nourbakhsh, and Davide

Scaramuzza.

p. cm. - (Intelligent robotics and autonomous agents series)

Slides de aula IA368N – Professor Eric Rohmer