

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS BLUMENAU

Trabalho I

Aluno Eduardo Mafra Pereira

Professores Leonardo Mejia Rincon
 Marcos Vinicius Matsuo

Blumenau, Julho de 2019

Conteúdo

1	Identificação da proposta	1
2	Introdução	2
3	Desenvolvimento	3
3.1	Algoritmo para o nível 1	3
3.1.1	Detecção dos pontos e armazenamento da trajetória . .	3
3.1.2	Rastreamento	3
3.1.3	Pseudo-Código	4
3.1.4	Resultados do nível 1	6
3.2	Algoritmo para o nível 2	7
3.2.1	Detecção dos pontos e armazenamento da trajetória . .	7
3.2.2	Rastreamento	7
3.2.3	Pseudo-Código	8
3.2.4	Resultados do nível 2	10
4	Referências	11

1 Identificação da proposta

Implemente no *Matlab* um programa capaz de rastrear objetos em um vídeo utilizando filtro de *Kalman*. O programa deve ser capaz de tratar situações em que a detecção do objeto falha, como por exemplo, a oclusão). O trabalho está dividido em dois níveis de dificuldade:

- *No primeiro nível de dificuldade, o programa deverá ser capaz de rastrear um único objeto de interesse;*
- *No segundo nível de dificuldade, o programa deve ser capaz de rastrear múltiplos objetos de interesse, sendo o número de objetos na cena conhecido e constante. Por exemplo, um vídeo onde sabe-se que há três objetos durante todo o intervalo.*

2 Introdução

O intuito deste projeto é realizar o rastreamento de objetos em vídeo e estimar suas posições em caso de oclusão total ou parcial desses objetos. Sendo assim, foi necessário realizar a detecção dos objetos em vídeo através de processamentos de imagem aplicados a cada *frame*, definir as matrizes de estado referentes a posição e velocidade de cada objeto em planos de duas dimensões, aplicar o filtro de *Kalman* para encontrar as posições estimadas destes objetos, além de outros procedimentos que serão abordados no decorrer do relatório.

Para solucionar os problemas propostos foram construídos dois algoritmos. O primeiro algoritmo realizará os rastreamentos de um único objeto em vídeo utilizando filtro de *Kalman*, já para o segundo algoritmo serão rastreados dois objetos em um vídeo utilizando o conceito de filtro de *Kalman* e algoritmo húngaro.

3 Desenvolvimento

Nesta seção serão apresentados os algoritmos descrevendo todas suas etapas de processamento.

3.1 Algoritmo para o nível 1

3.1.1 Detecção dos pontos e armazenamento da trajetória

Na primeira etapa do algoritmo são extraídas as coordenadas do centroide do objeto detectado e armazenadas em uma matriz de trajetórias “ z ”, em que as primeiras linhas corresponderão a posição do objeto referente ao eixo das abscissas e a segunda linha ao eixo das ordenadas, este vetor será necessário para realizar o rastreamento do objeto. O processo acontece dentro de um laço de repetição com os seus números de repetições iguais aos números de *frames* do vídeo. A cada repetição são obtidas as matrizes RGB correspondentes ao *frame* em questão e são convertidas para uma imagem *HSV*. A conversão foi proposta para facilitar a detecção do objeto a partir dos valores contidos nas camadas *HSV*, foi necessário também estimar um range de valores que contemple a cor do objeto na imagem *HSV*. Com o objeto detectado, finalmente são obtidas as coordenadas do centroide deste objeto a cada *frame* por meio da função “*iblobs*” e armazenadas a um vetor “ z ”. Os ciclos nos quais não são possíveis detectar nenhum objeto no vídeo, serão armazenados no vetor “ z ” elementos “*Not-a-Number*”.

3.1.2 Rastreamento

Com o vetor de trajetória é possível realizar o rastreamento do objeto. E para o rastreamento foi proposto um filtro de *Kalman* a partir do modelo de espaço de estados de posições e velocidade de um objeto em duas dimensões, e da determinação dos coeficientes de incerteza “ σ_W ” e “ σ_S ” aplicados ao filtro de *Kalman*, estes coeficientes foram definidos de maneira empírica. Para o rastreamento também foi necessário definir o período de amostragem “ T ”. Este tipo de estratégia é bastante útil para sistemas de rastreamento, principalmente em casos no qual ocorrem oclusão total de objetos e quando se deseja manter os índices atribuídos a cada objeto durante sua detecção. Este segundo exemplo será tratado novamente na seção referente ao nível 2, no qual serão rastreados múltiplos objetos.

O modelo fornece portanto as matrizes de modelo do sistema, “ $matriz_F$ ” e “ $matriz_H$ ”, já as incertezas possibilitam encontrar as matrizes de covariância do sistema, “ $matriz_W$ ” e “ $matriz_S$ ”.

A implementação do filtro de *Kalman* requer a criação de vetores de predição e estimação. A predição baseia-se no modelo dinâmico do sistema (matrizes F e H), no ponto de detecção atual e nas estimativas dos pontos anteriores. Já a estimação será a correção do estado predito a partir do ponto de detecção atual. Estes vetores são preenchidos no decorrer da utilização do filtro e representam a dinâmica de rastreamento do objeto, portanto caso haja oclusão de objeto, o vetor de rastreamento estimado irá considerar apenas as posições e velocidades preditas, no entanto se houver detecção do objeto no *frame*, serão considerados os valores estimados de posição e velocidade definidos como a correção do estado predito. Este processo será realizado no decorrer do algoritmo e será implementado por meio de condições “*if*” dentro de um laço de repetição.

Para realizar a utilização do filtro de *Kalman* *frame a frame* no algoritmo, se faz necessário inicialmente determinar as condições iniciais do processo, estas condições serão atribuídas a primeira coluna do vetor de estimação, a matriz de covariância do estado predito e a matriz de covariância do estado estimado, estas matrizes serão inicialmente descritas por uma matriz identidade de dimensão 4×4 e uma matriz quadrada também de dimensões 4×4 preenchida inteiramente por zeros, respectivamente. Já a condição inicial do vetor de estimação será representada pelo primeiro elemento do vetor de detecção que não for um elemento “*Not-a-Number*”.

Com as condições iniciais determinadas é possível realizar o rastreamento dos elementos através de um laço de repetição, o número de repetições para este laço é proporcional ao número de colunas presentes na matriz de elementos detectados anteriormente. A aplicação do filtro de *Kalman* será realizada a cada interação em que resultará nos valores de posições e velocidades estimadas e na matriz de covariância do estado estimado, para que esta possa ser utilizada na próxima interação. As posições e velocidades estimadas serão atribuídas e armazenadas na matriz de estado estimado, para que possam ser utilizadas na simulação dos resultados em vídeo.

Finalmente com os resultados estimados obtidos, foi implementado um algoritmo para sobrescrever os resultados obtidos no mesmo vídeo utilizado para a detecção do objeto.

3.1.3 Pseudo-Código

Nesta seção será desenvolvido um pseudo-código que representa a implementação do algoritmo original. Este terá 3 etapas principais, são elas: Aquisição das detecções, rastreamento e simulação dos resultados.

Aquisição das detecções:

- *Carrega o vídeo;*

- Atribui a variável “ n ” ao tamanho do vídeo;
- Defini uma matriz de detecção $2 \times n$;
- Laço de detecção;
 - Carrega frame;
 - Converte imagem para o formato “HSV”;
 - Extrai apenas os objetos dentro de um range definido na imagem “HSV”;
 - Dilata a imagem resultante;
 - Extrai propriedades apenas dos objetos com área definida e com valor igual a 1;
 - Se houver detecção, atribui-se a matriz de detecção as posições do centroide referente ao objeto detectado;
 - Se não houver detecção atribui-se a matriz de detecção valores NaN;
- Defini o tempo de amostragem;
- Defini os coeficientes de incerteza “ σ_W ” e “ σ_S ”;
- Defini as matrizes do modelo sistema “ $matriz_F$ ” e “ $matriz_H$ ”;
- Defini as matrizes de covariância;
- Atribui a condição inicial da matriz estimada;
- Defini as condições iniciais para as matrizes de covariância do estado predito e do estado estimado;
- Laço de rastreamento;
 - Defini o vetor predito;
 - Calcula as variáveis estimadas através do Kalman;
- Laço de simulação;
 - Mostra “frame”;
 - Plota os quatro últimos elementos detectados;
 - Plota os quatro últimos elementos estimados;

Vale ressaltar que para o cálculo de *Kalman* foi construída uma função, e esta é inicializada a cada laço de rastreamento com os seguintes parâmetros: “*matriz_F*”, “*matriz_H*”, detecção atual, vetor predito, matriz de covariância do estado estimado, “*matriz_W*” e “*matriz_S*”.

3.1.4 Resultados do nível 1

O algoritmo atendeu aos requisitos propostos, rastreando o objeto durante todo o percurso, inclusive nos momentos em que ocorrem oclusões. As figuras 2 e 3 representam os resultados obtidos através da implementação do algoritmo, sendo o *plot* em amarelo referente a detecção do ponto atual e dos quatro últimos pontos e o *plot* em azul referente ao rastreamento do objeto.



Figura 1: Frame original.

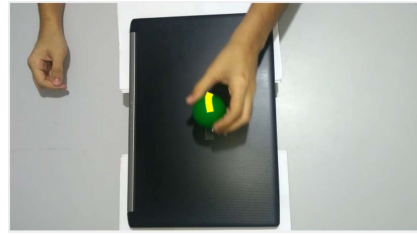


Figura 2: Frame apenas com detecção.



Figura 3: Resultado final, detecção e rastreamento.

3.2 Algoritmo para o nível 2

Para o nível dois foi necessário implementar um algoritmo que realize o rastreamento de múltiplos objetos em vídeo, portanto o algoritmo proposto foi implementado para dois objetos e utiliza como a base de sua implementação o algoritmo do nível 1.

3.2.1 Detecção dos pontos e armazenamento da trajetória

Na fase de detecção para este nível também é realizada a segmentação de objetos através da escala “*HSV*” e após segmentados estes objetos, suas imagens resultantes são somadas com o objetivo de tornar a resolução do problema mais próxima da realidade. Caso houver dois objetos detectados nesta nova imagem, serão atribuídos as matrizes de detecção “ Z_1 ” e “ Z_2 ” as posições dos centroides detectados. Caso exista apenas um objeto detectado, as coordenadas de seu centroide serão atribuídas a matriz de posição “ Z_1 ” e os elementos “*Not-a-Number*” serão atribuídos a matriz “ Z_2 ” e se não houver nenhum objeto detectado, os elementos “*Not-a-Number*” serão atribuídos a ambas as matrizes.

3.2.2 Rastreamento

As condições iniciais são atribuídas de características semelhantes ao nível um, todavia agora são construídas duas matrizes de estimação, cada uma corresponderá a um objeto. Além das matrizes de estimação, são inicializadas também as matrizes de covariância do estado predito e as matrizes de covariância do estado estimado para cada objeto.

Para o laço de rastreamento “*frame a frame*”, será necessário atribuir as coordenadas de detecção corretas a função de “*Kalman*” para cada objeto. A resolução deste problema se dá através do cálculo das distâncias entre os vetores preditos e a utilização do algoritmo húngaro para encontrar a solução ótima para este problema. Portanto, para a utilização do algoritmo húngaro deve-se inicialmente calcular o vetor de predição relativo a cada objeto, e calcular a distância de suas coordenadas em relação aos objetos detectados no *frame* em questão. Estas distâncias são atribuídas a uma matriz que será utilizada como entrada do algoritmo húngaro. A matriz de distâncias apresenta-se da seguinte maneira:

Elementos	<i>Posição Predita₁</i>	<i>Posição Predita₂</i>
Z_1	<i>Distância₁₁</i>	<i>Distância₁₂</i>
Z_2	<i>Distância₂₁</i>	<i>Distância₂₂</i>

Tabela 1: Matriz de distâncias

O elementos aplicados a matriz da tabela 1 são encontrados calculando a distância euclidiana entre os elementos das linhas e colunas da matriz. Por exemplo, o elemento *Distância₁₁* refere-se a distância entre as coordenadas da *Posição Predita₁* e as coordenados do elemento detectado Z_1 . Com a matriz calculada a função do algoritmo húngaro poderá ser utilizada. A entrada desta função é a matriz de distâncias representada na tabela 1, e ela retornará um vetor de elementos que apresenta a melhor solução para manter o rastreamento adequado aos resultados anteriores. Caso algum dos elementos retornados na matriz resultante seja igual a zero, significa que houve oclusão de algum dos objetos e portanto para este objeto será atribuído apenas o elemento predito correspondente.

Com o vetor de posição encontrado, é possível utilizar o algoritmo de “*Kalman*” de modo correto. Desta forma, atribuindo para as funções de “*Kalman*” os pontos de detecção correspondentes com as posições a serem estimadas.

Vale ressaltar que no nível 2, além de utilizar a função que calcula o “*Kalman*” já utilizada para o nível 1, da mesma forma, foi desenvolvida uma função para calcular a matriz de distâncias.

Assim como no nível 1, os resultados serão sobrescritos ao vídeo utilizado com o intuito de validá-los.

3.2.3 Pseudo-Código

Esta secção segue o mesmo modelo utilizado na secção 3.1.3 e consiste em três etapas principais, são elas: Aquisição das detecções, rastreamento e simulação dos resultados.

Aquisição das detecções:

- *Carrega o vídeo;*
- *Atribui a variável “n” ao tamanho do vídeo;*
- *Defini as matrizes de detecção $2 \times n$;*
- *Laço de detecção;*
 - *Carrega frame;*

- Converte imagem para o formato “HSV”;
- Extrai apenas os objetos dentro de um range definido na imagem “HSV”;
- Dilata os objetos presentes na imagem resultante;
- Extrai propriedades apenas dos objetos com área definida e com valor igual a 1;
- Se houver detecção, atribui-se a matriz de detecção as posições do centroide referente aos objetos detectados;
- Se houver apenas um elemento detectado atribui-se a matriz de detecção Z_2 os valores de NaN;
- Se não houver detecção atribui-se a ambas as matrizes de detecção os valores de NaN;
- Defini o tempo de amostragem;
- Defini os coeficientes de incerteza “ σ_W ” e “ σ_S ”;
- Defini as matrizes do modelo sistema “matriz_F” e “matriz_H”;
- Defini as matrizes de covariância;
- Atribui as condições iniciais das matrizes estimadas;
- Defini as condições iniciais para as matrizes de covariância dos estados preditos e dos estados estimados;
- Laço de rastreamento;
 - Defini os vetores preditos;
 - Calcula matrizes de distâncias;
 - Ordena os elementos através do algoritmo húngaro;
 - Calcula as variáveis estimadas através do “Kalman”;
- Laço de simulação;
 - Mostra “frame”;
 - Plota os quatro últimos elementos estimados;

3.2.4 Resultados do nível 2

Para o nível 2, do mesmo modo, pode-se afirmar que a implementação do algoritmo atendeu bem aos requisitos propostos, rastreando os objetos durante todos os seus percursos, inclusive em momentos de oclusão. Além disto, é importante ressaltar que o algoritmo consegue manter os índices atribuídos a cada objeto sem que estes sejam trocados, validando, portanto, o resultado do projeto.

A figura 5 representa o resultado obtido através da implementação do algoritmo, sendo os *plots* referentes aos quatro últimos elementos estimados.



Figura 4: Frame original.

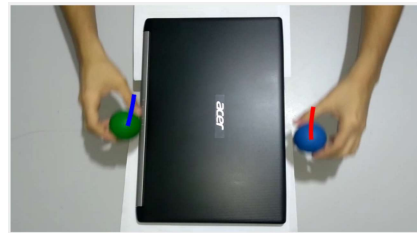


Figura 5: Frame com rastreamento.

4 Referências

CORKE, Peter. Robotics, Vision and Control: Fundamental Algorithms in MATLAB. Springer Verlag NY, 1, 2011. ISBN: 3642201431.

RINCON, Leonardo Mejia. Visão Computacional em robótica: Notas de aula. Blumenau, SC, 2019.

MATSU, Marcos. Visão Computacional em robótica: Notas de aula. Blumenau, SC, 2019.