

Universidade da Beira Interior

Departamento de Informática



Departamento de
Informática

**Nº 93 - 2020: *Sistema de detecção e tracking de
objectos em movimentocom base em análise de
imagem***

Elaborado por:

David Miguel Martins Pires

Orientador:

Professor Doutor Pedro Domingues de Almeida

26 de Fevereiro de 2020

Agradecimentos

A conclusão deste trabalho, bem como da grande maior parte da minha vida académica não seria possível sem a ajuda de ...

- do professor Pedro Domingues de Almeida por toda a orientação e ajuda providenciada durante a elaboração deste projeto
- todos os meus docentes, colegas e amigos que me acompanharam durante este percurso académico
- Ao professor Pedro Inácio na proveniência das bases para a realização deste Relatório e de uma Apresentação a ser realizada no final deste Trabalho.

Conteúdo

Conteúdo	iii
Lista de Figuras	v
Lista de Tabelas	vii
1 Introdução	1
1.1 Enquadramento	1
1.2 Motivação	1
1.3 Objetivos	2
1.4 Organização do Documento	2
1.5 Algumas Dicas – [RETIRAR DA VERSÃO FINAL]	3
2 Estado da Arte	5
2.1 Introdução	5
2.2 Estado atual do reconhecimento de Objetos	5
2.3 Funcionamento do Reconhecimento de Objetos	5
2.4 Algoritmos Estudados	5
2.4.1 Algoritmos de Detecção de Objetos	6
2.4.2 Tensorflow	7
2.4.3 OpenCV	7
2.4.4 Yolo	10
2.4.5 Hungarian Algorithm (Kuhn-Munkres)	10
2.4.6 Kalman Filter	10
2.4.7 Another ONE	11
2.5 Citações e Referências Cruzadas – [RETIRAR DA VERSÃO FI- NAL]	11
2.6 Secções Intermédias	11
2.7 Conclusões	11

3	Tecnologias e Ferramentas Utilizadas	13
3.1	Introdução	13
3.2	Python	13
3.3	Secções Intermédias	13
3.4	Conclusões	14
4	Implementação e Testes	15
4.1	Introdução	15
4.2	Secções Intermédias	15
4.3	Conclusões	16
5	Conclusões e Trabalho Futuro	17
5.1	Conclusões Principais	17
5.2	Trabalho Futuro	17

Lista de Figuras

Lista de Tabelas

3.1	Esta é uma tabela de exemplo.	13
-----	---------------------------------------	----

Acrónimos

Raspberry *Raspberry Pi 4*

frame
feed (de video)

Capítulo 1

Introdução

1.1 Enquadramento

Sistema de detecção e tracking de objectos em movimento com base em análise de imagem

- Visão Computacional - Identificação de objetos em imagem (2D, 3D) ou em um video sobre um ambiente de luz natural. Desafio para os sistemas de Visão Computacional, sendo que já foram implementadas muitas abordagens para estes sistemas ao longo dos anos.
- O Tracking de objetos é também um desafio para estes sistemas de Visão Computacional.

Neste projeto todo o código é desenvolvido usando um Raspberry Pi, com capacidades limitadas, o que torna este trabalho ainda um maior desafio ao nível da Complexidade Computacional.

- Complexidade Computacional

Usaremos também uma camera comum e conectada ao Raspberry Pi para obter as imagens para analisar em tempo real. Usarei uma camera da marca Logitech que tenho em casa, conectada via USB.

Para isto iremos usar irei usar um Raspberry Pi 4, Modulo B de 4GB RAM que tenho em casa e um cartão de memória de 64 Gb.

1.2 Motivação

O tema deste projeto é um tema atual e muito importante para as áreas de Visão Computacional e da Robotica. Daí ser uma grande motivação minha na escolha deste trabalho.

O reconhecimento de objetos será sempre uma area imprescindivel e de grande interesse para a robotica. Neste caso iremos reconhecer alguns objetos e rastrea-los em um fluxo de video. Isto ainda em um ambiente de desenvolvimento com

capacidades bastante limitadas, que é o Raspberry Pi, o que torna esta tarefa um desafio.

Apesar de nós humanos, conseguirmos reconhecer múltiplos objetos em diferentes imagens, e em diferentes disposições em segundos sem qualquer dificuldade. Muitas vezes mesmo Objetos obtruidos por outros objetos. Isto para a Visão Computacional é ainda um desafio, e é muito mais complexo reconhecer um objeto em uma imagem do que pensamos à primeira vista.

- Lab - Prof - Raspberry Pi

— Old:

A motivação da escolha deste projeto como projeto deve-se ao meu interesse neste tema de projeto. Tenho também muito interesse no aprendizado das ferramentas utilizadas como por exemplo a programação em um dispositivo com capacidades reduzidas, como um *Raspberry Pi 4* (Raspberry).

1.3 Objetivos

Como objetivo deste trabalho temos a elaboração de um Sistema de detecção e tracking de objetos em movimento com base em análise de imagem. Este sistema deve ser desenvolvido para posteriormente ser usado em um Raspberry Pi, com capacidade de processamento limitada. Devemos também usar uma webcam e ferremntas de programação como Tensorflow, Opencv e Yolo.

1.4 Organização do Documento

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – apresenta o projeto, a motivação para a sua escolha, o enquadramento para o mesmo, os seus objetivos e a respetiva organização do documento.
2. O segundo capítulo – **Tecnologias Utilizadas** – descreve os conceitos mais importantes no âmbito deste projeto, bem como as tecnologias utilizadas durante do desenvolvimento da aplicação.
3. ...

1.5 Algumas Dicas – [RETIRAR DA VERSÃO FINAL]

Os relatórios de projeto são individuais e preparados em \LaTeX , seguindo o formato disponível na página da unidade curricular. Deve ser prestada especial atenção aos seguintes pontos:

1. O relatório deve ter um capítulo Introdução e Conclusões e Trabalho Futuro (ou só Conclusões);
2. A última secção do primeiro capítulo deve descrever sucintamente a organização do documento;
3. O relatório pode ser escrito em Língua Portuguesa ou Inglesa;
4. Todas as imagens ou tabelas devem ter legendas e ser referidas no texto (usando comando `\ref{}`).

Capítulo 2

Estado da Arte

2.1 Introdução

Cada capítulo intermédio deve começar com uma breve introdução onde é explicado com um pouco mais de detalhe qual é o tema deste capítulo, e como é que se encontra organizado (i.e., o que é que cada secção seguinte discute).

—

Python

2.2 Estado atual do reconhecimento de Objetos

2.3 Funcionamento do Reconhecimento de Objetos

2.4 Algoritmos Estudados

subsec definições : se houver

2.4.1 Algoritmos de Detecção de Objetos

color thresholding + contour extraction

Haar cascades

HOG + Linear SVM

SSDs

Faster R-CNNs

Yolo

Yolo (You Only Look Once) é um algoritmo de detecção de objetos que aceita um feed de vídeo em tempo real. No Yolo, com uma única CNN ??? ??? simultaneamente prevemos múltiplas caixas delimitadoras e probabilidades de classes para essa caixa. O Yolo treina em imagens completas e otimiza diretamente o desempenho da detecção. Este modelo afirma ter um número de benefícios sobre outros métodos de detecção de objetos, como:

- YOLO encontra todos os objetos em uma imagem simultaneamente.
- Usa uma única rede convolucional para toda a imagem.
- Usa características da imagem inteira para prever cada caixa delimitadora. Também preve todas as caixas delimitadoras de todas as classes para uma imagem simultaneamente. Prevê as caixas delimitadoras e as probabilidades de classe para essas caixas.
- YOLO vê a imagem inteira durante o treino e tempo de teste, portanto codifica implicitamente informações contextuais sobre classes tal como a sua aparência.
- YOLO aprende representações generalizadas de objetos para que quando treinado em imagens naturais e testado em obras de arte, o algoritmo supera outros modelos de detecção.

Trabalho do YOLO

- YOLO pega em uma imagem e divide-a em uma rede $S \times S$. Cada célula prevê apenas um objeto. - A classificação e localização da imagem são aplicados em cada célula. - Se o centro de um objeto fica em uma célula, essa célula é responsável por detectar esse objeto. - Cada célula da rede prevê B caixas delimitadoras com classificações de confiança para essas caixas.

1. Caixas de confiança refletem o quão confiante o modelo está sobre a caixa conter o objeto e quão preciso ele pensa a caixa ser o que ele pensa. Se o objeto não existir, então a classificação de confiança será zero.

A previsão de confiança representa a IOU entre a caixa de previsão e qualquer

caixa de verdade do solo. $\text{Pr}(\text{Object}) * \text{IOU verdade previsão} - \text{interseção de união (IOU) entre a caixa prevista e o verdadeiro solo}$.

The output of the algorithm is a list of bounding box, in format [class, x, y, w, h, confidence]. The class is an id related to a number in a txt file (0 for car , 1 for pedestrian, ...). x, y, w and h represent the parameters of the bounding box. x and y are the coordinates of the center while w and h are its size (width and height). The confidence is a number expressed in

2.4.2 Tensorflow

Tensorflow Lite

2.4.3 OpenCV

Opencv é uma biblioteca *open source* criada em 2000 pela Intel e com Licença BSD Intel para a sua Distribuição. Utilizada para o desenvolvimento de aplicações na área de computação visual, seja para o uso acadêmico ou comercial. Possui módulos para processamento de imagens, estrutura de dados, álgebra linear, bem como uma interface gráfica para usuário e mais de 350 algoritmos de visão computacional, como filtros de imagem, calibração de câmera, reconhecimento de objetos, análise estrutural e outros.

Centroid tracking

Centroid tracking é um algoritmo presente na biblioteca OpenCV, que depende da distância euclidiana¹ entre centróides de objetos (ex. objetos que o rastreador de centróide já viu antes) e novos centróides de objetos entre quadros subsequentes em um video.

O algoritmo de rastreamento de centróides é um processo de multiplas etapas:

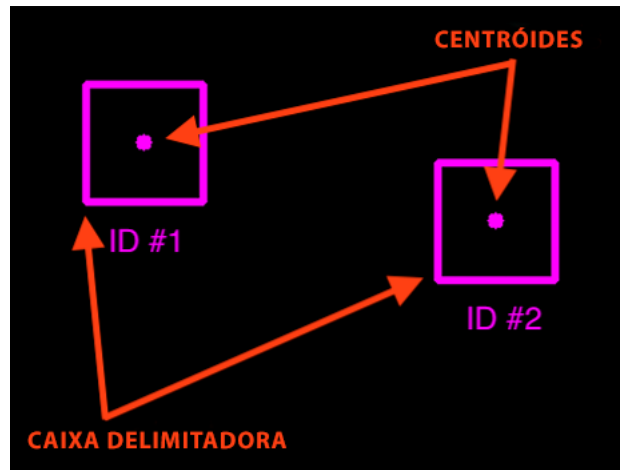
1- Aceitar as coordenadas da caixa delimitadora do objeto e calcular os centróides

Este algoritmo assume que estamos a passar um conjunto delimitador de coordenadas 2D de cada objeto detectado em *every single frame* (cada frame de video). Estas caixas delimitadoras podem ser produzidas por qualquer tipo de detetor de objetos que mais gostemos (temos em este capítulo alguns algoritmos de delimitadores explicados).

¹Distância entre dois pontos

Depois de obtermos as coordenadas das caixas delimitadoras, devemos determinar o centróide, ou simplesmente, o centro da caixa.

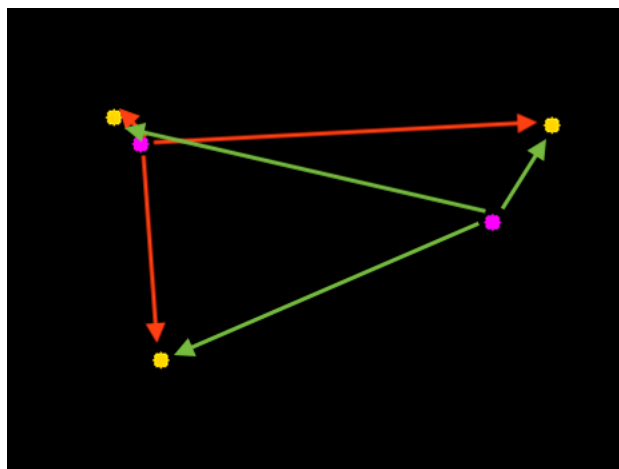
Visto que este é o conjunto inicial de caixas, devemos lhes atribuir IDs únicos.



2- Calcular distância euclidiana entre novas caixas delimitadoras e objetos existentes

Para cada subsequente frame do nosso vídeo, aplicamos a etapa 1 de calcular os centróides de cada objeto. Porém aqui, invés de atribuir um ID único a cada objeto detectado (que iria contrariar o sentido de rastrear um objeto), precisamos determinar primeiro se podemos acociar novos centróides de objetos (pontos amarelos) a aos centróides antigos (pontos roxos). Para este processo, nós calculamos a distância euclidiana entre cada par de objetos existente e novos objetos.

Mas como vamos usar as distâncias Euclidianas entre estes pontos para combiná-los e associá-los?

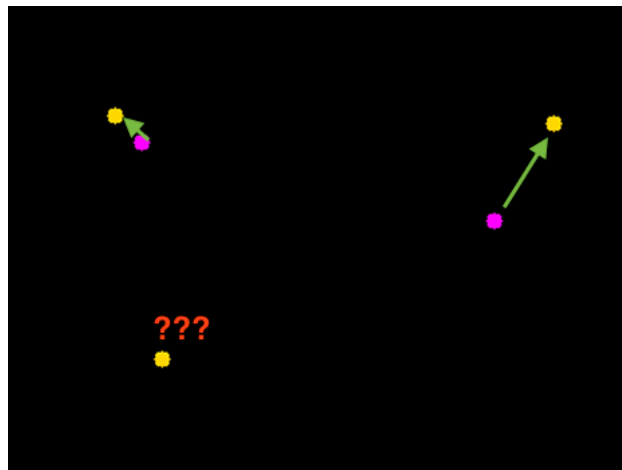


3- Atualizar coordenadas para objetos existentes

A suposição primária do algoritmo de rastreamento de centroides de objetos é de que o objeto irá se potencialmente mover entre frames subsequentes, porém a distância entre centróides será menor que outras distâncias entre objetos.

Portanto, se escolhermos associar centróides com a menor distância entre frames subsequentes, podemos construir o nosso rastreador de objetos.

Mas e se surgir um novo objeto?

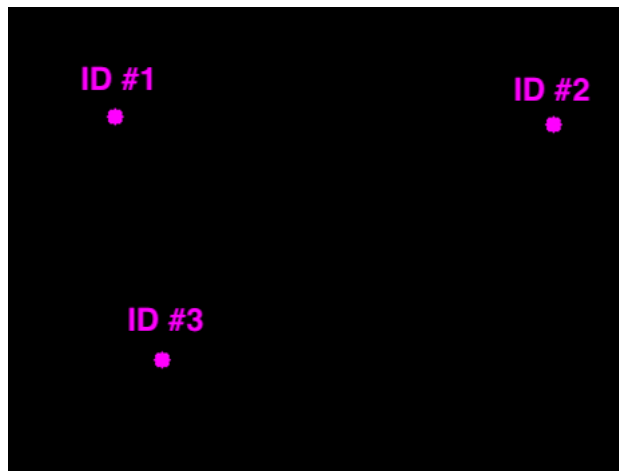


4- Registrar novos objetos

No caso de obtermos mais objetos detectados do que objetos a ser rastreados, precisamos registrar o novo objeto. Registrar o novo objeto na lista de objetos a ser rastreados significa que:

- Atribuímos um novo ID
- Guardamos o centróide da caixa delimitadora para esse objeto

Podemos ir até a etapa 2 e repetir o processo de etapas em cada frame de vídeo.



ref: <https://www.pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/>

2.4.4 Yolo

Yolo (You Only Look Once)

Um dos algoritmos de detecção de objetos mais popular.

The output of the algorithm is a list of bounding box, in format [class, x, y, w, h, confidence]. The class is an id related to a number in a txt file (0 for car , 1 for pedestrian, ...). x, y, w and h represent the parameters of the bounding box. x and y are the coordinates of the center while w and h are its size (width and height). The confidence is a number expressed in

2.4.5 Hungarian Algorithm (Kuhn-Munkres)

A Hungarian algorithm can tell if an object in current frame is the same as the one in previous frame. It will be used for association and id attribution

<https://towardsdatascience.com/computer-vision-for-tracking-8220759eee85>

2.4.6 Kalman Filter

A Kalman Filter is an algorithm that can predict future positions based on current position. It can also estimate current position better than what the sensor is telling us. It will be used to have better association.

<https://towardsdatascience.com/computer-vision-for-tracking-8220759eee85>

2.4.7 Another ONE

subsec outras funções ou bibliotecas consideradas importantes : se houver

2.5 Citações e Referências Cruzadas – [RETIRAR DA VERSÃO FINAL]

Para se referenciar outras secções, usar `\ref{label}`, e.g., para citar a secção da Introdução deste capítulo, usar `\ref{chap2:sec:intro}`. O resultado é: a secção 2.1 contém a introdução deste capítulo.

Para se citarem fontes bibliográficas, colocar a entrada certa no ficheiro `bibliografia.bib` e usar o comando `\cite{label-da-referencia}`, ligando o comando com a palavra que o antecede com um til. Por exemplo, para citar a referência eletrónica *The Not So Short Introduction to L^AT_EX* [?], deve incluir-se o trecho seguinte no ficheiro `bibliografia.bib` e usar `\cite{short}` para a citação (citação incluída nesta mesma frase):

```
@MISC{short,  
  author = {Tobias Oetiker and Hubert Partl and Irene Hyna and Elisabeth Schlegl},  
  title = "{The Not So Short Introduction to \LaTeX{}}",  
  year = 2018,  
  note = {[Online] \url{https://tobi.oetiker.ch/lshort/lshort.pdf}.  
        Último acesso a 12 de Março de 2019}  
}
```

2.6 Secções Intermédias

2.7 Conclusões

Cada capítulo intermédio deve referir o que demais importante se conclui desta parte do trabalho, de modo a fornecer a motivação para o capítulo ou passos seguintes.

Capítulo 3

Tecnologias e Ferramentas Utilizadas

3.1 Introdução

Cada capítulo intermédio deve começar com uma breve introdução onde é explicado com um pouco mais de detalhe qual é o tema deste capítulo, e como é que se encontra organizado (i.e., o que é que cada secção seguinte discute).

3.2 Python

Python é a linguagem de programação usada para o desenvolvimento deste projeto

3.3 Secções Intermédias

A tabela 3.1 serve apenas o propósito da exemplificação de como se fazem tabelas em \LaTeX .

campo 1	campo 2	campo 3
14	15	16
13	13	13

Tabela 3.1: Esta é uma tabela de exemplo.

3.4 Conclusões

Cada capítulo intermédio deve referir o que demais importante se conclui desta parte do trabalho, de modo a fornecer a motivação para o capítulo ou passos seguintes.

Capítulo 4

Implementação e Testes

4.1 Introdução

Cada capítulo intermédio deve começar com uma breve introdução onde é explicado com um pouco mais de detalhe qual é o tema deste capítulo, e como é que se encontra organizado (i.e., o que é que cada secção seguinte discute).

4.2 Secções Intermédias

O trecho de código seguinte mostra a função `main()` e o seu funcionamento:

```
#include <stdio.h>

int main() {
    int i = 0;
    for(i = 0; i < 100; i++)
        printf("%d\n", i);
}
```

Excerto de Código 4.1: Trecho de código usado no projeto.

Se quiser definir a distribuição de Pareto, posso colocar a fórmula *inline*, da seguinte forma $P(x) = \frac{x_i^{1/\Lambda}}{2}$, ou numa linha em separada, como se mostra a seguir:

$$y^2 = \sum_{x=0}^{20} (x^3 - 2x + 3).$$

Outra maneira, mas numerada, é usar o ambiente `equation`, como se mostra na (4.1):

$$y^2 = \sum_{x=0}^{20} (x^3 - 2x + 3). \tag{4.1}$$

$$2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + y^2 = \sum_{x=0}^{20} (x^3 - 2x + 3); \quad (4.2)$$

$$= x^4 - 2. \quad (4.3)$$

4.3 Conclusões

Cada capítulo intermédio deve referir o que demais importante se conclui desta parte do trabalho, de modo a fornecer a motivação para o capítulo ou passos seguintes.

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Conclusões Principais

Esta secção contém a resposta à questão:

Quais foram as conclusões principais a que o(a) aluno(a) chegou no fim deste trabalho?

5.2 Trabalho Futuro

Esta secção responde a questões como:

O que é que ficou por fazer, e porque?

O que é que seria interessante fazer, mas não foi feito por não ser exatamente o objetivo deste trabalho?

Em que outros casos ou situações ou cenários – que não foram estudados no contexto deste projeto por não ser seu objetivo – é que o trabalho aqui descrito pode ter aplicações interessantes e porque?

Bibliografia

- [1] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. The Not So Short Introduction to \LaTeX , 2018. [Online] <https://tobi.oetiker.ch/lshort/lshort.pdf>. Último acesso a 12 de Março de 2019.