

VISÃO COMPUTACIONAL

Tópicos selecionados e aplicações como a biblioteca openCV.



EMENTA TEÓRICA:

1. O que é visão computacional?
 1. Um problema difícil.
 2. A visão do ser humano.
 3. Aplicações praticas com visão computacional.
 4. Biblioteca OpenCV.
2. Imagens
 1. Câmeras
 1. O modelo da câmera escura simples.
 2. Imagens
 1. Amostragem.
 2. Quantização.
 3. Imagens coloridas
 1. RGB.
 2. CMY.
 3. YUV.
 4. HLS.
 5. Outros espaços coloridos.
 6. Algumas aplicações com cores.

EMENTA TEÓRICA:

- 4. Ruídos
 - 1. Tipos de ruído.
 - 2. Modelos de ruído.
 - 3. Geração de ruído.
 - 4. Avaliação do ruído.
- 5. Suavização
 - 1. Média de imagens
 - 2. Média localizada e blur gaussiano.
 - 3. Mascaras de rotação
- 3. Visão Binária
 - 1. Thresholding
 - 1. Problemas envolvendo Thresholding.
 - 2. Métodos de detecção do limiar
 - 1. Limiar de Otsu.
 - 3. Matemática Morfológica
 - 1. Dilatação.
 - 2. Erosão.
 - 3. Aberturas e Fechamentos.
 - 4. Conectividade

EMENTA TEÓRICA:

4. Transformações Geométricas

1. Especificando o problema
2. Transformações Afins
 1. Conhecidas
 2. Desconhecidas
3. Transformação de Perspectiva
4. Interpolação
 1. Vizinho mais próximo
 2. Bi linear
 3. Bi cúbica
5. Modelagem e remoção de distorção em câmeras

5. Vídeo

1. Detecção de objetos em movimento
 1. Objeto de interesse
 2. Problemas comuns
 3. Imagem da diferença
 4. Modelos de fundo
 5. Detecção de sombras

EMENTA TEÓRICA:

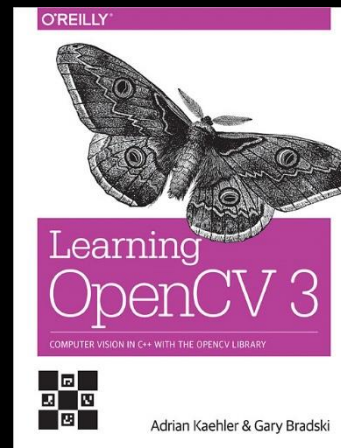
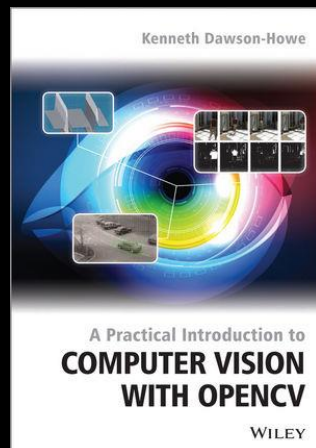
2. Rastreando

1. Busca Exaustiva
2. Deslocamento médio
3. Fluxo óptico denso
4. Fluxo óptico baseado em aspectos

6. Problemas de visão computacional

BIBLIOGRAFIA:

- A practical introduction to computer vision with OpenCV, Kenneth Dawson-Howe, Wiley 2014.
- Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library, Adrian Kaehler; Gary Bradski, O'Reilly 2016.
- <http://docs.opencv.org/3.1.0/>



O QUE É VISÃO COMPUTACIONAL?

- "É a análise automática de imagens e vídeos por computadores com o premissa de obter algum conhecimento do mundo."

Kenneth Dawson-Howe, 2014.

UM PROBLEMA DIFÍCIL

- "O primeiro problema que surgirá toda vez que alguém estudar este tópico é o de se convencer que este é um problema difícil"

Kenneth Dawson-Howe, 2014.

O QUE EU PRECISO PARA APRENDER VISÃO COMPUTACIONAL?

- Inglês
- Estatística
- Cálculo
- Álgebra Linear
- C++*
- Orientação a objetos*

COMO CRIAR UMA CLASSE EM C++

```
#ifndef CLASSE_HPP_  
#define CLASSE_HPP_  
#include <iostream>  
class image  
{      public:  
        int getAtributo();  
        void setAtributo(int);  
    private:  
        int atributo;  
  
};  
#endif /* CLASSE_HPP_ */
```

COMO CRIAR UMA CLASSE EM C++

```
#include "classe.hpp"
Int classe::getAtributo()
{
    return atributo;
}
void classe::setAtributo(int in)
{
    if(in > 0)
        atributo = in;
    else
        std::cout << "Entrada Invalida" << std::endl;
}
```

CONFIGURANDO O AMBIENTE LINUX PARA USAR A OPENCV

- https://docs.opencv.org/2.4/doc/tutorials/introduction/linux_install/linux_install.html
- https://docs.opencv.org/2.4/doc/tutorials/introduction/linux_eclipse/linux_eclipse.html

COMPARATIVO

Visão humana



Visão Computacional



VISÃO COMPUTACIONAL

Como é processada esta imagem:

Escala de cinza de 8 bits.

Resolução: 18x18.

67	67	66	68	66	67	64	65	65	63	63	69	61	64	63	66	61	60
69	68	63	68	65	62	65	61	50	26	32	65	61	67	64	65	66	63
72	71	70	87	67	60	28	21	17	18	13	15	20	59	61	65	66	64
75	73	76	78	67	26	20	19	16	18	16	13	18	21	50	61	69	70
74	75	78	74	39	31	31	30	46	37	69	66	64	43	18	63	69	60
73	75	77	64	41	20	18	22	63	92	99	88	78	73	39	40	59	65
74	75	71	42	19	12	14	28	79	102	107	96	87	79	57	29	68	66
75	75	66	43	12	11	16	62	87	84	84	108	83	84	59	39	70	66
76	74	49	42	37	10	34	78	90	99	68	94	97	51	40	69	72	65
76	63	40	57	123	88	60	83	95	88	80	71	67	69	32	67	73	73
78	50	32	33	90	121	66	86	100	116	87	85	80	74	71	56	58	48
80	40	33	16	63	107	57	86	103	113	113	104	94	86	77	48	47	45
88	41	35	10	15	94	67	96	98	91	86	105	81	77	71	35	45	47
87	51	35	15	15	17	51	92	104	101	72	74	87	100	27	31	44	46
86	42	47	11	13	16	71	76	89	95	116	91	67	87	12	25	43	51
96	67	20	12	17	17	86	89	90	101	96	89	62	13	11	19	40	51
99	88	19	15	15	18	32	107	99	86	95	92	26	13	13	16	49	52
99	77	16	14	14	16	35	115	111	109	91	79	17	16	13	46	48	51

A VISÃO DO SER HUMANO

- É muito complexa.
- A maior parte do tempo não sabemos como ela funciona.
- Tem como entrada duas lentes cujos sinais transmitem cerca de 50-60 quadros por segundo (Stereo Vision).

APLICAÇÕES PRÁTICAS DA VISÃO COMPUTACIONAL

- Verificar componentes e soldas de circuitos.
- Inspeccionar qualidade da impressão de rótulos.
- Inferir se garrafas estão corretamente preenchidas.
- Procurar por contusões em maçãs.
- Guiar robôs para na fabricação de produtos complexos como carros.
- Detecção de placas de transito automática (open ALPR).

BIBLIOTECA OPENCV

- Desenvolvida pela Intel em 2000.
- Multiplataforma e Livre.
- Licença BSD Intel.
- Contém vários módulos.
- Desenvolvida em C/C++, com suporte a Java, Python e Visual Basic.

COMO IMAGENS SÃO REPRESENTADAS NA OPENCV

- Geralmente representadas por uma instancia da classe `cv::Mat`.

Referência: http://docs.opencv.org/3.1.0/d3/d63/classcv_1_1Mat.html



MANIPULAÇÃO DE IMAGENS

Leitura, Exibição e Armazenamento

LEITURA DE IMAGENS

- É feita com o uso do método `Mat cv::imread(const String & filename, int flags = IMREAD_COLOR)`.

Referência: http://docs.opencv.org/3.1.0/d4/da8/group_imgcodecs.html

EXIBIÇÃO DE IMAGENS

- Para realizar esta tarefa precisaremos de quatro métodos `cv::imshow`, `cv::namedWindow`, `cv::waitKey` e `cv::destroyWindow`.

- `void cv::imshow(const String & winname, InputArray mat)`

Referência:

http://docs.opencv.org/3.2.0/d7/dfc/group_highgui.html#ga453d42fe4cb60e5723281a89973ee563

- `void cv::namedWindow (const String & winname, int flags = WINDOW_AUTOSIZE)`

Referência:

http://docs.opencv.org/3.2.0/d7/dfc/group_highgui.html#ga5afdf8410934fd099df85c75b2e0888b

- `int cv::waitKey(int delay = 0)`

Referência:

http://docs.opencv.org/3.2.0/d7/dfc/group_highgui.html#ga5628525ad33f52eab17feebcfba38bd7

- `void cv::destroyWindow(const String & winname)`

Referência:

http://docs.opencv.org/3.2.0/d7/dfc/group_highgui.html#ga851ccdd6961022d1d5b4c4f255dbab34

ARMAZENAMENTO DE IMAGENS

- É feita com o uso do método `bool cv::imwrite(const String & filename, InputArray img, const std::vector< int > & params = std::vector< int >())`

Referência:

http://docs.opencv.org/3.1.0/d4/da8/group_imgcodecs.html#gabbc7ef1aa2edfaa87772f1202d67e0ce

EXERCÍCIO 1

- Desenvolva uma classe em C++ chamada `images`. Crie dois arquivos `images.hpp` e `images.cpp`, esta classe deve conter métodos genéricos de leitura, exibição e escrita de imagens.