Vision 3D pour robots

Ludovic Hofer - Antoine Billy

L’objectif de ce projet est de concevoir un robot suiveur (c’est à dire un robot pouvant suivre une personne en respectant une certaine distance) de son algorithmique à son implémentation sur un système mobile à roues.

Infos pratiques

https://www.labri.fr/perso/lhofer/index.php?page=teaching/projets_technologiques

Premier semestre

Développer une application de stéréovision

- 6 TP (12 séances)
- Note finale: 0.6 TP + 0.4 Projet
- 5 meilleurs sur 6
- Rendus respectant strictement la deadline
- Avec tous les membres en copie
- C++, Qt et openCV

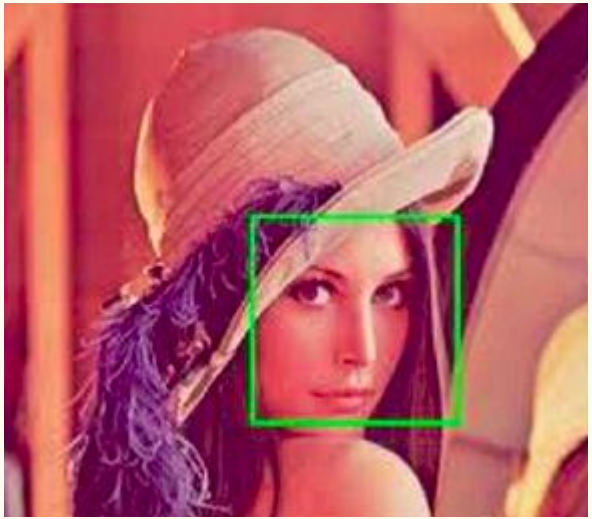
Introduction à openCV

Open Computer Vision

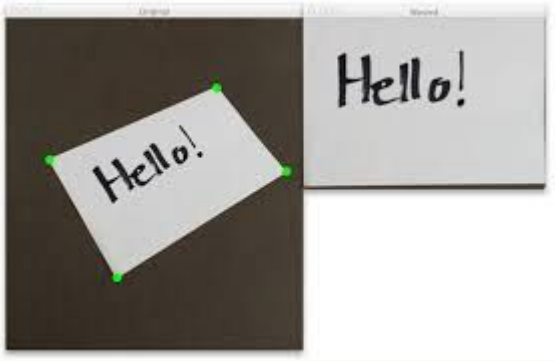
OpenCV est une bibliothèque de traitement d'images open sources originellement créée par Intel.

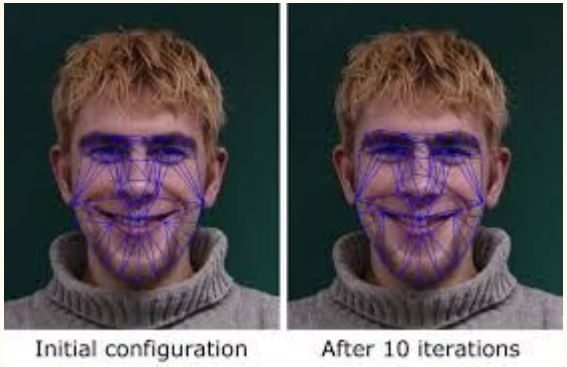
Disponible en C, C++ et python

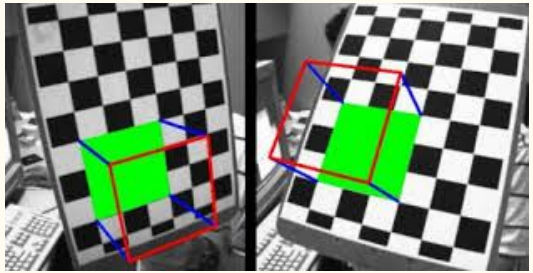
Dernière version 4.1.1

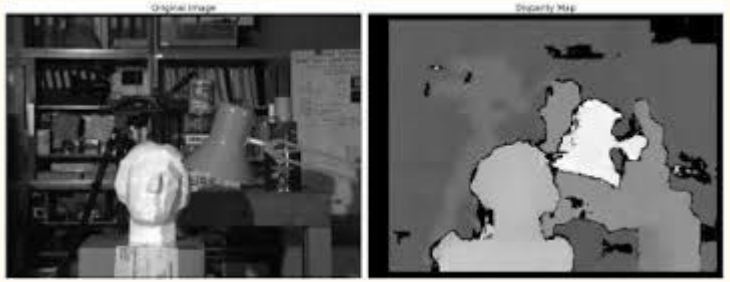


Exemples d'applications





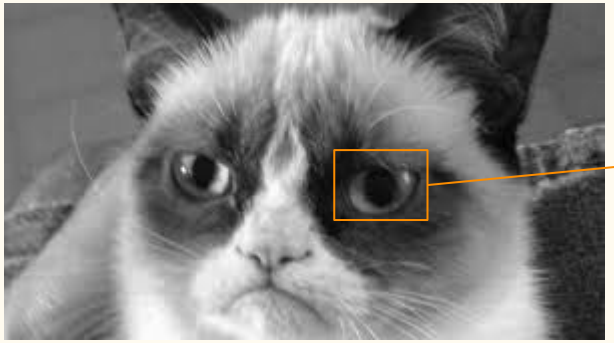




Structures de base

- **Point, Point2f** - 2D Point
- **Size** - 2D size structure
- **Rect** - 2D rectangle object
- **Mat** - image object

cv::Mat



216	201	133	134	65	221	164	177	191
202	201	155	102	66	67	66	187	52
188	97	111	80	53	22	58	86	133
107	108	128	211	203	170	94	57	142
169	160	177	97	86	154	63	134	187

cv::Mat

Public Attributes

MatAllocator *	**allocator** custom allocator More...
int	**cols**
uchar *	**data** pointer to the data More...
const **uchar** *	**dataend**
const **uchar** *	**datalimit**
const **uchar** *	**datastart** helper fields used in locateROI and adjustROI More...
int	**dims** the matrix dimensionality, >= 2 More...
int	**flags**
int	**rows** the number of rows and columns or (-1, -1) when the matrix has more than 2 dimensions More...
MatSize	**size**
MatStep	**step**
UMatData *	**u** interaction with **UMat** More...

cv::Mat

- Mat.**at**<datatype>(row, col)[channel] - returns pointer to image location
- Mat.**channels**() - returns the number of channels
- Mat.**clone**() - returns a deep copy of the image
- Mat.**create**(rows, cols, TYPE) - re-allocates new memory to matrix
- Mat.**cross**(<Mat>) - computes cross product of two matricies
- Mat.**depth**() - returns data type of matrix
- Mat.**dot**(<Mat>) - computes the dot product of two matrices

TP 1 - Manipuler des images avec openCV