



# Compte rendu : Reconstruction 3D

ANQUETIN Romain

20 décembre 2020

## Table des matières

Table des matières .....	0
Table des figures .....	1
Introduction.....	2
Calibrage de la caméra.....	3
Calibrage de la stéréovision .....	5
Géométrie épipolaire .....	6
Reconstruction 3D .....	9
Conclusion.....	10

## Table des figures

Figure 1 : Résultat 'True' avec le paterne affiché.....	3
Figure 2 : Matrices obtenues après la calibration .....	3
Figure 3 : Image rectifiée .....	4
Figure 4 : Matrice essentiel théorique.....	5
Figure 5 : Matrice essentiel pratique.....	5
Figure 6 : Matrice fondamentale théorique .....	5
Figure 7 : Matrice fondamentale pratique .....	5
Figure 8 : Fonctionnement de la géométrie épipolaire.....	6
Figure 9 : Résultat avec FT .....	6
Figure 10 : Différence de résultat entre théorique (en haut) et openCV (en bas) .....	7
Figure 11 : Résultat final avec les transformations sur les images .....	8
Figure 12 : Carte de profondeur .....	9

## Introduction

Dans ce TP, on va calibrer une caméra via des images d'un damier, on déterminera la matrice de la caméra et on analysera les résultats. Ensuite, on fera le calibrage de la stéréovision. On reconstruira les lignes épipolaires. Pour finir, on pourra calculer les informations de profondeurs de l'image. On utilisera la bibliothèque openCV en python que l'on a jamais manipulé.

Par manque de temps, la difficulté de prendre des images correctes et le temps de compréhension du programme on utilisera les images fournis avec le sujet.

## Calibrage de la caméra

La fonction `CameraCalibrate()`, cherche dans le damier des intersections entre les cases blanches et noires. On remarque que lorsqu'il ne trouve pas le paterne il renvoi la valeur 'false', et 'true' s'il trouve le paterne. Il affiche l'image avec des lignes reliant chaque point trouvé.



Figure 1 : Résultat 'True' avec le paterne affiché

La déformation de la figure 2d peut être dû à de la distorsion en coussinet. On observe l'image d'origine capté par le capteur du téléphone

```
cameraMatrix
[[1.41648092e+03 0.00000000e+00 6.67598677e+02]
 [0.00000000e+00 1.41360922e+03 9.16650426e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
dist
[[-0.00996315 0.66100357 0.00470359 -0.00439002 -1.82246322]]
newcameramtx
[[1.00423450e+03 0.00000000e+00 5.52874103e+02]
 [0.00000000e+00 7.90380249e+02 1.13082223e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

Figure 2 : Matrices obtenues après la calibration

La figure 2 montre les matrices qui représentent les paramètres intrinsèques et extrinsèques de la caméra. La matrice 'cameraMatrix' représente les paramètres intrinsèques et extrinsèques de la caméra calculer par openCV. La matrice 'newcameramtx' est une matrice calculer par la fonction 'getOptimalNewCameraMatrix()' de openCV.

Après avoir calculer les paramètres intrinsèques et extrinsèques de la caméra on peut calculer l'image rectifié, on obtient l'image suivante :



*Figure 3 : Image rectifiée*

On remarque sur la figure 3 des distorsions sur le bord de l'image, l'effet fish-eye n'est pas visible, par rapport à l'images de openCV. C'est une distorsion en coussinet.

## Calibrage de la stéréovision

Dans cette partie on va présenter le calibrage de la stéréovision.

La formule pour obtenir la matrice essentielle est  $E = [t]_x K R$ .

La formule pour obtenir la matrice fondamentale est  $F = K_1^{-T} * E * K_2^{-1}$

La matrice R est la matrice de rotation entre les 2 prises de vue.

La matrice T est la matrice de la translation entre les 2 prises de vue.

On a trouvé la matrice essentielle théorique suivante :

```
E théorique
[[-5.32928801e-09  3.63641196e-06 -4.45890434e-03]
 [-4.05475633e-06  4.39510720e-07 -9.17630235e-02]
 [ 4.87290515e-03  9.10557940e-02  1.00000000e+00]]
```

Figure 4 : Matrice essentiel théorique

On a trouvé la matrice essentielle pratique suivante :

```
E
[[-4.31423103e-07  8.60975694e-05 -7.46779304e-02]
 [-1.05478509e-04  1.46989758e-05 -7.03152327e-01]
 [ 8.93706285e-02  7.01436300e-01  1.02569069e-05]]
```

Figure 5 : Matrice essentiel pratique

On a trouvé la matrice fondamentale théorique est :

```
FT
[[-4.31423103e-07  8.60975694e-05 -7.46779304e-02]
 [-1.05478509e-04  1.46989758e-05 -7.03152327e-01]
 [ 8.93706285e-02  7.01436300e-01  1.02569069e-05]]
```

Figure 6 : Matrice fondamentale théorique

La matrice fondamentale pratique est :

```
F
[[-5.32928801e-09  3.63641196e-06 -4.45890434e-03]
 [-4.05475633e-06  4.39510720e-07 -9.17630235e-02]
 [ 4.87290515e-03  9.10557940e-02  1.00000000e+00]]
```

Figure 7 : Matrice fondamentale pratique

Les différences sont du aux méthodes de calculs, openCV utilise 2 points, tandis qu'on utilise des matrices. On que les matrice FT et E sont identique et que F et E théorique sont identiques aussi. On ne sait pas l'expliquer.

## Géométrie épipolaire

Dans cette partie on verra la notion de géométrie épipolaire.

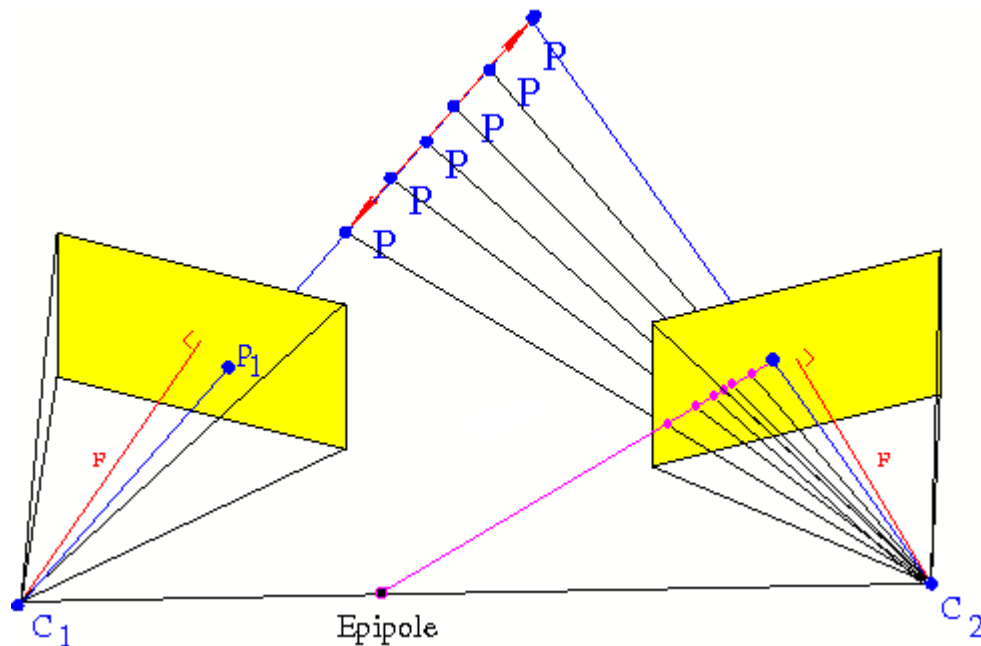


Figure 8 : Fonctionnement de la géométrie épipolaire

On observe que les résultats avec la matrice fondamentale calculée via la théorie correspondent à ce que l'on attend c'est-à-dire des droites qui convergent vers un point.

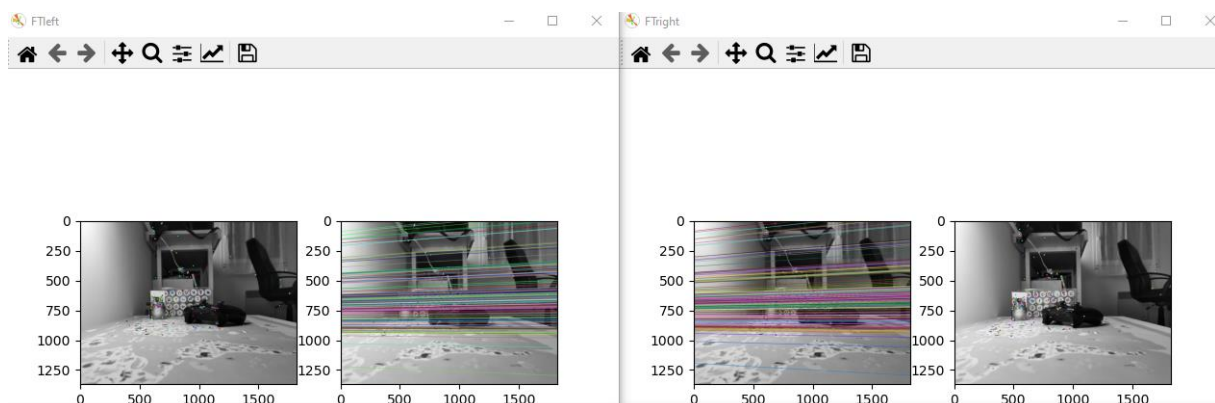


Figure 9 : Résultat avec FT

On peut aussi observer les différences de résultat entre les matrices  $FT$  et  $F$ . on observe clairement la différence là où les droites convergent avec  $FT$ , avec  $F$  les droites sont parallèles en elles.

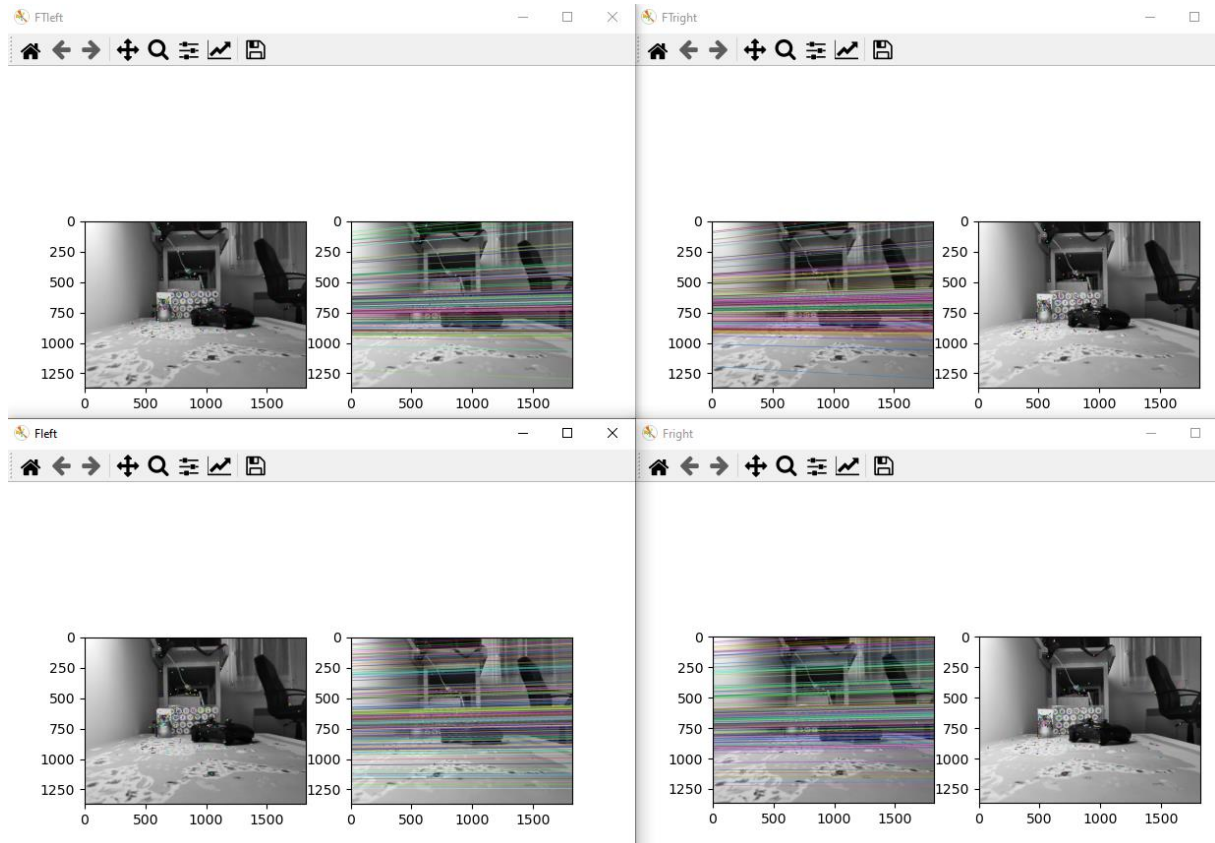


Figure 10 : Différence de résultat entre théorique (en haut) et openCV (en bas)



On observe le résultat final avec une rotation des images pour qu'elles se ressemblent le plus possibles. On observe que le résultat final n'est pas parfait, les lignes qui devraient être verticales se retrouvent inclinées. Il y a des étirements à certain endroit qui déforme la géométrie des objets, comme avec la manette.

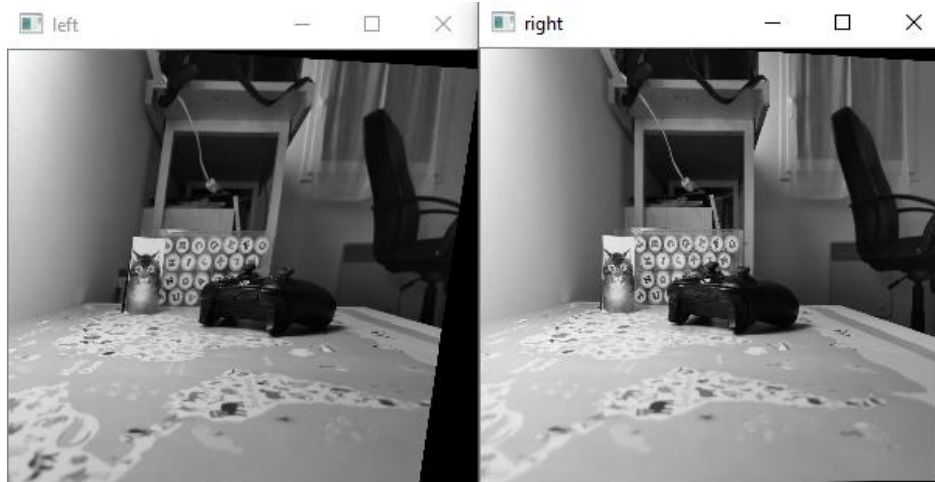


Figure 11 : Résultat final avec les transformations sur les images

L'étape suivante pour obtenir une carte 3D, est d'utiliser les résultats que l'on vient d'avoir pour calculer la profondeur dans l'image, car nous avons  $R$ ,  $t$ ,  $E$  et  $F$ .

## Reconstruction 3D

On obtient le résultat suivant :

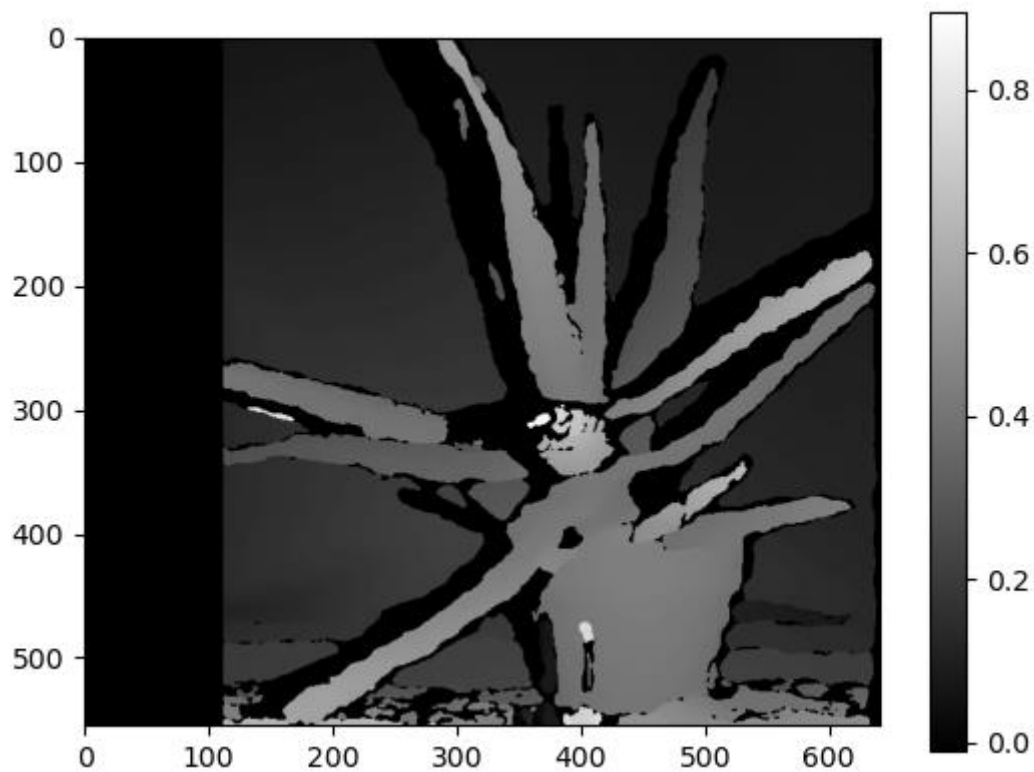


Figure 12 : Carte de profondeur

Sur la figure 10, on peut voir la carte de la profondeur. Cette carte représente la profondeur dans l'image grâce à l'intensité de la couleur. Ce qui est clair est proche, ce qui est loin est plus foncé. La précision de la carte n'est pas d'une précision incroyable.

## Conclusion

Dans ce TP, nous avons vu les étapes de calibrations d'une caméra. Nous avons vu la calibration de la stéréovision qui nous permet de faire plus d'analyse sur les images. Grace aux calibrations nous avons pu construire des lignes épipolaires entre deux images. Pour finir nous avons pu repérer l'information de la profondeur via deux images en stéréovision. Nous avons aussi manipulé la bibliothèque openCV pour python.