

Réalité augmentée

Par :

Koussaila Kadi
Mehdi Zenine
ISI Gr B

Motivation

Dans le cadre de ce projet, on souhaite mettre en œuvre une méthode de réalité augmentée sur une mire composée principalement d'un échiquier.

Le but étant une fois l'étude terminée de pouvoir situer n'importe quel point (qu'il soit virtuel ou non) présent dans la mire (et ayant des coordonnées connues dans le repère monde) sur n'importe quelle image représentant cette mire.

On prendra donc une série d'images (Avec la même caméra) et on cherchera à chaque fois à déterminer la matrice de projection de chacune d'entre elles.

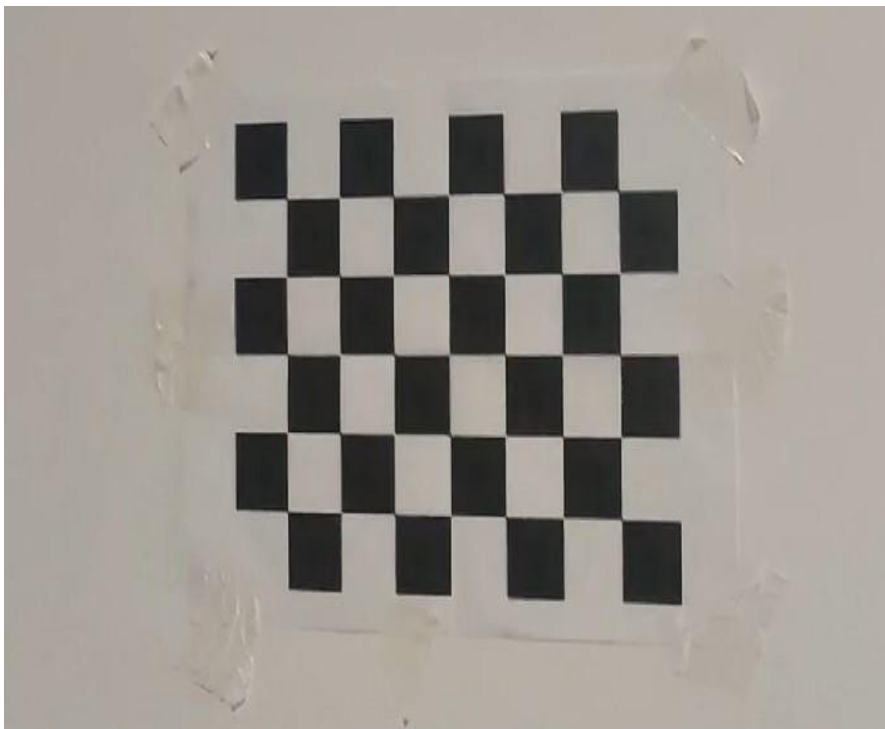


Image de la mire

Définition

Le concept de Réalité Augmentée est le dernier né de l'évolution galopante des Interfaces Homme-Machine (voir le numéro spécial de CACM, juillet 1993, vol. 36, no7).

C'est est une technique consistant à ajouter des éléments virtuels (texte, image, vidéo, animation, son) à notre environnement proche dans le but de l'enrichir et de le compléter. La réalité augmentée (RA) ne doit pas être confondue avec la réalité virtuelle (RV) d'un simulateur par exemple, qui plonge l'utilisateur dans un environnement virtuel en 3D[1].

Position du problème

D'après ce qui a été dit précédemment, faire de la réalité augmentée sur une mire, revient à prendre une série d'images de cette mire puis de trouver la matrice de projection pour chacune de ces images, il s'agit donc d'un problème de calibration.

Les points monde étant tous coplanaires, le calcul direct de la matrice de projection P par la méthode de la DLT n'est donc pas possible, on se propose de pallier de problème en déterminant la matrice P (pour chaque image) à partir de l'homographie H relie chaque plan image avec le plan de la mire.

Calcul de l'homographie

On se propose dans un premier tant de montrer la méthode de calcul de l'homographie pour une image donnée :

Soit un ensemble de points 3D $M(x\ y\ z)$ et leurs points images associés $m(u_i\ v_i)$.

L'homographie étant une relation qui relie 2 plans, on choisit les points 3D de la mire de sorte que ceux-ci soient coplanaires, on aura donc $z = 0$

Tout comme pour le calcul de la matrice de Projection P, on souhaite déterminer l'homographie H en utilisant une équation de la forme $Ah = 0$ ou h est un vecteur colonne contenant les éléments de H.

Soit un point 3D $M = (x, y, 0)$ et $m(u, v)$ sont point image associé, on a donc la relation :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

$$u = h_{11}x + h_{12}y + h_{13} \dots\dots\dots(1)$$

$$v = h_{21}x + h_{22}y + h_{23} \dots\dots\dots(2)$$

$$1 = h_{31}x + h_{32}y + h_{33} \dots\dots\dots(3)$$

En multipliant les u et v dans les 2 premières équations par l'expression de 1 dans (3) :

$$u(h_{31}x + h_{32}y + h_{33}) = h_{11}x + h_{12}y + h_{13}$$

$$v(h_{31}x + h_{32}y + h_{33}) = h_{21}x + h_{22}y + h_{23}$$

Finalement on obtient les 2 équations :

$$x h_{11} + y h_{12} + h_{13} - u x h_{31} - u y h_{32} - u h_{33} = 0$$

$$x h_{21} + y h_{22} + h_{23} - v x h_{31} - v y h_{32} - v h_{33} = 0$$

Qui peuvent être regroupées sous la forme matricielle :

$$\begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -ux & -uy & -u \\ 0 & 0 & 0 & x & y & 1 & -vx & -vy & -v \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ \vdots \\ h_{33} \end{pmatrix} = 0$$

Chaque couple de points (m, M) offre 2 équations linéairement indépendante. En considérant le facteur d'échelle, la matrice H aura 8 degrés de liberté, on utilisera donc 4 couples pour déterminer cette matrice.

Calcul de la matrice de Projection à partir de l'homographie

Ce paragraphe fait suite au précédent, on dispose d'un ensemble d'images d'une même mire, et on souhaite retrouver les matrices de projection de chacune de ces images à partir de la connaissance de l'homographie et de la matrice intrinsèque K

On illustrera les calculs pour 1 des images :

D'après le paragraphe ... on a les relations suivantes :

$$m = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = K \begin{pmatrix} r_1 & r_2 & r_3 & T \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = H \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}.$$

On aura donc, grâce à l'offset $z=0$:

$$\alpha K \begin{pmatrix} r_1 & r_2 & T \end{pmatrix} = H \text{ ou } \alpha : \text{Représente le facteur d'échelle}$$

Calcul du facteur d'échelle :

La relation précédente permet d'écrire : $\alpha \begin{pmatrix} r_1 & r_2 & T \end{pmatrix} = K^{-1} H$

On peut donc en déduire les vecteurs : αr_1 et αr_2 qui sont les 2 premiers vecteurs colonnes de $K^{-1} H$

On utilise à présent la Matrice :

$$B = \begin{pmatrix} \alpha r_1 & \alpha r_2 & \alpha r_1 \times \alpha r_2 \end{pmatrix} = \begin{pmatrix} \alpha r_1 & \alpha r_2 & \alpha^2 (r_1 \times r_2) \end{pmatrix}$$

$$\text{On a : } \det(B) = \alpha^4 \det(r_1 \ r_2 \ r_3) = \alpha^4$$

Car la matrice de rotation par définition à un déterminant égal à 1

On obtient donc 2 valeurs possibles pour α .

Ainsi on divise la matrice $K^{-1} H$ par α et on obtient : $(r_1 \ r_2 \ T)$

On en déduit donc la matrice de projection P :

$$K \begin{pmatrix} r_1 & r_2 & r_3 \end{pmatrix}$$

On teste les P pour les 2 valeurs de α afin de s'assurer de trouver le bon résultat

Extraction des points images

On a montré dans le paragraphe précédent comment retrouver la matrice de projection uniquement à partir de la connaissance des matrices H et K . Néanmoins, le calcul de la matrice H suppose que l'on est déjà accès aux points images associés à chaque point 3D et ce dans chaque image, ce qui n'est pas toujours évident.

On se propose dans le cadre de ce projet d'utiliser des outils de traitements d'images afin d'extraire les points images dans chacune des images.

La mire étant un échiquier, on souhaite trouver les coordonnées de certains points de cet échiquier (des coins de préférence). On utilisera pour cela l'algorithme de « **Harris corner detector** »

Détection par la méthode de Harris :

Harris Corner Detector est un opérateur de détection d'angle couramment utilisé dans les algorithmes de vision par ordinateur pour extraire les coins et déduire les caractéristiques d'une image[2].

Elle se base sur une fonction d'autocorrélation du signal c'est-à-dire sur les changements du signal dans plusieurs directions.

La méthode permet ainsi non seulement de détecter les coins présents dans l'image mais également de fournir les pixels associés à chaque coin.

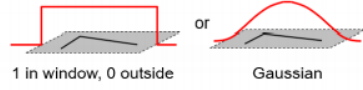
La détection se fait suivant la formule suivante :

Harris Detector formulation

Change of intensity for the shift $[u, v]$:

$$E(u, v) = \sum_{x, y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$$

Window function Shifted intensity Intensity

Window function $w(x, y) =$  1 in window, 0 outside or Gaussian

Source: R. Szeliski

Harris Detector formulation

This measure of change can be approximated by:

$$E(u, v) \approx [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

where M is a 2×2 matrix computed from image derivatives:

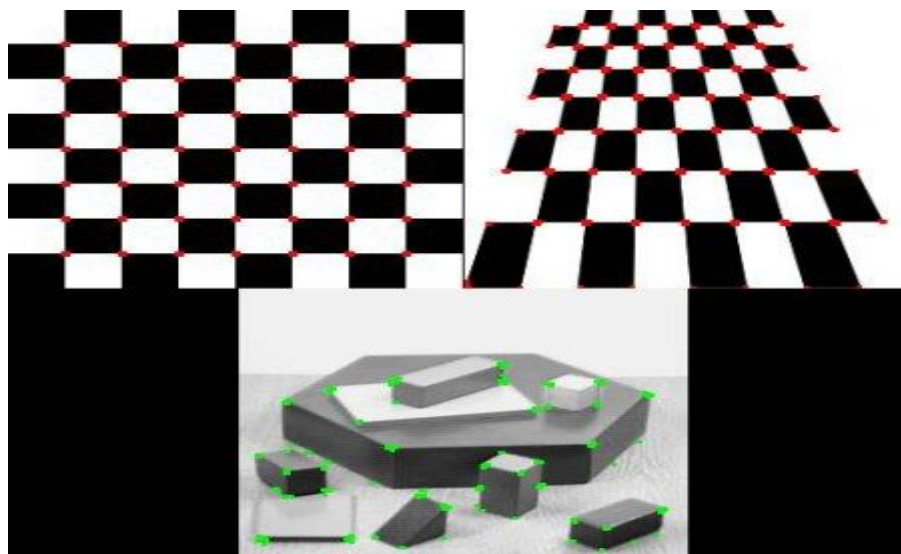
$$M = \sum_{x, y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Sum over image region – area we are checking for corner

Gradient with respect to x , times gradient with respect to y

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix}$$

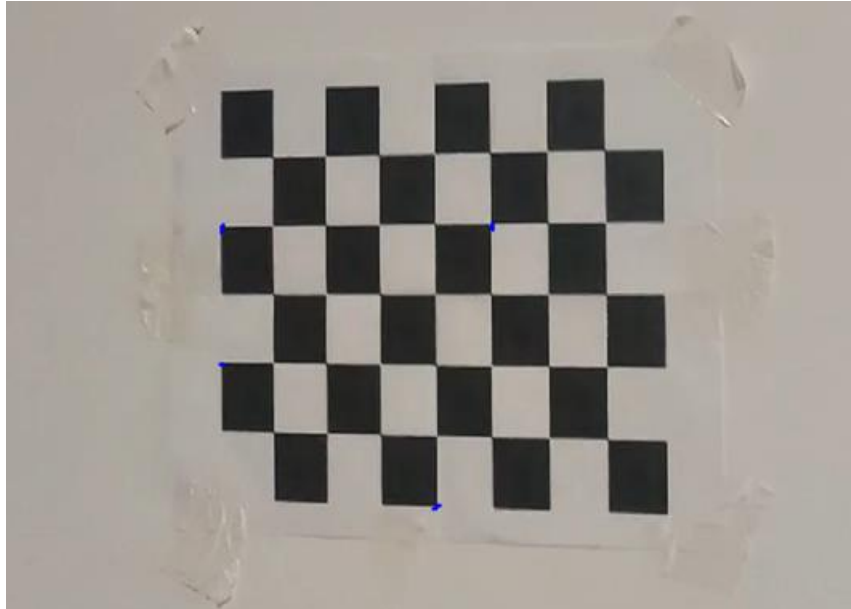
Formule de détection avec la méthode de Harris



Exemple d'utilisation de Harris Corner Detector sur OpenCV

Implémentation

On se propose d'illustrer cette méthode sur 1 seule image de la mire, on souhaite déterminer la matrice de projection associée à l'image suivante :



Les points bleus représentés sur la figure représentent les 4 points 3D utilisées pour le calcul de l'homographie.

Le repère monde est choisit de telle sorte que les coordonnées des 4 points images associés soient les suivantes :

$$M1 = \begin{pmatrix} 0 \\ 6 \\ 0 \end{pmatrix} \quad M2 = \begin{pmatrix} 0 \\ 12 \\ 0 \end{pmatrix} \quad M3 = \begin{pmatrix} 15 \\ 6 \\ 1 \end{pmatrix} \quad M4 = \begin{pmatrix} 12 \\ 18 \\ 1 \end{pmatrix}.$$

Les coordonnées sont des multiples de 3 car chaque segment de l'échiquier fait 3 cm

Notre implémentation contient :

- Un programme Python « Harris » qui fournit les pixels associés aux points monde
- Un programme Python qui à partir d'un ensemble d'images fournit la matrice intrinsèque de notre caméra
- Un programme MATLAB « homographie.m » qui calcule la matrice H
- Un programme MATLAB « Programme_Principal.m » qui calcule la matrice de projection P à partir des matrice H et K calculées précédemment

Résultats obtenus

En vérifiant le résultat obtenu (ndlr : La matrice de projection P de l'image) on ne retrouve pas les points images initialement associés mais des points images ayant des pixels « Largement » différents et ce pour les 2 valeurs de α . Ce mauvais résultat peut s'expliquer de l'une des façons suivantes :

- Erreurs sur le calcul de la matrice intrinsèque K
- La valeur de α est importante et donc le terme $1/\alpha$ introduit une « instabilité »
- Erreur d'offset pour la méthode Harris (la méthode détecte un point légèrement éloigné du coin)

Conclusions

- La méthode utilisée fonctionne uniquement avec une mire possédant des coins, dans le cas contraire le calcul des pixels devra se faire avec une autre méthode
- Le fait d'avoir des points 3D coplanaires fait que l'on ne peut pas calibrer directement notre caméra à partir des couples (m, M) , on est donc obligés de passer par la matrice d'homographie
- La difficulté principale de cette méthode réside dans l'extraction des points images
- La méthode utilisée ne prend pas en compte les possibles erreurs d'offset liées à l'extraction des images, pour une application concrète elle doit être améliorée
- Les erreurs de troncatures liées au calcul de la matrice K peuvent conduire à un résultat complètement erroné

Références bibliographiques

[1] https://cache.media.eduscol.education.fr/file/Numerique/45/0/EIM_REALITE_AUGMENTEE_628450.pdf

[2] https://ensiwiki.ensimag.fr/images/d/d3/Harris_corner.pdf

[3] R. Szeliski, " Computer Vision : Algorithms and Applications ", 2010.