Projet « Calcul Numérique » pour la 2ième année EMA.

Comparaison de méthodes d'appariement en vision par ordinateur par utilisation de stéréogrammes de points aléatoires

Introduction : En 1960, Julez[1] donne la preuve avec les R.D.S (Random Dot Stereogram) qu'un processus de vision bas niveaux permet de créer une carte de disparité (liée à la profondeur 3D) sans reconnaissance des objets de la scène. Depuis, il est admis que la détection d'aires visuelles identiques, dans les images issues de chaque rétine, s'effectue par interaction neuronale de bas niveau. Ce processus fournit la valeur de la disparité pour chaque zone visuelle.

En vision par ordinateur, et notamment en vision stéréoscopique (utilisation de deux caméras), une étape essentielle est d'obtenir, à partir des deux images gauche IG et droite ID, ce qui permettra le calcul de la profondeur de la scène : c'est l'étape d'appariement, c'est à dire faire correspondre un point de ID avec un point de IG. En sortie de l'étape d'appariement il y a une carte (matrice) des valeurs de *disparité.*
L'utilisation des RDS est un moyen simple permettant la comparaison de méthodes d'appariements basées sur les niveaux de gris (quantification par le capteur de la luminosité en 256 niveaux).

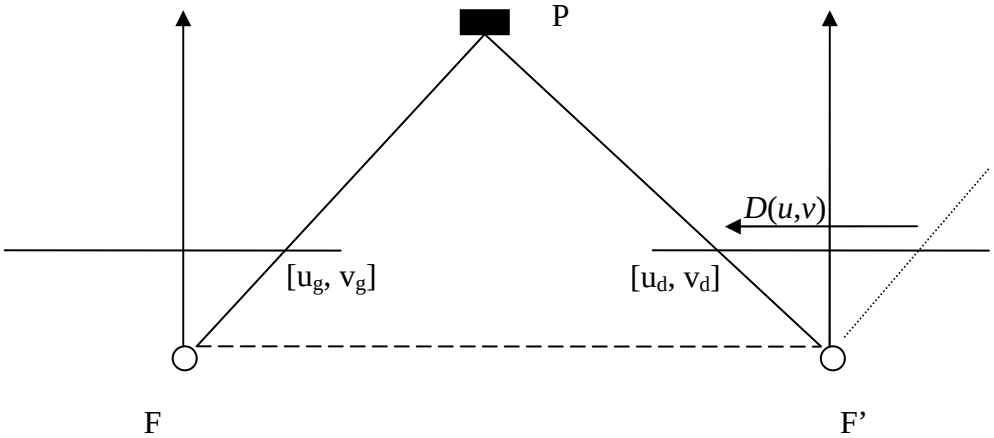


Figure 1: modèle mathématique d'un couple de caméra en géométrie rectifiée vue du dessus. F et F' leur centre de projection, P un point de l'espace 3D. V, V' les abscisses du projeté de P sur les plans images IG et ID.

La disparité est **l'écart entre les coordonnées** des projections d'un point 3D dans chacune des images (droite et gauche) et dans le cas de la **disposition géométrique des caméras utilisé** : cette disparité est simplement **l'écart entre les abscisses des projections** (figure 1)

$$D = D([u_g, v_g],[u_d, v_d]) = \left| v_d - v_g \right| \qquad \text{(eq.1)}$$

[1] Julesz,B., " Binocular Depth Perception of computer-generated patterns ", Bell Systems Techn. J 39, pp.1125-1162, 1960.

Dit autrement : connaissant la valeur du « décalage » *D* associé au point [u_g, v_g] de l'image de gauche nous pouvons retrouver son homologue [u_d, v_d] dans l'image de droite.

Le projet :

Nous proposons ici de tester et de comparer deux méthodes d'appariement sur les images artificielles que sont les RDS. Les méthodes étudiées dans ce projet serviront à retrouver une matrice de valeurs liées à la profondeur (carte des disparités) qui a servi à créer les matrices de points aléatoires. Les **méthodes** seront mis en œuvre sur des images RDS **fournies (**il n'est donc pas demandé de créer un tel couple mais de retrouver la carte des disparités originale**)** puis les **résultats** (erreur de reconstruction de la carte des disparités, temps de calcul) seront **comparés**.

Un RDS est un **couple de matrices aléatoires de points blanc et noir**. Si l'image de droite est initialisée aléatoirement, l'image de gauche est construite en utilisant la carte des disparités *D* en respectant l'équation suivante :

$$IG(u,v) = ID(u, v - D(u,v))$$

Ici, (*u*, *v*) sont les coordonnées d'un point dans l'image de gauche et *D*(*u*, *v*) la disparité associée.

Données du projet :

Sont fourni :

1. Pour la mise en œuvre :
   Un jeu d'images (gauche et droite) RDS ainsi que la carte de disparité qui les a générées. Ce premier jeu sera utilisé pour mettre en place et tester le programme en comparant les résultats obtenus par chacune des méthodes.

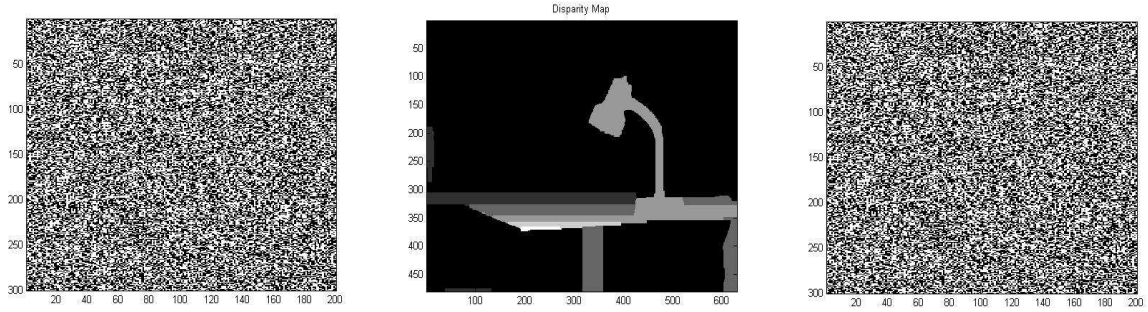


Figure 2 : couple RDS fourni. Au centre la carte des disparités (liée à la profondeur). A gauche et a droite respectivement les images du couple RDS généré par le biais de la carte des disparités qu'il faudra estimer par les méthodes proposées**.**

2. Pour l'utilisation :
   Un deuxième jeu d'images RDS est fourni pour lesquelles il ne sera demandé que de retrouver la carte des disparités qui les a générées.

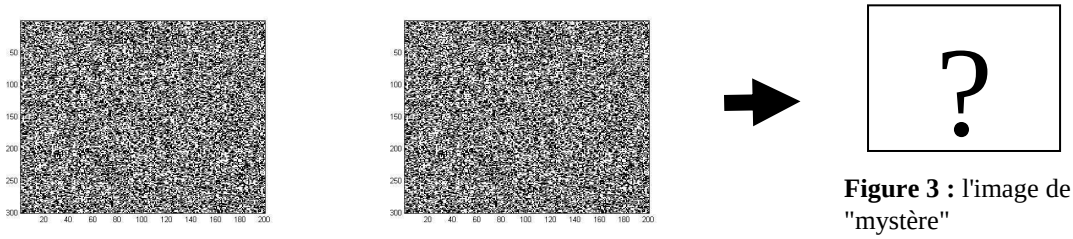
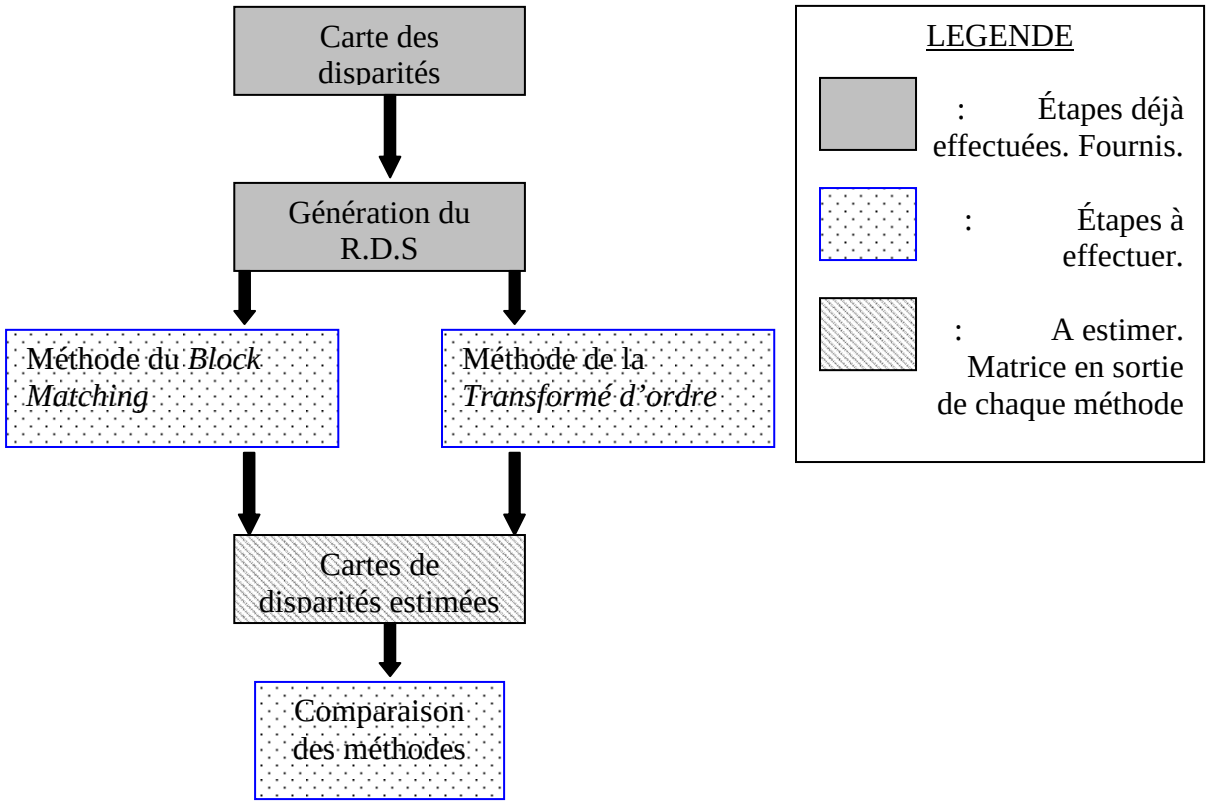


Figure 3 : l'image de disparité "mystère"

Étapes de l'expérimentation et construction d'un couple RDS :



Méthodes:

Première méthode : le ***Block matching***

Il s'agit de la recherche d'un maximum de ressemblance entre le voisinage (carré encadrant le pixel) du pixel de départ (IG) et celui d'arrivée (ID). Ce maximum est obtenu pour une disparité *D* minimisant la « distance » ou la « ressemblance » entre les blocs. Les deux méthodes fournies différent dans la définition de cette ressemblance. En effet, il s'agit de parcourir la matrice IG et pour chaque point (u_g, v_g), trouver le point homologue associé dans ID. L'écart de position de ces points est la valeur de disparité associée au point de l'image de gauche que nous recherchons.

Recherche du point homologue : à partir d'une position initiale (dans l'image de droite) (u_d, v_d) = (u_g, v_g) (disparité nulle), on parcourt les points de la ligne de droite à gauche en calculant un critère de ressemblance pour chaque position candidate pour le point homologue. Le décalage minimum (simple valeur absolue des écarts des abscisses v_g et v_d) par rapport à la position initiale qui minimise le critère de ressemblance est la **disparité**.

Critère de ressemblance :

C'est la somme des éléments d'une matrice 3x3 dont chaque élément est le carré de la différence terme à terme entre les valeurs de ID contenue dans un carré 3x3 centré en (*u*, *v*-*k*) où *k* est une valeur de disparité (décalage) testée, et le carré 3x3 centré sur le point (*u*, *v*) de IG.

Pour **limiter le temps de calcul**, nous fixons une disparité maximale **Dm** ce qui implique de ne pas vérifier certains appariements. Dans **l'intervalle k** $\in$ **[0, Dm]** nous devrons donc trouver le point homologue (plus Dm est petit plus le risque de faux appariement est élevé).

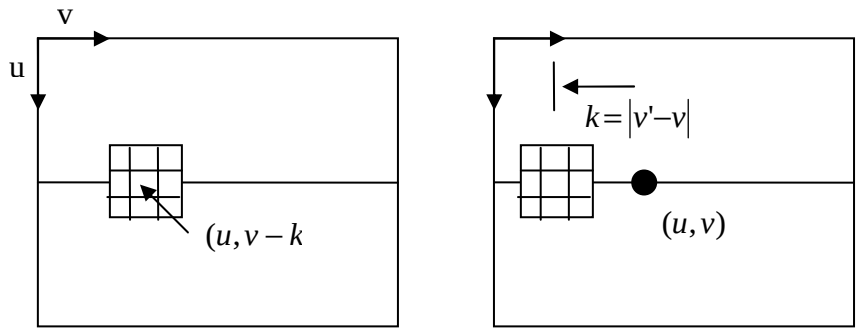


Figure 4 : illustration de la méthode du bloc matching. Soient (u, v) dans ID les coordonnées du point à apparier. On parcourt la ligne de gauche à droite dans l'autre image avec un décalage k. La valeur de k qui minimise le critère de ressemblance est la valeur de la disparité en (u,v).

Exceptions : comme on utilise un cadre 3x3 centré sur le pixel en cours, on évite les effets de bords en n'effectuant les calcules que lorsque le cadre est défini (soit coordonnée **bord +/- 1**). De plus, si pour la position courante du point (u, v) ont a v – Dm < 0 alors on ne fait pas les calculs (le point n'est peut être pas visible dans l'autre caméra, forte probabilité d'un faux appariement) et on pose dans ce cas comme dans le premier $D(u, v) = 0$.

Info : on initialise une matrice résultat avec ***zeros*** (MATLAB) de la même taille que les images. Dans cette matrice seront rangées les valeurs de disparités. Grâce à ***zeros*** les points hors du domaine de travail (les bords) sont directement initialisés à zéro. Vous aurez besoin des fonctions matlab ***tic/toc*** pour le temps de calculs (mesure entre le début et la fin du remplissage de la matrice).

Note : ces considérations facilitent les calculs et les images artificielles créées pour l'exercice sont conçues pour fonctionner avec ces contraintes. Connaissant ainsi les bornes du domaine d'étude (le reste ayant directement un résultat nul) il faut adapter les boucles de calculs.

Valeurs de Dm testées : de 4 à 16 (ainsi, k variera de 0 à 4 pour Dm = 4, de 0 à 5 pour Dm = 5, …).

En sortie: carte des disparités et temps de calcul pour chaque valeur de Dm.

Deuxième méthode : la Transformée d'ordre

Dans cette méthode, une fois les pixels de chacune des images étiquetés, les points homologues seront recherchés par une méthode de corrélation dense. Nous allons donc déterminer une signature du pixel de gauche et la comparer aux signatures de pixels candidat à l'appariement comme pour la méthode précédente mais nous n'utilisons pas directement les valeurs en niveaux de gris des pixels voisin du point considéré.

L' « étiquette » T (ou encore la signature du pixel à comparer) est définie comme suit :

L'étiquette est une analogie biomimétique de la notion de voisinage. Un pixel est alors représenté par **son voisinage en niveau de gris** (3x3 par exemple) et plus exactement par sa position une fois les valeurs de luminance de chaque pixel arrangées par ordre décroissant. Cela ce traduit par l'expression suivante (eq. 2) :

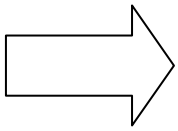
Ordre : la transformation qui renvoie le rang du pixel $p(i, j)$ si les niveaux de gris des pixels voisins ont été rangés par ordre décroissant.

$$T(p(i, j)) = (Ordre(P(i + k, j + l))), \qquad \text{(eq. 2)}$$

(k, l) parcourant le voisinage du pixel.

Valeurs du voisinage 3x3 :

250	190	110
170	150	80
70	50	0



Ordres :

1	2	5
3	4	6
7	8	9

Equivalent à :

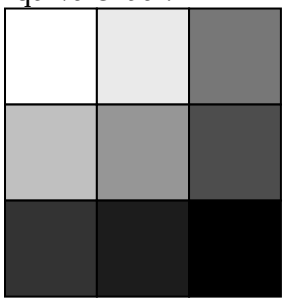


Figure 5: Exemple d'un voisinage 3x3. L'ordre du pixel (central), donc la valeur de l'étiquette, est $T(p(i, j)) = 4$.

Les matrices images sont parcourues de la même manière que dans la méthode « Block Matching ».

La mesure de ressemblance est la suivante :
Si le pixel candidat (le premier rencontré !) de ID à une étiquette identique à celle du pixel à apparier (u, v) de IG alors c'est le point ressemblant et la valeur de décalage est inscrite dans la matrice de disparité.

Comme pour la première méthode on fournira en sortie une matrice de disparité.

Info : Fonctions Matlab éventuellement nécessaires : ***return***, ***sort***. Les limites du domaine de travail sont les mêmes que pour la méthode 1.

Procédure :

Créer un répertoire de travail dans lequel seront situées toutes les fonctions créées ainsi que l'image de disparité initiale.

Créer un script principal « main.m » à partir duquel on exécutera toutes les fonctions créées.

%INITIALISATION :

- récupérer les images des disparités et RDS fournies (fonction ***imread***)
- conversion des valeurs en « double » (permet ensuite de faire des calculs),

% CALCUL
% Implémentation des méthodes « Block Matching et « Transformée d'Ordre »

% AFFICHAGE
L'erreur de reconstruction de la carte des disparités en fonction de Dm doit être exprimé en % (nombre pixels différents / nombre total de pixels). L'affichage d'une matrice A sous forme d'image est réalisé par la fonction ***imagesc***(A) ; **colormap gray** (pour l'affichage en niveaux de gris)

Nous aurons deux « figure » :
Dans « figure 1 » devra apparaître (help ***subplot***) l'image des disparités originale et celles reconstruites par chacune des méthodes.

Dans « figure 2 », nous aurons deux graphiques sur une même ligne :

- Graphique 1 : tracer en fonction de Dm, le temps de calculs des deux méthodes (en ms)

- Graphique 2 : tracer sous forme d'un diagramme en barres l'erreur de chaque méthode en fonction de Dm.

Chaque graphique devra contenir une légende. Les commentaires des fonctions doivent être clairs.
Ce document et tous les fichiers des scripts et fonctions doivent être compactés dans un fichier :

Nom1_Nom2_Nom3.zip

Nom1_Nom2_Nom3 sont les noms du groupe ayant réalisé le projet.

Le tout devra être envoyé par mail avant **9/01/2009, 12h aux enseignants respectifs :**

stefan.janaqi@ema.fr
pierre.Couturier@ema.fr
laurent.pissot@laposte.fr
guillaume.Tatur@ema.fr
Max.Nemoz-Gaillard@ema.fr
Ouael.Mouelhi@ema.fr