

# Traitement d'image - techniques d'appariements

## Ecole Doctorale MIIS - 2017

Sebastien Kramm

LITIS Rouen

28 juin 2017

Avec des diapos de Christian Unger, TUM



# Sommaire

- 1 Introduction
- 2 Exemple de problématiques à considérer
- 3 Appariement dense
- 4 Appariement épars



# Situations

- Le problème de l'appariement (*matching*) est une problématique permanente en CV, dans de nombreuses situations :
  - Calibrage d'un système stéréo : estimation de la géométrie épipolaire
  - Stereovision : construction d'une modélisation de l'environnement ("reconstruction 3D")
  - *Image registration* : recherche d'un objet dans une base d'images
  - Stitching (photos panoramiques)
  - Motion tracking
  - Indexing and content-based retrieval
  - Robot navigation, visual odometry
  - ...

VIDEO : AYMERIC JARDIN + NVIDIA



# Classification des méthodes

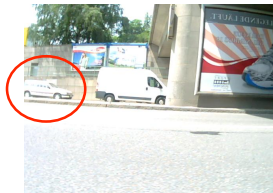
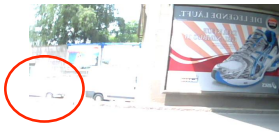
- Deux approches
  - matching dense : production de correspondance pour (presque) chaque pixel de l'image
  - matching épars : production de correspondance pour un ensemble de points donnés
- Solutions :
  - matching dense : techniques corrélatives, coupure de graphes, ...
  - matching épars : utilisation d'une caractérisation de l'image associée aux points 2D

# Sommaire

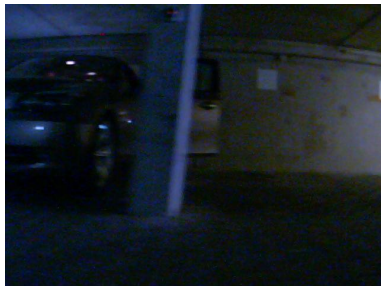
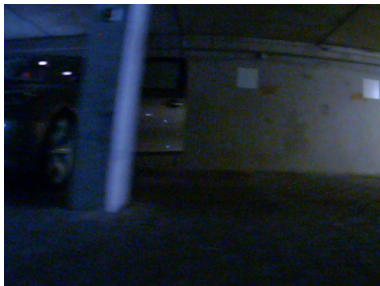
- 1 Introduction
- 2 Exemple de problématiques à considérer
- 3 Appariement dense
- 4 Appariement épars



## Challenges: Photometric Variations.



## Challenges: Image Sensor Noise.



## Challenges: Specularities.



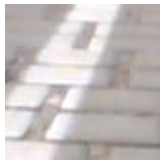


## Challenges: Foreshortening and the Uniqueness Constraint.

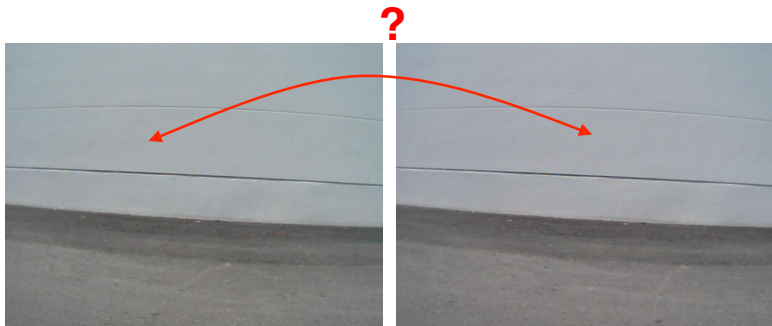


?

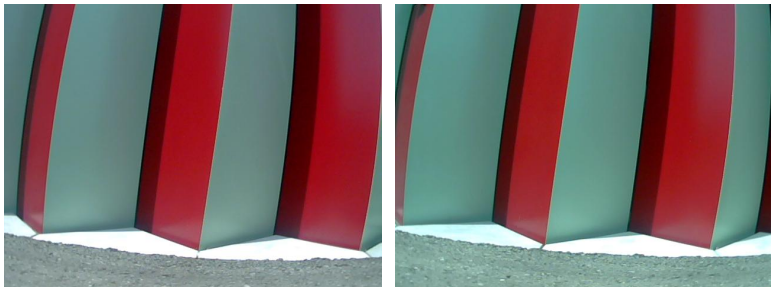
## Challenges: Perspective Distortions.



## Challenges: Textureless Regions.



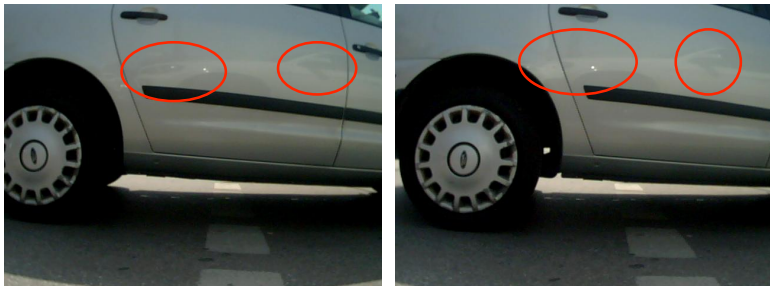
## Challenges: Repetitive Structures and Textures.



## Challenges: Reflections.



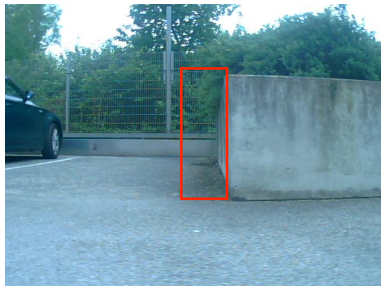
## Challenges: Reflections.



## Challenges: Transparency.



## Challenges: Occlusions.





## Challenges: Occlusions.



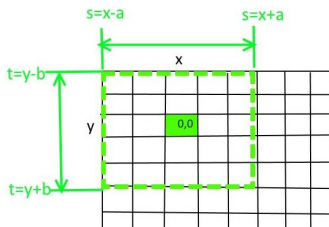
# Sommaire

- 1 Introduction
- 2 Exemple de problématiques à considérer
- 3 Appariement dense**
- 4 Appariement épars



## Techniques corrélatives

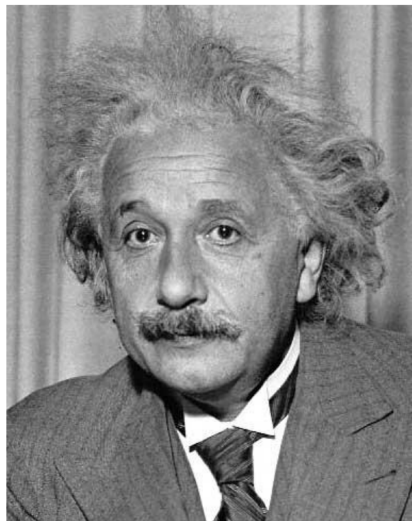
Principe : on recherche pour chaque pixel de  $I_1$  le pixel de  $I_2$  présentant la valeur minimale (ou maximale) d'un critère en considérant une fenêtre de taille fixe autour de  $p$ .



- On fait glisser une fenêtre contenant une "image" de  $I_2$  sur différentes positions de  $I_1$
- La fenêtre de corrélation peut être circulaire, rectangulaire, mais le plus souvent sera carrée

## Variante (mais même principe) : *Template Matching*

- Consiste à retrouver dans une image un élément d'une autre
- Exemple : retrouver dans l'image ceci :



## Techniques corrélatives

Deux types de critères de corrélation :

- ceux basés sur la distance euclidienne entre les deux vecteurs formés par les intensités des images  
SSD : *Sum of Squared Differences* ou SAD : *Sum of Absolute Differences*
- ceux basés sur le produit scalaire des ces deux vecteurs (*Cross Correlation*)

$$c_{SSD} = \sum_{i=-m}^m \sum_{j=-n}^n [(I_1(x_1 + i, y_1 + j) - (I_2(x_2 + i, y_2 + j))]^2$$

$$c_{SAD} = \sum_{i=-m}^m \sum_{j=-n}^n |(I_1(x_1 + i, y_1 + j) - (I_2(x_2 + i, y_2 + j)|$$

$$c_{CC} = \sum_{i=-m}^m \sum_{j=-n}^n [(I_1(x_1 + i, y_1 + j) (I_2(x_2 + i, y_2 + j))]$$

## Techniques corrélatives

Deux types de critères de corrélation :

- ceux basés sur la distance euclidienne entre les deux vecteurs formés par les intensités des images  
SSD : *Sum of Squared Differences* ou SAD : *Sum of Absolute Differences*
- ceux basés sur le produit scalaire des ces deux vecteurs (*Cross Correlation*)

$$c_{SSD} = \sum_{i=-m}^m \sum_{j=-n}^n [(I_1(x_1 + i, y_1 + j) - (I_2(x_2 + i, y_2 + j))]^2$$

$$c_{SAD} = \sum_{i=-m}^m \sum_{j=-n}^n |(I_1(x_1 + i, y_1 + j) - (I_2(x_2 + i, y_2 + j)|$$

$$c_{CC} = \sum_{i=-m}^m \sum_{j=-n}^n [(I_1(x_1 + i, y_1 + j) (I_2(x_2 + i, y_2 + j))]$$

Notation allégée :

$$c_{SSD} = \sum (I_1 - I_2)^2 \quad c_{SAD} = \sum |I_1 - I_2| \quad c_{CC} = \sum (I_1 \cdot I_2)$$

## Versions améliorées

- ZSSD (*Zero-mean Sum of Squared Differences*) : La luminance des images peut-être différente, on peut normaliser par la moyenne locale des niveaux dans la fenêtre.

$$c_{ZSSD} = \sum [(I_1 - \bar{I}_1) - (I_2 - \bar{I}_2)]^2$$

- ZNSSD (... *Normalized ...*) : on pondère par la moyenne et on normalise par la variance locale des intensités.  
→ plage de variation :  $[0, 1]$  (=0 pour des images identiques)


$$c_{ZNSSD} = \frac{\sum [(I_1 - \bar{I}_1) - (I_2 - \bar{I}_2)]^2}{\sqrt{\sum (I_1 - \bar{I}_1)^2 (I_2 - \bar{I}_2)^2}}$$

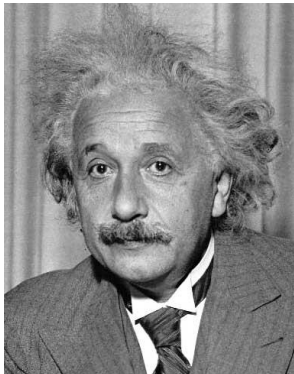
- ZNCC (*Zero-mean Normalized Cross Correlation*)  
→ plage de variation :  $[-1, +1]$  (=1 pour des images identiques)

$$c_{ZNCC} = \frac{\sum [(I_1 - \bar{I}_1) (I_2 - \bar{I}_2)]}{\sqrt{\sum (I_1 - \bar{I}_1)^2 (I_2 - \bar{I}_2)^2}}$$



# Matching with filters

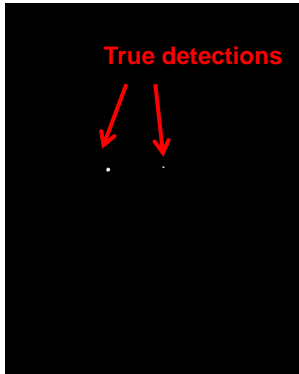
- Goal: find  in image
- Method 3: Normalized cross-correlation



Input




Normalized X-Correlation

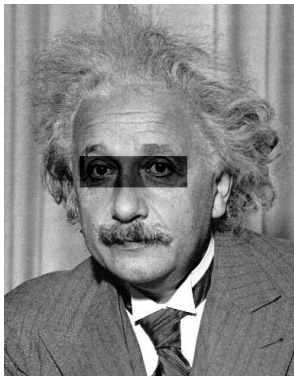


Thresholded Image



# Matching with filters

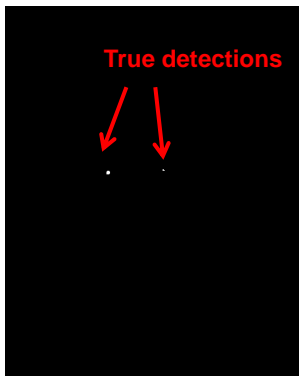
- Goal: find  in image
- Method 3: Normalized cross-correlation



Input



Normalized X-Correlation



Thresholded Image

## Choix & fiabilité

### Quel critère choisir ?

- En situation de type "industriel" (contrôle qualité) : l'environnement et l'éclairage est maîtrisé, les différences de performances seront minimales

## Choix & fiabilité

### Quel critère choisir ?

- En situation de type "industriel" (contrôle qualité) : l'environnement et l'éclairage est maîtrisé, les différences de performances seront minimales
- Dans les autres cas, il sera indispensable de choisir un critère plus élaboré (et plus coûteux)

## Choix & fiabilité

### Quel critère choisir ?

- En situation de type "industriel" (contrôle qualité) : l'environnement et l'éclairage est maîtrisé, les différences de performances seront minimales
- Dans les autres cas, il sera indispensable de choisir un critère plus élaboré (et plus coûteux)
- D'une façon générale : pas de vainqueur clair

## Choix & fiabilité

### Quel critère choisir ?

- En situation de type "industriel" (contrôle qualité) : l'environnement et l'éclairage est maîtrisé, les différences de performances seront minimales
- Dans les autres cas, il sera indispensable de choisir un critère plus élaboré (et plus coûteux)
- D'une façon générale : pas de vainqueur clair



## Choix & fiabilité

### Quel critère choisir ?

- En situation de type "industriel" (contrôle qualité) : l'environnement et l'éclairage est maîtrisé, les différences de performances seront minimales
- Dans les autres cas, il sera indispensable de choisir un critère plus élaboré (et plus coûteux)
- D'une façon générale : pas de vainqueur clair

### Amélioration de la fiabilité par validation croisée

On réalise deux passes d'appariement :

- 1 Recherche du meilleur appariement de  $I_1$  pour chaque pixel de  $I_2$
- 2 Recherche du meilleur appariement de  $I_2$  pour chaque pixel de  $I_1$

Et on ne retient que ceux dont l'appariement est symétrique.

## Diminution du coût

- Problème : en fonction de la taille de la fenêtre de corrélation, le coût peut être élevé
- Solutions :
  - limiter la taille de cette fenêtre ( $5 \times 5$ ,  $7 \times 7$ , ...)
  - limiter l'espace de recherche



## Diminution du coût par compromis CPU $\leftrightarrow$ RAM

- Implémentation naive des critères : de nombreuses opérations sont répétées
- Idée pour le SSD : optimiser le calcul en séparant les opérations en ligne et en colonne et en stockant les valeurs intermédiaires dans une matrice temporaire [PhD Pelcat, 2012].
- Résultats : quasi indépendance à  $k$

Optimisation	nombre d'opération $\oplus$	nombre d'opération $\otimes$
Sans	$(L - k)(H - k)(2k^2 - 1)$	$(L - k)(H - k)k^2$
Avec	$6L(H - k - 3) - 2k - 5$	$2(LH - 1) - k$

L, H : Largeur et Hauteur des images ; k : taille du noyau

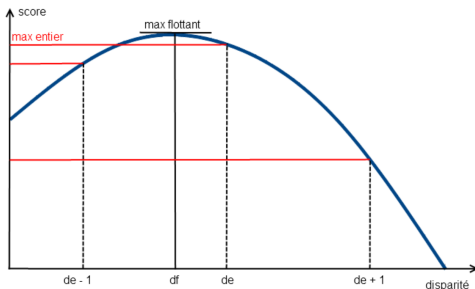
Pour une image 800 x 600 et k=11 :

Milliers d'opérations	opération $\oplus$	opération $\otimes$
Sans optimisation	111997	56231
Avec optimisation	2813	960
Rapport	39.8	58.6



## Améliorations qualitatives : précision subpixelique

- Par défaut, l'appariement se fait dans le domaine discret : un pixel de  $I_1$  en  $(u_1, v_1)$  sera associé à un pixel de  $I_2$  en  $(u_2, v_2)$
- On peut obtenir une précision subpixelique de la profondeur via une interpolation entre les valeurs entières des positions.

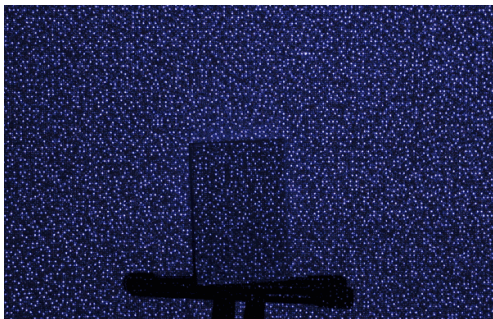


## Surfaces non texturées

- Les techniques corrélatives échouent pour des larges surface non texturées
- Une solution pour de l'environnement contrôlé : projection de mouchetis  
Mis en oeuvre en IR dans la Kinect / Microsoft

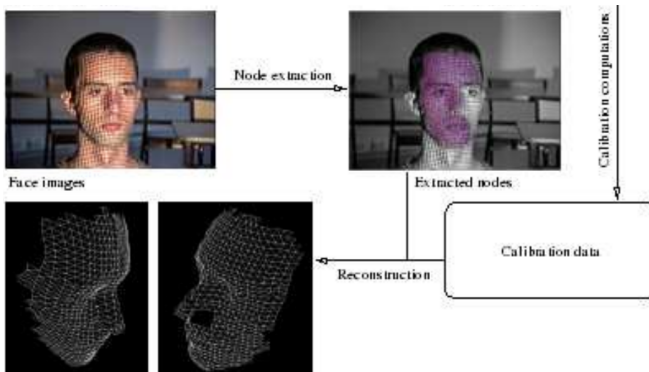


VIDEO KINECT



## Matching par projection

Si on projette une grille, on peut aussi récupérer la **topologie**, et on peut reconstituer un modèle 3D



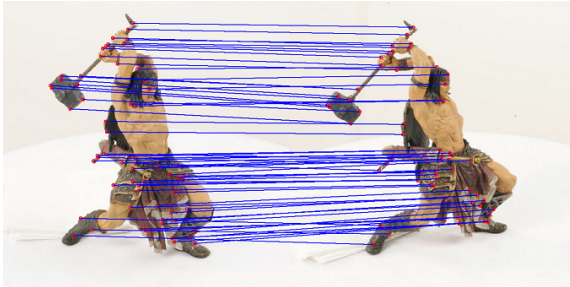
# Sommaire

- 1 Introduction
- 2 Exemple de problématiques à considérer
- 3 Appariement dense
- 4 Appariement épars**



# Principe

- On ne matche plus les pixels de l'image, mais des points prédéfinis
- Le matching se fait en utilisant la **caractérisation** de primitives (points, segments de droites, courbes, etc.) qui sont données en entrée



VIDEO : matching par BRIEF



## Mise en œuvre

- Rappel : Détecteur SIFT : fournit pour chaque point d'intérêt  $i$  détecté un vecteur de 128 valeurs

$$v_i = \{v_{i,1}, v_{i,2}, v_{i,3}, \dots, v_{i,128}\}$$

- Méthode *Brute force* : Pour chaque point  $i$  de  $I_1$ 
  - rechercher dans les points  $j$  de  $I_2$  le point qui minimise la (pseudo) distance euclidienne  $d$  :
$$d = (v_{i,1} - v_{j,1})^2 + (v_{i,2} - v_{j,2})^2 + \dots + (v_{i,128} - v_{j,128})^2$$
  - marquer ce point  $j$  comme apparié



## Mise en œuvre

- Rappel : Détecteur SIFT : fournit pour chaque point d'intérêt  $i$  détecté un vecteur de 128 valeurs

$$v_i = \{v_{i,1}, v_{i,2}, v_{i,3}, \dots, v_{i,128}\}$$

- Méthode *Brute force* : Pour chaque point  $i$  de  $I_1$ 
  - rechercher dans les points  $j$  de  $I_2$  le point qui minimise la (pseudo) distance euclidienne  $d$  :
$$d = (v_{i,1} - v_{j,1})^2 + (v_{i,2} - v_{j,2})^2 + \dots + (v_{i,128} - v_{j,128})^2$$
  - marquer ce point  $j$  comme apparié
- Problèmes :
  - coût
  - solution dépendant de l'ordre des points dans le jeu initial (solution non optimale)
  - on tombe dans des minimas locaux
- Problème difficile en théorie (*Curse of dimensionality*), mais qui se résout en général bien en pratique.

## Exemple (vecteurs de dimension 3)

- Soit les deux jeux de vecteurs pour  $I_1$  et  $I_2$ , contenant chacun 3 points A,B,C :

$$\begin{array}{ccc} A_1 & B_1 & C_1 \\ \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} & \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} & \begin{bmatrix} 1 \\ 2 \\ 6 \end{bmatrix} \\ A_2 & B_2 & C_2 \\ \begin{bmatrix} 5 \\ 5 \\ 7 \end{bmatrix} & \begin{bmatrix} 2 \\ 2 \\ 5 \end{bmatrix} & \begin{bmatrix} 3 \\ 3 \\ 2 \end{bmatrix} \end{array}$$

Calcul de distance :

$$d^2(A_1, A_2) = (1 - 5)^2 + (2 - 5)^2 + (3 - 7)^2 = 4^2 + 3^2 + 4^2 = 16 + 9 + 16 = 41$$



## Exemple (vecteurs de dimension 3)

- Soit les deux jeux de vecteurs pour  $I_1$  et  $I_2$ , contenant chacun 3 points A,B,C :

$$\begin{array}{ccc}
 A_1 & B_1 & C_1 & & A_2 & B_2 & C_2 \\
 \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} & \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} & \begin{bmatrix} 1 \\ 2 \\ 6 \end{bmatrix} & & \begin{bmatrix} 5 \\ 5 \\ 7 \end{bmatrix} & \begin{bmatrix} 2 \\ 2 \\ 5 \end{bmatrix} & \begin{bmatrix} 3 \\ 3 \\ 2 \end{bmatrix}
 \end{array}$$

Calcul de distance :

$$d^2(A_1, A_2) = (1 - 5)^2 + (2 - 5)^2 + (3 - 7)^2 = 4^2 + 3^2 + 4^2 = 16 + 9 + 16 = 41$$

- Tableau des valeurs de  $d^2$  :

	$A_2$	$B_2$	$C_2$
$A_1$	41	5	6
$B_1$	2	14	21
$C_1$	26	2	21

## Exemple (vecteurs de dimension 3)

- Soit les deux jeux de vecteurs pour  $I_1$  et  $I_2$ , contenant chacun 3 points A,B,C :

$$\begin{array}{ccc}
 A_1 & B_1 & C_1 & & A_2 & B_2 & C_2 \\
 \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} & \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} & \begin{bmatrix} 1 \\ 2 \\ 6 \end{bmatrix} & & \begin{bmatrix} 5 \\ 5 \\ 7 \end{bmatrix} & \begin{bmatrix} 2 \\ 2 \\ 5 \end{bmatrix} & \begin{bmatrix} 3 \\ 3 \\ 2 \end{bmatrix}
 \end{array}$$

Calcul de distance :

$$d^2(A_1, A_2) = (1 - 5)^2 + (2 - 5)^2 + (3 - 7)^2 = 4^2 + 3^2 + 4^2 = 16 + 9 + 16 = 41$$

- Tableau des valeurs de  $d^2$  :

	$A_2$	$B_2$	$C_2$
$A_1$	41	5	6
$B_1$	2	14	21
$C_1$	26	2	21

- Le résultat de l'appariement donnerai avec cet ordre :  $(A_1, B_2), (A_2, B_1), (A_3, B_3) : \sum d^2 = 2 + 5 + 21 = 28$  : pas optimal

## Exemple (vecteurs de dimension 3)

- Soit les deux jeux de vecteurs pour  $I_1$  et  $I_2$ , contenant chacun 3 points A,B,C :

$$\begin{array}{ccc}
 A_1 & B_1 & C_1 & & A_2 & B_2 & C_2 \\
 \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} & \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} & \begin{bmatrix} 1 \\ 2 \\ 6 \end{bmatrix} & & \begin{bmatrix} 5 \\ 5 \\ 7 \end{bmatrix} & \begin{bmatrix} 2 \\ 2 \\ 5 \end{bmatrix} & \begin{bmatrix} 3 \\ 3 \\ 2 \end{bmatrix}
 \end{array}$$

Calcul de distance :

$$d^2(A_1, A_2) = (1 - 5)^2 + (2 - 5)^2 + (3 - 7)^2 = 4^2 + 3^2 + 4^2 = 16 + 9 + 16 = 41$$

- Tableau des valeurs de  $d^2$  :

	$A_2$	$B_2$	$C_2$
$A_1$	41	5	6
$B_1$	2	14	21
$C_1$	26	2	21

- Le résultat de l'appariement donnerai avec cet ordre :  
 $(A_1, B_2), (A_2, B_1), (A_3, B_3) : \sum_{d^2} = 2 + 5 + 21 = 24$  : pas optimal
- En inversant l'énumération des points  $A_i$ , on obtiendrait :  
 $(A_1, B_3), (A_2, B_1), (A_3, B_2) : \sum_{d^2} = 6 + 2 + 2 = 10$  : bien mieux...

## Améliorations du *Brute force*

- Limitation de l'espace de recherche sur la base d'un critère géométrique (position du point dans l'image)



## Améliorations du *Brute force*

- Limitation de l'espace de recherche sur la base d'un critère géométrique (position du point dans l'image)
- Utilisation d'un critère d'ambiguïté (Lowe,2004)



## Améliorations du *Brute force*

- Limitation de l'espace de recherche sur la base d'un critère géométrique (position du point dans l'image)
- Utilisation d'un critère d'ambiguïté (Lowe,2004)
  - Recherche du 2<sup>e</sup> meilleur candidat à l'appariement et calcul de la distance correspondante  $d_2$



## Améliorations du *Brute force*

- Limitation de l'espace de recherche sur la base d'un critère géométrique (position du point dans l'image)
- Utilisation d'un critère d'ambiguïté (Lowe,2004)
  - Recherche du 2<sup>e</sup> meilleur candidat à l'appariement et calcul de la distance correspondante  $d_2$
  - validation de l'appariement uniquement si  $d/d_2 < \text{seuil}$  (Lowe propose 0,8)



## Améliorations du *Brute force*

- Limitation de l'espace de recherche sur la base d'un critère géométrique (position du point dans l'image)
- Utilisation d'un critère d'ambiguïté (Lowe,2004)
  - Recherche du 2<sup>e</sup> meilleur candidat à l'appariement et calcul de la distance correspondante  $d_2$
  - validation de l'appariement uniquement si  $d/d_2 < \text{seuil}$  (Lowe propose 0,8)
- Validation croisée : prise en compte de l'appariement que si la recherche du minima de  $I_2$  à partir de  $I_1$  donne le même résultat que la recherche du minima de  $I_1$  à partir de  $I_2$





## Améliorations du *Brute force*

- Limitation de l'espace de recherche sur la base d'un critère géométrique (position du point dans l'image)
- Utilisation d'un critère d'ambiguïté (Lowe, 2004)
  - Recherche du 2<sup>e</sup> meilleur candidat à l'appariement et calcul de la distance correspondante  $d_2$
  - validation de l'appariement uniquement si  $d/d_2 < \text{seuil}$  (Lowe propose 0,8)
- Validation croisée : prise en compte de l'appariement que si la recherche du minima de  $I_2$  à partir de  $I_1$  donne le même résultat que la recherche du minima de  $I_1$  à partir de  $I_2$
- Techniques avancées : LSH (*Local Sensitive Hashing*), kd-tree  
Principe : privilégier la *rapidité* en sacrifiant éventuellement l'exactitude (voir FLANN : *Fast Linear Approximate Nearest Neighbor search*)



## Connaissance à priori

- Si le contexte des données est connu, on peut utiliser des a-priori pour fiabiliser le matching
  - Si modèle algébrique, alors on valide les appariements via un estimateur robuste (RanSaC)  
(exemple : estimation de la géométrie épipolaire)
  - Si la topologie est connue, on peut rejeter les appariements incohérents

