Vision par ordinateur

Mise en correspondance stéréoscopique

Frédéric Devernay
avec de nombreux transparents de Richard Szeliski

Stéréoscopie

Étant données deux images ou plus d'une même scène, calculer une représentation de la forme 3D de cette scène

Quelles applications possibles ?

Modélisation de visage

Une paire stéréo = un modèle 3D de visage

[Frédéric Devernay, INRIA]

Z-keying : mélange réel/synthétique

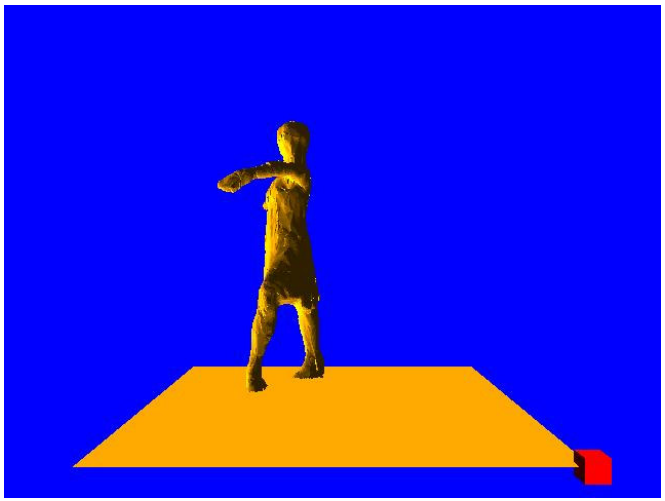
Takeo Kanade, CMU (Stereo Machine)

Virtualized Reality™

Takeo Kanade, CMU

- plus de 50 flux vidéo simultanés

reconstruction de séquences de modèles 3D

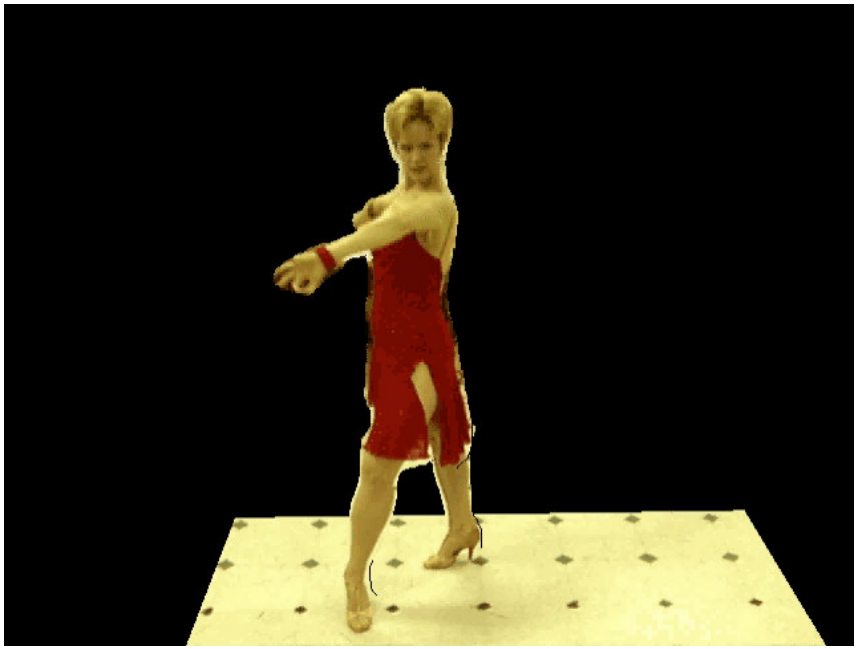


http://www.cs.cmu.edu/afs/cs/project/VirtualizedR/www/VirtualizedR.html

Virtualized Reality™

Takeo Kanade, CMU

- générer autre vidéo

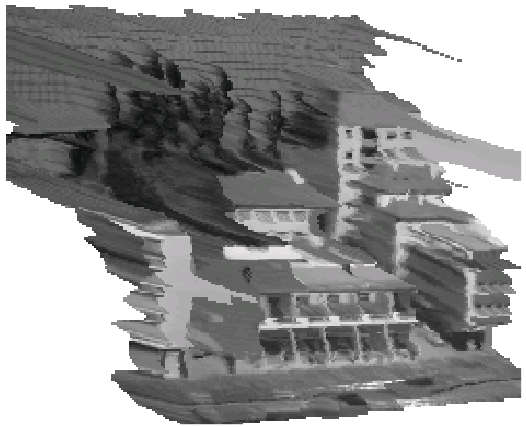


Interpolation de vues

Etant donné deux images avec des correspondences, faire du *morphing* entre elles [Chen & Williams, SIGGRAPH'93]

entrée

profondeur

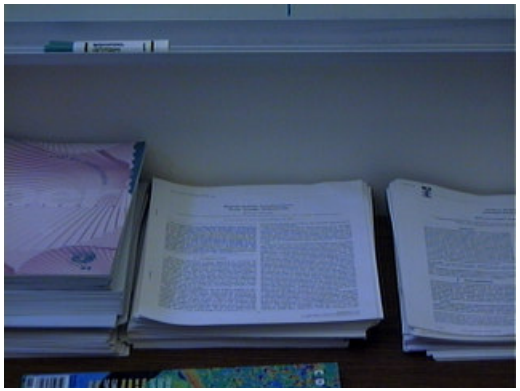


nouvelle vue

[Matthies,Szeliski,Kanade'88]

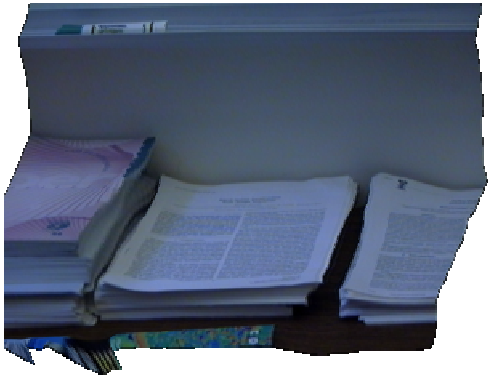
Interpolation de vues

Carte de profondeur sous forme de splines



entrée

profondeur

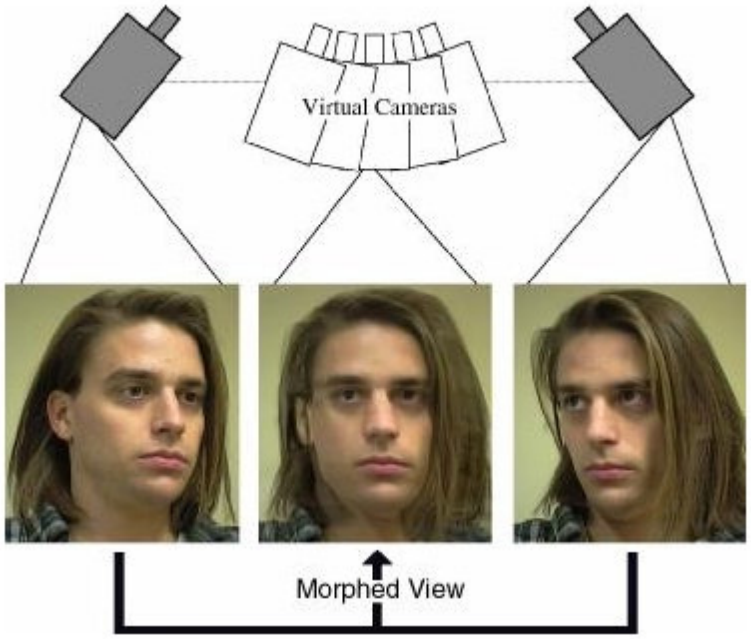


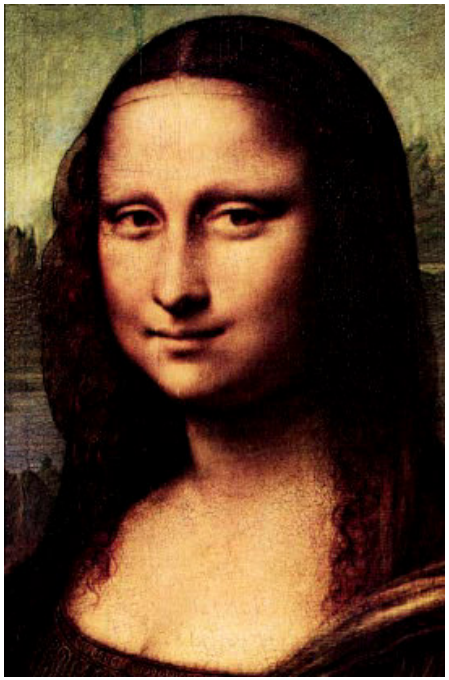
autre vue

[Szeliski & Kang ‘95]

View Morphing

Morph between pair of images using epipolar geometry [Seitz & Dyer, SIGGRAPH'96]

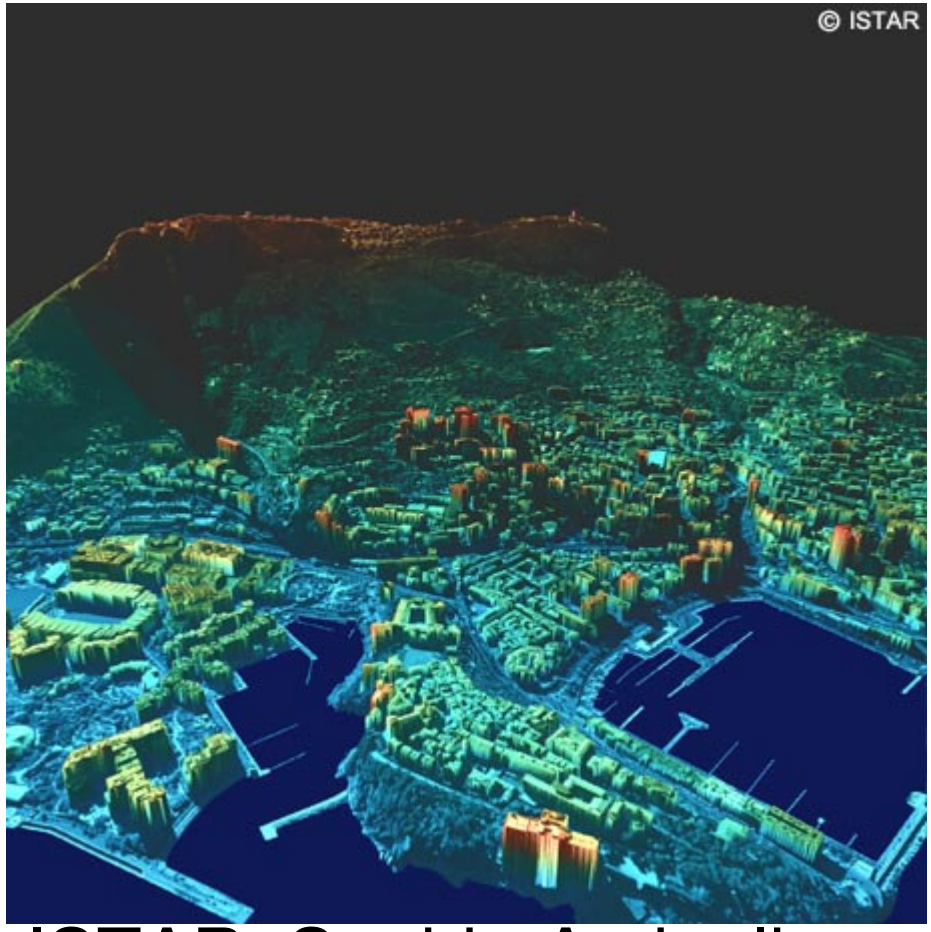


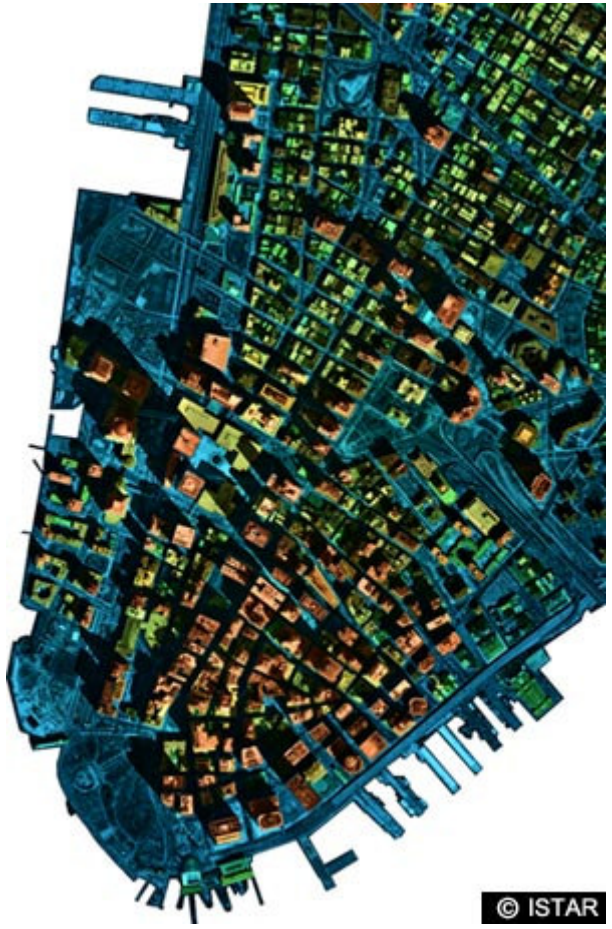


Modèles de terrain / ville

ISTAR, Sophia Antipolis

Modèles de terrain / ville





ISTAR, Sophia Antipolis

Autres applications

- Suivi temps réel de personnes (systèmes de Pt. Gray Research and SRI)
- Correction de regard pour la vidéo-conférence [Ott,Lewis,Cox InterChi'93]
- D'autres idées ?

Mise en correspondance stéréo

Étant données deux images ou plus d'une même scène, calculer une représentation de la forme de cette scène

Quelles représentations possibles ?

- cartes de profondeur
- modèles volumiques
- modèles de surfaces 3D
- superposition de calques

Mise en correspondance stéréo

Quels algorithmes possibles ?

- mettre en correspondance des “features” (points d'intérêts) et interpoler
- idem avec des contours
- mettre en corresp. des fenêtres de pixels

optimisation:

- mise à jour itérative
- programmation dynamique
- minimisation d'énergie (regularisation, stochastique)
- algorithmes de graphes

Nous allons voir...

Rectification des images

Critère de mise en correspondance

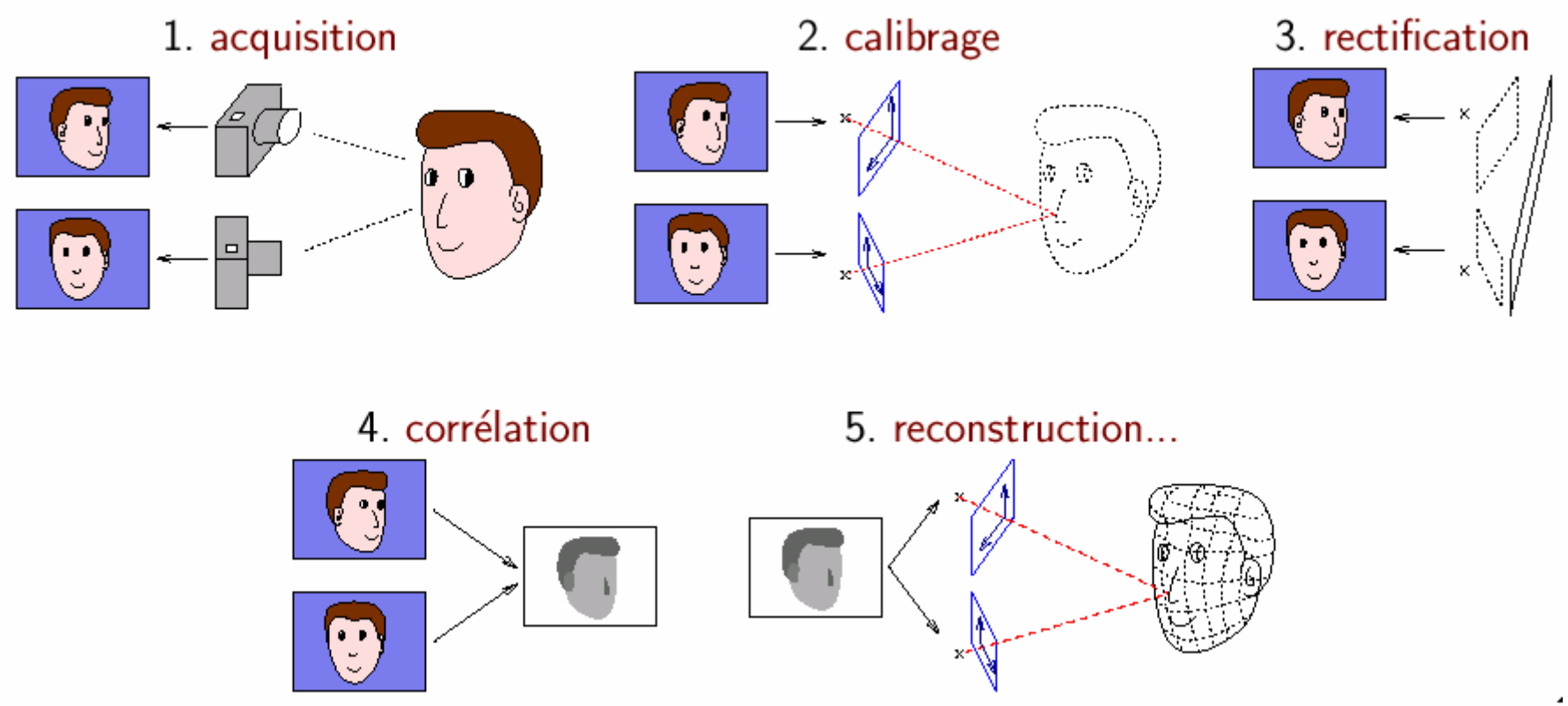
Algorithmes locaux (agrégation)

Mise à jour itérative

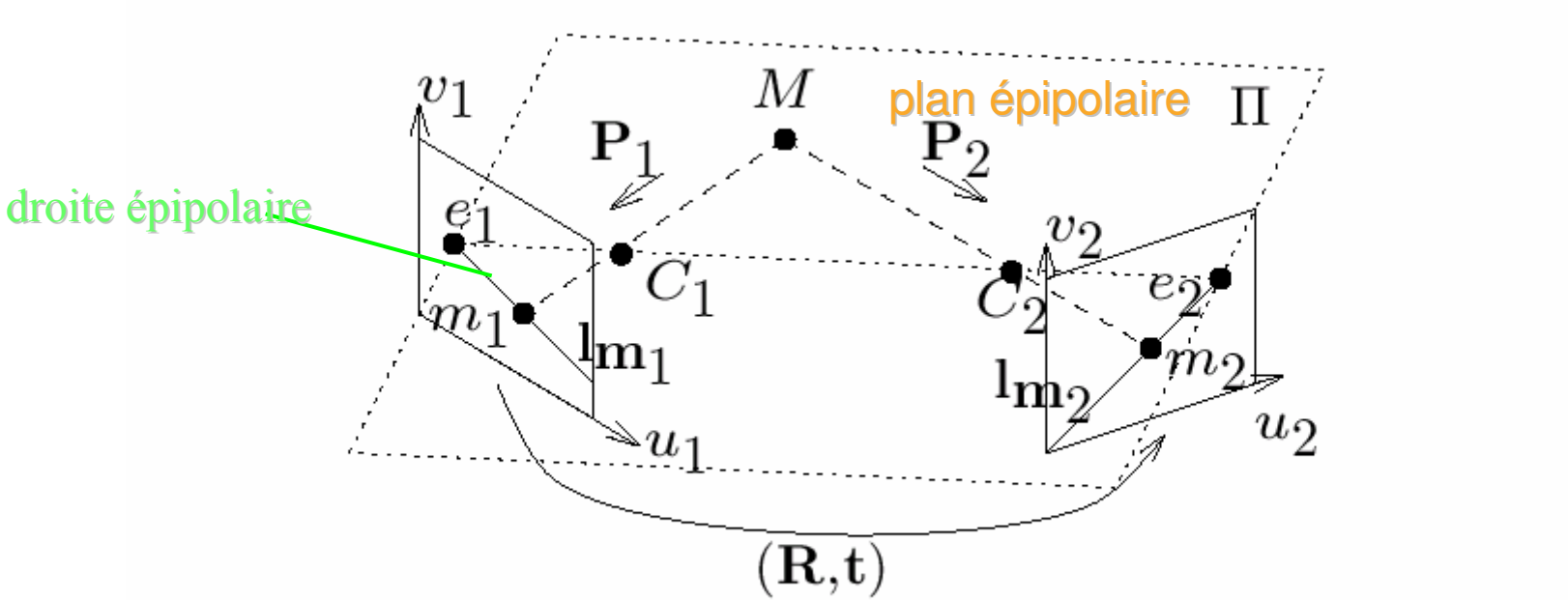
Algorithmes par optimisation :

- formulation énergétique
- Champs de Markov
  - champ moyen, stochastique, et algorithmes de graphe

Étapes de la stéréoscopie



Stéréo : géométrie épipolaire



$\mathbf{m}_1 = (x_1,y_1,1)$, $\mathbf{m}_2 = (x_2,y_2,1)$, et $\mathbf{m}_1^T \mathbf{F}_{12} \mathbf{m}_2 = 0$.

L'information sur la géométrie épipolaire est contenue dans la matrice (3x3) fondamentale $\mathbf{F}$, qui permet de faire une reconstruction projective

Stéréo : géométrie épipolaire

à partir de deux images, on peut calculer $\mathbf{F}$, qui nous donne les droites épipolaires

les droites épipolaires sont la projection du faisceau de plans passant par les centres optiques

Rectification : transformation homographique des images pour que les droites épipolaires soient horizontales

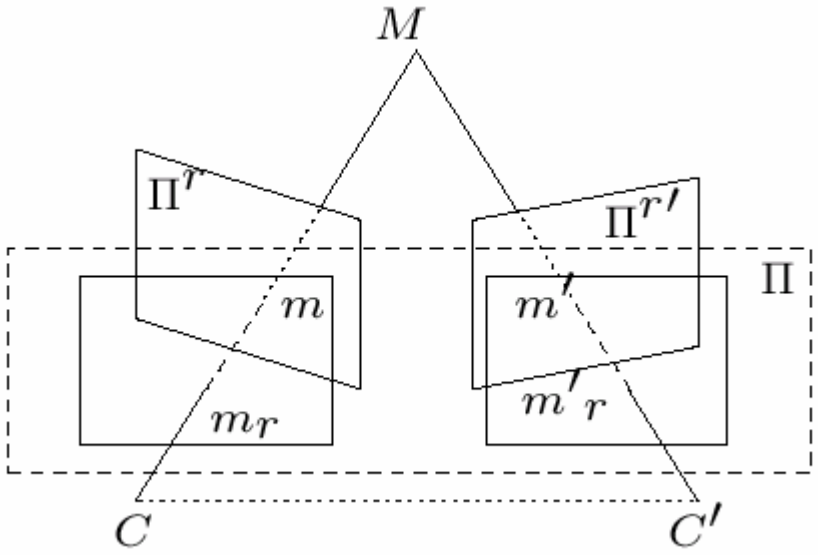
Rectification

Transformation de images pour préparer à la stéréo : les droites épipolaires deviennent horizontales, les points qui se correspondent ont le même y

En 3D : reprojection sur plan // à l'axe joignant les centres optiques

9 D.L. : orientation du plan (1) et params intrinsèques des caméras rectifiés (5+3)

Nécessite calibrage fort



Rectification et calibrage faible

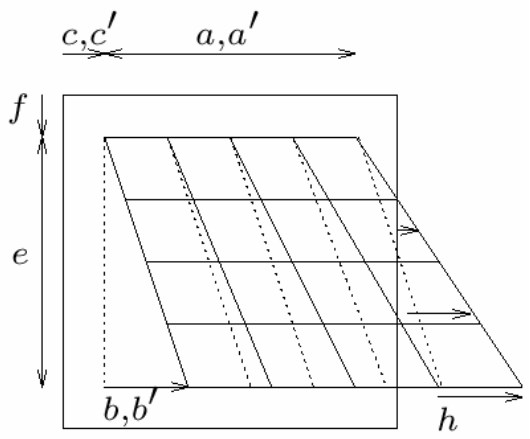
Rectifications compatibles avec $\mathbf{F}$: $\mathbf{F} \cong \mathbf{R'}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{R}$

Calculer une paire $\mathbf{R}_0$,$\mathbf{R'}_0$ par SVD, puis

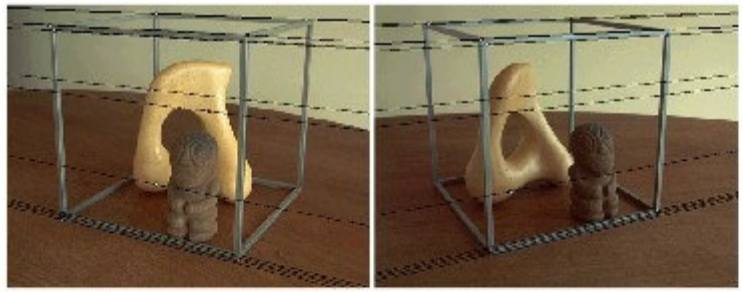
$$\mathbf{R} \cong \begin{bmatrix} a & b & c \\ 0 & e & f \\ 0 & h & i \end{bmatrix} \mathbf{R}_0 \text{ et } \mathbf{R}' \cong \begin{bmatrix} a' & b' & c' \\ 0 & e & f \\ 0 & h & i \end{bmatrix} \mathbf{R}'_0, \text{ avec } \begin{vmatrix} e & f \\ h & i \end{vmatrix} \neq 0.$$

Interprétation géométrique :

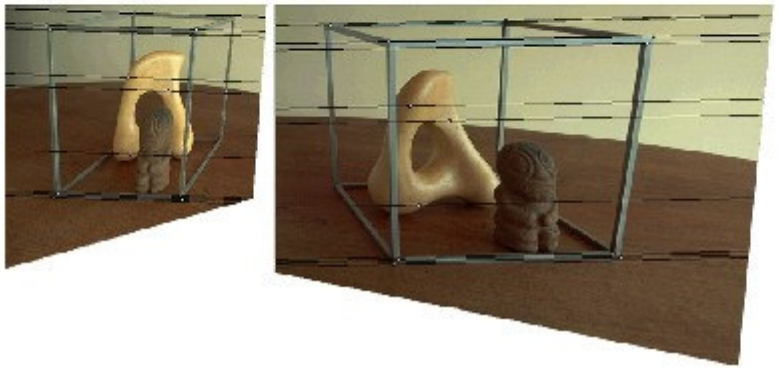
Optimisation par minimisation
de la déformation des images
[Zhang and Loop, MSR-TR-99-21]



Rectification

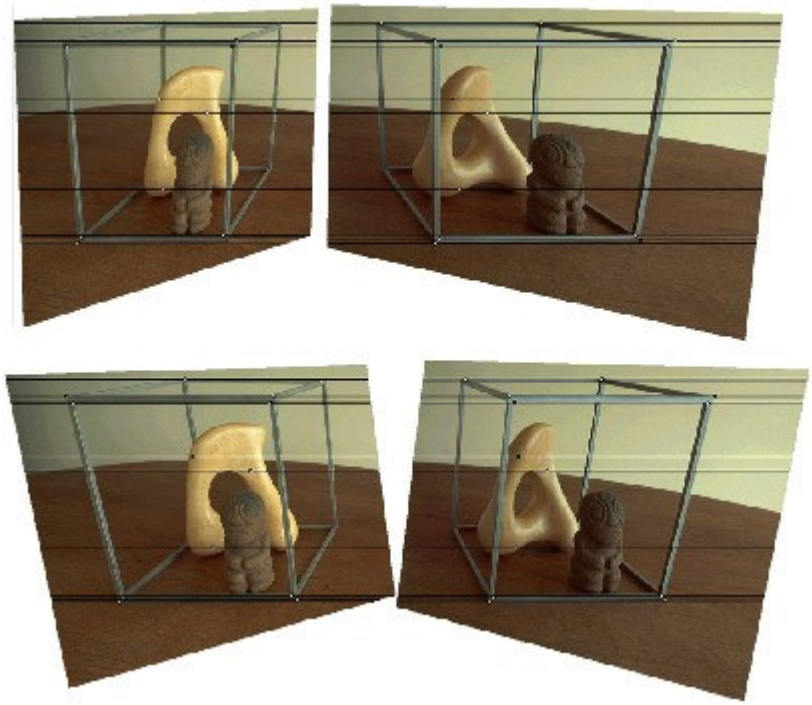


(a) Original image pair overlayed with several epipolar lines.



(b) Image pair transformed by the specialized projective mapping $\mathbf{H}_p$ and $\mathbf{H}'_p$. Note that the epipolar lines are now parallel to each other in each image.

Rectification

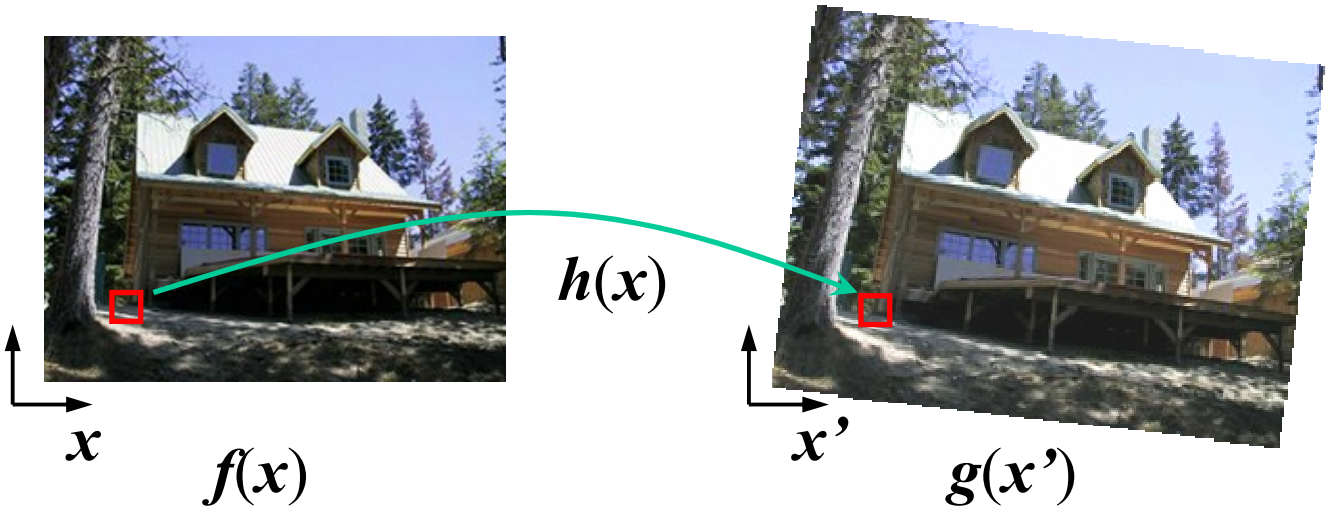


(c) Image pair transformed by the similarity $\mathbf{H}_r$ and $\mathbf{H}'_r$. Note that the image pair is now rectified (the epipolar lines are horizontally aligned).

(d) Final image rectification after shearing transform $\mathbf{H}_s$ and $\mathbf{H}'_s$. Note that the image pair remains rectified, but the horizontal distortion is reduced.

Transformation d'une image

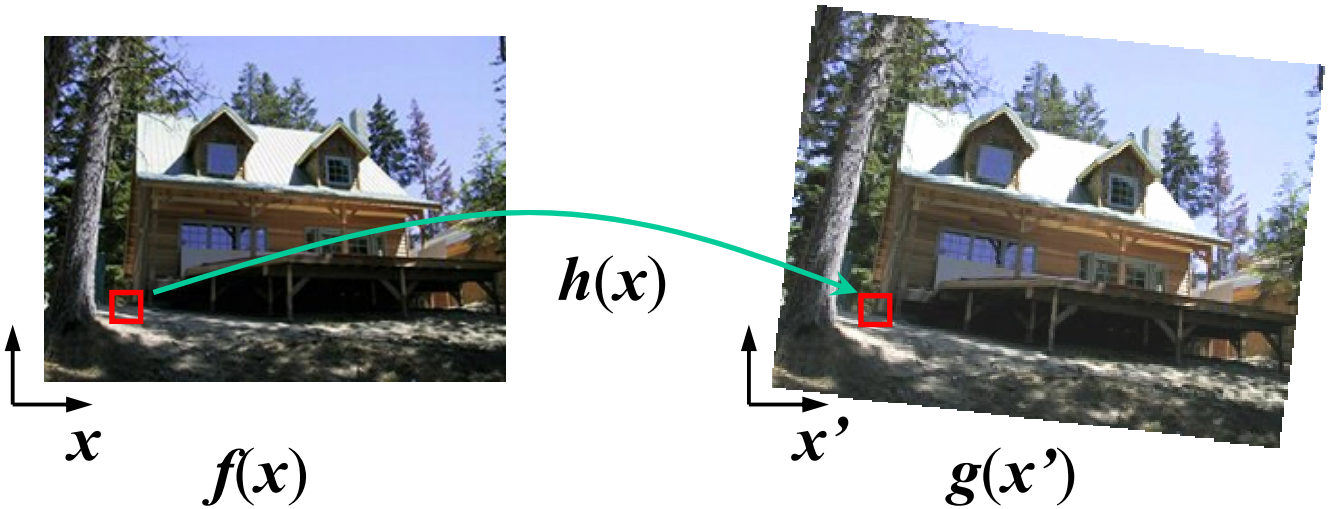
Étant donnée une transformation $x' = h(x)$ et une image source $f(x)$, comment calculer l'image transformée $g(x') = f(h(x))$?



Transformation directe

Envoyer chaque pixel $f(x)$ vers le lieu correspondant $x' = h(x)$ de $g(x')$

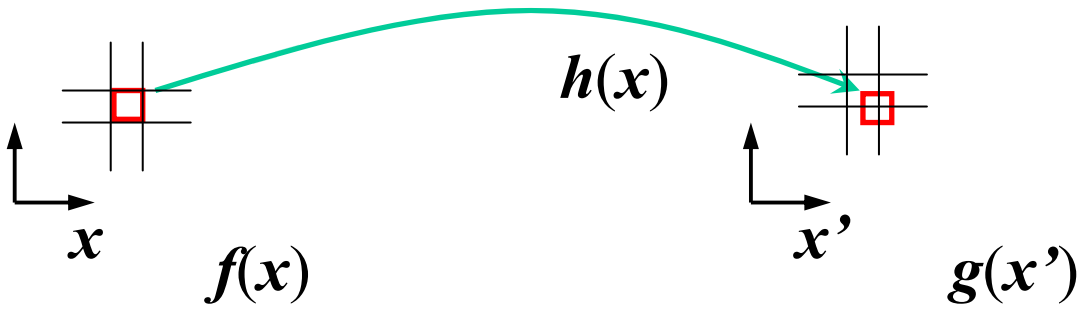
- Et si le pixel "tombe" entre deux pixels ?



Transformation directe

Envoyer chaque pixel $f(x)$ vers le lieu correspondant $x' = h(x)$ de $g(x')$

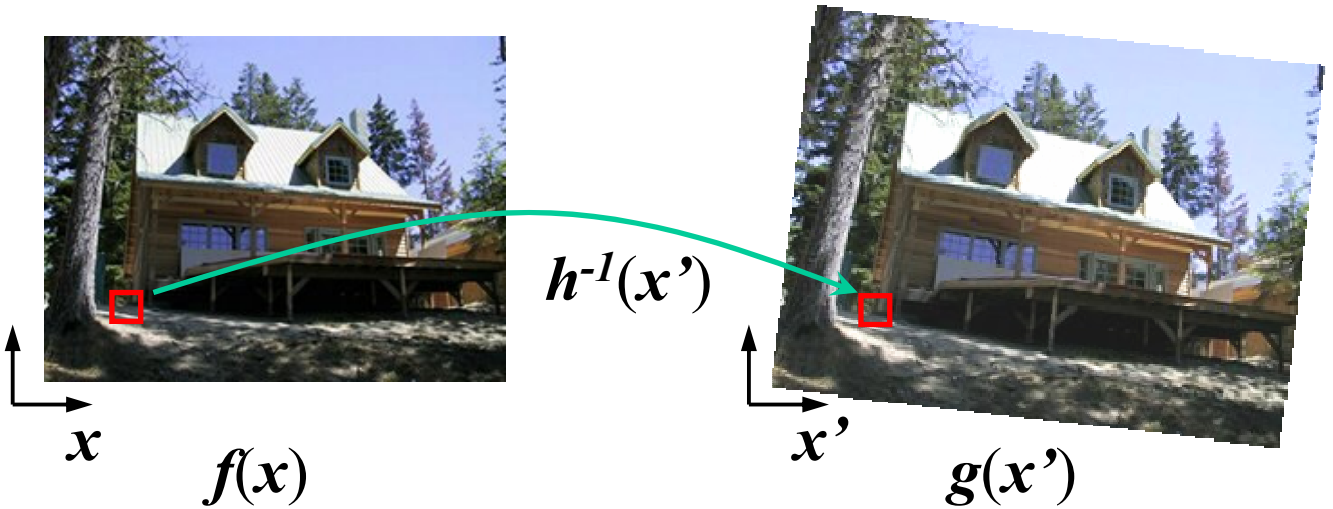
- Et si le pixel "tombe" entre deux pixels ?
- Réponse: ajouter la “contribution” à plusieurs pixels, normaliser plus tard (*splatting*)



Transformation inverse

Prendre chaque pixel $g(x')$ du lieu correspondant $x = h^{-1}(x')$ dans $f(x)$

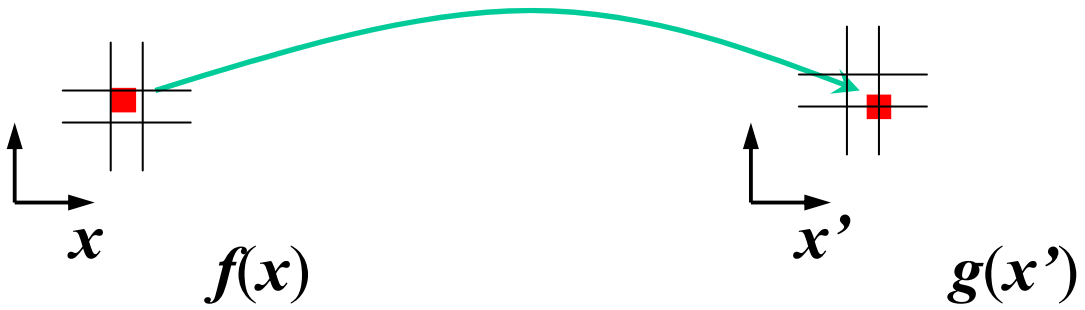
- Et si le pixel "tombe" entre deux pixels ?



Transformation inverse

Prendre chaque pixel $g(x')$ du lieu correspondant $x = h^{-1}(x')$ dans $f(x)$

- Et si le pixel "tombe" entre deux pixels ?
- Réponse : rééchantillonner les valeurs à partir de l'image originale *interpolée*



Interpolation

Filtres d'interpolation :

- plus proche voisin
- bilinéaire
- bicubique
- sinc / FIR

Nécessaire pour éviter les phénomènes d'aliasing et de moiré

Critère de correspondance

Valeurs d'intensité (corrélation)

Images filtrées passe-bande [Jones & Malik 92]

"Coins" [Zhang, …]

Contours [beaucoup de gens…]

Gradients [Seitz 89; Scharstein 94]

Statistiques sur le rang [Zabih & Woodfill 94]

Critères de corrélation

SSD (sum of squared differences). d = disparité.

$$c_{x,y}(d) = - \sum_{-l \leq i \leq l; -h \leq j \leq h} (I_1(x+i, y+j) - I_2(x+d+i, y+j))^2$$

Critère normalisé

$$c_{x,y}(d) = - \frac{\sum_{i,j}((I_1(x+i, y+j) - \overline{I_1}(x,y)) - (I_2(x+d+i, y+j) - \overline{I_2}(x,y)))^2}{\sqrt{\sum_{i,j}(I_1(x+i, y+j) - \overline{I_1}(x,y))^2} \sqrt{\sum_{i,j}(I_2(x+d+i, y+j) - \overline{I_2}(x,y))^2}}$$

Critère normalisé simplifié

$$I'_1(x+i, y+j) = I_1(x+i, y+j) - \overline{I_1}(x+i, y+j)$$

$$I'_2(x+i, y+j) = I_2(x+i, y+j) - \overline{I_2}(x+i, y+j)$$

$$c_{x,y}(d) = - \frac{\sum_{i,j}(I'_1(x+i, y+j) - I'_2(x+d+i, y+j))^2}{\sqrt{\sum_{i,j} I'_2(x+d+i, y+j)^2}}$$

Critères de corrélation

Cross-correlation = cosinus des vecteurs d'intensité

$$c_{x,y}(d) = \frac{\sum_{i,j}(I_1(x+i,y+j) - \overline{I_1}(x,y))(I_2(x+d+i,y+j) - \overline{I_2}(x,y))}{\sqrt{\sum_{i,j}(I_1(x+i,y+j) - \overline{I_1}(x,y))^2}\sqrt{\sum_{i,j}(I_2(x+d+i,y+j) - \overline{I_2}(x,y))^2}}$$

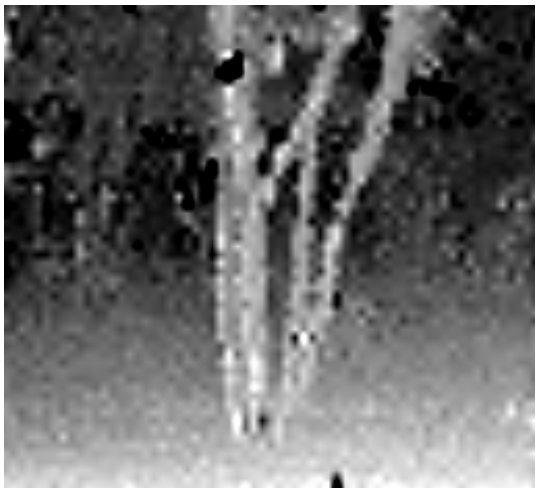
Critère simplifié

$$c_{x,y}(d) = \frac{\sum_{i,j} I_1'(x+i,y+j) I_2'(x+d+i,y+j)}{\sqrt{\sum_{i,j} I_2'(x+d+i,y+j)^2}}$$

Trouver les correspondances

calculer le critère de correspondance (par ex. correlation ou Lucas-Kanade) en tous les pixels simultanément

recherche des correspondances sur les droites épipolaires (moins de candidats)

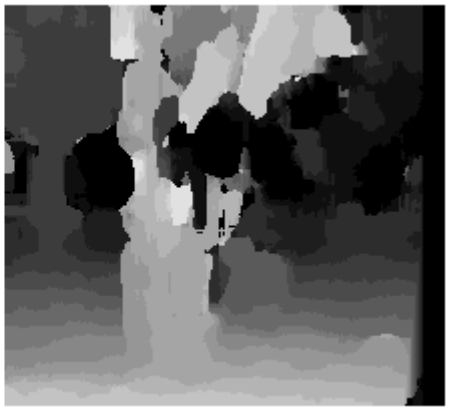


Taille de la fenêtre de corrélation

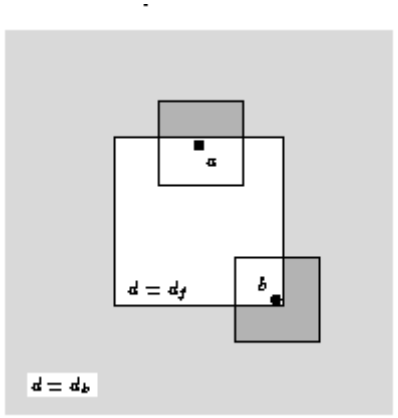
Petit voisinage : plus de détails

Grand voisinage : moins d'erreurs isolées

w = 3



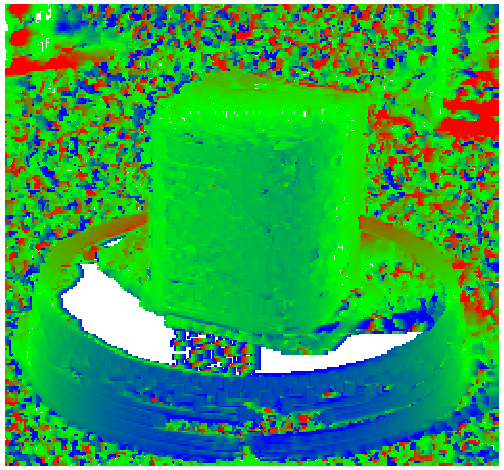
w = 20



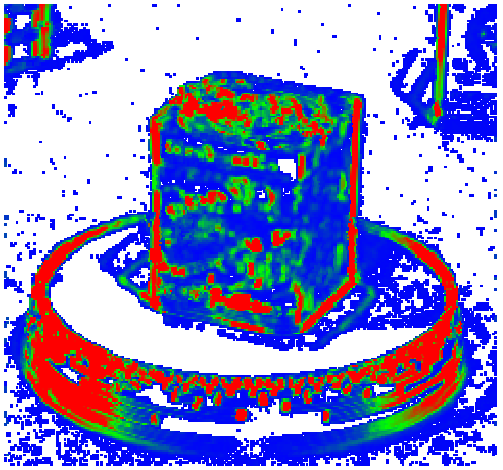
Stéréo : modéliser la confiance

Mesure de l'incertitude par le score de corrélation

entrée



disparité



confiance

Stéréoscopie par corrélation

Calculer les images I'_1, I'_2, variance(I'_2)

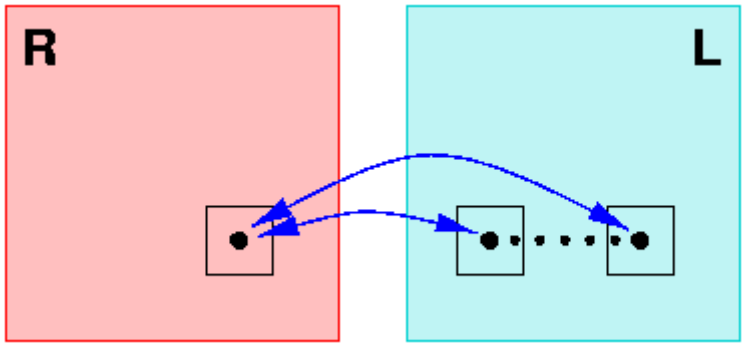
À chaque valeur de la disparité d correspond l'homographie

$$\mathbf{H}_d = \mathbf{R'}^{-1} \begin{bmatrix} 1 & 0 & d \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{R}$$

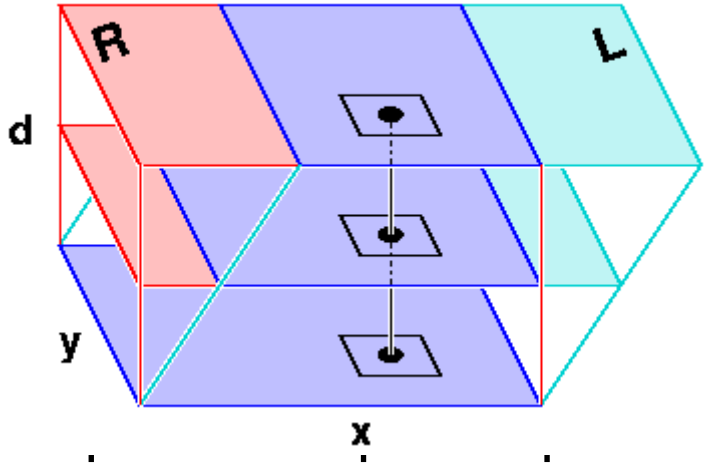
À une valeur de d correspond donc un *plan 3D* : balayer les valeurs de d revient à balayer l'espace par un plan 3D

Stéréoscopie par corrélation

Réordonnancement des boucles (pixel/disparité)



pour chaque pixel,
  pour chaque disparité
    calculer le score



pour chaque pixel
  calculer le score

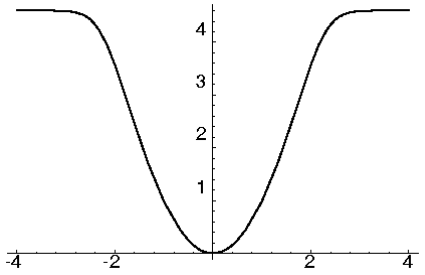
Stéréoscopie par corrélation

- Pour chaque disparité, calculer le côut en chaque pixel

$$E_0(x, y; d) = \rho(I_L(x' + d, y') - I_R(x', y'))$$

Pourquoi une fonction ρ robuste

- occlusions, autres erreurs

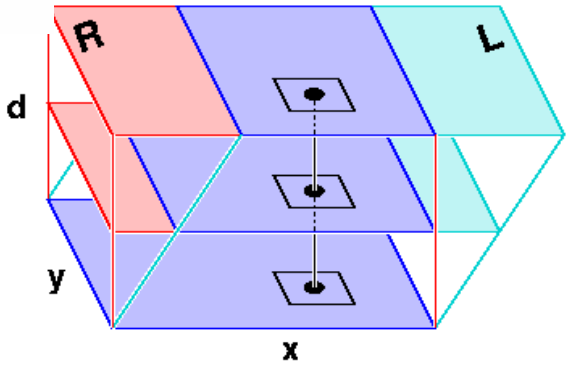


On peut également utiliser un autre critère (cosinus)

Stéréoscopie par corrélation

2. Score = moyenne spatiale

$$E(x, y; d) = \sum_{(x',y') \in N(x,y)} E_0(x', y', d)$$

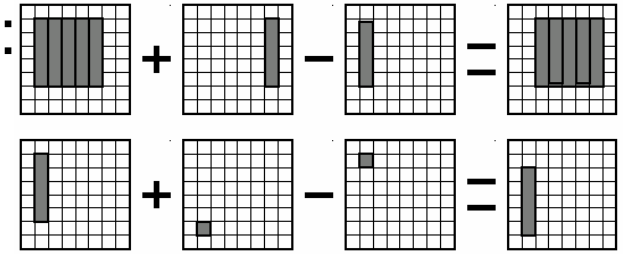


- On utilise une fenêtre carrée (facilement optimisable)

- On peut également utiliser une moyenne pondérée, de la diffusion [non-linéaire]...

Stéréoscopie par corrélation

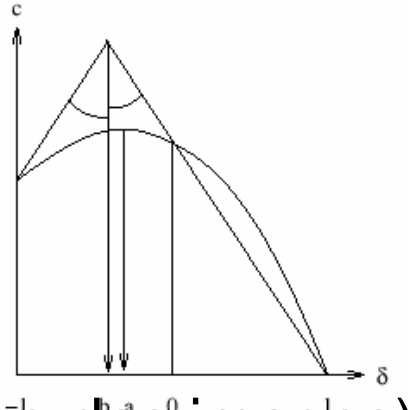
3. Optimisation par fenêtre glissante : complexité indépendante de la taille de la fenêtre de corrélation



4. En chaque pixel, meilleur score → disparité
5. Interpolation sous-pixélique parabole ou "toît"

$$d = d_0 + \frac{1}{2}\frac{c(d_0+1) - c(d_0-1)}{(c(d_0) - c(d_0+1)) + (c(d_0) - c(d_0-1))}$$

$$d = d_0 + \begin{cases} \frac{1}{2}\frac{c(d_0+1)-c(d_0-1)}{c(d_0)-c(d_0-1)} & \text{si } c(d_0+1) > c(d_0-1), \\ \frac{1}{2}\frac{c(d_0+1)-c(d_0-1)}{c(d_0+1)-c(d_0)} & \text{si } c(d_0+1) < c(d_0-1). \end{cases}$$



6. Validation par corrélation retour (échange des images)

Stéréo par corrélation classique

Avantages :

- donne une description détaillée de la surface
- algorithme rapide (fenêtre glissante)
- disparité sous-pixélique et confiance

Limitations :

- petite baseline $\Rightarrow$ estimations bruitées
- échoue dans les zones sans textures
- "bavures" au frontières des occlusions

Stéréo par corrélation fine

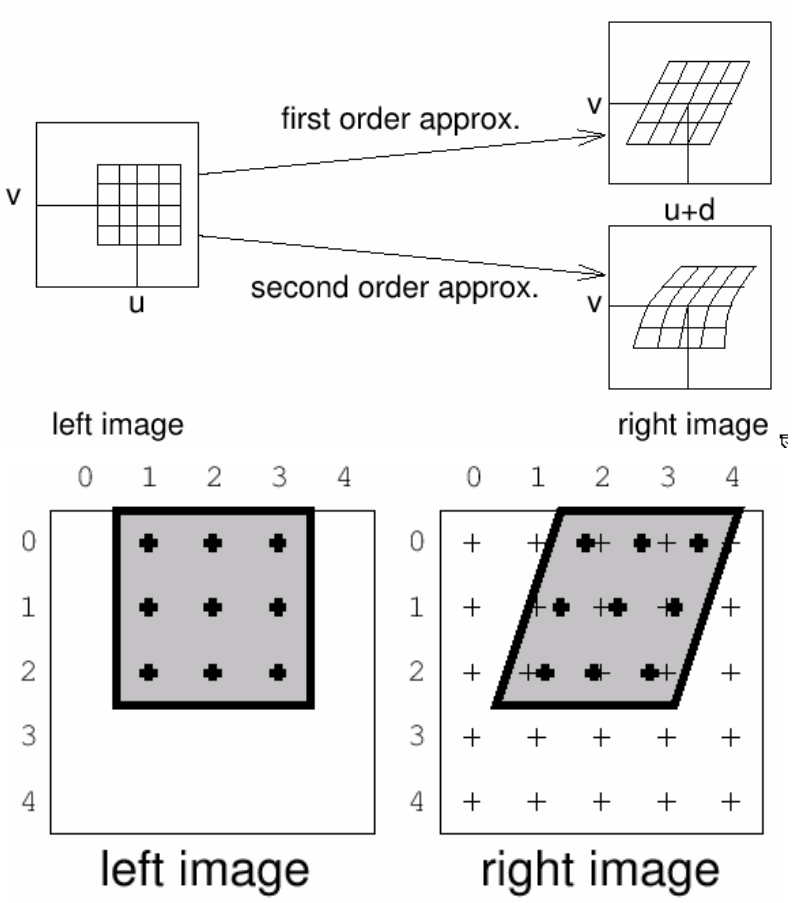
Si la surface n'est pas fronto-parallèle…

On optimise en chaque point des paramètres supplémentaires :

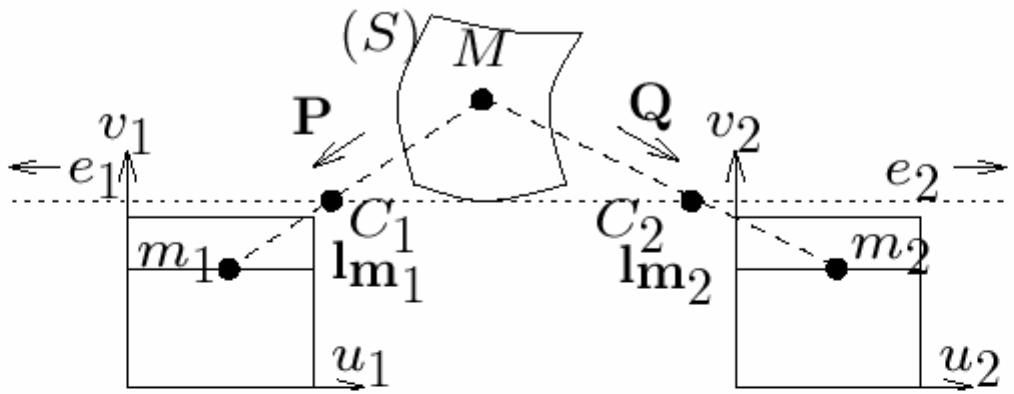
Dérivées de la disparité

- premières (1+2 pars.)
- secondes (1+2+3)

Descente de gradient



Reconstruction



matrices de projection *rectifiées* : $\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$ et $\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$

$$\mathbf{M} = (x,y,z,1), \quad \mathbf{m}_1 = \mathbf{PM} \cong (u_1,v_1,1), \quad \mathbf{m}_2 = \mathbf{QM} \cong (u_1 + d(u_1,v_1),v_1,1).$$

$$\text{donc} \quad \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{q}_1 - \mathbf{p}_1 \\ \mathbf{p}_3 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \cong \begin{pmatrix} u_1 \\ v_1 \\ d(u_1,v_1) \\ 1 \end{pmatrix}.$$

Quelques résultats

Résumé

Applications
Rectification
Critères de corrélation
Algorithmes locaux (par agrégation)
Méthodes surfaciques
- Stéréoscopie par corrélation classique
- Stéréoscopie par corrélation fine

La suite…

Algorithmes par optimisation

- formulation par minimisation d'énergie
- champs de Markov
- champ moyen ; programmation dynamique ;
- stochastique ; coupes de graphes

Stéréo multi-image

Méthodes volumiques

Méthodes par ensembles de niveau

Bibliographie

O. Faugeras, Three-Dimensional Computer Vision: a Geometric Viewpoint, MIT Press, 1993.

F. Devernay, Vision stéréoscipique et propriétés différentielles des surfaces, Thèse de l'École Polytechnique, 1997.

R. Szeliski. Stereo algorithms and representations for image-based rendering. In British Machine Vision Conference (BMVC'99), volume 2, pages 314-328, Nottingham, England, September 1999.

R. Szeliski and R. Zabih. An experimental comparison of stereo algorithms. In International Workshop on Vision Algorithms, pages 1-19, Kerkyra, Greece, September 1999.