Vision par ordinateur

Stéréoscopie par minimisation d'énergie

Frédéric Devernay
d'après le cours de Richard Szeliski

Mise en correspondance stéréo

Quels algorithmes possibles ?

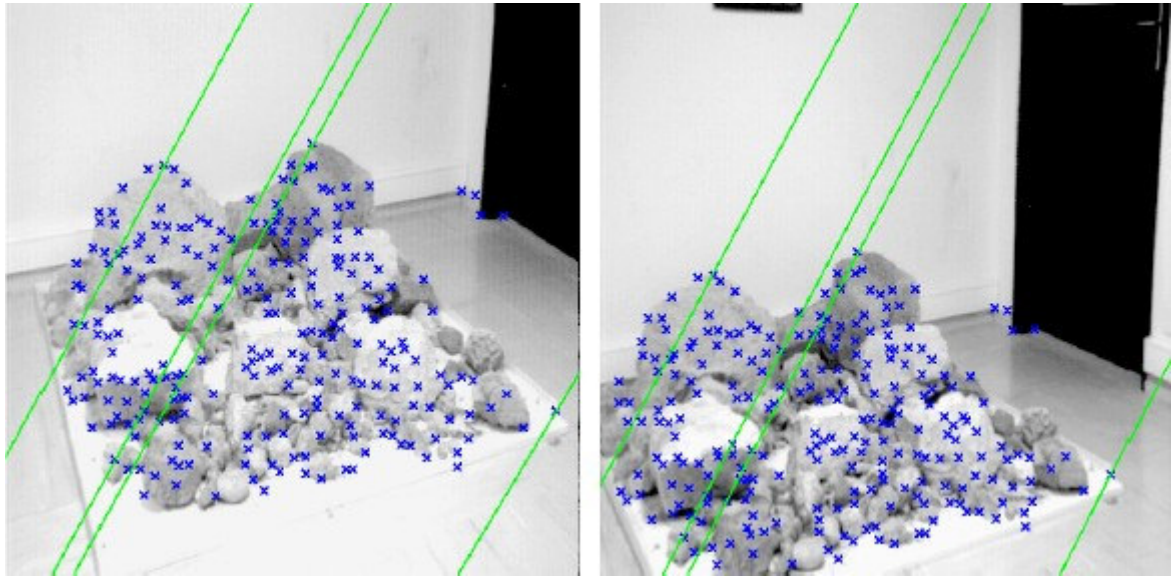
- mettre en correspondance des “features” (points d'intérêts) et interpoler
- idem avec des contours
- mettre en corresp. des fenêtres de pixels

optimisation:

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

Stéréo à base de points d'intérêt

Mise en correspondance de "coins" (pts d'intérêt)



Interpoler pour obtenir une carte de profondeur

Interpolation de données

Étant donné un ensemble de pts 3D, comment *interpoler* pour obtenir une surface 3D ?

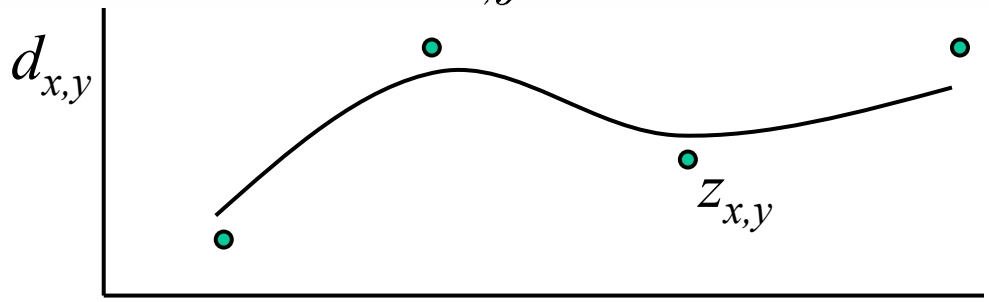
Interpolation de données éparses [Nielson93]

- trianguler dans une des deux images
- mettre dans un grille et remplir
- placer une *fonction de noyau* en chaque point
- minimiser une fonction d'énergie (membrane)

Minimisation d'énergie

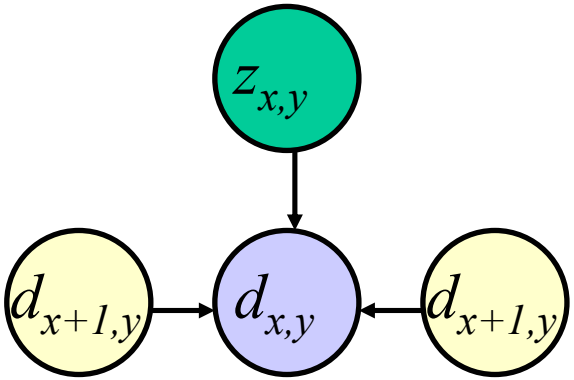
Exemple en 1D: splines d'approximation

$$
\begin{aligned}
E_{\text{total}}(\mathbf{d}) &= E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d}) \\
E_{\text{data}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - z_{x,y})^2 \\
E_{\text{membrane}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - d_{x-1,y})^2 \\
E_{\text{thin plate}}(\mathbf{d}) &= \sum_{x,y} (2d_{x,y} - d_{x-1,y} - d_{x+1,y})^2
\end{aligned}
$$



Relaxation

Améliorer itérativement une solution en minimisant localement l'énergie : *relaxation*



Intuitif, implantation simple, mais vitesse de convergence lente

Relaxation

Comment obtenir la formule de mise à jour ?

La différentielle de l'énergie doit être nulle

$$\begin{aligned} \frac{\partial E}{\partial d_{x,y}} &= 2(d_{x,y} - z_{x,y}) + \\ & \quad 2\lambda(2d_{x,y} - d_{x-1,y} - d_{x+1,y}) = 0 \\ d_{x,y} &\leftarrow \frac{1}{1+2\lambda}(z_{x,y} + d_{x-1,y} + d_{x+1,y}) \end{aligned}$$

Énergie non quadratique

Comment minimiser cette fonction de coût ?

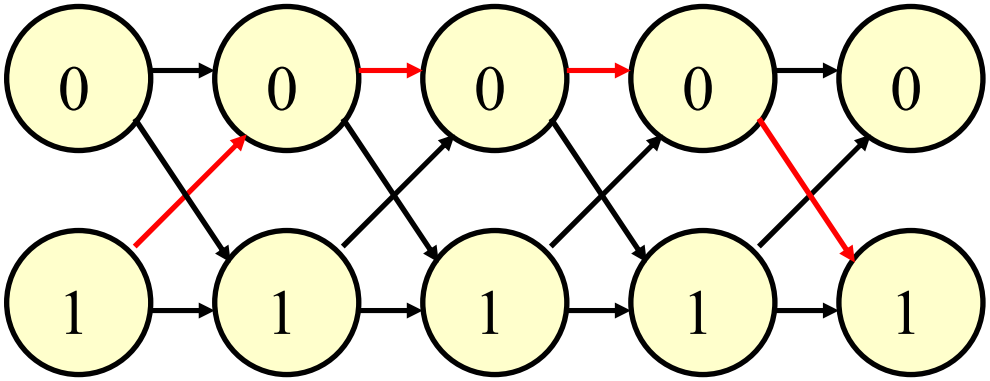
$$
\begin{aligned}
E_{\text{total}}(\mathbf{d}) &= E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d}) \\
E_{\text{data}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - z_{x,y})^2 \\
E_{\text{smoothness}}(\mathbf{d}) &= \sum_{x,y} |d_{x,y} - d_{x-1,y}|
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial E}{\partial d_{x,y}} &= 2(d_{x,y} - z_{x,y}) + \\
&\quad \lambda[\text{sgn}(d_{x,y} - d_{x-1,y}) + \text{sgn}(d_{x,y} - d_{x+1,y})] \\
&= 0
\end{aligned}
$$

Éspace d'optimisation discret

Et si les valeurs de *d* sont discrètes ?

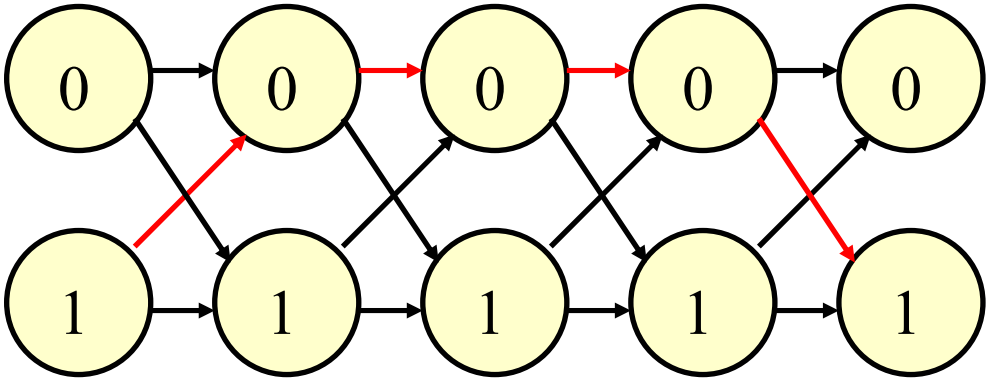
$$
\begin{aligned}
E_{\text{total}}(\mathbf{d}) &= E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d}) \\
E_{\text{data}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - z_{x,y})^2 \\
E_{\text{smoothness}}(\mathbf{d}) &= \sum_{x,y} |d_{x,y} - d_{x-1,y}|
\end{aligned}
$$



Programmation dynamique

Calculer le meilleur coût cumulatif en chaque pixel

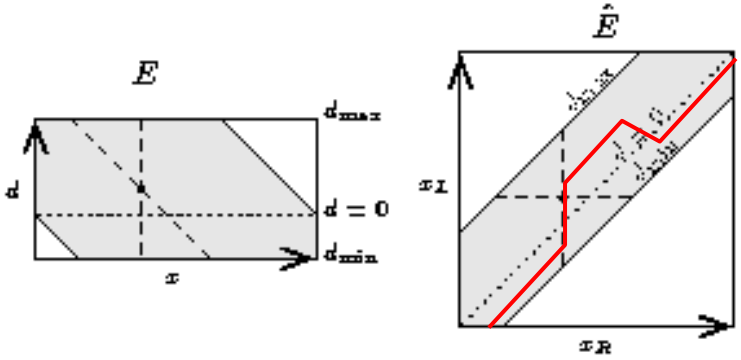
$$
\begin{aligned}
E_{\text{total}}(\mathbf{d}) &= E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d}) \\
E_{\text{data}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - z_{x,y})^2 \\
E_{\text{smoothness}}(\mathbf{d}) &= \sum_{x,y} |d_{x,y} - d_{x-1,y}|
\end{aligned}
$$



Programmation dynamique

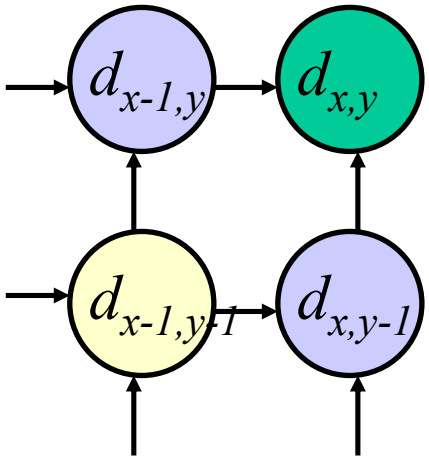
Fonction de coût 1D

$$
\begin{aligned}
E(\mathbf{d}) &= \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \sum_{x,y} E_0(x,y;d) \\
\tilde{E}(x,y,d) &= E_0(x,y;d) + \\
&\quad \min_{d'} \left(\tilde{E}(x-1,y,d') + \rho_P(d_{x,y} - d'_{x-1,y}) \right)
\end{aligned}
$$



Programmation dynamique

Est-ce que ça fonctionne aussi en 2D ?



Non : $d_{x,y-1}$ et $d_{x-1,y}$ peuvent dépendre de valeurs différentes de $d_{x-1,y-1}$

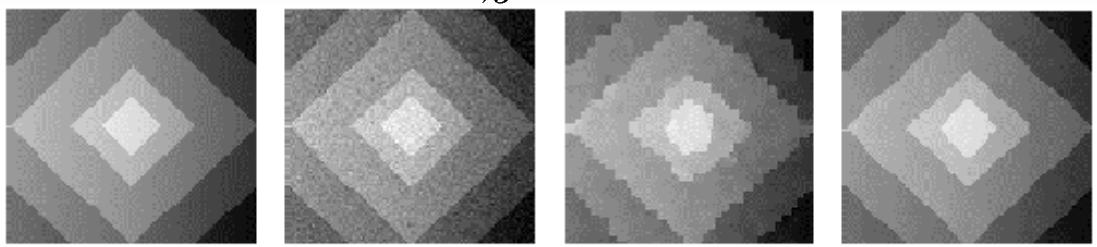
Coupes de graphe (graph cuts)

Technique de résolution d'un problème 2D

$$E_{\text{total}}(\mathbf{d}) = E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d})$$

$$E_{\text{data}}(\mathbf{d}) = \sum_{x,y} f_{x,y}(d_{x,y})$$

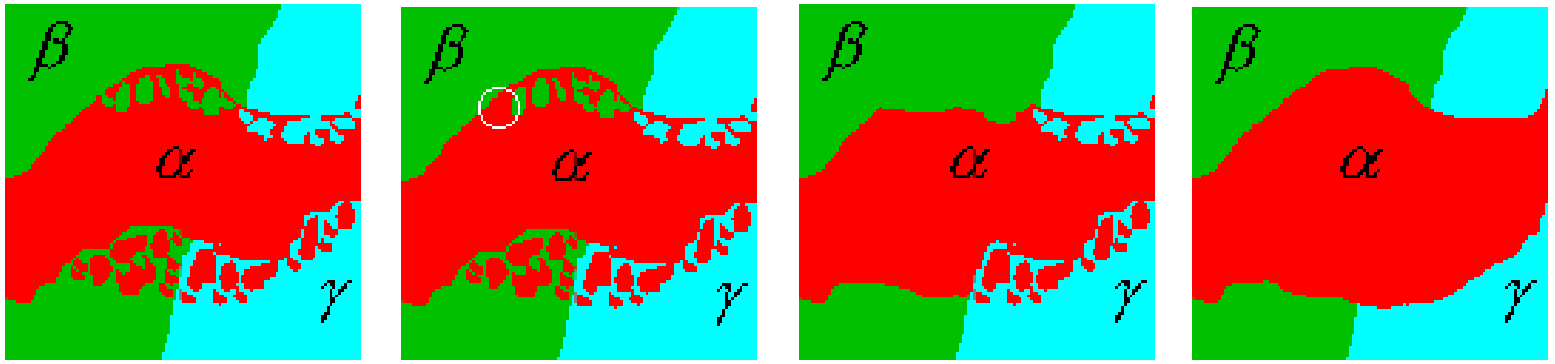
$$E_{\text{smoothness}}(\mathbf{d}) = \sum_{x,y} \rho(d_{x,y} - d_{x-1,y}) + \sum_{x,y} \rho(d_{x,y} - d_{x,y-1})$$



(a) original image (b) observed image (c) local min w.r.t. standard moves (d) local min w.r.t. α-expansion moves

Coupes de graphe (graph cuts)

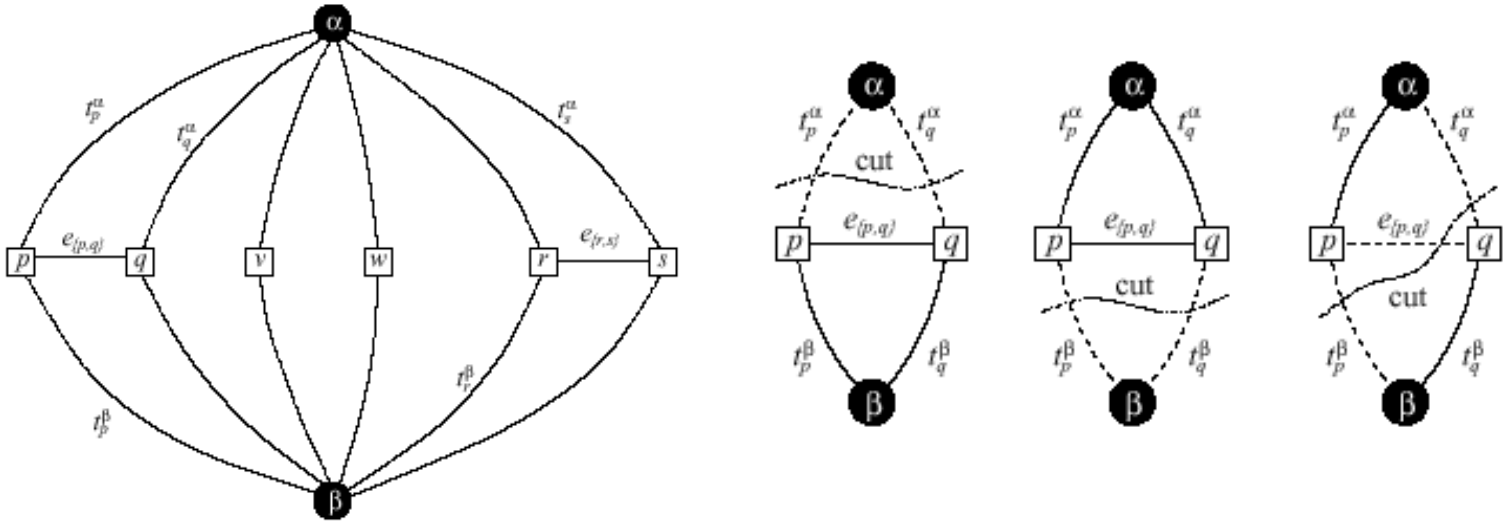
Deux types de mouvements :



(a) initial labeling (b) standard move (c) α-β-swap (d) α-expansion

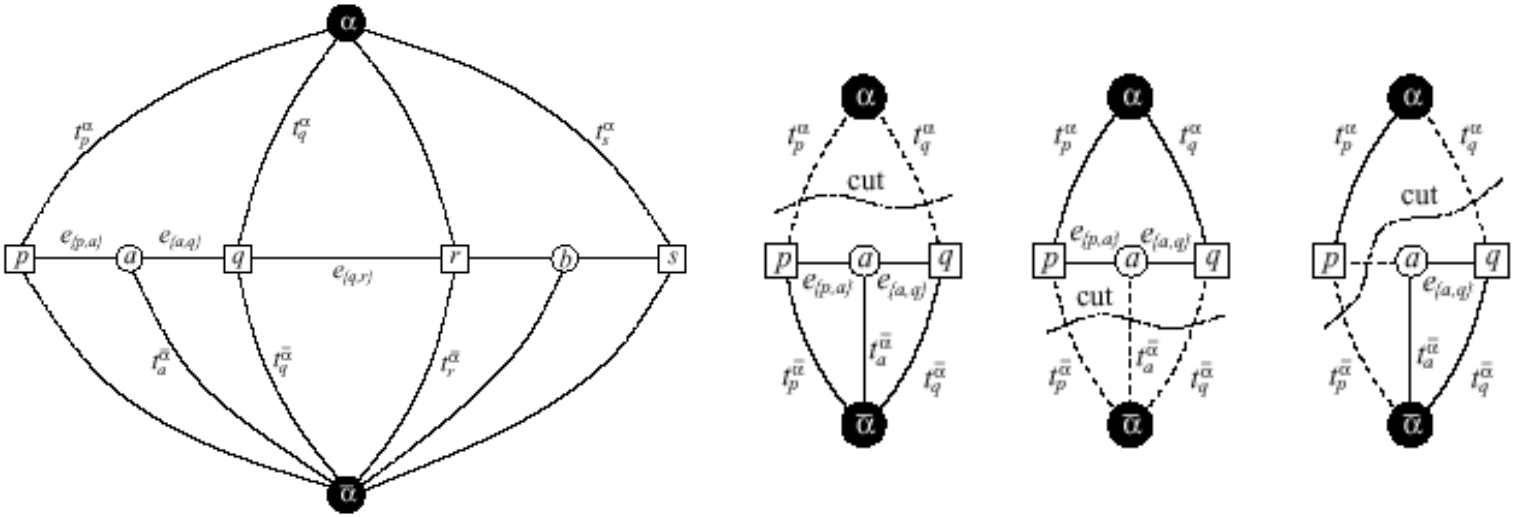
Coupes de graphe (graph cuts)

α–β swap: échanger les étiquettes α et β



Coupes de graphe (graph cuts)

α expansion: ajouter des pixels à la classe α



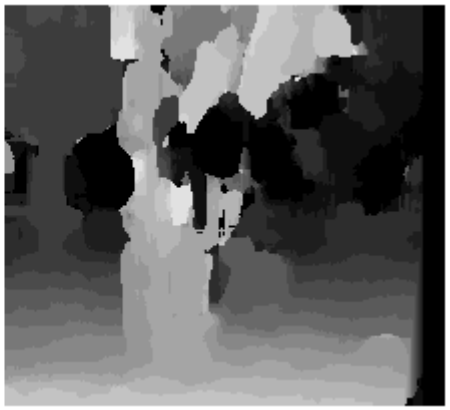
De retour à la stéréoscopie…

Taille de la fenêtre de corrélation

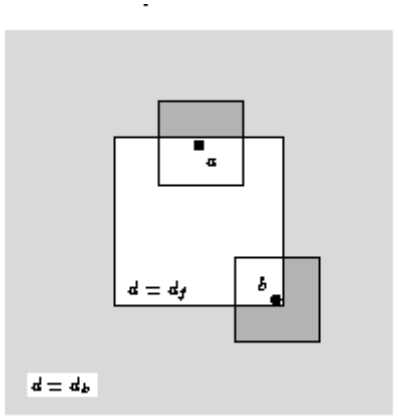
Petit voisinage : plus de détails

Grand voisinage : moins d'erreurs isolées

w = 3

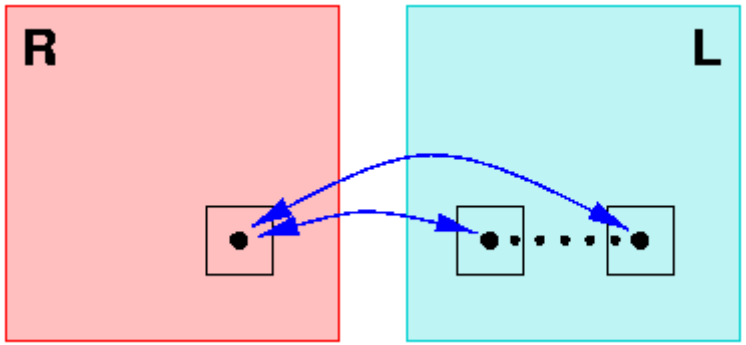


w = 20

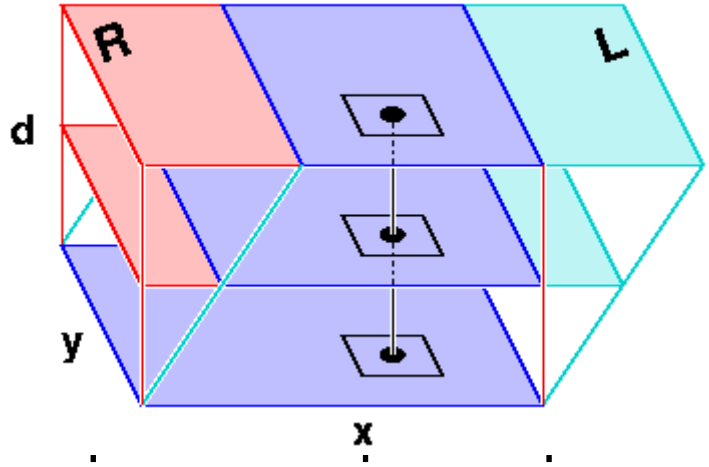


Stéréoscopie par corrélation

Réordonnancement des boucles (pixel/disparité)



pour chaque pixel,
pour chaque disparité
calculer le score



pour chaque pixel
calculer le score

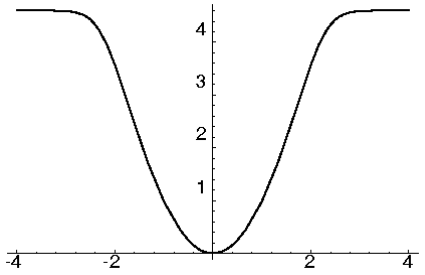
Stéréoscopie par corrélation

- 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, réflexion spéculaires, autres...



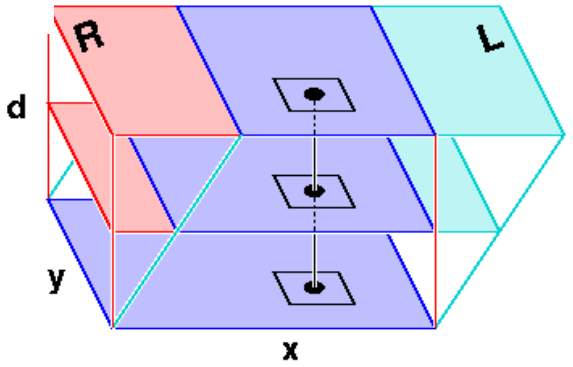
On peut également utiliser un autre critère (cosinus)

Stéréoscopie par corrélation

- Score = moyenne spatiale

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

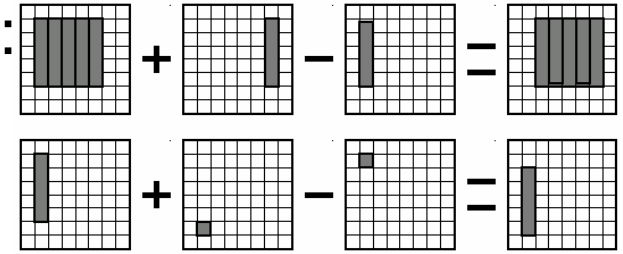
- On utilise une fenêtre carrée (facilement optimisable)



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

Stéréoscopie par corrélation

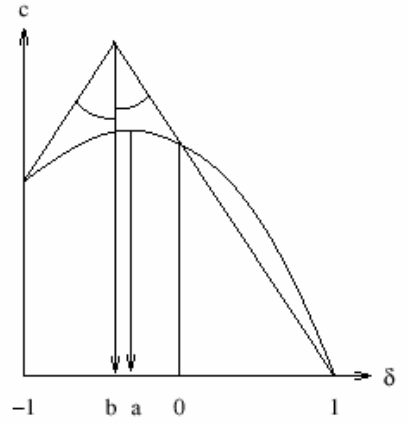
- Optimisation par fenêtre glissante : complexité indépendante de la taille de la fenêtre de corrélation



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

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

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

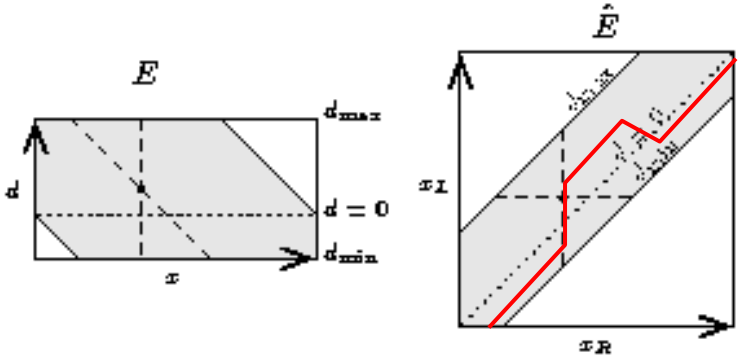


- Validation par corrélation retour (échange des images)

Programmation dynamique

Fonction de coût 1D [Intille & Bobick, IJCV 99]

$$
\begin{aligned}
E(\mathbf{d}) &= \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \sum_{x,y} E_0(x, y; d) \\
\tilde{E}(x, y, d) &= E_0(x, y; d) + \\
&\quad \min_{d'} \left(\tilde{E}(x, y, d') + \rho_P(d_{x,y} - d'_{x-,y}) \right)
\end{aligned}
$$



Programmation dynamique

Image de l'espace de disparité et chemin optimal

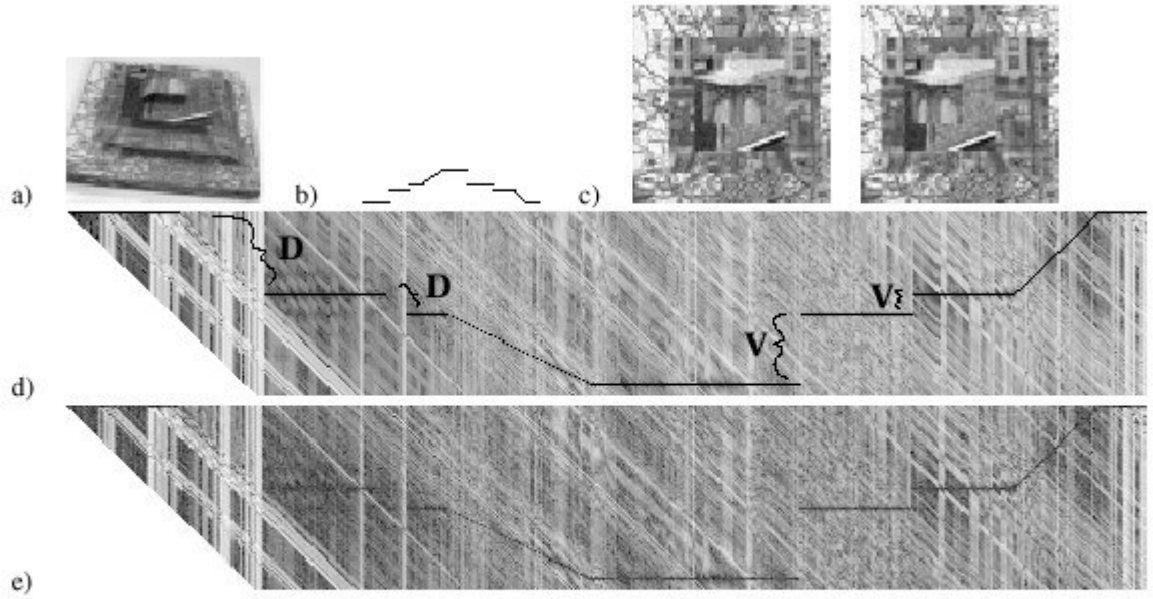


Fig. 4. This figure shows (a) a model of the stereo sloping wedding cake that we will use as a test example, (b) a depth profile through the center of the sloping wedding cake, (c) a simulated, noise-free image pair of the cake, (d) the enhanced, cropped, correlation $DSI=$ representation for the image pair in (c), and (e) the enhanced, cropped, correlation DSI for a noisy sloping wedding cake (SNR = 18 dB). In (d), the regions labeled "D" mark diagonal gaps in the matching path caused by regions occluded in the left image. The regions labeled "V" mark vertical jumps in the path caused by regions occluded in the right image.

Programmation dynamique

Image

Coupes de graphe

Programmation dynamique

(artefacts horiz.)

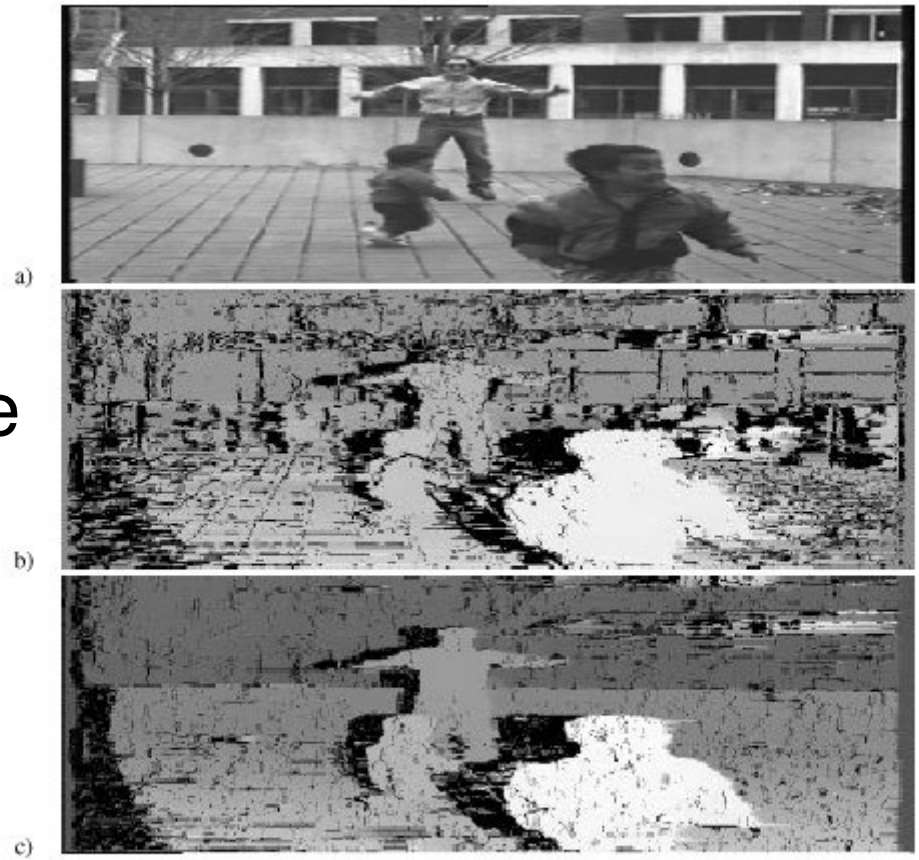


Fig. 12. Results of two stereo algorithms on Figure 1. (a) Original left image. (b) Cox *et al.* algorithm[14], and (c) the algorithm described in this paper.

Coupes de graphe (graph cuts)

α–β swap

α expansion

la pénalité de lissage dépend de la présence de contours

calcul des meilleurs appariements pour des valeurs entières de la disparité

Inférence bayésienne

Formulation statistique du problème

Modèle a priori $p_P(\boldsymbol{d})$

Modèle de mesure $p_M(\mathrm{I_L}, \mathrm{I_R} | \boldsymbol{d})$

Modèle a posteriori (Th. de Bayes)

$$p_M(\boldsymbol{d} \mid \mathrm{I_L}, \mathrm{I_R}) \propto p_P(\boldsymbol{d})\, p_M(\mathrm{I_L}, \mathrm{I_R} | \boldsymbol{d})$$

Maximum a posteriori (MAP) :

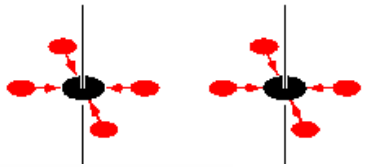
maximiser $p_M(\boldsymbol{d} \mid \mathrm{I_L}, \mathrm{I_R})$

Champs de Markov

Distribution de probabilité du champ de disparité $d(x,y)$ ne dépend que des voisins

$$p_P(d_{x,y}|\mathbf{d}) = p_P(d_{x,y}|\{d_{x',y'}, (x',y') \in \mathcal{N}(x,y)\})$$

$$p_P(\mathbf{d}) = \frac{1}{Z_P} e^{-E_P(\mathbf{d})}$$



$$E_P(\mathbf{d}) = \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \rho_P(d_{x,y+1} - d_{x,y})$$

Permet de gérer le *lissage* ou la *cohérence* du champ

Modèle de mesure

Vraisemblance de la correspondance d'intensité

$$p_M(I_L, I_R|\mathbf{d}) = \frac{1}{Z_M} e^{-E_0(x,y;d)}$$

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

Correspond à un bruit gaussien si ρ quadratique

Estimation MAP

Maximiser la vraisemblance à posteriori

$$
\begin{aligned}
E(\mathbf{d}) &= -\log p(\mathbf{d}|I_L, I_R) \\
&= \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \rho_P(d_{x,y+1} - d_{x,y}) \\
&\quad + \sum_{x,y} \rho_M(I_L(x + d_{x,y}, y) - I_R(x, y))
\end{aligned}
$$

Équivalent à une *régularisation* (minimisation d'énergie avec contraintes de lissage)

Pourquoi l'estimation bayésienne ?

Principe déterminant la fonction de coût

Modèle explicite du bruit et de la connaissance a priori

Grande variété d'algorithmes d'optimisation :

- descente de gradient (minimisation locale)
- optimisation stochastique (échantillonnage de Gibbs)
- approximation du champ moyen
- théorie des graphes (déterministe) [Zabih]

Approximation du champ moyen

Règle de diffusion non-linéaire bayésienne :

- mettre à jour la distribution de probabilité des disparités en supposant que les distributions des voisins sont indépendantes (vrai en 1D)

Équivalent à chercher la meilleure approximation *factorisée*

$$P(\mathbf{d}|I_L,I_R) \sim Q(\mathbf{d}) = \prod_i Q_i(d_i)$$

Approximation du champ moyen

log estimée MAP

$$-\log P(\mathbf{d}|I_L,I_R) = \sum_{ij} E_{ij}(d_i,d_j) + \sum_i E_i(d_i)$$

$$= \sum_{ij} \mathbf{s}_i \mathrm{A}_{ij} \mathbf{s}_i + \sum_i \mathrm{b}_i \mathbf{s}_i$$

divergence de Kullback-Leibler

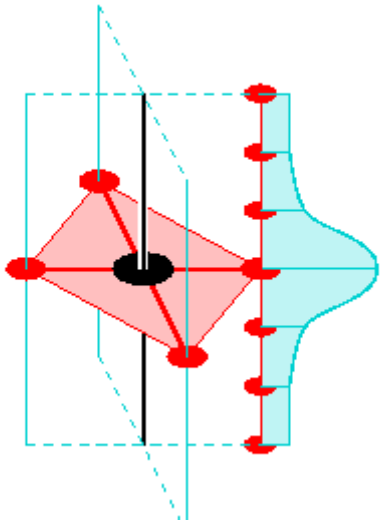
$$D_{KL} = H(Q) - \sum_{\mathbf{d}} Q(\mathbf{d}) \log P(\mathbf{d})$$

$$= \sum_{ik} q_{ik} \log q_{ik} + \sum_{ij} \mathbf{s}_i \mathrm{A}_{ij} \mathbf{s}_i + \sum_i \mathrm{b}_i \mathbf{s}_i$$

Approximation du champ moyen

minimiser la divergence K-L avec

$$\sum_k q_{ik} = 1$$



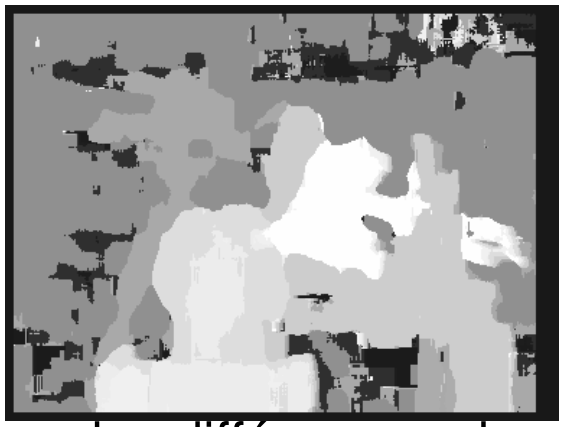
règle de mise à jour :

$$q_{ik} \propto \exp[- (\sum_j \mathbf{a}_{ij}^k \mathbf{q}_j + b_{ik})]$$

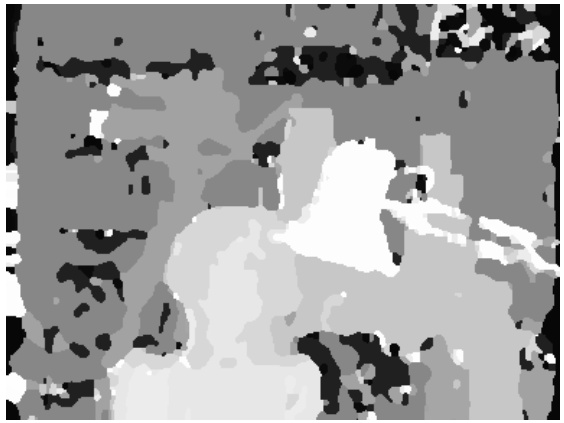
$$= \exp[- (\sum_{jl} E_{ij}(d_i=k,d_j=l) p(d_j=l) + E_i(d_i=k))]$$

Cartes de disparité

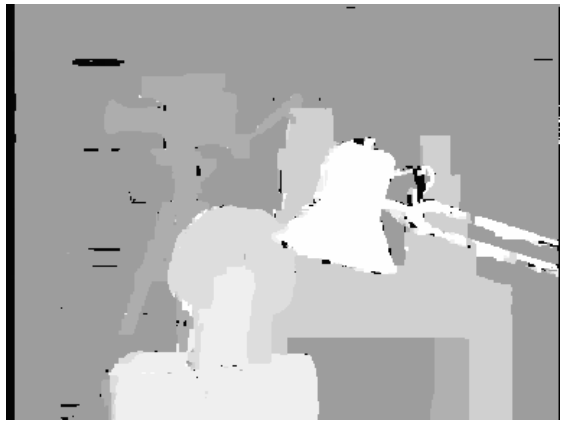
Image d'entrée



Somme des différences absolues



Champ moyen



Coupes de graphe

Stéréo avec diffusion non-linéaire

Avantages:

- fonctionne très bien dans dans régions sans occlusion

Inconvénients :

- limité à deux images (pas vraiment)
- se trompe dans les zones d'occlusion
- ne gère pas les *pixels mélangés*

Bibliographie

Y. Boykov, O. Veksler, and Ramin Zabih, Fast Approximate Energy Minimization via Graph Cuts, Unpublished manuscript, 2000.

A.F. Bobick and S.S. Intille. Large occlusion stereo. International Journal of Computer Vision, 33(3), September 1999. pp. 181-200

D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. International Journal of Computer Vision, 28(2):155-174, July 1998

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.

G. M. Nielson, Scattered Data Modeling, IEEE Computer Graphics and Applications, 13(1), January 1993, pp. 60-70.