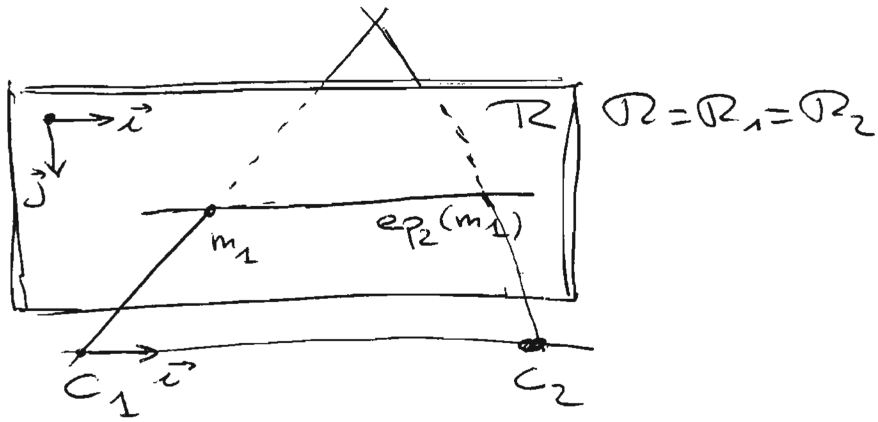
Séance 4
Rectification / Stéréo binoculaire / Graph Cuts

1) Rectification

On se place ici dans le cas de deux caméras de matrices $P_i = (A_i | b_i)$ connues

1.1) Configuration canonique

Supposons que les deux caméras partagent la même rétine, le même repère sur la rétine, et la direction $\overrightarrow{C_1C_2}$ comme horizontale de ce repère ($\overrightarrow{C_1C_2}$ est donc une des directions de la rétine).
<u>Les épipolaires sont alors les horizontales !</u>

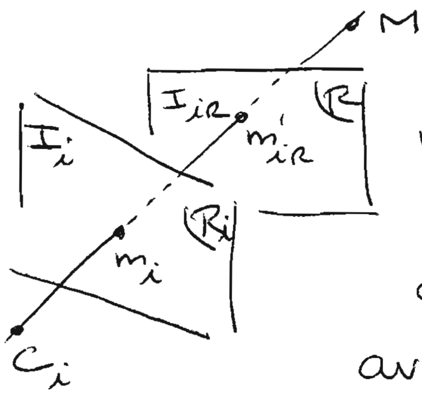


Une telle configuration est très confortable :
- (u_1, v_1) est associé à (u_2, v_1)
- on appelle $\boxed{d = u_2 - u_1 \text{ la disparité}}$
- apparier les images revient à estimer $d(u_1, v_1)$
- les fenêtres de corrélation se correspondent davantage.

On va voir qu'il est toujours possible (sauf cas dégénérés) de se ramener <u>a postériori</u> à ce cas canonique $\longrightarrow$ c'est la <u>rectification</u>

1.2) Principe

Il suffit de remarquer qu'on peut à postériori calculer l'image qui aurait été obtenue en changeant la rétine d'un appareil, et ce, sans connaître la position 3D des points



Le point M a donné naissance à $I_i(m_i)$ dans R_i. Avec R au lieu de R_i il aurait généré $\boxed{I_{iR}(m_{iR}) = I_i(m_i)}$

avec $\boxed{m_{iR} = (C_i m_i) \cap R}$

1.3) Calcul (NB : les points 3D sont ici en coord. non homogènes)

a) Plan R

On choisit pour R le plan (i) contenant $\overrightarrow{C_1C_2}$ et la direction $R_1 \cap R_2$ et (ii) passant par l'origine :

- $\vec{i} = \dfrac{\overrightarrow{C_1C_2}}{\|\overrightarrow{C_1C_2}\|}$ (Rappel : $C_i = -A_i^{-1} b_i$)
- $\vec{n_i}$ = 3ème ligne de A_i (normales à R_i)
- $\vec{j}' = \vec{n_1} \wedge \vec{n_2}$
- $\vec{k} = \vec{i} \wedge \vec{j}' / \|\vec{i} \wedge \vec{j}'\|$
- $\vec{j} = \vec{k} \wedge \vec{i}$

R est alors le plan $\{M \cdot \vec{k} = 0\}$

b) Passage de $m_i = (u_i\ v_i\ 1)^T$ à $m_{iR} = (\alpha, \beta)$ dans R

Il suffit d'intersecter $(C_i m_i)$ avec R :

résoudre $\begin{cases} M = \lambda A_i^{-1} m_i + C_i \\ M \cdot \vec{k} = 0 \end{cases}$ en λ, puis en M

Alors $\boxed{\alpha = M \cdot \vec{i},\ \beta = M \cdot \vec{j}}$

c) Projections

En réalité : (i) pour chaque image, on ramène (α,β) à des intervalles $[0,w]\times[0,h]$ (si les images initiales sont de dimension $w\times h$)

(ii) on remplit I_{iR} à partir de I_i. Il faut savoir passer de m_{iR} à m_i

Au total :

* Pour $i=1,2$
  - calculer $m_{iR}=(\alpha,\beta)$ pour $m_i=(0,0)\,(0,h)\,(w,0)$ et (w,h)
  - mémoriser les valeurs extrêmes $\alpha^i_{min}\ \alpha^i_{max}\ \beta^i_{min}\ \beta^i_{max}$
* $\beta_{min}=\min(\beta^1_{min}\,\beta^2_{min})$, $\beta_{max}=\max(\beta^1_{max}\,\beta^2_{max})$

  (NB : intervalle commun pour β)
* Pour $i=1,2$

  Pour $u_R=0\nearrow w$ et $v_R=0\nearrow h$
  - $\begin{cases}\alpha = \alpha^i_{min}+(\alpha^i_{max}-\alpha^i_{min})\dfrac{u_R}{w}\\ \beta=\beta_{min}+(\beta_{max}-\beta_{min})\dfrac{v_R}{h}\end{cases}$
  - $M=\alpha\vec{\imath}+\beta\vec{\jmath}$ (NB : point 3D !)
  - $m_i = A_iM+b_i=(u_i v_i w_i)^T$
  - $I_{iR}(u_R,v_R)\leftarrow I_i\left(\dfrac{u_i}{w_i},\dfrac{v_i}{w_i}\right)$

  (NB : - interpoler dans I_i
  - attention : on peut tomber en dehors de l'image)

D'où le schéma pour obtenir $d^* = \arg\min_d E(d)$

- $\forall d,\ \tilde{E}(0,d) = g(0,d)$
- $\forall u = 1 \nearrow u_{max}$

  $\forall d_u = 0 \nearrow d_{max}$

  $$\tilde{d}(u,d_u) = \arg\min_{d_v} \left(\tilde{E}(u-1,d_v) + f(d_u - d_v) \right)$$

  $$\tilde{E}(u,d_u) = g(u,d_u) + \tilde{E}(u-1,\tilde{d}(u,d_u)) + f(d_u - \tilde{d}(u,d_u))$$

- $d^*(u_{max}) = \arg\min_{d_u} \hat{E}(u_{max}, d_u)$

  $\forall u = u_{max}-1 \searrow 0,\quad d^*(u) = \hat{d}(u+1, d^*(u+1))$

→ ~~méthode souffrant du manque de lissage en v !~~

c) Croissance de régions

- Même idée que pour le seuillage par hystérésis dans le cas de l'extraction de contours. Deux seuils $s_1 > s_2$

(i) Extraction de graines :

- Points de Harris de I_1 associés sans ambiguïté avec I_2 :

$$\begin{cases} (u_1, v_1) \in \text{Harris}(I_1) \\ (u_2, v_1) \text{ et } (u_2', v_1) \text{ les deux points de } I_2 \text{ corrélant le mieux} \\ NCC((u_1,v_1)(u_2,v_1)) > s_1 \text{ et } NCC(u_2',v_1)/NCC(u_2,v_1) < 0.8 \end{cases}$$

→ pour ces "graines", on assigne $d(u_1,v_1) = u_2 - u_1$

(ii) Croissance :

- avec une structure de données adéquate, pour chaque voisin q d'un point p avec $d(p)$ connue, $d(q)$ non assignée :
  - essayer pour $d(q)$ les valeurs $d(p)-1$, $d(p)$, $d(p)+1$
  - garder la meilleure et seulement si $NCC > s_2$

2) Graph Cuts

2.1) Coupes minimales

$G = (V, E)$ graphe orienté pondéré
s, t deux sommets particuliers de V (source et puits)
$w(p,q)$ poids de l'arête $(p,q) \in E$ ($w \geq 0$)

- une s,t-coupe C est une partition de V en deux sous ensembles S et T avec $s \in S$, $t \in T$
- coût de la coupe : $\boxed{c(C) = \sum_{\substack{p \in S \\ q \in T \\ (p,q) \in E}} w(p,q)}$

NB : on compte les arêtes de S vers T

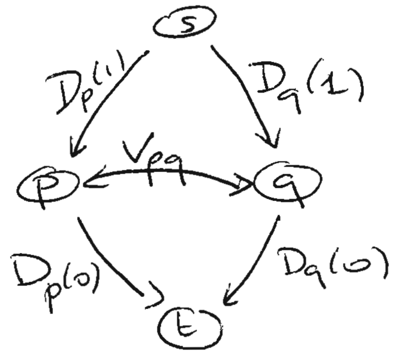
→ Il existe des algorithmes polynomiaux pour calculer la coupe de coût minimal. Ils se ramènent à calculer le flot maximal de s vers t en identifiant $w(p,q)$ à des capacités (Ford-Fulkerson). Leur complexité au pire est en $O(n^3)$ mais linéaire en pratique, en particulier sur les graphes utilisés en computer vision (algorithme des "GraphCut" [Boykov et al.])

2.2) Segmentation binaire

Le problème des régions actives peut se modéliser comme l'étiquetage des pixels p à $f_p = 0$ ou 1 (0 = intérieur, 1 = extérieur) qui minimise

$$\boxed{\begin{array}{l} E(f) = \sum_p D_p(f_p) + \sum_{p,q \text{ voisins}} V_{pq} \, \mathbb{1}\{f_p \neq f_q\} \\ \text{avec } D_p(0) = g_i(p),\ D_p(1) = g_e(p),\quad V_{pq} = g\left(\frac{p+q}{2}\right) \end{array}}$$

Or, minimiser $E(f)$ revient à trouver la min-cut dans ce graphe :



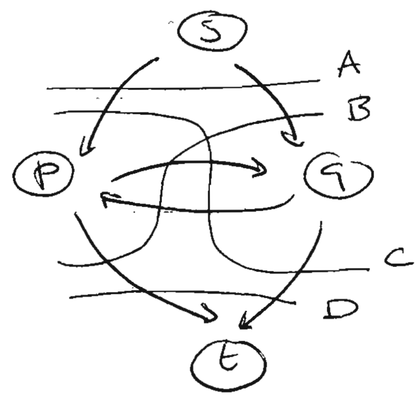
puis à faire : $f_p = 0$ si $p \in S$, $f_p = 1$ si $p \in T$

→ minimum global et rapide pour les régions actives ! (NB : le terme $\int g \, d\sigma$ n'est que grossièrement approché. On peut l'approcher mieux avec des arcs supplémentaires)

2.3) Autres cas

a) Il est possible de construire un graphe pour un cas plus général $\boxed{E(f) = \sum D_p(f_p) + \sum_{pq} V_{pq}(f_p, f_q)}$ à condition que V_{pq} soit submodulaire :

$$\boxed{V_{pq}(0,0) + V_{pq}(1,1) \leq V_{pq}(1,0) + V_{pq}(0,1)} \quad (1)$$

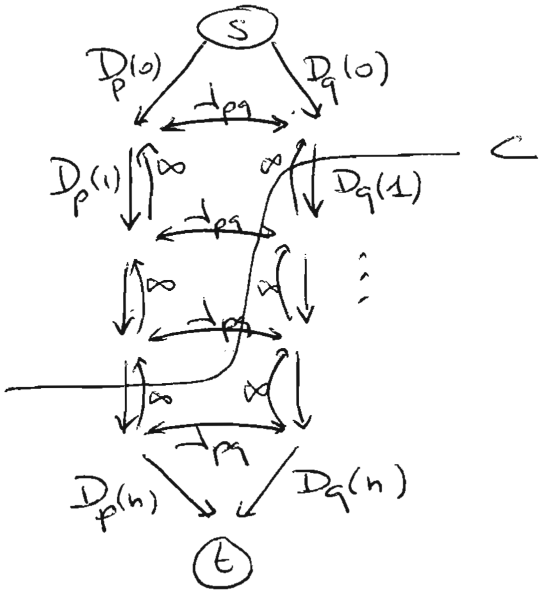


Il faut trouver les bons poids pour que les coupes A, B, C et D coûtent les valeurs d'énergie adéquates

→ ça n'est possible avec des poids ≥ 0 que si on a (1)

b) Cas où les étiquettes sont multiples ($f_p \in \{0, 1, \cdots, n\}$)
et où $V_{pq}(f_p, f_q) = \lambda_{pq} |f_p - f_q|$

Il est encore possible, en rajoutant une dimension, de construire un graphe donnant l'optimum global



- f_p est attribué à la "hauteur" de la coupe dans la "colonne" de p
- Pour éviter qu'une colonne soit coupée plusieurs fois, on ajoute des arcs de poids ∞

NB : Le terme de régularisation en $|f_p - f_q|$ est essentiel. En particulier, il n'y a pas de solution pour $1\!\!1_{\{f_p \neq f_q\}}$

c) Cas général : $f_p \in \mathcal{L} = \{\alpha, \beta, \cdots\}$ fini

Si $V_{pq}(\cdot,\cdot)$ est une métrique alors il est possible d'utiliser la méthode de l'α-expansion. Pour $\neq$ tes valeurs de $\alpha \in \mathcal{L}$, on optimise progressivement f_p en un nouveau f'_p tel que $\boxed{f'_p = f_p \text{ ou } f'_p = \alpha}$. Le choix du f'_p optimum est un choix binaire effectué par graph-cuts. Après convergence, on obtient un minimum local.

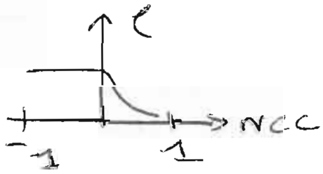
(NB : chaque itération est résolue de façon globalement optimale)

~~(NB : si $V_{pq}(\cdot,\cdot)$ est une semi-métrique (i.e. ne vérifie pas l'inégalité triangulaire), il est possible de procéder par α-β swap (échanger itérativement 2 étiquettes))~~

3) Application à la stéréovision

On suppose que les images sont rectifiées. Il n'y a plus qu'à estimer une disparité $d(u,v)$ optimale.

$$\left\{\begin{array}{l} p=(u,v) \\ d_p \in \mathcal{L} = \{d_{min}, \ldots, d_{max}\} \\ p,q \text{ voisins} \Leftrightarrow \text{voisins dans l'image} \\ D_p(d_p) = \rho(NCC(I_1, p, I_2, p+(d_p,0))) \\ \quad \text{avec } \rho \text{ fn} > 0 \text{ décroissante de la corrélation NCC} \\ V_{pq}(d_p, d_q) = \lambda |d_p - d_q| \end{array}\right.$$



→ [application directe de 2.3.b
(ou de 2.3.c si $V_{pq} = \lambda \mathbb{1}_{\{d_p \neq d_q\}}$)