Fiche BIMA

Charles Vin

Décembre 2022

Table des matières

Gausienne 2D : $\sigma_f = \frac{1}{\sigma_s \pi}$

$$\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}}.$$

Changement d'échelle : ?

Rotation :

$$P \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}.$$

Changement de variable 2D :

$$f(x,y) = (f_1(u), f_2(t))$$
$$\begin{cases} x = f_1(u) \\ y = f_2(t) \end{cases}$$
$$dxdy = \det(Jacobienne(f))$$

Pour inverser les variables : penser au matrice !

Généralité :

- Dynamique d'une image : $L = (k_{max} - k_{min}) + 1$
- Changer la dynamique $[k_{min}; k_{max}] \rightarrow [I_1, I_2]$: $f(k) = \frac{k - k_{min}}{k_{max} - k_{min}} * (I_2 - I_1) + I_1$
- Inverser une image : $L - p_i$
- Histogramme : compter les pixels de chaque couleurs
- Flat hist : $k' = Int(\frac{k_{max} - k_{min}}{N * M} H_c(k))$
- Gray value profile : line plot de la ligne de l'image

1 Digitalisation and subsampling

SIGNAL pour moi sorry

2 Filtrage

Pense à retourner h pour la convolution!!! Diamond formula

$$f(t)\delta(t-t_0) = f(t_0)\delta(t-t_0).$$

$$f(t) \star \delta(t-\tau) = f(t-\tau).$$

3 Edge Detection with filtering

— Un bord dans une image peut ressembler à une marche d'escalier ou à une rampe : il est plus ou moins nette

— On regarde la direction du gradient : $\|\nabla f\| = \sqrt{(\frac{\delta f}{\delta x})^2 + (\frac{\delta f}{\delta y})^2}$ que l'ont normalise $\frac{\nabla f}{\|\nabla f\|}$ pour obtenir un vecteur unitaire

— Par une méthode mathématique obscure nommée différence finis, on peut approximer les dérivés des images pas une convolution

3.1 Sobel Edge filter

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}.$$

$$G_y = G_x^T.$$

— la réponse impulsionnel de Sobel est en faite composé d'une matrice qui approxime la gaussienne et la matrice de dérivation horizontale $\begin{pmatrix} 1 & 0 & -1 \end{pmatrix}$

— $\|G\| = \sqrt{G_x^2 + G_y^2}$ cette norme est plus forte au niveau des contours (car dérivé d'un escalier $= +\infty$)

3.2 Second order

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \text{ ou } \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

— Ici on regarde quand la dérivée seconde s'annule pour trouver le max de la dérivé

— On utilise un laplacien pour approximer la matrice hessienne $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$

— Detecter les passages par zéros :
  — Fenetre 3x3 $\rightarrow$ max et min
  — zéro crossing = $max > 0, min < 0, max - min > S$

— Plus précis et moins sensible à la threshold que gradient

— **Pas** invariant par rotation!

— Thick edge

— bruit ++ $\rightarrow$ filtrage nécessaire $\rightarrow$ **On peut combiner les deux en une convolution** avec le laplacien de la gaussienne 2D

3.3 Approche pyramidale

Filtre gaussien $\rightarrow$ subsample 2 $\rightarrow$ filtre $\rightarrow$...

$$\begin{aligned} & fe > 2fmax \text{ (shannon)} \\ \Leftrightarrow & fe > 2 * \frac{3}{\sigma\pi} \\ \Leftrightarrow & \frac{\pi}{6} fe > \frac{1}{\sigma} \\ \Leftrightarrow & \frac{\pi}{6} * \frac{1}{2} > \frac{1}{\sigma} \text{ (1/2 car subsample 2)} \\ \Leftrightarrow & \frac{12}{\pi} < \sigma \end{aligned}$$

3.4 Canny-Deriche

Filtre gaussien plus optimisé + implémentation récursive possible pour éviter de faire deux fois la convolution(x et y)

3.5 Post processing

- Binarization Threshold : thick edge + bruit ou missed detection $\Rightarrow$ Gaussian smoothing
- Gaussian smoothing + Threshold :
  - flou ++ = moins de bruit // thick edge (imprecise localization)
  - Flou – = bruit // bonne localisation
- Non maxima suppression
  - Arrondie sur une des 8 directions
  - Interpolation à partir des deux voisins
  - $\rightarrow$ Bord fin

4 Corner Detection

- Point critique de l'image (local extrema, saddle points) = variation dans une ou plusieurs direction $\Leftrightarrow det(Hess) = 0$. Ca c'est la detection basique mais elle est vraiment pas ouf
- $\rightarrow$ On vas donc jouer avec la matrice Hessienne

4.1 Moravec keypoint detection

4.2 Harris detector

$$R_i(\sigma) = g_\sigma \star Hess(I) = g_\sigma \star \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}.$$

- Instinctivement : Dans le voisinage du filtre gaussien, on regarde pour des variations dans deux directions
- Les valeurs propre de la matrice Hessien indique les changement dans leurs direction respective x et y
- Mauvais sur les changements d'échelles
  - Multiscale Harris detector : On peut le stabiliser en changeant l'échelle σ du filtre gaussien en fonction de la scale de??? Puis en choisissant la meilleure valeur de R comme coin **pas compris** Bref on change la scale et on choisis le meilleur coin sur toutes les scales testées
  - Pareil pour Harris-Laplace + harris lapalce = flou gaussien + laplace = on peut combiner comme vu avant, je dirais même plus on peut approximer la combinaison pas une différence de gaussienne
- bien pour les scene texturé

Preuve : On part du détecteur de Moravec : il cherche des changements de variation dans les 8 directions possibles : coin = changement dans deux directions = $E_{u,v}(x,y)$ grands.
Forcément une image full bruit aura beaucoup de variation, donc pour éviter ça on vas lisser l'image avec un filtre gaussien.

Donc on cherche les endroits où $E_{u,v}(x,y)$ est grand aka on veux maximiser la fonctions et trouver les points critique max.

Point mathématique : Pour savoir la nature d'un point (critique normalement mais bon je sais pas où on annule le gradient) on regarde si la matrice Hessienne est définie positive/négative. Pour ça on regarde les valeurs propre, si elle sont strictement négative, c'est gagné on a trouvé un maximum local ! Par chance, on a pas besoin de diagonaliser à chaque fois car le determinant ($= \lambda_1\lambda_2$) et la trace ($= \lambda_1+\lambda_2$) sont invariant par changement de base et suffise pour déterminer le signe des valeurs propres.

- $\det Hess > 0$ Valeur propre de même signe
  - $Trace > 0 \rightarrow$ valeur propre positive $\rightarrow$ minimum local
  - $Trace < 0 \rightarrow$ valeur propre negative $\rightarrow$ maximum local

Finalement en posant $R = det - k * trace^2$ on reproduit le même principe. Ce critère permet de comparer λ_1 et λ_2.

- If λ_1 and λ_2 are approximately equal, we note $\lambda_1 \approx \lambda_2 = \lambda$ and get

  $$R(x,y) \approx \lambda^2 - k(4\lambda^2) = \lambda^2(1-4k).$$

  Since in practice k is chosen as a small value, i.e. $k << 1$, we get $R(x,y) \approx \lambda^2$. Then :
  - If $\lambda \rightarrow 0$, this means that the derivative of I are close to 0, i.e. the region is flat (homogenous gray levels), and $R \rightarrow 0$
  - If $\lambda > 0$ then $R > 0$, and we have locally a corner.
- If $\lambda_1 >> \lambda_2$ (or the reverse) then $R(x,y) \approx \lambda_1\lambda_2 - k\lambda_1^2 = \lambda_1^2(\frac{\lambda_2}{\lambda_1} - k) \approx -k\lambda_1^2$ If the pixel is an edge, it means that the variations of I are in one direction only, i.e. $\lambda_1 >> \lambda_2$, and we get $R < 0$.

$\square$