

Base du traitement d'image

Charles Vin

S1-2022

Nouveau cours du 13/09

<https://www-master.ufr-info-p6.jussieu.fr/parcours/ima/bima/>

TP : correction avec échantillonnage random Une semaine pour les faire, il veut qu'on aille plus loin que les question? 40% CC et 50% Exam, pas de DS1

Prise en compte de la perception dans le traitement d'image (sinon on ferait du traitement de signal) : exemple des illusions d'optiques

0.1 Encodage d'image

0.1.1 RBG

RGB = un cube

Limite de RGB :

- Les 3 canaux sont très corrélé → Information redondante
- Problème pour utiliser des distances euclidienne : certaine couleurs sont proche en distance mais sont pas du tout pareil

Extension :

- D'autre représentation : ACP
- Utiliser des model plus proche de l'humain : HSV

0.1.2 HSV

HSV = un cone

- Value : Moyenne de RGB
- Hue : De la trigo comme c'est un cercle
- Saturation : De la trigo comme c'est un cercle

Brightness : juste la moyenne des valeur de l'image

Contrast : C'est lié à la distance entre le min et le max qu'on définit pour le niveau de gris. On regarde la distance entre les valeur min et max, on peut utiliser l'écart type.

3 niveaux d'analyse/de compréhension des images :

- Low : image => image
 - Compression
 - Restauration (retirer le bruit)
 - Filtrage (trouver uniquement les contours)
 - Segmentation (un pixel = un label)
- Mid : image => Attributes
- High : image => understanding (semantic description)

Proche du processing fait par les réseaux de neurone

Nouveau cours du 20/09

1 Basic image transformations

VOIR LE DIAPO AVANT

- Transformation affine
- Translation

- Change of scale
- Rotation
- Linear transformation

Cordonnée homogène :

- On ajoute une nouvelle coordonnée qui vaut 1 partout. Ainsi maintenant une translation dans \mathbb{R}^2 peuvent s'exprimer comme une opération linéaire dans \mathbb{R}^3
- Ça permet d'être rapide comme les multiplications de matrice sont cablés dans les GPUs.
- Problème : En pivotant une image on crée des trous dans l'image (un losange dans un carré) et les pixels ne sont plus carrés. Deux solutions :
 - Direct transformation : pixels coordinates in the output image are determined from pixels coordinates in the input image
→ can generate missing data or superposition.
 - Inverse transformation : pixels coordinates in the input image are determined from pixels coordinates in the output image
→ can generate superposition and also missing data (due to bounded spatial domain).
- IMAGE DIAPO 12
- Exemple DIAPO 13

Méthode d'interpolation : Two examples of basic interpolation methods

- Nearest neighbor : pixel value is given by the value of the nearest neighboring pixel
- Bilinear interpolation : pixel value is determined from the 4 nearest neighboring pixels using a bilinear interpolation (inconveniant : lisse l'image, la rend flou)

Many other interpolation methods : B-splines, Hermite interpolation polynomials, ... IMAGE DIAPO 14

Note. Globalement deux opération en traitement d'image : Soit on moyenne, soit on intègre. On verra dans le TD3.

Application des transformations géométrique :

- Suivre les objets sur deux caméras différente (un passant qui passe d'une caméra à l'autre) Belle image dans le diapo 17. Tout peut se passer avec une matrice 3×3 .
- Faire le lien entre une carte dessinée et une image satellite. On peut détecter les points d'intérêt, ou en connaissant précisément la longitude et la latitude de chaque pixel
- Application médical : Superposition de plusieurs modalités provenant de plusieurs capteurs. Même reconstruire en 3D.
- Compression vidéo : principe on essaye de prédire l'image suivant à partir de la précédente avec une fonction. Bref on regarde uniquement les pixels qui se déplacent.

1.1 Operation between images

Application :

- Soustraction et réduction de bruit : Soit deux images $I, I^n, I - I^n$ permet de détecter le bruit. On obtient alors que des pixels noirs lorsqu'il n'y a pas de différence. Le reste est du bruit.
→ Permet également de trouver les changements dans le temps, détection de mouvement. (image diapo 31-32)

1.2 Image thresholding

- Thresholding : reduction of image values to few levels of intensity
- Binarization : image values are reduced to two intensity levels
- Binary thresholding, defined by :

$$k' = \begin{cases} k_1 & \text{si } k \leq S \\ k_2 & \text{si } k \geq S \end{cases}$$

with k_1, k_2 and S (threshold) are levels of intensity

→ Highlights regions but does not enhance the image

1.3 Image enhancement

Définition 1.1. —

Trois catégories :

- Pixel level enhancement :
 - the image brightness or contrast is modified
 - no spatial information, only radiometric value of the visited pixel is considered
 - Local enhancement : Prend en compte le voisinage
 - Enhancement in the frequency domain : on prend l'image, on quitte la représentation spatiale et on bascule dans le domaine fréquentiel pour faire des changements avant de revenir dans domaine spatial.
- Application :
- Regions to highlight
 - Images that are too bright or too dark
 - Intensity levels should be changed in order to make some details in the image more visible
 - Modify image brightness
 - Increase contrast (see Lecture 1)
 - Pixel-level enhancement is closely related to histogram transformation
 - IMAGE COOL DIAPO 41

1.4 Histogram

Définition 1.2 (Histogram). Histogram is an array / function describing the image values (intensities / gray values / colors) distribution. Provides image-specific information, such as :

- The statistical distribution of image values
- Minimal and maximal image values, moyenne, médiane
- **No spatial information at all (exemple diapo 44)**

Formule :

$$H(k) = \text{compteur pour } k.$$

En réalité on parcourt l'image et on regarde à quel interval il appartient et on incrémente celui-ci.

Exemple 1.1. Voir l'image diapo 43.

Définition 1.3 (Normalized Histogram). Permet d'approximer la densité de la loi. Car l'histogramme normalisé n'est pas conforme à la définition d'une densité (intégrale égal à 1).

Function H_n representing the probability (occurrence frequency here) for a pixel to have a given value k

$$H_n(k) = \frac{H(k)}{N \times M}.$$

with N and M are the image dimensions. Maintenant elle est à valeur dans $\in [0, 1]$

Définition 1.4 (Cumulative Histogram). Permet d'avoir une estimation de la fonction de répartition. On fait la somme cumulative

$$H_C(k) = \sum_{i \leq k} H(i).$$

1.5 Image négative

Définition 1.5. negative of the image obtained by the negative transformation in the range of $[0, L - 1]$:

$$k' = L - 1 - k.$$

with L the dynamic range of the image (number of intensity levels)

1.6 Transformation

1.6.1 Logarithmic transformation

Définition 1.6. Low values increase, high values decrease : allows increasing the contrast in dark parts of the image.

$$k' = \log(k).$$

Améliore le contraste dans les parties les plus sombres de l'image. EXEMPLE DIAPO 51

1.6.2 Exponential transformation

Définition 1.7. Low values decrease, high values increase : allows increasing the contrast in bright parts of the image.

$$k' = e^k.$$

1.7 Opération sur les histogramme

Définition 1.8 (Histogram Translation). Changes the **brightness** of an image, leaving the contrast unchanged. The new image is brighter or darker. Useful for images having a low dynamic range.

$$k' = k + t.$$

Définition 1.9 (Affine transformation).

Définition 1.10 (Image Normalization). On ramène les valeurs entre 0 et $L - 1$.

- Let k_{min} and k_{max} be the minimal and maximal intensity levels of an image, respectively :
- Transformation :

$$k' = \frac{L - 1}{k_{max} - k_{min}}(k - k_{min}).$$

- After transformation, $k' \in [0, L - 1]$, contrast is maximal
- No loss of information (same number of intensity levels)
- Before visualization, an image is often normalized (but not necessarily)

SUPER EXEMPLE DANS LE DIAPO 59

Définition 1.11 (Linear transformation with saturation). Cette fois ci on vas étirer l'histogramme uniquement dans un intervalle donnée. Voir diapo 60 pour plus d'information.

Cette fois il y a de la perte d'information.

Définition 1.12 (Histogram equalization). On vas le coder ahah. On prend l'histogramme et on le rend plat (équidistribué).

Perte d'information mais bien pour la visualisation.

- Each intensity level is represented in the same proportion
- Regions of lower local contrast gain a higher contrast
- Global contrast increases

L'idée de la méthode : On fusionne les pixels peu représentés avec les plus représentés.

$$k' = \text{Int}\left(\frac{L - 1}{N * M} H_c(k)\right).$$

Avec

- L the image dynamic range
- N and M the image size
- $H_c(k)$ the cumulative histogram
- Int rounding to the nearest integer

Super exemple dans le diapo 64.

Note (Histogram stretching versus histogram equalization : same operation?). — Stretching : changes the bins distribution in the histogram, but not their size

- Equalization : changes the bins distribution in the histogram and their size

Note (Application). — Image mosaic : a target image and a base of small image.

- Face recognition : On suppose que les histogramme d'image faciale ont la même loi de proba.
- Segmentation : En fusionnant beaucoup beaucoup les classes on peut segmenter en 5 couleurs (exemple diapo 70). Ou on utilise des algo de clustering

Nouveau cours du 27/09

15 min de retard, before diapo 11.

En gros la transformée de Fourier permet de représenter les fonctions périodiques avec des coordonnées dans l'espace infini des fonctions périodiques. Pourquoi

- Pour écrire les signaux de manière plus compact
- Pour pouvoir les compresser
- Pour pouvoir les comparer ensuite.
- CCL : DIAPO 14

Transformée de Fourier : La seule différence c'est qu'on obtient un continuum de fréquence

Interprétation du signal : On obtient une représentation de notre fréquence en terme de fréquence (haute ou basse)

- On utilise le module $|X(f)|$ pour obtenir la quantité de la fréquence pure f présente dans le signal x .
 - La phase : L'angle de $X(f)$ vue dans ce cas comme un vecteur du plans complexe
- Le signal x peut être reconstruit à partir de sa transformé de Fourier X .

Définition 1.13 (Convolution). Une sorte de moyenne locale de g pondéré par les valeur de f . Opération commutative, distributive, associative.

Retour fourier diapo 26 Liste des propriétés :

- Linéaire
- Time scaling : Quand on multiplie un signal par un scalaire, on le stretch et il s'étend. A l'inverse sa transformé de Fourier se rétrécis
- Time shifting : Si on translate le temps, on ne change pas la transformé
- Frequency shifting : translater les fréquence, translate la transformé de Fourier
- Théorème de la convolution :
 - Convolution assez complexe à calculer $O(n^2)$
 - Avec Fourier la convolution est super plus simple à calculer $O(n \log n)$
 - Voir formule dans le diapo 26 si nécessaire

Exemple 1.2. Exemple d'une transformé de Fourier avec une fonction porte.

Exemple 1.3. Exemple d'une transformé de Fourier avec une Gaussienne → On retrouve une gaussienne! Damn c'est fou

Exemple 1.4. Exemple d'une transformé de Fourier avec une fonction de Dirac $\delta(t)$. C'est l'élément neutre de la convolution. Au final, on pondère la moyenne mobile sur un unique point → ca donne la valeur de la fonction en ce point.

Exemple 1.5. Exemple d'une transformé de Fourier de \sin et \cos . C'est relativement facile en utilisant la formule d'Euler, qui donne deux exponentiel complexe, et une exponentielle complexe c'est une fonction de Dirac avec Fourier.

Fourier en 2D Globalement la même chose sauf qu'on intègre pour chaque dimension. On obtient en couple de fréquence, et pour le plot on utilise l'intensité lumineuse pour représenter l'amplitude.

Important : la transformé de Fourier donne des information sur l'orientation des objects, leur taille, ect EXEMPLE DIAPO 53

DIAPO 54, quelque fonction très utile. Et diapo 55 illustration de ce que fait la fonction $fftshift()$

Diapo 56 : On passe au log pour améliorer la visualisation.

Quelques Applications :

- Débruiter : en supprimant les hautes fréquences
- Compression : Les petits détails sont haute fréquences donc on peut les supprimer.
- Obtenir les directions des choses ect
- Filtrer : edge detection, point d'intérêt, ...

Désavantage d'une représentation fréquentielle :

- A cause de l'invariance par translation, on a pas d'information spacial.
- Signaux non dérivable : Avec que des portes (=des contours nettes dans images), pour l'approcher avec des \sin il faut une infinité de \sin

Nouveau cours du 04/10

2 2chantillonnage

On doit échantilloné les signaux continues

- Le fenétrage : on limite le support (domaine du signal) dans le temps
-

2.1 Fenêtrage

Classiquement on multiplie pas une fonction porte.

2.1.1 1D

La multiplication par une fonction rectangle \Leftrightarrow faire la convolution dans le domaine fréquentiel.

Mais visiblement ça complique pas mal la transformé en Fourier (voir image diapo 6). On retrouve des sinus cardiaux. On verra en TD comment faire et si la taille de la fenêtre a une conséquence.

2.1.2 2D

Ici on multiplie par deux fonctions portes, ... Mais on a la même chose en 3D au final, on retrouve le sinus cardinal

Quand on prend une fenetre petite, la Fourier s'élargie fortement, ce qui peut la rendre illisible. Avec une fenetre large on tend vers une dirac (diapo 12)

2.2 Echantillonnage

On prend une mesure tous les x temps. Si fréquence trop faible, on n'arrive pas à reconstituer l'aspect continue, mais si trop grand on prend beaucoup de ressource de calcul.

2.2.1 1D

Pour faire ça on multiplie notre signal par un peigne de Dirac III, c'est une collection de Dirac.

Qu'est ce qui se passe dans le domaine fréquentiel? La TF d'un peigne de Dirac c'est une peigne de Dirac, c'est plutôt contre intuitif. Mais en gros ça rend la TF périodique.

Perte d'information Y'a-t-il une perte d'information pendant l'échantillonnage? Oui et non

- Non Si on prend un signal borné séquentiellement. C'est à dire que la transformé de Fourier a un support borné sur l'axe des abscises
- Sinon oui il y a perte

Liste des signaux band-limité

- Rect non
- sinc oui
- sin oui
- cos oui
- Dirac oui
- BIEN ÉCRIRE DIAPO

Quel fréquence d'échantillonnage? Pour pouvoir échantillonner sans perte un signal, il faut utiliser une fréquence max deux fois plus grande que la fréquence max du signal. C'est le théorème de Shannon. DIAPO 23 J'suis pas sûr de ce que j'ai écrit.

Reconstituer un signal échantillonné? On doit re-fenêtrer. PAs compris vraiment mais en gros à la fin faux pas obtenir de courbe qui se chevauche.

CCL : Si Shannon est vérifié, toute l'information est dans la TF, on peut reconstruire le signal avec la formule de reconstruction de Shannon

Aliasing Si on ne vérifie pas les dégradations on a un phénomène d'aliasing. Difficile à prédire.

Exemple 2.1 (Cas d'une fonction sin). DIAPO 30 TO 39 Ici ça illustre les deux cas (correct et incorrect).

- Avec $4f_0$, c'est suffisant pour reconstruire le signal.
- Diapo 33 : à droite on a plot la formule d'interpolation de Shannon.
- On peut prendre une fréquence d'échantillonnage plus élevée mais ça change rien.
- DIAPO 35 : fréquence trop basse \rightarrow mauvais fenêtrage
- Diapo 37 : on peut voir une divergence des points rouge d'échantillonnage.

2.2.2 2D

Pareil, toujours les condition de Shannon : bande limité, fréquence d'échantillonnage. Plus restrictif car dans les deux direction. Le peigne de Dirac est représenté par une grille d'échantillonnage.

On peut utiliser une grille cartésienne ou hexagonale (qui a certain avantage). Diapo 45

Transformé de Fourier : il dit c'est pareil. On vas répéter le spectre dans toute les direction. Théorème de Shannon : pareil dans les deux directions.

Reconstruction : toujours possible avec une plus grosse formule.

Signaux à bande limité : Réaliste? En théorie non, jsp pk, mais en pratique on peut ignorer les fréquences de bruit créée.

- En pratique ça marche bien avec les signaux lisse et stationnaire comme un ECG.
- Pour les images naturels, chaque contours représente une forte fréquence (non lisse).
- Exemple d'aliasing diapo 58 59 60

Anti-aliasing

- Filtrer toute les fréquences qu'on peut pas représenter avec un filtre passe bas à partir de la moitié de la fréquence d'échantillonnage.
- Utile lorsque qu'on downsample une image. En retirant certain pixel, on perds certaine fréquence donc on crée de l'aliasing (?). Du coup on applique également un filtre aliasing.

2.3 Quantification

Après l'échantillonnage on applique la quantification. But : réduire le nombre de bits nécessaire pour encoder l'image (je crois). Deux type de quantifier

- Scalaire : chaque échantillon est quantifié
- Vecteur : une séquence d'échantillon est quantifié

2.3.1 Scalar quantification

Aucune idée de quoi on parle.

3 Transformé de Fourier discrète

C'est l'échantillonnage de la transformé de Fourier du signal échantillonné.

1. Transformé de Fourier du signal échantillonné. Calcul DIAPO 70 me semble pas mal.
2. On windows
3. On resample la DFT pour obtenir N valeur (comme donné à l'origine.) : ça ressemble beaucoup à la série de fourier mais avec un k discret plutôt qu'un x continue.

Si on a un signal de N sample, alors la FT nous rend N valeur tel que : la formule diapo 75.

Il vas vite mais globalement pas de différence en 2D

4 What you need to know

The difference between :

1. Fourier transform of a continuous signal \rightarrow a continuous function
2. Fourier transform of a windowed signal \rightarrow a continuous function
3. Fourier Transform of a discrete (sampled) signal \rightarrow again a continuous function
4. discrete Fourier transform of a discrete signal \rightarrow a discrete function

Nouveau cours du 11/10

5 Linear filtering

5.1 Linear time-invariant system

- Linéaire + invariant dans le temps \rightarrow Convolution.
- Filtrer en espace \Leftrightarrow filtrer en fréquence par la convolution. Voir les deux formules diapo 7
- Avantage et inconvénient des deux méthodes : dépendra de ce qu'on fait en pratique
- La même chose en 1D discret, 2D Discret.
- On retient que c'est une convolution quoi mdr

5.2 Spatial filtering

Pourquoi filter :

- Virer le bruit, trouver les bords
- Globalement on utilise une combinaison linéaire des pixel voisin pour déduire le nouveau pixel
- $\cdot \star h$ opérateur qui s'applique sur chaque pixel

Définition 5.1 (le bruit).

- processus indésirable dont on connaît idéalement la loi
- typiquement gaussien (type neige télé), ou impulse (arrive régulièrement dans les capteurs)
- Le flou est un bruit

Exemple 5.1. Exemple de bruit diapo 19 :

- bruit gaussien
- poivre et sel pour la deuxième image
- ?
- Bruit multiplicatif : grain

Filtre spatial :

- On prend chaque pixel et on regarde les pixel autour
- 2D : Formule plus simple à lire : pour le pixel i, j on regarde grâce aux sommes les pixel autour, pondéré par les valeurs de h
- Deux cas pour h : finit (FIR) ou infini (IIR)

FIR Filter :

- Deux versions du filtre : une antisymétrique, quand on regarde en bas à droite on vas chercher la valeur en haut à droite. La version symétrique existe. C'est comme ça pour garder la commutativité de la convolution il parrain.
- Algo pour le faire : DIAPO 24

On déplace une fenêtre sur chaque pixel. Comment faire pour les bords ?

- "Convolution linéaire" : En faite par la convolution si la fonction image vaut zéros alors la convolution vaut zéros. Fonctionne super bien quand la fonction est a support compacte. Mais sur une photo en réalité on a la scène qui est hors cadre qui est importante donc hypothèse bof validé.
- "Convolution circulaire" : j'ai loupé merde
- On rogne l'image en ne traitant pas les bords.

Smoothing linear filters :

- Pour réduire le bruit on peut faire la moyenne des pixel autour.
- Exemple de la matrice h diapo 29
- \rightarrow on réduit les détail
- Comment ça se comporte en fréquence ? Filtre passe bas, idéal ? Voir TME

Gaussian Smoothing :

- On pondère la moyenne par une gaussienne en choisissant bien σ . Il joue un rôle similaire à la taille du filtre moyenner. Plus il est grand plus on élimine de détail.
- Exemple numérique diapo 40
- Dans l'espace des fréquences, une gaussienne est une gaussienne. C'est un filtre passe bas non idéal.
- Exemple de la fonction de transfert plotée dans les diapos suivante

Binomial filter :

- Pour avoir la matrice on fait un produit cartésien du binôme de newton
- Lowpass non idéal

Autre filtre :

- Pyramidal : lowpass non idéal

- Conic : lowpass non ideal
- Sharpening : high pass non ideal, increase image details
- Gabor filter :
 - On s'en est beaucoup servie pour la classification d'image
 - Calcule des features pour différente orientation (?)
 - Pour la formule il a pas trop su expliqué mais l'exp représente une partie de la gaussienne orienté dans une certaine direction γ . et la partie cos représente un détecteur de contours.
 - Faire varier θ fait changer l'orientation
 - Faire varier s , rend la dirac de plus en plus dirac
- Filtre séparable :
 - La plupart des filtres 2D peuvent être exprimé par deux filtres 1D. (moyenne, gaussien, binomial, ...)
 - Permet d'aller plus vite et même faire des algo récursif.
- Filtrer les couleurs des images :
 - Comme h est une matrice, quand on multiplie le pixel vecteur de 3 couleurs
 - Si la matrice est diagonale on traite les 3 channels pareil
 - Sinon non
 - Pratique car les canaux RGB sont relativement indépendant
 - Mais attention on peut créer des fausses couleurs parfois comme l'indépendance n'est pas pur.

6 Filtering in the frequency domain

Définition 6.1. On considère un signal discret dont on va modifier certaine composante de fréquence

- Dans le domaine spatial = convolution
- La on va multiplier les valeur dans le domaine des fréquences

3 familles : lowpass, highpass, bandpass

ideal : les coef sont égal à 1 ou zéro, ça passe ou ça passe pas

Principe : DIAPHO 60

1. On prend la DFT
2. On créer notre H et on la complete avec des zéros pour faire un multiplication de matrice ez (voir diapo 61)
3. On multiplie element wise
4. Inverse DFT \rightarrow filtered image

Exemple diapo 62 : on cut le bruit net

Lowpass 2D filter :

—

ideal lowpass 2D filter :

- IMAGE diapo 66
- La ligne de la DFT de base : Pourquoi? parce que l'image originale peut être vu comme une fonction porte de FFT *sinc*. Puis après le filtre la FFT est proche d'une fonction porte du coup on retrouve une sorte de *sinc* dans l'image filtré == Gibbs artefacts
- Conséquence : ... Diapo 67
- Dans l'espace normal, c'est une convolution avec une fonction de Bessel

Les filtres idéaux cool quand on connaît la fréquence précise à cut, mais on a des gibbs artefact.

Lowpass butterworth 2D filter :

- Le paramètre n permet de faire varier la ressemblance avec une porte. Comme elle est continue est dérivable ça permet de limiter l'effet Gibbs.

Le filtre anti aliasing :

- Pour limiter les effet d'antialiasing, on va filter les fréquences qu'on capture autour appartenant au autre sinc.

Highpass 1D filtering :

- Même chose que tout à l'heure mais on inverse

Passe bande :

- "C'est le tric intermédiaire"
- on coupe à droite et à gauche

- Ca se présente comme un anneaux en 2D
- Dans l'exemple diapo 85, on isole une fréquence, on récupère deux diract, qui donne presque un cosinus.

7 Filtrage non linéaire

Le filtre mediant :

- Fait la médiane du voisinage
- Parfait pour les bruit "salt and pepper" car le bruit est ponctuel.
- Préserve les contours : on en discutera plus tard
- Réduit aussi les autres bruits

Autre filtre non linéaire :

- Filtre max : Supprime le bruit poivre
- Filtre min : Supprime le bruit sel
- On peut les composer et les stack
- Nagao filter : On peut prendre des fenêtre bizarre au sein de la grande fenêtre. Puis il prend la moyenne de plein de sous fenêtre bizarre et change la valeur du pixel sur la plus petite.

Nouveau cours6 du 18/10

BEFORE DIAPO 12 BIG RETARD

Edge Model

- On défini

8 Edge detection with filtering

8.1 First order approaches

Principe

- On peut pas faire tendre $h \rightarrow 0$ car on est sur une grille de pixel indexé sur \mathbb{N} , donc on vas supprimer la limite. On dit qu'on fait une différence fini.
- Il a dit c'est équivalent à une convolution.

Discrete approximation of gradient

1. VOIR DIAPO ETAPE de l'algo
2. 6) Edge linking : on prolonge le frontière dans l'espoir d'avoir un truc qui se ferme et d'avoir un contour.
 - Pas de choix meilleurs que d'autre
 - Roberts : calcule les gradients direction $\pi/4$
 - Où on met l'origine du masque? Il a dit "je vous laisse réfléchir" gdshgis
 - Diapo 23 : le premier c'est comme la matrice. Filtre passe haut dans la direction abscice
 - Diapo 24 : c'est le Robert, pareil la première image c'est comme la matrice. Finalement ça ressemble au filtre gradient mais pivoté, d'où la matrice. Passe bande dans une direction
 - Diapo 25 : Filtre de Prewitt avec les colonne de 1, deux pentes, deux sinus cardianux. Plutôt passe bande.
 - Diapo 26 : Sobel filter. Proche du précédent. Fait disparaître les lobes des sinc
 - Diapo 27 : On moyenne avec le vecteur vertical, et on dérive avec le vecteur horizontale. Pourquoi on moyenne? Pour prendre en compte les contours.
 - Diapo 28/29 : Filtre + réponse impulsionnel. Filtre moyennneur => Resemble à un sinc dans Fourier. Filtre qui ressemble à une gaussienne => Une gaussienne dans Fourier
 - Diapo 31 : Ce filtre permet de capturer les contours dans plusieurs direction en faisant pivoter les chiffres de la matrice.
 - Diapo 32 : belle exemple de comment faire sur une omage. Image θ c'est les directions, mais comme on fait une division dans le \arctan c'est pas très résistant au bruit
 - Diapo 34 : Ici on fait varier le seuil pour la carte de la norme $\|G\|$ Trop bas, on s'assure les vrais positifs mais peu de point. A l'inverse trop de point avec beaucoup de faux positif.
 - Diapo 35 : On peut lisser les contours avant la detection de contour avec un filtre gaussien ou le mediant. Celà a pour effet d'épaissir les trait de contour.

- Diapo 36 : Exemple avec un filtre gaussien
- Diapo 37 : High smoothing : robuste au bruit mais bord épais (localisation imprécise). Faible = inverse. Threshold :
- Digression : Texture in Image
- Définition : spatial repetition of the same pattern whatever the direction
- Processus stochastique et stationnaire.
- Diapo 40 : Texture : pas vraiment d'orientation de direction, contrairement à nos contours. Le plus simple est de lisser l'image (perte de loc) ou seuillage (autre problème)

8.2 Second order approach

- Le laplacien est invariant par rotation
- Diapo 43 : de toute manière on va approcher le laplacien. Marche mal sur les coins.
- Diapo 44 : méthode différence finie pour approcher le laplacien. Y'a un lien entre les matrices en bas et la formule, c'est les coef de la somme. la deuxième matrice est plus robuste aux rotations.
- Diapo 45 : passe haut
- Diapo 46 : passe haut
- Diapo 47 : Tuto appliquer la détection de bord avec le laplacien : On prend l'image, on convolue avec les laplaciens précédents. Et on regarde les passages par zéro : aka les changements de signes. Pour ça on regarde les min et max + un seuil pour éviter le bruit
- Diapo 48 réponse du laplacien sans seuil
- Diapo 49 :
- Diapo 50 : Pro and cons du laplacien vs le gradient : lire le diapo
- Diapo 51 : Pré-processing : les maths du pourquoi ça marche un filtre
- 1. .
- 2. dans l'équation (6) on utilise la dernière égalité
- 3. Car on connaît bien g
- Diapo 52 :
- Diapo 53 : C'est proche de ce qui se passe dans le cortex visuel. On aime bien les lissages à différentes valeurs de σ , il paraît que ça permet de voir les objets d'une certaine taille == pyramide de résolution
- Diapo 54 : Pyramide de résolution : filtre gaussien \rightarrow subsampling \rightarrow boucle. En fait ça permet de capturer les bords de toutes les tailles.
- Diapo 55 : Après on combine tous les résultats : multiscale fusion
- Diapo 57-60 : Exemple

9 Approche continue

9.1 Optimal filtering

Approche continue :

- Canny a maximisé des critères qu'il trouvait importants, en supposant un bruit gaussien
- 3 critères : Bonne détection, bonne localisation, unicité \Leftrightarrow un filtre dérivé qui vérifie des hypothèses mathématiques correspondant aux 3 critères.
- Finalement c'est proche d'une gaussienne. Diapo 65
- Y'a plusieurs solutions mathématiques.
- Diapo 67 : on peut simplifier la solution en passant sur un support infini (IIR) + implémentation récursive == Canny Deriche == rapide, paramètre α
- Canny Deriche : α même effet que pour la gaussienne : épaisseur, précision, bruit.

10 Post processing

10.1 Non maxima suppression, threshold and edge linking

Post processing

- Diapo 72-73-74
- Diapo 75 : Finalement on a que 8 direction comme on est sur une grille de point. On peut booster en gardant que les points estimé dans ces directions (pas sûre si j'ai bien compris)

Thresholding : diapo 77

- Technique basique
- Technique hysteresis : super puissant. Premier seuillage, puis deuxième seuillage en gardant les pixel qui ont un voisinage élevé (on vire les points seules finalement). Honnêtement pas tout compris.
- Finalement c'est une manière de prolonger les contours!

Edge linking and boundary detection : prologement de contour

- Et avoir des contours continue c'est important pour faire de la detection de zone.
- On peut transformer les pixel en un graph et chercher les cycles de ce graph en posant certaine condition.
- On peut essayer de retrouver une forme simple avec le bord (cercle, rectangle, ect) en utilisant une méthode de régression ou avec une transformé de Hough (on process par vote, mais coûteux)

Nouveau cours7 du 25/10

On vas chercher les autres features (descripteur) que les contours qui sont pas super robuste (au bruit). On veut qu'il soit robuste par rotation, traslation ect.

11 Point of interest

11.1 Hessian detector

Diapo 9 :

- On peut voir sur l'image I_{xx} et I_{yy} certain contour ne sont pas detecter entre les deux, les image se complete
- On vas essayer de regrouper toutes les informations dans un scalaire.

Diapo 10 :

- En utilisant le determinant : on peut detecter les max local, min, point selle.

Diapo 11, 12, 13 :

- Exemple de ce qu'on obtiens en faisant le determinant + une threshold
- → On obtient un détecteur de point critique, de coin (?)

11.2 Harris detector

Diapo 14 :

- Dix ans plus tard : harris detector, mieux que la simple matrice hessienne
- Utilise un detecteur d'autocorélation : qui regarde la corelation dans un voisinage
- Du coup on utilise une fenêtre gaussienne et on regarde si ça bouge dans deux direction (?)

Diapo 15-16-17-19 :

- TUTO COMMENT FAIRE CA
- Les valeurs propres de la matrice R_i diagonalisé nous dise???? Il m'a perdu
- Mais c'est beaucoup plus robuste

Diapo 19-20-21-22 :

- Robuste mais pas beaucoup au problème d'échelle, fonctionne pas trop avec les arbre dans la diapo 22 → On peut liser l'image pour se débarrasser des petit détail

Diapo 23 :

- Ahh c'est l'histoire avec les determinants

12 Multiscale detection

12.1 Sclae selection issue

Diapo 24-25 :

- Exemple avec la tête de clown
- Exemple avec l'église

Diapo 26 to 32 :

- On cherche un opérateur qui réponde de la même manière pour toutes les échelles
 - On peut trouver la scale optimal en regardant le max des graphiques
- Diapo 33 :
- Avec le filtre gaussien, on fait varier la taille des object qu'on peut detecter dans l'image.
 - En lissant on perd les petit détail
- Diapo 34 :
- ???????

12.2 Harris-Laplace detector

- Diapo 35 :
- idk en gros il a dit que ça marchait pas
- Diapo 36 :
- $x' = sx$ on fait un zoom
 - pareil on zoom le gaussien kernel
 - on caunait la dérivé de la gaussienne donc c'est elle qu'on dérive en premier
- Diapo 37 :
- On se retrouve avec deux gaussienne : une qui module la taillbe pour l'intégration, et un qui mesure la différentiation .
 - On se retrouve avec un detecteur robuste pour toute scale
 - Prix à payer : perte en loc
- Diapo 38 :
- illustre le problème de précision spatiale
- Diapo 39 :
- On peut se donner des critères supplémentaire pour améliorer la précision
 - On peut utiliser une normalisation de la des dérivé
- Diapo 40 :
- Image = toute les echelles VS echelle optimale
 - Il a dit d'autre truc

12.3 Blobs detector

- Diapo 41 :
- Laplacien de la gaussienne à différente echelle
 - Blob = une région qui ressemble à une gaussienne
 - horrible comme il est pas clair sdgfsbgfhehg
- Diapo 42 :
- jerqgugjeg
- Diapo 43 : Exemple Diapo 44-45-46 :
- Pour aller plus vite, on peut soustraire les gaussiennes pour obtenir un truc qui ressemble à un laplacien : $DifofGaussian \approx Laplacienofgaussien$
 - Diapo 46 : Exemple

13 Regions of interest

13.1 MSER

- Diapo 47-48 :
- On vas seuiller au maximum tout en obtenant des régions stables.
 - En divisant l'image en plusieurs partie, région convexe
 - Exemple diapo 49 assez insane

13.2 Quantitative Evaluation

- Diapo 51 :
- Pour évaluer
 - Repeatability

- Overlap error : On mesure la superposition
- On a des critères pour comparer nos algos sur différentes images qu'on fait varier par certaines transformations

Diapo 52-54 :

- Les types de transformation
- Scène plus ou moins compliquée : Structure ou texture exemple diapo 57-58
- Rien dit sur le 3ème point

Diapo 55-56 :

- On compare les différents filtres en fonction du point de vue

Diapo suivant : plein d'exemple Diapo CCL :

- Mathilde m'a déconcentré

Nouveau cours du 22/11

Je suis pas allé en cours pour réviser et parce que j'ouvre jamais ce document il se sera à rien.

Mais du coup je vais relire le diapo et prendre des notes de ce que je comprends :

- Mumford and Shah model :
 - Optimization based methods
 - Principe : on va minimiser une fonction mesurant la distance entre l'image segmentée et l'originale
 - On définit l'énergie de l'image segmentée comme la distance euclidienne entre chaque pixel de l'image segmentée I et les valeurs moyennes des pixels de l'image originale R
 - Par magie mathématique ça donne finalement la somme des valeurs propres de la matrice de variance-covariance des couleurs pixel (3x3)
 - Facile à optimiser comme un problème de dual + Lagrange
- Clustering based methods
 - On peut faire un seuil binaire comme on a fait en TME sur des images en noir et blanc : 6 catégories de méthodes diapo 18
 - Mais c'est pas de la segmentation : on a pas forcément des zones connectées entre elles → Spatial analysis
 - On divise l'histogramme en plusieurs parties
 - Pour chaque partie on trouve le meilleur seuil pour diviser l'image en la plus grosse section
 - Sur des images en couleurs on va clusteriser dans \mathbb{R}^3 , par exemple avec k-means : pros and cons diapo 34
 - Mean shift algo :
 - Finalement proche d'un k-mean mais au lieu de prendre tous les points les plus proches, on prend uniquement ceux dans un certain rayon. Puis centroïde puis boucle
 - On fait ça dans l'espace des features (couleurs, gradient, texture, ?)
 - Un point par feature, donc pas de dimension je crois ?
 - On merge les points proches à la fin
 - pros and cons diapo 41
- Region based segmentation
 - 2 approches : soit on part d'un cluster de la taille de l'image pour le réduire, soit on part de clusters de la taille de pixels (des seeds) puis qu'on aggrandit : fusion or division
 - Fusion : pros and cons diapo 48
 - Division : on part d'une zone de la taille de l'image, qu'on divise par exemple en 4, puis encore, puis jusqu'à qu'il n'y ait plus de régions non homogènes à diviser
 - Bien sûr on utilise un critère d'homogénéité choisi à l'avance
 - Algo split and merge == fusion des deux méthodes
 - On peut aussi couper le graph en plusieurs morceaux en fonction des valeurs d'une fonction de coût (?)
- Il y a encore beaucoup beaucoup d'autres méthodes : chacune avec leurs avantages et inconvénients en fonction du type d'image

Nouveau cours du 29/11

14 Feature descriptor

Comme la dernière fois je relis le diapo et prend juste des notes ici : pas de ce que le prof a Gabor filter

- filtre qui permet d'obtenir l'orientation et la scale dans les images
- Prhce du système humain avec les neurones spécialiser pour certaine orientation

Local derivative descriptors

- ??
- SIFT :???
- Autre descripteur : BRIEF, SURF, MORPS, Shape based descriptor
- → Plein de descripteur, souvent on les combine, le top c'est SIFT

Application en :

- Image matching : recherche d'image dans image avec de la robustness
- Indexing and searching in database :