

Bureau d'étude Twizy

Traitement d'images et reconnaissance de formes

Steven Le Cam

steven.le-cam@univ-lorraine.fr

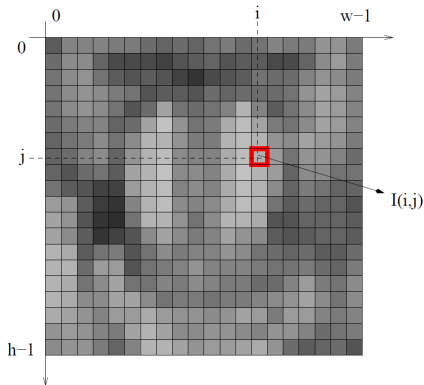
ISN parcours SA



- 1 Histogramme et seuillage
- 2 Opérations morphologiques
- 3 Détection de contours
- 4 Reconnaissance de formes

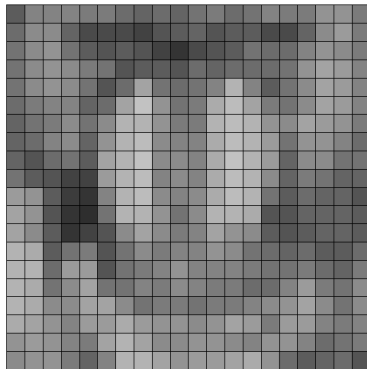
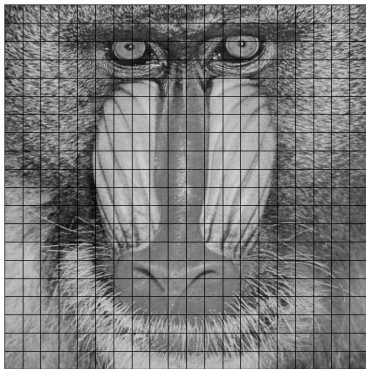
Image numérique

- Fonction à support discret, borné, et à valeur discrète
- Représentée sur un pavage rectangulaire découpé en pixels
- Chaque pixel possède une coordonnée dans l'espace 2D.



Qualité de l'image: 1) Résolution

Une image bien résolue permet de restituer les détails fins d'une image (les hautes fréquences)



Qualité de l'image: 1) Résolution

Une image bien résolue permet de restituer les détails fins d'une image (les hautes fréquences)



512x512



256x256



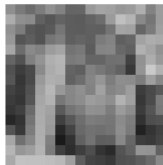
128x128



64x64



32x32



16x16

Qualité de l'image: 2) Quantification

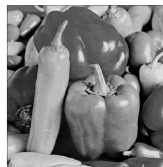
La quantification désigne la résolution tonale: détermine le nombre de valeurs que l'on peut attribuer à un pixel. (8 bits $\rightarrow$ 256 valeurs pour le jpg standard)



8 bits



5 bits



4 bits



3 bits



2 bits

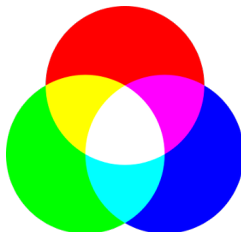


1 bit

La couleur

Codage RGB

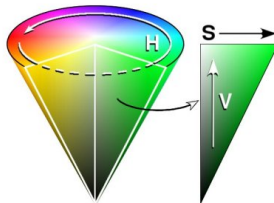
- Espace des couleurs primaires Rouge Vert Bleu
- Chaque pixel dans l'espace RGB est codé sur 3 valeurs (image multi-composantes)



La couleur

Codage HSV

- Codage intuitif de la façon dont les couleurs sont perçues par l'œil.
- Teinte (Hue): désigne la couleur codée suivant son angle sur le cercle des couleurs
- Saturation: taux de pureté de la couleur, entre la couleur pure et le gris (achromatique)
- Value: brillance de la couleur, noir lorsque la valeur est nulle. Permet de jouer sur les contrastes.



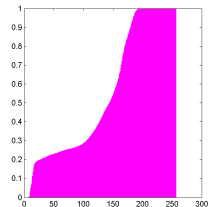
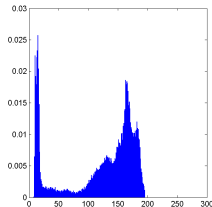
Plan

- 1 Histogramme et seuillage
- 2 Opérations morphologiques
- 3 Détection de contours
- 4 Reconnaissance de formes

Histogramme et seuillage

Histogramme

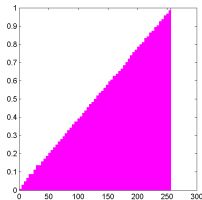
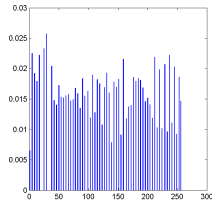
- L'histogramme permet de représenter la distribution des valeurs présentes dans une image. Pour chaque niveau de quantification n , on compte le nombre de pixels de l'image $H_I(n)$ prenant cette valeur.
- Histogramme normalisé: consiste à diviser les valeurs de H_I par le nombre total de pixels pour obtenir des valeurs entre 0 et 1
- Histogramme cumulé: on attribue à $H_I^c(n)$ la proportion de pixel de valeur inférieur à n .



Histogramme et seuillage

Augmentation du contraste: normalisation et égalisation d'histogramme

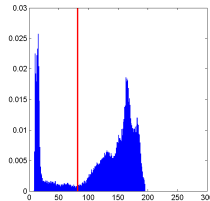
- Normalisation: utilisation de toute la dynamique de quantification
- Égalisation: imposer un histogramme plat (histogramme cumulé: pente linéaire de 0 à 1)



Histogramme et seuillage

Seuillage de niveaux de gris

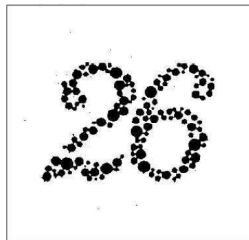
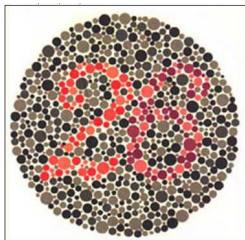
- Permet de faire ressortir des zones particulières de l'image ayant des valeurs de pixels similaires (ex: contenu dans un intervalle $[a,b]$). C'est une forme primitive de segmentation d'une image.
- En particulier on peut binariser l'image en mettant à 1 et 0 les pixels de part et d'autres d'un seuil.
- Détermination du seuil: manuelle en identifiant les modes de l'histogramme, détection de vallée, minimisation de variance intra-classe (Otsu), K-moyennes...



Histogramme et seuillage

Seuillage d'images couleurs

- Dans ce cas, l'image est composée de plusieurs bandes (RGB, HSV)
- Transformation en niveau de gris: peu satisfaisant car certaines couleurs bien distinctes auront des niveaux de gris similaires après transformation
- RGB: seuillages simultanés dans les trois bandes pour faire ressortir les objets d'une couleur déterminée.
- HSV: la bande H (tonale) correspond aux couleurs, le seuillage par couleur est facilité dans cette bande.



Histogramme et seuillage

Application: seuillage de couleurs

- Charger l'image 'pions.jpg' sous Matlab et la transposer dans l'espace HSV
- A partir de l'histogramme de la bande tonale, sélectionner les seuils relatifs aux jetons de couleurs jaunes et rouges
- Seuiller l'image et créer le masque des jetons (jetons blanc sur fond noir)



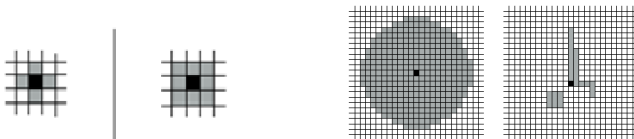
Commandes Matlab: `imhist` - `graythresh` - `rgb2hsv` - `imshowpair`

Plan

- 1 Histogramme et seuillage
- 2 Opérations morphologiques
- 3 Détection de contours
- 4 Reconnaissance de formes

Erosion-dilatation

Ce sont des opérations de base de la morphologie mathématique. Elles nécessitent la définition d'un élément structurant (ou masque), généralement carré (3×3 , 5×5) ou circulaire.



Principe

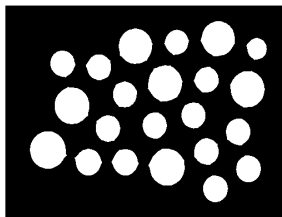
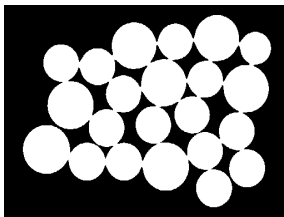
On parcourt l'image:

- On centre l'élément structurant (pixel noir) sur chaque pixel $I(i, j)$
- Un test est réalisé sur les pixels du masque (pixels gris) afin de déterminer la valeur du pixel $I'(i, j)$

Erosion-dilatation

L'érosion

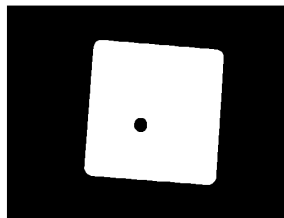
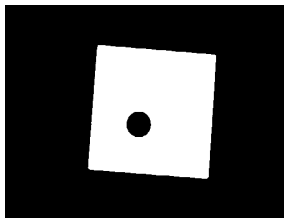
- Test: si un des pixels du masque est noir, mettre le pixel $I'(i,j)$ en noir
- Permet d'éliminer de petits objets détectés par le seuillage
- Permet de séparer des objets qui se touchent dans le but d'un comptage/extraction unitaire de ces objets



Erosion-dilatation

La dilatation

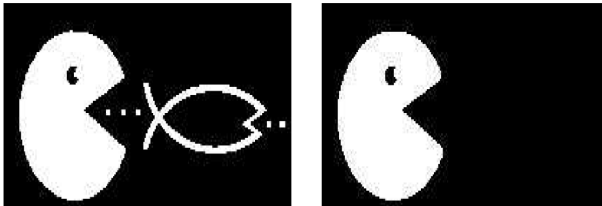
- Test: si un des pixels du masque est blanc, mettre le pixel $I'(i,j)$ en blanc
- Permet de restaurer les objets précédemment dégradés par l'érosion (erosion+dilatation=**ouverture**)
- Première approche pour la détection de contours par différence entre les images originale/dilatée/érodée



Erosion-dilatation

La dilatation

- Test: si un des pixels du masque est blanc, mettre le pixel $I'(i,j)$ en blanc
- Permet de restaurer les objets précédemment dégradés par l'érosion (erosion+dilatation=**ouverture**)
- Première approche pour la détection de contours par différence entre les images originale/dilatée/érodée



Erosion-dilatation

Application

- Appliquer une ouverture au masque des jetons afin d'éliminer les défauts du seuillage et de séparer les jetons rouges et jaunes.
Faire varier la taille et la forme des éléments structurants afin de parvenir à un résultat satisfaisant (formes circulaires/elliptiques)

Commandes Matlab: `strel` - `imerode` - `imdilate`



Plan

- 1 Histogramme et seuillage
- 2 Opérations morphologiques
- 3 Détection de contours**
- 4 Reconnaissance de formes

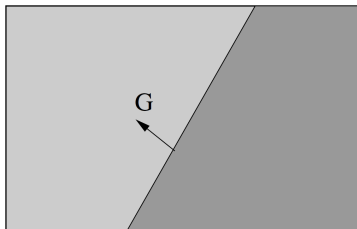
Détection de contours

Gradient

Le gradient d'une image est le vecteur $\nabla I(x, y)$ défini par:

$$\vec{G} = \nabla I(x, y) = \left[\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right]$$

- La direction du gradient maximise la dérivée directionnelle



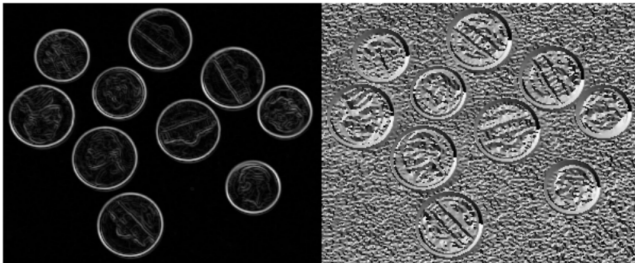
Détection de contours

Gradient

Le gradient d'une image est le vecteur $\nabla I(x, y)$ défini par:

$$\vec{G} = \nabla I(x, y) = \left[\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right]$$

- La direction du gradient maximise la dérivée directionnelle
- Norme: $G = \|\vec{G}\| = \sqrt{\frac{\partial I(x, y)}{\partial x}^2 + \frac{\partial I(x, y)}{\partial y}^2}$, Direction: $\vec{g} = \frac{\vec{G}}{\|\vec{G}\|}$



Détection de contours

Approche Gradient

- Détermination des extremas locaux dans le gradient de l'image
- Fonctions discrètes: les dérivées peuvent être approximées par différence finies

$$\nabla_x I(i, j) = I(i + n, j) - I(i, j) \text{ (en général } n = 1 \text{)}$$

- Ces dérivées sont calculées par convolution avec des opérateur de différences

Opérateurs de Roberts

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

- Dérivation selon les diagonales de l'image
- La valeur du gradient en chaque pixel est obtenu par $G = \sqrt{G_1^2 + G_2^2}$, avec $G_1 = h_1 * I$ et $G_2 = h_2 * I$

Détection de contours

Opérateurs de Prewitt

$$h_1 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

$$h_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Opérateurs de Sobel

$$h_1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$h_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- Chacun des filtre est séparable selon les directions x et y :

$$I'(x, y) = h_i(x, y) * I(x, y) = h_{i_y}(y) * (h_{i_x}(x) * I(x, y))$$

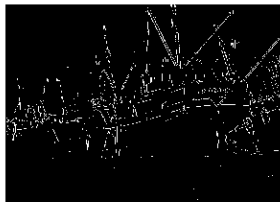
- Les deux opérateurs fonctionnent sur le même principe: filtrage passe-bas selon x puis passe-haut selon y pour h_1 , et inversement pour h_2 .
- Gradient: $G = \sqrt{G_1^2 + G_2^2}$, avec $G_1 = h_1 * I$ et $G_2 = h_2 * I$

Détection de contours

Image originale



Roberts



Sobel



Prewitt



Détection de contours

Filtre optimal: Canny

Critère d'optimalité de Canny:

- Bonne détection: identification de tous les contours
- Bonne précision: Minimisation de la distance entre les points sélectionnés et le vrai contours
- Réponse minimale: un contours n'est détecté qu'une seule fois
→ amène à la résolution d'une équation différentielle dont la solution est très proche de la dérivée d'un filtre gaussien:

$$h(x) = -\frac{x}{\sigma^2} \exp^{-\frac{x^2}{2\sigma^2}}$$

Détection de contours

Filtre optimal: Canny

Critère d'optimalité de Canny:

- Bonne détection: identification de tous les contours
- Bonne précision: Minimisation de la distance entre les points sélectionnés et le vrai contours
- Réponse minimale: un contours n'est détecté qu'une seule fois
→ amène à la résolution d'une équation différentielle dont la solution est très proche de la dérivée d'un filtre gaussien:

$$h(x) = -\frac{x}{\sigma^2} \exp^{-\frac{x^2}{2\sigma^2}}$$

- En 2D, actions croisées de deux filtres:

$$h_x(x, y) = -\frac{x}{\sigma^2} \exp^{-\frac{x^2}{2\sigma^2}} \exp^{-\frac{y^2}{2\sigma^2}}$$

$$h_y(x, y) = -\frac{y}{\sigma^2} \exp^{-\frac{x^2}{2\sigma^2}} \exp^{-\frac{y^2}{2\sigma^2}}$$

Détection de contours

Algorithme de Canny

- ➊ Application des deux filtres séparables selon les directions x et y
- ➋ Sélection des extremums dans l'image
- ➌ Applications de règles pour rejeter les faux positifs
- ➍ Filtrage Hysteresis (seuil double) pour éviter les trous dans les contours

Détection de contours

Algorithme de Canny

- 1 Application des deux filtres séparables selon les directions x et y
- 2 Sélection des extremums dans l'image
- 3 Applications de règles pour rejeter les faux positifs
- 4 Filtrage Hysteresis (seuil double) pour éviter les trous dans les contours

Image originale



Prewitt



Canny (par défaut)



Canny (customisé)



Détection de contours

Application

Utiliser la méthode *edge* de Matlab et tester les méthodes de Sobel, Prewitt et Canny sur les images suivantes:

- Détecter les contours des jetons à partir de l'image 'pions.jpg'
- Faire le même travail à partir de l'image seuillée (masque des jetons jaunes et rouges)

Commandes Matlab: `rgb2gray` - `edge`



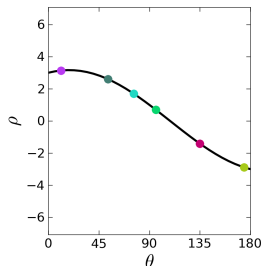
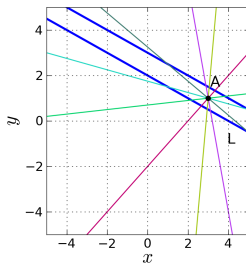
Plan

- 1 Histogramme et seuillage
- 2 Opérations morphologiques
- 3 Détection de contours
- 4 Reconnaissance de formes

La transformée de hough

Espace de Hough pour les droites

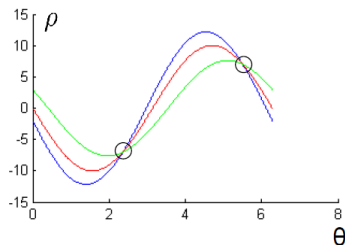
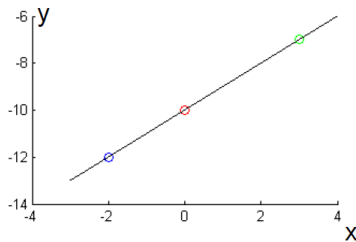
- Toute droite peut être définie par deux paramètres $y = ax + b$, ou encore $\rho = x \cos \theta + y \sin \theta$
- (ρ, θ) est l'espace de hough dans lequel une droite de l'espace d'origine est un point
- L'ensemble des droites passant par un point donné est représenté par une sinusoïde dans l'espace de Hough



La transformée de hough

Détection de droite

- A chaque point de contours de l'image, on lui associe sa sinusoïde dans l'espace de Hough
- Système de vote: les points de l'espace de Hough suffisamment représentés sont retenus comme étant des droites de l'image



La transformée de hough

Détection de droites: exemple

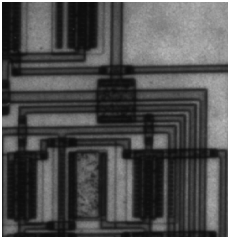
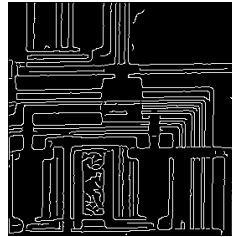
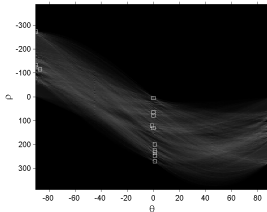


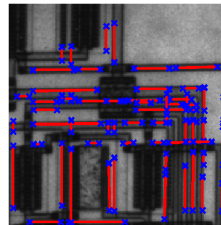
Image Originale



Contours (Canny)



Transformée de Hough



Détection

La transformée de hough

Espace de Hough pour les cercles

- Équation d'un cercle: $(y - y_0)^2 + (x - x_0)^2 = R$
- (x_0, y_0, R) : espace de hough pour les cercles

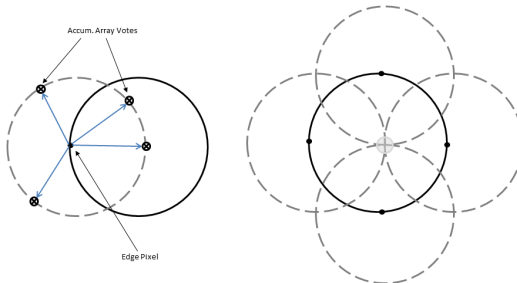
La transformée de hough

Espace de Hough pour les cercles

- Équation d'un cercle: $(y - y_0)^2 + (x - x_0)^2 = R$
- (x_0, y_0, R) : espace de hough pour les cercles

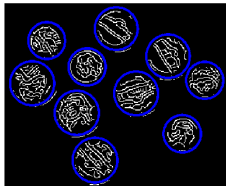
Détection de cercles

- Pour chaque point de contours, identification de l'ensemble des centres des cercles de rayon R passant par ce point.
- Système de vote: les couples (x_0, y_0) bien représentés sont détectés comme les centres de cercles (de rayon R) dans l'image.



La transformée de hough

Détection de cercles: exemple



Détection de cercles

Application

- Reprendre l'image des contours des jetons rouges et jaunes et détecter les cercles
- Jouez avec les paramètres d'érosion-dilatation, de détection de contours et de détection des cercles afin d'améliorer le résultat

Commande Matlab: `imfindcircles`

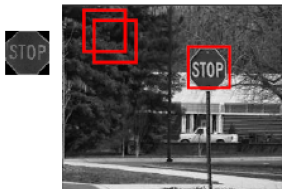


Template matching

On possède une vignette (ou template) de l'objet à identifier dans l'image

- Première approche: on recherche cette vignette à toute les positions de l'image → Très coûteux en calcul, problème d'échelle
- Seconde approche: extraction de fenêtres candidates dans l'image à partir de caractéristiques de l'objet (formes, couleurs...). Mise à l'échelle avec le template et détection si il y a correspondance, par corrélation par exemple:

$$\frac{\sqrt{\sum_{i=1}^h \sum_{j=1}^w (I_{fen}(i,j) \overline{T}(i,j))^2}}{\sigma_{I_{fen}} \sigma_T}$$



Template matching

Application

- 1 Détecter la lettre U dans l'alphabet à partir du template fourni par recherche exhaustive dans l'image
- 2 Extraire les fenêtres candidates dans l'image en se basant sur la couleur, et appliquer le template matching fenêtre par fenêtre

Commandes Matlab: `normxcorr2` - `corr2`



Extraction de caractéristiques

Principe

Autre solution: calcul de distance dans un espace de caractéristiques extraites de l'image

- Apprentissage: extraction de caractéristiques sur l'objet recherché
- Détection: recherche de ces caractéristiques dans l'image

Caractéristiques

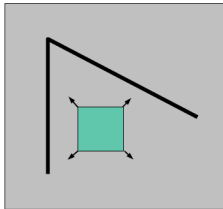
On distingue deux catégories:

- Caractéristiques globales: morphologie et couleurs
- Caractéristiques locales: **points d'intérêt**

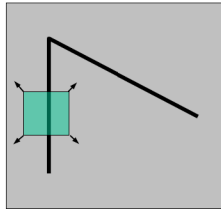
Extraction de caractéristiques

Points d'intérêt

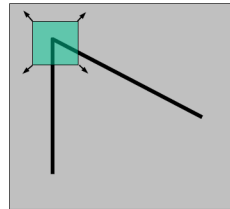
Points non usuels de l'image: points pour lesquels un changement maximal est observé dans l'image quelque soit la direction du décalage autour de ce point
→ ce sont les **coins** de l'image.



Plat



Contour



Coin

Avantages

- Local: segmentation non nécessaire
- Invariant aux transformations de l'image (rotation, luminance...)
- Robuste aux occultations

Extraction de caractéristiques

Points d'intérêt: illustrations



Extraction de caractéristiques

Points d'intérêt: les calculs...

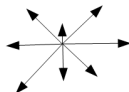
On considère l'erreur quadratique $E(u, v)$ (somme des carrés des différences) entre une vignette w de l'image I et une version décalée de cette vignette d'un incrément (u, v) :

$$E(u, v) = \sum_{(x,y) \in w} (I(x, y) - I(x + u, y + v))^2$$

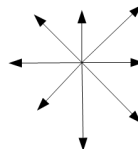
Coins: points pour lesquels $E(u, v)$ est *grand* quelque soit la direction (u, v)
 $E(u, v)$ peut se calculer à partir du gradient en chaque point de la vignette w



Plat



Contour



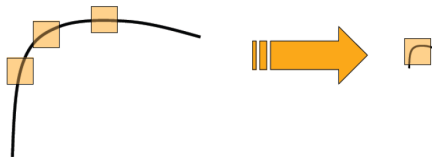
Coin

Extraction de caractéristiques

Méthode de Harris

Obtenu par analyse des valeurs propres de la matrice $E(u, v)$: le descripteur de Harris est associé à la plus petite des deux valeurs propres λ_-

- Avantages: robuste aux rotations et aux changements de luminance
- Inconvénient: sensible à l'échelle d'analyse!



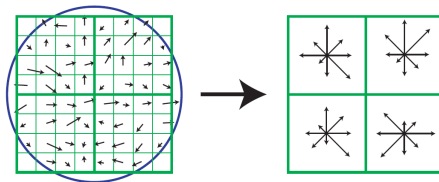
Solution: approche SIFT (Scale Invariant Feature Transform)

→ analyse multi-échelle par applications successives de filtres gaussiens.

Extraction de caractéristiques

Une fois les points d'intérêt détectés, comment les faire correspondre entre deux images?

- Calcul du gradient sur un patch (8×8 ou 16×16) autour du point d'intérêt
- Agrégation en sous-patch (2×2 ou 4×4) par calcul d'histogramme d'orientation (8 bins)
- Résultat: $4 \times 8 = 32$ ou $16 \times 8 = 128$ descripteurs
- Matching: distance quadratique entre les descripteurs



Gradient (patch 8×8)

Descripteurs (32 valeurs)

Extraction de caractéristiques

SIFT: illustrations



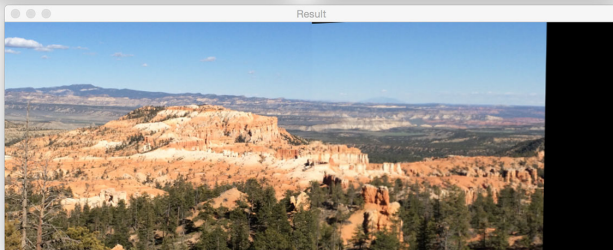
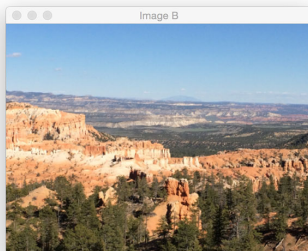
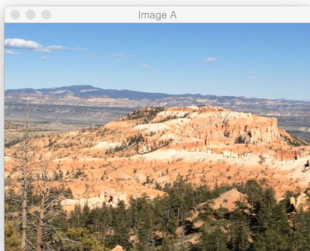
Extraction de caractéristiques

SIFT: illustrations



Extraction de caractéristiques

SIFT: illustrations



Extraction de caractéristiques

Application: Stan Matching

- Extraire les caractéristiques (point de Harris) à l'aide de la méthode SURF (approche multi-échelles similaire à SIFT) sur les deux images 'StanislasFace.jpg' et 'StanislasFace2.jpg'
- Identifier et afficher les points qui correspondent entre les deux images

Commandes Matlab: `detectSURFFeatures` - `extractFeatures` - `matchFeatures` - `showMatchedFeatures`



Références

Slides librement inspirées des ressources suivantes:

Florence Tupin, Antoine Manzanera, Henri Maître, Cours de traitement d'images, Télécom ParisTech

Nicolas Thome, Cours de Master de traitement d'images: transformée de Fourier, UPMC La Sorbonne

David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, 2004

Rajesh Rao, Features and Image Matching, Cours de l'Université de Washington

Ressources en ligne: The Hypermedia Image Processing Reference (HIPR), <http://homepages.inf.ed.ac.uk/rbf/HIPR2/>