ENSC – 2ème année

CO7SFTS0

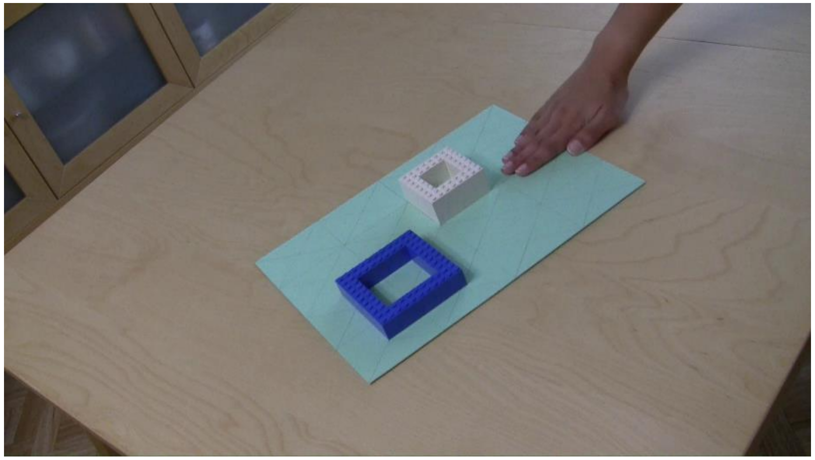
Projet de réalité augmentée

Marc Donias

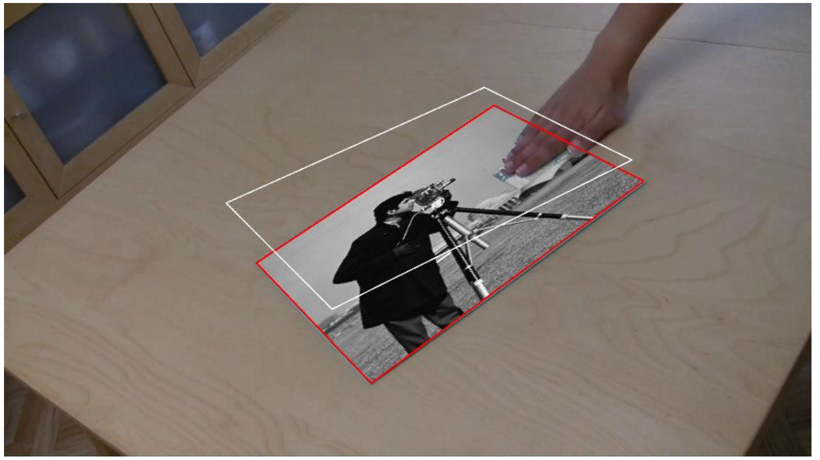
Initiation au Traitement des Images

I – Sujet

Application de réalité augmentée



vidéo initiale



remplacement et ajout de contenu

- Détection et suivi de coins
  - Analyse variationnelle et statistique
  - “Trajectographie”
- Réalité augmentée
  - Transformation d’espace
  - Incrustation de nouveaux contenus

Réalisations

Plan

- Chaîne algorithmique
- Suivi de points avec le détecteur de Harris
- Transformations géométriques planes rigides
- Prise en compte de la main
- Ajout de contenus 3D
- Implémentation

Initiation au Traitement des Images

II – Chaîne algorithmique

Remplacement de contenu (1/3)

- Détection du polygone de la feuille
  - Gradient d'intensité
    - Filtres de Canny (paramètres d'échelle)
    - Estimation des composantes horizontale et verticale
  - Initialisation/Suivi des 4 coins
    - Détecteur de Harris (variationnel et statistique)
      - Covariance des gradients
      - Combinaison Trace/Déterminant
      - Détecteur multi-échelle par association de deux détecteurs
    - Recherche de maxima locaux
      - Cribble exhaustif large
      - Prédiction linéaire de trajectoire et cribble exhaustif étroit

Remplacement de contenu (2/3)

- Remplacement de contenu du polygone
  - Détermination d'une homographie
    - Appariement coins du polygone/coins du contenu
    - Résolution exacte ou aux moindres carrés (si ajout d'autres points)
  - Transformation « homographique » d'une source
    - Application de l'homographie
    - Recopie du contenu (image ou image animée)
  - Prise en compte de la main
    - Modèle colorimétrique de la feuille
    - Segmentation partielle (« à droite »)
    - Masque de non-remplacement

Remplacement de contenu (3/3)

- Ajout de contenus 3D
  - Détermination d'une transformation 3D vers 2D
    - Détection et suivis des coins du polygone et des structures
    - Appariement points observés 2D/points théoriques 3D
    - Résolution aux moindres carrés
  - Ajout de structures « en fil de fer »
    - Modèle de contenu 3D statique ou dynamique
    - Application de la transformation 3D vers 2D
    - Tracé de segments

Initiation au Traitement des Images

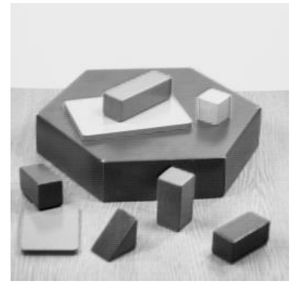
III – Suivi de points avec le détecteur de Harris

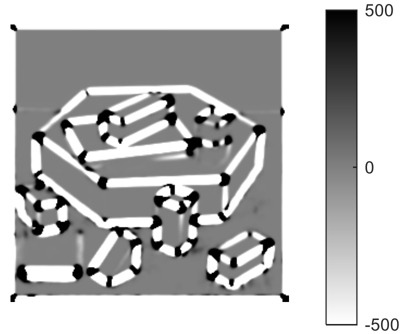
Approche variationnelle/statistique

- Covariance locale des gradients

Approche de Canny

$$C(x,y) = \begin{bmatrix} \sum_{\Omega} I_x(x,y) I_x(x,y) & \sum_{\Omega} I_x(x,y) I_y(x,y) \\ \sum_{\Omega} I_x(x,y) I_y(x,y) & \sum_{\Omega} I_y(x,y) I_y(x,y) \end{bmatrix}$$





- Combinaison Déterminant/Trace

$$D(x,y) = Det\,(C(x,y)) - \lambda\, Trace(C(x,y))^2$$

pondérateur ~ 0,05

Implémentation matricielle

- Estimation des dérivées en tout point par l'approche de Canny

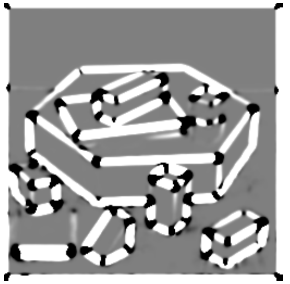
$$I_x^{\sigma_G} = I * G_x^{\sigma_G} \quad , \quad I_y^{\sigma_G} = I * G_y^{\sigma_G}$$

- Calcul de l'ensemble des covariances (moyennage gaussien)

$$C^{\sigma_G,\sigma_C} = \begin{bmatrix} C_{xx}^{\sigma_G,\sigma_C} & C_{xy}^{\sigma_G,\sigma_C} \\ C_{xy}^{\sigma_G,\sigma_C} & C_{yy}^{\sigma_G,\sigma_C} \end{bmatrix} \quad , \quad C_{xy}^{\sigma_G,\sigma_C} = (I_x^{\sigma_G} I_y^{\sigma_G}) * G_{\sigma_C}$$

- Calcul du détecteur en tout point

$$D^{\sigma_G,\sigma_C} = C_{xx}^{\sigma_G,\sigma_C} C_{yy}^{\sigma_G,\sigma_C} - C_{xy}^{\sigma_G,\sigma_C} C_{xy}^{\sigma_G,\sigma_C} - \lambda\,(C_{xx}^{\sigma_G,\sigma_C} + C_{yy}^{\sigma_G,\sigma_C})^2$$



$\sigma_C = 2$

$\sigma_C = 3$

Détecteur robuste multi-échelle

- Détecteur mono-échelle
  - Paramètre d'échelle σ_G du gradient
  - Paramètre d'échelle σ_C de la covariance

    $$\left.\begin{array}{l}\sigma_G \\ \sigma_C\end{array}\right\} D^{\sigma_G,\sigma_C}$$

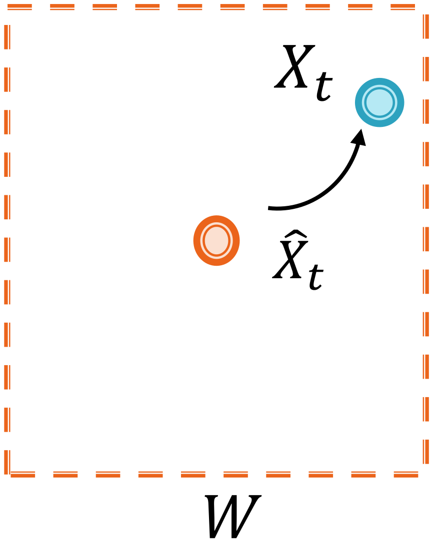
  - Intervalles de valeurs
    - Valeurs fortement positives : coins
    - Valeurs proches de 0 : régions quasi-uniformes
    - Valeur fortement négatives : contours

- Combinaison multi-échelle

$$D = \min\left(D_1^{\sigma_G,\sigma_{C_1}}\left|D_2^{\sigma_G,\sigma_{C_2}}\right|, \left|D_1^{\sigma_G,\sigma_{C_1}}\right|D_2^{\sigma_G,\sigma_{C_2}}\right)$$

Trajectographie linéaire

- Suivi/Poursuite de trajectoire
  - Position approximative $\hat{X}_t$
  - Position exacte X_t par recherche du maximum dans un voisinage W



- Stratégie « statique »

recopie $\hat{X}_t = X_{t-1}$

- Stratégie « dynamique »

prédiction linéaire $\hat{X}_t = X_{t-1} + \frac{X_{t-1} - X_{t-2}}{2}$

Paramètres « fonctionnels »

- Détecteur de Harris multi-échelle

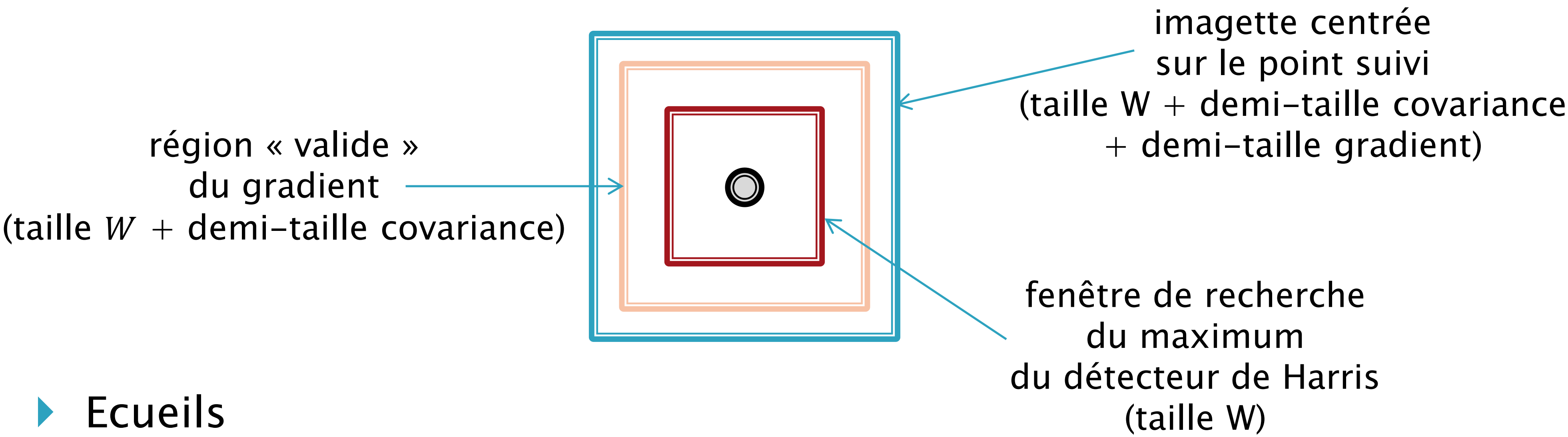
$$\begin{cases} \sigma_G \approx 2 \\ \sigma_{C_1} \approx 3 \\ \sigma_{C_2} \approx 5 \end{cases}$$

- Fenêtre de recherche du maximum

$$W = 37$$

Calculs accélérés

- Principe : calcul d'un détecteur de Harris centré sur chaque point suivi
  - Découpe d'une imagette de taille « adéquate »
  - Calculs convolutifs en conservant uniquement la partie « valide »



- Ecueils
  - Aspect multi-échelle
  - Maximum détecté en coordonnées relatives
  - Gestion de l'éventuelle « sortie » de l'image lors de la découpe

Initiation au Traitement des Images

IV – Transformations géométriques planes rigides

Contextes « plans »

- Prétraitement

Décodage
de QR-code

- Remplacement

Réalité
augmentée

- Alignement

Mosaïque de
prise de vues

Formalisme

<u>Transformation paramétrique</u>

$$I_1 \xrightarrow{T} I_2$$

point final

point initial

$$X_2 = T_\theta(X_1)$$

paramétrisation de la transformation géométrique

I_1

$$X_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

I_2

Nouveau cadre

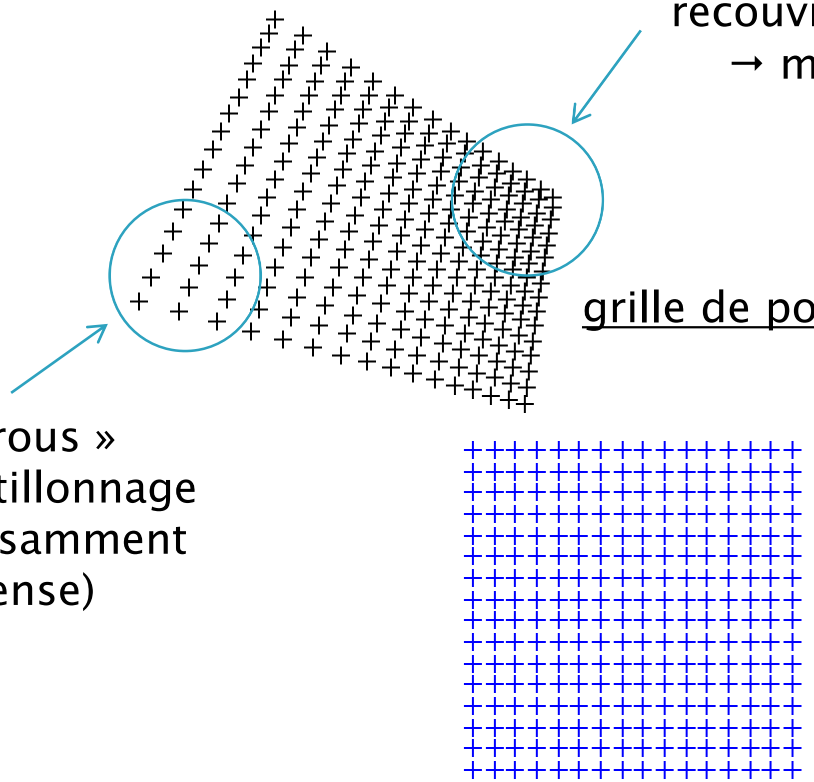
$$X_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

Ancien cadre

Hypothèse de conservation d'intensité $\quad I_1(X_1) = I_2(X_2)$

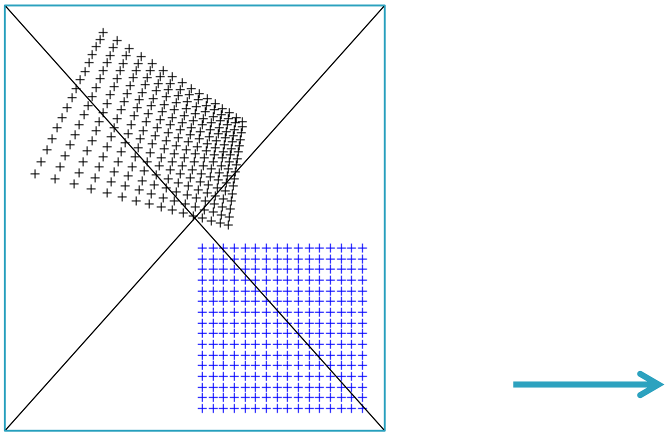
Algorithme « naïf »

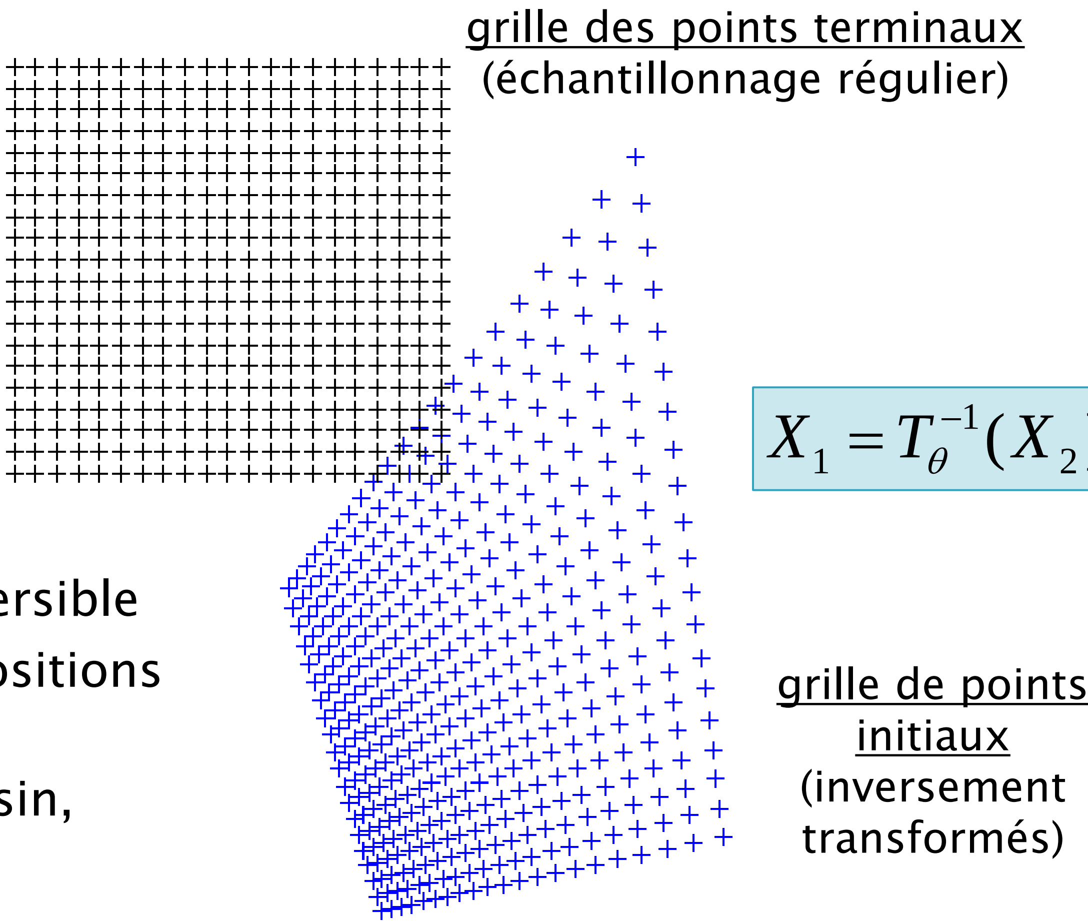
Principe : « déplacement » de pixels (coordonnées arrondies)



Algorithme

Principe : « remplissage » de l'image transformée



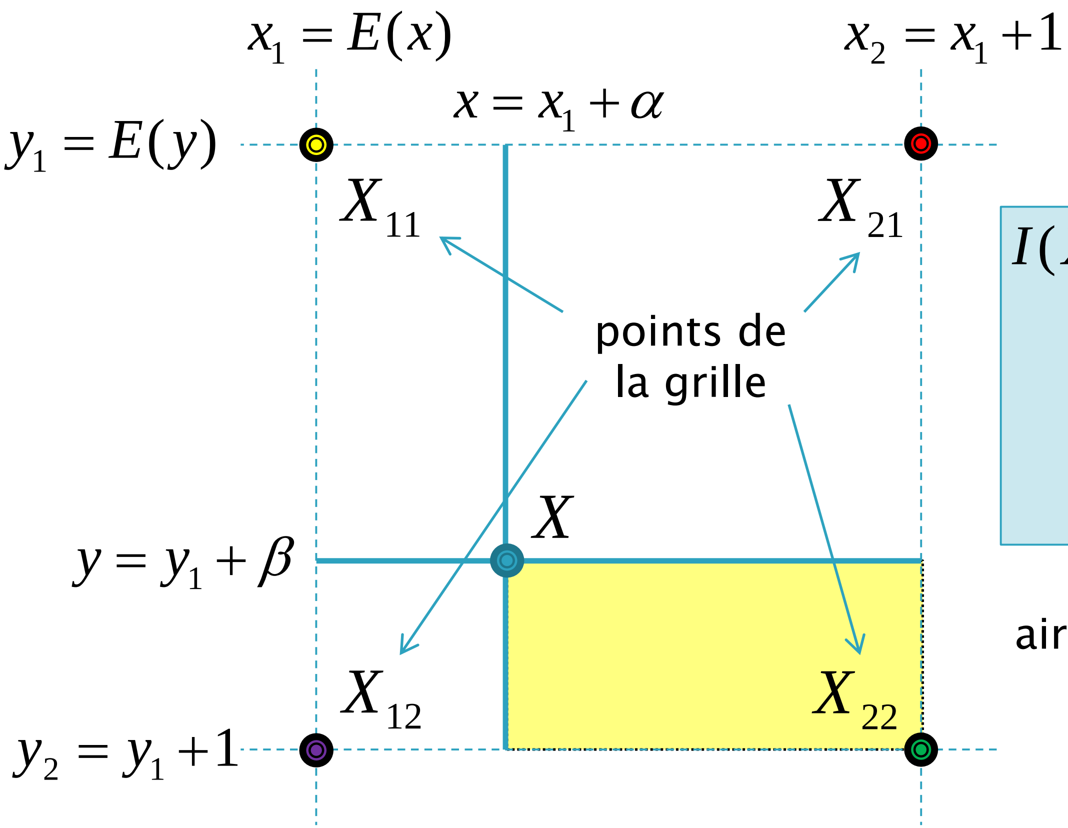


$$X_1 = T_\theta^{-1}(X_2)$$

- Transformation supposée inversible
- Valeur de « bord » pour les positions initiales en dehors de l'image
- Interpolation (plus proche voisin, bilinéaire, cubique, …)

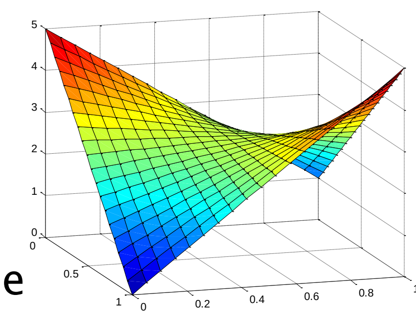
Interpolation bilinéaire

Utilisation de points voisins de la grille d'échantillonnage pour des positions à valeurs non entières



$$
\begin{aligned}
I(X) = \; & (1-\alpha)(1-\beta) && I(X_{11}) \\
& +(1-\alpha)\beta && I(X_{12}) \\
& +\alpha(1-\beta) && I(X_{21}) \\
& +\alpha\beta && I(X_{22})
\end{aligned}
$$

aire du rectangle opposé



interpolation non plane

Transformations affines

- Translation

$$X_2 = X_1 + T$$

- Rotation

centre

$$X_2 = R(X_1 - X_0) + X_0$$

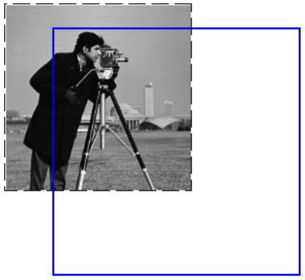
angle

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$

$X_0 = (128{,}128) \quad \alpha = 30°$

- Changement d'échelle

$$X_2 = s(X_1 - X_0) + X_0$$



$X_0 = (-200,-100) \quad s = 0.75$

- Transformation affine générique

$$X_2 = AX_1 + B$$

inversible pour une transformation non dégénérée

Coordonnées homogènes

$$X_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \longleftrightarrow \underline{X}_i = \begin{bmatrix} x_i \times s_i \\ y_i \times s_i \\ s_i \end{bmatrix}$$

facteur d'échelle

Coordonnées euclidiennes — Coordonnées homogènes

<u>Intérêts</u>

$$X_2 = AX_1 + B \longleftrightarrow \underline{X}_2 = H\,\underline{X}_1 \quad \text{avec } H = \begin{bmatrix} \boxed{\begin{matrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{matrix}} & \boxed{\begin{matrix} B_1 \\ B_2 \end{matrix}} \\ \begin{matrix} 0 & 0 \end{matrix} & 1 \end{bmatrix}$$

1 matrice et 1 vecteur de paramètres — 1 matrice de paramètres

changement d'échelle

rotation — translation

Cascade de transformations : $\underline{X}_2 = H_3 H_2 H_1 \underline{X}_1 = H\,\underline{X}_1$

Cas général – Homographie

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$$

$$\begin{cases} x_2 = \dfrac{H_{11}x_1 + H_{12}y_1 + H_{13}}{H_{31}x_1 + H_{32}y_1 + H_{33}} \\ y_2 = \dfrac{H_{21}x_1 + H_{22}y_1 + H_{23}}{H_{31}x_1 + H_{32}y_1 + H_{33}} \end{cases}$$

Transformation projective d'un plan
(8 degrés de liberté avec 9 coefficients à un facteur multiplicatif près)

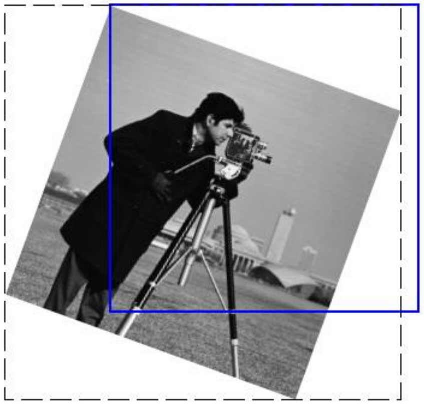
→ translation, rotation, miroir, transformation affine, projection, …

Exemples

- Agrandissement

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/1.25 \end{bmatrix}$$

- Rotation (autour de l'origine)



$$H = \begin{bmatrix} \cos 20° & -\sin 20° & 0 \\ \sin 20° & \cos 20° & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Cisaillement

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0.2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Homographie projective

$$H = \begin{bmatrix} 1 & 0 & 20 \\ 0 & 1 & 20 \\ -0.001 & 0.002 & 1 \end{bmatrix}$$

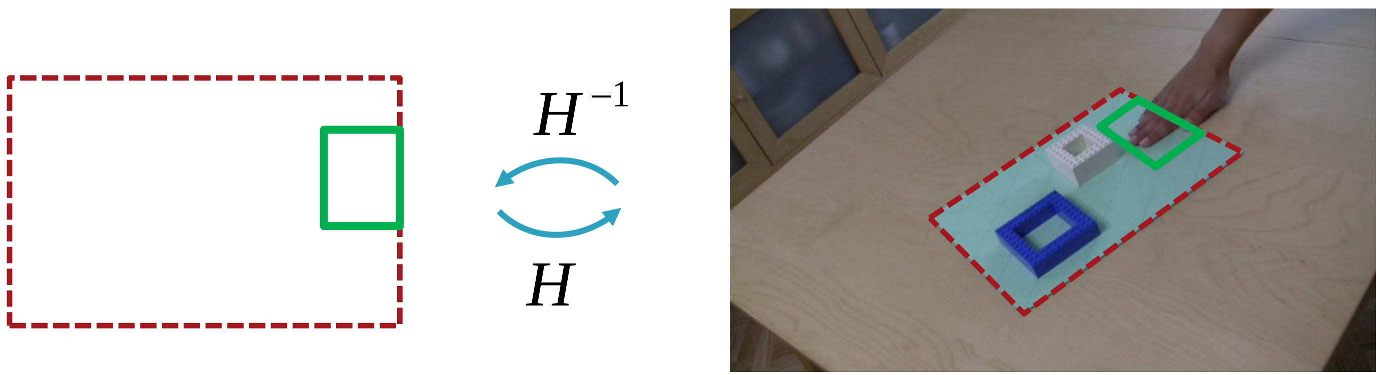
Initiation au Traitement des Images

V – Prise en compte de la main

Prise en compte de la main

- Observations
  - Le contact feuille-main est quasi-rigide
  - L’empreinte visuelle de la main qui occulte la feuille est quasi-plane

- Approche
  - Zone d’intérêt rectangulaire définie empiriquement dans l’espace du contenu à insérer
  - Segmentation colorimétrique de la feuille (détection de la main par complément)



Segmentation couleur (1/2)

- Principes
  - Choix d’un espace de représentation couleur
    - RGB
    - Lab
    - YCbCr, YUV
    - …
  - Evaluation binaire ou valuée de la similarité d’une couleur quelconque par rapport à un modèle colorimétrique de référence

- Approche basique (modèle « rectangulaire »)
  - 1 ou 2 seuils par composante couleur (similarité binaire)
  - Avantage : simplicité
  - Inconvénient : ensemble de couleurs très différentes identifiées comme semblables

Segmentation couleur (2/2)

- Approche adaptative (modèle « elliptique »)
  - Modèle $\Theta = \{\mu, \Sigma\}$ calculé à partir d'un échantillon $\{x_i\}$ représentatif

$$x_i = \begin{bmatrix} x_i^1 \\ x_i^2 \\ x_i^3 \end{bmatrix}, i = 1 \ldots N \qquad \mu = \begin{bmatrix} \mu^1 \\ \mu^2 \\ \mu^3 \end{bmatrix} = \overline{x_i} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\Sigma = \begin{pmatrix} \Sigma^{1,1} & \Sigma^{1,2} & \Sigma^{1,3} \\ \Sigma^{2,1} & \Sigma^{2,2} & \Sigma^{2,3} \\ \Sigma^{3,1} & \Sigma^{3,2} & \Sigma^{3,3} \end{pmatrix} \quad \text{avec} \quad \Sigma^{i,j} = \frac{1}{N}\sum_{k=1}^{N} \left(x_k^i - \mu^i\right)\left(x_k^j - \mu^j\right)$$

  - Métrique de ressemblance : distance de Mahalanobis

$$D_{\Theta}^{Maha}(y) = (y - \mu)^T \Sigma^{-1} (y - \mu)$$

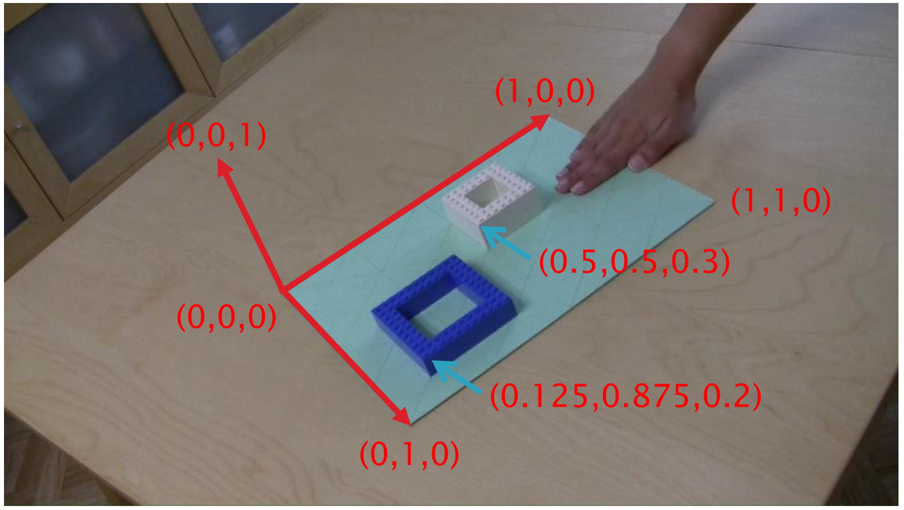
  - Similarité valuée (à seuiller)

Initiation au Traitement des Images

VI – Ajout de contenus 3D

Insertion de contenus 3D « fil de fer »

- Principes
  - Détermination des paramètres d'une transformation rigide 3D→2D
  - Application de la transformation à un modèle (à imaginer) composé de bipoints 3D (segments) définis dans un repère a priori



  - Tracés des bipoints 2D (segments) projetés

Transformation rigide 3D → 2D

- Formalisme en coordonnés homogènes

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix}$$

$$\boxed{\underline{X}^{2D}{}_2 = P\,\underline{X}^{3D}{}_1}$$

$$\begin{cases} x_2^{2D} = \dfrac{P_{11}x_1^{3D} + P_{12}y_1^{3D} + P_{13}z_1^{3D} + P_{14}}{P_{31}x_1^{3D} + P_{32}y_1^{3D} + P_{33}z_1^{3D} + P_{34}} \\ y_2^{2D} = \dfrac{P_{21}x_1^{3D} + P_{22}y_1^{3D} + P_{23}z_1^{3D} + P_{24}}{P_{31}x_1^{3D} + P_{32}y_1^{3D} + P_{33}z_1^{3D} + P_{34}} \end{cases}$$

(11 degrés de liberté avec 12 coefficients à un facteur multiplicatif près)

- Estimation au moindres carrés (avec au moins 6 points appariés)

$$AX = B \quad \rightarrow \quad X = \left(A^T A\right)^{-1} A^T B$$

pseudo-inverse (A rectangulaire)

Tracé de segments 2D (1/2)

- Tracé « fin » d'un segment AB

  - Détermination du nombre de points « utiles »

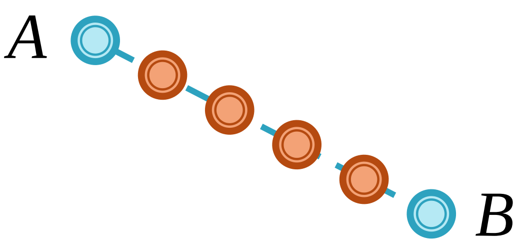
$$L = E\left[\alpha\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}\right] \quad \text{avec } 1 \le \alpha \le 2$$

partie entière

paramètre de sur-échantillonnage

  - Coordonnées des points à tracer

$$\begin{cases} x_i = x_A + \dfrac{i}{L-1}(x_B - x_A) \\ y_i = y_A + \dfrac{i}{L-1}(y_B - y_A) \end{cases} \quad i = 0 \ldots L-1$$



Tracé de segments 2D (2/2)

- Tracé « épais »

  - Calcul du vecteur normé orthogonal n au segment AB

$$\begin{cases} n_x = -\dfrac{y_B - y_A}{\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}} \\ n_y = \dfrac{x_B - x_A}{\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}} \end{cases}$$

  - Multiples tracés « fin » de segments décalés $A_j B_j$

$$\begin{cases} A_j = A + j \times n \\ B_j = B + j \times n \end{cases} \quad j = \{-1,0,1\} \text{ ou } j = \{-2,-1,0,1,2\}\ldots$$

Initiation au Traitement des Images

VII – Implémentation

Calculs accélérés par vectorisation

- Problématique
  - Diminution du temps d'exécution par substitution de calculs « par ensembles » à des calculs « point à point »
    - Substitution de boucles « compilées » à des boucles « interprétées » (de moins en moins vrai avec le JIT Matlab !!!)
    - Réduction du nombre d'appels de fonctions « atomiques »
  - Nécessite souvent de réorganiser la séquence de calculs et de créer des structures de données intermédiaires
  - Calculs par ensembles
    - Expressions arithmétiques et logiques
    - Calculs vectoriels et matriciels
    - Fonctions mathématiques
    - Algorithmes à fenêtre glissante (convolution, etc…)
  - Accès vectorisés en lecture/écriture aux structures de données
  - Applicable également dans le contexte de calculs parallélisés (GPU)

Accès lecture/écriture vectorisés

- Accès multidimensionnel (par coordonnées ou par plage)
  - vecteurs
  - matrice 2D
  - matrice nD

```
v(2) = w(3)
v(4:5) = w(7:8)
A(8, 7) = B(10, 8)
A(2:4, 3:8) = B(1:3, 4:9)
A(5:6, 7) = B(1:2, 3)
A(2:3, 6, 4:7, 6, 2) = B(3:4, 2, 1:4, 5, 1)
```

- Accès vectorisé
  - lois de conversion « multidimensionnel → ordre de stockage »

`A(y,x)`

(1,1)	(1,2)
(2,1)	(2,2)
(3,1)	(3,2)

⟺

1	4
2	5
3	6

`A(y+(x-1)*h)`

  - accès aléatoires par listes appariées (vecteurs) de même cardinal
    - non ordonnés
    - non structurés par plage

```
V([14 2 6]) = A([25 30 7])
```

Un exemple (1/4)

- Rotation d'une image
  - autour de son centre
  - paramétrée par un angle
  - conservant ses dimensions initiales

$\theta = 20°$

$\theta = 45°$

$\theta = 70°$

Un exemple (2/4)

- Implémentation classique « point à point »

```
function B = rotate(A, angle)

[h, w] = size(A);
B = zeros(h, w);
c = cos(-angle*pi/180); s = sin(-angle*pi/180);
H = [1 0 -w/2;0 1 -h/2; 0 0 1];
H = [c -s -w/2;s c -h/2;0 0 1]*H;
H = [1 0 w/2;0 1 h/2; 0 0 1]*H;
for y2 = 1:h
  for x2 = 1:w
    M2 = [x2; y2; 1];
    M1 = H*M2;
    x1 = round(M1(1)/M1(3));
    y1 = round(M1(2)/M1(3));
    if ( x1 >= 1 && x1 <= w && y1 >= 1 && y1 <= h)
      B(y2, x2) = A(y1, x1);
    end
  end
end

end
```

Un exemple (3/4)

- Implémentation vectorisée « par ensembles »

```
function B = rotate(A, angle)

[h, w] = size(A);
B = zeros(h, w);
c = cos(-angle*pi/180); s = sin(-angle*pi/180);
H = [1 0 -w/2;0 1 -h/2; 0 0 1];
H = [c -s -w/2;s c -h/2;0 0 1]*H;
H = [1 0 w/2;0 1 h/2; 0 0 1]*H;
[X2, Y2] =  meshgrid(1:w, 1:h);
M2 = [X2(:)'; Y2(:)'; X2(:)'*0+1];
M1 = H*M2;
X1 = round(M1(1, :)./M1(3, :));
Y1 = round(M1(2, :)./M1(3, :));
pos = find((X1 >= 1).*(X1 <= w).*(Y1 >= 1).*(Y1 <= h))
B(Y2(pos)+(X2(pos)-1)*h) = A(Y1(pos)+(X1(pos)-1)*h);

end
```

Un exemple (4/4)

- Comparaison
  - temps d'exécution réduits (rotation)

	256×256	1024×1024	4096×4096
« point à point »	13,9 ms	216 ms	3,53 s
« par ensembles »	4,2 ms	98 ms	1,53 s

Matlab R2019b, Intel i7-8750H 2,21Ghz

  - gains variables
    - selon la séquence de calculs (créations de données intermédiaires coûteuses en temps et en mémoire)
    - selon la version de Matlab (le « Just in Time » du moteur d'interprétation ne cesse de s'améliorer !)

Fonctions Matlab autorisées

- Gestion de vidéos
  - Lecture (**videoReader**, **hasFrame**, **readFrame**)
  - Ecriture (**videoWriter**, **open**, **writeVideo**, **close**)
- Interaction
  - Sélection d'une région d'intérêt (**ginput**)
  - Affichages basiques (**imshow**, **image**, **imagesc**, **plot**, **line**, …)
- Calculs
  - Fonctions mathématiques (**exp**, **sin**, …)
  - Fonctions élémentaires **(min**, **max**, **mean**, **sum**, **floor**, **round**, **ceil**, **find**, **conv2**, …)
- Environnement
  - Sauvegarde de variables du workspace (**save**)
  - Chargement de variables dans le workspace (**load**)