

ENSTA — Filtrage bayésien et approximation particulière

TP 2 : Suivi visuel par histogramme de couleur

vendredi 15 octobre 2021

On souhaite réaliser un algorithme de suivi dans une séquence d'images numériques couleur. A la lecture de la première image de la séquence, l'utilisateur sélectionne une zone de l'image, et le suivi s'effectue de façon séquentielle sur l'ensemble de la séquence, voir Figure 1.

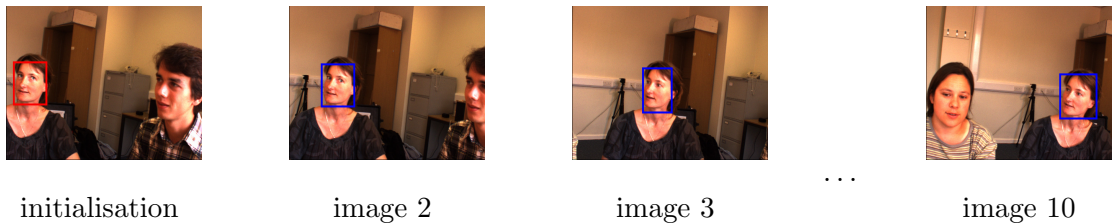


FIGURE 1 – Suivi d'un visage dans une séquence de 10 images

La méthode est construite sur l'algorithme SIR (souvent appelé algorithme CONDENSATION en vision par ordinateur). Elle repose sur l'hypothèse que l'histogramme de couleur de la zone à suivre est constant le long de la séquence. On pourra trouver dans [1] des informations détaillées sur cette méthode de suivi visuel.

1 Introduction aux images numériques

On désigne sous le terme d'image numérique toute image (dessin, icône, photographie, etc.) acquise, créée, traitée ou stockée sous forme binaire. On distingue généralement deux grandes catégories d'images :

- les images **vectorielles**, dont la description informatique est composée d'objets géométriques individuels (segments de droite, polygones, arcs de cercle, etc.), chacun défini par divers attributs de forme, de position, de couleur, etc.
- les images **matricielles**, représentées par un tableau à deux dimensions dont chaque case est un pixel (mot dérivé de l'anglais *picture element*, élément d'image). A chaque pixel est associée une ou plusieurs valeurs décrivant son niveau de gris ou sa couleur.

Les images vectorielles sont utilisées essentiellement en arts graphiques ou en CAO. Lorsque l'on s'intéresse au traitement d'images et à la vision par ordinateur, la représentation utilisée est la forme matricielle. Il existe plusieurs standards de codage de la couleur :

bitmap noir et blanc : en stockant un bit dans chaque case, il est possible de définir deux couleurs (noir ou blanc).

bitmap 256 niveaux de gris : en stockant un octet dans chaque case, il est possible de définir 256 dégradés de gris allant du noir au blanc.

palette de couleurs (colormap) : grâce à cette méthode, il est possible de définir une palette, ou table des couleurs, contenant l'ensemble des couleurs pouvant être présentes dans l'image, à chacune desquelles est associé un indice. Le nombre de bits réservé au codage de chaque indice de la palette détermine le nombre de couleurs pouvant être utilisées. On appelle ainsi image en couleurs indexées une image dont les couleurs sont codées selon cette technique.

couleurs vraies (true color) : le codage de la couleur est réalisé sur trois octets, chaque octet représentant la valeur d'une composante couleur par un entier de 0 à 255. Ces trois valeurs codent généralement la couleur dans l'espace RVB (rouge, vert, bleu), mais d'autres espaces de couleurs peuvent être utilisés. Le nombre de couleurs différentes pouvant être ainsi représentées est égal à $256 \times 256 \times 256$, soit près de 16 millions de couleurs.

Une image numérique est avant tout un signal $2D$. D'un point de vue mathématique, on considère l'image comme une fonction de $\mathbb{R} \times \mathbb{R}$ dans Ω où le couple d'entrée est une position spatiale sur la grille des pixels, et Ω est l'espace des valeurs de codage de la couleur (ou du niveau de gris). Par extension, on parlera d'images en dimension $2D + t$ (t pour le temps) pour désigner une séquence d'images numériques (ou vidéo numérique).

Remarque : Etant donné que l'écran effectue un balayage de gauche à droite et de haut en bas, on désigne généralement par les coordonnées $(0, 0)$ le pixel situé en haut à gauche de l'image, ce qui signifie que les axes de l'image sont orientés de la façon suivante : l'axe X est orienté de gauche à droite, l'axe Y est orienté de haut en bas, contrairement aux notations conventionnelles en mathématiques, où l'axe Y est orienté vers le haut.

2 Principe de l'algorithme de suivi visuel

Le but de ce TP est de construire un algorithme pour suivre une région d'intérêt dans une séquence d'images. Cette région est initialisée par l'utilisateur et sa forme est fixée *a priori*. On considèrera ici un rectangle, paramétré par la position, en pixel, du centre du rectangle $r = (x, y)$ et par un paramètre d'échelle s exprimé en %, c'est-à-dire que $s = 100$ signifie qu'un facteur d'échelle 1 est appliqué. Au pas de temps k (i.e. à l'image k), l'état du système à estimer sera donc $X_k = (r_k, s_k)$. Le paramètre d'échelle permet de suivre un objet même si celui-ci avance ou s'éloigne dans l'axe de la caméra (effet de zoom). A l'initialisation, l'utilisateur clique 4 points dans l'image, qui vont définir le rectangle initial. Celui-ci est décrit par les coordonnées du point haut/gauche, une largeur et une hauteur.

Equation d'état : On s'intéresse à la situation où aucune information *a priori* n'est disponible sur la nature de l'objet suivi. Dans ce cas, l'équation dynamique du système doit être peu informative. On supposera donc un modèle à position constante :

$$X_k = X_{k-1} + W_k, \quad (1)$$

où W_k est un bruit blanc gaussien, centré en 0 et de matrice de covariance C , matrice 3×3 diagonale. Les valeurs sur la diagonale sont c_1, c_2 et c_3 .

Notons que si la nature de l'objet suivi est connu, il est plus intéressant d'utiliser un modèle dynamique approprié. Par exemple, on pourrait imaginer utiliser un modèle à vitesse constante pour le suivi d'une voiture dans une vidéo acquise par une caméra sur autoroute.

Modèle de couleur : La zone initiale à suivre est caractérisée par un histogramme de couleur. Cet histogramme de référence est construit sur les Nb couleurs les plus représentatives de cette zone, comme montré sur la Figure 2 (la fonction `rgb2ind` permet de calculer les couleurs les plus représentatives d'une image). Cet histogramme de référence est noté $q^* = \{q^*(n), n = 1, \dots, Nb\}$, où $q^*(n)$ représente le nombre normalisé de pixels de la zone initiale dont la couleur la plus proche est la couleur n . On a $\sum_{n=1}^{Nb} q^*(n) = 1$. Pour plus d'informations sur les différents espaces de couleur, on pourra se reporter à la page *color space* sous Wikipedia.

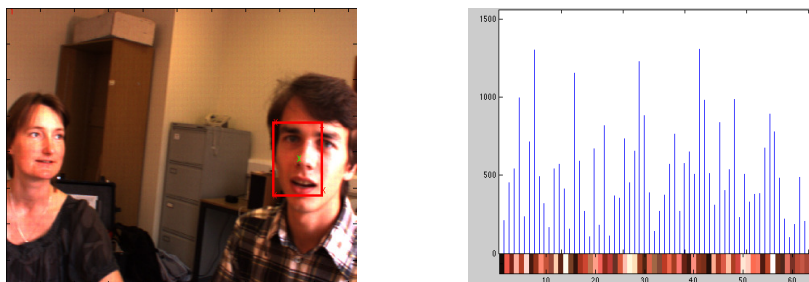


FIGURE 2 – Zone de l'image à suivre et histogramme de couleur associé pour $Nb = 64$

Comme décrit précédemment, le but est de suivre une zone de l'image le long de la séquence, sous l'hypothèse que son histogramme de couleur est invariant dans le temps. Au temps k , l'histogramme de couleur $q_k(x)$ d'un état hypothèse x sera comparé au modèle de couleur de référence q^* , et on définit la mesure de distance D entre ces deux histogrammes de couleur normalisés :

$$D(q^*, q_k(x)) = \left(1 - \sum_{n=1}^{Nb} \sqrt{q^*(n) q_k(x, n)}\right)^{1/2}. \quad (2)$$

Pour favoriser les états hypothèses dont l'histogramme de couleur associé est proche de l'histogramme de référence, on introduit la fonction :

$$g_k(x) \propto \exp\{-\lambda D^2(q^*, q_k(x))\}.$$

3 Mise en œuvre et validation de l'algorithme de suivi visuel

Structure de l'algorithme : La structure de l'algorithme de suivi visuel par méthode particulière est la suivante (le code MATLAB des étapes marquées d'une * est donné en annexe). On considère N particules à chaque pas de temps.

1. Lecture de l'ensemble des images de la séquence *
2. Initialisation des paramètres

3. Affichage de la première image *
4. Sélection et affichage de la zone à suivre *
5. Calcul de l'histogramme de couleur associé *
6. Initialisation des particules

7. Tant que la fin de la séquence n'est pas atteinte
 - (a) Diffusion des particules
 - (b) Calcul des histogrammes associés et des poids d'importance
 - (c) Normalisation des poids d'importance
 - (d) Affichage des particules sur l'image, et affichage de l'estimation (moyenne des particules)
 - (e) Ré-échantillonnage

Remarque sur le paramètre d'échelle : Pour la gestion du paramètre d'échelle s , il est conseillé d'utiliser les pourcentages. La valeur initiale s_0 est donc de 100%, puis cette valeur est perturbée lors de la diffusion des particules. La taille de la fenêtre associée à la i -ème particule de coordonnées $X_k^i = (r_k^i, s_k^i)$ est alors calculée en multipliant la taille de la fenêtre initiale par le facteur $s_k^i/100$.

Valeur des paramètres

N	100
Nb	10
λ	20
c_1, c_2	300
c_3	2

Séquences d'images disponibles : Deux séquences d'images `seq1.zip` et `seq2.zip` sont disponibles dans le répertoire

`people.rennes.inria.fr/Francois.Le_Gland/ensta/`

ainsi que le fichier `pre_visualisation.m` qui permet de pré-visualiser une séquence, et les fichiers `affichage_initial.m` et `selection_zone.m` présentés dans les Annexes. Ces séquences font partie d'une campagne d'acquisition audio-visuelle, dans le cadre du projet européen POP (Perception on Purpose).

Questions

1. En quoi la quantité D définie en (2) est-elle une mesure de distance ?
2. La diffusion des particules se fait selon la densité de probabilité de transition $p(x_k | x_{k-1})$. Quelle est l'expression de $p(x_k | x_{k-1})$? Comment sont évalués les poids d'importance des particules au temps k ?

3. Mettre en œuvre l'algorithme décrit ci-dessus. On prendra pour distribution de probabilité initiale des particules une distribution gaussienne, centrée autour de la zone initiale, et avec un petit écart-type.
4. Tester l'algorithme pour le suivi de différentes zones, i.e. une zone dont le mouvement est important, une zone qui ne bouge pas, une zone qui subit un mouvement de zoom, etc. Ne pas hésiter à modifier les valeurs de c_1 , c_2 , c_3 pour s'adapter aux différentes situations. Quelles sont vos conclusions ?
5. Tester l'algorithme sur une zone qui est occultée durant quelques images. Quelles sont vos conclusions ?
6. Tester l'algorithme sur une zone dont la couleur n'est pas discriminante dans l'image. Quelles sont vos conclusions ?
7. Selon vous, quelles sont les principales faiblesses de cet algorithme de suivi visuel ? Avez-vous des idées pour l'améliorer ?

Pour présenter les différents résultats et les commenter, afficher différentes images de la séquence et le résultat du suivi, comme sur la Figure 1 par exemple. Attention : il faut penser à arrêter le suivi si jamais l'estimation sort de l'image !

Référence bibliographique

- [1] Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet. Color-based probabilistic tracking. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Proceedings of the 7th European Conference on Computer Vision (ECCV'02), Copenhagen 2002*, volume 2350 of *Lecture Notes in Computer Science*, pages 661–675. Springer-Verlag, Berlin, June 2002.

4 Annexes

Lecture des images et affichage de la première image : Après initialisation du nom du répertoire où se trouve l'ensemble des images (`SEQUENCE`), ainsi que le numéro de la première image à traiter (`START`), le nom de l'ensemble des images est chargé. Puis, la première image est chargée (fonction `imread`) dans la variable `im` et affichée (fonction `imagesc`).

```
SEQUENCE = './seq2/';
START = 1;
% chargement du nom des images de la séquence
filenames = dir([SEQUENCE '*.png']);
filenames = sort({filenames.name});
T = length(filenames);
% chargement de la première image
tt = START;
im = imread([SEQUENCE filenames{tt}]);
% affichage de la première image
figure;
set(gcf, 'DoubleBuffer', 'on');
imagesc(im);
```

Sélection de la zone à suivre : L'utilisateur clique 4 points dans l'image dont les coordonnées sont récupérées. Ces coordonnées permettent d'initialiser la zone à suivre qui correspond au plus petit rectangle englobant ces 4 points. Les caractéristiques de rectangle, i.e. coordonnées du point haut/gauche, largeur, hauteur, sont sauvegardées dans le vecteur `zoneAT`.

```

disp('Cliquer 4 points dans l''image pour definir la zone a suivre.');
```

% sélection (à la souris) de la zone à suivre

```

zone = zeros(2,4);
compteur=1;
while(compteur ~= 5)
    [x,y,button] = ginput(1);
    zone(1,compteur) = x;
    zone(2,compteur) = y;
    text(x,y,'X','Color','r');
    compteur = compteur+1;
end
newzone = zeros(2,4);
newzone(1,:) = sort(zone(1,:));
newzone(2,:) = sort(zone(2,:));
% définition des paramètres de la zone à suivre
% x haut gauche, y haut gauche, largeur, hauteur
zoneAT = zeros(1,4);
zoneAT(1) = newzone(1,1);
zoneAT(2) = newzone(2,1);
zoneAT(3) = newzone(1,4)-newzone(1,1);
zoneAT(4) = newzone(2,4)-newzone(2,1);
% affichage du rectangle
rectangle('Position',zoneAT,'EdgeColor','r','LineWidth',3);
```

Construction d'un histogramme de couleur : A partir de l'image complète `im` et des coordonnées du rectangle `zoneAT`, on définit `littleim` qui est l'image de la zone à suivre (fonction `imcrop`). Puis on construit `Cmap` qui correspond aux Nb couleurs les plus représentatives de cette petite image (fonction `rgb2ind`). Enfin, on calcule `histoRef`, l'histogramme des couleurs de `littleim` défini sur la carte des couleurs `Cmap` et on le normalise. Cet histogramme est alors un tableau à Nb valeurs.

```

littleim = imcrop(im,zoneAT(1:4));
[~,Cmap] = rgb2ind(littleim,Nb,'nodither');
littleim = rgb2ind(littleim,Cmap,'nodither');
histoRef = imhist(littleim,Cmap);
histoRef = histoRef/sum(histoRef);
% affichage de l'histogramme
figure;
imhist(littleim,Cmap);
```

Pour le calcul de l'histogramme associé à chaque particule, il faut prendre garde de bien utiliser la même et unique base de couleur `Cmap` construite à l'initialisation.

```

impart = rgb2ind(impart,Cmap,'nodither');
histo = imhist(impart,Cmap);
histo = histo/sum(histo);
```

où `impart` est l'image qui correspond au rectangle défini par la particule.