

Chapitre 2



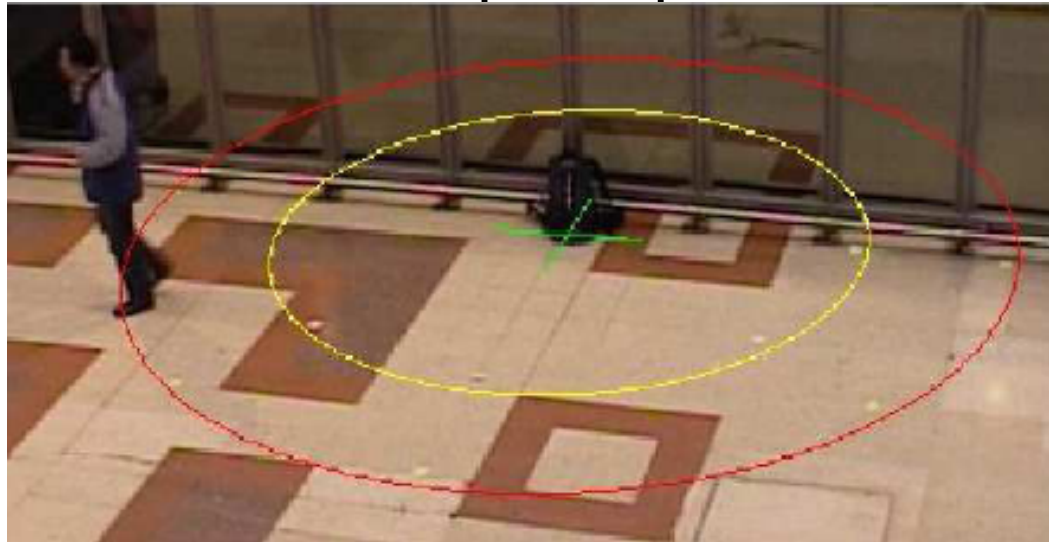
Extraction de régions d'intérêt par segmentation

Pourquoi la segmentation ?

- ❑ Détection d'objets abandonnés
 - Objectif: Déterminer si un objet est abandonné dans un lieu public pour prévenir, entre autre, des attentats.
 - Exemple:
 - ❑ Trouve les objets par soustraction d'arrière-plan.
 - ❑ Élimination des ombres.

Pourquoi la segmentation ?

- ❑ Détection d'objets abandonnés
 - ❑ Trouve le centroïde et le rectangle englobant.
 - ❑ Étudie le déplacement dans le temps des objets. Un bagage abandonné est un objet immobile durant 30 secondes, à plus de 3 m de celui ou celle qui le portait.



Pourquoi la segmentation ?

- ❑ Détection d'événements suspects
 - Objectif: Détecter des comportements définis comme anormaux.
 - Exemple:
 - ❑ Trouve les objets en mouvement par soustraction d'arrière-plan.
 - ❑ Trouve leurs centroïdes.
 - ❑ Calcule la trajectoire de chaque objet et leurs distances relatives à l'aide du centroïde.

Pourquoi la segmentation ?

- ❑ Détection d'événements suspects
 - ❑ Utilise un HMM (Hidden markov model) pour modéliser et détecter les états anormaux.
 - ❑ Permet de détecter des agressions ou vols.



Plan du chapitre

- ❑ Segmentation par détection de changements
 - Principe de base
 - Méthode de la moyenne temporelle
 - Méthodes paramétriques
 - ❑ Méthode simple distribution Gaussienne
 - ❑ Méthode amalgame de distributions Gaussiennes
 - Méthodes non-paramétriques
 - ❑ Différence avec méthodes paramétriques
 - ❑ Méthode ViBe
- ❑ Segmentation en régions
- ❑ Post-traitement
 - Composantes connectées, élimination des ombres

Détection de changements:

Principe de base

- S'il y a du mouvement dans une image, la valeur d'au moins un pixel sera différente sur deux images consécutives de la séquence vidéo.
 - Donc, faire la différence de deux images consécutives devrait nous permettre de détecter s'il y a eu un changement dans l'intervalle de temps.

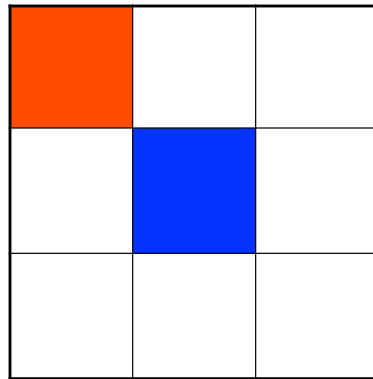


Image i

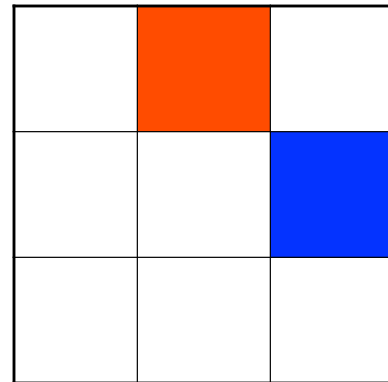


Image i+1

Détection de changements:

Principe de base

- ❑ Ne donne pas les résultats souhaités en pratique:
 - Le bruit du capteur et les changements de luminosité modifient les couleurs.
 - Détecte seulement les déplacements s'il y a changement de couleurs entre deux images consécutives.
 - Détecte l'ancien et le nouvel emplacement de l'objet.

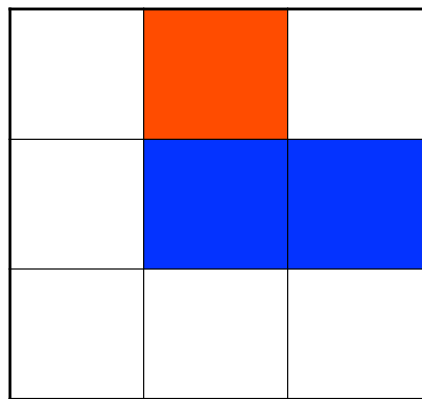


Image i

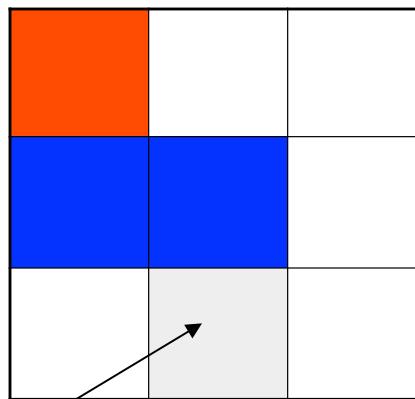
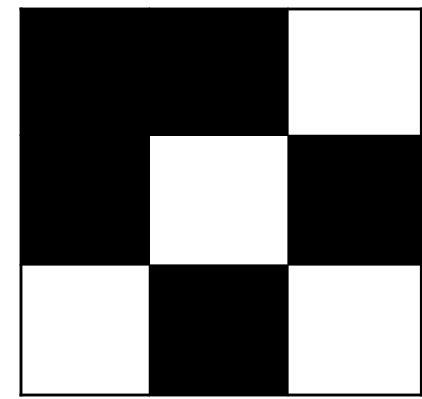


Image i+1



Différence

Détection de changements:

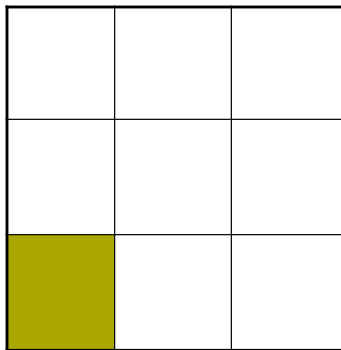
Principe de base

- ❑ La détection de changements est une forme de segmentation. Une segmentation est une opération qui vise à diviser l'image en n segments ayant des caractéristiques particulières (exemple, même couleur, même texture visuelle, même mouvement, etc.)
- ❑ En général, en détection de changement, on veut créer deux segments : 1) L'arrière-plan, 2) L'avant-plan (ce qui bouge).

Détection de changements:

Principe de base

- Utilisation d'une image de référence:
 - Fait la différence par rapport à un modèle d'arrière-plan où il n'y a pas de mouvement.
 - Détecte les nouveaux objets en mouvement et ceux arrêtés à leur emplacement courant.
 - Il reste à solutionner les problèmes liés au bruit et aux changements de la luminosité.



Référence

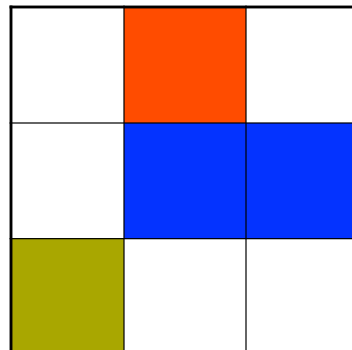
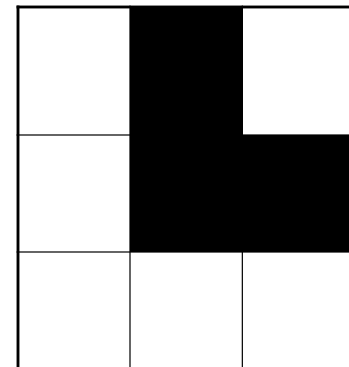


Image i



Différence

Méthode de la moyenne temporelle

- Pour éliminer le bruit et s'adapter aux changements, on fait une moyenne temporelle:
 - Création du modèle d'arrière-plan $B(x,y,t)$
 - Pour filtrer le bruit, on fait la moyenne des pixels sur les N premières images $I(x,y,t)$:

$$B(x,y,N) = \frac{1}{N} \sum_{t=1}^N I(x,y,t)$$

- Pour s'adapter aux changements, l'arrière-plan est mis à jour continuellement pour les pixels appartenant à l'arrière-plan dans l'image courante:

$$B(x,y,t) = \frac{B(x,y,t-1) * (t-1) + I(x,y,t)}{t}$$



Méthode de la moyenne temporelle

- Pour éliminer le bruit et s'adapter aux changements, on fait une moyenne temporelle:
 - Détection du mouvement:
 - Il y a du mouvement si:

$$|I(x, y, t) - B(x, y, t - 1)| > \tau$$

où τ est un seuil fixé manuellement.

Méthode de la moyenne temporelle



Méthodes paramétriques

- ❑ Les méthodes paramétriques supposent que le bruit affectant les pixels dans l'image suit une distribution de probabilité particulière, par exemple, une distribution Gaussienne (Loi normale).
- ❑ Ces méthodes essayent d'estimer les paramètres de la distribution de probabilité des pixels d'arrière-plan. Souvent, une distribution peut être représentée que par quelques paramètres, par exemple, moyenne et variance.

Méthode simple distribution Gaussienne

□ Hypothèses:

- Arrière-plan non-uniforme à cause du bruit du capteur.
- Le bruit suit approximativement une loi normale.
- Changements graduels de la luminosité.

□ Principe:

- Modélisation sous forme d'une distribution Gaussienne de chaque pixel de l'image de référence pour tenir compte du bruit.
- Mise à jour de l'image de référence pour tenir compte des changements graduels de la luminosité.

Méthode simple distribution Gaussienne

- Les valeurs d'un pixel est un processus dans le temps où à chaque instant t , l'historique des valeurs du pixel $\{x_0, y_0\}$ est connu et donné par:

- $$\{X_1, \dots, X_N\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\}$$

où I est la séquence vidéo.

- Ce processus de pixel peut être modélisé par une distribution Gaussienne:

- $$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-1/2 * (X_t - \mu_t)^T * \Sigma^{-1} (X_t - \mu_t)}$$

Méthode simple distribution Gaussienne

□ Modèle d'arrière-plan

- Une distribution Gaussienne est complètement définie par sa moyenne μ et sa variance σ^2 . Pour modéliser les pixels par cette distribution, il suffit de calculer la moyenne et la variance pour chaque couleur de chaque pixel sur quelques images.
- Chaque pixel a comme paramètres au temps t :

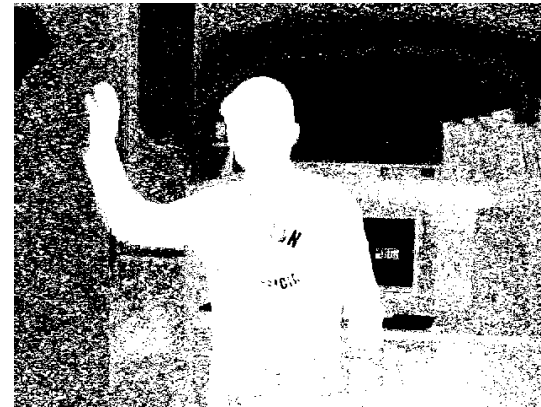
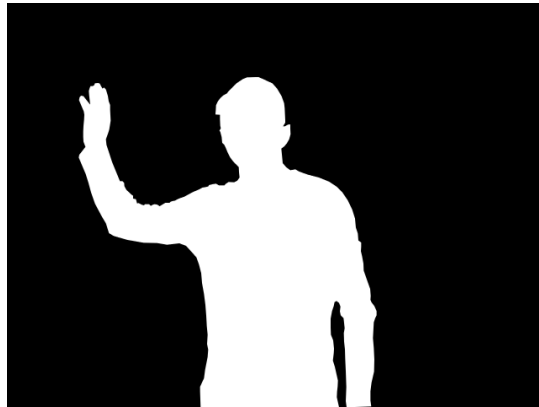
$$\mu_{t,r}, \mu_{t,g}, \mu_{t,b}, \sigma_{t,r}^2, \sigma_{t,g}^2, \sigma_{t,b}^2$$

- Un pixel X_t provient d'un objet en mouvement si:

$$|X_{t,r} - \mu_{t-1,r}| > n\sigma_{t-1,r} \text{ ou } |X_{t,g} - \mu_{t-1,g}| > n\sigma_{t-1,g} \text{ ou } |X_{t,b} - \mu_{t-1,b}| > n\sigma_{t-1,b}$$

où n est habituellement 2, 2.5 ou 3.

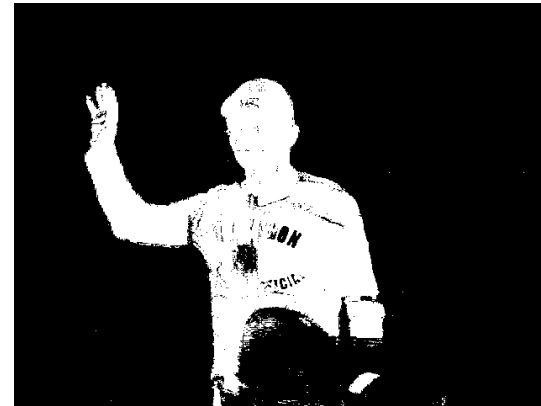
Méthode simple distribution Gaussienne



$n=2$



$n=5$



$n=10$

Méthode simple distribution Gaussienne

□ Mise à jour du modèle d'arrière-plan

- La distribution Gaussienne pour chaque pixel de l'arrière-plan est mise à jour pour tenir compte des changements de la luminosité.
- Cette mise à jour est sélective: seulement pour les pixels de l'image courante qui ne sont pas étiquetés comme pixels d'objets en mouvement.
- Pour la moyenne avec un pixel d'arrière-plan X_{t+1} :

$$\mu_{t+1} = \alpha \mu_t + (1 - \alpha) X_{t+1}$$

- Pour la variance:

$$\sigma_{t+1}^2 = \alpha (\sigma_t^2 + (\mu_{t+1} - \mu_t)^2) + (1 - \alpha) (X_{t+1} - \mu_{t+1})^2$$

où α est un coefficient d'adaptation

Méthode amalgame de distributions Gaussiennes (*Gaussian mixture*)

□ Hypothèse:

- Arrière-plan non-uniforme à cause du bruit du capteur et petits mouvements ou changements dans l'arrière-plan. Perturbations suivent une loi normale.
- Changements graduels de la luminosité.

□ Principe:

- Modélisation sous forme d'un amalgame de distributions Gaussiennes (*Gaussian mixture*) pour chaque pixel de l'image de référence pour tenir compte du bruit et des changements de l'arrière-plan.
- Mise à jour de l'image de référence pour tenir compte des changements graduels de la luminosité.

Méthode amalgame de distributions Gaussiennes

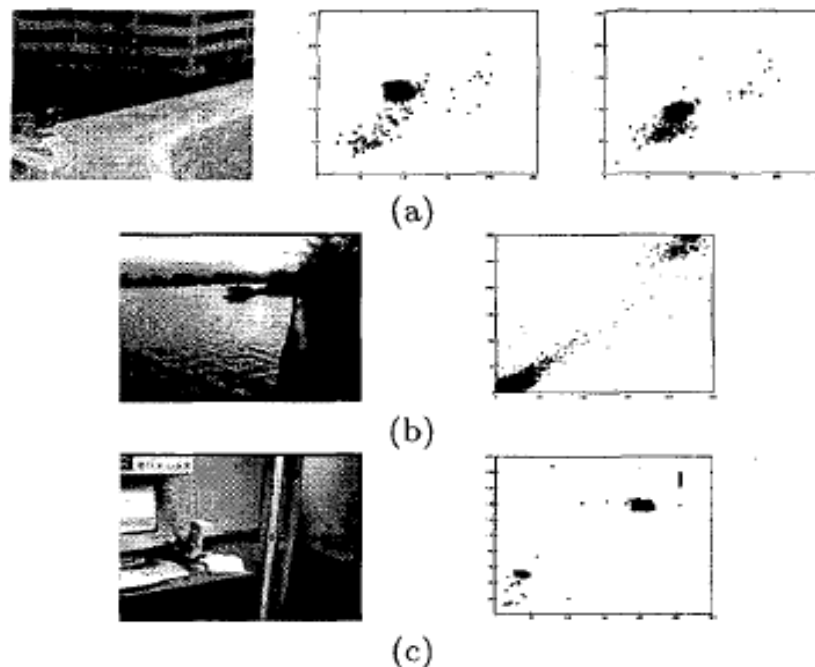


Figure 2: *This figure contains images and scatter plots of the red and green values of a single pixel from the image over time. It illustrates some of the difficulties involved in real environments. (a) shows two scatter plots from the same pixel taken 2 minutes apart. This would require two thresholds. (b) shows a bi-model distribution of a pixel values resulting from specularities on the surface of water. (c) shows another bi-modality resulting from monitor flicker.*

Méthode amalgame de distributions Gaussiennes

- Le processus de pixel est encore modélisé par une distribution Gaussienne:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-1/2 * (X_t - \mu_t)^T * \Sigma^{-1} (X_t - \mu_t)}$$

Avec $n=3$, pour 3 variables (R,G,B)

- Sauf que l'arrière-plan est considéré comme un processus qui peut avoir K distributions pour chaque pixel. Ainsi, la probabilité qu'une mesure X_t appartienne à l'arrière-plan est donnée par:

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

Méthode amalgame de distributions Gaussiennes

□ Modèle d'arrière-plan

- Pour faciliter les calculs, on suppose que les canaux R, G et B des pixels sont indépendants et de même variance, ainsi:

$$\Sigma_{i,t} = \sigma_i^2 I$$

- Ainsi, chaque pixel $X_t = [X_{t,r}, X_{t,g}, X_{t,b}]$ a comme paramètres au temps t :

$$\mu_t = [\mu_{t,r}, \mu_{t,g}, \mu_{t,b}], \sigma_t^2$$

- Un pixel X_t provient d'un objet en mouvement si:

$$|X_t - \mu_{t-1}| > 2.5\sigma_{t-1} \text{ pour les } B \text{ meilleures distributions.}$$

Méthode amalgame de distributions Gaussiennes

- Mise à jour de l'arrière-plan
 - Vérifie si la mesure X_t est dans une des K distributions:
 - Si $|X_t - \mu_{k,t-1}| \leq 2.5\sigma_{k,t-1}$ pour une distribution
 - $M_{k,t} = 1$ (Pixel d'arrière-plan)
 - Mise à jour des paramètres
 - Si $|X_t - \mu_{k,t-1}| > 2.5\sigma_{k,t-1}$ pour toutes les distributions
 - $M_{k,t} = 0$ (Nouvelle distribution).
 - Remplacement de la distribution de plus faible poids ω_k par la valeur de X_t comme moyenne, une variance élevée, et un poids faible.

Méthode amalgame de distributions Gaussiennes

□ Mise à jour des paramètres:

- 1) Ajustement des poids ω_k de chaque distribution:

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t})$$

- 2) Ajustement des paramètres de la distribution contenant X_t :

- $$\mu_{k,t} = (1 - \beta)\mu_{k,t-1} + \beta X_t$$

- $$\sigma_{k,t}^2 = (1 - \beta)\sigma_{k,t-1}^2 + \beta(X_t - \mu_{t,k})^T (X_t - \mu_{t,k})$$

Avec
$$\beta = \alpha\eta(X_t | \mu_{k,t-1}, \sigma_{k,t-1})$$

Méthode amalgame de distributions Gaussiennes

- Mise à jour des paramètres:
 - 3) Choix des B meilleures distributions sur les K au total:
 - Faible variance, grand échantillon.
 - Tri selon le ratio ω/σ des K distributions
 - Utilise les B meilleures distributions pour trouver les objets en mouvement.

$$B = \operatorname{argmin}_b \left(\sum_{k=1}^b \omega_k > T \right)$$

où T est la proportion minimum de données choisies pour modéliser l'arrière-plan.

Méthode amalgame de distributions Gaussiennes

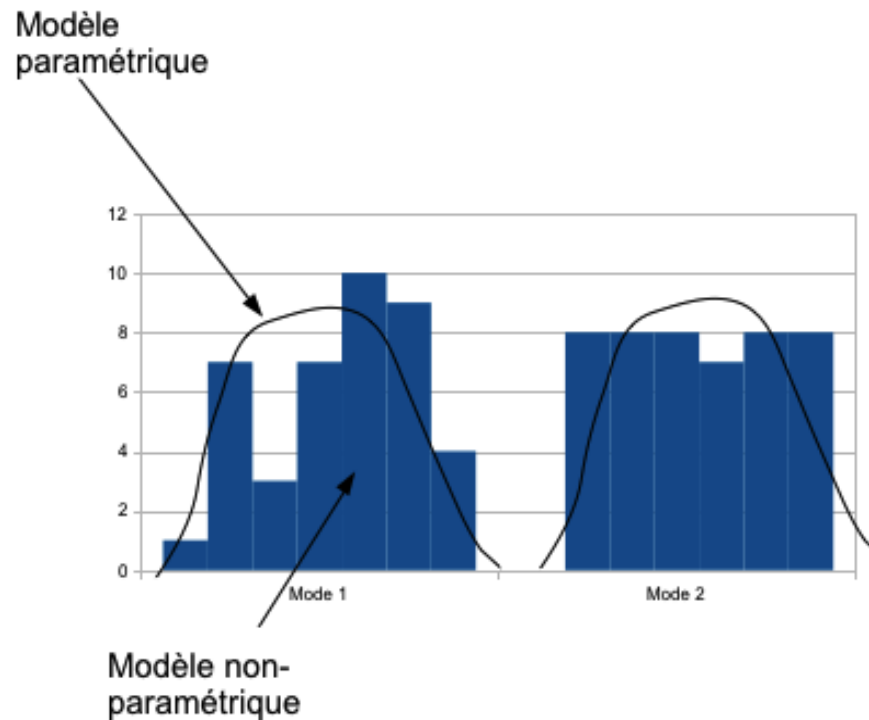


Méthodes non-paramétriques

- ❑ Les deux méthodes vues précédemment sont paramétriques, car l'arrière-plan est modélisé par une loi normale pour laquelle on estime les paramètres.
- ❑ En pratique, un arrière-plan n'est pas nécessairement paramétrisable avec une distribution de probabilité connue.
- ❑ Certains auteurs croient donc, que la distribution de probabilité de l'arrière-plan devrait être estimée sans paramètres.

Méthodes non-paramétriques

□ Différence avec méthodes paramétriques



Méthodes non-paramétriques

- ❑ Une méthode non-paramétrique estime la densité de probabilité avec un histogramme plutôt que par une loi de probabilité.
- ❑ Une méthode paramétrique requiert plus de mémoire pour stocker les densités de probabilité de chaque pixel.
- ❑ Par contre, modélise plus fidèlement les distributions.
- ❑ On peut utiliser plusieurs distributions d'arrière-plan pour un pixel.

- ❑ En pratique, les performances ne sont pas toujours meilleures que celles des méthodes paramétriques.

Méthode ViBe

- La méthode ViBe est une méthode non-paramétrique qui modélise l'arrière-plan par un nombre réduit d'échantillons. C'est-à-dire qu'on ne construit pas la distribution au complet pour limiter la mémoire requise.
- Principes de la méthode :
 - L'arrière-plan pour chaque pixel est modélisé par 20 échantillons (20 valeurs de couleur);
 - Les échantillons sont choisis au hasard dans le temps;
 - Pas de notion de temps pour les échantillons;
 - Un pixel d'arrière-plan aura une valeur proche de quelques valeurs parmi les 20 échantillons.

Méthode ViBe (suite)

- ▣ Modèle d'arrière-plan pour une position x

$$M(x) = \{v_1, v_2, \dots, v_n\}$$

où v_i est une valeur de pixel.

- ▣ Un pixel avec une valeur $v(x)$ fait partie de l'arrière-plan si

$$|S_R(v(x)) \cap \{v_1, v_2, \dots, v_n\}| \geq |\min|$$

où $S_R()$ est une sphère de rayon R centrée sur $v(x)$.
Habituellement, $\min = 2$ et $R=20$

Méthode ViBe (suite)

□ Initialisation

- À partir d'une seule trame. $M(x)$ est rempli avec 20 échantillons tirés au hasard autour de la position x .
- Utilise les 8 voisins pour le tirage.

□ Mise à jour du modèle

- La mise à jour est sélective;
- Un échantillon est remplacé au hasard selon une loi uniforme. Donc, on ne tient pas compte de l'âge de l'échantillon.
- La mise à jour est faite au hasard (pas à chaque trame). Typiquement : 1 chance sur 16

Méthode ViBe (suite)

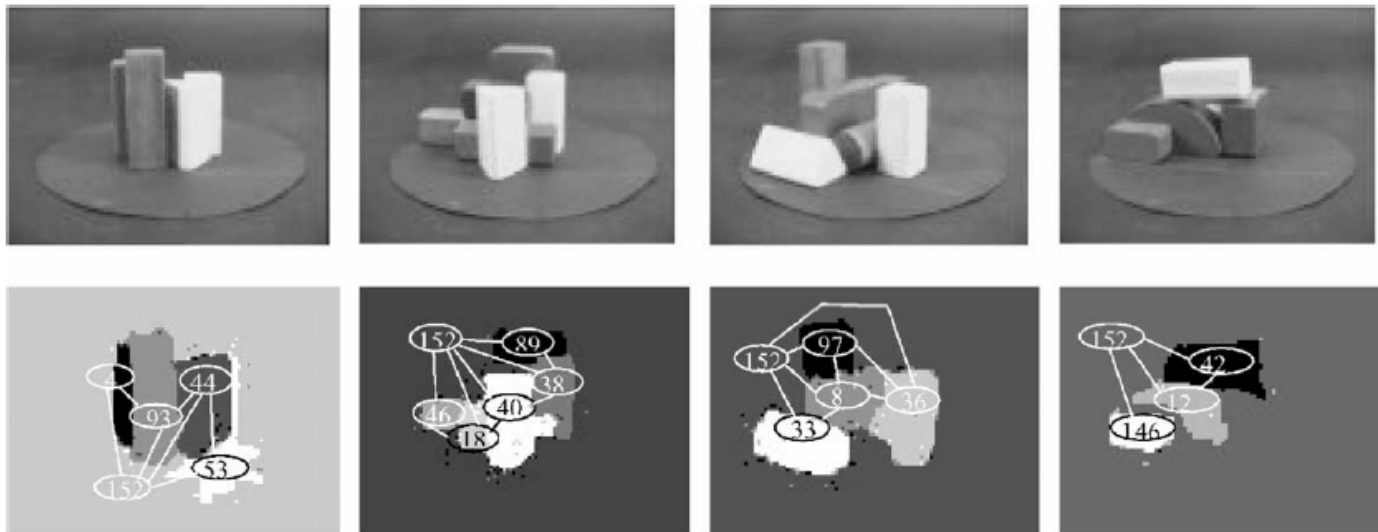
□ Mise à jour du modèle

- La mise à jour est propagée aux voisins de façon à mettre à jour aussi des pixels cachés par l'avant-plan.
- Met à jour un pixel voisin au hasard. (Voisinage 8 voisins).

Segmentation en régions

- Une segmentation est une opération qui vise à diviser l'image en n segments ayant des caractéristiques particulières (exemple, même couleur, même texture visuelle, même mouvement, etc.)
- En segmentation par régions, on segmente une seule image en régions homogènes. Le nombre de segments dépend habituellement du nombre d'objets de couleurs différentes.
- Le problème de la segmentation est de définir les seuils pour différencier les régions semblables de celles qui doivent être différentes.

Segmentation en régions



Segmentation en régions

- ❑ Segmenter un objet par les couleurs.
 - Exemple: Méthode par graphe.
 - ❑ Une image est un graphe où chaque pixel est un nœud et les relations de voisinage avec les pixels voisins sont des arcs.
 - ❑ Un nœud a un poids qui correspond à sa couleur et les arcs ont un poids correspondant à la différence des poids de la paire de nœuds liés.

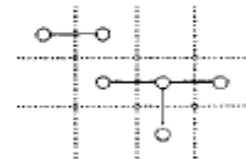
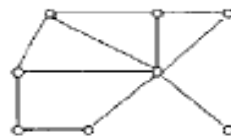
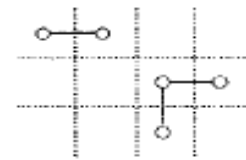
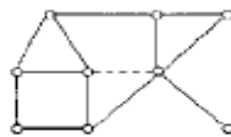
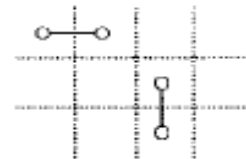
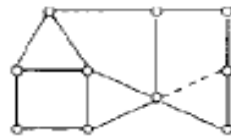
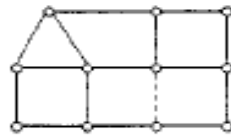
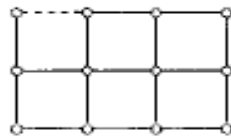
Segmenter un objet par les couleurs

■ (suite)

- Deux nœuds sont fusionnés si le poids de l'arc les liant est plus petit qu'un seuil.
- Le nœud fusionné a maintenant un poids qui correspond à la moyenne des couleurs des nœuds fusionnés.
- Les poids des arcs sont recalculés.
- On poursuit selon un ordre établi par le tri du poids des arcs, tant qu'il est possible de fusionner.

Segmentation en régions

■ (suite)



Segmentation en régions

- Segmentation en superpixels:
 - Idée: Réduire la complexité des traitements en groupant des pixels ensembles.
 - Nous allons étudier la méthode des SLIC superpixels:
 - Basée sur la méthode K -means;
 - Recherche limitée sur une région d'une taille proportionnelle au superpixel;
 - Le nombre de superpixels est K ;
 - Critère de groupement basé sur couleur et proximité spatiale.

Segmentation superpixels

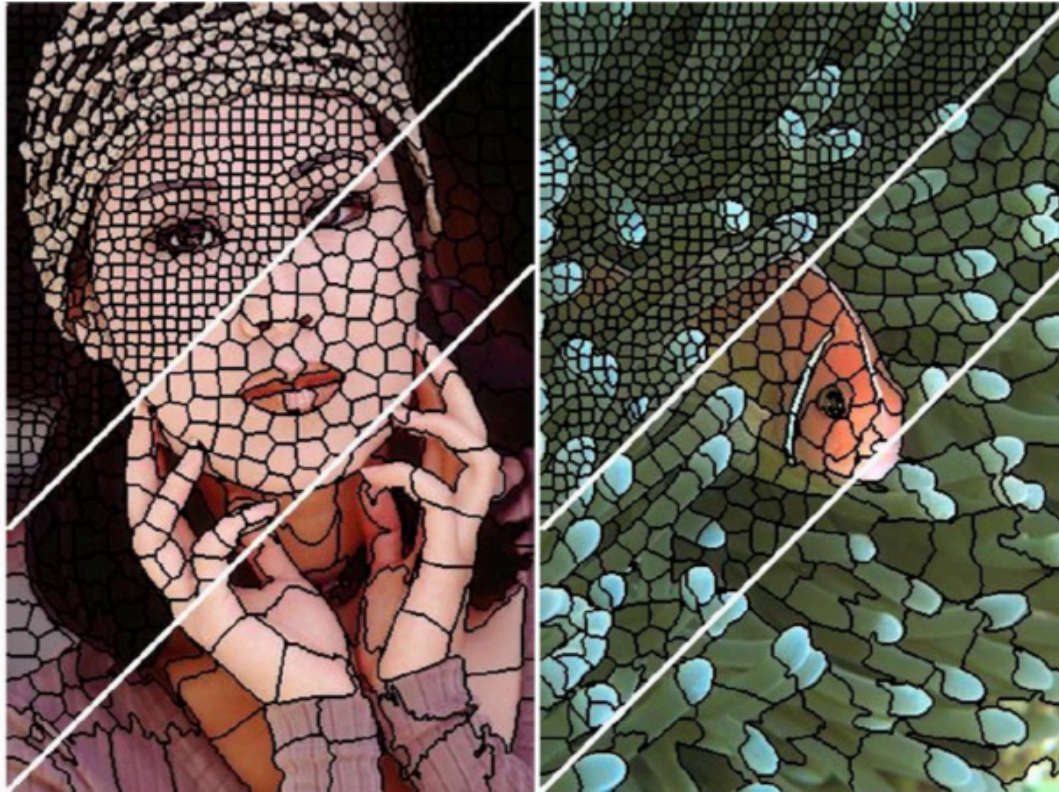


Fig. 1. Images segmented using SLIC into superpixels of size 64, 256, and 1,024 pixels (approximately).

Segmentation superpixels

Étape 1: Initialisation

- Les centres des groupes sont placés sur l'image en une grille régulière espacée de $s = \sqrt{\frac{N}{K}}$ où N est le nombre de pixels de l'image.
- Les centres sont ensuite ajustés à des positions de gradient plus faible en étudiant un voisinage de 3x3. Un superpixel ne devrait pas être initialisé sur une frontière.

Segmentation superpixels

Étape 2: Assignment des pixels aux centres

- Chaque pixel de l'image est assigné au centre le plus près selon la distance de similarité D et à une distance dans l'image plus petite que $2S$. C'est-à-dire que la distance D est calculée seulement pour les centres qui sont proches spatialement du pixel à assigner.
- La distance D est:

$$D = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2}$$

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

Segmentation superpixels

Étape 3: Mise à jour

- Le centre d'un groupe est ajusté à la moyenne des vecteurs $[l \ a \ b \ x \ y]$ de tous les pixels appartenant au groupe.
- On fait cette opération pour tous les centres.

Les étapes 2 et 3 sont répétées 10 fois (ou jusqu'à ce que les centres ne bougent plus).

On assigne les pixels disjoints au superpixel le plus près spatialement.

Post-traitement: Composantes connectées

- ❑ Pour grouper les pixels trouvés en région (BLOB: Binary Large Object).
- ❑ Méthode ligne par ligne avec structure "union-find"
 - Deux étapes:
 - ❑ 1) Propagation des étiquettes et recherche des équivalences.
 - ❑ 2) Assignment finale des étiquettes à l'aide des équivalences.
 - Structure "union-find": structure qui enregistre les étiquettes équivalentes (Étiquettes différentes pour une même région). Vecteur $1 \times N$, où N est le nombre d'étiquettes.

Post-traitement: Composantes connectées

- Propagation des étiquettes et recherche des équivalences.
 - Pour chaque pixel (ordre gauche à droite, haut vers le bas de l'image):
 - Si le pixel a une étiquette i , propager cette étiquette aux pixels immédiatement à droite et vers le bas qui ont été détectés comme pixels où il y a du mouvement. Si un des pixels où il y a propagation est déjà étiqueté par j , mettre dans la case j de la structure "union-find" l'étiquette i .
 - Si le pixel n'a pas d'étiquette et correspond à un pixel de mouvement, lui assigner l'étiquette $i+1$.

Post-traitement: Composantes connectées

- Propagation des étiquettes et recherche des équivalences.
 - Pour chaque pixel (ordre gauche à droite, haut vers le bas de l'image):
 - Si le pixel a une étiquette i , propager cette étiquette aux pixels immédiatement à droite et vers le bas qui ont été détectés comme pixels où il y a du mouvement. Si un des pixels où il y a propagation est déjà étiqueté par j , mettre dans la case j de la structure "union-find" l'étiquette i .
 - Si le pixel n'a pas d'étiquette et correspond à un pixel de mouvement, lui assigner l'étiquette $i+1$.

Post-traitement: Composantes connectées

- ❑ Assignation finale des étiquettes à l'aide des équivalences.
 - Pour chaque pixel (ordre gauche à droit, haut vers le bas de l'image):
 - ❑ 1) Si le pixel a une étiquette i , vérifier dans la case i de la structure "union-find", s'il y a une valeur d'étiquette.
 - ❑ 2) S'il y a une valeur j , vérifier ensuite dans la case j de la structure "union-find" s'il y a une valeur.
 - ❑ 3) Répéter 2), tant qu'il y a une valeur dans la case k .
 - ❑ 4) Assigner au pixel la dernière valeur d'étiquette trouvée.

Post-traitement: Composantes connectées

- Exemple MATLAB:
ConnectedComponentUF.m

Post-traitement: Élimination des ombres

- ❑ Après la soustraction d'arrière-plan des ombres peuvent faire partie de l'avant-plan
 - Elles peuvent être éliminées en partie.
 - Quelques méthodes:
 - ❑ Chromaticité et gradient
 - ❑ Seuil en HSV

Post-traitement: Élimination des ombres

□ Chromaticité et gradient

- Hypothèse: L'ombre cause un changement dans l'intensité, mais la chromaticité reste semblable.
- La chromaticité est donnée par:
 - $r_c = r / (r + g + b)$
 - $g_c = g / (r + g + b)$
- On souhaite faire la soustraction d'arrière-plan avec des distributions Gaussiennes pour r_c et g_c , mais si l'intensité n'est pas prise en compte, certains objets en mouvement ne pourront pas être détectés (si seule l'intensité les différencie de l'arrière-plan).

Post-traitement: Élimination des ombres

□ Chromaticité et gradient

- Donc, on tient compte aussi du gradient à chaque pixel de l'image.
- On calcule le gradient pour chaque canal de couleur RGB $(r_x, r_y, g_x, g_y, b_x, b_y)$ à chaque position (x, y) dans l'image.
- On intègre l'information du gradient par sa moyenne $(\mu_{xr}, \mu_{yr}, \mu_{xg}, \mu_{yg}, \mu_{xb}, \mu_{yb})$ et la variance de son module $(\sigma_{xyr}, \sigma_{xyg}, \sigma_{xyb})$ comme pour les couleurs. On calcule aussi la variance globale des modules des gradients calculés $(\sigma_{gr}, \sigma_{gg}, \sigma_{gb})$.

Post-traitement: Élimination des ombres

□ Chromaticité et gradient

- On a un pixel d'avant-plan si une des relations suivantes est vrai:

$$\sqrt{(r_x - \mu_{xr})^2 + (r_y - \mu_{yr})^2} > n * \max(\sigma_{xyr}, \sigma_{gr})$$

$$\sqrt{(g_x - \mu_{xg})^2 + (g_y - \mu_{yg})^2} > n * \max(\sigma_{xyg}, \sigma_{gg})$$

$$\sqrt{(b_x - \mu_{xb})^2 + (b_y - \mu_{yb})^2} > n * \max(\sigma_{xyb}, \sigma_{gb})$$

$$|r_c - \mu_{rc}| > n * \sigma_{rc}$$

$$|g_c - \mu_{gc}| > n * \sigma_{gc}$$

n est un seuil.

Post-traitement: Élimination des ombres



Post-traitement: Élimination des ombres

□ Seuil en HSV

- Hypothèse: L'ombre cause des changements faibles dans les canaux H, S et V. Si le changement est trop grand, ce n'est pas une ombre.



Post-traitement: Élimination des ombres

□ Seuil en HSV

- Le pixel I est un pixel d'ombre si par rapport au pixel d'arrière-plan B :

$$\alpha \leq \frac{I_v}{B_v} \leq \beta \wedge |I_s - B_s| \leq \tau_s \wedge \min(|I_h - B_h|, 360 - |I_h - B_h|) \leq \tau_h$$

- α et β définissent des bornes sur le noircissement de l'intensité dû à l'ombre.
- τ_s, τ_h sont des seuils.

Post-traitement: Élimination des ombres



(a)



(b)



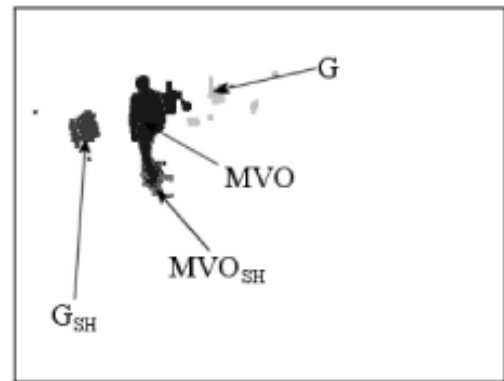
(c)



(d)



(e)



(f)

Bibliographie

- E. Auvinet et al., Left-Luggage Detection using Homographies and Simple Heuristics, in 9th IEEE International Workshop on PETS, 2006
- R. Cucchiara et al., Detecting Moving Objects, Ghosts, and Shadows in Video Streams, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 10, 2003, pp. 1337-1342
- O. Barnich, M. Van Droogenbroeck, ViBe: A universal background subtraction algorithm for video sequences, IEEE transaction on Image Processing, 20(6), 2011, pp. 1709-1724
- X. Lui, C.S. Chua, Multi-agent activity recognition using observation decomposed hidden Markov models, Image and Vision Computing 24 (2006), pp. 166–175
- S.J. McKenna et al., Tracking Groups of People, Computer Vision and Image Understanding Vol. 80, 2000, pp. 42–56
- C. Stauffer, W.E.L. Grimson, Adaptive background mixture models for real-time tracking, in CVPR 1999
- L.G. Shapiro, G.C. Stockman, Computer Vision, Prentice-Hall, 2001
- R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, SLIC Superpixels Compared to State-of-the-Art Superpixel Methods, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, No. 11, 2012, pp. 2274-2281