Modélisation probabiliste pour la segmentation multi-vues

Abdelaziz Djelouah, Jean-Sébastien Franco, Edmond Boyer, Francois Le Clerc, Patrick Pérez

Modelisation probabiliste pour la segmentation multi-vues

A.Djelouah[1,2] J-S.Franco[1] E.Boyer[1] F. Le Clerc [2] P.Pérez[2]

[1] LJK - INRIA Rhône-Alpes [2] Technicolor R&I

Résumé

Dans cet article nous abordons le problème de la segmentation multi-vues d'un objet vue par un ensemble de caméras calibrées. La principale difficulté est de formuler au mieux les contraintes inter-vues afin de propager un maximum d'information sur la segmentation de l'objet avec une complexité réduite. Pour cela, nous proposons un modèle probabiliste basé sur l'idée qu'examiner les informations à la projection d'un ensemble épars d'échantillons de points 3D doit être suffisant pour résoudre le problème de segmentation multi-vues. L'algorithme proposé classe chaque point comme "vide" s'il se projette sur une région du fond et "occupé" s'il se projette sur une région avant-plan dans toutes les vues. L'étude présente aussi un ensemble d'expérimentations qui valident l'approche proposée, avec des résultats équivalents à l'état de l'art et une complexité réduite.

Mots Clef

Segmentation, multi-vues.

Abstract

We address the problem of multiple view segmentation of objects viewed by a set of calibrated cameras. A key difficulty is to formulate inter-view propagation in a way that maximizes the propagation of segmentation information between views, while minimizing complexity and computational cost. To this goal, we first propose the n-tuple model, relying on the idea that examining measurements at the n projections of a sparse set of 3D points is sufficient to achieve this goal. The algorithm proposed classifies each point as empty if it projects on a background region in at least one view, or occupied if it projects in all foreground regions. The paper exposes a detailed set of experiments used to validate the algorithm, showing results comparable to the state of art, with reduced computational complexity.

1 Introduction

La segmentation d'objets d'intérêt est la première étape pour de nombreuses applications en vision par ordinateur, telles que l'analyse de scène, le "matting", le "compositing" pour la post-production, l'indexation d'image et la compression, ainsi que la reconstruction 3D. Dans de nombreuses situations, plusieurs points de vue de l'objet sont disponibles. Cette redondance d'information a été exploitée dans de nombreux travaux de recherche, afin d'améliorer la robustesse de la segmentation et d'en permettre une plus grande automatisation.

Il existe deux approches principales pour résoudre le problème de segmentation multi-vues : la co-segmentation, qui s'appuie sur une apparence commune de l'objet dans les différentes vues et une grande variabilité de l'apparence de l'arrière-plan, et l'extraction de silhouettes, qui cherche à extraire des segmentations géométriquement cohérentes. Dans cette deuxième approche, une forme de reconstruction dense de l'objet est utilisée, que ce soit dans l'espace 3D ou dans les vues. Ces deux types d'approches ont leurs limitations. La dépendance à l'apparence devient rapidement un obstacle à la co-segmentation dès que les points de vues sont éloignés, induisant des changements importants dans l'apparence de l'objet d'intérêt, ou dès lors que les arrière-plans des différentes vues présentent une trop grande similarité. La complexité de l'estimation de la représentation dense est le principal inconvénient de l'approche extraction de silhouettes. Un article récent [11], qui s'appuie sur des informations de profondeur obtenues par reconstruction stéréo, met en avant le potentiel de rapprochement entre ces deux méthodes.

Nous proposons ici un cadre probabiliste pour la segmentation multi-vues, qui formule le problème comme une estimation en maximum *a posteriori* des paramètres du modèle couleur de l'avant-plan et du fond. La solution est obtenue en utilisant une approche d'Espérance-Maximisation. Le modèle génératif proposé est décrit dans le §4. La méthode est ensuite testée dans différentes configurations, permettant une comparaison qualitative et quantitative avec l'état de l'art.

2 État de l'art

Segmentation monoculaire. Plusieurs approches existent pour la segmentation fond/objet. Les techniques bas-niveau raisonnent au niveau pixel, avec l'hypothèse d'une apparence d'arrière-plan constante [20]. Certaines de ces méthodes, telles que [17, 18], prennent aussi en compte une certaine variabilité du fond au cours du temps. L'avantage principal de ces approches est leur efficacité calculatoire, mais les hypothèses sur le fond sont souvent très restrictives. Des techniques plus récentes lèvent partiellement ces restrictions en formalisant le problème comme l'extraction

de l'objet d'intérêt à partir d'une distribution initiale [3] ou d'une ré-estimation itérative [13] de l'apparence de l'objet et de l'arrière plan. Elles nécessitent toutefois une interaction utilisateur pour définir les modèles initiaux d'apparence.

Segmentation multi-vues. Plusieurs méthodes ont été proposées pour segmenter un objet vu par plusieurs caméras en utilisant l'information couleur. L'une des premières approches orientées multi-vues a été proposée par Zeng *et al.* [21]. Ici, les silhouettes des objets sont définies comme étant la réunion des régions associées à un ensemble de superpixels, ces régions étant examinées de manière itérative afin de vérifier leur cohérence avec l'enveloppe visuelle des silhouettes courantes. Toutefois cette approche est sensible à toute mauvaise décision ou erreur de classification durant le processus.

D'autres travaux proposent d'extraire simultanément les modèles 3D et les silhouettes 2D correspondantes, fournissant une solution indirecte à la segmentation multi-vues [15, 6, 9]. Ces approches font l'hypothèse d'une certaine connaissance *a priori* de l'apparence de l'avant plan et/ou de l'arrière plan. Le problème est formulé de manière géométrique en utilisant des graph cuts [15], une approche probabiliste [6] ou une convexification du problème [9], avec de bonnes propriétés de convergence. D'autres méthodes procèdent également à une ré-estimation des paramètres d'apparence en se basant sur des processus plus complexes et plus coûteux [4, 5, 7]. Ces méthodes ont en commun de construire explicitement une enveloppe visuelle dense. D'autres travaux ciblant plus particulièrement la segmentation multi-vues proposent des solutions dédiées. Par exemple, Lee *et al*. [12] utilise la probabilité d'occupation le long des lignes de vue, et itère sur chaque vue pour propager l'information. Ces solutions restent tout aussi complexes.

Approches de co-segmentation. Dans l'un des premiers travaux sur cette approche, Rother*et al.* [14] définissent la co-segmentation comme étant la segmentation binaire simultanée de régions dans une paire d'images et, par extension, dans plusieurs images [1, 8, 19]. L'hypothèse clé de cette famille de méthodes est que l'objet d'intérêt présente la même apparence dans les différentes vues, alors que celle de l'arrière-plan varie suivant les images. Les approches de co-segmentation n'exploitent pas la cohérence géométrique entre les vues, à l'exception de [10], qui utilise un modèle 3D planaire par morceaux s'appuyant sur une reconstruction stéréo, et illustre la convergence entre les deux familles de méthodes. D'excellents résultats sont atteints mais au prix d'une formulation 3D avec une contrainte de distance inter-vues faible. [16] requiert également une faible distance inter-vues, mais sans utiliser de stéréo 3D.

Dans cet article, nous introduisons une nouvelle approche pour la segmentation multi-vues, évitant toute représentation 3D dense, et exprimant au mieux les différentes contraintes liées à la segmentation multi-vues.

3 Principe

La segmentation multi-vues vise à isoler un objet d'intérêt de l'arrière-plan de la scène, sous l'hypothèse que l'objet d'intérêt soit visible dans toutes les vues. [12] propose une définition plus formelle de l'objet d'intérêt, qui doit satisfaire deux contraintes : être entièrement visible dans toutes les vues, et avoir une apparence différente de l'arrière plan.

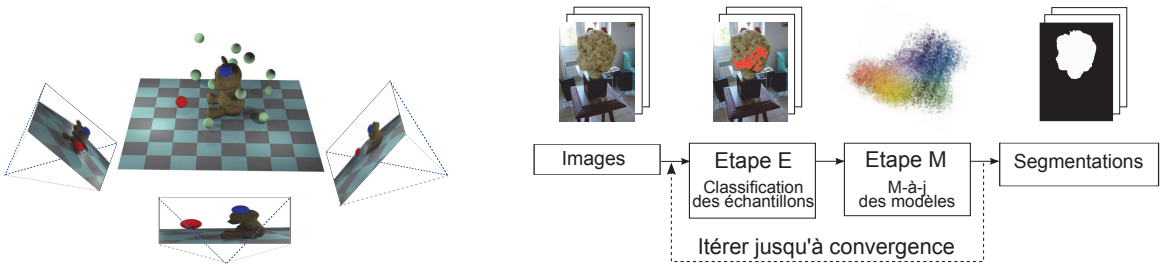
Nous proposons un cadre probabiliste innovant pour la segmentation multi-vues. Notre approche présente l'avantage d'une complexité calculatoire réduite, notamment parce qu'elle ne requiert pas une représentation dense de l'objet. Elle s'appuie sur un échantillonnage de l'espace 3D commun aux champs de vision des n caméras, et ne considère que l'ensemble des couleurs à la projection de chaque échantillon, dénommé n-uplet (voir Fig. 1(a)). La cohérence spatiale de l'avant-plan à travers les vues est exprimée exclusivement au moyen de ces n-uplets, et n'exploite pas la connaissance des coordonnées 3D des échantillons.

Un modèle génératif pour les étiquettes des échantillons est défini à partir de l'intuition suivante. Si un échantillon fait partie de l'objet d'avant-plan, alors toutes les couleurs devraient être simultanément décrites par le modèle couleur de l'avant-plan. En revanche, si un échantillon ne fait pas partie de l'objet d'intérêt, alors il existe une vue pour laquelle la couleur correspondante de l'échantillon est prédite par le modèle d'apparence d'arrière-plan de cette vue, et les autres couleurs sont indifférentes. Ainsi, on affecte à chaque échantillon s une variable d'étiquetage k_s, prenant ses valeurs dans $\mathcal{K} = \{\mathrm{f}, \mathrm{b}_1, \cdots, \mathrm{b}_n\}$. f est l'état avant-plan et b_i l'étiquette d'arrière-plan associée à la vue i. Le problème de segmentation multi-vues est formulé dans un cadre d'estimation de maximum *a posteriori* (MAP), où les variables de classification des n-uplets sont latentes (§4). La solution de ce problème de MAP fournit une première segmentation qui sera finalement raffinée pour chaque vue (§6).

4 Modélisation

4.1 Modèle génératif

Soit $\mathcal{S}$ l'ensemble des échantillons 3D sélectionnés. Le n-uplet associé à l'échantillon $s \in \mathcal{S}$ est $(I_1^s, \cdots, I_n^s)$, et $k_s \in \mathcal{K}$ est son étiquette. π_k sont les coefficients de mélange (à estimer), représentant la proportion d'échantillons expliqués par chaque hypothèse dans $\mathcal{K}$. On notera Θ_i^c les paramètres des distributions couleur (fond et objet) associées à chaque image i. Le modèle graphique probabiliste correspondant est représenté Fig. 2. Les couleurs de chaque n-uplet sont prédites à partir de sa variable de classification k_s avec comme *a priori* π_k et Θ_i^c. Il est intéressant de remarquer que cette modélisation traite le problème de segmentation multi-vues comme l'estimation d'un mélange de modèles objet/fond.



(a) Echantillonnage (b) Principales étapes

FIGURE 1 – Aperçu global de la méthode : (a) Les échantillons sont représentés par des sphères en 3D pour une bonne visualisation. L'échantillon bleu est étiqueté "objet" car il se projette dans la région avant-plan dans toutes les vues. L'échantillon rouge est étiqueté "fond", car deux caméras le classent dans l'arrière-plan. (b) Schéma algorithmique : la méthode itère entre classification des échantillons et mise à jour des modèles. Une segmentation finale est effectuée pour transférer la classification éparse des échantillons vers une classification dense des pixels.

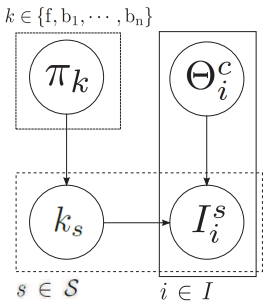


FIGURE 2 – Modèle graphique : I_i^s, La couleur de la projection dans la vue i de l'échantillon s, dépend des modèles couleurs Θ_i^c et de l'étiquette k_s. π_k représente les coefficients de mélange.

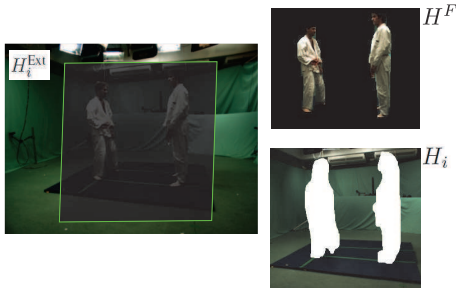


FIGURE 3 – Les régions des supports des différents histogrammes couleur, définis à partir de la région d'intérêt R_i pour la vue i : H_i^{Ext} pour les pixels d'arrière-plan connus (R_i^c), H_i pour les pixels "fond" dans R_i et H^F pour les pixels d'avant-plan (histogramme partagé entre les vues).

4.2 Modèles couleur

Différents modèles couleur peuvent être utilisés pour les Θ_i^c, comme les histogrammes ou les mélanges de gaussiennes [17]. On note R_i la région d'intérêt dans l'image i qui est supposée contenir tout l'avant plan. R_i peut être estimée automatiquement à partir du champ de vision commun des caméras ou initialisée plus finement à partir d'une interaction utilisateur. On fait le choix de représenter les distributions couleur par des histogrammes. En particulier, nous utiliserons un histogramme par vue pour l'arrière plan, noté H_i, et un histogramme partagé entre les vues pour l'avant plan, noté H^F. La région d'intérêt R_i sera elle aussi décrite grâce un histogramme H_i^{Int} (voir Fig. 3). De ce fait, l'ensemble des modèles couleur est entièrement paramétré par $\Theta^c = \{H^F, H_i | i \in \{1, \cdots, n\}\}$. Au vu de nos hypothèses, tous les pixels à l'extérieur de la région d'intérêt sont des pixels d'arrière plan. Ils seront utilisés plus tard comme *a priori* sur les distributions arrière plan. H_i^{Ext} est l'histogramme associé à cette région, notée R_i^c. On notera que $\Theta_i^c = \{H^F, H_i\}$.

4.3 Distribution de probabilité conjointe

Notre objectif est de trouver les paramètres qui maximisent la densité *a posteriori* connaissant les observations. En notant $K = \{k_s\}_{s \in \mathcal{S}}$, $I = \{I_i^s\}_{s \in \mathcal{S}, i \in \{1, \cdots, n\}}$, Θ^c et $\pi = \{\pi_k\}_{k \in \mathcal{K}}$, la distribution de probabilité conjointe se factorise suivant s :

$$p(\Theta^c, I, \pi, K) = p(\Theta^c) p(\pi) \prod_{s \in S} p(k_s, I_1^s, ..., I_n^s | \Theta^c, \pi), \tag{1}$$

où $p(\pi)$ est uniforme et sera ignorée par la suite. Pour un échantillon donné s on a (voir Fig. 2)

$$p(k_s, I_1^s, ..., I_n^s | \Theta^c, \pi) = \left[\prod_i^n p(I_i^s | \Theta_i^c, k_s) \right] p(k_s | \pi). \tag{2}$$

Si un échantillon est classé "objet" (avant-plan), alors toutes les couleurs du n-uplet associé doivent être prédites par le modèle couleur d'avant-plan. Mais si un échantillon est classé comme arrière-plan pour la vue i (étiquette b_i) alors la $i^{ème}$ couleur du n-uplet doit être prédite par le modèle couleur du fond pour cette vue i. Les autres couleurs

sont indifférentes, ce qu'on modélise au moyen de la distribution image H_i^{Int} :

$$p(I_i^s|\Theta_i^c, k_s) = \begin{cases} H_i(I_i^s) & \text{si} \quad k_s = \text{b}_i, \\ H^F(I_i^s) & \text{si} \quad k_s = \text{f}, \\ H_i^{\text{Int}}(I_i^s) & \text{sinon.} \end{cases} \tag{3}$$

C'est à ce niveau qu'est effectuée la classification par vue des échantillons. Si un échantillon vérifie le modèle couleur d'arrière-plan pour une vue i, il n'a pas besoin d'être comparé aux autres modèles couleur dans les autres vues. Il doit juste vérifier le modèle couleur de la région d'intérêt $H_j^{\text{Int}},\ j \neq i$.
Le terme $p(k_s|\pi_{k_s})$ représente l'*a priori* sur les proportions dans le mélange

$$p(k_s|\pi_{k_s}) = \pi_{k_s}. \tag{4}$$

4.4 Apriori à partir des pixels de l'arrière plan connu

Nous souhaitons renforcer la similarité entre la distribution des pixels d'arrière-plan et les couleurs présentes dans la région R_i^c. Dans ce but, nous considérons des échantillons 3D qui se projettent dans R_i^c et qui, de ce fait, ont une étiquette "fond", et nous définissons l'*a priori* suivant sur les modèles de fond Θ^c :

$$p(\Theta^c) = \prod_i \prod_{s \in \mathcal{S}_i} H_i(I_i^s) \tag{5}$$

avec $\mathcal{S}_i$ l'ensemble de ces échantillons 3D. Ainsi, un ensemble de paramètres d'histogrammes H_i est plus probable s'il explique correctement des pixels de R_i^c, dont l'étiquette arrière-plan est connue.
Exprimer cette contrainte en terme de pixels, au lieu d'échantillons 3D, se traduit par

$$p(\Theta^c) = \prod_i \prod_{p \in R_i^c} (H_i(I_i^p))^{t_p}, \tag{6}$$

où t_p indique pour chaque pixel le nombre d'échantillons 3D qui se projettent dessus.
Puisque nous ne voulons pas créer des échantillons à l'extérieur du champ de vision commun, nous allons approximer la valeur t_p par λ_i, le nombre moyen d'échantillons se projetant sur un unique pixel dans R_i. Notre contrainte s'écrit donc

$$p(\Theta^c) = \prod_i \prod_{p \in R_i^c} (H_i(I_i^p))^{\lambda_i}. \tag{7}$$

5 Algorithme d'estimation

Comme le problème se traduit en une estimation d'un maximum *a posteriori* avec des variables latentes, nous le résolvons au moyen d'un algorithme d'espérance-maximisation (EM). L'EM consiste en un processus itératif qui alterne entre une étape d'évaluation de l'espérance (E) et une étape de maximisation (M), où l'on estime le maximum de vraisemblance des paramètres en maximisant la vraisemblance trouvée à l'étape E. Dans notre cas, $\Phi = \{\Theta^c, \pi\}$. Nous construisons les étapes E et M en utilisant la fonctionnelle EM générique Q, dont les propriétés de convergence sont bien établies [2] :

$$Q(\Phi, \Phi^g) = \sum_K \log(p(I, K, \Phi)) p(K|I, \Phi^g) \tag{8}$$

$$\begin{aligned} Q(\Phi, \Phi^g) \quad = & \sum_K \log(\prod_s p(k_s, I_1^s, ..., I_n^s|\Phi)) \\ & \prod_{s\prime} p(k_{s\prime}|I_1^{s\prime}, ..., I_n^{s\prime}, \Phi^g) \\ & + \sum_i \lambda_i \sum_{p \in R_i^c} \log(H_i(I_i^p)). \end{aligned} \tag{9}$$

En simplifiant cette équation nous obtenons

$$\begin{aligned} Q(\Phi, \Phi^g) \quad = & \sum_s \sum_{k \in \mathcal{K}} \log\Big(p(k_s = k, I_1^s, ..., I_n^s|\Phi)\Big) \\ & p(k_s = k|I_1^s, ..., I_n^s, \Phi^g) \\ & + \sum_i \lambda_i \sum_{p \in R_i^c} \log(H_i(I_i^p)). \end{aligned} \tag{10}$$

La mise à jour de l'ensemble de paramètres est donnée par

$$\Phi = \arg\max_{\Phi} Q(\Phi, \Phi^g) \tag{11}$$

5.1 Étape d'estimation

Dans l'étape d'estimation, la probabilité de chaque étiquette k_s est calculée pour chaque échantillon s :

$$\forall k \in K,$$

$$p(k_s = k|I_1^s, ..., I_n^s, \Phi^g) = \frac{\pi_k^g \left[\prod_i^n p(I_i^s|\Theta_i^{g,c}, k_s = k)\right]}{\sum_z \pi_z^g \left[\prod_i^n p(I_i^s|\Theta_i^{g,c}, k_s = z)\right]}. \tag{12}$$

Cette quantité sera notée p_s^k.

5.2 Étape de maximisation

Dans cette étape, nous cherchons le nouvel ensemble de paramètres Φ qui maximisent la fonction Q. Cette fonction peut s'écrire comme la somme de termes indépendants :

$$\begin{aligned} Q(\Phi, \Phi^g) = \quad & \sum_{s,k} p_s^k \log \pi_k \\ & + \sum_i \Bigg[\sum_s p_s^{\text{b}_i} \log(p(I_i^s|\Theta_i^c, k_s = \text{b}_i)) \\ & \qquad + \lambda_i \sum_{p \in R_i^c} \log(H_i(I_i^p)) \Bigg] \\ & + \sum_i \left[\sum_s p_s^{\text{f}} \log(p(I_i^s|\Theta_i^c, k_s = \text{f})) \right] \end{aligned} \tag{13}$$

Chaque terme peut être maximisé de manière indépendante. Pour π_k

$$\pi_k = \frac{1}{N} \sum_s p_s^k \qquad (N \text{ nombre d'échantillons}). \tag{14}$$

Soit b une case d'histogramme et soit $H_{i,b}$ le nombre d'occurrences dans b pour l'histogramme H_i. Alors, on peut

montrer que maximiser $\Phi = \arg\max_{\Phi} Q(\Phi, \Phi^g)$ revient à mettre à jour les valeurs des cases des histogrammes du fond comme suit :

$$H_{i,b} = \sum_{s \in S, I_i^s \in b} p_s^{\mathrm{b}i} + \lambda_i H_{i,b}^{\mathrm{Ext}} \tag{15}$$

et

$$H_b^F = \sum_i \sum_{s \in S, I_i^s \in b} p_s^{\mathrm{f}} \tag{16}$$

pour l'histogramme de l'avant plan. Les histogrammes sont ensuite normalisés.

6 Segmentation finale

L'algorithme EM décrit dans la section précédente va converger vers une estimation des modèles couleur pour chaque vue et une table de probabilité de classification pour chaque échantillon. Les échantillons ne produiraient qu'une segmentation éparse des pixels de l'image s'ils étaient directement projetés dans les vues. Nous exploitons conjointement les probabilités sur les étiquettes des échantillons et les modèles couleur, tous deux issus de la convergence de l'EM, afin de générer une segmentation finale 2D dense. Cela revient à estimer, pour chaque pixel p de la $i^{ème}$ vue, l'étiquetage l_i^p (objet/fond) en fonction des modèles couleur (Fig. 4).

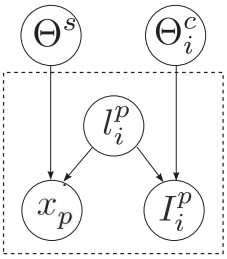


FIGURE 4 – Relation entre les différentes variables dans la segmentation finale.

Nous proposons d'utiliser une technique de "graph cut" similaire à [3] pour obtenir la segmentation finale dans chaque vue, en minimisant l'énergie discrète suivante :

$$E = \sum_p E_d(l_i^p | \Theta^s, \Theta_i^c, x_p, I_i^p) + \sum_{\{p,q\} \in N_i} \alpha E_s(I_i^p, I_i^q) \tag{17}$$

Notons que d'autres stratégies pourraient être utilisées ici. La terme lié aux données E_d au niveau du pixel p dépend des modèles couleur obtenus pour l'image i et de la distance entre sa position x_p et la projection dans l'image des échantillons 3D. Θ^s correspond à la position de l'échantillon et ses probabilités associées aux différentes étiquettes $\{p_s^k\}_{s,k}$:

$$E_d(l_i^p | \Theta^s, \Theta_i^c, x_p, I_i^p) = -\log(p(x_p | \Theta^s, l_i^p) p(I_i^p | \Theta_i^c, l_i^p)), \tag{18}$$

– $p(x_p | \Theta^s, l_i^p)$ est inversement proportionnel à la distance à la projection de l'échantillon avant plan le plus proche. Cela permet une projection plus harmonieuse de l'information inférée.
– $p(I_i^p | \Theta_i^c, l_i^p)$ est basé sur les histogrammes fond/objet obtenus à l'étape précédente :

$$p(I_i^p | \Theta_i^c, l_i^p) = \begin{cases} H_i(I_i^p) & \text{si} \quad l_i^p = \text{arrière-plan}, \\ H^F(I_i^p) & \text{si} \quad l_i^p = \text{avant-plan}. \end{cases} \tag{19}$$

E_s est un terme de lissage sur l'ensemble des pixels voisins (N_i), pour lequel n'importe quelle énergie favorisant un étiquetage cohérent dans les régions homogènes est utilisable. Dans notre implémentation nous utilisons l'inverse de la distance dans l'espace couleur entre pixels voisins.

7 Résultats expérimentaux

Dans cette section, nous nous proposons d'étudier la méthode proposées dans différents scénarios, en particulier en comparaison avec des méthodes monoculaires et multi-vues de l'état de l'art. Nous étudions aussi l'effet du nombre de vues et du nombre d'échantillons sur les résultats de segmentation.
Pour cela nous utilisons différents jeux de données calibrés. Les modèles couleurs utilisés sont des histogrammes 32^3 dans l'espace couleur HSV. Nous ne faisons aucune hypothèse particulière lors de l'initialisation. Toutes les étiquettes ont la même probabilité pour tous les échantillons, et le processus itératif commence par l'étape de maximisation. L'algorithme est exécuté sur un PC Intel Xeon 2.5 Ghz avec 12GB de RAM avec une implémentation C++ non optimisée. Le temps de calcul est typiquement de quelques secondes par itération de l'EM. L'algorithme converge en moins de 10 itérations pour tous les tests effectués.
L'échantillonnage est effectué à l'intérieur du volume de visibilité commun. Ce volume permet d'initialiser les régions R_i dans les vues et de trouver un premier ensemble de pixels d'arrière-plan. Dans la plupart des scénarios, moins de 50^3 échantillons ont été nécessaires pour converger rapidement vers une bonne estimation des silhouettes.
Les jeux de données utilisés pour ces tests sont *Couch*, *Bear*, *Car*, *Bike* and *Chair* de [11], *Buste* [1] utilisé dans [12] et certains jeux de données apparaissant dans [9] (*Pig* et *Rabbit*). Le dataset *Arts Martiaux* [2] est utilisé pour montrer la capacité de notre méthode à gérer plusieurs objets d'intérêt. Pour chaque jeu de données, le nombre de vues est indiqué dans la légende de la figure associée.

7.1 Résultats qualitatifs

Pour tous les jeux de données, nous montrons le résultat de l'algorithme sur les étiquettes des échantillons, ainsi que la segmentation finale obtenue. Nous pouvons observer sur la figure 6 les résultats intermédiaires de l'algorithme. A chaque itération, les échantillons étiquetés "objet" avec une

1. http ://www.cs.ust.hk/~quan/WebPami/pami.html
2. http ://4drepository.inrialpes.fr/public/viewgroup/4#sequence9 Séquence *Kick one*.

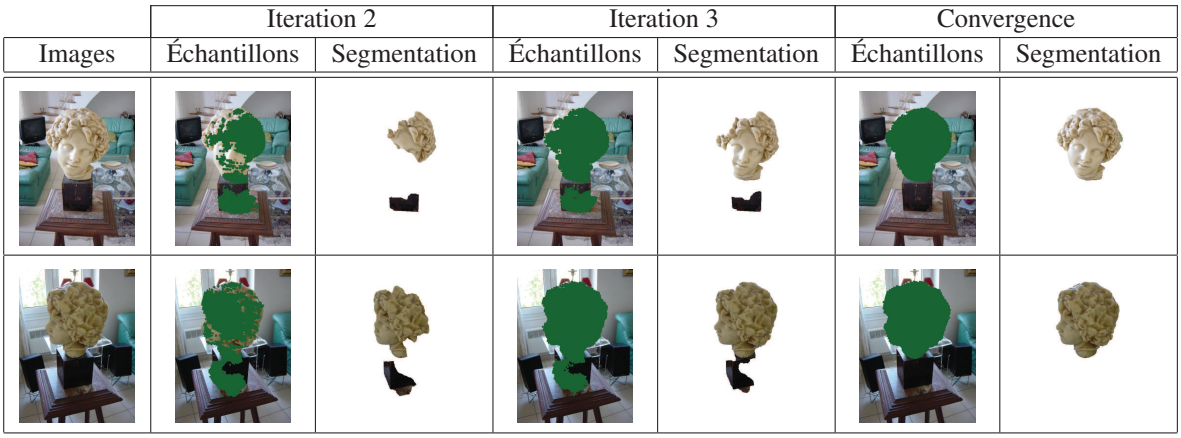


FIGURE 6 – Résultats intermédiaires sur *Buste* (13 vues). Les points verts représentent les projections des échantillons 3D avec $p_s^f > 0.8$. La segmentation à chaque itération est effectuée en utilisant la méthode décrite dans le §6. Ces résultats intermédiaires sont utilisés pour étudier la convergence de l'algorithme (Fig. 9).

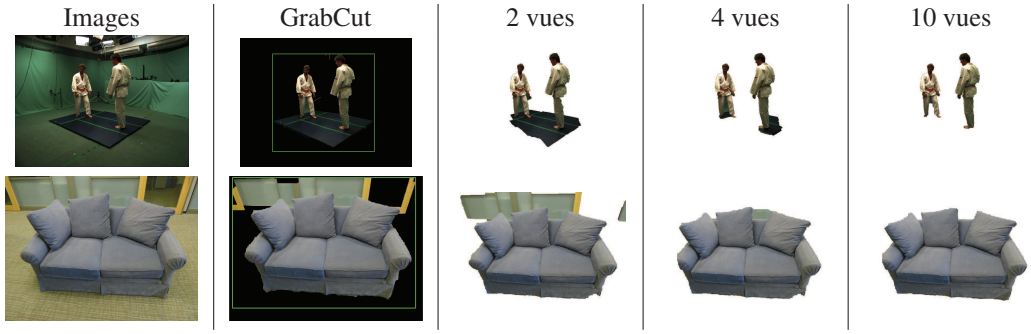


FIGURE 7 – Comparaison avec une approche monoculaire (GrabCut) et influence du nombre de vues sur le résultat de segmentation sur les séquences *Art martiaux* et *Couch*.

forte probabilité ($p_s^f > 0.8$) sont représentés. Les segmentations intermédiaires sont obtenues grâce à la méthode décrite dans le §6. Ces silhouettes intermédiaires sont utilisées pour étudier la convergence de l'algorithme.

Comparaison avec une méthode monoculaire. Pour montrer l'avantage d'utiliser une approche multi-vues, nous comparons la méthode proposée avec l'implémentation de GrabCut dans *OpenCV*. Les résultats (Fig. 7) montrent que pour une approche monoculaire, il est difficile d'éliminer les couleurs d'arrière-plan qui ne font pas partie de l'arrière plan-connu. A l'inverse, notre algorithme bénéficie de l'information des autres vues et produit une segmentation correcte. La Fig. 7 founit également une évaluation qualitative de l'influence du nombre de vues utilisées sur la segmentation.

Comparaison avec les méthodes multi-vues. Les résultats obtenus sur les jeux de données *Buste*, *Bear* et *Couch* (Figs. 6, 5(a) and 5(b)) sont très convaincants. La méthode peut gérer des situations avec plusieurs objets d'intérêt comme pour la séquence *Arts Martiaux*. Sur les séquences *Car* et *Bike* (Fig. 5(c) et Fig. 5(d)), les résultats sont de qualité comparable à ceux de [11] avec significativement moins de vues. Toutefois, dans certaines vues, les résultats de segmentation sont pénalisés par des ambiguïtés couleur entre l'avant-plan et l'arrière-plan.

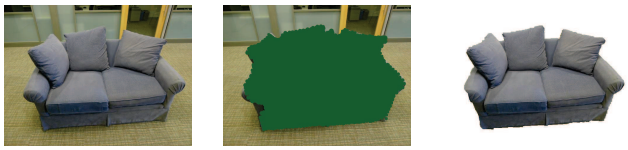
Nous comparons aussi notre approche avec [9], dont l'objectif principal est la reconstruction 3D de l'objet. Dans [9], le traitement est limité à une boîte englobante autour de l'objet, et l'estimation des modèles couleur d'avant-plan et d'arrière-plan fait appel à une interaction utilisateur. Notre approche peut aussi intégrer ce type d'information. Pour *Pig* et *Rabbit* (Fig. 8) l'utilisateur indique seulement les régions ambiguës. Il suffit de deux itérations pour converger vers une segmentation correcte.

7.2 Résultats quantitatifs

Dans le cadre de l'évaluation quantitative, 3 métriques principales sont utilisées : *Mean Error* (l'erreur moyenne), *Hit Rate* (taux de succès) et *False Alarms* (taux de fausses alarmes). Si on note W_b^a l'ensemble des pixels appartenant à l'ensemble a dans la vérité terrain et étiquetés b dans la segmentation finale, $N(.)$ la fonction qui compte le nombre de pixels dans chaque ensemble, alors

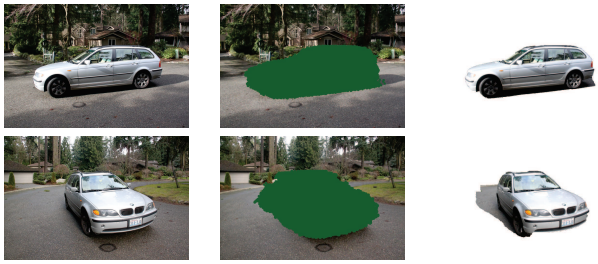
$$\textit{Mean Error} = \frac{N(W_F^B) + N(W_B^F)}{\textit{Nombre de pixels}} \tag{20}$$

$$\textit{Hit Rate} = \frac{N(W_F^F)}{N(W_F^F) + N(W_B^F)} \tag{21}$$



(a) *Couch* :10 vues - Runtime $\approx 15s$

(b) *Bear* :10 vues - Runtime $\approx 15s$



(c) *Car* :14 vues - Runtime $\approx 30s$

(d) *Bike* :11 vues - Runtime $\approx 20s$

FIGURE 5 – Résultats sur *Car* et *Bike* [11]. Les points verts indiquent les échantillons avec $p_s^f > 0.8$. La dernière colonne représente le résultat de la segmentation.

$$\textit{False Alarms} = \frac{N(W_F^B)}{N(W_F^B) + N(W_F^F)}. \qquad (22)$$

On définit aussi $\textit{Accuracy} = 1 - \textit{Mean Error}$ et $\textit{Missed Rate} = 1 - \textit{Hit Rate}$.

Pour ces mesures d'erreur, on calcule calcule la valeur moyenne sur l'ensemble des vues. La fig. 9 montre une évaluation de la vitesse de convergence : seulement 6 itérations sont nécessaires, voire même 2 itérations dans le cas de jeux de données plus simples comme *Couch* et *Bear*. Les résultats de segmentation sont obtenus après quelques secondes (voir fig. 5 pour des exemples de temps de calcul). A titre de comparaison, [11] indique des temps d'exécution de 2 minutes par image et [12] plusieurs minutes. Parmi les autres méthodes comparables, [9] utilise une implémentation GPU optimisée et obtient des temps de calcul de l'ordre de 5 secondes. On remarquera que notre méthode pourrait également tirer profit d'une implémentation sur GPU (parallélisation des étapes E et M) pour atteindre des temps de calcul comparables voire inférieurs.

La figure 10 montre une étude comparative avec [11] et

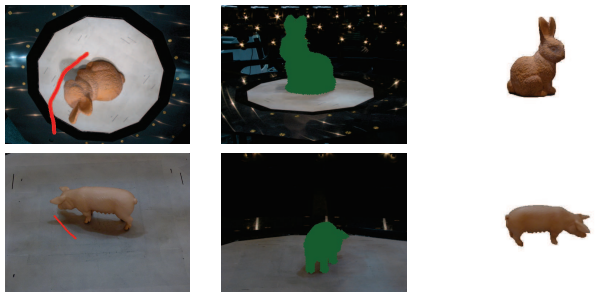


FIGURE 8 – Résultat sur les jeux de données *Rabbit* et *Pig* : L'utilisateur indique les régions d'arrière-plan dans une des vues.

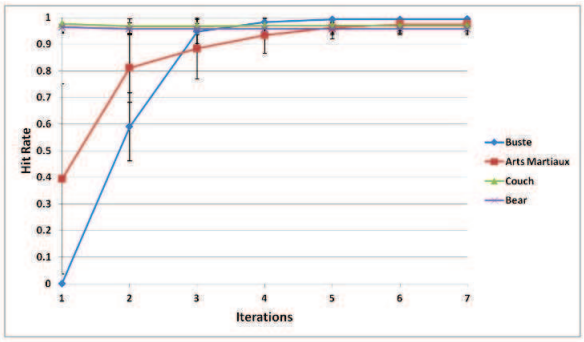


FIGURE 9 – Pour les 4 jeux de données, l'algorithme converge après 6 itérations.

[19] sur la qualité des segmentations obtenues. On peut voir que notre approche produit des résultats de meilleure qualité que ceux obtenus par [19], et de qualité comparable à [11] bien que seul un tiers des vues ait été utilisé.

Dataset	Notre méthode	Kowdle[11]	Vicente [19]
Couch	10 98.8 ± 0.8	10 99.6 ± 0.1	non disponible
Bear	10 98.8 ± 0.4	15 98.8 ± 0.4	non disponible
Car	14 96.2 ± 1.1	44 98.0 ± 0.7	44 91.4 ± 4.3
Bike	11 96.9 ± 1.4	35 99.4 ± 0.4	35 88.9 ± 6.3

FIGURE 10 – Résultats des tests comparatifs. La proportion de pixels correctement segmentés est utilisée comme mesure d'erreur (*Accuracy* en %). Pour chaque séquence, le nombre de vues utilisées est indiqué.

Influence du nombre d'échantillons. Pour étudier l'influence du nombre d'échantillons 3D sur les résultats de segmentation, nous utilisons un échantillonnage aléatoire épars à l'intérieur du volume de visibilité commun. Les résultats de ce test sont présentés dans le tableau de la figure 11. On peut voir qu'un maximum de 100 000 échantillons 3D sont suffisants pour atteindre des résultats de segmentation corrects, alors que [9] avec le double objectif de la

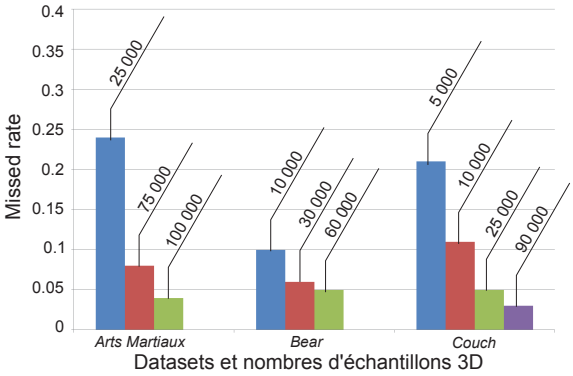


FIGURE 11 – Résultats avec différents nombre d'échantillons 3D. Le taux de pixels étiqueté "fond" à tort (*Missed rate*) est utilisé comme mesure d'erreur.

reconstruction 3D et de la segmentation utilise des grilles de 200^3 voxels.

8 Conclusion

Cet article présente une nouvelle approche, probabiliste, du problème de segmentation multi-vues. Des modèles couleur fond et objet sont simultanément estimés, et la cohérence inter-vues est exprimée au moyen de n-uplets représentant les couleurs des projections de points 3D, échantillonnés dans le volume de visibilité commun, dans chaque vue. Les résultats expérimentaux suggèrent que cette nouvelle approche fournit des résultats de performances comparables à ceux de l'état de l'art, avec une complexité calculatoire réduite.
Ce travail permet également d'avoir une meilleure compréhension des hypothèses pouvant être utilisées pour définir l'avant-plan et de leur influence sur les résultats, et suggère d'autres voies à explorer, en particulier une meilleure modélisation de l'apparence fond/objet qui permettrait de lever certaines ambiguïtés.

Références

[1] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg : Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, pages 3169–3176, 2010.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[3] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, 2001.

[4] N. D. F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Automatic 3d object segmentation in multiple views using volumetric graph-cuts. *Image Vision Comput.*, 2010.

[5] T. Feldmann, L. Dießelberg, and A. Wörner. Adaptive foreground/background segmentation using multiview silhouette fusion. In J. Denzler, G. Notni, and H. Süße, editors, *DAGM-Symposium*, volume 5748 of *Lecture Notes in Computer Science*. Springer, 2009.

[6] J.-S. Franco and E. Boyer. Fusion of Multi-View Silhouette Cues Using a Space Occupancy Grid. In *ICCV* , 2005.

[7] J. Gallego, J. Salvador, J. Casas, and M. Pardàs. Joint multi-view foreground segmentation and 3d reconstruction with tolerance loop. In *IEEE ICIP*, 2011.

[8] A. Joulin, F. R. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, pages 1943–1950, 2010.

[9] K. Kolev, T. Brox, and D. Cremers. Fast joint estimation of silhouettes and dense 3d geometry from multiple images. *IEEE PAMI*, 2011.

[10] A. Kowdle, Y.-J. Chang, D. Batra, and T. Chen. Scribble based interactive 3d reconstruction via scene co-segmentation. In *IEEE ICIP*, pages 2577–2580, 2011.

[11] A. Kowdle, S. N. Sinha, and R. Szeliski. Multiple view object cosegmentation using appearance and stereo cues. In *ECCV*, volume 7576 of *LNCS*, pages 789–803, 2012.

[12] W. Lee, W. Woo, and E. Boyer. Silhouette Segmentation in Multiple Views. *IEEE PAMI*, page 14 p., Oct. 2010.

[13] C. Rother, V. Kolmogorov, and A. Blake. "grabcut" : interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH*, 2004.

[14] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *CVPR*, pages 993–1000, 2006.

[15] D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. In *CVPR*, 2000.

[16] M. Sormann, C. Zach, and K. Karner. Graph cut based multiple view segmentation for 3d reconstruction. In *3DPVT*, pages 1085 –1092, june 2006.

[17] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[18] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower : principles and practice of background maintenance. *ICCV*, 1999.

[19] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, 2011.

[20] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder : Real-time tracking of the human body. *IEEE PAMI*, 1997.

[21] G. Zeng and L. Quan. Silhouette extraction from multiple images of an unknown background. In *ACCV*, 2004.