

Segmentation collective d'images à base d'objets polyédriques et courbés

Smaine Mazouzi*, Zahia Guessoum†, et Fabien Michel‡

* Dép. d'informatique, Université de Skikda, BP 26, 21000, Algérie.
mazouzi_smaine@yahoo.fr

† LIP6, Université Paris 6, 104, av. Président Kennedy, 75016 Paris, France.
zahia.guessoum@lip6.fr

‡ MODECO-CReSTIC, IUT de Reims, Rue des Crayers, 51100, France.
fmichel@univ-reims.fr

Résumé

Dans cet article, nous utilisons le paradigme multi-agents pour proposer une nouvelle méthode de segmentation d'images. Les images considérées dans ce travail sont les images de profondeur qui peuvent contenir à la fois des objets polyédriques et des objets courbés. La méthode proposée utilise une approche multi-agents pour l'alignement des bordures de régions afin de faire émerger une segmentation collective de l'image. Les agents se déplacent sur l'image et alignent les pixels de contours à leurs surfaces respectives. L'alignement compétitif qui en résulte permet à la fois l'émergence des contours et l'effacement des régions de bruit. Lors de leurs mouvements les agents s'auto-organisent dans l'espace de l'image via un champ de potentiel. Les résultats de test obtenus avec des images réelles montrent la pertinence de la méthode proposée pour une segmentation précise des images de profondeur.

Mots-clés : Segmentation d'images, système multi-agents, auto-organisation, champ de potentiel, image de profondeur.

Abstract

In this paper, we use the multi-agent paradigm in order to propose a new method for image segmentation. Considered images in this work are range images which can contain both polyhedral and curved objects. The proposed method uses a multi-agent approach for region boundary alignment to make a collective image segmentation emerge. Agents move within the image, and align the edge pixels to their respective surfaces. The resulted competitive alignment enables the emergence of edges as well as the erasing of noise regions. During their moves agents self-organize in the image space thanks to a potential field. The test results obtained with real images show the relevance of the proposed method for a precise segmentation of range images.

Key-words: Image Segmentation, Multi-Agent Systems, Self-organization, Potential field, Range Image.

1 INTRODUCTION

Une image de profondeur représente la surface visible d'une scène tridimensionnelle, où à chaque élément de l'image est associée la profondeur du point correspondant de la scène. Les images de profondeur sont utilisées principalement en reconnaissance d'objets 3D en vision robotique [20], car l'information 3D nécessaire à la reconnaissance des objets est immédiatement disponible. Cependant, ces images sont reconnues comme étant des images hautement bruitées [9] et sont par conséquent difficiles à segmenter et à interpréter. Plusieurs auteurs ont proposé des méthodes de segmentation pour cette classe d'images [5, 13, 11, 4, 1, 17]. Cependant, la plupart de ces auteurs se sont limités aux images à base d'objets polyédriques. Pour ces dernières, la détection des régions d'intérêt est largement plus facile que pour les images contenant des objets courbés [11]. La visualisation d'une image de profondeur par une procédure de rendu réaliste (Fig. 1) montre un degré important de déformation des surfaces, d'où la difficulté de situer les contours d'objets. Un lissage naïf sur ce genre d'images conduit souvent à des résultats de segmentation erronés. En effet, si le lissage est de faible impact, les déformations qui demeurent dans l'image la rendent inexploitable. Cependant, si le lissage est sévèrement opéré, il en résulte de larges contours lissés et élargis, formant eux-mêmes de nouvelles surfaces courbées et engendrant par la suite une fausse segmentation.

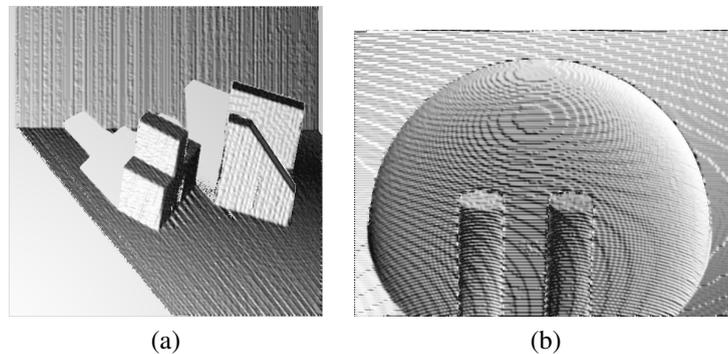


FIGURE 1 – Niveau de bruit dans une image de profondeur. (a) Scène à objets polyédriques; (b) Scène à objets courbés

Par ailleurs, la restriction du calcul au voisinage immédiat d'un pixel, imposée à la quasi-totalité des méthodes de segmentation, engendre des résultats imprécis. Cette contrainte de localité provient du fait que les données dans une image sont peu ou non modélisées. Même avec des modèles mathématiques raffinés tels que la distribution de Gibbs [7], les auteurs étaient contraints de supposer la markovianité des données de l'image. Ceci est en effet nécessaire pour pouvoir envisager des procédures de traitement qui

peuvent être exécutées en un temps raisonnable [15].

Dans le but de doter un système de segmentation d'images d'une faculté de perception plus large, nous avons recours au paradigme multi-agents et plus particulièrement aux agents réactifs [6]. Nous proposons dans cet article une nouvelle approche de segmentation d'images, basée sur l'utilisation d'une population dense d'agents réactifs. Il s'agit d'agents simples de granularité faible et dont le comportement est basé sur le mécanisme de stimulus-réaction. Les agents se déplacent d'un pixel à la fois sur l'image et agissent sur les pixels rencontrés. Un agent aligne le premier pixel non homogène à la surface sur laquelle il se déplace. Ainsi, les pixels appartenant aux régions de bruit ou situés sur les pourtours des surfaces sont systématiquement alignés aux surfaces qui les entourent. Nous montrons que l'alignement alternatif des pixels sur les frontières des régions permet de préserver les lignes de contour. Par contre, l'absence d'alternance d'alignement de pixels autour des régions de bruit conduit à l'effacement progressif de ces régions. Cette action collective des agents permet l'émergence des contours et l'effacement des régions de bruit.

Le mouvement de l'agent sur une surface homogène lui confère une perception plus large. Lorsqu'il agit sur un pixel, l'agent considère à la fois les données locales ainsi que des données globales construites le long de son parcours sur la même surface. Ceci nous a permis de surmonter l'ambiguïté inhérente à la restriction de la décision au seul voisinage du pixel traité. Afin de doter les agents d'un mécanisme d'auto-organisation spatiale, une méthode de coordination utilisant un champ de potentiel est introduite. Un champ de potentiel attractif se crée autour des pixels alignés et permet par conséquent de faire diriger les agents vers les régions d'intérêt qui se composent de pixels non homogènes (de contours ou de bruit). Ainsi, les actions des agents sont concentrées autour de ces régions.

La majorité des méthodes de segmentation utilisant le paradigme agent [2, 16, 21, 22, 19, 12] sont supervisées et par conséquent elles ne peuvent être appliquées qu'aux images pour lesquelles elles ont été conçues. La méthode proposée dans cet article est non supervisée. Elle ne fait aucune supposition sur le nombre et la forme des surfaces qui composent les objets dans l'image. Ceci permet l'adaptation de la méthode à tout type d'images et ce en définissant les critères d'homogénéité des régions qui composent les images traitées. La distribution des traitements et des décisions caractérisant notre approche, ainsi que le faible couplage des agents réactifs opérant selon le principe de la stygmergie [25], offrent une méthode parallèle bien appropriée aux applications temps réel. L'expérimentation de la méthode, en utilisant des images réelles, a permis de valider la nouvelle approche et de montrer son potentiel pour une segmentation exacte et efficace des images de profondeur.

La suite de l'article est organisée comme suit : dans la section 2, nous présentons les principales approches ayant utilisé les systèmes multi-agents pour la segmentation d'images. La section 3 est réservée à l'approche pro-

posée. Nous présentons d'abord le critère d'homogénéité de régions adopté dans notre cas. Ensuite, nous décrivons le fonctionnement du système multi-agents et nous montrons comment une segmentation collective émerge du comportement simple et réactif des agents. En section 4, nous détaillons la méthode de coordination spatiale des agents, en montrant comment les agents s'auto-organisent autour des pixels d'intérêt. Nos résultats de test sont présentés dans la section 5, dans laquelle nous discutons l'apprentissage des paramètres et nous commentons les résultats obtenus. Finalement, une conclusion résume notre travail et souligne ses potentielles extensions.

2 LES APPROCHES MULTI-AGENTS POUR LA SEGMENTATION D'IMAGES

Dès la publication des premiers travaux proposant des approches multi-agents pour le traitement d'images, la coopération et l'interaction entre les agents représentaient les nouveaux apports pour remédier au problème d'ambiguïté des données visuelles caractérisant les images [2, 16]. Par la suite, plusieurs systèmes multi-agents ont été proposés dans le domaine du traitement d'images et de la reconnaissance d'objets. Ces systèmes sont introduits pour résoudre des problèmes communs tels que le contrôle des opérations de traitement, la représentation et la manipulation des informations hétérogènes et la coordination et la parallélisation des traitements. Par contre, chaque système procède selon une approche différente qui lui y est exclusivement propre, comme on peut le constater pour la plupart des systèmes qui sont décrits dans la suite de la section.

Dans la catégorie des méthodes de segmentation basées contour et suivant une approche réactive, Ballet et al. [2] ont défini un système multi-agents pour la détection et le suivi de contours dans les images 2D. Un agent du système se situe sur un extremum du gradient de la luminance puis suit la crête de la ligne de contour et enregistre son chemin dans une mémoire partagée. Les segments de contour précédemment enregistrés sont utilisés par d'autres agents afin d'initialiser des lignes de contour ou d'achever d'autres. La méthode proposée par les auteurs repose sur une approche originale de détection parallèle et distribuée de contours. Cependant, les auteurs n'ont envisagé aucun mécanisme de coopération ou de coordination, pour renforcer la détection et améliorer les résultats de segmentation.

Les mêmes auteurs [22] ont repris ultérieurement la même approche et l'ont étendu en utilisant le langage multi-agents oRis [8]. En considérant la connaissance a priori sur la topologie des régions dans certains types d'images, les agents suivent les lignes d'extrema du gradient. Considérant des images à deux types de régions, deux catégories d'agents dits respectivement agents de noircissement et agents de luminance sont définis. Selon leur catégorie, les agents suivent respectivement les régions sombres ou les régions claires. En se maintenant sur les lignes de crête, les agents renforcent

la différence de contraste entre les paires de régions adjacentes. Le système proposé est bien dédié aux images contenant des contours de type toit (détectés par la discontinuité de la première dérivée de l'image). Ces contours caractérisent les images considérées par les auteurs.

Les bancs de poissons ont inspiré l'équipe de Mobahi pour concevoir un système multi-agents dédié à l'initialisation des contours actifs [19]. Chaque agent est soumis à trois forces distinctes : une force attractive exercée par les points de contour, une force répulsive produite par les agents eux-mêmes pour rester distants les uns des autres et une force d'intérêt incitant les agents à explorer l'image à la recherche des points de contour. Une approche de formation de patterns émergents est proposée par Jones et Saeed [12] en s'inspirant également des comportements de groupes sociaux. L'interaction des agents réactifs permet la perception des niveaux globaux de luminance à partir du contraste local qui représente le stimulus des agents. En fonction de son stimulus, un agent change d'orientation en modifiant le chemin suivi par l'ajout ou le retrait d'une unité de trace et ce selon que l'agent est un agent attiré par la lumière ou attiré par le noir. Les structures de trace émergentes sont interprétées comme une perception globale du degré de la luminance dans l'image.

Les approches multi-agents précédentes, à l'exception de celle de Rodin et al. [22] ont procédé à une détection sans aucune représentation (implicite ou explicite) des régions dans les images. Ceci ne facilite pas la segmentation basée région des images traitées.

Dans la catégorie des méthodes basées région, Liu et al. [16] ont utilisé un système à base d'agents réactifs pour la segmentation d'images à résonance magnétique (IRM) du cerveau. Les auteurs ont utilisé quatre types d'agents, qui correspondent aux quatre tissus qui forment la matière cérébrale pathologique. Une première génération d'agents est initialisée à l'intérieur des différents tissus. Ensuite, lorsque chaque agent arrive à reconnaître sa région, il crée des agents clones et les place de telle sorte que leur chance de trouver d'autres pixels homogènes augmente. Pour le même type d'images, Richard et ses collaborateurs [21] ont proposé un système multi-agents où les agents sont disposés hiérarchiquement comme suit : 1) Un agent de contrôle global en une seule instance pour la partition du volume de l'image en volumes partiels ; 2) Des agents de contrôle local, affectés chacun à un volume partiel ; 3) Des agents dédiés aux tissus, qui se lancent sous le contrôle d'un agent de contrôle local pour segmenter le tissu à l'intérieur d'un volume partiel. Ainsi, la détection est réalisée au niveau le plus bas par les agents dédiés aux tissus, puis synthétisée par les agents de contrôle local et l'agent du contrôle global. A la fin, une segmentation de l'image en ses différentes régions homogènes est disponible au niveau le plus haut de la hiérarchie.

Inspirés du comportement des araignées sociales, Bourjot et ses coauteurs [3] ont proposé un modèle multi-agents réactif reproduisant le tissage collectif de toiles. Le comportement stochastique de chaque agent consiste soit au mouvement entre les piquets implantés dans l'environnement, soit à

la fixation d'une ligne de soie sur le piquet courant. Un facteur d'attraction à la soie est utilisé pour réguler le comportement des agents, permettant à la fois l'exploration de l'environnement et la construction de la toile. Ensuite, le modèle proposé est transposé afin de produire une détection de régions dans les images à niveau de gris. La transposition est faite de telle sorte que chaque toile correspond à une région homogène. Pour assurer que les régions détectées soient connexes, un agent se limite dans ses déplacements aux pixels qui sont au voisinage du dernier pixel sélectionné.

A l'instar de la plupart des approches basées agent pour la segmentation d'images en régions, les trois travaux précédents suivent des approches supervisées, où il est nécessaire de disposer soit du nombre de régions soit de leurs formes. Nous introduisons dans cet article une nouvelle approche qui se veut générale et non supervisée pour la segmentation d'images. Nous montrons que l'alignement compétitif des pixels non homogènes, opéré par des agents réactifs, permet de faire émerger une segmentation collective de l'image. L'approche proposée est non supervisée. Elle ne fait aucune supposition préalable sur le nombre ni la nature des régions qui forment les images à segmenter.

3 SEGMENTATION D'IMAGES PAR ÉMERGENCE DE CONTOURS

Nous nous consacrons dans ce travail aux images de profondeur. Néanmoins, les concepts introduits et le système qui en découle peuvent être adaptés à d'autres types d'images. Une image de profondeur est représentée par une structure bidimensionnelle I , où au niveau de chaque élément (x, y) est enregistrée la profondeur du point correspondant $I(x, y)$. Cette profondeur représente la distance entre le point (x, y) et le plan du capteur télémétrique. Souvent, les surfaces des objets sont représentées par leurs équations de plan en ne considérant que les images à objets polyédriques. Dans ce travail, nous proposons une approche de segmentation étendue aux objets courbés. Par conséquent, il est nécessaire de définir une mesure sur les données brutes de l'image permettant de mettre en place un critère d'homogénéité de régions caractérisant les surfaces courbées.

3.1 Homogénéité de régions dans les images de profondeur

Afin de ne faire aucune hypothèse sur les formes de surfaces, nous avons opté pour l'utilisation des courbures directionnelles de surface comme critère d'homogénéité. Soit un pixel (x, y) , les deux courbures directionnelles, respectivement horizontale C_h et verticale C_v , sont calculées comme suit :

$$C_h(x, Y) = -\frac{I''(x, \cdot)}{1 + I'^2(x, \cdot)} \quad (1)$$

$$C_v(X, y) = -\frac{I''(\cdot, y)}{1 + I'^2(\cdot, y)} \quad (2)$$

où $I'(x, \cdot)$ et $I''(x, \cdot)$ sont les dérivées première et seconde de l'image selon x , en fixant la valeur de y à Y . De même, $I'(\cdot, y)$ et $I''(\cdot, y)$ sont les dérivées première et seconde de l'image selon y , en fixant la valeur de x à X .

C_h exprime la courbure de la projection de la surface sur le plan $y = Y$, tandis que C_v exprime la courbure de la projection de la surface sur le plan $x = X$.

Un ensemble R contigu de pixels $\{(x_k, y_k), k \in R\}$ est considéré homogène et représente une région de l'image, si tous ses pixels ont, à un seuil d'égalité Tr_c près, les mêmes courbures directionnelles :

$$\forall k, l \in R; |C_h(x_k, y_k) - C_h(x_l, y_l)| < Tr_c$$

et

$$|C_v(x_k, y_k) - C_v(x_l, y_l)| < Tr_c$$

Tr_c est le seuil de comparaison des courbures dont la valeur est fixée automatiquement lors de l'apprentissage des paramètres (cf. section 5.2).

Ainsi, les pixels d'intérêt qui sont situés aux frontières des vraies régions ou formant les régions de bruit, sont détectés lors du changement de l'une des deux courbures C_h ou C_v .

3.2 Dynamique des agents

De nombreux agents ($N=2500$ cf. section 5.2) de nature réactive sont placés à des positions arbitraires dans l'image de profondeur I . Après leur initialisation, les agents commencent à se déplacer dans l'image en suivant des directions aléatoires. Ils peuvent également être dirigés par un champ de potentiel attractif (cf. section 4). Un agent recherche une région homogène autour de sa position courante (x_c, y_c) en examinant les L derniers pixels visités. La longueur du chemin en pixels L (déterminée en section 5.2) permet à l'agent d'être certain qu'il est à l'intérieur d'une région homogène. Si ces derniers L pixels ont tous les mêmes courbures directionnelles, l'agent considère qu'il est à l'intérieur d'une région homogène (*regionCourante*). Il acquiert alors une capacité d'altération de l'image (*capaciteAlteration* ← VRAI). Cette capacité d'altération permet à l'agent de modifier les profondeurs des pixels rencontrés. Il procède dans ses futurs déplacements au lissage des pixels rencontrés si ces pixels appartiennent à la surface sur laquelle il se déplace. En arrivant sur le premier pixel non homogène, l'agent aligne partiellement ce

pixel à la surface courante. Il perd alors sa capacité d'altération et recommence à rechercher une nouvelle région homogène. L'algorithme suivant représente la méthode "*step()*" exécutée par un agent à chaque pas de son parcours.

Initialisation :

```

capaciteAlteration ← Faux
(xc, yc) ← (random(largeurImg - 1), random(hauteurImg - 1))
l ← 0 // initialisation de la longueur du chemin parcouru

```

Algorithme 1 Méthode *step()*

```

(xc, yc) ← seDeplacerSurUnPixelVoisin()
// (cf. section 4)
si NON capaciteAlteration alors
  // (cf. section 3.2.1)
  si voisinsHomogenes(xc, yc) alors
    l ← l + 1
    si l ≥ L alors
      capaciteAlteration ← VRAI
      regionCourante ← region(xc, yc)
    finsi
  finsi
sinon
  // (cf. section 3.2.2)
  si appartient(xc, yc, regionCourante) alors
    lisserPixel(xc, yc)
  sinon
    alignerPixel(xc, yc, regionCourante)
    mettreAJourChampDePotentiel(xc, yc)
    // (cf. section 4)
    capaciteAlteration ← FAUX
    l ← 0
  finsi
finsi

```

Toutes les variables utilisées dans l'algorithme 1 sont locales à l'agent et elles déterminent à tout moment son état interne. La fonction "*seDeplacerSurUnPixelVoisin()*" permet à l'agent de se déplacer (d'un seul pas) sur un des pixels voisins à celui sur lequel il est situé. Le vecteur de déplacement est aléatoire, mais biaisé par le vecteur de force du champ de potentiel si l'agent est en capacité d'altération (cf. section 4). La fonction "*voisinsHomogenes()*" teste si le pixel passé comme paramètre est entouré de pixels homogènes (appartiennent tous à la même surface). Le constructeur "*region()*" permet de construire une région en affectant à ses courbures directionnelles celles

du pixel passé comme paramètre. La fonction "*appartient()*" teste si le pixel passé comme premier paramètre appartient à la région passée comme second paramètre. La fonction "*lisserPixel()*" permet le lissage du pixel passé comme paramètre en considérant ses pixels voisins (cf. section 3.2.2). La fonction "*alignerPixel()*" aligne partiellement le pixel passé comme premier paramètre à la région passée comme second paramètre. La fonction "*miseAJourChampDePotentiel()*" met à jour le champ de potentiel autour du pixel passé comme paramètre (cf. section 4).

Les agents exécutent indéfiniment et d'une façon asynchrone la méthode "*step()*". La fin du processus est détectée suite à un état externe aux agents, atteint lorsque tous les contours sont détectés et lorsque toutes les régions de bruit sont effacées. Nous montrons dans la section 4 que l'amortissement total du champ de potentiel dans toute l'image est un indicateur de la fin du processus.

3.2.1 Initialisation d'une région homogène

Un agent reconnaît qu'il est en train de se déplacer sur une région homogène s'il arrive à parcourir L pixels après sa réinitialisation. Ces pixels doivent avoir tous les mêmes courbures directionnelles qu'on note C_h^0 et C_v^0 . L'agent mémorise ces deux courbures et les considère comme les deux courbures directionnelles caractérisant la surface sur laquelle il se déplace (*regionCourante*). A ce stade, l'agent acquiert une capacité d'altération des pixels de l'image. Cette capacité lui permet de lisser les pixels homogènes et d'aligner le premier pixel non homogène rencontré. Ce dernier pixel est soit un pixel du pourtour de la région en cours soit un pixel de bruit.

3.2.2 Lissage et alignement des pixels de l'image

Tant que l'agent est à l'intérieur d'une région homogène il lisse tout pixel qu'il visite. Pour cela, il utilise un filtre moyen opérant dans une fenêtre de taille 3×3 pixels. Le filtre moyen ne considère que les pixels dans la fenêtre qui ont les mêmes courbures directionnelles que le pixel en question (au centre de la fenêtre). Ainsi, la moyenne calculée ne sera pas biaisée par des pixels appartenant à une autre surface ou à une région de bruit. La nouvelle profondeur au pixel (x, y) peut être obtenue selon l'expression suivante :

$$I(x, y) = \frac{1}{V} \sum_{i=-1}^1 \sum_{j=-1}^1 I(x+i, y+j) \quad (3)$$

en ne considérant que les V pixels de l'ensemble $\{(x+i, y+j), i, j = -1 \dots 1\}$ vérifiant la condition suivante :

$$|C_h(x+i, y+j) - C_h(x, y)| < Tr_c \wedge |C_v(x+i, y+j) - C_v(x, y)| < Tr_c$$

Quand un agent arrive sur un pixel de la frontière de la région dans laquelle il se déplace, il le détecte comme étant un pixel non homogène. En effet, pour une région donnée les courbures directionnelles d'un pixel de la frontière sont différentes des courbures directionnelles de cette région. Le pixel non homogène est alors partiellement aligné à la surface de laquelle provient l'agent. Soit (x, y) le pixel non homogène, sa nouvelle profondeur $I(x, y)$ est remplacée par la moyenne des profondeurs de ses pixels voisins ayant au seuil Tr_c près, les mêmes courbures directionnelles. En fait, les agents procèdent par alignement partiel des pixels non homogènes car nous avons constaté qu'avec un alignement total, les contours engendrés sont déformés et mal localisés. Ainsi, ce n'est qu'après plusieurs alignements partiels du même pixel à la même surface, réalisés par une succession d'agents, que ce pixel devient totalement aligné. Nous utilisons pour cela le paramètre ξ représentant la force d'alignement avec laquelle l'agent aligne le pixel non homogène. La valeur utilisée de la force ξ est déterminée expérimentalement lors de la sélection des paramètres (cf. 5.2). La nouvelle profondeur du pixel en question est remplacée par la moyenne des profondeurs des pixels voisins, pondérée par la force d'alignement ξ .

$$I(x, y) = \frac{\xi}{V} \sum_{i=-1}^1 \sum_{j=-1}^1 I(x+i, y+j) \quad (4)$$

Dans ce cas également, on ne considère que les V pixels ayant au seuil Tr_c près, les mêmes courbures que celles de la surface courante (C_h^0, C_v^0) :

$$|C_h(x+i, y+j) - C_h^0| < Tr_c \wedge |C_v(x+i, y+j) - C_v^0| < Tr_c$$

Suite à l'alignement du premier pixel non homogène, l'agent devient à nouveau incapable d'altérer les pixels de l'image. A ce stade, plusieurs cas de figure se présentent : 1) Si le contour a une largeur d'un seul pixel, l'agent passe dans une nouvelle région homogène (Fig. 2b). Dans ce cas, il parcourt L pixels homogènes dans la nouvelle région, acquiert une capacité d'altération et commence ensuite à lisser les pixels homogènes dans la nouvelle région. 2) Si l'agent passe dans une région de bruit, il n'arrive pas alors à parcourir L pixels homogènes et en conséquence traverse et quitte cette région sans altérer aucun pixel. 3) Enfin, si le dernier pixel aligné est sur le bord d'un segment de contour épais, l'agent passe à l'intérieur de ce segment (Fig. 2a). Etant donné que la largeur du segment est inférieure à L , l'agent traverse le segment sans pour autant pouvoir parcourir L pixels homogènes. En effet, la valeur de L qui est sélectionnée de manière expérimentale (cf. section 5.2), doit être suffisamment grande pour qu'aucun contour ne dépasse en épaisseur cette valeur. Sinon, ce contour sera considéré comme une

vraie région car les agents dans ce cas peuvent parcourir au moins L pixels et peuvent par conséquent initialiser une région homogène. Cependant, la valeur de L ne doit pas dépasser un certain seuil, sinon certaines vraies régions de petite taille seront considérées comme de fausses régions (bruit).

3.3 Effacement des régions de bruit et émergence des contours

Une région de bruit dans une image de profondeur est soit une région homogène de faible taille, soit une région non homogène formée de pixels de bruit ayant des profondeurs aléatoires ou aberrantes. Lorsqu'il s'agit d'une région de faible taille, un agent qui la traverse n'arrive pas à parcourir L pixels homogènes contigus. Lorsqu'il s'agit d'une région formée de pixels de bruit, un agent qui la traverse ne trouve pas L pixels homogènes contigus pour pouvoir initialiser une région homogène ; et ce, quelle que soit la taille de cette région de bruit. Ces régions sont progressivement effacées par l'alignement des pixels de leurs bordures aux régions homogènes qui les entourent. En effet, lorsqu'un agent arrive sur un pixel de la bordure d'une région de bruit, il aligne ce pixel et passe à l'intérieur de cette région. Comme il n'arrive pas à parcourir L pixels homogènes contigus, l'agent demeure incapable d'altérer les pixels à l'intérieur de la région de bruit. Lorsqu'il quitte cette région, il n'aligne pas le premier pixel rencontré de la région homogène environnante. Ainsi, les bordures des régions de bruit sont, dans ce cas, continuellement alignées de l'extérieur au profit des régions environnantes. Ces régions finissent après un certain nombre de passages d'agents par être complètement effacées. Il est à noter qu'une région de bruit est entourée d'agents d'un même groupe. Ce groupe est formé des agents qui se déplacent sur la région homogène qui entoure la région de bruit.

Par contre, un segment de contour épais, situé entre deux régions homogènes, est entouré de deux groupes d'agents. Chacun de ces deux groupes est situé entièrement à l'intérieur d'une des deux régions. Les agents qui transitent par le segment alignent les pixels de la bordure du segment à la région dont ils proviennent. Le segment de contour entre les deux régions homogènes est alors continuellement aminci par les deux groupes d'agents (Fig. 2a et 2c). Quand le contour devient mince (d'un seul pixel de largeur), les agents des deux groupes se trouvent en compétition pour l'alignement des pixels du contour aminci. En effet, les pixels alignés par les agents d'un groupe donné, soit A dans la figure 2b, sont aussitôt réalignés à la seconde région (B) par les agents du second groupe. Ainsi, les pixels du segment de contour se trouvent en continuelle permutation entre les deux régions. En conséquence, quel que soit le nombre d'alignements de ces pixels, ils demeurent émergents dans l'image (Fig. 2d). Il est à noter que ce résultat n'est codé dans aucun des agents, mais il émerge des actions collectives des agents dans l'image.

A la fin du processus, les régions de bruit sont effacées et les contours

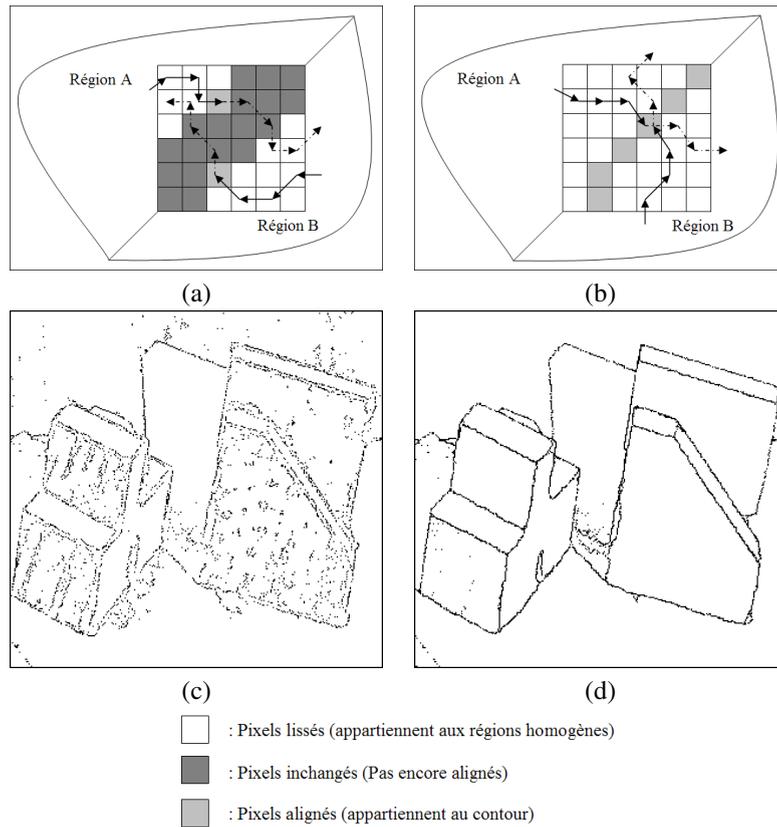


FIGURE 2 – Amincissement de contours (image abw.test.6). (a) Pixels de contour à $t=800$; (b) Pixels de contour à $t= 8000$

amincis. Ainsi, un accroissement simple de régions contrôlé par les contours détectés permet de produire une segmentation de l'image en ses différentes régions homogènes.

4 AUTO-ORGANISATION SPATIALE DES AGENTS

Partant du constat que les régions d'intérêt (contours et régions de bruit) ne représentent qu'un faible pourcentage de l'aire de l'image, nous avons opté pour l'auto-regroupement des agents autour de ces régions. En effet, les agents se déplacent au début aléatoirement, ensuite leurs mouvements sont pondérés par un champ de potentiel attractif qui se crée autour des pixels alignés. Le champ influence les agents à se diriger vers les régions d'intérêt.

Les actions des agents sont alors concentrées autour de ces régions, ce qui permet une amélioration considérable du temps de segmentation. Certains auteurs [6, 24, 23] ont utilisé le champ de potentiel pour la coordination spatiale d'agents. Cependant, ils ont procédé selon une approche statique, où le champ est créé à des positions prédéfinies correspondant aux emplacements des buts à atteindre ou des obstacles à éviter. Dans notre travail nous proposons une approche dynamique, où le champ de potentiel résulte de l'interaction des agents avec les objets de l'environnement (pixels de l'image). Le champ qui est exclusivement attractif dans notre cas est créé autour de tout pixel aligné. Ceci influence les agents qui sont dans le voisinage à se diriger vers la région où est situé le pixel aligné.

A une position (x, y) l'intensité $E(x, y)$ du champ engendré par un ensemble de M pixels qui viennent d'être alignés $\{(x_i, y_i), i = 1..M\}$ est exprimée comme suit :

$$E(x, y) = \sum_{i=1}^M \frac{1}{\sqrt{(x - x_i)^2 + (y - y_i)^2}} \quad (5)$$

Un agent situé à la position (x, y) et qui est sensible au champ de potentiel (ayant une capacité d'altération), est soumis à une force d'attraction \vec{F} . Elle se calcule par le gradient du champ, et s'exprime comme suit :

$$\vec{F} = \begin{cases} -\vec{\nabla} E(x, y), & \text{si } \text{capaciteAlteration} = \text{VRAI} \\ \vec{0}, & \text{sinon} \end{cases}$$

La force d'attraction générée par le champ de potentiel influence les agents à se diriger vers les pixels d'intérêt. Néanmoins, les agents gardent tout le temps une composante de mouvement arbitraire, leur permettant d'explorer toute l'image. Soit un vecteur unitaire de déplacement aléatoire \vec{R} et soit \vec{F} la force d'attraction du champ de potentiel ; le déplacement de l'agent se fait donc dans la direction du vecteur $\vec{D} = \vec{R} + \vec{F}$. Après avoir altéré un pixel, l'agent se libère du champ car il perd sa capacité d'altération et devient donc insensible au champ. Ceci lui permet de s'éloigner du centre d'attraction vers lequel il vient d'être dirigé. Plus tard, lorsqu'il aura acquis de nouveau une capacité d'altération, il pourra être suffisamment éloigné pour être éventuellement soumis à un autre centre d'attraction.

Le mécanisme de relaxation introduit opère de telle sorte que le potentiel créé autour d'une région donnée tend à s'amortir au fil du temps suite aux alignements répétés des pixels d'intérêt de la dite région. En effet, dans le cas d'une région de bruit, elle finit par s'effacer et le champ ne se créera désormais plus dans la zone correspondante de l'image. Cependant, dans le cas d'un contour, l'intensité du champ décroît après chaque altération d'un

de ses pixels. L'équation de la dynamique de relaxation en fonction du temps est exprimée comme suit :

$$E_{t+1}(x, y) = \mu \times E_t(x, y), \mu < 1 \quad (6)$$

où t représente le nombre d'alignements d'un même pixel d'intérêt. La valeur initiale $E_1(x, y)$ est l'intensité du champ créé suite à la première altération du pixel. Le taux d'amortissement μ est pris de manière empirique égal à 0.9. Le champ généré autour d'un pixel donné tend à s'annuler après plusieurs altérations. Lorsque toutes les régions de bruit sont effacées et tous les contours amincis, le champ est entièrement relaxé dans toute l'image. A cette étape, les agents reprennent un mouvement aléatoire libre de toute attraction et se dispersent uniformément dans l'image. Cet état du processus indique la fin de la segmentation et permettra ainsi d'arrêter le système.

5 EXPÉRIMENTATION ET ANALYSE

5.1 Méthode d'évaluation

Hoover et ses collaborateurs [9] ont proposé un framework dédié à l'évaluation des algorithmes de segmentation d'images de profondeur. Ce framework a été utilisé par plusieurs auteurs dans le domaine pour évaluer leurs algorithmes de segmentation et pouvoir les comparer aux autres [11, 10, 14, 4, 1, 18]. L'évaluation d'une segmentation selon ce framework consiste à comparer les images segmentées produites par une machine de segmentation donnée MS, aux images segmentées manuellement dont la segmentation correspond à la réalité terrain GT (Ground Truth). Cette comparaison permet de faire ressortir pour une image donnée les nombres respectivement de régions correctement détectées, de régions sur-segmentées, de régions sous-segmentées, de régions omises et de régions de bruit. La comparaison des régions de MS et de GT se fait selon un seuil de tolérance T dont la valeur varie de 50% pour une comparaison peu rigoureuse à 100% pour une rigueur maximale. Les auteurs du framework ont fourni également un ensemble d'images d'apprentissage et un ensemble d'images de test, le tout avec leur segmentation manuelle GT. Pour l'expérimentation de la méthode proposée nous avons utilisé deux ensembles d'images. Le premier est l'ensemble ABW dont les scènes sont composées exclusivement d'objets polyédriques. L'ensemble est composé de 10 images d'apprentissage et de 30 images de test, chacune de 512×512 pixels. Le second ensemble est K2T, formé de 20 images d'apprentissage et de 40 images de test, chacune de 640×480 pixels. Les images de cet ensemble représentent des scènes à objets en majorité courbés. Pour la comparaison de nos résultats avec ceux fournis avec le framework, nous avons considéré toutes les méthodes citées dans Hoover et al. [9]. Utilisant 5 paramètres ajustables, l'algorithme USF (University of South Florida) est basé sur l'estimation du plan tangent à la surface

en chaque pixel de l'image. Il procède ensuite à un accroissement de régions par ajout de points ayant les mêmes plans tangents. Ayant 10 paramètres à ajuster, l'algorithme WSU (Washington State University) procède à l'estimation des vecteurs normaux aux surfaces par une analyse en composantes principales. Les composantes de position et d'orientation sont regroupées et classifiées en spécifiant le nombre maximal de classes (régions). Certaines régions adjacentes sont ensuite fusionnées et ce, en fonction de leur degré de coplanarité. L'idée de l'algorithme UB (University of Bern) est que les pixels appartenant à un segment de droite de la ligne de scan (intersection du plan de Laser avec les objets de la scène) appartiennent tous à la même région plane. L'accroissement de régions est donc effectué en utilisant les segments de droite au lieu des pixels pris individuellement. L'algorithme utilise 11 paramètres qui doivent être préalablement ajustés. Suivant le même chemin que l'algorithme USF, l'algorithme UE (University of Edinburgh) utilise les courbures moyenne et gaussienne avec l'algorithme d'accroissement de régions en employant une douzaine de paramètres. Les résultats obtenus par l'accroissement de régions sont ensuite améliorés par un traitement spécifique des pourtours des régions.

Dans la suite de la section, nous présentons la sélection des paramètres du système et puis nous montrons les différents résultats expérimentaux obtenus par les deux ensembles d'images de test ABW et K2T.

5.2 Apprentissage de paramètres

Nous avons scindé les paramètres de notre système en deux ensembles. Le premier est formé des paramètres Tr_c et ξ représentant respectivement le seuil de comparaison de courbures et la force d'alignement des pixels. Cet ensemble de paramètres agit sur la qualité de détection de contours et d'effacement des régions de bruit. Le second ensemble est formé des paramètres N et L , représentant respectivement le nombre d'agents placés dans l'image et la longueur du parcours pour l'initialisation d'une région homogène. Ces deux paramètres influencent la dynamique du système. Pour le premier sous-ensemble de paramètres, nous avons lancé le système pour toutes les combinaisons $(Tr_c, \xi) \in \{0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07\} \times \{0.1, 0.25, 0.50, 0.75, 0.9\}$, à exécuter sur les images d'apprentissage des deux ensembles ABW et K2T.

Les valeurs considérées pour les paramètres Tr_c sont proposées suite à un échantillonnage de la courbure dans un certain nombre d'images, tandis que les valeurs de ξ sont proposées suite à un certain nombre d'essais de segmentation en considérant l'intervalle où les résultats visuels sont satisfaisants. Le critère de performance utilisé pour ces deux paramètres est le nombre moyen de régions correctement détectées dans les images d'apprentissage, avec le seuil de tolérance $T=80\%$. Les résultats d'apprentissage nous ont conduit à prendre $Tr_c = 0.03$ et $\xi = 0.25$.

Le même ensemble d'images d'apprentissage est utilisé pour estimer les

paramètres N et L . Pour ces derniers, nous avons considéré les combinaisons suivantes : $(N, L) \in \{1500, 2000, 2500, 3000, 3500\} \times \{3, 5, 7, 9, 11\}$. Par contre, nous avons considéré comme critère de performance le nombre de régions de bruit issues de la segmentation. En effet, ce choix vient du fait que ces deux paramètres lorsqu'ils ne sont pas bien sélectionnés, engendrent un nombre élevé de régions de bruit.

L'utilisation d'un nombre insuffisant d'agents pour une taille donnée des images traitées (512×512 ou 640×480 pixels), conduit à un sous-lissage de l'image et à un sous-amincissement et déformation des contours. Les régions qui en résultent sont amputées d'un nombre important de pixels qui auraient dû être inclus dans ces régions. Aussi, une faible valeur de la longueur de parcours L conduit à la prise en compte de petites régions homogènes qui devraient être considérées comme régions de bruit. Aussi, un contour épais dans ce cas peut être détecté comme étant une région courbée. Par contre, une valeur élevée de L peut conduire à l'omission de certaines vraies régions homogènes qui sont relativement étroites (cf. section 3.3). L'exécution de toutes les combinaisons a permis de fixer N à 2500 et L à 7.

5.3 Résultats expérimentaux

Afin de montrer comment les régions de bruit sont progressivement effacées et comment les lignes de contour sont progressivement amincies, nous avons affiché les résultats de détection à des intervalles réguliers de temps avant que la segmentation ne soit achevée. Le temps t exprime le nombre de pas effectués par chaque agent du système sur l'image depuis le début de la segmentation. La figure 3a montre une image de profondeur à base de surfaces courbées. Les figures 3b, 3c, 3d, 3e et 3f montrent le niveau de détection respectivement à $t=500, 3500, 7500, 10500$ et 13500 . Nous pouvons constater que les contours sont progressivement détectés et amincis. A la fin du processus nous avons des lignes de contour d'un seul pixel de largeur. Les régions de bruit quant à elles se rétrécissent progressivement et finissent par disparaître de l'image.

Afin de montrer l'intérêt de notre approche de segmentation, nous avons comparé nos résultats à ceux des méthodes citées par Hoover et al. [9]. Pour les objets polyédriques, nous avons pu confronter nos résultats à ceux des quatre méthodes USF, WSU, UB et UE. Cependant, il n'y a de résultats de segmentation d'images à base d'objets courbés que pour la méthode UB, auxquels nous confrontons nos résultats de segmentation d'images à base d'objets courbés.

5.3.1 Comparaison avec l'ensemble ABW

A l'instar de tous les auteurs qui ont utilisé le présent framework, nous avons calculé les résultats moyens de toutes les métriques de performance pour toutes les images de test, qui sont au nombre de 30 pour l'ensemble

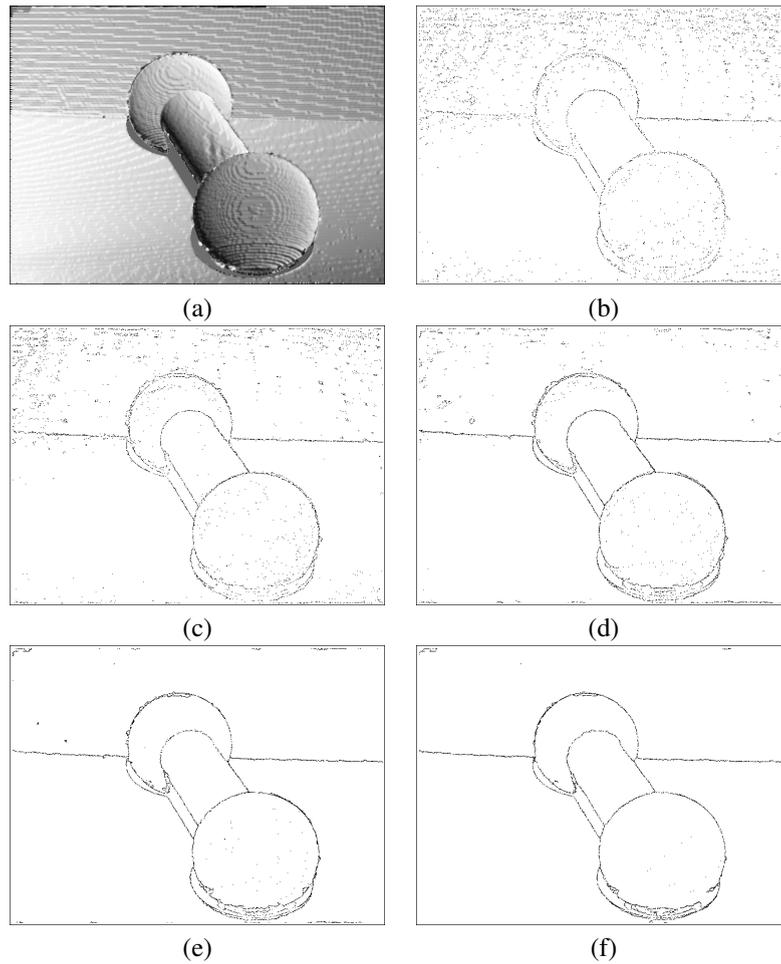


FIGURE 3 – Progression de segmentation. (a) Image de profondeur rendue réaliste ; (b) à $t=500$, (c) à $t=3500$; (d) à $t=7500$; (e) à $t=10500$; (f) à $t=13500$

ABW. Nous avons synthétisé nos résultats ainsi que ceux des quatre méthodes impliquées dans la comparaison dans la table 1, en considérant une valeur typique du seuil de tolérance $T=80\%$. Cette valeur typique a été prise par la majorité des auteurs dans le domaine [9, 11, 10, 14, 4, 1, 18].

Pour cet ensemble d'images, les résultats obtenus par notre système (AASI pour Approche Agent pour la Segmentation d'Images) sont très comparables à ceux du système UE, qui représente le meilleur système comparé aux trois autres. En effet, notre système a pu correctement détecter une moyenne de 13.1 régions sur une moyenne de régions de 15.2 de l'ensemble de test ABW.

TABLE 1 – Résultats moyens des cinq méthodes sur l'ensemble ABW avec $T=80\%$

Méthode	Régions GT	Délect. Correct.	Sur-segment.	Sou-segment.	Régions omises	Régions bruit
USF	15.2	12.7	0.2	0.1	2.1	1.2
WSU	15.2	9.7	0.5	0.2	4.5	2.2
UB	15.2	12.8	0.5	0.1	1.7	2.1
UE	15.2	13.4	0.4	0.2	1.1	0.8
AASI	15.2	13.1	0.5	0.1	1.8	0.8

D'autre part, il a affiché respectivement des moyennes de 0.5, 0.1, 1.8 et 0.8 respectives aux métriques de sur-segmentation, sous-segmentation, omission et bruit. Ces résultats sont comparables à ceux de UE et sont meilleurs que ceux de USF, WSU et UB.

La figure 4 représente les résultats moyens de détection correcte correspondant aux différentes valeurs du seuil T et en considérant toutes les images de l'ensemble ABW. Ces résultats montrent que notre système produit une détection correcte significativement meilleure que celle des systèmes USF, WSU et UB, et très proche de celle du système UE.

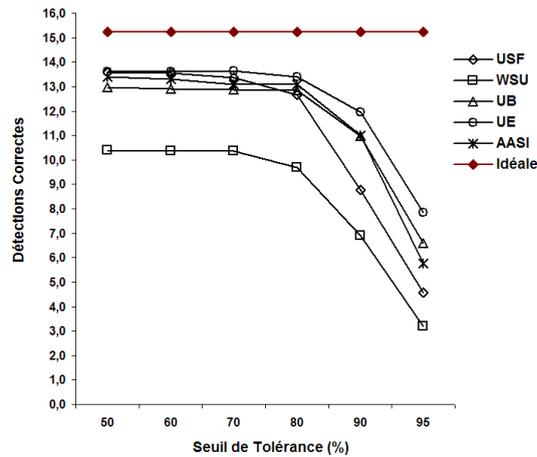


FIGURE 4 – Résultats moyens du nombre de régions correctement détectées de l'ensemble ABW de toutes les méthodes aux différentes valeurs du seuil de tolérance T ; $50\% \leq T < 100\%$

TABLE 2 – Résultats moyens des méthodes UB et AASI sur l'ensemble K2T avec $T=80\%$

Méthode	Régions	Délect.	Sur-	Sou-	Régions	Régions
	GT	Correct.	segment.	segment.	omises	bruit
UB	4.6	3.0	0.1	0.5	2.1	0.8
AASI	4.6	3.3	0.1	0.3	2.0	0.9

5.3.2 Comparaison avec l'ensemble K2T

Pour l'ensemble d'images K2T représentant des scènes contenant des objets en majorité courbés, il n'y a que l'algorithme UB qui est concerné par cet ensemble. Nous avons comparé nos résultats sur l'ensemble K2T avec les résultats de UB obtenus sur le même ensemble. La table 2 affiche les résultats des deux méthodes. Ces résultats sont exprimés sous forme de moyennes des différentes métriques de performance au seuil de tolérance $T=80\%$. A ce seuil de tolérance, notre système a permis de détecter une moyenne de 3.3 régions correctes sur une moyenne de 4.6 de l'ensemble de test K2T. Ceci représente un taux appréciable comparé à celui obtenu par la méthode UB.

La figure 5 montre les moyennes de détections correctes des deux méthodes pour toutes les valeurs considérées du seuil de tolérance T . Nous pouvons clairement constater l'écart en nombre de régions correctement détectées, notamment pour les valeurs 50%, 60%, 70% et 80% du seuil T . Les deux méthodes affichent des résultats similaires pour les valeurs 90% et 95% de T .

5.4 Temps d'exécution

Dans notre approche, le temps moyen de segmentation est indépendant de la nature des objets contenus dans les images à segmenter. Que ce soit pour des objets polyédriques ou pour des objets courbés, les agents procèdent par le même calcul pour l'altération des pixels de l'image. Les paramètres qui influencent le temps d'exécution sont : 1) La taille de l'image, où un nombre adéquat d'agents doit être utilisé pour sa segmentation ; 2) La force d'alignement ξ dont la valeur doit assurer un bon compromis entre la vitesse de segmentation et la qualité des contours. Nous avons enregistré le temps moyen de segmentation des images de l'ensemble ABW, en exécutant le système sur un PC Compaq nx8220 ayant une fréquence de l'horloge du processeur de 1.7 GHZ et une RAM de 512 MO. Le temps moyen de segmentation d'une image est de 11 secondes. Cette performance est comparable à celle de l'algorithme UB qui affiche un temps moyen sur le même ensemble de 7 secondes, exécuté sur Sun SparcStation 20. Cependant, elle est largement meilleure que celles des algorithmes USF, WSU et UE qui affichent respec-

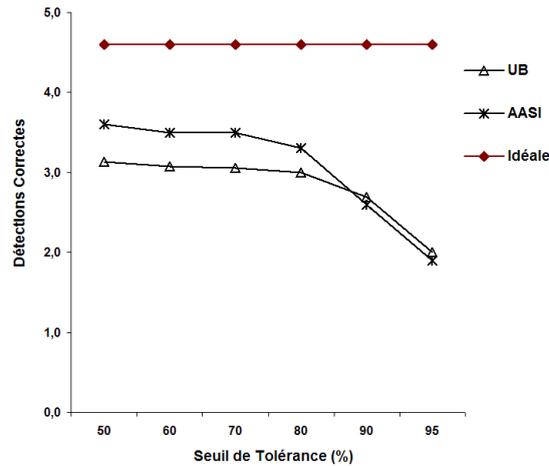


FIGURE 5 – Résultats moyens du nombre de régions correctement détectées de l'ensemble K2T des méthodes UB et AASI aux différentes valeurs du seuil de tolérance T ; $50\% \leq T < 100\%$

tivement 78 minutes sur une Sun SparcStation 20, 4.4 minutes sur une HP 9000/730 et 9.1 minutes sur une Sun SparcStation 5. Par ailleurs, la nature réactive des agents utilisés et la nature asynchrone de leurs actions permet le déploiement du système proposé sur une architecture massivement parallèle, où les agents sont affectés aux unités de traitement et partagent une mémoire commune mémorisant l'image à segmenter. Ceci permet encore de réduire considérablement le temps d'exécution et permet de déployer le système pour des applications temps réel.

6 CONCLUSION

Nous avons proposé dans cet article, un système à base d'agents réactifs pour la détection de contours et l'élimination de bruit dans une image de profondeur, permettant sa segmentation en ses différentes régions homogènes. L'utilisation de la courbure de surface a permis d'étendre l'approche aux images à base d'objets courbés. Ces objets sont reconnus comme étant des objets complexes à segmenter où peu d'auteurs ont proposé des méthodes de segmentation qui leur sont dédiées. Selon notre approche, la détection d'un segment de contour résulte de l'alignement alternatif de ses pixels. Cet alignement est opéré par des agents provenant des deux régions incidentes à ce segment. D'autre part, les régions de bruit sont progressivement effacées par l'alignement continu des pixels de leurs pourtours aux régions homogènes qui les entourent. Ces deux fonctionnalités ne sont codées dans aucun des

agents, mais émergent de l'action collective de tous les agents sur les pixels de l'image. Par ailleurs, le couplage faible qui caractérise les agents utilisés, ainsi que la nature parallèle de leurs actions permet de déployer le système sur une architecture parallèle, assurant un temps de segmentation souhaitable aux applications temps réel. L'approche a été rigoureusement évaluée en utilisant un framework dédié et deux ensembles d'images couvrant à la fois les scènes à base d'objets polyédriques et les scènes à base d'objets courbés. Les résultats obtenus nous ont permis de conclure à l'efficacité et la précision de la méthode proposée. L'abstraction faite sur la nature des surfaces qui peuvent former les objets permet à l'approche d'être utilisée pour d'autres types d'images. A cette fin, il suffit de définir le bon critère d'homogénéité des régions et de formuler les actions des agents en fonction de ce critère. En perspective à ce travail, nous envisageons l'intégration d'autres critères d'homogénéité de surface, tels que des expressions plus sophistiquées de la courbure et des opérateurs de morphologie de surfaces. Nous croyons que ceci permettra d'améliorer les résultats de segmentation sur les ensembles de test et d'étendre encore la méthode à des objets plus complexes.

RÉFÉRENCES

- [1] A. Bab Hadiashar et N. Gheissari. Range image segmentation using surface selection criterion. *IEEE Transactions on Image Processing*, 15(7) :2006–2018, July 2006.
- [2] P. Ballet, V. Rodin et J. Tisseau. Edge detection using a multiagent system. In *10th Scandinavian Conference on Image Analysis*, pages 621–626, Lappeenranta, Finland, 1997.
- [3] C. Bourjot, V. Chevrier et V. Thomas. How social spiders inspired an approach to region detection. In *AAMAS '02 : Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 426–433, New York, NY, USA, 2002. ACM.
- [4] Y. Ding, X. Ping, M. Hu et D. Wang. Range image segmentation based on randomized hough transform. *Pattern Recognition Letters*, 26(13) :2033–2041, 2005.
- [5] T.J. Fan, G.G. Medioni et R. Nevatia. Segmented description of 3-D surfaces. *IEEE J. Robotics Automat.*, 3(6) :527–538, December 1987.
- [6] J. Ferber. *Les systèmes multi-agents : vers une intelligence collective*. Informatique, Intelligence Artificielle. InterÉditions, 1995.
- [7] S. Geman et D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6) :721–741, 1984.
- [8] F. Harrouet, J. Tisseau, P. Reignier et P. Chevallier. oRis : un environnement de simulation interactive multi-agents. *Technique et Science Informatiques*, 21(4) :499–524, 2002.

- [9] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. W. Bowyer, D. W. Eggert, A. W. Fitzgibbon et R. B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7) :673–689, 1996.
- [10] X. Jiang, K. W. Bowyer, Y. Morioka, S. Hiura, K. Sato, S. Inokuchi, M. Bock, C. Guerra, R. E. Loke et J. M. Hans du Buf. Some further results of experimental comparison of range image segmentation algorithms. In *International Conference on Pattern Recognition*, volume 4, pages 4877–4882, 2000.
- [11] X. Jiang et H. Bunke. Edge detection in range images based on Scan Line approximation. *Computer Vision and Image Understanding*, 73(2) :183–199, 1999.
- [12] J. Jones et M. Saeed. Image enhancement, an emergent pattern formation approach via decentralised multi-agent systems. *Multiagent and Grid Systems Journal (ISO Press) Special Issue on Nature inspired systems for parallel, asynchronous and decentralised environments*, 3(1) :105–140, 2007.
- [13] S.B. Kang et K. Ikeuchi. The complex EGI : A new representation for 3-D pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7) :707–721, 1993.
- [14] S. Li et D. Zhao. Gradient-based polyhedral segmentation for range images. *Pattern Recognition Letters*, 24(12) :2069–2077, 2003.
- [15] S. Z. Li. *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [16] J. Liu et Y. Y. Tang. Adaptive image segmentation with distributed behavior-based agents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6) :544–551, 1999.
- [17] S. Mazouzi et M. Batouche. A new bayesian method for range image segmentation. In *6th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 4679 of *Lecture Notes in Computer Science*, pages 453–466. Springer, 2007.
- [18] S. Mazouzi, Z. Guessoum, F. Michel et M. Batouche. A multi-agent approach for range image segmentation. In *5th International Central and Eastern European Conference on Multi-Agent Systems*, volume 4696 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2007.
- [19] H. Mobahi, M. N. Ahmadabadi et B. N. Araabi. Swarm contours : A fast self-organization approach for snake initialization : Research articles. *Complex.*, 12(1) :41–52, 2006.
- [20] J. Park et G. N. DeSouza. 3-d modeling of real-world objects using range and intensity images. In Bruno Apolloni, Ashish Ghosh, Ferda Nur Alpaslan, Lakhmi C. Jain et Srikanta Patnaik, éditeurs, *Ma-*

- chine Learning and Robot Perception*, volume 7 of *Studies in Computational Intelligence*, pages 203–264. Springer, 2005.
- [21] N. Richard, M. Dojat et C. Garbay. Automated segmentation of human brain MR images using a multi-agent approach. *Artificial Intelligence in Medicine*, 30(2) :153–176, 2004.
 - [22] V. Rodin, A. Benzinou, A. Guillaud, P. Ballet, F. Harrouet, J. Tisseau et J. Le Bihan. An immune oriented multi-agent system for biological image processing. *Pattern Recognition*, 37(4) :631–645, 2004.
 - [23] O. Simonin. Construction of numerical potential fields with reactive agents. In *AAMAS '05 : Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1351–1352, New York, NY, USA, 2005. ACM Press.
 - [24] T. Tsuji, Y. Tanaka, P. Morasso, V. Sanguineti et M. Kaneko. Biomimetic trajectory generation of robots via artificial potential field with time base generator. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 32(4) :426–439, 2002.
 - [25] L. Tummolini et C. Castelfranchi. Trace signals : The meanings of stigmergy. In Danny Weyns, H. Van Dyke Parunak et Fabien Michel, éditeurs, *E4MAS*, volume 4389 of *Lecture Notes in Computer Science*, pages 141–156. Springer, 2006.