



**HAL**  
open science

## Fusion multi-vue d'informations de silhouettes à l'aide d'une grille d'occupation 3D

Jean-Sébastien Franco, Edmond Boyer

► **To cite this version:**

Jean-Sébastien Franco, Edmond Boyer. Fusion multi-vue d'informations de silhouettes à l'aide d'une grille d'occupation 3D. 15e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle, Jan 2006, Tours, France. pp.1747–1753. inria-00349093

**HAL Id: inria-00349093**

**<https://inria.hal.science/inria-00349093v1>**

Submitted on 23 Dec 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Fusion multi-vue d'informations de silhouettes à l'aide d'une grille d'occupation 3D

## Fusion of Multi-View Silhouette Cues Using a Space Occupancy Grid

Jean-Sébastien Franco

Edmond Boyer

INRIA Rhône-Alpes  
655, avenue de l'Europe, 38334 Saint-Ismier CEDEX, FRANCE  
nom.prenom@inrialpes.fr

### Résumé

Nous nous plaçons dans un contexte où de multiples caméras filment une scène. Nous disposons, pour chaque vue, d'images des probabilités d'appartenance à la silhouette d'objets de la scène, obtenues par un procédé de soustraction de fond. Dans cet article, nous explorons ce qui peut être déduit de cette information silhouette multi-vue. Dans ce but, nous proposons une nouvelle méthode pour fusionner de telles informations. Notre principale contribution est l'introduction du concept de grille d'occupation, popularisée chez les roboticiens, dans un contexte multi-caméras. L'idée centrale est de considérer que chaque pixel est un capteur apportant de l'information sur la scène observée, et de le modéliser statistiquement comme tel. Les observations rapportées par les pixels de toutes les caméras peuvent alors être conjointement utilisées pour déduire où la matière se trouve dans la scène, et avec quelle probabilité. Comme nos résultats l'illustrent, ce modèle simple présente de nombreux avantages. La plupart des sources d'incertitude peuvent être prises en compte explicitement, et aucune décision prématurée sur l'état des pixels, et leur appartenance à une silhouette, n'est nécessaire. Le modèle permet de s'affranchir des contraintes de visibilité communes aux méthodes classiques de reconstruction d'enveloppes visuelles.

### Mots Clef

Vision par ordinateur, silhouettes, fusion de capteurs, grille d'occupation, enveloppes visuelles, modélisation 3D à partir d'images, reconstruction 3D.

### Abstract

*In this paper, we investigate what can be inferred from several silhouette probability maps, in multi-camera environments. To this aim, we propose a new framework for multi-view silhouette cue fusion. This framework uses a space occupancy grid as a dense probabilistic 3D representation of scene contents. Such a representation is of great interest for various computer vision applications in perception,*

*or localization for instance. Our main contribution is to introduce the occupancy grid concept, popular in the robotics community, for multi-camera environments. The idea is to consider each camera pixel as a statistical occupancy sensor. All pixel observations are then used jointly to infer where, and how likely, matter is present in the scene. As our results illustrate, this simple model has various advantages. Most sources of uncertainty are explicitly modeled, and no premature decisions about pixel labeling occur preserving therefore pixel knowledge. Consequently, scene object localization and robust volume reconstruction can be achieved, with no constraint on camera placement and object visibility. In addition, it is possible to compute improved consistent silhouettes in original views using this representation.*

### Keywords

Computer vision, shape from filhouettes, sensor fusion, occupancy grid, visual hull, 3D Modeling from multiple views, 3D reconstruction.

## 1 Introduction

Les méthodes de reconstruction à partir de silhouettes sont populaires avec les environnements multi-caméra, du fait de leur simplicité et efficacité. Ces méthodes sont utiles pour la modélisation à partir d'images, la localisation de multiples objets dans une scène, et les applications de capture de mouvement, entre autres. Cependant de telles méthodes se basent souvent sur un étiquetage binaire des pixels en deux catégories, le fond et l'avant-plan, pour identifier la silhouette des objets. Cet étiquetage est généralement réalisé séparément dans chaque image, avant tout calcul en 3D, et porte le nom de *soustraction de fond*. Malheureusement, un tel étiquetage monoculaire est difficile à réaliser de façon fiable dans un environnement général et peu contrôlé. Diverses perturbations expliquent cette difficulté : le bruit des capteurs CCD, les ambiguïtés de couleur entre le fond et l'avant-plan, les changements d'illumination de la scène (ce qui inclut les ombres d'objets d'inté-

rêts), etc. De plus, des erreurs dans un tel processus peuvent avoir des conséquences drastiques sur la perception 3D multi-caméra, particulièrement en présence de bruit de calibration des caméras, ou si l'acquisition des images n'est pas simultanée pour toutes les caméras.

Notre but est donc de trouver une représentation de l'information silhouette multi-vue, où l'inférence sur les silhouettes est d'une plus grande robustesse aux différentes sources d'incertitude mentionnées plus haut. Intuitivement, la connaissance simultanée de toutes les images nous apporte plus d'information sur les silhouettes que la connaissance d'une seule de ces images. Cette idée nous a conduits à calculer une fusion des silhouettes en 3D, pour prendre en compte de manière optimale la contribution de toutes les images. Le résultat d'une telle fusion contient naturellement une information de forme, et peut donc être utilisée pour des applications de modélisation classique à partir d'images ; mais elle peut aussi être utilisée pour améliorer l'extraction des silhouettes dans les images, pour toute application basée sur les silhouettes.

Dans les méthodes existantes, on utilise généralement deux tâches distinctes pour inférer la forme des objets à partir de silhouettes : une décision sur l'occupation des silhouettes est d'abord effectuée pour chaque image, puis la forme et la position des objets sont inférées géométriquement à partir des silhouettes disponibles en utilisant les méthodes des *enveloppes visuelles* [13]. Ces méthodes peuvent conduire à une représentation surfacique des objets d'intérêt [14, 16], voxellique [19], ou une représentation à la résolution des images [17]. Bien que l'estimation par enveloppe visuelle peut-être exacte à partir de silhouettes [7], le modèle produit souffre d'une décision binaire dans chaque image, héritant des défauts dûs aux perturbations précédemment mentionnées. Notre approche permet de repousser la décision d'occupation et permet donc une meilleure intégration de l'information disponible.

Plusieurs méthodes ont déjà été proposées pour s'affranchir complètement de l'estimation des silhouettes. En effet beaucoup d'algorithmes reconstruisent la structure 3D d'une scène en se basant seulement sur l'information photométrique [12]. D'autres formulent cette structure comme la solution d'un problème d'optimisation global, en utilisant des techniques variationnelles [6], ou par coupe optimale de graphe [8, 11]. Cependant ces méthodes sont de plus grandes complexité par rapport aux méthodes à partir de silhouettes, car contrairement à celles-ci, elles doivent traiter le problème de visibilité des points sur la surface de l'objet. C'est pourquoi beaucoup d'applications privilégient les méthodes à partir de silhouettes (comme les plates-formes de réalité virtuelle, les applications temps-réel), ou s'en servent pour initialiser une méthode photométrique plus complexe [10].

Parmi les travaux les plus proches de notre problématique, Magnor *et al.* [8] proposent une estimation simultanée de la disparité stéréo et des silhouettes, avec cependant

le coût élevé de l'optimisation globale pour garantir la robustesse. Zeng *et al.* proposent une soustraction de fond multi-vue, avec un schéma itératif assez coûteux et dont la convergence n'est pas assurée, avec la contrainte supplémentaire de la visibilité de l'objet d'intérêt par toutes les caméras [21]. Des travaux ont été effectués en robotique pour localiser un objet à partir d'une séquence d'images acquises à partir d'un robot [15], avec cependant une problématique assez différente de la notre. Plusieurs travaux ont aussi exploré la possibilité de représenter l'espace avec une grille de probabilité, pour résoudre d'autres problèmes tels la stéréo (wide-baseline) [2] ou la reconstruction d'objets transparents [1]. Grauman *et al.* [9] proposent une méthode intéressante pour estimer le jeu de silhouettes le plus probable d'un même objet, en apprenant *a priori* les silhouettes humaines, présentant l'avantage d'une meilleure intégration de l'information sémantique, mais avec une généralité et une précision limitée. Toutes ses approches résolvent des problèmes connexes, avec certaines limitations. Nous présentons une approche bas niveau, et nous proposons d'enrichir l'information silhouette 2D en réalisant sa fusion dans une représentation 3D, tout en restant générique par rapport aux applications.

Nous proposons un nouvel outil basé sur les grilles d'occupation : il s'articule autour d'une représentation de la scène sous la forme d'une grille de voxels pour laquelle nous calculons la probabilité d'occupation d'un objet d'intérêt, associée à un modèle capteur. Les grilles d'occupation ont été beaucoup utilisées dans la communauté robotique [5, 4], pour représenter et reconstituer l'environnement de navigation d'un robot à partir d'observations de sonars. Notre contribution consiste à étendre le concept de grille d'occupation aux capteurs CCD, et à reformuler le problème d'estimation de forme à partir de silhouettes comme un problème de fusion de capteurs. Pour ce faire, nous associons à chaque pixel un modèle capteur *génératif*, où l'on modélise la réponse du pixel aux occupations de voxels dans la scène. Notre formulation tient compte de la région visible par chaque pixel, du problème d'échantillonnage des voxels, des petites erreurs de calibration, et de la fiabilité du pixel en tant que capteur. Ce modèle est alors utilisé pour résoudre la question inverse plus difficile : savoir où se trouve la matière dans la scène, connaissant les couleurs observées par tous les pixels. Nous montrons aussi que la grille d'occupation calculée peut être utilisée pour réaliser une soustraction de fond multi-vue, où l'estimation de la silhouette dans chaque vue bénéficie de la connaissance acquise dans d'autres vues.

## 2 Formulation du problème

Nous considérons le problème de fusion des informations silhouette multi-vue. Nous supposons disposer d'un jeu d'images *courantes*, obtenues à partir de caméras complètement calibrées. Nous supposons également disposer d'un

jeu d'images du fond, obtenues préalablement à partir de la scène dénuée d'*objets d'intérêt*. Aucune supposition n'est faite quant à l'existence d'une région de visibilité commune à toutes les caméras.

Nous formulons le problème comme l'estimation bayésienne pour chaque voxel, de la probabilité avec laquelle celui-ci est occupé par un objet d'intérêt. Nous utilisons un modèle capteur génératif : nous considérons que l'occupation des voxels est la cause, et les observations les effets. Nous modélisons donc l'influence de chaque voxel sur la formation des images. Ceci nous permet de résoudre, grâce à l'inférence bayésienne, le problème inverse : exprimer la vraisemblance de l'occupation des voxels à partir des images, traitées comme des mesures bruitées de la scène.

La résolution d'un problème bayésien requiert l'expression de la distribution de probabilité conjointe de co-occurrence de toutes les variables du problème (définies au paragraphe 2.1), avant toute inférence. Cette distribution de probabilité conjointe doit ensuite être décomposée et simplifiée, en fonction des dépendances statistiques que l'on considère entre les variables (paragraphe 3). En particulier, des formes paramétriques doivent être affectées aux différents termes de la décomposition pour donner une forme explicite aux relations liant plusieurs variables (paragraphe 3.2 et 3.3). Ceci réduit considérablement la complexité des calculs effectués à partir de la distribution conjointe, lors de l'inférence avec la règle de Bayes, utilisée pour calculer les distributions des variables d'occupation des voxels (paragraphe 4).

## 2.1 Principales variables du problème

Nous dénotons l'ensemble des  $n$  images courantes  $\mathcal{I}$ .  $\mathcal{I}^i$ ,  $i = 1 \dots n$  représente alors les données image de la caméra  $i$ , et  $\mathcal{I}_p^i$  représente les données image au point  $p$  dans l'image  $i$ , exprimées dans un certain espace de couleur (RGB, YUV, etc). Bien que non étudié dans cet article, on peut imaginer sans perte de généralité avoir plus de données image, tel le gradient ou le laplacien image par exemple, encapsulées dans le terme  $\mathcal{I}_p^i$ . Nous supposons que les données image correspondant aux  $m$  images du fond statique observées pour chaque caméra peuvent être résumées dans une seule image  $\mathcal{B}^i$ ,  $i = 1 \dots n$ , représentant les paramètres d'un modèle statistique utilisé pour la régression. Le but d'une telle régression étant habituellement de modéliser le bruit capteur. Tous les jeux de données image sont produits par  $n$  caméras dont on connaît les matrices de projection  $\mathbf{P}^i$ .

$\tau$  symbolise l'information *a priori* que nous introduisons dans le modèle. Ceci inclut notre connaissance de la scène, ce que nous connaissons des caractéristiques des capteurs, nos connaissances générales du problème.

Soit  $\mathcal{G}$  notre grille d'occupation. Pour chaque point de l'espace  $X$  de cette grille nous associons une variable d'occupation binaire  $\mathcal{G}_X \in \{0, 1\}$ , respectivement libre ou occupée. Nous supposons l'indépendance statistique entre les occupations de voxels et calculons chaque occupa-

tion de voxel de manière indépendante. Ceci est une hypothèse couramment utilisée pour rendre le traitement de grilles d'occupation possible en robotique [5]. Les résultats montrent que l'estimation indépendante telle que nous la proposons, même si elle n'est pas aussi exhaustive qu'une recherche globale sur tout l'espace des configurations possible de la grille, permet d'obtenir une information très robuste et utilisable, pour un coût en calcul beaucoup plus raisonnable.

Nous avons défini les variables d'entrée et de sortie de notre problème. Nous définissons un important jeu de variables cachées dans les images, les cartes de détection des silhouettes  $\mathcal{F}^i$ ,  $i = 1 \dots n$ . Ces cartes définissent, pour chaque pixel  $p$  de l'image  $i$ , une variable binaire de détection de silhouette  $\mathcal{F}_p^i$ .  $\mathcal{F}_p^i = 1$  si le capteur au pixel  $p$  de l'image  $i$  témoigne de la présence d'un objet sur sa ligne de vue. Nous insistons sur cette définition, car il y a une possibilité qu'il y ait effectivement un objet sur la ligne de vue du pixel  $p$ , mais que le capteur *se trompe* et ne rapporte pas cette information, en raison de causes internes ou externes (la modélisation des défaillances capteurs est discutée en détail au paragraphe 3.2). Ces cartes de détection représentent l'information silhouette dans notre modèle, que nous souhaitons marginaliser dans l'inférence.

## 3 Décomposition de la distribution conjointe

Notre but est d'inférer l'occupation  $\mathcal{G}_X$  d'un voxel à la position  $X$ , sachant  $\mathcal{I}$ ,  $\mathcal{B}$ , et  $\tau$ . De ce fait, nous devons modéliser l'effet de  $\mathcal{G}_X$  sur les observations. Pour modéliser les relations entre les variables du problème, nous devons calculer la distribution conjointe de probabilité de ces variables,  $p(\mathcal{G}_X, \mathcal{I}, \mathcal{B}, \mathcal{F}, \tau)$ . Nous proposons la décomposition suivante, basée sur les dépendances statistiques exprimées dans la figure 1 :

$$p(\mathcal{G}_X, \mathcal{I}, \mathcal{B}, \mathcal{F}, \tau) = p(\tau) p(\mathcal{B} | \tau) p(\mathcal{G}_X | \tau) p(\mathcal{F} | \mathcal{G}_X, \tau) p(\mathcal{I} | \mathcal{F}, \mathcal{B}, \tau)$$

- $p(\tau)$ ,  $p(\mathcal{B} | \tau)$  sont les distributions *a priori* de notre jeu de paramètre, et des paramètres de la régression des images du fond, respectivement. Comme nous n'avons aucune raison *a priori* de favoriser l'une ou l'autre des valeurs de ces paramètres, ou un aspect particulier des images du fond, nous affectons une distribution uniforme à ces termes. De ce fait ils disparaissent de toute inférence.
- $p(\mathcal{G}_X | \tau)$  est la vraisemblance *a priori* de l'occupation, qui pourrait varier en fonction de  $X$  par exemple. Celle-ci est indépendante de toute autre variable sauf  $\tau$ . Comme nous ne souhaitons pas favoriser une quelconque position de voxel dans l'espace, et que c'est surtout la régularisation de la grille induite par les observations image qui nous intéresse dans cet article, nous affectons aussi cette distribution à l'uniforme et l'ignorons donc par la suite.

- $p(\mathcal{F} | \mathcal{G}_X, \tau)$  est le terme de vraisemblance des silhouettes. Les dépendances considérées reflètent le fait que l’occupation des voxels explique la détection des silhouettes dans les images.
- $p(\mathcal{I} | \mathcal{F}, \mathcal{B}, \tau)$  est le terme de vraisemblance des images. Les dépendances considérées reflètent le fait que l’ensemble des observations dans les images n’est conditionné que par la détection de silhouette dans les images, et par la connaissance des images du fond.

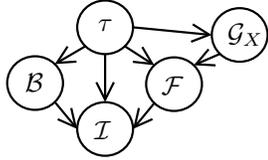


FIG. 1 – Les variables du système proposé, et leur graphe de dépendances.  $\tau$  : *a priori* et paramétrisation introduits dans le modèle.  $\mathcal{G}_X$  : occupation au voxel  $X$ .  $\mathcal{B}$  : carte des paramètres du modèle du fond.  $\mathcal{F}$  : carte de détection des silhouettes.  $\mathcal{I}$  : images observées.

### 3.1 Simplifications de la fusion de capteurs

Les couleurs des pixels des images d’entrée sont traitées comme des observations bruitées du modèle. Nous supposons, comme généralement dans un problème de fusion de capteurs, que le bruit est indépendamment et identiquement distribué. De plus, nous supposons que les observations de couleur obtenues en chaque pixel ne sont expliquées que par les images du fond ainsi que l’état de la détection de silhouette *en ce même pixel*. Ceci induit une indépendance conditionnelle entre toutes les observations de couleur  $\mathcal{I}_p^i$  :

$$p(\mathcal{I} | \mathcal{F}, \mathcal{B}, \tau) = \prod_{i,p} p(\mathcal{I}_p^i | \mathcal{F}_p^i, \mathcal{B}_p^i, \tau)$$

Nous exprimons que chaque variable de détection de silhouette ne dépend elle-même que de la connaissance que nous avons de l’état d’occupation de la grille, et qu’elle est indépendante de toute autre variable capteur. C’est également une hypothèse courante des problèmes de fusion de capteur : toutes les détections de silhouette sont conditionnellement indépendantes, sachant leur principale cause, à savoir l’occupation des voxels :

$$p(\mathcal{F} | \mathcal{G}_X, \tau) = \prod_{i,p} p(\mathcal{F}_p^i | \mathcal{G}_X, \tau)$$

En conséquence, la distribution conjointe des variables d’intérêt est réduite au produit suivant :

$$p(\mathcal{G}_X, \mathcal{I}, \mathcal{B}, \mathcal{F}, \tau) = \prod_{i,p} p(\mathcal{F}_p^i | \mathcal{G}_X, \tau) p(\mathcal{I}_p^i | \mathcal{F}_p^i, \mathcal{B}_p^i, \tau) \quad (1)$$

Nous avons donc réduit l’évaluation de la probabilité conjointe de toutes les images, de toutes les cartes de détection, et de notre occupation au voxel considéré, à deux

sous-problèmes beaucoup plus simples. D’une part, l’expression de la vraisemblance de la détection de silhouette en un seul pixel, sachant l’occupation de notre voxel. C’est le terme de formation des silhouettes (paragraphe 3.2). D’autre part, l’expression de la vraisemblance d’une observation couleur, sachant l’état de détection en ce pixel, et les paramètres du fond en ce pixel. Il s’agit du terme de formation des images (paragraphe 3.3). Penchons-nous maintenant sur ces deux termes.

### 3.2 Terme de formation des silhouettes

Le terme de formation des silhouettes  $p(\mathcal{F}_p^i | \mathcal{G}_X, \tau)$  modélise la réponse de la détection d’un pixel  $(i, p)$  à l’état d’occupation de notre voxel d’intérêt  $\mathcal{G}_X$ . Nous avons besoin d’introduire deux variables cachées locales  $\mathcal{S}$  et  $\mathcal{R}$  pour tempérer l’influence de ce voxel. La figure 2 présente les variables et dépendances statistiques de ce sous-problème. Dans un cas idéal et dénué de bruit, les deux variables  $\mathcal{F}_p^i$  et  $\mathcal{G}_X$  seraient auto-suffisantes, et liées par une relation de logique booléenne : si notre voxel  $X$  est occupé, et s’il se projette au point  $p$ , alors il y a une détection de silhouette au pixel  $p$ ,  $\mathcal{F}_p^i = 1$ . C’est la formulation implicite utilisée dans les méthodes classiques d’enveloppe visuelle.

Cependant, il existe des sources d’incertitude qui viennent perturber ce raisonnement intuitif. D’une part, le fait qu’un voxel se trouve sur la ligne de vue d’un pixel est lui-même incertain. Ceci peut-être dû à de nombreuses causes externes : les petites erreurs de calibration, la disparité des instants d’acquisition des images, qui introduisent des défauts d’alignement dans la scène. Il y a aussi des problèmes d’échantillonnage, car aucun voxel ne se projette parfaitement sur un pixel : sa surface de projection peut en couvrir plusieurs. D’autre part, des phénomènes autres que l’occupation du voxel lui-même peuvent expliquer la détection : une occupation due à un autre voxel que  $X$ , ou un changement d’apparence du fond (une cause *interne* de défaillance au vu du modèle capteur que nous définissons).

Il est possible de modéliser ces phénomènes en utilisant deux variables cachées booléennes  $\mathcal{S}$  et  $\mathcal{R}$ . Ceci nous conduit à deux expressions du terme de détection de silhouette  $p(\mathcal{F}_p^i | \mathcal{G}_X, \tau)$ . Tout d’abord, considérons le cas où nous savons que notre voxel  $X$  est occupé ( $\mathcal{G}_X = 1$ ) :

$$p(\mathcal{F}_p^i | [\mathcal{G}_X = 1], \tau) = p(\mathcal{S} = 0 | \tau) \mathcal{U}(\mathcal{F}_p^i) + p(\mathcal{S} = 1 | \tau) \mathcal{P}_d(\mathcal{F}_p^i) \quad (2)$$

Par définition, la *variable d’échantillonnage*  $\mathcal{S}$  est égale à 1 si le voxel  $X$  se trouve sur la ligne de vue du pixel  $(i, p)$ . Si ce n’est pas le cas ( $\mathcal{S} = 0$ ), alors la connaissance de l’occupation de notre voxel ne nous apporte aucune information sur la réponse du pixel. Ceci explique la distribution uniforme  $\mathcal{U}(\mathcal{F}_p^i)$  pour la détection de silhouette dans l’expression (2). Si le voxel se trouve sur la ligne de vue de  $p$  ( $\mathcal{S} = 1$ ), alors la détection à ce pixel est régie par la loi de probabilité  $\mathcal{P}_d(\mathcal{F}_p^i)$ . En pratique nous définissons cette distribution en utilisant une constante  $P_D \in [0, 1]$ , qui est un paramètre de notre système :  $\mathcal{P}_d([\mathcal{F}_p^i = 1]) = P_D$

est le taux de détection d'un pixel en tant que capteur, et réciproquement  $\mathcal{P}_d([\mathcal{F}_p^i = 0]) = 1 - P_D$  est son taux de défaut de détection. Une telle défaillance a lieu lorsque le pixel relate de manière erronée l'absence de matière sur sa ligne de vue. Ceci est fondamentalement utile pour notre problème : en effet il peut arriver que la détection de silhouette échoue localement dans une image. Le fait de modéliser explicitement cet état de fait au niveau de notre capteur donne la possibilité à notre système de corriger une telle erreur grâce à la contribution d'autres images.

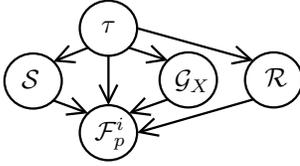


FIG. 2 – Variables et graphe de dépendances du sous-problème de détection de silhouette en un pixel.  $\tau$  : connaissance a priori.  $G_X$  : occupation de notre voxel.  $S$  : variable d'échantillonnage.  $\mathcal{R}$  : variable modélisant les causes externes de détection.  $\mathcal{F}_p^i$  : détection de silhouette au pixel  $(i, p)$ .

Considérons maintenant le cas où l'on sait que notre voxel n'est pas occupé ( $G_X = 0$ ) :

$$p(\mathcal{F}_p^i | [G_X=0], \tau) = p(S=0 | \tau) \mathcal{U}(\mathcal{F}_p^i) + p(S=1 | \tau) [ p(\mathcal{R}=1 | \tau) \mathcal{P}_d(\mathcal{F}_p^i) + p(\mathcal{R}=0 | \tau) \mathcal{P}_f(\mathcal{F}_p^i) ] \quad (3)$$

Comme dans le cas précédent, lorsque le voxel ne se trouve pas sur la ligne de vue de  $p$  (cas où  $S = 0$ ), on ne peut conclure sur la détection en ce pixel. Cependant même si le voxel se trouve sur la ligne de vue de ce pixel (cas où  $S = 1$ ), il n'est toujours pas possible de conclure sur l'état de la détection. Soit une variable de "causes extérieures"  $\mathcal{R}$ , valant 1 si un autre objet se trouve sur la même ligne de vue que notre voxel. Dans un tel cas, la détection est de nouveau régie par la loi de probabilité  $\mathcal{P}_d(\mathcal{F}_p^i)$ . Cependant, dans le cas où aucun autre objet n'obstrue cette ligne de vue ( $\mathcal{R} = 0$ ), la loi de probabilité de détection est régie par la distribution  $\mathcal{P}_f(\mathcal{F}_p^i)$ . Nous établissons cette distribution à l'aide d'une constante  $P_{FA} \in [0, 1]$ , qui est un paramètre de notre système :  $\mathcal{P}_f([\mathcal{F}_p^i = 1]) = P_{FA}$  est le taux de fausse alarme d'un pixel. C'est le taux avec lequel le pixel relate faussement la présence de matière le long de sa ligne de vue, lorsqu'elle est en fait absente.  $\mathcal{P}_f([\mathcal{F}_p^i = 0]) = 1 - P_{FA}$  est le taux de non-détections correctes et avérées. Nous devons donner une forme paramétrique à  $p(\mathcal{R} | \tau)$ . Il peut y avoir des causes de détection partout sur la ligne de vue de  $p$ . Nous ne faisons aucune supposition sur la nature de ces causes et considérons que la détection peut être provoquée aussi bien par l'occupation d'autres voxels de la grille ou par ces causes. Ceci nous conduit à affecter une distribution uniforme à ce terme. Ce faisant, nous considérons que le plus important n'est pas forcément

de donner une modélisation très élaborée à ces termes, mais que c'est le fait de laisser à l'inférence la possibilité de prendre en compte la présence de ces phénomènes extérieurs qui est déterminant en soi.

**Forme paramétrique pour le terme d'échantillonnage**  $p(S | \tau)$ . Ce terme dépend de  $i$ ,  $p$  et  $X$ . Nous utilisons un échantillonnage uniforme, avec  $p(S | \tau) = \mathcal{U}_{k \times k}(x - p)$ . Ceci donne un poids équivalent à tout les voxels se trouvant dans une fenêtre de taille  $k \times k$  autour du pixel  $p$ . Une fonction d'échantillonnage plus lisse (de forme Gaussienne) aurait aussi pu être utilisée, mais mobilise des ressources de calcul plus importantes pour intégrer l'information. De manière générale, la forme de cette fonction peut aisément être adaptée pour répondre à des besoins spécifiques.

La fonction d'échantillonnage permet d'avoir un contrôle sur les petites erreurs de calibration, de synchronisation entre les caméras, et sur les erreurs de classification dans les images : en effet elle permet à plusieurs pixels de participer à la classification du même voxel au cours de l'inférence. Grâce à l'introduction de ces deux variables cachées et de leur forme paramétriques, notre méthode unifie la gestion de l'incertitude dans le cadre des silhouettes et les schémas d'échantillonnage classiques utilisés dans certaines méthodes à base d'enveloppe visuelle [3].

### 3.3 Terme de formation des images

Le but du terme de formation des images  $p(\mathcal{I}_p^i | \mathcal{F}_p^i, \mathcal{B}_p^i, \tau)$  est de fournir un modèle qui explique l'information de couleur au pixel  $(i, p)$ , sachant les paramètres du modèle statistique du fond à ce pixel, et selon l'état de la détection à ce pixel. Nous exposons ici les formes paramétriques que nous choisissons de donner à ces termes.

Considérons tout d'abord le cas où une détection de silhouette a eu lieu au pixel  $(i, p)$ . La connaissance du fond ne nous apporte aucune information supplémentaire sur la couleur que l'on peut espérer observer à ce pixel. Nous savons en effet que les objets du fond sont occultés par un objet d'intérêt, dont le pixel observe la couleur. Nous ne faisons dans cet article aucune supposition sur la couleur des objets d'intérêt, ce qui nous conduit à donner une loi uniforme à la distribution des couleurs observées ici :

$$p(\mathcal{I}_p^i | [\mathcal{F}_p^i = 1], \mathcal{B}_p^i, \tau) = \mathcal{U}(\mathcal{I}_p^i)$$

Le second cas que nous devons considérer est celui où aucune silhouette n'a été détectée en ce pixel. Intuitivement, si l'on sait que le capteur ne voit pas de matière sur sa ligne de vue, alors la couleur observée en ce pixel doit ressembler à celle du fond. En pratique, nous choisissons de résumer toutes les images du fond dont nous disposons en estimant les paramètres  $\mathcal{B}_p^i = (\mu_p^i, \sigma_p^i)$  d'une distribution Normale dans l'espace de couleur (Y,U,V), pour chaque pixel. Nous pouvons alors formuler pour le pixel  $(i, p)$  le raisonnement exprimé ci-dessus de la manière suivante, à l'aide de cette distribution :

$$p(\mathcal{I}_p^i | [\mathcal{F}_p^i = 0], [\mathcal{B}_p^i = (\mu_p^i, \sigma_p^i)], \tau) = \mathcal{N}(\mathcal{I}_p^i | \mu_p^i, \sigma_p^i)$$

Cette représentation des couleurs du fond dérive des méthodes classiques de soustraction de fond [20]. Il est important de souligner cependant que l'outil présenté ici est indépendant du modèle statistique choisi pour représenter le fond. Il pourrait facilement utiliser d'autres représentations du fond, comme le mélange de Gaussiennes [18], qui est plus robuste aux ambiguïtés sous-pixelles et les variations périodiques de couleur dans les images du fond. Néanmoins, certains problèmes persistent quel que soit le modèle de couleur utilisé : les ambiguïtés de couleur entre les objets d'intérêt et ceux du fond, les changements dans la géométrie ou l'illumination de la scène. C'est le but de notre approche multi-caméra de compenser et d'être plus robuste à de tels problèmes, difficilement corrigibles dans un contexte monoculaire.

## 4 Inférence sur l'occupation d'un voxel

Maintenant que la distribution conjointe est complètement déterminée, il est possible d'utiliser la règle de Bayes pour inférer la distribution de probabilité de notre variable recherchée  $\mathcal{G}_X$ , sachant la valeur de nos variables connues  $\mathcal{I}, \mathcal{B}, \tau$ , et en marginalisant l'inférence par rapport à nos variables inconnues  $\mathcal{F}$  :

$$p(\mathcal{G}_X | \mathcal{I}, \mathcal{B}, \tau) = \frac{\sum_{\mathcal{F}} p(\mathcal{G}_X, \mathcal{I}, \mathcal{B}, \mathcal{F}, \tau)}{\sum_{\mathcal{G}_X, \mathcal{F}} p(\mathcal{G}_X, \mathcal{I}, \mathcal{B}, \mathcal{F}, \tau)}$$

$$= \frac{\sum_{\mathcal{F}} \prod_{i,p} p(\mathcal{F}_p^i | \mathcal{G}_X, \tau) p(\mathcal{I}_p^i | \mathcal{F}_p^i, \mathcal{B}_p^i, \tau)}{\sum_{\mathcal{G}_X, \mathcal{F}} \prod_{i,p} p(\mathcal{F}_p^i | \mathcal{G}_X, \tau) p(\mathcal{I}_p^i | \mathcal{F}_p^i, \mathcal{B}_p^i, \tau)} \quad (4)$$

$$= \frac{\prod_{i,p} \sum_{\mathcal{F}_p^i} p(\mathcal{F}_p^i | \mathcal{G}_X, \tau) p(\mathcal{I}_p^i | \mathcal{F}_p^i, \mathcal{B}_p^i, \tau)}{\sum_{\mathcal{G}_X} \prod_{i,p} \sum_{\mathcal{F}_p^i} p(\mathcal{F}_p^i | \mathcal{G}_X, \tau) p(\mathcal{I}_p^i | \mathcal{F}_p^i, \mathcal{B}_p^i, \tau)} \quad (5)$$

où (1) a été substituée dans (4). Cette inférence est simplifiée du fait que toutes les sommes sur les variables  $\mathcal{F}_p^i$  peuvent être réalisées au niveau de chaque pixel et donc factorisées (5). En particulier,  $\mathcal{F}_p^i$  peut elle-même être considérée comme une variable cachée intervenant au niveau du pixel. On peut dans ce cadre considérer que le modèle capteur se résume à un seul terme générique de formation des couleurs dans les images  $p(\mathcal{I}_p^i | \mathcal{G}_X, \mathcal{B}_p^i, \tau)$ . Ce terme lie directement la couleur observée en chaque pixel à l'état d'occupation de notre voxel. Il peut être exprimé à l'aide des termes de formation des silhouettes et des images présentés ci-dessus. Il suffit alors de marginaliser la détection de silhouette  $\mathcal{F}_p^i$ , comme écrit ci-dessous :

$$p(\mathcal{I}_p^i | \mathcal{G}_X, \mathcal{B}_p^i, \tau) = \sum_{\mathcal{F}_p^i} p(\mathcal{F}_p^i | \mathcal{G}_X, \tau) p(\mathcal{I}_p^i | \mathcal{F}_p^i, \mathcal{B}_p^i, \tau)$$

Cette vue du modèle capteur clarifie l'inférence (5), car elle montre comment chaque pixel contribue à la distribution de probabilité d'occupation de notre voxel :

$$p(\mathcal{G}_X | \mathcal{I}, \mathcal{B}, \tau) = \frac{\prod_{i,p} p(\mathcal{I}_p^i | \mathcal{G}_X, \mathcal{B}_p^i, \tau)}{\sum_{\mathcal{G}_X} \prod_{i,p} p(\mathcal{I}_p^i | \mathcal{G}_X, \mathcal{B}_p^i, \tau)} \quad (6)$$

Notons que l'expression finale de l'inférence (6) lie, dans son écriture, l'occupation d'un voxel à *tous* les pixels et leur observations. Un tel calcul est bien sûr inenvisageable, sachant qu'une telle inférence doit être réalisée *pour chaque voxel* de la grille. En pratique, les termes d'échantillonnage présentés au paragraphe 3.2 permettent de borner la région d'influence d'un pixel dans les images. Les expressions des probabilités de détection des pixels trop éloignés de la projection d'un voxel dégènerent en un terme uniforme, ce que les équations (2) et (3) expriment. Dans ce cas le contribution peut être factorisée et éliminée de l'expression d'inférence (6). Ceci rend l'inférence praticable, en calculant ce produit sur une fenêtre locale de pixels, centrée autour du point de projection de  $X$ , dans chaque image. Notons que le calcul d'un tel produit est rapidement limité par la capacité de représentation des nombres de la machine : en pratique nous calculons ce produit en utilisant une somme de log probabilités. Si  $k$  est la taille de la fenêtre, et  $N$  le nombre de voxels par dimension de l'espace, alors la complexité en calcul de l'inférence de tous les voxels de la grille est  $O(n k^2 N^3)$ .

## 5 Résultats et applications

Nous avons implémenté cet algorithme de fusion d'information silhouette multi-vue, en utilisant une stratégie d'échantillonnage uniforme pour les voxels dans nos expérimentations. Comparé à un échantillonnage Gaussien ceci s'est avéré être un bon compromis entre coût de calcul et capacité d'intégration de l'information de plusieurs pixels. Notons que la méthode dans son ensemble ne possède que trois paramètres  $\{P_D, P_{FA}, k\}$ , respectivement le taux de détection et de fausse alarme, et la taille de la fenêtre d'échantillonnage. Souvent ces paramètres peuvent être fixés une fois pour toute pour une application donnée. En pratique,  $P_D$  et  $P_{FA}$  pondèrent la confiance accordée aux observations. Si  $P_{FA} = 0$  et  $P_D = 1$ , nous accordons à ces observations une confiance aveugle. Si  $P_{FA}$  et  $P_D$  sont proches de 0.5 alors nos observations ne sont pas fiables : on a besoin de beaucoup plus d'observations pour conclure sur l'état d'occupation d'un voxel.  $k$  permet de décider du nombre d'observations à considérer localement dans chaque image et permet donc aussi un contrôle de la décision et de la robustesse de la méthode aux petites erreurs. Nous avons testé cet algorithme dans des conditions diverses. Il peut en effet être utilisé dans de nombreux champs d'application.

### 5.1 Modélisation à partir d'images

La grille est elle-même une estimation de forme. Nous illustrons ce fait dans la séquence dite de marche. Il s'agit de l'acquisition d'une scène où une personne est en train de marcher en rond, réalisée avec 8 caméras de caractéristiques et résolutions différentes (640x480, 780x580) avec une fréquence d'acquisition de 15Hz. Comme l'illustre la figure 3 l'information silhouette accessible par soustraction

de fond monoculaire est bruitée et comporte des erreurs. Notons aussi que certaines caméras ne voient pas toujours entièrement la personne au cours de la séquence. Ceci est le cas dans la deuxième vue de la figure, où l'avant bras de la personne est coupé. Le modèle de couleur utilisé pour les soustractions de fond monoculaires est le même que celui utilisé dans notre modèle (distribution normale). Ces soustractions de fond résument l'information silhouette dont notre algorithme dispose dans chaque image.

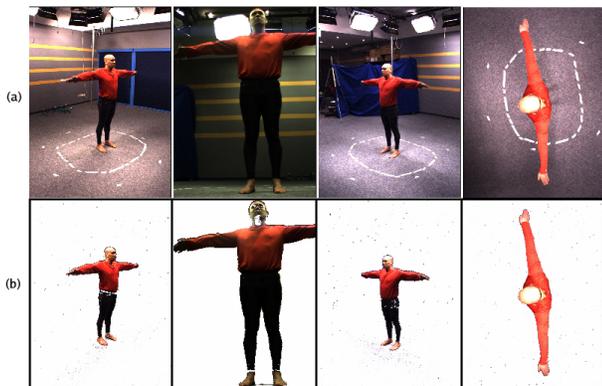


FIG. 3 – Entrées de l'algorithme. (a) Quatre des huit images de la séquence de marche (8 caméras, 15 images/sec) (b) Le résultat d'une soustraction de fond monoculaire en utilisant le même modèle de représentation des couleurs que dans notre algorithme (rendu semi-transparent pondéré par la probabilité des silhouettes). Notons les imperfections de ces silhouettes : la deuxième caméra ne voit pas l'avant bras gauche de la personne. Il y a des trous dans les silhouettes dans plusieurs vues.

La figure 4 montre le résultat de notre méthode à un instant de la séquence de marche, en utilisant une grille de  $120^3$  voxels. La figure comporte des coupes verticales et horizontales de la grille, qui montrent la nature de l'information disponible dans celle-ci. Une vue plus dynamique est fournie dans la vidéo adjointe à l'article<sup>1</sup>. Comme le montre la figure 4(c), la méthode fournit de bons résultats de modélisation, si l'on extrait une isosurface de la grille de probabilités. La méthode donne une surface finement détaillée, avec de l'information sous-voxellique. Les petits trous qui apparaissaient dans les soustractions de fond monoculaires sont comblés.

L'approche voxellique classique avec silhouettes binaires a été implémentée pour comparaison, et les résultats sont montrés en figure 4(d). Chaque voxel  $y$  est projeté dans les images et sculpté si cette projection est en dehors des silhouettes. Nous utilisons la soustraction de fond de la figure 3 pour cette expérience, et sélectionnons manuellement le meilleur seuil *dans chaque image* pour fournir les silhouettes binaires nécessaires à cet algorithme. Les trous qui apparaissent dans les silhouettes génèrent en général des trous et défauts dans le volume reconstruit. Notons que notre méthode retrouve une information d'occupation valide avec des vues qui ne voient pas entièrement l'objet.

<sup>1</sup><http://perception.inrialpes.fr/Publications/2005/FB05/SilhouetteCueFusion.avi>

Ceci est transparent pour notre algorithme, car celui-ci n'intègre que l'information des capteurs qui voient les voxels. Ceci est différent de toute approche classique à base d'enveloppe visuelle, aussi bien surfacique que volumétrique. En effet, celles-ci font toutes des hypothèses explicites qui forcent la décision pour les voxels se trouvant en dehors de la zone de visibilité d'une caméra (voir figure 4(d)).

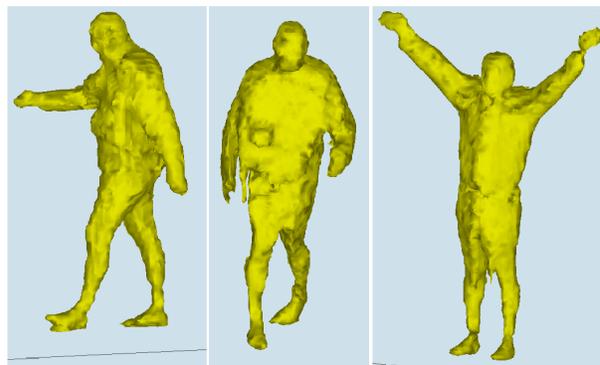


FIG. 5 – Isosurface de probabilité 0.80 à des instants différents de la séquence de marche.

## 5.2 Soustraction de fond multi-vue

Notre méthode calcule une fusion d'informations silhouette. Cette fusion peut être utilisée pour calculer des silhouettes cohérentes dans les images. En effet, il est possible de reprojeter et effectuer un rendu de la grille d'occupation, en projetant le maximum d'intensité de la grille (voir figure 6). Cette heuristique est une approximation du calcul d'inférence de l'état de détection de la silhouette d'un pixel sachant toutes les observations images, calcul possible mais pratiquement infaisable au vu de sa complexité. La projection du maximum d'intensité nous permet de localiser, à un coût moindre, les zones où les silhouettes sont plus probables dans les images.

Cette heuristique de rendu définit une procédure de soustraction de fond multi-caméra. En particulier, un seul seuil peut-être choisi simultanément pour toutes les images pour trouver les silhouettes binaires optimales, comme le montre la figure. Les détails fins de la silhouette sont préservés, et sont seulement limités par la résolution de la grille. Chaque vue bénéficie alors de la connaissance des informations silhouette des autres vues.

## 5.3 Détection d'objet

La méthode peut être utilisée dans des conditions beaucoup plus dures pour inférer de l'information sur une scène. En particulier, en présence de hauts niveaux de bruit, la taille de la fenêtre d'échantillonnage peut être augmentée pour une robustesse accrue, avec cependant un impact négatif sur la précision (cette opération tend en effet à dilater le

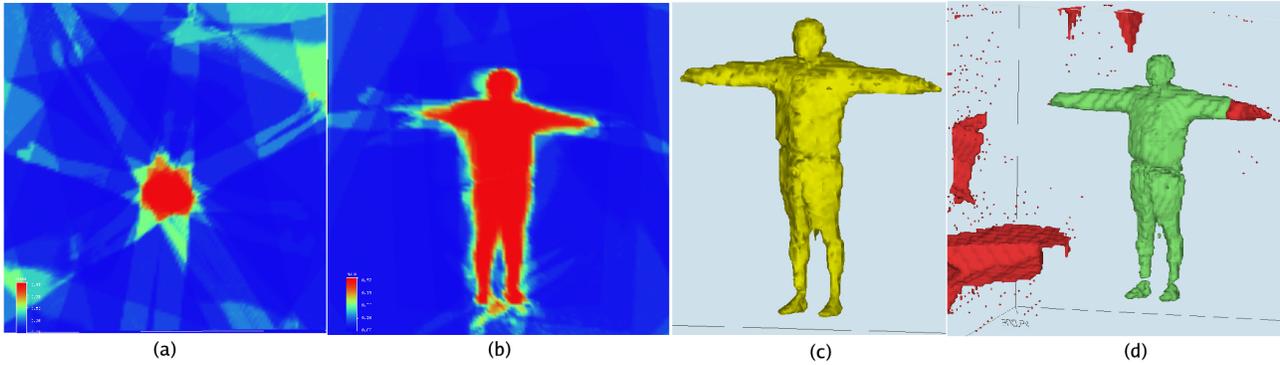


FIG. 4 – Un instant de la séquence de marche. Acquisition 15img/sec, 8 caméras, utilisant en entrée les données explicitées en figure 3. Grille de  $120^3$  voxels. Temps de calcul : approximativement 10 sec sur un PC 2.4 GHz. Paramètres utilisés :  $P_D = 0.9$ ,  $P_{FA} = 0.1$ ,  $k = 5$  (a) Coupe horizontale (au niveau de l’abdomen) dans la grille de probabilité. Les régions vertes au coin supérieur droit ne sont vues par aucune caméra (probabilité 0.5). (b) Coupe verticale de la grille. (c) Isosurface de probabilité 0.80, obtenue de la grille. Le bruit dû aux ombres ne perturbe pas l’estimation, la majorité des défauts des silhouettes sont comblés. (d) Deux approches classiques de reconstruction à base d’enveloppe visuelle : en vert, sous l’hypothèse que toutes les caméras voient entièrement l’objet. L’avant-bras gauche est perdu. En rouge, sous l’hypothèse que les voxels en dehors du domaine de visibilité d’une caméra peuvent faire partie de l’enveloppe visuelle. Le bras n’est cette fois-ci pas coupé, mais des objets fantômes apparaissent, résultant d’ambiguïtés visuelles à des endroits où toutes les caméras ne voient pas les voxels.

volume dans la grille, et par conséquent les isosurfaces que l’on peut en extraire). Lorsqu’il y a beaucoup de bruit dans les images, l’utilisation de la méthode peut devenir limitée pour les applications de modélisation 3D. Cependant la méthode peut encore être utilisée pour *localiser* les objets dans une scène, sans nécessairement avoir pour but de reconstruire précisément leur surface. Nous illustrons l’utilisation potentielle de notre méthode dans le cadre d’une application de localisation, avec des caméras non précisément réglées, et des images avec un contraste pauvre, dans la figure 7. Dans cette expérience, 8 caméras ont été placées dans une scène de manière à couvrir une zone relativement étendue ( $25m^2$ ) dans une pièce. Seul le centre de la pièce est vu par une majorité de caméras. Les zones périphériques de l’espace d’acquisition ne sont vues que par 3 ou 4 caméras tout au plus. Deux personnes marchent aléatoirement dans la pièce et sont localisées avec succès, lorsqu’elles sont vues par au moins 3 caméras. C’est un résultat empirique intéressant : deux caméras ne semblent pas être suffisantes pour apporter l’information nécessaire à la décision, et distinguer les vrais objets d’objets “fantômes” qui apparaissent dans les zones d’ambiguïté visuelle. Ces régions d’ambiguïté sont des régions vides de l’espace mal détectées par les caméras, qui sont en général dans l’ombre d’un véritable objet, par rapport au point de vue d’une ou plusieurs caméras.

## 6 Discussion

Nous avons présenté une nouvelle approche pour la fusion d’information silhouette provenant de plusieurs vues. Nous utilisons une approche de fusion de capteurs rigoureuse, pour lier directement l’information de la scène aux observations. Ceci présente plusieurs avantages : la chaîne en-

tière, des causes aux observations, est modélisée. Toutes les hypothèses faites sont rendues explicites. La méthode évite aussi par conséquent de prendre des décisions dures sur la segmentation des silhouettes, ce qui aurait requis l’ajustement manuel de paramètres pour chaque image. Ainsi, toute l’information silhouette disponible peut être intégrée, en utilisant seulement trois paramètres globaux du modèle capteur. Ces paramètres contrôlent intuitivement la fiabilité que l’on accorde aux observations. Cette approche a été validée dans le cadre de diverses applications, et de nombreuses nouvelles idées peuvent être expérimentées et rajoutées sans changer le coeur de la méthode.

Nous pourrions, en l’occurrence, prendre en compte plus de dépendances statistiques dans le modèle. Notamment, nous avons remarqué que la fiabilité des observations d’un pixel peut être liée à la couleur observée à ce pixel. Par exemple nous observons de nombreuses fois le cas où un objet d’intérêt noir apparaît devant un fond noir, qui génère spécifiquement des erreurs. Ceci pourrait être l’objet d’un traitement dédié et sera examiné. Plus généralement, notre modèle estime des grilles statiques, à un seul instant donné. Il serait très intéressant de prendre en compte la cohérence temporelle de la séquence, où les observations du passé sont utilisées pour inférer l’état d’occupation des voxels actuels. Les grilles d’occupation sont particulièrement bien adaptées à ce problème. C’est d’ailleurs leur principale utilisation dans la communauté de robotique [4]. Nous explorerons ces possibilités pour étendre les capacités de notre système.

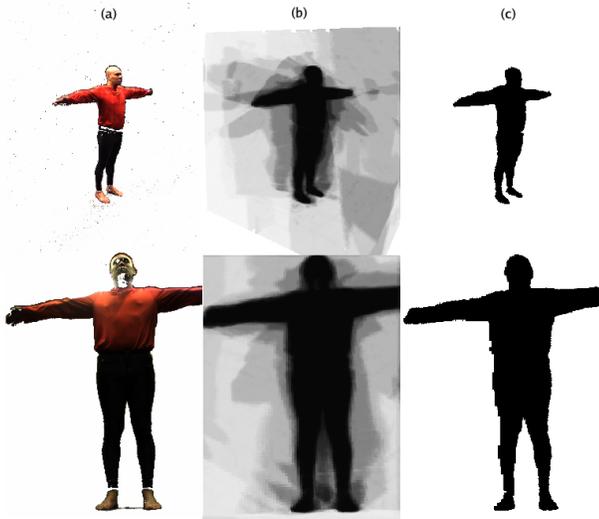


FIG. 6 – Deux exemples de rendu de silhouettes multi-vue. (a) Soustraction de fond monoculaire (rendu semi-transparent pondéré par la probabilité). On peut remarquer les imperfections : coupure de la silhouette au niveau du bassin dans l’image du dessus ; pieds séparés du corps, et erreur grossière au niveau du visage, dans l’image du dessous. Aucun algorithme de réparation d’une seule image (morphomat) ne peut corriger de tels erreurs et fournir une précision satisfaisante pour les silhouettes. (b) Projection du maximum d’intensité de la grille d’occupation ( $120^3$ ) depuis les points de vues de l’acquisition. (c) Les mêmes données après sélection manuelle d’un seuil pour toutes les silhouettes simultanément : les silhouettes sont améliorées par rapport aux silhouettes monoculaires initiales. Des artefacts d’échantillonnage apparaissent selon la résolution de la grille et la configuration de la scène.

## Références

- [1] J. S. D. Bonet and P. A. Viola. Roxels : Responsibility weighted 3d volume reconstruction. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, (Greece)*, volume I, pages 418–425, Sept. 1999.
- [2] A. Broadhurst, T. Drummond, and R. Cipolla. A probabilistic framework for the Space Carving algorithm. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, (Canada)*, volume I, pages 388–393, 2001.
- [3] G. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, (USA)*, volume II, pages 714 – 720, June 2000.
- [4] C. Coue. *Modèle bayésien pour l’analyse multimodale d’environnements dynamiques et encombrés : application à l’assistance à la conduite automobile en milieu urbain*. PhD thesis, Institut National Polytechnique de Grenoble, Dec. 2003.
- [5] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer, Special Issue on Autonomous Intelligent Machines*, 22(6) :46–57, June 1989.
- [6] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *Proceedings, 5th European*

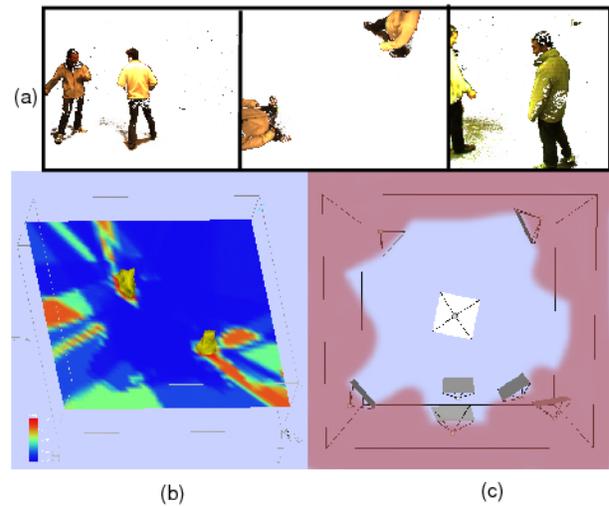


FIG. 7 – Séquence multi-objet, avec 8 caméras. (a) Soustractions de fond monoculaires de quelques vues d’entrée (rendu semi-transparent pondéré par la probabilité d’appartenir à une silhouette). Des conditions difficiles de lumière et contraste créent de grandes difficultés pour de telles soustractions monoculaires. (b) Notre méthode, utilisée pour reconstruire une grille grossière de la scène ( $50 \times 50 \times 18$ ), suffisante pour localiser des objets (avec  $k = 25$ ). Temps de calcul pour un instant de la séquence : 7s. Une coupe horizontale de la grille de probabilités est donnée, ainsi que des isosurfaces de probabilité 0.67 qui montrent les objets localisés. (c) Configuration des caméras dans cette scène. Les régions en rouge sombre sont moins fiables car elles ne sont vues que par un maximum de 2 caméras. Le système est capable de détecter la présence d’objets dans une zone de 5m x 5m.

- Conference on Computer Vision, Freiburg, (Germany)*, volume I of *Lecture Notes in Computer Science*, pages 379–393. Springer, June 1998.
- [7] J.-S. Franco and E. Boyer. Exact Polyhedral Visual Hulls. In *Proceedings of the British Machine Vision Conference, Norwich (UK)*, pages 329–338, Sept. 2003.
- [8] B. Goldlücke and M. Magnor. Joint 3-d reconstruction and background separation in multiple views using graph cuts. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Madison, (USA)*, volume I, pages 683–694, June 2003.
- [9] K. Grauman, G. Shakhnarovich, and T. Darrell. A bayesian approach to image-based visual hull reconstruction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Madison, (USA)*, volume I, pages 187–194, June 2003.
- [10] J. Isidoro and S. Sclaroff. Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints. In *Proceedings of the 9th International Conference on Computer Vision, Nice, (France)*, pages 1335–1342, 2003.
- [11] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *Proceedings, 7th European Conference on Computer Vision, Copenhagen, (Denmark)*, pages 82–96, 2002.

- [12] K. Kutulakos and S. Seitz. A Theory of Shape by Space Carving. *International Journal of Computer Vision*, 38(3) :199–218, 2000.
- [13] A. Laurentini. The Visual Hull Concept for Silhouette-Based Image Understanding. *IEEE Transactions on PAMI*, 16(2) :150–162, Feb. 1994.
- [14] S. Lazebnik, E. Boyer, and J. Ponce. On How to Compute Exact Visual Hulls of Object Bounded by Smooth Surfaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Kauai, (USA)*, volume I, pages 156–161, December 2001.
- [15] D. Margaritis and S. Thrun. Learning to locate an object in 3d space from a sequence of camera images. In *International Conference on Machine Learning*, pages 332–340, 1998.
- [16] W. Matusik, C. Buehler, and L. McMillan. Polyhedral Visual Hulls for Real-Time Rendering. In *Eurographics Workshop on Rendering*, 2001.
- [17] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image Based Visual Hulls. In *ACM Computer Graphics (Proceedings Siggraph)*, pages 369–374, 2000.
- [18] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, (USA)*, volume II, pages 246–252, June 1999.
- [19] R. Szeliski. Rapid Octree Construction from Image Sequences. *Computer Vision, Graphics and Image Processing*, 58(1) :23–32, 1993.
- [20] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfunder : Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) :780–785, 1997.
- [21] G. Zeng and L. Quan. Silhouette extraction from multiple images of an unknown background. In *Proceedings of the 6th Asian Conference on Computer Vision, Jeju Island, (Korea)*, Jan. 2004.