Capture de Mouvements Humains par Fusion de Multiples Données Squelettes

Jean-Thomas Masse, Frédéric Lerasle, Michel Devy, André Monin, Olivier Lefebvre, Stéphane Mas

Capture de Mouvements Humains par Fusion de Multiples Données Squelettes

Jean-Thomas Masse[1,2] Frédéric Lerasle[1,3] Michel Devy[1] André Monin[1] Olivier Lefebvre[2] Stéphane Mas[2]

[1] CNRS, Laboratoire d'Analyse et d'Architecture des Systèmes
7 avenue du colonel Roche, F-31400 Toulouse, France.
[2] Magellium SAS,
F-31520 Ramonville Saint-Agne, France.
[3] Univ de Toulouse, UPS, LAAS,
F-31400 Toulouse, France.

{jean-thomas.masse, frederic.lerasle, michel.devy, andre.monin}@laas.fr
{olivier.lefebvre, stephane.mas}@magellium.fr

Résumé
L'arrivée simultanée de capteurs de profondeur et couleur, et d'algorithmes de détection de squelettes super-temps-réel a conduit à un regain de la recherche sur la capture de mouvements humains. Cette fonctionnalité constitue un point clé de la communication Homme-Machine. Mais le contexte d'application de ces dernières avancées est l'interaction volontaire avec feedback, ce qui permet certaines approximations et requiert un positionnement spécifique des capteurs. Dans cet article, nous présentons une approche multi-capteurs, conçue pour améliorer la robustesse et la précision du positionnement des articulations de l'homme, et fondée sur un processur de lissage trajectoriel par logique différée, et le filtrage, des squelettes détectés par chaque capteur.

Mots Clef
Reconnaissance de mouvement, Assimilation de Données, Programmation Dynamique, Filtrage Bayésien, Capteur de profondeur

Abstract

Joint advent of affordable color and depth sensors and super-realtime skeleton detection, has produced a surge of research on Human Motion Capture. They provide a very important key to communication between Man and Machine. But the design was willing and closed-loop interaction, which allowed approximations and mandates a particular sensor setup. In this paper, we present a multiple sensor-based approach, designed to augment the robustness and precision of human joint positioning, based on delayed logic and filtering, of skeleton detected on each sensor.

Keywords
Motion Recognition, Data Assimilation, Dynamic Programming, Bayesian Filtering, Depth Sensor.

1 Introduction
Les recherches se portent de plus en plus sur la Capture de Mouvements Humains (abrégée CMH, ou MoCap, en général). Presque tous les fabricants majeurs de systèmes vidéo-ludiques ont introduit un périphérique permettant une CMH limitée. L'un d'eux, Microsoft, via son capteur RGB-D, la Kinect, en plus d'un succès commercial [1], a produit une alternative fiable et peu onéreuse aux caméras temps de vol et aux bancs stéréo dépendants de la texture. L'appareil repose sur la technologie brevetée de lumière structurée de PrimeSense [2]. La même puce équipe aussi le capteur équivalent Asus Xtion Pro Live [3], utilisé dans ce papier.

La CMH a aussi un intérêt pour la robotique et la vidéo-protection, où elle permet une interprétation des intentions et une meilleure communication. Ce principe présente un intérêt pour de nombreux autres domaines, par exemple l'éducation, la sociologie et la santé.
Plusieurs SDKs existent, i.e. le Microsoft Kinect SDK, propre à la Kinect, et l'environnement open-source OpenNI [4] de PrimeSense. Cette dernière a aussi développée le Middleware NiTE [5] pour OpenNI. C'est lui qui produit la segmentation de la silhouette utilisateur et la squelettisation. Les algorithmes détectent les parties corporelles dans les images de profondeur préalablement segmentées, en s'appuyant sur des techniques de Machine Learning. Un tel système ne nécessitant qu'une image de profondeur, même produite par une stéréo-caméra, caméra temps de vol ou une Kinect, nous appellerons *senseur* les Xtions que nous avons utilisées pour nos acquisitions.
Cependant, une telle approche est mono-senseur, et donc sensible à l'auto-occlusion et aux points de vue. La littérature s'est principalement focalisée sur la précision du capteur de profondeur [6] [7], ou réinventer une Motion Capture à partir des données senseur brutes [8]. Peu de travaux traitent du problème de MoCap par fusion multi-squelette, et de la précision de celle-ci comparée à un système commercial. Nos objectifs sont ici : (1) implémenter un procédé exploitant des techniques monoculaires existantes mais en multipliant les points de vue, et (2) utiliser un système de Capture de Mouvement industriel pour obtenir une vérité terrain afin de mesurer le gain de la multiplication des données.
L'article est structuré comme suit : d'abord, nous nous positionnons dans la littérature existante. Nous détaillons ensuite la formalisation de notre approche dans la section 3 . Avant la section des résultats, nous décrivons la plate-forme multi-capteurs et les données acquises. Enfin, nous listons nos conclusions et perspectives.

2 Travaux antérieurs
En supposant que NiTE/OpenNI soit similaire à la solution de la Microsoft Research Team [9], [10], il repose sur l'algorithme des Random Forests. Il est très rapide et relativement précis. Bien qu'il essaie de gérer l'occultation de certains membres, il est limité par son unique point de vue. De plus, le filtrage spatiotemporel de la reconstruction de posture est basique, sans prédiction.
La communauté Vision par Ordinateur a largement exploré cette problématique, comme l'atteste le survey de Moeslund et al. [11]. La résistance à l'occultation commence à trois ou quatre caméras [12]. De telles configurations requièrent une attention supplémentaire pour maintenir la cohérence temporelle et spatiale. En se basant sur les indices de couleur et de silhouette [13],

la Capture de Mouvement par caméras est possible. Cependant, elles présupposent la présence de scènes texturées, contrairement aux senseurs actifs [14].

Le corps humain étant une réalité physique, il est logique de s'appuyer sur un raisonnement temporel et de suivre le mouvement dans le temps à l'instar de [15]. Dans le suivi de piéton, les approches « tracking-by-detection » sont souvent combinées avec de la logique différée [16], [17], [18]. À notre connaissance, un tel principe n'a encore jamais été appliqué à la CMH à base de capteur de profondeur, où des approches purement basées détection [19] sont pour le moment plébiscitées. De plus, alors que de la CMH utilisant plusieurs capteurs de profondeur existe [8], elle se focalise sur la donnée brute au lieu d'utiliser des reconstructions de squelettes. Ainsi, il nous a semblé opportun de transposer ces approches de suivi par détection à notre problématique de suivi ici de postures.

Dans tous les cas, la vérité terrain constitue une base nécessaire pour mesurer une progression par rapport à l'existant. HumanEva [20] est une base publique issue d'un système multioculaires passif. Il existe peu de bases publiques RGB-D dédiées à la reconstruction de postures humaines.

Comparée à notre propre vérité terrain, notre stratégie multi-capteurs tire avantage des deux : la *temporalité* est exploitée par des techniques de filtrage, et est mise au même niveau que les données senseurs brutes dans un *lisseur* basé sur la logique différée, qui assure que la trajectoire est consistante avec les observations et les contraintes physiques.

3 Description de notre approche

Dans cette section, nous décrivons la formalisation sous-jacente à notre approche, et ensuite détailler chaque étape. Le schéma bloc de notre approche est décrite en sous-section 3.3 .

3.1 Formalisation

Comme dans de nombreux logiciel de CMH, le but est d'estimer $X = [X^j]_{j\in J}$, la position euclidienne d'un nombre réduit mais représentatif d'un ensemble J d'articulations du squelette humain. Ces articulations sont : mains, coudes, épaules, pieds, genoux, hanches, torse, cou et tête. A chaque pas de temps t, chaque senseur $k \in K$ produit une bitmap de segmentation du premier plan S_t^k par rapport à la bitmap de profondeur, et détecte une mesure du squelette $Y_t^k = \left[Y_t^{j,k}\right]_{j\in J}$. De plus, nous choisissons de ne pas nous limiter aux mesures produites par NiTE, donc nous considérons à la place un sur-ensemble $L \supset K$ d'hypothèses. Pour tous les ensembles tels que J, K et L, nous écrirons K au lieu de $|K|$ ou $card(K)$ pour alléger les notations.

Chaque mesure est de la forme $Y_t^l = X_t + V_t^l, \forall l \in L$, où V_t^l est l'erreur de reconstruction de squelette. Cette erreur n'est certainement ni bruit blanc ni gaussien.

Pour des raisons de coût CPU, nous choisissons de limiter la recherche à ces L hypothèses, recherchant la reconstruction avec la plus petite erreur. Pour une meilleure stabilité, nous optimisons cette erreur sur un ensemble de $M \in \mathbb{N}$ instants consécutifs.

Pour des raisons de simplicité, nous écrivons simplement Y_t^l l'événement $\hat{X}_t = Y_t^l$. Le but est donc de trouver, comme estimation :

$$Y_{t-M:t}^* = \underset{\{Y_s^{l_s}\}_{s=t-M:t}}{\operatorname{argmax}} \mathbb{P}\left(\{Y_s^{l_s}\}_{s=t-M:t} \middle| \{S_{t-M:t}^k\}_{k\in K}\right) \quad (1)$$

Cette problématique, communément appelé logique différée, ou « Modal Trajectory Estimation », est déjà résolue par programmation dynamique. Dont la sous-section suivante présente le formalisme.

3.2 Logique différée

D'après [21], le point terminal de la trajectoire optimale peut être trouvé par :

$$Y_t^* = \underset{Y_t^{l_t}}{\operatorname{argmax}}\, \mathcal{J}_t^*\left(Y_t^{l_t}, \{S_{t-M:t}^k\}_{k\in K}\right) \quad (2)$$

Où $\mathcal{J}_t^*$ est la vraisemblance Bayésienne marginale :

$$\begin{aligned}&\mathcal{J}_t^*\left(Y_t^{l_t}, \{S_{t-M:t}^k\}_{k\in K}\right)\\ &\triangleq \max_{Y_{t-M:t-1}} \mathbb{P}\left(\{S_{t-M:t}^k\}_{k\in K} \middle| \{Y_s^{l_s}\}_{s=t-M:t}\right)\\ &\qquad \cdot \mathbb{P}\left(\{Y_s^{l_s}\}_{s=t-M:t}\right)\end{aligned} \quad (3)$$

En commençant par la connaissance de la valeur initiale $\mathcal{J}_{t-M}^*\left(Y_{t-M}^{l_{t-M}}, \{S_{t-M}^k\}_{k\in K}\right) = \mathbb{P}\left(\{S_{t-M}^k\}_{k\in K} \middle| Y_{t-M}^{l_{t-M}}\right) \times \mathbb{P}\left(Y_{t-M}^{l_{t-M}}\right)$, elle peut être récursivement calculée par

$$\begin{aligned}&\mathcal{J}_t^*\left(Y_t^{l_t}, \{S_{t-M:t}^k\}_{k\in K}\right)\\ &= \max_{Y_{t-1}^{l_{t-1}}} \mathbb{P}\left(\{S_t^k\}_{k\in K} \middle| Y_t^{l_t}\right) \times \mathbb{P}\left(Y_t^{l_t} \middle| Y_{t-1}^{l_{t-1}}\right)\\ &\qquad \times \mathcal{J}_{t-1}^*\left(Y_{t-}^{l_t}, \{S_{t-M:t-1}^k\}_{k\in K}\right)\end{aligned} \quad (4)$$

Où $\mathbb{P}\left(\{S_t^k\}_{k\in K} \middle| Y_t^{l_t}\right)$ peut être considéré comme une probabilité d'observation pour laquelle nous devons trouver un modèle d'observation, et $\mathbb{P}\left(Y_t^{l_t} \middle| Y_{t-1}^{l_{t-1}}\right)$ une probabilité de transition pour laquelle nous devons trouver un modèle dynamique.

Enfin, une implémentation exploitant cette démonstration est le célèbre algorithme de Viterbi [22].

Avant de détailler ce qu'il nous reste à déterminer, résumons notre approche par un schéma-bloc.

3.3 Synoptique

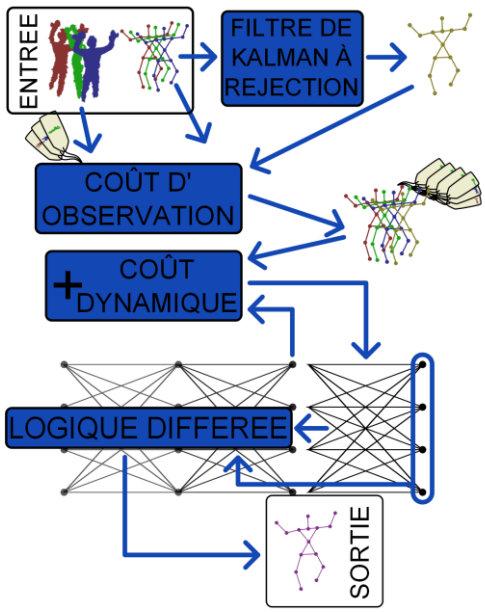


Figure 1. Schéma en blocs de notre approche.

La **Figure 1** donne un aperçu d'ensemble du processus. L'algorithme reçoit une segmentation S_t^k et le squelette Y_t^k reconstruit par NiTE. Nous supposons que la segmentation est fiable, elle sera exploitée comme observation pour mesurer la vraisemblance d'un squelette $\mathbb{P}\left(\{S_t^k\}_{k\in K} \middle| Y_t^{l_t}\right)$. Les squelettes calculés sont insérés dans un filtre de Kalman pour introduire une prédiction de posture. Tous les squelettes $Y_t^l, l \in L \supset K$ sont acco-

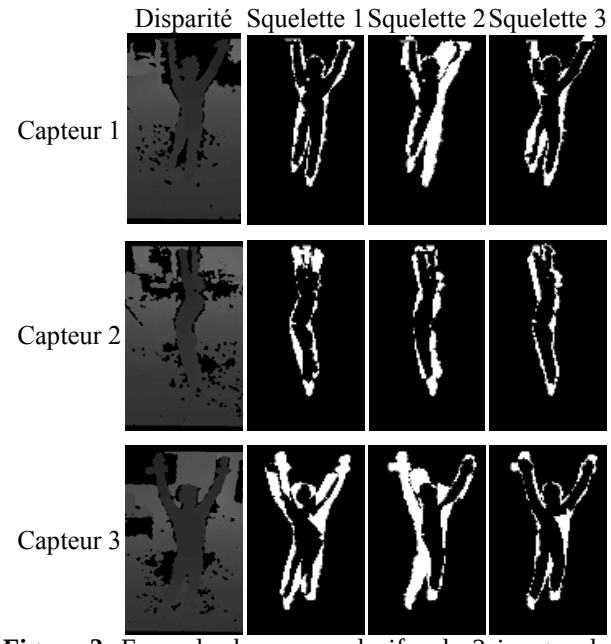


Figure 2. Exemple de « ou exclusif » de 2 images de segmentation, provenant de notre base de données acquise avec un système de 3 capteurs, présenté section 4 . Un pixel blanc signifie que seule une segmentation comprenait ce pixel. Les images sont tronquées horizontalement pour des raisons de présentation.

lés à la fin du treillis des trajectoires de longueur M possibles. Les probabilités de transition $\mathbb{P}\left(Y_t^{l_t} \middle| Y_{t-1}^{l_{t-1}}\right)$, basées sur la dynamique, sont combinées à la probabilité d'observation calculée précédemment. Enfin, un algorithme de Viterbi calcule la trajectoire la plus probable,

$$\left\{\hat{X}_{t-M}, \dots, \hat{X}_t\right\} = \underset{Y_{t-M}^{l_{t-M}}, \dots, Y_t^{l_t}}{\operatorname{argmax}} p\left(Y_{t-M}^{l_{t-M}}, \dots, Y_t^{l_t} \middle| S_{t-M}^K, .., S_t^K\right)$$

à travers tout le treillis. L'estimation de la position des articulations par le processus est $\hat{X}_t$.

À la manière d'une approche de type Monte Carlo, nous voulions d'abord compléter les K hypothèses de squelettes par de très nombreuses hypothèses générées aléatoirement selon la dynamique. Certes, le filtrage particulaire a souvent été privilégié dans ce contexte [23]. Ici, nous privilégions une approche de type programmation dynamique, et ensuite, pour des raisons de complexité des calculs de probabilité d'observation. Exploiter la prédiction issue d'un filtre pour produire une hypothèse supplémentaire nous est apparu comme étant une alternative intermédiaire, à moindre coût mais plus sophistiquée, à une approche stochastique.

Les sous-sections suivantes détaillent l'implémentation de notre lisseur, notamment : les modèles d'observation et de dynamique, et le filtre de Kalman à réjection.

3.4 Modèle d'observation

Afin de déterminer la vraisemblance d'une hypothèse $Y_t^{l_t}$ de squelette à l'instant t, la donnée la plus facilement disponible est S_t^K la segmentation des régions d'intérêt produite durant la détection des squelettes par NiTE, comme le montre la **Figure 2**.

On peut y voir qu'une fois reprojetée dans un autre point de vue (en dehors de la diagonale), le squelette d'apparence correct dans son propre point de vue (cas de la diagonale) peut produire des résultats variables.

Pour autant que nous puissions en juger, le middleware réalise une extraction de la silhouette, i.e. du premier plan, puisqu'il est perdu lorsque le senseur est déplacé. Les nouveaux acteurs entrant dans le champ de vision sont identifiés et maintenus indépendants ensuite, et la perception de la profondeur permet de s'affranchir de toute considération d'apparence (texture, couleur) de la cible.

Puisque la segmentation utilisateur $S_t^k(i,j)$ est la segmentation de premier plan en i,j de l'utilisateur tel que vu par le capteur k, elle peut être comparée à une segmentation artificielle $\mathbb{S}_t^{l,k}(i,j)$ basée sur la projection du squelette Y_t^l faite de cylindres blancs sur l'image.

Pour calculer une distance entre des cartes de segmentations, nous comptons simplement les pixels blancs dans exclusivement l'une ou l'autre image, c'est-à-dire, le ou exclusif ; ou bien la somme des différences absolues entre chaque pixel $d_t(k,l) = \sum_{i,j} |S_t(i,j) - \mathbb{S}(i,j)|$.

Telle-quelle, d_t varie avec la résolution et la place qu'occupe le corps de l'acteur dans l'image, qui est fonction de sa distance au senseur. Ainsi, il faut normaliser. Si nous le faisons par rapport à l'aire totale occupée sur l'image réelle, nous obtenons une mesure en grande partie insensible à la distance, si nous ignorons les effets d'échantillonnage dus au travail avec une bitmap. Il en résulte :

$$\mathrm{D_t}(k,l) = \frac{d_t(k,l)}{\sum_{i,j} S_t^k(i,j)} \quad (5)$$

Nous modelons ensuite la densité de probabilité $\mathbb{P}\left(\{S_t^k\}_{k\in K} \middle| Y_t^{l_t}\right)$ d'avoir une observation correcte, comme la distribution normale autour d'un D_t d'une moyenne $\mu = 0{,}1$ pour cause de bruit de mesure de profondeur et l'approximation des membres par des cylindres, et d'écart-type $\sigma = 0{,}1$, c'est à dire

$$\mathbb{P}\left(\{S_t^k\}_{k\in K} \middle| Y_t^{l_t}\right) \triangleq \prod_{k\in K} \frac{1}{\sqrt{2\cdot\pi\cdot\sigma^2}} e^{-\frac{(\mathrm{D_t}(k,l)-\mu)^2}{2\cdot\sigma^2}} \quad (6)$$

On peut noter que la pleine résolution de 640×480 n'a pas été utilisée dans l'implémentation. Nous utilisons une sous-résolution d'un quart à 160×120 pour accélérer les calculs et rester super-temps réel. De plus, au lieu de maximiser la probabilité, l'algorithme de Viterbi est appliqué tel-quel, minimisant pour le même effet, $\sum_{k\in K} \frac{(D_t(k,l)-\mu)^2}{\sigma^2}$, mais plus rapidement et sans dépassement de la capacité des nombre flottants par valeur inférieure. En effet, avec 45 degrés de liberté, les densités de probabilités (notamment dynamique) atteignaient des valeurs inférieures à $2{,}22507.e-308$, arrondie à 0. Nous avons transposé ce principe au modèle dynamique car il utilise lui aussi une probabilité gaussienne.

3.5 Modèle dynamique

Considérant la nature cartésienne de notre modèle, il est difficile d'obtenir un modèle cinématiquement fidèle. Nous pensons qu'utiliser un modèle articulaire aurait plus de sens, mais serait trop long à calculer, et donc incompatible avec nos contraintes de coût CPU.

Pour l'instant, l'objectif du modèle de dynamique est de réduire le jitter et de favoriser la cohérence spatiotemporelle et donc filtrer les inversions de labellisation gauche-droite classiquement observées lors de reconstruction OpenNI d'une personne de dos.

Ainsi, nous utilisons une densité gaussienne centrée sur chaque articulation, avec un écart-type ajusté pour chaque articulation, basé sur l'écart-type observé sur la vérité terrain

$$\mathbb{P}\left(Y_t^{l_t} \middle| Y_{t-1}^{l_{t-1}}\right) \triangleq \prod_{j\in J} \frac{1}{\sqrt{2\cdot\pi\cdot{\sigma_j}^2}} e^{-\frac{\left(Y_t^{j,l_t} - Y_{t-1}^{j,l_{t-1}}\right)^2}{2\cdot{\sigma_j}^2}} \quad (7)$$

3.6 Filtre de Kalman à réjection de mesures

Le filtre de Kalman que nous avons employé utilise un modèle d'état très simple pour limiter les temps de calcul. Chaque état X^j de l'articulation j est traqué séparément comme suit (les unités sont le mm, et le pas de temps $20^{-1}s$) :

$$X^j = (x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z})^T \quad (S)$$

$$X_t^j = A \cdot X_{t-1}^j + W_{t-1} \quad (E), \quad Z_t^j = H \cdot X_{t-1}^j + V_{t-1} \quad (O)$$

$$W \sim \mathcal{N}(0, Q) \quad and \quad V \sim \mathcal{N}(0, R) \quad (N)$$

$$H = (I_3 \quad 0_3) \quad (O_M), \quad A = \begin{pmatrix} I_3 & I_3 \\ 0_3 & 0.8 \cdot I_3 \end{pmatrix} \quad (E_M)$$

$$Q = \begin{pmatrix} 100^2 \cdot I_3 & 0 \\ 0 & 100^2 \cdot I_3 \end{pmatrix} \quad (N_{M1}), \quad R = 200^2 \cdot I_3 \quad (N_{M2})$$

Pour des raisons de simplicité et pour établir le nom des variables, nous rappelons les principales équations d'un filtre de Kalman :

$$\begin{aligned} \hat{X}_{t+1}^{j-} &= A \cdot \hat{X}_t^j \\ P_{t+1}^{j-} &= A \cdot P_t^j \cdot A^{-1} + Q \end{aligned} \qquad (8.1)$$

$$\begin{aligned} S_{t+1} &= H \cdot P_{t+1}^{j-} \cdot H^T + R \\ \boldsymbol{r}_{t+1}^j &= Y_{t+1}^j - H \cdot \hat{X}_{t+1}^{j-} \end{aligned} \qquad (8.2)$$

$$\begin{aligned} K_{t+1} &= P_{t+1}^{j-} \cdot H^T \cdot S_{t+1}^{-1} \\ \hat{X}_{t+1}^j &= \hat{X}_{t+1}^{j-} + K_{t+1} \cdot \boldsymbol{r}_{t+1}^j \\ P_{t+1}^j &= P_{t+1}^{j-} - K_{t+1} \cdot H \cdot P_{t+1}^{j-} \end{aligned} \qquad (8.3)$$

$$(\hat{X}_{t+1}^{j-}, P_{t+1}^{j-}) = (\hat{X}_{t+1}^j, P_{t+1}^j) \qquad (8.1')$$

L'étape (8.1) est effectuée à chaque pas de temps, (8.2) est utilisée afin de tester la réjection expliquée ci-après. La réjection annule éventuellement le calcul à ce moment, sinon, l'étape (8.3) est effectuée. Pour chaque mesure supplémentaire, l'étape (8.1') est faite (l'estimée est seulement la prédiction).

Ce modèle n'a qu'un désavantage, c'est la non-conformité à l'articularité du corps humain. Mais en pratique, puisque les mesures proviennent d'un système conformant, les résultats le sont approximativement. En revanche, aucune conversion n'est nécessaire depuis le format d'entrée, et les paramètres restent simples.

Cependant, toutes les articulations ne suivent pas un tel modèle même très approximativement. Par exemple, si l'utilisateur est de profil, les deux pieds peuvent se retrouver positionnés au même endroit. Pour éviter l'assimilation de telles mesures (ce qui dégrade sévèrement les performances), une réjection (ou vérification des résidus) est mise en place.

Cette technique, publiée pour la première fois dans [24], consiste à rejeter l'assimilation de Y_t^j si $(\boldsymbol{r}_t^j)^T \cdot S_t^{-1} \cdot \boldsymbol{r}_t^j > g_t^2$, où $\boldsymbol{r}_t^j$ est l'innovation, S_t la matrice de covariance des résidus, et g_t est le seuil de réjection. g_t est généralement 1, 2 ou 3, signifiant que « la norme L2 de l'innovation est à plus de g_t sigmas résiduels. » Ce qui indique généralement un senseur défaillant, car un tel phénomène a respectivement 32%, 5% ou 0,2% de probabilité de se produire pour $g_t = 1{,}2$ ou 3.

La mesure avec la meilleure innovation est assimilée en premier. Une bonne mesure suffit.

Si aucune mesure n'est prise en compte, la variance de l'erreur P_t^j devient éventuellement assez grande pour permettre une mesure au travers de la réjection.

4 Acquisition de données et plateforme multi-senseurs

Notre objectif est de quantifier la précision des estimations des positions 3D des articulations des systèmes mono-senseur et multi-senseurs. Par conséquent, nous avons décidé d'utiliser un système commercial de MoCap de Motion Analysis [25].

Cette acquisition a pour but d'atteindre trois objectifs :

Premièrement, quantifier les performances de NiTE depuis plusieurs orientations : frontale (telle que conçue), mais aussi de profil et en arrière ;

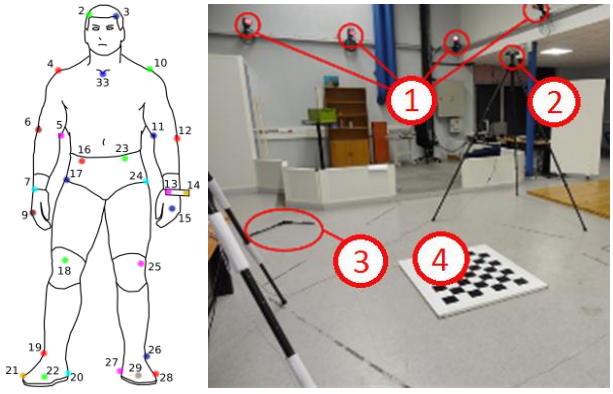
Deuxièmement, créer une base de données pour exploitation par un algorithme d'apprentissage ;

Troisièmement, créer une base de données pour évaluer quantitativement le gain apporté par la fusion multi-capteurs.

Nous avons placé et orienté le référentiel MoCap sur la mire de calibration couleur, notés respectivement ③ et ④ dans la **Figure 3**(b), afin de localiser tous les senseurs dans le référentiel MoCap. Localiser la partie capteurs couleurs des senseurs suffisait car la transformation rigide entre capteur D et capteur RGB est déterminée en usine, et assurée par OpenNI. Certains travaux [26] montrent qu'on peut améliorer de manière significative cette transformation, mais seulement à moins de 80 centimètres du capteur.

La synchronisation temporelle a été faite a posteriori par projection des marqueurs MoCap (**Figure 3**(a)) dans les images couleurs. Sur ces images, un mouvement rapide laissait une traînée floue par marqueur. Puisque nous avions 10 fois plus de mesures MoCap que d'images, nous avons finement ajusté la synchronisation de telle sorte que le marqueur était projeté au milieu de sa traînée. Enfin, puisque tous les flux de squelette et de segmentation étaient générés par un même processus utilisant OpenNI qui fournit un horodatage, nous pouvons les considérer comme synchronisés entre eux. De par la durée des séquences (90 secondes en moyenne) nous supposons que de la dérive temporelle entre les horloges MoCap et OpenNI n'est pas significative par rapport à la précision attendue.

Les données acquises sont scindées en deux ensembles, notés IRSS35, avec un ensemble complet de marqueurs (35), permettant la construction d'un squelette OpenNI complet, et NS-CAP13, avec un ensemble réduit de marqueurs (13), pour des tests rapides. Ces deux sous-ensembles présentent plusieurs séquences, couvrant des



(a) (b)

Figure 3. Expérience de mesure de la précision. (a) configuration des marqueurs de Motion Capture [2], et (b) configuration de la Motion Capture (① 4 des 10 caméras IR ou proche IR) et des Xtions (②), avec le référentiel MoCap (③) et la mire de 1 m² de calibration des caméras (④) présente aussi.

exercices et des mouvements sportifs, des postures de la vie courante, et enfin des poses aléatoires extrêmement variées afin d'augmenter la variabilité. IRSS35 a neuf séquences, certaines des répétitions, pour un total de 16 minutes, soit environ 21569 images de profondeur, exploitables.
Nous faisons remarquer que chaque séquence n'est pas un simple mouvement, mais un mélange d'actions, e.g. pour la séquence sportive : haltérophilie, course, des squats, et pour d'autres séquence, du nettoyage de sol, de la dance, et des déplacements de meubles.

5 Evaluations et résultats

Notre approche s'execute en temps réel, i.e. aussi rapidement que le flux des capteurs de profondeur (20Hz) avec une application mono-thread sur un processeur Inter Core i7 2760QM (2.40 GHz). Puisque le filtrage était réalisé hors-ligne, la décompression des images PNG de l'acquisition était faite par le même thread, et constituait 80% des calculs. Cela signifie que les instances de NiTE aurait pu fonctionner sur le même processeur, même s'il utilisait 100% d'un autre cœur, et notre framework serait resté temps-réel.
La **Figure 4** montre la proportion d'estimation squelette dont les articulations restaient en deçà d'une certaine distance de la vérité terrain (50, 100, 150 et 200 mm. Ces proportions sont donc déjà cumulatives). Chaque squelette proposé par le détecteur OpenNI mono-capteur a une entrée à des fins de comparaison (*Sensor 1* à *3*). De plus, nous avons généré l'oracle *Sensor Best5*, un pseudo-résultat dont la valeur est la valeur du *Sensor* ayant la plus petite distance cumulée à la référence (donc avec la plus petite erreur).
Les trois configurations testées sont

- *Notre approche*, basée sur l'algorithme de Viterbi opérant un choix sur les 3 squelettes et le filtre de Kalman du premier ordre à réjection.
- Un simple filtre de *Kalman* d'ordre 0 (« moyenneur »), pour référence
- L'implémentation décrite sans squelette supplémentaire fourni par filtrage. C'est l'entrée *Viterbi seulement.*

Les résultats nous permettent de tirer trois conclusions.
Premièrement, une implémentation de type Viterbi, sans squelette supplémentaire, c'est-à-dire agissant comme un sélecteur de senseur, peut approcher de très près la précision maximale concernant les distances à la vérité terrain des articulations.
Deuxièmement, l'ajout d'hypothèses supplémentaires apporte deux avantages : une fiabilité plus importante, puisque nous avons moins d'image où les positions ont plus de 200mm d'erreur, et troisièmement, une précision bien supérieure, comme l'indique le nombre d'images avec une précision inférieure à 50mm.
Par ailleurs, *Sensor 3* était l'estimation dont l'utilisateur faisait face au capteur (à l'exception des mouvements rotatifs). Les chiffres montrent que bien qu'il soit le plus fiable pour l'ensemble des articulations, il n'est pas nécessairement le plus précis. Les autres capteurs sont plus précis sur certaines articulations.
Le gain est fonction que la partie corporelle est globalement visible (typiquement, la tête) ou cachée (typiquement, les pieds), donnant principalement une augmentation de précision ou de fiabilité, respectivement.

6 Conclusions et perspectives

Dans cet article, nous décrivons une approche combinant les concepts de la programmation dynamique, de filtrage, et de vision. Les résultats s'avèrent être une

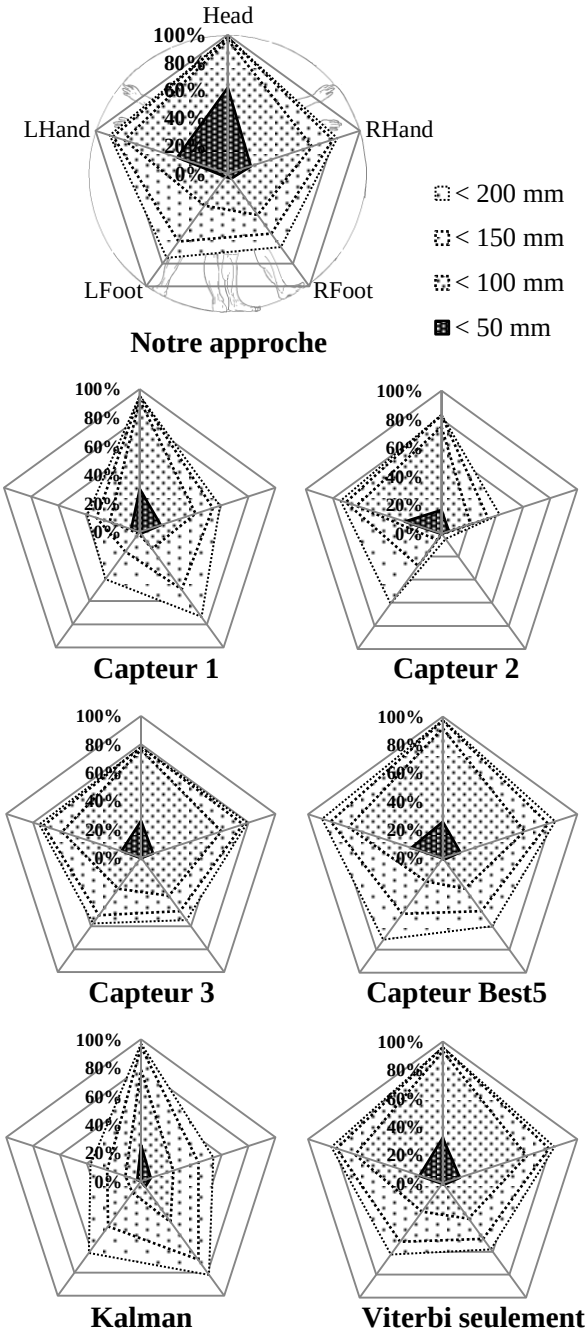


Figure 4. Evaluations quantitatives de notre approche.

nette amélioration sur des implémentations distribuées avec les capteurs utilisés, ceci en utilisant un système de Capture de Mouvement commercial afin d'acquérir une vérité terrain. L'implémentation fonctionne de manière super-temps réel, satisfaisant des contraintes temps-réel avec seulement une fraction du temps d'un cœur processeur disponible. Nous obtenons une unique estimation, logiquement plus précise, des positions des articulations, à partir de celles faites par reconstruction monoculaire.
Cependant, les résultats sont atteints au terme de nombreuses simplifications, telles que travailler avec un modèle cartésien, en utilisant qu'une seule hypothèse de prédiction, et en n'utilisant une reprojection 2D. Nous voudrions améliorer ces résultats en dépassant ces choix, par exemple en prédisant selon plusieurs modèles en parallèle, en privilégiant un modèle articulaire du squelette, ou en utilisant une observation basée voxel.

Par ailleurs, une implémentation GPU est à l'étude pour accroître le gain CPU et étendre au suivi de plusieurs postures simultanément.

Bibliographie

[1] Alfonso, Darren. Microsoft Investor Relations - Press Release. [Online] January 27, 2011. [Cited: September 25, 2012.] http://www.microsoft.com/investor/EarningsAndFinancials/Earnings/PressReleaseAndWebcast/fy11/q2/default.aspx.

[2] Freedman, Barak, Shpunt, Alexander et Arieli, Yoel. *Distance-Varying Illumination and Imaging Techniques for Depth Mapping. US 20100290698 A1* United States, 18 November 2010.

[3] ASUS. ASUS Xtion PRO LIVE. [En ligne] [Citation : 23 01 2013.] http://www.asus.com/Multimedia/Xtion_PRO_LIVE/.

[4] OpenNI, consortium. OpenNI | The standard framework for 3D sensing. [En ligne] [Citation : 23 01 2013.] http://www.openni.org/.

[5] PrimeSense. NiTE Middleware - PrimeSense. [En ligne] [Citation : 07 05 2013.] http://www.primesense.com/solutions/nite-middleware/.

[6] Binney, Daniel and Boehm, Jan. Performance Evaluation of the PrimeSense IR Projected Pattern Depth Sensor. [Poster]. London, England, United Kingdom : University College London, September 14, 2011.

[7] Andersen, M. R., et al., et al. *Kinect Depth Sensor Evaluation for Computer Vision Applications.* Aarhus : Aarhus University, 2012.

[8] *Real-Time Human Motion Tracking using Multiple Depth Cameras.* Zhang, L., et al., et al. 2012, Proc. of the International Conference on Intelligent Robot Systems (IROS).

[9] *Real-time human pose recognition in parts from single depth images.* Shotton, J., et al., et al. 2011, Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 1297 -1304. ISSN:1063-6919.

[10] *The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation.* Taylor, J., et al., et al. 2012. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 103-110.

[11] Moeslund, Thomas B., Hilton, Adrian et Krüger, Volker. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding.* 2006, Vol. 104, 2-3, pp. 90 - 126.

[12] *Articulated Body Motion Capture by Stochastic Search.* Deutscher, Jonathan and Reid, Ian. 2, s.l. : Kluwer Academic Publishers, 2005, International Journal of Computer Vision, Vol. 61, pp. 185-205.

[13] *Multi-view 3D Human Pose Estimation combining Single-frame Recovery, Temporal Integration and Model Adaptation.* Hofmann, Michael et Gavrila, Dariu M. 2009. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 2214-2221.

Remerciements

Ces travaux sont supportés par une Convention CIFRE de l'ANRT, numéro 2011/0734.

[14] *Depth-supported real-time video segmentation with the Kinect.* Abramov, A., et al., et al. 2012. Applications of Computer Vision (WACV), 2012 IEEE Workshop on. pp. 457-464.

[15] *Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis.* Forsyth, David A., et al., et al. s.l. : Now Publishers Inc., 2006. Foundations and Trends in Computer Graphics and Vision. DOI: 10.1561/0600000005.

[16] *Multi-cue onboard pedestrian detection.* Wojek, C., Walk, S. et Schiele, B. Miami, FL : s.n., 2009. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 794-801.

[17] *Stable Multi-Target Tracking in Real-Time Surveillance Video.* Benfold, Ben et Reid, Ian. 2011. CVPR. pp. 3457-3464.

[18] *Robust People Tracking with Global Trajectory Optimization.* Berclaz, Jerome, Fleuret, Francois et Fua, Pascal. [éd.] IEEE Computer Society. 2006. Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 1, pp. 744-750.

[19] *People-Tracking-by-Detection and People-Detection-by-Tracking.* Andriluka, M., Roth, S. et Schiele, B. Anchorage : s.n., 2008. Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.

[20] *HumanEva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion.* Sigal, L., Balan, A. O. et Black, M. J. 1-2, March 2010, International Journal of Computer Vision, Vol. 87, pp. 4-27.

[21] *A dynamic programming approach to trajectory estimation.* Larson, R.E. et Peschon, J. 3, 1966, Automatic Control, IEEE Transactions on, Vol. 11, pp. 537-540.

[22] *A personal history of the Viterbi algorithm.* Viterbi, A.J. 4, 2006, Signal Processing Magazine, IEEE, Vol. 23, pp. 120-142.

[23] *Cooperative passers-by tracking with a mobile robot and external cameras.* Mekonnen, A.A., Lerasle, F. et Herbulot, A. 0, 2012, Computer Vision and Image Understanding, pp. -. ISSN:1077-3142.

[24] Maybeck, Peter S. *Stochastic models, estimation, and control.* s.l. : Academic Press, 1979.

[25] Maloney, Rita. Movement Analysis Products. [En ligne] Motion Analysis Corporation, 4 January 2013. [Citation : 11 January 2013.] http://www.motionanalysis.com/html/movement/products.html.

[26] *Joint Depth and Color Camera Calibration with Distortion Correction.* Herrera C., Daniel, Kannala, Juho et Heikkila, Janne. 10, Los Alamitos : IEEE Computer Society, 2012, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34. ISSN:0162-8828.