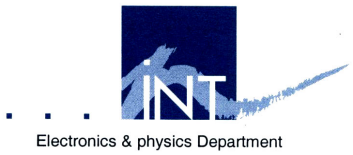


Reconnaissance du Geste Humain par Vision Artificielle: Application à la Langue des Signes

Présenté par:
Arnaud Deslandes
Arnaud.Deslandes@int-evry.fr

Rapport de stage dans le cadre du **DEA IARFA**: Intelligence Artificielle, Reconnaissance des Formes et Applications de l'**Université Paris 6** Pierre et Marie Curie.

Réalisé à l'**INT**, Institut National des Télécommunications en 2002 sous la responsabilité de:
Bernadette Dorizzi et Patrick Horain
Bernadette.Dorizzi@int-evry.fr
Patrick.Horain@int-evry.fr

Résumé:

Ce document présente un système de reconnaissance des mots isolés de la Langue des Signes Française (LSF) à partir de données en trois dimensions. Ces données sont obtenues en appliquant une méthode d'acquisition 3D des gestes à partir de séquences vidéo, sans marqueurs et sans connaissance à priori des mouvements attendus. Elle est basée sur le recalage d'un modèle 3D articulé sur des images en couleur, en respectant des contraintes morphologiques et biomécaniques. Ces paramètres sont ensuite présentés en entrée d'un Modèle de Markov Caché (MMC) pour réaliser la reconnaissance. Chaque modèle est appris pour un signe particulier par l'algorithme de Baum-Welch. La reconnaissance est effectuée par mise en compétition de ces modèles et comparaison de la valeur de log-vraissemblance renvoyée par l'algorithme de Viterbi pour chaque séquence de test.

Abstract:

This document present a system for recognizing isolated French Sign Language (LSF) words from three-dimensional data. The data are obtained by using three-dimensional gesture-acquisition methods applied on video sequence, without markers and without priori knowledge of the awaited movement. It is based on the registration of a 3D articulated model on color images with respect to morphological and biomechanical constraints. Those parameters are then given as input to a Hidden Markov Model (HMM) for the process of recognition. Each model of the set is trained for a specific sign by the Baum-Welch procedure. Confronting the models and comparing the log-likelihood value returned by the Viterbi algorithm for each test sequence does the recognition.

Mots clés:

Acquisition et Reconnaissance de gestes, Langue des Signes Française, Modèle de Markov Caché, Vision par ordinateur.

Key words:

Gesture acquisition and recognition, French Sign Language, Hidden Markov Models, Computer Vision.

I. Introduction:

La langue des signes française (LSF) est le moyen de communication naturel des sourds en France. L'acquisition des gestes des signeurs peut être en premier lieu effectuée au moyen de gants instrumentés mais ils se révèlent chers, fragiles et encombrants. L'utilisation de caméras permet de lever cette contrainte, mais utilise habituellement des marqueurs colorés ou lumineux placés sur la personne observée. Les techniques de vision par ordinateur, sans marqueurs, mono ou stéréoscopique apparaissent aujourd'hui comme une alternative prometteuse à ces méthodes contraignantes.

Une méthode pour l'acquisition des gestes de la main et des membres supérieurs procédant par recalage de modèles 3D articulés du corps et suivit dans les séquences vidéo, développée à l'INT, permet d'obtenir de bons résultats.

Cette étude vise à appliquer ce procédé pour l'acquisition de paramètres à partir de gestes de la Langue des Signes Française. Ceux–ci seront ensuite utilisés pour entraîner un système à base de Modèles de Markov Cachés (MMC) afin, dans un premier temps, d'évaluer les capacités d'un tel modèle dans le cadre de la reconnaissance de mots isolés.

Ce choix d'orientation a été effectué dans le but de faire abstraction des problèmes liés à la coarticulation qui apparaissent lorsque l'on cherche à reconnaître des gestes au sein d'une phrase. Il est difficile, même pour un observateur humain (non familier de la langue des signes) de distinguer les limites d'un mot dans une séquence, celles–ci dépendant de la façon dont le mot précédent a été réalisé et de comment le suivant le sera. Il n'y a en effet pas de passage par une position « zéro » qui permette facilement d'isoler les gestes. Résoudre ce problème impose de développer des outils spécifiques tels que des grammaires [1] reprenant la structure de la Langue des Signes ou l'entraînement de MMC dépendant d'un contexte [2] pour espérer obtenir des résultats satisfaisants.

II. Etat des lieux des travaux antérieurs:

A. Acquisition du geste:

L'acquisition du geste par vision artificielle peut être divisée en deux principales composantes : d'une part l'analyse des aspects 2D d'une image [3], [4], [5] et d'autre part la modélisation 3D de leur contenu [6]-[15].

Les méthodes basées sur la 2D ne peuvent généralement reconnaître qu'un nombre limité de gestes et ce souvent après un procédé d'apprentissage. La 3D tire avantage de la connaissance préalable de la forme d'un modèle et de la possibilité d'appliquer des transformations géométriques afin de le déformer. Il est alors nécessaire d'utiliser la stéréovision [10]-[13] ou plus de deux caméras [14], [15] pour acquérir l'information, ce qui constitue un procédé lourd et coûteux.

Il existe cependant des méthodes d'acquisition du geste qui ne demandent qu'une seule caméra. Cutler et Turk [5] utilisent ainsi la taille et le déplacement de taches dans l'image pour reconnaître le mouvement. Brand et Kettnaker [6] utilisent des Modèles de Markov Cachés pour estimer l'orientation 3D d'un corps à partir de silhouettes en basse résolution. Ouhaddi et Horain [9] utilisent des séquences d'images de la main pour recaler un modèle 3D à partir d'une seule caméra.

La méthode que nous utilisons est basée sur le principe de la minimisation d'une fonctionnelle de coût pour évaluer la qualité du recalage du modèle. Mochimaru et Yamazaki [16] puis Kush et Huang [17] ont recalé un modèle de la main par une minimisation de fonctionnelle de coût agissant sur des variations locales de degrés de liberté du modèle. Ohya et Kishino [18] ont recalé un modèle de corps humain par l'intermédiaire d'algorithme génétique. Gavrila et Davis [19] ont cherché à représenter hiérarchiquement un arbre regroupant toutes les configurations possibles de la main dans un espace de paramètres discret. Shimada et al. [20] utilisent des règles de dépendance statistiques entre des configurations successives.

B. Reconnaissance du geste:

Les études réalisées dans le domaine de la reconnaissance automatique des gestes sont relativement récentes, les premières étant liées à l'apparition des gants numériques en 1987 avec le DataGlove de VPL. Par la suite, nombre d'entre elles se sont d'abord attachées à la reconnaissance de gestes dans le cadre d'interactions homme-machine. D'autres, comme les travaux de Yamato et Al [21] ont porté sur la classification par MMC de mouvements de tennis en s'appuyant sur des techniques de quantification vectorielle dans des séquences d'images binaires.

Dans le domaine plus spécifique de la reconnaissance du langage des sourds, Braffort [22] a présenté un système pour la Langue des Signes Française basée sur l'acquisition de paramètres à partir de gants instrumentés. Les mots y sont analysés et caractérisés selon plusieurs critères, le système de reconnaissance étant basé sur des MMC divisés en deux modules : l'un d'une part pour classifier les signes conventionnels et le second d'autre part pour les signes variables et non conventionnels. Les taux de reconnaissance obtenus sont respectivement de 96 et 92%, parmi un dictionnaire de 44 phrases et un vocabulaire de 7 signes.

Starner et Pentland [23], ont développé un système basé sur la vidéo pour la reconnaissance des phrases de la Langue des Signes Américaine ASL, à partir d'un vocabulaire de 40 signes. Les gestes sont modélisés par un MMC à 4 états. Une seule caméra est utilisée pour l'enregistrement des images. Le taux de reconnaissance est compris entre 75 et 99%.

En 1997, Vogler et Metaxas [24] on décrit un système basé sur les MMC et destiné à la reconnaissance de l'ASL à partir d'un vocabulaire de 53 signes. Trois caméras sont utilisées pour permettre l'acquisition de paramètres 3D des mouvements des bras et des mains des signeurs. Une grammaire a été mise en place afin de traiter des phrases dans l'optique d'une reconnaissance continue des mots. 97 phrases de test ont été utilisées, les résultats obtenus varient entre 92.1% et 95.8% en fonction de la grammaire retenue.

La majorité des travaux actuels portent donc sur la reconnaissance des signes dans des séquences longues par utilisation de MMC continus, basés sur l'apprentissage de Baum-Welch et l'algorithme de Viterbi pour approximer la vraisemblance d'une observation étant donné le modèle.

III. Acquisition du geste par recalage d'un modèle 3D:

Notre méthode permet d'acquérir des paramètres 3D à partir d'images fournies par une seule caméra. Elle n'utilise pas de systèmes à base de gants instrumentés ou de marqueurs lumineux ce qui réduit la complexité de mise en œuvre. Elle ne requiert pas non plus de connaissance à priori du geste qui va être effectué.

La procédure d'acquisition consiste en la recherche de la correspondance entre l'image d'une séquence vidéo d'un mouvement, segmentée suivant des paramètres de couleur, et la projection d'un modèle 3D du corps contraint par des limitations morphologiques et biomécaniques. Par cette méthode, il est possible de compenser le manque d'information pour caractériser le geste en 3 dimensions du fait de la présence d'une seule caméra.

Un algorithme itératif d'optimisation est utilisé afin de minimiser le taux de non-recouvrement entre l'image segmentée et la projection du modèle sur cette image. La surface projetée dans le plan pouvant être identique pour différentes postures, il est nécessaire de mettre en place un système de régularisation du geste permettant d'éviter de trop fortes variations de la position du modèle entre deux images successives. Ce document va maintenant détailler les étapes successives du processus d'acquisition du geste. D'autres détails concernant ce module fournis par Bomb et Horain sont disponibles dans [37].

A. Le modèle 3D:

Le modèle 3D que nous utilisons possède 23 degrés de liberté qui permettent de restituer une infinité de postures. Il correspond à la partie supérieure du corps humain, tronc, bras tête et mains, puisque l'on se limite ici à de la reconnaissance de la langue des signes où seules ces parties sont significatives. Chaque mouvement autour d'une articulation est caractérisé par des contraintes liées aux possibilités physiques de mouvement. Celles-ci sont figurées par deux valeurs qui constituent les bornes minimales et maximales autorisées pour représenter un geste. Elles sont aussi intégrées au sein de l'algorithme d'optimisation dont procédé a été décrit en détail par Ouhaddi [25], et seront présentes à la base de la phase de régularisation du geste.

Numéro de Contrainte	Nom du paramètre	Minimum	Maximum
1	Chest Translation x	-70	70
2	Chest Translation y	-140	140
3	Chest Translation z	-10	10
4	Chest Rotation x	-45	45
5	Chest Rotation y	-360	360
6	Chest Rotation z	-45	45
7	Neck Rotation x	-15	45
8	Neck Rotation y	-79	79
9	Neck Rotation z	-41	41
10	Left Upper Arm Rotation x	-180	25
11	Left Upper Arm Rotation y	-150	70
12	Left Upper Arm Rotation z	-134	20
13	Right Upper Arm Rotation x	-180	25
14	Right Upper Arm Rotation y	-70	150
15	Right Upper Arm Rotation z	-20	134
16	Left For Arm Rotation x	180	0.1
17	Right For Arm Rotation x	180	0.1
18	Left Hand Rotation x	-20	20
19	Left Hand Rotation y	-10	10
20	Left Hand Rotation z	-37	27
21	Right Hand Rotation x	-20	20
22	Right Hand Rotation y	-10	10
23	Right Hand Rotation z	-27	37

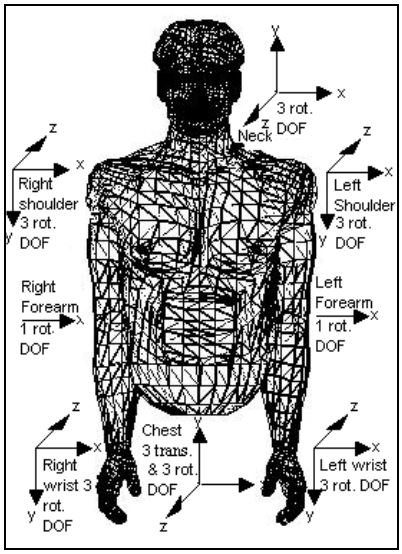


Fig. 1 : *Tableau des contraintes biomécaniques et présentation du modèle 3D en position initiale avec les axes indiquant l'orientation pour l'application des différents degrés de liberté*

Il faut préciser ici que le modèle est construit hiérarchiquement de manière à ce que, lorsqu'on applique une transformation à une partie père, elle s'étend également à toutes les sous-parties liées : une rotation de l'avant bras gauche se propagera par exemple à la main gauche uniquement.

Le buste constitue la racine du modèle, c'est sur lui que sera également appliqué un paramètre supplémentaire de mise à l'échelle qui ne figure pas dans le tableau ci-dessus, puisqu'il n'a pas de correspondance physique, il sert juste à ajuster au mieux la taille du modèle dans chaque vidéo.

Pour la suite, la liste de tous ces paramètres, que nous appellerons vecteur d'état du modèle, sera notée **q**. Elle suffit à définir intégralement une attitude du modèle.

B. Extraction des caractéristiques de l'image:

Le recalage du modèle 3D sur les vidéos dépend de la mise en correspondance de la projection du modèle et des caractéristiques extraites de chaque image. Dans de nombreux travaux d'acquisition du geste, les contours [9], [14], [15], le mouvement [5], les textures [26] ou encore la couleur [4], [9], [27] ont été utilisés avec succès pour réaliser la segmentation. Ainsi, les régions de l'image qui correspondent à la peau peuvent être par exemple efficacement détectées à partir de leurs caractéristiques de chrominance car elles sont alors moins sujettes aux variations liées à l'éclairage de la scène [27], [28]. Il en est de même pour les habits s'ils sont constitués de couleurs uniformes. C'est pour cette raison que nous avons choisi d'effectuer la segmentation de l'image dans l'espace des couleurs YC_bC_r.

Elle a lieu en deux étapes, la première correspond à la classification des différentes zones à identifier, la seconde assure l'élimination du bruit par filtrage.

- *Classification* : la moyenne et la matrice de covariance des différentes classes de couleur sont calculées à partir d'une image de la séquence vidéo déjà étiquetée. En pratique, on utilise la première trame et on isole chaque classe à l'aide d'un logiciel de traitement d'images *(*cf. *Fig.2)*.

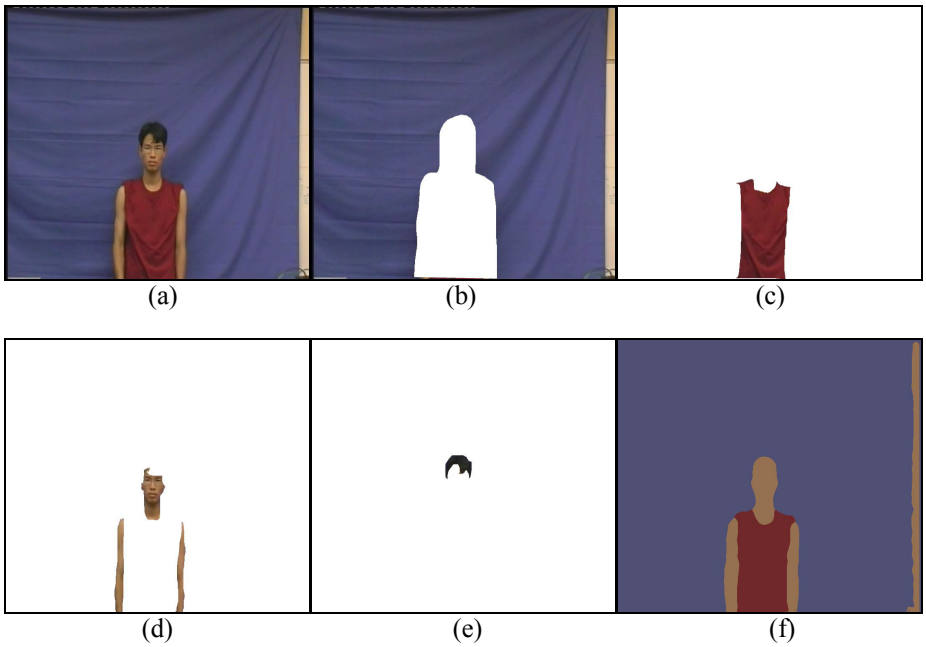


Fig. 2 : image initiale (a) et images correspondant à l'étiquetage des 4 classes qu'il faudra segmenter dans la séquence vidéo : (b) fond, (c) vêtements, (d) peau, (e) cheveux. (f) le résultat obtenu.

Chaque pixel de l'image est ensuite automatiquement discriminé à partir de ses paramètres de chrominance et placé dans la classe qui lui est la plus proche par évaluation de la distance de Mahalanobis [27], [29].
Si l'on note la chrominance d'un pixel $\mathbf{x} = (C_b, C_r)^T$, la distance à la moyenne $\boldsymbol{\mu}_i$ de la $i^{ème}$ classe de matrice de covariance $\boldsymbol{\Sigma}_i$ est alors : $\mathbf{d^2 = (x - \mu_i)^t \Sigma_i^{-1} (x - \mu_i)}$.
Si la valeur obtenue est supérieure à un seuil fixé, le pixel est considéré comme étranger à la classe et est rejeté par le classifieur.

- *Filtrage* : L'image segmentée contient généralement du bruit, le résultat final de la segmentation dépend donc de la qualité de la vidéo qui ne peut pas toujours être assurée. Nous utilisons alors une méthode de filtrage majoritaire (mode filtering) décrite dans [30] pour améliorer le résultat. Pour chaque pixel, on effectue le traitement dans une fenêtre de voisinage et l'on remplace le pixel courant par la nouvelle valeur. Une fenêtre de taille 13x13 a été choisie, elle représente pour notre application le meilleur compromis entre réduction du bruit satisfaisante et temps de calcul raisonnable.

(a) (b) (c) (d)

Fig. 3 : image initiale et images correspondant aux différents modes de segmentation testés:
(b) segmentation dans l'espace RGB : l'ombre près du tronc est reconnue comme appartenant à la classe peau.
(c) segmentation dans l'espace YC_bC_r : les erreurs de classification on baissé de manière significative.
(d) segmentation finale obtenue après filtrage.

C. Evaluation du recalage:

- *Taux de non-recouvrement* :

Après avoir extrait les caractéristiques de l'image, il faut pouvoir lui faire correspondre au mieux le modèle 3D de manière à obtenir une estimation de posture correcte. Pour cela, nous avons la possibilité de faire évoluer les 23 paramètres correspondant aux degrés de liberté du modèle jusqu'à atteindre un résultat satisfaisant.

Pour évaluer la qualité de ce résultat, on projette dans le plan, par application d'un algorithme de Z-buffer, la silhouette du modèle à laquelle on a ajouté des informations de couleur pour caractériser ses différentes sous-parties en accord avec les valeurs obtenues par la segmentation. Il ne reste plus ensuite qu'à mettre en parallèle les caractéristiques de l'image projetée avec celles extraites de la vidéo et à minimiser le taux de non-recouvrement. L'algorithme calcule donc itérativement pour les 23 paramètres le nombre de pixels d'une même classe de couleur qui ne sont pas recouverts jusqu'à ce qu'une valeur optimale soit atteinte.

L'attitude du modèle pour la $k^{ième}$ itération de l'algorithme sur la $t^{ième}$ image de la séquence étant définie par le vecteur $\mathbf{q_t}^k$, Le taux de non-recouvrement $\mathbf{F(q_t^k)}$ s'exprime alors suivant la relation :

$$F(\mathbf{q}_t^k) = \prod_{c=1}^{m} \left(\frac{\left| A_c \cup B_c^k \right| - \left| B_c^k \cap A_c \right|}{\left| A_c \cup B_c^k \right|} \right)^{\frac{1}{m}}$$

A_c caractérise l'ensemble des pixels correspondant à la classe de couleur c.
B_c^k représente la projection des parties du modèle associées à la classe de couleur c, pour l'attitude définie par le vecteur de paramètres q_t^k.
m indique le nombre de classes de couleurs considérées.

- *Optimisation* :

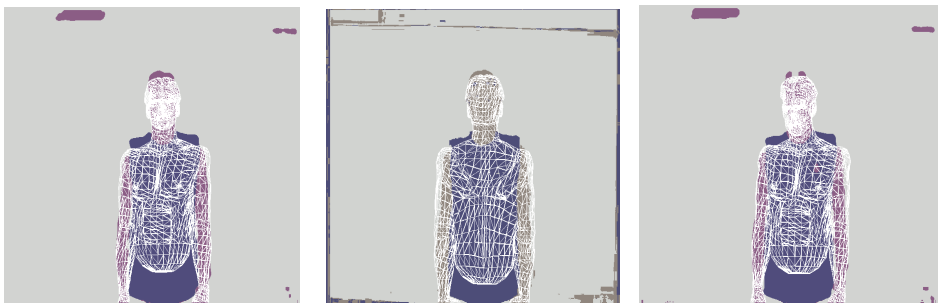
L'optimisation est obtenue par minimisation d'une fonctionnelle de coût liée au taux de non recouvrement décrit ci-dessus. $E(q_t^k) = F(q_t^k)$

Notre choix s'est porté sur l'utilisation d'une méthode de descente de simplexe conformément aux résultats obtenus par Ouhaddi [25] dans son étude d'algorithmes pour le recalage d'un modèle 3D de la main. La descente de simplexe nécessite en effet moins d'opérations d'évaluation de la fonctionnelle que la méthode de Powell [31] et permet de prendre aisément en compte les contraintes biomécaniques, de manière à réduire considérablement l'espace de recherche en éliminant immédiatement les configurations irréalistes.

L'arrêt de l'optimisation est obtenu par la spécification d'un nombre maximal d'itérations que l'algorithme ne doit pas dépasse ou par l'existence d'une différence minimale acceptable entre deux valeurs successives de la fonctionnelle de coût. Cette méthode nécessite également de procéder à l'initialisation des paramètres du modèle 3D en effectuant au préalable manuellement le recalage pour la première image de la séquence vidéo.

D. Régularisation du geste:

Une des limitations qui peuvent apparaître lors de l'utilisation de ce système d'acquisition du geste est liée au fait que la projection du modèle 3D dans le plan n'est pas unique pour des valeurs de contraintes différentes. Il se peut donc que lors du passage entre deux images successives, le modèle effectue par exemple une rotation à 180° sans qu'il soit possible pour le programme d'optimisation de détecter une variation notable dans la valeur de la fonctionnelle de coût (cf *Fig. 4*). C'est pour éviter ce problème, mais aussi pour éliminer les conséquences de la présence de bruit dans les séquences vidéo qui conduit à l'obtention de résultats incorrects, qu'il est nécessaire de mettre en place un système de régularisation du geste.



(a) (b) (c)

Fig.4:
(a) résultat obtenu pour le recalage d'une image quelconque dans une séquence vidéo
(b) résultat pour l'image suivante sans régularisation du geste : le modèle s'est retourné
(c) résultat pour la même image avec régularisation du geste.

Il doit permettre, à partir du résultat de l'optimisation des images précédentes, de prévoir quelle sera l'attitude la plus probable pour le modèle dans l'image suivante. Pour estimer le vecteur de paramètres attendu, on utilise une méthode dérivée du filtrage de Kalman, similaire à celle proposée par Lowe [32].

Si $\mathbf{q_t}$ correspond au vecteur des paramètres et $\dot{\mathbf{q}}_t$ exprime leur vitesse dans l'image t, on peut définir le vecteur d'état :

$$\mathbf{x}_t = [\mathbf{q}_t \quad \dot{\mathbf{q}}_t]^T$$

Kalman modélise ensuite ce vecteur comme une prédiction linéaire $\hat{\mathbf{x}}_t^-$ à laquelle s'ajoute un terme de bruit $\mathbf{w}_{t-1}$:

$$\mathbf{x}_t = \hat{\mathbf{x}}_t^- + \mathbf{w}_{t-1}$$

$$\hat{\mathbf{x}}_t^- = \mathbf{A}\mathbf{x}_{t-1}$$

A représente une matrice 2x2.

Il est possible de transformer ces relations sous la forme :

$$\hat{\mathbf{q}}_t^- = \mathbf{q}_{t-1} + \dot{\mathbf{q}}_{t-1} \qquad (1)$$

$$\hat{\dot{\mathbf{q}}}_t^- = \dot{\mathbf{q}}_{t-1}$$

Le terme de régularisation correspondant s'écrit alors:

$$R(\mathbf{q}_t^k) = \sqrt{\left(\sum_{i=1}^{n} \left(\frac{\hat{q}_t^{i-} - q_t^{k,i}}{\sigma_i} \right)^2 \right)}$$

σ_i est la déviation standard du paramètre i.

$\hat{q}_t^{i-}$ est la valeur prédite pour le paramètre i à partir de l'équation (1).

q_t^{ki} est la valeur du paramètre i à l'itération k de l'algorithme d'optimisation.

Le terme $R(\mathbf{q}_t^k)$ est enfin intégré à la fonction de coût et sera par la suite pris en compte lors de l'optimisation :

$$E(\mathbf{q}_t^k) = F(\mathbf{q}_t^k) + R(\mathbf{q}_t^k)$$

Il introduit une pondération qui va, en cas de différence trop importante entre le vecteur prédit et celui obtenu, augmenter la valeur de la fonctionnelle et conduire le programme à poursuivre ses itérations.

E. Confrontation de l'application à la LSF:

Pour le moment, ce système d'acquisition des paramètres à partir du geste est limité aux mouvements des bras, de la tête et des mains des signeurs. Il ne prend donc en compte qu'une sous-partie des postures qui peuvent être réalisées en LSF. En effet, Braffort [22] a distingué l'existence de 5 caractéristiques qui sont susceptibles de distinguer les mots :

- ***La dynamique*** du mouvement de la main mais aussi des bras du signeur qui différencie les divers aspects d'un verbe : on peut exprimer des notions aussi différentes que brièvement, longtemps, presque, souvent… par ce biais.
- ***L'orientation*** du mouvement, pour conjuguer les verbes ou préciser l'orientation d'un objet.
- ***L'emplacement*** de la main par rapport au corps qui indique l'endroit spécifique où a lieu l'action.
- ***La configuration*** de la main : par exemple, si elle prend la forme d'une boule, elle peut désigner un bol alors que si sa forme est celle d'une pince, c'est une tasse. Elle nécessite de pouvoir distinguer la position des doigts.
- ***La mimique faciale*** pour exprimer un mode de discours : interrogation… ou une émotion : colère, plaisir.

On peut constater, d'après ce qui a été décrit lors de la présentation de la phase d'acquisition des gestes, qu'il n'est pas encore possible en l'état actuel du système, de pouvoir caractériser le mouvement des doigts de la main ou l'expression du visage du fait même de la conception du programme. Les séquences vidéo sur lesquelles nous travaillons utilisent une vue en plan américain de la personne qui réalise le geste. Il y est difficile d'appliquer les mêmes méthodes de recalage pour les doigts que pour le reste du corps car les zones de l'image correspondant à ces parties ne contiennent que très peu de pixels. Il serait nécessaire de posséder une vue en gros plan de la main pour espérer obtenir des résultats identiques à ceux de Ouhaddi [25].
Le procédé d'acquisition des paramètres de l'image 2D permet de la segmenter en diverses régions, mais pas de manière assez précise pour pouvoir conserver des informations sur la position des doigts. La mimique faciale est elle aussi ignorée par ce procédé.

Un autre problème lié au recalage de la projection du modèle 3D apparaît lorsque deux régions qui sont identifiées comme appartenant à une même classe s'intersectent dans l'image dont on a extrait les caractéristiques. Il n'y a, en l'état actuel du programme, pas de procédé mis en œuvre pour guider le positionnement du modèle dans ces conditions. On constate alors des erreurs lors de mouvements qui consistent à passer par exemple une main devant le visage (cf *Fig. 5*) ou superposer les deux bras. Cependant, on peut considérer le résultat comme qualitativement acceptable si les régions sont petites et si le geste s'effectue avec une bonne amplitude.

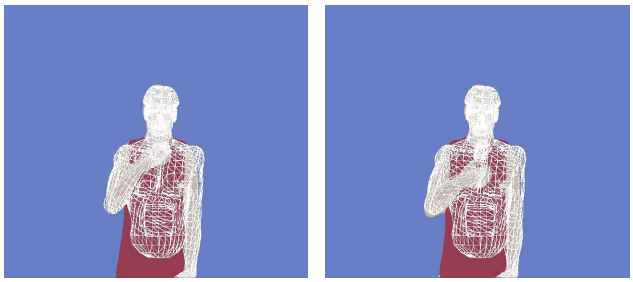


Fig.5 : la main reste recalée sur le cou du fait d'un problème d'auto-occultation dans les images précédentes.

Il faudra donc prendre en compte ces limitations lorsque l'on abordera la phase de reconnaissance automatique de gestes, notamment lors du choix du corpus de données qui sera traité.

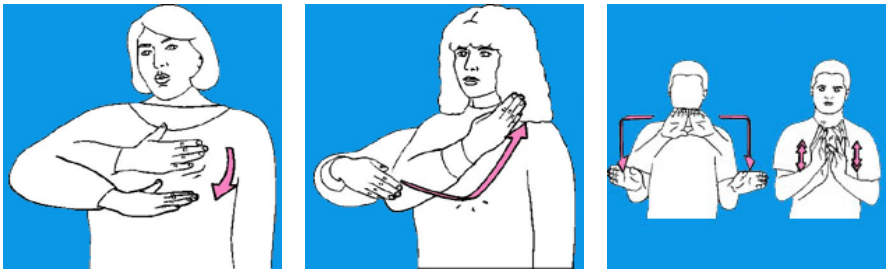
F. Constitution d'un corpus de données:

Pour pouvoir mettre en place un système de reconnaissance automatique de la Langue des Signes Française basée sur les Modèles de Markov Cachés, il est nécessaire d'avoir à notre disposition un certain nombre d'occurrences de paramètres étiquetés, qui vont être utilisés pour constituer d'une part la base d'apprentissage et d'autre par la base de tests. Pour cela, il faut posséder un nombre suffisant d'itérations d'un même geste et ce, réalisées par des personnes différentes, pour espérer obtenir un modèle offrant une bonne capacité de généralisation.

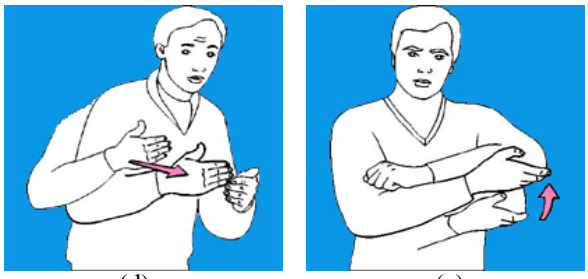
Un corpus de vidéos a donc été filmé à l'INT spécifiquement dans ce but. Cinq mots de la langue des signes Française ont été choisis à partir d'un dictionnaire multimédia d'apprentissage de la LSF [33] parmi une liste de 200 possibilités, pour les raisons suivantes :

- Gestes dans lesquels les doigts ou le visage ne jouent pas de rôle.
- Amplitude importante dans les mouvements pour faciliter le recalage.
- Absence d'auto-occultations entre les bras ou les mains des signeurs.
- Présence équilibrée de gestes impliquant un seul bras ou les deux.
- Facilité d'exécution par des personnes ne connaissant pas la langue des signes.

Il s'agit des mots Bon (a), Blessé (b), Cheminée (c), Approcher (d) et Aider (e).



(a) (b) (c)



(d) (e)

Fig. 6 : Les mots choisis pour faire partie du corpus de données.

Les vidéos ont été prises sur fond bleu, les signeurs portent un vêtement rouge pour faciliter la segmentation et leurs bras sont apparents. Les séquences à raison de 25 images par seconde, varient en longueur en fonction de la complexité du geste exécuté. Elles sont comprises entre 40 et 130 images, leur résolution est de 720x576 pixels.

Les 5 gestes ont été répétés 5 fois, et ce par 4 signeurs différents. La base est donc constituée par environ 100 séquences auxquelles il faut parfois retirer quelques réalisations qui se sont avérées inutilisables et en ajouter d'autres qui ont été tournées par la suite en remplacement, mais parfois pour un autre mot.

Le temps de calcul nécessaire au traitement de ces vidéos a été évalué à près d'une minute par trame sur une station de travail Pentium 4 standard. Une illustration des résultats obtenus est fournie en annexe à ce document et une partie est visible et téléchargeable sur Internet à l'adresse :

http://www-eph.int-evry.fr/~deslande/resultats.html.

On constate que les performances du recalage sont généralement bonnes, les problèmes liés aux auto-occultations des membres n'apparaissent pas dans les cas choisis excepté pour le mot Aider. Un léger retard de suivit du geste par le modèle à l'initialisation du mouvement a été constaté, il ne représente pas un problème notable dans l'optique de la reconnaissance, puisqu'il est très bref et qu'il se répète de manière similaire dans presque toutes les séquences.

Pour chaque vidéo, on dispose donc d'un nombre de fichiers de paramètres équivalent au nombre de trames de la séquence. Ces fichiers contiennent chacun le vecteur d'état **q** du modèle.

IV. Les Modèles de Markov Cachés:

Les Modèles de Markov Cachés (MMC) sont un type de modèle statistique largement employé depuis quelques années en reconnaissance de la parole et plus récemment en reconnaissance de l'écrit, reconnaissance des gestes et par extension reconnaissance du langage des signes. Le point qui suit présente les bases de la théorie des MMC. Une explication détaillée des modèles discrets étant disponible dans [34] et [35], nous ne présenterons ici que les modèles continus que nous allons utiliser.

A. Définition d'un MMC:

Un MMC est caractérisé par les éléments suivants:

- N est le nombre d'états cachés du modèle. On note $S=\{s_1, s_2, \ldots, s_N\}$ l'ensemble des états cachés. A l'instant t, un état est représenté par q_t ($q_t \in S$).
- M est le nombre de symboles distincts que l'on peut observer dans chaque état. On les représente par l'ensemble $V=\{v_1, v_2, \ldots, v_M\}$. A l'instant t, un symbole observable est désigné par O_t ($O_t \in V$).
- Une matrice de probabilité de transition, notée $A=[a_{ij}]$, où a_{ij} est la probabilité à priori de transition de l'état i vers l'état j. Dans le cadre d'un MMC stationnaire du premier ordre, cette probabilité ne dépend pas de t. on définit $a_{ij}=P(q_{t+1}=s_j \mid q_t=s_i)$, $1 \le i, j \le N$.
- Une matrice de distribution des probabilités, notée $B=[b_j(k)]$, associée à chaque état où $b_j(k)$ est la probabilité d'observer le symbole v_k en étant à l'état s_j à l'instant t. on définit $b_j(k)=P(o_t=v_k \mid q_t=s_j)$, $1 \le i \le N$, $1 \le k \le M$.
- Un vecteur $\Pi=[\pi_i]$ de distribution des probabilités de transitions initiales, où π_i est la probabilité de commencer dans l'état i. On définit $\pi_i=P[q_1=s_i]$ avec $1 \le i \le N$.

En résumé, la spécification totale d'un MMC nécessite la spécification des paramètres du modèle, N et M, la spécification des symboles d'observation, et la spécification des trois mesures de probabilités, A, B et Π. Sous une forme plus réduite, on peut dire qu'un MMC noté λ est défini complètement par $\lambda = (A, B, \Pi)$, N et M étant sous-entendus dans la matrice A et B ainsi que le vecteur Π.

Un exemple de MMC est fourni ci-dessous (cf. *Fig. 7*), il s'agit d'un type de modèle appelé « gauche-droite », c'est-à-dire que pour tout $a_{ij} > 0$ on a $j \ge i$. En d'autres mots, les transitions ne sont autorisées que vers un état d'indice supérieur à l'état courant et non l'inverse. C'est le type de topologie le plus répandu lorsqu'il s'agit de modéliser des processus évoluant dans le temps.

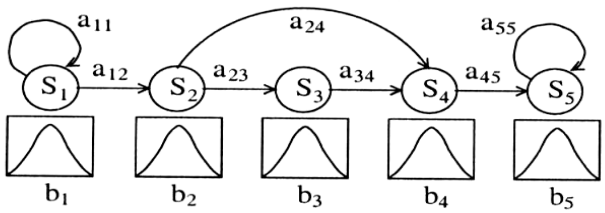


Fig.7 : Un MMC « gauche-droite » avec ses transitions et probabilités d'émission.

B. Les trois problèmes fondamentaux des MMC:

Pour qu'un Modèle de Markov Caché puisse être utilisé dans des conditions réelles, il faut pouvoir résoudre les trois problèmes de base suivants :

(1) Pour une séquence d'observations $O= O_1, O_2, \ldots, O_T$ donnée et un modèle λ, calculer la probabilité $P(O|\lambda)$ que le MMC génère O.

(2) Pour une suite d'observations O et un modèle λ donné, trouver la suite d'états $S_1, S_2, \ldots, S_T$ qui génère O.

(3) Ajustement optimal des paramètres d'un MMC λ de telle manière qu'ils permettent de maximiser $P(O|\lambda)$ pour une suite d'observations O.

(1) On cherche à calculer la probabilité d'une séquence O, connaissant le modèle λ. Pour calculer $P(O|\lambda)$ et si l'on définit $Q= Q_1, Q_2, \ldots, Q_T$ comme une séquence d'états de λ, il suffit de définir une variable $\alpha_t(i)$ telle que:

$$\alpha_t(i) = P(O_1, O_2, ..., O_t, Q_t = S_i | \lambda) \qquad 1 \le i \le N \qquad (1)$$

et qui correspond à la probabilité d'observer la séquence partielle $O_1, O_2, \ldots, O_t$ et l'état S_i à l'instant t.
Il est alors possible de résoudre cette équation inductivement, en trois étapes :

Initialisation :

$$\alpha_1(i) = \pi_i b_i(O_1) \qquad (3)$$

Induction :

$$\alpha_{t+1}(i) = b_i(O_{t+1}) \sum_{j=1}^{N} \alpha_t(j) a_{ij} \qquad 1 \le t \le T-1 \qquad (4)$$

Terminaison :

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \qquad (2)$$

Les calculs supposent que les observations O_i soient indépendantes et que l'on se place dans le cade des Chaînes de Markov d'ordre 1, c'est-à-dire que toutes les transitions vers un autre état ne dépendent que de l'état courant. Cette méthode est appelée algorithme de forward-backward et permet de calculer $P(O|\lambda)$ avec une complexité en $O(N^2T)$.

(2) Le second problème consiste à trouver le chemin Q le plus probable pour un MMC λ, étant donnée une séquence d'observations O, ce qui revient à maximiser $P(Q,O|\lambda)$.

Si l'on définit :

$$\delta_t(i) = \max_{Q_1, \ldots, Q_{t-1}} P(Q_1 Q_2 \ldots Q_t = S_i, O|\lambda) \qquad (5)$$

Par induction on a :

$$\delta_{t+1}(i) = b_i(O_{t+1}) . \max_{1 \le j \le N} \{\delta_t(j) a_{ij}\} \qquad (6)$$

$$\max_Q P(Q, O|\lambda) = \max_{1 \le i \le N} \{\delta_t(i)\} \qquad (7)$$

$\delta_t(i)$ correspond à la probabilité du meilleur chemin partiel amenant en l'état S_i à l'instant t. Les équations (6) et (7) découlent de l'équation (5) par induction sur t. Cet algorithme de programmation dynamique est appelé algorithme de Viterbi. Il permet de calculer à la fois le maximum de probabilité $P(Q,O|\lambda)$ ainsi que la séquence d'états correspondante en une complexité de $O(N^2T)$.

(3) Le troisième problème consiste en l'entraînement du MMC par des données, de telle manière qu'il soit ensuite capable de reconnaître des informations qui ne lui ont encore jamais été présentées mais qui sont proches. Il n'existe pas de solution analytique pour maximiser $P(O|\lambda)$ pour un nombre donné de séquences d'observation, mais il existe une procédure itérative, appelée algorithme de Baum-Welch, qui permet de maximiser $P(O|\lambda)$ localement.

Dans le cas de densités de probabilité *continues*, le processus de réestimation se déroule de la manière suivante :

Soit : $$b_j(O) = \sum_{m=1}^{M} c_{jm} G(O, \mu_{jm}, U_{jm}) \qquad (8)$$

M indique le nombre de gaussiennes
j est le numéro de l'état
c correspond au poids de la gaussienne m dans l'état j
G est une gaussienne de moyenne μ et de matrice de covariance U.

On définit une variable de backward β par :

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, ..., O_T | Q_t = S_i, \lambda) \quad (9)$$

$$\beta_T(i) = 1 \quad (10)$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_{ij}(O_{t+1}) \beta_{t+1}(j) \quad 1 \le i \le N,\ 1 \le t \le T-1 \quad (11)$$

Ainsi que ξ et γ tels que:

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} \quad (12)$$

$$\gamma_t(i) = \sum_{i=1}^{N} \xi_t(i, j) \quad (13)$$

$\sum_{i=1}^{N} \xi_t(i, j)$ peut être interprété comme le nombre attendu de transitions suivies de S_i vers S_j.

De la même façon, $\sum_{i=1}^{N} \xi_t(i, j)$ correspond au nombre de transitions suivies depuis S_i.

Avec cette interprétation, les formules de réestimation pour les transitions et les probabilités d'émission s'écrivent :

$$\overline{\pi}_i = \gamma_1(i) \quad (14)$$

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (15)$$

$$\overline{c}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j, m)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j, k)} \quad (16)$$

$$\overline{\mu}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j, m) O_t}{\sum_{t=1}^{T} \gamma_t(j, m)} \quad (17)$$

$$\overline{U}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j, m)(O_t - \mu_{jm})(O_t - \mu_{jm})T}{\sum_{t=1}^{T} \gamma_t(j, m)} \quad (18)$$

En répétant cette procédure on converge vers une probabilité maximum, généralement au bout de 5 à 10 itérations.

C. Application des MMC à la reconnaissance isolée de mots de la LSF:

L'application de la théorie des MMC à notre problème de reconnaissance s'est traduite par l'adaptation d'un outil existant basé sur la toolbox HMM qui s'intègre au programme Scilab développé par l'INRIA. Cette toolbox a déjà été utilisée à l'INT avec succès dans le cadre de la vérification d'identité et plus précisément l'apprentissage de modèles de signatures [36]. Il permet d'entraîner des MMC continus par l'algorithme de réestimation de paramètres de Baum-Welch en paramètrant le nombre d'états et de gaussienne souhaités. C'est l'algorithme de Viterbi qui permet ensuite d'évaluer la vraisemblance de la séquence la plus probable pour une observation donnée.

Pour nos modèles, nous avons choisi d'utiliser une topologie de type Bakis ou encore appelée « gauche-droite » qui s'adapte particulièrement au problème de reconnaissance de gestes et donc par extension à celui de la langue des signes.

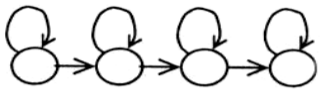


Fig.8 : Un exemple de la topologie choisie pour les modèles.

Le protocole qui a été suivi consiste à créer un modèle spécifique par geste à apprendre. Chaque modèle est entraîné indépendamment à partir d'exemplaires du mouvement qu'il doit permettre de reconnaître, d'après la méthode présentée pour résoudre le problème 3. Le corpus de données fourni en entrée 12 réalisations du geste pour l'apprentissage et 8 pour les tests. La reconnaissance s'effectue étant donné une observation, c'est à dire pour chaque occurrence d'un geste dans la base de test. On calcule alors par l'algorithme de Viterbi (solution du problème 2), pour chaque modèle de geste, la vraisemblance que le mouvement ait été généré par ce modèle. Il ne reste plus alors qu'à classer l'observation comme correspondant au geste pour lequel le modèle renvoi la valeur de vraisemblance maximale.

Cette méthode s'applique bien évidemment uniquement au cas où l'on souhaite reconnaître des mouvements isolés. Elle est basée sur l'hypothèse que chaque geste peut être extrait et traité individuellement. Pour cette raison, nous avons travaillé à partir d'un corpus de données dans lequel les signes sont effectués depuis une position de repos, bras le long du corps.

Les paramètres qui ont été choisis pour être passés en entrée des MMC correspondent aux valeurs des différents degrés de liberté du modèle 3D recalés tels qu'ils sont présentés dans le tableau de la *Figure 1* et illustrés à titre d'exemple en annexe de ce document pour le mot Bon.

Dans l'immédiat, pour traiter les gestes Bon et Blessé pour lesquels seul le bras droit est utilisé, nous avons seulement conservé 7 paramètres parmi 23 afin d'éviter d'effectuer l'apprentissage à partir de données qui ne varient pas.

RightUppArm rotx
RightUppArm roty
RightUppArm rotz
RightForArm rotx
RightHand rotx
RightHand roty
RightHand rotz

Fig. 9 : tableau des noms des paramètres pris en entrée du MMC pour l'apprentissage des séquences Bon et Blessé

V. Résultats obtenus:

Le corpus de données qui a pu être traité à ce jour ne nous a pas permis d'effectuer suffisamment de tests pour formuler des conclusions précises puisqu'il ne couvre de manière significative que deux mots différents. Cependant, nous avons pu apprendre des modèles pour ces deux signes, Bon et Blessé, avec respectivement 12 et 7 séquences de données pour l'apprentissage et 14 et 5 séquences pour les tests.

	2 états	3 états	4 états
1 gaussienne	Geste Bon : 71,5% Geste Blessé : 100%	Geste Bon : 85,7% Geste Blessé : 100%	Geste Bon : 92,8% Geste Blessé : 100%
2 gaussiennes	Geste Bon : 71,5% Geste Blessé : 100%	Geste Bon : 100% Geste Blessé : 100%	Geste Bon : 85,7% Geste Blessé : 100%

Le tableau ci-dessus présente les taux de reconnaissance obtenus pour différentes configurations de modèles : nombre d'états variant entre 2 et 4 pour une et deux gaussiennes. Ces valeurs ne sont pas significatives, particulièrement en ce qui concerne les modèles pour lesquels on a choisi plusieurs gaussiennes. En effet, dans ce cas, le nombre de paramètres fournis lors de l'apprentissage est un élément très important dans la qualité du résultat : il est nécessaire de disposer d'une grande base d'apprentissage pour espérer obtenir un modèle qui fonctionne de manière satisfaisante.

Le taux de reconnaissance obtenu pour 2 états cachés n'est pas très élevé dans le cas de la reconnaissance du mot Bon. Cela peut s'expliquer d'une part par le fait que les deux gestes qui ont été choisis ici à titre d'exemple sont assez proches (cf *Fig.6*) et donc plus difficiles à discriminer que d'autres, mais aussi parce que le nombre d'états est petit et qu'il est peut être insuffisant dans le cadre de nos données. Cela expliquerait aussi l'amélioration de performance obtenue pour 4 états.

L'acquisition des paramètres se poursuit encore actuellement, les autres modèles seront entraînés lorsque suffisamment de paramètres auront été acquis.

VI. Conclusion:

Ce stage a contribué à mettre en place un procédé complet de reconnaissance automatique des gestes à partir de paramètres en 3 dimensions extraits de vidéos filmées par une seule caméra. Un corpus de données a ainsi été constitué afin d'apprendre des Modèles de Markov dédiés à la reconnaissance de signes isolés.

Il a permis de tester la validité des paramètres fournis par le module d'acquisition en montrant que la reconnaissance de geste était possible. Il reste cependant à poursuivre cette étude afin de tester les performances du système sur une plus grande quantité de mots à reconnaître.

Une étude plus poussée pourra ensuite être mise en place afin d'évaluer quel types de modèles sont les plus adaptés pour une application des MMC à la LSF et quels sont les paramètres les plus pertinents à fournir au modèle.

C'est alors qu'il sera possible d'envisager d'étendre le système de reconnaissance à l'identification de signes dans une séquence continue.

VII. Remerciements:

Je tiens à remercier particulièrement ***Patrick Horain*** et ***Bernadette Dorizzi*** qui, par les conseils qu'ils m'ont prodigués, m'ont permit de mener à bien ce stage dans les meilleures conditions. J'aimerais aussi citer ici le travail des stagiaires qui m'ont précédé sur ce projet tels ***Rami Kanhouche*** et ***Mayank Bomb*** pour la partie acquisition du geste, ***Marc Fuentes*** en reconnaissance automatique de signature et sans lesquels cette étude n'aurait pu avoir lieu.

Bibliographie:

[1] H. Hienz, B. Bauer and K. F. Kraiss, *"HMM Based Continuous Sign Language Recognition Using Stochastic Grammars"*, Gesture Workshop 99.

[2] C. Vogler et D. Metaxas, *"ASL Recognition Based on a Coupling Between HMMs and 3D Motion Analysis"*, International Conference on Computer Vision, Mumbay India, 4-7-1998, pp363-369.

[3] J. Martin and J. L. Crowley, "*An Appearance Based Approach to Gesture Recognition*", in Proc. 9th ICIAP, Lecture notes in computer science 1311, Springer, Verlag, Florence, Italy, 1997, pp. 340–347.
Disponible: http://www-prima.imag.fr/~jmartin/english-publis.html

[4] L. Bretzner, I. Laptev, and T. Lindeberg, "*Hand Gesture Recognition Using Multi-Scale Color Features, Hierarchical Models and Particle Filtering*," in Proc. 5th IEEE Int. Conf. on Automatic Face and Gesture Recognition, May 20–21, 2002, pp. 405–410.

[5] R. Cutler and M. Turk, "*View-Based Interpretation of Real-Time Optical Flow for Gesture Recognition*", in Proc. IEEE Conference on Automatic Face and Gesture Recognition, April 14–16, 1998, Nara Japan.
Disponible: http://www.cs.ucsb.edu/~mturk/research.htm

[6] M. Brand and V. Kettnaker, *"Discovery and Segmentation of Activities in Video*", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, 2000, pp. 844–851.

[7] J. M. Regh and T. Kanade, *"Visual Tracking of Height dof Articulated Structures, an Application to HumanTtracking*", in Proc. 3rd ECCV, 1994, vol. 2, pp. 37–46.

[8] S. B. Gokturk, J. Y. Bouguet, C. Tomasi, and B. Girod, *"Model-Based Face Tracking for View-Independent Facial Expression Recognition*", in Proc. 5th IEEE Int. Conf. on Automatic Face and Gesture Recognition, May 20-21, 2002, pp. 272–278.

[9] H. Ouhaddi and P. Horain, *"3D Hand Gesture Tracking by Model Registration*", in Proc. Int. Workshop on Synthetic - Natural Hybrid Coding and Three Dimensional Imaging, September 15–17 1999, Santorini Greece.
Disponible: http://www-eph.int-evry.fr/~horain//Publications/iwsnhc3di99-ouhaddi.pdf

[10] S. Malassiotis, F. Tsalakanidou, N. Mavridis, V. Giagourta, N. Grammalidis, and M. G. Strintzis, *"A Face and Gesture Recognition System Based on an Active Stereo Sensor*", in Proc. Int. Conf. on Image Processing, 2001, vol. 2, pp. 955–958.

[11] R. Yang and Z. Zhang, *"Model-based head pose tracking with stereovision"*, in 5th IEEE Int. Conf. on Automatic Face and Gesture Recognition, May 20-21 2002, pp. 242–247.

[12] R. Plaenkers and P. Fua, *"Model-based silhouette extraction for accurate people tracking*", in Proc. European Conf. on Computer Vision, Copenhagen, Denmark, May 2002.
Disponible: http://vrlab.epfl.ch/Publications/pdf/Plaenkers_Fua_ECCV_02.pdf

[13] A. V. Nefian, R. Grzeszczuk, and V. Eruhimov, *"A Statistical Upper Body Model for 3D Static and Dynamic Gesture Recognition From Stereo Sequences",* in Proc. Int. Conf. on Image Processing, 2001, vol. 2, pp. 286–289.

[14] Q. Delamarre and O. Faugeras, *"3D Articulate Models and Multi-View Tracking with Physical Forces",* CVIU journal, 2001, vol. 81, pp. 328–357. Disponible: http://www-sop.inria.fr/robotvis/personnel/qdelam/PUBLI.html

[15] I. Kakadiaris and D. Metaxas, *"Model-Based Estimation of 3D Human Motion",* IEEE Trans. Pattern Analysis and Machine Intelligence, Dec. 2000, vol. 22, no. 12, pp. 1453–1459.

[16] M. Mochimaru and Y. Yamazaki, *"The Three Dimensional Measurement of Unconstrained Motion Using a Model-Matching Method"*, Ergonomics, vol. 37, N 3, 1994, pp. 493–510.

[17] J. J. Kush and T. S. Huang, *"Vision Based Modeling and Tracking for Virtual Teleconferencing and Telecollaboration",* in Proc. ICCV, June 1995, pp. 666–671.

[18] J. Ohya and F. Kishino, *"Human Posture Estimation from Multiple Images Using Genetic Algorithm"*, in Proc. 12th IAPR, vol. I, pp. 750–753.

[19] D. M. Gavrila and L. S. Davis, *"3D Model Based Tracking of Humans in Action: a Multi-View Approach"*, in Proc. IEEE CVPR, San Francisco, CA, 1996, pp. 73–80.

[20] N. Shimada, Y. Shirai, Y. Kuno, and J. Miura, "Hand Gesture Estimation and Model Refinement Using Monocular Camera – Ambiguity Limitations by Inequality Constraints", in Proc. 3rd IEEE Int. Conf. on Automatic Face and Gesture Recognition, Nara, Japan, 1998, pp. 268–273.
Disponible: http://www-cv.mech.eng.osaka-u.ac.jp/~shimada/research/research2.html

[21] I. Yamato, I. Ohya and K. Ishii, "Recognizing Human Action In Time-Sequential Images Using Hidden Markov Models", in IEEE Conference on Computer Vision and Pattern Recognition, p 379-385, June 1992.

[22] A. Braffort, *"Reconnaissance et Compréhension des Gestes, Application à la Langue des Signes",* Thèse de Doctorat préparée au sein du LIMSI, Université Paris XI, soutenue le 28 juin 1996.

[23] T. Starner, J. Weaver and A.Pentland, *"Real Time American Sign Language Recognition Using Desk and Wearable Computer Based Video"*, IEEE Trans. On Pattern Analysis and Machine Intelligence, 20(12):1371-1375, December 1998.

[24] C. Vogler and D. Metaxas, "Adapting Hidden Markov Models for ASL Recognition by Using Three Dimensional Computer Vision Methods", In Proceedings of IEEE International Conference and Systems, Man, and Cybernetics, pp 156-161, Orlando (USA), 1997.

[25] H. Ouhaddi, "Contribution à l'Analyse de Gestes par Vision Monoscopique", Thèse de Doctorat préparée au sein de l'INT, Université Paris 6, soutenue le 20 octobre 1999.

[26] F. Lerasle, G. Rives, and M. Dhome, *"Human Body Limbs Tracking by Multi-Ocular Vision"*, in Actes 11ème congrès Reconnaissance des Formes et Intelligence Artificielle, vol. II, 1998, Clermont-Ferrand, France, pp. 193–199.

[27] N. Habili, *"Automatic Segmentation of the Face and Hand Sign Language Video Sequences"*, Technical Report, Dept of Electrical and Electron. Eng., Adelaide University, SA 5005, Australia, July 3 2001.
Disponible : http://www.eleceng.adelaide.edu.au/Personal/nhabili/Research.html 916–919

[28] S. Kewei, F. Xitian, C. Anni and S. Jingao, *"Automatic face Segmentation in YcrCb Images",* in Proc. 5th Asia-Pacific Conf. on Communications and 4th Optoelectronics and Communication Conf., vol 2, 1999, pp.

[29] M. Patridge and M. Jabri, *"Face Recognition Using a New Distance Metric"*, in Proc. IEEE Signal Processing Society Workshop Neural Networks for Signal ProcessingX, vol. 2, 2000, pp. 584-593.

[30] E. R. Davis, *"Machine Vision , Theory Algorithms Practicalities"*, 2nd edition, Academic Press, 1995.

[31] W. H Press, B. P. Flannery, S. A. Teykolsky and W. T. Vetterling, *"Numerical Recipes in C"*, Cambridge University Press, 1992.

[32] D. G. Lowe, *"Fitting Parameterized Three-Dimensional Models to Images"*, IEEE Trans. Pattern Analyses and Machine Intelligence, vol. 13, no. 5, may 1991, pp.441-450.

[33] "Les Signes de Mano", CD-Rom, IVT, ABAQUE –Micro, Je, Tu, Il.
Disponible : http://www.ivtcscs.org/produits

[34] M. Slimane, *"Les Chaînes de Markov Cachées: définitions, Algorithmes, Architectures"*, Université François Rabelais Tours.

[35] L. R. Rabiner, "*A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*", Proceedings of the IEEE, vol. 77, No. 2, February 1989.

[36] M. Fuentes, D. Mostefa, J. Khanoubi, S. Garcia-Salicetti, *"Vérification de l'Identité par Fusion de Données Biométriques : Signature en Ligne et Parole"*.

[37] M. Bomb and P. Horain, "3D Model Based Gesture Aquisition Using a Single Camera", IEEE WACV 02, Orlando FL, Dec. 3-4 2002.

Annexe:

Résultats du traitement du corpus vidéo :

A. Les quatre signeurs choisis pour la réalisation des gestes :

Merci à Bao Ly Van (d), Hichem Atti (b) et Waheb Larbi (c), également stagiaires au département EPH de l'INT, d'avoir bien voulu consacrer un peu de leur temps à la réalisation de ces séquences.


(a) : signeur 1

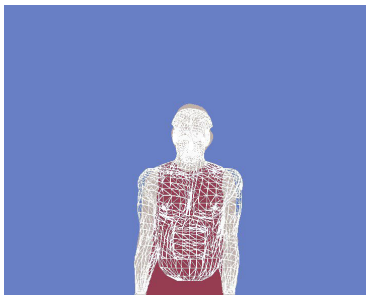
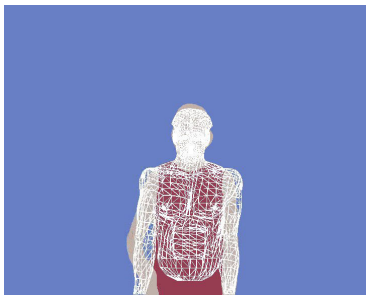
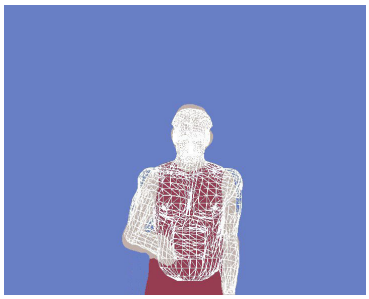
(b) : signeur 2

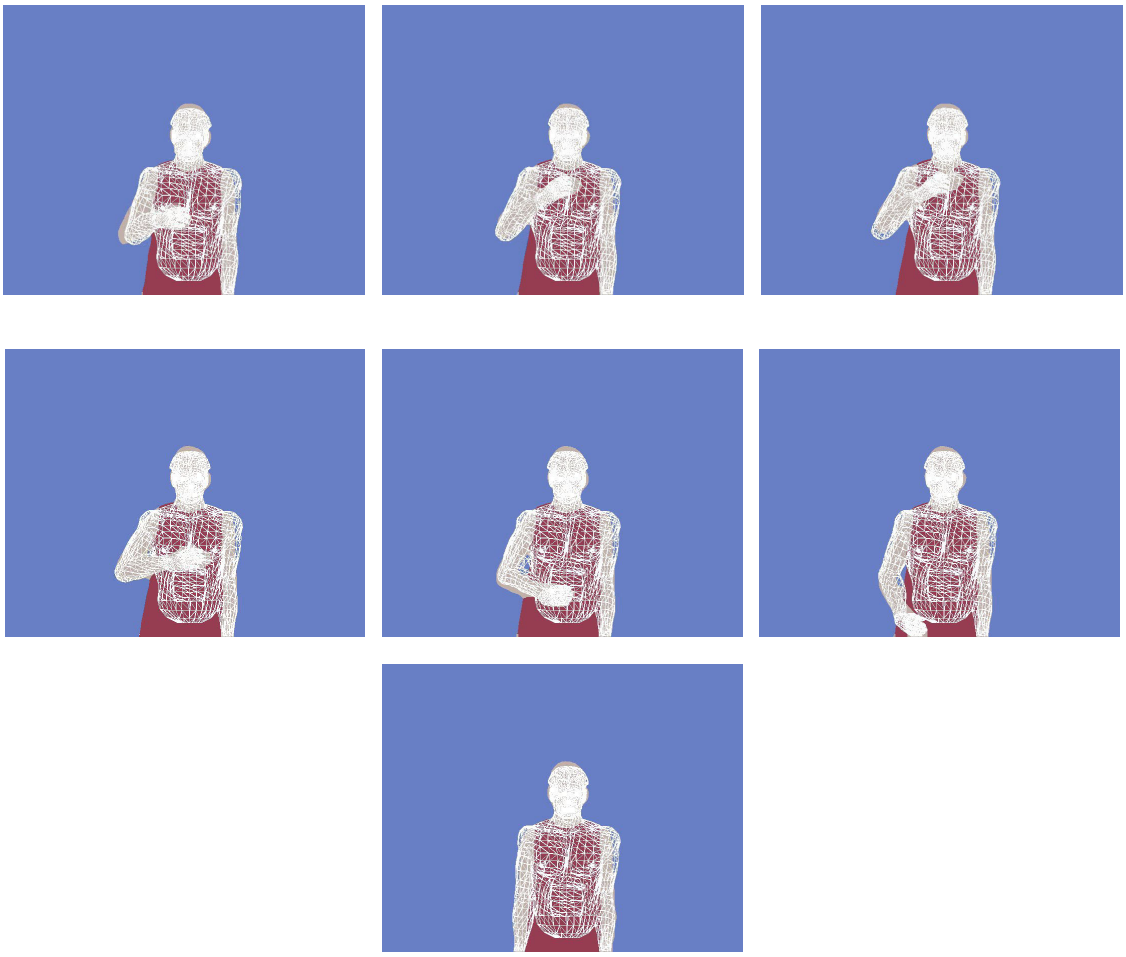
(c) : signeur 3

(d) : signeur 4

B. Résultats du traitement d'une séquence correspondant au mot Bon :

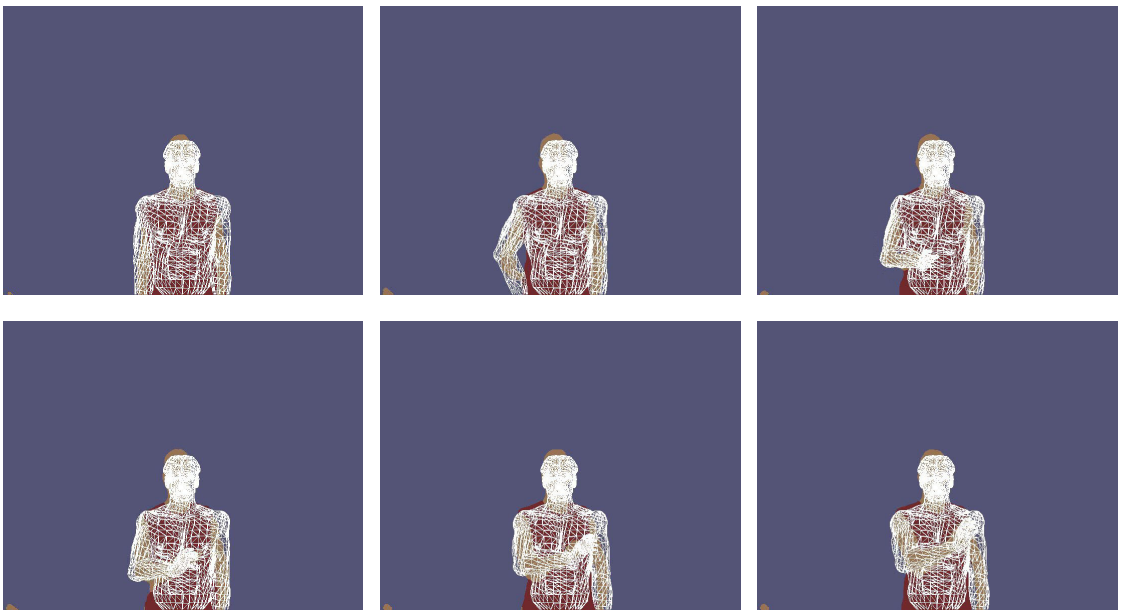
(signeur1)

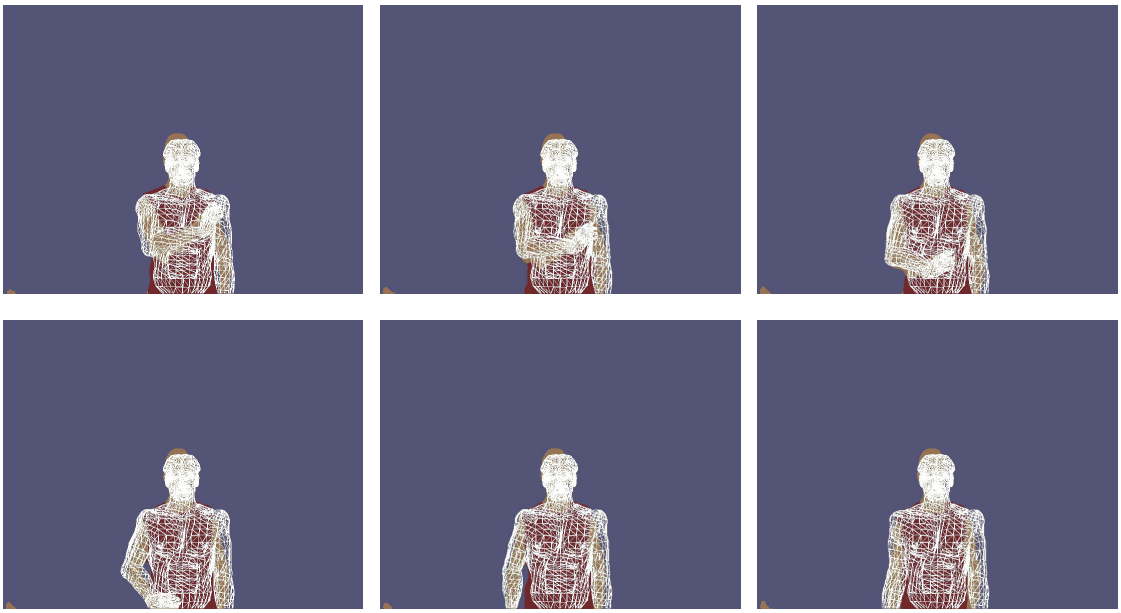






C. Résultats du traitement d'une séquence correspondant au mot Blessé :

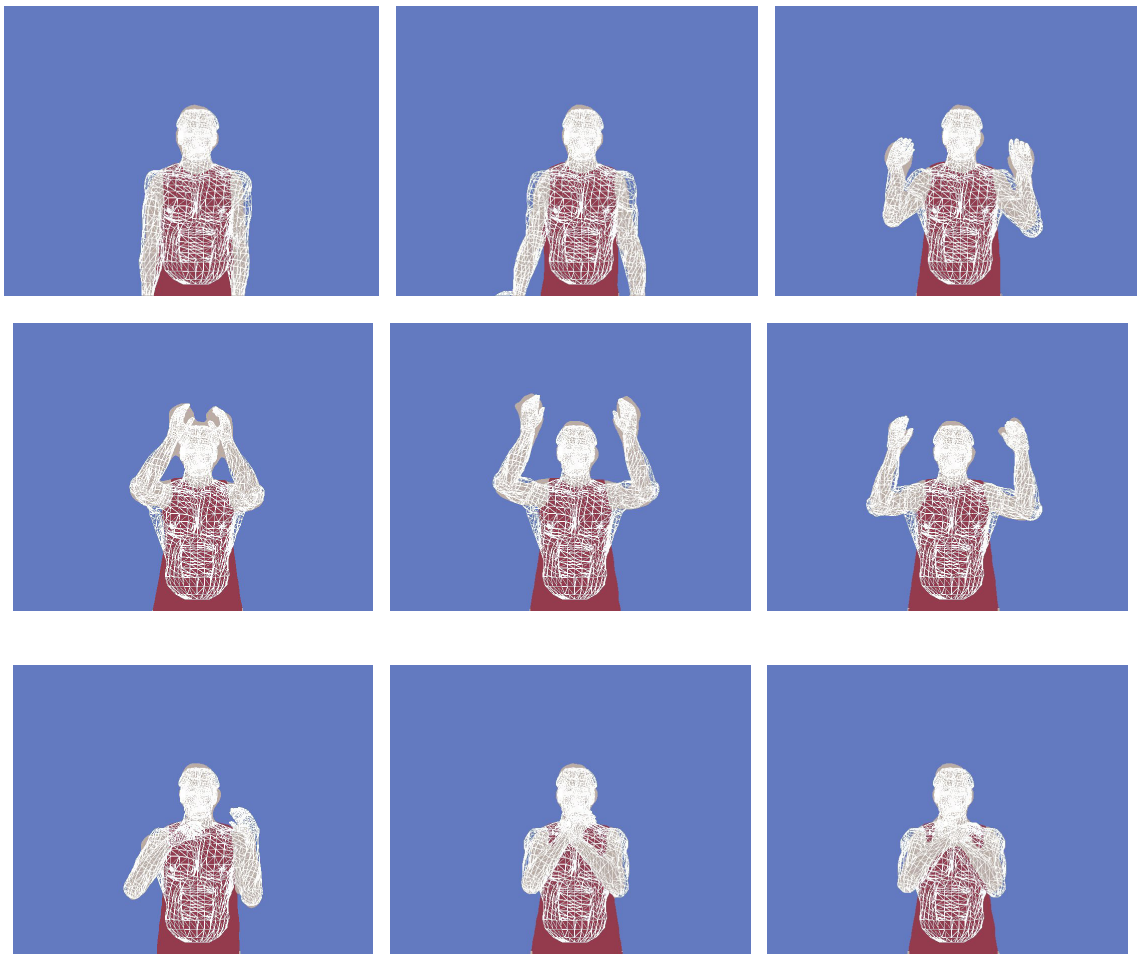
(signeur3)

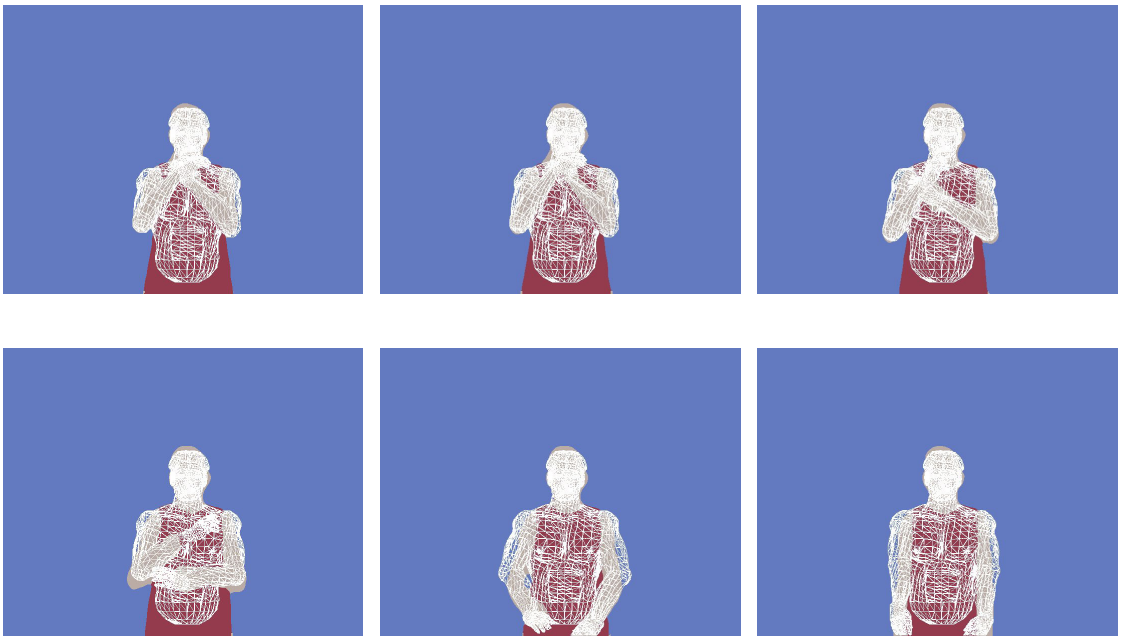




D. Résultats du traitement d'une séquence correspondant au mot Cheminée :

(signeur1)





E. Un exemple des fichiers de paramètres obtenus :

Les fichiers choisis pour illustrer ce point correspondent aux valeurs des positions du modèle 3D qui figurent dans l'illustration du recalage pour la séquence Bon. A chaque image correspond un fichier contenant le vecteur d'état du modèle.

```
Chest.trax : 31.000021
Chest.tray : 35.000034
Chest.traz : -0.000203
Chest.roty : -0.000000
Chest.rotx : -0.000002
Chest.rotz : 0.000002
Neck.rotz : -0.057843
Neck.roty : -0.021081
Neck.rotx : 0.175913
LeftUppArm.rotz : 0.029451
LeftUppArm.roty : -0.393129
LeftUppArm.rotx : -0.018847
RightUppArm.rotz : 0.032397
RightUppArm.roty : 0.497700
RightUppArm.rotx : -0.002112
LeftForArm.rotx : -0.217673
RightForArm.rotx : -0.309009
LeftHand.rotz : -0.090714
LeftHand.roty : -0.045070
LeftHand.rotx : 0.077811
RightHand.rotz : 0.024374
RightHand.roty : 0.097236
RightHand.rotx : 0.028512
Chest.cameraScale : 1.900297
```

```
Chest.trax : 31.000021
Chest.tray : 35.000034
Chest.traz : -0.000203
Chest.roty : -0.000000
Chest.rotx : -0.000002
Chest.rotz : 0.000002
Neck.rotz : -0.057120
Neck.roty : -0.018315
Neck.rotx : 0.174580
LeftUppArm.rotz : 0.029451
LeftUppArm.roty : -0.393129
LeftUppArm.rotx : -0.018847
RightUppArm.rotz : 0.145223
RightUppArm.roty : -0.562325
RightUppArm.rotx : 0.022174
LeftForArm.rotx : -0.217673
RightForArm.rotx : -0.044582
LeftHand.rotz : -0.090714
LeftHand.roty : -0.045070
LeftHand.rotx : 0.077811
RightHand.rotz : 0.152206
RightHand.roty : -0.132417
RightHand.rotx : -0.173686
Chest.cameraScale : 1.900297
```

```
Chest.trax : 31.000021
Chest.tray : 35.000034
Chest.traz : -0.000203
Chest.roty : -0.000000
Chest.rotx : -0.000002
Chest.rotz : 0.000002
Neck.rotz : -0.057364
Neck.roty : -0.019248
Neck.rotx : 0.174977
LeftUppArm.rotz : 0.029451
LeftUppArm.roty : -0.393129
LeftUppArm.rotx : -0.018847
RightUppArm.rotz : 0.320144
RightUppArm.roty : -1.207751
RightUppArm.rotx : 0.155950
LeftForArm.rotx : -0.217673
RightForArm.rotx : -1.380397
LeftHand.rotz : -0.090714
LeftHand.roty : -0.045070
LeftHand.rotx : 0.077811
RightHand.rotz : 0.638526
RightHand.roty : 0.062158
RightHand.rotx : 0.202651
Chest.cameraScale : 1.900297
```

```
Chest.trax : 31.000021
Chest.tray : 35.000034
Chest.traz : -0.000203
Chest.roty : -0.000000
Chest.rotx : -0.000002
Chest.rotz : 0.000002
Neck.rotz : -0.058321
Neck.roty : -0.054090
Neck.rotx : 0.158751
LeftUppArm.rotz : 0.029451
LeftUppArm.roty : -0.393129
LeftUppArm.rotx : -0.018847
RightUppArm.rotz : 0.314414
RightUppArm.roty : -0.283719
RightUppArm.rotx : 0.174340
LeftForArm.rotx : -0.217673
RightForArm.rotx : -2.054288
LeftHand.rotz : -0.090714
LeftHand.roty : -0.045070
LeftHand.rotx : 0.077811
RightHand.rotz : 0.643854
RightHand.roty : 0.125392
RightHand.rotx : 0.344717
Chest.cameraScale : 1.900297
```

```
Chest.trax : 31.000021
Chest.tray : 35.000034
Chest.traz : -0.000203
Chest.roty : -0.000000
Chest.rotx : -0.000002
Chest.rotz : 0.000002
Neck.rotz : -0.028535
Neck.roty : -0.036267
Neck.rotx : 0.125530
LeftUppArm.rotz : 0.044720
LeftUppArm.roty : -0.526167
LeftUppArm.rotx : -0.021288
RightUppArm.rotz : 0.469517
RightUppArm.roty : -0.404508
RightUppArm.rotx : 0.173802
LeftForArm.rotx : -0.197907
RightForArm.rotx : -2.497141
LeftHand.rotz : -0.556512
LeftHand.roty : -0.024021
LeftHand.rotx : 0.030222
RightHand.rotz : 0.634377
RightHand.roty : -0.116765
RightHand.rotx : 0.115587
Chest.cameraScale : 1.900297
```

```
Chest.trax : 31.000021
Chest.tray : 35.000034
Chest.traz : -0.000203
Chest.roty : -0.000000
Chest.rotx : -0.000002
Chest.rotz : 0.000002
Neck.rotz : -0.025896
Neck.roty : -0.034596
Neck.rotx : 0.140561
LeftUppArm.rotz : 0.044720
LeftUppArm.roty : -0.526167
LeftUppArm.rotx : -0.021288
RightUppArm.rotz : 0.524447
RightUppArm.roty : -0.322705
RightUppArm.rotx : 0.172968
LeftForArm.rotx : -0.197907
RightForArm.rotx : -2.564801
LeftHand.rotz : -0.556512
LeftHand.roty : -0.024021
LeftHand.rotx : 0.030222
RightHand.rotz : 0.644395
RightHand.roty : -0.037676
RightHand.rotx : 0.137786
Chest.cameraScale : 1.900297
```

```
Chest.trax : 31.000021
Chest.tray : 35.000034
Chest.traz : -0.000203
Chest.roty : -0.000000
Chest.rotx : -0.000002
Chest.rotz : 0.000002
Neck.rotz : -0.012599
Neck.roty : 0.012334
Neck.rotx : 0.101205
LeftUppArm.rotz : 0.060241
LeftUppArm.roty : -0.528514
LeftUppArm.rotx : -0.016307
RightUppArm.rotz : 0.528104
RightUppArm.roty : -0.657808
RightUppArm.rotx : 0.173083
LeftForArm.rotx : -0.225838
RightForArm.rotx : -2.173359
LeftHand.rotz : -0.116630
LeftHand.roty : -0.016368
LeftHand.rotx : -0.007606
RightHand.rotz : 0.573946
RightHand.roty : -0.035537
RightHand.rotx : 0.188728
Chest.cameraScale : 1.900297
```

```
Chest.trax : 31.000021
Chest.tray : 35.000034
Chest.traz : -0.000203
Chest.roty : -0.000000
Chest.rotx : -0.000002
Chest.rotz : 0.000002
Neck.rotz : -0.012743
Neck.roty : 0.010430
Neck.rotx : 0.100220
LeftUppArm.rotz : 0.060606
LeftUppArm.roty : -0.531425
LeftUppArm.rotx : -0.016317
RightUppArm.rotz : 0.423429
RightUppArm.roty : -0.739833
RightUppArm.rotx : 0.155166
LeftForArm.rotx : -0.226418
RightForArm.rotx : -1.558169
LeftHand.rotz : -0.105506
LeftHand.roty : -0.010512
LeftHand.rotx : -0.041552
RightHand.rotz : 0.066016
RightHand.roty : 0.006073
RightHand.rotx : -0.066578
Chest.cameraScale : 1.900297
```

```
Chest.trax : 31.000021
Chest.tray : 35.000034
Chest.traz : -0.000203
Chest.roty : -0.000000
Chest.rotx : -0.000002
Chest.rotz : 0.000002
Neck.rotz : -0.012550
Neck.roty : 0.011816
Neck.rotx : 0.100105
LeftUppArm.rotz : 0.060606
LeftUppArm.roty : -0.531425
LeftUppArm.rotx : -0.016317
RightUppArm.rotz : 0.350065
RightUppArm.roty : -0.461015
RightUppArm.rotx : 0.153801
LeftForArm.rotx : -0.226418
RightForArm.rotx : -1.020386
LeftHand.rotz : -0.105506
LeftHand.roty : -0.010512
LeftHand.rotx : -0.041552
RightHand.rotz : 0.511987
RightHand.roty : 0.033549
RightHand.rotx : -0.067263
Chest.cameraScale : 1.900297
```

```
Chest.trax : 31.000021
Chest.tray : 35.000034
Chest.traz : -0.000203
Chest.roty : -0.000000
Chest.rotx : -0.000002
Chest.rotz : 0.000002
Neck.rotz : -0.028651
Neck.roty : 0.005319
Neck.rotx : 0.124257
LeftUppArm.rotz : 0.060606
LeftUppArm.roty : -0.531425
LeftUppArm.rotx : -0.016317
RightUppArm.rotz : 0.106761
RightUppArm.roty : -0.243659
RightUppArm.rotx : 0.158361
LeftForArm.rotx : -0.226418
RightForArm.rotx : -0.002904
LeftHand.rotz : -0.105506
LeftHand.roty : -0.010512
LeftHand.rotx : -0.041552
RightHand.rotz : 0.148873
RightHand.roty : -0.004312
RightHand.rotx : -0.036475
Chest.cameraScale : 1.900297
```