

# Analyse de trajectoires pour la synthèse d'un mouvement de marche

## Trajectory analysis for synthetizing walking animation

Alexandru-Robert-Ciprian Guduvan<sup>1</sup>

Vincent Charvillat<sup>1</sup>

Géraldine Morin<sup>1</sup>

<sup>1</sup> IRIT - Université de Toulouse

2 rue Camichel, F-31071 Toulouse

alexandru.guduvan@etu.enseiht.fr, vincent.charvillat@enseiht.fr geraldine.morin@enseiht.fr,

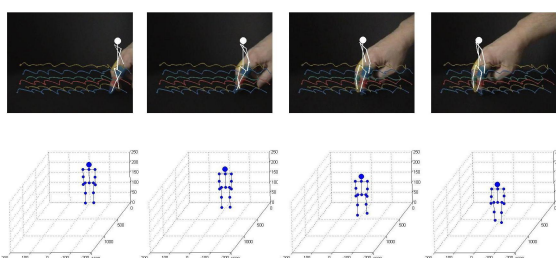


FIG. 1 – Analyse des trajectoires de doigts pour paramétrer un modèle d'animation 3D et la synthèse du mouvement associé.

### Résumé

Cet article présente l'animation d'un modèle de personnage contrôlé par des paramètres issus de l'analyse de mouvement de doigts préalablement colorés dans une vidéo. L'intérêt de cette approche est la facilité de la mise en oeuvre, et la faible complexité des algorithmes proposés. Pour le traitement de la vidéo, une analyse colorimétrique des trames isole des régions de couleur. Les données manquantes sont reconstruites et des trajectoires lisses correspondant au mouvement continu des doigts sont déduites. Les paramètres de mouvement sont estimés et utilisés pour définir le mouvement d'un personnage 3D.

### Mots Clef

Analyse de vidéo, synthèse de mouvement, trajectoires.

### Abstract

This article presents the animation of a character model controlled by parameters extracted from the analysis of the movement of fingers in a video. The interest of this approach is its ease of implementation and the limited complexity of the algorithms. For the video-processing a colorimetric analysis of each frame isolates regions of different colors. Missing data is reconstructed and smooth trajectories corresponding to the continuous motion of the fingers are deduced. We infer the parameters for the bipedal movement and use them to define the walk of a 3D character.

### Keywords

Video analysis, motion synthesis, trajectories.

## 1 Introduction

La détection des gestes dans un fichier vidéo, l'analyse des caractéristiques du geste, puis l'utilisation des paramètres déterminés (par exemple : la forme du geste, la vitesse) pour la synthèse d'un mouvement de marche 3D, représentent un domaine d'actualité qui a des applications dans l'industrie cinématographique et celle des jeux informatiques. Une première approche utilise des systèmes de "motion capture" [AJ05], que l'on retrouve notamment dans l'industrie cinématographique, un personnage 3D reproduit exactement les données capturées par le système. Tournier et al. [TWC\*08], d'une part, et Arikan [Ari06], d'autre part, proposent deux méthodes pour compresser les données acquises par la capture de mouvement. La difficulté principale de cette approche repose sur le matériel lourd utilisé, donc nous proposons une alternative en faisant appel à l'analyse de séquences vidéo. Deux approches de génération de mouvement sont proposées : la première consiste à utiliser les trajectoires trouvées directement pour l'animation d'un personnage (comme dans [BH07]). La deuxième approche utilise des caractéristiques essentielles du geste pour paramétrer un modèle d'animation. Ce modèle est basé sur le principe du pendule simple pour simuler le mouvement des articulations les unes par rapport aux autres. Sa simplicité par rapport aux modèles classiques, qui proviennent d'une analyse physique du mouvement humain [BC89], le rend idéal pour des applications en temps réel.

Par rapport aux travaux précédents, cet article propose l'analyse ainsi que la synthèse du mouvement afin de créer un système complet de reconnaissance de gestes, leur analyse et la production de l'animation correspondante. Les résultats montrent que la méthode trajectographique est précise et que le modèle d'animation est d'une part une bonne approximation du mouvement naturel et d'autre part facile à contrôler par un nombre réduit de paramètres.

Dans la suite, la section 2 présente le traitement de la vidéo comprenant la gestuelle par analyse chromatique (détermination des régions de couleur comme dans [GH96]). En section 3, nous construisons les trajectoires des centres des régions identifiées en utilisant des courbes de Bézier de degré variable. Ensuite nous utilisons un modèle de personnage 3D paramétré par les caractéristiques de la gestuelle afin de produire un mouvement de synthèse.

## 2 Analyse de la vidéo et éléments de trajectographie

### 2.1 Format des données d'entrée

La reconnaissance du mouvement dans un fichier vidéo comprend l'analyse de chacune de ses trames au niveau des pixels (cf. figure 7 où nous décrivons l'algorithme utilisé). Pour identifier chaque geste de la main, les doigts dans la vidéo ont été préalablement séparés (segmentés) en régions de couleur (cf. figure 2). Pour simplifier l'acquisition des données il est possible d'utiliser un gant avec la même construction [GH96] ou d'utiliser une méthode de détection de la peau [VSA03]. La séparation des doigts en régions facilite la reconnaissance du geste. Dans une première approche nous considérons les régions des doigts et nous leur faisons correspondre les régions de la jambe d'un personnage virtuel qui va suivre le mouvement exact des doigts.

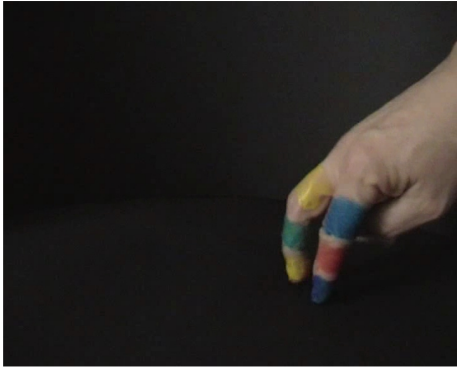


FIG. 2 – Trame dans la vidéo.

Dans la suite du papier une vidéo  $V$  est considérée comme un ensemble de  $D$  trames ordonnées :

$$V = \{T_n \mid n \in \mathbb{N}^*, n \leq D\} \quad (1)$$

où  $T_n$  représente la  $n$ -ième trame de la vidéo.

### 2.2 Prétraitement de la vidéo et analyse chromatique

Pour une trame donnée  $T_i$ , nous nous sommes intéressés à la détection des régions de couleur de la main. A partir de ces positions, nous pourrions identifier le geste et déterminer ses caractéristiques.

Pour faciliter le processus d'identification des régions de couleur et de leur centre, la vidéo subit un prétraitement. La première étape représente l'application d'un filtre Gaussien à chaque trame de la vidéo, pour éliminer le bruit et homogénéiser les zones de couleur.

Nous avons choisi d'utiliser les composantes RGB pour permettre un traitement direct sur les composantes RGB des trames. Ceci permet de restreindre le nombre de calculs afin de garder un aspect temps réel. Il aurait été possible d'envisager l'utilisation d'autres espaces de couleurs (comme dans [TC92] ou encore [TLF95]). Dans la plupart des vidéos les variations de couleurs sont dues aux conditions dynamiques de la capture, aux variations de l'éclairage ou aux ombres. Nous utilisons la représentation normalisée des couleurs. Si dans une trame un pixel a dans l'espace RGB les composantes  $(R, G, B)$ , alors après la normalisation elles seront :

$$(R^n, G^n, B^n) = \left( \frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B} \right). \quad (2)$$

Tenant compte du fait qu'une des nouvelles composantes peut être déduite de la propriété :  $R^n + G^n + B^n = 1$ , seules deux composantes suffisent.

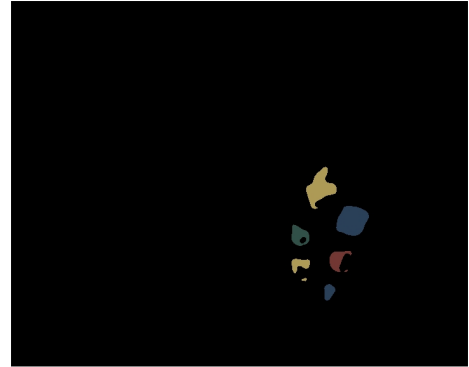


FIG. 3 – Masque de séparation pour une trame (identification des régions de couleur).

Des informations supplémentaires sur les couleurs à reconnaître sont données en entrée : le nombre de couleurs, leur codage RGB et le nombre de régions de chaque couleur. En mode d'apprentissage supervisé, on sélectionne un nombre significatif de trames uniformément distribuées dans la vidéo et l'utilisateur peut identifier les régions de couleur (dans notre cas : rouge, verte, bleu et jaune). Nous calculons ensuite la moyenne de chacune des couleurs, que l'on utilise pour identifier les régions dans la vidéo, en comparant la différence entre la couleur de chaque pixel et les couleurs moyennes à un certain seuil, comme proposé dans [OKS80]. Ce procédé nous offre pour chacune des couleurs un masque qui montre la zone qu'elle occupe dans la trame (cf. figure 3).

Le centre d'une région est défini comme le point moyen des pixels de sa couleur dans le masque. Dans le cas où

une certaine couleur ne se retrouve pas dans le masque, on l'ignore et la position du centre de la région correspondante sera approximée par une moyenne des valeurs de  $k$  plus proches voisins (cf. sous-section 2.4).

Après l'application des principes décrits ci dessus, pour chaque région, nous disposons d'un ensemble de paires  $(x_n, y_n)$  indiquant la position du centre de la région dans chacune des trames  $T_n$  (cf. figure 4).

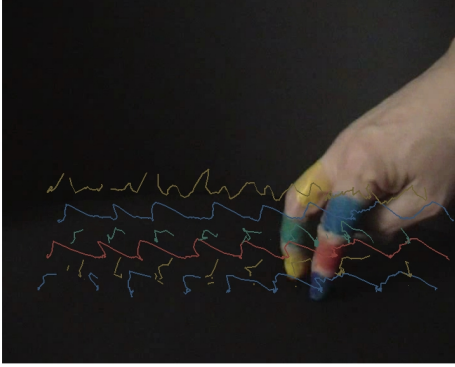


FIG. 4 – Centres des régions de couleurs identifiées après le traitement complet de la vidéo. Des données sont manquantes, en particulier à cause de problèmes d'occultation.

### 2.3 Détermination robuste des régions

La détection des régions et donc le calcul de leurs centres peut manquer de précision. Ceci peut être dû à la mauvaise détection des couleurs (par exemple, si les couleurs sont proches). Deux techniques permettent d'améliorer ces calculs. La première utilise l'élimination des points isolés. Les points isolés sont éliminés, considérant qu'ils font partie d'une mauvaise détection des couleurs. La deuxième méthode consiste à imposer une limite sur la vitesse des gestes. Si dans deux trames successives (correspondant à deux instants consécutifs), la position d'un centre varie trop, la position dans la deuxième trame est ignorée. C'est à dire, si dans une trame  $T_i$ , on identifie correctement le centre de la région jaune en haut puis, dans la trame  $T_{i+1}$ , à cause d'une mauvaise détection de la couleur de la peau, le centre se déplace vers le poignet, la distance entre les deux centres peut être trop importante, auquel cas la donnée aberrante est éliminée.

### 2.4 Remplissage de l'ensemble des données

Parfois des régions de couleur sont complètement ou partiellement occultées (quand elles sortent de l'écran ou qu'elles sont cachées par un autre objet). Dans ce cas, par l'élimination des données erronées, on obtient un ensemble incomplet de centres. Pour éliminer le risque d'obtenir des centres approximés faux, et régulariser les variations, des nouvelles données sont insérées. Les positions manquantes sont estimées par la méthode des  $k$  plus proches voisins. Pour une région dans une trame  $T_i$ , on suppose

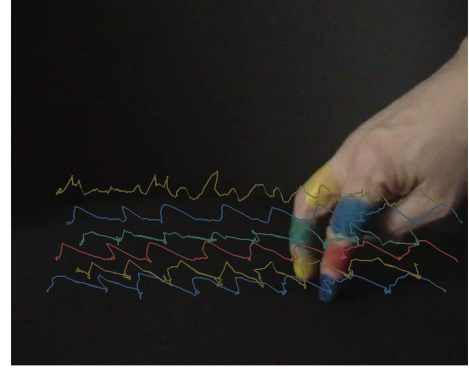


FIG. 5 – Les centres des régions après le remplissage.

que la détection du centre a échoué. On remplit alors la valeur de son centre  $C_i$  avec la moyenne des valeurs des  $k$  plus proches centres bien détectés. Cette méthode tiendra compte des données déjà insérées pour le remplissage des données manquantes suivantes. Ceci assure un aspect continu tout en évitant les redondances. En effet plusieurs centres non-détectés sont approximés avec des valeurs différentes. Cette étape est nécessaire car l'estimation des paramètres des courbes de Bézier menée ultérieurement est sensible aux endroits où la densité des points d'apprentissage n'est pas suffisante (comme dans l'ensemble de données brutes). Sans cette étape de remplissage des données, l'approximation des trajectoires par des courbes de Bézier donne des résultats inutilisables : la courbe approximante oscille fortement dans les zones où les données manquent.

### 2.5 Approximation des trajectoires

Pour tracer une approximation des trajectoires suivies par chacune des régions de couleur, nous utilisons un modèle polynomial, sous forme de courbes de Bézier. Arïkan utilise ce même modèle pour compresser des bases de données issues de la capture de mouvement [Ari06]. Notre but est de lisser les trajectoires brutes (éliminer leurs variations locales) afin d'offrir au mouvement un aspect régulier.

Une courbe de Bézier de degré  $d$  avec l'ensemble  $P$  des  $d + 1$  points de contrôle est décrite de la manière suivante :

$$B(t, P) = \sum_{i=0}^d PC_i b_{i,d}(t) \quad (3)$$

où  $PC_i \in P$  est le  $i$ ème point de contrôle et  $b_{i,d}$  le  $i$ ème polynôme de la base de Bernstein de degré  $d$ , qui s'écrit :

$$b_{i,d}(t) = \binom{d}{i} t^i (1-t)^{d-i}. \quad (4)$$

Chaque centre  $C_i$  d'une région a deux composantes  $(x_i, y_i)$  qui définissent sa position dans la trame  $T_i$ . On considère l'ensemble  $C = \{(i, (x_i, y_i)) \mid i = 1..D\}$ . On calcule par régression la courbe de Bézier qui est la meilleure approximation de ces données, au sens des moindres carrés, c'est

à dire, qui minimise le critère :

$$\sum_i \|B(i, P) - (x_i, y_i)\|^2. \quad (5)$$

Pour avoir une valeur précise de la qualité de l'approximation et de sa capacité de généralisation, on procède de la manière suivante : comme cet ensemble est de grande taille, il est séparé en un ensemble d'apprentissage et un ensemble de test (éventuellement de la même taille, tenant compte des indices pairs et impairs par exemple, ou d'un autre ensemble d'indices  $I$ ) :  $C^{app} = \{(i, (x_i, y_i)) \mid i \in I\}$ ,  $C^{test} = \{(i, (x_i, y_i)) \mid i \notin I\}$ . L'ensemble d'apprentissage est utilisé pour déterminer des courbes de Bézier de divers degrés ( $d$ ) ou, plus spécifiquement, leurs paramètres : les points de contrôle ( $PC_i, i = 1..d + 1$ ). Pour chaque degré entre  $d_{min}$  et  $d_{max}$ , on calcule l'ensemble des points de contrôle solution du problème d'estimation suivant :

$$\hat{P}_d = \underset{P_c}{\operatorname{argmin}} \frac{1}{N} \sum_{i \in I} \|(x_i, y_i) - B(i, P_d)\|^2, N = \operatorname{card}(I). \quad (6)$$

Parmi les courbes de divers degrés, on choisit celle qui a la meilleure capacité de généralisation, c'est-à-dire celle qui obtient la valeur la plus basse de l'erreur de généralisation pour l'ensemble de test (la distance entre les valeurs de l'ensemble de test et celles prédites) :

$$Err_{gen} = \frac{1}{M} \sum_{i \notin I} \|x_i - B(i, P_d)\|^2, M = D - N \quad (7)$$

On notera que le degré retenu n'est pas toujours le plus élevé, mais celui qui permet la meilleure adaptation de la courbe aux données. Les courbes de Bézier obtenues offrent un ensemble de données :  $C^{pred}$  avec les coordonnées des centres prédites des régions de couleur pour chaque trame (cf. figure 6). Ces trajectoires sont ensuite utilisées pour induire un mouvement (première approche) ou paramétrer le mouvement d'un personnage (deuxième approche) (cf. figure 8).

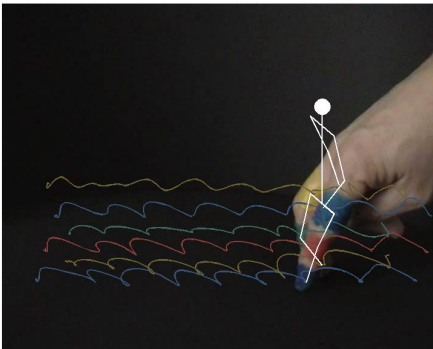


FIG. 6 – La première approche : influence directe du personnage par le geste à l'aide des courbes de Bézier.

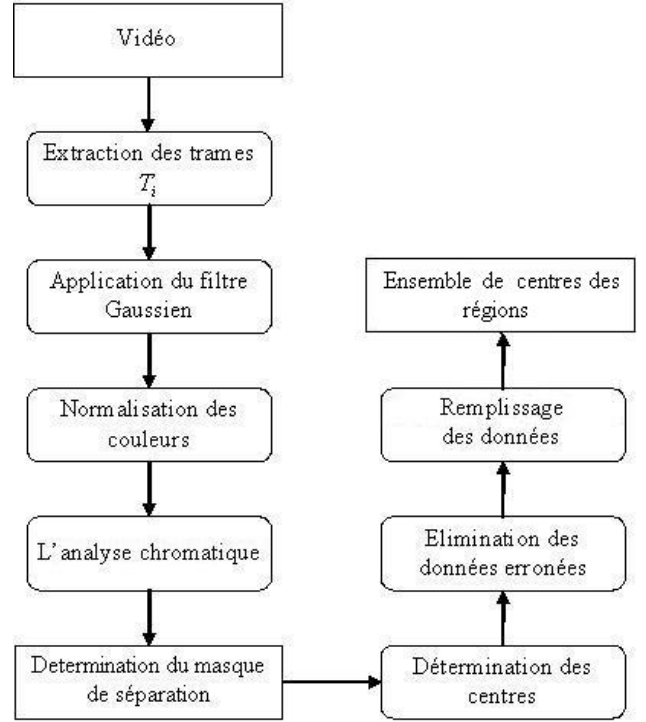


FIG. 7 – Schéma des procédures proposées pour l'analyse des trames.

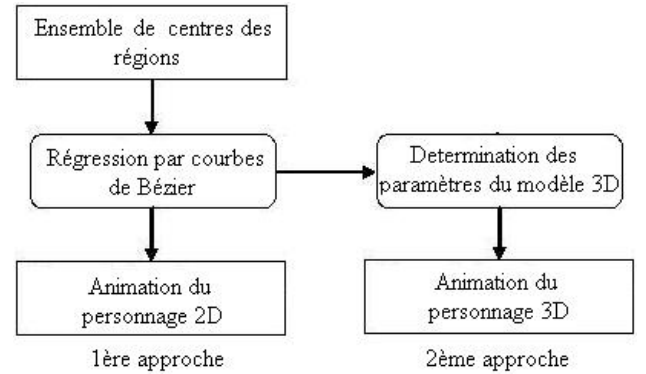


FIG. 8 – Schéma des procédures proposées pour la synthèse du mouvement du personnage.

## 3 Modèle d'animation

### 3.1 Construction du personnage

Les éléments constitutifs du personnage sont (cf. figure 9) :

- 1 *Elément\_Tête*,
- 1 *Elément\_Cou*,
- 1 *Elément\_Bassin*,
- 2 × *Elément\_Coude*,
- 2 × *Elément\_Main*,
- 2 × *Elément\_Genou*,
- 2 × *Elément\_Chevilles*.

Dans ce modèle il n'y a pas d'éléments correspondant aux épaules et les hanches. Ces deux composantes ana-

tomiques sont modélisées à partir de l'*Elément\_Cou* et l'*Elément\_Bassin* respectivement, grâce à leur alignement.



FIG. 9 – Le modèle utilisé pour le personnage 3D.

Le mouvement des éléments du personnage se fait dans des plans parallèles au plan XOY, en gardant les composantes Z constantes dans le temps. Le personnage avance / recule dans la direction de l'axe OX.

### 3.2 Animation directe du personnage

On utilise directement les trajectoires reconstruites pour animer un personnage dont les jambes suivent les centres des régions de couleur. Les bras ont été animés de façon cohérente par rapport au mouvement des jambes : le bras gauche suit le mouvement de la jambe droite et le bras droit suit le mouvement de la jambe gauche -par symétrie par rapport à l'axe OY pour rendre l'animation la plus naturelle possible.

### 3.3 Modèle de mouvement paramétré

L'élément principal du personnage est l'*Elément\_Bassin* par rapport auquel les positions de tous les autres éléments sont calculées. Les composantes : *Elément\_Bassin*, *Elément\_Cou* et *Elément\_Tête* modifient leur composante X dans le cas du mouvement dans la scène (en temps fini) ou restent constantes dans le cas du mouvement infini.

Pour simuler un mouvement naturel du personnage et garder les contraintes imposées par son squelette (longueur constante des composantes : bras, jambes) le mouvement a été produit comme la somme des changements des positions des composantes les unes par rapport aux autres selon le modèle suivant :

- l'*Elément\_Main* change sa position tenant compte de la position de l'*Elément\_Coude* qui à son tour est influencé par la position de l'*Elément\_Cou*,
- similairement l'*Elément\_Cheville* change sa position tenant compte de la position de l'*Elément\_Genou* qui à son tour est influencé par la position de l'*Elément\_Bassin*.

### Le principe du pendule double

Chacun des éléments mentionnés effectue le même mouvement paramétré différemment (modèle du "pendule simple" [Moj03]) par rapport à l'élément le plus proche qui l'influence, comme dans le schéma (cf. figure 10). La somme de ces mouvements impose au personnage un mouvement selon le modèle d'un ensemble de "pendules doubles".

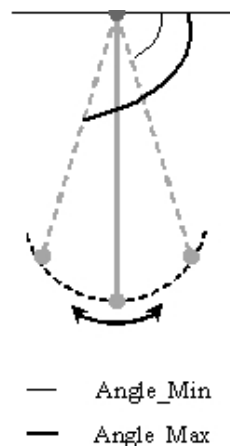


FIG. 10 – Principe du "pendule simple".

Par exemple : l'*Elément\_Main* effectue un mouvement "pendule simple" par rapport à l'*Elément\_Coude* qui à son tour effectue le même genre de mouvement par rapport à l'*Elément\_Cou*. Donc l'*Elément\_Main* et l'*Elément\_Coude* ensemble suivent un mouvement de "pendule double" par rapport à l'*Elément\_Cou*.

Le mouvement type "pendule simple" est caractérisé par trois paramètres :

- *Angle\_Min* : l'angle minimal fait par le segment qui joint les deux éléments du pendule avec l'horizontale pendant le mouvement,
- *Angle\_Max* : l'angle maximal fait par le segment qui joint les deux éléments du pendule avec l'horizontale pendant le mouvement,
- *Période* : l'intervalle de temps pendant lequel le pendule partant d'une de ses positions extrêmes (*Angle\_Min* ou *Angle\_Max*) passe par l'autre position extrême et finalement arrive dans son état initial.

Pour simplifier : dans le système d'un pendule simple, un *Elément\_1* effectue un mouvement par rapport à un *Elément\_2* qui représente le parcourt d'un arc de cercle centré dans la position de l'*Elément\_2*, avec un rayon égal à la distance euclidienne entre les deux éléments (R). Le cercle sera parcouru entre *Angle\_Min* et *Angle\_Max*, donc l'arc aura  $\Delta Angles = (Angle\_Max - Angle\_Min)$  degrés. Ce parcours sera fait pendant l'intervalle  $\frac{Période}{2}$  dans un sens puis dans l'autre sens pendant la même durée.

Les équations qui gèrent le mouvement pendule simple dans le temps sont :

$$\begin{aligned}
\text{Elément}_1 &= \\
&[\text{Elément}_2(1) + R \cdot \cos(\text{Angle\_Min} + \Delta\text{Angles} \cdot \Delta t), \\
&\text{Elément}_2(2) + R \cdot \sin(\text{Angle\_Min} + \Delta\text{Angles} \cdot \Delta t), \\
&\text{Elément}_2(3)]
\end{aligned}$$

### 3.4 Paramètres de contrôle

Pour animer le personnage, nous utiliserons trois paramètres : *Vitesse*, *Période* et *Saut*. Le premier est un réel qui prend des valeurs dans l'intervalle  $[0, 1]$  et le troisième un booléen qui prend la valeur 1 quand le personnage doit effectuer un saut. La valeur du paramètre *Période* est déduite en analysant les courbes de Bézier. Nous observons que le mouvement des jambes est périodique dans le temps (cf. figure 11). De plus, des informations sur cette périodicité peuvent être déduites facilement en regardant l'ensemble des centres des régions prédits sur l'axe OY. La tangente à une courbe de Bézier s'exprime simplement car sa dérivée est une courbe de Bézier de degré  $d - 1$  dont les points de contrôle sont les différences des points de contrôle successifs (voir par exemple [Far01]) :

$$B'(t, P) = \sum_{i=0}^{d-1} (PC_{i+1} - PC_i) b_{i,d-1}(t). \quad (8)$$

Les valeurs de la *Période* au cours du temps sont données par la détection des endroits où l'ordonnée de la tangente change de signe. Pour cela, il est nécessaire que l'ordonnée des différences des points de contrôle change de signe. En utilisant cette *Période* déduite du mouvement, et qui est corrélée avec la *Vitesse*, nous obtenons un mouvement très naturel du personnage 3D.

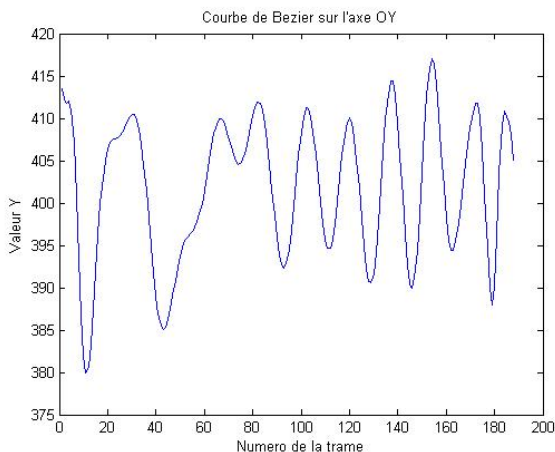


FIG. 11 – Exemple de courbe de Bézier qui montre la périodicité du mouvement sur l'axe OY.

Le paramètre *Vitesse* influence les deux autres de la manière suivante :

- *Angle\_Min* a une variation inversement proportionnelle à celle de *Vitesse* (quand la vitesse augmente le personnage fait des pas en avant plus longs),
- *Angle\_Max* a une variation directement proportionnelle à celle de *Vitesse*.

C'est à dire que le mouvement des bras et des jambes du personnage devient de plus en plus ample avec l'augmentation de *Vitesse*, en respectant la réalité physique. Le paramètre *Saut* n'influence que la position des composantes du squelette du personnage sur l'axe OY. Il va imprimer au personnage, en plus de son mouvement le long de l'axe OX, une translation sur OY pendant une durée prédéterminée *Durée\_Saut*.

Nous avons établi deux états du personnage : un état Repos (qui correspond à une vitesse nulle) et un état Courir (qui correspond à la valeur 1 de la vitesse) (cf. table 1). Les états intermédiaires sont obtenus par interpolation linéaire quand *Vitesse* parcourt l'intervalle  $[0, 1]$  :

$$L'\text{état\_Intermédiaire} = (1 - \text{Vitesse}) \cdot L'\text{état\_Repos} + \text{Vitesse} \cdot L'\text{état\_Courir} \quad (9)$$

TAB. 1 – Les deux états significatifs du personnage.

Paramètres	L'état Repos	L'état Courir
<i>Angle_Genou_Min</i>	$\pi/2$	$-0.20\pi$
<i>Angle_Genou_Max</i>	$\pi/2$	$-0.70\pi$
<i>Angle_Cheville_Min</i>	$\pi/2$	$-0.20\pi$
<i>Angle_Cheville_Max</i>	$\pi/2$	$-0.90\pi$
<i>Angle_Coude_Min</i>	$\pi/2$	$-0.20\pi$
<i>Angle_Coude_Max</i>	$\pi/2$	$-0.75\pi$
<i>Angle_Main_Min</i>	$\pi/2$	0
<i>Angle_Main_Max</i>	$\pi/2$	$-0.60\pi$

Des vidéos illustrant l'analyse du mouvement des doigts, puis le mouvement induit d'une part en 2D puis en 3D sont disponibles sur {gduvan,morin}.perso.enseiht.fr .

## 4 Liaison entre les deux approches

Les paramètres contrôlant le personnage 3D peuvent être déduits très facilement en partant des trajectoires estimées. Une vitesse  $V_d$  peut être extraite des courbes de Bézier en calculant la somme des distances entre les positions des centres des régions dans des trames consécutives. A la fin, une normalisation est nécessaire pour obtenir une vitesse variant dans l'intervalle  $[0, 1]$ . De plus le calcul de la *Période* est simple grâce aux propriétés des courbes de Bézier. La détection des sauts se fait en calculant seulement la somme des distances sur la verticale (la hauteur de la vidéo). En comparant cette valeur à un certain seuil choisi auparavant, on peut donner une valeur à la variable booléenne *Saut*. Au cours de l'animation du personnage cette variable va ajouter une translation sur l'axe OY au mouvement (au début vers le haut puis vers le bas pour revenir

à la position Y initiale) pendant une certaine durée choisie par l'utilisateur.

Par rapport à la première approche (voir section 3.2) l'influence du mouvement analysé est modérée ; le personnage modifie son comportement avec une certaine inertie. Sa réaction est ralentie et proportionnelle au geste. En effet, pour pondérer cette inertie, la vitesse du personnage sera une fonction qui dépend de sa vitesse précédente et du paramètre *Vitesse*. Dans ce cas, la vitesse offerte au modèle d'animation à un moment de temps  $i + 1$  (correspondant à la trame  $T_{i+1}$ ) aura une loi de la forme suivante :

$$Vitesse(i + 1) = Vitesse(i) + CI \cdot f(V_d, Vitesse(i)) \quad (10)$$

où CI représente un "coefficient d'inertie" du personnage qui détermine la rapidité de sa réaction aux stimuli. La fonction  $f$  est celle qui analyse les paramètres du geste et, en conséquence, modifie les paramètres du personnage. Dans le cas le plus simple, on a :

$$f(V_d, Vitesse) = V_d - Vitesse \quad (11)$$

Dans ce cas, le personnage aura tendance à suivre, avec sa propre inertie, le mouvement du geste et approcher, à son propre rythme, sa vitesse de celle du geste.

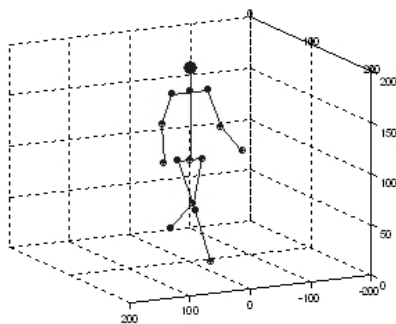


FIG. 12 – La deuxième approche : le personnage 3D influencé par le geste (simulation en MATLAB).

## 5 Limites de notre approche

Dans la section 2.2 nous avons utilisé la méthode colorimétrique pour la détection des couleurs dans les trames de la vidéo. Cette méthode est stable aux changements d'illumination de lumière blanche grâce à la normalisation des composantes des couleurs. Par contre dans le cas où on ajoute une source d'illumination monochromatique ou qui ne couvre pas la totalité du spectre visible de la caméra avec une intensité constante pour chaque longueur d'onde, le procédé perd sa robustesse. Dans les cas habituels d'utilisation, l'étape de calibration par apprentissage supervisé garantit un bon fonctionnement de l'algorithme même avec un changement d'intensité lumineuse des sources de lumière déjà présentes. La méthode de remplissage des données présentée dans 2.4 offre de très bons résultats quand

les occultations sont de courtes durées. Dans le cas de la gestuelle que nous avons analysé, des occultations de grandes durées apparaissent plutôt quand les doigts restent au repos (le doigt avant occulte complètement le doigt arrière). Mais dans ce cas la méthode des  $k$  plus proches voisins garde sa robustesse à cause du fait que dans l'intervalle de temps traité les données restent constantes (il n'y a pas de mouvement). Une autre limitation de notre solution est liée à la qualité de la caméra utilisée (le nombre de trames par seconde qu'elle enregistre). Si l'enregistrement de la vidéo est lent par rapport à la vitesse de la gestuelle, la continuité du mouvement réel est perdue. Le mouvement représenté dans la vidéo n'est plus cohérent. Notre méthode d'analyse, basée en particulier par une représentation fonctionnelle du mouvement, et un remplissage aux  $k$  plus proches voisins, présuppose un mouvement continu ; l'analyse n'est plus adaptée.

## 6 Conclusion et perspectives

La méthode colorimétrique avec construction des trajectoires par régression représente un moyen simple et robuste d'identification d'un geste dans une vidéo, préenregistré ou en streaming (grâce à la vitesse rapide de l'algorithme proposé, cette fonctionnalité sera implémentée prochainement). En munissant l'utilisateur d'un gant en couleurs, on pourra facilement interpréter une grande variété de gestes. Au cas où les données collectées sont incomplètes -à cause par exemple d'occultations- les données manquantes sont estimées. De plus, un système d'élimination des erreurs dues à une vitesse trop grande du geste ou à une mauvaise identification des zones de couleur rend la détection robuste. Ces deux méthodes permettent d'obtenir des données de bonne qualité sur les positions des zones de couleur. Ces informations ont été exploitées pour trouver des trajectoires par régression, selon le modèle des courbes de Bézier. L'analyse de ces trajectoires donne une grande variété d'informations sur le geste, permettant une identification directe ou la paramétrisation d'un geste. Dans notre approche, le geste simple montrait une vitesse, un sens et des éventuels sauts. Ces informations ont pu être transmises à un personnage 3D facilement contrôlable par quelques variables.

Il serait possible d'augmenter la complexité du personnage (pour éventuellement pouvoir contrôler un robot [Moj03]) en tenant compte des propriétés physiques du mouvement humain [BC89]. Notre approche généraliste devrait permettre l'identification des gestes plus complexes, leur analyse et leur synthèse dans un mouvement virtuel. Par ailleurs, il nous semble aussi pertinent, étant donné la simplicité du système, de faire évoluer nos algorithmes vers une analyse et une interprétation en ligne du geste.

## Références

- [AJ05] ATSUSHI N., JUNICHI H. : Video motion capture for virtual characters. *Modeling decisions for artificial intelligence* (2005), 431–441.

- [Ari06] ARIKAN O. : Compression of motion capture databases. *ACM SIGGRAPH* (2006), 890 – 897.
- [BC89] BRUDERLIN A., CALVERT T. W. : Goal-directed, dynamic animation of human walking. *Proceedings of the 16th annual conference on Computer graphics and interactive techniques* (1989), 233 – 242.
- [BH07] BRUBAKER M.A. ; FLEET D., HERTZMANN A. : Physics-based person tracking using simplified lower-body dynamics. *IEEE Conference on Computer Vision and Pattern Recognition, 2007.* (2007).
- [Far01] FARIN G. : *Curves and Surfaces for Cagd : A Practical Guide.* Elsevier Ltd, Oxford, 2001.
- [GH96] GROBEL K., HIENZ H. : Video-based recognition of fingerspelling in real-time. *Proc. des Workshops Bildverarbeitung für die Medizin, Aachen, 1996* (1996).
- [Moj03] MOJON S. : Realization of a physic simulation for a biped robot. *Semester Project, School of Computer and Communication Sciences EPFL* (2003).
- [OKS80] OHTA Y., KANADE T., SAKAI T. : Color information for region segmentation. *Computer Graphics and Image Processing* 13, 3 (July 1980), 222 – 241.
- [TC92] TSENG D.-C., CHANG C.-H. : Color segmentation using perceptual attributes. *Pattern Recognition, 1992. Vol.III. Conference C : Image, Speech and Signal Analysis, Proceedings.* (1992), 228–231.
- [TLF95] THIERRY C., LAMBERT P., FOULLOY L. : Segmentation d’images couleur dans la base teinte luminance saturation : approche numérique et symbolique, 1995.
- [TWC\*08] TOURNIER M., WU X., COURTY N., ÉLISE ARNAUD, REVERET L. : Compression de mouvement par analyse en géodésiques principales. *AFIG* (2008).
- [VSA03] VEZHNEVETS V., SAZONOV V., ANDREEVA A. : A survey on pixel-based skin color detection techniques. *Proc. Graphicon* (2003).