

Un système de synthèse et d'annotation automatique à partir de données capturées pour l'animation faciale expressive en LSF

Clément Reverdy, Sylvie Gibet, Caroline Larboulette, Pierre-François Marteau

Université Bretagne Sud, IRISA

Résumé

Ce papier présente le fonctionnement d'un système visant à créer automatiquement des corpus d'animations faciales annotées pour la Langue des Signes Française (LSF) à partir de données issues de capture du mouvement (MoCap). L'animation produite est basée *blendshapes*. Les coefficients associés à ce modèle sont calculés à partir des données capturées puis utilisés par le module de reconnaissance afin d'annoter automatiquement ces animations.

Mots clé : LSF, animation faciale, données *MoCap*, reconnaissance automatique, expressions faciales, HMM, *thin-shell*, *blendshapes*

1. Introduction

La Langue des Signes Française (LSF), utilisée en France par la communauté des sourds et leur entourage, représente une part majeure de la culture et de l'identité de cette dernière. Le développement d'outils informatiques éducatifs, de divertissement et de travail tenant compte de leur langue naturelle est apprécié des 100000 locuteurs français (environ 169000 dans le monde)[†]. Outre la vidéo, l'un des moyens permettant la prise en charge de cette langue qui ne connaît à l'heure actuelle pas de forme écrite est la génération de contenu par le biais de personnages virtuels animés appelés *avatars signeurs*.

Le système que nous proposons s'intègre dans un projet plus général de synthèse gestuelle de la LSF par concaténation qui permet d'éditer des morceaux de mouvements capturés, et de les couper / coller / transformer / mixer afin de construire de nouvelles phrases [GCDLN11, LAGT*]. Ce système nécessite d'avoir à disposition une base de données annotée qui comprend à la fois des mouvements et des informations sémantiques qui étiquettent ces mouvements [ACD*] de façon à produire de nouvelles animations à partir de phrases énoncées comme une combinaison d'annotations. Cependant, son utilisation nécessite de disposer d'un corpus conséquent de données annotées, ce qui représente un travail humain important et fastidieux.

En LSF, l'expressivité faciale revêt une importance particulière dans les langues signées [HLR11] car elle est le vecteur de nombreuses informations (e.g., prosodiques, clausales ou adjectivales) sans lesquelles la compréhension de la langue ne peut être que partielle.

Nos corpus, que ce soit pour les mouvements corporels ou faciaux, sont constitués à partir de données *MoCap* qui présentent l'avantage de fournir une grande précision tant spatiale que temporelle (fréquence d'acquisition > 200 fps) ainsi que l'exploitation d'une technologie unique et synchrone pour ces deux canaux.

Le système proposé permet, à partir de mouvements capturés, (i) de synthétiser les paramètres nécessaires pour l'animation faciale, et (ii) de produire de manière semi-automatique les annotations correspondantes à partir de ces mêmes paramètres.

En matière d'animation faciale, les traitements peuvent varier suivant la nature des données dont on dispose initialement [RGL15]. Dans le cadre des méthodes employées avec des données issues de la capture du mouvement basée marqueurs, nous évoquerons deux principales familles de méthodes. Tout d'abord, nous retrouvons les méthodes basées Laplacien comme le modèle *thin-shell* [BS08, BBA*, LZD13] où un sous-ensemble de sommets du maillage cible est contraint et suit les mouvements des marqueurs qui leur sont associés, alors que les autres sommets sont modifiés de façon à minimiser la déformation du maillage. Ce type de méthode est jusqu'à un certain point compatible avec du temps-réel, cela dépend du nombre total de sommets du maillage cible. Le second type principalement utilisé est celui basé *blendshapes* [LAR*, SLS*12, DCFN]. Il s'agit d'une représentation constituée d'une expression

[†]. <https://www.ethnologue.com/language/fsl>

neutre du maillage et d'un ensemble d'expressions de base (dites bases) représentant chacune une expression faciale particulière exprimée différenciellement par rapport à l'expression neutre. Une expression E quelconque est alors représentée comme l'expression neutre B_{neutre} à laquelle on ajoute une combinaison linéaire de l'ensemble des bases B_i :

$E = B_{neutre} + \sum_{i=0}^b w_i B_i$. Cette représentation par *blendshapes* est intéressante pour plusieurs raisons. Elle permet un stockage relativement compact (dans notre cas une cinquantaine de coefficients par *frame*). De plus il s'agit d'une représentation avec un haut niveau d'abstraction pouvant constituer un descripteur efficace pour un module de reconnaissance visant à annoter automatiquement nos corpus. Enfin, le vecteur de coefficients représente une couche d'abstraction entre les données brutes capturées et le module d'annotation car il ne dépend pas de la morphologie de l'acteur mais uniquement du modèle (maillage + bases *blendshapes*) ce qui facilitera la tâche lors de la constitution de corpus multi-acteurs.

2. Méthodologie, modèles et expérimentations

Notre objectif est dual. D'une part nous souhaitons obtenir automatiquement les animations correspondant à nos données *MoCap* ; d'autre part nous voulons annoter ces données aussi automatiquement que possible. La figure 1 schématise le système de synthèse / reconnaissance que nous proposons.

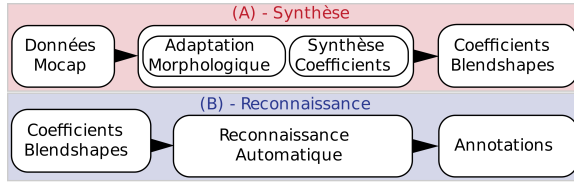


Figure 1: Schéma d'ensemble.

Au niveau de la synthèse, nous partons des données brutes auxquelles nous appliquons une adaptation morphologique via une régression RBF (fonction à base radiale). Les coefficients de *blendshapes* sont ensuite optimisés afin de minimiser la distance entre ces positions et celles des sommets correspondants du maillage. Le module destiné à l'annotation reprend ces coefficients en entrée et effectue ses prédictions sur la base d'un modèle HMM (*modèle de Markov à états cachés*).

2.1. Corpus

Les données ont été capturées à une fréquence de 200 *frames* par seconde via un dispositif de motion capture sur 2 acteurs (un homme et une femme). Chaque acteur a réalisé une séquence par émotion (*Expressions Isolées (EI)*) où il alterne successivement 5 à 6 fois l'expression neutre puis l'émotion en question ainsi qu'une série (1 par émotion) d'autres séquences (*Séquences d'Expression (SE)*) au cours desquelles il alterne chaque émotion les unes après les autres (exemple : neutre joie surprise joie colère joie peur joie tristesse joie dégoût joie). L'unique pré-traitement effectué sur ces données est l'annulation des déplacements (rotation + translation) liés aux mouvements corporels et de la tête.

2.2. Système de synthèse

Adaptation Morphologique Dans un premier temps, une adaptation des coordonnées 3D est nécessaire afin de gérer les différences morphologiques entre le visage de l'acteur et celui de l'avatar. Pour résoudre ce problème de correspondance entre les données *MoCap* et 3D, nous appliquons une méthode classique de régression RBF (voir [NN01, BBA*, SLS*12]). Soit M la matrice de dimension $(f, 3 \times m)$ contenant les coordonnées 3D des m marqueurs au cours des f *frames* de la séquence. Soit B la matrice de dimension $(b+1, 3 \times v)$ représentant le maillage de l'avatar et les différentes bases du modèle *blendshapes* avec v le nombre de sommets, b le nombre de bases et où la première ligne B_0 contient les positions des coordonnées 3D des sommets de l'expression neutre. Soit V l'ensemble des indices des m sommets correspondant aux marqueurs. La régression RBF permet de calculer les coordonnées $\hat{M}$ des marqueurs adaptées à la morphologie de l'avatar cible pour chaque *frame* i et marqueur j :

$$\hat{M}_{i,j} = \sum_{k \in V} u_k (||M_{i,j} - B_{0,k}||^3) + q(M_{i,j}) \quad (1)$$

avec u_k les coefficients associés aux fonctions radiales de base et q un polynôme trivarié $q(x,y,z) = ax + by + cz + d$.

Optimisation Les coefficients de *blendshapes* sont ensuite calculés par optimisation. Soit X un vecteur de coefficients de *blendshapes*, soit, pour une *frame* i donnée, le vecteur différentiel Dm_i entre les positions *MoCap* adaptées et l'expression neutre $Dm_i = \hat{M}_i - B_0$ et soit Da_i le vecteur des déformations synthétisées de l'avatar $Da_i = X_i \cdot B_{I \in [1,b]}$ (on ne considère que les sommets correspondant aux marqueurs). Le problème d'optimisation revient à déterminer les coefficients de *blendshapes* à chaque *frame* i :

$$\hat{X}_i = \underset{X}{\operatorname{argmin}} (||Dm_i - Da_i||) \quad (2)$$

La solution à ce problème n'est pas unique car différentes combinaisons de coefficients peuvent minimiser la distance entre les positions des marqueurs et leur sommet correspondant. Ainsi, en minimisant la distance donnée par l'équation 2, on peut tomber sur des solutions qui incluent des jeux de coefficients largement inférieurs à 0 ou supérieurs à 1. Or les expressions de base sont en général créées en considérant que les coefficients qui leur sont associés seront compris entre ces bornes : l'utilisation de coefficients qui en sortent peut alors conduire à la génération d'artefacts.

Pour éviter cela nous définissons deux énergies de régularisation. La première, E_{bounds} , définit l'espace dans lequel les coefficients sont autorisés à évoluer. La seconde, $E_{thinShell}$, pénalise une déformation trop importante du maillage. Des coefficients α définissent l'importance donnée à ces énergies. L'équation 2 devient :

$$\hat{X}_i = \underset{X}{\operatorname{argmin}} (||Dm_i - Da_i|| + \alpha_{bounds} E_{bounds} + \alpha_{thinShell} E_{thinShell}) \quad (3)$$

Soient s un paramètre de lissage (ici 0.1), l_{inf} et l_{sup} les bornes respectivement inférieures et supérieures (ici -0.1

et 1.1), E_{bounds} est décrite par l'équation ci-dessous :

$$E_{bounds} = \sum_{j=1}^{b+1} \left(\exp\left(\frac{l_{inf} - X_{j-1}}{s}\right) \right) + \sum_{j=1}^{b+1} \left(\exp\left(\frac{X_{j-1} - l_{sup}}{s}\right) \right)$$

$E_{thinShell}$ découle quant à elle de l'équation du modèle *thin-shell* (cf. [BS08]) :

$$\underbrace{(-k_s \mathcal{L}_V + k_b \mathcal{L}_V^2)}_A D = 0 \quad (4)$$

$\mathcal{L}_V$ et $\mathcal{L}_V^2$ sont respectivement le laplacien et le bi-laplacien du maillage et D représente les déformations appliquées à ce dernier.

Pour intégrer cette énergie au système de synthèse basé *blendshapes*, on définit la matrice $AB = A \cdot B_{j \in [1, b+1]}^T$, avec AB' la matrice contenant le sous ensemble de ses lignes $k \notin V$ (non contraints). À partir des équations 3 et 4 on obtient :

$$\hat{X}_i = \underset{X}{\operatorname{argmin}} (||Dm_i - Da_i|| + \alpha_{thinShell} ||AB' \cdot X^T|| + \alpha_{bounds} E_{bounds}) \quad (5)$$

La minimisation est effectuée au sens des moindres carrés en utilisant un algorithme classique.

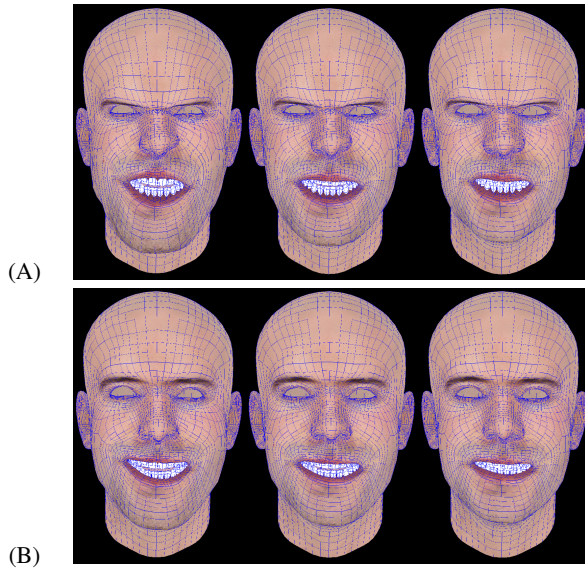


Figure 2: De gauche à droite : résultat avec uniquement E_{bounds} ; mélange de E_{bounds} et $E_{thinShell}$; résultat avec uniquement $E_{thinShell}$. (A) Colère, (B) Joie .

Les deux énergies de régularisation employées influent différemment sur l'animation produite. E_{bounds} définit simplement l'espace dans lequel le vecteur des coefficients est libre d'évoluer. Si l'on se limite à appliquer cette contrainte, les distances entre marqueurs et sommets correspondants seront réduites, peu importe la façon dont le maillage est déformé pourvu que les coefficients de *blendshapes* restent dans cet espace. Il en résulte des expressions très expressives mais parfois irréalistes. En revanche, la contrainte définie par $E_{thinShell}$ impose que ces distances soient minimisées tout en réduisant le plus possible la déformation du maillage. La plupart du temps l'expression obtenue paraît plus naturelle mais légèrement moins expressive et surtout on retrouve beaucoup de coefficients inférieurs à 0 ou supérieurs

à 1 ce qui n'est pas souhaitable. Dans certains cas il serait préférable d'utiliser l'une ou l'autre de ces deux énergies. Combiner les deux offre néanmoins un compromis avantageux, qui se traduit par la limitation des coefficients à leur espace autorisé, l'augmentation de l'expressivité et la plausibilité de l'expression générée (voir fig. 2).

Une fois les énergies de régularisation normalisées, nous avons choisi les paramètres α de telle façon que $\alpha_{bounds} + \alpha_{thinShell} = 1$. En effet, si cette somme est > 1 alors les expressions ont tendance à être trop atténuées, si cette somme tend vers 0 on voit apparaître des artefacts. Dans les exemples présentés ici : $\alpha_{bounds} = 0.5$, et $\alpha_{thinShell} = 0.5$.

2.3. Annotation basée HMM

Un HMM (modèle de Markov caché) est un modèle génératif permettant de traiter des données à variables latentes temporelles. Entraîner un HMM consiste à déterminer 3 jeux de paramètres optimaux : le vecteur Π des probabilités de se trouver dans chacun des états lors de la première observation, la matrice $\mathcal{A}$ des probabilités de transition de chaque état vers chaque état à chaque observation et les paramètres $\mathcal{B}$ relatifs à la fonction d'émission de chaque état (voir [Rab89] pour plus d'informations sur les HMM). Dans le système proposé, les fonctions d'émission sont toutes modélisées par des GMM (*modèle de mixture gaussien*).

En premier lieu, nous posons le problème trivial de segmentation suivant : pour chaque séquence d'expression isolée EI (voir sec. 2.1), nous souhaitons effectuer une clusterisation qui isole 3 classes : la classe neutre $_ [expr]$, la classe transitoire *trans* et la classe pic expressif $[expr]$. Entraîner un HMM par séquence avec 3 états cachés ($nbStates = 3$) et 20 composantes ($nbComp = 20$) pour les GMM permet d'effectuer efficacement cette tâche (fig. 3).

Nous entraînons ensuite un HMM par expression ($HMM[expr]$) en prenant pour données d'entraînement chacune des sous-séquences précédemment isolées correspondant aux pics expressifs. Nous effectuons le même traitement ($HMM_ [expr]$) pour les classes neutres. Les paramètres de ces modèles ($nbStates = 5$ et $nbComp = 20$) ont été choisis arbitrairement.

À partir de ces modèles, nous avons déterminé les paramètres d'un HMM plus général. Soit $nbClass$ le nombre de classes totales (pics expressifs + neutres), $class(i)$ la classe à laquelle appartient l'état caché i , $\Pi_{concat} = [\Pi^{joy}, \Pi^{fear}, \dots, \Pi^{anger}, \Pi^{surprise}]$ et $probOut = 0.01$ la probabilité (choisie arbitrairement) de passer d'une classe émotionnelle à une autre à chaque observation, les paramètres de ce modèle sont les suivants :

$$\Pi = \frac{1}{nbClass} \times \Pi_{concat} \quad (6)$$

$$\mathcal{A}_{i,j} = \begin{cases} \frac{1}{nbClass-1} \times probOut \times \Pi_{concat}, & \text{if } class(i) \neq class(j) \\ (1 - probOut) \times \mathcal{A}_{i,j}^{class(i)}, & \text{else} \end{cases} \quad (7)$$

$$\mathcal{B}_i = \mathcal{B}_i^{class(i)} \quad (8)$$

L'annotation est ensuite effectuée *frame par frame* en déterminant via ce modèle la séquence d'états la plus probable.

La fig. 4 montre un exemple d'annotations obtenues

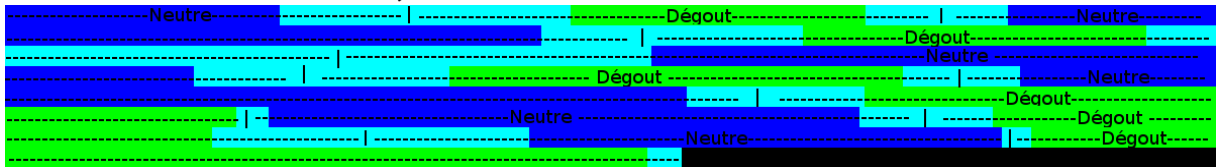


Figure 3: Segmentation automatique d'expressions isolées via un HMM à 3 états. On distingue ici en bleu l'expression neutre, en cyan un état intermédiaire et en vert les pics expressifs. L'expression enregistrée est le dégoût répété 6 fois (textuellement en noir : l'annotation manuelle). 1 ligne = 6s.

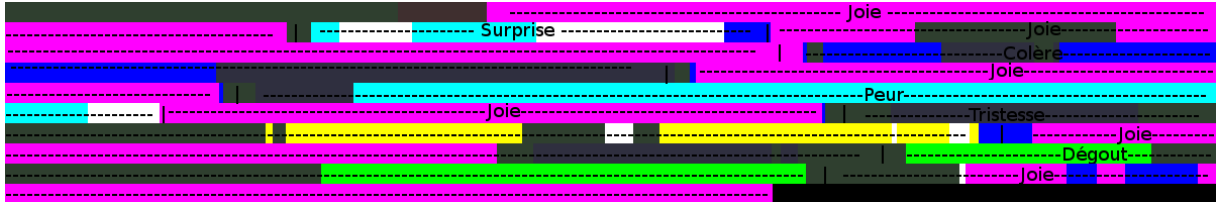


Figure 4: Annotation automatique en couleur (bleu : colère, rose : joie, blanc : surprise, cyan : peur, jaune : tristesse, vert : dégoût, autres couleurs : expressions neutres). Textuellement en noir : l'annotation manuelle. 1 ligne = 6 secondes.

sur une séquence d'expressions effectuées successivement. On peut voir que la méthode n'est pas encore au point, notamment certaines classes (en particulier surprise/peur, joie/colère) sont fréquemment confondues alors même que la tâche reste simple (le nombre de classes est très limité). Toutefois cette première expérience reste suffisamment significative pour valider notre démarche. À l'avenir nous envisageons de tester d'autres méthodes basées CRF (*Conditional Random Fields*) ou RNN (*Recurrent Neural Network*).

3. Conclusion et perspectives

Les deux principaux apports des travaux présentés ici sont i) une démarche globale visant à faciliter la constitution de corpus faciaux annotés, ii) une méthode de synthèse qui tire partie de l'approche *thin-shell* en tant qu'énergie de régularisation pour améliorer les résultats, tout en restant dans un système de synthèse de type *blendshapes*.

Concernant la partie synthèse, les résultats obtenus sont satisfaisants. Ils devront cependant être validés de deux façons différentes : (i) les animations devront faire l'objet d'études perceptuelles auxquelles participent des personnes sourdes et signantes, (ii) les différents paramètres peuvent être réévalués automatiquement via une validation croisée sur la base des taux de reconnaissance obtenus par le système d'annotation.

En ce qui concerne l'annotation automatique, nous en sommes encore à une phase exploratoire. La démarche employée nous paraît encourageante et différentes pistes (en particulier des méthodes basées CRF et RNN) sont envisagées afin d'améliorer notre système. Les méthodes développées devront par ailleurs être testées sur des jeux de données faciales expressives plus conséquents.

Références

[ACD*] AWAD C., COURTY N., DUARTE K., LE NAOUR T., GIBET S. : A Combined Semantic and Motion Capture Database for Real-Time Sign Language Synthesis. In *9th Int. Conf. on Intelligent Virtual Agent (IVA 2009)*, pp. 432–438.

[BBA*] BICKEL B., BOTSCH M., ANGST R., MATUSIK W., OTADUY M., PFISTER H., GROSS M. : Multi-scale capture of

facial geometry and motion. In *ACM SIGGRAPH 2007 Papers* (New York, NY, USA).

- [BS08] BOTSCH M., SORKINE O. : On linear variational surface deformation methods. *IEEE Trans. on Visualization and Computer Graphics*. Vol. 14, Num. 1 (janvier 2008), 213–230.
- [DCFN] DENG Z., CHIANG P.-Y., FOX P., NEUMANN U. : Animating blendshape faces by cross-mapping motion capture data. In *Proce. of the 2006 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA), pp. 43–48.
- [GCDLN11] GIBET S., COURTY N., DUARTE K., LE NAOUR T. : The SignCom System for Data-Driven Animation of Interactive Virtual Signers : Methodology and Evaluation. *ACM Trans. on Interactive Intelligent Systems*. Vol. 1, Num. 1 (2011), 6.
- [HLR11] HUENERFAUTH M., LU P., ROSENBERG A. : Evaluating importance of facial expression in american sign language and pidgin signed english animations. In *Proc. of the 13th International ACM SIGACCESS Conf. on Computers and Accessibility* (New York, NY, USA, 2011), ASSETS, pp. 99–106.
- [LAGT*] LEFEBVRE-ALBARET F., GIBET S., TURKI A., HAMON L., BRUN R. : Overview of the Sign3D Project High-fidelity 3D recording, indexing and editing of French Sign Language content. In *Third Int. Symp. on Sign Language Translation and Avatar Technology (SLTAT) 2013* (Chicago, United States).
- [LAR*] LEWIS J. P., ANJYO K., RHEE T., ZHANG M., PIGHIN F., DENG Z. : Practice and Theory of Blendshape Facial Models. In *Eurographics 2014 - State of the Art Reports*.
- [LZD13] LE B. H., ZHU M., DENG Z. : Marker optimization for facial motion acquisition and deformation. *IEEE Trans. on Visualization and Computer Graphics*. Vol. 19, Num. 11 (2013), 1859–1871.
- [NN01] NOH J.-Y., NEUMANN U. : Expression cloning. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), SIGGRAPH, pp. 277–288.
- [Rab89] RABINER L. R. : A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*. Vol. 77, Num. 2 (1989), 257–286.
- [RGL15] REVERDY C., GIBET S., LARBOULETTE C. : Animation faciale basée données : un état de l'art. In *28èmes journées de l'Association Française en Informatique Graphique* (2015).
- [SLS*12] SEOL Y., LEWIS J., SEO J., CHOI B., ANJYO K., NOH J. : Spacetime expression cloning for blendshapes. *ACM Trans. Graph.*. Vol. 31, Num. 2 (avril 2012), 14 :1–14 :12.