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, P-F Marteau

Journées Françaises d'Informatique Graphique,
Grenoble, 30 nov - 2 dec 2016

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

C. Reverdy[1,2], S. Gibet[1,2], C. Larboulette[1,2] et P-F. Marteau[1,2]

[1]Université de Bretagne Sud [2]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 basée données, reconnaissance automatique d'expressions faciales

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*13]. 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*09] 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.

†. https ://www.ethnologue.com/language/fsl

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*07, LZD13] où un sous-ensemble de sommets du maillage cible est contraint et suit les mouvement 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*14, SLS*12, DCFN]. Il s'agit

d'une représentation constituée d'une expression neutre du maillage et d'un ensemble d'expressions de base (dites bases) représentant chacune une expression faciale particulière exprimée différentiellement par rapport à l'expression neutre. Une expression E quelconque est alors représentée comme l'expression neutre B_{neutre} plus une combinaison linéaire de chacune de ces bases : $E = B_{neutre} + \sum_{i=0}^{b} w_i B_i$. 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 *blendshape*) 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 correspondantes 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.

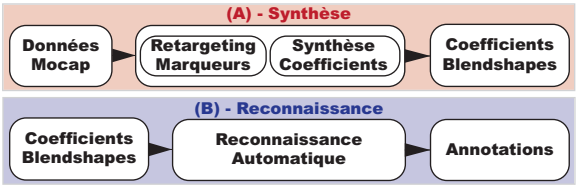


Figure 1: *Schéma d'ensemble.*

Au niveau de la synthèse, nous partons des données brutes auxquelles nous appliquons une adaptation morphologique via une régression RBF. 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.

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 (IE)*) 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 dégout 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 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, nous appliquons une méthode de régression RBF (voir [NN01, BBA*07, 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 :

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

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 $Da_i = X_i \cdot B_{l \in [1,b]}$ (on ne considère que les sommets correspondant aux marqueurs). Le problème d'optimisation revient à déterminer les coefficients de *blendshapes* à chaque *frame* i :

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

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 1 ou supérieurs à 0. Or les expressions de base sont en général créées en considérant que les coefficients qui leur sont associées seront compris entre ces bornes, et l'utilisation de coefficients qui en sortent peut 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. L'équation 2 devient :

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

Soient l_{inf} et l_{sup} les bornes respectivement inférieures et

supérieures, E_{bounds} est décrite par l'équation ci-dessous :

$$E_{bounds} = \sum_{j=1}^{b+1}(exp(\frac{l_{inf} - X_{j-1}}{s})) + \sum_{j=1}^{b+1}(exp(\frac{X_{j-1} - l_{sup}}{s})) \tag{4}$$

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

$$\underbrace{(-k_s\mathcal{L}_V + k_b\mathcal{L}_V^2)}_{A} D = 0 \tag{5}$$

Pour intégrer cette énergie au système de synthèse basé *blendshapes*, on définit la matrice $AB = A \cdot B^T_{i\in[1,b+1[}$, et ABu le sous ensemble des lignes $i \notin V$ (non contraints). En exploitant les équations 3 et 5 on obtient :

$$\hat{X}_i = \underset{X}{\operatorname{argmin}}(||Dm_i - Da_i|| + \alpha_{thinShell}||ABu \cdot X^T|| + \alpha_{bounds}E_{bounds}) \tag{6}$$

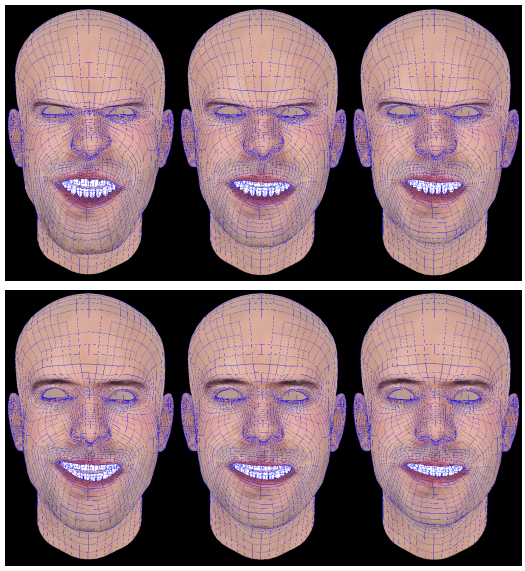
La minimisation est effectuée au sens des moindres carrés en utilisant un algorithme classique.

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 parait 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).

2.3. Annotation basée HMM

La méthode proposée ici est un premier essai réalisé en vue de délivrer rapidement une preuve de concept. Le système est basé sur des modèles de Markov cachés (HMM) construits à partir de *Gaussian Mixture Models* (GMM) pour modéliser les fonctions de densité de probabilité (voir [Rab89] pour plus d'informations sur les HMM). 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.

Dans un premier temps, 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 $[expr]$. Il se trouve qu'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. 4).



(A) (B)

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

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, $\Pi_{concat} = [\Pi^{joy}, \Pi^{fear}, \cdots, \Pi^{\_anger\_}, \Pi^{\_surprise\_}]$ 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 :

$$\Pi = \frac{1}{nbClass} \times \Pi_{concat} \tag{7}$$

$$\mathcal{A}_{i,j} = \begin{cases} \frac{1}{nbClass-1} \times probOut \times \Pi_{concat}, & \text{if } class(i) \neq class(j) \\ (1 - probOut) \times A_{i,j}^{class(i)}, & \text{else} \end{cases} \tag{8}$$

$$\mathcal{B}_i = \mathcal{B}_i^{class(i)} \tag{9}$$

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

La fig. 3 montre un exemple d'annotations obtenues sur une séquence d'expressions effectuées successivement.

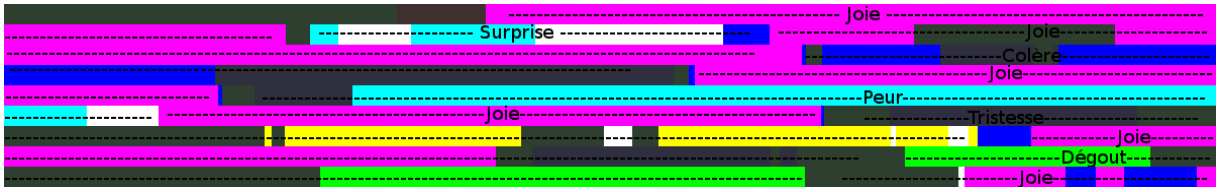


Figure 3: *Annotation automatique en couleur (bleu : colère, rose : joie, blanc : surprise, cyan : peur, jaune : tristesse, vert : dégout, autres couleurs : expressions neutres). Textuellement en noir : l'annotation mannuelle. Une ligne correspond à* 6 *secondes d'enregistrement.*

.

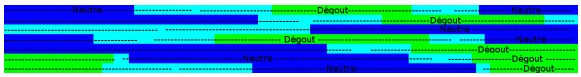


Figure 4: *Segmentation automatique d'expressions isolées via un HMM à* 3 *états. On distingue ici en bleu l'expression neutre, en cyan un état intermédiaire et en vert les pics expressifs. Chaque ligne correspond à* 6 *secondes d'enregistrement, l'expression enregistrée est le dégoût répété* 6 *fois (textuellement en noir : l'annotation manuelle).*

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 relativement 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 ou RNN.

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 mériteraient cependant d'être validés par comparaison entre plusieurs méthodes, voire en effectuant des études perceptuelles. En ce qui concerne l'annotation automatique, nous en sommes encore à une phase exploratoire. La démarche employée nous parait encourageante et différentes pistes sont envisagées afin d'obtenir un système entièrement fonctionnel. À l'avenir nous envisageons également de tester d'autres méthodes basées CRF ou RNN ou des méthodes à noyaux. Enfin les méthodes développées devront être testées sur des jeux de données faciales expressives plus conséquents.

Références

[ACD*09] AWAD C., COURTY N., DUARTE K., LE NAOUR T., GIBET S. : A Combined Semantic and Motion Capture Database for Real-Time Sign Language Synthesis.

[BBA*07] BICKEL B., BOTSCH M., ANGST R., MATUSIK W., OTADUY M., PFISTER H., GROSS M. : Multi-scale capture of facial geometry and motion. *ACM Trans. Graph.* (2007).

[BS08] BOTSCH M., SORKINE O. : On linear variational surface deformation methods. In *IEEE Transactions on Visualization and Computer Graphics* (2008).

[DCFN] DENG Z., CHIANG P.-Y., FOX P., NEUMANN U. : Animating blendshape faces by cross-mapping motion capture data. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*.

[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 Transaction on Interactive Intelligent Systems* (2011).

[HLR11] HUENERFAUTH M., LU P., ROSENBERG A. : Evaluating importance of facial expression in american sign language and pidgin signed english animations.

[LAGT*13] 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 International Symposium on Sign Language Translation and Avatar Technology (SLTAT)* (2013).

[LAR*14] LEWIS J. P., ANJYO K., RHEE T., ZHANG M., PIGHIN F., DENG Z. : Practice and Theory of Blendshape Facial Models.

[LZD13] LE B., ZHU M., DENG Z. : Marker optimization for facial motion acquisition and deformation. *Visualization and Computer Graphics, IEEE Transactions on* (2013).

[NN01] NOH J.-Y., NEUMANN U. : Expression cloning.

[Rab89] RABINER L. R. : A tutorial on hidden markov models and selected applications in speech recognition.

[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), Actes des 28èmesjournées de l' AFIG.

[SLS*12] SEOL Y., LEWIS J., SEO J., CHOI B., ANJYO K., NOH J. : Spacetime expression cloning for blendshapes. *ACM Trans. Graph.* (2012).