Titre: Title:	Amélioration de la capture des visages pour l'industrie du jeu vidéo
Auteur: Author:	Yannick Marion
Date:	2017
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Marion, Y. (2017). Amélioration de la capture des visages pour l'industrie du jeu vidéo [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/2804/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/2804/
Directeurs de recherche: Advisors:	Benoît Ozell
Programme: Program:	Génie informatique

UNIVERSITÉ DE MONTRÉAL

AMÉLIORATION DE LA CAPTURE DES VISAGES POUR L'INDUSTRIE DU JEU VIDÉO

YANNICK MARION
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
OCTOBRE 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

AMÉLIORATION DE LA CAPTURE DES VISAGES POUR L'INDUSTRIE DU JEU VIDÉO

présenté par : MARION Yannick
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. BOYER François-Raymond, Ph. D., président
M. OZELL Benoît, Ph. D., membre et directeur de recherche
M. PAL Christopher J., Ph. D., membre

DÉDICACE

À ma famille de l'autre côté de l'Atlantique,
je vous dédie ce mémoire, vous qui avez toujours cru en moi.

REMERCIEMENTS

Merci à Benoit Ozell, mon directeur de recherche, pour son support et ses conseils avisés qui ont grandement aidé à la réalisation de ce mémoire.

Merci au groupe Ubisoft de m'avoir accueilli pour travailler en partenariat avec eux.

Tout particulièrement, merci aux équipes Reflex et Facebuilder du Technology Group. Merci à l'aide apportée par François Lesveque et Marc-André Bleau sur le suivi du projet, ainsi qu'à Souleymene Rouchou et Arnaud Hubert. Merci également à Joël Tremblay pour sa bonne humeur pendant ces deux dernières années.

Merci à l'équipe de Ubisoft La Forge d'avoir permis cette opportunité de recherche entre nos deux entités. Merci particulièrement à Yves Jacquier et Olivier Pomarez pour leur aide.

Merci au studio Alice d'Ubisoft de nous avoir laissé réaliser des sessions de Motion Capture afin d'avoir des données publiques de MoCap faciales de qualité professionnelle.

Merci à Luce-Claire Ndagano pour la gestion de mon dossier du côté d'Ubisoft.

Merci à Chi-Chun Nguyen, Pablo Toscano, Jerome Eippers et Matthew Lee d'avoir accepté de tester mon pipeline et de m'avoir donné de la précieuse rétroaction des artistes.

Merci à Pauline Marion, ma soeur, d'être venu me visiter à Montréal et d'avoir accepté de réaliser ces performances artistiques de qualité professionnelle.

Merci à MITACS d'avoir financé ma recherche.

RÉSUMÉ

Réaliser une animation réaliste d'un visage dans l'industrie du jeu vidéo est un défi technique. Il est extrêmement complexe, y compris pour les meilleurs animateurs 3D, de produire une animation faciale réaliste. Une approche à ce problème est de capturer la performance d'un acteur et de la reproduire automatiquement sur un modèle de visage paramétrable, communément appelé un «rig». Le processus reste toutefois toujours peu automatisé et repose énormément sur les artistes qui l'effectuent.

La question est alors de savoir comment arriver à trouver les paramètres d'animations sur un rig de visage en 3D à partir d'une source vidéo 2D mono-caméra ?

Nous proposons de ramener le problème dans un contexte physique : déterminer ces paramètres d'animation sans intervention humaine à partir de la session de capture de mouvement. L'objectif est alors de proposer aux artistes animateurs un nouvel outil qui leur permettra d'obtenir des résultats semblables à ceux du processus précédent, d'une façon plus rapide sans pour autant changer leurs habitudes de travail.

Notre nouveau pipeline consiste en un transfert de déformation entre la performance de l'acteur retranscrite dans un suivi de points en 3D vers les mêmes points sur le Rig visé, afin de réaliser une cible atteignable. L'artiste n'a pas besoin de fournir une pose neutre au système, elle est déterminée sans intervention humaine en projetant la pose neutre du Rig dans l'espace du suivi. Les paramètres d'animation optimaux sont ensuite déterminés itérativement en minimisant une énergie des moindres carrés entre la cible et le rig à l'aide d'une descente de gradient dans l'espace des paramètres d'animation. Afin de ne pas tomber dans un minimum local, un réalignement au cours du processus itératif est réalisé par zones du visage entre le rig et la cible. Si l'artiste n'est pas satisfait du résultat alors produit, il peut le corriger à l'aide des mêmes outils qu'il a l'habitude d'utiliser, que nous avons adaptés à la correction d'erreurs.

En utilisant notre nouveau pipeline, un artiste peut désormais réaliser une animation faciale réaliste en une demi-heure contre presque dix heures pour le même résultat avec le pipeline précédent. L'artiste n'a en effet plus qu'à se concentrer sur la correction des résultats insatisfaisants du nouveau pipeline plutôt que de réaliser l'animation entière du début.

ABSTRACT

Performing a realistic 3D facial animation in the video game industry is a real technical challenge. It is indeed extremely complex to produce a realistic facial animation, even for the most skilled artists. One way to solve this issue is to look at it differently: why simulate something we can capture? One approach is thus to capture and reproduce the actor's performance automatically on a parametrable model. This process, however, still relies heavily on artists and is poorly automatized.

How to automatically determine facial animation parameters on a 3D face rig from a Monocular 2D video source?

We propose to bring the problem back to the physical world: to automatically determine these animation parameters without human intervention from the motion capture session. Our goal is to give the artists a new tool which would be yielding similar results to the previous process in a more efficient way, without changing the way they work.

Our new pipeline consists in a deformation transfer between the actor's performance and the 3D model by tracking a set of 3D points on the actor's face and reproducing their movements on the same points on the rig to create a reachable target. The artist doesn't have to input a neutral pose to the system since it is automatically determined by projecting the neutral pose of the rig into the tracking space. We then iteratively determine the optimal animation parameters by minimizing a least square energy between the rig and the target in animation parameter space. To prevent local minimas, we realign the rig and the target by zones through the iterative process. If the artist is not satisfied with the output, he can correct it using the same tools he is used to use that we adapted to allow error corrections.

An artist can now produce a realistic facial animation in half an hour down ten hours using the previous pipeline for the same results. The artist indeed only has to focus on correcting the frames he judges to be not satisfying rather than starting animating from scratch.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

FACS	Facial Action Coding System
RBF	Radial Basis Function
PCA	Principal Components Analysis
LBS	Linear Blend Skinning
IHM	Interface Homme-Machine
MAP	Maximum a posteriori
MPACP	Mélange Probabiliste d'Analyseurs à Composantes Principales

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

Dans ce mémoire, nous chercherons à améliorer le pipeline actuel d'animation faciale utilisé par Ubisoft Montréal, notre partenaire de recherche. En effet, le pipeline d'Ubisoft est un pipeline classique d'animation faciale tel qu'utilisé dans l'industrie aujourd'hui.
Dans un pipeline basé sur un solveur basé sur des Radial Basis Functions (RBF), tel que celui de notre partenaire, l'animateur reproduit à certaines images clés ce qu'il voit dans la vidéo issue de la séance de Motion Capture à l'aide des paramètres d'animations du rig facial. Ces paramètres sont ensuite interpolés entre les différentes images clés à l'aide d'un mélange de fonctions gaussiennes.
Bien qu'efficace en termes de résultat graphique, cette méthode requiert un travail qui est très fastidieux et qui gagnerait à être automatisé. L'objectif sera donc d'améliorer l'ancienne implémentation de retargeting de paramètres d'animations faciale sur les visages de nouveaux acteurs en ce qui concerne la qualité et la productivité.

1.1 Définitions et concepts de base

Avant de se focaliser plus en détail sur la problématique du projet, il convient d'introduire certains concepts et éléments de langage propre au sujet.

`Os` : Os du squelette sous notre modèle 3D. Un artiste contrôle leurs mouvements pour produire une animation.

`Empeçage` : Déformation de la surface d'un modèle 3D basé sur le mouvement des os de son squelette. Un modèle courant d' Empeçage est le Linear Blend Skinning (LBS).

`Glissoirs` : Désigne l'espace de paramètre déformant les os du squelette lié à notre modèle 3D, produisant ainsi la pose voulue. Conceptuellement, ce sont les ficelles de notre marionnette. Nommé ainsi, car les outils pour contrôler l'animation dans l'Interface Homme-Machine (IHM) sont bien souvent des glissoirs.

`Rig` : Modèle 3D liés à un squelette et un empeçage. Contrôlable en bougeant les os, dont le mouvement est déterminé par l'utilisation de glissoirs. Conceptuellement, c'est une marionnette virtuelle qu'un animateur sait contrôler pour produire une animation.

`Blendshape` : Modèle 3D de haute qualité sous une posture déterminée. En mélangeant un ensemble de blendshapes par somme pondérée, on peut produire une animation. Il s'agit d'une méthode courante dans le cinéma, où l'affichage en temps réel des animations n'est pas un problème et où l'on peut donc se permettre de mélanger une multitude de modèles de haute qualité à chaque image pour produire une animation.

`RigMorph` : Nouveau concept introduit dans ce mémoire. Il s'agit, pour un même glissoir, d'un ensemble de positions du squelette, chacune associée à un état du glissoir. Notre modèle repose sur un ensemble de RigMorphs : chaque glissoir générant un nombre de poses 3D égal au nombre d'états de notre RigMorph. La différence fondamentale entre un Blendshape et un RigMorph est qu'un Blendshape ne correspond qu'à une pose 3D du maillage, tandis qu'un RigMorph correspond à une multitude de poses 3D sur le squelette lui même.

`MoCam` : Pipeline d'animation facial d'Ubisoft. Se base sur un solveur RBF. Permet de mettre des valeurs à des images-clés dans le temps dans l'IHM.

1.2 Éléments de la problématique

La problématique de la capture de visage est de reproduire la performance d'un acteur sur un modèle 3D synthétique. Cette problématique soulève deux questions de recherche importantes : comment arriver à capturer l'expressivité d'un acteur et comment la transmettre ensuite sur un modèle virtuel ?
En effet, si l'on pense aux applications qui utilisent ce genre de procédés, on remarque qu'on ne souhaite pas seulement reproduire la performance de l'acteur sur un modèle le représentant — bien que ce cas de figure puisse arriver — mais plutôt la transférer sur un autre modèle totalement différent. On pourrait tout à fait souhaiter transférer l'expressivité d'un acteur sur une tête de créature comme un animal par exemple.
Pour répondre à ce problème de cohérence des données, une même production reposera bien souvent sur un même ensemble de paramètres, qu'ils soient basés sur des Blendshapes ou un squelette. Par exemple, on cherchera à reproduire l'animation sur un visage humain et réutiliser les paramètres appris sur une tête d'animal.

Un autre problème important est l'automatisation. En effet, réaliser une animation faciale est une tâche fastidieuse qui requiert des dizaines d'heures de travail à un animateur professionnel pour donner un résultat satisfaisant.

Dans notre contexte de capture de visage pour l'industrie du jeu vidéo, un facteur limitant majeur est le temps de calcul. En effet, contrairement au cinéma d'animation, le média du jeu vidéo exige un rendu en temps réel, ce qui rend difficile l'utilisation de technique de pré rendu. Il faut donc que le transfert d'expressivité soit le plus simple possible. Lorsque l'on veut jouer l'animation, on souhaite donc transmettre le minimum d'information au moteur de jeu. On ne transmettra donc que les valeurs de nos glissoirs, abstraction du mouvement du squelette du modèle, que le moteur de jeu sait interpréter en temps réel. On ne peut pas mélanger des centaines de Blensdhapes de plusieurs millions de triangles.
On souhaite également ne transmettre que les glissoirs puisque ce sont les paramètres qu'un humain sait interpréter. Un animateur réussira plus facilement à travailler avec cet espace basé sur un squelette plutôt qu'avec un ensemble de Blendshapes souvent plus abstraits. Cela permettra donc de corriger la sortie et de même accentuer certaines expressions et en diminuer d'autres.

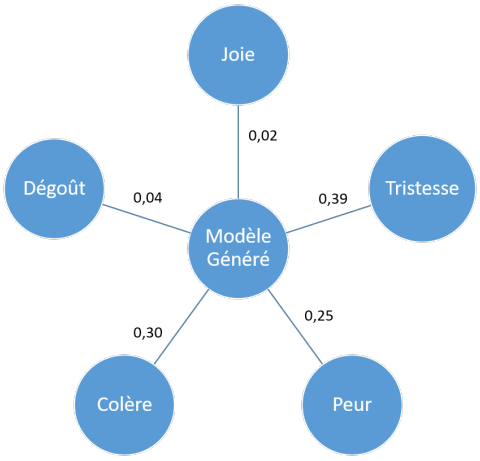

Figure 1.1 Exemple de mélange de Blendshape représenté en base Simpliciale

Une différence majeure existe entre un modèle basé sur un squelette et un modèle à ensemble de Blendshapes. Pour déterminer une expression faciale dans un ensemble de Blendshape, faire une somme pondérée des différents modèles suffit. Un exemple pour un ensemble de cinq Blendshapes est présenté à la figure 1.1. Il n'est pas possible de réaliser ceci avec un modèle basé sur un squelette, car le modèle 3D y est généré par opération d'empeçage sur les os dans une position donnée. On ne cherche donc pas à faire une somme pondérée de modèles 3D, mais de trouver quelle serait la position optimale des os pour produire un effet.

Les glissoirs nous donnent donc un moyen de faire abstraction du modèle 3D pour arriver à animer un visage. Mais il convient de lier à la performance de l'acteur capturée cette abstraction du modèle. En effet, réaliser une animation faciale, ce n'est que faire varier ces glissoirs en fonction du temps.

Une première approche, qui est celle utilisée communément pour l'animation de corps entier, serait de déterminer des images représentant des extrémums dans l'animation, et d'interpoler ensuite les valeurs des glissoirs entre ces extrémums.
Cette méthode fonctionne bien pour un corps, car on a généralement un faible nombre de degrés de liberté sur ce genre de modèles. D'un point de vue mécanique, il est possible d'interpoler la position et l'orientation de chacune des jointures entre ces extrémums pour reproduire l'animation fidèlement.

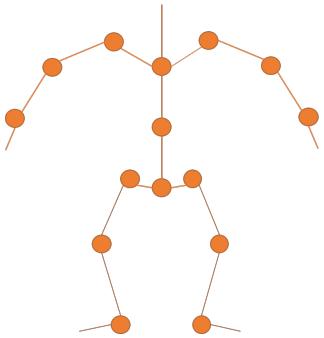

Figure 1.2 Schéma cinématique d'un squelette de corps entier.

Le problème est qu'un squelette facial ne fonctionne pas d'une façon mécaniquement si simple. Premièrement, le nombre d'os le composant n'est pas du même ordre de grandeur. Un squelette de corps, tel que celui présenté à la figure 1.2, contient généralement une dizaine d'os, contre une centaine pour un squelette facial. Aussi, les mouvements des os d'un squelette de corps ont très peu d'influence entre eux, contrairement aux mouvements d'un squelette facial qui sont très corrélés. En définissant une image-clé comme une image présentant un extrémum sur la position de l'une des jointures du squelette, on risque d'avoir la totalité de nos images comme images clés.

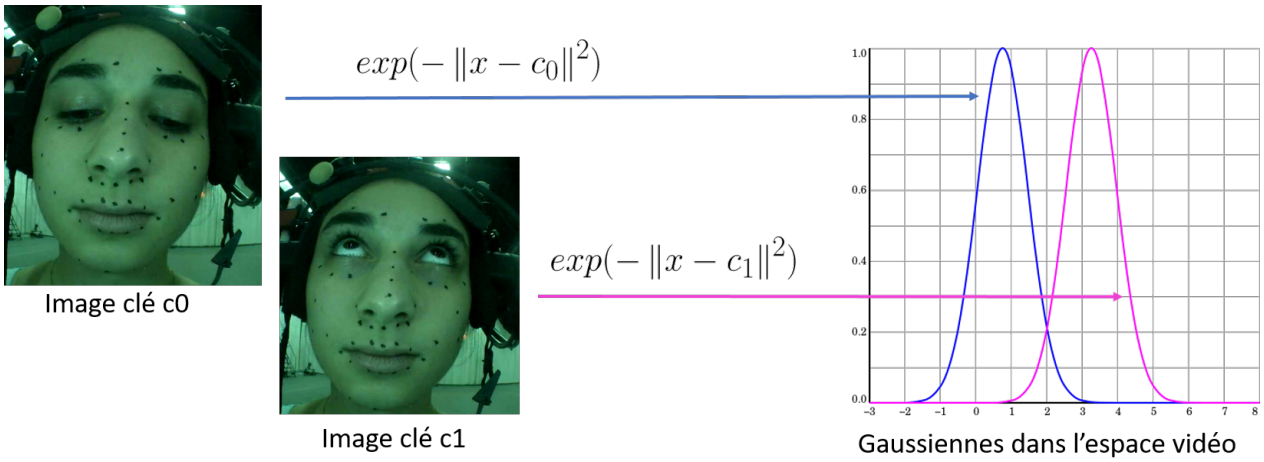

Figure 1.3 Apprentissage de l'espace des gaussiennes du RBF.

Une solution courante pour trouver les glissoirs réalisant la pose voulue automatiquement — qui est celle qu'utilise notre partenaire Ubisoft dans ses productions — est d'utiliser une méthode basée sur un solveur RBF. Avec une telle méthode, l'animateur «capture» l'expressivité de l'acteur à des images clés en reproduisant l'expression de son visage sur le modèle 3D à l'aide de l'espace d'animation. Chaque «capture» représentera une gaussienne dans l'espace d'entrée (les images de la capture, des points suivis, ...). Par mélange de gaussiennes entre tous ces apprentissages, on reproduit l'animation à toutes les images inconnues.

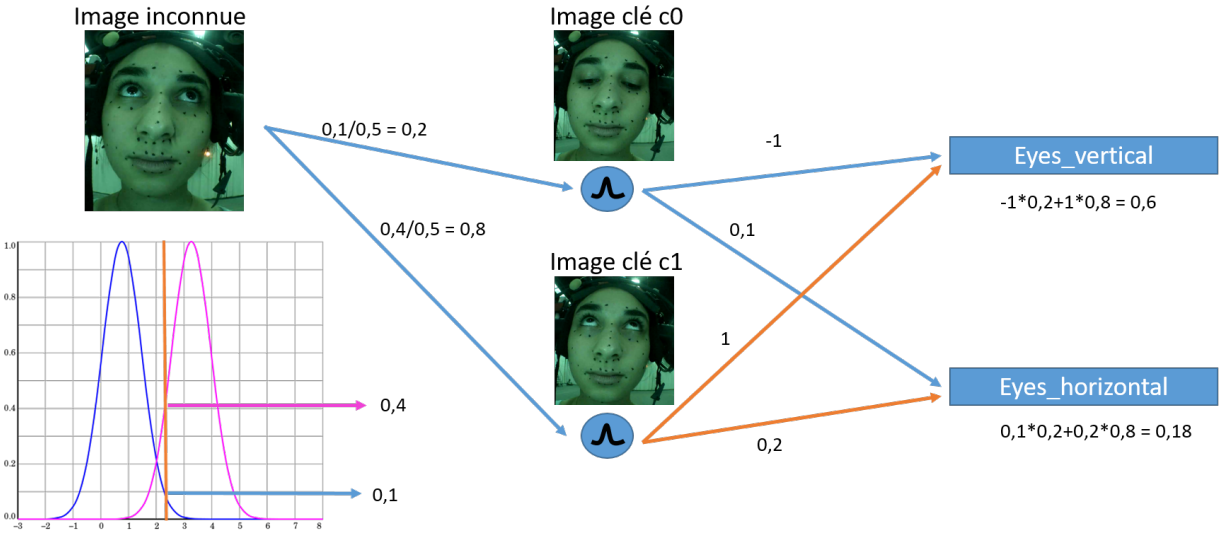

Figure 1.4 En projetant une image inconnue dans l'espace des gaussiennes, on trouve le mélange des paramètres d'animations.

Le problème de cette méthode, c'est qu'elle requiert un grand temps de travail de la part des animateurs afin d'arriver à un résultat correct. En effet, le visage est une partie très complexe du corps et l'on a donc bien souvent plus de 100 paramètres de contrôle différents sur les différents rigs faciaux. Il faut également en moyenne 100 à 200 images clés par animation pour un résultat correct. Il faut donc remplir à la main 10 000 à 20 000 paramètres. Pour un animateur entrainé, cela représente 12 heures de travail pour 5 minutes d'animation.

Figure 1.5 Capture d'écran de «Assassin's Creed : Unity» (©Ubisoft 2014-2017). Tous les personnages secondaires, comme ces passants, possèdent une animation faciale.

De plus, contrairement à l'industrie du cinéma, le jeu vidéo nécessite un bien plus grand nombre d'animations pour arriver à un résultat.
En effet, le média du cinéma est un média linéaire, tandis que le jeu vidéo est un média interactif. Cette différence subtile entraine un problème majeur : on ne choisit pas ce que l'on veut montrer à l'écran dans le média du jeu vidéo, puisque ce sont les joueurs eux-mêmes qui vont créer leur histoire. Il convient alors d'avoir une animation faciale de qualité et cohérente pour tous les protagonistes que le joueur va rencontrer.

Prenons l'exemple de «Assassin's Creed : Unity» (©Ubisoft 2014-2017), production de notre partenaire Ubisoft. Dans ce jeu, on suit un personnage fictif pendant les évènements de la Révolution française de 1789. On est alors amené à rencontrer les personnalités majeures de cet évènement (Danton, Robespierre, Louis XVI, . . .). Néanmoins, réaliser l'animation faciale de ces personnages ne suffit pas, il faut également avoir une animation faciale cohérente pour tous les personnages «sans importances» que le joueur va rencontrer dans sa session de jeu, à savoir les passants, les marchands, les mendiants, etc. Ce sont ces personnages, plus que les personnages «importants», qui vont renforcer l'immersion du joueur dans l'univers du jeu.
À l'opposé de cette idée, un film sur la Révolution française pourra ne se concentrer que sur les personnages principaux, puisqu'il va se focaliser sur les moments importants de l'évènement et non sur la cohérence d'un monde vivant.
Paradoxalement, un jeu vidéo requiert donc beaucoup plus de données d'animation qu'un film reposant sur l'animation faciale au cinéma.

Pour résumer, la problématique de notre projet est de produire un outil simple pour les animateurs, c'est-à-dire basé sur des glissoirs, comme ceux avec ils ont l'habitude de travailler aujourd'hui, pour réaliser beaucoup d'animation faciale, puisque le jeu vidéo est le média reposant le plus sur ce genre de données, en un minimum de temps.
Cela devrait entrainer un gain de productivité, puisque le temps passé à produire l'animation sera réduit à une simple correction d'erreurs, ainsi qu'un gain de qualité, puisque ce temps précédemment gagné pourra être réinvesti dans le polissage final des animations déjà présentes dans le jeu.

L'objectif final de notre recherche est donc d'améliorer l'ancien pipeline d'animation faciale utilisé actuellement dans l'industrie en termes de qualité et de productivité.

1.3 Plan du mémoire

Dans un premier temps, la section 2 présente l'état de l'art actuel de la recherche dans le domaine de l'animation faciale et du transfert d'expressivité.

La section 3 présente les contributions de ce mémoire. Tout d'abord sera introduite notre contribution majeure : la détermination de glissoirs d'animation sans intervention humaine par cinématique inversée dans l'espace d'animation de notre rig. Ensuite, nous y présenterons nos autres contributions, à savoir la génération de la pose neutre adaptative par RBF, le transfert de déformation, l'enrichissement du suivi par flux optique et la correction de la sortie utilisant les outils habituels des animateurs.

À travers la section 4, nous nous intéresserons aux résultats de notre nouveau pipeline, tout d'abord bruts, puis avec l'aide des images-clés de corrections réalisés par des artistes.

La section 5 discute des points forts et faibles de notre méthode, ainsi que des améliorations futures que l'on pourrait y apporter.

Enfin, la section 6 fait le point et conclue ce mémoire. Des résultats supplémentaires sont ensuite montrés en annexe.

CHAPITRE 2 REVUE DE LITTÉRATURE

Les travaux de recherches récents se focalisent sur deux processus distincts du pipeline d'animation faciale afin d'arriver à animer automatiquement un visage en 3D depuis une vidéo.

Tout d'abord, une partie des travaux cherchent à réaliser un transfert de déformation entre deux modèles de contexte semblables. Une extension de ces travaux mène au transfert d'expressivité, qui est le domaine qui nous intéresse dans ce mémoire.

Une autre partie des articles s'intéresse au problème de la reproduction de l'expression d'un visage par création et mélange d'un ensemble de blendshape basé sur les Facial Action Coding System (FACS) [Ekman et Friesen (1978)], un modèle de représentation des expressions du visage par paramètres permettant de reproduire conceptuellement la totalité des mouvements d'un visage humain.

2.1 Transfert d'expressivité

Un des points clés des méthodes de transfert d'expressivité est de réussir à transférer les expressions d'un acteur vers un modèle d'une physionomie de visage très différente, comme un animal, un monstre ou encore des objets. Rien ne nous indique que le modèle de sortie aura deux yeux, un nez et une bouche. Une solution est de ne pas essayer de transférer le mouvement en tant que tel, mais plutôt sa sémantique.

[Noh et Neumann (2001)] est un des premiers articles à s'intéresser au transfert d'expressivité. C'est également l'un des articles qui inspire le plus la méthode présentée dans ce mémoire. Une première étape y est de représenter l'expression comme une différence entre la pose neutre d'un modèle 3D facial animé et toutes les images de la vidéo dont est issu ce modèle. En transférant cette différence sur un autre modèle 3D, on reproduit l'animation.
Cette différence n'est toutefois pas naïvement reproduite : elle suit un morphage par un réseau RBF afin de respecter la physionomie du visage cible. Ainsi, on peut penser transférer une expression d'un humain à un renard, par exemple.
Également, une contrainte est ajoutée sur les sommets des lèvres intérieures afin de ne pas pouvoir les superposer, évitant ainsi les auto-occlusions.

[Na et Jung (2004)] réutilise en partie cette idée, mais la pousse plus loin : plutôt que d'avoir une correspondance entre les poses neutres uniquement, la correspondance se fait également

sur un sous ensemble de sept poses issues de FACS [Ekman et Friesen (1978)]. L'expression est donc transférée via une moyenne pondérée des différences entre l'animation entière et l'ensemble des poses d'apprentissage sur lesquelles on a correspondance.
La méthode présente également un moyen de récupérer des détails fins comme les rides. Passé l'étape de transfert de déformation, une différence des décalages des normales entre toutes les poses et la pose jugée comme étant la pose neutre est effectuée. Le modèle cible est alors déformé pour que les normales transférées soient orthogonales aux faces présentes. Cette étape permet de capturer les détails fins.
Le problème de cette méthode est que la vidéo du modèle 3D source pour le transfert d'expressivité est acquise via un dispositif lourd de lumière structurée, peu adapté aux besoins des caméras montées sur casque des acteurs de capture de mouvement dans le cinéma et le jeu vidéo.

[Sumner et Popović (2004)] propose une méthode pour copier automatiquement une déformation d'un maillage à un autre. Pour peu que les modèles d'entrées et de sorties possèdent une physionomie similaire, la méthode fonctionnera. L'utilisateur n'a qu'à indiquer un ensemble de points sur les deux modèles où la déformation est contrainte pour être similaire.
Pour chaque triangle de la source, la méthode calcule la transformation optimale entre sa pose neutre et sa pose déformée. Pour trouver la déformation optimale sur la cible et donc transférer au mieux la déformation, [Sumner et Popović (2004)] résout un problème d'optimisation en essayant de réutiliser ces ensembles de transformations, tout en imposant qu'un vertex partagé par des triangles différents de la cible doit se trouver au même endroit après déformation de chaque triangle. Ce problème se résout par un simple système d'équations linéaires.
Dans le cas d'un transfert de déformation entre deux modèles ne partageant pas le même nombre de triangles, un Iterative Closest Point (ICP) est réalisé afin d'associer les triangles de la source aux triangles de la cible. La résolution du problème est ensuite contrainte comme une minimisation de trois énergies différentes : l'énergie de déplacement, issue de la méthode pour un nombre de triangles cohérent, l'énergie de lissage, qui contraint le modèle à rester lisse après la déformation, et l'énergie d'identité, qui contraint les mouvements à ne pas être trop grands. En trouvant l'ensemble de transformations minimisant ces trois énergies, le modèle est déformé convenablement.

[Vlasic et al. (2005)] propose un transfert d'expressions faciales utilisant un tenseur d'ordre 3 dont les modes sont $\{Identités * Expressions * Visèmes\}$ sur une base de visages 3D et l'algèbre multilinéaire.

Le tenseur est tout d'abord rempli via la capture de performance d'un grand nombre de personnes réalisant différentes expressions et reproduisant les sons liés aux visèmes cherchés à être reproduits.
Afin de reproduire l'animation faciale depuis une source monocaméra, un suivi de points par flux optique est réalisé. Une descente de gradient sur les paramètres de chaque mode du tenseur (Identité, Expression, Visème) est alors réalisée afin de trouver le mélange des modes du tenseur donnant le résultat le plus proche du modèle cherché à être reproduit, en ne prenant en compte que les paramètres d'un mode du tenseur à la fois.

[Dutreve et al. (2008)] présente la méthode encore globalement communément utilisée de nos jours dans l'industrie : la réalisation d'une animation par réseau RBF.
Dans un premier temps, le réseau RBF est entrainé à partir d'un ensemble de marqueurs suivi sur le visage de l'utilisateur à l'aide d'un flux optique de Lucas-Kanade [Lucas et al. (1981)]. Chaque image issue du suivi de points est associée au modèle 3D qui lui correspond. Ce modèle 3D n'a pas besoin d'être de physionomie semblable au visage de l'utilisateur, on peut tout à fait utiliser un modèle d'animal ou d'objet pour ce transfert. L'ensemble des modèles doit néanmoins être cohérent en termes de topologie.
Afin de trouver comment reproduire l'expression d'une image inconnue, il suffit de projeter le suivi de points issu du Flux-Optique dans le réseau RBF. En sortira le mélange des modèles 3D d'apprentissage qui reproduit l'expression sur la cible au mieux.
Cette méthode est implémentée sur GPU et est relativement légère en termes de coût de calcul, rendant ainsi une utilisation temps réel possible. En effet, seul le mélange de modèles est coûteux, et le GPU gère très bien ce genre de calculs.

[Kholgade et al. (2011)] propose une méthode permettant de résoudre le problème du transfert d'expressivité entre deux modèles structurellement et sémantiquement dissimilaires
Le point central est de diviser l'expressivité du modèle et de l'acteur en trois couches distinctes : une couche d'émotion, une couche de parole et une couche de clignements pour les yeux. À chacune de ses trois couches s'ajoute un masque pour déterminer comment les mélanger à chaque image.
À partir d'un suivi de points «traditionnel» de capture de visage 3D basé sur des réflecteurs infrarouges, ainsi que d'un Modèle à Apparence Actif, le même découpage en couches est effectué pour chaque modèle de suivi de points issu de la capture de performance. Chaque paramétrisation de couche est ensuite effectuée à l'aide de distances à l'intérieur d'un n-simplexe, n étant le nombre de paramètres dans la couche courante. Les extrêmes du simplexe sont déterminés par des artistes pour le modèle cible, et par une performance préenregistrée

de poses prédéfinies pour l'acteur. Par exemple, trois des extrêmes du simplexe de la couche d'émotion seront «Dégout», «Peur» et «Joie».
Vu que les simplexes sont cohérents entre eux, trouver le paramétrage dans l'un implique de trouver le paramétrage dans l'autre. En projetant le modèle de l'acteur issu de la capture de mouvement dans les différents simplexes, on trouve donc les différentes couches.

[Bouaziz et Pauly (2014)] s'attaque à l'un des problèmes des méthodes de transfert d'expressivité basées sur de l'apprentissage (réseaux RBF, apprentissage profond, etc.) : la taille de l'ensemble d'apprentissage. En effet, afin d'avoir un modèle en sortie où l'expression est correcte, il convient d'avoir un grand nombre d'exemples ou un suivi de point est associé au modèle en sortie, fut-il généré via un ensemble de paramètres ou les points 3D directement.
L'article propose d'utiliser les images sans correspondances apprises pour enrichir l'ensemble d'apprentissage : à savoir les images issues de la capture de mouvements non annotées et les animations sur la cible déjà existantes.
Cette étape est rendue possible via l'apprentissage d'un espace partagé latent de façon semi-supervisée entre les données issues de la capture de mouvements et les paramètres d'animation faciale. Cet apprentissage est réalisé grâce à l'utilisation de Modèles Partagés de Processus Gaussien Latent Variables.
Cette méthode repose toutefois sur une hypothèse très forte. En effet, même si la taille de l'espace d'apprentissage nécessaire a été drastiquement réduite, cette méthode requiert tout de même la présence d'animation où la correspondance entre la performance vidéo capturée et les paramètres d'animation est faite. En d'autres termes, cette méthode requiert des animations déjà faites, ce qui n'est pas toujours disponible.

[Thies et al. (2016)] s'attaque à un problème encore plus complexe : le transfert de l'expressivité d'un visage humain à un autre dans des vidéos. Ce problème est plus complexe, car contrairement à un modèle 3D, une cible vidéo 2D photo-réaliste est extrêmement difficile à produire. En effet, un humain va perceptuellement immédiatement voir si un problème est présent dans le résultat.
Tout d'abord, l'identité de la source est récupérée à partir d'une simple source monocaméra 2D par regroupement non-rigide basé sur un modèle de Principal Components Analysis (PCA) multi-linéaire. Ensuite, les expressions faciales sont détectées sur les visages de la source et de la cible à l'aide d'une mesure de conformité photométrique dense.
Un transfert de déformation est alors produit entre la source et la cible, puis l'intérieur de la bouche est reproduit en trouvant dans la séquence cible l'intérieur de bouche pour laquelle l'expression était celle reproduite.

2.2 Générations de blendshapes spécifiques à l'utilisateur

Certains articles s'efforcent d'améliorer la création d'un modèle spécifique à l'utilisateur de blendshape. Ces articles consistent souvent en une étape de calibration entrainant la création d'un ensemble de Blendshapes, puis d'un mélange des Blendshapes à l'exécution.

2.2.1 Méthodes nécessitant une étape d'initialisation hors-ligne

Un des premiers articles à proposer une solution à ce problème est [Weise et al. (2009)]. Tout d'abord, le visage de l'acteur est suivi en 3D de façon fidèle grâce à un système de lumière structurée. L'acteur réalise ensuite un ensemble d'expressions, capturé par ce système. Ces numérisations d'expressions 3D sont ensuite transmises sur un modèle 3D générique par recalage non rigide. La dimensionnalité de cet espace est ensuite réduite par PCA pour ne garder qu'un sous-ensemble de paramètres.
L'ensemble de Blendshapes «cibles» est ensuite créé en transférant les déformations des composantes propres issues de la PCA vers le modèle cible souhaité en utilisant [Sumner et Popović (2004)].
Pour trouver le mélange de blendshape optimal à chaque instant est réalisé le même recalage sur un modèle générique qu'a l'étape de création de l'espace de Blendshape. La numérisation 3D «inconnu» recalée est ensuite projetée dans l'espace PCA. L'expression de ce visage dans l'espace PCA nous donne le mélange de Blendshape.
Les gros désavantages de cette méthode sont l'infrastructure nécessaire à la création des numérisations 3D et l'obligation de réaliser tout un ensemble d'expressions afin de les reproduire (due à l'utilisation d'un espace PCA). On ne peut en effet pas reproduire une expression non présente dans l'ensemble d'apprentissage. Par exemple si notre acteur n'a jamais réalisé de sourires lors de la capture des expressions et décide d'en faire un au moment de la capture de performance, ce sourire ne sera pas reproduit sur la cible.

Dans [Li et al. (2010)], publié l'année suivante, la même équipe améliore son modèle pour surmonter certains des problèmes évoqués plus haut. Plutôt que de générer un ensemble de Blendshape basé sur une PCA faite à partir d'expressions enregistrées, ce modèle génère un ensemble de Blendshape basée sur un ensemble de Blendshape générique préexistant
Cet ensemble de Blendshape générique est rendu spécifique à l'acteur en fournissant un sous-ensemble de numérisations 3D d'expressions d'apprentissage. Le transfert de déformation de l'ensemble de blendshapes générique vers l'ensemble spécifique à l'utilisateur est une fois de plus fait à l'aide de [Sumner et Popović (2004)].

Néanmoins, une étape supplémentaire est ajoutée afin de s'assurer que la transformation correspond bien au contexte du visage considéré à l'aide d'une régularisation basée sur la correspondance des gradients de déformation au sein d'un même blendshape entre les 2 ensembles. Cette régularisation utilise les exemples fournis pour trouver cette correspondance et faire en sorte qu'elle soit présente de la même manière sur les blendshapes sans exemples.

L'année suivante, la même équipe a retravaillé la partie de déterminations des poids sur les blendshapes dans [Weise et al. (2011)]. La partie de génération de l'ensemble de blendshape spécifique à l'utilisateur est similaire à [Weise et al. (2009)] et [Li et al. (2010)], à l'exception près que les numérisations de visages sont réalisées avec un simple capteur Kinect disponible au grand public.
La partie de détermination des poids est par contre très différente : [Weise et al. (2011)] s'est rendue compte que certains mélanges de blendshapes ne sont pas souhaitables, car ils ne permettent pas la réalisation de poses réalistes pour un humain, ce qui est perceptuellement très choquant pour un observateur et vient donc au détriment de la qualité de la reproduction de l'animation.
Afin de résoudre ce problème, l'article propose une régularisation des poids sur les blendshapes via un a priori appris sur des animations à partir de l'ensemble de Blendshape déjà existantes. Cet a priori est représenté comme un Mélange Probabiliste d'Analyseurs à Composantes Principales (MPACP), et le mélange optimal est ensuite inféré par Maximum a Posteriori (MAP).

[Cao et al. (2013)] propose pour la première fois de faire un reciblage à l'aide d'un ensemble de Blendshape à partir d'une source 2D. Dans cet article, un régresseur est entrainé à partir d'image 2D d'un acteur réalisant les expressions de l'ensemble de blendshapes et un ensemble de points 2D suivis sur ces visages. En projetant les mêmes points issus d'un ensemble de blendshapes 3D en 2D et réalisant une déformation rigide contrainte, l'ensemble de blendshape spécifique à l'utilisateur est créé.
Un régresseur est ensuite entrainé pour associer les bonnes valeurs de mélange de blendshape à des images connues. À partir de cet apprentissage, il est alors possible de déterminer le mélange de blendshape optimal depuis une source 2D en utilisant le régresseur entrainé. Puisque tous les modèles utilisés utilisent le même espace de Blendshape, il est aisé de transférer l'animation.

2.2.2 Méthodes sans initialisation en ligne

L'année 2013 fut prolifique vis-à-vis de la recherche en ce domaine. Dans [Li et al. (2013)], l'équipe de retargeting facial d'Industrial Light and Magic (ILM) améliore [Weise et al. (2011)] avec une première approche ne nécessitant pas d'initialisation.
Cette méthode commence par utiliser une numérisation 3D du visage de l'acteur en pose neutre, elle génère ensuite l'ensemble de Blendshape «initial» par la méthode présentée dans [Li et al. (2010)]. De cet ensemble est déduit un modèle linéaire par PCA adaptative. Ce modèle consiste en un ensemble de «anchor shapes», les blendshapes initiaux, et de «correctives shapes», qui permettent de découvrir les poses qui ne sont pas dans l'ensemble de départ et de corriger celles qui y sont déjà.
Pour se faire, le résultat de la PCA adaptative est déformé vers la carte de profondeur en entrée (fournis par une caméra Kinect, comme dans [Weise et al. (2011)]) à l'aide de 40 points d'intérêts faciaux suivis sur le visage par déformation laplacienne par vertex.

[Bouaziz et al. (2013)] a convergé vers une solution similaire dans l'idée. Tout comme dans [Li et al. (2013)], ce modèle se base sur une PCA pour la gestion de l'identité. L'ensemble de blendshapes est ensuite déduit de cette identité via la méthode de [Li et al. (2010)] et amélioré grâce à un champ de déformation Laplacien.
Mais contrairement à [Li et al. (2013)], [Bouaziz et al. (2013)] optimise ses blendshapes manquants sur son «Modèle Dynamique d'Expression», qui regroupe à la fois l'identité, les blendshapes et les champs de déformations. Plutôt que de n'optimiser le résultat que sur les déformations, le résultat est optimisé sur toutes ces valeurs. Le champ de déformation ne requiert également pas de suivi de points 2D et utilise le visage 3D reproduit à l'image précédente comme entrée considérée comme «correcte».

Dans [Cao et al. (2014a)], Chen Cao et son équipe améliorent leur précédente méthode [Cao et al. (2013)] en permettant de se passer de l'étape d'initialisation. Le régresseur précédemment utilisé est enrichi pour prendre en compte ce que l'équipe appelle les Displaced Dynamic Expression (DDE), une fonction qui fait correspondre un ensemble de paramètres inconnus (identité, expressivité, rotation globale, translation globale) à une image 2D enrichie d'un suivi de points 2D.
Le régresseur DDE est appris sur la base de données Facewarehouse [Cao et al. (2014b)], constituée de 150 visages en 3D sous différentes poses avec les photos 2D des acteurs sous ces mêmes positions sous différents angles de vues associés.
En utilisant le régresseur sur une image inconnue, il est possible de reproduire l'expression

du visage en 3D.

[Garrido et al. (2013)] propose également une méthode de retargeting facial utilisant des Blendshapes depuis une source monocaméra 2D sous des conditions d'éclairages non contraintes. Cette méthode est composée de quatre étapes distinctes.
Tout d'abord, l'ensemble de Blendshapes est créé à partir d'un même ensemble réalisé par un artiste et d'une reconstruction 3D binoculaire du visage de l'acteur cherché à être reproduit. Le transfert de déformation de l'ensemble générique à la numérisation du visage est réalisé tout d'abord en alignant 29 points en 3D du Blendshape neutre et de la numérisation, puis en réalisant une recherche de correspondance globale et une déformation laplacienne régularisée. Le reste de l'ensemble de Blendshapes est ensuite transmis en utilisant la méthode proposée par [Li et al. (2010)].
Ensuite, un suivi d'un ensemble de points 2D est effectué sur la source vidéo à partir d'une méthode de suivi de visage probabiliste régularisée par un modèle de visage 3D. Le mélange de Blendshapes et la transformation globale sont obtenus par minimisation de l'énergie des moindres carrés sur la distance entre les points suivis depuis la source vidéo et les mêmes points sur le modèle de Blendshapes mélangés.
La troisième étape est de corriger les expressions fausses causées par le manque d'information dans le suivi de points. En effet, certains points d'intérêt du visage ne sont pas suivis, notamment les joues. Il faut alors corriger le modèle issu de l'étape précédente afin de prendre en compte ces déformations. [Garrido et al. (2013)] propose de calculer un flux optique dense sur la vidéo capturée et sur une vidéo synthétique, issue de l'étape précédente. En trouvant la déformation totale qui minimise la différence entre les deux flux, [Garrido et al. (2013)] corrige le modèle.
Une dernière étape rajoute les détails les plus fins tels que les rides via une approche de «Forme depuis ombrage» (Shape from Shading). Cette étape est réalisée en adaptant la méthode de [Valgaerts et al. (2012)] à une source monoculaire.

Récemment, [Garrido et al. (2016)] a continué l'approche de [Thies et al. (2015)] en la rendant compatible aux entrées monoculaires 2D : tout d'abord, un modèle d'identité est appris sur une vidéo 2D, comprenant la géométrie, la texture, l'albédo. Ce modèle est ensuite transformé en un ensemble de Blendshapes.
Cet ensemble de Blendshapes est enrichi par une approche similaire à la troisième étape de [Garrido et al. (2013)] pour obtenir un modèle plus fin. Enfin, toujours de façon similaire à [Garrido et al. (2013)], les détails les plus fins tels que les rides sont ajoutées au modèle via l'ombrage sur la vidéo en entrée.

Cette méthode s'avère robuste même sur des vidéos bruitées comme des extraits de films des années 80 ou des vidéos Internet.

2.3 États de l'art pour suivi de points 2D/3D

On remarque à travers la lecture des articles précédents que l'étape d'initialisation est des plus importantes : il convient d'avoir une méthode permettant de suivre les mouvements du visage en 3D. Certains résolvent ce problème en utilisant des procédés de capture RGBD ou par lumière structurée, mais ces procédés sont complexes.
L'autre méthode est de suivre un ensemble de points en 2D ou en 3D sur la vidéo 2D monocaméra capturée lors de la performance de l'acteur, et de trouver la correspondance entre ces points et le modèle 3D, ou encore de déterminer à partir de cette source 2D quelle est la composante de profondeur manquante.
Nous allons ici nous intéresser à deux états de l'art en suivi de points 2D et 3D sur un visage.

Le défi «300 Videos-in-the-Wild», dont les résultats sont présentés dans [Shen et al. (2015)], repose entièrement sur ce problème. Le but du défi est de trouver le meilleur algorithme permettant de suivre 68 points sur des visages extraits de vidéos d'une minute environ.
Les vidéos sont triés selon trois scénarii de difficultés croissantes : le premier présente des vidéos dans un environnement à éclairage contrôlé, sans grands mouvements de tête ni occlusions. C'est le cas le plus proche d'une capture en laboratoire. Le second scénario présente des vidéos dans des situations d'éclairages réelles et avec des mouvements de tête non contrôlés. Ce scénario évite toutefois les occlusions. Le dernier scénario présente le reste des vidéos, les plus difficiles pour le suivi de points, avec un éclairage non contrôlé et présence d'occlusions. Les méthodes présentant les meilleurs résultats pour les 3 problèmes sont [Xiao et al. (2015)] et [Yang et al. (2015)].

[Xiao et al. (2015)] propose un processus basé sur une régression multi étape. Cette régression se base sur une initialisation de certains points clés dans des coins forts de l'image (coins de la bouche et des yeux) puis détermine toujours plus de points à chaque étape jusqu'à avoir déterminé la position de tous les points suivis. Cette méthode s'avère très robuste aux occlusions et différences d'éclairages.

[Yang et al. (2015)] utilise une méthode basée sur une régression de forme spatio-temporelle en cascade. La régression en cascade permet de réduire la variance des entrées, tandis que la régression par séries temporelles assure une transition douce entre deux images dans la vidéo. Cette méthode possède également un nouveau détecteur de visage pour réinitialiser

le suivi en cas de perte de détection. Elle est également très robuste aux occlusions et aux changements d'éclairages.

«The First 3D Face Alignment in the Wild (3DFAW) Challenge» est la suite du défi précédent. Puisque le suivi de points en 2D semble être un problème «résolu», l'étape logique suivante était de proposer un suivi de points en 3D depuis des sources vidéos 2D. Les résultats du défi sont présentés dans [Jeni et al. (2016)]. Le but y était de comparer le suivi des mêmes 68 points du défi précédent en 3D comparée à la réalité du terrain basée sur des numérisations 4D tels que «Multi-PIE» [Gross et al. (2010)] ou «BP4D-Spontaneous» [Zhang et al. (2014)]. La plupart des participants aux défis ont utilisé des approches d'apprentissages profonds par réseaux de neurones convolutionnels. Les deux articles présentant les meilleurs résultats sont, quasi à égalité parfaite, [Bulat et Tzimiropoulos (2016)] et [Zhao et al. (2016)].

[Bulat et Tzimiropoulos (2016)] utilise une approche à deux étapes : la première calcule des cartes de chaleur de probabilité de présence du point suivi dans l'image via régression par réseaux convolutionnels très profonds. La deuxième détermine la profondeur en utilisant ces cartes de chaleurs et l'image RGB initiale et en les mettant en entrée d'un réseau convolutionnel profond. Cette méthode a gagné le défi 3DFAW et est donc considérée état de l'art en suivi de point 3D sur un visage.

[Zhao et al. (2016)] quant à elle, utilise un réseau de neurones convolutionnel profond qui fait correspondre directement l'image 2D d'un visage à ses points suivis en 3D. Comme [Bulat et Tzimiropoulos (2016)] cette méthode fonctionne en deux étapes, la première servant à déterminer la position des points en 2D et la 2e déterminant la profondeur du point.
Cette méthode est 22% moins robuste que [Bulat et Tzimiropoulos (2016)], mais plus rapide à l'exécution.

Ces deux méthodes s'avèrent robustes à l'occlusion, les rendant potentiellement très intéressantes à l'utilisation dans des cas extrêmes. Néanmoins, comme toute solution reposant sur de l'apprentissage profond, elles requièrent un coût d'apprentissage très élevé.
En effet, ces méthodes requièrent d'avoir énormément de données d'apprentissage, et ces données d'apprentissages se doivent de capturer convenablement l'essence du problème, à savoir la multitude des cas de résolutions possibles. Aussi, entrainer des réseaux de neurones convolutionnels profonds demande un temps de calcul considérable.

2.4 Hypothèses de recherche

Notre question de recherche est de savoir comment arriver à capturer l'expressivité d'un acteur et comment la transmettre ensuite sur un modèle virtuel. Bien que des avancées des plus intéressantes aient été réalisées dans l'animation faciale à base d'ensemble de Blendshapes ou de transfert d'expressivité direct, les problèmes inhérents à ces deux sujets subsistent.

Tout d'abord, les méthodes de transfert d'expressivité direct ne permettent généralement pas de corrections de la part d'artistes puisque la sortie y est un modèle 3D et non un ensemble de paramètres générant ce modèle 3D ([Na et Jung (2004)], [Sumner et Popović (2004)], [Dutreve et al. (2008)]).
Les articles les permettant sont basés sur un ou plusieurs ensembles de Blendshape ([Vlasic et al. (2005)], [Kholgade et al. (2011)], [Thies et al. (2015)]). [Vlasic et al. (2005)] ne déclare pas utiliser un ensemble de Blendshape puisque la parution de l'article est antérieure à l'émergence du concept de Blendshape, mais en effet leur tenseur $\{Identité, Expression, Visème\}$ de visages en 3D est un ensemble d'ensembles de Blendshape dont l'identité varie.

Même s'il permet des retouches de la part d'artistes, un mélange sur un ensemble de Blendshape propose moins de contrôle à un artiste qu'un modèle basé sur un squelette. En effet, un ensemble de Blendshape représente un espace de modèle 3D. Réaliser une expression dans cet espace revient à trouver les coordonnées d'un point dans l'espace dont les dimensions sont les expressions apprises, comme le présente [Kholgade et al. (2011)]. Il devient alors complexe pour un artiste de se représenter ce mélange d'un nombre parfois très important de modèles 3D, qui peut atteindre des milliers, comme c'est le cas avec la réutilisation d'anciennes animations de [Bouaziz et Pauly (2014)].
De plus, la sémantique de ces extrémums peut parfois être difficile à cerner pour un être humain, puisqu'elle peut être basée sur les composantes principales issues d'une PCA, comme c'est le cas dans [Weise et al. (2009)].

Notre volonté est donc d'adapter ces méthodes tout en gardant les avantages que présente un modèle basé sur un squelette suivi d'une opération d'empeçage. Nous souhaitons donc trouver les paramètres d'animations, les «glissoirs», qui sont une couche d'abstraction des transformations des joints du squelette, par opérations de cinématique inversée.

Notre première hypothèse, qui est l'hypothèse majeure sur laquelle repose la contribution principale de ce mémoire, est donc que l'on peut abstraire les mouvements des joints du squelette par leurs composantes de translation uniquement. En effet puisque les angles de rotations sur le visage sont faibles, les négliger ne devrait pas avoir d'impact trop important sur la résolution du problème de cinématique inversée sur la couche d'abstraction des

glissoirs. Au contraire, les négliger permettra de simplifier grandement le problème, l'espace d'animation devenant multilinéaire.

Notre seconde hypothèse est qu'un suivi de points sur un visage en 3D comme ceux présentés dans [Jeni et al. (2016)] nous permettra d'obtenir une abstraction des mouvements et de l'expression du visage de l'acteur. En transmettant ces mouvements de points sur les mêmes points de notre modèle à animer, d'une manière similaire à [Noh et Neumann (2001)], nous produirons une cible atteignable sur notre Rig. Afin de rendre notre système indépendant de la morphologie des visages sources ou cibles, nous ferons l'hypothèse que la variété des visages humains possibles se trouve dans des déformations locales de points. Nous envisagerons donc d'aligner les visages par zones pour produire notre cible.

Ensuite, pour ne pas avoir à choisir une pose neutre, puisque rien ne nous indique qu'elle existe dans notre performance capturée, nous ferons l'hypothèse qu'un mélange linéaire des positions des points suivis dans le temps nous permet de reconstruire une pose neutre adaptée. Nous projetterons la pose neutre de notre Rig dans le réseau RBF constitué de toutes nos positions de points suivis dans le temps issues de la capture de performance, le mélange de ces positions ainsi produit nous donnera notre pose neutre de la performance.

D'une manière similaire à [Garrido et al. (2013)], nous réaliserons une minimisation de l'énergie des moindres carrés sur la distance, à chaque image, entre les points suivis et la cible générée. La différence fondamentale reposera sur la résolution de ce problème, qui dans notre cas est un problème de cinématique inversé, puisque l'on cherche à trouver une position du squelette qui minimise cette distance, et non un mélange de modèles 3D.

Une troisième hypothèse est que certaines poses ne sont pas atteignables par le Rig basé sur un squelette du fait des mouvements cinématiques que permet le squelette. Un alignement non plus sur la pose neutre, mais sur chaque image du suivi de point, par zone, pendant le processus itératif de détermination des glissoirs devrait nous éviter ce problème et nous éviter de trouver un minimum local, par opération de recuit.

Combiner ces différentes idées et les adapter à notre problème nous permettra d'amener les avancées académiques du transfert d'expressivité dans des modèles se passant de Blendshape, facilement corrigibles par des animateurs professionnels.

2.5 Objectifs de recherche

L'objectif principal de cette maitrise a été de définir et mettre en place un nouveau pipeline pour la capture de performance sur des visages en 3D qui donne les paramètres d'animations sur un rig de visage 3D préexistant, tout en permettant aux artistes animateurs de retravailler

cette sortie à leur guise via l'utilisation des mêmes outils et interfaces qu'ils ont l'habitude d'utiliser.

Nos objectifs de recherches sont donc :

1. Trouver les paramètres d'animations associés à un rig de visage correspondant au déplacement d'un ensemble de points 3D sur un visage par cinématique inversée.
   a. Ajuster un ensemble de points en 3D en modifiant les paramètres d'animations afin d'avoir un minimum de distance entre les points 3D sur le visage de l'acteur et ces mêmes points sur le modèle 3D cible.
   b. Permettre aux artistes de corriger les résultats de cette méthode à leur guise.

2. Transmettre l'expressivité de la performance de l'acteur vers le rig afin de produire une cible atteignable par notre opération de cinématique inversée.
   a. Créer un outil permettant d'établir la correspondance entre un ensemble de points suivis dans la performance de notre acteur et le même ensemble de points sur le rig cible afin de pouvoir se placer dans le même repère.
   b. Produire une pose neutre dans l'espace du suivi de points qui correspond à la pose neutre du rig cible afin d'avoir une même base pour les déformations.
   c. Réaliser un alignement des poses neutres par zone afin de prendre en compte la diversité des visages humains possibles, en source comme en cible du pipeline.
   d. Réaliser un alignement image par image lors du processus de détermination des paramètres d'animations afin de s'assurer que nos cibles soient atteignables et mettre en place un système de simulation de recuit pour éviter les minimas locaux.

CHAPITRE 3 MÉTHODOLOGIE

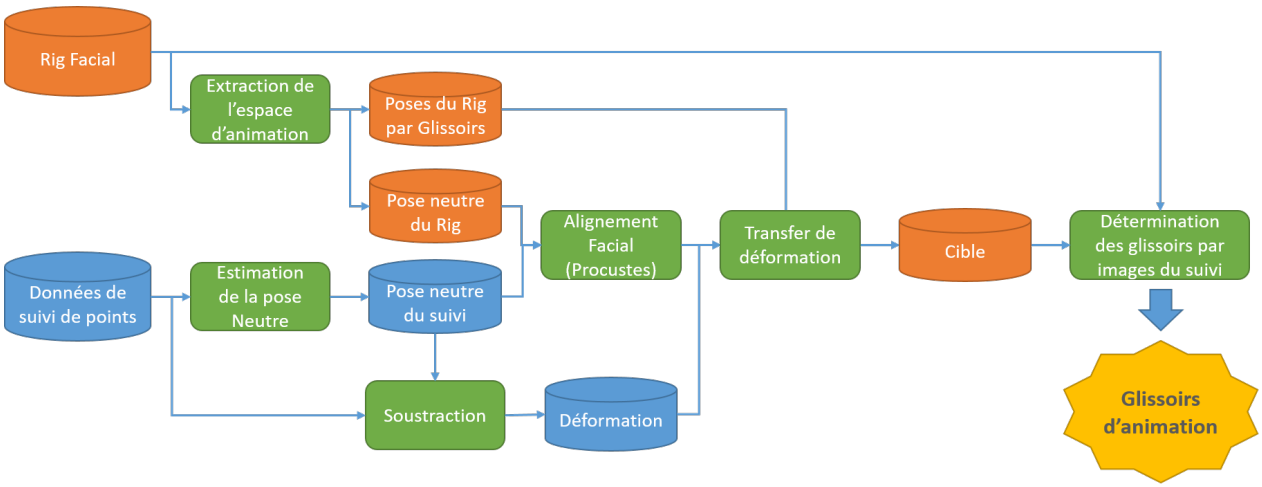

Figure 3.1 Pipeline de notre méthode de transfert d'expressivité.

Le corps principal de notre méthode est présenté dans le pipeline de la figure 3.1. À partir d'un suivi de points 3D et d'un Rig facial, notre méthode détermine sans intervention humaine les glissoirs d'animation adaptés à la performance de l'acteur enregistrée en appliquant des principes de cinématique inversée sur une couche d'abstraction des mouvements possibles de notre squelette d'animation.

La section 3.1 présente la contribution majeure de ce mémoire : à savoir la détermination sans intervention humaine des glissoirs d'animations optimaux dans l'espace d'animation du rig par descente de gradient.

La section 3.2 présente comment le suivi de points est réalisé, la façon de l'enrichir et le processus à suivre pour le lier au rig facial sur lequel on souhaite reproduire l'animation. En effet, notre méthode repose grandement sur la qualité du suivi mis à notre disposition et de l'étape de correspondance réalisée par un humain.

Afin de pouvoir réaliser notre descente de gradient et de trouver nos glissoirs optimaux, nous avons besoin d'une cible à chaque image de la performance. Un alignement facial est donc réalisé entre les deux poses neutres du suivi de points et du Rig par analyse Procustéenne. S'en suit un transfert de déformation afin de générer cette cible atteignable dans l'espace du Rig. Le détail de la méthode générant ces poses neutres et cet alignement se retrouvera dans la section 3.3.

Puisque le résultat du pipeline est un ensemble de glissoirs variant dans le temps, un artiste pourra facilement le retravailler. En effet il s'agit du genre de données avec lesquelles il est habitué à travailler. Nous verrons dans la section 3.4 comment procéder pour corriger et retravailler la sortie à l'aide d'un réseau RBF.

3.1 Trouver les meilleurs glissoirs par descente de gradient

La contribution majeure de ce mémoire se trouve dans cette étape, qui permet de réaliser une descente de gradients dans l'espace des glissoirs afin de trouver les meilleures valeurs de ces paramètres d'animations pour une performance donnée.

3.1.1 Notation tensorielle utilisée

La plupart des notions utilisées dans notre pipeline (espace d'animations, performance en suivi de point, performance sur le rig, ...) peuvent aisément être expliquées grâce à une notation tensorielle.

On définit l'opérateur [.] sur un tenseur comme l'opérateur de sélection d'un tenseur d'ordre $n-1$ dans un tenseur d'ordre n sur son dernier mode.
Par exemple, définissons un tenseur T de dimensions $\{A, B, C\}$. On aura :

$$\forall i \in [1, C] \quad | \quad T[i] = \begin{bmatrix} T_{(1,1,i)} & T_{(1,2,i)} & \cdots & T_{(1,B-1,i)} & T_{(1,B,i)} \\ T_{(2,1,i)} & T_{(1,2,i)} & \cdots & T_{(1,B-1,i)} & T_{(1,B,i)} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ T_{(A-1,1,i)} & T_{(A-1,2,i)} & \cdots & T_{(A-1,B-1,i)} & T_{(A-1,B,i)} \\ T_{(A,1,i)} & T_{(A,2,i)} & \cdots & T_{(A,B-1,i)} & T_{(A,B,i)} \end{bmatrix} \tag{3.1}$$

Cette notion de sélection tensorielle simplifiera grandement nos calculs par la suite.

3.1.2 Notions et vocabulaire utilisé

On appellera $\tilde{V}$ l'ensemble des points suivis sur la performance et V ce même ensemble de points sur le Rig. L'association entre ces deux points est réalisée dans l'étape présentée dans la partie 3.2.1.
En utilisant la notation présentée en 3.1.1, $\tilde{V}$ et V sont deux tenseurs d'ordre 3 dont les modes sont $(coordonnées * points * temps)$. Les dimensions de ces tenseurs sont donc le nombre de dimensions de l'espace 3, le nombre de points suivis K, et le nombre d'images de la

performance P. Il est important de noter l'ordre dans lequel sont définis les modes de notre tenseur. Un tenseur de modes ($coordonnées * points * temps$), comme celui du suivi de points issu de la performance de l'acteur, n'est pas équivalent à travers notre notation à un tenseur de modes ($temps * points * coordonnées$).

En utilisant l'opération de sélection dans un tenseur présentée à la partie 3.1.1, on pourra donc trouver l'ensemble de points suivi à une image donnée en réalisant l'opération :

$$V[j] = \begin{bmatrix} V_{(1,1,j)} & V_{(1,2,j)} & \cdots & V_{(1,K-1,j)} & V_{(1,K,j)} \\ V_{(2,1,j)} & V_{(2,2,j)} & \cdots & V_{(2,K-1,j)} & V_{(2,K,j)} \\ V_{(3,1,j)} & V_{(3,2,j)} & \cdots & V_{(3,K-1,j)} & V_{(3,K,j)} \end{bmatrix} \tag{3.2}$$

Dans la même optique, on pourra aussi récupérer le point i de l'image j comme tel :

$$V[j][i] = \begin{bmatrix} V_{(1,i,j)} \\ V_{(2,i,j)} \\ V_{(3,i,j)} \end{bmatrix} \tag{3.3}$$

Introduisons maintenant notre notion de RigMorphs. Un RigMorph est un ensemble de positions du squelette associé à différents états d'un glissoir. Dans un même RigMorph, les positions du squelette sont interpolées linéairement entre deux états. Chaque RigMorph possède un état 0 dans lequel la position du squelette est celle qui génère $Neutre_{Rig}$.

En supposant les rotations des joints du squelette associé au rig faibles, on fait l'hypothèse que les positions des points du maillage relié à ce squelette par empeçage peuvent également être interpolées linéairement entre les différents états. Cette hypothèse forte est au centre de notre méthode et de notre réflexion.

On pourra alors sous cette hypothèse représenter un RigMorph comme un tenseur R d'ordre 3 ($coordonnées * points * états$) associé à une liste d'état B. On introduira alors une méthode pour générer un modèle 3D dans un RigMorph de $L+1$ états sous un glissoir de valeur x comme :

$$g(R, B, x) = \begin{cases} R[0] & si \quad x < B_0 \\ \frac{x-B_l}{B_{l+1}-B_l} \cdot R[l] - \frac{x-B_{l+1}}{B_{l+1}-B_l} \cdot R[l+1] & si \quad B_l < x < B_{l+1} \\ R[L] & si \quad x \geq B_L \end{cases} \tag{3.4}$$

On remarquera que l'on peut inclure notre notation de RigMorph dans un tenseur A de modes ($coordonnées * points * états * glissoirs$) d'un ordre plus élevé pour ainsi définir notre espace

d'animation. Cet espace d'animation est lié aux valeurs des états par glissoirs B, on définit donc notre espace d'animation comme $\{A, B\}$.

En reprenant les notations de la génération de la pose neutre de la partie 3.3.1, on peut maintenant définir la fonction f qui génère l'expression faciale à partir d'un ensemble de N glissoirs comme la sommation des déformations dues à chaque valeur de glissoir S_j dont la liste d'état est $B[j]$:

$$f(\{A, B\}, S) = Neutre_{Rig} + \sum_{j=1}^{N} \left[g(A[j], B[j], S_j) - Neutre_{Rig} \right] \tag{3.5}$$

3.1.3 Minimisation d'énergie

Comme introduit dans la partie précédente, le but de la méthode est de minimiser la distance entre les points suivis $\tilde{V}$ et les points sur le rig V, dont les positions vont varier en fonction des glissoirs, pour toutes les images t de la performance de durée Z. On reformulera donc le problème sous la forme d'une minimisation d'énergie des moindres carrés.

$$E = \sum_{t=1}^{Z} R_t^2 = \sum_{t=1}^{Z} \left\| \tilde{V}[t] - V[t] \right\|_2 \tag{3.6}$$

Or, notre animation du côté du rig est une succession d'ensemble de valeurs de glissoirs variants dans le temps, qu'on a donc choisi de représenter sous la forme d'un tenseur S de modes ($glissoirs * temps$). Ces glissoirs permettent de récupérer une position de points dans notre espace d'animation $\{A, B\}$ à l'aide de l'équation 3.5. On a donc :

$$E = \sum_{t=1}^{Z} \left\| \tilde{V}[t] - f(\{A, B\}, S[t]) \right\|_2 \tag{3.7}$$

C'est sur ce tenseur S que l'on souhaite agir afin de converger vers notre solution. Le problème est un problème de minimisation d'énergie des moindres carrés, la solution est connue et consiste à trouver le gradient des paramètres agissant sur cette distance. La direction dans laquelle faire varier S pour réduire l'énergie est donc donnée par l'équation suivante :

$$\nabla S = \frac{\delta E}{\delta S} = 2 \sum_{t=1}^{Z} R_t \frac{\delta R_t}{\delta S} \tag{3.8}$$

Or S est un tenseur d'ordre 2, on peut donc écrire cette dérivée comme tel :

$$\frac{\delta R_t}{\delta S} = \begin{bmatrix} \frac{\delta R_t}{\delta S_{1,1}} & \cdots & \frac{\delta R_t}{\delta S_{1,t}} & \cdots & \frac{\delta R_t}{\delta S_{1,Z}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\delta R_t}{\delta S_{j,1}} & \cdots & \frac{\delta R_t}{\delta S_{j,t}} & \cdots & \frac{\delta R_t}{\delta S_{j,Z}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\delta R_t}{\delta S_{N,1}} & \cdots & \frac{\delta R_t}{\delta S_{N,t}} & \cdots & \frac{\delta R_t}{\delta S_{N,Z}} \end{bmatrix} \tag{3.9}$$

Il convient de trouver ces dérivées partielles. Puisque la position des points cibles $\tilde{V}$ est indépendante des glissoirs, on a :

$$\frac{\delta R_t}{\delta S_{j,t}} = \frac{\delta \left\| \tilde{V}[t] - f(\{A, B\}, S[t]) \right\|_2}{\delta S_{j,t}} = -\frac{\delta f(\{A, B\}, S[t])}{\delta S_{j,t}} \tag{3.10}$$

On remarquera déjà que puisque l'image considérée t n'est pas un paramètre de la fonction déplaçant les points sur le maillage en fonction des glissoirs f, toutes les colonnes k de la matrice des dérivées partielles de l'équation 3.9 où $t \neq k$ seront nulles.

L'équation 3.5 nous donne l'expression de la fonction déplaçant les points sur le maillage f, on peut donc dériver facilement :

$$\frac{\delta f(\{A, B\}, S[t])}{\delta S_{j,t}} = \frac{\delta \sum_{i=1}^{N} g(A[i], B[i], S_{i,t})}{\delta S_{j,t}} = \frac{\delta g(A[j], B[j], S_{j,t})}{\delta S_{j,t}} \tag{3.11}$$

En effet, la dérivée sera nulle pour tout glissoir d'index différent de j.

En dérivant l'équation 3.4, on trouve :

$$\frac{\delta g(R, B, x)}{\delta x} = \begin{cases} 0 & si \quad x < B_0 \\ \frac{R[l+1]}{B_{l+1} - B_l} - \frac{R[l]}{B_{l+1} - B_l} & si \quad B_l < x < B_{l+1} \\ 0 & si \quad x \geq B_L \end{cases} \tag{3.12}$$

En injectant l'équation 3.12 dans 3.11, on a :

$$\frac{\delta R_t}{\delta S_{j,t}} = -\frac{\delta f(\{A, B\}, S[t])}{\delta S_{j,t}} = \begin{cases} 0 & si \quad S_{j,t} < B[j]_0 \\ \frac{A[j][l]}{B[j]_{l+1} - B[j]_l} - \frac{A[j][l+1]}{B[j]_{l+1} - B[j]_l} & si \quad B[j]_l < S_{j,t} < B[j]_{l+1} \\ 0 & si \quad S_{j,t} \geq B[j]_L \end{cases} \tag{3.13}$$

Les dérivées partielles de la fonction d'erreur ne sont que de simples fonctions constantes par morceaux. On pourra donc très facilement remplir la matrice des dérivées partielles présentée dans l'équation 3.9.

En reprenant la remarque faite sur l'équation 3.10, on remarquera également que l'équation de calcul du gradient ∇S de l'équation 3.8 se simplifie beaucoup. En effet, on somme dans cette équation sur t, mais pour toutes $k \neq t$ les valeurs des colonnes k seront nulles. La sommation ne remplira alors que les colonnes t une par une dans cette matrice. On pourra développer cette étape ainsi, afin d'obtenir l'expression simplifiée de détermination de notre gradient :

$$\nabla S = 2\sum_{t=1}^{Z} R_t \frac{\delta R_t}{\delta S} = 2\begin{bmatrix} R_1\frac{\delta R_1}{\delta S_{1,1}} & \cdots & R_t\frac{\delta R_t}{\delta S_{1,t}} & \cdots & R_Z\frac{\delta R_Z}{\delta S_{1,Z}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ R_1\frac{\delta R_1}{\delta S_{j,1}} & \cdots & R_t\frac{\delta R_t}{\delta S_{j,t}} & \cdots & R_Z\frac{\delta R_Z}{\delta S_{j,Z}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ R_1\frac{\delta R_1}{\delta S_{N,1}} & \cdots & R_t\frac{\delta R_t}{\delta S_{N,t}} & \cdots & R_Z\frac{\delta R_Z}{\delta S_{N,Z}} \end{bmatrix} \tag{3.14}$$

Le gradient du tenseur de glissoirs pourra très simplement se remplir en récupérant la distance à chaque image de la performance entre le suivi de points et les points du rig et en multipliant cette distance aux constantes issues de la dérivée de f, s'obtenant par une simple sélection dans un espace précalculé.

3.1.4 Descente de gradient et recherche dichotomique

Une fois le gradient des glissoirs récupérés, il convient de faire varier ces derniers afin de réduire notre distance. On applique la formule

$$S := S - \lambda \cdot \nabla S \tag{3.15}$$

Comme dans toute descente de gradient, le problème est alors de déterminer le pas λ optimal minimisant notre énergie. En d'autres termes, on cherche à savoir de combien «descendre» dans la direction de ∇S.

Pour faire cela, on déterminera le pas λ par un processus dichotomique récursif. Pour cela on va chercher à trouver de quel μ faire bouger λ pour être optimal. On arrêtera la récursion lorsque μ tombera en dessous d'un seuil ε jugé suffisant. Empiriquement, $\varepsilon = 0.005$ est un bon compromis entre temps d'exécution et qualité visuelle en sortie du pipeline.

Algorithme 1 Recherche dichotomique du pas λ

$\lambda \leftarrow 1$
$\mu \leftarrow \lambda$
$distActuelle \leftarrow \infty$
function RechercheLineaire(λ, μ, $distActuelle$)
  if $\mu < \varepsilon$ **then**
    return λ
  else
    $\mu \leftarrow \mu/2$
    $distGauche \leftarrow$ CalculDistance($S - (\lambda - \mu)\nabla S$)
    $distDroite \leftarrow$ CalculDistance($S - (\lambda + \mu)\nabla S$)
    if $distGauche < distDroite$ et $distGauche < distActuelle$ **then**
      $distActuelle \leftarrow distGauche$
      $\lambda \leftarrow$ RechercheLineaire($\lambda - \mu$, μ, $distActuelle$)
    else if $distDroite < distGauche$ et $distDroite < distActuelle$ **then**
      $distActuelle \leftarrow distDroite$
      $\lambda \leftarrow$ RechercheLineaire($\lambda + \mu$, μ, $distActuelle$)
    else
      $\lambda \leftarrow$ RechercheLineaire(λ, μ, $distActuelle$)
    end if
    return λ
  end if
end function

Dans l'algorithme 1, la fonction «CalculDistance» réalise l'opération présentée dans l'équation 3.7 sous les paramètres d'animations S qui lui sont passés.

Afin de retrouver les meilleures valeurs de glissoirs S qui minimisent l'énergie des moindres carrés et reproduisent la performance faciale de notre acteur, on réalise l'algorithme suivant :

Algorithme 2 Détermination des glissoirs optimaux S

$distActuelle \leftarrow \infty$
$\Delta dist \leftarrow \infty$
$\lambda \leftarrow 1$
$iter \leftarrow 0$
while $\Delta dist > \gamma$ **do**
  majPose(S)
  $iter \leftarrow iter + 1$
  if $(iter \,\%\, facteurRealignement) = 0$ **then**
    Realignement( )
  end if
  $distPrev \leftarrow distActuelle$
  $distActuelle \leftarrow$ CalculDistance(S)
  $\Delta dist \leftarrow |distPrev - distActuelle|$
  $\nabla S \leftarrow$ CalculGradient(S)
  $\lambda \leftarrow$ RechercheLineaire(λ, λ, $distActuelle$)
  $S \leftarrow S - \lambda \nabla S$
  $\lambda \leftarrow 2\lambda$
end while

Dans l'algorithme 2, la fonction «Realignement» correspond à l'application de l'équation 3.30. La fonction «majPose» correspond à l'application de l'équation 3.5 sous les paramètres d'animations S. On considère l'espace d'animation $\{A, B\}$ déjà connu par notre système. La fonction «CalculGradient» correspond quant à elle à l'application de l'équation 3.14.

Empiriquement, on a déterminé que $\gamma = \frac{10}{Z}$, avec Z le nombre d'images dans la performance, était un critère d'arrêt offrant un compromis acceptable en termes de vitesse de convergence et de qualité de la sortie.

3.2 Suivi de points d'intérêts sur un visage

L'entrée de notre pipeline consiste en un suivi de points en 3D. Les points suivis sur le visage importent peu, de même que leur ordre. Il faut juste qu'ils soient cohérents sur une animation entière.

Ils peuvent être basés sur des points de maquillage sur le visage de l'acteur, ou sur les méthodes présentées dans le défi 3DFAW [Jeni et al. (2016)]. Dans le cadre de notre partenariat avec Ubisoft, nous utilisons le logiciel de suivi de points en 3D Performer (©Dynamixyx [Per]) , qui suit 97 points par apprentissage de façon robuste sur un visage. Un exemple de ces 97 points est présenté dans la figure 3.2

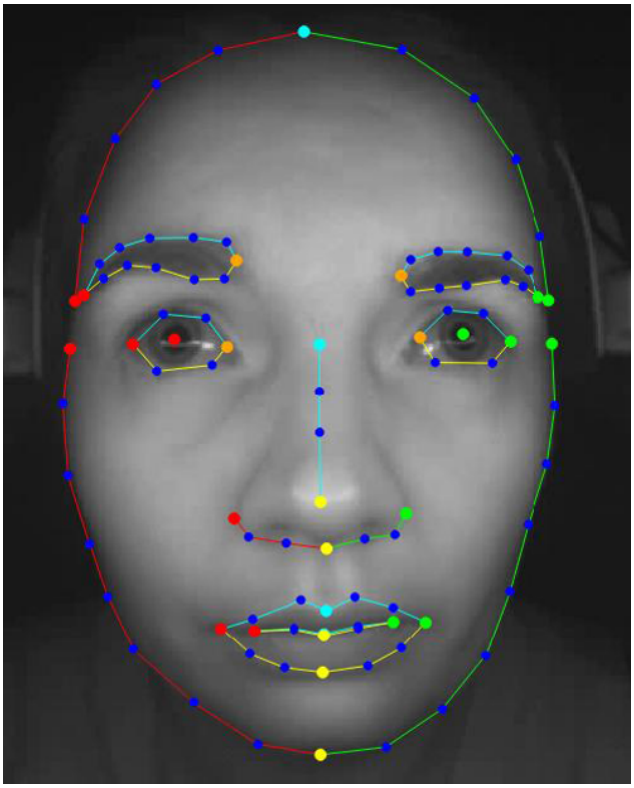

Figure 3.2 Les 97 points suivis par le logiciel Performer, l'entrée de notre pipeline

On définira les «zones» du suivi de points comme les points portant sur une même zone d'intérêt du visage. Par exemple, sur la figure 3.2, on a neuf zones distinctes, huit représentées par les points interconnectés entre eux, ainsi que la zone du nez, qui concatène les points connectés sous les narines et sur la crête du nez.
Dans le modèle utilisé avec notre partenaire, on a donc une zone de mâchoire, une zone pour chaque sourcil, une zone pour chaque œil, une pour le nez, pour l'intérieur des lèvres, l'extérieur des lèvres et enfin, une zone pour le front.

On remarque néanmoins qu'aucun suivi n'est effectué au niveau des joues ou des rides du front, ce qui entrainera nécessairement une perte de qualité une fois ces données entrées dans notre pipeline.

3.2.1 Associer les points suivis au Rig

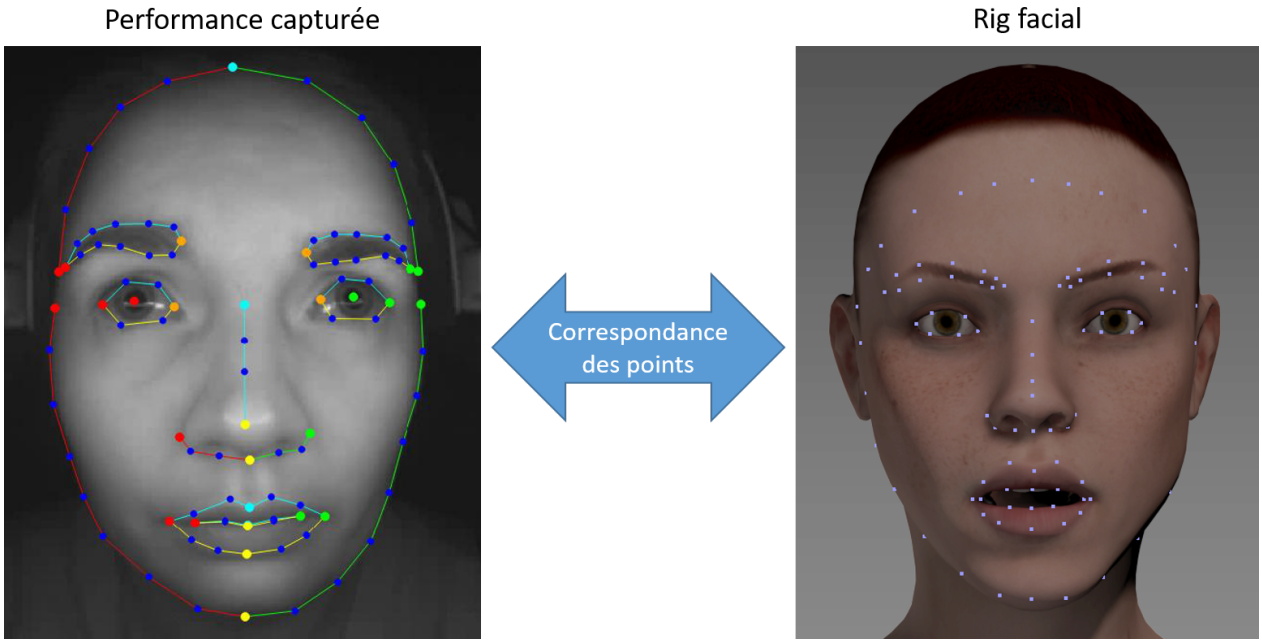

Figure 3.3 Correspondance entre les 97 points suivis sur l'acteur et sur le Rig d'Elise. Cette Correspondance se fait dans une étape de prétraitement une fois par Rig.

Une fois le suivi de points récupéré, il convient de le lier à un Rig. Pour cela, nous avons développé un outil sur 3DSMax permettant de lier ces points.

Il convient de choisir à la main ces mêmes points suivis sur le Rig, afin d'avoir une correspondance entre le suivi et le Rig.

Tout d'abord, l'utilisateur créée les zones qu'il souhaite mettre en place. Par exemple, dans l'exemple du suivi de points issus du logiciel Performer que l'on voit en figure 3.3, les zones sont : Sourcil Gauche, Sourcil Droit, Œil Gauche, Œil Droit, Front, Menton, Nez, Bouche Intérieure et Bouche Extérieure.

Ensuite, par zones, l'utilisateur associe les points suivis de la performance aux vertex du modèle 3D en sélectionnant dans l'ordre des points de la zone les vertex du modèle 3D un par un. L'ordre des points est régi par une règle simple : les points sont sélectionnés dans l'ordre horaire si la zone est cyclique, comme les Yeux, la Bouche ou les Sourcils dans notre

exemple, et de haut en bas et de gauche à droite si non, comme le Front, le Menton ou le Nez sur la figure 3.3.

Cette étape de prétraitement hors ligne prend environ une demi-heure à effectuer, mais n'a ensuite plus jamais à être réalisée pour un rig donné.

3.3 Créer une cible atteignable à partir du suivi de points

La première étape de notre pipeline de transfert d'expressivité sera de reproduire notre suivi de points, c'est-à-dire notre cible que l'on cherche à atteindre, dans l'espace 3D du rig sur lequel on cherche à reproduire l'animation.

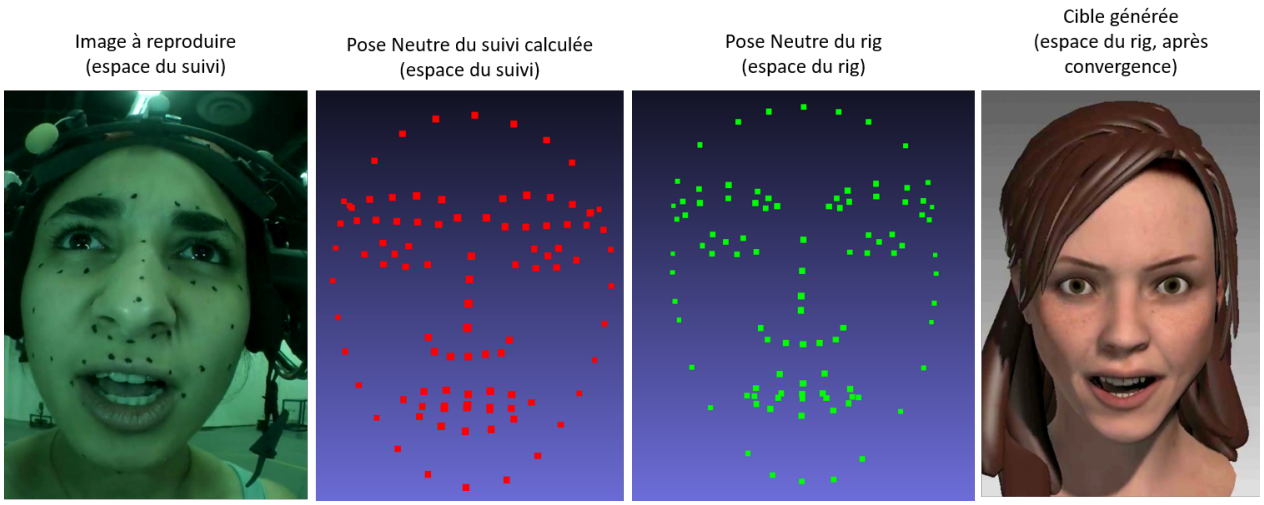

Figure 3.4 Transférer l'expressivité, c'est passer de l'image de gauche à l'image de droite

On utilise la formule suivante :

$$Cible_{Rig} = (Cible_{Suivi} - Neutre_{Suivi}) \cdot T + Neutre_{Rig} \tag{3.16}$$

Avec T la transformation optimale entre $Neutre_{Rig}$ et $Neutre_{Suivi}$.

Bien que l'on connaisse $Cible_{Suivi}$ et $Neutre_{Rig}$, il nous faudra déterminer les 2 autres éléments afin de pouvoir utiliser cette formule.

Ce que l'on souhaite, c'est d'utiliser ce pipeline sur toutes les images de notre animation. Une animation sera une suite de cibles à atteindre qui s'enchaine dans le temps. Notons t le temps actuel et Z le nombre d'images total dans l'animation. On peut donc réécrire 3.16

$\forall t \in \{1, \cdots, Z\}$ comme :

$$Cible_{Rig}[t] = (Cible_{Suivi}[t] - Neutre_{Suivi}) * T + Neutre_{Rig} \tag{3.17}$$

3.3.1 Génération de la pose neutre et de l'alignement automatiquement

La création de la cible se fait en transmettant la différence entre une image donnée et la pose neutre du suivi de points sur la pose neutre du rig. Afin d'avoir une cible dont l'expressivité correspond vraiment à ce que l'on a capturé, il faudra donc tout d'abord s'assurer que les poses neutres sont cohérentes entre elles.

Figure 3.5 Poses neutres sur les Rigs des productions «Assassin's Creed Unity» (©Ubisoft 2014-2017), «Watch Dogs 2» (©Ubisoft 2016-2017) et «The Division» (©Ubisoft 2016-2017)

.

Sur la figure 3.5, on remarque bien le problème de la cohérence de la pose neutre. À gauche, on a la pose neutre du rig sur la Production «Assassin's Creed Unity» (©Ubisoft 2014-2017) présentée ici sur le personnage d'Elise. Au centre, la pose neutre du rig sur la production «Watch Dogs 2» (©Ubisoft 2016-2017) présentée sur Marcus. À droite, la pose neutre du rig de «The Division» (©Ubisoft 2016-2017).

Ces trois rigs différents ont un nombre d'«os» d'animations différent et des espaces d'animations différents, se traduisant par un nombre différent de glissoirs.

On remarque bien que la pose neutre peut varier drastiquement d'une production à l'autre. On veut générer une pose neutre de suivi de points qui correspond bel et bien à la pose Neutre du rig. Bien que la plupart des productions choisissent de mettre la pose neutre avec

la bouche fermée, ce n'est pas le cas de «Watch Dogs 2» (©Ubisoft 2016-2017), il faut donc prendre ce requis en compte lorsque l'on génère $Neutre_{Suivi}$ et T.

Pose neutre et alignement par moyennage global

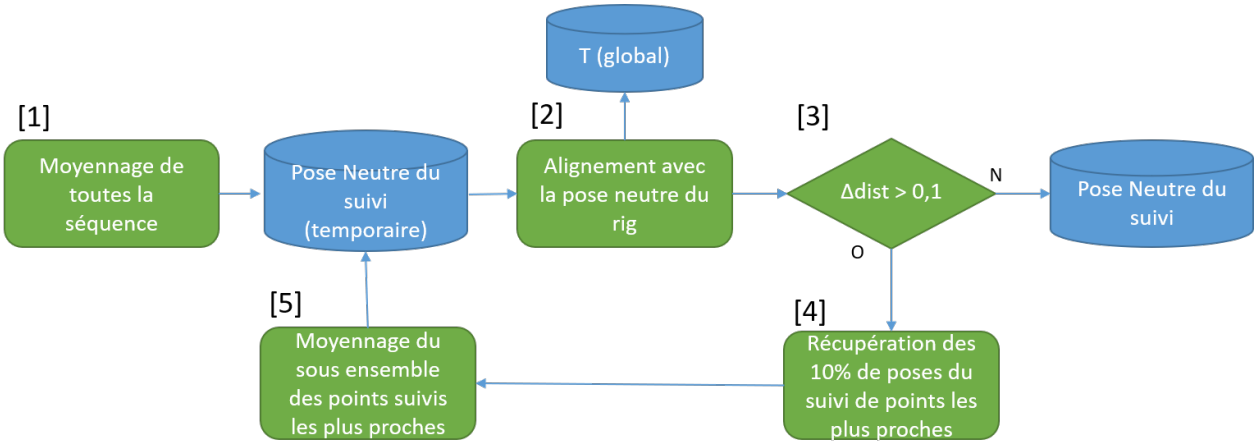

Figure 3.6 Pipeline de génération de la pose neutre et des transformations pour l'alignement Rig-Suivi. Cette version se base sur une suite de moyennage et calcule un alignement global

Naïvement, on commencera par considérer qu'un moyennage des positions sur notre suivi au fil du temps est une bonne estimation de notre pose neutre. On réalisera donc l'opération (1) sur le pipeline de la figure 3.6.

$$Neutre_{Suivi} = \sum_{t=1}^{Z} \frac{Cible_{Suivi}[t]}{Z} \tag{3.18}$$

On a donc désormais une première estimation de notre suivi neutre, il faudra donc tester la validité de cette estimation. Pour faire cela, on cherchera à aligner $Neutre_{Rig}$ et $Neutre_{Suivi}$.

Pour effectuer cet alignement, on réalise une analyse procustéenne simple. Dans notre cas, on cherche à minimiser la distance euclidienne post-alignement entre les poses neutres. On cherche alors :

$$\min_{T} \|Neutre_{Suivi} \cdot T - Neutre_{Rig}\|_2 \tag{3.19}$$

La solution de ce problème est connue. Si on se place en coordonnées homogènes pour plus de confort de calcul, T est donnée par la formule suivante, avec V représentant la pose neutre du suivi de points $Neutre_{Suivi}$ et L la pose neutre du rig $Neutre_{Rig}$:

$$T = (V^{\mathsf{T}}V^{-1})V^{\mathsf{T}}L \tag{3.20}$$

Le problème de l'équation 3.20, c'est que le facteur d'échelle n'y est pas constant dans toutes les dimensions : on peut avoir un facteur différent sur X, Y et Z. Il existe également une solution au problème de Procustes qui contraint l'orthogonalité des transformations (et donc un facteur d'échelle constant). Cette solution est connue depuis les années 60 et a été démontrée dans [Schönemann (1966)]

On cherche la matrice orthogonale R qui minimise notre distance post-transformation. Pour trouver la transformation optimale entre A et B, le problème est donc :

$$\min_{R} \|RA - B\| \quad | \quad R^T R = I \tag{3.21}$$

On réalisera tout d'abord une décomposition en valeurs singulières.

$$M = A^T B = U\Sigma V^T \tag{3.22}$$

Dans l'équation 3.22, U et V sont deux matrices de rotations en coordonnées homogènes et Σ une matrice de mise à l'échelle en coordonnées homogènes. La valeur de R se trouve simplement via cette décomposition.

$$R = UV^T \tag{3.23}$$

Maintenant que l'étape (2) du pipeline de la figure 3.6 est effectuée, on vérifie si notre pose neutre est valable. Si ce n'était que la première itération, on jugera qu'elle ne l'est pas et on continuera dans le pipeline, sinon, on jugera que le minimum est atteint quand la différence de distance $\Delta dist$ entre 2 itérations tombera en dessous d'un seuil, que l'on a fixé à 0,001. Dans les faits, après environ 4 itérations, $\Delta dist$ atteint l'epsilon machine.

On projette tous les points de nos suivis à chaque image dans l'espace du rig afin de calculer une distance euclidienne à chaque image. On sélectionne alors les 10% d'images les plus proches de notre rig en distance. On considèrera que cet ensemble d'images est alors une meilleure approximation de la pose neutre que l'ensemble total.

On réitère ensuite les étapes (2) (3) (4) et (5) du pipeline jusqu'à ce que $\Delta dist$ respecte le critère d'arrêt. On considère alors que ce moyennage et cet alignement sont les meilleurs pour représenter une pose neutre cohérente vis a vis de celle du Rig.

Pose neutre et alignement par moyennage par zones

Bien que le pipeline de la figure 3.6 soit intéressant, il ne prend pas en compte un détail important du transfert d'expressivité : le contexte du visage source et du visage cible peut être très différent. On peut par exemple tout à fait vouloir transférer les mouvements d'un autre visage sur celui d'Elise de la figure 3.5. Ou encore, on peut vouloir transmettre les mouvements de l'acteur sur un personnage fantaisiste comme un animal anthropomorphique, il faut alors que l'expressivité du visage reste, bien que les contextes soient totalement différents.

Pour résoudre ce problème, plutôt que d'aligner la totalité des points du suivi sur le suivi via une unique transformation, on alignera par zones (Sourcils, Yeux, Nez, Bouche, tour des lèvres, ...). Seule l'étape (2) du pipeline de la figure 3.6 change.

En notant [.] la sélection des points et transformations reliés à nos zones, on cherche donc :

$$\min_{T[z]} \|Neutre_{Suivi}[z] * T[z] - Neutre_{Rig}[z]\|_2 \tag{3.24}$$

On trouve donc un alignement optimal par zones qui respecte la différence de sémantique des visages.

Pose neutre et alignement par RBF

Bien que l'acteur reste bien souvent dans une pose neutre (lorsqu'un autre protagoniste lui parle ou lors de moments de silence). On ne peut pas être assuré que la pose neutre se trouve en moyennant la séquence de suivi de points et donc que la pose neutre sera correcte.

Reprenons l'exemple de Marcus présentée à la figure 3.5. Sa pose neutre n'est pas une pose «Naturelle» du visage humain, il y a donc très peu de chances qu'un moyennage arrive à la reproduire, même si ce moyennage a lieu sur les 10% d'images les plus proches.

L'approche appropriée est donc de récupérer les paramètres de mélange à l'aide d'un solveur RBF, en réalisant un mélange de gaussiennes sur le suivi d'entrée, chaque image représentant une gaussienne dans l'espace des points du suivi, et mixer les entrées pour générer la pose neutre.

Ce nouveau pipeline est présenté en figure 3.7. Comme pour les deux méthodes précédentes, on aura besoin d'une première estimation de la pose neutre du suivi ainsi que de l'alignement optimal par zones pour respecter la différence de cohérence des visages.

Les choses changent à partir de la deuxième étape : il faut créer notre représentation du suivi facial sous forme de gaussienne.

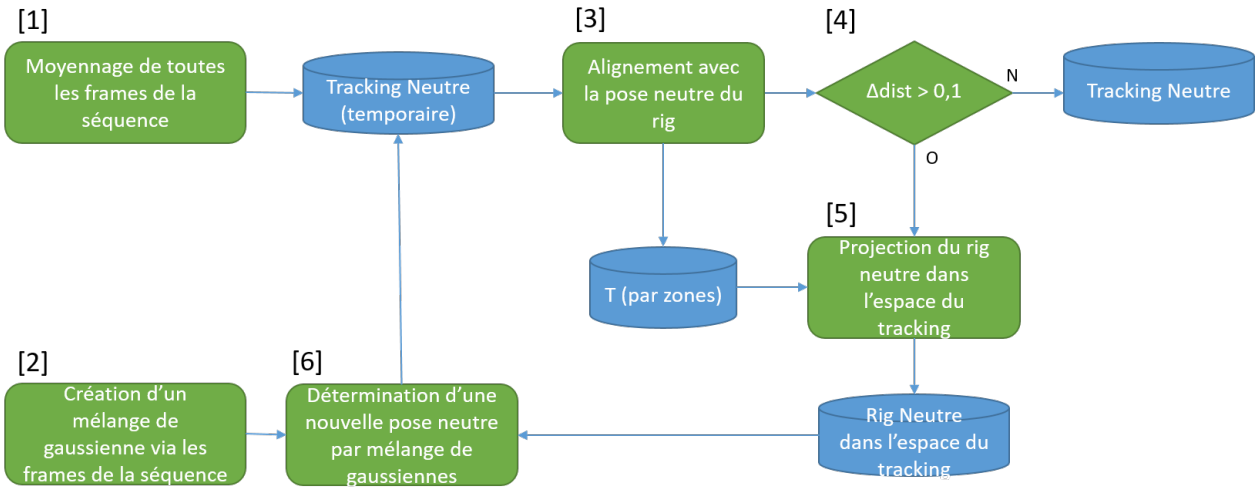

Figure 3.7 Pipeline de génération de la pose neutre et des transformations pour l'alignement Rig-Suivi basée sur un mélange de gaussienne RBF et un alignement global

$$\forall t \in \{1, \cdots, N\}\, G_t(X) = exp(-\left\|X - Cible_{Suivi}[t]\right\|_2) \tag{3.25}$$

Avec X un ensemble de points de suivi cohérent au suivi enregistré et dans l'espace du suivi. Pour trouver comment produire un $Neutre_{Suivi}$ convenable, il suffit de projeter $Neutre_{Rig}$ dans l'espace du suivi et de trouver son mélange M de gaussienne optimal.

$$M = \begin{bmatrix} G_1(Cible_{Rig} \cdot T^{-1}) \\ \cdots \\ G_N(Cible_{Rig} \cdot T^{-1}) \end{bmatrix} \tag{3.26}$$

N étant le nombre d'images total de notre animation. En normalisant cette valeur, on trouve le mélange des images de suivi optimal qui produit la pose neutre. Il suffit ensuite de faire une somme pondérée par le mélange de gaussienne du suivi pour produire cette pose neutre la plus proche du rig neutre.

$$Neutre_{Suivi} = \sum_{t=0}^{Z} \left[\frac{M[t]}{\sum_{i=0}^{Z} G_i(Cible_{Rig} \cdot T^{-1})} \cdot Cible_{Suivi}[t] \right] \tag{3.27}$$

Pose neutre et alignement par RBF par zones

Avec la méthode précédente, le problème de contexte des visages est toujours présent. On peut toutefois encore utiliser une approche par zones pour avoir un alignement optimal respectant la différence de sémantique des visages, renforcée de plus d'un mélange RBF par zones, donc d'autant plus robustes, les points des autres zones n'ayant plus aucune influence dans le mélange d'une zone donnée.

Il convient donc de modifier l'équation 3.26 afin de refléter l'existence de ces poids sur le mélange de Gaussiennes par zones.

$$M[z] = \begin{bmatrix} exp(-\left\|Cible_{Rig}[z] \cdot T^{-1}[z] - Cible_{Suivi}[z][1]\right\|_2) \\ \cdots \\ exp(-\left\|Cible_{Rig}[z] \cdot T^{-1}[z] - Cible_{Suivi}[z][N]\right\|_2) \end{bmatrix} \tag{3.28}$$

La pose neutre devient alors l'union des sommes pondérés des mélanges de gaussiennes associés à chaque zone. En posant $S[z] = \sum_{i=0}^{Z} M[z]$ on a :

$$\begin{cases} Neutre_{Suivi} = \emptyset \\ \underset{z}{\forall} \left(Neutre_{Suivi} \parallel \left[\frac{M[z][t]}{S[z]} \cdot Cible_{Suivi}[z][t] \right] \right) \end{cases} \tag{3.29}$$

3.3.2 Alignement «par image»

Un autre problème d'alignement peut avoir lieu lors de la tentative de résolution des glissoirs. En effet, bien que notre alignement soit correct pour la pose neutre, il ne l'est pas forcément pour toutes les autres poses. Certaines cibles ne sont donc pas forcément atteignables.

C'est pourquoi toute les 10 itérations de notre optimisation par descente de gradient, présentée dans la section 3.1, on réalise un réalignement par image.

À chaque image se rajoutera, en plus de la transformation globale générant la cible, une transformation locale pour rendre la cible atteignable par notre rig.

On cherche donc à modifier l'équation 3.17 pour prendre en compte cette nouvelle déformation. L'équation devient donc, en reprenant la notation par zone, à :

$$\begin{cases} Cible_{Rig}[t] = \emptyset \\ \underset{z}{\forall} \left(Cible_{Rig}[t] \parallel \left[\left(Cible_{Suivi}[z][t] - Neutre_{Suivi}[z] \right) \cdot T[z] + Neutre_{Rig}[z] \right] \cdot T_{local}[z][t] \right) \end{cases} \tag{3.30}$$

Comme pour l'alignement des poses neutres, une analyse procustéenne est réalisée pour déterminer $T_{local}[z][t]$ pour chaque zone z et chaque image t. Cet alignement est calculé entre la position actuelle des points du Rig V et la cible.

$$\min_{T_{local}[z][t]} \|Cible_{Suivi}[z][t] \cdot T_{local}[z][t] - V[z][t]\|_2 \tag{3.31}$$

En plus de rendre la cible plus atteignable, cette opération de réalignement par image réalisera un effet de réchaud sur notre optimisation, nous permettant d'obtenir une distance plus proche dans notre espace suite à la descente de gradient de la section 3.1

3.4 Correction de la sortie par réseau RBF

Bien que la sortie du pipeline soit acceptable, elle n'est pas parfaite. En effet, un artiste va parfois vouloir retravailler l'émotion ou l'expressivité de certaines poses qu'il juge incorrectes ou encore pas assez expressives. C'est pourquoi on pourra corriger la sortie à l'aide de la méthode classique d'animation utilisée, en introduisant notre notion d'«Images-clés de correction».

À une image donnée, si l'artiste décide de réaliser une correction, le suivi de point associé à l'image va être projeté dans l'espace RBF du suivi de points constitué de toutes les images. Si la distance RBF à une image est supérieure à un seuil donné, on considérera que cette image doit également être corrigée. On utilisera alors le mélange RBF des glissoirs utilisé dans le pipeline classique à ces images, et le pipeline procédural à toutes les autres.

Pour éviter que le résultat soit discontinu, on réalisera une convolution de ce mélange par une gaussienne d'une largeur de 10 images.

CHAPITRE 4 RÉSULTATS

Nous présentons nos résultats sur principalement trois performances différentes, «Comprendre Antigone», un extrait d'une tirade dans Antigone, de Jean Anouilh, «Femme Moderne», extrait d'une tirade de Cendrillon de Joëlle Pommerat et «Laideur et Amour».

Ces trois performances théâtrales ont été effectuées par Pauline Marion.

Ces résultats ont été produits sur une machine équipée d'un processeur Intel Xeon E5-1650 à 6 cœurs cadencé à 3.20 GHz, de 32 Go de mémoire vive, d'une carte graphique NVidia Geforce GTX 670 possédant 1344 cœurs CUDA, 4 Go de mémoire vidéo et cadencé à 967 MHz. Cette machine a pour système d'exploitation Windows 7 Entreprise version 6.1.7601 SP1 Build 7601.

Figure 4.1 Rigs utilisés pour nos tests, issus des productions «Assassin's Creed Unity» (©Ubisoft 2014-2017), «Watch Dogs 2» (©Ubisoft 2016-2017) et «The Division» (©Ubisoft 2016-2017)

.

4.1 Sortie brute du pipeline

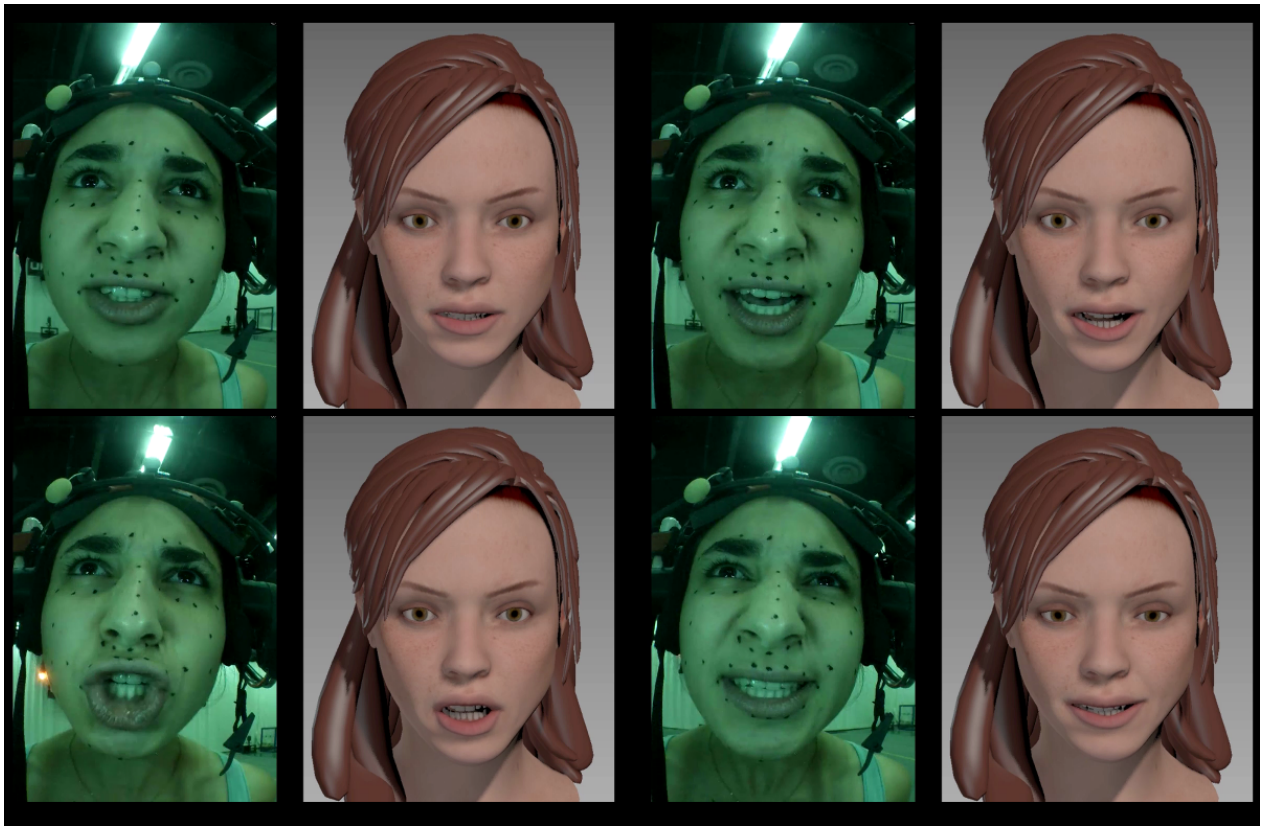

Figure 4.2 Extraits des résultats du pipeline présentés sur le Rig de Elise pour la performance «Comprendre Antigone».

Les résultats sur la performance présentée dans la figure 4.2 ont été obtenus en 80 secondes, pour une séquence d'une durée de 38 secondes. Dans cet extrait, Antigone est énervée et se rebelle contre l'ordre établi. Le but était donc de pouvoir transcrire cette colère et ce dégout sur le rig via notre transfert d'expressivité.

Bien que l'expression générale du visage et la forme de la bouche soient acceptables, même si imparfaite, on remarquera que l'on manque encore d'expressivité avant d'atteindre le résultat escompté. Tout particulièrement, le bas de l'œil manque de mouvement après le transfert de déformation, faisant ainsi perdre de l'effet «colérique» de la performance.

Ces problèmes s'expliquent par une disposition des points suivis peu avantageuse : en effet, des points mal répartis sur le visage ne permettront pas à notre méthode de reproduire toutes les expressions. On ne pourra par exemple pas reproduire les mouvements des joues, puisque celles-ci ne sont pas présentes dans le suivi de points.

Ces problèmes se corrigent néanmoins très bien via l'utilisation des images-clés de corrections, tel que nous pourrons le voir dans la prochaine section.

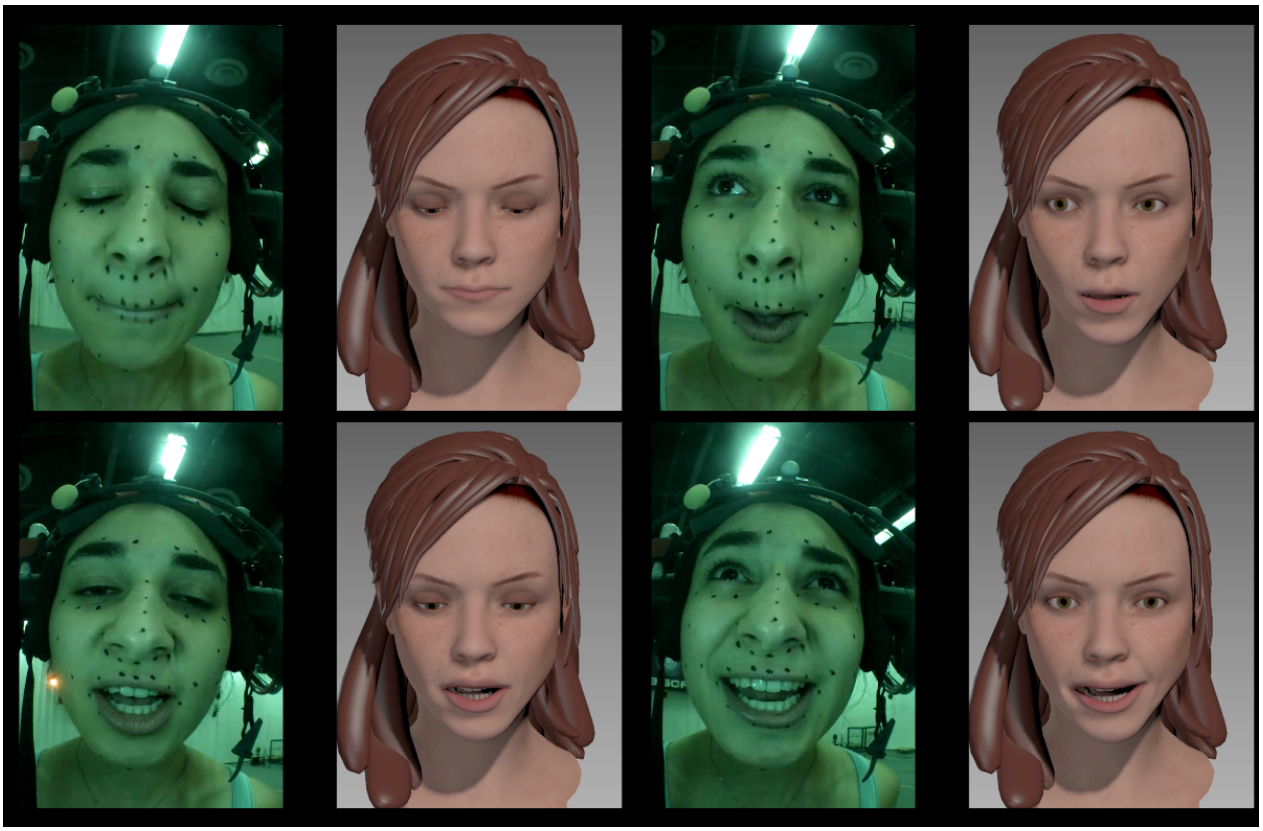

Figure 4.3 Extraits des résultats du pipeline présentés sur le Rig de Elise pour la performance «Femme Moderne».

Le résultat présenté dans la figure 4.3 a été obtenu en 182 secondes, pour une séquence d'une durée de 87 secondes.

Dans «Femme Moderne», la performance est celle d'un échange. Le personnage y a une approche de maître à son auditoire, avec un sentiment presque maternaliste et mélancolique.

C'est cette mélancolie qui était le point important à transcrire. Contrairement à «Comprendre», l'expressivité y est mieux transmise. Néanmoins, le problème du manque de mouvement sur le bas de l'œil est toujours présent, de même qu'au niveau de la mâchoire

En effet, un problème connu de la méthode est que la mâchoire n'est pas assez activée. Lors de la descente de gradient, le mouvement des lèvres propose souvent un gradient plus important que le mouvement du menton pour la minimisation de l'énergie. Cet effet peut s'expliquer par le fait que l'on ne peut pas suivre de points sur les dents directement, les mouvements des

points sur le menton sont souvent sous-estimés. Un faible nombre de points suivis au niveau du menton ou un suivi de points peu robuste dans cette zone entrainera donc un résultat moins convaincant.

Figure 4.4 Extraits des résultats du pipeline présentés sur le Rig de Elise pour la performance «Laideur et Amour».

Ce problème de mâchoire est d'autant plus présent dans la performance «Laideur et Amour», présenté dans la figure 4.4, obtenu en 152 secondes, pour une séquence d'une durée de 50 secondes.

On remarque que dans la première partie de la vidéo, le sourire est haut et large. Ces poses extrêmes ne sont pas atteignables par le rig, malgré l'adaptation des poses neutres. Le pipeline sort donc l'ensemble de glissoirs qui approxime au mieux les informations qu'il possède.

On remarquera également que le problème ne survient plus sur les trois dernières images,

issues de la deuxième partie de la performance, où les mouvements du visage sont moins accentués.

Bien que des glissoirs existent pour activer l'animation des joues, ils ne peuvent pas être pris en compte par notre approche en utilisant le suivi de points de Performer, puisqu'aucun point n'est suivi au niveau des joues.

Figure 4.5 Extraits des résultats des trois performances sur le Rig de Marcus de «Watch Dogs 2» (©Ubisoft 2016-2017).

La figure 4.5 présente les résultats sur un personnage au physique très différent du visage de notre actrice. Marcus n'est en effet ni du même sexe ni de la même ethnie que notre actrice.

Ces résultats ont été obtenus en respectivement 159, 305 et 213 secondes pour reproduire les animations entières, dont ces images sont extraites.

Les résultats en transfert d'expressivité sont pourtant nettement meilleurs. On remarquera par exemple que les poses extrêmes des sourires de «Laideur et Amour» sont désormais correctement reproduites pour les poses atteignables. Cela peut s'expliquer par le fait que le rig de «Watch Dogs 2» (©Ubisoft 2016-2017) possède un espace d'animation bien plus large que celui d'Elise. De nombreuses poses qu'Elise ne pouvait pas produire sont donc possibles pour Marcus.

L'animation au niveau des joues n'est toujours pas satisfaisante pour les mêmes problèmes que sur le rig d'Elise : le manque de point suivis au niveau des joues ne permet jamais d'activer ces glissoirs là lors de la descente de gradient.

On remarquera également que le problème d'expressivité au niveau des yeux est plus faible. Néanmoins, il est toujours présent au niveau de la mâchoire. Dans les vidéos annexes de présentation de ce mémoire, il est apparent que le système préfère bouger les lèvres pour minimiser l'énergie plutôt que de bouger le menton, ce qui est logique vis-à-vis de nos données de suivis.

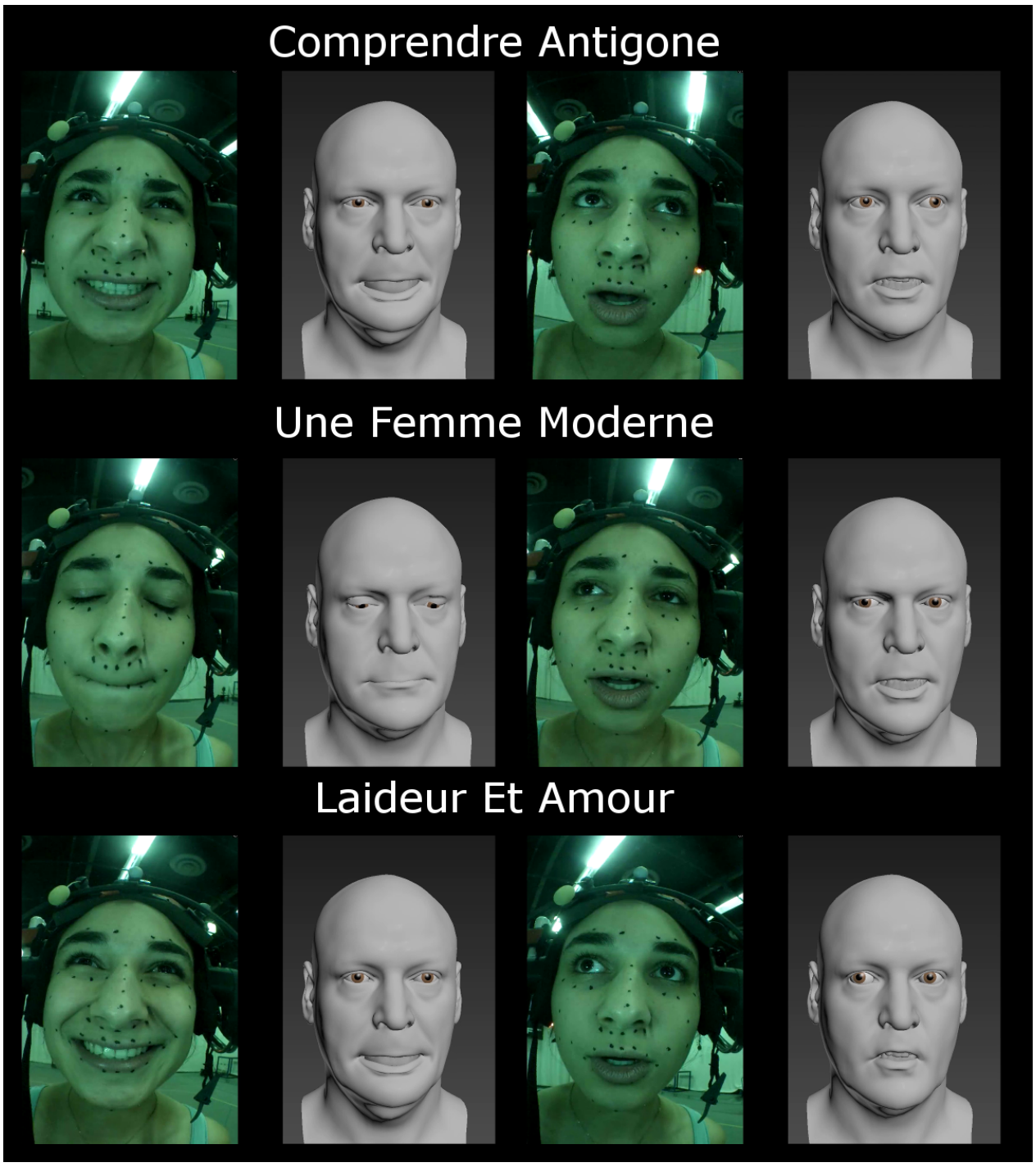

Figure 4.6 Extrait des résultats des trois performances sur le Rig de «The Division» (©Ubisoft 2016-2017).

Sur le rig de «The Division» présenté en figure 4.6, le problème de la mâchoire est encore plus apparent. En effet, il n'y a pas de glissoirs d'animations entrainant un mouvement de l'un des 97 points suivis pour un mouvement de la mâchoire interne. Le glissoir n'est donc jamais activé par notre pipeline, résultant en un mouvement des lèvres correct, mais aucun mouvement des dents.

On peut donc affirmer que le choix du rig influence grandement la sortie de notre pipeline. Les problèmes survenus sur nos trois rigs de tests ont été à chaque fois liés à la construction du rig lui-même.

Dans le cas d'Elise, le rig ne permettait pas d'effectuer physiquement la pose «sourire», donnant un résultat peu satisfaisant sur la performance «Laideur et Amour». Aussi, le bas de l'œil ne s'activait pas, du fait de l'absence d'un glissoir d'animation directement lié au clignement de l'œil.

Dans le cas de Marcus, la trop grande dimensionnalité de l'espace d'animation liée au trop faible nombre de points suivis sur notre performance sous-contraint notre problème, entrainant plus de mouvements au niveau des lèvres qu'au niveau du menton pour reproduire la parole. Les résultats sur le rig de «The Division» souffrent encore plus de ce problème, puisque la mâchoire n'est même jamais activée du fait de l'absence de suivi au niveau des dents.

Les différents problèmes qui sont survenus sur le rig d'Elise sont donc indépendants de notre méthode et dépendent plus de la façon dont le rig est construit et des données de suivi passées au pipeline. Si le rig est mal construit ou les données de suivis mauvaises, le reciblage sera également mauvais.

Malgré ces différents problèmes, le pipeline a été nettement plus rapide à donner un résultat de qualité qu'un animateur entrainé. Il faut en moyenne une dizaine d'heures pour faire l'animation complète d'un visage, contre quelques minutes dans notre cas.

4.2 Ajouts d'images-clés de corrections

On a pu voir dans la section précédente que malgré des résultats acceptables, la méthode ne fonctionne pas bien dans les cas extrêmes (grands sourires, clignements d'œil rapides, ...) et que le résultat manque globalement d'expressivité.

C'est pourquoi nous avions proposé aux artistes un pipeline permettant de corriger des images-clés à leur guise, dont le fonctionnement du côté utilisateur est en tout point identique au fonctionnement du pipeline précédent.

Nous présenterons donc ici un comparatif entre la méthode brute, dont le temps d'exécution est de quelques minutes, avec la méthode après une vingtaine de corrections.

Ces corrections ont nécessité moins d'une demi-heure à être réalisées, ce qui est toujours nettement inférieur à la dizaine d'heures nécessaire pour faire une animation de qualité auparavant.

Dans la figure 4.7, on remarque que le problème du sourire est désormais corrigé. Un faible nombre d'images-clés dans cette partie de la vidéo a suffi à corriger la totalité de la séquence présentant des difficultés majeures. Il convient de noter que les images choisies dans la figure ne se trouvent pas sur une image-clé et ont donc été automatiquement corrigées.

On remarquera également que les clignements d'yeux, qui posaient problème dans la sortie brute de notre pipeline sont également automatiquement corrigés. Ces clignements d'yeux ne sont pas instantanés, puisque les corrections induites par les animateurs sont convolués dans le temps par une gaussienne de dix images de large pour que le résultat soit continu.

Cette augmentation de l'expressivité est également flagrante sur les figures 4.8 et 4.9. On remarquera tout particulièrement le bas de l'œil et les joues, qui présentent désormais une expression bien plus ancrée.

Ces corrections ont également permis de rendre le rendu de la parole plus naturel en pouvant animer la langue, non-animée lors de l'exécution du pipeline brut, et en pouvant forcer certains visèmes sur des sons importants, comme les consonnes occlusives bilabiales $/b, m, p/$, ou encore les voyelles $/a, o, i, u, y/$.

Par exemple, sur la deuxième ligne de la figure 4.7 ainsi que sur la deuxième ligne de la figure 4.9, un visème $/m/$ a été forcé.

Ces corrections sont également utiles pour forcer la phonologie de certaines voyelles, comme c'est le cas sur la première ligne de la figure 4.8 où le visème $/o/$ a été forcé pour produire la bonne expression liée au phonème français «on».

Figure 4.7 Comparatif entre les résultats bruts et avec une vingtaine de corrections pour la performance «Laideur et Amour».

Figure 4.8 Comparatif entre les résultats bruts et avec une vingtaine de corrections pour la performance «Comprendre Antigone».

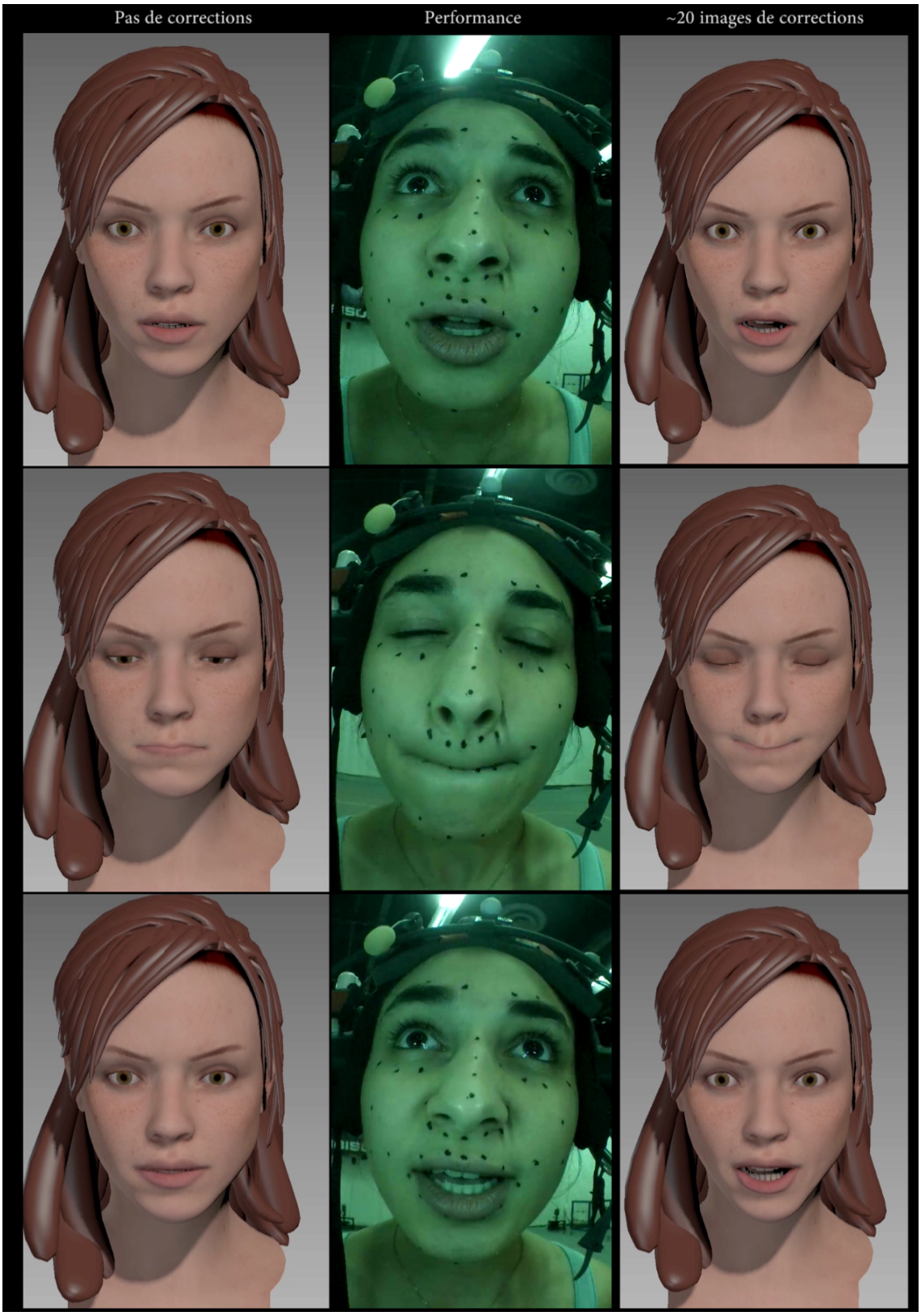

Figure 4.9 Comparatif entre les résultats bruts et avec une vingtaine de corrections pour la performance «Femme Moderne».

Avec ces corrections, nous proposons donc un outil combinant les bons points de la nouvelle méthode et de l'ancienne, à savoir le coté procédural de la nouvelle méthode, permettant d'avoir un résultat proche de la sortie voulue, bien que manquant un peu d'expressivité, en un temps très rapide, et le coté «images-clés» de l'ancienne méthode, proposant une interface que les artistes en animations ont l'habitude d'utiliser et leur permettant donc de retravailler la sortie afin de donner leur propre touche à la sortie.

L'objectif final de notre recherche, à savoir d'améliorer l'ancien pipeline d'animation faciale en termes de qualité et de productivité est donc atteint.

En effet, nous avons bien réussi à trouver les paramètres d'animations associés à un rig de visage correspondant à un déplacement d'un ensemble de points 3D sur un visage par cinématique inversée, tout en permettant à un artiste de retravailler la sortie comme il le souhaite, tel qu'était notre premier objectif. Pour faire cela, nous avons bien pu transmettre l'expressivité de la performance de l'acteur vers le rig pour avoir une cible à donner à notre méthode de reciblage, tel qu'était notre deuxième objectif.

De notre première hypothèse, en abstrayant les mouvements des joints du squelette par leurs composantes de translation uniquement, et en ignorant ainsi les composantes de rotations, nous avons pu dériver une méthode simple pour calculer le gradient des glissoirs d'animation, nous permettant ainsi de minimiser l'énergie des moindres carrés entre une performance cible et un modèle de visage en 3D à chaque image de la performance itérativement.

Notre seconde hypothèse, à savoir que la variété des visages humains possibles se trouve dans la déformation locale de points et donc qu'un alignement des visages par zones donnerait une meilleure cible, est également vérifiée. En effet, on a pu remarquer que cette nouvelle méthode de reciblage fonctionne sur des rigs 3D de sexes, âges et ethnies différentes.

On peut donc affirmer que nos hypothèses sont vérifiées.

CHAPITRE 5 DISCUSSION

5.1 Limitations de la solution proposée

La limitation principale de cette solution est l'impact de la qualité de l'entrée du pipeline.

En effet, si le suivi de point donné en entrée n'est pas correct, la sortie du pipeline sera également incorrecte. Les défauts du suivi de points, c'est-à-dire la mauvaise estimation de la matrice de caméra, la trop grande sphéricité de la lentille, etc. impactent grandement la sortie, puisque c'est sur ce suivi de points que se base la totalité de la méthode.

Un autre impact inhérent à la qualité de l'entrée est le nombre de points suivis. On remarque sur les différents résultats qu'il y a un certain manque d'expressivité. Ce manque pourrait être comblé par un suivi robuste de plus de points sur le visage. L'ajout d'images-clés de corrections corrige en partie ce problème, mais si de meilleurs résultats sont possibles en amont du pipeline, de meilleurs résultats surviendront également en aval.

Une autre limitation de cette méthode est qu'elle requiert que la déformation induite par le transfert d'expressivité, à savoir la cible à atteindre à chaque image, soit atteignable par le rig. On a pu remarquer dans les résultats de «Laideur et Amour» qu'un manquement à cette règle impacte fortement la qualité de la sortie. Bien que l'on puisse corriger cette sortie dans l'étape suivante du pipeline, mieux vaudrait atteindre la meilleure qualité possible plus tôt.

De ces deux limitations découle un défaut majeur que l'on constate sur les résultats présentés dans la section 4, le manque de mouvements au niveau de la mâchoire en sortie de notre pipeline. En effet, le logiciel de suivi de points utilisé a tendance à sous-estimer les mouvements des points suivis sur la mâchoire, d'une part par la sphéricité de la lentille utilisée pour capter la performance, et d'autre part par le caractère mono-caméra de la méthode de suivi, entraînant donc des erreurs dans l'estimation de la profondeur.

Dans cette section, nous allons discuter plus en détail de solutions mises en place pour arriver à contourner ces limitations.

5.1.1 Descente de gradient

Afin de nous assurer de ne pas nous retrouver coincés dans un minimum local, nous avons simulé un phénomène de recuit qui se traduisait dans notre méthode par le réalignement à certaines itérations de notre modèle avec notre cible. La figure 5.1 présente ces résultats. On pourra trouver les résultats issus des performances «Femme Moderne» et «Laideur et Amour» en annexe A.1 et A.2.

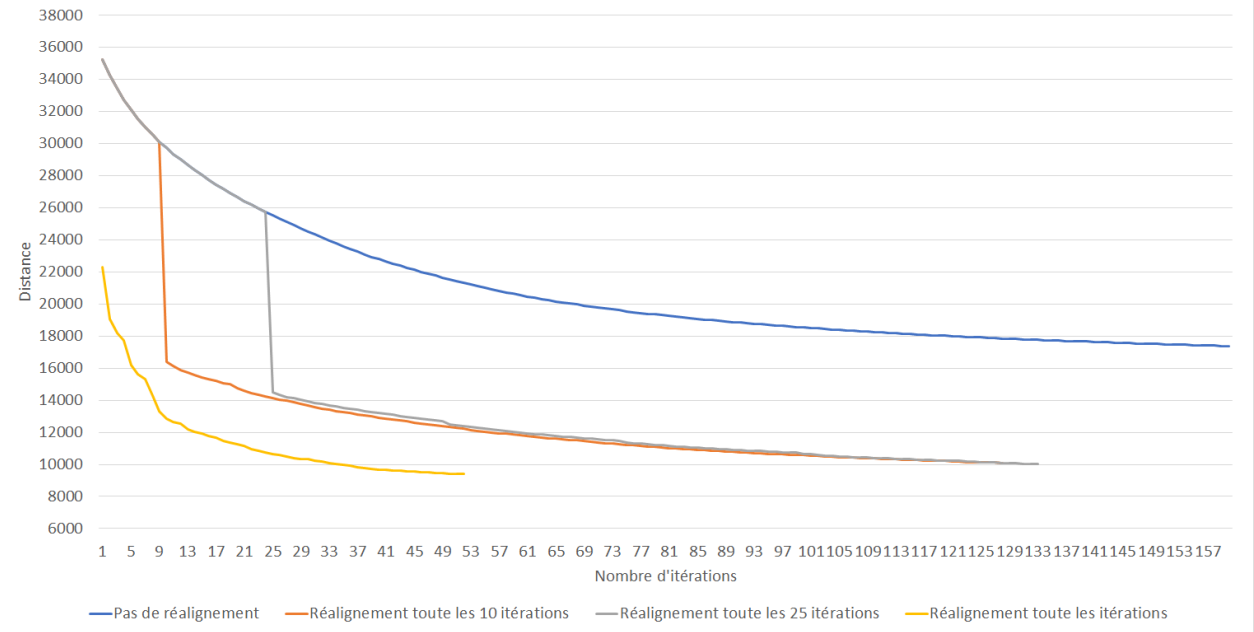

Figure 5.1 Influence du taux de réalignement sur la performance «Comprendre Antigone» sur le rig d'Elise. La pose neutre est générée par mélange RBF et transformations par zones.

On remarque que la solution la plus préférable semble être d'effectuer un réalignement à chaque itération de la descente de gradient.

Ce résultat peut s'expliquer simplement : certaines poses de l'espace d'animations ne correspondent pas à un mouvement de visage «réel». Le modèle du rig ne prend pas en compte l'élasticité de la peau. On a donc un grand nombre de poses potentielles qui ne sont pas atteignables par notre rig.

Le réalignement par itération nous permet donc de rendre ces cibles atteignables, et donc de continuer la descente.

Néanmoins, il convient de pouvoir faire varier ce facteur, car des expérimentations récentes nous ont indiqué que lorsque le nombre de points suivis est plus élevé et réparti de façon plus homogène sur le visage de l'acteur, alors un réalignement à chaque itération n'est pas

forcément plus efficace qu'un réalignement toutes les 5 ou 10 itérations. Le réalignement donne néanmoins un résultat toujours plus satisfaisant qu'une animation réalisée sans aucun réalignement pendant la descente de gradient.

Le facteur limitant est donc encore ici la qualité du suivi donné en entrée.

5.1.2 Génération de la pose Neutre

Comme exprimé à la section 3.3, la génération de la pose neutre est une étape ayant un des impacts les plus gros sur la sortie de notre pipeline. En effet, tout le transfert de déformation, et donc toute la génération de la cible que l'on va chercher à reproduire, dépend de cette étape.

Le but de cette étape était d'être en ligne et non prétraité, afin de s'adapter automatiquement aux données de suivi de points passées en entrée du pipeline.

Dans cette section, nous allons discuter de l'impact des trois méthodes de générations de la pose Neutre non «naïves» sur le résultat final.

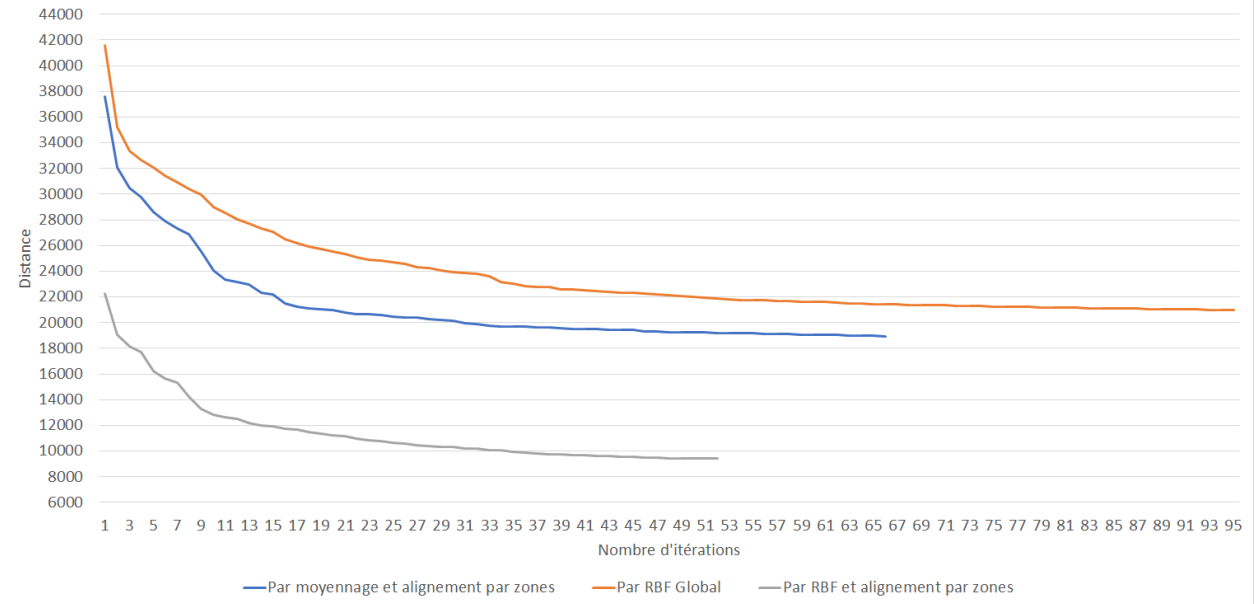

Figure 5.2 Influence de la génération de la pose neutre sur la performance «Comprendre Antigone» sur le rig d'Elise. Le facteur de réalignement est 1.

Sur la figure 5.2, on remarque que l'hypothèse selon laquelle la pose neutre influence énormément la sortie est vérifiée. On a en effet une énergie en sortie variant du simple au double, ce qui est non négligeable.

On pourra trouver les résultats issus des performances «Femme Moderne» et «Laideur et Amour» en annexe A.3 et A.4.

On remarquera que les méthodes basées sur des transformations par zones donnent de meilleurs résultats. Ce résultat n'est pas très surprenant, puisque les transformations par zones permettent de s'adapter au contexte du visage cible et donc d'être indépendant de l'identité de l'acteur de la performance.

La méthode donnant le meilleur résultat est le mélange par RBF local par zones. En effet, cette méthode va nous permettre d'approximer au mieux par sous-zones notre cible. Plutôt que de projeter la totalité des points du rig et de reconstruire un modèle à partir de cette projection dans l'espace RBF des points, on ne projette que la zone qui nous intéresse. On déconnecte ainsi les différentes zones. Un clignement d'œil ne va pas affecter un mouvement de lèvres, et ainsi de suite.

Figure 5.3 Les visages de notre actrice et d'Elise semblent bien semblables à première vue, mais des différences locales subsistent.

Sur la figure 5.4, Les points rouges correspondent à notre actrice, à gauche sur la figure 5.3, et les points verts à Elise, à droite sur la figure 5.3.

On remarque bien que dans le cas du mélange RBF par zones, la pose neutre du suivi de

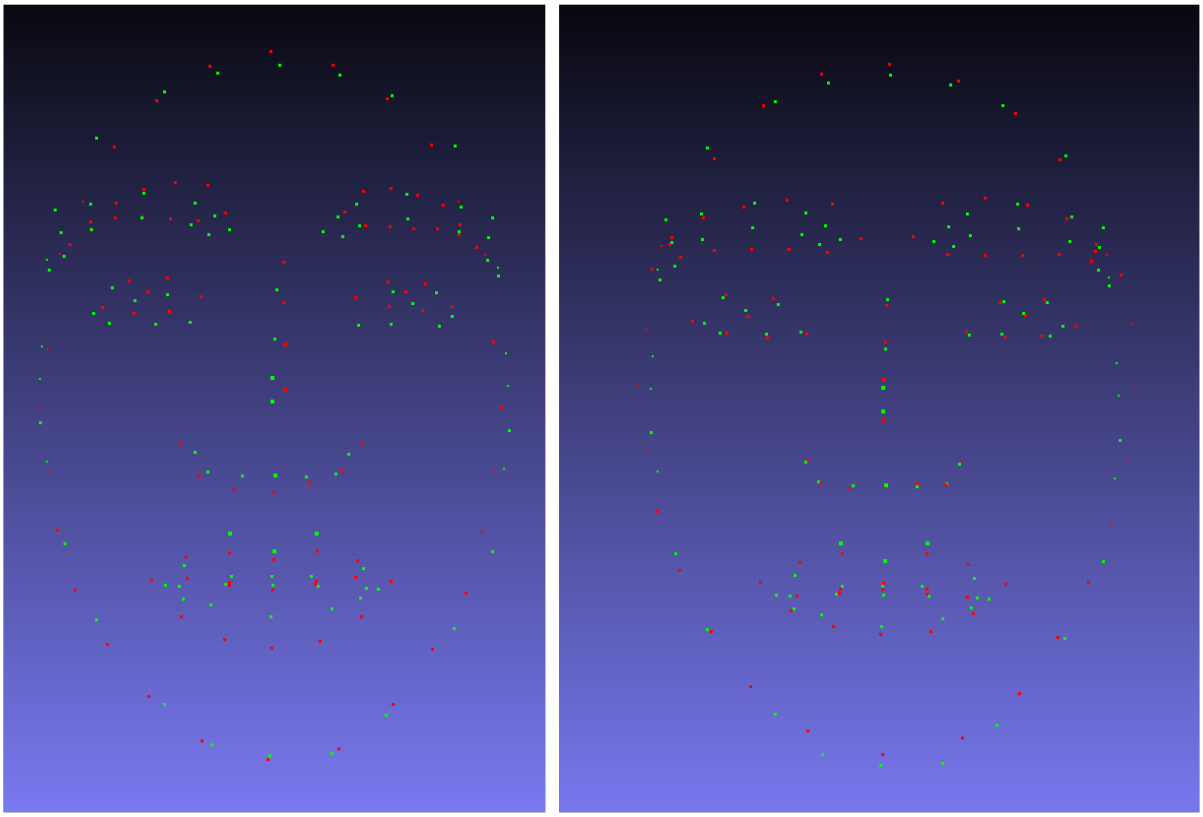

Figure 5.4 À gauche : réalignement global. À droite : réalignement par zones.

points générés automatiquement transformée respecte la morphologie de la cible. En effet, l'alignement global présenté sur la gauche de la figure montre des différences en termes d'échelle et de positionnement pour la plupart des zones du visage.

On remarquera tout particulièrement les points placés sur les lèvres. Dans le cas du réalignement global, le bas de la lèvre n'est pas du tout aligné entre la performance de l'acteur et la cible. Ce mauvais alignement va contribuer à un manque d'intensité dans l'animation au niveau des lèvres en sortie de notre pipeline.

Au contraire, dans le cas de l'alignement local par zones, l'alignement au niveau des lèvres est bien meilleur, entrainant un meilleur résultat en sortie du pipeline.

C'est ce résultat qui nous permet une animation plus satisfaisante lors de l'étape de descente de gradient.

5.1.3 Enrichissement du suivi par Flux Optique

Afin de contrer le problème du manque d'information sur notre suivi de points introduit dans la section 3.2, nous avons expérimenté de l'enrichir avec de nouveaux points suivis par flux optique Lucas-Kanade [Lucas et al. (1981)].

Tout d'abord, nous avons sélectionné des points de maquillage sur l'acteur qui correspondent à des points suivis dans Performer, le logiciel de suivi de point utilisé jusqu'à présent. Les points sélectionnés pour la correspondance avec Performer sont représentés en rouge sur la figure 5.5.

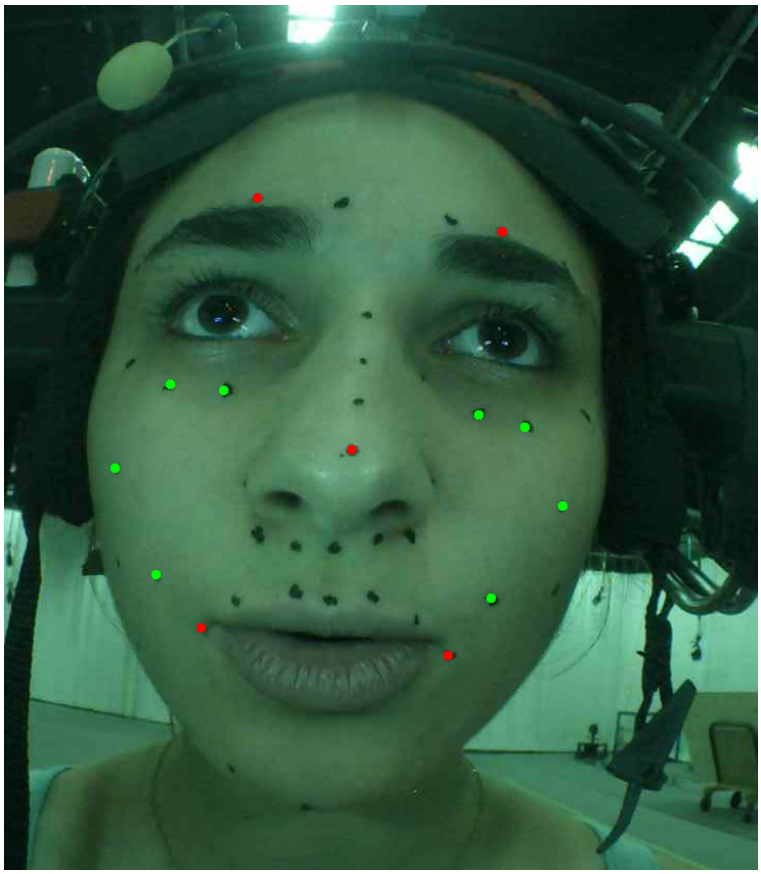

Figure 5.5 En rouge : les points utilisés pour stabiliser notre enrichissement avec le suivi de points pré existant. En vert : les points ajoutés au suivi par flux optique.

En plus de ces points, nous avons sélectionné les points que l'on souhaitait rajouter au suivi, à savoir ceux présents dans une zone qui manquait d'expressivité dans nos résultats : les joues et le bas de l'oeil. Un suivi de points par flux optique Lucas-Kanade [Lucas et al. (1981)] a alors été réalisé.

Afin de projeter les nouveaux points suivis dans l'espace des points suivis déjà présents, la

transformation 2D entre les points «rouges» de la figure 5.5 et les mêmes points issus du suivi déjà effectué est réalisée à chaque image par analyse Procustéenne.

En appliquant à chaque image la transformation calculée sur les nouveaux points, on les projette dans l'espace du suivi.

La composante de profondeur de ces nouveaux points est estimée par moyenne des profondeurs des points 3D du suivi à enrichir, pondérée par la distance 2D du point considéré à chacun de ces points.

Des premiers résultats avec cet enrichissement ont permis d'obtenir de légères améliorations au niveau de l'animation des joues, mais de nouvelles expérimentations récentes avec un nouveau suivi de points comprenant beaucoup plus de points répartis d'une façon bien plus homogène se sont avérés être encore plus prometteurs.

Ces expérimentations vont dans le sens de notre argumentation en faveur d'un suivi de meilleure qualité en entrée de notre pipeline pour obtenir les meilleurs résultats possible à l'utilisation de ce dernier.

5.2 Améliorations futures

Afin d'améliorer la qualité en sortie du pipeline, on se doit d'améliorer l'entrée. On cherchera donc à utiliser ou améliorer une méthode robuste de suivi de points en 3D sur un visage afin d'avoir plus de points de contrôles sur notre rig, ainsi qu'un suivi plus robuste. On pourra par exemple s'intéresser plus en détail aux résultats du challenge 3DFAW [Jeni et al. (2016)].

On cherchera également à réaliser notre optimisation non pas uniquement sur les points suivis, mais également sur la texture, comme cela a été utilisé dans plusieurs publications récentes telles que [Thies et al. (2015)] et [Garrido et al. (2016)].

Afin de résoudre le problème de cible difficile à atteindre, on cherchera à adapter la méthode à une approche multi-cible. Plutôt que de n'avoir qu'une unique pose neutre depuis laquelle la déformation sera transmise, on produira l'ensemble d'animation à plusieurs autres poses que l'on sait difficile, et on optimisera la distance pour trouver les glissoirs optimaux sur ces différents espaces. L'ensemble de glissoirs qui minimise la somme des distances sur tous les espaces sera alors utilisé.

Une autre amélioration future envisagée est de combler les points faibles de notre méthode par les points forts d'une autre. Une des limitations majeures de notre méthode est qu'elle a tendance à sous-estimer les mouvements de la mâchoire. Ces limitations proviennent de l'estimation du suivi de points, qui a des difficultés à estimer la profondeur des mouvements

des points suivis sur la mâchoire.

Des avancées récentes dans la génération d'animation de lèvres à partir d'une source sonore, telle que [Edwards et al. (2016)] nous encouragent dans cette démarche. L'animation des lèvres et de la mâchoire y est des plus impressionnantes et ne requiert pas de source vidéo à l'opposé de notre méthode. Mixer les deux sorties devrait permettre de corriger les plus gros défauts de notre sortie.

De premières expérimentations avec des systèmes simplistes d'animation labiale à partir de sources sonores aux résultats médiocres ont pourtant montré, une fois mixées avec la sortie de notre méthode, des résultats des plus prometteurs.

CHAPITRE 6 CONCLUSION

Nous avons présenté à travers ce mémoire une nouvelle méthode et un nouvel outil de transfert d'expressivité à l'attention des artistes en animation faciale afin de faciliter leur travail.

Cette méthode combine les attraits d'un système entièrement procédural, le résultat en première passe étant entièrement réalisé sans intervention humaine, et les attraits des systèmes basés sur des mélanges d'images-clés que les artistes sont habitués à utiliser.

Cette méthode s'avère robuste sous une multitude de rig faciaux humains différents, de sexes et ethnies différentes.

Cette méthode permet donc de simplifier et fluidifier grandement leur travail, tout en ne changeant pas la façon de travailler des animateurs 3D.

6.1 Synthèse des travaux

À partir d'un suivi de point robuste en 3D sur un visage d'acteur, la performance de cet acteur est reproduite au mieux sur le rig cible, dans l'espace des paramètres de ce rig.

Tout d'abord, la pose neutre du visage de l'acteur est déterminée automatiquement, ainsi que l'alignement optimal entre le visage de l'acteur et le rig cible. À la suite d'un transfert de déformation à chaque image de la performance de l'acteur vers le rig, une cible est produite sur le rig pour chaque image.

Par notre nouvelle méthode de reciblage dans l'espace des paramètres d'animations, les glissoirs d'animations optimaux sont déterminés par descente de gradient dans l'espace tensoriel des animations possibles pour chaque image de la performance.

L'artiste animateur peut ensuite à sa guise retravailler les images qu'il juge incorrectes ou pas assez expressives à l'aide des mêmes outils habituels d'animation faciale, à savoir l'utilisation d'images-clés. Ces corrections seront ensuite répercutées automatiquement sur les endroits nécessaires de l'animation par mélange RBF. Afin qu'une correction ait lieu, il faut que la distance des points 3D suivis à cette image vers les points 3D d'une des images-clés tombe sous un seuil.

RÉFÉRENCES

Dynamixyz. Consulté le 2017-05-05. En ligne : `http://www.dynamixyz.com/`

S. Bouaziz et M. Pauly, "Semi-supervised facial animation retargeting", EPFL, Rapp. tech. EPFL-REPORT-202143, 2014.

S. Bouaziz, Y. Wang, et M. Pauly, "Online modeling for realtime facial animation", *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 40, 2013.

A. Bulat et G. Tzimiropoulos, "Two-stage convolutional part heatmap regression for the 1st 3d face alignment in the wild (3dfaw) challenge", dans *European Conference on Computer Vision*. Springer, 2016, pp. 616–624.

C. Cao, Y. Weng, S. Lin, et K. Zhou, "3d shape regression for real-time facial animation", *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 41, 2013.

C. Cao, Q. Hou, et K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation", *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 43, 2014.

C. Cao, Y. Weng, S. Zhou, Y. Tong, et K. Zhou, "Facewarehouse : A 3d facial expression database for visual computing", *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 413–425, 2014.

L. Dutreve, A. Meyer, et S. Bouakaz, "Feature points based facial animation retargeting", dans *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*. ACM, 2008, pp. 197–200.

P. Edwards, C. Landreth, E. Fiume, et K. Singh, "Jali : an animator-centric viseme model for expressive lip synchronization", *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 127, 2016.

P. Ekman et W. V. Friesen, "Facial action coding system", Consulting Psychologists Press, Stanford University, Palo Alto, 1978.

P. Garrido, L. Valgaerts, C. Wu, et C. Theobalt, "Reconstructing detailed dynamic face geometry from monocular video", *ACM Trans. Graph.*, vol. 32, no. 6, p. 158, 2013.

P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, et C. Theobalt, "Reconstruction of personalized 3d face rigs from monocular video", *ACM Transactions on*

Graphics (TOG), vol. 35, no. 3, p. 28, 2016.

R. Gross, I. Matthews, J. Cohn, T. Kanade, et S. Baker, "Multi-pie", *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.

L. A. Jeni, S. Tulyakov, L. Yin, N. Sebe, et J. F. Cohn, "The first 3d face alignment in the wild (3dfaw) challenge", dans *European Conference on Computer Vision*. Springer, 2016, pp. 511–520.

N. Kholgade, I. Matthews, et Y. Sheikh, "Content retargeting using parameter-parallel facial layers", dans *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2011, pp. 195–204.

H. Li, T. Weise, et M. Pauly, "Example-based facial rigging", dans *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4. ACM, 2010, p. 32.

H. Li, J. Yu, Y. Ye, et C. Bregler, "Realtime facial animation with on-the-fly correctives", *ACM Trans. Graph.*, vol. 32, no. 4, pp. 42–1, 2013.

B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision", dans *Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, 1981, pp. 674–679.

K. Na et M. Jung, "Hierarchical retargetting of fine facial motions", dans *Computer Graphics Forum*, vol. 23, no. 3. Wiley Online Library, 2004, pp. 687–695.

J.-y. Noh et U. Neumann, "Expression cloning", dans *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 277–288.

P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem", *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos, et M. Pantic, "The first facial landmark tracking in-the-wild challenge : Benchmark and results", dans *Computer Vision Workshop (ICCVW), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1003–1011.

R. W. Sumner et J. Popović, "Deformation transfer for triangle meshes", *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 399–405, 2004.

J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, et C. Theobalt, “Real-time expression transfer for facial reenactment”, *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 183, 2015.

J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, et M. Niessner, “Face2face : Real-time face capture and reenactment of rgb videos”, dans *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, et C. Theobalt, “Lightweight binocular facial performance capture under uncontrolled lighting”, *ACM Trans. Graph.*, vol. 31, no. 6, p. 187, 2012.

D. Vlasic, M. Brand, H. Pfister, et J. Popović, “Face transfer with multilinear models”, dans *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 426–433.

T. Weise, H. Li, L. Van Gool, et M. Pauly, “Face/off : Live facial puppetry”, dans *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation*. ACM, 2009, pp. 7–16.

T. Weise, S. Bouaziz, H. Li, et M. Pauly, “Realtime performance-based facial animation”, dans *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 77.

S. Xiao, S. Yan, et A. A. Kassim, “Facial landmark detection via progressive initialization”, dans *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 33–40.

J. Yang, J. Deng, K. Zhang, et Q. Liu, “Facial shape tracking via spatio-temporal cascade shape regression”, dans *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 41–49.

X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, et J. M. Girard, “Bp4d-spontaneous : a high-resolution spontaneous 3d dynamic facial expression database”, *Image and Vision Computing*, vol. 32, no. 10, pp. 692–706, 2014.

R. Zhao, Y. Wang, C. F. Benitez-Quiroz, Y. Liu, et A. M. Martinez, “Fast and precise face alignment and 3d shape reconstruction from a single 2d image”, dans *European Conference on Computer Vision*. Springer, 2016, pp. 590–603.

ANNEXE A Autres courbes traitant de la minimisation de la distance

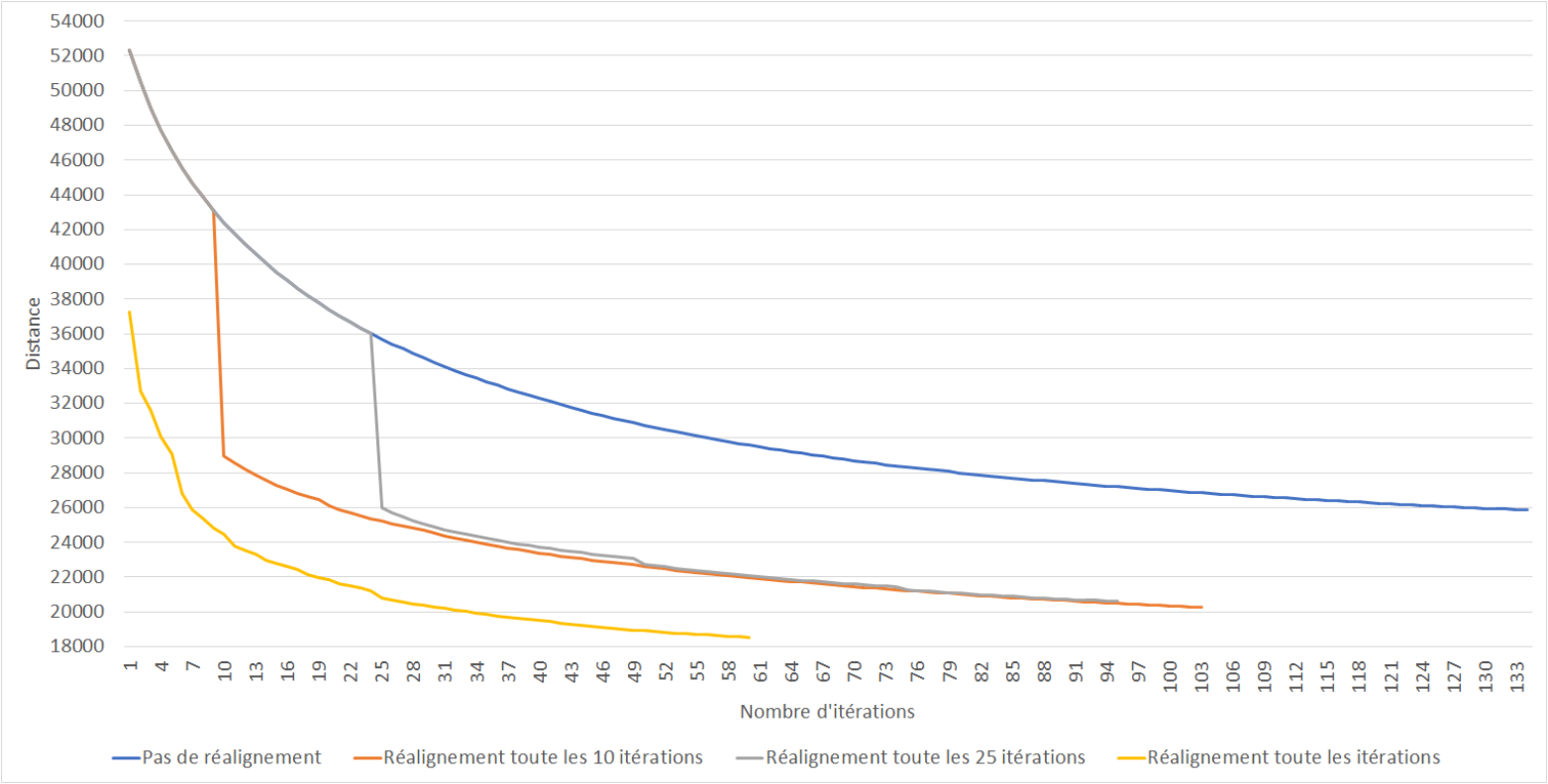

Figure A.1 Influence du taux de réalignement sur la performance «Femme Moderne» sur le rig d'Elise. La pose neutre est générée par mélange RBF et transformations par zones.

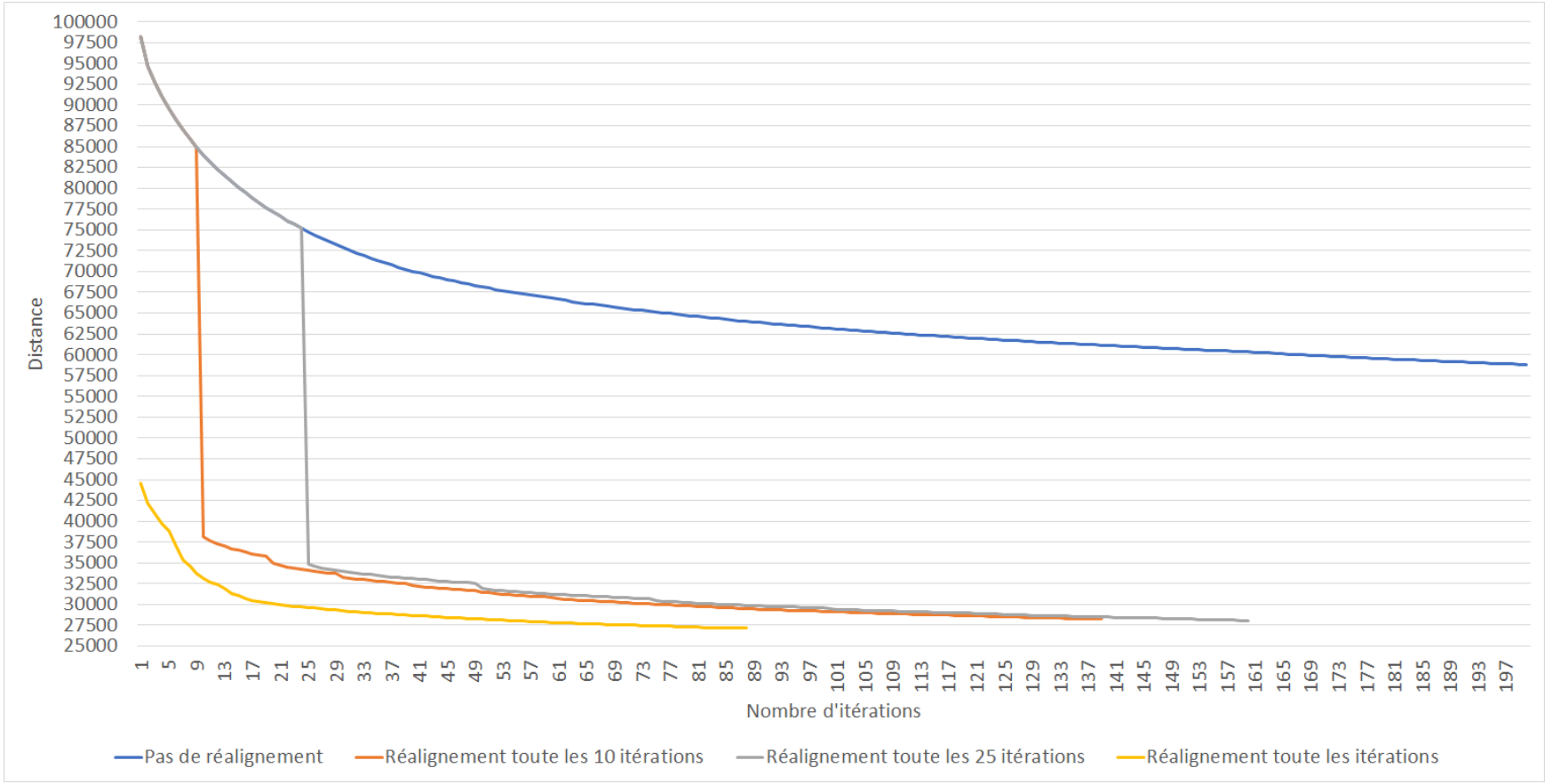

Figure A.2 Influence du taux de réalignement sur la performance «Laideur et Amour» sur le rig d'Elise. La pose neutre est générée par mélange RBF et transformations par zones.

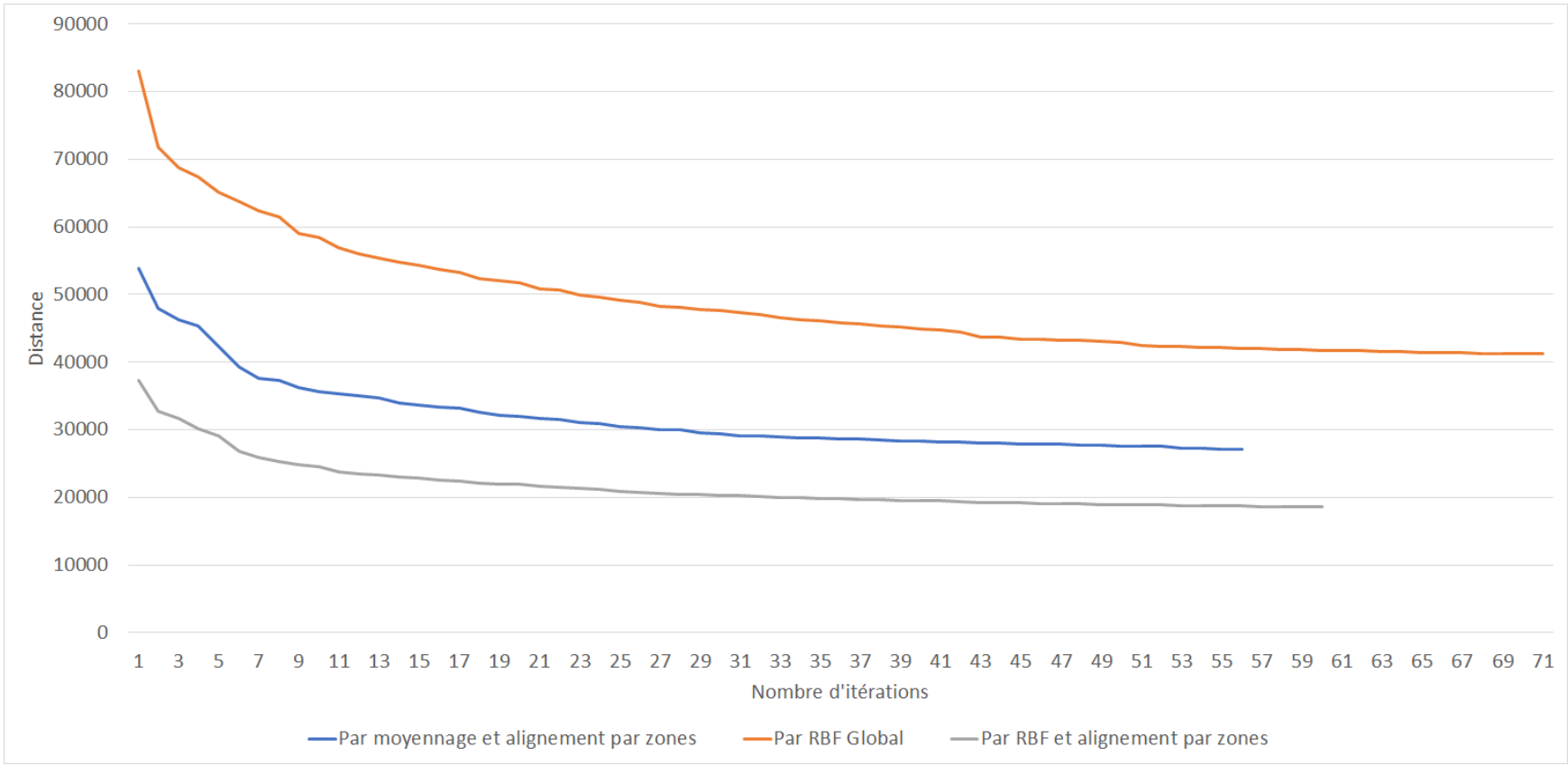

Figure A.3 Influence de la génération de la pose Neutre sur la performance «Femme Moderne» sur le rig d'Elise. Le facteur de réalignement est 1.

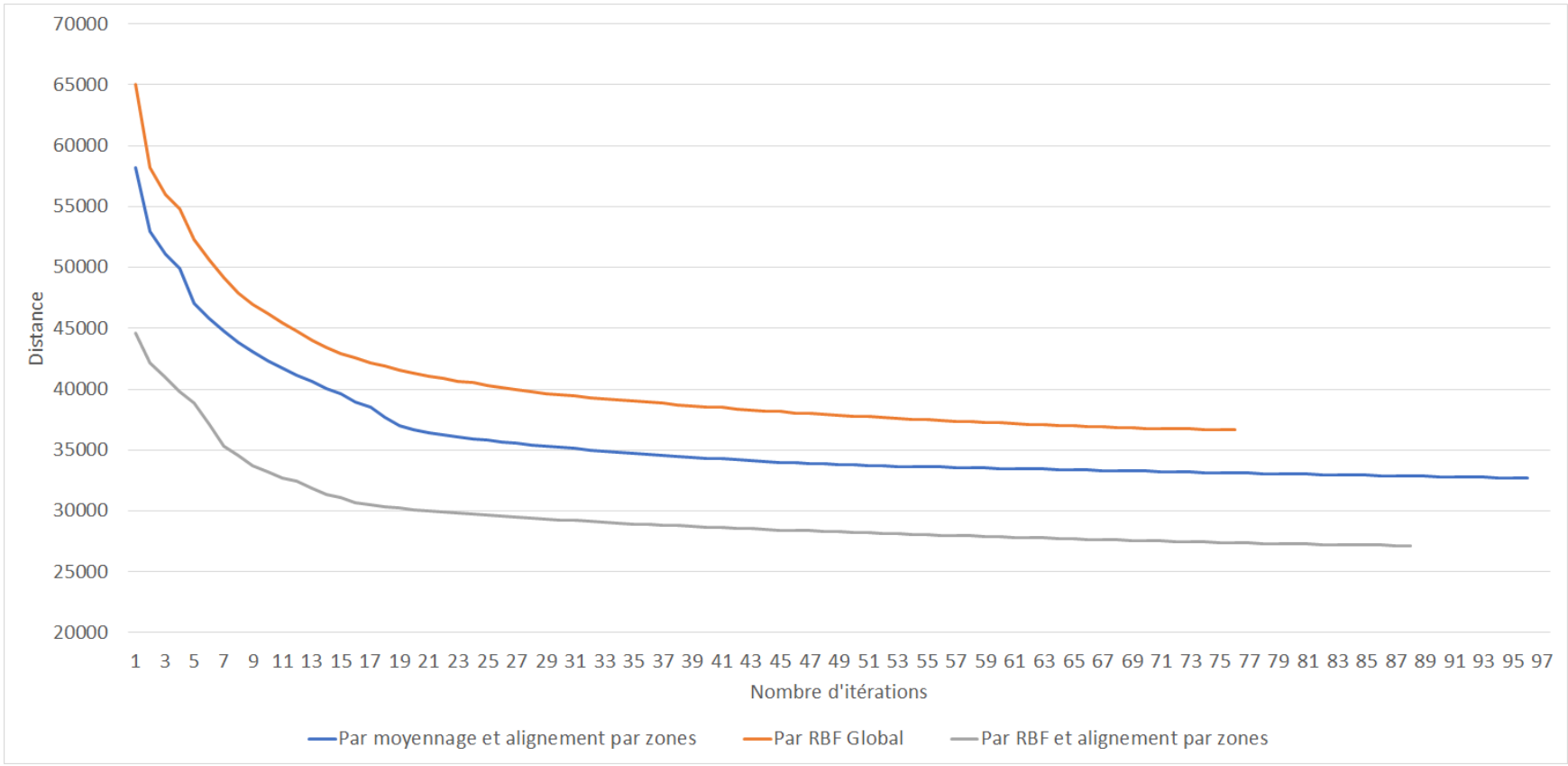

Figure A.4 Influence de la génération de la pose Neutre sur la performance «Laideur et Amour» sur le rig d'Elise. Le facteur de réalignement est 1.