



THÈSE DE L'UNIVERSITÉ DE LYON

Délivrée par

L'UNIVERSITÉ CLAUDE BERNARD - LYON 1

École Doctorale Informatique et Mathématiques

DIPLÔME DE DOCTORAT

(arrêté du 7 août 2006)

Paramétrisation et Transfert d'Animations Faciales
3D à partir de Séquences Vidéo :
vers des Applications en Temps Réel.

soutenue publiquement le 24 mars 2011

par Ludovic DUTREVE

Composition du jury:

Rapporteurs

Mr. Jean-Pierre JESSEL Professeur à l'Université Paul Sabatier

Mr. Marc NEVEU Professeur à l'Université de Bourgogne

Examineurs

Mr. Jean-Claude MARTIN Professeur à l'Université Paris-Sud 11

Mr. Renaud SÉGUIER Maître de Conférence à Supélec Rennes

Directeur

Mme. Saïda BOUAKAZ Professeur à l'Université Claude Bernard Lyon 1

Co-encadrant

Mr. Alexandre MEYER Maître de Conférence à l'Université Claude Bernard Lyon 1

Résumé

Paramétrisation et Transfert d'Animations Faciales 3D à partir de Séquences Vidéo : vers des Applications en Temps Réel.

L'animation faciale est l'un des points clés dans le réalisme des scènes 3D qui mettent en scène des personnages virtuels. Ceci s'explique principalement par les raisons suivantes : le visage et les nombreux muscles qui le composent permettent de générer une multitude d'expressions ; ensuite, notre faculté de perception nous permet de détecter et d'analyser ses mouvements les plus fins. La complexité de ce domaine se retrouve dans les approches existantes par le fait qu'il est très difficile de créer une animation de qualité sans un travail manuel long et fastidieux. Partant de ce constat, cette thèse a pour but de développer des techniques qui contribuent au processus de création d'animations faciales.

Trois thèmes sont principalement abordés. Le premier concerne la paramétrisation du visage pour l'animation. La paramétrisation a pour but de définir des moyens de contrôle pour pouvoir déformer et animer le visage. Le second s'oriente sur l'animation, et plus particulièrement sur le transfert d'animation. Le but est de proposer une méthode qui permette d'animer le visage d'un personnage à partir de données variées. Ces données peuvent être issues d'un système de capture de mouvement, ou bien elles peuvent être obtenues à partir de l'animation d'un personnage virtuel qui existe déjà. Enfin, nous nous sommes concentrés sur les détails fins liés à l'animation comme les rides. Bien que ces rides soient fines et discrètes, ces déformations jouent un rôle important dans la perception et l'analyse des émotions. C'est pourquoi nous proposons une technique d'acquisition mono-caméra et une méthode à base de poses références pour synthétiser dynamiquement les détails fins d'animation sur le visage.

L'objectif principal des méthodes proposées est d'offrir des solutions afin de faciliter et d'améliorer le processus de création d'animations faciales réalistes utilisées dans le cadre d'applications en temps réel. Nous nous sommes particulièrement concentrés sur la facilité d'utilisation et sur la contrainte du temps réel. De plus, nous offrons la possibilité à l'utilisateur ou au graphiste d'interagir afin de personnaliser sa création et/ou d'améliorer les résultats obtenus.

Mots clés : Animation faciale, temps réel, transfert d'animation, détails fins dynamiques, mise en correspondance de surface, capture de mouvement, fonctions à base radiale.

Abstract

Rigging and Retargeting of 3D Facial Animations from Video : towards Real-Time Applications.

Facial animation is one of the key points of the realism of 3D scenes featuring virtual humans. This is due to several reasons : face and the many muscles that compose it can generate a multitude of expressions; then, our faculty of perception provides us a great ability to detect and analyze its smallest variations. This complexity is reflected in existing approaches by the fact that it is very difficult to create an animation without a long and a tedious manual work. Based on these observations, this thesis aims to develop techniques that contribute to the process of creating facial animation.

Three main themes have been addressed. The first concerns the rigging issue of a virtual 3D face for animation. Rigging aims at defining control parameters in order to deform and animate the face. The second deals with the animation, especially on the animation retargeting issue. The goal is to propose a method to animate a character's face from various data. These data can be obtained from a motion capture system or from an existing 3D facial animation. Finally, we focus on animation fine-scale details like wrinkles. Although these are thin and discreet, their deformations play an important part in the perception and analysis of emotions. Therefore we propose a monocular acquisition technique and a reference pose based method to synthesise dynamically animation fine details over the face.

The purpose is to propose methods to facilitate and improve the process of creating realistic facial animations for interactive applications. We focused on ease to use in addition to the real-time aspect. Moreover, we offer the possibility to the user or graphist to interact in order to personalize its creation and/or improve the results.

Keywords : Facial animation, real-time, animation retargeting, dynamic fine-scale details, surface fitting, motion capture, radial basis functions.

Table des matières

1	Introduction Générale	1
1.1	Motivation	1
1.2	Domaines d'application	3
1.3	Projets PlayAll et PlayOnline	5
1.4	Contributions	6
1.5	Organisation du document	7
2	État de l'art	9
2.1	La tête : tour de contrôle de la communication	11
2.2	Génération d'un visage 3D	12
2.3	Systèmes de codage	14
2.3.1	<i>Facial Action Coding System (FACS)</i>	14
2.3.2	Standard MPEG-4	15
2.4	Paramétrisation (<i>Rigging</i>)	16
2.4.1	Interpolation de formes (<i>Blendshape</i>)	17
2.4.2	Modélisation physique	20
2.4.3	Déformation de surface	22
2.4.3.1	Déformation directe	22
2.4.3.2	Déformation de forme libres	23
2.4.3.3	Déformation à base de points de contrôle	23
2.4.3.4	Animation par squelette (<i>Skinning</i>)	24
2.4.4	Paramétrisation automatique ou assistée	26
2.5	Animation	26
2.5.1	Création manuelle d'animation et interpolation	27
2.5.2	Capture de mouvement	29
2.5.2.1	Techniques de capture	29
2.5.2.2	Capture photoréaliste	31
2.5.2.3	Détails fins	32
2.5.2.4	Adaptation des données capturées	33
2.5.3	Transfert d'animation (<i>Retargeting</i>)	36

2.5.4	Détails fins d'animation	38
2.5.5	Synthèse d'animation multimodale	40
2.6	Conclusion	40
3	Paramétrisation du visage	45
3.1	Introduction et motivation	45
3.2	Fonctions à base radiale pour l'interpolation	48
3.2.1	Aspects mathématiques	48
3.2.2	Exemples de fonctions radiales	49
3.2.3	Interpolation de données éparses par un modèle <i>RBF</i>	50
3.2.4	Ajout d'une transformation rigide à l'interpolation	51
3.3	Paramétrisation par approche procédurale	52
3.4	Paramétrisation par transfert	55
3.4.1	Transfert des points de contrôle	55
3.4.1.1	Détection et appariement des points repères	56
3.4.1.2	Transfert des points de contrôle	63
3.4.2	Transfert des poids os-sommet	63
3.4.2.1	Recalage dense de surfaces	63
3.4.2.2	Affectation des influences	66
3.5	Résultats	67
3.6	Conclusion	69
4	Transfert d'animation et évaluation	71
4.1	Introduction	71
4.2	Transfert d'animation	72
4.2.1	Transfert d'expressions	73
4.2.2	Transfert partiel	74
4.3	Transfert pour la capture de mouvement	74
4.4	Résultats et évaluation	75
4.5	Conclusion	78
5	Détails fins dynamiques	83
5.1	Introduction	83
5.2	Acquisition des détails d'animation	86
5.2.1	Image ratio	86
5.2.2	Calcul de la carte de normales	87
5.2.3	Adaptation au personnage 3D	90
5.3	Génération des détails fins d'animation en temps réel	91
5.3.1	Pose référence	91

5.3.2	Évaluation d'une pose quelconque par rapport à une pose référence	92
5.3.3	Masques d'influence	94
5.3.4	Fonction de modulation	96
5.3.5	Mélange des détails	97
5.4	Résultats	98
5.5	Discussion	101
5.6	Conclusion	105
6	Conclusion Générale	107
	Glossaire	129

Table des figures

1.1	Chaîne de production de l’animation faciale des méthodes proposées .	8
2.1	Anatomie musculaire du visage	13
2.2	Extrait du <i>FACS</i>	14
2.3	Les 84 <i>FDP</i> définis par le standard <i>MPEG-4</i>	16
2.4	Exemple de <i>blendshape</i>	18
2.5	Interférences lors d’un mélange de forme (<i>blendshape</i>)	20
2.6	Modélisation de la biomécanique du visage par un système masse-ressort à 3 couches	21
2.7	Modélisation de la biomécanique du visage par éléments finis	22
2.8	Personnage contrôlé par <i>skinning</i>	24
2.9	Illustration de quelques attributs d’un visage contrôlé par <i>skinning</i> .	26
2.10	Exemple d’une technique de <i>posing</i>	28
2.11	Capture de mouvement avec marqueurs	30
2.12	Système de capture utilisé pour le <i>Digital Emily Project</i>	32
2.13	Système de capture de mouvement proposé par Bickel <i>et al.</i>	34
2.14	Système de capture de mouvement utilisé par Na <i>et al.</i>	34
2.15	Système de capture de mouvement proposé par Bradley <i>et al.</i>	35
2.16	Système de transfert d’animation par Noh <i>et al.</i>	37
2.17	Bases de visèmes	41
3.1	Modification de la taille de l’enveloppe dans la tâche de paramétrisation d’un modèle	46
3.2	Influence de l’enveloppe de l’os dans la tâche de paramétrisation d’un modèle	46
3.3	Fonction de pondération	54
3.4	Exemple de génération de cartes de hauteur des visages	57
3.5	Coupe latérale d’une segmentation par montée des eaux	58
3.6	Opérateur de symétrie	60
3.7	Exemple du critère de sélection des graines	61

3.8	Séquence de segmentations	62
3.9	Transfert automatique de points de contrôle	64
3.10	Recalage dense de surface	65
3.11	Résultat du transfert des poids du <i>skinning</i>	67
3.12	Exemple de recalage sur des visages de type <i>cartoon</i>	68
3.13	Extraits d'animation des visage sources et cibles	69
4.1	Vue d'ensemble de notre méthode de transfert	73
4.2	Exemple de transfert d'une expression	74
4.3	Exemple de transfert partiel	75
4.4	Extraits d'une séance de capture	76
4.5	Comparaison de transfert avec et sans interpolation	77
4.6	Évaluation de l'erreur de transfert entre Simon et Barak	78
4.7	Évaluation de l'erreur de transfert entre Proog et Barak	79
4.8	Évaluation de l'erreur de transfert : résultats visuels	80
4.9	Exemple de transfert d'expression	81
5.1	Interpolation de l'expression vers le visage neutre	87
5.2	Calcul de l'image ratio	87
5.3	Calcul de la carte de normales à partir de l'image ratio	89
5.4	Adaptation de la carte de normales	90
5.5	Vue d'ensemble de la technique de détails fins dynamiques	92
5.6	Évaluation des poses	94
5.7	Masque d'influence	95
5.8	Fonction de modulation	97
5.9	Mélange de carte de normales	99
5.10	Utilisation d'une carte de normale capturée	100
5.11	Acquisition des cartes de normales	101
5.12	poses références de <i>Proog</i>	102
5.13	poses références de <i>Mister B.</i>	102
5.14	Les 6 expressions de base de <i>Proog</i>	103
5.15	Les 6 expressions de base de <i>Barak</i>	104

Chapitre 1

Introduction Générale

Sommaire

1.1 Motivation	1
1.2 Domaines d’application	3
1.3 Projets PlayAll et PlayOnline	5
1.4 Contributions	6
1.5 Organisation du document	7

1.1 Motivation

Le visage est une partie majeure du corps humain, autant pour la reconnaissance d’un individu que pour la communication par les expressions qu’il est capable de produire. C’est pourquoi c’est un élément indispensable dans un grand nombre d’applications qui mettent en scène des personnages virtuels. La complexité et la richesse des détails du visage humain en fait un des composants les plus difficiles à animer dans une scène en trois dimensions. Notre aptitude à donner une signification aux mouvements les plus fins ou à traduire l’expression d’un visage en émotions augmente encore plus la difficulté de créer une animation faciale réaliste. En outre, les contraintes supplémentaires imposées par les différents contextes d’utilisation rendent cette tâche encore plus délicate.

Depuis les travaux précurseurs de Frederic I. Parke [Par72], de nombreuses recherches ont contribué à l’évolution de l’animation faciale, et son utilisation a considérablement augmentée. Ses domaines d’application peuvent être regroupés dans deux grandes familles, la première concerne les applications en temps réel, dont l’animation et le rendu doivent être effectués en temps interactif, et dont les ressources

sont partagées entre différentes tâches (intelligence artificielle, réseau, graphe d'animation, interface, etc.) comme dans les jeux-vidéos, les mondes virtuels ou encore les systèmes d'interaction entre l'homme et la machine, etc. La seconde concerne les applications, dites « hors-ligne », qui n'exigent pas le temps réel, et pour lesquelles la qualité du résultat prime sur la rapidité des calculs, comme dans l'industrie cinématographique avec les films d'animations en 3D de plus en plus répandus, les effets spéciaux, les spots publicitaires, etc.

La création d'une animation faciale passe par différentes étapes :

- la modélisation ;
- la paramétrisation ;
- l'animation ;
- le rendu.

Il n'existe pas de système généralisé qui permette de répondre à la majorité des attentes et chacune de ces étapes peut être réalisée à l'aide de nombreuses méthodes. La modélisation peut par exemple être créée par un artiste qui sculpte virtuellement la morphologie du visage souhaité, ou bien par des techniques d'acquisition telles que les scanners qui permettent d'obtenir les données numériques d'un visage réel. La modélisation et le rendu sont des techniques indépendantes de l'animation, nous nous concentrons ici sur la paramétrisation ainsi que sur l'animation qui pilote les paramètres créés lors de l'étape de la paramétrisation pour déformer la géométrie du visage. Actuellement, pour générer une animation faciale en 3D, deux questions principales doivent être posées :

- Comment rendre le visage « animable » (paramétrisation, *rigging*) ?
- Comment animer le visage ?

Pour répondre à ces questions, il est nécessaire de connaître le contexte d'utilisation de l'animation faciale : en temps réel ou hors-ligne ; ressources disponibles (temps, ressources matérielles, ressources humaines, ressources financières) ; etc. Pour répondre au mieux à ces besoins, il est également important de prendre en compte les nombreuses difficultés qu'implique la création d'une animation faciale réaliste. En effet, celles-ci ont souvent besoin des compétences et d'un savoir faire important de la part des artistes, le temps de travail nécessaire est conséquent. Les systèmes mis en place sont souvent complexes et coûteux, comme pour la capture de mouvement par exemple, qui pour plus de réalisme vise à reproduire les expressions d'un acteur réel sur un personnage virtuel.

1.2 Domaines d'application

L'animation faciale peut être utilisée dans de nombreux domaines, allant des interfaces basées dialogue pour l'interaction homme-machine, d'un « speaker » virtuel pour les informations, aux loisirs numériques et aux divertissements. On peut considérer qu'il existe deux familles d'applications : les applications hors-ligne où l'utilisateur n'est que simple spectateur, et celles qui travaillent en temps réel où l'interactivité permet d'accroître l'immersion de l'utilisateur. Avant de rentrer dans des considérations techniques, nous présentons ici quelques domaines d'utilisation majeurs de l'animation faciale en trois dimensions.

Cinéma et télévision

L'industrie cinématographique a été l'une des premières à utiliser des visages virtuels. Le film *Tony de Peltrie* a mis en scène un personnage 3D avec une animation faciale expressive en 1985 [BL85]. Les studios *Pixar* ont également présenté dès le milieu des années 80 des courts métrages qui introduisent des visages animés comme le film *Tim Toy* (1988). Des longs métrages d'animation qui présentent des personnages « cartoons » ont vu le jour : *Toy Story* (1995), *Stuart Little* (1999), *Shrek* (2002), *L'âge de glace* (2003), *Wall-E* (2009), *Là-Haut* (2010), etc., ainsi que des productions avec des personnages plus réalistes comme *Final Fantasy* (2001) ou *Polar Express* (2004). L'animation faciale a également été utilisée dans des films « classiques », soit pour réaliser certaines scènes de cascades comme dans *Matrix Reloaded* (2002) [BPL⁺03, BL03], soit pour insérer un personnage virtuel dans des scènes réelles : *King-Kong* (2005) [Sag06], la trilogie *Le Seigneur des Anneaux* (2001, 2002, 2003), *Pirate des Caraïbes 2* (2006), *Avatar* (2010), etc. Aujourd'hui, de plus en plus de films d'animation sortent chaque année. Les moyens métrages pour enfants, les spots publicitaires ou encore les clips musicaux utilisent aussi abondamment la 3D et les personnages virtuels. Les chaînes de production peuvent cependant varier d'un film à un autre, en effet, certains d'entre eux animent des personnages non-réels comme les jouets de *Toy-Story*, d'autres animent des personnages fantastiques mais dirigés par un acteur réel, d'autres encore animent des personnages qui se doivent d'être aussi réels que l'acteur qu'ils incarnent. Ces productions hors-ligne, et le cinéma en particulier, disposent de moyens importants. Notamment en terme de temps et de ressources. De plus, la non-interactivité des scènes produites permet aux artistes de travailler sur un socle scénaristique fixe. Cependant, les exigences de certaines productions impliquent des efforts de plus en plus conséquents et le besoin d'outils, à tous les niveaux de la chaîne de production, augmente.

Jeux vidéo et mondes virtuels

Les jeux vidéo et les mondes virtuels ont également un besoin croissant d'animations faciales de qualité. Depuis les premiers jeux vidéo 3D du milieu des années 90 mettant en scène des personnages réalistes, jusqu'à aujourd'hui avec les jeux massivement multijoueurs (*Massively Multiplayer Online Role-Playing Game*, **MMORPG**) et les mondes virtuels, le nombre et la variété de visages et d'animations ont considérablement augmentés. Le réalisme d'un personnage joue un rôle primordial dans l'immersion du joueur ou de l'utilisateur. L'animation et le rendu des personnages doivent être réalisés en temps réel, ce qui impose des contraintes importantes, sans oublier le fait que le ou les visages ne constituent qu'une partie minime d'une scène complexe pour laquelle les ressources sont partagées. De plus, l'interactivité de telles applications passe par des personnages qui agissent en fonction des événements de l'utilisateur et dont les actions et les paroles sont beaucoup moins fixées et scénarisées que pour les applications hors-ligne. Ces personnages virtuels peuvent être utilisés sous trois formes différentes : le personnage de l'utilisateur, les personnages des autres utilisateurs, qui doivent refléter au mieux les attitudes et émotions pour améliorer l'immersion, ainsi que les personnages non-joueurs, dirigés par l'application, mais agissant tout de même en fonction des utilisateurs. On peut également noter d'importantes différences liées à la nature des personnages qui, comme dans les productions cinématographiques, dans de nombreux jeux-vidéos ou des mondes virtuels comme *Second life*, peuvent incarner des personnages réalistes, des personnages non-réalistes (citons par exemple les **MMORPG**, comme *World of Warcraft* où les personnages sont des humanoïdes issus de l'univers de la *fantasy*) et des personnages réels (principalement dans les jeux de sport). L'importance du réalisme est d'autant plus grande dans les environnements multi-utilisateurs puisque les personnages sont les avatars de personnes réelles.

Talking Head

Des visages virtuels peuvent également être utilisés dans le cadre de « têtes parlantes » pour lire l'actualité en *streaming* sur Internet par exemple. L'animation est alors créée à partir de textes ou de sons et travaille sur la synchronisation des lèvres. Une autre utilisation concerne les dialogues homme-machine : l'avancée des techniques d'intelligence artificielle permet en effet de créer des agents virtuels capables de répondre aux questions d'un utilisateur. En plus de la parole et de l'animation de la bouche qui lui est liée, ils peuvent être dotés d'émotions et d'expressions faciales, permettant de rendre leurs messages et leurs réponses plus humains. Ces agents peuvent être utilisés pour le commerce en ligne, le service après-vente, le support

technique, la réhabilitation psychiatrique¹, etc. On peut également citer les visioconférences où chaque participant peut être incarné par son avatar. Dans ce cadre et en considérant que les utilisateurs sont novices dans le domaine de l'animation faciale, la création de l'avatar et de son animation doit se faire le plus simplement possible.

1.3 Projets PlayAll et PlayOnline

Les travaux menés au cours de cette thèse ont été financés par les projets PlayAll (2007-2009) puis PlayOnline (2009-2011). PlayAll est un projet dont le but est de fédérer les talents et technologies de quatre sociétés de jeux vidéo (Darkworks, Kylo-tonn, White Birds Productions et Wizarbox) pour créer et valoriser une plateforme commune ainsi qu'un centre de développement. Il associe également des sociétés de *middleware* spécialisées (Spirops, Bionatics et Voxler) et des partenaires académiques (LIRIS, CNAM, Telecom Paris Tech, Telecom Sud Paris, ENJMIN). L'objectif est de développer une chaîne de production de jeux vidéo complète et d'un standard technologique unifié et élargi. Les plateformes visées (présentes ou futures) sont les consoles de salon, les consoles portables, et les PC. PlayAll est financé par le pôle de compétitivité Cap Digital et est membre du programme IDEES de Microsoft France.

PlayOnline fait suite au projet PlayAll, dont il conserve la structure collaborative entre des studios et laboratoires français. Résolument tourné vers l'avenir du jeu vidéo, PlayOnline s'attache à mettre en place une nouvelle manière d'aborder le jeu vidéo au sein d'applications réseaux où les moyens de communication et d'interaction avec les autres joueurs seront grandement améliorés.

Les travaux de l'équipe SAARA dans le cadre de PlayAll et PlayOnline s'inscrivent dans le domaine de l'animation, aussi bien au niveau de la capture que de la synthèse. Trois thématiques différentes sont abordées : le corps entier, la main et le visage. Ces travaux poursuivent différents buts tels que proposer des outils pour la capture de mouvement, des outils pour faciliter les tâches des graphistes, ou encore l'accessibilité pour un utilisateur novice. Ils visent également à donner au joueur la possibilité de transmettre ses propres émotions et/ou gestes à son avatar afin de rendre les jeux plus proches de la réalité, plus conviviaux, et de développer ainsi de nouveaux modes d'interaction. Ces travaux ont pour ambition d'enrichir les jeux vidéo de nouvelles métaphores d'interaction basées vidéo en utilisant de simples webcams dont

1. Par exemple pour l'étude de la perception des émotions chez les personnes qui possèdent des déficiences mentales

la qualité ne cessent d'accroître et dont le coût les rend de plus en plus accessibles pour le grand public. Différents aspects sont alors explorés :

- capture avec ou sans marqueur d'une personne en entier et/ou d'une partie spécifique du corps ;
- analyse de mouvements ;
- transfert de ses mouvements à un personnage virtuel affiché sur une machine distante ;
- niveaux de détails pour l'animation ;
- etc.

1.4 Contributions

Les animations faciales nécessitent de plus en plus d'outils pour les concevoir, en raison notamment du nombre progressif de leurs utilisations et de l'augmentation de leur complexité. C'est pourquoi les travaux présentés dans ce document sont orientés vers l'aide à la création d'animations faciales pour les applications en temps réel. Les studios de développement ont un besoin croissant de techniques pour produire et intégrer des animations faciales à leurs productions. Par ailleurs, une meilleure immersion passe par un personnage encore plus proche au niveau de l'attitude et de l'émotion qu'il doit transmettre, ce qui nécessite à la fois des outils de capture, de traitement, et de rendu de ces caractéristiques. Les interactions avec l'utilisateur, même novice, sont de plus en plus présentes, c'est pourquoi la facilité d'utilisation est un critère important en plus de l'aspect temps réel.

Nos travaux se divisent en trois contributions principales, chacune se déclinant en différentes apports :

- La première partie de nos travaux concerne la paramétrisation du visage 3D pour l'animation. La paramétrisation est une étape délicate et longue à réaliser, c'est pourquoi nous nous sommes concentrés sur des méthodes permettant d'accélérer, de faciliter ou même d'automatiser cette tâche. Nos apports quant à la **paramétrisation d'un visage** sont :
 - le transfert automatique des points de contrôle et des paramètres du *skin-ning* basé sur un nouvel algorithme automatique de recalage non-rigide d'un visage 3D ;
 - la paramétrisation semi-automatique à base d'heuristiques et de données morphologiques.

- Une fois la paramétrisation effectuée, le visage doit être animé. Les avancées récentes en terme de capture de mouvement permettent d’obtenir des animations réalistes de qualité. Cependant, les moyens nécessaires à leur mise en place sont complexes et les résultats sont peu flexibles. C’est pourquoi nous avons proposé une nouvelle méthode de transfert. Nos contributions quant à la **génération d’animation par transfert** sont :
 - le transfert en temps réel sur un nouveau visage d’une animation 3D existante d’un autre visage à la morphologie différente ;
 - le transfert en temps réel sur un visage 3D d’une animation 2D ce qui permet de créer un système de capture de mouvement à faible coût ;
 - le transfert partiel pour plus de flexibilité qui permet de fusionner plusieurs animations ou de ne sélectionner qu’une partie du visage à transférer ;
 - l’étude qualitative des différentes fonctions à base radiale pour l’interpolation dans le cadre de l’animation faciale.

- Les détails fins d’animation liés aux déformations faciales comme les rides sont très importants pour l’analyse et la reconnaissance des émotions. Cependant, il est difficile de les appliquer à un visage dans le contexte d’une application en temps réel. C’est pourquoi nous proposons deux méthodes pour capturer ces détails à partir d’un visage réel et les appliquer ensuite à un visage virtuel. Nos apports dans l’**ajout de détails fins dynamiques à l’animation en temps réel** concernent :
 - l’acquisition mono-caméra de détails d’animation ;
 - la génération en temps réel de détails fins à partir de poses références ;
 - l’utilisation des paramètres de la paramétrisation par *skinning* pour associer chaque point de la surface de la peau avec les poses références.

Le schéma présenté sur la Figure 1.1 illustre la chaîne de production d’animation faciale que nous proposons.

1.5 Organisation du document

Les travaux réalisés au cours de la thèse sont présentés comme suit :

Le Chapitre 2 présente un état de l’art sur l’animation faciale. Il présente les différentes étapes nécessaires à la réalisation d’une animation faciale et les diverses méthodes proposées dans la littérature. Une conclusion donnée en fin de chapitre permet de faire le bilan des méthodes exposées et donne les lignes directrices des

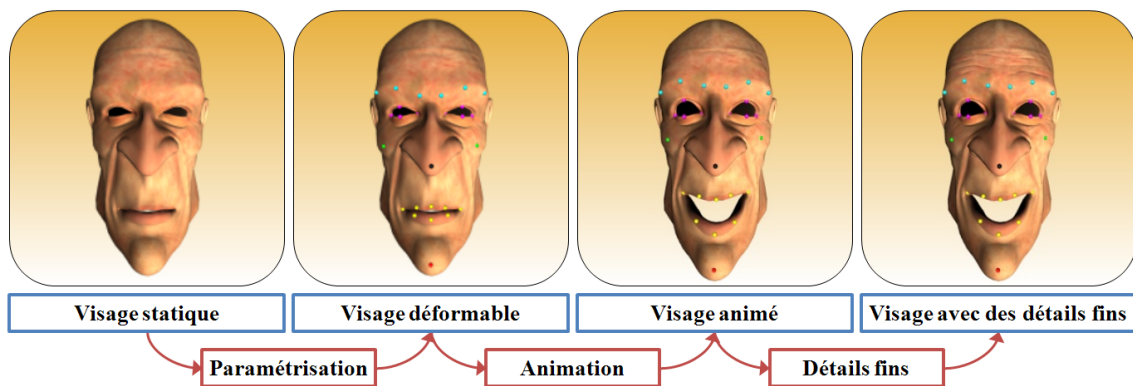


FIGURE 1.1 – Chaîne de production de l’animation faciale des méthodes proposées. Nos travaux concernent chacune des étapes entourées en rouge.

travaux que nous avons réalisés.

Le Chapitre 3 présente les premières contributions de cette thèse qui concernent la paramétrisation du visage 3D pour l’animation. Il s’agit de deux techniques qui visent à aider de façon automatique, ou semi-automatique le graphiste pour la paramétrisation d’un visage par *skinning*.

Le Chapitre 4 propose une technique de transfert d’animation à base de points de contrôle, qui permettent d’animer un visage à partir d’un autre visage animé en 3D, ou bien à partir de données issues d’un système de capture de mouvement quelconque.

Le Chapitre 5 présente une technique d’acquisition des détails fins liés aux déformations faciales comme les rides, et une technique pour ajouter facilement ces détails fins dynamiques et en temps réel sur un visage animé.

Chapitre 2

État de l'art

Sommaire

2.1	La tête : tour de contrôle de la communication	11
2.2	Génération d'un visage 3D	12
2.3	Systèmes de codage	14
2.3.1	<i>Facial Action Coding System (FACS)</i>	14
2.3.2	Standard MPEG-4	15
2.4	Paramétrisation (<i>Rigging</i>)	16
2.4.1	Interpolation de formes (<i>Blendshape</i>)	17
2.4.2	Modélisation physique	20
2.4.3	Déformation de surface	22
2.4.4	Paramétrisation automatique ou assistée	26
2.5	Animation	26
2.5.1	Création manuelle d'animation et interpolation	27
2.5.2	Capture de mouvement	29
2.5.3	Transfert d'animation (<i>Retargeting</i>)	36
2.5.4	Détails fins d'animation	38
2.5.5	Synthèse d'animation multimodale	40
2.6	Conclusion	40

Le visage est au centre de nombreux thèmes abordés par la recherche dans les secteurs de la vision par ordinateur et de l'informatique graphique, allant de la biométrie et la sécurité, aux divertissements et aux loisirs numériques, en passant par la médecine et les applications thérapeutiques. Nous nous concentrons ici sur l'animation faciale en trois dimensions dont le domaine a été ouvert par les travaux précurseurs de F. I. Parke [Par72]. Selon J.-Y. Noh [NN98], le but ultime de la recherche dans le domaine de l'animation faciale est un système qui :

1. créé une animation réaliste ;
2. fonctionne en temps réel ;
3. est automatisé autant que possible (en laissant une possibilité d'intervention pour l'utilisateur) ;
4. s'adapte facilement à différents visages.

Un tel système n'a pas encore vu le jour, mais certains points ont été obtenus plus ou moins indépendamment. Par exemple, le photo-réalisme est pratiquement atteint dans les productions cinématographiques récentes. Cependant, le travail manuel à fournir est colossal et ces techniques sont loin d'être exploitables en temps réel. De même, des systèmes interactifs et flexibles ont été proposés, avec un niveau d'automatisation plutôt avancé. Ainsi, pour réaliser un système d'animation, des compromis entre les quatre points cités ci-dessus doivent être faits.

Il est difficile de donner un classement strict de l'ensemble des travaux qui ont été proposés dans la littérature. Ceci pour les raisons : qu'il n'existe pas toujours de frontière entre les différentes techniques ; qu'une partie d'une méthode peut être appliquée à une autre ; que certains travaux se concentrent sur un problème très spécifiques de l'animation faciale alors que d'autres cherchent à proposer un système complet ; etc. Cette diversité nous a amené à adopter un plan de présentation de l'état de l'art différent de ceux que l'on trouve habituellement dans les études de synthèse [NN98, OW04, Wil06, RP06, PL06, DN07].

Cette étude est divisée en cinq parties. Afin de comprendre les mécanismes mis en jeu par les expressions et les animations faciales, la première section décrit brièvement l'anatomie du visage humain (Section 2.1). Puis nous décrivons quelques techniques pour générer un visage en trois dimensions (Section 2.2). Ensuite nous présentons les systèmes de codage utilisés dans de nombreux travaux sur l'animation faciale (Section 2.3), il s'agit d'abstractions permettant de définir une expression ou une animation faciale sans connaissance sur la technique d'animation sous-jacente. La suite de l'état de l'art est dédiée à l'animation. La quatrième partie aborde les différentes méthodes de paramétrisation du visage pour l'animation (Section 2.4), c'est-à-dire comment, à partir d'un visage statique, il est possible d'obtenir un visage dynamique déformable. La paramétrisation est souvent désignée par le terme *rigging* dans la littérature internationale. Elle décrit comment, concrètement, le visage virtuel est déformé en fonction d'un certain nombre de paramètres. Enfin la dernière partie traite des différentes façons d'animer le visage une fois celui-ci paramétrisé

(Section 2.5). Nous nous concentrons ici sur l'animation faciale d'une manière générale. Le rendu sera peu abordé étant donné sa spécificité et son indépendance vis-à-vis de l'animation. Nous introduirons cependant la technique du *bump-mapping* que nous avons utilisée dans notre dernière contribution.

2.1 La tête : tour de contrôle de la communication

Le visage intervient fortement dans le processus de communication entre êtres humains, ceci sur deux niveaux :

- au niveau de la parole pour laquelle la bouche et la mâchoire jouent un rôle important ;
- à travers la communication non verbale réalisable grâce à la multitude de déformations que le visage est en mesure de produire, des plus larges aux plus fines.

Ces deux formes de communications sont rendues possibles grâce à un système biomécanique complexe, composé de trois structures majeures : la peau, les os et les muscles ainsi que d'autres organes comme les yeux, la langue, les oreilles, etc.

La peau, plus grand organe du corps humain, est structurée en trois couches distinctes : l'épiderme, le derme et l'hypoderme. Elle joue un rôle de protection face aux agressions extérieures (bactéries, substances toxiques, etc.), d'isolation thermique pour réguler la température du métabolisme, et de stockage (d'eau, de graisse et de sang). De nombreuses terminaisons nerveuses lui confèrent également un rôle prépondérant dans la perception du toucher.

Les os de la tête sont regroupés en deux parties : le crâne et le visage. Le crâne est formé de huit os qui forment la boîte crânienne qui contient l'encéphale. Le visage est quant à lui composé de quatorze os, eux-mêmes répartis en deux massifs osseux, la mâchoire supérieure et la mâchoire inférieure. L'unique articulation du visage qui met en opposition ces deux parties est utilisée pour la mastication et pour la parole. Les os de la tête lui donnent sa forme générale et offre une stabilité à l'ensemble des muscles et organes qui la constituent.

Les muscles sont présents en grande quantité sur le visage, ils sont les acteurs principaux des expressions faciales. Il en existe environ une cinquantaine identifiable ; certains d'entre eux peuvent être considérés comme des groupes de plus petits muscles. On peut les classer par leur zone d'influence : le scalpe, la bouche, le nez, le cou, la mâchoire et les paupières. Contrairement aux muscles du corps, ils ne

sont pas toujours attachés directement aux os mais à la peau et leurs contractions entraînent alors des déformations subtiles du visage. De plus, ces déformations de la peau provoquent également des déformations encore plus fines, comme les rides par exemple, perpendiculaires à l'action du ou des muscles. La Figure 2.1 illustre la complexité du système musculaire faciale.

Enfin, quelques organes jouent un rôle très important dans la communication, comme les yeux, aussi bien pour la perception d'informations visuelles que pour la diffusion d'émotions ; la langue pour la communication orale et la perception du goût ; le nez ou les oreilles pour les perceptions des odeurs ou des sons ; ou encore le cou qui sert de support à la tête et qui intervient également dans ses mouvements et sa posture. D'une manière générale, la tête, et le visage plus particulièrement, est la zone centrale de la communication puisque les cinq sens y sont perçus (exclusivement pour le goût, l'odorat, l'ouïe et la vue, partiellement pour le toucher), et qu'elle est fortement impliquée dans la diffusion de la parole et des émotions.

Cette complexité de la tête explique en partie pourquoi il est difficile de modéliser et d'animer un visage virtuel de façon réaliste. Le nombre de déformations possibles est conséquent ainsi que les degrés de liberté nécessaires à une paramétrisation réussie. Sans compter sur notre aptitude développée depuis notre naissance à analyser les déformations faciales et à en obtenir des informations sur l'état d'esprit, l'humeur, les émotions, ou le caractère de la personne face à nous.

2.2 Génération d'un visage 3D

La première étape dans la construction d'un visage 3D animé est de générer la géométrie du visage. Celui-ci peut être complètement synthétique, c'est-à-dire issu de l'imagination et du travail manuel d'un graphiste ou bien il peut être obtenu à partir d'un visage réel. La modélisation doit respecter certaines contraintes pour que le visage puisse être animé correctement comme les ouvertures au niveau des yeux et de la bouche par exemple. Il faut aussi associer au maillage ou à la surface des coordonnées textures. Les visages peuvent être créés directement avec l'aide de logiciels de modélisation 3D lorsqu'il s'agit de visages imaginés ou même lorsqu'il s'agit de reproduire un visage réel (l'utilisation de calques de photo aide alors l'utilisateur à la réalisation cette tâche). Les scanners ou les systèmes d'acquisition stéréo sont le plus fréquemment utilisés pour reproduire avec fidélité les formes d'un visage réel [LTW95, GGW⁺98].

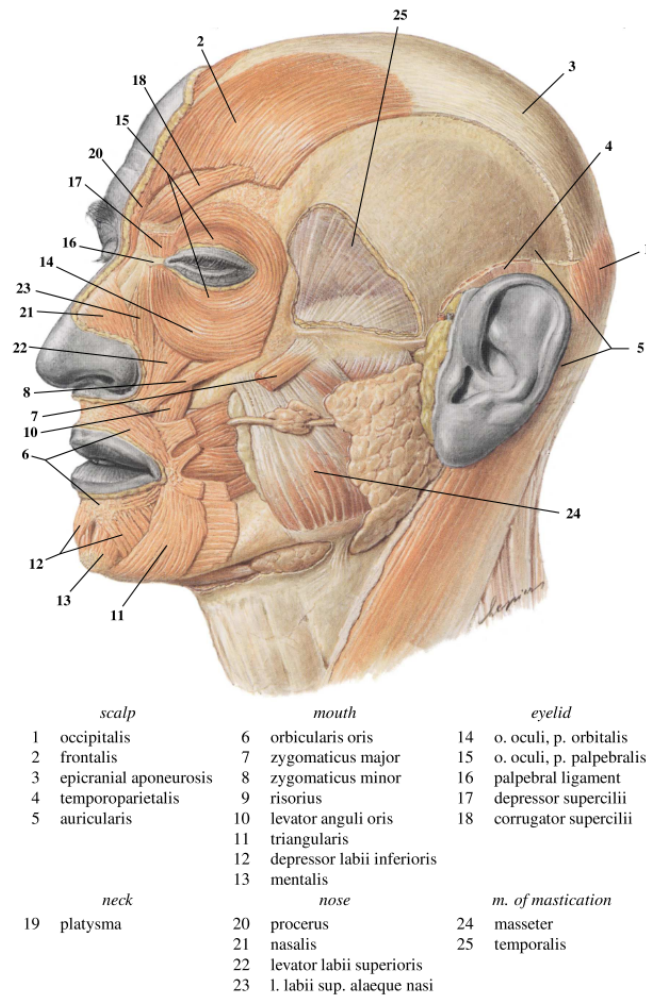


FIGURE 2.1 – Vue de côté d'une partie des muscles du visage [Sob01].

Des méthodes qui visent à réduire les coûts de production liés à l'usage d'un scanner et à faciliter la réalisation d'un visage virtuel à partir d'un visage réel ont été proposées. Une majorité d'entre elles vise à déformer un visage virtuel générique vers le visage voulu. Pour cela des techniques de déformation et d'interpolation de formes à partir de points caractéristiques positionnés sur des photographies sont utilisées [PHL⁺98, KHYS02, KHS03]. Ambrosini *et al.* utilisent les points de contrôle définis par la norme *MPEG-4* et les fonctions à base radiale (*Radial Basis Functions*, *RBF*) pour déformer un modèle polygonal vers la forme d'un autre modèle [ACLP98]. Weissenfeld *et al.* proposent une technique plus complexe dont le but est d'adapter un modèle générique aux données d'un visage scanné [WSQO05]. Un filtre passe-bas va progressivement être appliqué et réduire les hautes-fréquences de la géométrie des modèles scannés et génériques. En utilisant également des points de contrôle et des déformations par *RBF*, le modèle scanné va être adapté progressivement avec l'aide de la hiérarchie des maillages obtenus. Itérativement, le modèle générique est adapté

au maillage filtré de la plus basse résolution jusqu'à la plus fine.

2.3 Systèmes de codage

Cette partie de l'état de l'art est consacrée à différents systèmes de codage d'expressions ou d'animations faciales. Bien que ces systèmes ne soient pas utilisés systématiquement par chacune des méthodes existantes d'animation faciale, il est important de les connaître. Ici, nous ne nous posons pas la question de savoir comment le visage virtuel est déformé, mais nous présentons des méthodes permettant de décrire une expression ou une animation faciale. Certaines sont issues de travaux dans le domaine de la psychologie [EF78, EFH02] et n'ont pas pour vocation initiale d'être utilisées en informatique graphique. D'autres ont été proposées uniquement dans ce but, par exemple la partie dédiée à l'animation faciale de la norme *MPEG-4* [MPE01], ou les systèmes *MPA* [KMMTT91] ou *AMA* [MTPT87].

2.3.1 Facial Action Coding System (*FACS*)

Le *FACS* est un système de taxinomie des expressions faciales développé par les psychologues P. Ekman et W. Friesen en 1978 [EF78] puis révisé en 2002 [EFH02]. Il décompose les déformations faciales en 46 unités d'action (*Action Units, AU*). Une unité d'action définit un changement d'apparence, ainsi que le ou les muscles intervenant dans cette déformation. Les expressions faciales sont alors décrites par des combinaisons d'unités d'action et de leurs intensités. La Figure 2.2 présentent les *AU* de la partie supérieure du visage.

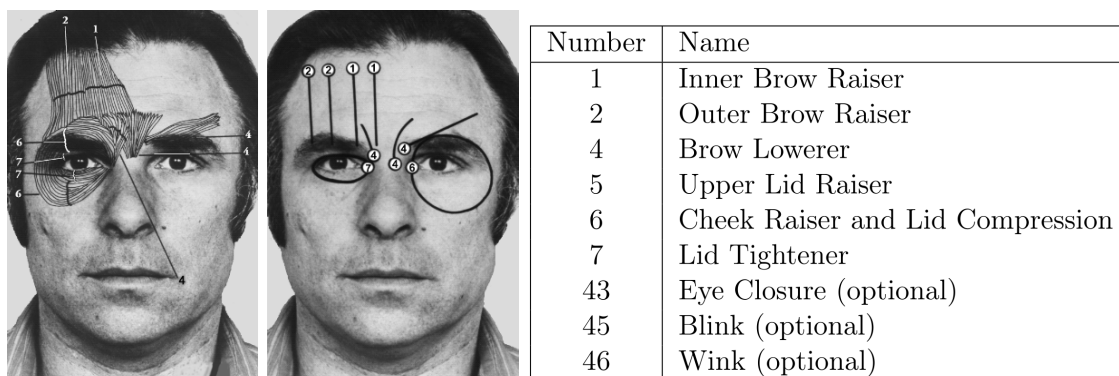


FIGURE 2.2 – *AU* de la partie supérieure du visage. La première image présente les différents muscles intervenant dans chacune des *AU*, la seconde image présente l'action produite par ces muscles. Le tableau récapitule les *AU* et leurs noms.

Cette indexation des déformations faciales a été utilisée dans de nombreux travaux s'intéressant à la reconnaissance des expressions. Parmi les travaux récents, nous pouvons citer [LKCL98, WO06, WJL09, LBL09, PPGG10]. Cette indexation a également été utilisée dans le cadre de la génération d'expressions faciales et la modélisation physique du visage [WF94, ZPS01, RDW09]. Il est intéressant de noter que l'atomisation des déformations du visage ne se fait pas au niveau de l'action de chacun des muscles, mais au niveau de l'action combinée de groupements musculaires. On peut ainsi déduire que pour obtenir une animation faciale complète, il n'est pas forcément nécessaire de modéliser l'action de chaque muscle, mais plutôt d'être capable de produire chacune des unités d'action.

2.3.2 Standard MPEG-4

La norme *MPEG-4* contient une partie dédiée à l'animation du corps humain, dont le visage, notamment à travers un système de codage d'animation faciale [MPE01, Ost98], celui-ci a été spécifiquement créé pour permettre à différents terminaux ou implémentations d'utiliser un protocole commun.

Le standard définit entre autres :

- 84 points caractéristiques (*Facial Definition Parameters, FDP*)
- 68 paramètres d'animation (*Facial Animation Parameters, FAP*)
- 6 unités basées sur des mesures anthropométriques du visage (*Facial Animation Parameter Units, FAPU*)

Les *FAP* sont les paramètres d'animation qui décrivent comment le visage se déforme, aussi bien pour produire des visèmes que pour générer des mouvements de la tête et des expressions. Les *FDP* sont des points caractéristiques du visage (voir la Figure 2.3). Ils jouent deux rôles principaux : le premier est de permettre la mesure des caractéristiques d'un visage ainsi que le calibrage d'un modèle générique sur un modèle défini par des données obtenues à partir d'un visage réel ; le second rôle est directement lié à l'animation puisque les *FAP* décrivent les déplacements des *FDP*. Les *FAPU* sont des unités de normalisation basées sur des distances et des angles entre certains points caractéristiques du visage qui permettent d'interpréter les valeurs des *FAP* en fonction de la morphologie du visage animé. Les distances, déplacements ou autres paramètres sont alors décrits de façon relative plutôt que de façon absolue.

Beaucoup de travaux s'appuient sur cette normalisation de l'animation faciale. Des techniques simples à base d'interpolation de données éparses (voir la Section 3.2)

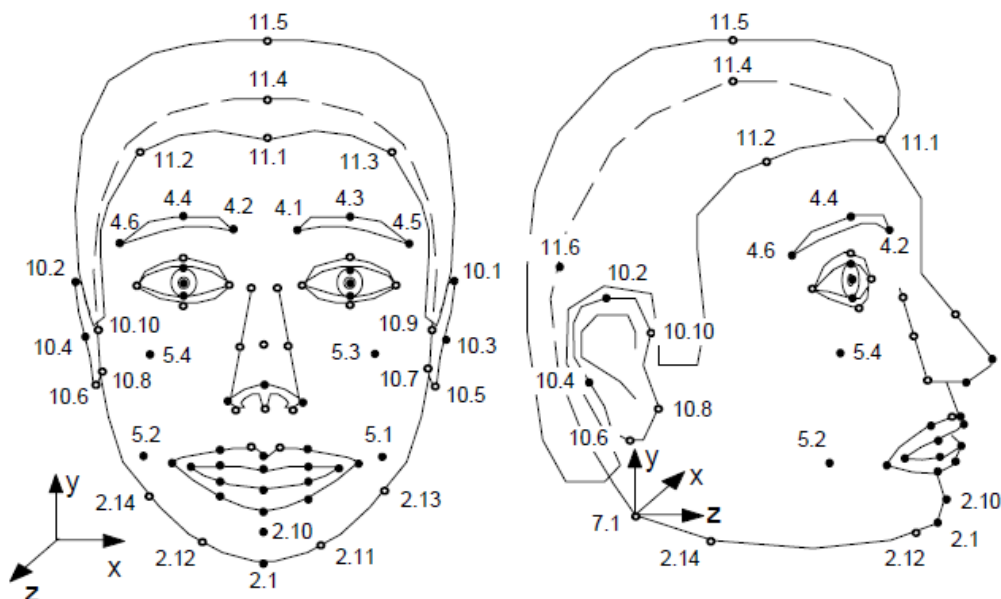


FIGURE 2.3 – Les 84 *FDP* définis par le standard *MPEG-4* [MPE01]. Les points pleins sont ceux influencés par les paramètres d’animation *FAP*. Les points creux sont fixes et sont utilisés pour le calibrage des visages. Les *FDP* sont groupés par zone (9 pour le nez et 3 pour les yeux par exemple).

permettent de calibrer un visage générique sur n’importe quel autre visage à partir d’un ensemble restreint de points caractéristiques [ACLP98]. D’autres chercheurs ont proposé des méthodes de paramétrisation permettant d’animer un visage en accord avec la norme *MPEG-4* [LP99, PCLB01, GEMT05], ou ont développé des algorithmes permettant de compresser les données utilisées pour l’animation (les *FAP*) [LP01]. Balci *et al.* ont proposé une librairie *open-source* qui permet la génération d’animation faciale basée sur la norme *MPEG-4* [Bal04] puis une plateforme complète pour la création d’agents conversationnels [BNZP07]. Courgeon *et al.* utilisent un modèle basé sur la norme *MPEG-4* dans le cadre de leurs travaux sur la perception des émotions et des expressions d’un personnage virtuel [CBM09, CAG10]. Pasquariello *et al.* ont proposé un système basé sur des pseudo-muscles compatible avec cette norme [PP01].

2.4 Paramétrisation (*Rigging*)

Les systèmes de codage précédemment cités peuvent être considérés comme des paramétrisations abstraites de l’animation faciale, les techniques que nous présentons dans cette section concernent un aspect plus technique qui décrit comment différents paramètres définis vont réellement agir sur la géométrie du visage pour le

déformer. Les maillages ou les surfaces utilisés dans la plupart des applications 3D sont statiques par défaut. Pour les animer, il faut alors définir un certain nombre de paramètres qui constituent la paramétrisation. La modification de leurs valeurs au cours du temps permet ensuite de réaliser l’animation. Trois catégories principales de paramétrisation existent pour remplir cette tâche : il s’agit de la famille des méthodes de simulation physique ; de la famille des méthodes de mélange de formes, appelées communément *blendshape* ou interpolation de formes ; et de la famille des méthodes de déformations de surface. Nous décrivons par la suite les avantages et inconvénients de chacune et les types d’applications pour lesquelles elles sont adaptées.

2.4.1 Interpolation de formes (*Blendshape*)

Les techniques de *blendshape* consistent à créer un espace linéaire d’expressions faciales et de réaliser des interpolations entre les formes de base pour en obtenir une nouvelle. Une expression est alors définie par une combinaison linéaire de n vecteurs, chaque vecteur définissant une forme. Dans le cas d’un maillage, il s’agit des coordonnées de chacun de ses sommets sachant que la topologie entre chaque forme doit être identique. Dans le cas d’un modèle *spline*, il s’agit de la position des points de contrôle. Ainsi, la position du $j^{\text{ème}}$ sommet \vec{v}_j d’un maillage est définie par la formule :

$$\vec{v}_j = \sum_{i=1}^n \alpha_i \vec{v}_{ji} \quad (2.1)$$

avec \vec{v}_{ji} la position de ce même sommet pour la forme i ($1 < i < n$) et α_i le poids appliqué à la forme i . Une autre formulation également utilisée, appelée *delta blendshape*, se base sur des vecteurs déplacements $\delta \vec{v}_{ji}$ qui sont ensuite pondérés et ajoutés à la forme initiale à la place d’une pondération directe des formes complètes :

$$\vec{v}_j = \vec{v}_{j0} + \sum_{i=1}^n \alpha_i \delta \vec{v}_{ji} \quad (2.2)$$

Étant donné qu’il est très facile de passer d’une formulation à une autre, le choix a peu d’importance, bien que les modelleurs aient une préférence pour la première et les animateurs pour la seconde selon [LMDN05]. La Figure 2.4 présente un exemple de pondération de formes. La solution couramment utilisée pour générer de nouvelles expressions avec un modèle contrôlé par *blendshape* est de manipuler des *sliders*¹, qui correspondent aux poids attribués aux formes de base. Ces fonction-

1. composants d’une interface graphique qui permettent de déplacer un curseur sur une échelle graduée

nalités existent nativement dans les logiciels commerciaux tels que *Maya*² ou *3ds Max*³ ou les logiciels ouverts tels que *Blender*⁴.

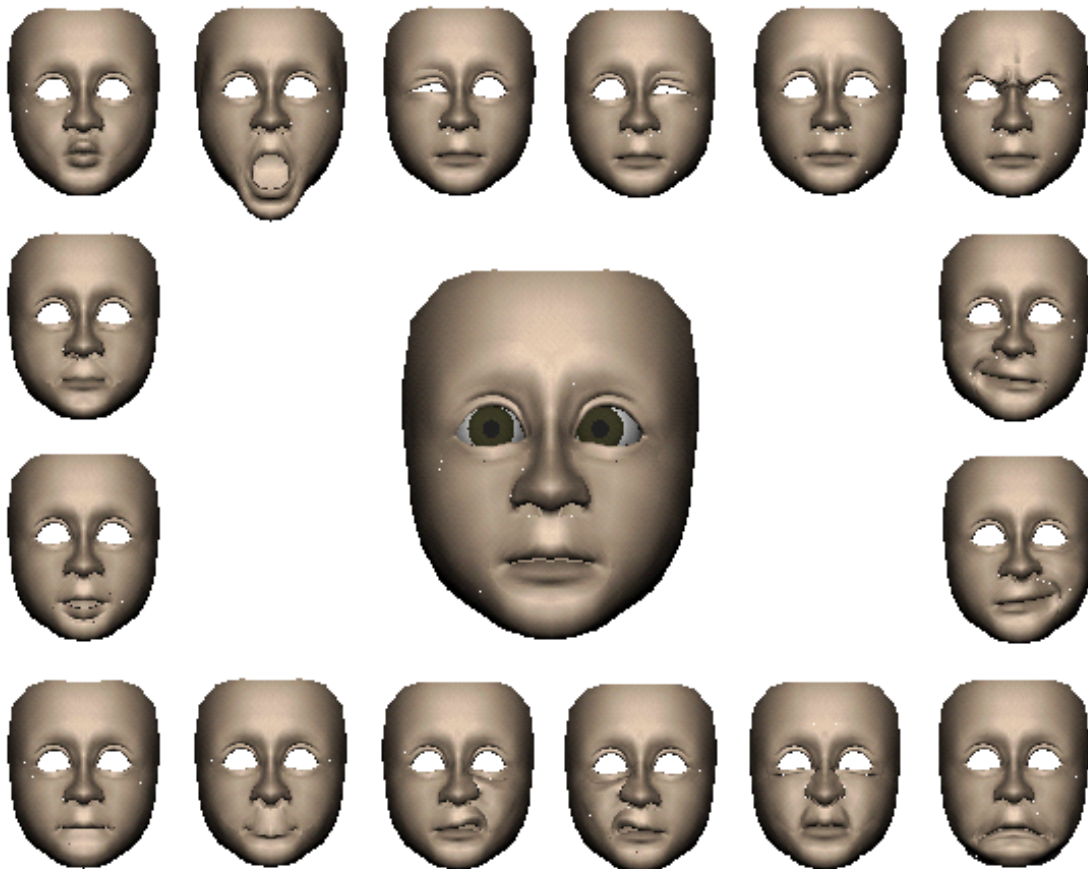


FIGURE 2.4 – Exemple d’un visage paramétrisé par *blendshape*. Les expressions du pourtour de l’image représentent la base d’expressions créée pour le visage. L’expression au centre de l’image est obtenue par pondération des différentes formes de base.

La qualité d’un modèle animé par *blendshape* est très fortement liée à la qualité et à la quantité des expressions de base fournies. Étant donné qu’elles définissent le domaine des déformations réalisables, elles doivent être créées de sorte que chaque expression désirée puisse être obtenue par mélange des expressions initiales. Ce nombre de formes clés peut être très important et le travail nécessaire à leur création conséquent. En effet, générer 10, 50, 100 (ou plus) expressions nécessite de nombreuses heures de travail pour un graphiste confirmé, et ce, uniquement pour rendre le modèle déformable. Cependant, avec suffisamment de temps, l’artiste peut

2. <http://www.autodesk.com/maya>

3. <http://www.autodesk.com/3dsmax>

4. <http://www.blender.org/>

obtenir des modèles contrôlés de très grande qualité, et dans certains domaines, comme celui du cinéma, il s'agit du critère principal. C'est pourquoi de nombreuses productions cinématographiques ont utilisé cette famille de contrôle pour leurs personnages virtuels. Pour minimiser ces efforts de paramétrisation du modèle, certains articles proposent des outils pour faciliter la création des formes de base ou les optimiser [CK01, PKC⁺03, LD08, LWP10]. Les techniques de *blendshape* ont récemment démontré leurs qualités dans des projets tels que *Digital Emily Project* (2009)⁵ [ARL⁺09b, ARL⁺09a] en association avec *Image Metrics*⁶, ou des films tels que *Lord of the rings* (2002), *King Kong* (2005) [Sag06], *Avatar* (2010) [LA10] ou *The Curious Case of Benjamin Button* (2008) dont l'animation faciale est produite par *Weta Digital*⁷ pour les trois premiers et par *Digital Domain*⁸ pour le dernier.

Outre la quantité de travail nécessaire, la technique standard des *blendshape* présente quelques inconvénients. Le premier est lié au fait que les formes sont mélangées dans leur intégralité, ainsi il n'est pas possible d'agir localement sur certaines zones. Prenons l'exemple du clin d'œil, il n'est pas possible de l'obtenir en utilisant une forme de base avec les yeux fermés, il faut avoir défini une déformation avec un seul œil fermé. Pour résoudre ce problème, certains auteurs ont proposé de découper le visage en zones indépendantes [JTDP03, PCNS05, LD08]. Ainsi, la pondération ne se fait plus globalement mais localement. Un autre inconvénient assez proche concerne les interférences entre plusieurs formes, la non-orthogonalité de l'espace défini par les formes de base peut empêcher l'animateur d'obtenir la déformation désirée. La Figure 2.5 est une illustration de ce problème présentée par Lewis *et al.* qui ont proposé une méthode permettant de supprimer cet effet d'interférence [LMDN05]. Ils souhaitent imiter une expression de Jack Nicholson, qui consiste à relever un sourcil, tout en conservant les yeux partiellement fermés. La forme du sourcil est facilement obtenue, mais elle implique une ouverture de l'œil non désirée. Pour remédier à ce problème et rendre le système moins rigide, des sommets du maillage sont sélectionnés par l'utilisateur, et le système minimise leurs déplacements lorsque des poids de *blendshape* sont modifiés en agissant automatiquement sur le poids d'autres *blendshape*. Il s'agit donc de minimiser l'erreur entre les sommets contraints et la déformation désirée qui risque de les déplacer.

Une autre difficulté essentielle dans l'animation d'un visage par *blendshape* réside dans le fait que pondérer chaque forme de base manuellement peut s'avérer être

5. <http://gl.ict.usc.edu/Research/DigitalEmily/>

6. <http://www.image-metrics.com/>

7. <http://www.wetafx.co.nz/>

8. <http://www.digitaldomain.com/>



FIGURE 2.5 – A gauche l'utilisateur définit l'expression initiale pour les yeux. Au centre, l'utilisateur cherche à rehausser le sourcil droit, cependant, une interférence avec l'œil droit intervient et celui-ci est ouvert contre la volonté de l'utilisateur. A droite, l'expression corrigée par les contraintes définies à l'aide de la méthode de Lewis *et al.* [LMDN05].

un travail extrêmement fastidieux et peu intuitif. C'est pourquoi une grande partie des travaux de recherche focalisés sur les *blendshape* se sont orientés sur le calcul automatique des poids à partir de données issues de capture de mouvement [KPL98, CB02, PKC⁺03, DCFN06, LMX⁺08, LD08, LWP10]. Étant donné que ces techniques répondent plus à la question « comment le visage est animé ? », qu'à la question « comment préparer le visage pour qu'il soit déformable ? », ces techniques sont détaillées dans la Section 2.5 consacrée à l'animation.

2.4.2 Modélisation physique

Pour obtenir des résultats réalistes et physiquement corrects, la recherche s'est également penchée sur des modélisations que nous nommerons « physiques » pour représenter le visage humain et ses déformations. Ces paramétrisations visent à simuler les mécanismes de déformations tels qu'ils existent réellement sur le visage humain. Cette partie aurait pu être associée à la paramétrisation par déformation de surface et être incluse dans la Section 2.4.3, mais étant donné les différences notables entre un système visant à produire des déformations visuellement réalistes et un autre visant à modéliser la biomécanique du visage, il nous a semblé plus indiqué de différencier clairement ces deux familles. Ces modélisations portent sur deux aspects : la modélisation de la dynamique de la peau, et la modélisation des muscles et de leurs actions, ces deux modélisations sont très fortement liées. Elles peuvent être définies par des systèmes masses-ressorts, des systèmes vectoriels, ou à partir d'une représentation par éléments finis.

Le système masse-ressort proposé en 1981 par Platt et Badler est le premier qui s'intéresse à la modélisation physique du visage [PB81]. La peau est représentée par un maillage de ressorts sur lesquels les muscles agissent. Terzopoulos et Waters ont

présenté une modélisation sur plusieurs couches de ressorts de la peau (la surface cutanée, la couche graisseuse et les muscles) [TW90]. Plus récemment, d'autres travaux se sont basés sur des modélisations similaires [ZPS01, LTW95, KHS01] de la peau. Waters a défini une modélisation devenue populaire des muscles à base de vecteurs [Wat87, WF94, BLK01, BHN03]. Chaque vecteur est défini par un point d'origine, une direction du champ vectoriel et un point d'insertion au niveau de la peau (le maillage). Son système basé sur le *FACS* permet de définir une expression indépendamment de la morphologie du visage. Le court métrage *Tin Toy* (Pixar) met en scène Billy, un bébé, dont le visage est animé en utilisant 47 muscles inspirés de la méthode de Waters. Tang *et al.* ont proposé une modélisation des muscles par des *NURBS* dont la pondération des différents points de contrôle permet de modifier leur forme, et donc la surface du visage [TLY04a]. Contrairement aux méthodes à base de déformation de surface, les *NURBS* servent ici à définir les muscles.

Kähler *et al.* ont proposé un modèle à trois couches [KHS01, KHYS02, KHS03] :

- un crâne rigide ;
- une couche de muscles simulée par des ensembles de fibres et qui ont la possibilité de se contracter de façon linéaire ou circulaire ;
- la peau représentée par un maillage triangulaire.

Ces 3 couches (os, muscles, peau) sont reliées entre elles par un système masse-ressort (voir la Figure 2.6).

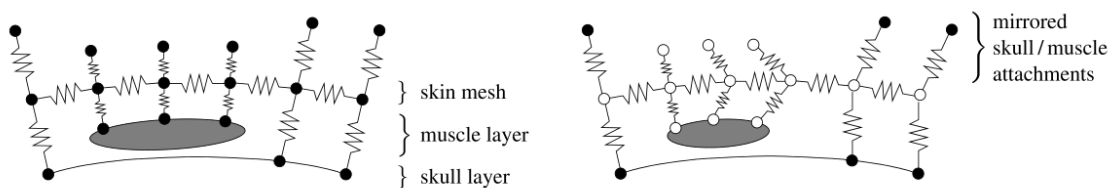


FIGURE 2.6 – Modélisation physique à base d'un système masse-ressort à 3 couches proposée par Kähler *et al.* [KHS01].

Lee *et al.* ont développé une technique pour adapter un visage pré-paramétrisé à de nouveaux personnages [LTW95]. Le visage référence est paramétrisé par un système masse-ressort à trois couches. D'autres ont remplacé la modélisation masse-ressort de la peau par une modélisation basée sur les éléments finis [BLK01, SNF05, SSRMF06]. Ces méthodes offrent d'excellents résultats, cependant, les temps de calcul nécessaires sont très importants, de l'ordre de 8 minutes par frame [SNF05]. La modélisation proposée dans [SNF05] est présentée sur la Figure 2.7.

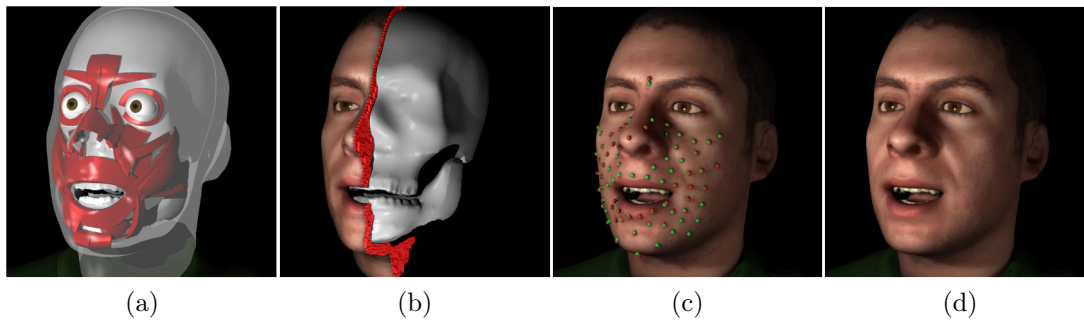


FIGURE 2.7 – Le modèle de Sifakis *et al.* utilise 32 muscles (a) qui déforment un maillage tétraédrique d'éléments finis (b). Le positionnement des points de contrôle lors d'une expression permettent de définir automatiquement l'activité des muscles avec, en rouge, la position recherchée des marqueurs, et en vert, la position obtenue par simulation ((c) et (d)). [SNF05]

2.4.3 Déformation de surface

La famille de paramétrisations par déformation de surface consiste à définir des paramètres directement sur la surface de l'objet, dont les modifications entraînent des déformations géométriques du visage. Tout en prenant en compte la structure physique sous-jacente du visage, ces techniques se basent principalement sur l'observation pour définir ces paramètres, contrairement à la modélisation physique décrite dans la Section 2.4.2. L'idée n'est pas de reproduire la mécanique du visage, mais de reproduire ses effets visuels.

2.4.3.1 Déformation directe

La première paramétrisation proposée dans la littérature par Parke permet d'agir sur certaines parties du visage indépendamment [Par74b, Par74a, Par82], comme le degré d'ouverture des paupières, la position des sourcils, etc. Cohen *et al.* ont ensuite proposé une amélioration de cette paramétrisation dans le cadre d'un système de synthèse vocale [CM93]. Le système d'animation faciale proposé par Valve Software⁹ et utilisé dans de nombreuses productions de jeux vidéo consiste en une paramétrisation directe du visage par les *AU* du *FACS* (*Faceposer*¹⁰). Un paramètre consiste en une interpolation entre deux positions d'un ensemble de sommets. Le fait de travailler sur un ensemble de paramètres offre des avantages non négligeables. Tout d'abord, il est beaucoup plus aisé d'animer un visage en manipulant un nombre limité de paramètres plutôt qu'une géométrie complète, le gain de temps pour l'animateur est conséquent. Ensuite, les paramètres offrent la possibilité de travailler

9. <http://www.valvesoftware.com/>

10. <http://developer.valvesoftware.com/wiki/Faceposer>

localement et d’agir sur différentes parties du visage indépendamment. Enfin, en terme de quantité de données, les valeurs peuvent se situer entre une dizaine ou une centaine de paramètres, alors que sans paramétrisation il est nécessaire de stocker la géométrie complète du visage pour chaque frame ou frame clé de l’animation.

2.4.3.2 Déformation de forme libres

Pour générer les déformations faciales, il est possible d’utiliser les déformations libres (*Free Form Deformation, FFD*) [SP86] ou ses extensions *EFFD* (*Extended FFD*) [Coq90], *RFFD* (*Rational FFD*) [KMMTT92] disponibles dans les logiciels de modélisation 3D comme *Maya*, *3ds Max* ou *Blender* et qui ont été développées initialement pour la déformation de surfaces géométriques. Leur principe est d’envelopper un objet ou une surface dans un espace et de déformer cet espace. Les déformations réalisées sur l’espace sont alors appliquées à l’objet enveloppé. La déformation ne se fait pas directement sur l’objet, mais par le déplacement de points de contrôle positionnés à intervalle régulier sur ou dans l’espace défini. Les *EFFD* permettent d’avoir des matrices de points de contrôle de formes arbitraires, et les *RFFD* permettent de pondérer les points de contrôle pour leur donner plus ou moins d’importance. De nombreuses variantes ont été proposées, pour plus de détails, un état de l’art consacré à ce sujet peut être trouvé dans [Bec98]. Les surfaces à base de splines ont également été utilisées dans le cadre de la paramétrisation d’un visage virtuel [NHS88, WF94, HFG94]. Par rapport à des modèles à base de polygones, elles offrent l’avantage d’offrir une continuité jusqu’à C^2 , et surtout d’animer la surface en ne déplaçant qu’un minimum de points.

2.4.3.3 Déformation à base de points de contrôle

Pour contrôler un visage virtuel identique au visage réel d’un acteur et pour l’animer par capture de mouvement, Guenter *et al.* utilisent une paramétrisation basée sur des points de contrôle, mis en correspondance avec les marqueurs positionnés sur le visage de l’acteur [GGW⁺98]. Chaque sommet est influencé par une pondération des points de contrôle de son voisinage et se déplace de façon linéaire en fonction du déplacement des points de contrôle qui l’influencent. Une étape intermédiaire transforme préalablement les 180 points de contrôle initiaux en une grille de 1400 points régulièrement positionnés et un étiquetage manuel est effectué sur les points appartenant aux zones susceptibles de conflits que sont les contours de la bouche et des yeux. Ce problème est d’ailleurs récurrent et nécessite souvent un traitement particulier. Helzle *et al.* utilisent une paramétrisation similaire du visage [HBSL04]. Les *RBF* ont été utilisées par Noh *et al.* pour déplacer les sommets en fonction du déplacement d’un point de contrôle dans un espace restreint par des points d’an-

crage. Plus récemment et toujours à partir de points de contrôle, la déformation large du visage (en opposition aux déformations fines que sont les rides et autres bosses) proposée par Bickel *et al.* se fait en utilisant un modèle linéaire [BBA⁺07, BLB⁺08]. Le déplacement des sommets est obtenu en minimisant les contraintes d'étirement et de torsion de la peau à partir des déplacements des points de contrôle. Bien qu'efficace en terme de réalisme, cette déformation est trop coûteuse en terme de calculs pour être utilisée en temps réel. Les déformations à base de points de contrôle diffèrent des déformations de formes libres par le fait que les points sont à la surface du visage et ne sont plus utilisés pour définir un espace cubique déformable. Les applications interactives optent régulièrement pour la paramétrisation à base de points de contrôle et l'implémentent sous la forme d'une hiérarchie d'articulations décrite dans la section suivante.

2.4.3.4 Animation par squelette (*Skinning*)

L'animation d'un personnage 3D par squelette, également appelée *skinning*, *Skeleton Subspace Deformation (SSD)* ou bien *Linear Blend Skinning (LBS)*, est une technique très répandue pour déformer le maillage d'un personnage ou d'un objet articulé. Le principe consiste à animer un squelette rigide, auquel est lié le maillage du personnage ou de l'objet (voir la Figure 2.8). Pour réaliser ceci, le *skinning* décrit des associations entre les sommets du maillage et les os (*bones*) du squelette, on parle de l'influence d'un os sur un sommet. Lorsque le squelette est animé, les sommets sont déplacés en fonction des os qui les influencent. La technique n'a jamais été directement publiée mais on la retrouve cependant décrite dans certains articles [MTLT88]. Des évolutions ont été proposées pour contrer certains problèmes d'artefacts qui peuvent survenir aux niveau des articulations [LCF00, KJP02, MG03, JZP⁺08].

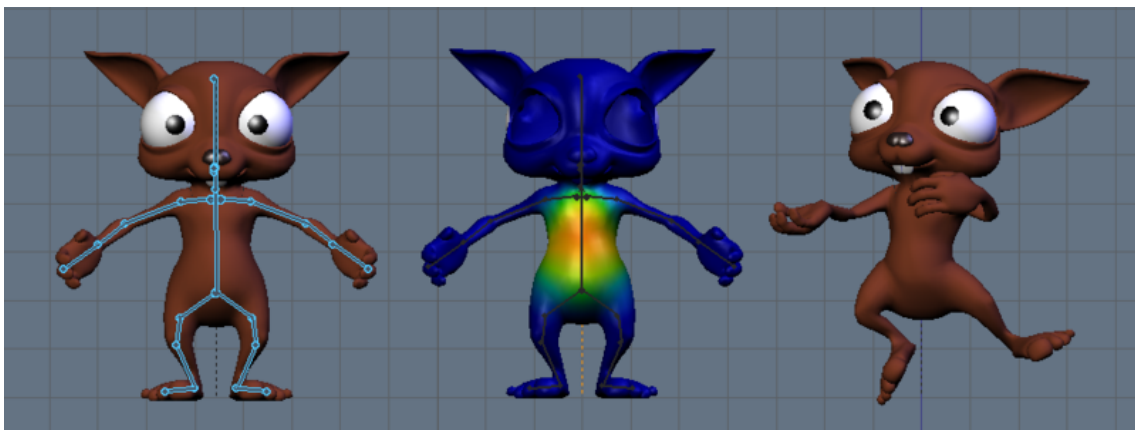


FIGURE 2.8 – Exemple de personnage animé par *skinning*. A gauche : le modèle 3D avec le squelette adapté dans une position de repos. Au milieu : l'influence de l'os de la colonne vertébrale sur la surface du personnage.

La position \vec{v} d'un sommet v d'un objet articulé est lié aux transformations rigides \mathcal{M} appliquées aux articulations du squelette :

$$\vec{v} = \sum_{k=1}^n w_k \mathcal{M}_k(v) v \quad (2.3)$$

avec $M_k(v) = M_k^\delta M_k^{0^{-1}} M_v^0$. M_k^0 est la transformation allant du système de coordonnées de l'os k au repos au système de coordonnées du monde, M_k^δ est la transformation allant du système de coordonnées de l'os k animé au système de coordonnées du monde, et M_v^0 est la transformation allant du système de coordonnées de la surface contenant le sommet v au système de coordonnées du monde. Soulignons que $M_k^{0^{-1}} M_v^0$ est donc la transformation de v dans le repère de l'os k au repos. Soit n le nombre d'os et w_k le poids de l'os k associé au sommet v . Si on note par $W = \{w_0, \dots, w_i, \dots, w_{n-1}\}$ où w_k représente le poids de M_k avec $\sum w_i = 1$ et $w_i \in [0, 1]$, W est un ensemble convexe. Dans le cadre de l'animation de personnage, les transformations appliquées aux os sont des rotations, et peuvent être décrites dans des matrices de taille 3×3 , 3×4 ou 4×4 , des quaternions [Hej04] ou des dual-quaternions [KCZO07]. Le *skinning* a rencontré un succès très important dans l'animation en temps réel grâce à sa facilité d'implémentation, et notamment la possibilité d'effectuer les calculs aisément sur le processeur graphique (*Graphics Processing Unit*, *GPU*).

L'animation du visage à base de points de contrôle peut être utilisée dans le cadre d'une implémentation par *skinning* bien que la majeure partie des déformations du visage soient plus adaptées aux translations qu'aux rotations (excepté pour la mâchoire qui est l'unique articulation présente sur le visage et les yeux qui ont un mouvement de rotation uniquement). En effet, la translation nécessaire à l'animation des points de contrôle peut être insérée dans les matrices de transformation M_k de l'Équation 2.3. L'animation d'un visage par points de contrôle offre donc un réel avantage au niveau de l'implémentation, puisqu'il est possible de n'utiliser qu'un système d'animation pour le personnage entier (corps et visage). De plus, l'édition d'une expression est beaucoup plus intuitive lorsqu'il s'agit de travailler directement sur les points de contrôle, contrairement aux *blendshape* qui nécessitent l'utilisation de sliders et les risques de conflits entre les différentes expressions produisant des artefacts. La Figure 2.9 présente un visage dont la paramétrisation à base de points de contrôle est utilisée dans une implémentation classique d'animation par squelette.

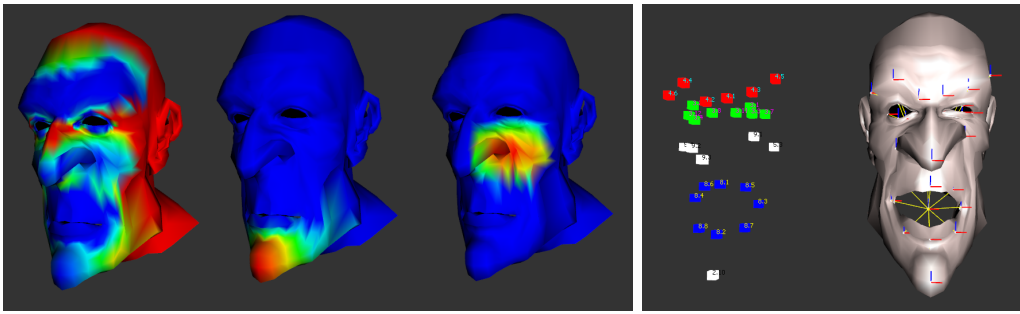


FIGURE 2.9 – Visage paramétrisé par déformation à base de points de contrôle implémentée sur un système d’animation par *skinning*.

2.4.4 Paramétrisation automatique ou assistée

Paramétriser un visage nécessite un travail, très souvent manuel, important. Définir un ensemble de paramètres offrant la possibilité de déformer le visage selon ses très nombreux degrés de liberté, ou au moins selon les degrés de liberté nécessaires à l’application est une tâche ardue. Par conséquent, certains chercheurs se sont intéressés à l’automatisation de cette tâche. Zhang *et al.* proposent une méthode pour générer facilement la paramétrisation par modélisation physique d’un nouveau visage dont la géométrie est obtenue par le moyen d’un scanner [ZST04]. Un modèle générique est utilisé et sa paramétrisation est adaptée morphologiquement au nouveau visage grâce au positionnement manuel de quelques points de contrôle. De façon similaire, Kähler *et al.* utilisent un modèle générique paramétrisé pour l’adapter à un visage scanné dont les données sont incomplètes [KHYS02]. La méthode de transfert proposée par Orvalho *et al.* permet d’adapter différents types d’attributs d’une paramétrisation et notamment la modélisation de muscles sur un nouveau personnage [OZS06]. La différence majeure par rapport aux deux autres techniques présentées réside dans le fait que la topologie du visage à paramétriser est conservée. Ceci offre un avantage important quant à la flexibilité du système : le modèle générique ne contraint plus un type de maillage prédéfini. Ce progrès permet donc de paramétriser des personnages créés pour divers types de plateformes ou d’applications. Cependant, la mise en correspondance entre les deux modèles, souvent réalisée manuellement, ne permet pas d’automatiser le processus. Nous proposons une méthode pour résoudre ce problème dans le Chapitre 3.

2.5 Animation

Comme il a été souligné auparavant, nous avons fait le choix de distinguer la paramétrisation qui définit les paramètres associés au visage, de l’animation qui modifie ces paramètres pour déformer la géométrie du visage au cours du temps. L’animation

constitue un domaine qui a suscité autant d'intérêt auprès des chercheurs ces trente dernières années que celui de la paramétrisation, et souvent, ces deux sujets sont abordés ensembles. Les méthodes les plus simples et les plus naturelles pour animer un visage consistent à faire varier les paramètres de contrôle manuellement au cours du temps pour produire l'animation, en allant de l'interpolation entre expressions jusqu'à l'utilisation de courbes d'animation pour chaque paramètre (la courbe définit la valeur du paramètre au cours du temps). Constatant les problèmes liés à la gestion manuelle des paramètres de contrôle, Williams a introduit le concept de capture de mouvement pour le visage [Wil90] (*Performance-Driven Facial Animation, PDFA*). L'idée est d'utiliser l'animation d'un visage réel pour l'appliquer à un visage virtuel. Certains travaux se sont également orientés vers la synthèse d'animation, notamment pour l'animation de la bouche en fonction de la parole. Et enfin, le transfert d'animation, c'est-à-dire, la réutilisation d'une animation existante sur un nouveau visage, suscite également l'intérêt de la recherche.

2.5.1 Création manuelle d'animation et interpolation

La première approche pour générer une animation est très fortement basée sur l'aspect artistique et le travail manuel fournit par un animateur 3D. Elle consiste à créer des expressions clés (ou frames clés) et d'effectuer une interpolation linéaire entre elles pour déformer de façon continue le visage au court du temps [Par72, BL85, PHL⁺98]. Selon la méthode de paramétrisation choisie, la génération d'une expression nécessite un travail plus ou moins important. Il s'agit de définir la position de chaque sommet « à la main » pour un modèle non paramétrisé ; de positionner les points de contrôle pour les techniques de paramétrisation par déformation de surface ; de déterminer la force appliquée par chaque muscle pour les méthodes de simulation ; de pondérer les formes pour les modèles à base de *blendshape* ; etc. Cette méthode apporte quelques inconvénients. Tout d'abord, même avec un contrôle du visage évolué, permettant de définir toutes les expressions possibles, synthétiser une animation réaliste reste extrêmement compliqué à cause de la dynamique complexe du visage qu'il est difficile de reproduire par une succession d'expressions clés et d'interpolations. Pour résoudre ces problèmes, certains ont utilisé des interpolations bilinéaires [Par74b], ou à base de fonctions cosinus [WL93], etc. Aujourd'hui, la valeur de ces paramètres peut être décrite à l'aide de courbes d'animation, indépendantes les unes des autres, et qui facilitent la création d'animations fluides. La gestion de la vitesse et de l'accélération des différentes parties du visage sont ainsi plus facilement contrôlables.

Le nombre important de degrés de liberté pour définir une seule expression ajoute une difficulté supplémentaire à la génération d'animation « à la main ». Récemment, des méthodes spécifiques pour la création d'expression ont été proposées [CJ06, LCXS07, NF07]. Elles ont pour but de faciliter la création de poses en offrant des outils intuitifs pour l'édition d'expression. Stoiber *et al.* ont proposé une interface 2D pour générer les expressions d'un visage à partir d'un espace de dimension réduite des expressions possibles obtenues par le biais d'un visage réel [SSB09, SBS10]. De plus, leur méthode permet d'obtenir une dynamique réaliste entre plusieurs expressions grâce à la nature continue de leur représentation, cependant, seules les émotions du visage sont contrôlables et ce système ne permet pas de produire les visèmes associés à la parole. Certaines méthodes, dites de *sketching*, consistent à déformer un visage à partir de dessins réalisés en 2D. Natanelli *et al.* utilisent les machines à vecteurs de support (*Support Vector Machine, SVM*) pour effectuer un apprentissage qui permet ensuite de définir des paramètres de déformation à partir de lignes dessinées [NF07]. D'autres méthodes, dites de *posing*, consistent quant à elles à déformer un maillage en travaillant directement sur sa surface à l'aide de traits, de courbes, de formes ou de points [NSACO05, CJ06, LCXS07]. La génération d'expressions peut alors être comparée à de la sculpture numérique. L'idée générale consiste à minimiser l'énergie nécessaire pour déformer une forme initiale dessinée par l'artiste à la surface du visage, vers une forme désirée (voir la Figure 2.10).

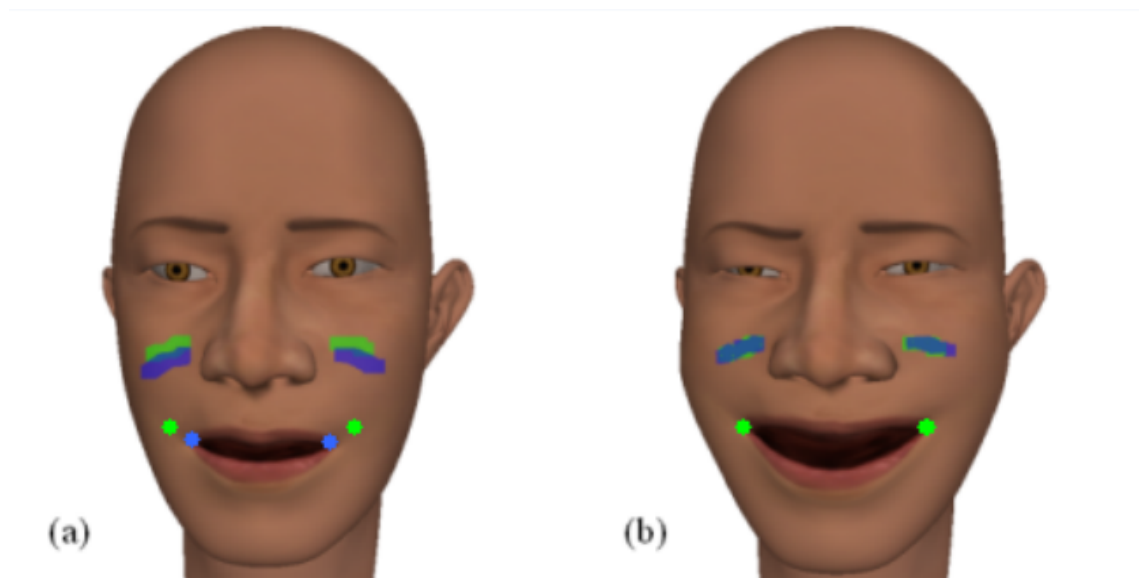


FIGURE 2.10 – Exemple de génération d'expression à partir de traits et de points [LCXS07]. À gauche le maillage original, l'utilisateur définit des contraintes à base de formes sources (en bleu) et de formes cibles (en vert). À droite le maillage déformé en respectant les contraintes de l'utilisateur et celles obtenues à partir du modèle (base de données d'expressions).

2.5.2 Capture de mouvement

Les méthodes de capture de mouvement consistent à recueillir des données d'animation du visage à partir d'une personne réelle pour les utiliser sur un visage synthétique. Travailler sur des données réelles d'animation permet de limiter les problèmes liés à la création manuelle et orientée artiste. La fluidité et la dynamique des déformations sont alors mieux respectées visuellement. Les systèmes de capture reposent sur des bases communes qui consistent à détecter et/ou suivre des points de contrôle ou des contours dans les images d'une vidéo, issues d'une ou plusieurs caméras. Une fois ces données obtenues, il faut les corriger si nécessaire, puis les transformer en données utilisables par un système de paramétrisation (par exemple, déterminer les forces appliquées par les muscles d'un visage contrôlé par simulation à partir des mouvements de points de contrôle capturés).

2.5.2.1 Techniques de capture

Suivi de marqueurs

L'un des moyens le plus utilisé pour réaliser une animation faciale pilotée par une personne réelle (*PFDA*) consiste à positionner des marqueurs sur le visage de l'acteur et de les suivre pendant l'animation. Ces marqueurs peuvent être des billes réfléchissantes collées sur le visage (voir la Figure 2.11) ou bien des points dessinés sur la peau (voir la Figure 2.13). Les premiers travaux visant à reproduire l'animation d'un visage réel travaillent sur des images statiques et font varier les coordonnées textures 2D au cours du temps pour reproduire l'animation [Wil90]. Guenter *et al.* ont présenté plus tard un système beaucoup plus évolué à partir de 6 caméras et de 182 marqueurs apposés sur le visage de l'acteur [GGW⁺98]. La difficulté majeure réside dans la classification de chacun des marqueurs sur les images des six caméras pour effectuer les mises en correspondance et déterminer leurs positions 3D à chaque instant de l'animation. Le visage virtuel, construit à l'identique de celui de l'acteur (texture et géométrie), est ensuite animé par une technique à base de points de contrôle. Les techniques de flots optiques sont régulièrement utilisées pour le suivi des marqueurs ou même de chacun des pixels du visage [EDP94, EBDP96, BPL⁺03].

Suivi de points caractéristiques sans marqueur

Le suivi de points caractéristiques a également été abordé avec ou sans les contours. Un point caractéristique définit un point naturel du visage en opposition aux marqueurs ajoutés dans le cadre de l'acquisition. Les travaux de Pandzic *et al.* permettent par exemple d'extraire en temps réel des paramètres de l'animation faciale comme l'ouverture des yeux, le déplacement vertical des sourcils, etc. Ces paramètres sont ensuite appliqués sur un visage virtuel en utilisant le système de codage *MPA*



FIGURE 2.11 – Capture des mouvement du visage par suivi de marqueurs utilisée pour le film *Pirates des Caraïbes 2* ©Industrial Light & Magic/Walt Disney Pictures.

[PKMTT94, PCMTT96]. L'utilisateur, après avoir indiqué manuellement la position de ces points pour une première expression neutre, peut intervenir à tout moment pour corriger les erreurs de capture. De façon similaire, Bourel *et al.* déterminent des paramètres d'animation du visage en associant au suivi des points des heuristiques et des connaissances préalables sur les déformations d'un visage humain [BCL00]. Les méthodes de capture sans marqueur sont cependant limitées à des captures peu précises (basse résolution, bruit, erreurs). Chai *et al.* proposent une solution à ce problème en associant au système de suivi de points caractéristiques une base de données de l'espace du possible des déformations du visage [CXH03]. Cependant, cette base de donnée est obtenue grâce à un système de capture avec marqueurs et limite le suivi de points caractéristiques sans marqueur à une seconde étape de l'acquisition.

Modèles déformables

Bien que très répandu, le suivi de points n'est pas toujours utilisé. Les modèles déformables sont également beaucoup employés pour la capture d'animation. Les *snakes* déforment des courbes pour suivre les contours du visage et de ses organes comme les yeux ou la bouche. La déformation des courbes est réitérée par minimisation d'énergie jusqu'à stabilisation sur les zones de contours [KWT88]. Cootes *et al.* ont développé de façon similaire les modèles de forme active (*Active Shape Models, ASM*) qui s'appuient sur une modélisation statistique des contours [CTCG95]. Ces modèles ont été étendus aux modèles d'apparence active (*Active Appearance Models, AAM*) par

la prise en compte des informations de couleur au modèle [CET98, CET99, CT01]. Les *AAM* ont également été améliorés pour une acquisition basée sur plusieurs caméras [SSSB09], pour réduire les problèmes liés aux variations d'éclairage [SGBG09] ou bien pour répondre au mieux aux différences de morphologie entre le modèle statistique créé et le visage réel à suivre en décomposant le visage en zones distinctes [ZC05].

2.5.2.2 Capture photoréaliste

Pour obtenir des visages de plus en plus réalistes, les systèmes de capture de mouvement sont devenus de plus en plus complexes. Car pour obtenir un rendu photoréaliste, il ne suffit pas que les déformations soient réalistes, il faut également que les textures et les paramètres d'illumination du visage soient d'une grande précision. Pour cela, des systèmes très évolués d'acquisition ont été mis au point. Hawkins *et al.* utilisent un système rotatif doté d'un arc vertical semi-circulaire sur lequel 30 lumières stroboscopiques sont fixées et de 6 caméras haute-définition dans le but d'obtenir 480 images de la tête, correspondant à 480 directions d'éclairage [HCD01, HWT+04]. L'acquisition d'une seule expression nécessite 8 secondes (1 tour complet du bras) pendant lesquelles l'acteur doit rester immobile, et ceci est effectué pour 60 expressions. Une autre technique développée spécialement pour le film *Matrix Reloaded* a nécessité l'utilisation de visages virtuels pour mettre en scène les figures acrobatiques réalisées non pas par les acteurs Keanu Reeves et Hugo Weaving, mais par leurs avatars virtuels [BPL+03, BL03]. Une première acquisition haute-résolution des visages est effectuée à l'aide d'un scanner, puis l'animation est obtenue à l'aide de 6 caméras filmant les performances des acteurs. Les images produites par ces caméras permettent également de créer des textures qui varient au cours du temps pour reproduire au mieux la dynamique visuelle du visage.

Les résultats les plus aboutis à ce jour ont été obtenus dans le cadre du *Digital Emily Project*¹¹ [ARL+09b, ARL+09a] (voir la Figure 2.12). L'actrice est placée au centre d'une sphère sur laquelle sont positionnées 156 LED blanches et deux caméras en stéréo qui permettent d'enregistrer 17 clichés avec des directions d'éclairage différentes (pour capturer en haute-définition les composantes diffuses et spéculaires de la peau), mais aussi des lumières structurées permettant d'obtenir en plus des données géométriques très précises. À partir de ces données, le système de paramétrisation du visage à base de *blendshape* est créé, suivi de l'animation et du rendu. Le résultat obtenu est de grande qualité, l'animation complète de la tête est incrustée dans la

11. Projet réalisé en association par la société *Image Metrics* (<http://www.image-metrics.com>) spécialisée dans la capture de mouvement du visage et *USC ICT* (*University of Southern California's Institute for Creative Technologies*).

vidéo qui filme l'actrice, et il est presque impossible de dire que le visage présent sur la vidéo est synthétisé. Néanmoins, une telle qualité s'obtient à l'aide d'une quantité de travail et d'efforts considérables. Quelques chiffres sur le temps nécessaire à la réalisation d'une animation de 90 secondes au format *Full HD* (1920x1080 pixels) sont donnés dans [ARL+09a] :

- numérisation (scanner) : 1h30min (3 secondes par scan, 3 techniciens)
- travail sur les données scannées : 10 jours (1 artiste)
- paramétrisation du visage : 3 mois (75 expressions, 1 artiste)
- animation : 2 semaines (2 animateurs)
- rendu : 3 mois (1 artiste)

On est donc encore très loin d'une méthode automatique et en temps réel et de telles techniques ne sont utilisables que pour quelques projets disposant de moyens très importants.

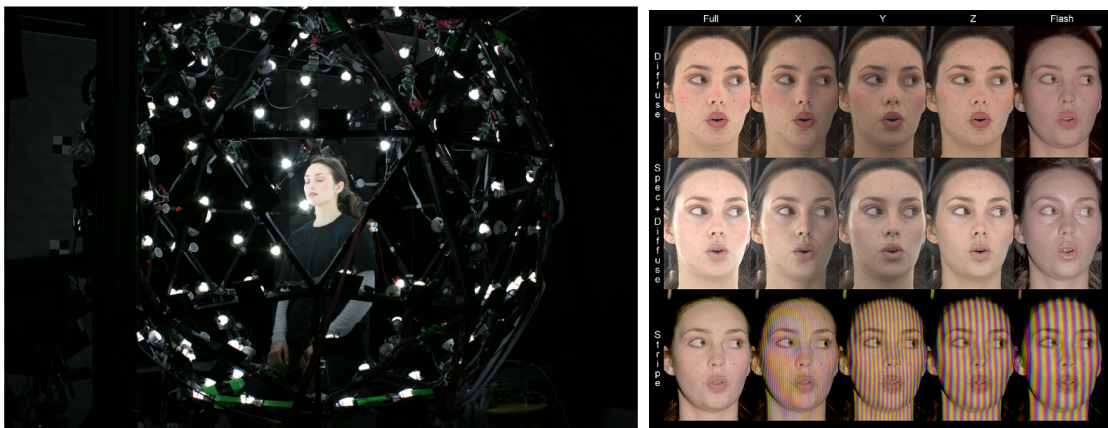


FIGURE 2.12 – Système de capture utilisé pour le *Digital Emily Project* [ARL+09b, ARL+09a]. À gauche, vue d'ensemble du système avec l'actrice à l'intérieur de la sphère sur laquelle sont positionnées 156 lumières et deux caméras. À droite, les 15 images prises par une paire de caméras frontales pour une unique pose. Les images de la ligne supérieure permettent la capture de la composante diffuse du visage. La seconde ligne concerne les composantes diffuse et spéculaire. La dernière permet de capturer la géométrie 3D.

2.5.2.3 Détails fins

Plus récemment, la recherche s'est intéressée à la capture fine de l'animation faciale, *i.e.* en plus de capturer l'animation générale du visage, il s'agit de détecter les rides et autres déformations fines de la peau. Pour ce faire, des techniques de vision telles que les techniques travaillant sur l'auto-ombrage des surfaces [BBA+07, LLZ+08] ou les images ratios [TLY+04b] ont été utilisées. Des systèmes de capture à base de projection de lumières structurées permettent d'obtenir des détails

fins du visage [MVR⁺95, JLP04, NJ04, WHL⁺04, WLGP09] (voir la Figure 2.14). Bickel *et al.* détectent les rides sur les images et les représentent par des *B-Splines* en 2D. La forme de la ride est ensuite estimée grâce aux effets d'éclairage de la surface (voir la Figure 2.13). Beeler *et al.* ont réussi récemment à obtenir des détails d'acquisition aussi précis que ceux obtenus avec des lumières actives (lasers, lumières structurées) grâce à un système de stéréo vision passif qui ne nécessite pas d'éclairage spécifique [BBB⁺10]. Les détails les plus fins sont extraits des images à l'aide d'un filtre passe-haut et reconstruits géométriquement grâce à une nouvelle formulation macroscopique. Bradley *et al.* ont également proposé un système d'acquisition passif permettant de capturer les déformations faciales très précises [BHPS10]. Ce système utilise 9 plaques d'illumination dotées chacune de 192 LED. Pour obtenir un suivi très précis sans l'utilisation de marqueur, des paires de caméras vont chacune filmer en stéréo-vision une petite partie du visage, avec un zoom optimisé pour obtenir la meilleure résolution possible sur chacune de ces parties. 14 caméras haute-définition sont alors utilisées. Ce sont les détails fins de la peau comme les pores qui permettent d'effectuer la mise en correspondance dans le cadre de la reconstruction stéréo (voir la Figure 2.15). Les systèmes d'acquisition présentés dans cette section permettent de capturer une animation faciale détaillée : ils permettent en effet de détecter à la fois les déformations larges du visage ainsi que les déformations fines. Cependant, leur mise en place est longue et difficile, elle nécessite une quantité et une qualité de matériel conséquentes. Nous développons plus en détail dans la Section 2.5.4 l'animation des détails fins qui peut être abordée indépendamment de l'aspect capture.

2.5.2.4 Adaptation des données capturées

Une fois la géométrie, la position des points caractéristiques ou la position des marqueurs obtenue, il faut ensuite adapter ces données au système de paramétrisation d'animation choisi. En effet, il est souvent nécessaire d'effectuer des traitements supplémentaires car les différents types de paramétrisation ne prennent pas ou rarement des données géométriques capturées en entrée. Par exemple, la position d'un point caractéristique détecté automatiquement est inutilisable sur un modèle animé par *blendshape* sans traitement. Certains auteurs ont fait le choix d'utiliser un système de codage intermédiaire pour rendre les données d'animation génériques (c'est-à-dire utilisables par n'importe quel système implémentant ce codage) alors que d'autres se sont concentrés sur un système de contrôle cible spécifique.

Les systèmes de codage comme le *FACS* ont été utilisés pour faire abstraction des données brutes de capture et les réutiliser sur un modèle contrôlé [EBDP96, EP97]. Heltzle *et al.* transforment les déplacements des marqueurs en valeur de *AU*

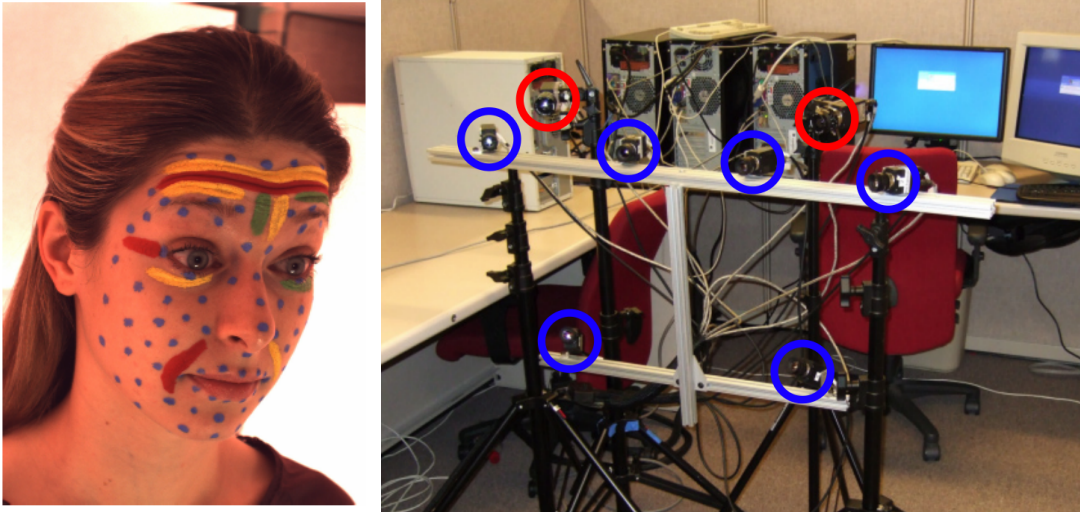


FIGURE 2.13 – Système de capture à 3 échelles proposé par Bickel *et al.* [BBA⁺07, BLB⁺08]. À gauche, visage équipé des marqueurs pour les déformations larges et des zones de rides peintes pour la détection de celles-ci. À droite, le matériel d’acquisition : 6 caméras (entourées de bleu) pour les déformations larges et 2 caméras (entourées en rouge) pour les détails intermédiaires.

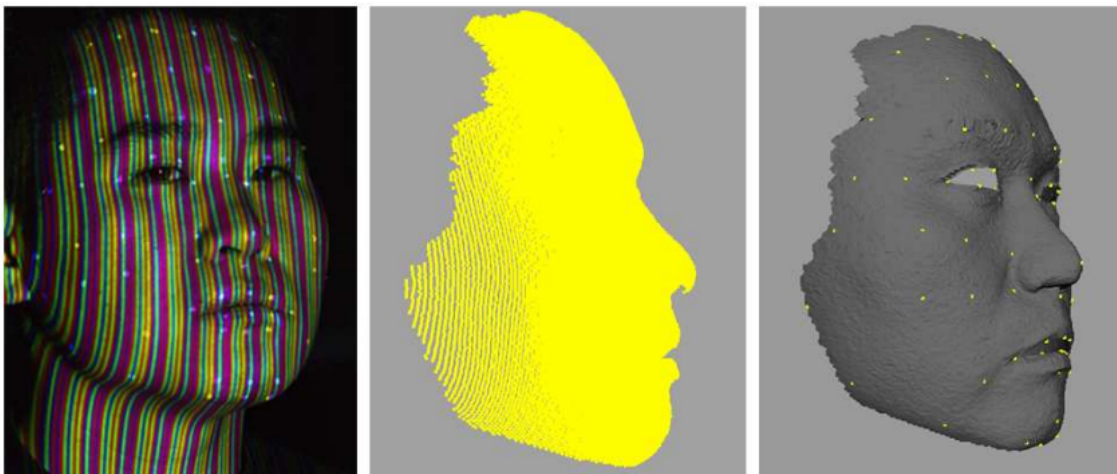


FIGURE 2.14 – Système de capture utilisé par Na *et al.* faisant intervenir marqueurs et lumières structurées [NJ04].

[HBSL04]. Des courbes d’animation non linéaires sont alors créées au cours du temps et réutilisables sur des modèles qui utilisent ce même codage.

Définir une expression pour un modèle paramétrisé par *blendshape* consiste à déterminer les poids de chaque forme ou expression de la base. Pour trouver ces poids à partir de données de *MoCap*, plusieurs méthodes ont été proposées. La plupart des travaux reposent sur la connaissance préalable de certaines correspondances entre les données acquises par caméra et les valeurs des paramètres du modèle à animer.

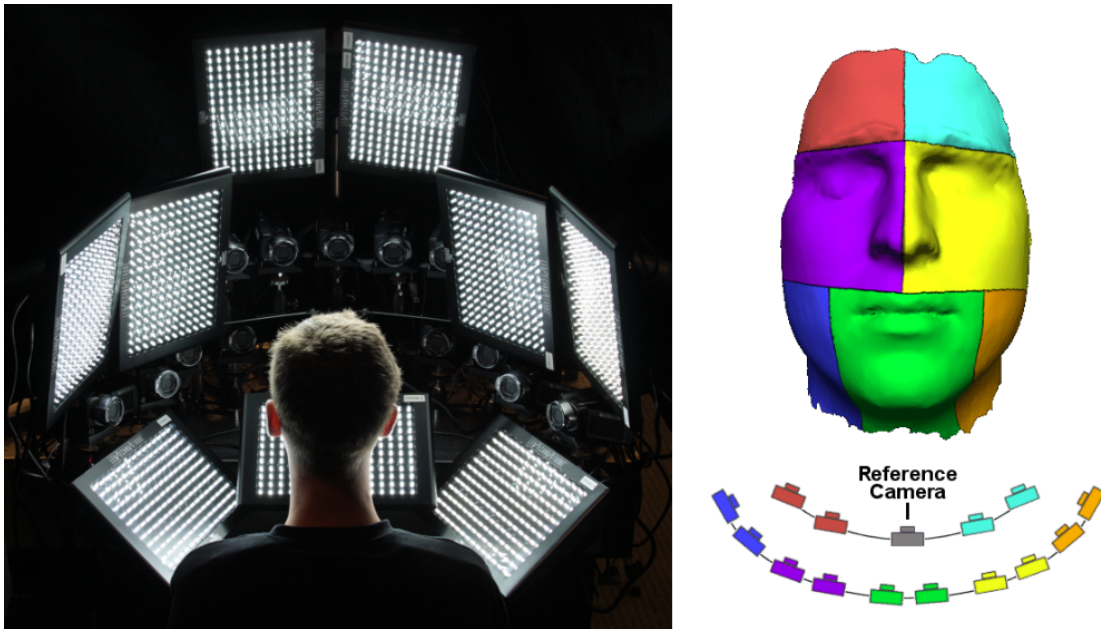


FIGURE 2.15 – Système de capture passif proposé par Bradley *et al.* [BHPS10]. Il utilise 9 plaques d’illumination de 192 LED chacune et 7 paires de caméras dont les zooms ciblent des zones réduites du visage.

Il est ensuite possible de minimiser, avec un solveur basé sur les moindres carrés par exemple, la distance entre la configuration d’une nouvelle pose et la pondération des poses de la base [KPL98]. Certains auteurs ont également eu l’idée de construire le modèle *blendshape* en fonction des données capturées et une sélection de poses clés [CB02], puis d’utiliser un solveur des moindres carrés. Pour éviter des distorsions non souhaitables, l’espace de pondération est contraint à des données positives. Inversement, Lauthelie *et al.* définissent une base d’expressions pour le visage réel à partir des déplacements d’un ensemble de points de contrôle [LN00]. Pendant l’animation, les poids des expressions du visage réel sont calculés pour chaque frame et appliqués au visage virtuel. Deng *et al.* proposent une approche semi-automatique pour trouver les pondérations [DCFN06]. Une fois l’acquisition effectuée, quelques frames de l’animation sont sélectionnées avec soin et un animateur va définir les pondérations souhaitées pour chacune de ces frames. Ces paires références sont ensuite utilisées pour générer une approximation par *RBF*. Chacune des frames de l’animation peut alors être transformée en poids et permet ainsi d’animer le visage virtuel. Les *RBF* ont également été utilisées dans ce contexte par [JTDP03, LMX+08].

Pour animer les modèles contrôlés par des systèmes de simulation physique, l’adaptation des données de capture consiste à les traduire en « forces » produites par les muscles modélisés. Pour cela, la majorité des techniques visent à minimiser la distance entre les points caractéristiques réels du visage du comédien et les marqueurs

virtuels positionnés sur la surface du visage 3D. Choe *et al.* utilisent un solveur des moindres carrés itérativement pour minimiser cette distance [BLK01]. La modélisation physique par éléments finis de Sifakis *et al.* nécessite une résolution du problème plus complexe à base de cinématique inverse et en suivant les contraintes d'une biomécanique non linéaire [SNF05].

2.5.3 Transfert d'animation (*Retargeting*)

S'il est possible maintenant d'obtenir des animations de qualité, le travail nécessaire pour produire une animation réaliste est important. Ainsi, comme pour la paramétrisation d'un visage, des techniques de transfert d'animation ont vu le jour [PL06]. L'idée principale est d'économiser du temps et des ressources en réutilisant l'animation existante pour un visage donné sur un nouveau visage. Les différences de morphologie entre les visages, les modèles dont les traits sont parfois exagérés (caricatures, cartoons, etc.) font que le transfert n'est pas trivial et nécessite des techniques particulières. Le transfert d'animation peut parfois être confondu avec les techniques d'adaptation de données présentés dans la Section 2.5.2.4. Bien que certaines méthodes peuvent s'adapter à ces deux applications, le transfert consiste à réutiliser des données d'animation existantes alors que l'adaptation consiste à adapter des données de *MoCap* sur un visage donné et sa méthode de paramétrisation.

La première technique proposée pour le visage consiste à transférer les vecteurs d'animation d'un maillage à un autre maillage, possédant à la fois une géométrie et une topologie différentes [NN01] (voir la Figure 2.16). Un vecteur d'animation correspond au déplacement d'un sommet au cours de l'animation. Après avoir positionné des points de contrôle sur les deux visages (semi-automatiquement à l'aide d'heuristiques), une mise en correspondance entre les deux maillages est effectuée. La position des points de contrôle permet d'initialiser une transformation par *RBF*, qui est ensuite appliquée à chacun des sommets du maillage source. Celui-ci va approximer la forme du maillage cible, une projection permet de plaquer les sommets cibles déformés sur la surface du maillage source. Une fois la correspondance dense obtenue, les vecteurs d'animation sont adaptés en orientation et en amplitude puis appliqués au maillage cible. Na *et al.* ont étendu cette méthode pour l'appliquer de façon hiérarchique, offrant la possibilité d'effectuer le transfert avec un certain degré de précision [NJ04]. Les détails fins d'animation comme les rides peuvent ainsi être transférées en plus de la déformation globale du visage.

De nombreux travaux sur le transfert se sont concentrés sur les modèles contrôlés par *blendshape*. Bien que le transfert entre deux modèles dont les formes de base

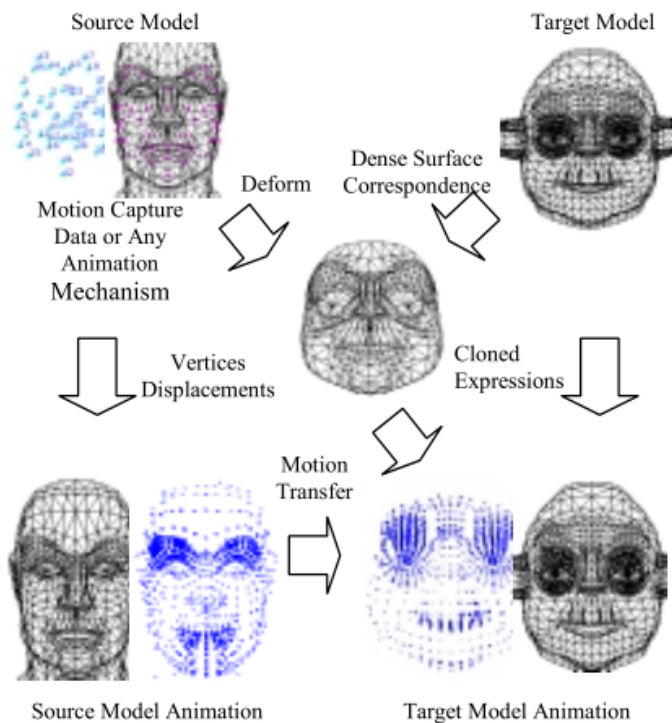


FIGURE 2.16 – Schéma des différentes étapes pour le clonage d’expressions proposé par Noh *et al.* [NN01].

possèdent les mêmes sémantiques soit trivial, puisque il suffit de réutiliser les poids du premier sur le second [PHL⁺98], ce cas s’avère être rare en pratique. Pyun *et al.* utilisent une paramétrisation intermédiaire par points de contrôle pour effectuer le transfert par le biais d’une interpolation de données éparses par *RBF* [PKC⁺03]. Cependant, leur méthode nécessite tout de même de générer des poses sources et cibles similaires. La plupart du temps, les modèles source et cible ne sont pas construits sur une même base d’expressions [PL06], il est alors demandé à un artiste de sculpter les mêmes poses du visage source sur le visage cible. Soulignons que le terme sculpter désigne ici le fait de modifier les poids des formes de base du visage cible et non de recréer la géométrie manuellement. Deng *et al.* utilisent une base de correspondances entre des données issues de capture de mouvement réduites par *ACP* (*Analyse en Composante Principale*) avec les poids du modèle cible sculpté [DCFN06]. De façon similaire à [PKC⁺03], une interpolation par *RBF* permet ensuite de passer de l’espace de paramétrisation du modèle source vers l’espace de paramétrisation du modèle cible. Vlastic *et al.* ont soumis l’idée d’une modélisation statistique multi-linéaire pour contrôler et animer un visage [VBPP05]. Les données décomposées en 3 attributs (identité, expression et visème) permettent de transférer l’animation d’un visage connu de la base (même identité mais avec des expressions et visèmes qui varient au cours du temps) vers un autre visage de la base en modifiant l’attribut

identité.

2.5.4 Détails fins d'animation

L'ajout de détails à une animation faciale, particulièrement dans un contexte de temps réel, est toujours une tâche difficile malgré les efforts fournis dans le domaine de l'animation faciale. Certaines méthodes proposent une modélisation physique pour le vieillissement de la peau et la simulation de rides [WKMT96, BKMTK00, KG03, VLR05]. D'autres recherches se sont focalisées sur l'acquisition de détails à partir de visages réels [ZSCS04, LLZ⁺08, BBA⁺07]. Certains auteurs ont également soumis des techniques de transfert de détails [TLY⁺04b, NJ04, BLB⁺08]. Nous nous concentrons dans cette partie de l'état de l'art sur les travaux qui se sont intéressés aux détails fins d'animation, en opposition aux détails considérés comme statiques.

Volino *et al.* ont présenté une technique pour ajouter des rides sur une surface triangulée animée [VMT99]. Basée sur les variations de taille des arêtes du maillage, l'amplitude des cartes de hauteur appliquées sont modifiées. Pour chaque arête et texture, un coefficient de forme est calculé pour connaître l'influence de l'arête avec la texture donnée. Plus une arête est perpendiculaire aux détails présents sur la texture, plus son élongation ou son raccourcissement perturbent la carte de hauteur. Le rendu est réalisé par lancé de rayons et *bump-mapping*, le maillage est de plus subdivisé automatiquement dans les zones nécessaires. Bando *et al.* ont proposé une méthode pour générer différents types de rides (fines ou larges) sur le corps [BKN02]. Les rides fines sont affichées par *bump-mapping* et obtenues en utilisant un champ de direction défini par l'utilisateur. Les rides larges sont affichées en déplaçant les sommets d'un maillage haute-résolution et obtenues en utilisant des courbes de Bézier dessinées manuellement par l'utilisateur. L'amplitude des rides varie pendant l'animation du maillage en calculant la contraction des triangles. Na *et al.* ont amélioré la technique de transfert de Noh *et al.* [NN01] en offrant la possibilité d'un transfert hiérarchique [NJ04]. Celui-ci peut être appliqué à différents niveaux de détails, d'une résolution basse jusqu'à une résolution élevée où les rides dynamiques apparaissent.

Alors que des techniques ont été développées pour la génération de rides, très peu ont été développées spécifiquement pour des applications en temps réel [LC04, Oat07]. Larboulette *et al.* ont proposé une technique pour générer des rides dynamiquement et en temps réel [LC04]. L'utilisateur doit déterminer une zone sur laquelle les rides doivent apparaître et un modèle de ride (ensemble de segments 2D utilisés pour contrôler une courbe 3D). La courbe de contrôle conserve sa longueur,

ainsi, son amplitude varie lorsque la surface est déformée. Les détails sont affichés par subdivision du maillage le long des zones de rides.

Beaucoup des méthodes présentées nécessitent des maillages fins ou des techniques de subdivision « à la volée » coûteuses pour générer ou afficher les rides [BLB⁺08, LLZ⁺08, NJ04, KG03]. À cause des contraintes liées au temps réel, ces techniques sont difficilement utilisables dans le cadre d'applications où les ressources sont partagées. Cependant, les récentes avancées en informatique graphique permettent de représenter efficacement les détails fins en utilisant les techniques de *bump-mapping* pour un rendu interactif.

Oat a présenté une technique pour mélanger facilement des cartes de rides appliquées sur un maillage [Oat07]. Les cartes sont subdivisées en régions, pour chaque région, des coefficients permettent de pondérer l'influence de chaque carte. Cette technique nécessite peu de ressources en temps de calcul et de stockage (3 cartes sont utilisées). De plus, elle est facilement adaptable à une implémentation existante. L'inconvénient principal de cette méthode réside dans le fait qu'elle nécessite de définir manuellement les coefficients pour chaque région au cours de l'animation. Nous proposons dans le Chapitre 5 une méthode automatique adaptée aux visages contrôlés par *skinning* qui génère des détails dynamiques à partir d'exemples. Nous faisons également la proposition d'une méthode d'acquisition qui permet de fournir des données d'entrée à notre technique de rides dynamiques.

Étant donné que l'approche que nous présentons au Chapitre 5 est basée sur une unique vue pour l'acquisition des cartes de normales, nous nous concentrons dans ce paragraphe sur la reconstruction de formes à partir d'une simple image, et plus précisément, sur les approches qui utilisent les informations d'éclairage que l'on peut classer dans la famille du problème du *shape-from-shading*. Ce problème a été largement traité dans le domaine de la vision par ordinateur [DFS08]. Il est reconnu comme étant difficile du fait qu'il s'agisse d'un problème mal posé [PF05], ainsi, peu d'approches ont travaillé sur des photographies réelles [VVBW09]. Wu *et al.* ont proposé une technique intéressante de reconstruction interactive de normales à partir d'une simple image [WSTS08]. Ils décomposent la reconstruction de la surface en deux étapes : la première détecte automatiquement les normales dans le domaine des hautes-fréquences ; la seconde permet de reconstruire la forme de l'objet dans sa globalité grâce à l'intervention de l'utilisateur qui définit les normales dans le domaine des basses-fréquences. Dans le cadre des détails fins du visage, il est possible de se limiter à la reconstruction des détails fins uniquement plutôt qu'à la reconstruction d'une forme 3D complète. Ceci diminue légèrement la difficulté du problème inhérent

au *shape-from-shading* [WSTS08]. En cherchant à ne reconstruire que des éléments locaux comme les rides, il est possible de limiter l'étape d'interaction avec l'utilisateur qui est principalement dédiée à fournir des informations globales. De plus, travailler avec des visages permet d'avoir des indications sur la forme de l'objet à reconstruire. Ces indications peuvent offrir une réponse au problème du *shape-from-shading* et permettent même de reconstruire l'aspect global du visage [SH08]. À notre connaissance, l'état de l'art sur la problématique du *shape-from-shading* ne propose pas de résultat convaincant sur l'acquisition des rides d'animation de la peau et plus particulièrement du visage, c'est pourquoi une phase de reconstruction procédurale est souvent associée à l'acquisition [BBA⁺07, LAGP09].

2.5.5 Synthèse d'animation multimodale

La création manuelle d'une animation par un artiste offre un large panel de possibilités, cependant le réalisme est difficile à obtenir. En revanche, les animations obtenues par la performance d'un acteur réel le sont, mais ces techniques sont très peu flexibles et il n'est pas toujours facile d'éditer ou de modifier une animation acquise par caméra. C'est pourquoi certains travaux se sont orientés vers la synthèse d'animation qui ne se base pas sur des données capturées. Beaucoup de ces travaux se sont concentrés sur la production de « parole visuelle ». Il s'agit de générer les mouvements de la bouche, à partir de données audio [DN06, BP04, GGMT07, SSRMF06], ou à partir d'un texte [WL93, PBS96, AHK⁺02, ASHS05]. Le plus souvent, une base de *visèmes* est utilisée ainsi qu'une base d'émotions. Un *visème* est l'équivalent visuel d'un *phonème*, un *phonème* est lui même la plus petite segmentation d'une unité de son utilisée dans le cadre de la parole (voir la Figure 2.17). L'expression du visage est alors un mélange entre l'état émotionnel du personnage et le texte qu'il énonce. Pour rendre la synthèse d'animation plus réaliste, les expressions faciales ne suffisent pas et des travaux concernant la gestuelle de la tête ont également été proposés (mouvements et clignements des yeux, mouvements de la tête, etc.) [GLB07, JL08]. La synthèse d'animation est souvent accompagnée de langages d'animation, de scripts, ou de systèmes permettant de décrire une animation complète du visage. Parmi ceux-ci nous pouvons citer *FacEMOTE* [BB02], *BEAT* (*Behavior Expression Animation Toolkit*) [CVB01], *RUTH* [DRSV02], etc.

2.6 Conclusion

Nous avons présenté un grand nombre de travaux concernant l'animation faciale en trois dimensions. Nous les avons classés principalement sous la forme de deux familles, celle des méthodes de paramétrisation et celle des méthodes d'animation.



FIGURE 2.17 – Bases de visèmes de plusieurs personnages utilisées pour la synthèse d’animations à partir de sons [JL08].

En effet, ce sont deux questions primordiales que doivent se poser les concepteurs d’un projet qui fait intervenir des personnages virtuels. Selon les applications visées et les contraintes qui leur sont liées, le choix des techniques peut fortement varier. Il faut donc déterminer quels sont les critères principaux à respecter. Parmi ces critères, on peut lister :

- application temps réel ou non : si oui, ressources disponibles pendant l’animation ;
- interactivité ;
- moyens financiers à disposition ;
- temps disponible à la création de la paramétrisation et de l’animation ;
- précision de détail désirée ;
- utilisateurs novices ou experts ;
- utilisation de nouveaux visages à chaque utilisation ;
- ...

Par exemple, l’animation faciale dans le cadre des jeux vidéo impose la contrainte du temps réel, de ressources faibles et partagées entre le rendu, l’intelligence artificielle, la physique, le réseau, l’interface, etc. Contraintes qui ne sont pas présentes dans le cadre des productions cinématographiques où la qualité visuelle est le critère le plus important et dont les moyens financiers permettent de mettre en place des systèmes beaucoup plus lourds et coûteux.

L’avantage principal des techniques de *blendshape* est que l’artiste maîtrise fortement son animation et ses déformations étant donné qu’il crée lui-même les formes de base. Il peut ainsi définir l’espace linéaire dans lequel son visage sera animé. Le réalisme d’une expression dépend principalement de la qualité de sa modélisation. Cependant, ce qui est considéré comme un avantage dans certains contextes devient un inconvénient dans d’autres. En effet, pour obtenir un ensemble de contrôles suf-

fisant sur un visage, l'artiste doit modéliser un grand nombre de formes dont le temps nécessaire à la conception est conséquent. De plus, la quantité de données (un maillage complet du visage ou de la tête par expression) peut s'avérer être une contrainte importante pour certaines applications ou terminaux en terme de mémoire, sachant que le visage, bien que primordial pour le réalisme d'une scène, n'est souvent qu'une petite partie de celle-ci.

Les techniques de modélisations physiques sont souvent difficiles à mettre en œuvre. Sur ce point, elles souffrent du même problème que les techniques de *blendshape*, bien que des approches de transfert de paramétrisation proposées facilitent cette tâche [Orv07]. Elles souffrent également de la complexité des calculs qui les rendent souvent difficilement utilisables en temps réel à ce jour malgré des résultats visuels très prometteurs [SNF05, SSRMF06]. Cependant, elles permettent de reproduire plus fidèlement les déformations du visage (globales ou fines), et il peut être judicieux de les utiliser pour paramétriser un visage pour une autre méthode (par exemple, générer les expressions de base d'une modélisation par *blendshape*).

Les méthodes de paramétrisation par déformation de surface nécessitent quant à elles moins de ressources, à la fois en terme de temps de calcul et en mémoire, notamment celles à base de points de contrôle. C'est pourquoi elles sont souvent choisies dans le cadre d'applications interactives. Cependant il est plus difficile d'obtenir des résultats réalistes et les efforts nécessaires pour paramétriser un visage restent importants. Par rapport aux méthodes de *blendshape*, elles aussi utilisables en temps réel, les méthodes de paramétrisation par déformation de surface offrent l'avantage de n'avoir besoin que de peu de mémoire. Elles offrent également une manipulation plus aisée. En effet, manipuler des points de contrôle directement à la surface du visage peut être beaucoup plus intuitif que de manipuler les poids d'un modèle de *blendshape*.

La création de l'animation en elle-même continue d'occuper les chercheurs, pour proposer des systèmes toujours plus performants et qui proposent des effets de plus en plus fins et subtiles. Après avoir proposé des techniques nécessitant la création manuelle d'animations, la capture de mouvement apporte plus de réalisme, car elle est basée sur la dynamique d'un visage réel. Avec notamment la capture des détails fins qui a fait son apparition depuis peu, elle est presque devenue incontournable. Cependant, elle implique des contraintes non négligeables. La capture en elle-même est souvent une étape délicate qui nécessite des moyens importants, de part le matériel parfois conséquent (plusieurs paires de caméras, spots lumineux, environnement d'éclairage complètement contrôlé), la préparation compliquée (la pose précise

de très nombreux marqueurs), etc. La session d'acquisition se doit d'être parfaitement préparée, étant donné qu'une fois l'animation capturée, elle est difficilement modifiable. Les frais engendrés par l'oubli d'une simple séquence d'animation ou l'acquisition d'une expression peuvent être importants.

Plus généralement, l'animation faciale est un point délicat à cause notamment de la complexité de sa création. On peut remarquer qu'un axe principal de la recherche est de proposer des techniques pour faciliter, améliorer, rendre plus accessible et plus rapide le travail du graphiste et/ou de l'animateur. Il s'agit là d'un objectif majeur, qui n'a pas encore été atteint. Les systèmes proposés nécessitent souvent beaucoup de temps et du matériel onéreux. Pour la capture, les efforts se porteront sur la facilité d'acquisition et la facilité d'adaptation d'une animation, notamment pour les détails fins d'animation qui ne sont pas encore ou peu exploitables en temps réel. Au niveau qualitatif, des résultats photoréalistes ont déjà été obtenus au prix d'efforts considérables : par exemple le *Digital Emily Project* décrit une méthode très réaliste mais qui a demandé plusieurs mois de travail d'artistes et de techniciens pour 90 secondes d'animation et pour un visage unique [ARL⁺09b, ARL⁺09a]. Au niveau des techniques de paramétrisation, l'avenir va se porter sur l'association de techniques. Dans le cadre d'applications interactives, l'association des paramétrisations à base de points de contrôle (sous forme de *skinning*) et de *blendshape* permettra d'apporter à l'animation faciale les avantages obtenus depuis peu pour l'animation du corps entier [LCF00, KM04b, RLN06, LH09, WSLG07, YY09].

Les travaux que nous avons menés au cours de cette thèse se place dans ce contexte. Les constats dressés ci-dessus ont orienté notre recherche. Notre motivation est de proposer des techniques simples et faciles d'utilisation pour la réalisation d'animations faciales. Ces techniques doivent permettre de répondre à des contraintes de temps réel et doivent être adaptées à la majorité des moteurs graphiques actuels et futurs. Nos travaux se portent sur 3 niveaux : automatisation de la paramétrisation par *skinning* ; aide à la création d'animation et au transfert d'animation ; ainsi que la capture et l'animation de détails fins comme les rides de la peau apparaissant durant l'animation.

Chapitre 3

Paramétrisation du visage

Sommaire

3.1	Introduction et motivation	45
3.2	Fonctions à base radiale pour l'interpolation	48
3.2.1	Aspects mathématiques	48
3.2.2	Exemples de fonctions radiales	49
3.2.3	Interpolation de données éparses par un modèle <i>RBF</i>	50
3.2.4	Ajout d'une transformation rigide à l'interpolation	51
3.3	Paramétrisation par approche procédurale	52
3.4	Paramétrisation par transfert	55
3.4.1	Transfert des points de contrôle	55
3.4.2	Transfert des poids os-sommet	63
3.5	Résultats	67
3.6	Conclusion	69

3.1 Introduction et motivation

Nous avons vu dans le Chapitre 2 que paramétriser un visage pour l'animation s'avère être une tâche délicate et très coûteuse en temps, et qu'il existe un réel manque d'outils permettant d'effectuer ce travail plus aisément. Malgré les besoins dans ce secteur, aussi bien l'automatisation que l'assistance ont été jusque là assez peu explorées pour la paramétrisation d'un visage par *skinning*. Nous avons également souligné qu'actuellement, les logiciels d'animation tels que *3ds Max*, *Maya* ou *Blender* disposent de moyens mais qu'ils sont principalement destinés au reste du corps, le visage étant peu concerné.

Les étapes principales d'une paramétrisation par *skinning* consistent à créer un squelette et à le recaler sur le maillage du personnage dans une position que nous appellerons « neutre » ou « au repos » puis d'associer chaque sommet du maillage aux os du squelette. La longueur de ces os permet aux logiciels d'estimer des enveloppes d'influence que l'utilisateur peut éditer et de les appliquer au maillage (voir les Figures 3.1 et 3.2). Ensuite, le graphiste affine les associations os-somets avec des outils « de peinture » qui permettent d'agir directement sur les poids avec le curseur de la souris. Cependant, si ces différents outils offrent la possibilité d'accélérer le processus de paramétrisation, ils ne permettent pas de l'automatiser.

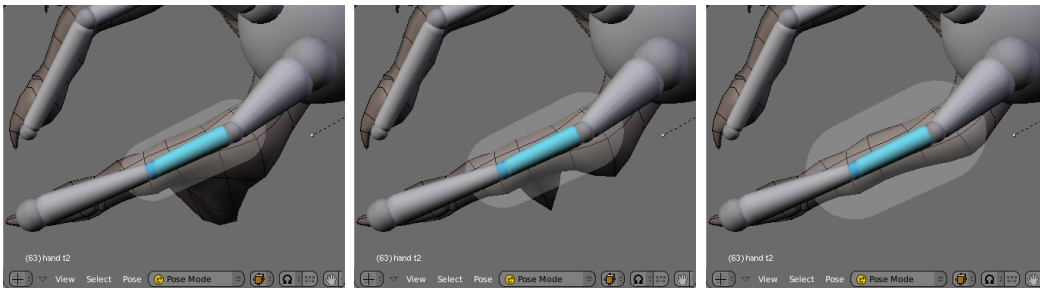


FIGURE 3.1 – Modification de la taille de l'enveloppe d'influence associée à un os sous *Blender*¹.

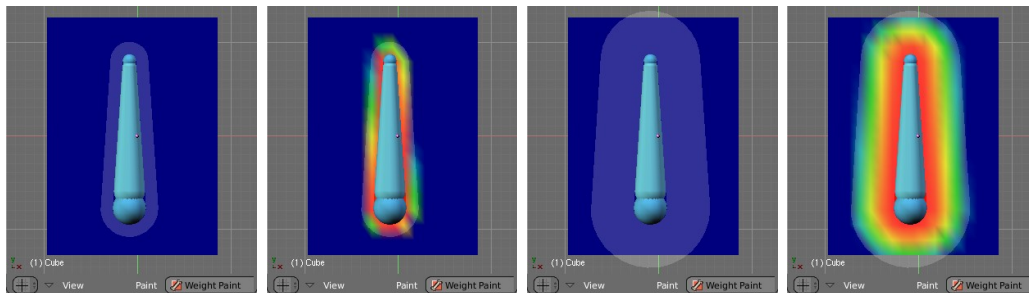


FIGURE 3.2 – Diffusion de l'influence en fonction de l'enveloppe sous *Blender*.

L'aide à la paramétrisation par *skinning* a toutefois fait l'objet de quelques travaux. Nous pouvons notamment ceux présentés dans [JZP⁺08]. Le principe général de paramétrisation s'appuie sur un système à base de cages qui s'ajoute au *skinning*. Bien que le but principal de cette approche n'est pas de proposer une réponse au problème d'automatisme, ce système définit un template de paramétrisation adaptable à différents personnages et qui facilite le transfert de paramétrisation. D'autres travaux comme le système *Pinocchio* se concentrent sur l'automatisation de la paramétrisation [BP07]. Le principe général est le suivant : une première étape consiste à recaler et à adapter un squelette générique au maillage du personnage à paramétriser ; une seconde étape emploie un algorithme de diffusion de chaleur à partir de

chaque os pour les associer correctement aux sommets du maillage, évitant ainsi le problème de l'outil « enveloppe » qui souvent englobe des parties non désirées. Par exemple, l'enveloppe du bras peut toucher le tronc du corps ou l'enveloppe d'une jambe peut toucher l'autre jambe avec les outils classiques, ce qui nécessite par la suite l'intervention du graphiste pour rectifier ces défauts. *Pinocchio* est un système efficace et entièrement automatique. Cependant, comme pour les outils proposés par les logiciels d'animation, il n'est que peu applicable au visage. En effet, alors que le corps est déformé de façon rigide par le squelette, les déformations du visage sont plus fines et sont liées, non pas aux mouvements des os, mais aux déplacements des points de contrôle qui décrivent l'action des muscles sur la peau. On comprend alors que la dénomination généralement utilisée d'animation par squelette pour le visage peut induire en erreur, il serait plus naturel de parler de paramétrisation par points de contrôle. Cependant, les aspects implémentation et génération des déformations par paramétrisation par points de contrôle sont similaires à la technique de l'animation par squelette ; ce qui constitue d'ailleurs un des avantages de l'animation par *skinning*. Ceci justifie l'emploi du terme *skinning* ou déformation par squelette en parallèle au terme de déformation par points de contrôle.

Assez peu de travaux ont été dédiés au visage dans ce domaine. Orvalho *et al.* ont proposé une méthode pour transférer la paramétrisation existante d'un visage à un autre [OZS06]. L'idée est qu'une fois un visage paramétrisé, il n'est plus nécessaire d'effectuer le travail pour chacun des personnages. Pour ce faire, il faut tout d'abord positionner manuellement des points repères sur chacun des deux visages : le visage source, qui est paramétrisé ; et le visage cible, non paramétrisé. Une déformation par *RBF* dont l'initialisation est effectuée grâce aux points repères permet ensuite d'adapter les paramètres sources au visage cible. Bien que destinée tout d'abord à une paramétrisation par modélisation physique du visage, cette approche est facilement adaptable à une paramétrisation par points de contrôle.

Dans cette partie nous sommes brièvement revenu sur l'état de l'art pour souligner le manque d'outils spécifiques au visage. Ce qui a motivé nos travaux sur ce sujet se rapporte à l'assistance et à la génération automatique de paramétrisation par point de contrôle du visage. Ceci a aboutit à deux méthodes complémentaires. La première est une approche procédurale qui a pour but d'assister l'artiste dans cette tâche, de façon similaire aux outils proposés par les logiciels d'animation qui suggèrent une zone d'influence des os selon leur taille. Notre méthode utilise les informations de morphologie que la position des points de contrôle fournit pour associer les sommets aux points de contrôle. Notre deuxième approche s'inspire du concept de transfert de paramétrisation. Notre technique permet de réutiliser la pa-

ramétrisation par *skinning* d'un visage source vers un visage cible avec l'avantage de positionner de façon automatique les points de contrôle. Ceci permet d'éviter une intervention manuelle. Nous commençons ce chapitre par décrire le principe et les calculs de l'interpolation de données éparses à partir de fonctions à base radiale (*Radial Basis Functions*, *RBF*) que nous utilisons dans chaque chapitre de ce document. Nous décrivons ensuite les deux approches proposées.

3.2 Fonctions à base radiale pour l'interpolation

Les *RBF* sont largement utilisées dans le domaine de l'informatique graphique pour l'approximation de surface ou la régression. Dans le cadre de nos travaux, nous les avons notamment utilisées pour effectuer l'interpolation d'un visage à un autre en utilisant des ensembles de points repères appariés comme échantillons. L'initialisation d'un modèle de *RBF* consiste à minimiser une distance et donc à trouver la meilleure approximation entre les échantillons d'entrée et les valeurs obtenues par l'approximation. Le terme d'interpolation de données éparses est régulièrement utilisé dans le contexte de l'animation faciale pour désigner cette approximation. Pour rester conforme avec cette terminologie et par abus de langage, nous adopterons cette dénomination au détriment de l'exactitude mathématique.

3.2.1 Aspects mathématiques

Dans cette section, nous faisons un rappel de certaines notions qui sont nécessaires à la compréhension des *RBF* et de leur utilisation dans le cadre de l'animation faciale. Une fonction à base radiale est une fonction symétrique autour d'un centre μ_j (les plus fréquemment utilisées sont listées dans la Section 3.2.2). Soit $\phi_j(x) = \phi(\|x - \mu_j\|)$ la fonction radiale de centre μ_j avec $\|\dots\|$ une norme. Soit $W = (w_1, \dots, w_l)^t$ le vecteur des coefficients de pondération, et y une fonction telle que $y : \mathbb{R}^d \rightarrow \mathbb{R}$. Un modèle de *RBF* est une combinaison linéaire de fonctions radiales de la forme :

$$y(x) = \sum_{j=1}^l w_j \phi_j(x) \quad (3.1)$$

Si les centres μ_j sont connus, l'Équation 3.1 peut s'écrire :

$$y(x) = \sum_{j=1}^l w_j \phi(\|x - \mu_j\|) \quad (3.2)$$

On cherche alors à déterminer le vecteur W qui minimise les erreurs $E = (e_1, \dots, e_n)^t$ pour chacune des n valeurs théoriques attendues $y(x_i)$ et les valeurs

obtenues $\tilde{y}(x_i)$:

$$\tilde{Y} = Y + E \Leftrightarrow \begin{bmatrix} \tilde{y}(x_1) \\ \vdots \\ \tilde{y}(x_i) \\ \vdots \\ \tilde{y}(x_n) \end{bmatrix} = \begin{bmatrix} y(x_1) \\ \vdots \\ y(x_i) \\ \vdots \\ y(x_n) \end{bmatrix} + \begin{bmatrix} e_1 \\ \vdots \\ e_i \\ \vdots \\ e_n \end{bmatrix} \quad (3.3)$$

Soit $Y = (y_1, \dots, y_l)^t$, avec $y_i = y(x_i)$, le vecteur de taille l qui regroupe les sorties théoriques, on note par H la matrice de taille $l \times n$ telle que

$$H_{ij} = \phi_j(x_i) \quad (3.4)$$

L'Équation 3.3 peut s'écrire sous forme matricielle :

$$\tilde{Y} = Y + E = HW + E \quad (3.5)$$

En explicitant l'Équation 3.5, on obtient :

$$\begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_i \\ \vdots \\ \tilde{y}_n \end{bmatrix} = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_j(x_1) & \cdots & \phi_l(x_1) \\ \vdots & \ddots & \vdots & & \vdots \\ \phi_1(x_i) & & \phi_j(x_i) & & \phi_l(x_i) \\ \vdots & & \vdots & \ddots & \vdots \\ \phi_1(x_n) & \cdots & \phi_j(x_n) & \cdots & \phi_l(x_n) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_j \\ \vdots \\ w_l \end{bmatrix} + \begin{bmatrix} e_1 \\ \vdots \\ e_i \\ \vdots \\ e_n \end{bmatrix} \quad (3.6)$$

3.2.2 Exemples de fonctions radiales

Voici quelques exemples de fonctions à base radiale qui sont fréquemment utilisées avec $r = \|x - \mu_j\|$ et σ un paramètre à déterminer :

- **Multiquadrique** $\phi(r) = \sqrt{r^2 + \sigma^2}$;
- **Gaussienne** $\phi(r) = \exp(-r^2/2\sigma^2)$;
- **Bi-harmonique** $\phi(r) = r$;
- **Tri-harmonique** $\phi(r) = r^3$;
- **Thin plate spline** $\phi(r) = r^2 \ln(r)$.

La fonction *Thin Plate spline* est particulièrement adaptée pour l'interpolation de fonctions lisses à deux variables, les gaussiennes sont principalement utilisées pour les réseaux de neurones, et la multiquadrique est utilisée dans de nombreuses applications, notamment pour l'adaptation de données topographiques. Pour l'adaptation de fonctions à trois variables, le choix inclue la bi-harmonique et la tri-harmonique [CBC+01].

3.2.3 Interpolation de données éparses par un modèle *RBF*

Dans le cadre de l'interpolation de données éparses en trois dimensions, on cherche à déterminer trois fonctions $\tilde{Y}_x : \mathbb{R}^3 \rightarrow \mathbb{R}$, $\tilde{Y}_y : \mathbb{R}^3 \rightarrow \mathbb{R}$ et $\tilde{Y}_z : \mathbb{R}^3 \rightarrow \mathbb{R}$ qui correspondent aux interpolations sur chaque axe de l'espace \vec{x} , \vec{y} et \vec{z} . Pour cela on s'appuie sur les données d'entrée qui correspondent à deux ensembles de points mis en correspondance : $\mathcal{S} = \{s_i\}$, $i = [1..n]$, l'ensemble des n points sources et $\mathcal{T} = \{t_i\}$, $i = [1..n]$, l'ensemble des n points cibles². Une étape d'appariement des points établie une mise en correspondance entre ces deux ensembles telle que le point s_i de l'espace source correspond au point t_i de l'espace cible.

Soit $T_x = \{t_{i,x}\}$ (resp. $T_y = \{t_{i,y}\}$ et $T_z = \{t_{i,z}\}$) les composantes sur l'axe \vec{x} (resp. \vec{y} et \vec{z}) des points t_i , $E_x = \{e_{i,x}\}$ (resp. $E_y = \{e_{i,y}\}$ et $E_z = \{e_{i,z}\}$) les erreurs d'approximation à minimiser de la fonction \tilde{Y}_x (resp. \tilde{Y}_y et \tilde{Y}_z) et $W_x = \{w_{j,x}\}$, $j \in [1..n]$ (resp. W_y et W_z) les coefficients de pondération à déterminer de la fonction \tilde{Y}_x (resp. \tilde{Y}_y et \tilde{Y}_z). À partir de l'Équation 3.2 et en utilisant les points s_i comme centre des fonctions radiales, on obtient :

$$\begin{bmatrix} Y_x(s_i) \\ Y_y(s_i) \\ Y_z(s_i) \end{bmatrix} = \begin{bmatrix} \tilde{Y}_x(s_i) \\ \tilde{Y}_y(s_i) \\ \tilde{Y}_z(s_i) \end{bmatrix} + \begin{bmatrix} e_{i,x} \\ e_{i,y} \\ e_{i,z} \end{bmatrix} = \begin{bmatrix} t_{i,x} \\ t_{i,y} \\ t_{i,z} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n w_{j,x} \cdot \phi(\|\vec{s}_i - \vec{s}_j\|) \\ \sum_{j=1}^n w_{j,y} \cdot \phi(\|\vec{s}_i - \vec{s}_j\|) \\ \sum_{j=1}^n w_{j,z} \cdot \phi(\|\vec{s}_i - \vec{s}_j\|) \end{bmatrix} \quad (3.7)$$

En utilisant la notation matricielle définie par l'Équation 3.4, la matrice H est maintenant une matrice carrée de taille $n \times n$ puisque le nombre de centres des fonctions radiales et le nombre d'échantillons sont identiques. On peut alors écrire :

$$\begin{aligned} \tilde{Y}_x + E_x &= T_x = H.W_x \\ \tilde{Y}_y + E_y &= T_y = H.W_y \\ \tilde{Y}_z + E_z &= T_z = H.W_z \end{aligned} \quad (3.8)$$

Il s'agit alors de résoudre trois systèmes d'équations à n inconnues chacune (les poids $w_{j,x}$, $w_{j,y}$ et $w_{j,z}$) dont les solutions existent et sont uniques [Pow87]. La minimisation par les moindres carrés peut être obtenue en utilisant la matrice pseudo-inverse H^+ de H : $H^+ = (H^T H + \lambda I)^{-1} H^T$, on obtient alors les calculs matriciels suivant :

$$\begin{aligned} W_x &= H^+.T_x \\ W_y &= H^+.T_y \\ W_z &= H^+.T_z \end{aligned} \quad (3.9)$$

2. Nous utilisons fréquemment des notations avec les lettres 's' et 't' pour désigner des éléments en relation avec les espaces *source* et *target*

3.2.4 Ajout d'une transformation rigide à l'interpolation

Il est possible de prendre en compte la transformation rigide (rotation, échelle et translation) éventuelle de l'interpolation entre les deux ensemble de points \mathcal{S} et \mathcal{T} pour obtenir une meilleure approximation. Cette transformation est une étape préliminaire qui permet de rapprocher les points sources des points cibles avant de calculer l'interpolation par *RBF*. Une telle transformation A peut s'écrire sous forme matricielle :

$$A = \begin{bmatrix} a_{x,x} & a_{y,x} & a_{z,x} & a_{w,x} \\ a_{x,y} & a_{y,y} & a_{z,y} & a_{w,y} \\ a_{x,z} & a_{y,z} & a_{z,z} & a_{w,z} \end{bmatrix} \quad (3.10)$$

Soit $s_i = (s_{i,x}, s_{i,y}, s_{i,z}, s_{i,w})^t$ le point auquel on a rajouté la quatrième coordonnée $s_{i,w} = 1$ qui permet de le multiplier par la matrice de transformation et donc de lui appliquer la translation qui est définie par les éléments $a_{w,x}$, $a_{w,y}$ et $a_{w,z}$ de A . La multiplication de A par s_i qui donne le point transformé s'_i s'écrit :

$$s'_i = A \times s_i = \begin{bmatrix} a_{x,x} & a_{y,x} & a_{z,x} & a_{w,x} \\ a_{x,y} & a_{y,y} & a_{z,y} & a_{w,y} \\ a_{x,z} & a_{y,z} & a_{z,z} & a_{w,z} \end{bmatrix} \times \begin{bmatrix} s_{i,x} \\ s_{i,y} \\ s_{i,z} \\ s_{i,w} \end{bmatrix} \quad (3.11)$$

On peut définir 3 polynôme q_x , q_y et q_z tels que :

$$\begin{aligned} q_x(s_i) &= a_{x,x} \times s_{i,x} + a_{y,x} \times s_{i,y} + a_{z,x} \times s_{i,z} + a_{w,x} \\ q_y(s_i) &= a_{x,y} \times s_{i,x} + a_{y,y} \times s_{i,y} + a_{z,y} \times s_{i,z} + a_{w,y} \\ q_z(s_i) &= a_{x,z} \times s_{i,x} + a_{y,z} \times s_{i,y} + a_{z,z} \times s_{i,z} + a_{w,z} \end{aligned} \quad (3.12)$$

que l'on peut ajouter à l'Équation 3.7 :

$$\begin{bmatrix} t_{i,x} \\ t_{i,y} \\ t_{i,z} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n w_{j,x} \cdot \phi(\|\vec{s}_i - \vec{s}_j\|) + q_x(s_i) \\ \sum_{j=1}^n w_{j,y} \cdot \phi(\|\vec{s}_i - \vec{s}_j\|) + q_y(s_i) \\ \sum_{j=1}^n w_{j,z} \cdot \phi(\|\vec{s}_i - \vec{s}_j\|) + q_z(s_i) \end{bmatrix} \quad (3.13)$$

Soit S la matrice $4 \times n$ des points sources, T la matrice $4 \times n$ des points cibles, W la matrice $3 \times n$ des poids, et A la matrice 4×4 de la transformation affine :

$$S = \begin{bmatrix} s_{1,x} & s_{1,y} & s_{1,z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ s_{n,x} & s_{n,y} & s_{n,z} & 1 \end{bmatrix}, T = \begin{bmatrix} t_{1,x} & t_{1,y} & t_{1,z} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ t_{n,x} & t_{n,y} & t_{n,z} & 1 \end{bmatrix} \quad (3.14)$$

$$W = \begin{bmatrix} w_{1,x} & w_{1,y} & w_{1,z} & \\ \vdots & \vdots & \vdots & \vdots \\ w_{n,x} & w_{n,y} & w_{n,z} & \end{bmatrix} \quad (3.15)$$

Soit A_x (resp. A_y et A_z) la première (resp. deuxième et troisième) ligne de A , \vec{o} un vecteur nul de dimension 4, et O une matrice nulle de dimensions 4×4 . Les équations matricielles présentée dans l'Équation 3.8 peuvent s'écrire sous la forme (présentée uniquement pour la composante x ici) :

$$\begin{bmatrix} T_x \\ \vec{o} \end{bmatrix} = \left[\begin{array}{c|c} H & S \\ \hline S^t & O \end{array} \right] \begin{bmatrix} W_x \\ A_x^t \end{bmatrix} \quad (3.16)$$

et se résolvent de la même façon que précédemment, en calculant la matrice pseudo-inverse :

$$\begin{bmatrix} W_x \\ A_x^t \end{bmatrix} = \left[\begin{array}{c|c} H & S \\ \hline S^t & O \end{array} \right]^+ \begin{bmatrix} T_x \\ \vec{o} \end{bmatrix} \quad (3.17)$$

Nous utiliserons la notation $\mathcal{RBF}(\mathcal{S} \rightarrow \mathcal{T})$ pour désigner l'interpolation par *RBF* initialisée entre l'ensemble de points sources \mathcal{S} et l'ensemble de points cibles \mathcal{T} et qui applique les 3 fonctions \tilde{Y}_x , \tilde{Y}_y et \tilde{Y}_z à un point s dans l'espace source pour obtenir son interpolation t dans l'espace cible.

3.3 Paramétrisation par approche procédurale

L'approche procédurale que nous avons développée s'appuie sur des points de contrôle positionnés sur le visage et une fonction d'influence. Cette fonction détermine la pondération qui associe chaque sommet du maillage avec chaque point de contrôle considéré. Notre approche consiste en une première étape pendant laquelle l'utilisateur place manuellement des points de contrôle, puis d'une seconde durant laquelle l'influence des points de contrôle sur le maillage est générée automatiquement. Contrairement à la paramétrisation du corps par enveloppe, qui utilise la longueur des os pour initialiser la paramétrisation, le visage nécessite une initialisation différente et adaptée. L'idée principale est que la position des points de contrôle offre une connaissance préalable sur la morphologie du visage, ce qui suppose indirectement qu'il est possible de connaître la zone d'influence de chacun des points de contrôle. On peut souligner la possibilité de se reporter à des études statistiques anthropométriques ou à des bases de données existantes pour apporter des informations utiles pour la configuration d'une telle approche [HR98, KM04a].

Logiquement, l'influence du point de contrôle P_j sur un sommet v_i doit varier en fonction de la distance qui les sépare : plus P_j est éloigné de v_i , et moins son influence est importante. De plus, il faut que l'influence diminue progressivement. En effet, deux sommets voisins doivent être influencés de façon similaire par un point de contrôle donné. Le cas échéant, le déplacement du point de contrôle entraîne un fort déplacement pour un sommet, et non pour l'autre, ce qui provoque des défauts visuels non désirés. De façon plus formelle, soit \vec{v}_i la position d'un sommet du maillage donné par ses coordonnées, et P_j un point de contrôle désigné par sa numérotation en tant que *FDP* de la norme *MPEG-4* de coordonnées \vec{P}_j . On pose $\vec{d} = \vec{v}_i - \vec{P}_j = (d_x, d_y, d_z)^t$. L'influence de P_j sur v_i peut être quantifiée par :

$$\alpha_{ij} = \max \left(0, 1 - \left(\frac{d_x^2}{r_x^2} + \frac{d_y^2}{r_y^2} + \frac{d_z^2}{r_z^2} \right) \right) \quad (3.18)$$

$\alpha_{ij} \rightarrow 1$ lorsque le sommet v_i tend à être à la même position que P_j et $\alpha_{ij} = 0$ si le sommet v_i est trop éloigné de P_j et sort de sa zone d'influence. Les valeurs sont ensuite normalisées pour obtenir les coefficients de pondération affectés aux sommets du maillage tels que $\sum_j w_{ij} = 1$:

$$w_{ij} = \frac{\alpha_{ij}}{\sum_{k=1}^n \alpha_{ik}} \quad (3.19)$$

Signalons que nous ne conservons que les quatre poids les plus importants, il s'agit ici d'une question d'implémentation, la notre en utilise quatre mais il est tout à fait possible d'en utiliser un nombre maximal différent

Nous avons opté pour une ellipsoïde pour définir la fonction de pondération. Ce choix est motivé par le fait qu'une telle fonction a l'avantage d'avoir trois paramètres r_x , r_y et r_z (les trois axes de l'ellipsoïde), il est ainsi possible de configurer la taille de la zone d'influence pour chaque point de contrôle indépendamment (voir la Figure 3.3). Prenons l'exemple de la paupière supérieure de l'œil, dont l'influence est déterminée par un unique point de contrôle. La région d'influence de ce point est large (d'un coin de l'œil à l'autre) alors que sa hauteur est plus réduite, ainsi il est intéressant de pouvoir paramétrer l'influence du point indépendamment sur chaque axe. Dans ce cas, r_x dépend de la distance entre les coins de l'œil, et r_y dépend de la distance de la paupière avec le sourcil. L'indexation *MPEG-4* nous permet de connaître la sémantique de chaque point et apporte des informations complémentaires à la position indispensable au calcul de paramètres relatifs aux positions des points de contrôle entre eux.

Voici quelques exemples de paramètres que nous avons utilisés :

- **4.4 et 4.3**, points situés au centre des sourcils gauche et droit³ :
 - r_x = distance entre les deux extrémités d'un même sourcil /2,
 - $r_y = r_x$,
 - $r_z = r_x$.
- **3.1 et 3.2**, points situés au centre de la paupière supérieure des yeux :
 - r_x = largeur de l'oeil /1.5,
 - r_y = distance entre les paupières inférieure et supérieure de l'œil,
 - $r_z = r_x$.
- **8.1**, point situé au centre de la lèvre supérieure :
 - r_x = largeur de la bouche /3.,
 - r_y = distance du point avec le nez $\times 0.8$,
 - $r_z = r_x$.
- **8.2**, point situé au centre de la lèvre inférieure :
 - r_x = largeur de la bouche /2.5,
 - r_y = distance du point avec le menton /1.5f,
 - $r_z = r_x$.

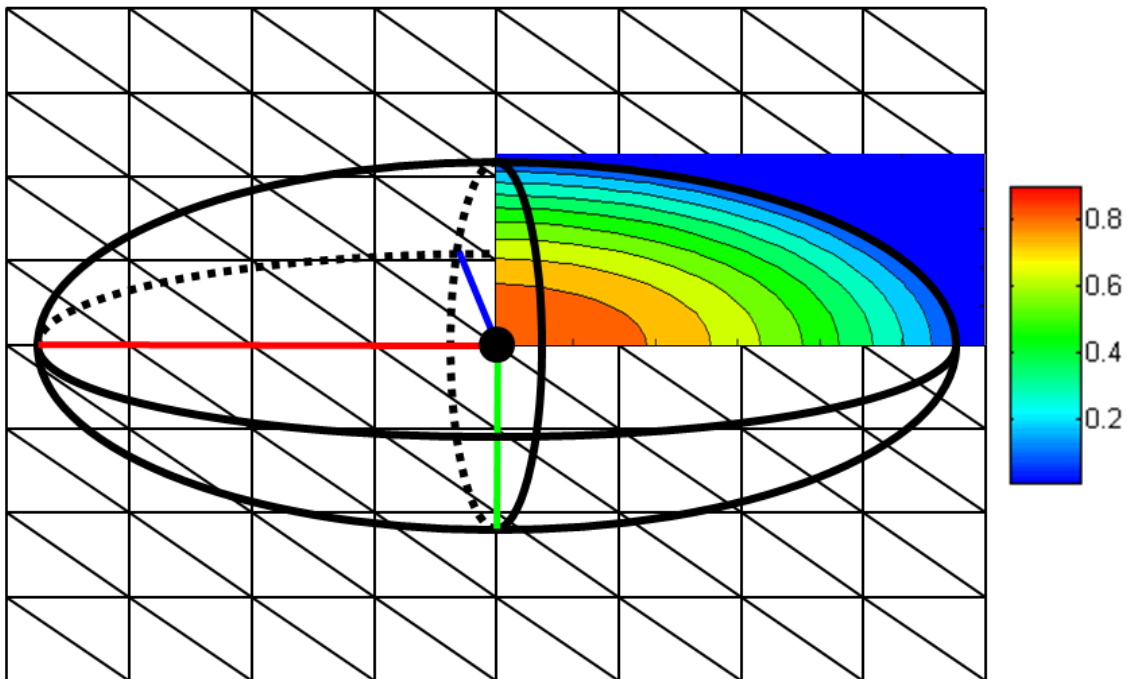


FIGURE 3.3 – Représentation de la fonction de pondération associée au point de contrôle (en noir au centre du schéma). Les segments rouge, vert et bleu représentent les valeurs des trois paramètres r_x , r_y et r_z . Le dessin 2D de la partie supérieure droite de l'ellipsoïde illustre l'influence décroissante du point de contrôle sur les sommets

3. Définition des *FDP* de *MPEG-4*

3.4 Paramétrisation par transfert

L'approche procédurale semi-automatique d'un visage par heuristiques que nous venons de présenter permet d'aider l'artiste dans sa tâche de paramétrisation, cependant, elle n'est pas automatique et son utilisation dans le cadre d'applications grand public peut s'avérer compliquée. Notre deuxième approche de paramétrisation se base sur l'idée qu'une technique permettant de réutiliser une paramétrisation existante offrirait un gain de temps important. Plutôt que de réaliser une paramétrisation pour chaque visage d'une application, une seule paramétrisation transférée à chaque personnage serait suffisante. En nous inspirant de la méthode semi-automatique proposée par Orvalho *et al.* destinée à une paramétrisation physique, nous proposons dans cette partie une approche automatique de transfert de paramétrisation pour le *skinning*. Une étape importante et présente dans de nombreux travaux sur l'animation faciale consiste à positionner de façon manuelle des points repères sur le visage. Nous avons donc développé une méthode de détection et de mise en correspondance automatique de ces points repères entre deux visages dans le but de transférer les points de contrôle d'un visage source à un visage cible. Étant donné que le visage peut être représenté par une carte de hauteur par rapport au plan de la caméra, la première étape de la méthode que nous proposons utilise des techniques issues du domaine de la vision et consiste en une mise en correspondance de ces cartes de hauteur. À l'issue de cette phase, nous transférons les points de contrôle, et finalement, un recalage dense des deux surfaces 3D permet de transférer les paramètres du *skinning* (influence de chaque os sur les sommets) du visage source au visage cible.

3.4.1 Transfert des points de contrôle

Trouver la correspondance entre deux surfaces représentées par un maillage est un point clé dans de nombreux problèmes de l'informatique graphique : modélisation à partir de données issues de capteurs, compression, transfert d'information, application de texture ou déformation, etc. Cette mise en correspondance peut prendre plusieurs formes selon la nature des surfaces traitées : objets rigides et objets déformables, images numérisées et maillages, etc. Notre problème se situe dans le cadre de la mise en correspondance d'instances différentes d'un même objet. Cela signifie que les deux objets ont des caractéristiques communes mais des formes et géométries différentes. Notre méthode se rapproche des méthodes de recalage de formes dans le domaine de la vision [SMFF07]. Néanmoins, une différence importante est que nous ne travaillons pas sur deux vues de la même instance d'un objet. En effet, nous disposons de deux visages différents, chaque visage est représenté par une sur-

face qui correspond à une vue de face de celui-ci. Les deux visages peuvent être de morphologies différentes.

Soulignons que la mise en correspondance de surfaces déformables est un problème récurrent dans le domaine de l’animation du corps et du visage ainsi que dans le domaine de la capture de mouvement. Les techniques employées visent souvent à recalculer un modèle générique sur la forme numérisée [DN07]. Ce recalage peut être obtenu par la minimisation d’un critère d’erreur. Ces méthodes sont robustes sur des surfaces ayant les mêmes reliefs, y compris dans le cas de la présence de trous ou d’un faible échantillonnage. Cependant elles deviennent limitées lorsqu’il y a de grandes différences de morphologie entre les deux surfaces. Pour pallier cette limite, il est souvent efficace de travailler sur la projection des surfaces représentées alors dans un espace en deux dimensions [LDRS05], c’est ce que nous faisons dans la méthode proposée. Ceci offre la possibilité de faire abstraction de la résolution des maillages et de pouvoir s’appuyer sur des descripteurs existants en deux dimensions. Pour la plupart des méthodes de mise en correspondance, le processus se réalise en deux étapes. Tout d’abord la détection et l’appariement de points repères sur chacun des visages sont réalisés ; cette étape est souvent réalisée de façon manuelle. En s’appuyant sur ces points repères, un recalage dense des deux surfaces est réalisé par des déformations à base de *RBF* [NN01, OZS06]. Dans notre cas, nos travaux se sont surtout focalisés sur l’étape de détection des points repères pour rendre notre méthode entièrement automatique.

3.4.1.1 Détection et appariement des points repères

Pour déterminer les points repères nécessaires à une mise en correspondance dense des deux visages, nous avons mis en place une méthode incrémentale qui réalise à chaque étape : une segmentation, une détection de points et leur mise en correspondance. Comme nous l’avons noté ci-dessus, sans nuire à la généralité et sans risquer de perdre de l’information, il est parfois plus pratique de travailler sur une carte de hauteur plutôt que sur un maillage 3D. Cela évite certains inconvénients liés au traitement des maillages, comme la résolution qui peut fortement varier entre un modèle et un autre. Et il est admis qu’effectuer des calculs sur une grille régulière lorsque cela est possible est plus efficace que de les effectuer sur un maillage. L’obtention des cartes de hauteur est très rapide, elle consiste simplement à récupérer les valeurs du *Z-Buffer* lors du rendu orthogonal de chaque visage (voir la Figure 3.4).

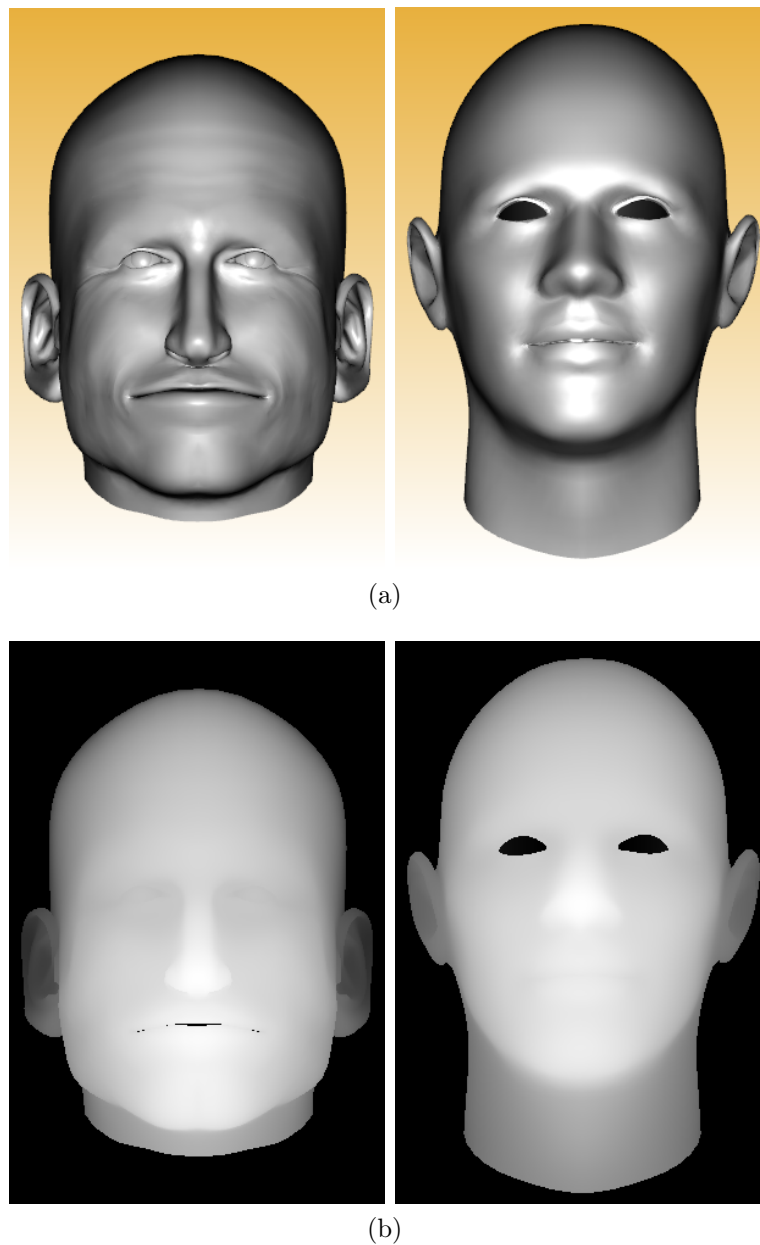


FIGURE 3.4 – Exemple de génération de cartes de hauteur (b) obtenues grâce au *Z-Buffer* d'un rendu 3D orthogonal (a).

Algorithme général

De par sa nature incrémentale, notre méthode utilise les résultats de l'étape n pour réaliser l'étape $n + 1$ et ajoute les nouveaux points repères appariés entre les deux visages. À une étape n donnée, notre algorithme se déroule comme suit :

- calcul d'un descripteur appliqué sur chaque pixel des deux cartes de hauteur et obtention des images traitées ;
- segmentation par montée des eaux (*watershed segmentation*) des images traitées [BL79] ;

- sélection des points d'intérêt (graines issues de la segmentation) ;
- test de mise en correspondance des points d'intérêt et ajout des points vérifiant le test aux ensembles des points repères.

Segmentation par algorithme de montée des eaux

Nous utilisons la segmentation par montée des eaux, qui consiste à considérer une image en niveau de gris comme une surface topographique que l'on immerge dans l'eau [MW99]. À chaque fois qu'un minimum local est immergé, une région se crée et s'agrandit au fur et à mesure de l'immersion. Quand deux régions se rencontrent, une digue est créée, empêchant ainsi leur fusion⁴. À la fin de la segmentation, chaque pixel appartient à une région qui correspond à un bassin de rétention d'eau, le minimum local de chaque région est conservé comme représentant de la région et nous l'appelons graine. Ce sont ces graines qui seront candidates à la mise en correspondance et ajoutées à la liste des points repères si l'appariement est validé. Nous appliquons la segmentation sur les images obtenues à l'issue de l'étape de traitement qui applique les descripteurs aux cartes de hauteur. La Figure 3.5 illustre une coupe latérale d'une segmentation par montée des eaux.

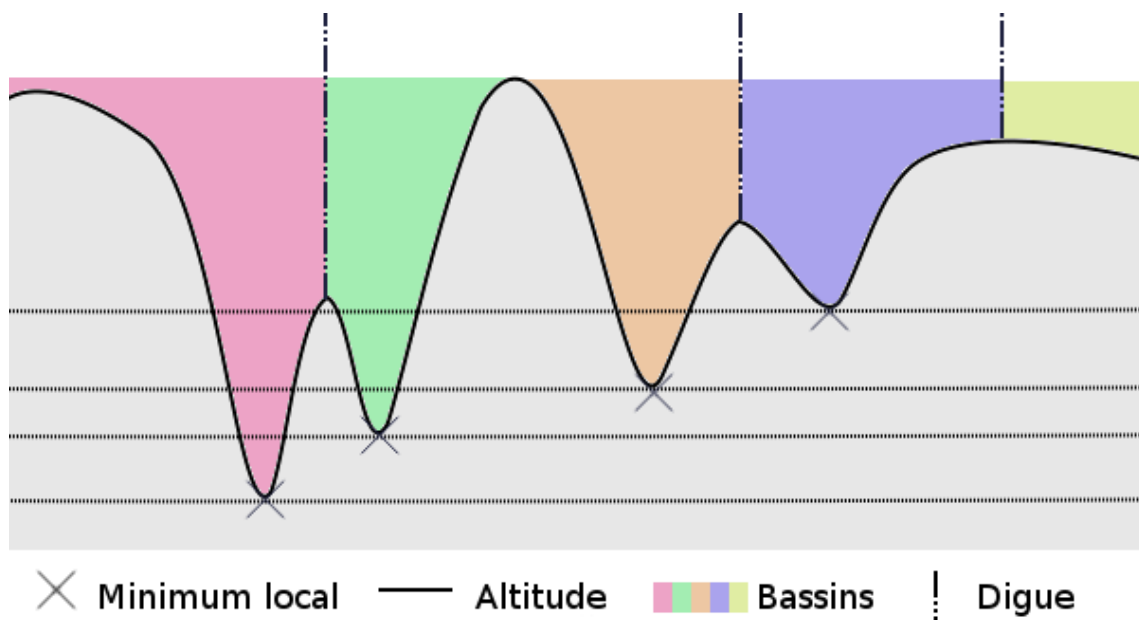


FIGURE 3.5 – Coupe latérale d'une segmentation par montée des eaux. L'eau monte progressivement jusqu'à rencontrer un minimum local, un bassin est alors constitué et se remplit parallèlement à la montée des eaux. Lorsque deux bassins se rejoignent, une digue est construite empêchant les deux régions de fusionner.

4. En pratique, des critères peuvent être définis pour déterminer si oui ou non les deux régions sont fusionnées.

Une segmentation est définie par une méthodologie qui s'applique à un attribut (un descripteur dans notre cas). Cet attribut peut représenter une valeur de niveau de gris, une couleur, une texture, un gradient, etc. La méthode incrémentale que nous proposons permet de changer de descripteur à chaque étape. L'utilisation de différents descripteurs permet de faire ressortir des caractéristiques spécifiques du visage comme par exemple des régions à fortes courbures ou des régions où la symétrie est importante, etc. Ainsi, l'utilisation de plusieurs descripteurs offre un large choix de caractéristiques à extraire, et les faiblesses des uns peuvent être comblées par les avantages des autres. Nous proposons dans cette partie différents traitements à appliquer aux cartes de hauteur. Le choix de ces traitements est un problème classique en vision qui consiste à choisir des descripteurs pertinents pour la mise en correspondance d'images ou de surfaces [ZF03, SMFF07]. Étant donné que notre algorithme est basé sur une séquence de segmentations/mises en correspondance, autant de descripteurs que nécessaire peuvent être utilisés. Néanmoins, le choix de ceux-ci doit être ordonné de façon à commencer par les descripteurs qui extraient les caractéristiques globales pour terminer par les caractéristiques plus locales. La Figure 3.8 illustre un exemple d'une séquence où les régions segmentées sont d'abord peu nombreuses et larges puis s'affinent. Le calcul des descripteurs passe par des traitements. Dans notre cas, nous appliquons dans un premier temps un filtre gaussien qui permet de segmenter les images en peu de régions et donc d'extraire les extrema globaux comme le nez ou le front. Il est également possible d'extraire des extrema plus locaux comme ceux localisés sur les coins des yeux. Un filtre laplacien permet d'extraire les arêtes, en général sur le contour du visage ou du nez. Comme le visage possède des régions symétriques, un descripteur permettant de mettre en avant cette propriété peut être également efficace. La liste n'est pas exhaustive et bien sûr d'autres descripteurs peuvent être introduits pour enrichir la mise en correspondance. En pratique, la séquence montrée dans la Figure 3.8 est celle que nous avons utilisée pour nos résultats (voir la Figure 3.9) dont voici quelques descripteurs utilisés :

Filtre gaussien : opérateur Canny Deriche [Can86, Der90] : il permet d'extraire les extrema locaux ou globaux selon la taille du filtre et la région explorée.

Filtre laplacien : descripteur de courbure :

$$F(x, y) = \frac{f(x+h, y) + f(x-h, y) - 2 \times f(x, y)}{h^2} + \frac{f(x, y+h) + f(x, y-h) - 2 \times f(x, y)}{h^2} \quad (3.20)$$

Il permet d'extraire des points situés sur les zones de contour où les variations de courbure sont importantes comme sur la frontière du visage, le contour du

nez, etc.

Opérateur de symétrie : il détermine si l'axe des segments verticaux et horizontaux passant par ce point est un axe de symétrie. Pour cela, l'opération consiste à soustraire l'intensité des valeurs situées dans le rectangle en haut (resp. à gauche) à l'intensité des valeurs situées dans le rectangle opposé en bas (resp. à droite) (voir la Figure 3.6). Il ne s'agit pas d'une symétrie ponctuelle, le pixel correspond au centre du segment de la symétrie axiale. Exemple de calcul pour la symétrie horizontale avec r et $r \times 2$ les longueurs du rectangle :

$$S(x, y) = \sum_{i=-r}^r \sum_{j=1}^r (f(x+i, y+j) - f(x+i, y-j)) \quad (3.21)$$

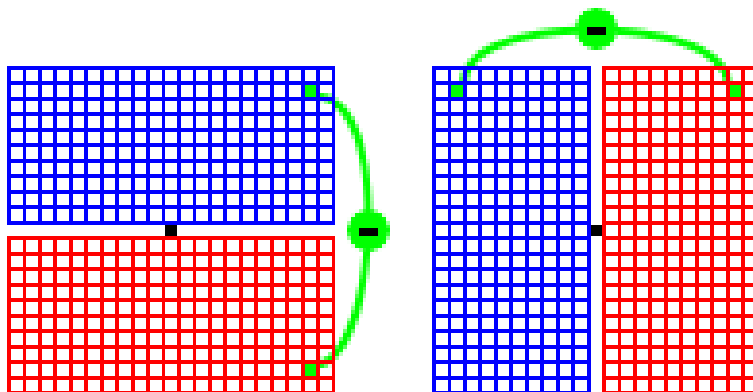


FIGURE 3.6 – Illustration de l'opérateur de symétrie horizontale et verticale. Les pixels noirs représentent les pixels sur lesquels nous souhaitons appliquer les descripteurs. La valeur des pixels des zones rouges sont soustraites aux valeurs des pixels des zones bleues correspondants.

De la segmentation à la mise en correspondance

Notre algorithme d'appariement nécessite une phase d'initialisation, pour l'automatiser, nous effectuons cette phase en positionnant des points repères sur le pourtour boîte englobante à intervalles réguliers. Ensuite, après chaque segmentation, nous possédons un ensemble de graines et de régions pour chacune des deux images. L'étape suivante consiste à déterminer les correspondances qui peuvent exister entre les régions et/ou graines d'un visage à l'autre. Lorsque deux graines sont mises en correspondance, elles sont alors ajoutées aux ensembles de points repères. Considérons que k étapes de segmentations/mises en correspondance ont été effectuées. Nous possédons deux ensembles \mathcal{R}_s^k et \mathcal{R}_t^k de points repères mis en correspondance, c'est-à-dire que le $i^{\text{ème}}$ point $r_s^i \in \mathcal{R}_s^k$ est associé au $i^{\text{ème}}$ point $r_t^i \in \mathcal{R}_t^k$. Nous

utilisons ces points pour générer deux interpolations par *RBF* que nous notons $\mathcal{RBF} = \mathcal{RBF}(\mathcal{R}_s^k \rightarrow \mathcal{R}_t^k)$ et $\mathcal{RBF}^{-1} = \mathcal{RBF}(\mathcal{R}_t^k \rightarrow \mathcal{R}_s^k)$, l'une étant la transformation entre l'ensemble des points repères du visage source vers le visage cible et l'autre la transformation entre l'ensemble des points repères du visage cible vers le visage source. Soit \mathcal{P}_s^{k+1} et \mathcal{P}_t^{k+1} les graines obtenues après la segmentation de l'étape $k + 1$. Une graine $p_s \in \mathcal{P}_s^{k+1}$ est sélectionnée comme point repère si et seulement si elle peut être associée à une graine $p_t \in \mathcal{P}_t^{k+1}$. Soit W_s une région (un bassin de la segmentation) de la source de graine p_s et W_t une région de la cible de graine p_t . Le test de correspondance est vérifié si et seulement si :

$$\begin{aligned} \mathcal{RBF}(p_s) &\in W_t \\ \mathcal{RBF}^{-1}(p_t) &\in W_s \end{aligned} \quad (3.22)$$

Ce test consiste donc à vérifier que l'interpolation de la graine source p_s dans l'espace cible tombe dans la région cible segmentée W_t dont la graine p_t interpolée dans l'espace source tombe dans la région W_s de la graine source. Le critère de correspondance entre deux graines est illustré sur la Figure 3.7. Ce test est effectué entre toutes les paires possibles de graines sources et de graines cibles. Les paires de graines satisfaisant ce critère sont promues points repères puis ajoutées à leur ensemble de points repères respectif. Le calcul de ce critère est rapide et permet de trouver facilement les régions similaires d'un visage à l'autre. D'après nos tests, il semble suffisant et il n'est donc pas utile de considérer d'autres critères.

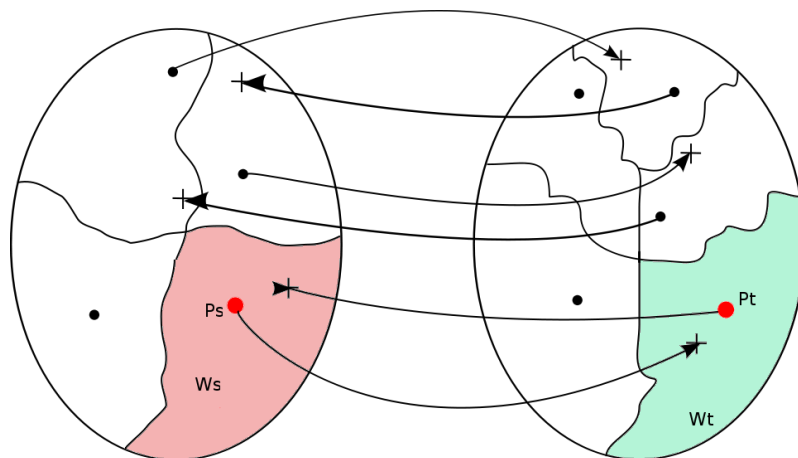


FIGURE 3.7 – Exemple du critère de sélection des graines. Ce test vérifie que l'interpolation de la graine source p_s dans l'espace cible tombe dans la région cible segmentée W_t dont la graine p_t interpolée dans l'espace source tombe dans la région W_s de la graine source

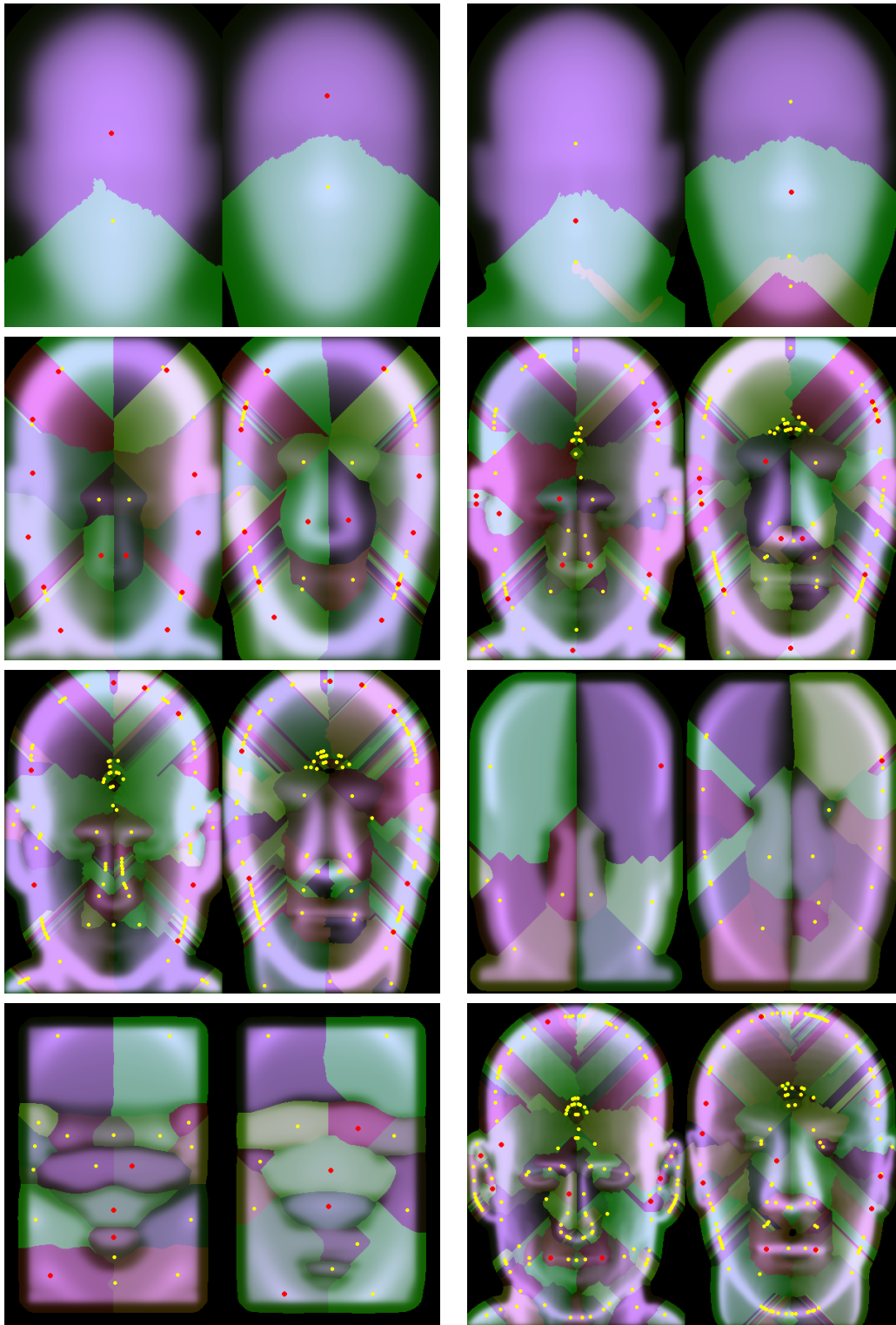


FIGURE 3.8 – Séquence de segmentations avec des traitements différents. Les points jaunes sont les graines non conservées, les points rouges sont les graines conservées comme points repères.

3.4.1.2 Transfert des points de contrôle

A l'issue de la séquence de segmentations/mises en correspondance, nous avons donc deux ensembles de points repères mis en correspondance. C'est grâce à eux que nous allons pouvoir transférer les points de contrôle, étape préliminaire pour toute paramétrisation du visage par *skinning*. Étant donné que les points de contrôle sont définis dans l'espace 3D du visage source et que les points repères le sont dans l'espace 2D des cartes de hauteur, la première étape consiste à projeter les points de contrôle dans l'espace de la carte de hauteur. Cette carte a été obtenue grâce au *Z-Buffer* qui utilise les paramètres de la caméra virtuelle (*matrice modelview*, la *matrice projection* et la *matrice viewport*). Nous utilisons ces mêmes paramètres pour projeter les points de contrôle. Une fois les coordonnées 2D des points de contrôle sources obtenues, nous calculons une nouvelle interpolation par *RBF* entre l'espace défini par les points repères du visage source et l'espace défini par les points repères du visage cible. En appliquant cette transformation aux points de contrôle sources, nous obtenons les points de contrôle cibles en 2D. Une projection inverse nous permet finalement d'obtenir leur position 3D. La Figure 3.9 présente quelques exemples du transfert de points de contrôle obtenus automatiquement par notre méthode.

3.4.2 Transfert des poids os-sommet

Une fois les points de contrôle du visage cible obtenus, il faut transférer les autres informations nécessaires à l'animation par *skinning*. Il s'agit de l'influence des points de contrôle sur chacun des sommets du maillage. L'idée consiste à effectuer un recalage dense des deux surfaces 3D : après avoir mis en correspondance les deux visages grâce à deux ensembles de points repères qui nous ont permis de transférer les points de contrôle, nous allons chercher à déterminer à quel point de la surface du visage source chaque sommet du visage cible correspond. Nous utilisons les points de contrôle pour initialiser ce recalage dense.

3.4.2.1 Recalage dense de surfaces

La correspondance complète consiste à trouver à quel point p_s de la surface du visage source correspond le sommet v_t du maillage du visage cible, sachant que p_s n'est pas nécessairement un sommet mais un point sur la surface d'un triangle. Avant d'effectuer le recalage, une étape préliminaire est effectuée. Elle consiste à découper le maillage cible pour ne conserver que les sommets appartenant au visage. Cette opération permet d'optimiser les temps de calcul et d'éviter des erreurs de recalage liées au fait que les sommets du crâne et de l'arrière de la tête ne sont pas influen-

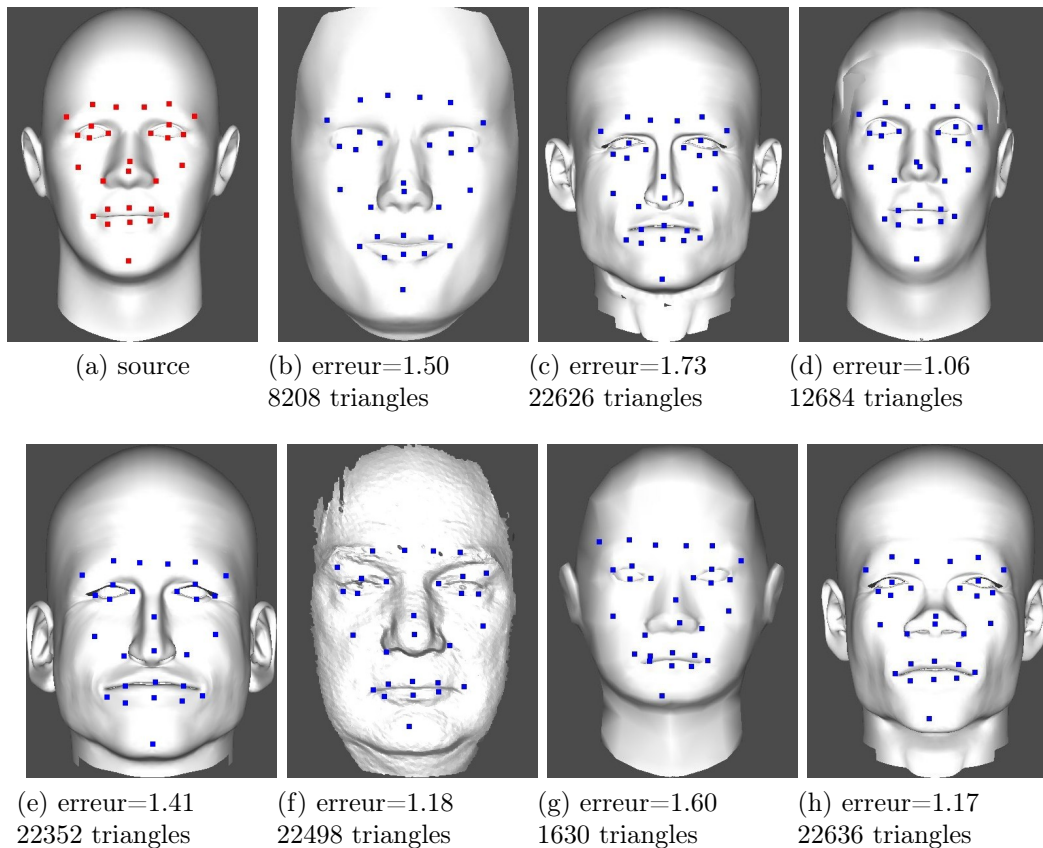


FIGURE 3.9 – Exemples de transferts de points de contrôle effectués automatiquement grâce à notre séquence incrémentale de segmentations/mises en correspondance.

cés par les points de contrôle du visage. Ces sommets exclus du recalage sont alors influencés par l'os articulant la tête entière. Le découpage est effectué automatiquement en utilisant la distance qui sépare chaque sommet du point de contrôle qui lui est le plus proche. Si cette distance est supérieure au critère choisi, alors le sommet est exclu du visage. Expérimentalement, il s'avère que la distance euclidienne entre les deux yeux du visage nous permet d'obtenir les résultats escomptés.

Nous utilisons une interpolation par *RBF* pour effectuer la première étape de notre recalage dense. L'interpolation est cette fois basée sur les coordonnées 3D des points de contrôle et est calculée dans le sens visage cible vers visage source. Étant donné que les points de contrôle ne permettent pas de définir complètement la surface d'un visage, mais uniquement quelques points clés liés à l'animation, la déformation par *RBF* de chaque sommet du maillage cible ne garantit pas l'obtention d'une surface strictement identique à celle du visage source. La Figure 3.10d montre que la surface du visage cible (surface maillée) ne correspond pas exactement à la surface du visage

source (surface pleine). La déformation par *RBF* permet principalement de mettre en correspondance la surface proche des points de contrôle. Pour corriger ce biais et obtenir un recalage dense, nous appliquons une projection des sommets cibles interpolés sur la surface du visage source. Étant donné que la géométrie de la tête se rapproche le plus de celle de la sphère, nous appliquons une projection sphérique dont le centre de projection est l'os racine de la tête. Cet os est fréquemment positionné au centre de la tête, proche de son centre de gravité. Le centre de la boîte englobante des sommets de la tête est un choix efficace expérimentalement. La Figure 3.10e illustre le recalage dense du visage cible sur le visage source.

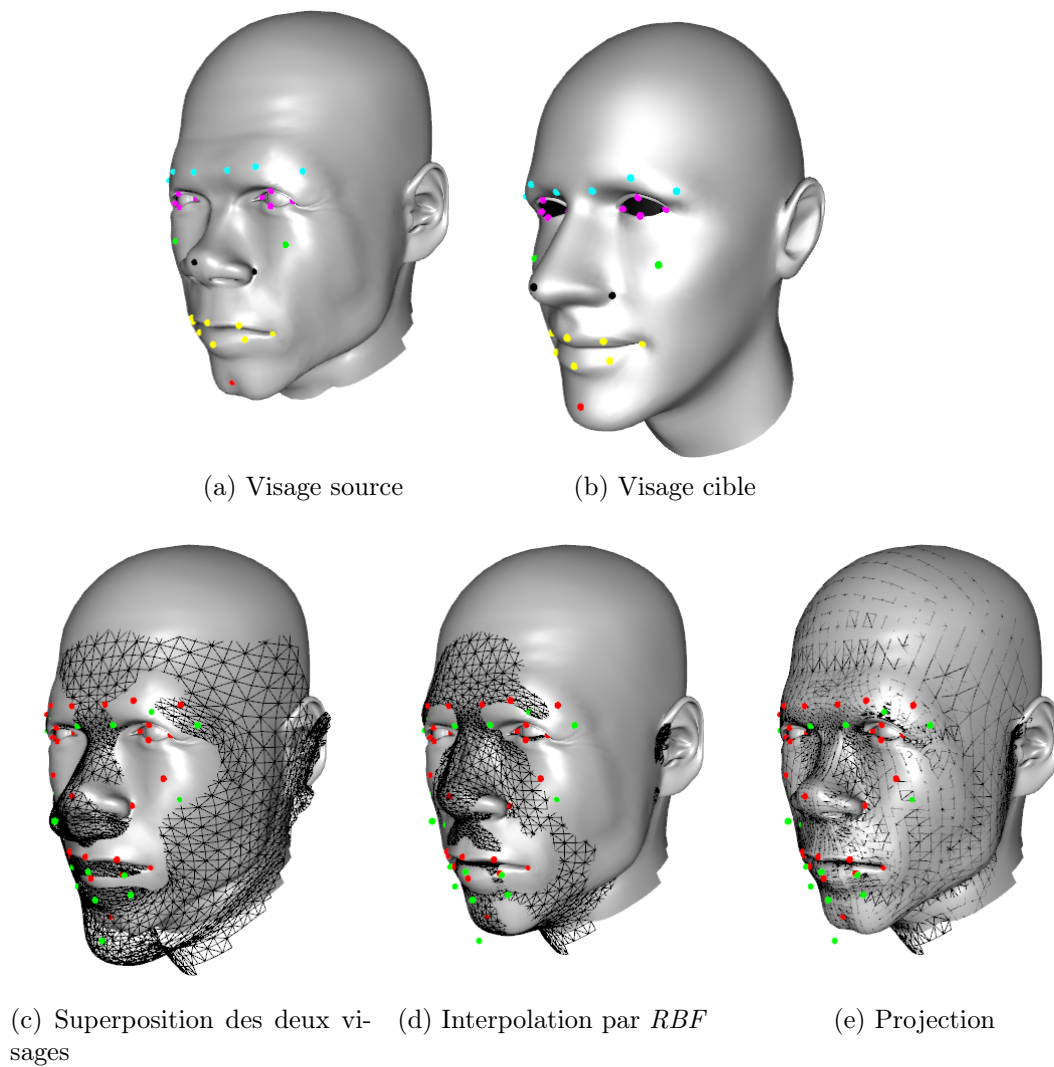


FIGURE 3.10 – Recalage dense de surface.

3.4.2.2 Affectation des influences

Une fois la projection effectuée, chaque sommet v_t su visage cible est positionné sur un point p_s appartenant au triangle $T_s = (v_s^1, v_s^2, v_s^3)$ du visage source. L'influence $w_P(p_s)$ d'un point de contrôle P sur le point p_s , et donc sur le sommet cible v_t , est calculée en fonction des distances le séparant des 3 sommets du triangle. Soit $w_P(v_s^i)$ l'influence de P sur v_s^i . Cette interpolation est effectuée en utilisant une pondération inverse à la distance (*PID*) :

$$w_P(v_t) = w_P(p_s) = \sum_{i=1}^3 \left(\frac{d_{p_s}(v_s^i)}{\sum_{k=1}^3 d_{p_s}(v_s^k)} w_P(v_s^i) \right) \quad (3.23)$$

avec

$$d_{p_s}(v_s^i) = \frac{1}{\|\vec{v}_s^i - \vec{p}_s\|} \quad (3.24)$$

Le choix de la pondération inverse à la distance nous permet d'obtenir une interpolation spatiale en fonction de la distance. Plus p_s est proche d'un sommet v_s^i , plus ce sommet est influent dans la pondération.

Certains problèmes peuvent être détectés au niveau de la bouche. En effet, la projection d'un point situé sur la lèvre supérieure du visage cible peut être recalé sur la lèvre inférieure du visage source car les points de contrôle constituent une information insuffisante pour effectuer un recalage précis sur cette zone. Ceci produit alors une influence incorrecte qui génère des incohérences pendant l'animation. Cet aspect n'intervient pas au niveau de l'ouverture des yeux si l'on considère que les visages sont dans une position neutre (les yeux ouverts). Pour résoudre ce problème, un traitement particulier est réalisé sur chaque sommet influencé : soit par un point de contrôle de la lèvre supérieure ; soit par un point de contrôle de la lèvre inférieure. Ce traitement consiste à calculer les distances géodésiques qui lient le sommet influencé avec le point de contrôle et son point opposé par rapport à la bouche. Si la distance avec le point opposé est inférieure à celle avec le point courant, alors les influences de ces deux points de contrôle sont interverties. Par exemple, imaginons que nous avons obtenu pour le sommet v_t une influence du point de contrôle de la lèvre supérieure (point de contrôle défini par 8.1 dans la norme *MPEG-4*). Nous souhaitons vérifier la distance géodésique qui sépare v_t des points 8.1 et 8.2 (lèvre inférieure). Si la distance avec 8.2 est inférieure de la distance avec 8.1, alors nous inversons les valeurs de pondération de ces deux points de contrôle :

$$w_{8,1}(v_t) = w_{8,2}(v_t) \quad (3.25)$$

$$w_{8,2}(v_t) = w_{8,1}(v_t) \quad (3.26)$$

La Figure 3.11 montre un exemple de transfert de pondération obtenu suite à notre recalage dense.

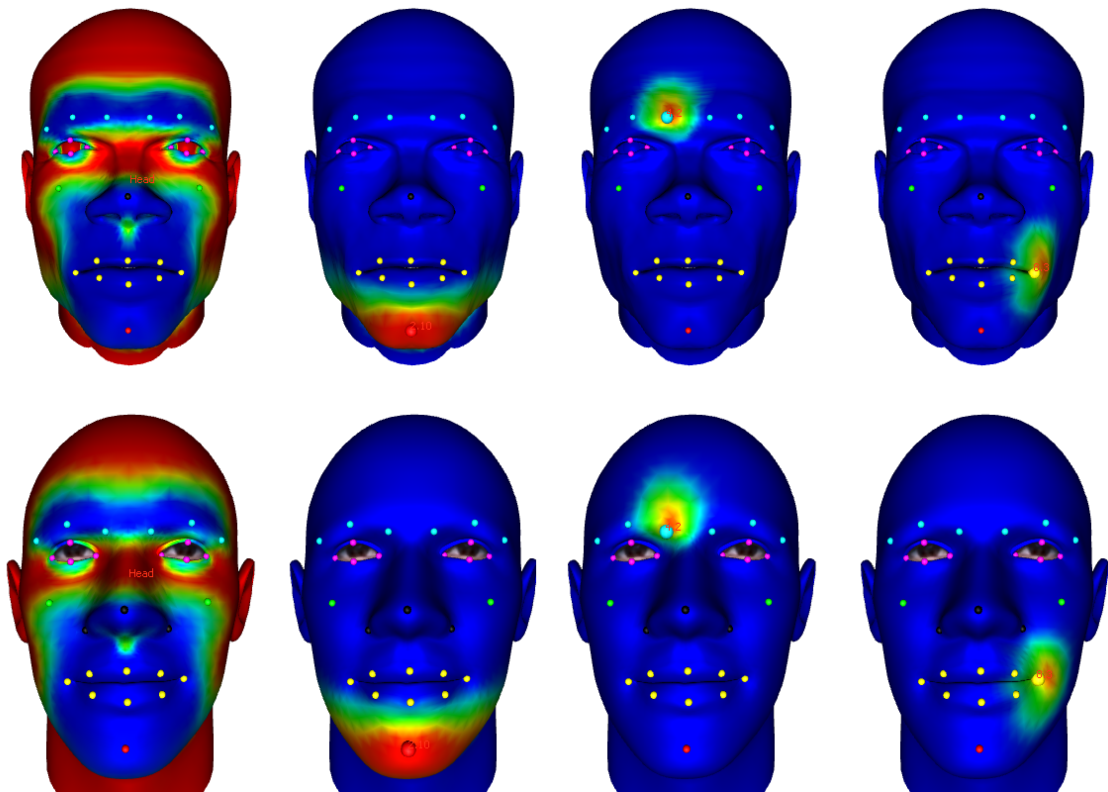


FIGURE 3.11 – Résultat du transfert des poids du *skinning*. Le visage du haut est le visage source, le visage cible (en bas) a été paramétrisé automatiquement par transfert à partir de la paramétrisation du visage source.

3.5 Résultats

Nous avons testé notre méthode de recalage sur dix modèles 3D différents qui possèdent chacun une taille, une morphologie, et une topologie de maillage propre. Nous supposons que les visages sont correctement positionnés et orientés : regard pointé sur l'axe \vec{z} , haut de la tête sur l'axe \vec{y}). Les modèles sont pour certains issus de visages numérisés, d'autres ont été créés manuellement par un graphiste, et d'autres ont été générés de façon procédurale à l'aide de logiciels spécifiques à la modélisation du corps humain. Pour démontrer la capacité de notre approche à faire abstraction

de la résolution du maillage, les modèles 3D possèdent entre un millier et plusieurs dizaines de milliers de triangles. Dans un souci de généralité, nous avons également testé le recalage sur des visages de personnages *cartoons* (voir la Figure 3.12). Il est à noter que les visages cartoons sont extrêmement difficiles à apparier en pratique à cause de leur non-hétérogénéité.

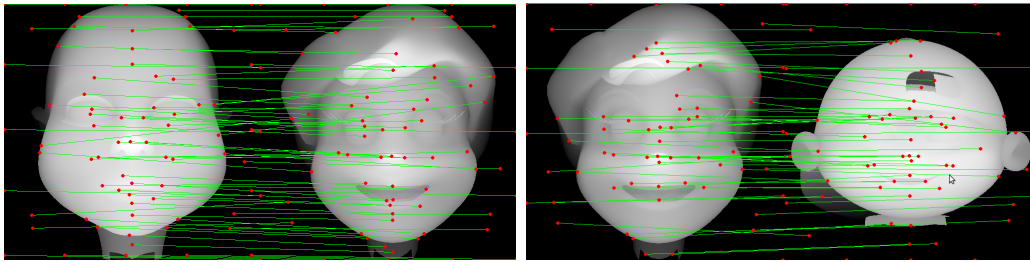


FIGURE 3.12 – Exemple de mise en correspondance sur des visages de type *cartoon*.

Pour évaluer la qualité de la mise en correspondance des points de contrôle, nous avons positionné manuellement 20 points de contrôle sur les visages cibles. Pour être capable de comparer les résultats obtenus sur des têtes ayant des tailles différentes, nous avons normalisé les résultats par la taille de la boîte englobante des visages. L'erreur que nous calculons est la somme des distances entre les points de contrôle obtenus automatiquement et ceux de références déterminés manuellement. Notre but est de proposer une aide à la création de la paramétrisation du visage, pour respecter la part de créativité de l'utilisateur, l'intervention de celui-ci est toujours possible si le besoin s'en fait ressentir sur le positionnement des points de contrôle sans perturber le reste de la méthode.

Le calcul de la carte de hauteur correspond au rendu d'une image du visage 3D et à la lecture des données contenues dans le *Z-Buffer* de la carte graphique. Cette première étape est très rapide. La mise en correspondance des points de contrôle est dépendante de la résolution de la carte de hauteur et du nombre d'étapes de notre mise en correspondance incrémentale. Nous avons utilisé des cartes de hauteur de dimensions 512×512 pixels avec la séquence de segmentation présentée sur la Figure 3.8. La mise en correspondance dure environ 20 secondes sur un processeur Athlon X2 3800+ avec 2GO de mémoire vive. Les temps de calcul du recalage dense des deux surfaces est quant à lui dépendant de la résolution des maillages utilisés, cette étape dure environ 30 secondes pour un maillage de 12000 triangles. La Figure 3.13 montre des extraits d'animations obtenues grâce à notre méthode de capture et de transfert présentée dans le Chapitre 4. La deuxième ligne correspond au visage source qui a été paramétrisé manuellement (plusieurs heures de travail),

et les deux lignes suivantes présentent le visage cible dont la paramétrisation a été obtenue automatiquement par notre méthode en quelques secondes sans intervention manuelle.

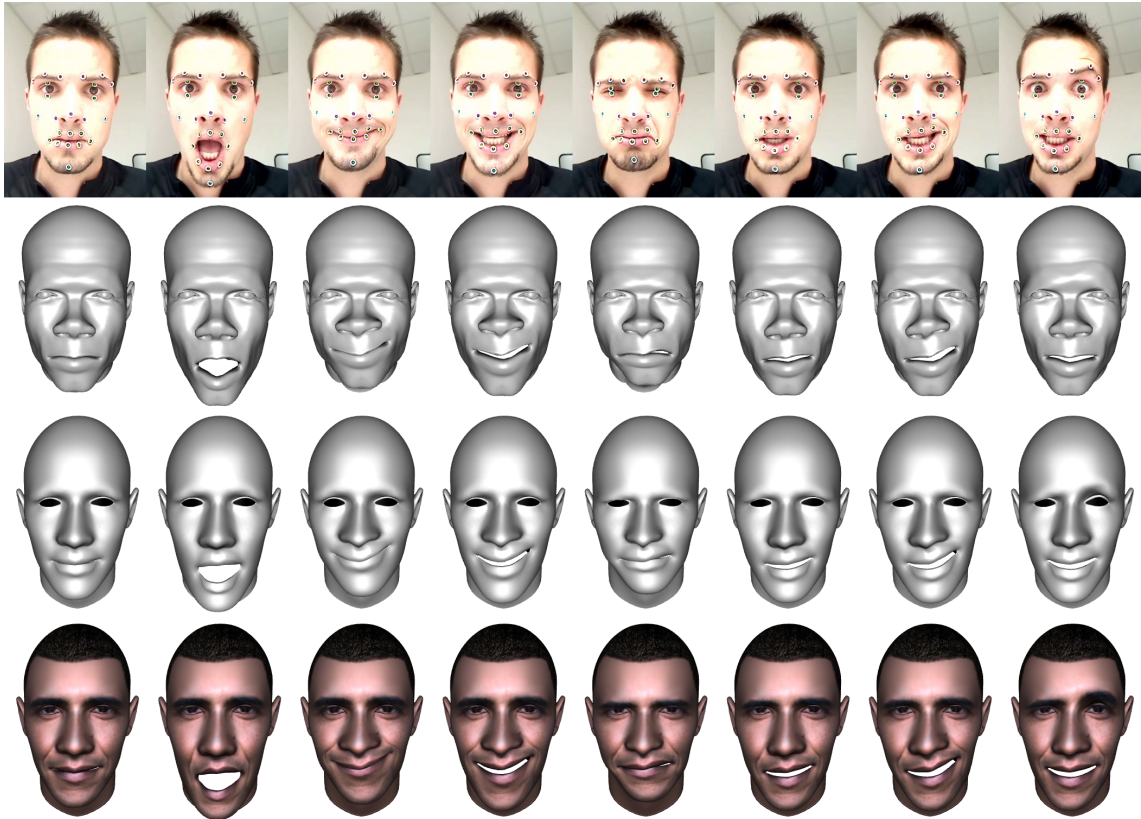


FIGURE 3.13 – Extraits d’animation des visages source et cible. La paramétrisation du visage cible (en bas) a été transférée automatiquement.

Étant donné que nous souhaitons proposer un algorithme aussi générique que possible, nous n’avons pas utilisé d’information de textures. Nous sommes bien conscient que cet attribut pourrait grandement aider au recalage au niveau des sourcils et de la bouche, cependant, tous les modèles ne sont pas texturés et/ou ne possèdent pas de coordonnées textures. Dans le cas spécifique où la mise en correspondance est employée au visage, des heuristiques comme celles proposées par Noh *et al.* pourraient compléter notre méthode, qui serait à partir de ce moment, complètement spécifique au visage humain [NN01].

3.6 Conclusion

Nous avons proposé deux étapes pour la paramétrisation du visage basée sur une déformation de surface à base de points de contrôle. La première permet d’accélérer

le travail du graphiste et serait une alternative idéale aux enveloppes d'influence classiques utilisées pour le corps dans les logiciels d'animation actuels. Elle offre un gain de temps non négligeable malgré le fait qu'elle nécessite l'intervention de l'utilisateur pour le positionnement des points de contrôle. Disposant d'une première paramétrisation réalisée sur un visage, notre deuxième méthode permet de la réutiliser sur de nouveaux personnages. Elle consiste à transférer automatiquement la paramétrisation du visage source à un visage cible grâce à deux étapes de recalage. Pour laisser place à la créativité et aux exigences de l'utilisateur, il lui est possible d'intervenir et de modifier la paramétrisation obtenue : paramétrage des ellipsoïdes, positionnement des points de contrôle ou affinement des influences point de contrôle/sommet. Ces deux approches sont complémentaires : la première permet d'accélérer la réalisation d'une paramétrisation et la seconde offre la possibilité de réutiliser la paramétrisation créée sur de nouveaux personnages. Les résultats obtenus prouvent la viabilité de ce système basé sur une représentation par cartes de hauteur des visages qui rend la méthode plus rapide et flexible. Une partie des travaux présentés dans ce chapitre a été publiée dans des conférences internationales avec comité de lecture : « Feature Points Based Facial Animation Retargetting » dans VRST'08 (*15th ACM Symposium on Virtual Reality Software and Technology*) [DMB08], et « Easy Rigging of Face by Automatic Registration and Transfer of Skinning Parameters » dans ICCVG'10 (*International Conference on Computer Vision and Graphics*) [DMOB10].

Chapitre 4

Transfert d'animation et évaluation

Sommaire

4.1 Introduction	71
4.2 Transfert d'animation	72
4.2.1 Transfert d'expressions	73
4.2.2 Transfert partiel	74
4.3 Transfert pour la capture de mouvement	74
4.4 Résultats et évaluation	75
4.5 Conclusion	78

4.1 Introduction

La création d'une animation faciale se compose de deux phases principales, la première consiste à paramétrer le visage, c'est-à-dire que le modèle 3D statique devient « animable ». La seconde consiste à créer l'animation proprement dite, c'est-à-dire à faire varier les paramètres de la paramétrisation pour obtenir une animation réaliste et de qualité. Cette étape peut être réalisée en créant manuellement des poses clés qui sont ensuite interpolées pour obtenir l'animation. Cependant, la dynamique du visage est complexe et il est difficile d'obtenir un résultat satisfaisant et réaliste par ce processus. C'est dans le but de simplifier cette tâche et permettre de générer des animations convaincantes que la capture de mouvement du visage a été mise à contribution avec le système précurseur *PDFA* [Wil90]. Elle s'est depuis largement démocratisée et un grand nombre d'animations faciales exploitées de nos jours est générée par capture de mouvement. Cependant, tout comme pour la paramétrisation, l'animation est une tâche exigeante. Les systèmes de capture de mouvement nécessitent souvent la pose et l'identification de très nombreux marqueurs sur le visage de l'acteur. De plus, les données obtenues ne peuvent être appliquées directe-

ment qu'à un visage virtuel dont la géométrie est identique à celle de l'acteur. Une animation faciale transférée d'un visage à un autre sans traitement particulier entraînera une dégradation visuelle de l'animation qui pourra alors paraître accentuée ou atténuée par endroit, provoquant une dégradation du réalisme et de l'expressivité. Ainsi le transfert d'une animation existante d'un visage vers un autre visage dont les morphologies sont différentes est un problème encore ouvert.

Soulignons que le transfert d'animation a d'abord été proposé pour l'animation du corps [Gle98]; les travaux sur le transfert de l'animation faciale sont arrivés plus tard. Noh *et al.* ont proposé un système spécifique au visage [NN01]. La méthode qu'ils présentent consiste à effectuer un recalage dense entre les deux maillages des visages source et cible, puis de transférer les vecteurs d'animation de chaque sommet après les avoir adaptés en taille et en orientation. Un vecteur d'animation correspond au déplacement d'un sommet pour une expression par rapport à sa position lorsque le visage est dans une configuration neutre. Le recalage est effectué par une interpolation par *RBF*, dont l'initialisation a été effectuée à partir de points repères. Bien qu'efficace, cette technique possède quelques contraintes importantes qui peuvent rendre son utilisation délicate. Tout d'abord il n'est pas possible de l'utiliser pour animer un visage paramétrisé, aussi bien par *blendshape* que par *skinning*, car elle travaille directement sur le maillage et ses déformations. Ensuite elle est globale et très peu flexible, il est impossible d'intervenir dans le transfert et enfin le temps réel est difficilement envisageable à cause du calcul à chaque frame et pour chaque sommet de son vecteur déplacement. Enfin, elle est très spécifique et son utilisation est réservée uniquement aux transferts d'animation de maillages 3D vers d'autres maillages 3D.

La technique que nous proposons dans ce chapitre permet de résoudre ces problèmes : elle est à la fois adaptée au temps réel et à l'animation par *skinning* ; elle est flexible et laisse la possibilité à l'utilisateur d'intervenir dans le transfert ; et enfin elle permet d'exploiter directement des données issues de la capture de mouvement.

4.2 Transfert d'animation

La méthode de transfert que nous proposons est basée sur l'interpolation par des *RBF*. Elle se décompose en deux étapes, la première est une phase d'initialisation et la seconde est une phase de transfert qui est calculée en temps réel pour chaque image de l'animation. La Figure 4.1 illustre le schéma général de la technique.

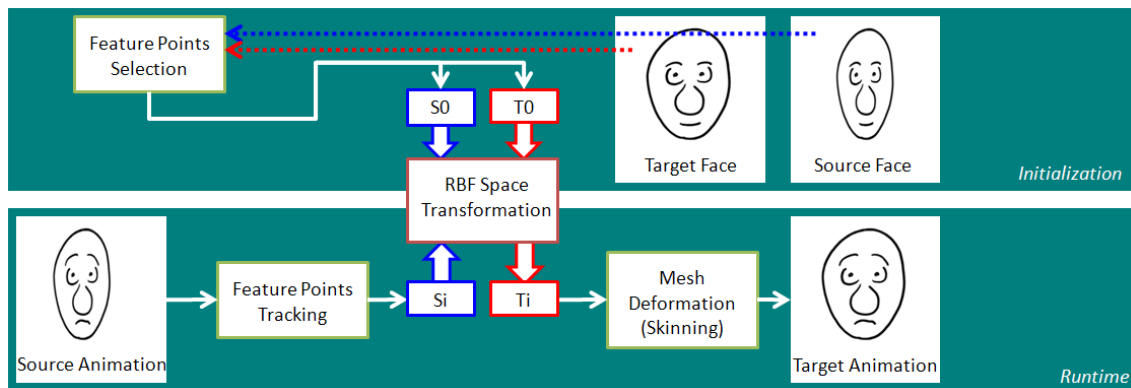


FIGURE 4.1 – Vue d’ensemble de notre méthode de transfert. Une première phase d’initialisation effectue l’initialisation d’une interpolation par *RBF* à partir des points de contrôle sources et cibles. Puis, en temps réel et pour chaque frame de l’animation, les points de contrôle sources sont transformés pour obtenir les nouveaux points de contrôle cibles, et ainsi, déformer le visage cible.

4.2.1 Transfert d’expressions

Les modèles à base de *RBF* permettent comme nous l’avons vu au Chapitre 3 de calculer une approximation qui permet de situer un point de l’espace du visage source dans l’espace du visage cible. Chacun des espaces est défini par les points de contrôle qui représentent les points caractéristiques liés à l’animation faciale. Pour transférer une expression, nous allons travailler sur la position des points de contrôle, le but étant de transférer leurs déplacements du visage source au visage cible. La phase d’initialisation de l’interpolation par *RBF* est calculée sur les points de contrôle des visages en position neutre, c’est-à-dire sans expression (voir la Figure 4.2a). Cette étape n’est réalisée qu’une seule fois pour les deux visages. Pour transférer une expression, nous appliquons l’interpolation aux points de contrôle déformés pour obtenir leurs positions dans l’espace du visage cible (voir la Figure 4.2b). Les sommets du maillage sont alors déplacés en fonction des mouvements des points de contrôle grâce à la paramétrisation par *skinning*. L’interpolation est appliquée à chaque frame de l’animation et permet ainsi d’obtenir l’animation du visage cible dont le déplacement des points de contrôle a été adapté à sa propre morphologie. L’initialisation du transfert ne nécessite pas beaucoup de ressources, le calcul le plus important consiste à inverser une matrice carrée de dimension le nombre de points de contrôle. Pendant l’animation, la complexité est linéaire et les calculs limités permettent d’animer plusieurs visages en temps réel ce qui est important dans une application dont les ressources sont partagées comme les jeux vidéo.

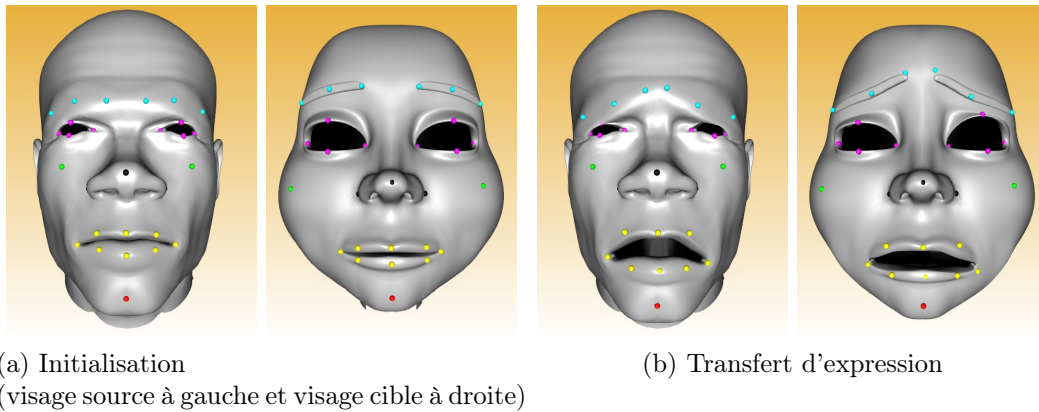


FIGURE 4.2 – Exemple de transfert d’une expression. À gauche, les deux visages avec des expressions neutres utilisées pour l’initialisation du transfert. À droite, le visage source est doté d’une expression qui est transférée au visage cible par notre technique.

4.2.2 Transfert partiel

Notre système offre également un avantage majeur : il est possible de ne transférer qu’une partie de l’animation. En effet, étant donné que le transfert est basé directement sur les points de contrôle, il devient possible de n’appliquer le transfert qu’à un certain nombre de points, n’utilisant qu’une partie de l’expression source. Il est ainsi aisé de n’appliquer l’animation qu’à une partie du visage comme par exemple au niveau des yeux ou de la bouche. Ce transfert local n’interfère pas sur le réalisme du visage puisqu’aucun artefact visuel n’est produit si une partie du visage reste statique. Il est également possible d’animer un même visage cible à partir de deux animations sources, en appliquant chacune des animations à des zones différentes du visage. La Figure 4.3 illustre différentes possibilités du transfert partiel offert par notre méthode.

4.3 Transfert pour la capture de mouvement

Notre méthode de transfert peut également être utilisée au sein d’un système de capture de mouvement, pour adapter les données capturées depuis un visage réel sur le visage virtuel. Dans le cadre d’un système permettant d’obtenir les positions des points caractéristiques dans l’espace 3D et non dans le plan 2D d’une image (systèmes stéréoscopiques ou faisant intervenir plusieurs caméras), il s’agit d’un cas similaire à celui du transfert d’une animation 3D existante d’un visage à un autre. En revanche, notre technique permet également de transférer les données issues d’un système monoculaire, et donc en deux dimensions. Pour démontrer la faisabilité

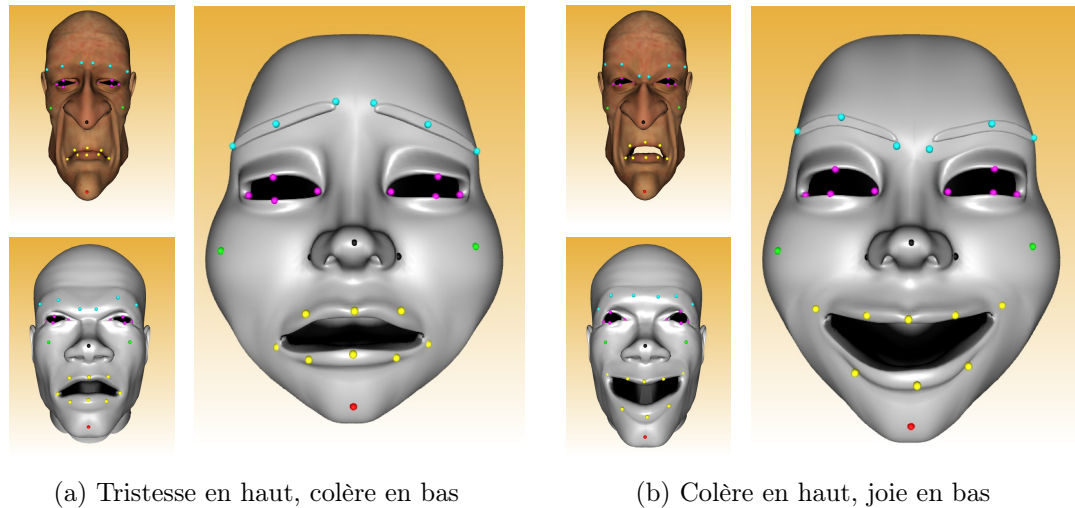


FIGURE 4.3 – Deux exemples de transfert partiel. Dans chaque cas, le visage source en haut à gauche transmet l’animation au niveau des sourcils et des yeux, et le visage source en bas à gauche transmet l’animation au niveau de la bouche et de la mâchoire.

d’un transfert basé sur des données en deux dimensions, nous avons implémenté une application de suivi de marqueurs. Nous avons utilisé l’algorithme pyramidale du traqueur de points caractéristiques de Lucas Kanade [Bou02]. Nous avons positionné une vingtaine de marqueurs sur le visage de l’acteur, filmé par une webcam grand public produisant des images dont la définition est de 640×480 pixels à une fréquence de 30 images par seconde. La Figure 4.4 illustre quelques extraits d’une séquence de transfert à partir du suivi de points en 2 dimensions. L’acquisition 2D impose quelques limitations : une grande partie des déformations faciales se produisent dans le plan 2D du visage et notre approximation nous permet donc d’appliquer une majorité de mouvements ; cependant, les mouvements perpendiculaires au visage, comme sont capables de produire les lèvres ou le menton, ne pourront être capturés et donc transférés au visage virtuel.

4.4 Résultats et évaluation

La Figure 4.5 montre la différence entre un transfert effectué à l’aide de notre méthode et un transfert direct qui se contente de normaliser le déplacement des points de contrôle en fonction de la taille des deux visages. La différence est notable et le transfert direct restitue des expressions atténuées. Le transfert par *RBF* permet au contraire de prendre en compte les différences de morphologies entre les deux visages. Les temps de calcul supplémentaires pour un transfert adapté sont constants

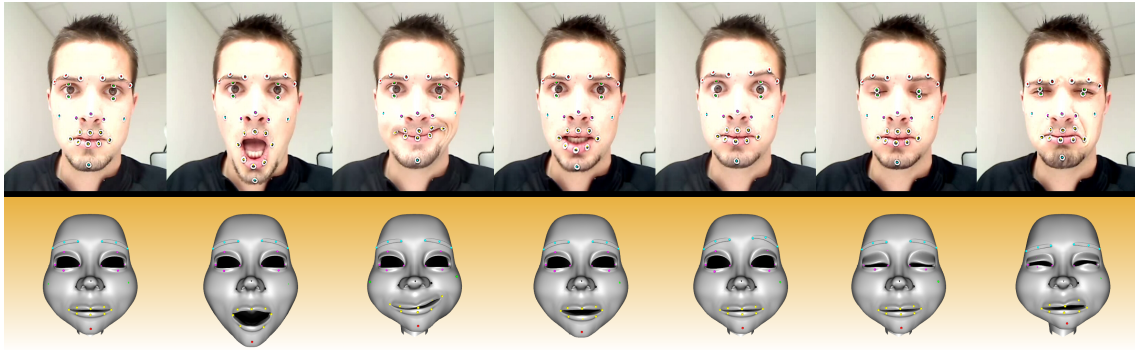


FIGURE 4.4 – Extraits d’une séance de capture pendant laquelle le suivi des marqueurs en 2 dimensions permet d’animer le visage cible virtuel en 3 dimensions.

et négligeables (3 multiplications d’une matrice de taille n^1 par un vecteur de taille n).

Pour comparer et évaluer les différents types d’interpolation par *RRF*, nous avons mis au point un protocole d’évaluation. Il consiste à transférer en « aller/retour » une expression d’un visage source à un visage cible, c’est-à-dire d’effectuer un transfert d’expression du visage source vers le visage cible, puis de re-transférer l’expression obtenue du visage cible sur le visage source, et enfin, de comparer le positionnement initial et final des points de contrôle sur le visage source. Nous avons utilisé les six expressions de base pour effectuer nos tests [EF78] (joie, tristesse, dégoût, surprise, peur et colère) ainsi que l’expression neutre. Pour une expression, le taux d’erreur est la somme des distances euclidiennes de chaque point entre la position attendue et la position obtenue :

$$\delta_e = \sum_{j=1}^n \|P_{j,init.}^{\vec{}} - P_{j,trans.}^{\vec{}}\| \quad (4.1)$$

avec n le nombre de points de contrôle, $P_{j,init.}^{\vec{}}$ la position du point de contrôle j dans l’expression initiale, et $P_{j,trans.}^{\vec{}}$ sa position après le double transfert. Nous avons effectué l’évaluation sur deux paires de visages, la première correspond à deux visages aux morphologies humaines réalistes, la seconde correspond à deux visages dont l’un est de type « cartoon », le but étant de tester le transfert sur deux visages aux différences importantes. Les résultats sont présentés sur les Figures 4.6 et 4.7, nous avons testé 5 fonctions différentes :

- **TPS** pour *thin plate spline* ($\phi(r) = r^2 \ln(r)$);
- **MQ** pour multiquadrique ($\phi(r) = \sqrt{r^2 + \sigma^2}$);
- **R** pour bi-harmonique ($\phi(r) = r$);

1. le nombre de points de contrôle



FIGURE 4.5 – Comparaison de transfert avec et sans interpolation sur 3 expressions différentes. À gauche le visage source, au milieu le visage cible avec notre méthode de transfert, et à droite le visage cible sans transfert adapté (le déplacement des points de contrôle est directement utilisé après normalisation selon la taille des deux visages).

- \mathbf{R}^3 pour tri-harmonique ($\phi(r) = r^3$);
- \mathbf{G} pour la fonction gaussienne ($\phi(r) = \exp(-r^2/2\sigma^2)$).

Les fonctions multiquadrique et gaussienne nécessitent un coefficient σ . Il permet de favoriser la localité de l'interpolation pour la multiquadrique [NN01] et il correspond à la largeur de la fonction pour la gaussienne. La valeur utilisée pour chaque point de

contrôle correspond à la distance minimale le séparant d'un autre point de contrôle :

$$\sigma_j = \min_{j \neq l} (\|\vec{P}_j - \vec{P}_l\|) \quad (4.2)$$

Les résultats montrent que la fonction $\phi(r) = r$ génère le moins d'erreurs sur nos deux séries de test et sur l'ensemble des expressions, hormis l'expression neutre pour laquelle les résultats sont meilleurs avec la gaussienne. Ceci traduit une erreur d'initialisation moindre pour la gaussienne, mais une faculté d'adaptation aux différences de morphologies meilleure pour la bi-harmonique. Les résultats visuels de ce double transfert avec cette fonction sont présentés sur la Figure 4.8.

	Neutre	Joie	Tristesse	Dégoût	Surprise	Peur	Colère
<i>TPS</i>	0.0085	0.0095	0.0084	0.0082	0.0098	0.0090	0.0083
<i>MQ</i>	0.0034	0.0063	0.0038	0.0044	0.0059	0.0049	0.0044
<i>R</i>	0.0007	0.0045	0.0023	0.0031	0.0043	0.0034	0.0025
<i>R</i> ³	0.0097	0.0099	0.0099	0.0095	0.0102	0.0102	0.0094
<i>G</i>	0.0001	0.0237	0.0060	0.0112	0.0218	0.0129	0.0111

	Moy.
<i>TPS</i>	0.0088
<i>MQ</i>	0.0047
<i>R</i>	0.0030
<i>R</i> ³	0.0098
<i>G</i>	0.0124

FIGURE 4.6 – Tableau des erreurs obtenus par transfert « aller/retour » avec Simon comme modèle source et Barak comme modèle cible.

La Figures 4.9 illustrent deux séries de transferts avec un visage source différent pour chacune. Il est à noter que la position, la taille et l'orientation ne pose pas de problème dans le cadre du transfert étant donné que la régression par *RBF* prend en compte les transformations rigides (translation, rotation, échelle).

4.5 Conclusion

La méthode de transfert que nous proposons respecte les objectifs fixés. Tout d'abord, les résultats obtenus prouvent la qualité du transfert, de plus, pour la première fois à notre connaissance, nous proposons une comparaison de différentes fonctions à base radiale dans le cadre de l'animation faciale. Les interpolations par *RBF* sont souvent choisies arbitrairement et le choix de la fonction radiale est rarement justifié dans le domaine de l'animation faciale. Notre système offre la possibilité à un

	Neutre	Joie	Tristesse	Dégoût	Surprise	Peur	Colère
<i>TPS</i>	0.1430	0.1556	0.1391	0.1421	0.1439	0.1439	0.1409
<i>MQ</i>	0.0546	0.0935	0.0673	0.0932	0.0797	0.0748	0.0639
<i>R</i>	0.0060	0.0344	0.0212	0.0386	0.0232	0.0178	0.0233
<i>R</i> ³	0.1243	0.1654	0.1209	0.1241	0.1483	0.1376	0.1189
<i>G</i>	0.0004	0.0826	0.0479	0.0842	0.0586	0.0514	0.0499

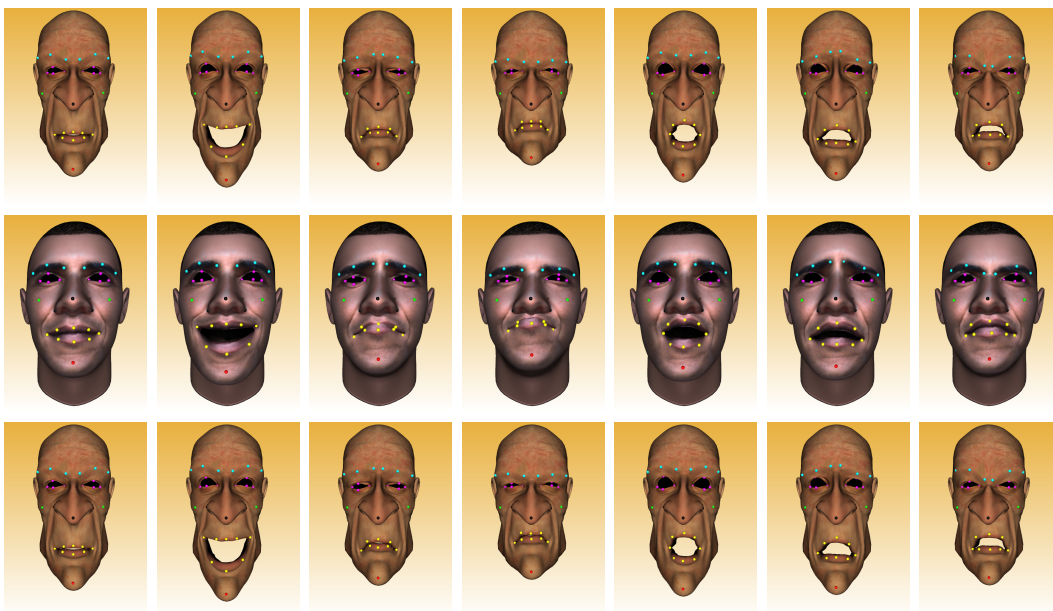
	Moy.
<i>TPS</i>	0.1441
<i>MQ</i>	0.0753
<i>R</i>	0.0235
<i>R</i> ³	0.1342
<i>G</i>	0.0536

FIGURE 4.7 – Tableau des erreurs obtenus par transfert « aller/retour » avec Proog² comme modèle source et Barak comme modèle cible.

graphiste de réutiliser une animation existante sur un nouveau personnage. Il s'agit d'un gain de temps considérable étant donné les efforts nécessaires à produire pour la réalisation d'une animation, aussi bien pour une animation créée manuellement que pour une animation obtenue par capture de mouvement. De plus, elle permet aisément d'animer un visage en temps réel. Ceci est très utile pour des applications dont le nombre de personnages est conséquent, plutôt que d'avoir les animations pour chaque personnage, les animations d'un seul visage peuvent être suffisantes, ce qui peut fournir un gain en mémoire notable. Le système de capture de mouvement que nous avons présenté permet également de confirmer la possibilité d'utiliser notre technique à partir de données réelles, aussi bien à partir de systèmes monoculaires en 2 dimensions que de systèmes dotés d'un plus grand nombre de caméras. Enfin, la possibilité de notre système de ne transférer qu'une zone du visage permet de l'utiliser pour la génération de nouvelles animations. Une partie des travaux présentés dans ce chapitre a été publiée dans une conférence internationale avec comité de lecture : « Feature Points Based Facial Animation Retargetting » dans VRST'08 (15th ACM Symposium on Virtual Reality Software and Technology) [DMB08].



(a) Simon -> Barak -> Simon



(b) Proog -> Barak -> Proog

FIGURE 4.8 – Sur les deux exemples présentés, la première ligne correspond au visage source, la seconde correspond au visage cible sur lequel nous avons transféré l'expression du visage source et la troisième correspond au visage source sur lequel nous avons transféré l'expression du visage cible. Un bon résultat se retrouve donc avec un visage source identique sur les lignes 1 et 3.



(a) Source : Betaboy, Cibles : Simon, Proog, Barak



(b) Source : Barak, Cibles : Simon, Proog, Betaboy

FIGURE 4.9 – La première ligne est le visage source en position neutre et avec les six expressions de base. Les trois autres lignes montrent les visages cibles sur lesquels les expressions ont été transférées.

Chapitre 5

Détails fins dynamiques

Sommaire

5.1	Introduction	83
5.2	Acquisition des détails d’animation	86
5.2.1	Image ratio	86
5.2.2	Calcul de la carte de normales	87
5.2.3	Adaptation au personnage 3D	90
5.3	Génération des détails fins d’animation en temps réel	91
5.3.1	Pose référence	91
5.3.2	Évaluation d’une pose quelconque par rapport à une pose référence	92
5.3.3	Masques d’influence	94
5.3.4	Fonction de modulation	96
5.3.5	Mélange des détails	97
5.4	Résultats	98
5.5	Discussion	101
5.6	Conclusion	105

5.1 Introduction

Les avancées récentes en rendu et en animation ont permis d’améliorer le réalisme des scènes virtuelles. Cependant, une animation faciale réaliste et en temps réel est toujours difficile à obtenir car les déformations fines de la peau comme les rides ou les renflements générés par l’action des muscles sur la surface du visage sont difficiles à créer. Bien que « petits », ces détails contribuent pourtant beaucoup à la perception des émotions et des expressions faciales [CBM09], et donc au réalisme du personnage virtuel. Mais l’ajout de ce phénomène dans le cadre d’une application temps réel est

délicat. Les difficultés se situent aussi bien au niveau de la génération de ces détails (obtenus par capture et/ou par synthèse) que sur la restitution de ceux-ci sur un visage 3D en temps réel.

Nous parlons ici des détails fins dans un contexte d'animation, c'est-à-dire les détails dynamiques qui apparaissent lorsque les muscles déforment le visage, et non des détails statiques comme les rides de vieillesse ou les pores de la peau. De nombreux travaux relatifs aux déformations larges ont été proposés [ED07, GB08], en revanche, très peu se sont consacrés à l'animation en temps réel des détails fins.

Oat a proposé une technique permettant de mélanger des « cartes de rides » pour le rendu de détails animés sur le visage humain [Oat07]. Les cartes de rides sont des cartes de normales utilisées sous forme de texture (*bump-mapping*) qui contiennent des informations sur les variations fines de surface que le maillage ne peut modéliser. Un masque ou calque est introduit pour définir différentes zones du visage, des coefficients sont fournis manuellement pour chaque zone afin de mélanger les cartes de détails au sein des zones. L'idée est intéressante et d'après les résultats fournis par les auteurs, cette technique produit de bons résultats en temps interactif, mais le fait de devoir définir les coefficients pour chaque frame ou pose clé de l'animation rend son application longue et délicate. De plus, chaque nouvelle animation nécessite de nouveaux coefficients et le travail réalisé précédemment ne peut pas être réutilisé. En nous basant sur les idées que nous avons développées pour le transfert, nous avons cherché à améliorer cette technique. L'idée est de compléter notre méthode d'animation en lui ajoutant des détails fins dynamiques et en l'adaptant à un visage paramétrisé par *skinning*.

Les cartes de rides peuvent être difficiles à obtenir. Trois solutions peuvent être envisagées. La première consiste à ce qu'un artiste les crée manuellement sur le personnage avec un logiciel de modélisation 3D, la seconde consiste à les synthétiser à base d'une modélisation physique ou de façon procédurale, et la dernière consiste à les capturer sur un visage réel. Extraire ces détails fins d'animation sur un visage réel est toujours un travail délicat malgré les avancées dans le domaine de la capture de mouvement. D'une part, les méthodes actuelles se concentrent sur la capture du visage avec l'aide de systèmes actifs (basés sur des lasers, des lumières structurées, etc.) et/ou de plusieurs caméras [WLVG07, ARL+09b, WGP+10]. Même si des travaux récents proposent des systèmes passifs [BBB+10, BHPS10], les contraintes liées à la stéréo-vision sont toujours présentes, et il est même souvent nécessaire d'utiliser au moins six caméras haute-définition. D'autre part, il n'est pas utile de capturer la géométrie complète du visage si seuls les détails fins sont souhaités.

Dans ce chapitre, nous proposons deux techniques pour à la fois :

- extraire les détails fins d’animation sur un visage réel ;
- et les appliquer sur une animation faciale 3D en temps réel.

Pour offrir une alternative aux systèmes de *MoCap* traditionnels, nous proposons une technique simple et à faible coût pour la partie acquisition. À partir de plusieurs images prises de face d’un même visage doté d’expressions différentes, nous extrayons les détails liés à chacune des expressions avec peu d’interventions manuelles qui se limitent à sélectionner des points repères sur les photographies. Ces détails sont obtenus sous forme de cartes de normales, en passant par le calcul d’une image ratio. L’utilisation d’une image ratio permet d’éviter le problème qui est de déterminer la couleur intrinsèque de la peau à partir d’une seule photo (voir l’Équation 5.5). Notre nouvelle approche n’utilise que des vues frontales et nous permet de nous concentrer sur le problème difficile de reconstruction de surface en utilisant les informations d’éclairage. Ces images sont obtenues à l’aide d’un seul appareil photo ou d’une seule caméra, alors que les approches classiques de capture faciale utilisent la stéréo. Une simple vue offre une flexibilité importante, mais l’impossibilité d’utiliser une méthode de triangulation à partir de plusieurs vues rend la recherche des déformations plus délicate.

Notre méthode de génération de détails fins dynamiques sur le visage 3D en temps réel est une technique de synthèse à partir d’exemples. Elle consiste à ajouter une couche d’animation à base de cartes de rides sur une couche d’animation classique par *skinning*. Il s’agit ainsi d’ajouter une couche d’animation des détails fins à une couche d’animation globale. Pour appliquer les cartes de rides, nous utilisons quelques poses références (en pratique deux poses références sont suffisantes). Une pose référence est l’association d’une déformation large (c’est-à-dire l’expression du visage défini par la position des points de contrôle) et des déformations fines qui lui sont associées (la carte de rides). Pendant l’animation, l’expression courante (*i.e.* la pose courante) du visage est comparée aux poses références et les coefficients des cartes de rides sont automatiquement calculés à l’aide d’une fonction non linéaire à base de cosinus. Il est important de noter que la comparaison est effectuée au niveau des points de contrôle et non au niveau de l’expression entière, ceci permet de mélanger localement les différentes cartes de rides des poses références sur différentes parties du visage. Une nouvelle carte de normales est ensuite générée dans le *GPU* à partir des cartes fournies en entrée. Notre approche de synthèse à partir d’exemples peut être utilisée avec le système d’acquisition que nous proposons, mais aussi avec d’autres techniques d’acquisition, avec des données produites par un graphiste, ou avec des données générées (de façon procédurale ou par simulation physique).

5.2 Acquisition des détails d’animation

Dans cette section, nous proposons une méthode d’acquisition qui détecte des détails de la peau de quelques expressions faciales et les transforme en cartes de normales. Étant donné que nous ne cherchons pas à capturer la géométrie complète du visage, notre approche peut utiliser les variations d’éclairage à la surface de la peau entre plusieurs images d’un même visage doté d’expressions différentes. Les photos peuvent être extraites d’une séquence vidéo ou prises par un appareil photo. Le système requiert peu de contraintes : la personne capturée doit être de face et rester le plus immobile possible entre les prises de vue ; l’éclairage doit être frontale au visage ou au moins dans l’axe vertical face au visage ; concernant le fond, aucune contrainte n’est imposée. Soit \mathcal{I} l’image du visage dans une expression neutre et \mathcal{I}' l’image du visage avec une expression à partir de laquelle nous souhaitons extraire les détails. Nous cherchons à calculer la carte de normales, cette carte sera ensuite utilisée pendant l’animation 3D de notre personnage virtuel pour produire les détails fins dynamiques d’animation.

5.2.1 Image ratio

La première étape de notre méthode consiste à calculer une image ratio pour extraire les variations de lumières [LSZ01]. Étant donné que l’expression de \mathcal{I}' implique une déformation par rapport à \mathcal{I} et que nous voulons comparer la variation pixel par pixel, nous commençons par déformer \mathcal{I}' pour que le visage expressif reprenne la position du visage neutre \mathcal{I} . Ceci permet d’obtenir un recalage entre chaque pixel, et donc, de connaître la différence d’illumination en chaque point de la surface. L’image de l’expression déformée \mathcal{I}'_d est obtenue en utilisant une interpolation par *RBF* entre deux ensembles de points repères positionnés manuellement sur les deux images \mathcal{I} et \mathcal{I}' (voir la Figure 5.1). L’image ratio \mathcal{R} est obtenue en divisant la luminance de \mathcal{I}'_d par celle de \mathcal{I} et est définie par :

$$\mathcal{R}(x, y) = \frac{\mathcal{I}'_d(x, y)}{\mathcal{I}(x, y)} \quad (5.1)$$

Nous pouvons raisonnablement estimer que $\mathcal{I}(x, y) \neq 0$ si $\mathcal{I}(x, y)$ est un pixel situé sur la peau du visage. Par abus de notation, nous adopterons par la suite :

$$\mathcal{R} = \frac{\mathcal{I}'_d}{\mathcal{I}} \quad (5.2)$$

Pour éviter les artefacts éventuels dus à un recalage imparfait, notamment au niveau des cheveux ou des sourcils ou d’autres contours du visage, nous offrons la possibilité de définir des régions d’intérêt, ce sont des zones sur lesquelles nous

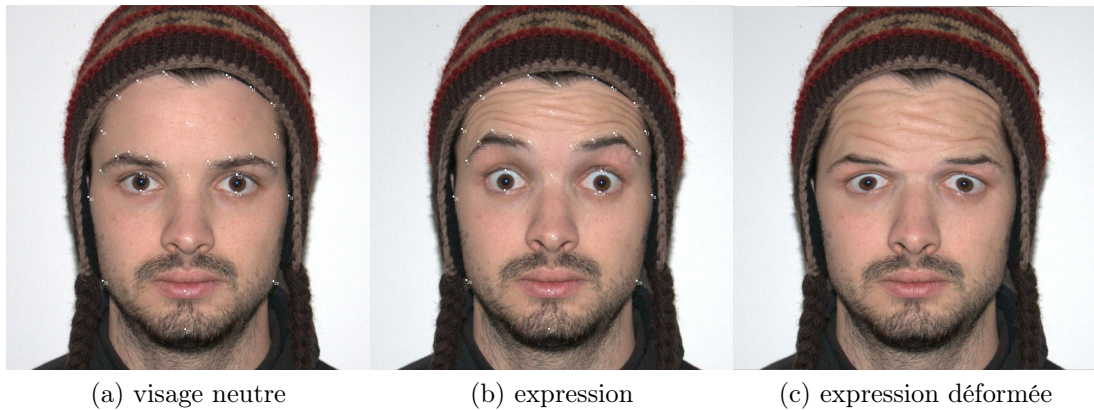


FIGURE 5.1 – Déformation de « l'expression » vers la forme du « visage neutre » pour obtenir « l'expression déformée ».

souhaitons extraire les déformations fines. Sur le plan pratique, notre logiciel permet de sélectionner ces zones à l'aide d'un outil « pinceau ». La Figure 5.2 illustre le masque utilisé et l'image ratio obtenue.

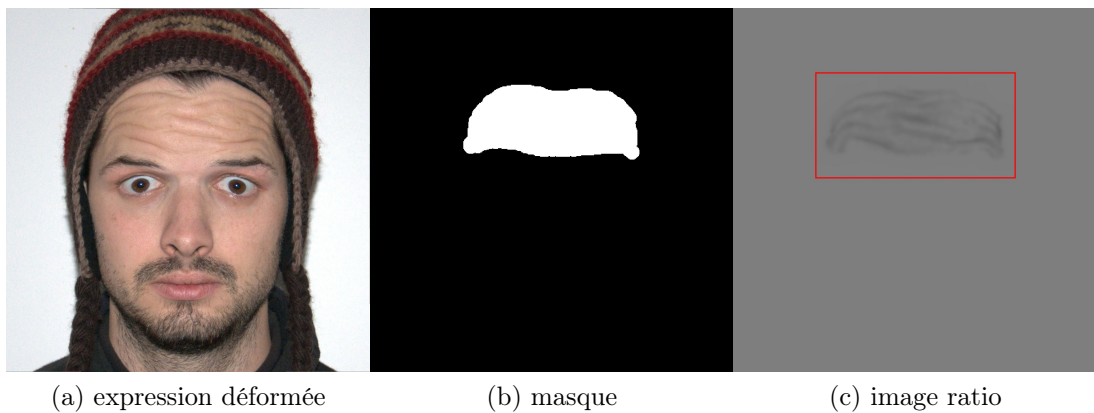


FIGURE 5.2 – L'image ratio est calculée en divisant l'image déformée par l'image neutre. Les zones définies par le masque représentent des zones d'intérêt et permettent d'éviter des erreurs de recouvrement dues aux frontières ou aux cheveux.

5.2.2 Calcul de la carte de normales

Dans ce paragraphe, nous notons par :

- C_d la couleur intrinsèque de la peau ;
- C_l la couleur de la lumière ;
- $\vec{N} = (N_x, N_y, N_z)^t$ la normale à la surface de la peau de l'expression neutre ;
- $\vec{N}' = (N'_x, N'_y, N'_z)^t$ la normale à la surface de la peau de l'expression interpolée ;
- \vec{L} le vecteur direction de la lumière ;

– et \cdot le produit scalaire.

Rappelons que nous cherchons \vec{N}' défini dans le repère tangent de la surface neutre dont la normale est \vec{N} car nous cherchons la variation de surface entre l'image neutre et l'image interpolée. L'image ratio \mathcal{R} qui divise \mathcal{I}' par \mathcal{I} est bien porteuse de ces informations relatives. En supposant que la peau est une surface diffuse, selon le modèle lambertien, nous avons :

$$\mathcal{I} = C_d \times C_l \times \vec{N} \cdot \vec{L} \quad (5.3)$$

et

$$\mathcal{I}'_d = C_d \times C_l \times \vec{N}' \cdot \vec{L} \quad (5.4)$$

La Formule 5.1 donne :

$$\mathcal{R} = \frac{\mathcal{I}'_d}{\mathcal{I}} = \frac{C_d \times C_l \times \vec{N}' \cdot \vec{L}}{C_d \times C_l \times \vec{N} \cdot \vec{L}} = \frac{\vec{N}' \cdot \vec{L}}{\vec{N} \cdot \vec{L}} \quad (5.5)$$

En utilisant le flash de l'appareil photo comme lumière directe, nous pouvons approximer la direction de la lumière à $\vec{L} = (0, 0, -1)^t$ dans le repère de l'image, ce qui donne :

$$\mathcal{R} = \frac{N'_x \times 0 + N'_y \times 0 + N'_z \times -1}{N_x \times 0 + N_y \times 0 + N_z \times -1} = \frac{N'_z}{N_z} \quad (5.6)$$

Nous pouvons ainsi estimer la composante N'_z de \vec{N}' par :

$$N'_z = N_z \times \mathcal{R} = N_z \times \frac{\mathcal{I}'_d}{\mathcal{I}} \quad (5.7)$$

En supposant que les zones du visage neutre définies par le masque sont planes, N_z possède une valeur constante par zone. Nous avons fait le choix de considérer le plan de ces zones comme étant parallèle au plan de l'image, ainsi $\vec{N} = (0, 0, 1)^t$. Cependant, nous laissons la possibilité à l'utilisateur, s'il l'estime nécessaire, d'affiner le résultat. Pour cela il lui suffit de préciser la valeur de N_z pour chaque zone. L'Équation 5.6 montre que l'intensité de l'image ratio est directement dépendante de la variation d'inclinaison de la surface déformée par rapport à la surface neutre. Plus \mathcal{R} est sombre, plus N'_z est petit et plus la déformation de la ride est importante. Notons que cette utilisation de l'image ratio permet d'éviter la difficulté qui consiste à trouver la couleur intrinsèque de la peau à partir d'une photo. Maintenant que nous connaissons N'_z , que nous pouvons interpréter comme une valeur relative à l'inclinaison de \vec{N}' par rapport à \vec{N} , nous devons déterminer le couple (N'_x, N'_y) qui représente la direction de la normale de l'image déformée. Ces valeurs ne peuvent être obtenues directement à partir de \mathcal{R} et N'_z . N'_z permet seulement de déterminer un cône des \vec{N}' possibles (même inclinaison mais directions (\vec{N}'_x, \vec{N}'_y) différentes).

Nous proposons d'utiliser le gradient de l'image ratio pour calculer la direction de la normale en faisant l'hypothèse que le gradient indique la direction perpendiculaire aux déformations. Ceci est une approximation qui nous semble réaliste comme le montre nos résultats. Dans le même esprit, Robles-Kelly *et al.* ajoutent des itérations de l'équation de diffusion de chaleur sur l'image d'illumination initiale en tant que pré traitement avant d'aligner les normales sur la direction du gradient. Ceci permet de lisser le résultat des directions [RKH07]. Soit G_x et G_y les valeurs du gradient 2D normalisé de l'image \mathcal{R} sur les deux axes \vec{x} et \vec{y} . Nous cherchons \vec{N}_x et \vec{N}_y tels que $\vec{N}_x^2 + \vec{N}_y^2 + \vec{N}_z^2 = 1$, et $N_x = \alpha G_x$ et $N_y = \alpha G_y$ avec α un coefficient de normalisation. Nous calculons alors la direction de la normale comme suit :

$$\begin{aligned} N'_x &= G_x \times \sqrt{1 - N'_z{}^2} \\ N'_y &= G_y \times \sqrt{1 - N'_z{}^2} \end{aligned} \quad (5.8)$$

Enfin, nous convertissons \vec{N}' dans un format de carte de normales. Notre implémentation utilise le format couramment utilisé qui définit les composantes (r, v, b) de l'image telles que :

- $r = ((N'_x + 1)/2.) \times 255$ (*i.e.* $N'_x \in [-1, 1]$ entraine $r \in [0, 255]$),
- $v = ((N'_y + 1)/2.) \times 255$ (*i.e.* $N'_y \in [-1, 1]$ entraine $v \in [0, 255]$),
- $b = ((N'_z + 1)/2.) \times 255$ (*i.e.* $N'_z \in [0, 1]$ entraine $b \in [127, 255]$).

La Figure 5.3 illustre une carte de normales obtenue à partir de l'image ratio.

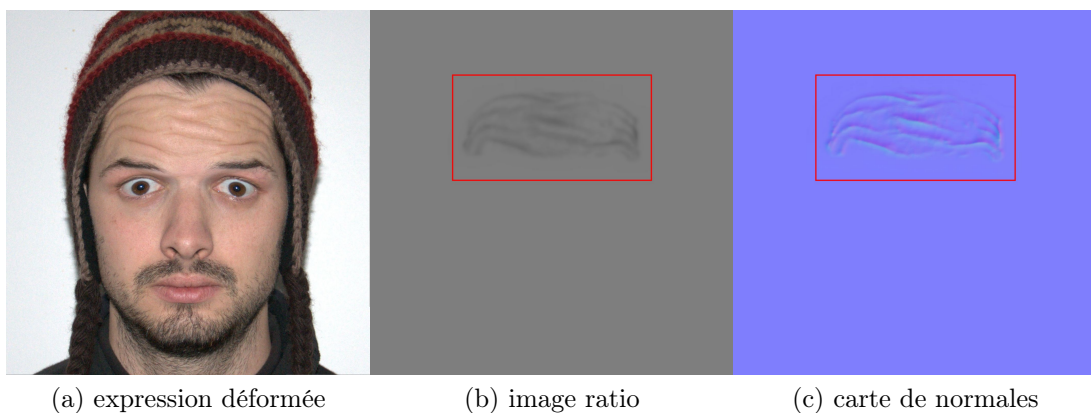


FIGURE 5.3 – À partir de l'image déformée (a), nous calculons l'image ratio (b) puis la carte de normales avec l'aide de l'image ratio et de son gradient (c) (dans un souci de lisibilité, les zones où les détails sont présents sont entourées d'un rectangle rouge, il en sera de même pour les images suivantes).

5.2.3 Adaptation au personnage 3D

Les étapes accomplies ci-dessus nous ont permis d'obtenir une carte de normales. Nous devons maintenant l'adapter pour pouvoir l'utiliser avec notre personnage virtuel. Cette opération est réalisée par une interpolation entre « l'espace » d'acquisition de la photographie vers « l'espace » des coordonnées de texture du maillage 3D. Nous utilisons une méthodologie similaire à l'interpolation réalisée dans le Paragraphe 5.2.1 : nous positionnons des points repères sur la carte de normales et sur la texture cible pour l'initialisation des *RBF*. La texture diffuse du personnage est particulièrement indiquée comme texture cible car nous utilisons les mêmes coordonnées textures pour la texture diffuse et la carte de normales. De plus, les couleurs de la peau, des sourcils, des lèvres, etc. permettent de situer les différentes zones du visage. Il est également possible de projeter le maillage sur la texture pour obtenir des informations complémentaires comme cela est illustré sur la Figure 5.4a. Le résultat de l'adaptation de la carte de normale de la Figure 5.3c sur le modèle 3D *Proog* est présenté sur la Figure 5.4b.

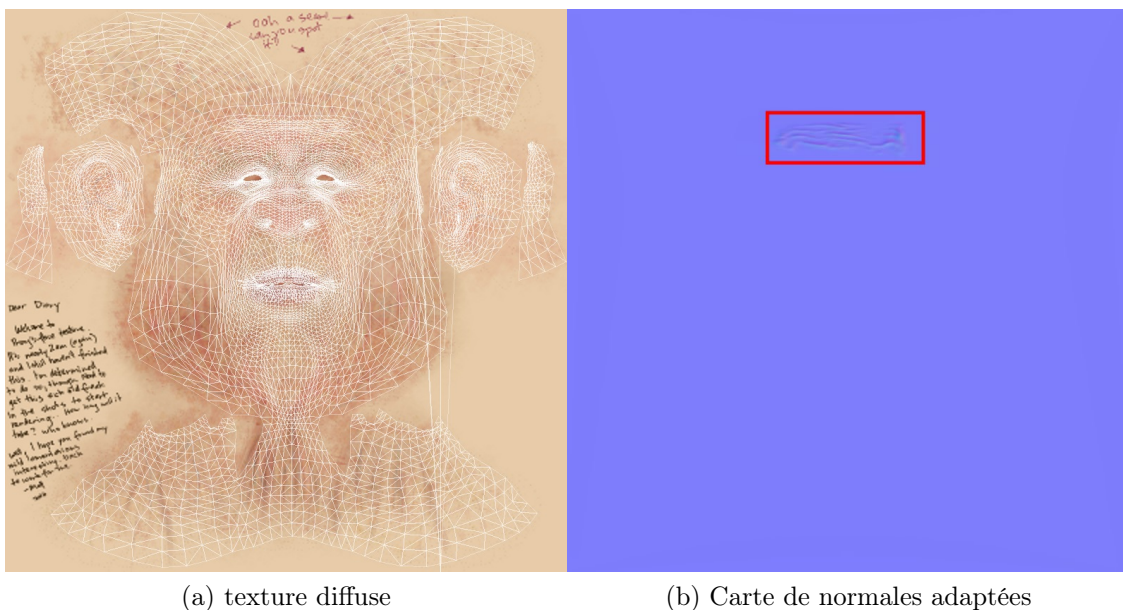


FIGURE 5.4 – La carte (b) est le résultat de l'adaptation de la carte de normales calculée (voir la Figure 5.3c) vers l'espace défini par la texture du personnage 3D (a).

5.3 Génération des détails fins d'animation en temps réel

Le but de la méthode que nous présentons maintenant est d'utiliser les détails extraits à partir de photos (voir la section précédente) pour générer des déformations fines et dynamiques à la surface de la peau. Ce procédé est une alternative aux différentes méthodes existantes qui nécessitent des calculs coûteux pour la génération des détails dynamiques de façon procédurale ou physique. Cette section décrit comment des poses références sont utilisées en temps réel pour une expression quelconque du visage. La notion de poses références est importante et sera décrite dans le Paragraphe 5.3.1. La Figure 5.5 présente l'ensemble des étapes de notre technique. Nous rappelons qu'en entrée notre approche utilise ρ poses références. Chacune d'entre elles est l'association d'une expression, *i.e.* du positionnement des points de contrôle, et d'une carte de détails. À chaque frame de l'animation, nous devons déterminer comment chaque pixel à la surface du visage utilise les poses références pour faire apparaître les détails. Pour cela, notre algorithme se décompose en trois étapes principales :

- calculer les influences de chaque pose référence sur chaque point de contrôle ;
- associer les poses références et les sommets du maillage en utilisant les poids du *skinning* et les influences calculées à la première étape. Les points de contrôle jouent un rôle essentiel ici en permettant de faire le lien entre les poses références et les sommets du maillage ;
- générer la nouvelle carte de normales après la phase de *rastérisation* avec un rendu par pixel¹.

5.3.1 Pose référence

Une pose référence est une expression définie par ses déformations larges et ses déformations fines. Sur le plan technique, elle est représentée par :

- la position des points de contrôle, *i.e.* la pose du squelette (déformations larges) ;
- une carte de normales (déformations fines).

Pour créer une pose référence, l'utilisateur déforme le visage à l'aide des points de contrôle pour obtenir l'expression sur laquelle il souhaite appliquer des détails. Cette expression doit être maximale² pour obtenir les meilleurs résultats, *i.e.* l'expression qui génère les rides les plus prononcées. Une fois cette pose créée, nous lui associons

1. En opposition au rendu par sommet qui ne calcule que la lumière au niveau des sommets

2. Nous parlons ici du point de vue de l'artiste car il est difficile de quantifier une expression maximale. Il s'agit de la pose extrême pour une expression donnée.

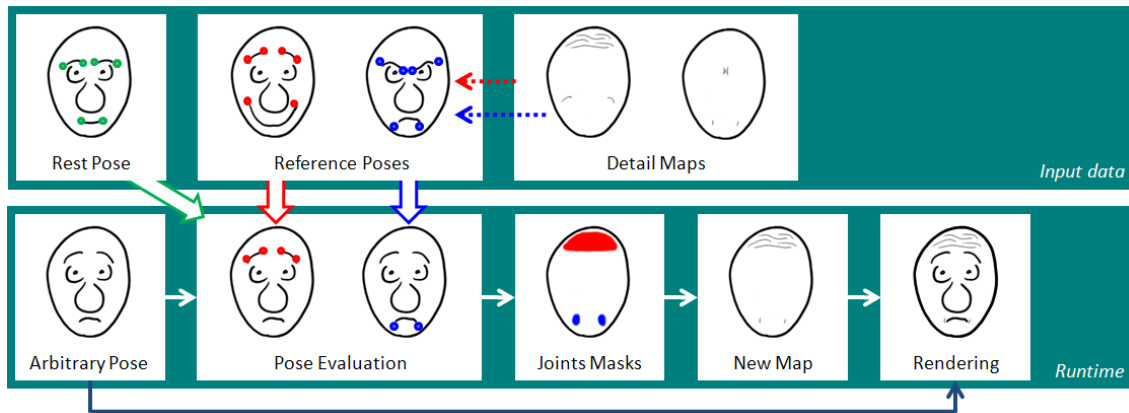


FIGURE 5.5 – Aperçu de la méthode de détails fins dynamiques. Les données d’entrée nécessaires sont un visage paramétrisé par *skinning* en position neutre et quelques poses références (pose référence = pose du squelette et carte de détails). Pendant l’animation et en temps réel, la pose courante est comparée aux poses références, point de contrôle par point de contrôle. Ensuite, les influences des points de contrôle sur les sommets du maillage sont utilisées comme « masques » pour appliquer les détails qui sont mélangés localement en fonction des coefficients obtenus. L’éclairage par pixel permet ensuite de faire apparaître les rides et autres déformations fines à la surface du visage.

une carte de rides. La méthode d’acquisition que nous proposons dans la section précédente peut être utilisée, cependant, tout autre moyen permettant de générer une telle carte de normales est employable. Étant donné que l’influence des poses références est calculée au niveau des points de contrôle (donc localement), il est possible de définir une pose référence où les rides apparaissent sur diverses zones du visage indépendamment : l’apparition de certains détails à un instant donné n’implique pas obligatoirement l’apparition des autres. Par exemple, des détails sur les joues peuvent apparaître indépendamment de ceux du front, même si ils sont définis dans une même pose référence.

5.3.2 Évaluation d’une pose quelconque par rapport à une pose référence

L’évaluation d’une pose quelconque par rapport à une pose référence consiste à estimer leur ressemblance. Elle traduit l’influence de chaque pose référence sur chaque point de contrôle. Il s’agit de trouver comment les positions des points de contrôle pour une expression quelconque de l’animation s’apparentent à leurs positions définies dans la pose référence. Soulignons que les calculs nécessaires à la synthèse des rides se font à l’échelle des points de contrôle, ou de groupes de points de contrôle, et non sur l’expression complète du visage. Effectuer cette évaluation

au niveau des points de contrôle permet de travailler localement et d'obtenir des masques d'influence. Ceux-ci ne pourraient pas être produits si les influences étaient calculées au niveau de l'expression complète. Ces masques d'influence offrent la possibilité d'appliquer différentes poses références à un instant donné, et donc réduit le besoin quantitatif des poses références. Deux poses références sont en général suffisantes pour reproduire les détails fins d'animation du visage : une expression ouverte (*stretch pose*) et une expression crispée (*compress pose*). Il est important de prendre en considération le fait que les positions des points de contrôle avec lesquelles nous travaillons sont définies dans le repère local de la tête. Nous pouvons donc considérer que les transformations rigides appliquées à celle-ci ou que les déplacements du corps du personnage ne conditionnent pas notre évaluation.

Dans ce paragraphe, nous notons par :

- Π_k la pose référence k avec $k \in \{1, \dots, \rho\}$;
- Π_0 la pose neutre ;
- Π_f la pose relative à la frame courante ;
- P_j un point de contrôle avec $j \in \{1, \dots, n\}$;
- \vec{P}_{jk} la position de P_j dans la pose référence Π_k ;
- \vec{P}_{j0} la position de P_j dans la pose neutre Π_0 ;
- \vec{P}_{jf} la position de P_j dans la pose courante Π_f ;
- $\|\dots\|$ la norme euclidienne ;
- et \cdot le produit scalaire.

Nous définissons l'influence α_{jfk} de la pose Π_k pour le point de contrôle P_j de position \vec{P}_{jf} dans la pose courante Π_f par l'équation :

$$\alpha_{jfk} = \min \left(1, \max \left(0, \frac{(\vec{P}_{jf} - \vec{P}_{j0}) \cdot (\vec{P}_{jk} - \vec{P}_{j0})}{\|(\vec{P}_{jk} - \vec{P}_{j0})\|} \right) \right) \quad (5.9)$$

α_{jfk} est la taille du segment entre la projection orthogonale de \vec{P}_{jf} sur le segment $[\vec{P}_{jk}, \vec{P}_{j0}]$ normalisé par la taille du segment $[\vec{P}_{jk}, \vec{P}_{j0}]$ (voir la Figure 5.6). Pour la frame courante de l'animation, les influences des poses références pour le point de contrôle P_j peuvent s'écrire comme un vecteur de dimension ρ :

$$\vec{\alpha}_{jf} = \begin{pmatrix} \alpha_{jf1} \\ \vdots \\ \alpha_{jfk} \\ \vdots \\ \alpha_{jf\rho} \end{pmatrix} \quad (5.10)$$

On peut remarquer que l'Équation 5.9 vérifie les propriétés suivantes :

- $\alpha_{jfk} = 0$ si P_j n'a pas bougé par rapport à la pose neutre ;
- $\alpha_{jfk} = 0$ si le déplacement de P_j est opposé à son déplacement dans la pose référence k ;
- $\alpha_{jfk} = 1$ si P_j se trouve à la même position que dans la pose référence k ;
- $\alpha_{jfk} = 1$ si P_j s'est déplacé dans la même direction que dans la pose référence mais de façon plus exagérée (ceci n'arrive pas en pratique si comme nous l'avons suggéré, la pose référence est une « expression maximale ») ;
- et que α_{jfk} est borné : $0 \leq \alpha_{jfk} \leq 1$.

À chaque frame de l'animation, ce vecteur des influences est calculer pour chacun des points de contrôle.

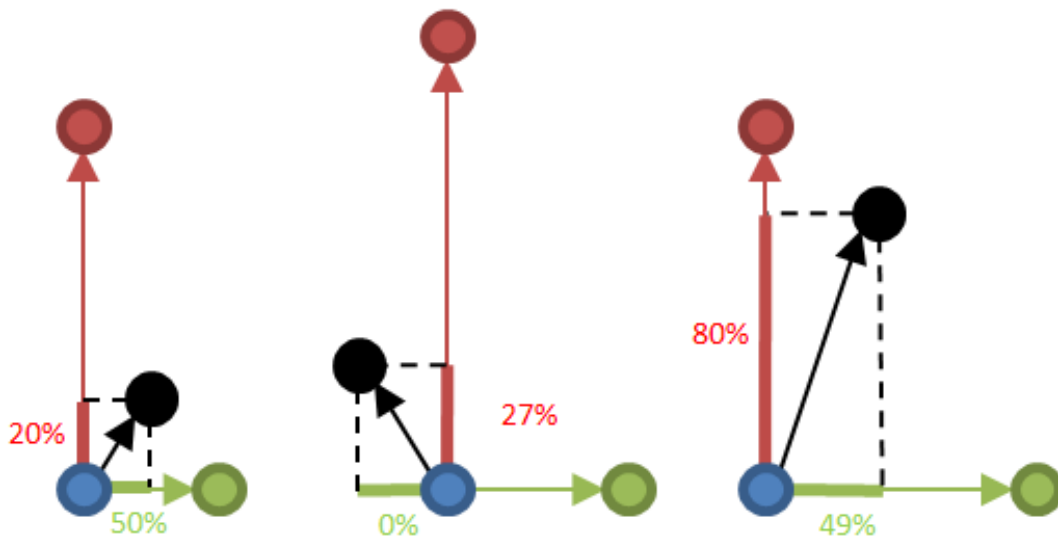


FIGURE 5.6 – Exemple 2D de l'évaluation d'une pose quelconque pour 3 points de contrôle. En bleu les points dans la pose neutre, en rouge les points dans la pose référence 1, en vert les points dans la pose référence 2 et en noir les points dans la pose courante à évaluer. L'évaluation se fait en projetant orthogonalement les points de la pose courante sur les axes reliant les points de la pose neutre et les points des poses références.

5.3.3 Masques d'influence

Une fois que nous connaissons les influences des poses références pour chaque point de contrôle, nous pouvons les utiliser pour le rendu des détails fins. Il s'agit de transmettre l'information connue à l'échelle des points de contrôle, aux sommets du maillage, puis jusqu'aux pixels affichés de la surface. L'idée principale consiste à utiliser les poids du *skinning* qui associent sommets du maillage et points de contrôle

pour déterminer l'influence des poses références sur chaque sommet, et donc, grâce à l'interpolation effectuée pendant l'étape de *rastérisation* par le *GPU*, pour chaque pixel (voir la Figure 5.7). Étant donné que les rides et autres détails fins d'animation sont fortement liés aux déformations faciales, nous pouvons déduire que ces détails sont également liés aux déplacements des points de contrôle. Ainsi, nous associons l'influence des points de contrôle à l'influence des poses références. Autrement dit, le masque d'influence d'un point de contrôle correspond à l'ensemble des sommets pour lesquels sa pondération (pondération du *skinning*) est non nulle. Un sommet v_i du maillage dont la pondération qui l'associe au point de contrôle P_i est de valeur w_{ij} sera influencé par la pose référence Π_k à la frame f tel que :

$$\beta_{ifk} = \sum_{i=1}^n \lambda_{ij} \times \theta(\alpha_{jfk}) \quad (5.11)$$

avec

- $\lambda_{ij} = 1$ si $w_{ij} > 0$;
- $\lambda_{ij} = 0$ sinon ;
- $\theta(x)$ la fonction de modulation que nous présentons dans la Section 5.3.4.

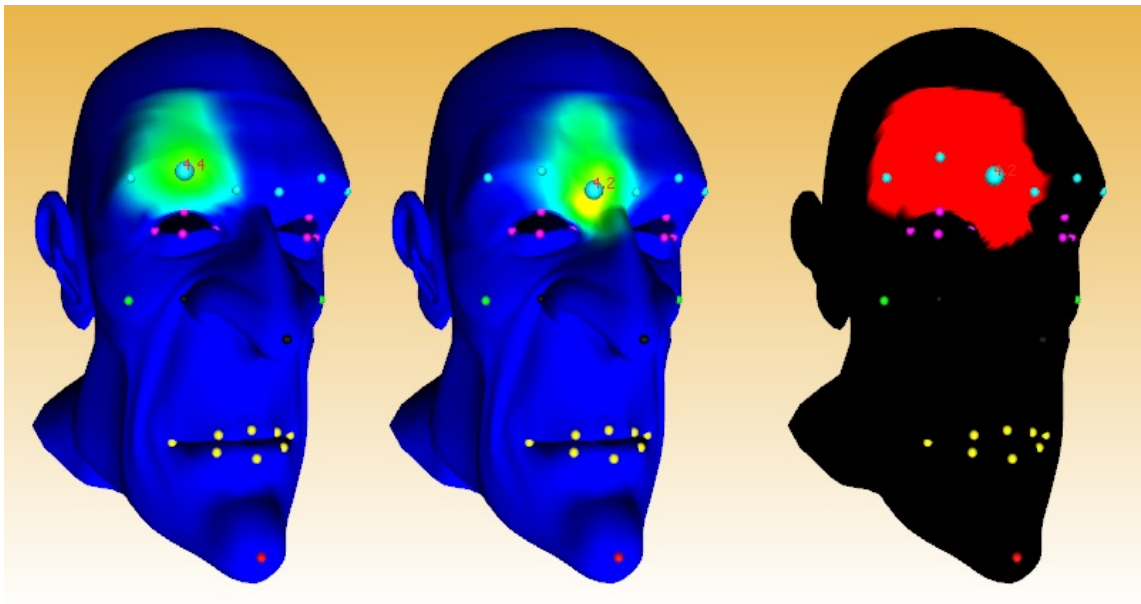


FIGURE 5.7 – Les deux images à gauche montrent les influences de deux points de contrôle du sourcil droit sur les sommets du maillage (pondération du *skinning*). Le masque d'influence défini par ces deux points de contrôle est constitué de tous les sommets du maillage dont les pondérations illustrées sur les deux images de gauche sont non nulles. le masque d'influence est colorié en rouge sur l'image de droite.

La paramétrisation d'un visage peut fortement varier selon l'application et selon la complexité de l'animation souhaitée. Le nombre de points de contrôle dans une même zone peut varier et une même ride peut être alors obtenue suite aux déplacements de plusieurs points de contrôle. Si ces points de contrôle ne se déplacent pas de façons similaires pendant l'animation, il est possible que des artefacts apparaissent. Ce phénomène est lié au fait qu'une même ride peut être plus ou moins prononcée le long de son sillon. Pour éviter ce type de problème, l'outil mis en place offre la possibilité à l'utilisateur de grouper les points de contrôle pour une pose référence donnée. On affecte la même valeur aux différents points de contrôle du groupe. Cette valeur est la moyenne de leurs influences pour une pose référence donnée.

5.3.4 Fonction de modulation

L'élasticité de la peau fait que les rides ne semblent pas apparaître de façon linéaire en fonction de la contraction des muscles. Pour répondre à cet aspect, nous avons cherché à déterminer une fonction capable de représenter au mieux la réalité observée. Pour simuler cet effet, nous introduisons pendant la phase d'évaluation des poses références une fonction θ telle que $\theta : [0, 1] \rightarrow [0, 1]$, que nous appelons fonction de modulation. Cette fonction a pour but de moduler la façon dont les rides apparaissent pendant l'action des muscles. Ceci se traduit dans notre approche à base de poses références par la modulation de l'influence des poses références sur les points de contrôle α_{jfk} . Pour obtenir une fonction basée sur l'apparition des rides d'une personne réelle, nous avons effectué notre algorithme d'acquisition de détails sur chaque image d'une séquence vidéo qui affiche le passage de l'expression neutre à l'expression sur laquelle les détails sont extraits. Cela fournit pour chaque pixel et pour chaque étape de la vidéo, un vecteur déplacement entre la normale neutre et la normale de l'expression. L'amplitude de ce vecteur traduit l'intensité des rides. Cependant, les limitations en mémoire disponible dans une application interactive rendent difficile l'utilisation directe de ces valeurs. L'analyse des données obtenues nous a permis de proposer une fonction analytique qui permet d'approximer l'amplitude des rides en fonction de l'amplitude des déplacements des points de contrôle :

$$\theta(x) = \left(\frac{\cos(\pi \times (1 + x)) + 1}{2} \right)^2 \quad (5.12)$$

La Figure 5.8 illustre la courbe composée de segments que nous avons capturée et la courbe que nous proposons. La fonction cosinus nous apparaît comme une base intéressante pour cette fonction si nous lui appliquons une translation et une normalisation pour vérifier la contrainte $\theta : [0, 1] \rightarrow [0, 1]$. Nous avons essayé de représenter dans cette courbe l'accélération lisse et progressive de l'affichage des

rides. Nous avons donné la priorité aux plus forts pourcentages de la déformation, là où les rides sont les plus visibles, alors que l'écart entre les deux courbes qui semble important au début de la déformation n'a qu'un faible impact visuel.

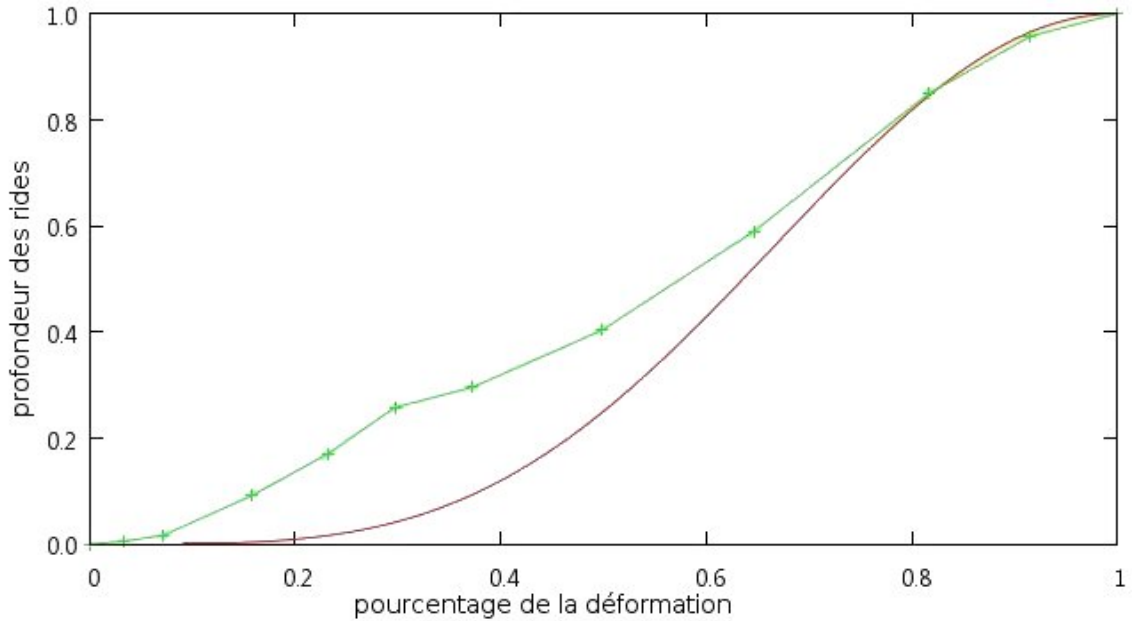


FIGURE 5.8 – Les données obtenues à partir de notre extraction de détails sur une séquence vidéo sont dessinées en vert. La fonction de modulation que nous proposons pour approximer la profondeur des rides en fonction du pourcentage de la déformation est tracée en rouge.

5.3.5 Mélange des détails

Une fois que nous avons appliqué la fonction de modulation aux influences des poses références sur chaque point de contrôle et obtenu cette influence $\beta_{i fk}$ au niveau des sommets du maillage, nous pouvons utiliser ces valeurs pour synthétiser la nouvelle carte de normales à partir de la carte de normales neutre et les cartes de normales associées aux poses références. Cette carte de normales sera utilisée pour le rendu. Rappelons que les cartes de normales ne contiennent que les déformations fines associées à l'expression, une simple moyenne des normales, même pondérée, entraîne la perte des déformations fins statiques présentes sur la carte de normales neutre (*i.e.* les détails comme les pores, les cicatrices et les autres détails statiques de la peau). Par exemple, un pixel influencé à cent pour cent par une carte de rides ne serait dessiné qu'avec cinquante pour cent des détails statiques de la carte neutre et cinquante pour cent des détails dynamiques. Néanmoins, si les cartes de rides ont été obtenues par un moyen différent de celui que nous proposons avec notre méthode d'acquisition, et dont les détails statiques sont présents dans les cartes de rides, cette

contrainte disparaît et la moyenne pondérée des normales neutre et de rides est alors suffisante.

Nous utilisons les notations suivantes pour un pixel donné :

- \vec{n}_f la normale finale utilisée pour le rendu et que nous voulons calculer ;
- \vec{n}_k les ρ normales fournies par les cartes de rides ;
- \vec{n}_0 la normale fournie par la carte de normales neutre ;
- γ_{fk} l'influence de la pose référence Π_k à la frame f sur le pixel obtenue (suite à la **rastérisation** des influences β_{ifk} des sommets) ;

Nous commençons par calculer les normales pondérées $\vec{\eta}_k$ par les coefficients γ_{fk} des cartes des poses références :

$$\vec{\eta}_k = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \gamma_{fk} \times \left(\vec{n}_k - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right) \quad (5.13)$$

Puis nous calculons la normale finale utilisée pour le rendu :

$$\vec{n}_f = \left\| \begin{pmatrix} n_x + \sum_{k=1}^{\rho} \eta_k^x \\ n_y + \sum_{k=1}^{\rho} \eta_k^y \\ n_z \times \prod_{k=1}^{\rho} \eta_k^z \end{pmatrix} \right\| \quad (5.14)$$

L'Équation 5.14 présente le calcul que nous utilisons pour éviter le problème de perte des informations statiques de la carte neutre. La somme appliquée aux deux premières composantes (sur les axes \vec{x} et \vec{y}) permet d'obtenir la direction moyenne des deux normales. Les composantes sur l'axe \vec{z} sont multipliées, ce qui entraîne une augmentation des détails obtenus à partir des deux normales. Plus la valeur sur \vec{z} est petite et plus le relief de la surface est prononcé dans le cas où le détail neutre et le détail dynamique sont orientés dans la même direction. Dans le cas contraire, une normalisation de la normale permet de la rapprocher de la normale unitaire perpendiculaire à la surface. Cette multiplication permet d'ajouter les variations de surface de la carte des détails statiques sur la carte des détails dynamiques. La Figure 5.9 illustre la différence entre un mélange simple des cartes de normales et la synthèse que nous présentons.

5.4 Résultats

La Figure 5.10 montre un exemple d'utilisation d'une carte de normales obtenue à partir d'une image réelle avec notre système monoculaire. Cette exemple démontre la possibilité de n'utiliser qu'une partie des détails d'une même carte. Avec les deux

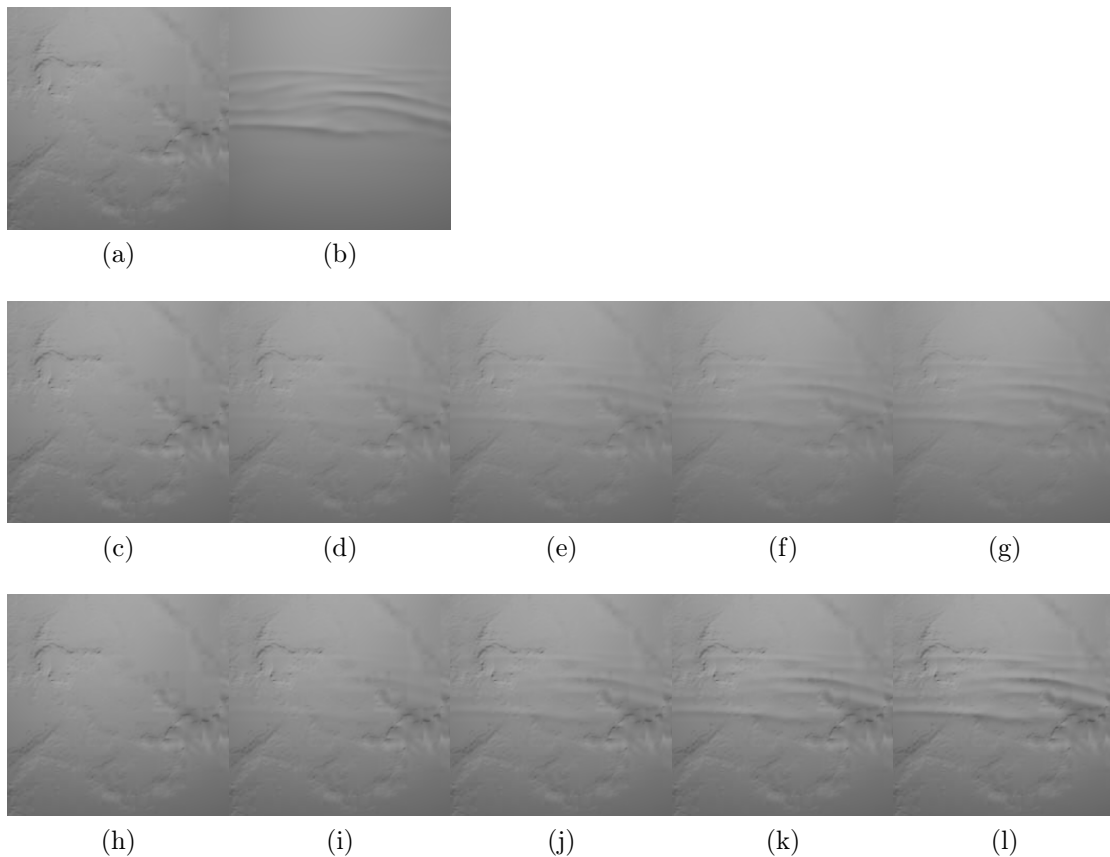


FIGURE 5.9 – Comparaison entre deux méthodes de mélange de carte de normales. (a) et (b) montrent le rendu obtenu avec les deux cartes de normales que nous voulons mélanger.

(c) → (g) : simple pondération telle que (a) est pondérée par 1. et (b) est pondérée successivement par 0., 0.25, 0.5, 0.75 et 1.

(h) → (l) : mélange présenté dans la Section 5.3.5 avec les mêmes valeurs de pondération. Ce mélange permet de conserver les détails alors que le mélange simple atténue les effets des deux cartes, la synthèse présentée permet de conserver les détails de chacune des cartes.

yeux grands ouverts et les sourcils relevés, les rides apparaissent sur toute la largeur du front. Par contre, si un seul sourcil est levé, les rides n'apparaissent qu'au dessus de celui-ci. Cela nous assure du fait qu'il n'est pas nécessaire de capturer toutes les expressions comme en aurait besoin un système à base de *blendshape*, mais qu'un ensemble limité permet d'obtenir de nombreuses possibilités de détails.

Nos tests ont été effectués avec nos modèles *Proog* et *Barak* qui possèdent chacun environ 10 000 triangles. Ils sont paramétrisés par *skinning* et possèdent 28 points de contrôle. Le rendu est réalisé avec l'*API* graphique *OpenGL* et NVidia Cg³ pour

3. http://developer.nvidia.com/page/cg_main.html

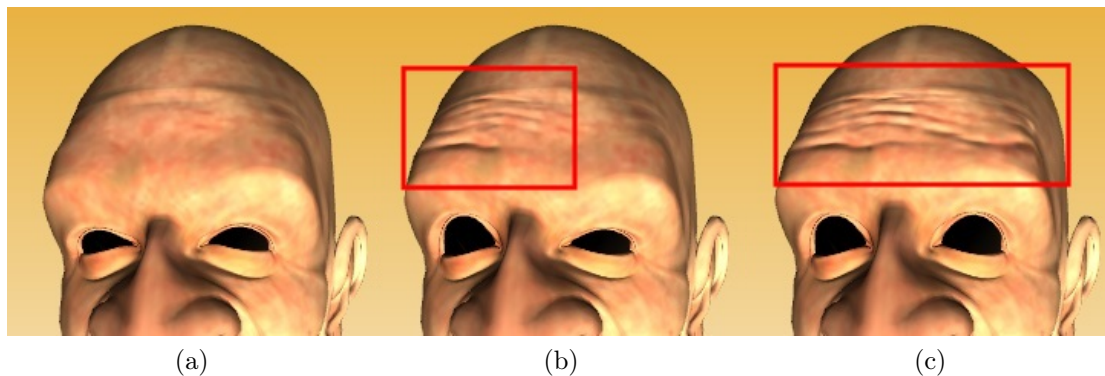


FIGURE 5.10 – Utilisation d’une carte de normales obtenue par notre méthode d’acquisition. La carte de normales est celle utilisée pour illustrer la Section 5.2. (a) expression neutre, aucun détail ne s’affiche. (b) un seul sourcil est relevé, une seule partie des rides s’affiche. (c) les deux sourcils sont relevés, les rides d’affichent tout le long du front.

la programmation sur carte graphique. L’ajout de la couche de détails fins n’entraîne pas de dégradation des performances et l’animation tourne aisément à plus de 100 frames par seconde sur un ordinateur classique doté d’un *CPU* AMD Athlon 64 3800+, 3GO de mémoire RAM et une carte graphique NVidia Geforce GTX 275. Sur nos deux modèles, nous avons utilisé deux poses références présentées sur les Figures 5.12 et 5.13. Ces poses références ont été générées avec les cartes de détails capturées sur la Figure 5.11. Grâce à la possibilité de n’agir que localement, nous avons pu fusionner deux cartes dont les détails ne se situaient pas au même endroit. La capture a été réalisée avec l’aide d’un appareil photo numérique d’une définition de 12 Mpx. Les Figures 5.14 et 5.15 montrent nos personnages avec les six expressions de base. La première ligne représente le personnage sans détail, la seconde avec notre système de rides dynamiques, et la dernière propose un code couleur par rapport à l’influence des poses références (le rouge correspond à l’influence de la pose référence 1, le vert correspond à l’influence de la pose référence 2, et le jaune est un mélange des deux).

Dans notre implémentation, l’évaluation des poses est réalisée sur le *CPU*, les coefficients obtenus sont envoyés au *GPU*. L’étape des masques d’influence est calculée dans le *vertex shader*, ce qui résulte en ρ nombres flottants (une valeur pour chaque pose référence). L’étape de *rastérisation* est ensuite calculée par le *GPU* et nous obtenons les valeurs interpolées pour chaque fragment dans le *pixel shader* où nous utilisons ces coefficients pour synthétiser la nouvelle carte de normales et calculer l’éclairage. Les déformations larges ainsi que le rendu ne sont pas perturbés, l’ajout de notre méthode à une implémentation existante est facile. Aucune passe de

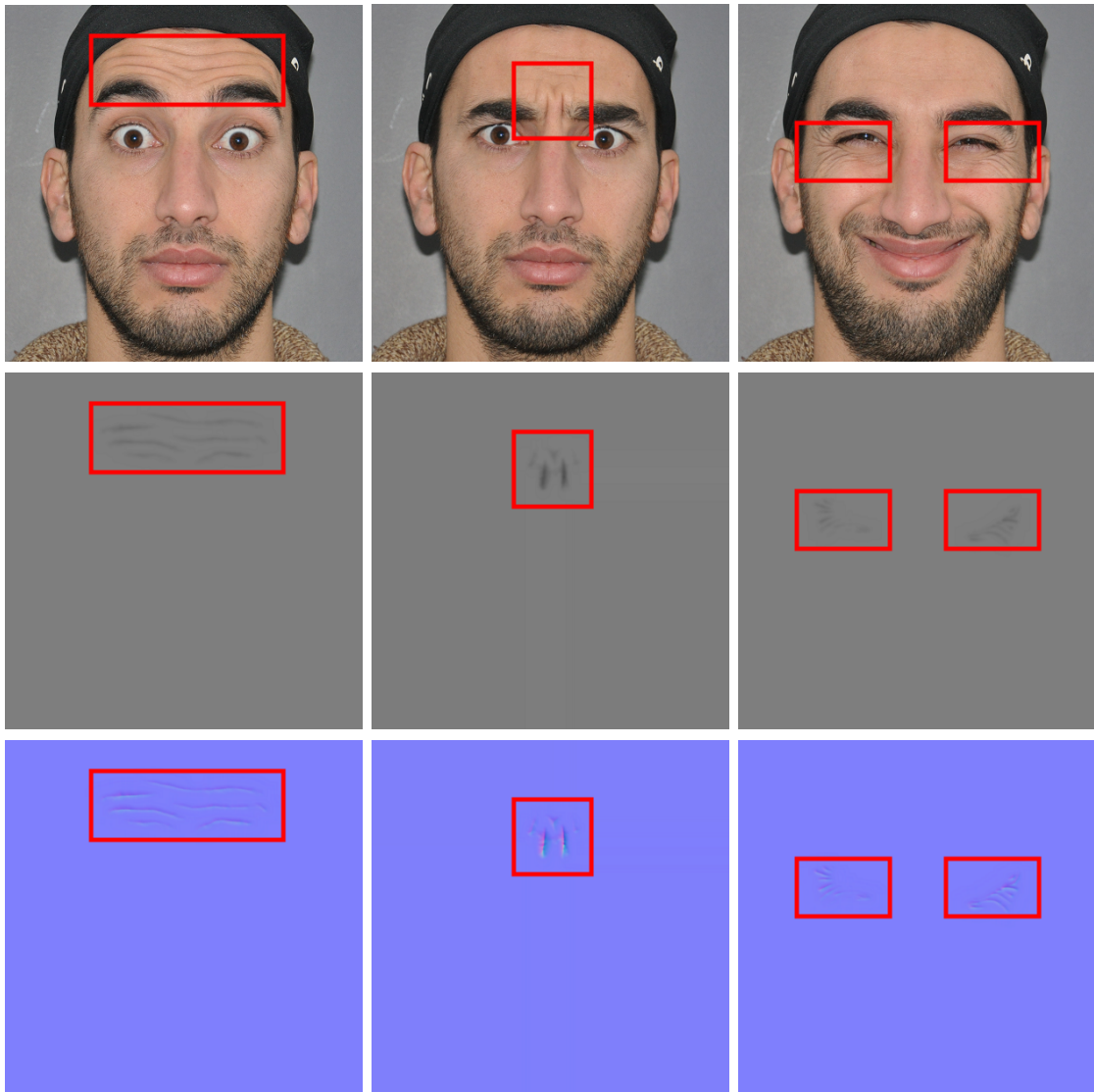


FIGURE 5.11 – Résultats de l’acquisition de trois cartes de détails utilisées sur nos personnages 3D.

rendu supplémentaire n’est requise, quelques fonctions doivent être ajoutées aux différentes étapes citées précédemment. Les données envoyées au *GPU* correspondent à $\rho \times n$ nombres flottants avec ρ le nombre de poses références et n le nombre de points de contrôle.

5.5 Discussion

Les performances de notre méthode dépendent de trois éléments. Premièrement, la paramétrisation joue un rôle important étant donné que nous utilisons directement les poids du *skinning* comme masques d’influence, et donc, ils définissent la

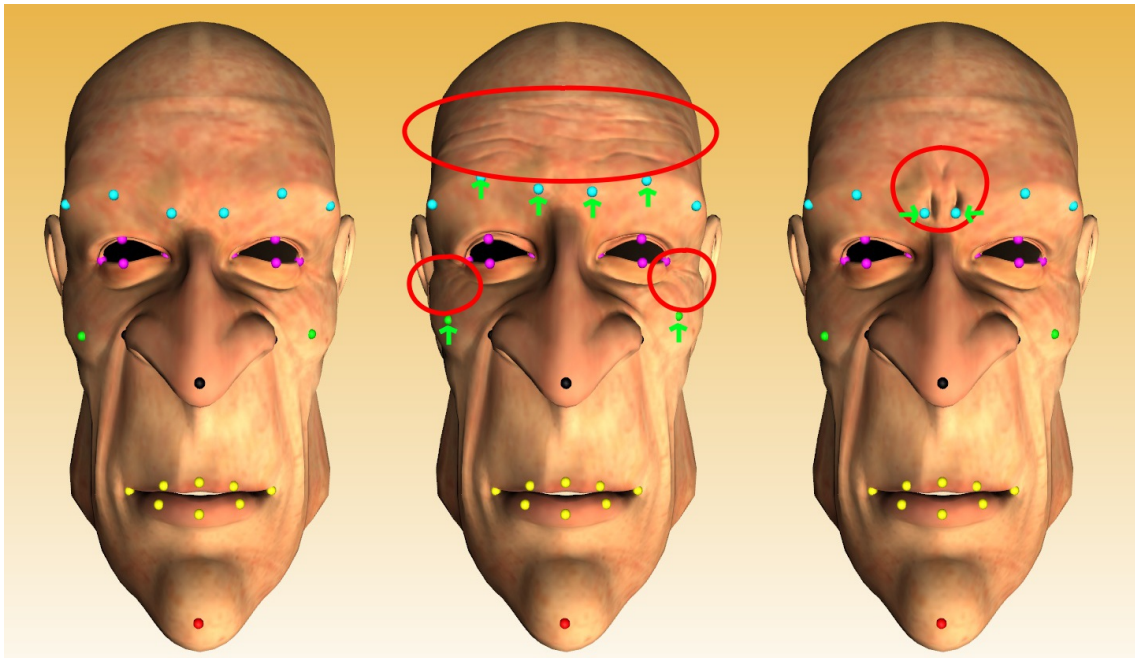


FIGURE 5.12 – La pose neutre (à gauche) et les deux poses références. Les zones entourées de rouge indiquent où les détails apparaissent. Les flèches vertes indiquent le déplacement des points de contrôle qui définissent la pose référence.

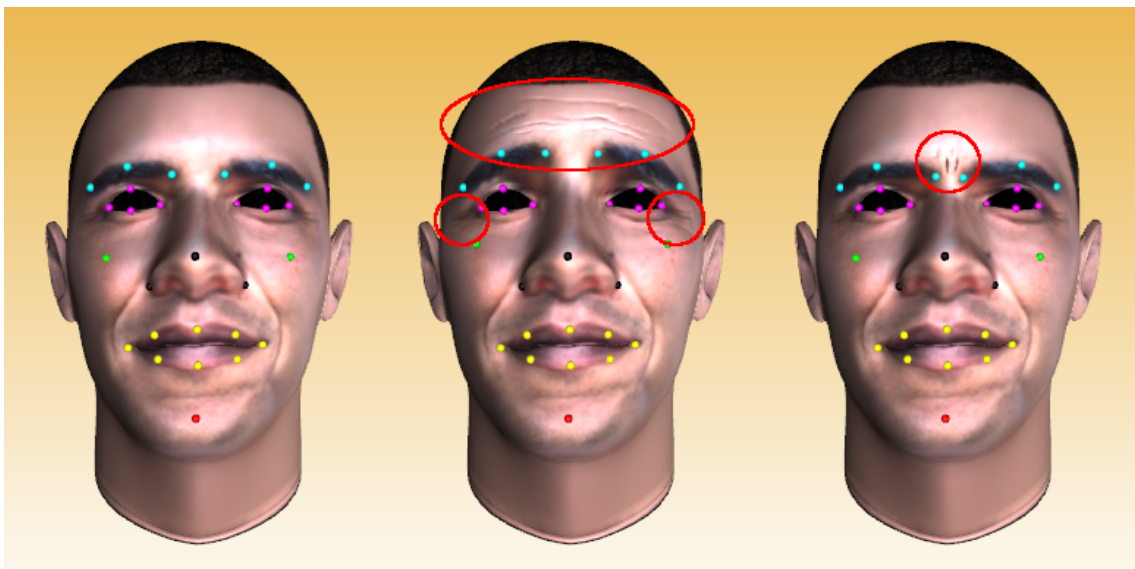


FIGURE 5.13 – La pose neutre (à gauche) et les deux poses références. Les zones entourées de rouge indiquent où les détails apparaissent.

façon dont chaque sommet est influencé par les différentes poses références. Une paramétrisation qui permet de modifier correctement la géométrie peut provoquer des défauts au niveau des détails si la pondération n'est pas assez lisse entre les sommets voisins. Deuxièmement, les poses références doivent être autant que possible orthogonales pour éviter qu'elles interfèrent entre elles de façon « non désirée » pen-

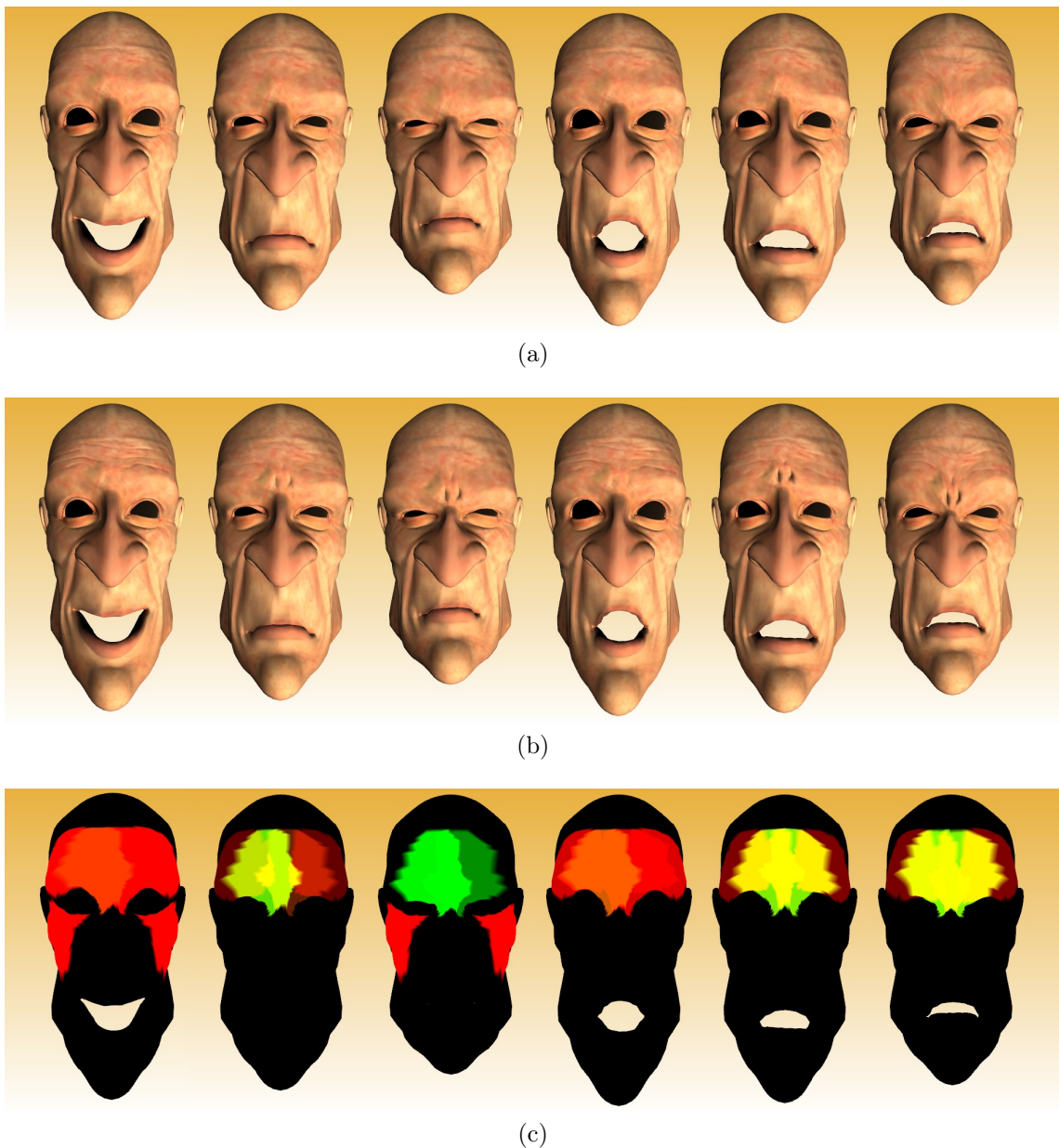


FIGURE 5.14 – (a) montre les 6 expressions de base (joie, tristesse, dégoût, surprise, peur, colère) sans détail. (b) montre les mêmes expressions en utilisant notre méthode de rides dynamiques avec nos données capturées. (c) montre l’influence des poses références sur le maillage (les pixels rouges sont ceux influencés par la pose référence 1, les pixels verts sont ceux influencés par la pose référence 2, les pixels jaunes sont ceux influencés par les deux poses références).

dant l’animation (concept similaire à la paramétrisation par *blendshape*). Signalons que l’utilisation de plusieurs poses références sur une même zone est possible, mais cela devient problématique si des déplacements similaires des points de contrôle engendrent différents détails fins. Compte tenu de notre définition des poses références, ce cas de figure est considéré comme un défaut de conception des poses références.

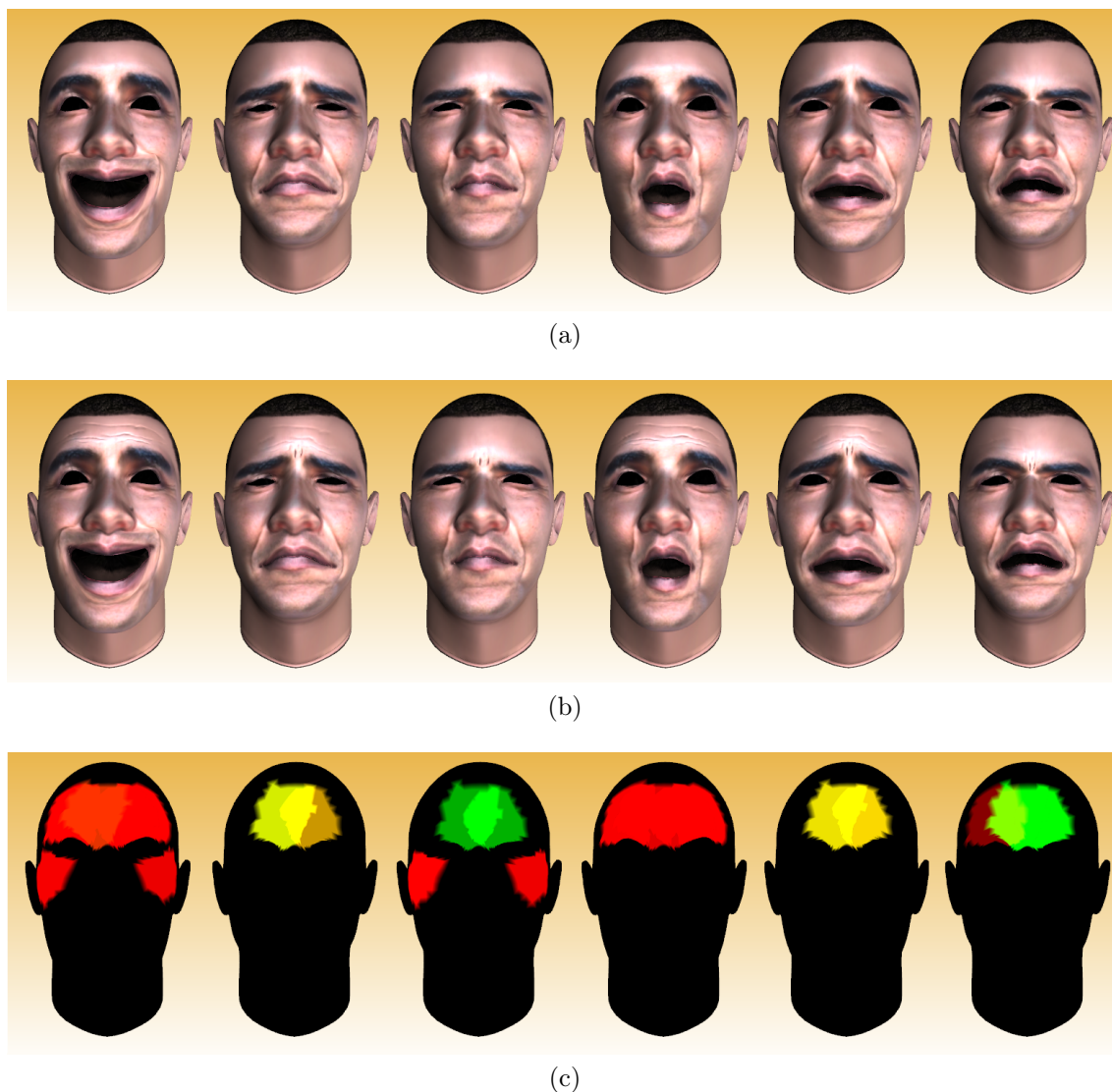


FIGURE 5.15 – (a) montre les 6 expressions de base (joie, tristesse, dégoût, surprise, peur, colère) sans détail. (b) montre les mêmes expressions en utilisant notre méthode de rides dynamiques avec nos données capturées. (c) montre l’influence des poses références sur le maillage (les pixels rouges sont ceux influencés par la pose référence 1, les pixels verts sont ceux influencés par la pose référence 2, les pixels jaunes sont ceux influencés par les deux poses références).

Enfin, la qualité des cartes de détails influence très fortement la qualité des résultats visuels. Concernant la phase d’acquisition, des problèmes apparaissent si le visage porte une barbe empêchant d’extraire les détails autour de la bouche. Les poils perturbent le calcul de l’image ratio et peuvent même cacher les rides.

Le choix d’utiliser les poids du *skinning* comme masques d’influence offre plusieurs avantages. Il permet de mettre en relation les déformations larges et les déformations fines par le biais des points de contrôle. Contrairement à d’autres travaux

[Oat07], nous n'avons pas besoin de masques sous forme de textures additionnelles pour définir les différentes zones des rides. Il permet également d'assurer le fait que les sommets influencés par une pose référence sont des sommets qui se déplacent en accord avec les points de contrôle. L'introduction d'une fonction de modulation des rides permet d'obtenir une dynamique plus réaliste de la peau et de son élasticité. L'analyse de la courbe d'apparition peut être réalisée sur un nouveau sujet possédant des caractéristiques différentes que le sujet utilisé dans cette étude. Elle est néanmoins facilement modifiable et l'utilisateur peut aisément définir et appliquer une nouvelle fonction de modulation plus en accord avec ses besoins.

Par rapport aux méthodes existantes, notre approche permet une diminution importante des coûts, aussi bien au niveau des ressources matérielles que des ressources humaines. Le prix d'un appareil photo numérique grand public ou d'une caméra de type webcam est faible en comparaison des différentes caméras stéréos haute-résolution ou des projecteurs de lumières structurées habituellement utilisés. De plus, il n'est pas nécessaire de poser des marqueurs sur le visage capturé, et l'intervention manuelle nécessaire se limite au positionnement d'une trentaine de points repères sur chaque image et au dessin d'un masque de sélection.

5.6 Conclusion

Ce chapitre a traité de l'ajout de détails fins dynamiques dans une animation en temps réel. Deux méthodes complémentaires ont été proposées. La première vise à capturer les détails fins d'animation du visage à partir de deux images d'une même personne avec et sans expression. Elle ne nécessite que peu d'efforts manuels et n'exige pas de matériel spécifique. Un appareil photo grand public ou une caméra suffit. Malgré l'approximation du modèle d'éclairage, nous obtenons des résultats visuels encourageants. Notre méthode propose une alternative aux coûteux systèmes traditionnels si l'on vise à obtenir les détails fins d'animation. La seconde permet l'utilisation de poses références pour générer en temps réel, à partir d'exemples, et pendant l'animation par *skinning* d'un visage, des détails fins d'expression.

Notre système de détails dynamiques fonctionne en temps réel, l'utilisation des méthodes d'éclairage par pixel permet d'éviter l'utilisation de maillages haute-résolution ou de techniques de subdivision⁴. La réalisation effective de notre méthode s'appuie sur des techniques très répandues dans les moteurs graphiques ac-

4. Notons tout de même que les récentes évolutions des cartes graphiques permettent de subdiviser un maillage en temps réel. L'utilisation des cartes de normales que nous proposons peut facilement être adaptée à de telles techniques qui utilisent une carte de hauteur.

tuels comme le *skinning* pour l'animation et le *bump-mapping* pour l'éclairage et le rendu des détails. Elle a l'avantage d'être compatible à un partage des ressources souvent nécessaire dans des applications interactives (animation, rendu, intelligence artificielle, interface utilisateur, réseau, etc.). Son implémentation ne présente pas de difficultés techniques et ne modifie pas les chaînes d'animation et de rendu habituelles. Une partie des travaux présentés dans ce chapitre a été publiée dans une conférence internationale avec comité de lecture : « Real-Time Dynamic Wrinkles of Face for Animated Skinned Mesh » dans ISVC'09 (*5th International Symposium on Advances in Visual Computing*) [DMB09], et dans une revue internationale avec comité de lecture « Easy Acquisition and Real-Time Animation of Facial Wrinkles » dans JCAVW (*Computer Animation and Social Agents (CASA2010)/ Journal of Computer Animation and Virtual World* (à paraître)) [DMB11].

Chapitre 6

Conclusion Générale

Nous nous sommes intéressés dans cette thèse à la problématique de l'animation faciale pour les applications en temps réel. Nous nous sommes particulièrement concentrés sur l'accessibilité des solutions proposées. Cette accessibilité se traduit par une facilité d'utilisation, même pour un utilisateur novice, et par des besoins limités en matériel. Nous avons également pris en compte l'objectif du temps réel qui impose de nombreuses contraintes. Au cours de ces travaux, nous avons tenté d'apporter des solutions à trois problèmes principaux :

- La paramétrisation d'un visage est une étape longue et ardue, comment peut-on aider le graphiste dans sa réalisation ; voire comment faire pour rendre ce travail accessible à un utilisateur novice ?
- La création d'une animation est également une étape cruciale, comment faciliter sa création, et surtout, comment la réutiliser sur des personnages différents que celui pour lequel elle a été créée ?
- Les détails fins d'animation comme les rides sont importants pour l'expressivité d'un visage, comment ajouter à une animation en temps réel de telles déformations ?

Nos travaux sont basés sur des techniques connues et largement utilisées dans des applications interactives, à savoir le *skinning* pour la paramétrisation du visage et le *bump-mapping* pour le rendu par pixel (en opposition au rendu par sommet). Le choix du *skinning* est orienté par les contraintes liés à nos objectifs et s'avère être le choix le plus judicieux quant à une utilisation facile et intuitive en temps réel. Nos contributions s'articulent autour de trois points complémentaires pour proposer une démarche complète pour l'animation faciale.

Concernant la paramétrisation du visage pour l'animation, nous avons développé deux techniques distinctes. La première, semi-automatique, propose d'aider le gra-

phiste dans la tâche d'association des points de contrôle aux sommets du maillage et offre une alternative plus adaptée pour le visage que les enveloppes d'influence traditionnellement proposées par les logiciels de modélisation 3D. La seconde contribution complète la première étape. Au lieu de recréer à chaque fois la paramétrisation pour chacun des nouveaux visages, nous sommes partis du principe qu'il serait plus judicieux de réutiliser une paramétrisation existante et de l'appliquer à un nouveau modèle. Nous avons ainsi développé une nouvelle méthode de transfert de paramétrisation en nous basant sur la mise en correspondance de surface. Cette technique permet de détecter des points repères dans chacun des deux visages 3D que nous analysons dans une représentation 2D. Les paires de points repères obtenues permettent ensuite de transférer les points de contrôle du visage *source* au visage *cible*. Nous avons ensuite développé une technique de mise en correspondance de surfaces denses pour transférer les influences du *skinning*. Cette approche de transfert de paramétrisation est entièrement automatique, mais l'intervention de l'utilisateur reste néanmoins possible si il souhaite personnalisé les résultats.

Au niveau du travail d'animation, nous avons proposé une méthode de transfert flexible et rapide. Elle se base sur une interpolation par *RBF*. Notre approche, basée sur les points de contrôle du *skinning*, permet avec un même processus : d'animer un visage 3D à partir d'une animation existante d'un autre visage 3D à la morphologie différente ; et d'animer un visage 3D à partir d'un système de capture de mouvement 2D (ou 3D). Notre système est flexible. Il offre une indépendance entre les régions qui permet de n'effectuer qu'un transfert partiel de l'animation ou de fusionner plusieurs sources dans une seule animation. Cette technique de transfert peut être employée de plusieurs façons :

- pour générer une nouvelle animation à partir d'un système de *MoCap*, en offrant un résultat visuel instantané du fait de sa rapidité ;
- dans un esprit d'optimisation de mémoire pour une application interactive, en ne stockant qu'un ensemble d'animations destinées à un même visage pour les transférer ensuite en temps réel sur les autres personnages de l'application ;
- nous pouvons imaginer une application dont les émotions transmises par les yeux seraient obtenues à partir d'un système de suivi et les mouvements de la bouche et de la mâchoire seraient obtenus à partir de la parole de l'utilisateur.

Nos dernières contributions concernent la problématique des détails fins d'animation pour le temps réel. Nous avons proposé une technique qui conjugue les déformations larges d'un visage paramétrisé par *skinning* et les déformations fines d'animation stockées dans des cartes de détails. Pour cela, notre système se base sur un petit ensemble de poses références qui constituent des paires de déformations

finies et larges et permettent de générer à la volée à partir d'exemples la carte de détails utilisée à un instant donné de l'animation. L'utilisation de la technique de *bump-mapping* est largement utilisée pour le rendu de scène 3D et permet d'éviter l'utilisation de maillages haute-résolution. L'utilisation d'une couche de détails fins qui se rajoute à la couche d'animation large offre une flexibilité importante. Son implémentation ne présente pas de difficultés techniques et ne modifie pas les chaînes d'animation et de rendu habituelles. Il est facile de l'activer ou de la désactiver dans le cadre de l'utilisation de niveaux de détails. De plus, notre méthode est basée sur des techniques très répandues dans les moteurs graphiques actuels. Pour obtenir les cartes de détails, nous avons développé une technique d'acquisition qui permet d'extraire les détails d'animation d'une personne réelle. Cette technique ne nécessite qu'un environnement faiblement contrôlé et peu de moyens matériels en comparaison des systèmes proposés dans la littérature. En effet, un seul appareil de capture est nécessaire (appareil photo, caméra, ou webcam) et l'éclairage se limite à la source de lumière de l'appareil de capture ou d'une source de lumière frontale.

Les méthodes que nous proposons apportent des solutions aux problèmes posés. Elles respectent les objectifs et les contraintes que nous nous sommes fixés, à savoir, le temps réel, la facilité d'utilisation, la flexibilité et le faible coût matériel. Elles s'adressent aussi bien aux professionnels qu'aux utilisateurs de petites structures du monde du graphisme qui réalisent des animations à partir de systèmes de capture ad-hoc, par le biais d'applications adaptées. Associées entre elles, ces méthodes peuvent constituer une chaîne de production qui permet de générer facilement des animations à partir d'un simple maillage 3D.

Un aspect intéressant pour la continuation de nos travaux concerne l'association de techniques de paramétrisation, l'idée est de mixer différents types de paramétrisation pour bénéficier des avantages de chacun. Des travaux ont déjà été effectués par Lewis sur l'animation du corps *et al.* [LCF00]. Concernant le visage, notre utilisation d'une couche de détails ajoutée à une couche de déformation large par points de contrôle peut s'apparenter au mélange du *skinning* et du *blendshape*. Malgré l'évolution rapide des techniques et la qualité des travaux proposés dans le domaine de l'animation faciale, de nombreux points sont encore à améliorer. Les systèmes de capture les plus performants sont encore beaucoup trop lourds et coûteux, bien qu'ils obtiennent de très bons résultats. Un axe important de recherche sera de trouver des techniques plus rapides et moins exigeantes. Notre méthode d'acquisition de détails s'appuie sur cette logique de capture facile et démocratisée. L'automatisation et la facilité d'utilisation telles que nous les proposons dans nos travaux démontrent la possibilité de réduire notablement une partie des contraintes. Nos recherches se sont

principalement basées sur l'aspect géométrique et visuel de l'animation faciale en temps réel, il serait intéressant d'aborder des travaux sur un niveau supérieur d'abstraction de l'animation, comme sur les différents aspects cognitifs et de perception. Comme par exemple capturer un état émotionnel du visage, et le retranscrire sur un visage virtuel, indépendamment de sa parole et de sa situation.

Bibliographie

- [ACLP98] L. Ambrosini, M. Costa, F. Lavagetto, and R. Pockaj. 3d head model calibration based on mpeg-4 parameters. In *The 6th ISPACS - IEEE International Workshop on Intelligent Signal Processing and Communication Systems*, pages 626–630, 1998. [13](#), [16](#)
- [AHK⁺02] Irene Albrecht, Jörg Haber, Kolja Kähler, Marc Schröder, and Hans-Peter Seidel. "may i talk to you? :-)" facial animation from text. In *PG '02 : Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, page 77, Washington, DC, USA, 2002. IEEE Computer Society. [40](#)
- [ARL⁺09a] Oleg Alexander, Mike Rogers, William Lambeth, Matt Chiang, and Paul Debevec. Creating a photoreal digital actor : The digital emily project. In *CVMP '09 : Proceedings of the 2009 Conference for Visual Media Production*, pages 176–187, Washington, DC, USA, 2009. IEEE Computer Society. [19](#), [31](#), [32](#), [43](#)
- [ARL⁺09b] Oleg Alexander, Mike Rogers, William Lambeth, Matt Chiang, and Paul Debevec. The digital emily project : photoreal facial modeling and animation. In *SIGGRAPH '09 : ACM SIGGRAPH 2009 Courses*, pages 1–15, New York, NY, USA, 2009. ACM. [19](#), [31](#), [32](#), [43](#), [84](#)
- [ASHS05] Irene Albrecht, Marc Schröder, Jörg Haber, and Hans-Peter Seidel. Mixed feelings : expression of non-basic emotions in a muscle-based talking head. *Virtual Real.*, 8(4) :201–212, 2005. [40](#)
- [Bal04] K. Balci. Xface : Mpeg-4 based open source toolkit for 3d facial animation. In *AVI '04 : Proceedings of the working conference on Advanced visual interfaces*, pages 399–402, New York, NY, USA, 2004. ACM. [16](#)
- [BB02] M. Byun and N. I. Badler. Facemote : qualitative parametric modifiers for facial animations. In *SCA '02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 65–71, New York, NY, USA, 2002. ACM. [40](#)

- [BBA⁺07] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister, and M. Gross. Multi-scale capture of facial geometry and motion. In *SIGGRAPH '07 : ACM Trans. on Graphics*, volume 26, 2007. 24, 32, 34, 38, 40
- [BBB⁺10] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross. High-quality single-shot capture of facial geometry. *ACM Trans. Graph.*, 29(4) :1–9, 2010. 33, 84
- [BCL00] F. Bourel, C. Chibelushi, and A. Low. Robust facial feature tracking. In *BMVC '00 : Proceedings of the 11th British Machine Vision Conference*, volume 1 of 1, pages 232–241, Bristol, England, September 2000. 30
- [Bec98] D. Bechmann. Multidimensional free-form deformation tools. In *State of The Art Report, EUROGRAPHICS' 98*, 1998. 23
- [BHN03] T. D. Bui, D. Heylen, and A. Nijholt. Improvements on a simple muscle-based 3d face for realistic facial expressions. In *CASA '03 : Proceedings of the 16th International Conference on Computer Animation and Social Agents*, page 33, Washington, DC, USA, 2003. IEEE Computer Society. 21
- [BHPS10] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer. High resolution passive facial performance capture. In *ACM SIGGRAPH 2010 papers, SIGGRAPH '10*, pages 41 :1–41 :10, New York, NY, USA, 2010. ACM. 33, 35, 84
- [BKMTK00] L. Boissieux, G. Kiss, N. Magnenat-Thalmann, and P. Kalra. Simulation of skin aging and wrinkles with cosmetics insight. In *Proc. of Eurographics Workshop on Animation and Simulation*, 2000. 38
- [BKN02] Y. Bando, T. Kuratate, and T. Nishita. A simple method for modeling wrinkles on human skin. In *Proc. of the 10th Pacific Conference on Computer Graphics and Applications*, pages 166–175, 2002. 38
- [BL79] S. Beucher and C. Lantuejoul. Use of Watersheds in Contour Detection. In *International Workshop on Image Processing : Real-time Edge and Motion Detection/Estimation, Rennes, France.*, September 1979. 57
- [BL85] P. Bergeron and P. Lachapelle. Controlling facial expressions and body movements in the computer generated animated short 'tony de peltrie'. In *SIGGRAPH '85 Tutorial Notes, Advanced Computer Animation Course*, 1985. 3, 27

- [BL03] G. Borshukov and J. P. Lewis. Realistic human face rendering for "the matrix reloaded". In *SIGGRAPH '03 : ACM SIGGRAPH 2003 Sketches & Applications*, New York, NY, USA, 2003. ACM. 3, 31
- [BLB⁺08] B. Bickel, M. Lang, M. Botsch, M. Otaduy, and M. Gross. Pose-space animation and transfer of facial details. In *SCA '08 : Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, July 2008. 24, 34, 38, 39
- [BLK01] Choe B., H. Lee, and H.-S. Ko. Performance-driven muscle-based facial animation. *The Journal of Visualization and Computer Animation*, 12(2) :67–79, 2001. 21, 36
- [BNZP07] K. Balci, E. Not, M. Zancanaro, and F. Pianesi. Xface open source project and smil-agent scripting language for creating and animating embodied conversational agents. In *ACM MULTIMEDIA '07 : Proceedings of the 15th international conference on Multimedia*, pages 1013–1016, New York, NY, USA, 2007. ACM. 16
- [Bou02] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker : Description of the algorithm. urlJean-Yves Bouguet, 2002. 75
- [BP04] E. Bevacqua and C. Pelachaud. Expressive audio-visual speech : Research articles. *Comput. Animat. Virtual Worlds*, 15(3-4) :297–304, 2004. 40
- [BP07] I. Baran and J. Popović. Automatic rigging and animation of 3d characters. In *SIGGRAPH '07 : ACM SIGGRAPH 2007 papers*, volume 26, page 72, New York, NY, USA, 2007. ACM. 46
- [BPL⁺03] G. Borshukov, D. Piponi, O. Larsen, J. P. Lewis, and C. Tempelaar-Lietz. Universal capture : image-based facial animation for "the matrix reloaded". In *SIGGRAPH '03 : ACM SIGGRAPH 2003 Sketches & Applications*, 2003. 3, 29, 31
- [CAG10] M. Courgeon, M.-A. Amorim, and J.-C. Giroux, C. and. Martin. Do users anticipate emotion dynamics in facial expressions of a virtual character? In *CASA '10 : Proceedings of the 23rd International Conference on Computer Animation and Social Agents*, 2010. 16
- [Can86] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8 :679–698, November 1986. 59
- [CB02] E. Chuang and C. Bregler. Performance driven facial animation using blendshape interpolation. Cstr-2002-02, Stanford University Computer Science, April 2002. 20, 35

- [CBC⁺01] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76, New York, NY, USA, 2001. ACM. [49](#)
- [CBM09] M. Courgeon, S. Buisine, and J.-C. Martin. Impact of expressive wrinkles on perception of a virtual character's facial expressions of emotions. In *Proc. of the 9th International Conf. on Intelligent Virtual Agents*, volume 5773 of *Lecture Notes in Computer Science*, pages 201–214. Springer Berlin / Heidelberg, 2009. [16](#), [83](#)
- [CET98] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In H. Burkhardt and B. Neumann, editors, *ECCV '98 : Proceedings of the 1998 European Conference on Computer Vision*, volume 2, pages 484–498. Springer, 1998. [31](#)
- [CET99] T. F. Cootes, G. Edwards, and C. J. Taylor. Comparing active shape models with active appearance models. In *BMVC '99 : Proceedings of the 10th British Machine Vision Conference*, volume 1, pages 173–182, September 1999. [31](#)
- [CJ06] E. Chang and O. C. Jenkins. Sketching articulation and pose for facial animation. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 271–280, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. [28](#)
- [CK01] B. Choe and H.-S. Ko. Analysis and synthesis of facial expressions with hand-generated muscle actuation basis. *Computer Animation, 2001. The Fourteenth Conference on Computer Animation. Proceedings*, pages 12–19, 2001. [19](#)
- [CM93] M. M. Cohen and D. W. Massaro. Modeling coarticulation in synthetic visual speech. In *Models and Techniques in Computer Animation*, pages 139–156. Springer-Verlag, 1993. [22](#)
- [Coq90] S. Coquillart. Extended free-form deformation : a sculpturing tool for 3d geometric modeling. In *SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 187–196, New York, NY, USA, 1990. ACM. [23](#)

- [CT01] T. F. Cootes and C. J. Taylor. Statistical models of appearance for medical image analysis and computer vision. In *SPIE Medical Imaging*, 2001. 31
- [CTCG95] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1) :38–59, January 1995. 30
- [CVB01] J. Cassell, H. H. Vilhjálmsón, and T. Bickmore. Beat : the behavior expression animation toolkit. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 477–486, New York, NY, USA, 2001. ACM. 40
- [CXH03] J.-X. Chai, J. Xiao, and J. Hodgins. Vision-based control of 3d facial animation. In *SCA '03 : Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 193–206, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 30
- [DCFN06] Z. Deng, P.-Y. Chiang, P. Fox, and U. Neumann. Animating blendshape faces by cross-mapping motion capture data. In *I3D '06 : Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 43–48, New York, NY, USA, 2006. ACM Press. 20, 35, 37
- [Der90] R. Deriche. Fast algorithms for low-level vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12 :78–87, January 1990. 59
- [DFS08] J.-D. Durou, M. Falcone, and M. Sagona. Numerical methods for shape-from-shading : A new survey with benchmarks. *Comput. Vis. Image Underst.*, 109 :22–43, January 2008. 39
- [DMB08] L. Dutreve, A. Meyer, and S. Bouakaz. Feature points based facial animation retargeting. In *VRST '08 : 15th ACM Symposium on Virtual Reality Software and Technology*, pages 197–200, October 2008. 70, 79
- [DMB09] L. Dutreve, A. Meyer, and S. Bouakaz. Real-time dynamic wrinkles of face for animated skinned mesh. In *Proceedings of the 5th International Symposium on Advances in Visual Computing : Part II*, ISVC '09, pages 25–34, Berlin, Heidelberg, 2009. Springer-Verlag. 106
- [DMB11] L. Dutreve, A. Meyer, and S. Bouakaz. Easy acquisition and real-time animation of facial wrinkles. *Proceedings of the 2008 Computer Animation and Social Agents (CASA2010)/ Journal of Computer Animation and Virtual World (to appear)*, 2011. 106

- [DMOB10] L. Dutrevez, A. Meyer, V. Orvalho, and S. Bouakaz. Easy rigging of face by automatic registration and transfer of skinning parameters. In *Proceedings of the 2010 international conference on Computer vision and graphics : Part I, ICCVG'10*, pages 333–341, Berlin, Heidelberg, 2010. Springer-Verlag. 70
- [DN06] Z. Deng and U. Neumann. eface : expressive facial animation synthesis and editing with phoneme-isomap controls. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 251–260, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. 40
- [DN07] Z. Deng and J. Y. Noh. *Computer Facial Animation : A Survey*, chapter 1, pages 1–28. Springer, Ulrich Neumann, Zhigang Deng (eds.), September 2007. 10, 56
- [DRSV02] D. DeCarlo, C. Revilla, M. Stone, and J. Venditti. Making discourse visible : Coding and animating conversational facial displays. In *In Proc. Computer Animation 2002*, pages 11–16, 2002. 40
- [EBDP96] I. Essa, S. Basu, T. Darrell, and A. Pentland. Modeling, tracking and interactive animation of faces and heads using input from video. In *CA '96 : Proceedings of the Computer Animation*, page 68, Washington, DC, USA, 1996. IEEE Computer Society. 29, 33
- [ED07] N. Ersotelos and F. Dong. Building highly realistic facial modeling and animation : a survey. *Vis. Comput.*, 24 :13–30, 2007. 84
- [EDP94] I. A. Essa, T. Darrell, and A. Pentland. Tracking facial motion. In *In Proceedings of the Workshop on Motion of Nonrigid and Articulated Objects*, pages 36–42. IEEE Computer Society, 1994. 29
- [EF78] P. Ekman and W. V. Friesen. *Facial Action Coding System (FACS) : A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, 1978. 14, 76
- [EFH02] P. Ekman, W. V. Friesen, and J. C. Hager. *Facial Action Coding System*. A Human Face, 2002. 14
- [EP97] I. A. Essa and A. P. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7) :757–763, 1997. 33
- [GB08] J. Gain and D. Bechmann. A survey of spatial deformation from a user-centered perspective. *ACM Trans. Graph.*, 27 :107 :1–107 :21, November 2008. 84

- [GEMT05] S. Garchery, A. Egges, and N. Magnenat-Thalmann. Fast facial animation design for emotional virtual humans. In *Proc. Measuring Behaviour, Wageningen, N.L.* CD-ROM Proceeding, September 2005. 16
- [GGMT07] T. Di Giacomo, S. Garchery, and N. Magnenat-Thalmann. *Expressive Visual Speech Generation*, chapter 2, pages 29–59. Springer, September 2007. 40
- [GGW⁺98] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 55–66, New York, NY, USA, 1998. ACM. 12, 23, 29
- [GLB07] E. Gu, J.B. Lee, S. P. and Badler, and N. I. Badler. *Eye Movements, Saccades, and Multiparty Conversations*, chapter 4, pages 79–97. Springer, Ulrich Neumann, Zhigang Deng (eds.), September 2007. 40
- [Gle98] M. Gleicher. Retargetting motion to new characters. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 33–42, New York, NY, USA, 1998. ACM. 72
- [HBSL04] V. Helzle, C. Biehn, T. Schl"omer, and F. Linner. Adaptable setup for performance driven facial animation. In *SIGGRAPH '04 : ACM SIGGRAPH 2004 Sketches*, page 54, New York, NY, USA, 2004. ACM. 23, 34
- [HCD01] T. Hawkins, J. Cohen, and P. Debevec. A photometric approach to digitizing cultural artifacts. In *VAST '01 : Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, pages 333–342, New York, NY, USA, 2001. ACM. 31
- [Hej04] J. Hejl. Hardware skinning with quaternions. pages 487–495, 2004. 25
- [HFG94] M. Hoch, G. Fleischmann, and B. Girod. Modeling and animation of facial expressions based on b-splines. *Visual Computer*, 11 :87–95, 1994. 23
- [HR98] C. R. Harrison and K. M. Robinette. Caesar : Summary statistics for the adult population (ages 18-65) of the united states of america. Technical report, Defense Technical Information Center OAI-PMH Repository [<http://stinet.dtic.mil/oai/oai>] (United States), 1998. 52
- [HWT⁺04] T. Hawkins, A. Wenger, C. Tchou, A. Gardner, F. G"oransson, and P. Debevec. Animatable facial reflectance fields. In *Rendering Tech-*

- niques 2004 : 15th Eurographics Workshop on Rendering*, pages 309–320, jun 2004. [31](#)
- [JL08] E. Ju and J. Lee. Expressive facial gestures from motion capture data. In G. Drettakis and R. Scopigno, editors, *Computer Graphics Forum (Proceedings of EUROGRAPHICS '08)*, volume 27, pages 381–388, April 2008. [40](#), [41](#)
- [JLP04] C. Je, S. W. Lee, and R.-H. Park. High-contrast color-stripe pattern for rapid structured-light range imaging. *Computer Vision - ECCV*, 3021 :95–107, 2004. [33](#)
- [JTDP03] P. Joshi, W. C. Tien, M. Desbrun, and F. Pighin. Learning controls for blend shape based realistic facial animation. In *SCA '03 : Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 1–1, New York, NY, USA, 2003. ACM. [19](#), [35](#)
- [JZP⁺08] T. Ju, Q.-Y. Zhou, M. van de Panne, D. Cohen-Or, and U. Neumann. Reusable skinning templates using cage-based deformations. *ACM Trans. Graph.*, 27 :122 :1–122 :10, December 2008. [24](#), [46](#)
- [KCZO07] L. Kavan, S. Collins, J. Zara, and C. O’Sullivan. Skinning with dual quaternions. In *I3D '07 : ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 39–46. ACM Press, April/May 2007. [25](#)
- [KG03] H. Kono and E. Genda. Wrinkle generation model for 3d facial expression. In *ACM SIGGRAPH 2003 Sketches & Applications*, pages 1–1, 2003. [38](#), [39](#)
- [KHS01] K. Kähler, J. Haber, and H.-P. Seidel. Geometry-based muscle modeling for facial animation. In *GRIN'01 : No description on Graphics interface 2001*, pages 37–46, Toronto, Ont., Canada, Canada, 2001. Canadian Information Processing Society. [21](#)
- [KHS03] K. Kähler, J. Haber, and H.-P. Seidel. Reanimating the dead : reconstruction of expressive faces from skull data. In *SIGGRAPH '03 : ACM SIGGRAPH 2003 Papers*, pages 554–561, New York, NY, USA, 2003. ACM. [13](#), [21](#)
- [KHYS02] K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel. Head shop : generating animated head models with anatomical structure. In *SCA '02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 55–63, New York, NY, USA, 2002. ACM. [13](#), [21](#), [26](#)

- [KJP02] P. G. Kry, D. James, and D. Pai. Eigenskin : Real time large deformation character skinning in hardware. In *ACM SIGGRAPH/Eurographics SCA*, pages 153 – 160, July 2002. 24
- [KM04a] M. Kouchi and M. Mochimaru. Analysis of 3d face forms for proper sizing and cad of spectacle frames. *Ergonomics*, 47 :1499–1516, 2004. 52
- [KM04b] T. Kurihara and N. Miyata. Modeling deformable human hands from medical images. In *SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 357– 365, 8 2004. 43
- [KMMTT91] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann. SMILE : a multilayered facial animation system. In *Proceedings of the IFIP WG 5.10 Working Conference*, 1991. Centre d’Inf., Geneva Univ., Switzerland. 14
- [KMMTT92] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann. Simulation of facial muscle actions based on rational free form deformations. In A. Kilgour and L. Kjelldahl, editors, *Computer Graphics Forum (Proceedings of EUROGRAPHICS '92)*, volume 11, pages 59–69, 1992. 23
- [KPL98] C. Kouadio, P. Poulin, and P. Lachapelle. Real-time facial animation based upon a bank of 3d facial expressions. In *CA '98 : Proceedings of the Computer Animation*, page 128, Washington, DC, USA, 1998. IEEE Computer Society. 20, 35
- [KWT88] M. Kass, A. Witkin, and d. Terzopoulos. Snakes : Active contour models. *International Journal of Computer Vision*, 1(4) :321–331, 1988. 30
- [LA10] J. P. Lewis and K.-I. Anjyo. Direct manipulation blendshapes. *IEEE Computer Graphics and Applications*, 30 :42–50, 2010. 19
- [LAGP09] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.*, 28(5), December 2009. 40
- [LBL09] G. C. Littlewort, M. S. Bartlett, and K. Lee. Automatic coding of facial expressions displayed during posed and genuine pain. *Image Vision Comput.*, 27(12) :1797–1803, 2009. 15
- [LC04] C. Larboulette and M.-P. Cani. Real-time dynamic wrinkles. In *Computer Graphics International*, 2004. 38

- [LCF00] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation : a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. [24](#), [43](#), [109](#)
- [LCXS07] M. Lau, J.-X. Chai, Y.-Q. Xu, and H.-Y. Shum. Face poser : Interactive modeling of 3d facial expressions using model priors. In *SCA '07 : Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on computer animation*, August 2007. [28](#)
- [LD08] Q. Li and Z. Deng. Orthogonal-blendshape-based editing system for facial motion capture data. *IEEE Comput. Graph. Appl.*, 28(6) :76–82, 2008. [19](#), [20](#)
- [LDRS05] N. Litke, M. Droske, M. Rumpf, and P. Schröder. An image processing approach to surface matching. In *SGP '05 : Proceedings of the third Eurographics symposium on Geometry processing*, 2005. [56](#)
- [LH09] G. S. Lee and F. Hanner. Practical experiences with pose space deformation. In *SIGGRAPH ASIA '09 : ACM SIGGRAPH ASIA 2009 Sketches*, pages 1–1, New York, NY, USA, 2009. ACM. [43](#)
- [LKCL98] J. J. Lien, T. Kanade, J. F. Cohn, and C.-C. Li. Automated facial expression recognition based on face action units. In *FG '98 : Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, page 390, Washington, DC, USA, 1998. IEEE Computer Society. [15](#)
- [LLZ⁺08] Y.-S. Lo, I.-C. Lin, W.-X. Zhang, W.-C. Tai, and S.-J. Chiou. Capturing facial details by space-time shape-from-shading. In *CGI '08 : Proceedings of the Computer Graphics International 2008*, pages 118–125, June 2008. [32](#), [38](#), [39](#)
- [LMDN05] J. P. Lewis, J. Mooser, Z. Deng, and U. Neumann. Reducing blendshape interference by selected motion attenuation. In *I3D '05 : Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 25–29, New York, NY, USA, 2005. ACM Press. [17](#), [19](#), [20](#)
- [LMX⁺08] X. Liu, T. Mao, S. Xia, Y. Yu, and Z. Wang. Facial animation by optimized blendshapes from motion capture data. *Comput. Animat. Virtual Worlds*, 19(3-4) :235–245, 2008. [20](#), [35](#)
- [LN00] T. Lauthelien and M. Neveu. Facial animation by reverse morphing on a sequence of real images : application to film and vi-

- deo production. *Annals of Telecommunications*, 55 :143–148, 2000. 10.1007/BF03001907. **35**
- [LP99] F. Lavagetto and R. Pockaj. The facial animation engine : towards a high-level interface for the design of mpeg-4 compliant animated faces. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(2) :277–289, 1999. **16**
- [LP01] F. Lavagetto and R. Pockaj. An efficient use of mpeg-4 fap interpolation for facial animation at 70 bit/frame. *IEEE Trans. on Circuits and Systems for Video Technology*, 1(10), October 2001. **16**
- [LSZ01] Z. Liu, Y. Shan, and Z. Zhang. Expressive expression mapping with ratio images. In *Proc. of the 28th annual conference on Computer graphics and interactive techniques*, pages 271–276, 2001. **86**
- [LTW95] Y. Lee, D. Terzopoulos, and K. Walters. Realistic modeling for facial animation. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 55–62, New York, NY, USA, 1995. ACM. **12, 21**
- [LWP10] H. Li, T. Weise, and M. Pauly. Example-based facial rigging. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2010)*, 29(3), July 2010. **19, 20**
- [MG03] A. Mohr and M. Gleicher. Building efficient, accurate character skins from examples. In *SIGGRAPH '03 : ACM SIGGRAPH 2003 papers*, volume 22, pages 562–568, New York, NY, USA, 2003. ACM. **24**
- [MPE01] MPEG4. Iso/iec 14496-2 - mpeg-4 - coding of audio-visual objects - part 2 - visual, 2001. **14, 15, 16**
- [MTLT88] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings on Graphics interface '88*, pages 26–33, Toronto, Ont., Canada, Canada, 1988. Canadian Information Processing Society. **24**
- [MTPT87] N. Magnenat-Thalmann, E. Primeau, and D. Thalmann. Abstract muscle action procedures for human face animation. *The Visual Computer*, 3(5) :290–297, 1987. **14**
- [MVR⁺95] A. M. McIvor, R. J. Valkenburg, I. Robert, J. Valkenburg, and J. Valkenburg. Calibrating a structured light system. Technical report, Industrial Research Limited, 1995. **33**
- [MW99] A. P. Mangan and R. T. Whitaker. Partitioning 3d surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 5(4) :308–321, 1999. **58**

- [NF07] G. Nataneli and P. Faloutsos. Sketching facial expressions. In *SIGGRAPH '07 : ACM SIGGRAPH 2007 sketches*, page 60, New York, NY, USA, 2007. ACM. [28](#)
- [NHS88] M. Nahas, H. Huitric, and M. Saintourens. Animation of a b-spline figure. *The Visual Computer*, 3 :272–276, 1988. [23](#)
- [NJ04] K. Na and M. Jung. Hierarchical retargetting of fine facial motions. *Computer Graphics Forum (Proceedings of EUROGRAPHICS '04)*, 23(3) :687–695, 2004. [33](#), [34](#), [36](#), [38](#), [39](#)
- [NN98] J.-Y. Noh and U. Neumann. A survey of facial modeling and animation techniques. Tr-99-705, USC, 1998. [9](#), [10](#)
- [NN01] J.-Y. Noh and U. Neumann. Expression cloning. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 277–288, New York, NY, USA, 2001. ACM. [36](#), [37](#), [38](#), [56](#), [69](#), [72](#), [77](#)
- [NSACO05] A. Nealen, O. Sorkine, M. Alexa, and D. Cohen-Or. A sketch-based interface for detail-preserving mesh editing. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 papers*, volume 24, pages 1142–1147, New York, NY, USA, 2005. ACM. [28](#)
- [Oat07] C. Oat. Animated wrinkle maps. In *ACM SIGGRAPH 2007 courses*, pages 33–37, 2007. [38](#), [39](#), [84](#), [105](#)
- [Orv07] V. C. Orvalho. *Reusable Facial Rigging And Animation : Create Once, Use Many*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, June 2007. [42](#)
- [Ost98] J. Ostermann. Animation of synthetic faces in mpeg-4. In *CA '98 : Proceedings of the Computer Animation*, page 49, Washington, DC, USA, 1998. IEEE Computer Society. [15](#)
- [OW04] J. Ostermann and A. Weissenfeld. Talking faces - technologies and applications. In *ICPR '04 : Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3*, pages 826–833, Washington, DC, USA, 2004. IEEE Computer Society. [10](#)
- [OZS06] V. C. Orvalho, E. Zacur, and A. Susin. Transferring a labeled generic rig to animate face models. In *AMDO '06 : 4th International Conference on Articulated Motion and Deformable Objects*, pages 223–233, 2006. [26](#), [47](#), [56](#)
- [Par72] F. I. Parke. Computer generated animation of faces. In *ACM '72 : Proceedings of the ACM annual conference*, pages 451–457, New York, NY, USA, 1972. ACM. [1](#), [9](#), [27](#)

- [Par74a] F. I. Parke. A model for human faces that allows speech synchronized animation. In *SIGGRAPH '74 : Proceedings of the 1st annual conference on Computer graphics and interactive techniques*, pages 2–2, New York, NY, USA, 1974. ACM. 22
- [Par74b] F. I. Parke. *A parametric model for human faces*. PhD thesis, 1974. 22, 27
- [Par82] F. I. Parke. Parameterized models for facial animation. *IEEE Comput. Graph. Appl.*, 2(9) :61–68, 1982. 22
- [PB81] S. M. Platt and N. I. Badler. Animating facial expressions. *SIGGRAPH Comput. Graph.*, 15(3) :245–252, 1981. 20
- [PBS96] C. Pelachaud, N. I. Badler, and M. Steedman. Generating facial expressions for speech. *Cognitive Science*, 20 :1–46, 1996. 40
- [PCLB01] R. Pockaj, M. Costa, F. Lavagetto, and C. Braccini. A solution for model-independent animation of mpeg-4 faces. In *ICAV3D'01 International Conference on Augmented, Virtual Environments and 3D Imaging*, pages 327–330, 2001. 16
- [PCMTT96] I. Pandzic, T. Capin, N. Magnenat-Thalmann, and D. Thalmann. Towards Natural Communication in Networked Collaborative Virtual Environments. In *Proc.FIVE 1996*, 1996. 30
- [PCNS05] B. Park, H. Chung, T. Nishita, and S. Y. Shin. A feature-based approach to facial expression cloning : Virtual humans and social agents. *Comput. Animat. Virtual Worlds*, 16(3-4) :291–303, 2005. 19
- [PF05] E. Prados and O. Faugeras. Shape from shading : a well-posed problem? In *Proc. of CVPR*, volume II, pages 870–877. IEEE, 2005. 39
- [PHL⁺98] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 75–84, New York, NY, USA, 1998. ACM. 13, 27, 37
- [PKC⁺03] H. Pyun, Y. Kim, W. Chae, H. W. Kang, and S. Y. Shin. An example-based approach for facial expression cloning. In *SCA '03 : Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 167–176, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 19, 20, 37

- [PKMTT94] I. S. Pandzic, P. Kalra, N. Magnenat-Thalmann, and D. Thalmann. Real-time facial interaction. *Displays*, 15(3) :157–63, 1994. MIRALab, Geneva Univ., Switzerland. 30
- [PL06] F. Pighin and J. P. Lewis. Introduction (session : Performance-driven facial animation). In *SIGGRAPH '06 : ACM SIGGRAPH 2006 Courses*, page 1, New York, NY, USA, 2006. ACM. 10, 36, 37
- [Pow87] M. J. D. Powell. Radial basis functions for multivariable interpolation : a review. pages 143–167, 1987. 50
- [PP01] S. Pasquariello and C. Pelachaud. Greta : A simple facial animation engine. In *In Proc. of the 6th Online World Conference on Soft Computing in Industrial Applications*, 2001. 16
- [PPGG10] C. Puklavage, A. R. Pirela, A. J. Gonzalez, and M. Georgiopoulos. Imitating personalized expressions in an avatar through machine learning. In *Twenty-Third International FLAIRS Conference*, pages 68–73, 2010. 15
- [RDW09] L. Rothkrantz, D. Datcu, and P. Wiggers. Facs-coding of facial expressions. In *CompSysTech '09 : Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing*, pages 1–6, New York, NY, USA, 2009. ACM. 15
- [RKH07] A. Robles-Kelly and E. R. Hancock. Shape-from-shading using the heat equation. *IEEE Trans. on Image Processing*, 16, 2007. 89
- [RLN06] T. Rhee, J. P. Lewis, and U. Neumann. Real-time weighted pose-space deformation on the gpu. In *Computer Graphics Forum (Proceedings of EUROGRAPHICS '06)*, volume 25, pages 439–448, September 2006. 43
- [RP06] M. Radovan and L. Pretorius. Facial animation in a nutshell : past, present and future. In *SAICSIT '06 : Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, pages 71–79, Republic of South Africa, 2006. South African Institute for Computer Scientists and Information Technologists. 10
- [Sag06] M. Sagar. Facial performance capture and expressive translation for king kong. In *SIGGRAPH '06 : ACM SIGGRAPH 2006 Sketches*, page 26, New York, NY, USA, 2006. ACM. 3, 19

- [SBS10] N. Stoiber, G. Breton, and R. Séguier. Modeling short-term dynamics and variability for realistic interactive facial animation. *IEEE Computer Graphics and Applications*, 30 :51–61, 2010. 28
- [SGBG09] R. Séguier, S. Le Gallou, G. Breton, and C. Garcia. Adapted active appearance models. *J. Image Video Process.*, 2009 :13 :2–13 :2, February 2009. 31
- [SH08] W. A. Smith and E. R. Hancock. Facial shape-from-shading and recognition using principal geodesic analysis and robust statistics. *Int. J. Comput. Vision*, 76 :71–91, 2008. 40
- [SMFF07] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image Vision Comput.*, 25(5) :578–596, 2007. 55, 59
- [SNF05] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *SIGGRAPH '05 : ACM SIGGRAPH 2005 papers*, 24(3) :417–425, 2005. 21, 22, 36, 42
- [Sob01] J. Sobotta. *Atlas of Human Anatomy, Volume 1 : Head, Neck, Upper Limb*. Lippincot Williams and Wilkins, 13th english / english edition, 2001. 13
- [SP86] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20(4) :151–160, 1986. 23
- [SSB09] N. Stoiber, R. Séguier, and G. Breton. Automatic design of a control interface for a synthetic face. In *Proceedings of the 13th international conference on Intelligent user interfaces, IUI '09*, pages 207–216, New York, NY, USA, 2009. ACM. 28
- [SSRMF06] E. Sifakis, A. Selle, A. Robinson-Mosher, and R. Fedkiw. Simulating speech with a physics-based facial muscle model. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 261–270, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. 21, 40, 42
- [SSSB09] A. Sattar, N. Stoiber, R. Séguier, and G. Breton. Gamer’s facial cloning for online interactive games. 2009. 31
- [TLY04a] S.-S. Tang, A. W.-C. Liew, and H. Yan. Lip-sync in human face animation based on video analysis and spline models. In *MMM '04 : Proceedings of the 10th International Multimedia Modelling Conference*, page 102, Washington, DC, USA, 2004. IEEE Computer Society. 21

- [TLY⁺04b] P.-H. Tu, I.-C. Lin, J.-S. Yeh, R.-H. Liang, and M. Ouhyoung. Surface detail capturing for realistic facial animation. *J. Comput. Sci. Technol.*, 19(5) :618–625, 2004. 32, 38
- [TW90] D. Terzopoulos and K. Waters. Physically-based facial modeling, analysis, and animation. *The Journal of Visualization and Computer Animation*, 2 :73–80, 1990. 21
- [VBPP05] D. Vlastic, M. Brand, H. Pfister, and J. Popovi'c. Face transfer with multilinear models. *SIGGRAPH '05 : ACM SIGGRAPH 2005 papers*, 24(3) :426–433, 2005. 37
- [VLR05] K. Venkataramana, S. Lodhaa, and R. Raghava. A kinematic-variational model for animating skin with wrinkles. *Computers & Graphics*, 29 :756–770, October 2005. 38
- [VMT99] P. Volino and N. Magnenat-Thalmann. Fast geometrical wrinkles on animated surfaces. In *Proc. of the 7-th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media*, 1999. 38
- [VVBW09] O. Vogel, L. Valgaerts, M. Breuß, and J. Weickert. Making shape from shading work for real-world images. pages 191–200, Berlin, Heidelberg, 2009. 39
- [Wat87] K. Waters. A muscle model for animation three-dimensional facial expression. In *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 17–24, New York, NY, USA, 1987. ACM. 21
- [WF94] C. L. Wang and D. R. Forshey. Langwidere : A new facial animation system. Technical report, Vancouver, BC, Canada, Canada, 1994. 15, 21, 23
- [WGP⁺10] C. A. Wilson, A. Ghosh, P. Peers, J.-Y. Chiang, J. Busch, and P. Debevec. Temporal upsampling of performance geometry using photometric alignment. *ACM Trans. Graph.*, 29 :17 :1–17 :11, 2010. 84
- [WHL⁺04] Y. Wang, X. Huang, C. S. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. In *Computer Graphics Forum (Proceedings of EUROGRAPHICS '04)*, pages 677–686, 2004. 33
- [Wil90] L. Williams. Performance-driven facial animation. In *SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics*

- and interactive techniques*, pages 235–242, New York, NY, USA, 1990. ACM. 27, 29, 71
- [Wil06] L. Williams. Performance-driven facial animation. In *SIGGRAPH '06 : ACM SIGGRAPH 2006 Courses*, page 16, New York, NY, USA, 2006. ACM. 10
- [WJL09] T.-H. Wang and J.-J. James Lien. Facial expression recognition system based on rigid and non-rigid motion separation and 3d pose estimation. *Pattern Recogn.*, 42(5) :962–977, 2009. 15
- [WKMT96] Y. Wu, P. Kalra, and N. Magnenat-Thalmann. Simulation of static and dynamic wrinkles. In *Proc. Computer Animation '96*, pages 90–97, 1996. 38
- [WL93] K. Waters and T. M. Levergood. Decface : An automatic lip-synchronization algorithm for synthetic faces. Technical report, Multimedia Tools and Applications, 1993. 27, 40
- [WLGP09] T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/off : Live facial puppetry. In *SCA '09* : Eurographics Association, August 2009. 33
- [WLVG07] T. Weise, B. Leibe, and L. Van Gool. Fast 3d scanning with automatic motion compensation. In *CVPR*, pages 1–8, June 2007. 84
- [WO06] J. Whitehill and C. W. Omlin. Haar features for face recognition. In *FGR '06 : Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, pages 97–101, Washington, DC, USA, 2006. IEEE Computer Society. 15
- [WSLG07] O. Weber, O. Sorkine, Y. Lipman, and C. Gotsman. Context-aware skeletal shape deformation. *Computer Graphics Forum (Proceedings of Eurographics)*, 26(3), 2007. 43
- [WSQO05] A. Weissenfeld, N. Stefanoski, S. Qiuqiong, and J. Ostermann. Adaptation of a generic face model to a 3d scan, berlin, germany. In *ICOB 2005 - Workshop On Immersive Communication And Broadcast Systems*, volume 0, 2005. 13
- [WSTS08] T.-P. Wu, J. Sun, C.-K. Tang, and H.-Y. Shum. Interactive normal reconstruction from a single image. *ACM Trans. Graph.*, 27 :119 :1–119 :9, 2008. 39, 40
- [YY09] Y. Yoshiyasu and N. Yamazaki. Pose space deformation with rotation-invariant details. In *SIGGRAPH ASIA '09 : ACM SIGGRAPH ASIA 2009 Sketches*, pages 1–1, New York, NY, USA, 2009. ACM. 43
- [ZC05] Cuiping Zhang and Fernand Cohen. Component-based active appearance models for face modelling. *3832* :206–212, 2005. 31

- [ZF03] B. Zitova and J. Flusser. Image registration methods : a survey. *Image and Vision Computing*, 2003. 59
- [ZPS01] Y. Zhang, E. C. Prakash, and E. Sung. Real-time physically-based facial expression animation using mass-spring system. In *CGI '01 : Proceedings of the International Conference on Computer Graphics*, page 347, Washington, DC, USA, 2001. IEEE Computer Society. 15, 21
- [ZSCS04] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces : High-resolution capture for modeling and animation. In *ACM Annual Conference on Computer Graphics*, pages 548–558, August 2004. 38
- [ZST04] Y. Zhang, T. Sim, and C. L. Tan. Rapid modeling of 3d faces for animation using an efficient adaptation algorithm. In *GRAPHITE '04 : Proc, International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, pages 173–181, June 2004. 26

Glossaire

AAM

Active Appearance Model, modèle actif d'apparence, algorithme basé sur les contours et sur les informations de texture pour la mise en correspondance d'un visage statistique sur un visage réel.. 30

ACP

Analyse en Composante Principale. 36

AMA

Abstract Muscle Action. 14

API

Application Programming Interface, ensemble de fonctions, procédures ou classes mises à disposition par une bibliothèque logicielle, un système d'exploitation ou un service.. 99

ASM

Active Shape Model, modèle actif de forme, algorithme basé sur les contours pour la mise en correspondance d'un visage statistique sur un visage réel.. 30

AU

Action Unit, Unité d'action du FACS. 14, 22, 33

blendshape

Littéralement mélange de formes, utilisé également pour définir l'ensemble des techniques utilisant l'interpolation de formes.. 16–19, 25, 27, 31, 33, 34, 36, 41–43, 72, 98, 101, 109

bump-mapping

Technique qui consiste à simuler un relief à la surface d'un objet en modifiant sa normale en chaque point, et donc la façon dont la surface est éclairée.. 10, 38, 39, 84, 105, 107, 108

CPU

Central Processing Unit, Processeur. 99, 100

FACS

Facial Action Coding System, système de taxinomie des expressions du visage humain développé par le psychologue Paul Ekamn en 1978. 14, 20, 22, 33

FAP

Facial Animation Parameters, paramètres d'animation du visage de la norme MPEG-4.. 15

FAPU

Facial Animation Parameter Units, unités utilisées par les FAP basées sur les mesures anthropométriques du visage.. 15

FDP

Facial Definition Parameters , points caractéristiques du visage de la norme MPEG-4.. 15, 52, 53

GPU

Graphic Processor Unit, Processeur graphique. 25, 85, 94, 100

LBS

Linear Blend Skinning, voir skinning. 24

matrice *modelview*

Matrice de transformation qui permet de passer du repère monde au repère caméra.. 61

matrice *projection*

Matrice qui permet de projeter les points 3D sur l'espace 2D de la caméra.. 61

matrice *viewport*

Matrice qui adapte la projection de la scène en fonction de la fenêtre de rendu.. 61

MMORPG

Massively multiplayer online role-playing game, Jeu de rôle en ligne massivement multijoueur, aussi appelé *MEUPORG* par certains journalistes.. 3

MoCap

MOtion CAPture, Capture de mouvement. 34, 36, 85, 108

MPA

Minimum Perceptible Action. 14, 29

MPEG-4

Norme de codage d'objets audiovisuels spécifiée par le *Moving Picture Expert Group*. 12, 14, 15, 52, 53, 66

OpenGL

Open Graphics Library, API multiplateforme pour la conception d'applications graphiques.. 99

PDFA

Performance-Driven Facial Animation, Animation faciale par capture des mouvements d'un acteur). 26, 71

PFDA

Performance-Driven Facial Animation, animation faciale pilotée par un un acteur réel.. 29

phonème

Plus petite unité de son associée à la parole.. 40

PID

Pondération inverse à la distance. 65

pixel shader

Tâche au sein du pipeline graphique dont le but est de calculer la couleur de chaque pixel individuellement.. 100

rastérisation

Algorithme qui permet de projeter un triangle 3D (scène virtuelle) sur un plan 2D (écran). Il intègre une fonction d'interpolation des paramètres (couleur, texture, etc.) entre les sommets 3D du triangle ou du polygone et chaque pixel 2D dessiné.. 91, 94, 98, 100

RBF

Radial Basis Function, Fonctions à base radiale. 12, 23, 34, 36, 47, 48, 50, 52, 56, 60, 61, 64, 72, 75, 76, 78, 86, 89, 108

rigging

Terme usuellement utilisé pour définir une technique de paramétrisation d'un objet virtuel pour l'animation.. 2, 10

shape-from-shading

Techniques pour l'estimation de surface à partir des variations d'éclairage.. 39

skinning

Appellation qui désigne l'ensemble des techniques de déformation de maillage à partir d'un squelette ou d'une hiérarchie d'articulations.. 6–8, 24, 25, 39, 43, 45–47, 54, 61, 63, 72, 84, 85, 91, 94, 95, 99, 101, 104, 105, 107–109

SSD

Skeleton Subspace Deformation, voir skinning. 24

vertex shader

Tâche au sein du pipeline graphique dont le but est de calculer la projection de coordonnées des sommets des primitives à partir de l'espace 3D dans l'espace écran.. 100

visème

Unité de représentation visuelle associée à un son (phonème).. 15, 27, 36, 40

Z-Buffer

Carte de profondeur que l'on obtient sur de la carte graphique qui fournit la distance qui espace chaque point d'une scène 3D de la caméra virtuelle.. 56, 61, 68