Clonage réaliste de visage.

Jérôme Manceau

CentraleSupélec

N° d'ordre : 2016-04-TH

CentraleSupélec

Ecole Doctorale MATISSE

« Mathématiques, Télécommunications, Informatique, Signal, Systèmes Electroniques »

Institut d'Électronique et de Télécommunication de Rennes

THÈSE DE DOCTORAT

DOMAINE : STIC
Spécialité : Traitement du Signal

Soutenue le 04 mai 2016

par :

Jérôme MANCEAU

Clonage réaliste de visage

Composition du jury :

Examinateur :	Pascal BOURDON	Maitre de conférences (Université de Poitiers)
Rapporteur :	Mohamed DAOUDI	Professeur (Télécom Lille 1)
Examinateur :	Kidiyo KPALMA	Professeur (Insa de Rennes), Président du jury
Rapporteur :	Alain PRUSKI	Professeur des Universités (Université de Lorraine)
Directeur de thèse :	Renaud SÉGUIER	Professeur (CentraleSupélec Rennes)
Encadrant :	Catherine SOLADIÉ	Professeur adjoint (CentraleSupélec Rennes)

Le visage est un livre où l'on peut lire d'étranges choses.
William Shakespeare - Macbeth - 1605

Remerciements

Ce mémoire de thèse est le résultat d'un travail effectué pendant trois ans dans les locaux de CentraleSupélec au sein des équipes Signal, Communication et Électronique Embarquée (SCEE pendant 1 an) et Facial, Analysis, Synthesis and Tracking (FAST pendant 2 ans). Elles sont membres de l'Institut d'électronique et de télécommunications de Rennes (IETR)- UMR CNRS 6164.

Je souhaite en tout premier lieu remercier Mohamed Daoudi, Professeur à Télécom Lille 1, ainsi que Alain Pruski, Professeur à l'université de Lorraine, de me faire honneur d'avoir accepté la charge de rapporteur de cette thèse. Je remercie également Pascal Bourdon, Maître de Conférence à l'université de Poitiers et Kidiyo Kpalma, Professeur à l'Insa de Rennes, d'avoir accepté de juger mon travail en tant que membre du jury.

Je tiens à remercier Renaud Séguier, mon directeur de thèse pour m'avoir donné l'opportunité de commencer et de poursuivre ce travail de recherche au sein de l'équipe FAST. Sa grande disponibilité malgré un emploi du temps très chargé et ses remarques toujours pertinentes ont été précieuses pour la réussite de ce travail. Je tiens aussi à remercier Catherine Soladié, mon encadrante, pour m'avoir guidée et suivie tout au long de ma thèse, ainsi que pour ses conseils qui m'ont permis de bien mener ce projet.

Je tiens à saluer et remercier chaleureusement les membres des équipes (post-doctorants, doctorants et stagiaires) que j'ai côtoyés au quotidien pour l'accueil chaleureux durant ces trois années. Merci à vous Ziad, Salma, Patricia, Hanan, Caroline, Abdel, Samba, Oussama, Lama, Marwa, Abir, Eren, Lara, Amine, Alya, Vincent G, Malek, Vincent S, Quentin, Rémi, Takoua, Raphaël, Salah, Slim, Corentin et tous les autres que j'ai oubliés. L'ambiance chaleureuse et stimulante au sein de CentraleSupélec m'a beaucoup aidé personnellement et professionnellement.

Je remercie l'ensemble des membres des équipes SCEE et FAST ainsi que tous les permanents de CentraleSupélec que j'ai pu côtoyer durant ces trois ans pour leur accueil sympathique.

Je remercie les membres de l'équipe FAST et des entreprises 3D Sound Labs et Dynamixyz pour leurs précieux conseils pendant les nombreux workshop de l'équipe.

Je tiens également à remercier le personnel de CentralSupélec du campus de Rennes pour sa disponibilité et sa bonne humeur au quotidien.

Pour finir, je ne pourrais clore ces remerciements sans remercier du fond du cœur ma famille pour leurs soutiens inconditionnels et leurs encouragements. Merci à ma mère, mon père, mes 2 sœurs et mon beau-frère pour avoir été toujours à mes côtés pendant ces 3 années. Enfin, je tiens à remercier ma compagne, Maïté, qui m'a toujours soutenu et m'a épaulé dans les choix, les moments de stress et les épreuves rencontrées.

Jérôme Manceau.

Résumé

Aujourd'hui, de nombreuses applications utilisent des clones 3D de visage. En effet, ils peuvent être utilisés comme prétraitements dans de nombreuses méthodes de synthèse (immersion...) et d'analyse d'un visage (analyse d'émotion...). Cette thèse porte sur le clonage réaliste de visages à partir d'une caméra RVB-Z basse-résolution.

Pour pouvoir être utilisés dans ces applications, les clones doivent modéliser avec précision la forme du visage, tout en conservant les spécificités des individus. Ils doivent aussi être sémantiques, c'est-à-dire que la position des différentes parties du visage (yeux, nez ...) sur le maillage 3D est connue. La texture du visage doit être précise, sans flou et doit contenir un maximum de spécificités de la personne. Nos travaux portent sur une solution permettant d'obtenir un clone 3D sémantique réaliste.

Nous proposons une approche qui utilise des patchs de forme et de texture pour **préserver les caractéristiques du visage de la personne** et un modèle déformable de visage 3D pour obtenir **des clones sémantiques**. Les patchs sont les parties adéquates de données de profondeur et de texture. Ils sont détectés à partir d'une distance d'erreur et de la direction des vecteurs normaux en chaque point du maillage 3D. Ensuite ces patchs, qui mettent l'accent sur les spécificités de l'individu, sont fusionnés pour reconstruire la forme et la texture complète du visage.

Nous avons comparé notre méthode de reconstruction de la forme 3D et de la texture du visage avec les méthodes de l'état de l'art. Ces tests ont montré que notre approche de reconstruction de la forme est plus performante que les méthodes classiques de *fitting*. En effet, elle permet **d'être plus précise** et **de retrouver plus de spécificités des personnes**. Les tests qualitatifs réalisés avec les méthodes de reconstruction de texture de l'état de l'art ont montré que notre méthode de reconstruction de texture est **robuste** et qu'elle permet de **reconstruire une texture précise, sans couture gardant les spécificités des individus**.

Mots clef Clonage de visage, Maillage sémantique, Détection de patchs, Fusion de patchs, Carte de profondeur, *Fitting*, *Warping* de la texture

Table des Matières

Première partie

Introduction

Dans cette thèse nous nous intéressons au clonage de visage. C'est un domaine intéressant et émergent mais il est très difficile, encore aujourd'hui, de reproduire la complexité du visage. Pour que le clone reflète l'identité de la personne, il faut que les traits caractéristiques propres à la personne apparaissent. En effet, l'œil humain a besoin de nombreuses spécificités à la fois de forme et de texture pour reconnaître l'identité d'une personne. Nous verrons dans cette thèse qu'il est compliqué de reconnaitre l'identité d'une personne en observant la forme du visage sans sa texture. La figure I.1.1 montre des exemples de clone 3D sans texture.

Nous avons travaillé sur la reconstruction de la forme 3D et la texture du visage pendant cette thèse. Nous présentons en premier dans cette introduction, le contexte et les motivations. Puis, nous énonçons les contraintes induites et les contributions de nos travaux. Pour conclure cette introduction, nous précisons l'organisation de ce document.

Sommaire

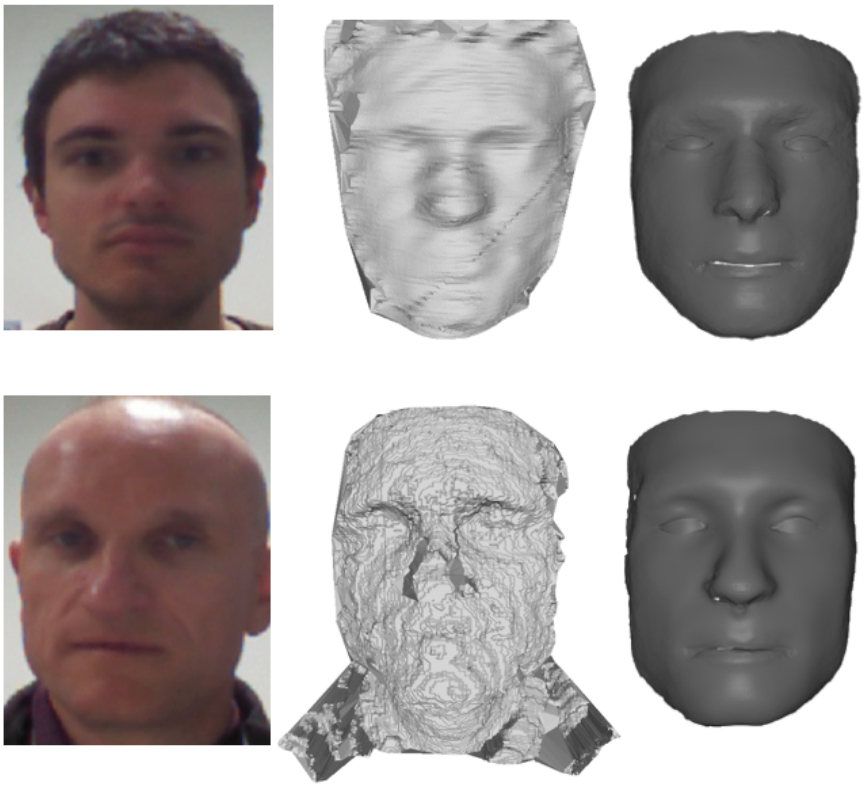


FIGURE I.1.1: La figure présente 2 maillages 3D sans texture. Les maillages 3D du milieu sont des trames de profondeur fournies par la caméra Kinect. Les maillages 3D de droite sont des clones obtenus avec notre méthode de reconstruction de la forme 3D.

I.1.1 Contexte et motivations

Le clonage de visage est un domaine important en infographie et modélisation 3D. Un clone est composé d'une forme et d'une texture. La forme est caractérisée par une surface 2D déployée dans l'espace 3D et représentant l'apparence (nez, menton...) alors que la texture est la couleur qui représente le grain de la peau (grain de beauté, tache de rousseur...). Il existe plusieurs types d'applications qui utilisent ces clones (synthèse et analyse) :

- Les applications dont le but est **d'immerger l'utilisateur dans un monde virtuel**. Aujourd'hui, les jeux vidéo et les films d'animation sont très présents dans notre société. Le but des créateurs de ces applications est d'améliorer le réalisme pour que le joueur soit immergé dans le jeu vidéo. L'utilisation de clones de visage permet de donner l'apparence de l'utilisateur à l'un des personnages, et donc d'augmenter son niveau d'immersion dans l'application. Cela permet au joueur de s'identifier au personnage et d'avoir plus d'interaction avec les autres personnages. Ce type d'application est appelé immersion anthropologique par la socialisation.
- Les applications où **l'utilisateur doit être en mesure d'interagir avec des ordinateurs**. Ce genre d'applications, telles que les serious game, l'e-learning, les simulateurs, la détection des émotions, l'authentification de personne et les nouvelles interfaces de commande,

sont en pleine expansion et sont de plus en plus présentes dans notre quotidien. Beaucoup de recherches sont menées pour améliorer ces applications. R.Gross et al [1], C.Soladie et al [2] et A.Väljamäe et al [3] montrent que les systèmes qui s'adaptent aux spécificités des sujets obtiennent de meilleures performances que les systèmes génériques. Pour cette raison, l'utilisation d'un clone de visage 3D de l'utilisateur plutôt qu'un modèle de visage générique comme pré-traitement augmente les performances de ces applications. K.A.Funes Mora et J.Odobez [4] montrent par exemple que l'utilisation d'un clone 3D pour détecter la pose de la tête et des yeux donne d'excellents résultats.

Pour que les clones de visage puissent être utilisés dans toutes ces applications, ils doivent répondre à un certain nombre de contraintes que nous détaillons dans la section suivante.

I.1.2 Les contraintes et notre proposition

Notre méthode de clonage de visage 3D doit respecter plusieurs contraintes. Nous les avons décomposées en 3 catégories : les contraintes matérielles, les contraintes sur le maillage 3D du clone et les contraintes sur la texture.

Les contraintes matérielles : Pour que les systèmes répondent au cahier des charges des applications décrites dans la section précédente, il faut que le matériel ait un coût peu élevé et qu'il soit facile à utiliser. Un tel système doit être aussi entièrement automatique et non intrusif. Il existe beaucoup de scanners dans la littérature, présentés dans le chapitre II.1, qui permettent d'obtenir des clones de visage 3D de très bonne qualité. Mais ces scanners ont souvent un prix très élevé. De plus, certains nécessitent des étapes de calibration qui ne peuvent pas être effectuées par des personnes n'ayant aucune notion en vision par ordinateur (stéréovision...). Les caméras RVB-Z à faible résolution du type Kinect ont l'avantage de fournir des données de couleur et de profondeur et d'avoir un faible coût. C'est pourquoi elles ont été récemment utilisées dans le domaine du clonage de visage. L'inconvénient de ce type de capteur est la mauvaise qualité de ses données de profondeur et de couleur. L'utilisation d'un modèle déformable de visage 3D permet d'augmenter la résolution et de supprimer le bruit des données. Il existe peu de modèles déformables de ce type dans la littérature. En effet, il faut utiliser un scanner haute résolution qui fournit des scans très réalistes et précis (light stage [5]) pour la construction de la base du modèle déformable de Paysan et al [6].

Les contraintes sur le maillage 3D du clone : Les méthodes de reconstruction de la forme du visage doivent respecter plusieurs contraintes. La forme doit être représentée par un maillage 3D précis, de haute résolution et réaliste. De plus, les clones doivent être sémantiques pour pouvoir ensuite être utilisés dans des applications du type interaction homme-machine. Cela signifie que la correspondance entre chaque point du maillage et le visage doit être connue. Par exemple, nous devons connaître quel point 3D correspond au coin gauche de l'œil. Si cette contrainte est respecté, un maillage sémantique peut directement être utilisé dans des applications comme l'identification d'émotions d'une personne [7]. Cela évite d'effectuer une étape de détection des points caractéristiques qui n'est pas toujours très précise. Pour obtenir un clone sémantique, une étape de *fitting* utilisant un modèle déformable de visage 3D est nécessaire. Ces techniques de *fitting* classique [8] dépendent fortement de la qualité des visages utilisés pour créer le modèle. En effet, les spécificités des individus ne peuvent être trouvées que si elles appartiennent à la base de données ayant servi à la création du modèle. Il est donc essentiel d'utiliser une base de données composée de visages diversifiés. Comme le visage possède énormément de caractéristiques physiques propres à chaque personne, il est très difficile d'obtenir des clones très réalistes de visage par cette méthode.

Les contraintes sur la texture du visage : Pour que le clone soit réaliste, il faut qu'il soit texturé. L'œil humain n'est pas performant pour reconnaître l'identité d'une personne à partir uniquement de la forme de son visage, la texture joue un rôle très important. C'est pourquoi, dans

une application d'immersion dans un jeu vidéo, il faut que le clone ait une texture réaliste pour que le joueur s'identifie à lui. Elle doit être précise, sans flou et elle doit contenir un maximum de spécificités de la personne. Le système doit pouvoir *mapper* une texture sur un maillage sémantique pour qu'il puisse être utilisé facilement dans des applications du type interaction homme-machine telle que la détection de regard.

I.1.3 Contributions

Nous proposons un système entièrement automatique qui permet de reconstruire un visage 3D à partir d'un capteur RVB-Z à faible coût (Kinect). Il permet d'obtenir un clone 3D sémantique de haute résolution. Ce capteur a l'avantage d'être facilement accessible et d'avoir un coût peu élevé. Mais il fournit des données bruitées basse résolution de profondeur et de couleur. Notre système se compose de deux grandes parties : la reconstruction de la forme du visage et la reconstruction de sa texture.

Reconstruction de la forme

Notre méthode de reconstruction de la forme du visage ne contient pas d'étape manuelle et utilise un modèle déformable de visage 3D [6]. L'utilisation d'un modèle déformable présente deux avantages : 1) il permet d'augmenter la résolution et de réduire le bruit de chaque trame de profondeur, 2) il permet aussi de connaître la structure du maillage 3D du visage et donc d'obtenir des maillages 3D sémantiques. Néanmoins, l'utilisation de modèles déformables a aussi des inconvénients. Les attributs morphologiques de certaines personnes que nous voulons cloner peuvent ne pas être reconstruits. Dans un modèle global, certains attributs des visages de la base de données sont corrélés et il est compliqué de trouver toutes les formes et les détails possibles d'un visage inconnu. Ces modèles sont donc limités par leur base donnée d'apprentissage : il faut qu'elle soit la plus variée possible. Notre méthode permet d'obtenir un clone sémantique plus réaliste que les clones obtenus avec les méthodes classiques de *fitting* [8]. Elle met l'accent sur les caractéristiques individuelles de chaque visage.

La principale contribution de notre méthode de reconstruction de la forme 3D de visage est **la conservation des attributs morphologiques locaux** des individus clonés en utilisant un modèle global de visage 3D. La première originalité de cette méthode est au niveau du système. Les méthodes de l'état de l'art, présentées dans la partie II, effectuent d'abord la fusion des données de profondeur puis le *fitting* avec un modèle déformable de visage [6, 8]. La particularité de notre méthode est l'inversion du processus. Nous réalisons d'abord le *fitting* sur chaque trame de profondeur, puis la fusion (voir figure I.1.2). Cela implique de fusionner à posteriori des informations jugées fiables ; le système proposé est alors moins dépendant des alignements et des erreurs de *fitting*. Ainsi, notre méthode reconstruit plus facilement les caractéristiques d'un visage inconnu de la base de données.

La deuxième contribution est notre technique de **détection et de fusion des patchs** qui permet de reconstruire entièrement le visage. Premièrement, comme chaque trame de profondeur ne contient pas tous les détails du visage, notre technique élimine les parties des trames qui ne contiennent pas d'informations pertinentes. Lorsque nous utilisons un modèle déformable de visage, certaines des spécificités morphologiques de l'individu que nous voulons cloner peuvent disparaître. En effet, l'ensemble du modèle ne contient pas toutes les formes et les détails possibles du visage inconnu dans son intégralité. Les spécificités de l'individu ne peuvent être

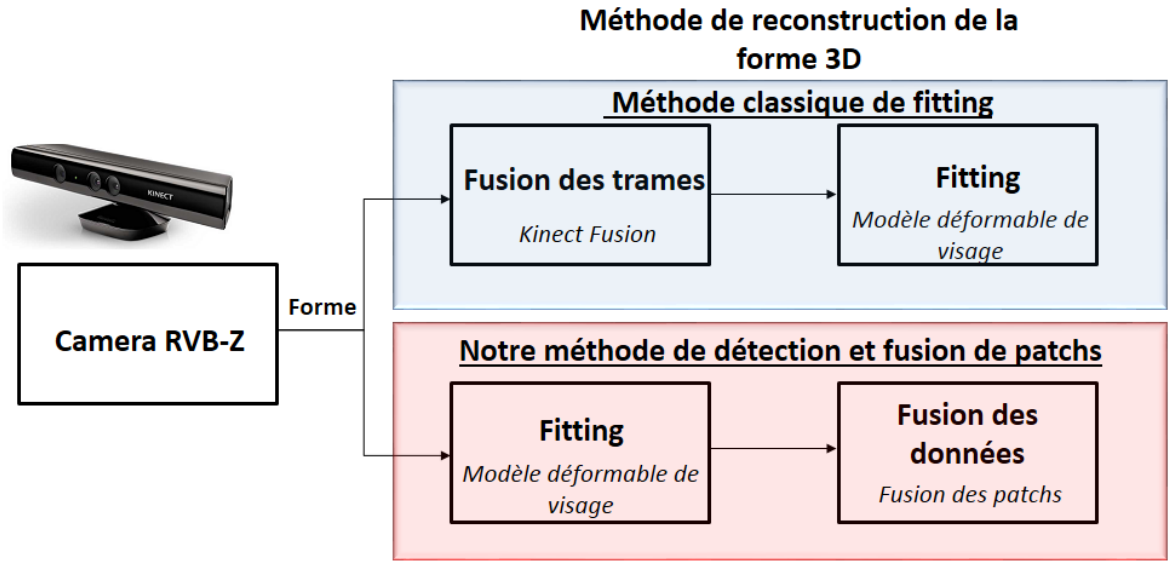


FIGURE I.1.2: Dans notre méthode, nous inversons le système. En effet, nous réalisons l'étape de *fitting* avant la fusion des données.

trouvées que si elles appartiennent à la base de données. Voilà pourquoi nous avons utilisé une méthode qui choisit de petites parcelles (des patchs : ensemble de points soigneusement choisis) qui mettent l'accent sur les détails des spécificités de l'individu. Autrement dit, nous identifions les parties de chaque maillage 3D qui sont pertinentes en utilisant une distance d'erreur et la direction des vecteurs normaux à chaque point de la face. Notre approche permet de trouver les spécificités des personnes qui ne se retrouvent pas avec un procédé classique de *fitting* [8]. De plus, nous fusionnons les données du *fitting* obtenues avec le modèle déformable et les données du capteur de profondeur, ce qui est une nouvelle approche. Ce procédé de fusion augmente le réalisme et l'exactitude du clone 3D.

Reconstruction de la texture

Nous **reconstruisons automatiquement une texture dépliée précise et détaillée avec un capteur RVB-Z à faible coût**. Le principal problème des méthodes de reconstruction de la texture de l'état de l'art [9] est l'apparition de coutures. L'originalité de cette méthode tient dans l'utilisation de patchs de texture pour préserver les caractéristiques de la personne (grain de beauté, barbe ...) et résoudre ainsi le problème des coutures. Nous utilisons plusieurs trames de texture pour récupérer toutes les informations de texture du visage. Chaque trame de texture du capteur RVB-Z contient des détails sur le visage, mais aussi du bruit et des mauvaises informations (contour de la trame...). Nous ne voulons pas fusionner ces mauvaises informations de texture qui peuvent réduire la qualité de la texture finale. Par conséquent, nous **détectons les parties de trames de profondeur**, que nous appelons là aussi patchs, qui sont appropriées et précises. Nous connaissons la correspondance entre la forme et la texture de chaque trame.

Les trames de profondeur sont alors utilisées pour détecter les données de couleur qui ne sont pas correctement saisies par le capteur. En effet, les zones de texture précises sont situées aux endroits où les vecteurs normaux sont parallèles à l'axe optique de la caméra. Pour détecter ces patchs, nous calculons les vecteurs normaux de chaque point 3D et l'erreur entre chaque trame de profondeur et le clone sémantique. La distance d'erreur élimine les erreurs d'alignement rigide et de *fitting*. Les vecteurs normaux sont utilisés pour détecter les endroits où le capteur ne donne pas les informations correctes (trous, bords...). Enfin, nous fusionnons les patchs de texture détectés.
Le deuxième problème des méthodes de reconstruction de texture de la littérature est l'alignement des données de texture des différentes trames. Dans notre méthode, nous **utilisons les données de profondeur et de texture** pour effectuer un alignement efficace.

I.1.4 Organisation de la thèse

La thèse est organisée en 4 parties composées de plusieurs chapitres.

Partie II

Cette première partie présente un état de l'art des méthodes de clone de visage. Nous exposons notre vision des différentes techniques. Nous avons séparé ces méthodes en deux sections :

- **Les scanners 3D** : nous présentons les capteurs passifs et actifs.
- **Les méthodes de reconstruction d'un clone 3D à partir d'un capteur RVB-Z basse résolution**. Nous décrivons en premier les méthodes de reconstruction de la forme 3D du visage, puis les techniques pour la reconstruction de la texture de ce visage.

Partie III

Dans cette deuxième partie, nous décrivons nos travaux de thèse. Elle est composée de 2 sections :

- **La reconstruction de la forme :** Nous décrivons notre méthode de reconstruction de la forme 3D. Elle est basée sur la détection et la fusion de patchs de forme. Nous utilisons aussi un modèle déformable de visage.
- **La reconstruction de la texture :** Comme pour la forme, nous utilisons une technique de détection et de fusion de patchs. Nous utilisons à la fois les données de forme et de texture pour créer la carte de texture du clone.

Partie IV

Dans cette troisième partie, nous présentons les résultats obtenus sur une base de 15 personnes. Nous les comparons qualitativement et quantitativement avec plusieurs méthodes de la littérature. Pour finir, nous discutons des avantages, des inconvénients et des caractéristiques de chaque technique.

Partie V

Dans cette dernière partie, nous réalisons le bilan des travaux effectués pendant cette thèse. Et enfin, nous décrivons les perspectives de ces travaux.

Deuxième partie

Etat de l’art

Cette partie est consacrée à l'état de l'art des techniques de clonage 3D de visage. La figure II.1.1 présente les différentes sections de cette partie du manuscrit. La reconstruction du visage est composée de 2 étapes : l'acquisition des données et les post traitements pour reconstruire le maillage 3D. Dans la première partie, nous discutons des différents scanners 3D génériques qui existent dans le domaine de la vision par ordinateur pour numériser un objet ou une scène (du petit objet au bâtiment). Dans la deuxième partie, nous nous intéressons aux techniques qui permettent de reconstruire la forme et la texture d'un visage 3D à partir des données de profondeur et de couleur d'un capteur RVB-Z. Nous présentons les techniques de reconstruction de la forme qui permettent d'obtenir un maillage sémantique d'un visage 3D, puis celles qui permettent d'obtenir un maillage non sémantique. Pour finir, nous décrivons les méthodes de reconstruction de texture 3D à partir de données RVB obtenues grâce à un capteur 3D. Il existe plusieurs états de l'art de ce type dans la littérature. Daoudi et al [10] ont écrit un livre qui présente les différentes méthodes pour modéliser un visage 3D. K.Ouji [11] présente dans l'état de l'art de son manuscrit de thèse les techniques de numérisation optique 3D. Enfin, B.Loriot [12] a réalisé un état de l'art sur les systèmes d'acquisition 3D.

Chapitre II.1

Scanners 3D haute résolution

Dans cette première partie, nous présentons les différents scanners 3D qui permettent de numériser des objets en 3D. Il existe deux catégories de scanners 3D dans la littérature : les scanners sans et avec contact. Les scanners 3D avec contact sont essentiellement utilisés dans l'industrie. Ils récupèrent l'information de profondeur en palpant l'objet 3D. Ils sont très précis (précision de l'ordre du micron) mais très lents et sont principalement utilisés pour numériser des objets de géométrie simple. Dans ce chapitre, nous nous intéressons aux scanners 3D optiques sans contact. Cette famille de capteur est la plus utilisée dans le domaine de la vision par ordinateur et se base sur l'émission et/ou la réflexion d'ondes. Les capteurs qui émettent des rayonnements sont actifs et ceux qui n'émettent aucun rayonnement sont passifs. Tout d'abord, nous présentons différents scanners passifs (stéréoscopie...) (section II.1.1). Et ensuite les scanners actifs qui existent dans le domaine de la vision (scanners laser, scanners temps de vol et les capteurs 3D à lumière structurée)(section II.1.2).

Sommaire

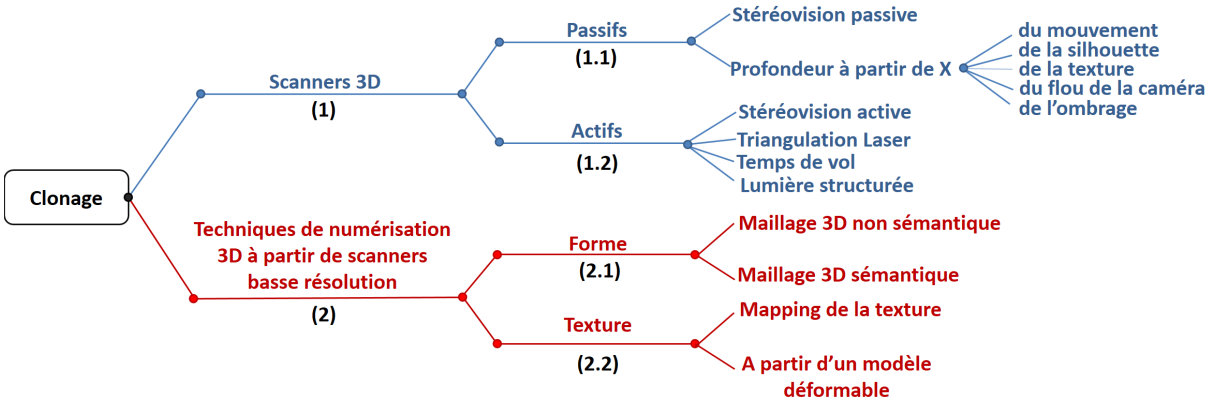


FIGURE II.1.1: Schéma des différentes parties de l'état de l'art.

II.1.1 Capteurs passifs

Les capteurs passifs numérisent en 3D des objets sans émettre de rayonnement. Ils calculent la profondeur à partir du rayonnement de la lumière visible. En effet, la reconstruction de l'objet 3D est réalisée à partir d'un ou plusieurs capteurs RVB. Dans cette partie, nous présentons d'abord la stéréoscopie passive qui permet d'obtenir des maillages 3D à partir de plusieurs capteurs RVB [13], ensuite plusieurs méthodes qui permettent de retrouver la forme d'un objet à partir de certaines caractéristiques d'une ou de plusieurs images RVB [14].

II.1.1.1 Stéréovision passive

Les méthodes de stéréovisions *passives* permettent d'obtenir la forme d'un objet à partir de plusieurs caméras RVB [15] avec des résultats d'une grande précision. Certains paramètres comme l'augmentation de la résolution des caméras permettent d'augmenter la précision des résultats. La diminution de la distance entre les caméras améliore aussi les résultats mais diminue la surface visible de l'objet. C'est pourquoi il faut trouver un compromis en fonction de l'objet et du résultat que l'on veut obtenir. Les méthodes de stéréovisions *passives* sont composées de 3 étapes :

- la calibration des caméras
- l'appariement des points des différentes images
- la reconstruction 3D

Les caméras doivent être disposées autour de l'objet et légèrement espacées (voir figure II.1.2). Comme pour la vision humaine, la distance de chaque point est déterminée en analysant les différences entre les images de chaque capteur. Les disparités entre les images sont analysées et la profondeur est calculée grâce à une méthode de triangulation optique. Avant d'acquérir des images de l'objet, chaque caméra doit être calibrée. Le modèle du sténopé [16] est le plus simple et le plus souvent utilisé. Les paramètres intrinsèques (ex : distance focal) et extrinsèques (ex : matrice de transformation qui permet de passer de l'espace réel à l'espace de la caméra) des

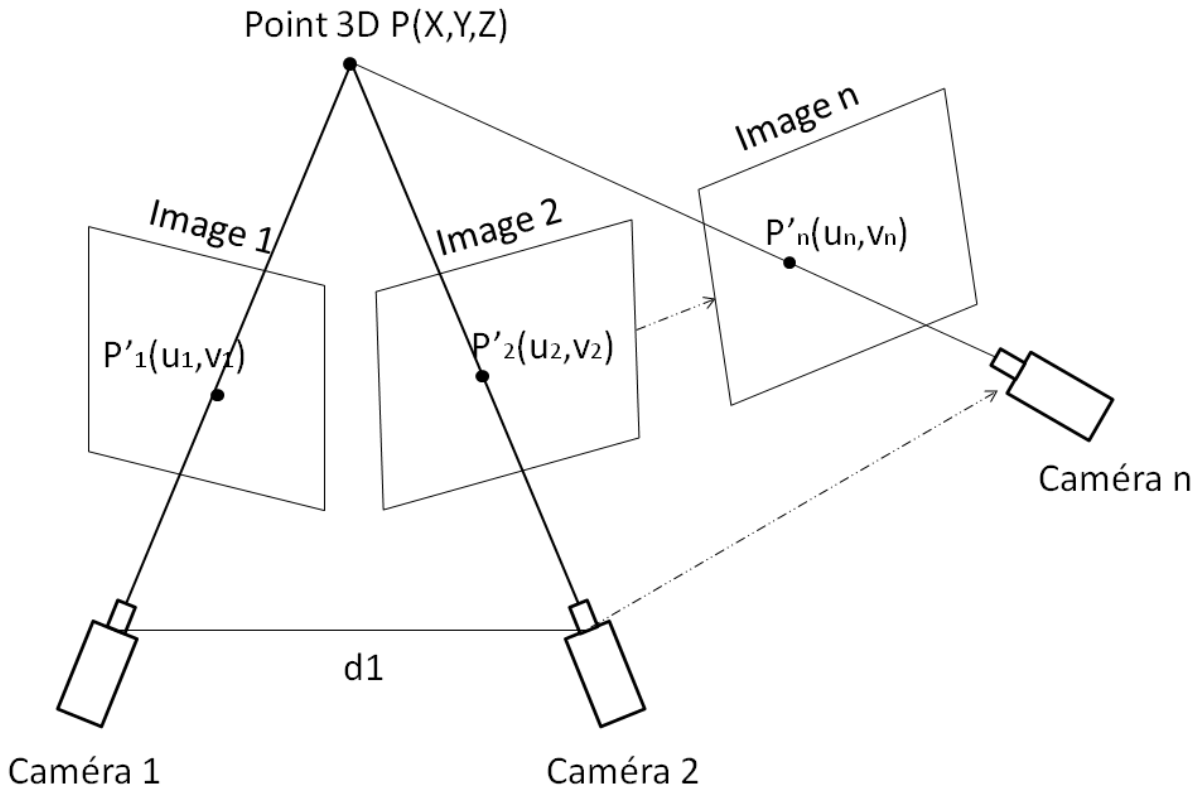


FIGURE II.1.2: Schéma de fonctionnement de la stéréovision. Les caméras (1 à 3) doivent être disposées autour de l'objet (point 3D P). P'1, P'2 et P'3 représentent les images de l'objet P dans les plans images de chaque caméra (Image 1, Image 2 et Image 3). La disparité entre les images est analysée et une triangulation est ensuite réalisée pour calculer la profondeur.

caméras doivent être connus. Une mire de calibration est aussi souvent utilisée pour cette étape [17].

La principale difficulté est la deuxième étape : l'appariement des points des différentes images RVB. En effet, un point d'une image n'a pas forcément de correspondance dans les autres images (occlusion). La réflexion de la lumière qui diffère selon le point de vue rend aussi l'étape d'appariement des points plus difficile. Il y a principalement dans la littérature deux catégories de méthodes : les globales [18] et les locales [19].

Pour réaliser la troisième étape, il existe plusieurs techniques de reconstruction 3D (triangulation...) en fonction du type d'objet à scanner [20]. La plupart des techniques de stéréovision ne sont pas accessibles au grand public. En effet, la calibration des caméras et les post-traitements sur les données capturées ne peuvent pas être effectués par tout le monde. Certains logiciels permettent de reconstruire un maillage 3D à partir de plusieurs photos d'un objet. Agisoft est un logiciel payant qui permet de numériser n'importe quel objet à partir d'image RVB (voir figure II.1.3). AutoDesk 123D Catch [1] est une application gratuite qui permet de reconstruire automatiquement un maillage 3D à partir de plusieurs photos (voir figure II.1.4). Ces deux

1. `http://www.123dapp.com/catch`

FIGURE II.1.3: Résultats obtenus avec le logiciel Agisoft (résolution des images 5184*3456).

FIGURE II.1.4: Exemple d'un scan de visage 3D obtenu avec Autodesk (19 photos de résolution 3264*2448).

techniques sont simples à utiliser mais pour obtenir des résultats précis, ces méthodes nécessitent l'utilisation de capteurs de haute résolution onéreux.

II.1.1.2 Profondeur à partir du mouvement de l'objet

Cette technique permet de reconstruire le maillage 3D d'un objet à partir de son mouvement [21]. Une caméra capture plusieurs images successives de l'objet. Pour estimer la profondeur, il faut évaluer la disparité entre les différentes images. Le mouvement d'un objet devant une

caméra fixe revient à avoir un objet fixe et une caméra en mouvement. Ce problème peut donc être traité comme un problème de stéréovision [22]. L'utilisation de trames successives permet de retrouver plus facilement la correspondance entre deux trames mais conduit à un calcul de la disparité erroné. De plus, il est plus difficile de trouver l'orientation et la position de l'objet car il est en mouvement, ce qui rend la calibration extrinsèque plus difficile à réaliser. Contrairement à une méthode de stéréovision classique, cette méthode est très sensible au bruit et ne permet pas d'obtenir des résultats aussi performants.

II.1.1.3 Profondeur à partir de la silhouette de l'objet

Ce type de scanner 3D calcule la profondeur à partir de plusieurs photos prises de l'objet que l'on veut scanner [23]. La position des appareils photographiques est calculée grâce à une mire de calibration. L'objet doit se trouver devant un arrière-plan contrasté pour que les silhouettes de l'objet sur les différentes photos puissent être segmentées. C'est à partir de ces silhouettes qu'une approximation de la forme de l'objet, appelée *enveloppe visuelle*, est construite. Cette technique permet de reconstruire un modèle 3D texturé mais elle ne permet pas de reproduire les parties concaves des objets. Cette technique fonctionne uniquement pour des objets de formes simples. L'augmentation du nombre de photos et de leur résolution permet d'améliorer la qualité des résultats.

II.1.1.4 Profondeur à partir de la texture d'une image

Cette technique permet de retrouver la forme de l'objet à partir de la texture d'une seule image [24]. Pour que cette méthode fonctionne correctement, il faut que la texture soit particulière. En effet, il faut qu'elle soit composée de motifs homogènes qui se répètent appelés *texels*. La forme est retrouvée à partir de la déformation de ce motif [25]. Cette méthode a l'avantage d'avoir un coût très faible. Mais elle ne fonctionne que dans peu de cas et ne produit pas des modèles 3D de bonne qualité.

II.1.1.5 Profondeur à partir du flou de la caméra

Cette méthode permet de retrouver la forme d'un objet à partir du flou d'une caméra équipée d'une profondeur de champ assez faible (quelques microns) [26, 27]. Cette technique utilise les caractéristiques intrinsèques de l'objectif de la caméra pour calculer la forme de l'objet. En effet, il faut que l'ouverture de l'objectif et la profondeur de champ de la caméra soit connus. L'objet doit être photographié à partir de différents points de vue. C'est à partir des zones qui ne sont pas floues de ces images que la profondeur est calculée. Cette technique est très lente et ne permet de reconstruire que des objets de petite taille. Elle est principalement utilisée pour de la numérisation 2.5D.

II.1.1.6 Profondeur à partir de l'ombrage

La méthode de reconstruction de la profondeur à partir des ombrages d'une image RVB, ou *shape from shading* a été introduite par Horn et al [28] en 1970. Elle utilise les variations d'intensité de l'image pour retrouver la forme d'un objet ou d'une scène. L'ombrage correspond à l'apparence plus ou moins claire d'une surface sur une image. Cette méthode est très dépendante de la position et de l'intensité de l'éclairage. Pour qu'elle fonctionne, il faut que l'illumination soit uniforme et la surface de l'objet lambertienne [28], c'est-à-dire que l'intensité de la lumière réfléchie dépend de son incidence. La qualité de la reconstruction dépend beaucoup des facteurs extérieurs (lumière, réflectance de l'objet...).

II.1.1.7 Conclusion

Les scanners optiques passifs permettent de reconstruire la forme d'un objet à partir de capteurs RVB. La stéréovision passive est la méthode qui donne les meilleurs résultats. En effet, les autres techniques décrites dans cette section ne fonctionnent que sous certaines conditions et donnent des résultats moins performants. Le principal inconvénient de la stéréovision est le coût des capteurs. En effet, pour obtenir des résultats performants les appareils photos ou les caméras doivent être dotés d'un niveau de résolution important. De plus, il faut pour la plupart de ces méthodes effectuer une étape de calibration.

II.1.2 Capteurs actifs

Un scanner actif émet un rayonnement et calcule la distance de l'objet à partir de la réflexion du rayonnement. Nous présentons tout d'abord dans cette partie la stérévision active, puis les scanners à triangulation laser et à temps de vol, enfin les scanners 3D à lumière structurée.

II.1.2.1 Stéréovision active

Il existe plusieurs méthodes de stéréoscopie *active* dans la littérature [29, 11]. Salvi et al [30] projettent sur l'objet des motifs lumineux pour résoudre les problèmes d'appariement des points. Le light stage est une méthode de stéréoscopie active récente proposée par les chercheurs de l'université de Californie du Sud [5]. Il permet de capturer les propriétés lumineuses de n'importe quel objet et de générer un modèle lumineux 3D ultra-haute définition. Il se compose de plusieurs sources de lumière (LED), de plusieurs caméras numériques et d'un système électronique pour réguler la lumière et les caméras. Les sources de lumière permettent de créer une lumière ambiante. Ce qui veut dire que la quantité de lumière réfléchie (réflectance) sera identique dans toutes les directions de l'espace. C'est pourquoi chaque élément de l'objet aura la même apparence quelque soit le point de vue. Il sera donc plus facile de réaliser l'appariement des différents points des images. Cette méthode utilise les propriétés spécifiques de la peau.

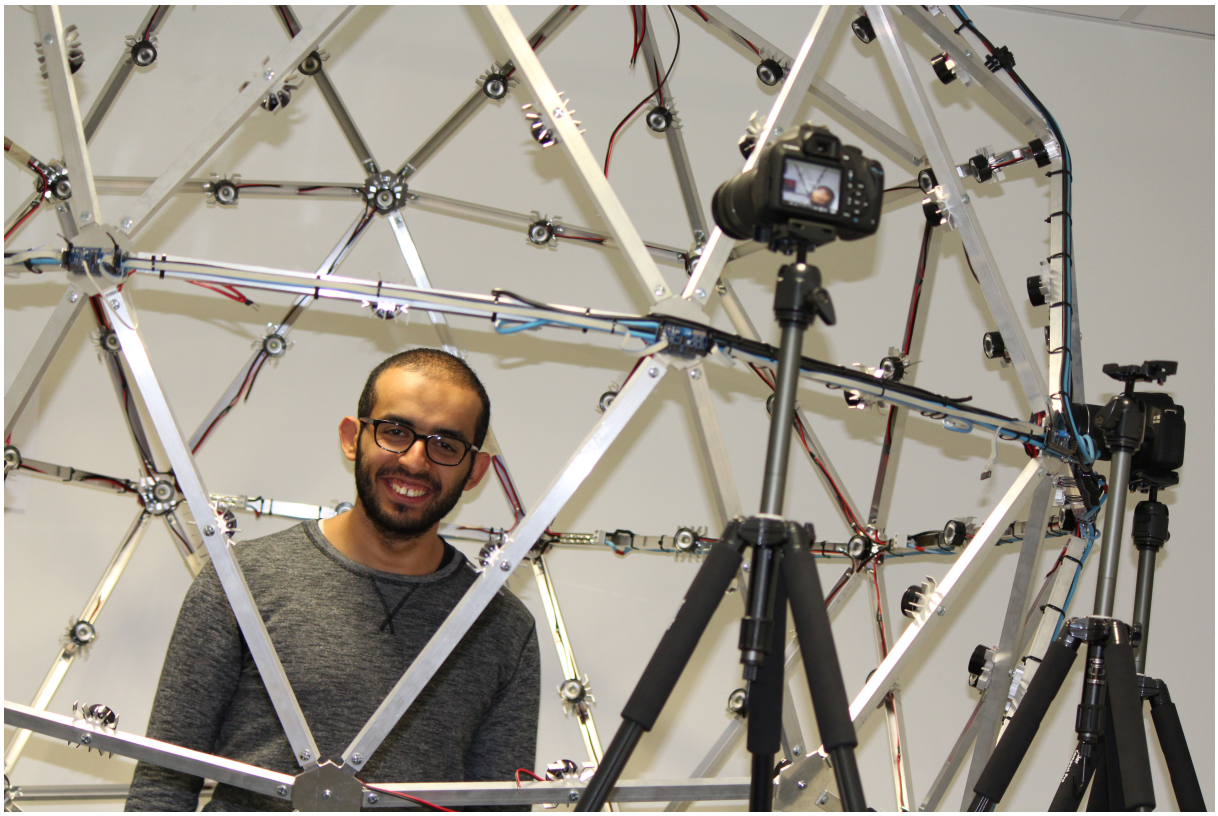


FIGURE II.1.5: Light stage de l'équipe FAST de CentraleSupélec Rennes.

Elle est basée sur le fait que la lumière garde sa polarisation quand elle se déplace sur la peau (lumière spéculaire), alors qu'elle perd sa polarisation quand elle pénètre dans la chair avant de se réfléchir (lumière diffuse). La réflexion est dite spéculaire quand il n'y a qu'un seul rayon réfléchi et diffuse lorsque la lumière est réfléchie dans de nombreuses directions. Il est donc possible d'extraire, pour la synthèse, les cartes de texture diffuse et spéculaire du visage dans toutes les directions de l'espace 3D par filtrage et différence de texture et de spéculaire. En général, une frange de lumière structurée est envoyée sur le visage pour aider la reconstruction. La figure II.1.9 montre un exemple de stéréovision avec des franges de lumière structurée (voir section II.1.2.4). Lorsque les capteurs sont de haute résolution, nous obtenons avec ce type de méthode, un maillage 3D ultra fins du visage (avec les pores de la peau), une texture et une matrice de réflectance. Ce type de méthode peut donc permettre de scanner très précisément n'importe quel type de visage et donc de créer des bases de données de très bonne qualité. A notre connaissance, il n'existe pas encore de base de données publique créée à partir d'un light stage. Les premières versions ont été utilisées dans de nombreux films depuis les années 2000 (Matrix Reloaded...) pour cloner les visages des acteurs. L'inconvénient de cette méthode est son coût et son accès limité au grand public. En effet, il en existe très peu dans le monde. Un Light stage est actuellement en construction dans l'équipe FAST de CentraleSupelec de Rennes (voir figure II.1.5).

FIGURE II.1.6: Exemple de scan 3D obtenu avec le scanner de Faro.

II.1.2.2 Triangulation laser

Les scanners laser permettent de numériser en 3D des objets. Ce sont des capteurs 3D actifs principalement utilisés dans l'industrie (voir figure II.1.6). Il existe deux types de scanner à triangulation laser : le laser à points et le laser à lignes. Le scanner à lignes permet d'acquérir plus de données à partir d'une image. Il est composé d'un projecteur laser et d'un capteur qui permet de connaître la position du laser. La profondeur est calculée à partir d'une triangulation optique [31]. En effet, la caméra, l'émetteur laser et l'objet forment un triangle [24]. Ces scanners permettent d'obtenir des résultats très précis et sont très répandus. Plusieurs systèmes de numérisation laser sont commercialisés (le système VITUS Smart XXL, BodyLine de la compagnie Hamamatsu Photonics...). Les principaux défauts de ces scanners sont leur prix élevé et leur lenteur d'acquisition. En effet, les scanners laser à lignes et les scanner laser à points ne permettent pas d'effectuer des acquisitions en temps réel. De plus, ce type de scanner ne peut pas être utilisé pour numériser un visage car les lasers sont dangereux pour les yeux.

II.1.2.3 Temps de vol

La plupart des scanners temps de vol sont des scanners 3D actifs qui ont une grande portée (300 mètres). Ils sont appropriés pour numériser des objets de taille importante comme des immeubles [32]. Comme les scanners à triangulation laser, ils projettent une onde sur l'objet. Cette onde est réfléchie et captée par le scanner. Le temps nécessaire à l'onde pour faire le trajet aller-retour permet de calculer la distance à laquelle se trouve l'objet. Ces scanners utilisent un télémètre laser pour mesurer le temps d'aller-retour du rayon [33]. La vitesse de la lumière permet de calculer la distance des points 3D. C'est pourquoi la qualité des mesures dépend de la précision de mesure du temps. Ils ont l'avantage d'être très rapides mais ils ne permettent pas d'obtenir des résultats très précis. En effet, la grande célérité de la lumière empêche d'effectuer

des mesures précises en deçà de plusieurs millimètres. Plusieurs scanners Temps de vol ont été commercialisés (Lidar...). Ces scanners nécessitent souvent un post-traitement (filtrage...) et peuvent scanner de 10 000 à 100 000 points par seconde. Récemment, des capteurs temps de vol avec une courte portée ont été commercialisés (Kinect version 2, softkinectic...). Cui et al [34] utilisent une méthode de super-résolution pour débruiter les données fournies par ce genre de capteur (caméra TOF Swissranger SR4000). Ils présentent l'avantage d'avoir un faible coût et de pouvoir être utilisés facilement par le grand public, mais ils fournissent des données de moins bonne qualité.

Il existe aussi des scanners laser qui évaluent la distance en mesurant les décalages de phase [35]. Le scanner émet aussi un rayon qui est réfléchi au contact de l'objet. Le scanner n'envoie plus une impulsion mais une onde modulée en amplitude. La distance est calculée en analysant le décalage de phase entre le rayon émis et le rayon reçu. Ces scanners sont plus précis que les scanners temps de vol classiques mais ils ont une portée plus petite.

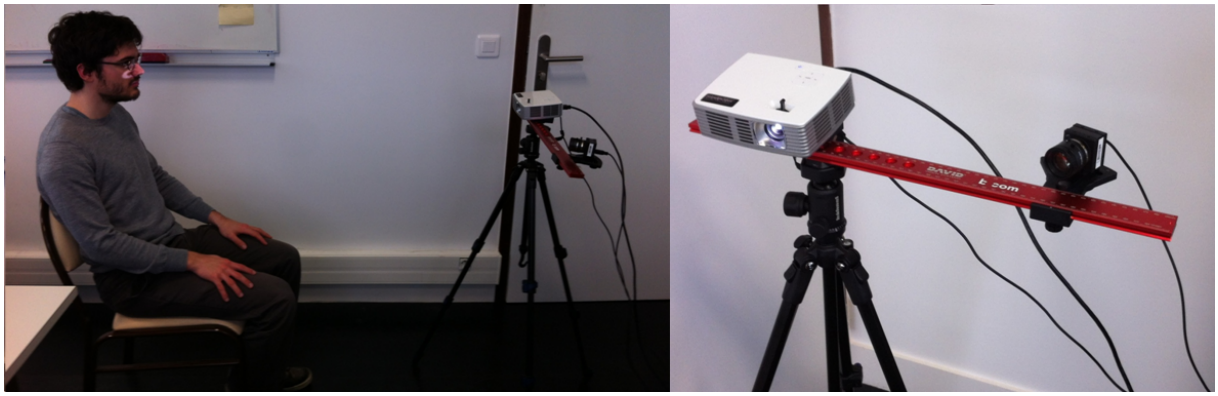


FIGURE II.1.7: Scanner David Pro Edition SLS-2.

Nous trouvons aussi dans la littérature les systèmes Conoscopique [36]. Comme pour les deux précédentes méthodes, un rayon laser est envoyé vers l'objet. Ensuite, le rayon réfléchi passe à travers un cristal biréfringent [37] avant d'être envoyé sur un capteur. La distance de l'objet est calculée en analysant la fréquence des motifs de diffraction du rayon. Ce type de système est compliqué à mettre en place et a un coût élevé.

II.1.2.4 Lumière structurée

Les techniques de numérisation qui utilisent des lumières structurées permettent de scanner des objets en 3D. De nombreuses recherches sont réalisées pour améliorer la précision des résultats obtenus avec ce type de scanner. Les scanners 3D de cette catégorie projettent un motif lumineux sur l'objet et observent ensuite les déformations subies par le motif. C'est pourquoi ils sont composés d'un projecteur et d'une ou plusieurs caméras qui enregistrent les déformations du motif. Premièrement, la distance est calculée à partir des déformations du motif projeté. Puis, une technique de triangulation est utilisée pour calculer la position dans l'espace des différents

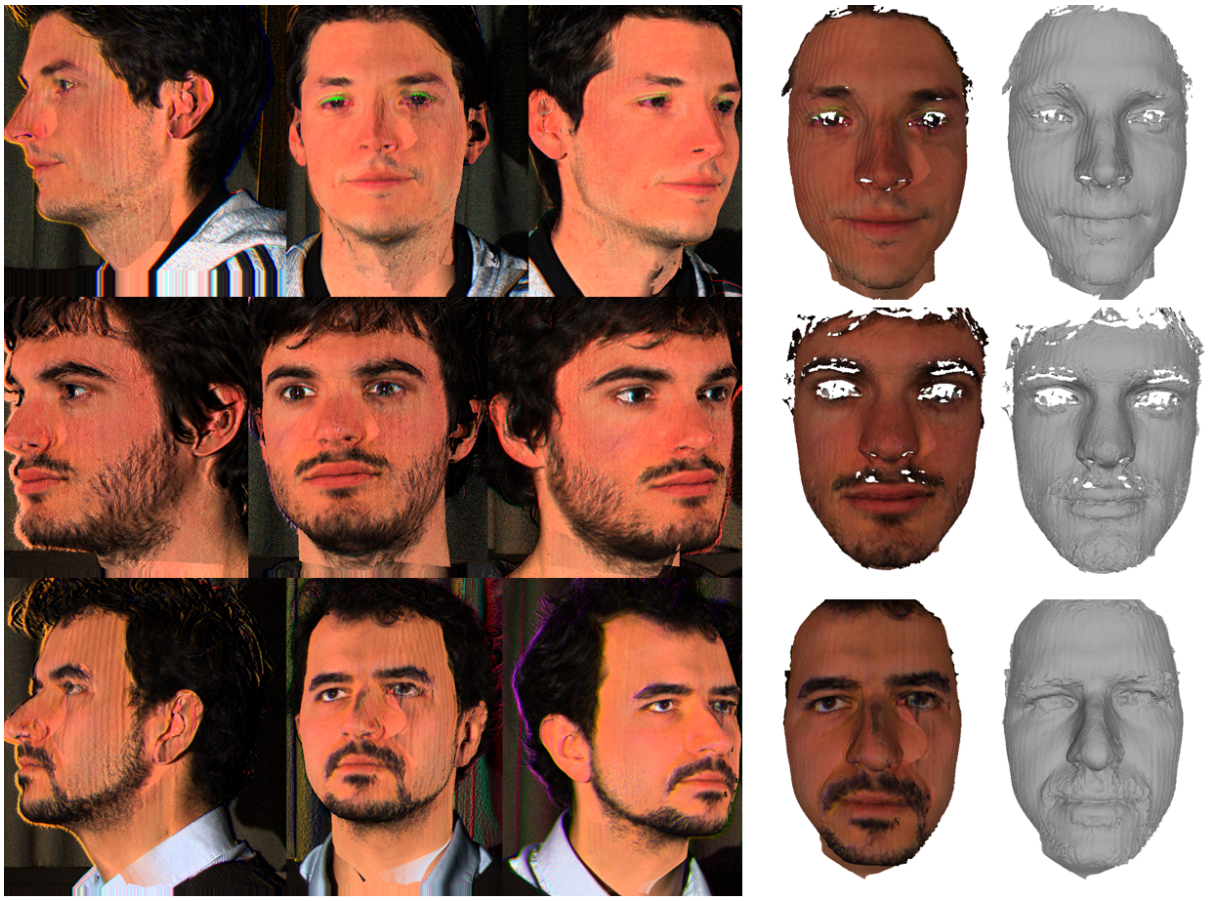


FIGURE II.1.8: Exemples de scans de visage 3D obtenus avec le scanner David SLS-2 (3 trames par visage). Les clones obtenus sont de bonne qualité mais des franges verticales apparaissent sur le maillage.

points du motif. Inspeck Mega Capturor II 3 qui acquiert des données de profondeur grâce à la lumière structurée a permis de créer la base de données Bosphorus avec une précision de 0,3 mm ref [38]. Paysan et al [6] utilise un système *coded light* créé par ABW-3D. Il reconstruit la forme d'un objet en utilisant une séquence de motifs lumineux. Ce scanner permet d'obtenir des clones réalistes avec une résolution élevée. Il a été utilisé pour concevoir la base de données du modèle déformable de visage de Basel Face Model [6]. Pan et al [39] décrivent une technique pour mesurer la profondeur à partir d'un motif structuré de couleur et d'une caméra. La texture de l'objet peut aussi être acquise en calculant les différences de teintes entre plusieurs couleurs projetées sur l'objet [40]. Le scanner SLS-2 de David Vision Systems est un scanner 3D à lumière structurée (voir figure II.1.7). Il permet d'obtenir des résultats précis. Mais des franges peuvent apparaître sur le maillage 3D reconstruit (voir figure II.1.8). De plus, il a un coût assez élevé. Le Scanner HDI Advances Scanner est aussi un scanner de ce type qui utilise deux caméras RVB pour reconstruire le maillage 3D. Il est très performant et nous a permis d'obtenir des vérités terrains pour notre système de clonage bas coût (voir figure II.1.9). La caméra Kinect de Microsoft est un capteur à lumière structurée infrarouge qui fournit des données de profondeur et de couleur de base résolution. Elle projette un mouchetis infrarouge sur l'objet. Elle a l'avantage

d'être rapide, mais aussi d'être accessible au grand public et d'avoir un faible coût. Mais les données qu'elle fournit sont de basse résolution et très bruitées.

FIGURE II.1.9: Exemple de scan de visage 3D obtenu avec le scanner HDI Advances de LMI technologies.

II.1.2.5 Conclusion

Les scanners 3D actifs sont des scanners très performants. Ils permettent d'obtenir des résultats très réalistes et précis. Le principal inconvénient est le coût souvent très élevé de ces scanners. C'est pourquoi ils sont principalement utilisés dans l'industrie ou pour construire des bases de données. Récemment, des capteurs actifs à bas coût ont été commercialisés (Kinect, capteur temps de vol basse résolution...). Mais ils fournissent des données de basse résolution et très bruitées. De nombreuses recherches ont été effectuée pour améliorer les résultats obtenus avec ce type de capteur.

II.1.3 Conclusion

Dans notre méthode de clonage, nous avons choisi d'utiliser un capteur de type lumière structurée basse résolution. Nous utilisons une caméra Kinect version 1 qui est équipée d'un capteur de couleur et d'un capteur de profondeur. Elle a été développée par la société israélienne PrimeSense. Elle offre une résolution de 640*480 à 30 trames par seconde et une portée assez courte (0,5 mètre à 3.5 mètres). Elle ne fonctionne pas bien en présence de la lumière du soleil. En effet, la mire infrarouge envoyée par la Kinect est fortement perturbée par les infrarouges du soleil. C'est le capteur qui correspond le mieux à nos contraintes. En effet, il a un prix abordable et est accessible au grand public. Il peut être utilisé sur des objets de la taille d'un visage humain. De plus, il permet d'acquérir la texture et la forme de l'objet très rapidement. Il est utilisable seul et est automatique. Ce capteur possède néanmoins certains inconvénients. En effet, les données de profondeur et de texture sont très bruitées et sont de basse résolution. De plus, le capteur ne numérise pas bien les surfaces réfléchissantes de types miroir (la pupille). C'est pourquoi il ne donne pas la bonne information au niveau des yeux. Dans notre méthode, nous avons besoin d'un

capteur qui fournit une carte de profondeur qui soit peu cher et facile à utiliser. Certains capteurs temps de vol répondent aussi au cahier des charges et peuvent être utilisés dans notre méthode (caméra TOF Swissranger SR4000, softkinetic, Kinect version 2). Dans la partie suivante, nous présentons les méthodes de la littérature qui permettent d'améliorer les résultats obtenus avec les données basse résolution de ces capteurs.

Chapitre II.2

Scanners 3D basse résolution

Dans cette deuxième partie de l'état de l'art, nous décrivons les différentes méthodes de reconstruction d'un visage 3D à partir des données fournies par un scanner 3D basse résolution. Il existe plusieurs techniques de clonage de visage 3D qui utilisent des capteurs RVB-Z basse résolution (Kinect : figure II.2.1,Time-off Flight). Au cours de la dernière décennie, ces capteurs ont souvent été utilisés dans les laboratoires de recherche notamment pour leur faible coût.
Chaque visage est composé d'une forme et d'une texture. C'est pourquoi dans la littérature, la plupart des méthodes de reconstruction de visage 3D présentent séparément la reconstruction de la forme et la reconstruction de la texture. Cette section est composé de deux grandes parties (voir figure II.1.1). Tout d'abord, nous présentons les méthodes de reconstruction 3D de la forme (voir section II.2.1), puis les méthodes de reconstruction de la texture (voir section II.2.2).

Sommaire

FIGURE II.2.1: Caméra Kinect : capteur utilisé dans notre méthode de clonage de visage 3D. La photo au milieu représente une carte de profondeur de résolution 640*480. La photo à droite de la figure montre l'image RVB de résolution 640*480 fournie par la caméra Kinect

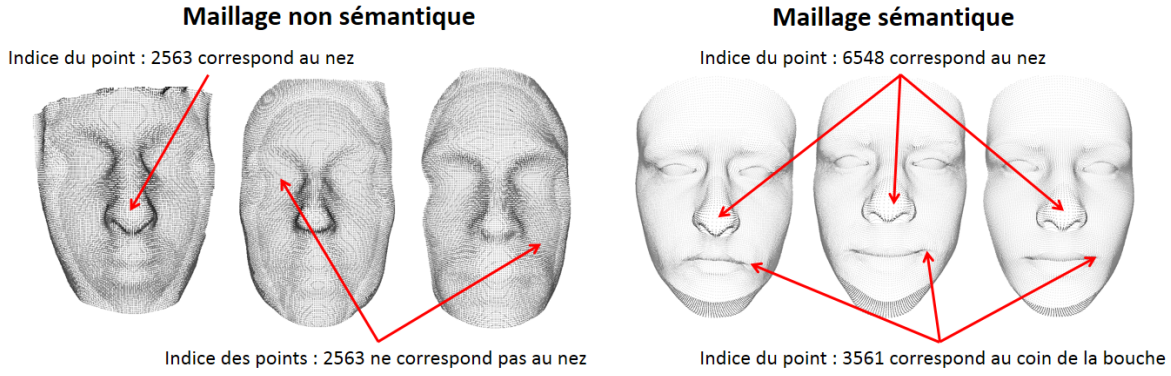


FIGURE II.2.2: Cette figure explique la différence entre un maillage sémantique et un maillage non sémantique. Dans des maillages non sémantiques (les trois maillages à gauche), le point correspondant à l'indice 2563 ne correspond pas à la même partie du visage. Au contraire, pour les trois maillages sémantiques (à droite), le point d'indice 6548 correspond au bout du nez dans les trois maillages.

II.2.1 Reconstruction de la forme 3D

Dans cette section, nous présentons les méthodes de reconstruction de la forme 3D à partir d'un scanner basse résolution (Kinect : figure II.2.1, temps de vol). Ces méthodes peuvent être classées en deux sous-sections : les techniques qui permettent d'obtenir un maillage 3D non sémantique et celles qui permettent d'avoir un maillage 3D sémantique. Un maillage sémantique est un maillage dont nous connaissons la correspondance de chacun des points 3D avec les zones du visage que nous voulons cloner. La figure II.2.2) montre la différence entre trois maillages non sémantiques (à gauche) et trois maillages sémantiques (à droite). Dans les maillages sémantiques, l'indice du point 3D du coin gauche de la bouche (indice 3561 dans la figure II.2.2 est identique pour les trois maillages. De plus, les maillages sémantiques possèdent un nombre de points identiques. Contrairement aux maillage non sémantiques qui ont un nombre de points quelconque.

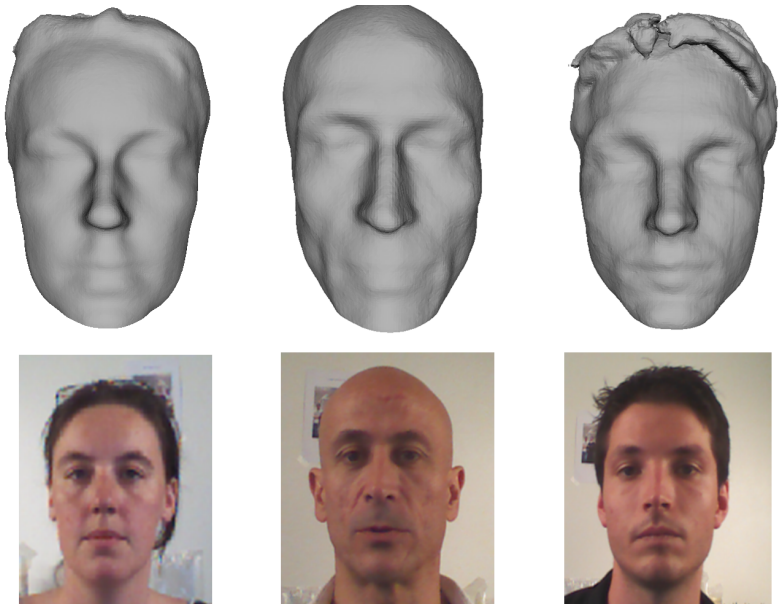


FIGURE II.2.3: Exemples de reconstruction de visages 3D à partir de l'algorithme Kinect Fusion [42].

II.2.1.1 Maillage 3D non sémantique

Il existe de nombreuses méthodes dans la littérature qui permettent de reconstruire la forme 3D d'un visage à partir de données de couleur et de profondeur basse résolution. Les techniques présentées dans cette partie n'utilisent pas de modèle déformable de visage 3D et conduisent, de ce fait, à la production d'une forme 3D non sémantique. Microsoft a présenté en 2012 un système de reconstruction 3D en temps réel de scènes ou d'objets à l'aide d'une caméra Kinect appelé Kinect Fusion [41]. Il permet de d'obtenir des maillages 3D de bonne qualité. Il est disponible gratuitement dans le SDK fournit sur le site internet de Microsoft (voir figure II.2.3). L'algorithme est composé de trois étapes :

- le suivi de la caméra
- l'intégration volumétrique
- le *Raycasting*

Cui et al [34] proposent un algorithme qui permet de scanner des objets en 3D à partir d'une caméra temps de vol basse résolution (MESA Swissranger SR4000 ToF). Cette caméra envoie une lumière infrarouge qui est ensuite réfléchie par l'objet pour calculer la profondeur [43]. Comme la Kinect, cette caméra a un coût peu élevé et est accessible au grand public mais fournit des données de profondeur bruitées et de basse résolution. Les auteurs démontrent dans leur article que leur méthode fonctionne aussi avec une caméra Kinect et un système de stéréovision actif [34]. Elle est aussi composée de trois étapes : la super résolution, l'alignement des scans et la reconstruction du maillage 3D. Sun et al [44] proposent une méthode pour reconstruire un visage 3D à partir de données RVB et de profondeur captées avec une Kinect basse résolution. Elle est composée de trois étapes :

– la segmentation du visage
– l'alignement et l'augmentation de la résolution des trames de profondeur
– la combinaison et le lissage de ces trames de profondeur haute résolution

Hernandez et al [45] décrivent une méthode de reconstruction d'un visage 3D qui utilise une caméra de profondeur à bas coût. Elle est composée de trois étapes principales :

– l'alignement
– la création d'une carte 2D canonique
– le filtrage des données

La plupart de ces techniques sont structurées de la même manière. Premièrement dans ces méthodes, les auteurs détectent et segmentent le visage. Ensuite, ils filtrent et augmentent la résolution des trames de profondeur pour améliorer leur qualité. Puis ils utilisent un algorithme pour aligner les données de profondeur des différentes vues du visage. Pour finir, ils reconstruisent un maillage 3D à partir des données de profondeur alignées.

Après avoir récupéré les données de profondeur et de couleur fournies par le capteur RVB-Z, certaines méthodes dans la littérature détectent et segmentent le visage. En effet, les données sont constituées du visage mais aussi d'autres informations qui ne sont pas utiles pour reconstruire la forme 3D (ex : les oreilles). Sun et al [44] acquièrent quatre trames de profondeur de face du visage pour reconstruire le maillage 3D. Pour détecter et segmenter le visage de la personne, ils utilisent les données RVB fournies par la caméra Kinect. Premièrement, ils détectent la personne en utilisant le SDK de Microsoft et identifient la tête en émettant l'hypothèse qu'elle se trouve dans une zone rectangulaire entre l'épaule droite et l'épaule gauche.

Les données de profondeur fournies par les capteurs RVB-Z sont de basse résolution et souvent très bruitées. C'est pourquoi, certaines méthodes de reconstruction de la forme 3D d'un visage utilisent des techniques pour filtrer et augmenter la résolution des données. Cui et al [34] utilisent une méthode de super-résolution [46] pour augmenter la résolution des cartes de profondeur. Cette méthode permet de créer une carte de profondeur haute résolution à partir de plusieurs cartes de profondeur basse résolution. Pour que cela fonctionne, il faut que les différentes cartes de profondeur basse résolution aient un point de vue très légèrement différent de l'objet statique. Cette approche permet d'obtenir des trames de profondeur moins bruitées et lissées. Sun et al [44] utilisent une interpolation bilinéaire [47] pour augmenter la résolution de ces trames de profondeur. Chaque trame de profondeur basse résolution (128*128) devient une trame de profondeur haute résolution (512*512). Pour finir, ils combinent les quatre trames de profondeur pour augmenter le réalisme du résultat en minimisant une fonction d'énergie. Ils utilisent une variante de la fonction d'énergie utilisée dans l'algorithme de super-résolution Lidarboost [46]. En effet, il utilise un algorithme Laplacien [48] pour lisser le résultat et remplacer le terme de régularisation du Lidarboost.

Hernandez et al [45] utilisent un modèle cylindrique comme certaines méthodes de la littérature [49, 50] pour traiter plus facilement les données de profondeur. Une carte de profondeur dépliée est obtenue en projetant l'information 3D sur un cylindre placé autour du visage. Ce procédé

permet de transformer les données 3D en données 2D (carte de profondeur) et a l'avantage de limiter le nombre de données et de permettre l'utilisation de traitements 2D. Pour finir, ils filtrent la carte de profondeur. Pour cela, ils utilisent un filtre moyenneur, un filtre bilatéral et une interpolation linéaire pour éliminer le bruit et lisser les données.

Chaque trame de profondeur obtenue grâce à un capteur RVB-Z ne fournit pas l'information globale du visage. En effet, une trame de face ne contient pas d'informations sur le profil de la personne (zone cachée). C'est pourquoi de nombreuses méthodes de reconstruction de la forme 3D utilisent des trames de différentes vues du visage. Avant de pouvoir reconstruire le maillage 3D, il faut aligner ces différentes trames de profondeur. Newcombe et al [41] utilisent d'abord l'algorithme Iterative Closest Point (ICP) [51] pour déterminer la position de la caméra et aligner les différentes trames de profondeur fournies par la Kinect. Cet algorithme est présenté dans la partie IV de ce manuscrit. Kinect Fusion se sert de chaque point des trames et de leurs plans tangents pour aligner les données de profondeur. Sun et al [44] utilisent dans la deuxième étape de leur méthode de reconstruction de la forme, l'algorithme *Iterative Closest Point* [52] pour aligner quatre trames de profondeur fournies par une caméra Kinect. Hernandez et al [45] se servent aussi d'une variante de cet algorithme, EM-ICP implémenté sur CUDA [53], pour estimer la pose de la tête. Cui et al [34] alignent chaque trame de haute résolution pour reconstruire le visage 3D en utilisant un procédé d'alignement probabiliste [54]. Ce procédé est adapté aux données fournies par le capteur temps de vol.

Après avoir traité et aligné les données, il faut reconstruire le maillage 3D du visage pour obtenir une représentation 3D de la forme. Dans la dernière étape de leur méthode, Hernandez et al [45] créent le maillage 3D en utilisant une méthode des plus proches pixels voisins sur la carte de profondeur. Newcombe et al [41] se servent d'une représentation de surface volumique [55] pour créer une représentation 3D des différentes trames de profondeur alignées. Les différents sommets sont intégrés dans une grille 3D de voxels à résolution fixe en utilisant une variante de *Signed Distance Functions (SDFs)* [56]. A partir de la pose globale calculée, les points alignés sont convertis en coordonnées globales et la grille de voxels mise à jour. Pour finir ils réalisent un *Raycasting* [57] pour le rendu de données de profondeur. Ils obtiennent une carte de profondeur synthétique qui est moins bruitée et qui est utilisée dans l'itération suivante de l'algorithme ICP. A la fin de leur algorithme, Cui et al [34] obtiennent un maillage 3D à partir d'une reconstruction de Poisson [58].

Conclusion

Toutes ces méthodes permettent de reconstruire des visages 3D à partir d'une caméra répondant à notre cahier des charges. La plupart de ces méthodes utilisent des algorithmes pour augmenter la résolution et filtrer les données de profondeur et de RVB. Compte tenu du bruit que fournit la caméra Kinect, la technique de Newcombe et al [42] permet d'obtenir des résultats performants. En effet, elle fournit des maillages 3D avec une précision au millimètre et sans bruit. C'est pourquoi, elle est très utilisée pour scanner des objets de taille moyenne. Mais cette

méthode est moins performante pour cloner des visages. En effet, certains détails caractéristiques d'un visage humain n'apparaissent pas. Par exemple, nous pouvons observer sur la figure II.2.3 que les yeux et les lèvres ne sont pas marqués et ne correspondent pas à la réalité. En effet, les données de profondeur fournies par la Kinect sont de basse résolution et bruitées et ne donnent pas d'informations très détaillées sur les parties du visage. En plus, en raison de la réflexion des rayons infrarouges dans les yeux, le capteur ne renvoie pas la forme du globe oculaire. La méthode de Cui et al [34] permet d'obtenir de bons résultats. Mais, comme Kinect Fusion, elle n'est pas spécifiquement adaptée au clonage de visage. C'est pourquoi elle ne fournit pas non plus toutes les caractéristiques du visage de la personne que l'on veut cloner. Contrairement aux deux méthodes présentées précédemment, la méthode de Sun et al [44] est spécifiquement utilisée pour cloner des visages humains. Les méthodes de super-résolution permettent d'augmenter la qualité des données fournies par la caméra, mais elles ne permettent pas d'obtenir toutes les caractéristiques humaines d'un visage. En effet, certains traits du visage comme les yeux ne sont pas marqués. La méthode de Hernandez et al [45] permet d'obtenir des résultats sans bruit et de haute résolution mais qui ne sont pas marqués au niveau des yeux et de la bouche. Ces méthodes sont donc limitées en performance à cause de la mauvaise qualité des données fournies par les caméras. De plus, elles ne fournissent pas de clones 3D sémantiques et ne peuvent donc pas être directement utilisées comme pré traitement dans des applications nécessitant des connaissances sur la correspondance des points du maillage avec le visage. Pour résoudre ces deux importants inconvénients, certaines méthodes utilisent des modèles déformables de visage 3D. Dans la partie suivante nous décrivons les différentes méthodes de ce type qui existent dans la littérature.

II.2.1.2 Maillage 3D sémantique

Il existe dans la littérature de nombreuses méthodes qui utilisent des modèles déformables de visage 3D pour réaliser un clone de visage. Ils permettent de reconstruire des clones sémantiques et d'éliminer le bruit des données de profondeur fournies par les capteurs RVB-Z. De plus, l'utilisation d'un modèle déformable de visage 3D permet d'obtenir un maillage sémantique haute résolution qui possède les caractéristiques physiques d'un visage humain. Tout d'abord dans cette partie, nous présentons les différents modèles déformables de visages. Ensuite, nous décrivons les méthodes de *fitting* à partir de données 2D. Et pour finir, nous détaillons les techniques de la littérature qui adaptent un modèle déformable sur des données de profondeur fournies par un capteur de basse résolution.

Plusieurs équipes scientifiques ont créé des modèles déformables de visage 3D. La création de ces modèles nécessite une base de données de visages importante. De plus, les visages doivent êtres numérisés avec un scanner qui permet d'obtenir des résultats très réalistes et de haute résolution (type Light Stage). C'est pourquoi nous ne trouvons pas de énormément de modèles déformables dans la littérature. Blanz et Vetter [59] présentent le modèle déformable de visage dont ils se servent dans leur algorithme de clonage de visages 3D. Ils ont été les premiers à

utiliser un modèle déformable pour faire de la synthèse de visage. Pour créer leur modèle, ils ont scanné 200 visages neutres (100 masculins, 100 féminins) avec un scanner laser (Cyberware TM). Paysan et al [6] présentent une méthode de reconnaissance faciale à partir de visages 3D. Ils ont créé un modèle déformable (Basel Face Model) de visage 3D à partir de 200 scans de visage (100 femmes, 100 hommes). Ils ont utilisés le scanner à lumière structurée construit par ABW-3D. Ce modèle déformable est très utilisé dans le domaine. En effet, il est gratuit et facilement accessible sur internet.

Les modèles déformables de visage ont d'abord été utilisés pour reconstruire des visages 3D à partir d'images 2D. Il existe plusieurs méthodes de ce type dans la littérature qui permettent de reconstruire la forme et la texture. Dans cette section, nous présentons la reconstruction de la forme du visage. Dans leur article, Blanz et Vetter [59] ont testé leur modèle déformable en l'utilisant dans une méthode de reconnaissance faciale. Leur méthode permet de reconstruire le visage 3D à partir d'une ou plusieurs photos prises dans des conditions différentes et d'un modèle déformable de visage. Premièrement, un alignement manuel du visage moyen du modèle 3D est effectué. Ensuite, une procédure de correspondance automatique adapte le modèle déformable 3D sur l'image. Dans cette étape automatique, une image I_{model} est estimée à partir des visages 3D fournis par le modèle déformable. Une projection en perspective du modèle déformable et un modèle d'illumination de Phong [60] sont utilisés pour effectuer cette estimation. Ensuite les coefficients du modèle sont optimisés pour que l'image estimée à partir du modèle déformable I_{model} soit la plus proche de l'image d'entrée en terme de couleur de pixels. Ce type de méthode peut être aussi utilisé pour faire de la reconnaissance faciale. Comme dans la méthode de Blanz et Vetter [59], Paysan et al [6] adaptent leur modèle déformable sur les images de deux bases de test (base de donnée CMU-PIE [61] et FERET [62]). Pour réaliser le *fitting*, ils optimisent trois termes d'erreurs basés sur les points caractéristiques, le contour et l'ombrage. Pour augmenter la flexibilité de l'algorithme, ils adaptent d'abord quatre régions (le nez, les yeux, la bouche et le reste du visage) séparément avant de les combiner. Ces deux méthodes donnent des résultats cohérents et satisfaisants. En effet, elles permettent d'obtenir des visages 3D sémantiques de haute résolution. De plus comme elles utilisent un modèle déformable de visage, les clones possèdent les caractéristiques physiques d'un visage humain notamment au niveau des yeux et de la bouche. Mais la qualité des résultats dépend beaucoup de la diversité de la base de test qui a permis de créer le modèle déformable. Les résultats pour un visage de type africain seront moins performants si le modèle déformable de visage a été créé à partir de visages de type caucasien. De plus, la méthode de Blanz et Vetter [59] n'est pas complètement automatique. Il faut effectuer un alignement manuel.

Les techniques les plus récentes adaptent généralement le modèle déformable sur des données 3D. En effet, les données de profondeur possèdent plus d'informations sur les régions du visage qui disposent de nombreux détails (côtés du nez, les joues...) que des images RVB. Zollhofer et al [63] présentent un algorithme pour cloner des visages 3D de haute résolution à partir de données RVB et de profondeur obtenues avec une caméra Kinect. Cet algorithme est composé

de 3 grandes étapes : le filtrage des données de profondeur, la segmentation du visage et le *fitting*. Cao et al [8] présentent la méthode qui leur à permis de créer une base de données de visages 3D de 150 individus. Pour chaque personne, ils ont capturé avec une caméra Kinect les données RVB-Z de vingt expressions différentes dont une neutre. Zollhofer et al [64] présentent une méthode itérative pour cloner le visage 3D d'une personne. Leur méthode est composée de trois étapes : l'estimation de la pose tête, la fusion des données et le *fitting*. Les maillages 3D sémantiques peuvent ensuite être utilisés dans diverses applications telles que la détection d'émotions [7], l'animation de visage [65] ou encore le vieillissement [66]. En effet, Odobez et al [4] utilisent des clones 3D pour estimer la pose de la tête et la détection du regard d'une personne. La première étape de sa méthode pour estimer la direction du regard est la création d'un clone spécifique au visage de la personne. C'est pourquoi il utilise le modèle déformable de visage de Basel [6]. Ce type de méthode est souvent composé de deux étapes : le filtrage et la fusion des données de profondeur et le *fitting* du modèle déformable de visage sur les données. Toutes ces méthodes sont structurées de la même manière. Premièrement, les données du capteur sont traitées pour éliminer le bruit et augmenter la résolution. Si plusieurs trames du visage sont utilisées, alors ils emploient une méthode pour les fusionner et obtenir un maillage 3D. Pour finir, ils adaptent un modèle déformable de visage sur les données traitées précédemment.

Avant d'adapter le modèle déformable de visage sur les données de profondeur fournies par le capteur RVB-Z, les données sont traitées avec différents algorithmes. Zollhofer et al [63] effectuent plusieurs traitements pour éliminer le bruit et lisser les trames de profondeur. Premièrement, ils moyennent les données de 8 trames de face successives pour obtenir un maillage 3D plus lisse. L'utilisateur doit rester immobile pendant l'acquisition pour éviter un effet de flou. Ensuite, ils appliquent un filtre de Gauss et un algorithme de remplissage de trous pour améliorer les données et remplir les zones de trous du maillage. Dans la deuxième étape, ils détectent le visage et les zones caractéristiques (les yeux et le nez) en utilisant OpenCV. Cette opération est effectuée sur les données RVB. La correspondance entre les données RVB et les données de profondeur de la Kinect permet de retrouver les zones caractéristiques du visage sur les données de profondeur. Ils détectent notamment deux zones caractéristiques correspondant aux yeux. Pour détecter l'emplacement des pupilles, ils prennent le milieu de ces deux zones. Pour détecter le bout du nez, ils prennent le point le plus proche de la caméra de la zone du nez. Enfin, ils détectent le menton en utilisant le point du bout du nez détecté et les cartes de profondeur. Les points caractéristiques sont ensuite utilisés pour segmenter le visage. Cao et al [8] utilisent l'algorithme de Kinect Fusion [41] pour reconstruire la forme du visage. Cela leur permet d'éliminer le bruit et de lisser les données de profondeur fournies par la caméra Kinect. Pour chaque expression, ils utilisent l'algorithme Active Shape Model (ASM) [67] pour détecter 74 points caractéristiques du visage sur l'image RVB fournie par la Kinect. Nous pouvons noter que leur méthode n'est pas complètement automatique. Certains points sont réajustés manuellement pour une meilleure précision. Les points caractéristiques sont divisés en deux catégories : les points du contour du visage et les points sur le visage (yeux,

nez, bouche...). La correspondance entre l'image couleur et la carte de profondeur est connue. C'est pourquoi nous pouvons facilement connaître les cordonnées 3D des points caractéristiques. Zollhofer et al [64] estiment la pose de la tête en utilisant l'algorithme Procruste [68] et détectent les points caractéristiques (bouche, nez et yeux) à partir du classifieur de Haar [69]. Ces deux algorithmes permettent d'initialiser l'algorithme ICP point-plan [70] qui permet d'effectuer l'alignement rigide. La deuxième étape est la fusion des données de profondeur alignées dans l'étape précédente. Ils utilisent une méthode de fusion similaire à Kinect Fusion [41].

Après avoir traité les données de profondeur, le modèle déformable est adapté aux données. Cette étape est composée d'une transformation rigide (translation, rotation et scale) et d'une transformation non rigide. Zollhofer et al [63] utilisent les points caractéristiques détectés dans la première étape pour aligner les données de profondeur. Ensuite, ils adaptent un modèle déformable de visage 3D sur le maillage 3D cible obtenu dans l'étape précédente en minimisant un terme d'énergie. Tout d'abord ils alignent grossièrement le visage moyen du modèle et le maillage obtenu en utilisant les points caractéristiques et une généralisation de l'analyse Procruste [68]. Le terme d'énergie est composé d'un terme de *fitting* et d'un terme de régularisation. Le terme de *fitting* est une erreur de distance entre le maillage du modèle que l'on veut déformer et le maillage cible. Le terme de régularisation permet de d'aligner plus précisément les données et d'empêcher les déformations non pertinentes. Cao et al [8] adaptent le modèle déformable de Blanz et Vetter [59] sur le maillage du visage neutre obtenu avec Kinect Fusion [41]. Le but de leur algorithme est de trouver les coefficients du modèle qui minimisent un terme d'énergie. Le modèle se déforme pour s'adapter le mieux possible au maillage 3D cible tout en faisant correspondre les points caractéristiques. Le *fitting* se déroule en deux parties. Premièrement, ils optimisent une énergie composée de trois termes. Le premier terme est la distance entre les points caractéristiques détectés sur le maillage cible et le maillage du modèle déformable. La distance est calculée à partir des coordonnées 3D pour les points à l'intérieur du visage et à partir des coordonnées 2D pour les points du contour du visage. Le deuxième terme correspond à la distance entre les points du maillage cible et les points les plus proches du maillage du modèle. Le dernier terme est un terme de régularisation (Tikhonov [71]) qui permet d'éviter que le maillage ne se déforme de façon importante. Deuxièmement, au bout de cinq à huit itérations de l'algorithme d'optimisation, le terme de régularisation est remplacé par un algorithme de déformation en fonction du Laplacien [72]. Ce terme permet de raffiner le maillage. Ensuite pour les 19 autres expressions, ils utilisent un algorithme de transfert d'expression [73]. Finalement, ils obtiennent, pour 150 personnes, un maillage 3D sémantique pour chaque expression. Zollhofer et al [64] adaptent le modèle déformable de Basel [6] sur les données fusionnées. Pour ce faire, ils optimisent un terme d'énergie qui permet de retrouver la forme, l'albédo et l'illumination. Le terme d'énergie est composé d'un terme de *fitting* de la profondeur, d'un terme de *fitting* de la couleur et d'un terme de régularisation. Le terme de *fitting* de la profondeur est l'erreur de distance entre le maillage cible et le maillage du modèle déformable. Le terme *fitting* de la couleur est la différence d'albédo et d'intensité entre les deux maillages. Le dernier terme est un

régularisateur statistique [59]. Il permet d'éviter le sur-apprentissage des données d'entrée. Leur méthode est temps réel et itérative. En effet, à chaque nouvelle trame en entrée de l'algorithme, les trois étapes sont effectuées. Odobez et al [4] adaptent ce modèle sur des données de profondeur capturées avec une caméra de type Kinect. Ils minimisent une fonction de coût composée de trois termes. Le premier terme correspond à la distance entre les points du visage cible et les points les plus proches du visage du modèle. Le deuxième terme correspond à la distance entre les points caractéristiques des deux maillages 3D et le troisième est un terme de régularisation. Dans leur méthode, ils utilisent un placement manuel de points caractéristiques sur les données de la Kinect. Pour détecter la pose de la tête et le regard à partir du clone 3D, ils utilisent ensuite un algorithme du type *Iterative Closest Points plan* [70].

Conclusion

L'ensemble des méthodes décrites précédemment utilise des modèles déformables de visage 3D. Ils permettent d'obtenir des clones sémantiques de haute résolution. Ces techniques dépendent beaucoup de la qualité du modèle de visage. En effet, les spécificités d'un individu ne peuvent être retrouvées que si leurs déformations appartiennent à la base de données. Il est donc primordial d'utiliser une base de données composée de nombreux visages diversifiés. L'utilisation de points caractéristiques peut permette d'améliorer le *fitting*. Mais les méthodes sont sensibles à la précision de la détection de ces points. C'est pourquoi certaines méthodes réajustent les points manuellement. Après avoir reconstruit la forme du visage, il faut reconstruire la texture du visage pour que le clone soit plus réaliste. Pour toutes ces raisons, dans notre méthode, nous utilisons un modèle déformable de visage pour reconstruire la forme 3D. Les modèles déformables de visage sont aussi utilisés dans nombreuses applications comme la reconnaissance faciale [6], le transfert d'expressions [8], l'animation de visage [65] ou encore la génération de stimuli pour des expériences psychologiques [74].

II.2.2 Reconstruction de la texture

Dans cette partie, nous présentons les méthodes de reconstruction de texture que l'on peut trouver dans la littérature. Il existe des méthodes manuelles mais aussi des méthodes automatiques. Nous allons présenter principalement le deuxième type de méthodes. Dans la première partie, nous présentons les méthodes qui *mappent* une texture fournie par un capteur sur un maillage 3D. Dans la deuxième partie, nous présentons les méthodes qui estiment la texture du visage en utilisant un modèle déformable.

II.2.2.1 *Mapping* de la texture fournie par le capteur

Certaines techniques dans la littérature permettent de reconstruire une texture 3D à partir des données RVB fournies par un capteur. Nous détaillons dans cette sous-partie les techniques

qui sont utilisées pour *mapper* une texture sur un visage ou un objet 3D. He et al [9] présentent une méthode pour reconstruire la texture d'un visage à partir d'une image RVB. Elle permet de *mapper* une texture sur n'importe quel maillage 3D de visage. Leur méthode est composée de quatre étapes : la détection du visage et des points caractéristiques, la création des coordonnées de texture, l'estimation de la pose de la tête et la reconstruction de la carte de texture. La méthode de Zhang et Luo [75] permet de *mapper* une texture sur un maillage 3D de visage à partir de 2 images RVB orthogonales. Elle est composée de quatre parties : la génération de la texture, la projection cylindrique, le *warping* et la reconstruction du modèle 3D. Xu et al [76] décrivent une technique qui permet de générer une texture 3D et de la *mapper* sur un maillage 3D. Cette méthode n'est pas conçue spécifiquement pour des maillages 3D de visages. Elle est composée de quatre étapes et utilise en entrée un maillage 3D et des images RVB calibrées. Hwang et al [77] présentent une méthode qui permet de *mapper* une carte de texture sur un maillage 3D. Elle est composée de 2 étapes : l'extraction de la texture et la génération de la carte de texture. Ces méthodes sont souvent constituées de trois étapes. Tout d'abord, le maillage 3D est déplié pour créer une carte de texture et des coordonnées de texture. Ensuite, la carte de texture est complétée par les différentes images RVB fournies par le capteur. Pour finir, des traitements sont appliqués pour améliorer la qualité de la texture.

Pour pouvoir *mapper* une texture 2D sur un maillage 3D, il faut créer une carte de texture 2D à partir du maillage 3D et des coordonnées de texture. Ces coordonnées permettent de faire la correspondance entre chaque point du maillage et la carte de texture. Il existe plusieurs techniques dans la littérature pour déplier un maillage 3D sur une carte 2D. He et al [9] utilisent le logiciel FaceGen produit par Singular Inversions Inc qui permet d'obtenir une carte de texture qui contient la texture dépliée du visage et les coordonnées de texture qui aident à déterminer comment cette carte de texture est *mappée* sur le visage 3D. Pour créer la carte de texture, Zhang et Luo [75] projettent le maillage 3D sur un plan 2D cylindrique. Cela permet d'obtenir une carte de texture dépliée du maillage 3D. Ils n'utilisent pas de coordonnées de texture. Après avoir complété la carte de texture avec la texture d'image RVB, ils reconstruisent le modèle 3D en utilisant la projection cylindrique inverse. Xu et al [76] utilisent la méthode de Katz et al [78] pour trouver les relations entre les images et le maillage 3D à partir des matrices de calibration. Hwang et al [77] se servent d'une cartographie cylindrique pour extraire la texture. Tout d'abord, un cylindre virtuel est placé autour du maillage 3D du visage. Ensuite, les sommets du maillage 3D sont projetés sur un plan image, puis intersectés par le rayon passant par le centre du cylindre.

Après avoir obtenu la carte de texture, il faut compléter cette carte avec la texture que nous voulons *mapper* sur le maillage 3D. En effet, la carte de texture est vierge et ne contient donc pas la texture du visage. He et al [9] reconstruisent la carte de texture à partir d'une image d'entrée RVB. Ils détectent d'abord les points caractéristiques sur l'image RVB à partir d'un algorithme Active Appearance Model (AAM) [79]. Ensuite, ils calculent la pose de la tête en utilisant les points caractéristiques détectés sur l'image d'entrée et des points caractéristiques du maillage 3D. Tous les points du maillage 3D sont alors projetés sur l'image RVB d'entrée.

Cette technique permet de remplir la région centrale de la carte de texture (yeux, nez, bouche...). Les autres parties sont remplies avec une texture par défaut fournie par le logiciel FaceGen. Si l'image d'entrée ne permet de compléter qu'une partie de la carte de texture (profil droit), une symétrisation de la texture est réalisée. Ces méthodes peuvent aussi être utilisées à partir de plusieurs images d'entrée. Des traitements sont ensuite effectués pour améliorer la texture. Zhang et Luo [75] utilisent une image de face et une image de profil du visage pour compléter la carte de texture. Dans la première étape, ils génèrent la texture à l'aide d'une technique multi-résolution. Ils fusionnent les 2 images en utilisant une ligne de référence (racine des cheveux, des sourcils, coins des yeux, la bouche et le menton) qui a été définie sur les images. Dans l'étape suivante, ils déterminent les correspondances entre la texture et la carte de profondeur dépliée. Pour cela, ils utilisent l'algorithme de *warping 2D* en fonction des paires de lignes caractéristiques [80]. Cette technique permet d'obtenir des résultats satisfaisants. Mais l'étape de *warping* doit être correctement effectuée pour obtenir les résultats souhaités. Lee et Magnenat Thalmann [81] proposent une méthode pour *mapper* une texture sur un visage 3D. Dans la première étape de leur technique, ils utilisent une image de face et une image de chaque profil du visage pour former une carte de texture dépliée. D'abord, les 2 images représentant les 2 profils sont déformées pour être alignées avec l'image de face. Ils utilisent une méthode manuelle pour fusionner les images en faisant correspondre des points caractéristiques et des lignes du visage. Dans la dernière étape, cette carte de texture est adaptée sur le maillage 3D. La texture est projetée sur le maillage 3D en utilisant un alignement manuel. Cette technique permet de *mapper* une texture sur un maillage 3D mais n'est pas complètement automatique. Xu et al [76] projettent sur les différentes images les faces visibles des maillages 3D. Ensuite, ils utilisent la méthode de Lempitsky et Ivanov [82] pour trouver, pour chaque face du maillage 3D, la texture la plus adaptée. Hwang et al [77] extraient et mettent en correspondance les pixels de trois images prises de points de vue différents avec la carte de texture. Ils utilisent une *Modified Image Stitching Method* pour générer la carte de texture [83]. Ce procédé permet de créer une texture de l'ensemble du visage.

Pour pouvoir obtenir des résultats de qualité, des traitements sont ensuite effectués sur la carte de texture complétée. En effet, les textures reconstruites à partir de plusieurs images RVB contiennent souvent des coutures. Elles sont dues aux différences de teintes et de luminosité entre les différentes images. Plusieurs traitements permettent de les éliminer. He et al [9] effectuent un lissage et un filtrage pour éliminer le flou et le changement de teinte au niveau de la transition entre les différentes parties de la carte de texture. Zhang et Luo [75] utilisent une méthode multirésolution de décomposition d'image RVB pour éliminer les frontières et lisser les images de texture [84].

Lee et Magnenat Thalmann [81] fusionnent trois images à partir d'une méthode de décomposition pyramidale qui utilise un opérateur Gaussien [85]. Ensuite ils corrigent les couleurs de la texture du maillage. Xu et al [76] observent des coutures à certains endroits de la texture. Elles sont dues aux conditions d'éclairage différentes entre les images RVB. C'est pourquoi ils transforment l'espace couleur RVB en espace TSV (Teinte, Saturation et Valeur) [86]. Après

avoir compensé les différences d'éclairage, ils utilisent une méthode de mélange de Gradien [87] pour compenser les différences d'intensité et améliorer la qualité de la texture. Pour pouvoir supprimer les coutures de la carte de texture, Hwang et al [77] se servent du gradient de la texture. Pour finir, ils utilisent des opérations morphologiques pour compléter les trous et ajoutent des yeux artificiels. Certaines méthodes utilisent les équations de Poisson pour pouvoir éliminer ces coutures. Ge et all [88] présentent dans leur article une méthode de reconstruction d'une texture d'un visage à partir des équations de Poisson [89]. Elles permettent d'éliminer les coutures d'une texture. Cette approche est basée sur l'équation différentielle partielle de Poisson avec les conditions aux limites de Dirichlet. Ces équations sont le plus souvent utilisées sur des données 2D. Dans leur méthode, ils les utilisent sur les données 3D d'un visage. Les pixels des données 2D sont remplacés par des points géométriques 3D. Pour pouvoir utiliser les équations de Poisson, les voisins de chaque point doivent être connus. La base de données de visage 3D BJUT-3D sert de base de test. Cette base de données est composée de 1500 visages 3D qui sont unifiés dans un système de coordonnées. Chaque maillage 3D de cette base de donnée est une maillage sémantique. Dans leur méthode, les auteurs découpent chaque maillage en 122 patchs de formes et de texture. Pour tester leur méthode, ils collent un patch d'un visage 3D (ex : le nez) sur un autre visage de la base. Pour améliorer leur technique, ils réalisent un pré-traitement sur le patch avant qu'il ne soit fusionné avec le visage cible. Ils modifient l'état d'éclairage du patch en utilisant une méthode inspirée par les méthodes de transfert de couleur [90, 91]. Ils utilisent une régression linéaire pour transférer la distribution des couleurs du patch au visage cible. Dessein et al [89] proposent aussi une méthode pour utiliser les équations de Poisson [89] sur un maillage 3D. Elle est composée de trois étapes : l'échantillonnage de l'image, la segmentation du maillage et l'utilisation du gradient pour éliminer les coutures. Dans la première étape, une interpolation bilinéaire de couleur sur la grille de pixel de chaque image est réalisée. Pour chaque vue, l'image est échantillonnée sur les pixels visibles du maillage 3D en utilisant une rétro-projection. Dans la deuxième étape, ils segmentent le maillage 3D en se basant sur une méthode de *fast marching* [92]. Dans la dernière étape, ils éliminent les coutures en utilisant les équations de Poisson avec les conditions de Dirichlet. Comme dans la méthode précédente, ces équations sont utilisées sur les points 3D du maillage. Pour améliorer leurs résultats, ils proposent une étape facultative pour transformer la couleur de la texture. Ils utilisent la technique de Bannai et [93]. Pour tester leur méthode, ils utilisent la base de données de Rooster. Elle se compose d'un maillage 3D avec une texture obtenue à partir de huit cartes de profondeur. Les textures fournies contiennent du bruit et des coutures. Ils réalisent une deuxième série de tests sur des photographies de visage de la base CMU PIE [94]. La technique de Paysan et al [6] est utilisée pour reconstruire la forme et la texture du visage à partir de trois photographies du visage. Leur méthode permet d'améliorer la texture 3D obtenue. Weise et al [95] utilisent aussi les équations de Poissons pour éliminer les coutures de la texture dépliée obtenue à partir des données d'une caméra Kinect. Certains articles de la littérature ne décrivent pas explicitement leur méthode de *mapping* de la texture. Par exemple, Hernandez et al [45] et Sun et al [44] expliquent juste qu'ils projettent une image

RVB de face sur le maillage 3D. Han et Jain [96] *mappent* une image RVB de face en utilisant les points caractéristiques détectés dans l'étape de reconstruction de la forme. Schneider et Eisert [97] projettent des images de face et de profil du visage en utilisant des points caractéristiques manuellement annotés.

Ils existent beaucoup de méthodes dans la littérature pour *mapper* une texture 2D sur un maillage 3D. La principale difficulté est de faire correspondre les points du maillage avec les différentes images RVB. Cette étape est primordiale pour obtenir des résultats satisfaisants. La deuxième difficulté est l'élimination des coutures de la texture. En effet, des transitions de couleur et de luminosité apparaissent quand plusieurs images RVB sont fusionnées. Ces techniques ont l'avantage de reconstruire les spécificités d'une personne. Par exemple, les grains de beauté ou les moustaches apparaissent dans les résultats.

II.2.2.2 Estimation de la texture à partir d'un modèle déformable

Certaines techniques de reconstruction de texture utilisent des modèles déformable de visage 3D. Comme pour la forme, une optimisation des paramètres du modèle permet de retrouver la texture la plus proche du visage cible. Plusieurs articles dans la littérature utilisent ce type de méthodes.

Blanz et Vetter [98] utilisent un modèle déformable de visage 3D pour reconstruire la forme et la texture du visage à partir d'une images RVB. La reconstruction de la forme est expliquée dans la partie II.2.1. Le modèle déformable utilisé permet de reconstruire la forme des visages mais aussi la texture. Comme pour la forme, la texture du visage cible est retrouvée à partir de l'optimisation décrite dans la partie 2.2.1. Ce type de technique n'est pas performant pour des visages possédant des caractéristiques physiques individuelles (grain de beauté, barbe...). En effet, elles sont très dépendantes de la variabilité des caractères physiques des visages qui constituent la base de données. Pour résoudre ce problème, Blanz et Vetter extraient ces caractéristiques individuelles de l'image qui représente le visage cible. Ils utilisent la technique de Pighin et al [99] qui permet d'extraire une texture à partir d'image RVB. Ensuite, ils comparent la texture retrouvée avec le modèle déformable et avec celle extraite de l'image cible. Dans les zones occultées de l'image, ils comptent sur la prédiction faite par le modèle. Sur le reste du visage, ils modifient la texture à partir de la comparaison effectuée précédemment. L'inconvénient de cette méthode est qu'il faut utiliser en plus une extraction de texture à partir d'une image pour améliorer la texture obtenue avec le modèle déformable.

Zollhofer et al [64] présentent aussi dans leur article une méthode de reconstruction de la forme et la texture à partir des données fournies par une Kinect. Ils utilisent donc des données de profondeur et des données de couleur. La reconstruction de la forme est détaillée dans la partie 2.2.1. Avant d'utiliser un modèle déformable, les données de forme et de texture sont fusionnées en utilisant une méthode semblable à Kinect Fusion [41]. Ils utilisent le Basel Face modèle [6] et se servent d'une optimisation non linéaire pour retrouver la forme, l'albédo et l'illumination du

visage cible. L'illumination est modélisée dans la fonction de coût par un terme de *fitting* pour la couleur. Ils utilisent des harmoniques sphériques pour la modélisation de l'illumination. Le principal défaut de cette méthode de reconstruction de la texture est qu'elle ne permet pas de reconstruire les caractéristiques spécifiques d'un individu (barbe...).

Les méthodes de reconstruction de texture à partir d'un modèle déformable de visage permettent d'obtenir des textures de visage de haute résolution. Mais ces méthodes sont limitées. En effet, comme pour la forme, elles dépendent beaucoup de la qualité du modèle du visage. Il faut que la base de données soit la plus diversifiée possible pour pouvoir retrouver certaines spécificités d'un visage. Mais, contrairement à la forme, la texture d'un visage est beaucoup plus diversifiée. C'est pourquoi une caractéristique physique individuelle (barbe, grain de beauté...) a très peu de chance d'être retrouvée par le modèle. Certaines méthodes utilisent les données RVB fournies par une caméra pour modifier les résultats. Dans notre méthode, nous n'utilisons pas de modèle déformable pour reconstruire la texture. En effet, nous avons préféré directement générer et *mapper* directement la texture fournie par la caméra RVB-Z.

II.2.3 Conclusion

Nous avons présenté des méthodes qui permettent de reconstruire la forme et la texture d'un visage. Les méthodes qui ne fournissent pas de maillages sémantiques ne peuvent pas être directement utilisées dans des applications du type détection du regard [4]. De plus, il est très difficile d'obtenir les caractéristiques physiques de certaines parties du visage. Notamment au niveau des yeux et de la bouche. En effet, les données des capteurs basse résolution sont limitées au niveau de la précision. Ils ne captent pas correctement les yeux qui reflètent les infrarouges. C'est pourquoi, certaines méthodes utilisent des modèles déformables pour résoudre ces problèmes. Mais ces méthodes dépendent beaucoup de la qualité de la base de données du modèle. En effet, les spécificités d'un individu ne peuvent être retrouvées que si elles se trouvent dans cette base de données. Ces méthodes permettent donc d'obtenir les caractéristiques physiques de certaines parties du visage qui ne sont pas fournies par les capteurs, mais elles ne permettent pas de retrouver certaines spécificités d'un individu. C'est pourquoi, nous proposons une méthode qui fusionne les données fournies par le capteur et les données obtenues avec un modèle pour obtenir notre clone de visage. Notre méthode de détection et de fusion par patchs appartient à la catégorie des techniques qui utilisent un modèle déformable de visage 3D pour la reconstruction de la forme. Notre technique permet d'être moins dépendant de la qualité du modèle utilisé. En effet, nous utilisons un système de détection et de fusion de patchs de forme pour ne sélectionner que les parties correctes des trames de profondeur fournies par le capteur. Pour reconstruire la texture, nous utilisons une méthode de *mapping*. En effet, elles sont plus performantes pour scanner la texture de visages diversifiés. Nous utilisons aussi une technique de patchs pour reconstruire la texture afin d'éviter les coutures et reconstruire toutes les spécificités de la personne.

Troisième partie

Clonage de visage 3D par patchs

Notre méthode permet de numériser un visage 3D à partir d'une caméra basse résolution RVB-Z. Elle est composée de 2 parties qui sont présentées dans ce chapitre. La figure III.0.4 décrit les différentes étapes de notre technique. L'acquisition des données est effectuée avec une caméra du type Kinect. Elle permet d'obtenir des données de couleur et de profondeur du visage. Dans le premier chapitre (III.1), nous détaillons comment nous avons reconstruit la forme 3D du visage. Ce chapitre est composé 3 sections : le *fitting*, la détection des patchs de forme et la fusion de ces patchs. Tout d'abord, nous utilisons un modèle déformable de visage 3D pour augmenter la résolution et supprimer le bruit des trames de profondeur (étape de *fitting*). Ensuite, nous détectons les parties adéquates et précises (patchs) des différents maillages obtenus avec le *fitting*. Pour finir, nous fusions les parties détectées pour reconstruire la forme entière du visage 3D. Dans le deuxième chapitre (III.2), nous décrivons notre algorithme qui permet de reconstruire la texture du visage 3D. Ce chapitre est composée de 4 sections : l'alignement rigide des trames de profondeur et du clone, la détection des patchs de texture, le *warping* et la fusion des patchs de texture. Premièrement, nous alignons chaque trame de profondeur du capteur RVB-Z avec le clone sémantique. Deuxièmement, nous détectons les zones adéquates et précises (patchs) des trames de texture, c'est-à-dire les zones où la caméra RVB-Z a correctement capté la texture (sans bruit). Puis dans la troisième étape, nous créons la carte de texture du clone et nous *warpons* les patchs de texture détectés pour compléter cette carte. Pour finir, nous fusionnons les différents patchs de texture pour reconstruire entièrement la texture du clone 3D.

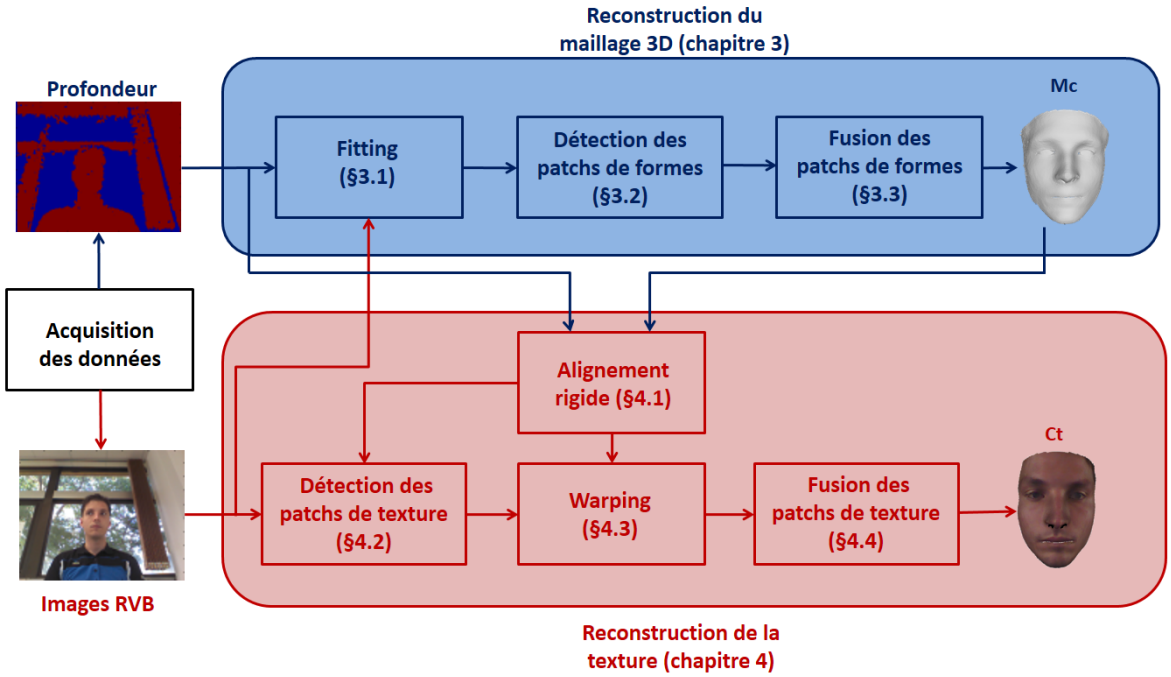


FIGURE III.0.4: Notre système de clonage de visage. Pour reconstruire la forme 3D du visage, nous réalisons d'abord une étape de *fitting* sur chaque trame de profondeur fournie par le capteur RVB-Z. Nous utilisons les données de couleur pour initialiser l'alignement rigide du *fitting*. Ensuite, nous détectons et fusionnons les patchs de forme qui correspondent aux zones adéquates et précises des **n** maillages 3D obtenus dans la première étape. La fusion permet de reconstruire entièrement la forme du visage 3D. Pour reconstruire la texture du clone, nous utilisons à la fois les trames de forme et de texture pour détecter les patchs de texture. Ensuite, nous *warpons* ces patchs pour compléter entièrement la carte de texture du clone. Pour finir, nous fusionnons les patchs qui contiennent la même information de texture du visage.

Chapitre III.1

Reconstruction de la forme du visage 3D

Dans ce premier chapitre, nous décrivons notre technique de reconstruction de la forme 3D du visage. La figure III.1.1 décrit les différentes étapes de la reconstruction de la forme. Cette partie est composée de 3 sections : le *fitting* à partir d'un modèle déformable de visage, la détection des patchs de forme et la fusion de ces patchs. Pour chaque personne, nous capturons les données RVB-Z de **n** vues du visage avec une caméra Kinect. Les données de profondeur fournies par le capteur sont très bruitées et de basse résolution. C'est pourquoi dans la première section, nous réalisons un *fitting* pour augmenter leur résolution et éliminer le bruit. Tout d'abord dans cette section, nous filtrons les données de profondeur en utilisant un filtre bilatéral. Ensuite nous ajustons un modèle déformable de visage M(section III.1.1) sur chaque trame de profondeur D_P (p = 1 à n). C'est-à-dire que nous effectuons d'abord un alignement rigide entre le modèle M et la trame traitée D_p, puis nous déformons le modèle pour qu'il s'adapte à la trame (transformation non rigide). Après cette étape de *fitting*, nous obtenons **n** maillages sémantiques M_p correspondant aux **n** trames D_P. Dans la deuxième section, nous détectons **n** patchs de profondeur P_n (section III.1.2) correspondant aux données de différents maillages M_p. En effet, chaque trame de profondeur ne contient pas toute l'information du visage. Une trame de profil droit ne contient pas l'information du profil gauche du visage. C'est pourquoi certaines parties des maillages sémantiques M_p obtenues à partir de ces trames ne sont pas adéquates et précises. Donc c'est pour cela que nous utilisons des patchs de forme. Enfin dans la dernière section, nous fusionnons (section III.1.3) les différents patchs que nous avons détectés pour générer un clone 3D complet M_c. Nous présentons différents types de fusion des données des patchs de profondeur.

Sommaire

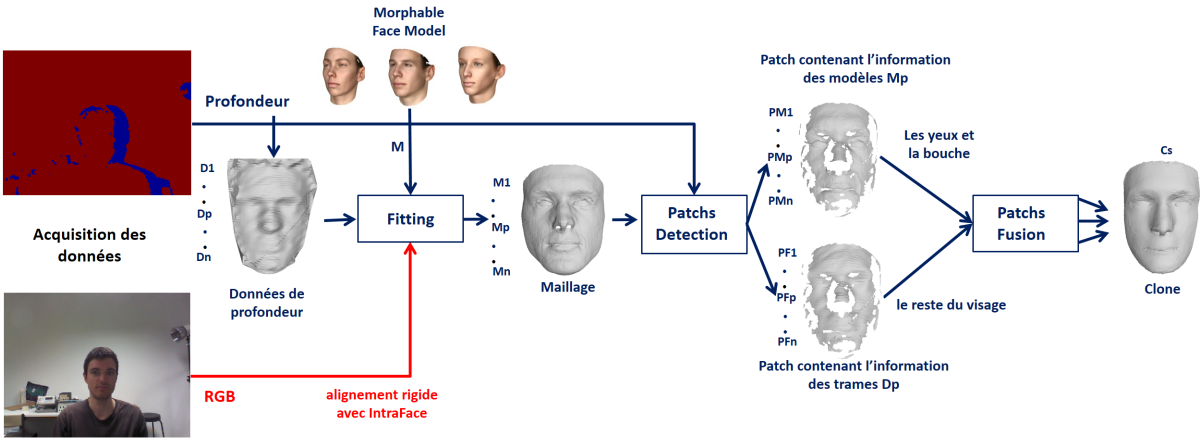


FIGURE III.1.1: Notre système de détection et de fusion des patchs de forme : il est composé de 3 sections. Tout d'abord, nous réalisons un *fitting* sur chaque trame de profondeur fournie par le capteur RVB-Z. Nous obtenons **n** maillages 3D sémantiques. Ensuite nous détectons les zones (patchs de forme) précises contenant l'information du visage de chaque maillage 3D. Nous détectons 2 types de patchs : ceux contenant les points des trames et ceux contenant les points des maillages. Dans la dernière section, nous fusionnons ces différents patchs de forme pour reconstruire la forme 3D entière du visage.

III.1.1 *Fitting* avec un modèle déformable de visage 3D

Dans cette section, nous décrivons notre technique de *fitting*. La figure III.1.2 montre les différentes étapes du *fitting*. Cette étape permet d'éliminer le bruit et d'augmenter la résolution des trames de la caméra RVB-Z. Elle est composée d'un prétraitement (filtrage) et de 2 étapes itératives (transformation rigide et non rigide). Nous effectuons d'abord un prétraitement pour filtrer chaque trame de profondeur D_P fournie par le capteur. Nous utilisons un filtre bilatéral [100] pour lisser les données et éliminer une partie du bruit. Puis, à chaque itération, nous calculons les matrices de rotation et de translation qui alignent chaque trame D_P avec le maillage M (transformation rigide). Nous utilisons l'algorithme Iterative Closest Points [101] pour trouver ces 2 matrices. Pour réduire le nombre d'itérations de l'algorithme ICP, nous initialisons l'angle de rotation en utilisant la pose de la tête (estimation des trois angles de rotation) calculée par Intraface [102]. IntraFace permet de trouver l'angle de rotation à partir des données de couleur 2D. Ensuite, nous déformons le maillage M, de sorte qu'il prenne la forme de la trame D_P (transformation non rigide). Nous utilisons l'algorithme d'optimisation de Gauss-Newton pour trouver les paramètres du modèle déformable qui minimise l'erreur de distance entre la trame D_P et le maillage M afin de générer autant de maillages déformés M_p que de trames en entrée.

III.1.1.1 Segmentation et filtrage bilatéral

Les capteurs RVB-Z fournissent des cartes de profondeur D_p et des images RVB I_p ou le visage n'est pas segmenté. Pour pouvoir éliminer les points qui n'appartiennent pas au visage de la personne, nous détectons le visage en utilisant le détecteur de Viola et Jones [103]. Il permet

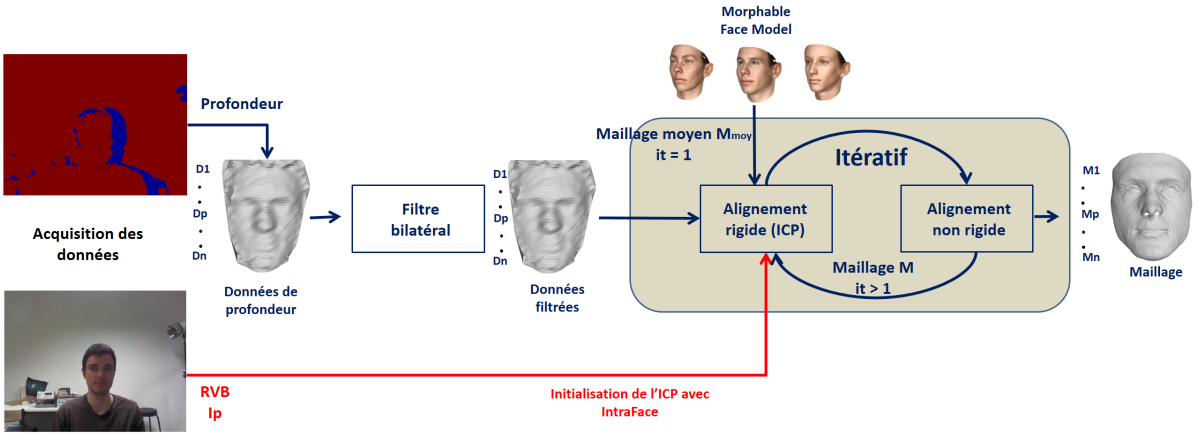


FIGURE III.1.2: Schéma explicatif de notre méthode de *fitting*. Elle est composée d'un prétraitement et de 2 étapes itératives. Chaque trame de profondeur est d'abord filtrée par un filtre bilatéral. Ensuite nous utilisons un modèle déformable de visage pour augmenter la résolution des données, éliminer le bruit et obtenir des maillages sémantiques. Nous déformons le maillage moyen du modèle en optimisant les paramètres du modèle. À chaque itération, le maillage du modèle et la trame sont alignés (transformation rigide) et le maillage déformé (transformation non rigide).

de détecter sur les cartes de texture RVB le visage. Grâce à la correspondance entre les cartes de profondeur D_p et les cartes de texture I_p fournies par le capteur, nous connaissons les points 3D correspondant au visage pour chacune des trames.

Pour éliminer le bruit et lisser les trames de profondeur D_p, nous les filtrons. Les filtres passe-bas sont les filtres les plus utilisés et les plus simples pour supprimer le bruit d'une carte de profondeur. Ce type de filtre est linéaire et ne s'adapte pas aux données d'entrée. Par exemple le filtre Gaussien remplace chaque élément par la moyenne pondérée de ses voisins. Il ne prend en compte que la distance spatiale pour déterminer la contribution de chacun des éléments. C'est pourquoi, quand nous utilisons ce genre de filtre, les contours de l'objet ne sont pas préservés. Le filtre bilatéral est un filtre non linéaire qui s'adapte au contenu de l'objet et donc permet de lisser efficacement le bruit des zones uniformes sans détériorer les contours. En effet, la forme du filtre dépend du contenu de l'élément que l'on veut filtrer. La valeur de chaque élément est remplacée par la moyenne pondérée des valeurs semblables des éléments voisins. Dans la version la plus utilisée, la moyenne pondérée est calculée à partir de fonctions Gaussiennes. Les coefficients de ce type de filtre sont calculés en fonction de la distance euclidienne entre les pixels et de la ressemblance des valeurs entre le pixel traité et les pixels du masque. Dans une zone ou la valeur des pixels est semblable, le lissage sera de forte intensité. A contrario, dans une zone où les pixels ont des valeurs très différentes (frontière), le lissage sera de faible intensité. Le filtre bilatéral est décrit dans les équations ci-dessous :

$$I_f(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) h_S(||x_i - x||) h_I(||I(xi) - I(x)||) \qquad \text{(III.1.1)}$$

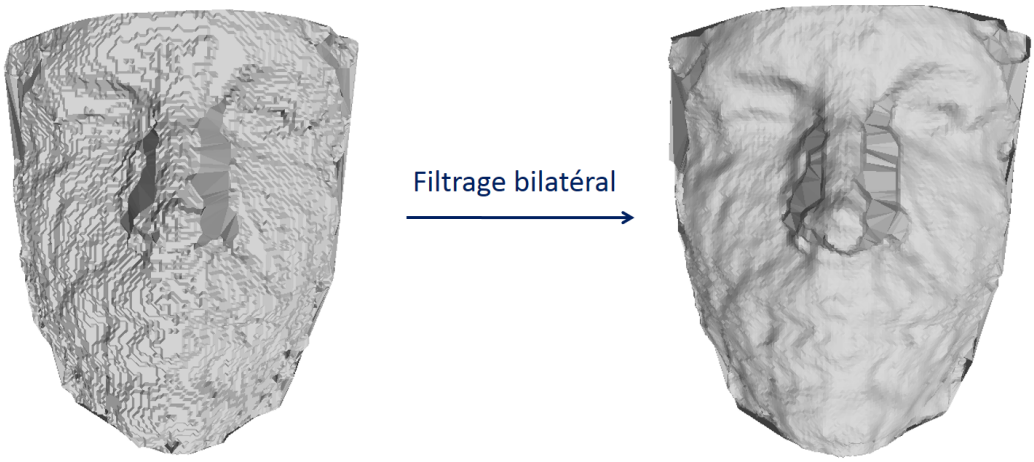


FIGURE III.1.3: Exemple de filtrage bilatéral des trames de profondeur. Nous obtenons un maillage 3D lissé.

$$W_p = \sum_{x_i \in \Omega} h_S(||x_i - x||) h_I(||I(xi) - I(x)||) \tag{III.1.2}$$

avec

- I_f : l'image filtrée
- I : image originale non filtrée
- x : coordonnées de l'élément traité
- W_p : terme de normalisation
- h_S : filtre spatial
- h_I : filtre d'intensité
- Ω : fenêtre centrée sur x

Les équations III.1.1 et III.1.2 définissent le filtre bilatéral. Le filtre bilatéral est composé de 2 termes de pondérations. Le filtre spatial h_S permet de pondérer le lissage en fonction de la distance des éléments de la fenêtre de traitement Ω et le filtre d'intensité h_I pondère en fonction des différences d'intensités entre les éléments de la fenêtre Ω. Le plus souvent ce sont des fonctions gaussiennes. Dans ce cas, la déviation standard σ permet de régler l'intensité du lissage. Nous utilisons un filtre bilatéral avant d'utiliser les cartes de profondeur fournies par le capteur RVB-Z (résolution 320*240). Ce filtre non linéaire est efficace sur ce type de données. Chaque valeur d'un pixel correspond à sa distance par rapport à la caméra. Dans notre méthode, nous utilisons une fenêtre Ω de 11*11 pixels. Nous utilisons des fonctions Gaussiennes pour le filtre d'intensité h_I et le filtre spatial h_S et un σ égal à 5 dans les deux cas. La figure III.1.3 montre une trame de profondeur avant et après le filtrage. Nous pouvons voir que la trame de profondeur a été lissée et la plupart des artéfacts éliminés. Le principal inconvénient de ce filtre est le temps de calcul coûteux. Nous segmentons le visage avant le filtrage pour que le temps de calcul soit le plus petit possible.

Le modèle déformable que nous utilisons dans notre méthode fournit des maillages séman-

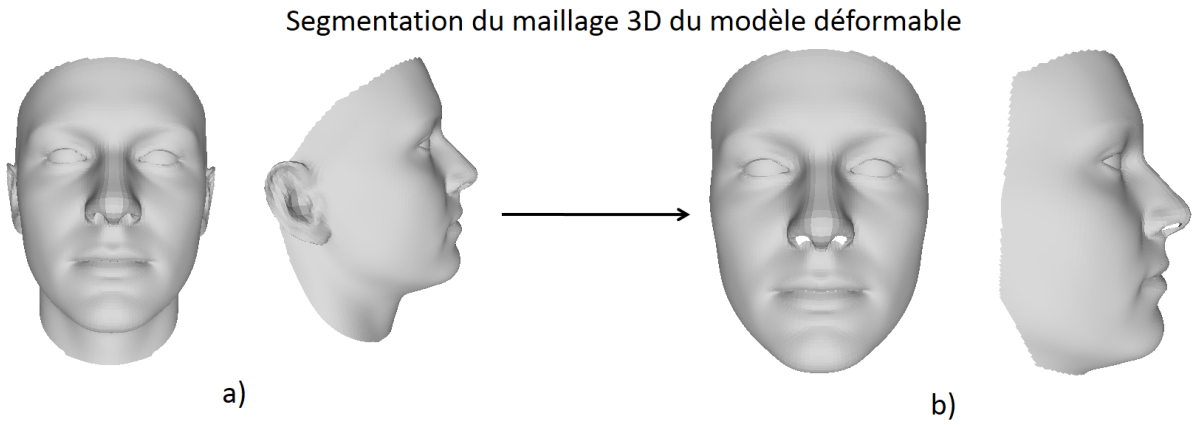


FIGURE III.1.4: Le maillage a) représente le maillage 3D moyen du modèle déformable de visage. Nous segmentons le cou et les oreilles car nous n'utilisons pas ces 2 parties du visage dans notre méthode. Le maillage b) est le maillage moyen du modèle après segmentation.

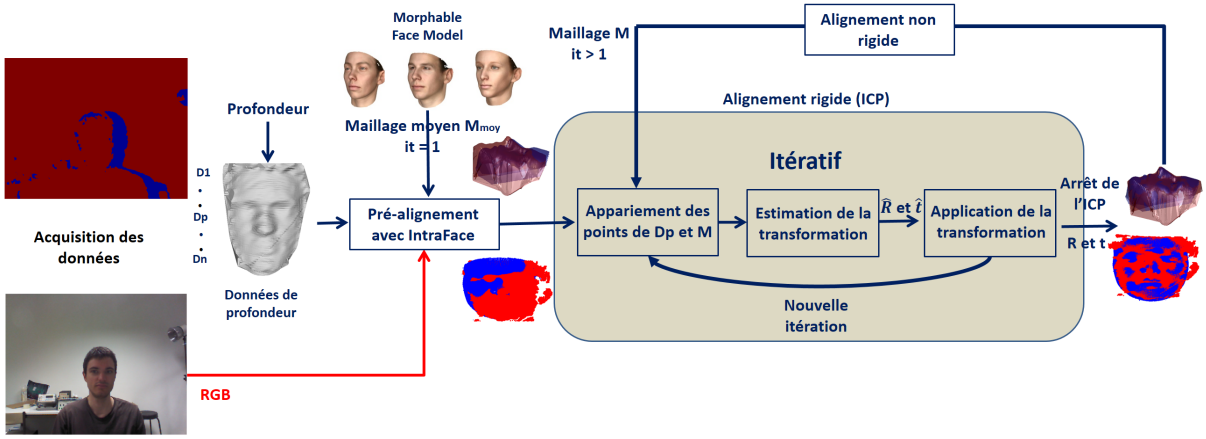


FIGURE III.1.5: Schéma explicatif de notre méthode d'alignement rigide. Elle est composée d'un préalignement et de 4 étapes itératives. Tout d'abord, nous utilisons Intraface [102] pour préaligner le maillage moyen M_{moy} du modèle déformable et la trame de profondeur D_p. Ensuite, nous utilisons l'algorithme ICP pour réaliser l'alignement rigide. Premièrement, nous apparions les points des 2 maillages en utilisant le critère du plus proche voisin. Puis, nous estimons la matrice de transformation qui aligne les 2 maillages. Pour finir, nous appliquons cette transformation et nous décidons si l'algorithme doit effectuer ou pas une nouvelle itération.

tiques composés du visage mais aussi du cou et des oreilles de la personne. Dans notre technique de reconstruction de la forme, nous segmentons le cou et les oreilles des maillages du modèle (voir figure III.1.4). Comme les maillages sont sémantiques, il est très facile d'éliminer les faces et points 3D correspondant à ces parties des maillages 3D.

III.1.1.2 Alignement rigide

À chaque itération du *fitting*, nous alignons chaque trame de profondeur D_p avec le maillage M du modèle déformable pour que la transformation non rigide se déroule correctement. La

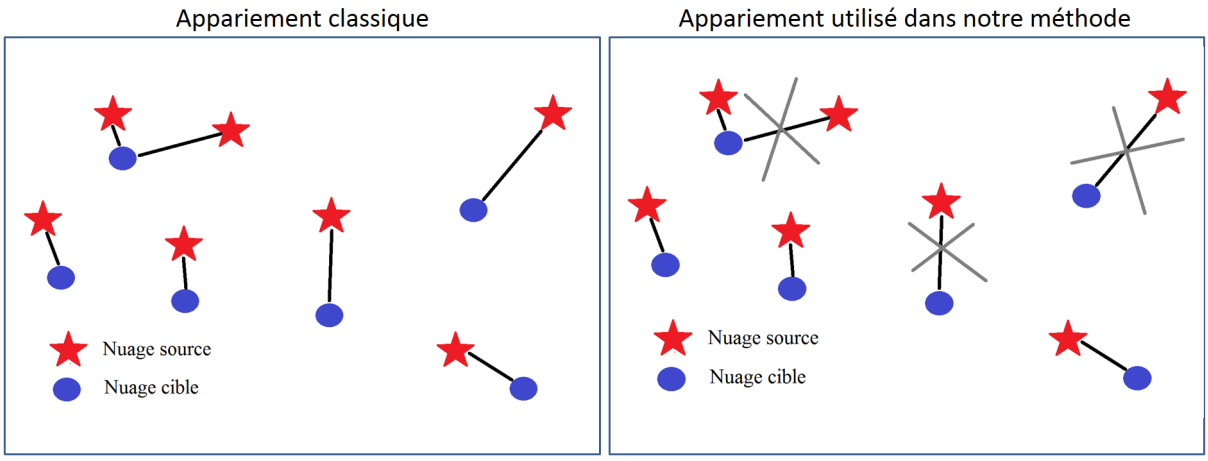


FIGURE III.1.6: La figure de gauche montre un exemple d'appariement de point en utilisant le critère du plus proche voisin. La distance euclidienne est utilisée pour calculer ce critère. La figure de droite présente notre méthode d'appariement. Nous utilisons aussi le critère du plus proche voisin. Mais, nous éliminons les appariements où la distance euclidienne est trop grande.

figure III.1.5 présente le schéma de notre méthode d'alignement rigide. Il existe beaucoup de méthodes dans la littérature qui permettent de calculer les transformations pour aligner 2 nuages de points 3D [101]. L'algorithme Iterative Closest Point est un algorithme souvent utilisé dans ce genre de méthode. Il est très populaire et très utilisé dans le domaine de la vision par ordinateur. Douadi et al [104] ont comparé plusieurs variantes de cet algorithme. Ils ont montré qu'il existe différentes techniques pour améliorer les résultats de l'algorithme ICP (utilisation de la couleur...). Cet algorithme est itératif et est composé de 4 étapes :

- l'appariement des points
- l'estimation de la transformation (translation et rotation)
- l'application de cette transformation
- la décision de réaliser ou pas une nouvelle itération

Il permet de calculer une matrice de transformation rigide (rotation et translation) à partir d'un nuage cible et d'un nuage source fournis en entrée de l'algorithme. Une étape d'initialisation est parfois utilisée pour aider l'algorithme à converger plus facilement et éviter les minimums locaux. L'étape d'appariement des points est la plus importante (voir figure III.1.6). En effet, les résultats peuvent ne pas être corrects si cette étape n'est pas bien réalisée. Le principal inconvénient de cet algorithme est sa lenteur. En effet, le coût de calcul des distances euclidiennes pour l'appariement des points 3D est très important.

La première étape de l'algorithme (appariement des points) peut avoir un impact sur sa convergence. En effet, l'appariement de mauvaises paires de points peut provoquer un ralentissement ou même faire converger l'algorithme dans un minimum local. Tout d'abord, il faut *présélectionner les points 3D* des 2 nuages que l'on va apparier. Cette présélection peut permettre à la fois d'accélérer l'algorithme mais aussi d'éliminer le bruit des nuages et donc d'éviter les

mauvais appariements. Masuda et al [105] proposent un sous-échantillonnage aléatoire des points 3D pour augmenter la vitesse de l'ICP et la robustesse aux faux appariements. Kim et al [106] ont montré qu'un échantillonnage suivant la direction radiale à partir du centre de gravité du nuage permettait d'obtenir des résultats identiques à un ICP classique en accélérant l'algorithme. Rusinkiewitcz et al [107] utilisent un sous-échantillonnage en fonction des normales des points pour améliorer leurs algorithmes. Ses différentes méthodes de sélection sont efficaces et permettent d'augmenter notablement la vitesse de l'algorithme. Nous utilisons un sous-échantillonnage aléatoire dans notre méthode pour accélérer et augmenter la robustesse de notre algorithme.

Après avoir réalisé la présélection, il faut *apparier les paires de points* (voir figure III.1.6). Il existe plusieurs critères pour réaliser cette étape. Dans l'ICP classique, le critère du plus proche voisin à partir de la distance euclidienne entre les points des 2 nuages est utilisé. Pour chaque point du nuage cible, il faut calculer le point le plus proche du nuage source. Cette opération a un coût très élevé en temps de calcul. Certaines méthodes utilisent d'autres critères pour réaliser l'appariement. Par exemple, il est possible d'utiliser les données de couleur quand elles sont disponibles. Jost [108] utilise d'abord un critère de distance couleur pour trouver les points compatibles et ensuite la distance euclidienne pour apparier ces points. Certaines méthodes utilisent une distance mixte basée sur un critère de distance couleur et un critère de distance euclidienne. Douadi et al [104] montrent que l'utilisation de la couleur permet d'augmenter la robustesse de l'ICP. Mais la couleur des points 3D n'est pas toujours disponible. C'est pourquoi, certaines méthodes utilisent d'autres critères. En effet, les normales de surfaces [109] ou encore les moments des courbes [110] peuvent être utilisés car ils permettent d'améliorer l'étape d'appariement. Pour finir, certaines méthodes utilisent un seuil de distance (euclidienne...) pour éliminer les fausses paires de points. Mais ce seuil n'est pas toujours facile à choisir car les distances entre les paires de points évoluent au cours des itérations. C'est pourquoi, Zhang et al [52] proposent d'utiliser un seuil qui s'adapte en fonction du critère de distance choisi.

Ces critères permettent d'améliorer les résultats de l'algorithme mais augmentent le temps de calcul. Pour diminuer ce coût, certaines techniques *utilisent un arbre KDtree*. Les arbres KDtree (k-dimensional tree) sont des arbres binaires qui divisent l'espace 3D de points en plusieurs sous-espaces. Ils permettent d'accélérer les algorithmes de recherches du type plus proche voisin. En effet, il y a moins de distance entre les points appariés (distance euclidienne, couleur...) à calculer. Pour chaque point du nuage cible, il faut calculer la distance avec les sous-espaces et non plus avec chaque point du nuage source. La figure III.1.7 montre un exemple d'arbre KD tree pour un nuage de points 2D.

La deuxième étape consiste à estimer les matrices de rotation (3*3) et de translation (1*3) qui permettent d'aligner le nuage de points source p avec le nuage de points cible q selon la formule III.1.3. Dans la première étape, nous avons apparié les différents points des nuages. C'est à partir de cela que nous allons calculer la distance entre ces nuages. C'est pourquoi la première étape est très importante. En effet, si l'appariement n'est pas correct alors la distance entre les 2 nuages non plus, ce qui va entrainer une mauvaise estimation des matrices de transformation. Le critère

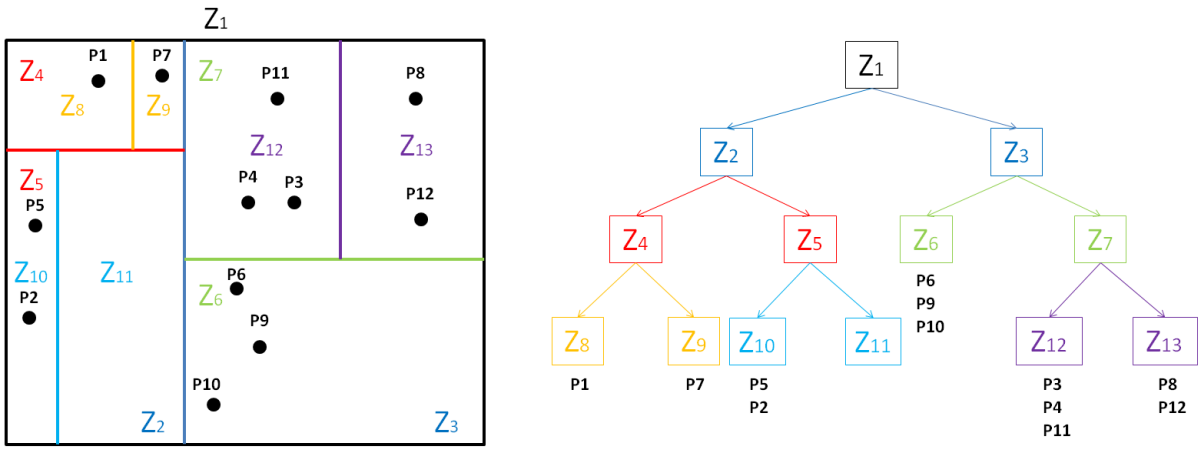


FIGURE III.1.7: Exemple d'arbre KD tree pour un nuage de point 2D. Les points (P_1 à P_{12}) se trouvent dans différentes zones (Z_2 à Z_{13}) de l'espace 2D (Z_1). L'arbre KD permet de diviser l'espace et donc d'accélérer la recherche des plus proches voisins. En effet, pour chaque point (P_1 à P_{12}), nous cherchons d'abord la zone (Z_2 à Z_{13}) la plus proche et ensuite le point le plus proche de cette zone.

de distance le plus simple est la somme des erreurs quadratiques entre les paires de points. Pour améliorer la robustesse de l'algorithme, il est possible d'utiliser une pondération. Luck et al [111] explique dans leur article, qu'une pondération des paires des points en fonction de la qualité de leur appariement permet d'augmenter les performances de l'ICP. Ils utilisent la médiane des carrés des distances entre les points appariés. Les poids sont alors calculés à partir de la distance (couleur, euclidienne...) entre 2 points appariés. Les poids sont inversement proportionnels à la distance entre les 2 points. Les équations III.1.3 et III.1.4 montrent comment s'effectue le calcul de la distance. Quand le critère est la somme des erreurs des distances quadratiques entre les paires de points, l'erreur obtenue est alors fonction de la matrice de rotation $\hat{R}$ et de la matrice de translation $\hat{t}$.

$$q = R * p + t \tag{III.1.3}$$

avec

- q : nuage cible
- p : nuage source

$$E_{ICP}(\hat{R}, \hat{t}) = argmin \sum ||(qs - (\hat{R} * ps + \hat{t}))||^2, qs \subset q, ps \subset p \tag{III.1.4}$$

avec

- $\hat{R}$: matrice de rotation estimée
- $\hat{t}$: matrice de translation estimée
- $E_{ICP}(\hat{R}, \hat{t})$: erreur à minimiser
- qs : Sous-ensemble de points du nuage cible appariés
- ps : Sous-ensemble de points du nuage source appariés

E_{ICP} est la somme de la distance au carré entre chaque point de la source (ps) et le point de destination correspondant du nuage cible (qs). Dans cette équation III.1.4, R et t sont les matrices de transformation. qs et ps sont les points appariés selon leur distance euclidienne. Ce sont des sous-ensembles de points de q et p.

Ensuite, il faut estimer les matrices de transformation en minimisant $E_{ICP}(\hat{R}, \hat{t})$. Il existe plusieurs méthodes dans la littérature pour résoudre ce problème. En effet, Arun et al [112] utilisent la décomposition en valeur singulière pour trouver les 2 matrices de transformations. Leur méthode permet de résoudre le problème en séparant la translation et la rotation. Pour estimer la rotation et la translation, certaines méthodes utilisent un système avec des quaternions. Faugeras et al [113] utilisent les quaternions unitaires et Walker et al [114] les quaternions duaux. L'estimation peut aussi être effectuée en utilisant un algorithme d'estimation des moindres carrés. Par exemple, Fitzgibbon et al[115] se servent d'un algorithme itératif de Gauss Newton pour réaliser cette opération. L'inconvénient de ce type d'algorithmes est leur temps de convergence. Certaines méthodes dans la littérature utilisent d'autres critères de distance pour le calcul de l'erreur entre les 2 nuages de points. Chen et al [109] et Low [101] présentent un critère de distance point à plan. C'est-à-dire qu'il calcule la distance entre les points d'un des nuages et les plans tangents contenant les points du deuxième nuage. L'estimation des 2 matrices de transformation est alors réalisée en effectuant une approximation linéaire et une décomposition en valeur singulière. Cette technique est plus robuste quand il faut aligner des nuages de points avec du bruit et des trous.

La troisième étape de l'algorithme est l'application des 2 matrices de transformation sur le nuage de points source. À chaque itération, la distance entre les points des nuages n'est plus la même. C'est pourquoi, il faut réeffectuer l'appariement des points et l'estimation de la matrice de rotation et de la matrice de translation.

Pour finir, dans la dernière étape, il faut déterminer à quel moment l'algorithme doit être arrêté. Il existe plusieurs possibilités de critère d'arrêt pour l'ICP. En effet, nous pouvons arrêter l'algorithme si un seuil minimal d'erreur est atteint, si l'erreur ne varie plus ou encore si le nombre maximal d'itérations est dépassé.

Initialisation

Dans notre méthode, nous voulons à chaque itération du *fitting* aligner la trame de profondeur D_p qui est traitée avec le maillage du modèle déformable M_p. L'algorithme ICP est plus performant quand il doit aligner 2 nuages de point 3D peu bruité et que le recouvrement entre les nuages est important. Le maillage fourni par le modèle contient l'information du visage en entier et n'est pas bruité. Mais souvent les trames de profondeur à aligner contiennent du bruit et ne sont pas forcément de bonne qualité. C'est pourquoi, nous initialisons l'algorithme d'alignement rigide en utilisant l'algorithme d'IntraFace [102] (voir la figure III.1.8). Il permet à la fois de détecter des points caractéristiques d'un visage dans une image RVB, mais aussi de calculer la pose de la tête. Chaque trame de profondeur a une correspondance avec une image RVB fournie

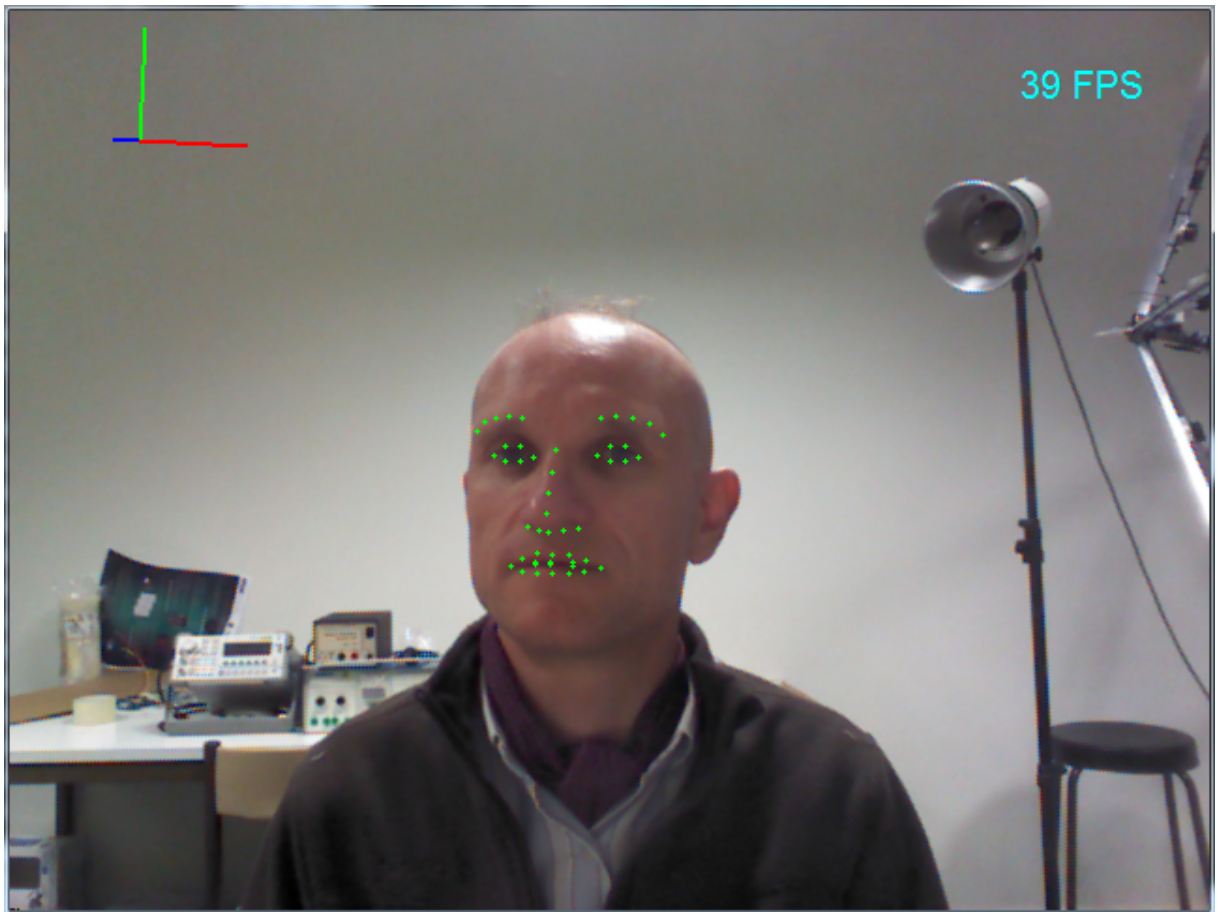


FIGURE III.1.8: Exemple d'une image RVB traitée avec l'algorithme IntraFace.

par le même capteur. Pour initialiser la rotation, nous appliquons l'angle de pose fourni par IntraFace sur les données de profondeur correspondant à l'image RVB. Et pour la translation, nous alignons les points détectés par IntraFace correspondant au bout du nez du maillage et de la trame.

Paramétrage de l'algorithme ICP

Pour le calcul de l'erreur de distance entre les 2 nuages, nous utilisons l'ICP point plan. En effet, il permet d'obtenir des meilleurs résultats avec des données bruitées (artéfact et trou dans le maillage). Nous ne possédons pas les données couleur correspondant à la texture du visage pour les maillages 3D fournis par le modèle déformable. C'est pourquoi, nous ne pouvons pas utiliser un critère de couleur dans notre algorithme d'alignement rigide. Nous utilisons les distances euclidiennes comme critère pour l'appariement. Nous ne gardons qu'un certain pourcentage d'appariement (50 %). C'est-à-dire, nous ne gardons que les points appariés avec une faible distance euclidienne et des vecteurs normaux ayant des directions proches. De plus, dans le calcul de l'erreur de distance, nous pondérons les paires de points en utilisant des poids inversement proportionnels à la distance entre les 2 points appariés (voir équation III.1.5). Pour pouvoir trouver la matrice de rotation et de translation, nous utilisons une approximation linéaire

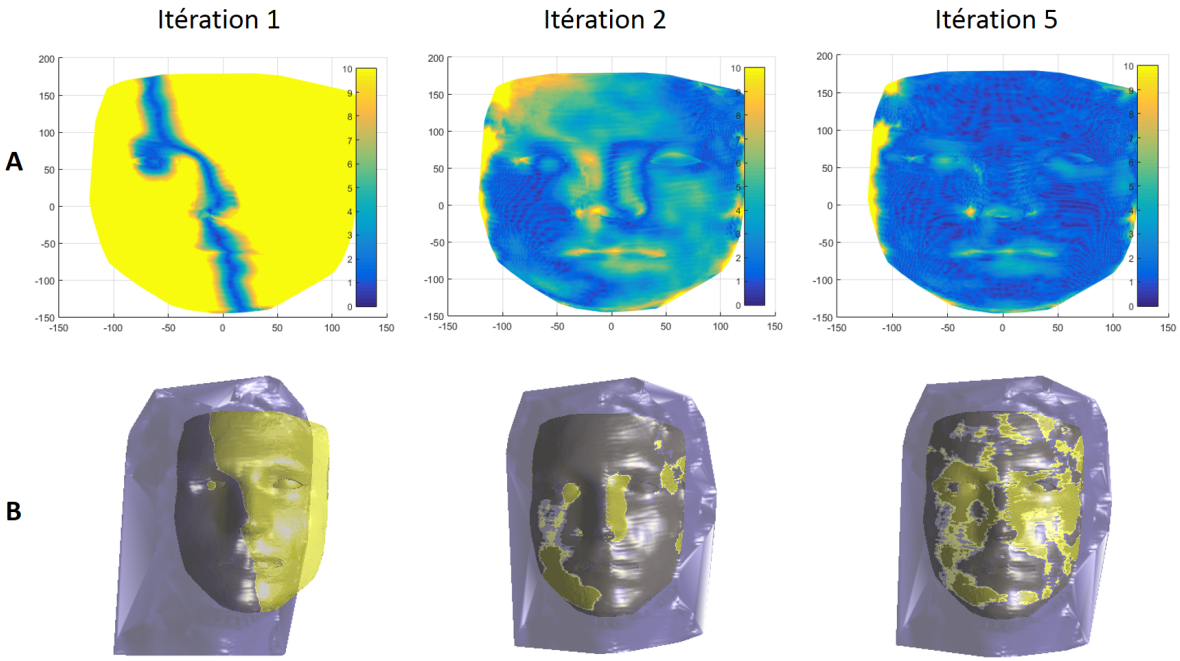


FIGURE III.1.9: Alignement de 2 maillages avec l'algorithme ICP. A) Erreur entre les 2 maillages, B) Alignement entre les 2 maillages au cours des itérations.

et une décomposition en valeur singulière.

$$E_{ICP}(\hat{R},\hat{t}) = argmin \sum ||W_p.((Ds_p - (\hat{R} * Ms_p + \hat{t})).n_p)||^2, Ds_p \subset D_p, Ms_p \subset M_p \quad \text{(III.1.5)}$$

avec

- $\hat{R}$: matrice de rotation estimé
- $\hat{t}$: matrice de translation estimée
- $E_{ICP}(\hat{R},\hat{t})$: erreur à minimiser
- n_p : vecteurs normaux des points de D_p
- Ds_p : Sous-ensemble de points de le trame de profondeur D_p appariés
- Ms_p : Sous-ensemble de points du maillages M_p appariés
- W_p : poids calculés en fonction de la distance entre les paires de points

E_{ICP} est la somme de la distance au carré entre chaque point du maillage (Ms_p) et le plan tangent de son point de destination correspondant (Ds_p). Dans cette équation, $\hat{R}$ et $\hat{t}$ sont les matrices de transformation estimées et n_p est les vecteurs normaux à Ds_p. Les poids W_p sont calculés à partir de la distance entre Ds_p et Ms_p. Les paires de points avec une courte distance sont les plus importants, de sorte que les poids W_p sont inversement proportionnels à la distance entre les points appariés. La figure III.1.9 montre un exemple d'alignement rigide.

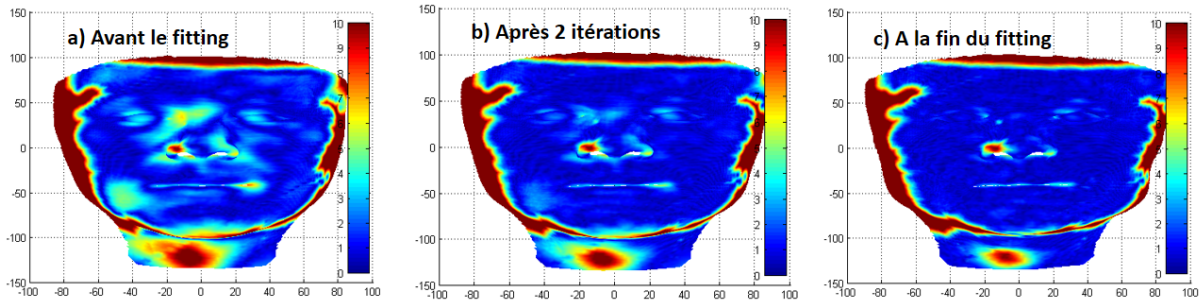


FIGURE III.1.10: Exemple de *fitting* : a) Erreur de distance entre la trame de profondeur et le maillage moyen du modèle déformable de visage avant l'étape de *fitting*. b) Erreur de distance après 2 itérations du *fitting*. c) Erreur de distance à la fin de l'étape de *fitting*.

III.1.1.3 Transformation non rigide

La transformation non rigide permet de *fitter* le maillage moyen du modèle déformable sur un autre maillage. Contrairement à une transformation rigide, tous les points du maillage ne subissent pas la même transformation (voir figure III.1.10). Les modèles déformables permettent de réaliser une transformation de ce type. Il existe plusieurs modèles déformables de visage dans la littérature. Nous avons présenté plusieurs méthodes qui utilisent des modèles déformables dans la partie II.2.1.2 de l'état de l'art de ce manuscrit. Un modèle déformable de visage 3D est créé à partir de plusieurs scans de visage. Les maillages doivent être de haute résolution et sans bruit. C'est pourquoi, il faut utiliser des scanners 3D haute résolution. Ce type de scanner est présenté dans la section II.1. Les personnes scannées doivent avoir un visage neutre et des caractéristiques physiques variées pour que le modèle soit le plus riche possible. Le plus souvent, les personnes numérisées sont de groupe ethnique, de sexe et d'âge différents. De plus, les maillages doivent posséder le même nombre de points et avoir une correspondance point à point entre les sommets des faces (maillages 3D sémantiques). Ensuite une Analyse en Composante Principale (ACP) est réalisée sur ces données. À partir de ce modèle, nous pouvons obtenir un nouveau visage à partir d'une combinaison linéaire des visages de la base de données. Il faut noter qu'il est difficile de réaliser un modèle déformable de visage 3D. C'est pourquoi, nous ne trouvons pas énormément de modèles de bonne qualité dans la littérature. Blanz et Vetter [98] expliquent dans leur article comment ils ont construit leur modèle déformable (voir section II.2.1.2). L'équation ci-dessous III.1.6 décrit le modèle déformable.

$$M(\alpha) = \mu + U * diag(\sigma)\alpha \tag{III.1.6}$$

- $M(\alpha)$: Modèle paramétré par les coefficients α
- μ : visage moyen du modèle
- U : Base orthonormale de l'ACP
- σ : l'écart type de l'ACP
- α : vecteur des coefficients du modèle

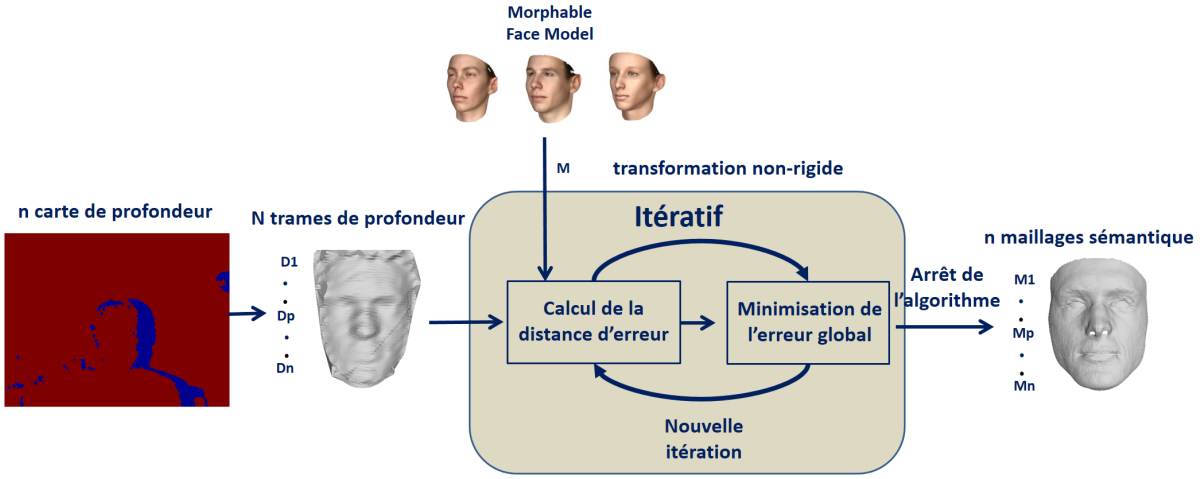


FIGURE III.1.11: Schéma de notre méthode de transformation non rigide. Tout d'abord, nous calculons l'erreur de distance entre la trame de profondeur traitée D_p et le maillage M du modèle déformable. Puis nous minimisons cette erreur de distance en optimisant le vecteur de paramètres α du modèle. Nous obtenons en sortie le maillage sémantique M_p correspondant à la trame de profondeur D_p.

Le vecteur de paramètre α permet de déformer le maillage 3D du modèle. En effet, ce vecteur permet de se déplacer dans l'espace formé par les vecteurs propres de l'analyse en composante principale (voir équation III.1.6). Quand ce vecteur de coefficient est nul, le maillage 3D obtenu est le visage moyen du modèle déformable.

Notre méthode de transformation non rigide est aussi itérative. Dans cette étape, nous utilisons le modèle déformable de visage 3D MorphFace de Paysan et al [6] (voir section IV.1.3). Il a été créé à partir de 200 visages (100 femmes et 100 hommes) numérisés avec un scanner laser (Cyberware TM). Après avoir aligné la trame de profondeur avec le maillage 3D, nous déformons le maillage pour qu'il s'adapte le mieux possible à la trame. Cette transformation est dite non rigide. En effet, le maillage ne garde pas la même forme après la transformation. La transformation non rigide est composé de 3 étapes : le calcul de la distance d'erreur, la minimisation de l'erreur globale et le critère d'arrêt. La figure III.1.11 décrit ces différentes étapes. Dans la première étape, il faut calculer l'erreur de distance entre la trame de profondeur traitée et le maillage 3D du modèle déformable. Nous n'utilisons pas tous les points des 2 maillages. En effet, chaque trame de profondeur fournies par la caméra Kinect ne contient pas que des informations de profondeur correctes (bord, courbure importante...). C'est pourquoi, nous éliminons les points de la trame dont la direction des vecteurs normaux est éloignée de la direction de l'axe optique de la caméra (seuil prédéfini). Nous ne considérons que les points qui n'ont pas ce type de normale comme du bruit. De plus, nous éliminons les points de la trame qui sont isolés et qui sont en bordure du maillage. En effet, ces points ont plus de chances d'être du bruit que d'être des points corrects du visage. Ensuite, nous utilisons plusieurs critères pour apparier les points de la trame et du maillage du modèle. Pour que 2 points soient appariés, il

faut que leurs distances euclidiennes soient plus petites qu'un certain seuil et que leurs vecteurs normaux aient une direction quasiment similaire. L'appariement est l'étape fondamentale du *fitting*. En effet, si cette étape n'est pas correctement effectuée, le résultat final ne sera pas correct. À partir de ces appariements, nous obtenons une erreur de distance E_f qui est la somme au carré des distances euclidiennes entre les différentes paires de points.

$$E_f = argmin_E \sum ||W_p . * (M_p(\alpha) - D_p)||^2 \tag{III.1.7}$$

avec

- E_f : Distance d'erreur
- W_p : pondération en fonction de la distance entre les paires de points et de la direction des vecteurs normaux
- $M_p(\alpha)$: maillage 3D du modèle déformable
- D_p : trame de profondeur traitée

Nous utilisons une optimisation de Gauss Newton pour minimiser cette erreur. La modification du vecteur de paramètres α du modèle permet de déformer le maillage 3D du modèle et donc de modifier l'erreur de distance E_f. Après avoir effectué l'étape de *fitting* sur les *n* trames de profondeur D_p, nous obtenons *n* maillages 3D sémantiques M_p (voir figure III.1.12). Cette étape a permis d'éliminer le bruit et d'augmenter la résolution des trames de profondeur.

III.1.2 Détection des patchs de forme

Dans cette section, nous détectons les patchs de forme des différents maillages 3D obtenus dans la section précédente. En effet, nous voulons détecter les zones des données de profondeur (forme) qui ne contiennent pas de bruit et sont adéquates. La figure III.1.13 montre que les trames de profondeur ne sont pas correctes au niveau des yeux et que les lèvres sont lissées et donc peu marquées. Le principal avantage de l'utilisation d'un modèle déformable est qu'il fournit des maillages de résolution élevés et sémantiques. De plus, les maillages ne sont pas bruités. Contrairement aux capteurs RVB-Z qui fournissent des données avec très peu de détails sur certaines parties du visage, ce type de modèle permet d'obtenir des visages 3D avec des traits marqués au niveau des yeux et de la bouche. C'est pourquoi, les maillages fournis par le modèle déformable sont beaucoup plus précis. Par contre, le principal inconvénient de l'utilisation d'un modèle déformable est sa dépendance à sa base de données. C'est-à-dire qu'il est difficile d'obtenir un résultat avec toutes les spécificités du visage que l'on veut cloner en n'effectuant qu'un seul *fitting*. En effet, même si toutes les spécificités se trouvent dans la base de données, le modèle déformable ne les retrouve pas forcément toutes en n'effectuant qu'un seul *fitting*. C'est pourquoi, nous effectuons d'abord le *fitting* sur chaque trame avant de faire une fusion des données. En effet, chaque spécificité du visage se trouve dans plusieurs trames de profondeur et donc peut être retrouvée plus facilement par le modèle déformable. Dans cette étape, nous voulons détecter les parties des maillages M_p et les parties des trames de profondeur D_p qui

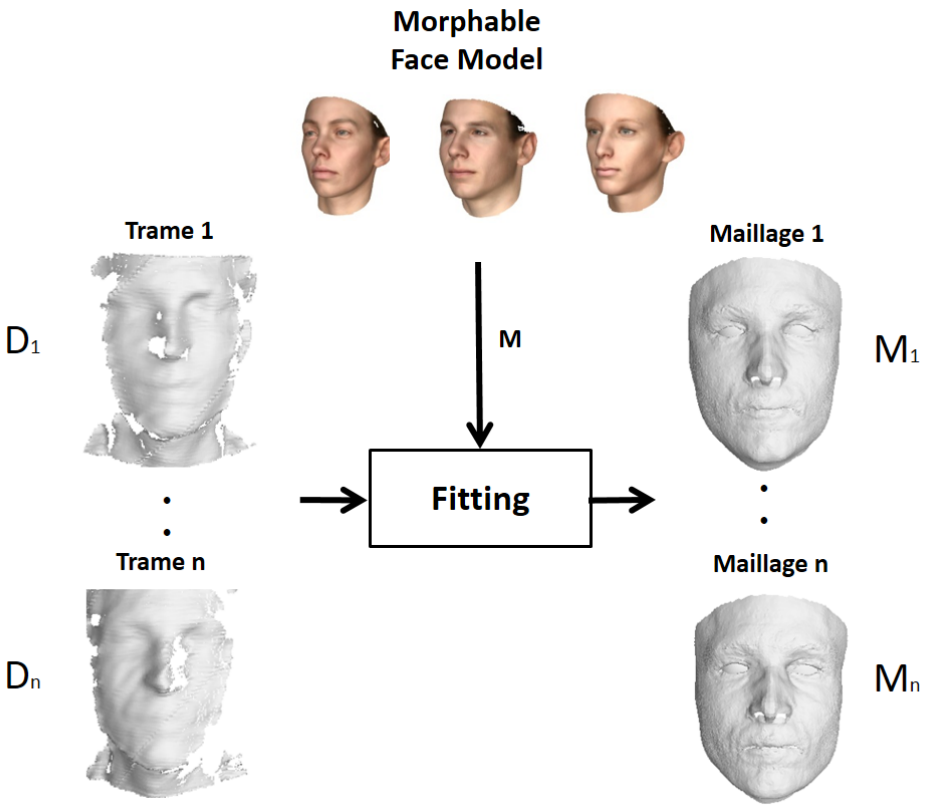


FIGURE III.1.12: Cette figure présente le schéma de l'étape de *fitting*. Les *n* trames de profondeur D_p sont les données d'entrée. Après avoir réalisé le *fitting*, nous obtenons *n* maillages sémantiques M_p.

sont les plus précises, sans erreur et qui contiennent le plus de spécificités. Nous appelons ces parties des différents maillages détectés, des patchs. Ce sont des ensembles de points 3D pré-sélectionnés pour reconstruire le maillage de point 3D du visage. Dans cette partie, nous décrivons nos critères de sélection et les raisons de l'utilisation de ces critères. La figure III.1.13 décrit les différentes sections de cette partie de détection de patchs de forme.

L'étape précédente III.1.1, nous a permis de créer *n* maillages correspondant aux trames de profondeur fournies par le capteur RVB-Z (voir figure III.1.12). Chaque trame de profondeur ne contient pas toute l'information du visage (voir figure III.1.13 : (a1 et b1)). En effet, une trame de profondeur de profil droit ne contient pas l'information du profil gauche. C'est pourquoi, nous voulons garder le patch du maillage qui correspond au bon profil. De plus, certaines parties des trames de profondeur ne sont pas captées correctement par le capteur. Nous voulons éliminer ces parties peu précises. Pour finir, l'étape de *fitting* ne permet pas d'obtenir toutes les spécificités du visage. Ces zones doivent aussi être détectées et supprimées. Pour détecter les patchs de forme correctement, nous utilisons 2 critères que nous décrivons ci-dessous.

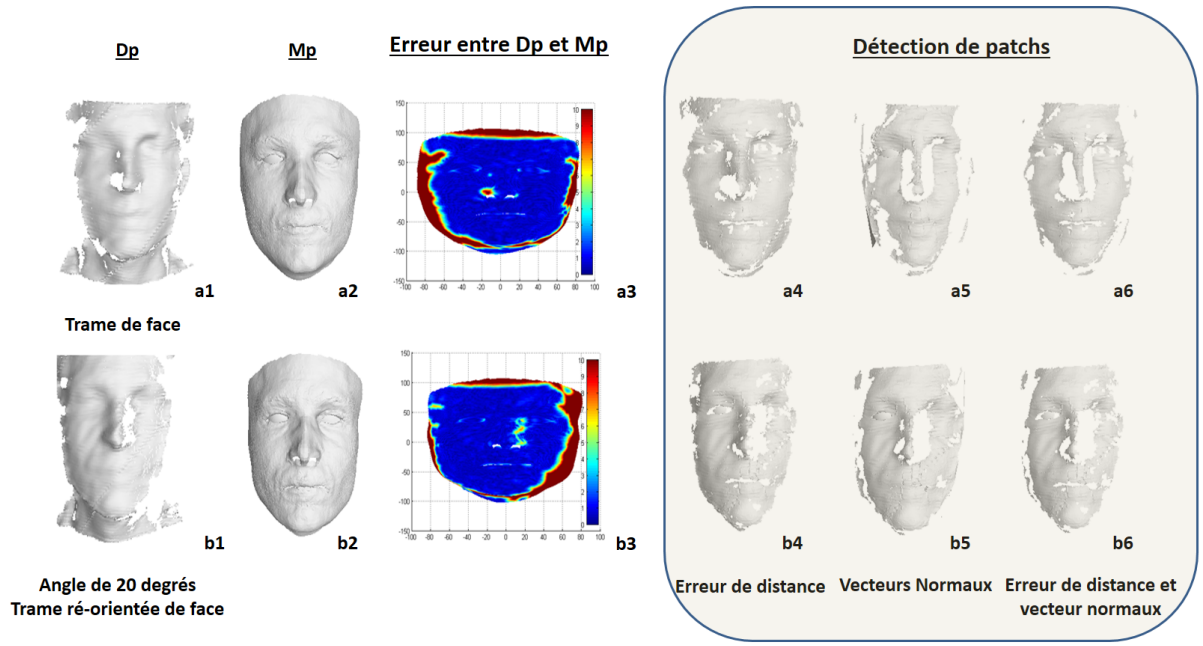


FIGURE III.1.13: Exemples de détection des patchs : a1 : une trame de face. b1 : une trame de profil droit (le visage est re-orienté de face). Nous utilisons l'algorithme de *fitting* pour calculer les 2 maillages sémantiques (a2 et b2). L'erreur entre la trame et le maillage (a3 et b3) donne les patchs qui sont adéquats. Nous sélectionnons les parties du maillage (patch) où l'erreur est petite (a4 et b4) et où les vecteurs normaux sont parallèles à l'axe de la caméra (a5 et b5). Dans notre processus, nous utilisons ces 2 critères (a6 et b6).

III.1.2.1 Erreur de distance

Le premier critère que nous utilisons est un critère de distance. Après avoir réalisé le *fitting* sur une trame de profondeur D_p, nous obtenons un maillage M_p. Ce critère est basé sur la distance d'erreur point à point entre la trame D_p et le maillage M_p correspondant. C'est-à-dire que nous apparions chaque point de la trame avec le point le plus proche du maillage. Cet appariement a déjà été réalisé et est détaillée dans l'étape du calcul de l'erreur E_f de la transformation non rigide du *fitting* dans l'équation III.1.7. Nous utilisons l'erreur entre les différents points appariés pour détecter les patchs de forme. Cette erreur de distance entre 2 points doit être inférieure à un certain seuil (1 mm) pour que le point du maillage M_p soit conservé. L'utilisation d'une erreur de distance permet d'éliminer le bruit du capteur et les erreurs de *fitting*. Sur la vue de face de la figure III.1.13 : (a4), il y a un trou au niveau du nez car le capteur ne donne pas cette information. Le maillage M_p (figure III.1.13 : (a2)), calculé par le *fitting*, ne possède pas un tel trou. Les données de maillage M_p ne reflètent donc pas l'identité du sujet pour cette partie du visage. L'erreur de distance supprime ces informations (figure III.1.13 : (a4)). De même, pour le côté gauche du nez de la trame III.1.13 : (b1)).

III.1.2.2 Vecteurs normaux

Le deuxième critère est basé sur la direction des vecteurs normaux des points des trames de profondeur. En effet, une caméra RVB-Z capture plus précisément les zones où l'axe optique est perpendiculaire à la surface de l'objet. C'est-à-dire que le vecteur normal d'un point de la trame de profondeur doit être plus ou moins parallèle à l'axe optique. Nous utilisons un seuil d'angle de détection pour éliminer les points avec une normale dont la direction est éloignée de celle de l'axe optique. Dans la figure III.1.13 : (a5), nous constatons que ce critère de sélection élimine les points que la caméra ne saisit pas correctement. Par exemple, nous notons que pour une trame de profondeur de face III.1.13 : (a2 et a4), les points situés sur les côtés du nez sont mal capturés par la caméra et ne sont pas précis.

III.1.2.3 Critères utilisés dans notre méthode

Dans notre méthode, nous utilisons les 2 critères détaillés ci-dessus. Pour qu'un point soit conservé, il doit avoir un vecteur normal parallèle à l'axe optique de la caméra et la distance entre le maillage M_p et la trame D_p doit être inférieure à un seuil. Nous tolérons un angle de plus ou moins vingt degrés pour les vecteurs normaux. Cela nous permet de détecter les zones qui ne sont pas capturées correctement par la caméra et d'éliminer le bruit et les erreurs de *fitting*. La figure III.1.13 : (a6, b6) montre que seuls les points appropriés sont conservés. L'étape de détection de patchs est très importante dans notre méthode. En effet, elle permet d'éliminer les erreurs qui ont eu lieu dans l'étape de *fitting*. De plus, elle permet aussi de récupérer l'information pertinente qui contient les spécificités du visage. Si cette étape ne fonctionne pas correctement, les résultats sont très fortement dégradés.

III.1.3 Fusion des patchs de forme

Dans l'étape de détection des patchs, nous avons détecté les patchs de forme de chaque maillage M_p (un par trame D_p). Pour pouvoir reconstruire la forme complète du visage, nous devons fusionner ces patchs (voir figure III.1.14). Les maillages M_p que nous avons obtenus avec le modèle déformable sont sémantiques. C'est pourquoi, nous connaissons la position exacte de chaque patch sur le visage (yeux, front ...). Cette information est très importante et nous permet de fusionner les différents patchs en fusionnant chaque point des maillages sémantiques. Nous pouvons faire varier dans notre méthode le type de fusion (section III.1.3.1) et le type de donnée des patchs (section III.1.3.2). Nous avons testé quatre types de fusion dans notre méthode que nous présentons ci-dessous : la moyenne, la moyenne pondérée, la médiane et la moyenne robuste (section III.1.3.1). De plus, nous avons utilisé plusieurs types de patchs. C'est-à-dire les patchs composés par les données des maillages M_p, mais aussi les patchs composés par les

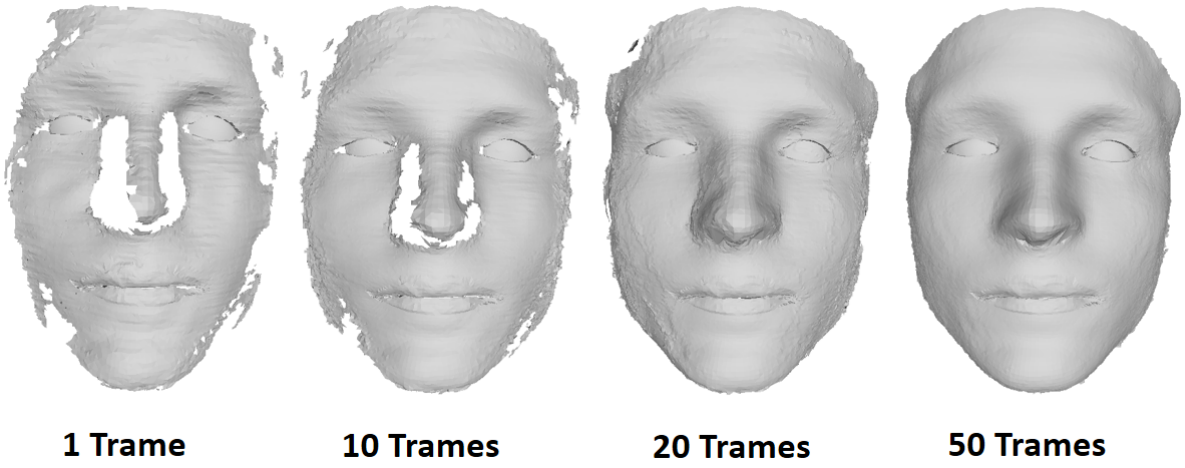


FIGURE III.1.14: La figure présente les résultats obtenus avec notre méthode de clonage à différentes itérations de notre algorithme. Le traitement d'une seule trame ne permet pas de reconstruire la forme du visage entièrement. En effet, certaines informations du visage ne sont pas présentes dans cette première trame (profil,...). L'utilisateur doit effectuer des mouvements de rotation de la tête pour que toutes les informations du visage soient captées par la caméra. On peut voir sur la figure III.1.14, qu'au bout de 50 trames, le visage est reconstruit intégralement.

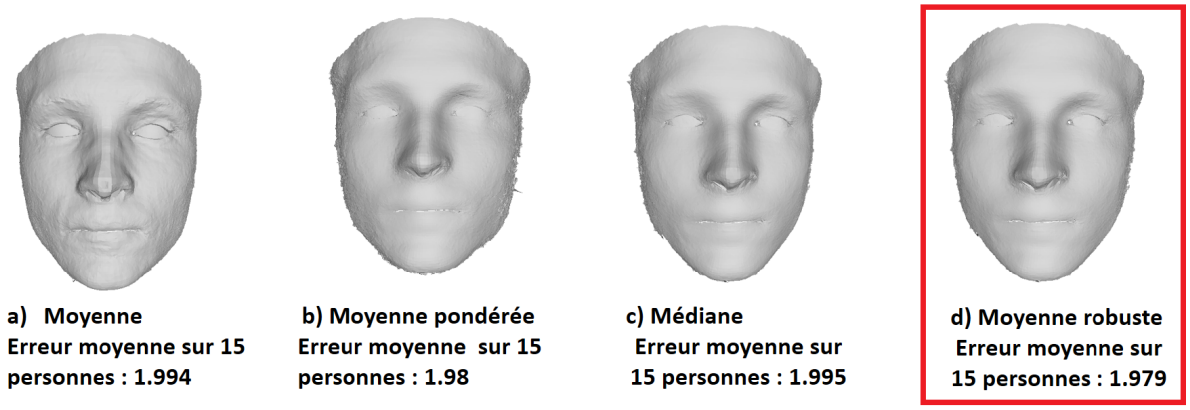


FIGURE III.1.15: Comparaison qualitative des différents types de fusion. Le clone a) est obtenu avec la fusion par moyenne. Le clone b) avec la fusion par moyenne pondérée et le clone c) avec la fusion par médiane. Nous obtenons les meilleurs résultats avec la fusion par moyenne robuste (clone d). En effet, nous avons comparé ces différents résultats avec une vérité terrain. Nous avons calculé une erreur de distance point à point (moyenne : 1.994, moyenne pondérée : 1.98, médiane : 1.995, moyenne robuste : 1.979, voir section IV.2.3).

données des trames de profondeur correspondantes (section III.1.3.2). La figure III.1.15 montre des exemples de fusion de patchs. Nous présentons ses différentes techniques dans cette section.

III.1.3.1 Les quatre types de fusion

Dans cette section, nous voulons fusionner les patchs pour obtenir un clone complet du visage. Mais certains patchs possèdent la même information du visage. C'est pourquoi il faut fusionner

ces informations. Nous connaissons la sémantique des patchs, donc nous savons exactement sur quelle partie du clone doit se positionner chaque point des différents patchs de forme.

Pour le premier type de fusion, nous moyennons simplement les coordonnées des points qui se superposent sur le maillage du clone.

Pour la fusion avec une moyenne pondérée, nous ajoutons des poids qui permettent de donner plus d'importance aux points de patch qui ont une probabilité plus grande d'être correcte. Les poids sont calculés en fonction du critère de distance. Plus la distance entre le point d'un patch P_p et le point apparié de la trame correspondante D_p est grande et plus le poids sera petit. En effet, un point d'un patch avec une distance petite aura plus de chance d'être correct.

Pour la fusion avec la médiane, nous sélectionnons le point médian. Si pour un point du clone, il n'y a que 2 points possibles, nous effectuons une moyenne des coordonnées.

Pour le quatrième type de fusion, nous réalisons une moyenne robuste. C'est-à-dire que nous éliminons les points qui sont éloignés du point médian et nous effectuons une moyenne pondérée des autres points. Nous utilisons le même type de poids que pour le deuxième type de fusion.

La figure III.1.15 montre les résultats obtenus pour les 4 types de fusion.

Dans notre méthode finale, nous avons utilisé la méthode de fusion avec la moyenne robuste. En effet, nous pouvons voir sur la figure III.1.15 que les résultats sont meilleurs avec ce type de fusion. Nous détaillons les caractéristiques des résultats obtenus avec les 4 types de fusion dans la partie IV.

III.1.3.2 Les trois types de données des patchs de forme

Les maillages M_p sont précis au niveau des yeux et de la bouche, mais ne contiennent pas forcément toutes les spécificités du visage. À l'inverse, les trames de profondeur D_p ne sont pas précises au niveau des yeux et de la bouche (pas de globe oculaire...) mais contiennent plus d'informations sur les spécificités du visage. C'est pourquoi, nous avons testé aussi plusieurs types de données pour les patchs de forme dans notre méthode. Dans la section III.1.2, nous avons détecté les patchs P_p des différents maillages M_p. Chacun des patchs détectés est composé des données (point 3D) des maillages M_p obtenus à partir du *fitting*. Chacun de ces patchs contient donc une partie de l'information du visage. Pour chaque point de ces patchs P_p, nous connaissons le point de la trame de profondeur D_p apparié. Dans notre deuxième type de patchs, nous avons remplacé les points des patchs correspondant aux données des maillages, par leurs points appariés. Dans notre troisième type de patch, nous avons effectué un mixte des 2 types précédent ; c'est-à-dire, nous avons utilisé des patchs composés des données des maillages M_p pour certaines parties du clone et le deuxième type de patchs pour les autres parties. La figure III.1.16 montre les résultats obtenue à partir de ces 3 techniques différentes.

Données des maillages (M_1 à M_N, voir figure III.1.16 : c) : nous utilisons les données des patchs de profondeur des différents maillages M_p. Ce type de données fournit un clone réaliste

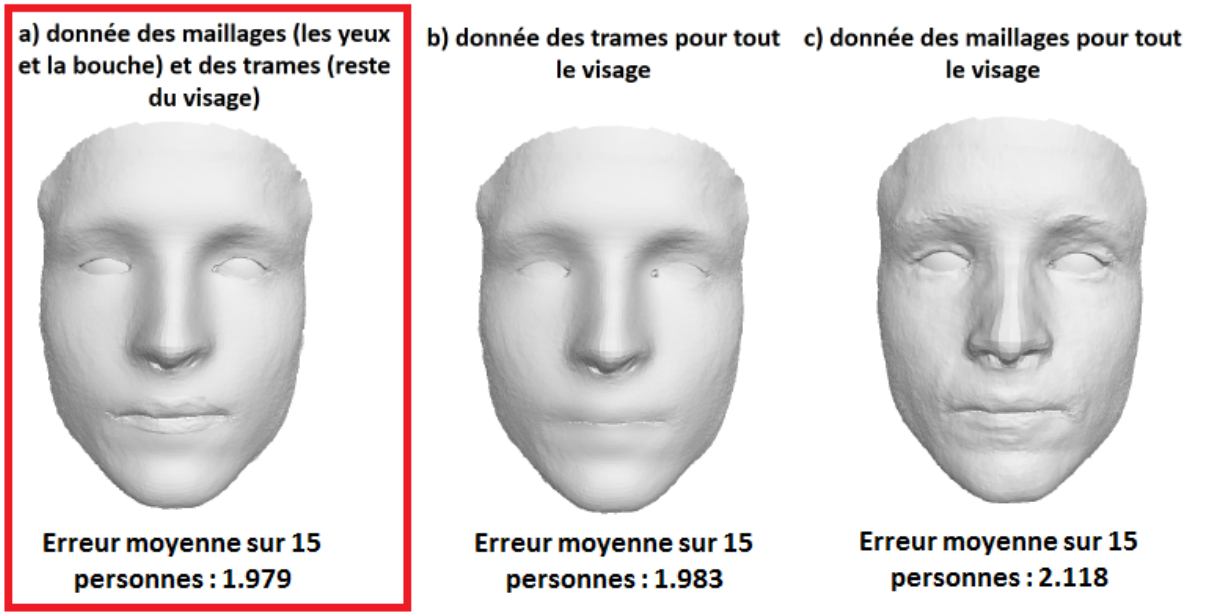


FIGURE III.1.16: Différents clones obtenus avec les trois types de patchs de forme. Pour le clone a), nous utilisons des patchs composés des informations des maillages M_p pour les yeux et la bouche et les informations des trames D_p pour le reste du visage. Pour le clone b), nous utilisons des patchs composés des données des trames de profondeur D_p. Enfin, pour le clone c), nous utilisons des patchs composés des informations des maillages M_p pour tout le visage. La distance d'erreur avec la vérité terrain est la plus petite pour le clone a) (Distance d'erreur : 1.979).

de la personne. L'avantage d'utiliser l'information des maillages M_p est que le clone obtenu a les traits du visage marqués au niveau des yeux et des lèvres. Mais certaines spécificités peuvent ne pas apparaître dans le clone si elles n'ont pas été apprises lors de la construction du modèle déformable de visage 3D.

Données des trames (D_1 à D_n, voir figure III.1.16 : b) : nous utilisons les données des trames D_p pour reconstruire le clone sémantique M_C. Pour chaque point des patchs P_p, nous connaissons le point de la trame correspondant D_p le plus proche. Nous remplaçons les valeurs des points des patchs par les valeurs des points des trames les plus proches. Avec ce type de donnée, nous obtenons un clone sémantique de haute résolution qui contient de nombreuses spécificités de l'individu. Mais les données acquises par le capteur RVB-Z ne sont pas précises et sont très bruitées au niveau des yeux et de la bouche.

Dans notre processus (voir figure III.1.16 : a), nous utilisons les 2 types de données décrites ci-dessus. Plus précisément, nous utilisons les points des maillages M_p pour les yeux et la bouche et les points des trames D_P pour le reste du visage. L'utilisation des données des trames fournit un clone avec de nombreuses spécificités de la personne et l'utilisation des données des maillages donne un clone réaliste au niveau des yeux et de la bouche. Pour chaque point du clone, il peut y avoir plusieurs patchs qui se chevauchent (comme le front du visage dans la figure III.1.13). C'est pourquoi, nous faisons une fusion de chaque point de ces patchs. Nous effectuons la moyenne pondérée robuste sur les points qui se chevauchent (cf section III.1.3.1). Ainsi, nous ne prenons

pas en compte les valeurs aberrantes dans le calcul de la moyenne. La pondération est calculée à partir de la distance d'erreur point à point calculée au paragraphe III.1.2. Nous éliminons les points qui sont loin de la valeur médiane avec un seuil (2 mm). Cette méthode permet d'obtenir les meilleurs résultats qualitatifs et quantitatifs. Dans la partie IV, nous présentons et comparons les différents résultats obtenus avec ses différentes types de fusion et de données.

III.1.4 Conclusion

Dans cette partie, nous avons décrit une méthode qui permet de reconstruire la forme 3D d'un visage à partir de plusieurs trames de profondeur fournies par un capteur RVB-Z. Nous utilisons une technique qui utilise un modèle déformable de visage 3D. Les 2 particularités de notre méthode sont l'inversion du système et l'utilisation de patchs de forme. Nous réalisons en premier le *fitting* sur chaque trame de profondeur, puis la fusion (voir figure III.1.1). Notre système est moins dépendant des alignements et des erreurs de *fitting* parce que nous fusionnons a posteriori des informations fiables (les patchs). C'est pourquoi, notre méthode est mieux adaptée aux caractéristiques d'un visage inconnu de la base de données. L'utilisation de patchs de forme permet de mettre l'accent sur les spécificités du visage. En effet, notre technique élimine les parties des trames qui ne contiennent pas d'informations adéquates du visage. Notre méthode permet donc de retrouver plus facilement les spécificités de la personne.

Chapitre *III.2*

Reconstruction de la texture

Dans cette deuxième étape (figure III.2.1), nous expliquons notre méthode de reconstruction de la texture du visage. Dans la première section, nous avons reconstruit la forme 3D du visage. Pour que le résultat soit plus réaliste, nous voulons *mapper* une texture sur le maillage 3D obtenu. Notre méthode est composée de 4 étapes itératives : l'étape d'alignement de chaque trame et du clone est décrite au paragraphe III.2.1, la détection des patchs de texture au paragraphe III.2.2, le *warping* des patchs au paragraphe III.2.3 et la fusion des patchs de texture au paragraphe III.2.4. Pour reconstruire cette texture, nous utilisons les informations de couleur et de profondeur du capteur RVB-Z. Nous utilisons une résolution de 1280*960 pour les trames de texture I_p sachant que la résolution en profondeur est bien inférieure (640*480) (voir figure III.2.2). Pour obtenir la carte de texture du clone, nous déplions le maillage 3D sur un cylindre. Nous appelons chacun des points dépliés, les points d'ancrage de la carte de texture du clone 3D. La particularité de notre méthode est que nous utilisons la forme des trames de profondeur et du clone 3D pour *warper* la texture des trames sur la carte de texture du clone.

Sommaire

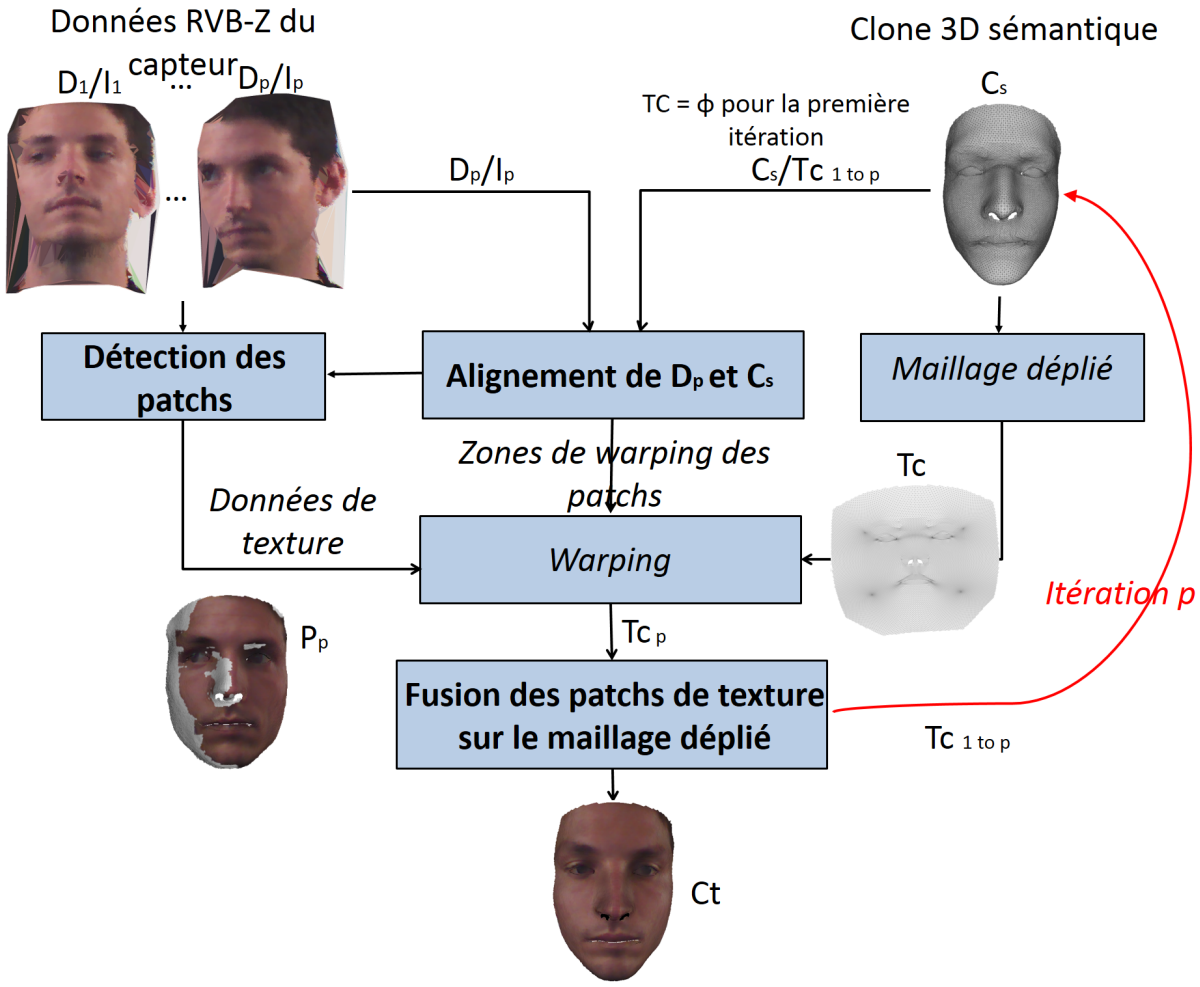


FIGURE III.2.1: Schéma global de notre méthode de reconstruction de texture. Elle est composée de 4 étapes. Tout d'abord, nous alignons la trame de profondeur D_p traitée avec le clone 3D sémantique C_s. Ensuite, nous détectons le patch de texture correspondant P_p. Nous effectuons ces 2 étapes pour chaque trame traitée. Après avoir déplié le clone 3D C_s et obtenu la carte de texture Tc, nous *warpons* les patchs de texture détectés P_p sur cette carte. Pour finir, nous fusionnons les zones des patchs P_p de texture qui se chevauchent.

III.2.1 Alignement rigide des trames de profondeur et du clone sémantique

La première étape de notre méthode pour pouvoir reconstruire la texture du visage 3D est d'aligner les trames de profondeur D_p et le clone C_s. Dans notre méthode, il faut que l'alignement entre le clone 3D C_s et les trames de profondeur D_p soit le plus précis possible. En effet, un mauvais alignement provoque un décalage sur le positionnement de la texture sur le clone. Pour cela, nous utilisons l'algorithme ICP [70] détaillé dans la partie III.1.1. Cet algorithme est lent et nécessite de nombreuses itérations pour converger. Pour réduire le nombre d'itérations de notre méthode, nous initialisons l'angle de rotation avec la pose calculée par le processus Intraface [102]. Dans notre méthode, nous utilisons l'ICP point plan [109]. C'est-à-dire que nous calculons

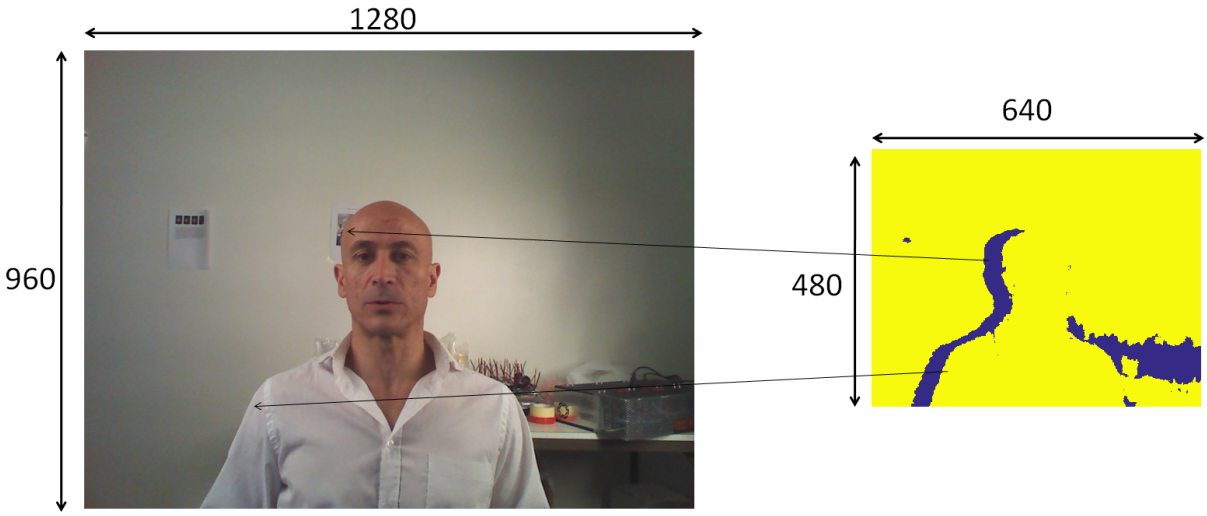


FIGURE III.2.2: La caméra RVB-Z permet d'obtenir des images RVB de résolution 1280*960 et une carte de profondeur de résolution 640*480 (après interpolation). Nous connaissons la correspondance entre la carte de texture et la carte de profondeur de la caméra RVB-Z.

la somme des distances au carré entre chaque point du clone et le plan tangent de son point de destination correspondant (D_p). L'algorithme ICP classique permet d'obtenir des résultats précis quand il faut aligner une trame de face D_p avec le clone C_s. En effet, il y a une grande surface de recouvrement entre une trame vue de face et le maillage 3D du clone. Voilà pourquoi, la première trame I_1 que nous traitons dans notre méthode est une trame vue de face. En revanche, il est plus difficile de correctement aligner une trame de profil. Il y a beaucoup moins de recouvrement et les angles de rotation sont plus importants. De plus, IntraFace [102] est moins performant quand l'angle de rotation est important.

Pour augmenter les performances de l'algorithme ICP classique, notamment pour l'alignement des trames qui ne sont pas de face, nous avons ajouté un critère de couleur. En effet, dans la partie II, nous présentons des méthodes qui montrent que l'utilisation de la couleur dans l'ICP permet d'augmenter les performances de l'algorithme. Il est donc possible d'utiliser la couleur pour améliorer l'appariement des points qui est l'étape la plus importante dans l'algorithme ICP. Douadi et al [104] ont testé 2 types de méthode avec le critère de couleur. La première méthode consiste à utiliser une distance mixte, au lieu d'une distance géométrique, pour apparier les points. C'est-à-dire que la mise en correspondance des points se fait à partir de la distance euclidienne et de la distance couleur entre les 2 nuages de points. Cette méthode permet d'améliorer l'alignement entre 2 nuages 3D. La deuxième méthode consiste à utiliser la couleur et la distance séparément. C'est-à-dire que l'appariement est effectué selon la distance géométrique entre les points des nuages. La distance couleur sert à éliminer les mauvais appariements. Si la distance couleur entre 2 points appariés est plus grande qu'un certain seuil prédéfini, alors le couple de point est rejeté. Douadi et al [104] montrent que cette deuxième méthode est la plus performante. Pour que ces 2 méthodes soient efficaces, il faut que la texture soit de bonne

qualité et que la variation de la luminosité ne soit pas importante. Pour limiter l'impact de la luminosité, il est possible de travailler dans l'espace couleur YIQ. La composante Y correspond à la luminosité et les composantes I et Q correspondent à la chrominance. La distance couleur est alors calculée à partir des 2 composantes de chrominance (voir équation III.2.1). Ce changement d'espace de représentation de la couleur permet d'améliorer la robustesse de l'algorithme. Dans notre méthode, nous rejetons les paires de points avec une distance de couleur au-dessus d'un certain seuil.

$$D_{couleur}(q_i, p_i) = \sqrt{(I_{q_i} - I_{p_i})^2 + (Q_{q_i} - Q_{p_i})^2} \tag{III.2.1}$$

avec

- $D_{couleur}$: Distance couleur entre 2 points appariés.
- I_{q_i} et Q_{q_i} : composantes de chrominance du point apparié q_i du nuage q.
- I_{p_i} et Q_{p_i} : composantes de chrominance du point apparié p_1 du nuage p.

Dans notre technique, chaque trame de profondeur D_p fournie par le capteur RVB-Z est associée à une trame de couleur I_p. Les trames sont traitées les unes après les autres et permettent de compléter la carte de texture du clone. Il faut noter que lors du traitement de la première trame, nous ne disposons d'aucune information de texture. C'est pourquoi, nous ne pouvons pas utiliser le critère de couleur au début de notre méthode et que la première trame de profondeur traitée par notre technique est une trame de face. Nous utilisons IntraFace [102] pour détecter l'orientation de la pose de la tête à partir des cartes de texture fournies par le capteur RVB-Z. Comme l'alignement est plus performant avec ce type de trame, nous savons que le traitement de cette trame va être efficace. Une fois que la première trame a été traitée, la texture du clone commence à être complétée (voir section III.2.4). Ensuite, nous utilisons cette texture dans l'algorithme ICP pour aligner les autres trames. En effet, pour la seconde trame de profondeur D_p, qui n'est pas forcément de face, nous utilisons le critère de couleur pour rejeter les mauvais appariements. Chaque point du clone est apparié avec un point de la trame de profondeur. Nous connaissons la sémantique du maillage du clone, donc nous savons quels points ont été texturés par le traitement de la première trame. Nous rejetons les appariements des points du clone texturé qui ont une distance couleur (voir l'équation III.2.1) plus grande qu'un seuil prédéfini (seuil de 3 dans notre méthode). Cette technique permet d'améliorer l'alignement des trames D_p avec le clone C_s et donc la reconstruction de la texture 3D. Dans notre méthode, nous n'utilisons la texture que de la première trame de face pour certaines parties du visage. En effet, l'alignement entre la première trame de face et le clone est très précis. C'est pourquoi, nous ne *warpons* que la texture de cette trame sur les trois parties suivantes du visage : la bouche, les yeux et les sourcils. De plus, les trames suivantes fournies par le capteur peuvent être bruitées sur ces trois zones. C'est pourquoi, le *warping* de la texture de la première trame permet d'éviter du bruit dans la texture au niveau de ces trois zones. Les trames suivantes sont utilisées pour compléter la texture du reste du visage. En outre, au cours de l'acquisition (rotation de la tête), la pupille change de position. Elle n'est donc jamais située au même endroit dans les différentes trames. Notre

méthode permet d'éviter l'apparition de plusieurs pupilles dans chaque œil. Elle permet aussi d'atténuer les différences de luminosité entre les trames. Notre méthode permet donc d'améliorer la qualité de la texture du visage.

III.2.2 Détection des patchs de texture

Dans cette deuxième section, nous voulons détecter sur chaque trame de texture I_p, les parties de la texture qui sont correctes et précises. Comme pour la reconstruction de la forme, nous appelons ces zones, des patchs. La caméra capture mal la texture de certains endroits des trames. En effet, les contours et les zones du visage où les vecteurs normaux ne sont pas parallèles à l'axe optique de la caméra ne sont pas correctement restitués par la caméra RVB-Z. C'est pourquoi, nous repérons les points de chaque trame de profondeur proche du clone obtenu dans la première étape et avec un vecteur normal cohérent. Par exemple (figure III.2.3), pour une trame de texture de profil droit (b_1) nous ne voulons garder les pixels qui correspondent au profil droit du visage. Nous appelons patch tous les pixels isolés de chaque texture I_p (p allant de 1 à n) que nous voulons garder. Nous utilisons 3 critères pour la détection de patch : la distance d'erreur entre le clone C_S et la trame de profondeur D_p, la direction des vecteurs normaux et la couleur associée à chaque point.

III.2.2.1 Erreur de distance

L'utilisation d'une erreur de distance élimine le bruit du capteur et les erreurs d'alignement. Ce critère est basé sur la distance point à point entre chaque trame de profondeur D_p et le clone sémantique C_S. Nous gardons les patchs de texture P_p correspondant aux points des trames de profondeur détectés. Cette erreur de distance doit être inférieure à un seuil (1 mm). Sur la vue de face de la figure III.2.3 : (a1), il y a un trou dans le nez parce que le capteur ne donne pas l'information. Les données de texture I_p pour cette partie du visage ne reflètent donc pas l'identité de l'objet. L'erreur de distance supprime ces informations (III.2.3 : a2).

III.2.2.2 Vecteurs normaux

La caméra RVB-Z capture plus précisément les zones où l'axe optique est perpendiculaire à l'objet de surface. Par conséquent, nous ne gardons que les points de trames de profondeur D_p qui ont un vecteur normal parallèle à l'axe optique de la caméra. Dans la figure III.2.3, nous constatons que ce critère de sélection élimine les points que la caméra ne saisit pas bien (III.2.3 : a3 et b3). Par exemple, pour une trame de profondeur dans la vue de face (III.2.3 : a1), nous n'obtenons pas toutes les informations du nez. En effet, les points situés sur les côtés du nez ne sont pas bien capturés par la caméra.

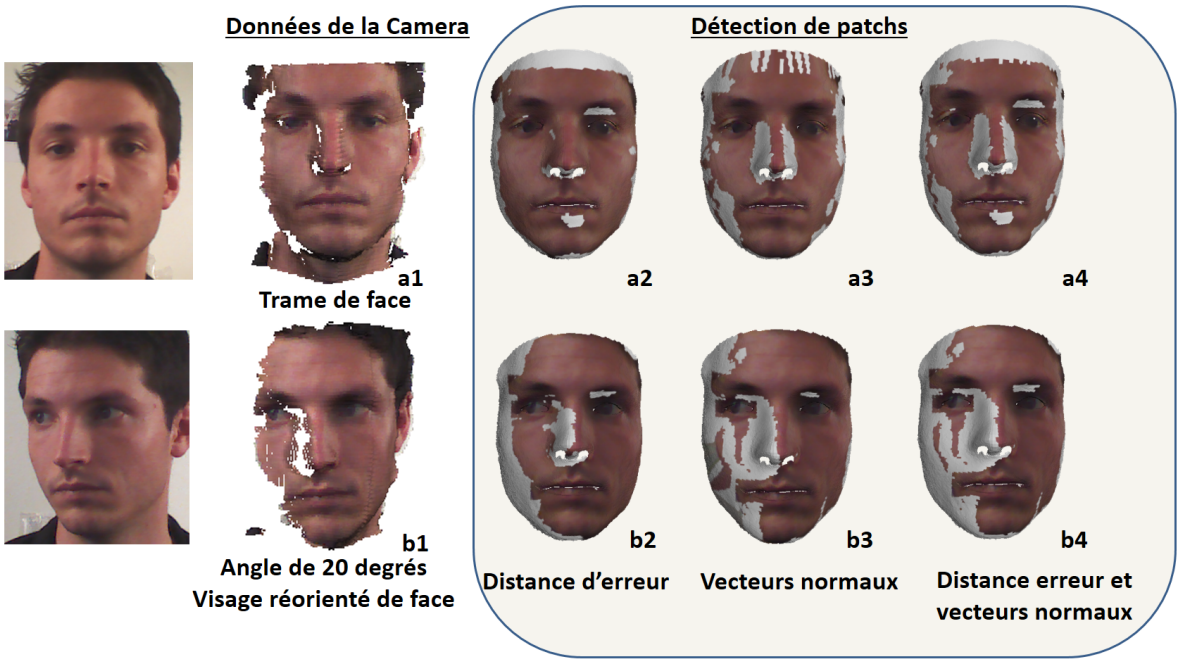


FIGURE III.2.3: Détection des patchs de texture pour chaque trame : a1 : une trame de face. b1 : une trame de profil droit (le visage est re-orienté de face). Nous sélectionnons les parties des textures (patch) où la distance d'erreur est petite (a2 et b2) et où les vecteurs normaux sont pertinents (a3 et b3). Dans notre processus, nous utilisons ces 2 critères (a4 et b4). Le critère de texture n'intervient que lors du traitement de la deuxième trame de la caméra RVB-Z.

III.2.2.3 Couleur

Pour améliorer la qualité de la texture, nous utilisons un critère de couleur. C'est-à-dire que nous calculons la distance couleur entre chaque point de la trame et le point apparié du clone en cours de construction. Si cette distance est trop importante, alors nous éliminons, la zone du patch de texture P_p correspondante. Il faut noter que ce critère ne peut pas être utilisé pour la première trame car la carte de texture du clone est encore vierge.

III.2.2.4 Critères utilisés dans notre méthode

Nous utilisons les trois critères détaillés ci-dessus : les points du patch de texture doivent avoir un vecteur normal parallèle à l'axe optique de la caméra, la distance entre le clone sémantique C_S et la trame de profondeur D_P doit être inférieure à un seuil (nous tolérons un angle de plus ou moins vingt degrés pour les vecteurs normaux) et le critère de couleur aussi inférieur à un seuil. Il est utilisé à partir du traitement de la deuxième trame de texture. Ces trois critères sont utilisés pour détecter les zones qui ne sont pas bien capturées par la caméra et aussi pour éliminer le bruit du capteur et de l'erreur de *fitting*.

III.2.3 *Warping* de la texture des trames sur la carte de texture

Dans cette section, nous voulons *warper* les patchs de texture détectés dans la section précédente sur la carte 2D de texture du clone. La première étape de notre technique de *warping* est la création de la carte de texture T_c du clone C_S. Pour cela, nous projetons le maillage 3D du clone sur un cylindre en utilisant le logiciel Blender [116] [1]. Ensuite, nous utilisons les trames de profondeur D_p pour trouver la position de chaque patch de texture P_p sur la carte T_c. Enfin, nous *warpons* chaque patch de texture pour reconstruire la texture complète du visage.

III.2.3.1 Création de la carte de texture du clone sémantique

Pour créer la carte de texture T_c du clone sémantique C_S, nous utilisons la bibliothèque de Blender [116]. Elle permet de déplier un maillage 3D en le projetant sur un cylindre. Nous obtenons donc une carte T_c du maillage 3D déplié de résolution 3000*3000. Cette projection nous permet d'obtenir les coordonnées de texture du clone 3D. Chaque point 3D (X,Y,Z) du maillage du clone à une correspondance dans la carte de texture (U,V). Nous appelons alors ces points, les points d'ancrage. C'est-à-dire que pour chaque sommet du maillage dans le repère réel (X,Y,Z), nous connaissons ses coordonnées U et V dans la carte de texture. La figure III.2.4 montre le maillage du clone déplié et non déplié. Dans les étapes suivantes, nous complétons cette carte avec les textures des différentes trames obtenues avec la caméra RVB-Z.

III.2.3.2 Positionnement des patchs de texture sur la carte de texture

Après avoir créé la carte de texture T_c du clone C_S, nous calculons le positionnement des différentes trames de texture I_p en utilisant les trames de profondeur D_p correspondantes. La figure III.2.5 présente le schéma de notre méthode pour calculer le positionnement des patchs de texture P_p. Nous cherchons la correspondance entre la carte de texture T_c et les trames de texture I_p. Nous utilisons la correspondance entre les trames de texture I_p (u,v) et les trames de profondeur D_p (x,y,z) fournies par le capteur RVB-Z. C'est-à-dire que pour chaque point des trames de profondeur D_p, nous connaissons ses coordonnées x, y et z, mais aussi ses coordonnées u et v qui permettent de trouver la correspondance avec I_p. Dans la section III.2.1, nous avons aligné chaque carte de profondeur D_p avec le clone sémantique C_S. Nous cherchons donc pour chaque point de la trame D_p traitée, le point 3D du clone sémantiqueC_S correspondant. Nous trouvons cette correspondance en utilisant un critère de distance euclidienne. Pour chaque point de la trame D_p, nous calculons les 4 points du clone C_S les plus proches. Nous réalisons une interpolation en fonction de la distance entre les points. La figure III.2.6 présente notre méthode

1. `https://www.blender.org/`

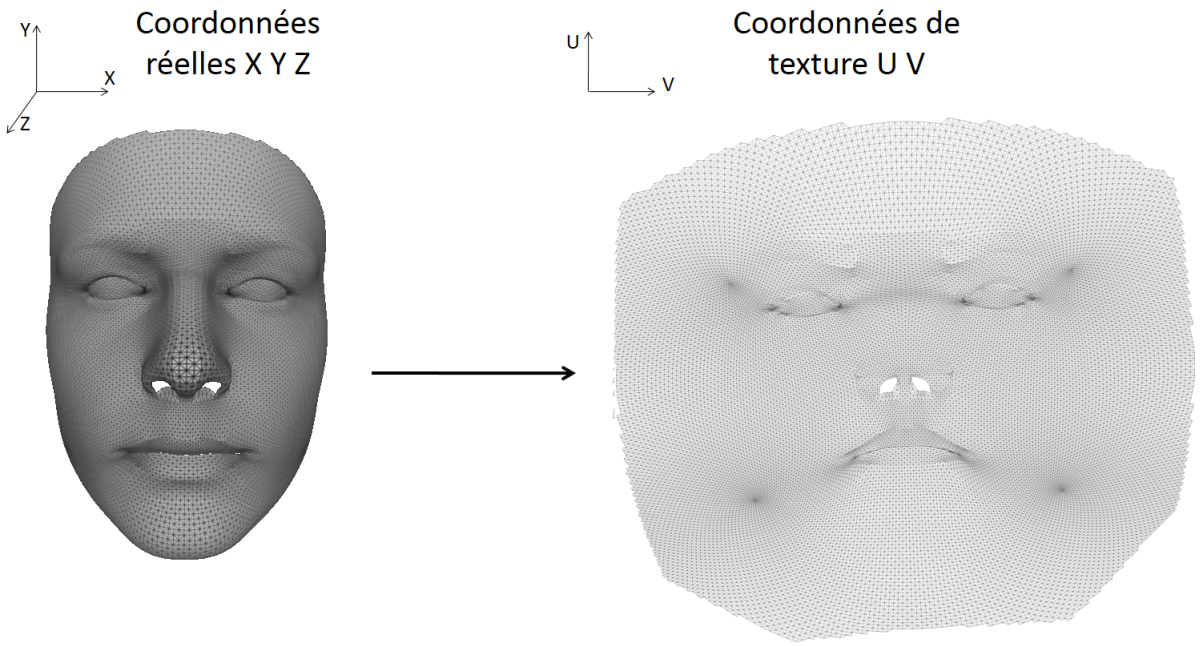


FIGURE III.2.4: Nous créons la carte de texture Tc à partir du maillage 3D du clone sémantique M_c en utilisant le logiciel Blender. Chaque point 3D (X,Y, Z) du clone a une correspondance 2D dans la carte de texture (U,V).

d'interpolation. Cela nous permet de trouver pour chaque point (x,y,z) de la trame D_p, les coordonnées (U,V) de C_S et donc de T_c associées. Comme chaque point (X,Y,Z) du clone C_S est associé à un point d'ancrage (U,V) de sa carte de texture T_c, nous connaissons la position de chaque patch de texture P_p sur la carte de texture du clone T_c. Dans l'étape suivante, nous voulons *warper* ces différents patchs de texture sur cette carte de texture T_c.

III.2.3.3 *Warping* des patchs de texture

Après avoir créé la carte de texture du clone sémantique C_S et trouvé le positionnement de chaque patch de texture P_p sur cette carte, nous voulons *warper* ces patchs. Nous utilisons un *warping* bilinéaire dans notre méthode. La figure III.2.7 décrit notre méthode de *warping*. Nous connaissons pour chaque point d'ancrage des trames de texture I_p , ses coordonnées U et V dans T_c. Nous *warpons* alors les patchs de texture P_p pour compléter la texture des triangles entre les différents points d'ancrage. Nous utilisons une résolution des trames de texture I_p (1280*960) supérieures à la résolution (640*480) des trames de profondeur D_p (voir III.2.2). Cette différence de résolution entre les trames de profondeur D_p et les trames de texture I_p, permet de pouvoir obtenir une texture du clone T_c avec une résolution élevée (3000*3000).

III.2.3.4 Conclusion

Dans cette section, nous avons créé et complété avec les patchs de texture P_p la carte de texture du clone T_c. Dans notre technique, nous utilisons les trames de profondeur D_p pour

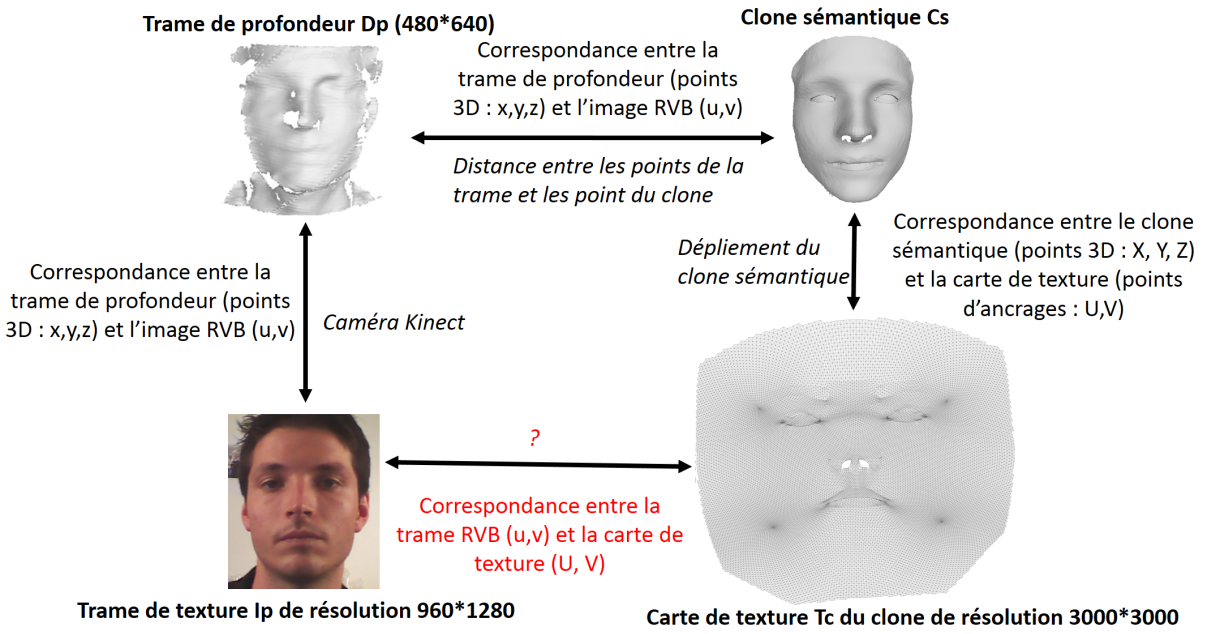


FIGURE III.2.5: Cette figure présente les correspondances entre les trames de profondeur D_p (x,y,z), les trames de texture I_p (u,v), le clone sémantique C_S (X,Y,Z) et sa carte de texture T_c (U,V). Pour pouvoir compléter la carte de texture T_c, nous cherchons la correspondance entre les trames de texture I_p (u,v) et la carte de texture T_c (U,V). Pour trouver cette correspondance, nous utilisons les trames de profondeur D_p (x,y,z) et le clone sémantique C_S (X,Y,Z). Nous cherchons pour chaque point des trames D_p, le point du clone C_s correspondant en utilisant les distances euclidiennes.

trouver le positionnement de chacun de ces patchs P_p sur la carte de texture T_c. Le *warping* de **n** patchs de textures sur la carte de texture du clone T_c, nous permet de reconstruire complètement la texture du clone. Chaque patch de texture I_p ne contient pas toute l'information de texture du visage (face et profil droit et gauche). Mais certains patchs de texture P_p peuvent contenir la même information. C'est pourquoi, nous devons réaliser une fusion de ces données.

III.2.4 Fusion des patchs de texture

La troisième et dernière partie de notre méthode de reconstruction de la texture est la fusion des différents patchs *warpés* sur la carte de texture du clone T_c (voir figure III.2.7). Ces patchs de texture P_p ont été créés dans la section III.2.2. La figure III.2.8 montre la reconstruction de la texture avec notre méthode de patchs. Chacun de ces patchs contient l'information et les caractéristiques de la texture du visage. Mais la même information peut se trouver dans plusieurs patchs de texture P_p obtenus avec des trames différentes. C'est pourquoi, nous devons réaliser une fusion de cette information. La figure III.2.9 illustre le fait que certains patchs P_p sur la carte de texture T_c se chevauchent.

Nous connaissons la sémantique de la carte de texture du clone T_c, c'est pourquoi nous savons exactement quels patchs de texture se chevauchent. Pour chaque pixel de la carte de

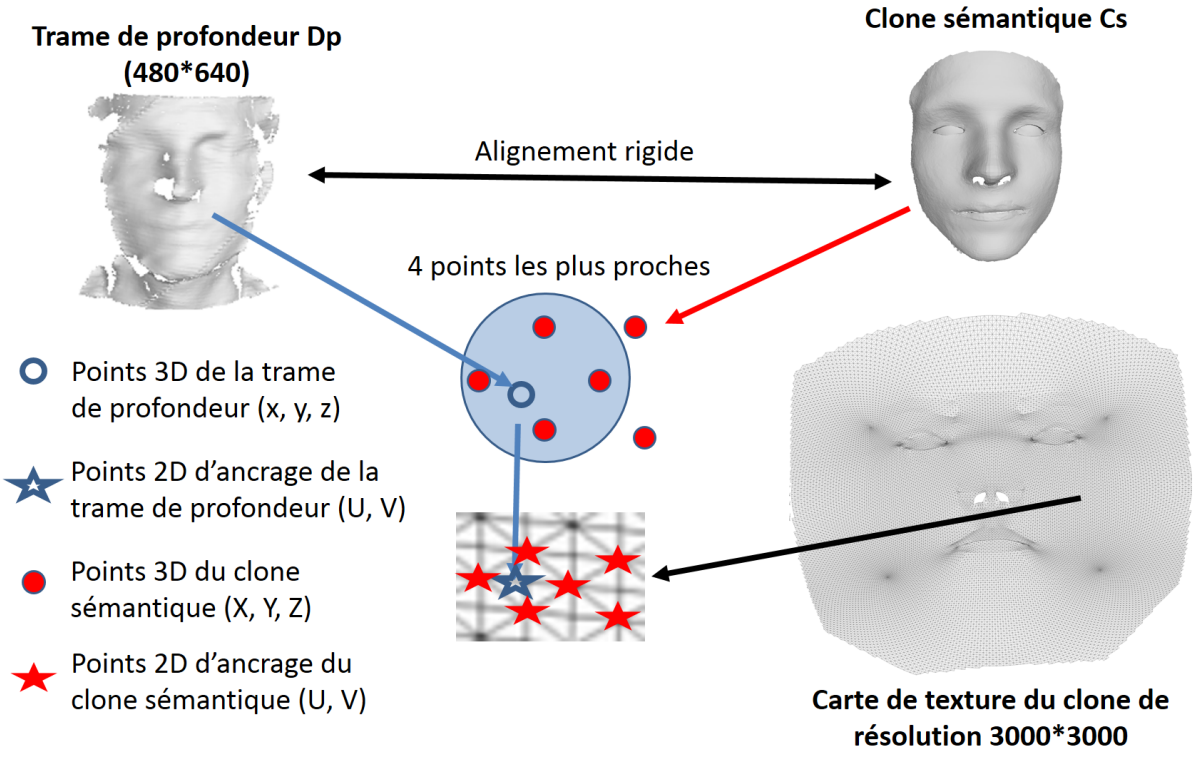


FIGURE III.2.6: La figure décrit notre méthode pour trouver la correspondance entre les trames de profondeur D_p (x,y,z) et le clone sémantique C_S (X,Y,Z). Nous utilisons les distances euclidiennes entre les points des 2 maillages. Pour chaque point de la trame de profondeur D_p, nous cherchons les 4 points les plus proches du clone C_S. De plus, nous connaissons la correspondance entre les points (X,Y,Z) du clone C_S et sa carte de texture T_c (U,V). Grâce à ces correspondances, nous pouvons trouver les coordonnées U,V dans la carte de texture T_c de chaque point de la trame de profondeur D_p. Chaque point de la trame D_p (x,y,z) est positionné sur la carte de texture T_c (U,V) en utilisant sa distance euclidienne (interpolation) avec les 4 points du clone sémantique C_S les plus proches.

texture T_c, nous fusionnons les données RVB des patchs correspondants. Nous avons testé quatre types de fusion : moyenne, médiane, moyenne pondérée et moyenne robuste.

Pour la moyenne, nous moyennons simplement les données RVB qui se superposent sur les pixels de la carte de texture du clone T_c.

Pour la fusion avec une moyenne pondérée, nous ajoutons des poids qui permettent de donner plus d'importance aux données RVB des patchs qui ont une probabilité plus grande d'être correcte. Les poids sont calculés en fonction du critère de distance sur la profondeur. Nous utilisons les distances calculées dans l'étape de détection de patchs III.2.2.

Pour le quatrième type de fusion, nous réalisons une moyenne robuste. Nous ne prenons pas en compte les pixels aberrants. Avant de procéder à la moyenne, nous éliminons les pixels loin de la médiane en utilisant un seuil de couleur. C'est-à-dire que nous éliminons les pixels qui sont éloignés du pixel médian et nous effectuons une moyenne pondérée des autres pixels. Nous utilisons le même type de poids que pour le deuxième type de fusion.

Pour la fusion avec la médiane, nous sélectionnons le pixel médian. Si pour un pixel de

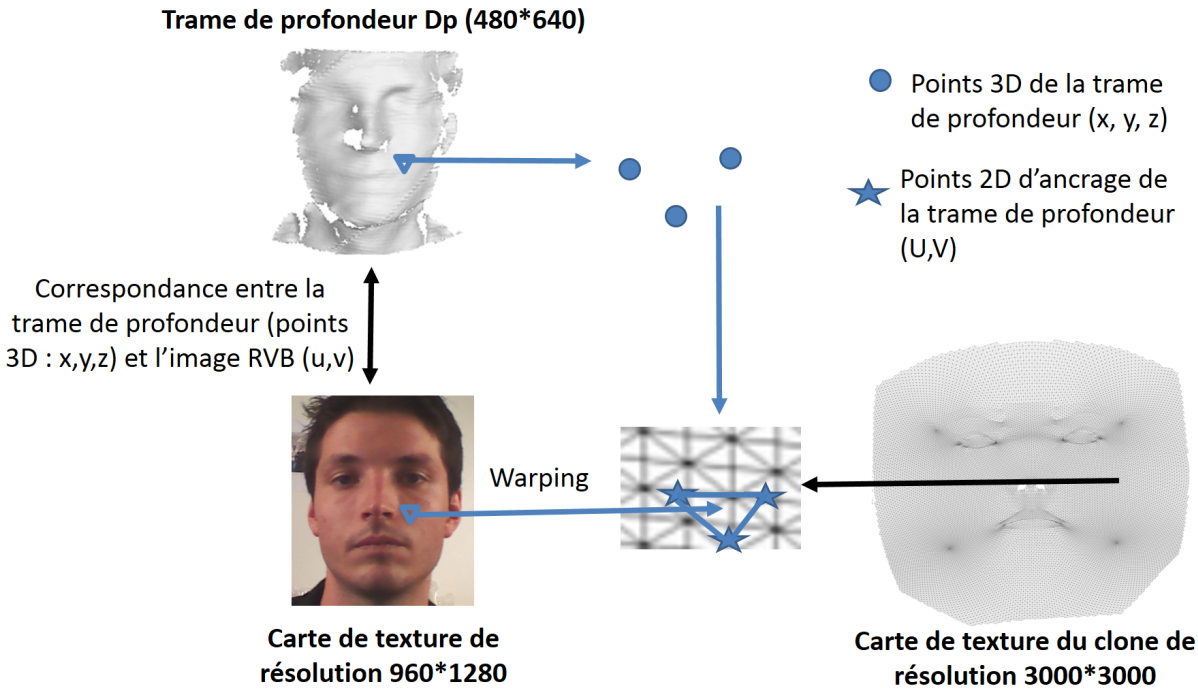


FIGURE III.2.7: Schéma du *warping* des patchs de texture. Le capteur fournit des cartes de texture I_p de résolution 1280*960. Nous utilisons ces trames de texture pour remplir la carte de texture du clone de résolution 3000*3000. Dans la section III.2.2, nous avons détecté les patchs de texture P_p de chaque trame de texture I_p. Ensuite dans la section III.2.3, nous avons trouvé la correspondance entre les points d'ancrage (U,V) de la carte de texture T_c et les trames de texture I_p (u,v). Les patchs de texture P_p sont ensuite *warpés* sur la carte de texture T_c pour remplir les triangles entre les points d'ancrage (U,V). L'utilisation de plusieurs patchs de texture (résolution : 1280*960) permet d'obtenir une texture globale et de grande résolution (3000*3000).

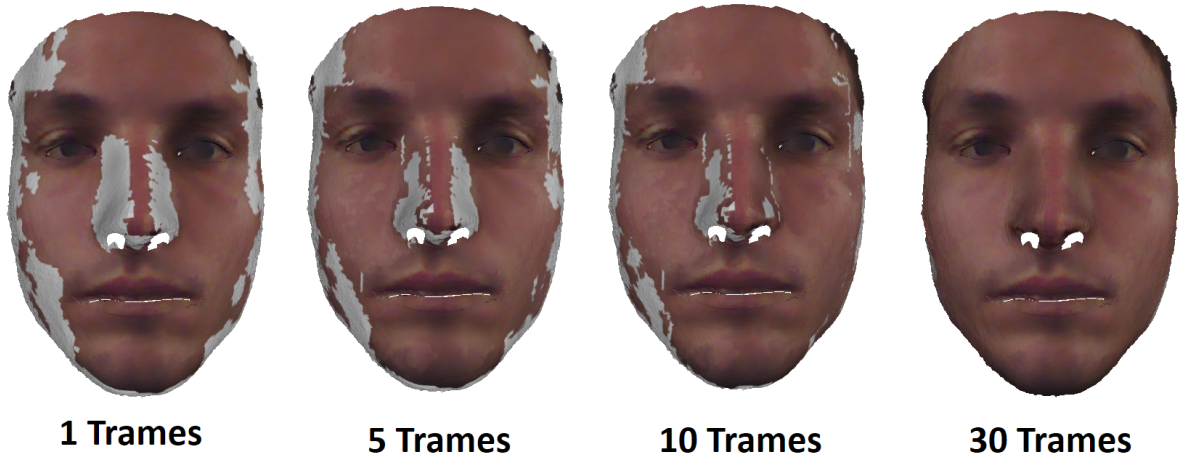


FIGURE III.2.8: Cette figure présente notre méthode de fusion par patchs de texture. En effet, nous pouvons voir le résultat de la reconstruction de la texture à différentes itérations de notre méthode. Après le traitement d'une trame, la texture n'est pas complètement reconstruite. Une trame de face ne contient pas toute l'information du visage (côtés du nez...). Au bout d'une vingtaine de trames, la texture est complètement reconstruite.

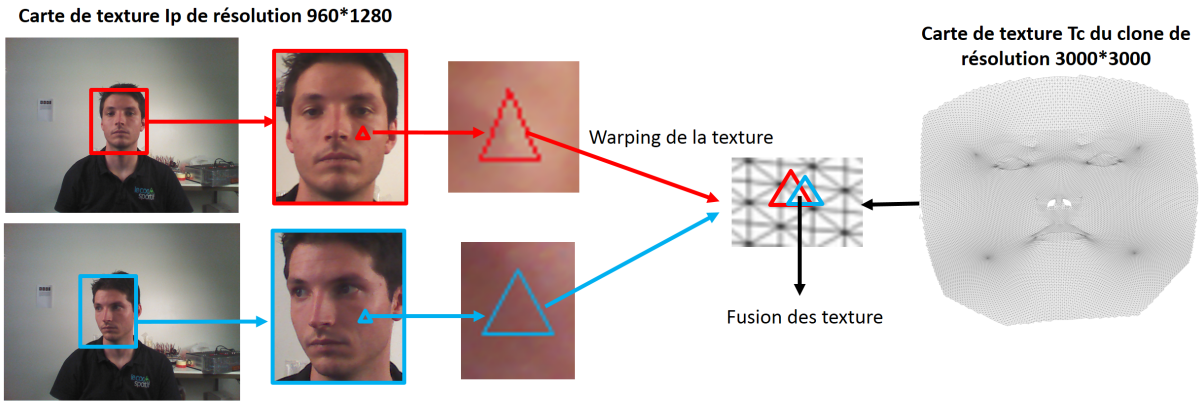


FIGURE III.2.9: Cette figure présente le schéma de notre méthode de fusion des patchs. Chaque patch de texture P_p a été *warpé* sur la carte de texture du clone C_S. Mais certain de ces patchs se chevauchent. C'est pourquoi, nous fusionnons les pixels de ces patchs.

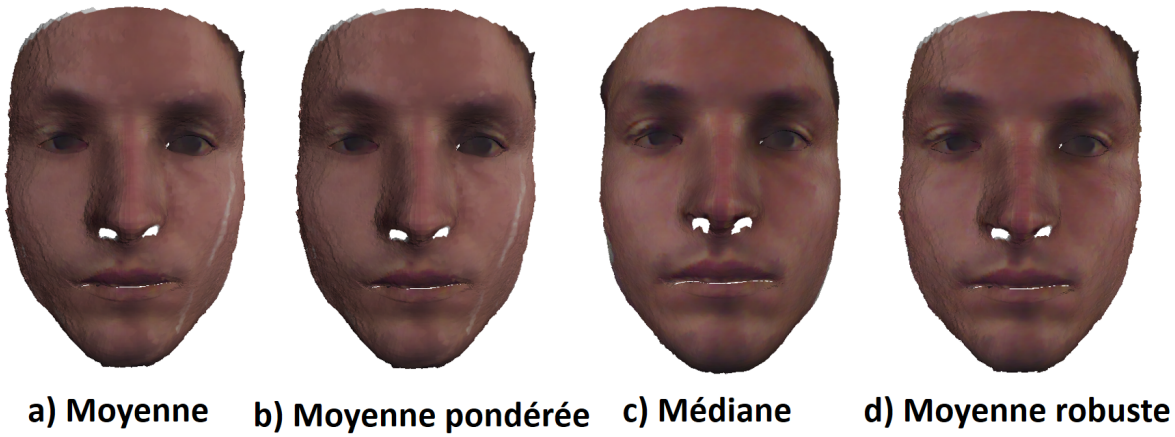


FIGURE III.2.10: Comparaison qualitative des différents types de fusion. Dans notre méthode de reconstruction de la texture, nous avons testé 4 types de fusion de patchs de texture : la moyenne, la moyenne pondérée, la médiane et la moyenne robuste. La moyenne et la moyenne pondérée donnent de moins bons résultats. En effet, certains patchs de texture peuvent contenir des artefacts dus aux erreurs d'alignement (bruit sur la joue gauche des clones) vu que la moyenne et la moyenne pondérée conservent tous les pixels des patchs de texture. La médiane et la moyenne robuste éliminent les valeurs des pixels qui sont éloignés du pixel médian. Ce qui donne en revanche des résultats moins bruités.

la carte de texture du clone, il n'y a que 2 pixels possibles, nous effectuons une moyenne des coordonnées.

Dans notre méthode finale, nous avons utilisé la méthode de fusion avec la médiane. En effet, nous pouvons voir sur la figure III.2.10 que les résultats sont meilleurs avec ce type de fusion.

La figure III.2.10) montre les résultats obtenus avec ces 4 types de fusion. La médiane donne les meilleurs résultats. En effet, elle permet d'éviter l'apparition de flou sur la texture. Les 3 autres méthodes donnent sensiblement les mêmes résultats. Notre méthode à l'avantage

de reconstruire une texture sans coutures. Nous détaillons ces résultats dans la partie IV de ce manuscrit.

III.2.5 Conclusion

Notre technique présentée dans cette partie permet de reconstruire la texture d'un maillage 3D à partir de différentes trames de profondeur d'un visage. Comme pour la reconstruction de la forme, nous utilisons une technique avec des patchs. Ces patchs permettent de ne conserver que les parties de textures cohérentes de chaque trame. Ces patchs de texture ont la particularité d'être sélectionnés à partir de la forme de chaque trame correspondante. Notre technique permet de reconstruire les spécificités de la personne et d'obtenir une texture de haute résolution.

Chapitre *III.3*

Conclusion de notre méthode de clonage

Notre méthode permet de numériser un visage humain en utilisant un capteur RVB-Z bas coût. Ces capteurs ont l'avantage d'être grand public mais ils fournissent des données très bruitées et de basse résolution. Notre méthode est constituée de 2 grandes parties : la reconstruction de la forme et la reconstruction de la texture. Dans notre méthode de reconstruction de la forme, nous utilisons un modèle déformable de visage 3D pour obtenir un clone haute résolution sans bruit qui soit sémantique. Mais ce modèle déformable est dépendant de sa base d'apprentissage. C'est pourquoi, il est très difficile de retrouver certaines spécificités d'une personne en utilisant ce type de modèle. Notre principale contribution est de pouvoir reconstruire plus facilement les caractéristiques physiques des personnes en utilisant un modèle déformable de visage. Contrairement aux techniques classiques, nous réalisons le *fitting* avant la fusion des données. L'inversion du système permet d'être moins dépendant des erreurs d'alignement rigide et de *fitting*. De plus, comme, nous réalisons plusieurs *fitting*, il est plus facile de retrouver les spécificités des individus à partir du modèle. Notre technique est donc mieux adaptée aux caractéristiques physiques des personnes. La deuxième originalité est notre technique de détection et de fusion de patchs. Elle permet de détecter et d'éliminer les zones des trames de profondeur qui ne contiennent pas d'information du visage. C'est pourquoi, nous ne fusionnons que les zones (patchs) fiables qui mettent l'accent sur les spécificités de chaque personne. Pour la reconstruction de la texture, nous utilisons aussi une technique de détection et de fusion de patchs. Notre technique permet d'obtenir une texture précise et de haute résolution à partir d'un scanner basse résolution. Ainsi, notre système de fusion de patch permet de préserver les spécificités physiques (grain de beauté...) et d'éliminer les coutures.

Quatrième partie

Résultats

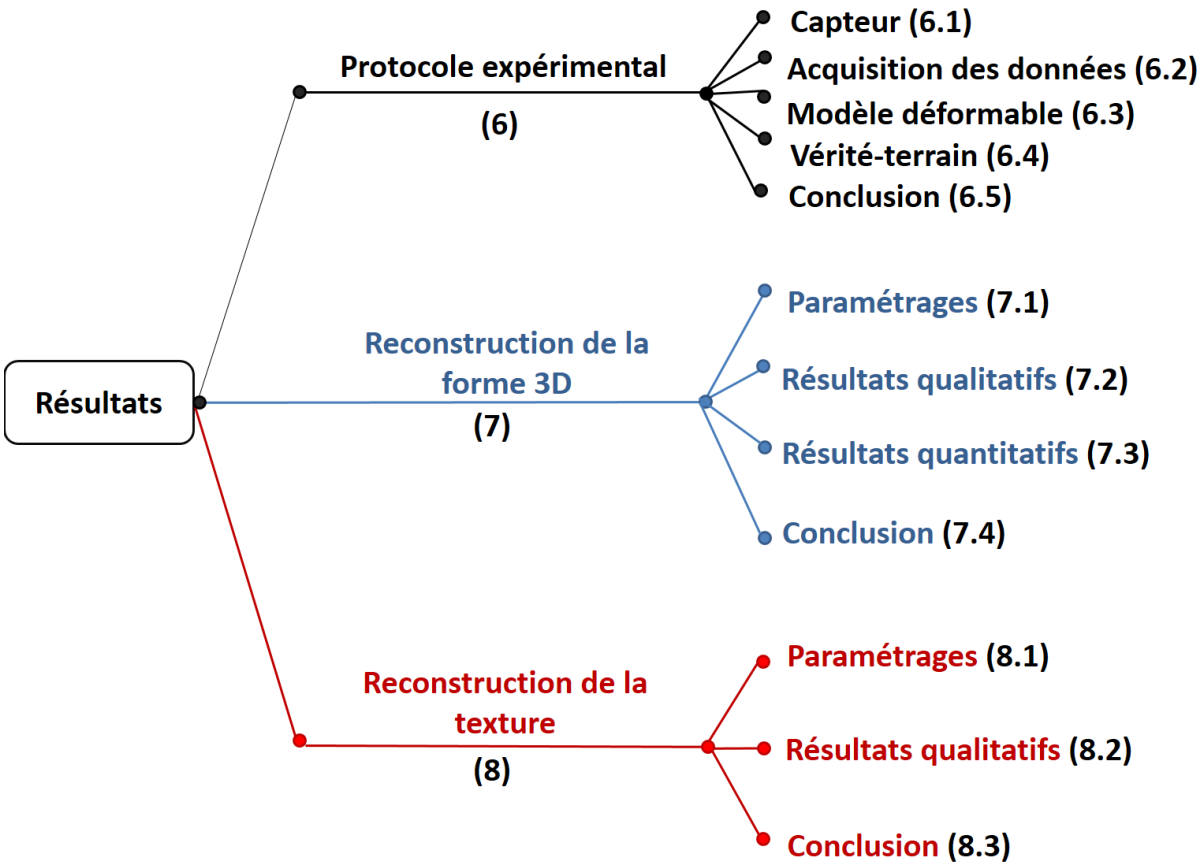


FIGURE IV.0.1: Plan de la partie résultats de ce manuscrit.

Dans ce chapitre, nous présentons les différents résultats obtenus pendant nos travaux de thèse. La figure IV.0.1 présente le plan de notre partie résultats. Dans la première section, nous présentons notre protocole expérimental. Tout d'abord, nous décrivons le protocole que nous utilisons pour acquérir les données de profondeur et de texture de 15 sujets. Ensuite, nous détaillons les caractéristiques du capteur RVB-Z, du modèle déformable de visage 3D et du logiciel que nous utilisons pour la vérité terrain. Dans la deuxième section, nous montrons les résultats obtenus avec notre méthode de reconstruction de la forme du visage 3D. Nous comparons nos résultats avec plusieurs méthodes de l'état de l'art [8, 41] et nous testons sa robustesse. Dans la dernière section, nous décrivons les résultats obtenus avec notre méthode de reconstruction de la texture. Nous testons notre méthode (éclairage, spécificités des visages) et nous la comparons avec des méthodes de l'état de l'art [41, 45]. Pour finir, nous expliquons les avantages de notre méthode.

Chapitre IV.1

Protocole expérimental

Dans ce chapitre, nous présentons notre protocole expérimental. Tout d'abord, nous décrivons les caractéristiques du capteur RVB-Z et le protocole que nous utilisons pour acquérir les données de profondeur et de texture de 15 sujets. Ensuite, nous détaillons les caractéristiques du modèle déformable de visage 3D. Pour finir, nous décrivons le logiciel et le protocole d'acquisition que nous utilisons pour créer la vérité terrain.

Sommaire

IV.1.1 Capteur

Nous utilisons une caméra Kinect version 1 qui est équipée d'un capteur de couleur et d'un capteur de profondeur. La carte de profondeur a une résolution de 320*240 (30 images/seconde) et les images RVB une résolution de 640*480 (30 images/seconde) ou de 1280*960 (15 images/seconde). Les données de profondeur sont souvent interpolé pour obtenir des cartes de profondeur de résolution 640*480. Elle fournit aussi la correspondance entre la profondeur et la couleur (coordonnées de texture). C'est-à-dire que l'on peut reconstruire un nuage de points 3D en couleur (coordonnées 3D et de texture). Elle a une portée de 0.50 à 3.5 mètres. Les caméras Kinect ont l'avantage d'avoir un faible coût et d'être accessibles au grand public. Mais elles fournissent des données de profondeur très bruitées et de faible résolution.

IV.1.2 Acquisition des données

Nous utilisons ce capteur pour réaliser l'acquisition des données de profondeur et de couleur que nous utilisons dans notre méthode de clonage.

Nous testons notre méthode de reconstruction de la forme 3D et de la texture sur 15 sujets (voir figure IV.2.7). Chaque sujet doit être à une distance de la caméra minimale de 0.5 mètre et maximale d'un mètre. Ensuite, il effectue un mouvement de rotation de la tête pour que l'ensemble des zones du visage soient capturées (profil droit, profil gauche..). Cela permet d'avoir un maillage 3D entièrement complété et sans trous. Lors de l'acquisition de données, la personne réalise une expression neutre. Pour chaque sujet, nous capturons environ 100 trames de profondeur et 100 trames de texture RVB.

Pour la reconstruction de la texture, nous travaillons avec des données de profondeur de 640*480 (après interpolation) et des données RVB de résolution 1280*940. Dans notre méthode, la résolution de la texture est donc nécessairement plus importante que celle des cartes de profondeur (voir section II.2.2).

IV.1.3 Modèle déformable

Après avoir acquis les données de couleur et de profondeur avec le capteur Kinect, nous utilisons un modèle déformable de visage 3D [6] pour réaliser une transformation non rigide. La base de données d'apprentissage de ce modèle est constitué de 200 scans de visage 3D (100 femmes, 100 hommes). C'est un scanner à lumière structurée construit par ABW-3D qui a été utilisé pour obtenir ces 200 scans. Chaque maillage 3D est constitué de 53490 sommets connectés par 160 470 triangles. De plus, les maillages sont sémantique. Ils sont composés du visage, des oreilles et du cou de la personne. Une analyse en composante principale (ACP) a été réalisée sur la forme 3D, mais aussi sur la texture des visages et constitue le modèle. Ainsi

une combinaison linéaire des 199 principales composantes de l'ACP permet de reconstruire un visage avec une identité différente.

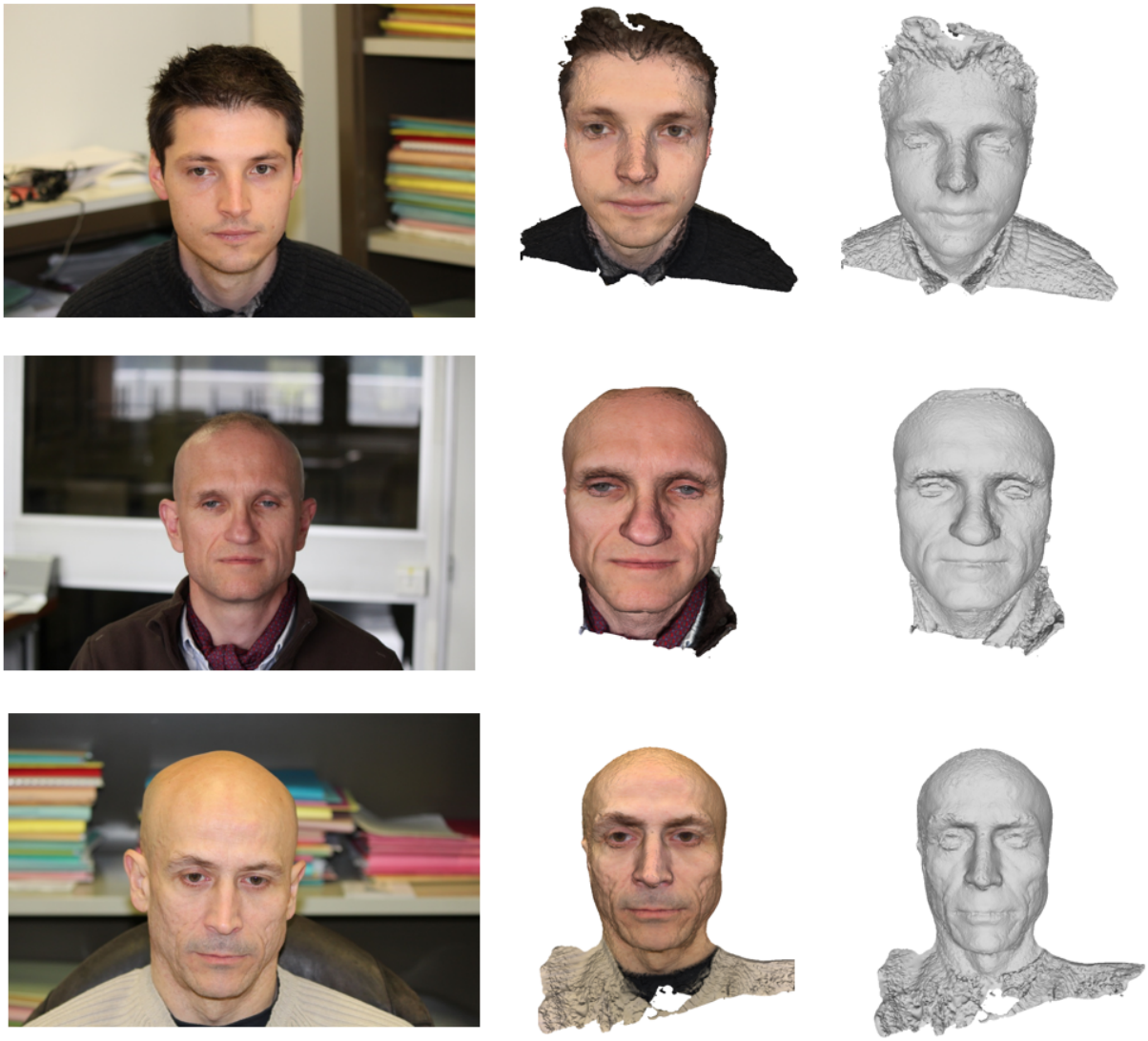


FIGURE IV.1.1: Exemple de clones obtenus avec le logiciel Agisoft [117]. Les clones obtenus sont très précis (paupières,...) et de haute résolution. Nous utilisons ces résultats comme vérité terrain.

IV.1.4 Vérité terrain

Pour comparer nos résultats quantitativement, nous utilisons une vérité terrain. Il est compliqué d'obtenir une vérité terrain de bonne qualité. En effet, les scanners performants coûtent très cher. C'est pourquoi, il est difficile d'avoir accès à ce type de scanner. Nous utilisons le logiciel Agisoft [117] pour générer la vérité terrain. C'est un logiciel de stéréovision qui permet de reconstruire un maillage 3D à partir de plusieurs photos du visage. Le visage doit être photographié sous différents angles de vue. Pour que les résultats soient performants, il faut utiliser

des images RVB de haute résolution et que l'éclairage soit le plus homogène possible. Nous utilisons quinze images de résolutions 5184*3456 pour reconstruire la vérité terrain de chacun de nos sujets. Pendant l'acquisition, le sujet doit être immobile et faire une expression neutre. Les quinze photos doivent être prises à une distance de 1 mètre et en effectuant un arc de cercle en face du sujet. La figure IV.1.1 montre trois exemples de clones obtenus avec ce logiciel.

IV.1.5 Conclusion

Nous avons présenté dans cette partie notre protocole d'acquisition, le matériel et les bases de données qui nous utilisons. Le capteur Kinect nous permet d'obtenir des données d'entrée pour notre méthode. Nous utilisons le modèle déformable de visage [6] pour ensuite traiter les données de profondeur. Grâce au logiciel Agisoft [117], nous obtenons des vérités terrain pour comparer les résultats obtenus avec notre méthode. Dans le chapitre suivant, nous présentons les différents résultats que nous avons obtenus au cours de cette thèse.

Chapitre IV.2

Résultats sur la reconstruction de la forme

Dans ce chapitre, nous présentons les résultats obtenus avec notre méthode de reconstruction de la forme du visage 3D. Nous comparons d'abord nos 4 méthodes de fusion qualitativement et quantitativement et nous testons sa robustesse. Ensuite, nous comparons nos résultats avec plusieurs méthodes de l'état de l'art [8, 41] (figure IV.2.6). Enfin, nous présentons les caractéristiques de nos résultats. La figure IV.2.7 montre les résultats obtenus avec notre méthode sur 15 personnes.

Sommaire

IV.2.1 Paramétrages

IV.2.1.1 Types de fusion

Pour la reconstruction de la forme, nous comparons qualitativement et quantitativement nos quatre méthodes de fusion : la moyenne, la moyenne pondérée, la médiane et la moyenne robuste (voir section III.1.3.1). La figure IV.2.1 montre 2 clones obtenus avec ses différents types de fusion. Pour ces 4 clones, nous utilisons des patchs constitués des données des maillages M_p pour les yeux et des patchs constitués des données des trames D_p pour le reste du visage (voir section III.1.3.2). Les 4 types de fusion sont présentés dans la section III.1.1. Nous obtenons les meilleurs résultats quantitatifs (moyenne : 1.994, moyenne pondérée : 1.98, médiane : 1.995 et moyenne robuste : 1.979) et qualitatifs avec la fusion par moyenne robuste. En effet, elle élimine beaucoup de bruit en gardant les détails. L'erreur affichée sur la figure IV.2.1 est la distance moyenne entre les 15 clones et leurs vérités terrain. Pour calculer cette distance d'erreur, nous apparions chaque point du clone que nous voulons comparer avec le point le plus proche de la vérité terrain, et nous calculons l'erreur de distance globale entre les paires de points. Dans la section suivante, nous comparons les résultats obtenus avec les différents types de patchs.

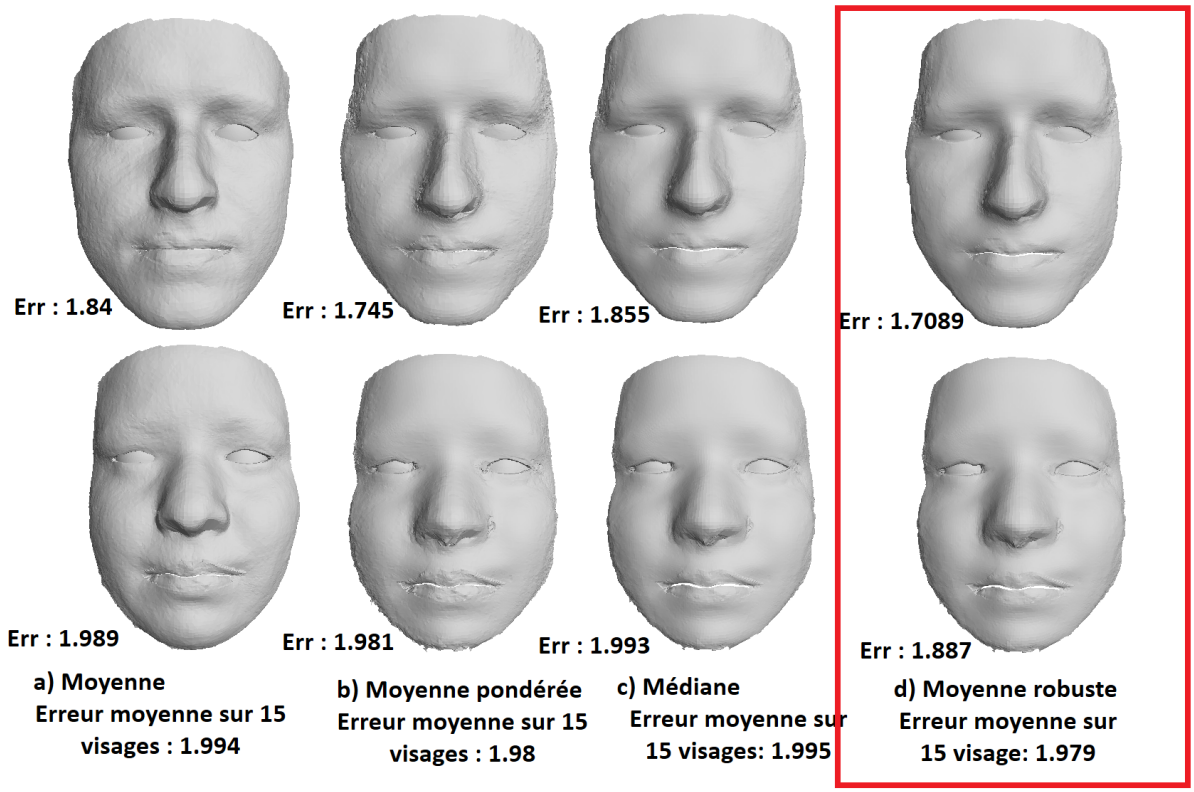


FIGURE IV.2.1: Cette figure présente pour 2 visages, les résultats de nos 4 types de fusion des patchs.

IV.2.1.2 Types de patchs

Pour la reconstruction de la forme, nous testons aussi plusieurs types de patchs (voir section III.1.3.1). La figure IV.2.2 montre les clones (pour trois visages) obtenus avec trois types de patchs de forme différents. Le type de fusion utilisé pour obtenir les clones de la figure IV.2.2 est la moyenne robuste. La figure IV.2.2 confirme que l'on obtient les meilleurs résultats avec les données des maillages et des trames (clones a). De plus, nous observons sur la figure IV.2.2 qu'il n'y a aucune différence quantitative significative entre les clones obtenus avec les trois types de patchs (moyenne sur 15 personnes : 1.979, 1.983, 2.118). Néanmoins, les résultats qualitatifs sont meilleurs avec les patchs contenants les données des trames D_p et des maillages M_p.

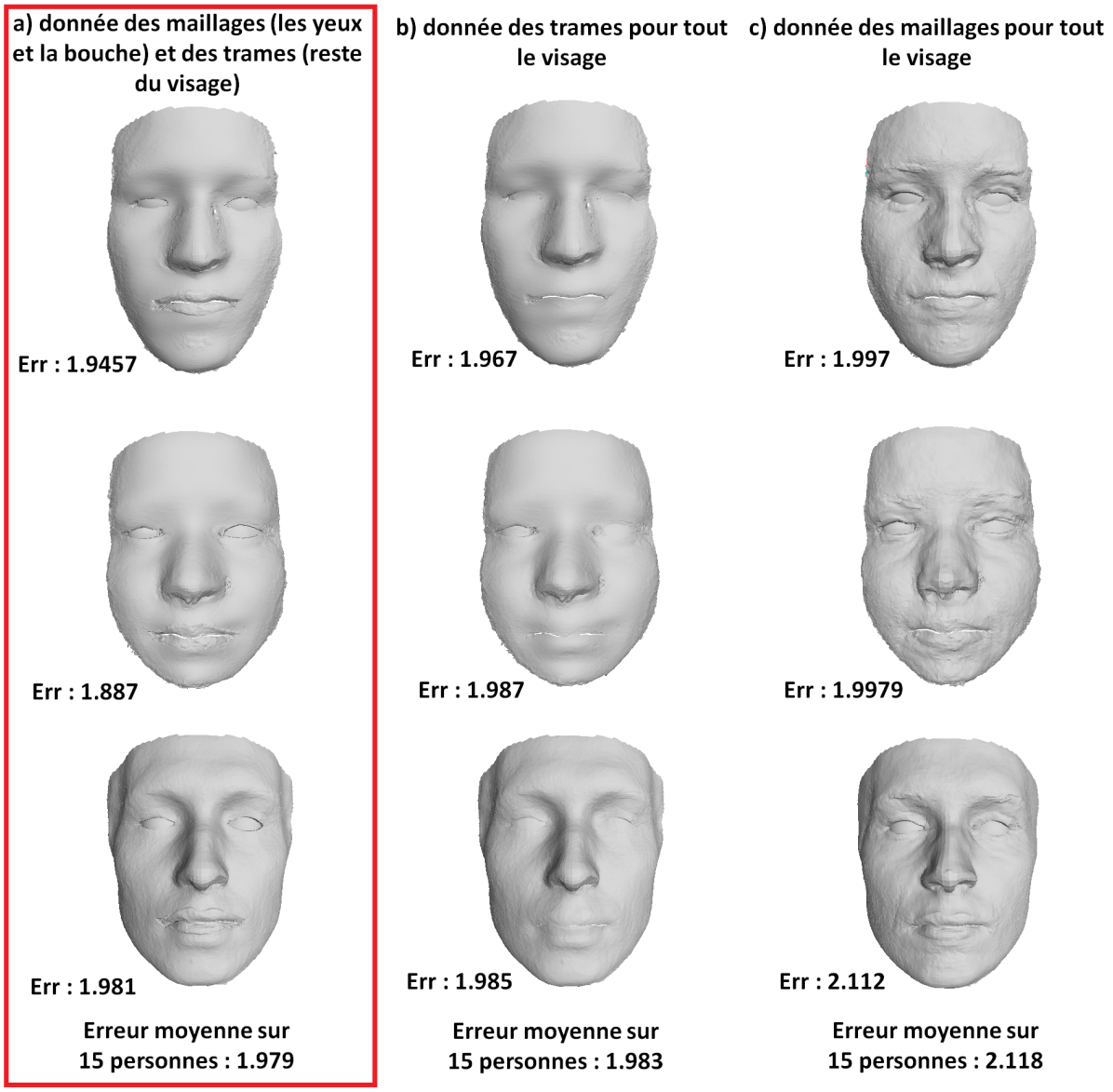


FIGURE IV.2.2: Différents clones obtenus avec les trois types de patchs de forme (pour 2 visages). L'erreur est la distance moyenne entre les 15 clones et leurs vérité terrain.

IV.2.1.3 Robustesse de notre méthode

Nous réalisons des tests pour montrer que notre méthode est stable aux données d'entrée. Pour cela, nous montrons que l'on obtient le même clone 3D si on effectue notre méthode plusieurs fois sur le même visage. La figure IV.2.3 représente des clones de différents visages (3 visages) obtenus avec différentes acquisitions de données. Premièrement, nous reconstruisons un clone avec notre méthode en utilisant 50 trames de profondeur fournies par le capteur RVB-Z (clone a)). Ensuite, nous réalisons une seconde acquisition de données du même visage et nous reconstruisons la forme du visage à partir de ces cinquante nouvelles trames (clone b)). Nous constatons que les résultats obtenus sont quasiment identiques. En effet, notre méthode converge toujours vers un résultat cohérent. De plus, l'ordre de traitement des trames n'a pas d'importance. En effet, les trames de profondeur sont traitées séparément avant que leurs données soient fusionnées.

Nous avons aussi comparé ces clones avec la vérité terrain. Cette comparaison quantitative nous confirme que les résultats obtenus sont quasiment similaires. Ces différents tests montrent que notre méthode est robuste au changement de données d'entrée. En effet, nous obtenons des résultats possédants les mêmes caractéristiques avec différentes données d'entrée.

IV.2.1.4 Avec ou sans patchs

Dans la figure IV.2.4, nous comparons un clone obtenu avec notre méthode (clone a)) et un clone obtenu sans réaliser la détection des patchs (clone b)) (III.1.2). C'est-à-dire nous fusionnons tous les points de chaque maillage M_p sans détecter les patchs (zone adéquate des maillages M_p). Les résultats sont meilleurs quand on utilise des patchs. En effet, ils permettent d'éliminer les erreurs d'acquisitions, de *fitting* et d'alignement, et donc des erreurs de forme sur le résultat final. Nous réalisons aussi une comparaison quantitative avec la vérité terrain (Erreur moyenne sur 15 personnes avec des patchs : 1.979, sans patch : 2.437). Ces résultats montrent que la partie détection de patchs permet d'obtenir des meilleurs résultats et elle est donc très importante dans notre méthode.

IV.2.1.5 Projection dans l'espace ACP du modèle déformable

La figure IV.2.5 montre que le clone obtenu avec notre méthode (clone a)) ne se trouve pas dans l'espace formé par les 200 visages de la base de données du modèle déformable. Pour montrer cela, nous projetons le clone a) dans l'espace du modèle déformable. C'est-à-dire que nous calculons les 199 paramètres du modèle déformable correspondant à notre clone a) en utilisant le maillage moyen, les écarts types et les composantes principales fournis avec le modèle. Nous obtenons donc la combinaison linéaire des 199 principales composantes de l'ACP du maillage 3D obtenue avec notre méthode. À partir de ces 199 paramètres, nous reconstruisons un nouveau clone 3D b) (figure IV.2.5). Nous pouvons voir que le nouveau clone b) n'est pas

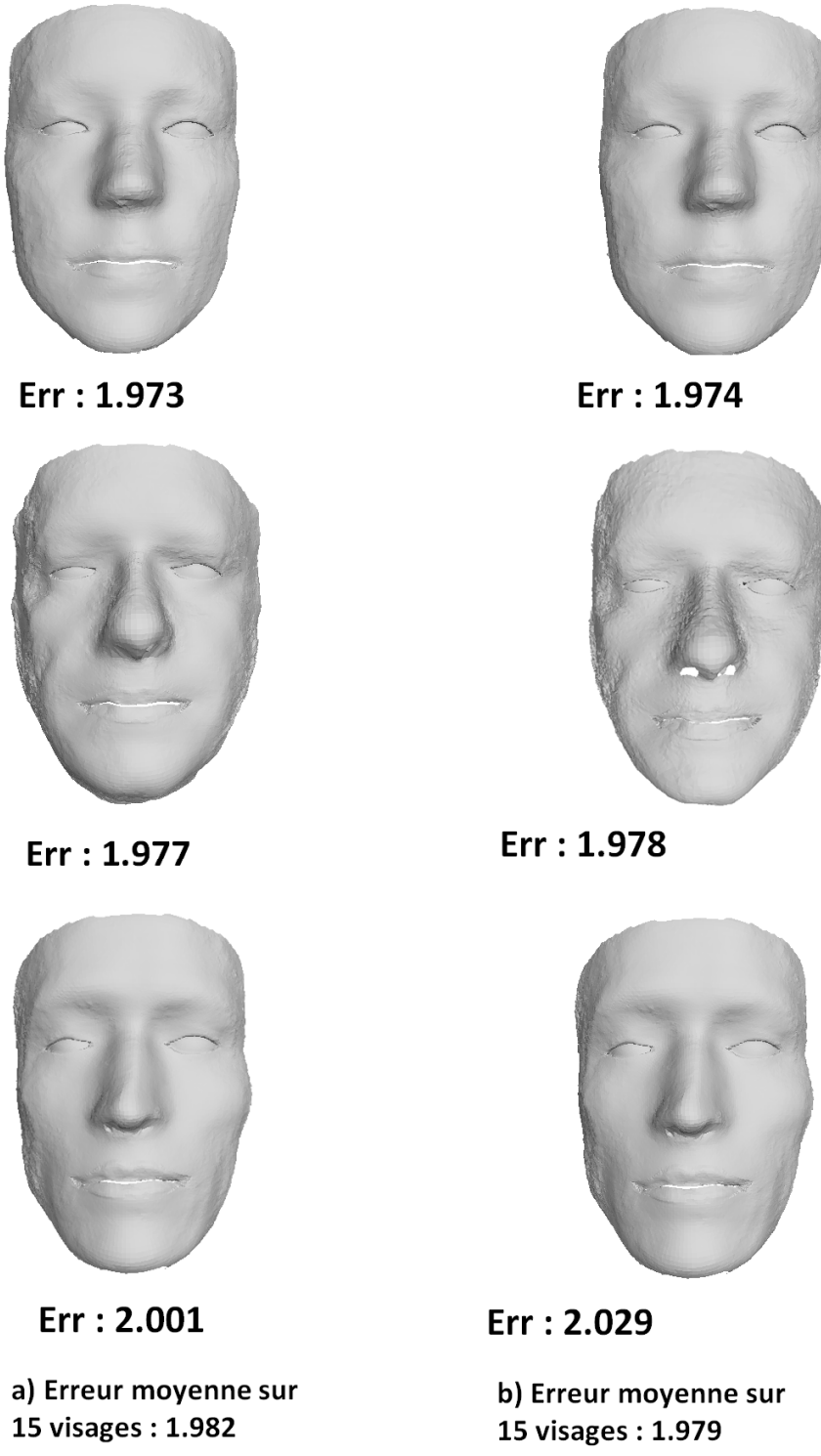


FIGURE IV.2.3: Test de la stabilité de notre méthode. Nous avons reconstruit 2 clones (a et b) du même visage avec différentes acquisitions de données. Nous obtenons des résultats sensiblement identiques (distance d'erreur avec la vérité terrain). Nous avons réalisé ce test sur 15 personnes.

identique au clone obtenu avec notre méthode. Nous montrons que le clone est moins précis en

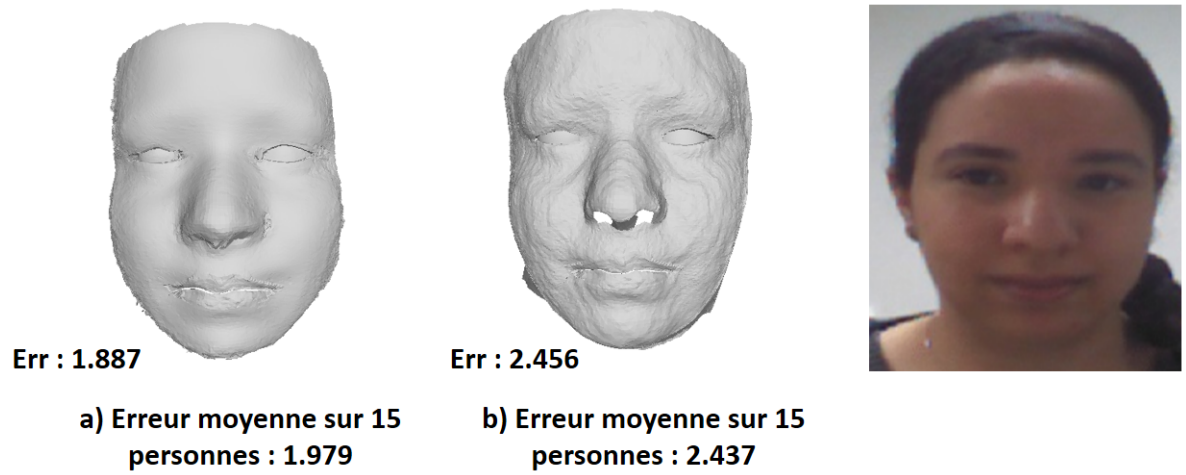


FIGURE IV.2.4: Résultats obtenus avec notre méthode (clone a) et notre méthode sans patch (clone b). C'est-à-dire que nous fusionnons toutes les données de chaque trame de forme sans utiliser notre méthode de détection de patchs. Nous obtenons une distance d'erreur plus grande quand nous n'utilisons pas de patchs. Ces résultats montrent que l'utilisation des patchs de forme permet d'améliorer les résultats.

réalisant une comparaison quantitative avec la vérité terrain (Erreur moyenne sur 15 personnes : clone a : 1.979, clone b : 2.245)). De plus, on peut voir sur la figure IV.2.5 que le clone b) est moins réaliste que le clone a) (nez plus large, visage moins fin). Notre méthode est donc moins dépendante du modèle déformable. En effet, il n'est pas possible de retrouver notre clone en utilisant une méthode classique de *fitting*.

IV.2.1.6 Conclusion

Après avoir réalisé ces différents tests, nous choisissons d'utiliser, dans notre méthode de reconstruction de la forme, la technique de fusion par moyenne pondérée. En effet, elle permet d'obtenir le clone 3D le plus proche de la vérité terrain. Nous choisissons aussi d'utiliser des patchs composés des données des maillages M_p pour les yeux et la bouche et des patchs contenant les données des trames D_p pour le reste du visage. Ces paramétrages nous semblent les plus cohérents car ils permettent d'éliminer les caractéristiques physiques propres au modèle, le bruit du capteur Kinect et les erreurs d'alignement rigide. De plus, la distance d'erreur avec la vérité terrain est plus petite en utilisant ces paramétrages (voir les figures IV.2.2 et III.1.15). La figure IV.2.7 montre les résultats obtenus avec ces réglages pour 15 sujets différents.

IV.2.2 Résultats qualitatifs

Dans cette section, nous présentons les résultats qualitatifs. Tout d'abord, nous comparons qualitativement notre méthode avec 2 méthodes de l'état de l'art [8, 41]. Dans la deuxième section, nous présentons les résultats obtenus avec notre méthode sur 15 personnes.

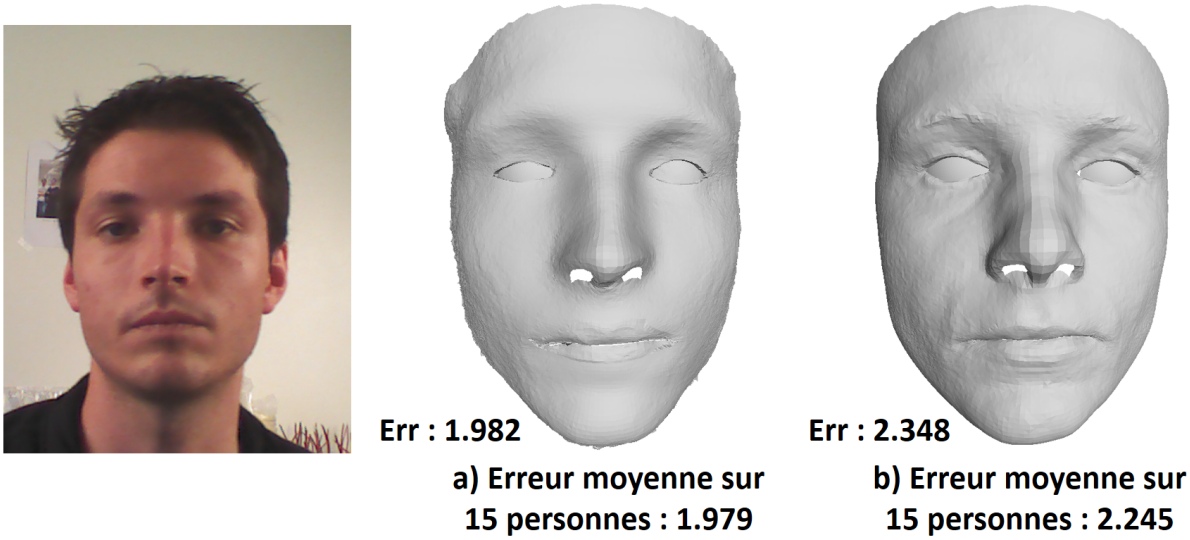


FIGURE IV.2.5: Le clone a) est obtenu avec notre méthode de reconstruction de la forme 3D. Le clone b) est la projection du clone a) dans l'espace du modèle déformable de MorphFace [6]. À partir de ces 199 coefficients, nous avons obtenu le clone b). Les 2 clones de la figure IV.2.5 ne sont pas identiques. Donc le clone obtenu ne se trouve pas dans l'espace ACP du modèle. De plus l'erreur de distance du clone b) avec la vérité terrain est plus grande, ce qui montre l'importance d'inverser le processus et d'utiliser des patchs de forme.

IV.2.2.1 Comparaisons qualitatives avec les méthodes de l'état de l'art

Nous comparons qualitativement notre méthode aux processus classiques de *fitting* [8] et à Kinect Fusion [41]. La figure IV.2.6 montre des clones (3 visages) obtenus avec ces différentes méthodes de clonage. Les clones obtenus avec la méthode de Kinect Fusion [41] (figure IV.2.6 : b) ne sont pas des clones sémantiques. De plus, on peut voir sur la figure IV.2.6 que les zones des yeux et des lèvres sont très lissées car la caméra Kinect fournies des données incorrectes dans ces zones du visage. C'est pourquoi, sur les clones b) obtenus avec des méthodes qui n'utilisent pas de modèle déformable, souvent le globe oculaire n'apparaît pas. Les méthodes de *fitting* classiques [8] (figure IV.2.6 : c) réalisent d'abord une fusion des trames de profondeur et ensuite un *fitting*. Ici, la méthode de *fitting* auquel nous nous comparons utilise l'ICP décrite au paragraphe III.1.1, la fusion des trames étant réalisée par Kinect Fusion [41]. La méthode de *fitting* classique fournit des clones sémantiques. De plus, le scanner qui est utilisé pour créer la base de données du modèle est beaucoup plus performant que le capteur RVB-Z. C'est pourquoi les clones c) sont beaucoup plus précis au niveau des yeux et de la bouche. Mais ces clones obtenus avec la méthode de *fitting* classique possèdent moins de spécificités de l'individu. En effet, les modèles déformables de visage ne contiennent pas toutes les formes possibles et détails du visage du sujet et leurs bases de données d'apprentissage sont limitées. Dans notre procédé (figure IV.2.6 : d), les zones des yeux et de la bouche ne sont pas lissées et les clones possèdent plus de spécificités de la personne que les clones 3D créés par la méthode de *fitting* classique [8].

On peut voir que le globe oculaire est présent et que le visage est plus cohérent. En effet, notre méthode de détection et de fusion de patch permet d'être moins limité par la base de données du modèle. De plus, les résultats quantitatifs confirment que les clones obtenus avec notre méthode sont plus proche de la vérité terrain que les clones obtenus avec [8].

IV.2.2.2 Résultats sur 15 personnes

Nous pouvons voir sur la figure IV.2.7 : b) (front, joues) et j) (nez) que la barbe peut entrainer du bruit sur le résultat final. En effet, les barbes augmentent le bruit sur les trames de profondeur fournies par la Kinect et accroissent les erreurs d'alignement rigide. Les artefacts au niveau des joues et de la moustache peuvent perturber l'algorithme ICP (voir section III.1.1) et donc augmenter les erreurs de *fitting*. On remarque sur la figure IV.2.7 : a) et f) qu'une moustache et une barbe courte ne perturbent pas notre méthode. En effet, les maillages a) et f) ne sont pas bruités. Ce type de barbe entraîne moins de bruit sur les joues des trames fournies par le capteur.

IV.2.3 Résultats quantitatifs

Pour comparer nos résultats avec la vérité terrain (voir section IV.1.4), nous calculons une distance d'erreur entre les 2 maillages 3D (entre les clones C_s et la vérité terrain). Cette comparaison quantitative illustre la précision de nos résultats. La figure IV.2.8 montre 2 exemples de comparaisons quantitatives avec la vérité terrain. Nous observons dans le tableau IV.2.1 que l'erreur globale moyenne est plus petite avec notre méthode (Erreur : 1,979) qu'avec le processus classique de *fitting* (erreur = 2,356). Notre méthode permet donc d'obtenir des meilleurs résultats quantitatifs que la méthode de l'état de l'art [8]. Le tableau IV.2.1 montre aussi que nous obtenons les meilleurs résultats en utilisant des patchs composés des données des maillages M_p et des trames de profondeur D_p (voir section III.1.3.2). La figure IV.2.9 montre les histogrammes des erreurs de distance avec la vérité terrain. L'histogramme de gauche présente les erreurs entre les 15 clones obtenus avec [8] et la vérité terrain. L'histogramme de droite présente les erreurs pour les 15 clones obtenus avec notre méthode. Pour finir, nous avons comparé nos résultats avec le maillage moyen du modèle déformable (voir figure IV.2.11). Nous pouvons voir que les clones que nous obtenons sont plus ou moins loin du maillage moyen. Ce qui montre que les 15 visages de notre base de test ont des formes 3D diversifiées.

La figure IV.2.10 montre la moyenne des erreurs de distance entre les clones (15 visages) et la vérité terrain. Nous avons effectué cette moyenne pour les clones obtenus avec notre méthode et les clones obtenus avec la méthode [8]. L'erreur montre que notre méthode est plus précise au niveau du nez et du menton des visages.

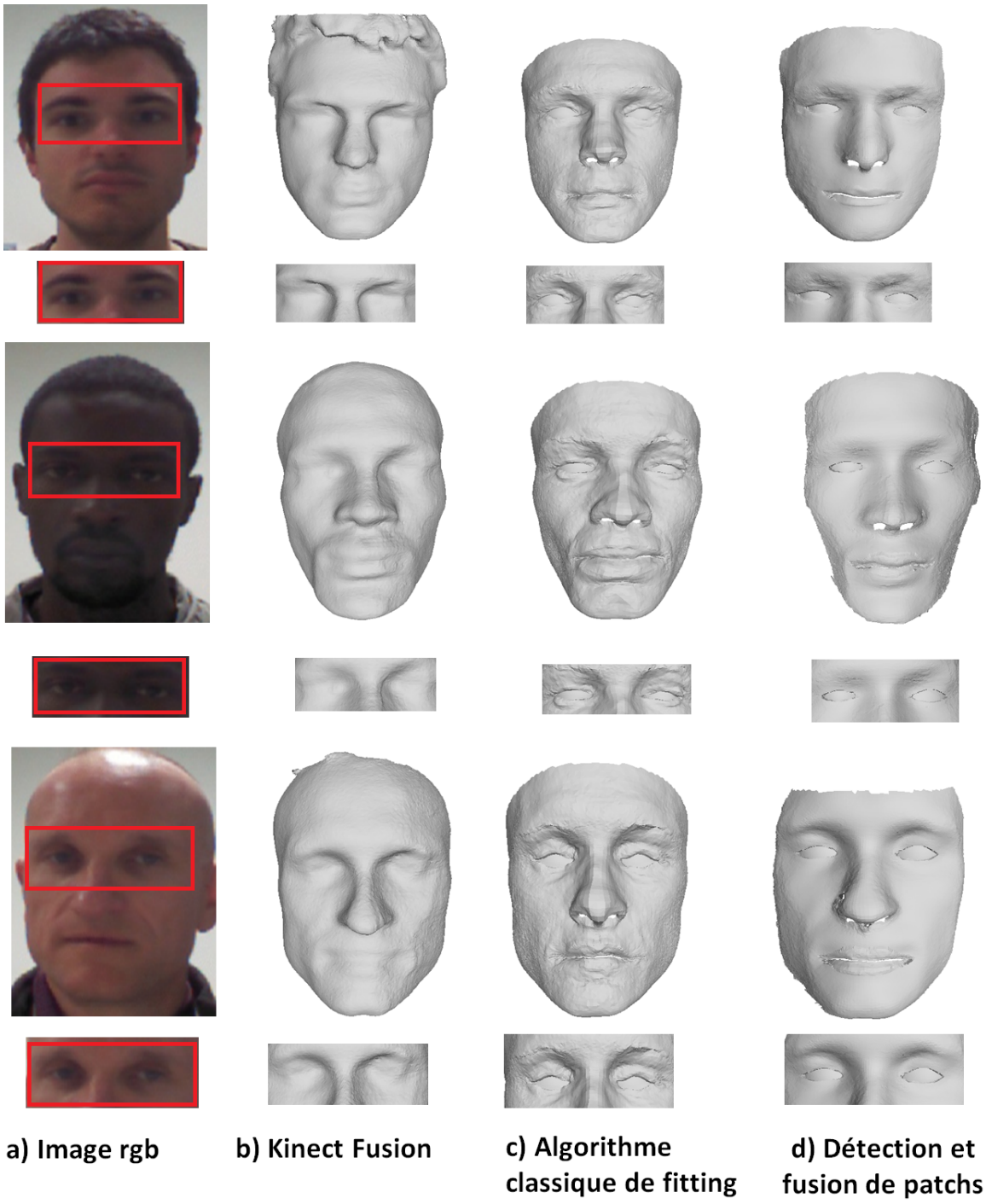


FIGURE IV.2.6: Comparaisons qualitatives entre les clones obtenus avec notre méthode et les clones obtenus avec les méthodes de l'état de l'art [8, 41]. Kinect Fusion [41] permet d'obtenir un clone de bonne qualité. Mais le clone obtenu n'est pas sémantique et n'est pas précis au niveau des yeux et de la bouche (pas de globe oculaire). Au contraire, le clone obtenu avec la méthode de Cao et al [8] est sémantique et possède un globe oculaire mais est moins réaliste. Notre méthode (colonne d) permet d'obtenir un clone réaliste, sémantique et qui possède les caractéristiques physiques d'un visage au niveau des yeux (globe oculaire).

IV.2.4 Conclusion

Notre méthode permet d'obtenir un clone sémantique avec plus de spécificités que les méthodes classiques de *fitting* [8]. Dans notre méthode, nous inversons le processus car nous

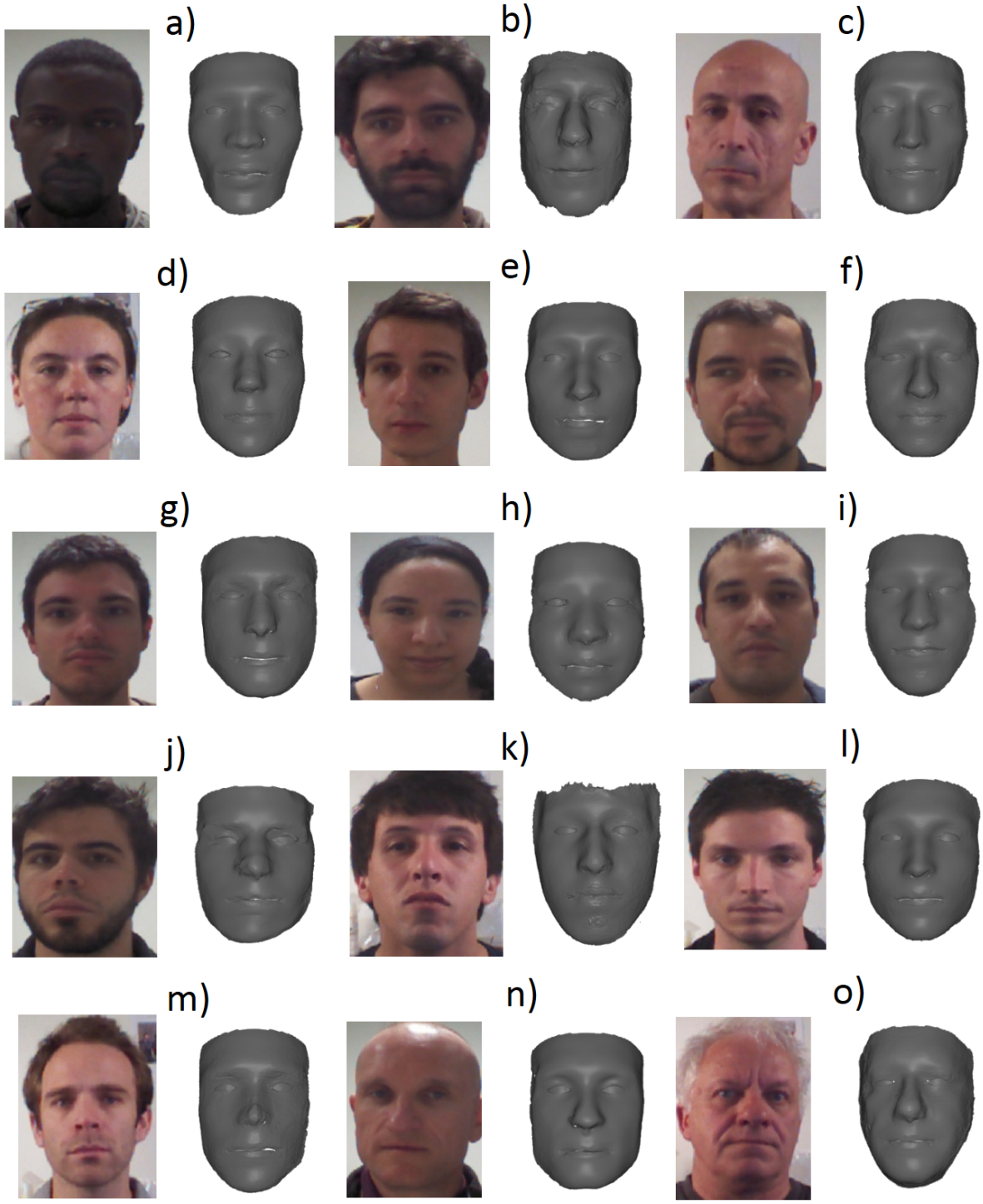


FIGURE IV.2.7: Nous testons notre méthode de reconstruction de la forme du visage sur 15 sujets. Nous avons sélectionné des personnes avec des caractéristiques physiques différentes (genre, barbe...). Notre méthode fonctionne correctement pour les 15 personnes (voir tableau IV.2.1). Les barbes peuvent entrainer des erreurs d'alignements et donc du bruit sur le résultat.

réalisons d'abord le *fitting* sur chaque trame de profondeur D_p, puis la fusion. Nous savons que la base de données du modèle déformable ne contient pas toutes les spécificités physiques possibles et que ces spécificités sont corrélées. C'est-à-dire que même si 2 spécificités du visage

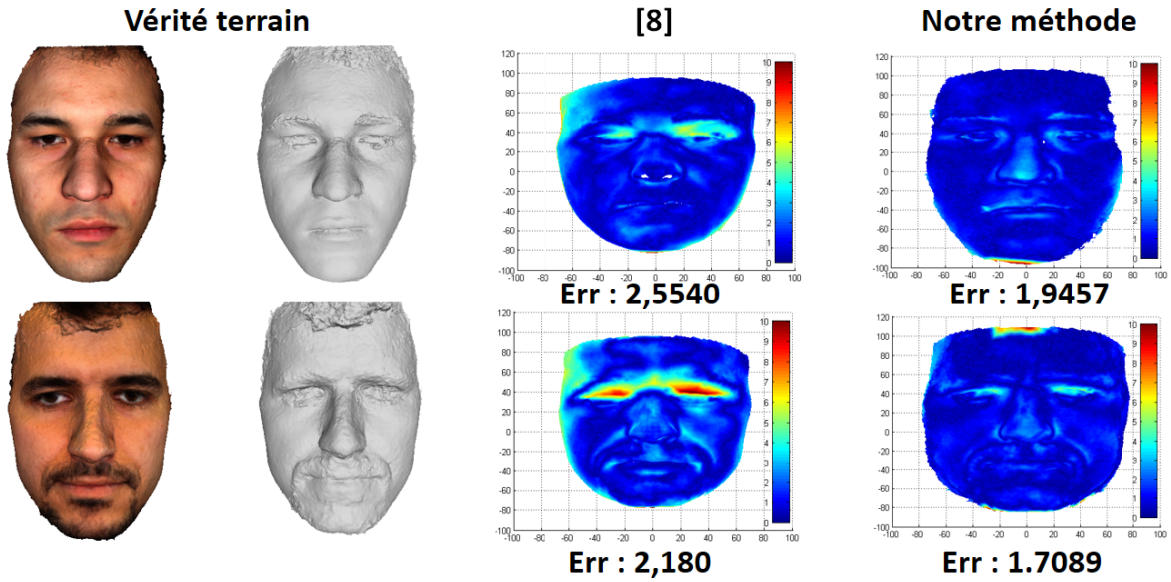


FIGURE IV.2.8: Comparaison quantitative avec la vérité terrain. Elle est obtenue avec le logiciel de stéréovision Agisoft [117]. Nous comparons notre méthode et la méthode de l'état de l'art [8] avec la vérité terrain. L'erreur globale est plus petite avec notre méthode. Nous avons réalisé cette comparaison sur 15 personnes différentes.

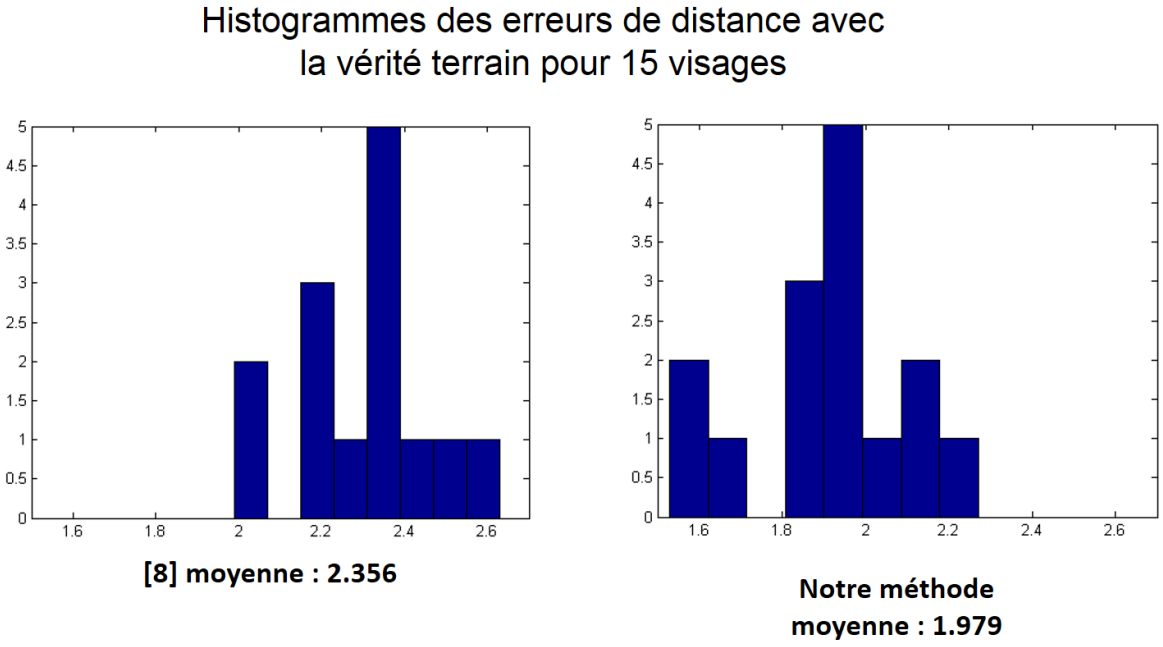


FIGURE IV.2.9: Histogramme des erreurs avec la vérité terrain. La distance d'erreur moyenne entre la vérité terrain et le clone obtenu avec notre méthode est 1.979. La distance d'erreur moyenne avec le clone obtenu avec [8] est de 2.356.

se trouvent dans la base de données du modèle, nous ne sommes pas sûr de pouvoir les retrouver en n'effectuant qu'une seule étape de *fitting*. En effet, si elles ne sont pas corrélées, nous pouvons

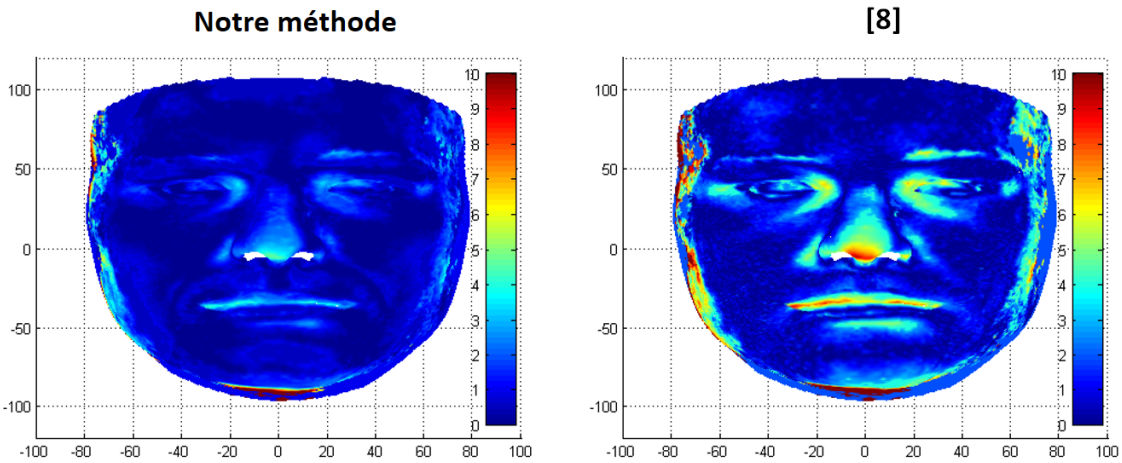


FIGURE IV.2.10: Nous avons calculé pour chaque clone (15 personnes) obtenu avec notre méthode, la distance d'erreur de son maillage 3D avec la vérité terrain. La figure montre la moyenne de ses erreurs pour les 15 visages. Nous avons réalisé la même opération avec les clones obtenus avec la méthode [8]. Notre méthode est plus précise au niveau du nez et du menton.

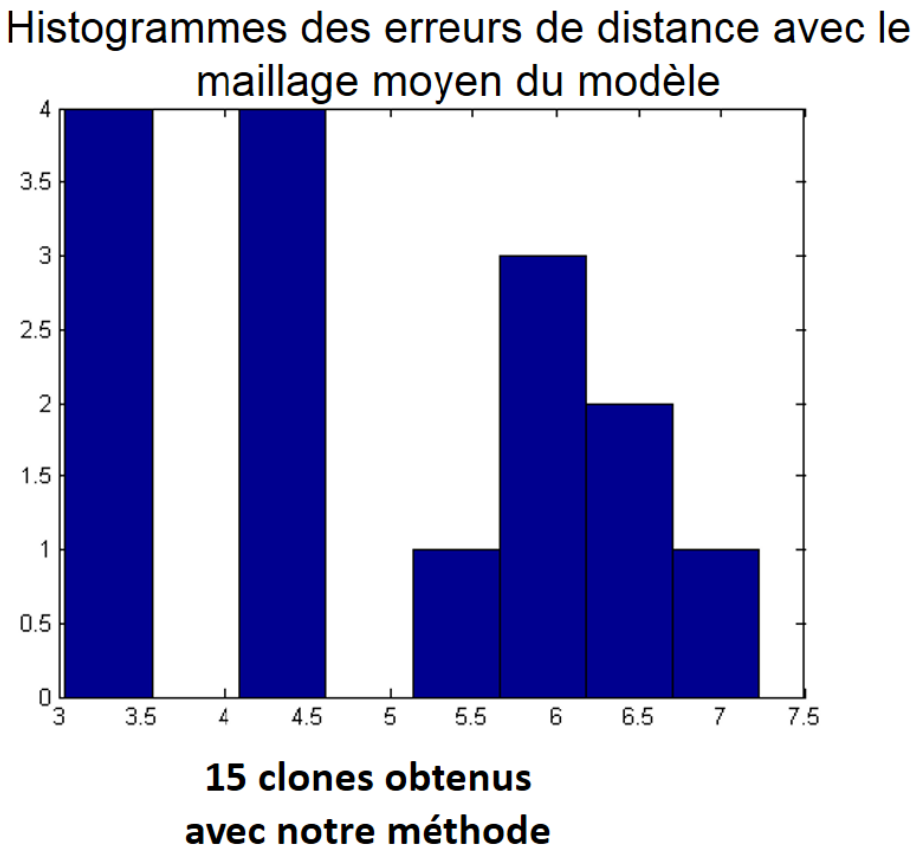


FIGURE IV.2.11: Histogramme des erreurs avec le maillage moyen du modèle déformable de visage et les 15 clones obtenus avec notre méthode.

TABLE IV.2.1: Comparaison quantitative avec la vérité terrain.

Méthode	[8]	Notre méthode : données des maillages	Notre méthode : données des trames de profondeur	**Notre méthode : données des maillages et des trames de profondeur**
Erreur moyenne sur 15 visages	2.356	2.118	1.983	**1.979**

ne reconstruire qu'une seule de ces 2 spécificités physiques. C'est pourquoi, dans notre méthode, nous effectuons plusieurs étapes de *fitting*. Cela nous permet de retrouver plus facilement les caractéristiques physiques. Notre technique de détection de patchs permet aussi d'être moins dépendant des alignements et des erreurs de *fitting* parce que nous fusionnons a posteriori des informations fiables. Donc notre méthode est mieux adaptée aux caractéristiques d'un visage inconnu de la base de données. De plus, notre méthode permet de fournir, comme toutes les méthodes qui utilisent des modèles déformables, des clones sémantiques. Nous gardons les avantages de ces techniques en améliorant les résultats. Notre technique de détection et de fusion de patch permet d'être moins limitée par le modèle.

Chapitre IV.3

Résultats sur la reconstruction de la texture du visage

Dans ce chapitre, nous décrivons les résultats obtenus avec notre méthode de reconstruction de la texture. Nous testons les différents types de fusion de notre méthode et sa robustesse. Puis, nous comparons qualitativement les résultats obtenus avec notre technique avec plusieurs méthodes de l'état de l'art. Pour finir, nous expliquons les avantages de notre technique et les caractéristiques de nos résultats.

Sommaire

IV.3.1 Paramétrages

IV.3.1.1 Types de fusion

Nous comparons les 4 types de fusion de notre méthode de reconstruction de la texture : moyenne, médiane, moyenne pondérée et moyenne robuste. La figure IV.3.1 montre la reconstruction de la texture 3D obtenue avec les 4 types de fusion pour 2 visages. Ces 4 techniques sont décrites dans la section III.2.4. Nous avons testé notre méthode sur 15 personnes différentes. La figure IV.3.1 confirme que l'on obtient les meilleurs résultats avec la fusion par médiane et la fusion par moyenne robuste.

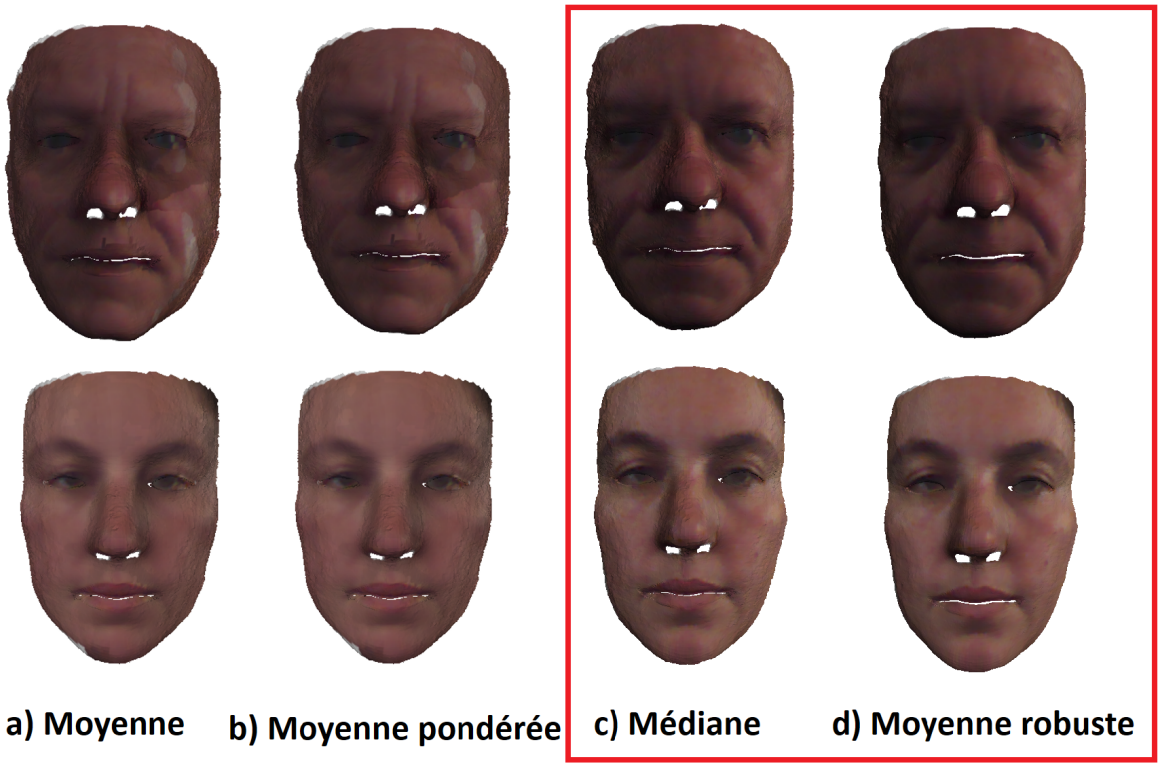


FIGURE IV.3.1: Cette figure montre des clones obtenus avec nos 4 type de fusion. La médiane et la moyenne robuste donnent les meilleurs résultats.

IV.3.1.2 Robustesse de notre méthode

Dans cette section, nous expliquons comment nous évaluons la stabilité de notre méthode. Nous montrons que nous obtenons des clones similaires en utilisant des données d'entrée différentes du même visage. La figure IV.3.2 représente des textures obtenues avec notre méthode pour trois visages. Nous effectuons les mêmes tests que pour la reconstruction de la forme (IV.2.1.3). Les textures des clones a) sont reconstruites à partir de 30 trames de texture. Les clones b) ont été reconstruits avec 30 trames différentes des mêmes visages. Les résultats sur la figure IV.3.2 a) et b) sont identiques. Nous obtenons donc des résultats satisfaisants avec

différentes données d'entrée. Cela permet de montrer que notre méthode de reconstruction de texture est robuste aux données d'entrée. De plus, il n'y a pas d'opération aléatoire dans notre méthode. C'est pourquoi, nous obtenons strictement les mêmes résultats avec notre méthode sur des données d'entrée identique.

IV.3.1.3 Avec ou sans patchs

Nous testons aussi notre méthode sans utiliser les patchs de texture. La figure IV.3.3 compare notre méthode sans patch et avec patch. On peut remarquer que le résultat sans l'utilisation de patchs est de moins bonne qualité. On peut voir sur la figure IV.3.3 que la texture ne contient aucune spécificité. La texture du visage est devenue homogène. Les erreurs d'alignement et les parties des trames de texture mal capturées par le capteur ne sont pas éliminées. Elles sont donc prises en compte dans la fusion des résultats. La détection des patchs de texture est donc une étape très importante de notre méthode. Elle permet d'éviter que les erreurs de *fitting* et le bruit du capteur (bord...) viennent perturber la texture.

IV.3.2 Résultats qualitatifs

Dans cette section, nous décrivons les résultats que nous avons obtenus sur 15 sujets de notre base. La figure IV.3.7 montre que notre méthode fonctionne sur différentes personnes. Les textures obtenues ne contiennent pas de bruit et sont représentatives de la personne (barbe, grain de beauté...).

IV.3.2.1 Reconstruction des spécificités

Nous comparons notre méthode avec la méthode [45] qui projette une trame de texture sur un maillage 3D. La figure IV.3.5 montre que notre méthode fonctionne mieux que la méthode [45] pour certaines spécificités (grain de beauté ...). Nous avons pris l'exemple d'une personne avec un grain de beauté sur le nez. Dans [45], ils projettent une image de vue de face unique sur le clone 3D. Nous notons, dans la figure IV.3.5, que le grain de beauté est déformé sur leurs résultats parce que la trame vue de face ne contient pas l'information correcte des côtés du nez. En effet, la courbure du nez est très importante. Par conséquent, sur les résultats obtenus avec [45], le grain de beauté est très étendu et déformé. Dans notre méthode, nous utilisons les trames de profondeur pour détecter les patchs de texture. La direction des vecteurs normaux permet d'éliminer les zones du visage qui ne sont pas perpendiculaires à l'axe optique de la caméra (les bords...). Par conséquent, nous avons atteint de meilleurs résultat, car nous éliminons les côtés du nez pour une trame vue de face. Dans nos résultats, le grain de beauté n'est pas étendu et est placé au bon endroit (voir la figure IV.3.5).

FIGURE IV.3.2: Comme pour la forme, nous testons la stabilité de notre technique de reconstruction de la texture. Les textures des clones a) ont été reconstruites à partir de 30 trames de texture. Les clones b) ont été reconstruits avec 30 trames différentes des mêmes visages. Nous pouvons voir que les textures représentant le même visage sont identiques. Ces différents tests montrent que notre technique est robuste aux données d'entrée.

IV.3.2.2 Mauvaise conditions d'éclairage

Nous comparons notre méthode avec celle de Kinect Fusion [41]. La figure IV.3.4 montre un clone obtenu avec notre méthode et un clone obtenu avec la méthode de Kinect Fusion. Nous

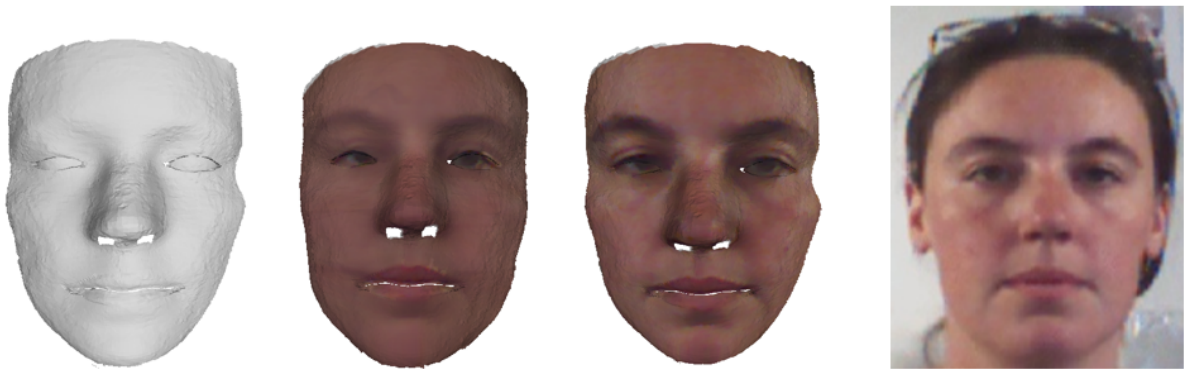


FIGURE IV.3.3: Résultats obtenus avec notre méthode et notre méthode sans patchs. C'est-à-dire que nous fusionnons toutes les données de chaque trame de texture. La texture obtenue est de moins bonne qualité. En effet, les détails du visage n'apparaissent pas sur la texture. Les erreurs d'alignement et les erreurs d'acquisitions ne sont pas éliminées par la détection de patchs. De plus, nous pouvons voir que le résultat obtenu est plus flou.

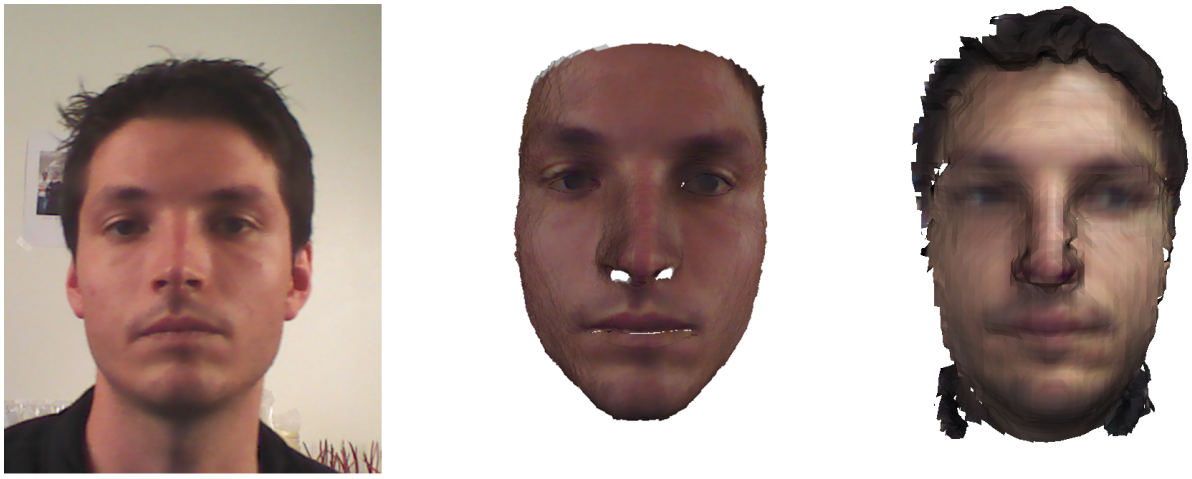


FIGURE IV.3.4: Nous comparons notre méthode (clone b)) avec la méthode Kinect Fusion [41] (clone c)). Ces 2 méthodes permettent d'obtenir des textures de bonne qualité. Il faut noter que la méthode de Kinect Fusion [41] ne permet pas d'obtenir un maillage sémantique.

comparons, notre méthode avec KinectFusion [41] car ils utilisent aussi des données fournies par une caméra Kinect. Cette figure montre que ces 2 méthodes permettent d'obtenir des textures de bonne qualité. A noter que Kinect Fusion [41] ne fournit pas des clones 3D sémantiques et ne peut donc pas être directement utilisée comme prétraitement dans des applications ayant besoin de la correspondance des points de maillage avec le visage. Nous testons ces 2 méthodes dans des mauvaises conditions. La figure IV.3.6 montre que notre méthode est plus efficace avec un mauvais éclairage que [41]. Nous avons délibérément utilisé une lumière de côté, de sorte que la lumière ne soit pas uniforme. Les résultats obtenus avec [41] contiennent du bruit (figure :

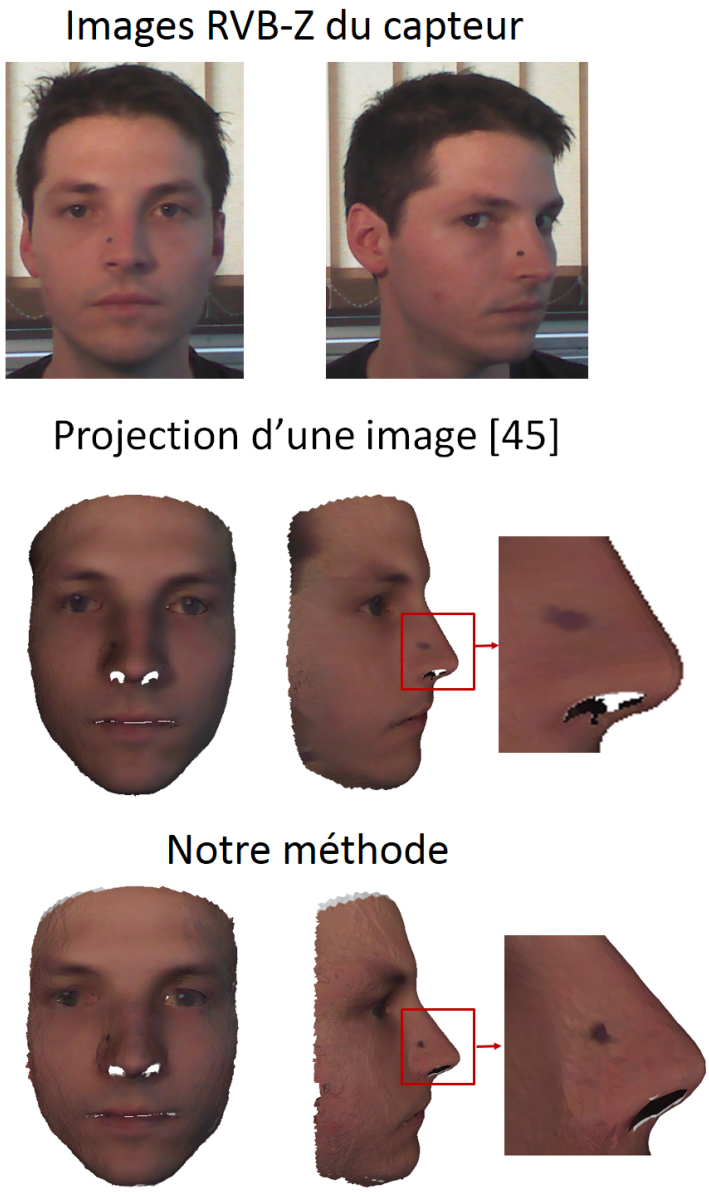


FIGURE IV.3.5: Nous comparons notre méthode avec la méthode d'Hernandez et al [45] qui projettent une image de vue de face unique sur le clone 3D. Notre méthode permet d'obtenir des meilleurs résultats. En effet, nous pouvons voir que le grain de beauté est mieux représenté avec notre méthode. Avec la méthode d'Hernandez et al [45], il est déformé et n'est pas au bon endroit.

IV.3.6 : côté gauche de la face). Notre méthode permet d'éviter ce type de bruit parce que nous ne conservons que les zones perpendiculaires à l'axe de la caméra. C'est pourquoi, notre système de patch ne fusionne que des informations adéquates. Mais nous voyons l'apparition de couture au niveau des yeux, des sourcils et de la bouche parce que nous utilisons les informations de la première image pour les yeux, la bouche et les sourcils (figure IV.3.6). En effet, le mauvais éclairage est la cause de ces coutures. Ces coutures peuvent être éliminées en utilisant les équations de Poisson comme dans [88]. En effet, nous connaissons la position des coutures parce

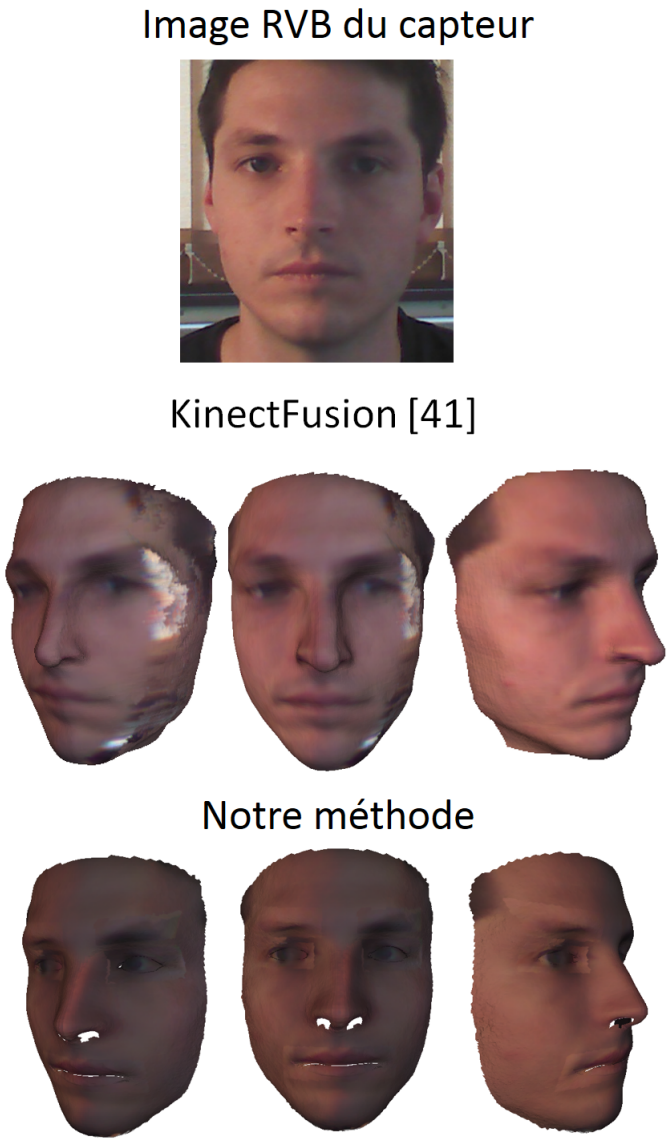


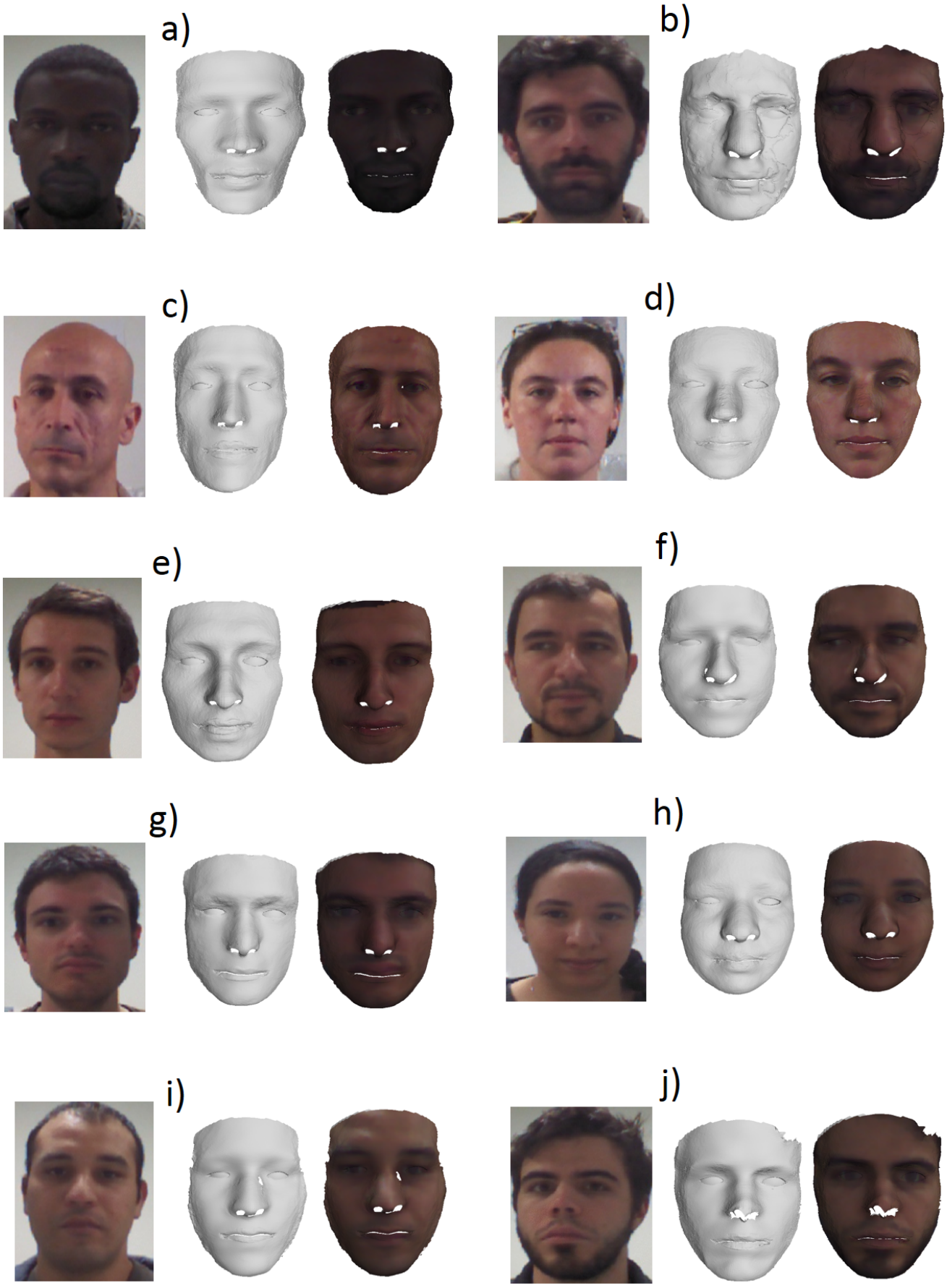
FIGURE IV.3.6: Nous comparons notre méthode et la méthode de Kinect Fusion [41] dans des mauvaises conditions d'éclairages. En effet, nous positionnons une lampe qui éclaire seulement un profil du visage (image RVB du capteur). Les résultats obtenus avec Kinect Fusion [41] sont très bruités (artefact sur le côté droit du visage). Notre méthode permet d'obtenir de meilleurs résultats.

que nous utilisons un clone sémantique.

IV.3.3 Conclusion

Notre technique permet d'obtenir une texture précise, de bonne qualité et qui contient les spécificités des personnes. De plus, elle permet de reconstruire la texture d'un maillage sémantique. La figure IV.3.7 montre les résultats obtenus sur différentes personnes (15 personnes

dans notre base de test). Nous avons choisis des sujets avec des caractéristiques physiques différentes (barbe, grain de beauté...). Notre technique fonctionne correctement pour tous les sujets. En effet, nous pouvons voir que sur les sujets b) et j) la barbe et la moustache sont présentes. Nous pouvons aussi voir que même si le clone sémantique contient des trous dans son maillage (clone j) au niveau du nez), la reconstruction de la texture fonctionne correctement. Nous n'avons pas comparé nos résultats quantitativement car il est difficile d'obtenir une vérité terrain. En effet, la texture dépend de la luminosité de la pièce et du capteur utilisé (capteur actif, flash...).


a)
b)
c)
d)
e)
f)
g)
h)
i)
j)

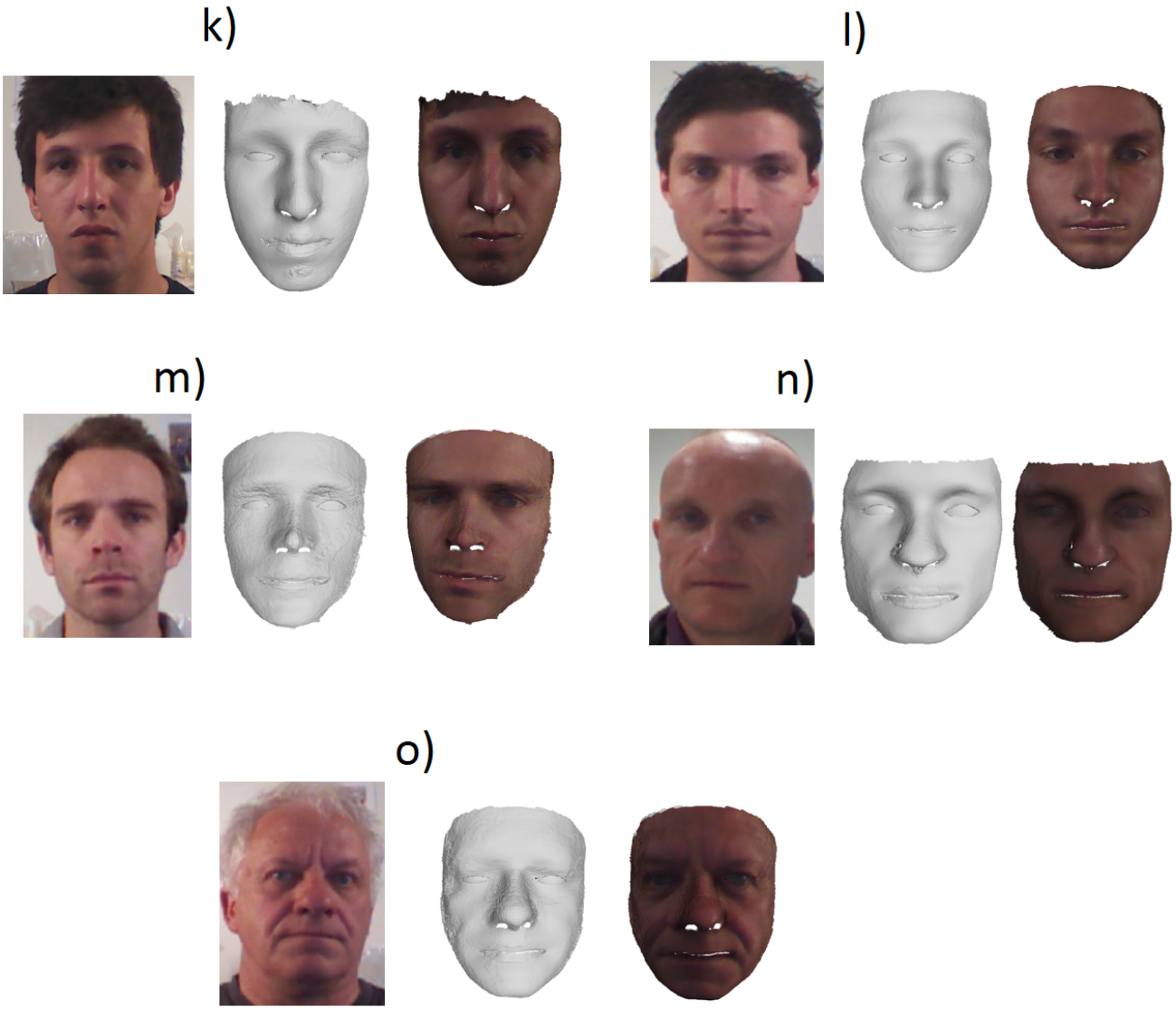


FIGURE IV.3.7: Résultats de notre méthode de reconstruction de texture pour quinze personnes. Notre méthode fonctionne correctement pour tous les sujets. Les caractéristiques physiques des personnes sont représentées sur la texture.

Chapitre IV.4

Conclusion

Dans cette partie, nous avons présenté les résultats que nous avons obtenus avec notre méthode de clonage de visage 3D. Nous avons d'abord décrit notre protocole expérimental. Puis nous avons détaillé les résultats de notre méthode de reconstruction de la forme du visage. Puis pour finir, nous avons montré les résultats obtenus avec notre méthode de reconstruction de la texture du visage. Notre méthode de reconstruction de la forme du visage permet d'obtenir des résultats précis et sémantiques. Nous avons comparé notre méthode avec 2 méthodes de l'état de l'art [8, 41] en utilisant une vérité terrain. Nous avons pu observer que notre méthode permettait d'obtenir les meilleurs résultats. En effet, notre clone est sémantique et possède les spécificités de la personne.

Notre méthode de reconstruction de texture permet d'avoir une texture précise, de haute résolution et avec les spécificités la personne. Nous avons comparé notre méthode avec 2 méthodes de l'état de l'art [41, 45]. Nous avons pu montrer que notre méthode était meilleure pour reconstruire les spécificités de la personne et qu'elle était plus robuste aux mauvaises conditions d'éclairages. Pour finir, nous avons montré la robustesse de notre méthode et nous avons aussi montré l'utilité des patchs de texture.

Cinquième partie

Conclusion de la thèse

Dans cette section, nous réalisons le bilan de nos travaux (caractéristiques, limites...) et nous discutons des perspectives futures. Tout d'abord, nous détaillons comment nous avons effectué les différents choix au cours de cette thèse. Puis, nous présentons les contributions que nous avons réalisées et les différents résultats obtenus. Pour finir, nous décrivons les limites de notre méthode et nous nous projetons dans l'avenir en discutant des différentes perspectives futures.

Sommaire

V.1.1 Bilan de nos travaux

Dans cette première section, nous réalisons le bilan de nos travaux décrit dans ce manuscrit. Nous avons présenté une nouvelle méthode automatique de clonage de visage qui répond au cahier des charges détaillé dans la section *Les contraintes et notre proposition*. Tout d'abord, nous expliquons les contributions et ensuite les limites de notre méthode.

Contributions de notre méthode

Notre méthode permet de reconstruire la forme et la texture d'un visage à partir d'un capteur à faible coût. Elle est complètement automatique et permet d'obtenir un clone sémantique. Nos différentes contributions ont permis d'améliorer les résultats et de résoudre les problèmes que nous avons rencontrés. Ces contributions sont divisées en 2 parties : celle sur la reconstruction de la forme et celle sur la reconstruction de la texture.

Dans notre méthode de reconstruction de la forme, nous utilisons un modèle déformable de visage 3D qui permet d'obtenir un clone 3D sémantique. La principale contribution de notre méthode est la conservation **des spécificités morphologiques de l'individu** en utilisant un modèle déformable de visage. Les 2 originalités sont l'inversion du système (*fitting* puis fusion des données) et l'utilisation de patchs de forme.

L'inversion du système permet d'être moins dépendant des erreurs d'alignements et de *fitting* parce que nous ne fusionnons que des informations fiables du visage. De plus, dans un modèle déformable de visages, les spécificités physiques des visages de la base de test sont corrélées. C'est-à-dire qu'il n'est parfois pas possible de retrouver 2 spécificités sur le maillage 3D si elles ne sont pas corrélées. En effectuant, plusieurs étapes de *fitting* (une sur chaque trame de profondeur), nous pouvons retrouver plus facilement 2 spécificités non corrélées.

La deuxième originalité de notre méthode de reconstruction de la forme est **l'utilisation de patchs de forme**. Premièrement ces patchs permettent d'éliminer les zones des maillages qui ne contiennent pas d'informations sur le visage. Chaque trame contient du bruit et des mauvaises informations (bord...). Ces mauvaises informations sont éliminés. De plus, ils permettent de mettre l'accent sur les détails des spécificités du visage. Nous utilisons une distance d'erreur et la direction des vecteurs normaux de chaque point du nuage 3D pour détecter les parties qui sont pertinentes. C'est pourquoi, notre approche permet de trouver les spécificités des personnes qui ne se retrouvent pas avec un procédé classique de *fitting* [8].

Dans notre méthode de reconstruction de la texture, nous *warpons* sur un clone sémantique la texture de plusieurs trames de texture. La principale contribution est que **nous reconstruisons automatiquement une texture dépliée précise et détaillée avec un capteur RVB-Z à faible coût**. La première originalité est **l'utilisation de patchs pour préserver les caractéristiques de la personne et résoudre le problème des coutures**. Ils permettent de ne conserver que les parties adéquates des trames de texture. Nous utilisons une distance d'erreur et les vecteurs normaux des points des trames de profondeur correspondantes. Notre méthode de fusion des

patchs permet donc de reconstruire une texture sans coutures. Notre deuxième originalité est **l'utilisation de la forme et de la texture pour effectuer l'alignement de ses différents patchs de texture**.

Limites de notre méthode

Nous avons pu voir dans la partie résultat de ce manuscrit (voir chapitre IV) que notre méthode de clonage a des limites de fonctionnement. En effet, notre méthode permet d'avoir de bons résultats, mais certaines spécificités physiques sont plus difficiles à retrouver (moustache,...). De plus, un visage avec une barbe peut entrainer des erreurs d'alignement et donc du bruit sur le clone. De plus, des mauvaises conditions d'éclairages (de côté) peuvent entrainer l'apparition de coutures. C'est pourquoi, dans la section suivante, nous présentons les travaux futurs à réaliser pour améliorer notre méthode.

V.1.2 Perspectives

Dans cette section, nous présentons les travaux futurs à réaliser pour augmenter les performances de notre méthode. Plusieurs axes peuvent être envisagés pour nos travaux futurs. Les méthodes de super-résolution (V.1.2.1) permettent d'augmenter de façon non négligeable la résolution des cartes de profondeur et de texture des capteurs RVB-Z. En effet, le principal problème des données de ce genre de capteur est leur faible résolution. De plus, l'utilisation de la couleur (V.1.2.2) pour la partie reconstruction de la forme peut aussi permettre d'améliorer les résultats. Enfin, l'utilisation d'un algorithme ICP non rigide (V.1.2.3) peut augmenter le réalisme des résultats (nombre de spécificités).

V.1.2.1 Super résolution

Le principal inconvénient des caméras du type Kinect est la faible résolution de leurs données. En effet, les trames de profondeur ne sont pas précises au niveau des yeux (pas de paupière...) et de la bouche (les lèvres ne sont pas marquées). C'est pourquoi, il est difficile de reconstruire correctement ces zones du visage même en utilisant un modèle déformable. Il existe des méthodes dans la littérature pour augmenter la résolution de cartes de profondeur [118, 119]. Schuon et al [118] utilisent un ensemble de cartes de profondeur basse résolution d'une scène statique pour reconstruire une carte de profondeur haute résolution de cette scène. Langmann et al [119] comparent plusieurs méthodes de super résolution dans leur article. L'utilisation de ce type de méthodes sur les trames de profondeur pour augmenter leur résolution peut permettre d'augmenter la qualité de nos résultats. L'augmentation de la résolution des cartes de texture peut aussi permettre d'améliorer notre méthode de reconstruction de la texture. Yang et al [120] présentent une méthode pour augmenter la résolution d'une image RVB. Ils utilisent un

dictionnaire de patchs (zones de plusieurs pixels d'une image) construit à partir de plusieurs images RVB de haute résolution pour augmenter la résolution de l'image traitée.

V.1.2.2 Utilisation de la couleur

Pour pouvoir améliorer les résultats de notre méthode de reconstruction de la forme 3D, il est possible d'utiliser la couleur dans l'étape de *fitting* (voir section III.1.1). L'idée est de boucler notre système de clonage de visage. C'est-à-dire qu'après avoir reconstruit la forme et la texture du visage, il faut utiliser la texture qui a été reconstruite pour affiner la reconstruction de la forme 3D. La couleur peut permettre notamment d'améliorer les étapes de transformations rigides et non rigides du *fitting*.

V.1.2.3 Transformation non rigide

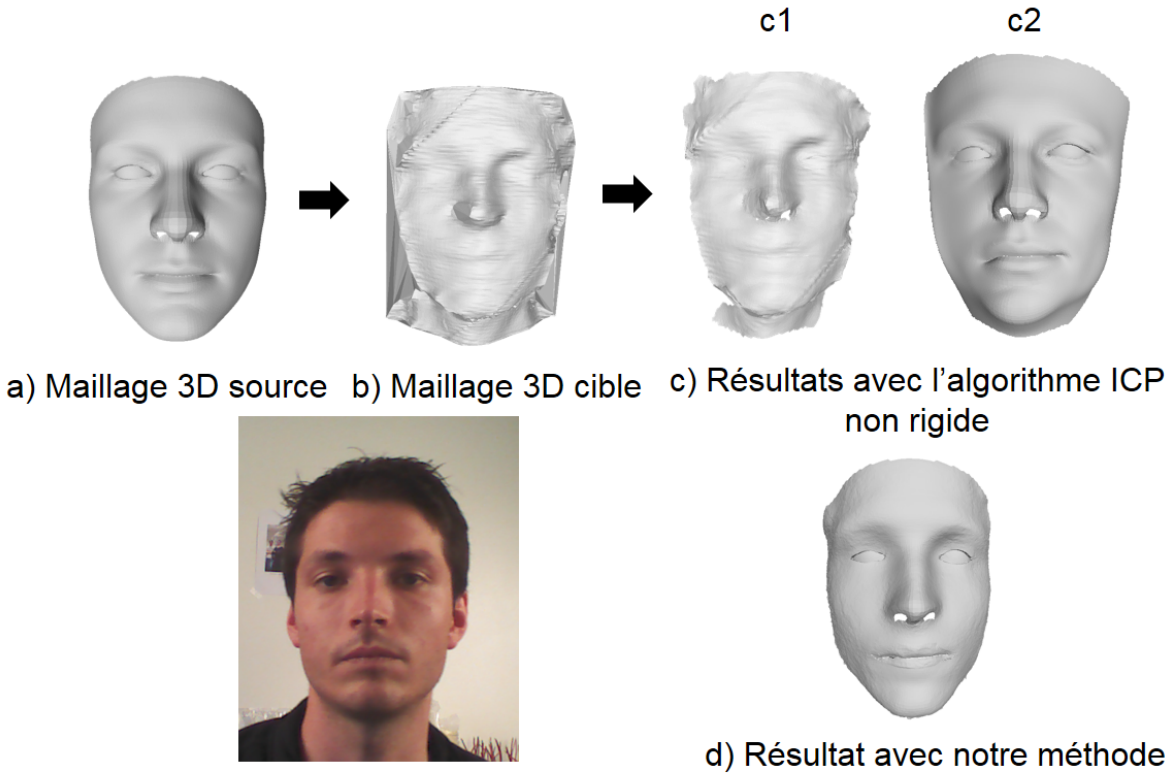


FIGURE V.1.1: Cette figure montre deux résultats obtenus avec l'ICP non rigide (c1 et c2). Le maillage 3D source (a : maillage moyen du modèle déformable) est déformé pour s'adapter à la forme du maillage 3D cible (b : trame de profondeur de la caméra Kinect). Dans le premier cas, le clone c1 a été trop déformé. Contrairement au clone obtenu avec notre méthode (d), il possède du bruit et des trous de la trame de profondeur. Dans le deuxième cas, le clone c2 ne s'est pas assez déformé. En effet, sa forme 3D est proche de la forme du maillage 3D source.

Dans nos travaux futurs, nous pourrons intégrer un algorithme d'alignement non rigide à notre méthode de reconstruction de forme 3D. Cet algorithme permet de déformer un maillage

3D pour qu'il s'adapte plus précisément à un autre maillage 3D. Il est très efficace et permet d'obtenir des résultats très performants. L'ICP non rigide que nous avons testé est composé de 2 parties : la transformation globale et la transformation locale. Dans l'étape de transformation globale, une translation est appliquée à tous les points du nuage source. Mais contrairement à l'ICP rigide, cette translation est différente pour chaque point du nuage. Dans l'étape de transformation locale, une transformation rigide est appliquée sur certaines zones du nuage de points (plusieurs points 3D). C'est-à-dire que l'on applique sur tous les points de la zone sélectionnée, la même transformation (translation et rotation). La figure V.1.1 montre 2 résultats extrêmes obtenus avec cet algorithme (V.1.1 : c1 et c2) : V.1.1 : (c1) s'est trop déformé (du bruit et des trous dans le maillage) et V.1.1 : (c2) pas assez (proche du maillage moyen du modèle déformable). On constate que pour que l'algorithme puisse être utilisé dans une méthode de clonage, il faut que les paramètres de cet algorithme soient réglés précisément. Un autre point sur lequel il faut être vigilant est de garder la sémantique. La figure V.1.2 montre que le clone V.1.1 : (c1) n'est plus un clone sémantique (nez, front). En effet, on peut voir que le point d'indice 2000 (sur le côté du nez) est maintenant situé sur le bout du nez.

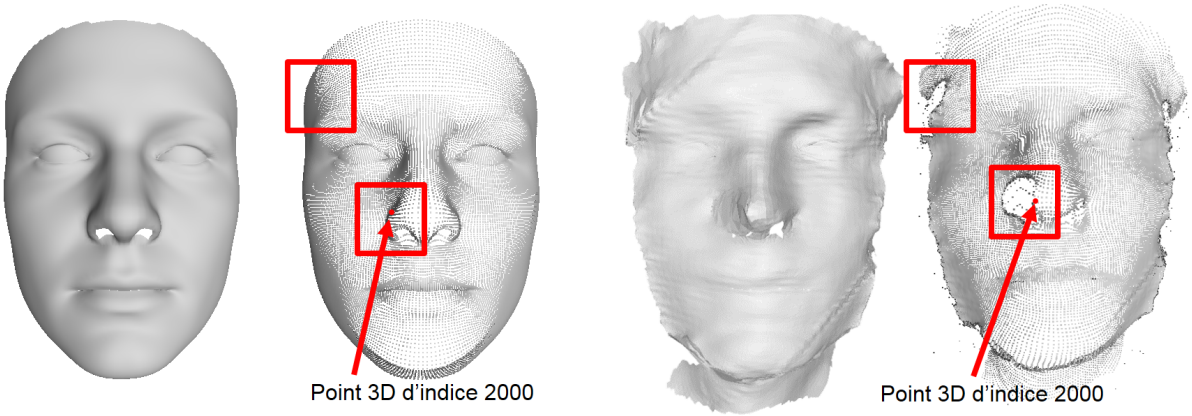


FIGURE V.1.2: Cette figure montre que le résultat (b) obtenu avec l'ICP non rigide n'est plus sémantique. Contrairement au maillage 3D source (a : maillage moyen du modèle déformable), qui est un maillage sémantique, certains points 3D du maillage après ICP non rigide (b : nez et front) ne correspondent plus à la même zone du visage (point d'indice 2000).

Publications

Publications dans des revues internationales avec comité de lecture :

Jérôme Manceau, Catherine Soladié, Renaud Seguier, *Patches Detection and Fusion for 3D face Cloning*, In Advances in Image and Video Processing, Society for Science and Education, United Kingdom.

Publications dans des conférences internationales avec comité de lecture et proceedings :

Jérôme Manceau, Catherine Soladié, Renaud Seguier, *Reconstruction of face texture based on the fusion of texture patches*, International Symposium on Visual Computing, Dec 2015, Las Vegas, United States.

Jérôme Manceau, Catherine Soladié, Renaud Seguier, *3D Facial clone based on depth patches*, IEEE International Conference on Visual Communication and Image Processing - VCIP 2015, Dec 2015, Singapore.

Publications dans des conférences nationales avec comité de lecture et proceedings :

Jérôme Manceau, Catherine Soladié, Renaud Seguier, *Reconstruction d'un clone de visage 3D à partir de patchs forme*, Colloque GRETSI 2015, Sep 2015, Lyon, France.

Bibliographie

[1] Ralph Gross, Iain Matthews, and Simon Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(11) :1080–1093, November 2005.

[2] Catherine Soladie, Nicolas Stoiber, and Renaud Séguier. Invariant representation of facial expressions for blended expression recognition on unknown subjects. *Computer Vision and Image Understanding*, 117(11) :1598–1609, November 2013.

[3] Aleksander Väljamäe, Pontus Larsson, Daniel Västfjäll, and Mendel Kleiner. Auditory presence, individualized head-related transfer functions, and illusory ego-motion in virtual environments. *Proc. of 7th Annual Workshop Presence*, 2004.

[4] Kenneth Alberto Funes Mora and Jean-Marc Odobez. Gaze estimation from multimodal kinect data. In *IEEE Conference in Computer Vision and Pattern Recognition, Workshop on Gesture Recognition*, June 2012.

[5] Paul Debevec. The Light Stages and Their Applications to Photoreal Digital Actors. In *SIGGRAPH Asia*, Singapore, November 2012.

[6] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In Stefano Tubaro and Jean-Luc Dugelay, editors, *AVSS*, pages 296–301. IEEE Computer Society, 2009.

[7] Satyadhyan Chickerur and Kartik Joshi. 3d face model dataset : Automatic detection of facial expressions and emotions for educational environments. *British Journal of Educational Technology*, 46(5) :1028–1037, 2015.

[8] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse : A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3) :413–425, March 2014.

[9] X.C. He, S.C. Yuk, K.P. Chow, K.K. Wong, and R.H.Y. Chung. Super-resolution of faces using texture mapping on a generic 3d model. In *Image and Graphics, 2009. ICIG '09. Fifth International Conference on*, pages 361–365, Sept 2009.

[10] Mohamed Daoudi, Srivastava Anuj, and Remco Veltkamp. *3D Face Modeling, Analysis and Recognition*. Wiley, June 2013.

[11] Karima Ouji. *Numérisation 3D de visages par une approche de super-résolution spatio-temporelle non-rigide*. Theses, Ecole Centrale de Lyon, June 2012.

[12] Benjamin Loriot. *Automation of Acquisition and Post-processing for 3D Digitalisation*. Theses, Université de Bourgogne, March 2009.

[13] Olivier Faugeras. *Three-dimensional Computer Vision : A Geometric Viewpoint*. MIT Press, Cambridge, MA, USA, 1993.

[14] Bellmann Anke, Hellwich Olaf, Rodehorst Volker, and Yilmaz Ulas. A benchmark dataset for performance evaluation of shape-from-x algorithms, 2008.

[15] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3) :7–42, April 2002.

[16] C. Zeller. *Calibration projective, affine et euclidienne en vision par ordinateur et apllication à la perception tridimensionelle*. INRIA, 1996.

[17] V Rodin and A Ayache. Stéréovision axiale : modélisation et calibrage du système de prises de vues, reconstruction 3d d'objets naturels. *Traitement du Signal*, 11(5), 1994.

[18] Yuichi Ohta and Takeo Kanade. Stereo by intra - and inter - scanline search using dynamic programming. Technical Report CMU-CS-83-162, Carnegie-Mellon University.Computer science. Pittsburgh (PA US), 1983.

[19] H. Hirschmuller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9) :1582–1599, Sept 2009.

[20] Sylvie Chambon. *Color stereo matching with occlusions*. Theses, Université Paul Sabatier - Toulouse III, December 2005.

[21] T. Jebara, A. Azarbayejani, and A. Pentland. 3d structure from 2d motion. *Signal Processing Magazine, IEEE*, 16(3) :66–84, May 1999.

[22] P. Fua. Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data. *Int. J. Comput. Vision*, 38 :153–171, July 2000.

[23] J. Lee, B. Moghaddam, H. Pfister, and R. Machiraju. Silhouette-based 3D face shape recovery. *Graphics Interface*, 2003.

[24] Giovanna Sansoni, Marco Trebeschi, and Franco Docchio. State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors*, 9(1) :568–601, 2009.

[25] Yu-ichi Ohta, Kiyoshi Maenobu, and Toshiyuki Sakai. Obtaining surface orientation from texels under perspective projection. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 746–751, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[26] Jenn-Kwei Tyan. *Analysis and application of autofocusing and three-dimensional shape recovery techniques based on image focus and defocus*. PhD thesis, State University of New York at Stony Brook, 1997.

[27] Yalin Xiong and Steven Shafer. Depth from focusing and defocusing. Technical Report CMU-RI-TR-93-07, Robotics Institute, Pittsburgh, PA, March 1993.

[28] B. K.P. Horn. Shape from shading : A method for obtaining the shape of a smooth opaque object from one view. Technical report, Cambridge, MA, USA, 1970.

[29] R VAILLANT and I SURIN. Reconstruction de visages par stéréovision active. *TS. Traitement du signal*, 12(2) :201–211, 1995.

[30] Joaquim Salvi, Jordi Pagès, and Joan Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37 :827–849, 2004.

[31] Josep Forest Collado et al. *New methods for triangulation-based shape acquisition using laser scanners*. Universitat de Girona, 2004.

[32] Norbert Haala and Claus Brenner. Generation of 3d city models from airborne laser scanning data. In *Proceedings EARSEL workshop on LIDAR remote sensing on land and sea*, pages 105–112, 1997.

[33] H. G. Maas. The Suitability For Airborne Laser Scanner Data For Automatic 3D Object Reconstruction. In *Ascona01*, 2001.

[34] Yan Cui, Sebastian Schuon, Sebastian Thrun, Didier Stricker, and Christian Theobalt. Algorithms for 3d shape scanning with a depth camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1039–1050, 2013.

[35] J. I. San José Alonso, J. Martínez Rubio, J. J. Fernández Martín, and J. García Fernández. Comparing time-of-flight and phase-shift. the survey of the royal pantheon in the basilica of san isidoro (leÓn). *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-5/W16 :377–385, 2011.

[36] Didier Gava. *Vision conoscopique 3D : Calibration et reconstruction*. Theses, Université René Descartes - Paris V, June 1998.

[37] Theis P Hansen, Jes Broeng, Stig EB Libori, Erik Knudsen, Anders Bjarklev, Jacob Riis Jensen, and Harald Simonsen. Highly birefringent index-guiding photonic crystal fibers. *Photonics Technology Letters, IEEE*, 13(6) :588–590, 2001.

[38] Arman Savran, Neşe Alyüz, Hamdi Dibeklioğlu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. Biometrics and identity management. chapter Bosphorus Database for 3D Face Analysis, pages 47–56. Springer-Verlag, Berlin, Heidelberg, 2008.

[39] Jiahui Pan, Peisen S Huang, Song Zhang, and Fu-Pen Chiang. Color n-ary gray code for 3-d shape measurement. *proceedings of ICEM, Italy*, 2004.

[40] A. Lathuilière. *Génération de mires colorées pour la reconstruction 3D couleur par système stéréoscopique de vision active*. 2007.

[41] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion : Real-time dense surface mapping and tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 127–136, Washington, DC, USA, 2011. IEEE Computer Society.

[42] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion : Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[43] A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-flight sensors in computer graphics (state-of-the-art report), 2009.

[44] Qi Sun, Yanlong Tang, Ping Hu, and Jingliang Peng. Kinect-based automatic 3d high-resolution face modeling. In *Image Analysis and Signal Processing (IASP), 2012 International Conference on*, pages 1–4, Nov 2012.

[45] M. Hernandez, Jongmoo Choi, and G. Medioni. Laser scan quality 3-d face modeling using a low-cost depth camera. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1995–1999, Aug 2012.

[46] Sebastian Schuon, Christian Theobalt, James Davis, and Sebastian Thrun. Lidarboost : Depth superresolution for tof 3d shape scanning. In *CVPR'09*, pages 343–350, 2009.

[47] Kim T Gribbon and Donald G Bailey. A novel approach to real-time bilinear interpolation. In *Field-Programmable Technology, 2004. Proceedings. 2004 IEEE International Conference on*, pages 126–131. IEEE, 2004.

[48] Glen A Hansen, Rod W Douglass, and Andrew Zardecki. *Mesh enhancement : selected elliptic methods, foundations and applications*. Imperial College Press, 2005.

[49] Lance Williams. Performance-driven facial animation. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '90, pages 235–242, New York, NY, USA, 1990. ACM.

[50] Yuping Lin, Gérard Medioni, and Jongmoo Choi. Accurate 3d face reconstruction from weakly calibrated wide baseline images with profile contours. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1490–1497. IEEE, 2010.

[51] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2) :239–256, February 1992.

[52] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision*, 13(2) :119–152, October 1994.

[53] Toru Tamaki, Miho Abe, Bisser Raytchev, and Kazufumi Kaneda. Softassign and em-icp on gpu. *2013 International Conference on Computing, Networking and Communications (ICNC)*, 0 :179–183, 2010.

[54] Andriy Myronenko, Xubo Song, and Á. Carreira-Perpiñán. Non-rigid point set registration : Coherent point drift (cpd). In *In Advances in Neural Information Processing Systems 19*. MIT Press, 2006.

[55] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996.

[56] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.

[57] Scott D Roth. Ray casting for modeling solids. *Computer graphics and image processing*, 18(2) :109–144, 1982.

[58] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[59] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9) :1063–1074, September 2003.

[60] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics : Principles and Practice (2Nd Ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[61] Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12) :1615 – 1618, December 2003.

[62] P. Jonathon Phillips, Hyeonjoon Moon, Patrick Rauss, and Syed A. Rizvi. The feret evaluation methodology for face-recognition algorithms. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 137–, Washington, DC, USA, 1997. IEEE Computer Society.

[63] Michael Zollhofer, Michael Martinek, Gunther Greiner, Marc Stamminger, and Jochen SuBmuth. Automatic reconstruction of personalized avatars from 3d face scans. *Comput. Animat. Virtual Worlds*, 22(2-3) :195–202, April 2011.

[64] Michael Zollhofer, Justus Thies, Matteo Colaianni, Marc Stamminger, and Gunther Greiner. Interactive model-based reconstruction of the human head using an rgb-d sensor. *Computer Animation and Virtual Worlds*, 25(3-4) :213–222, 2014.

[65] Stéphane Valente. *Analyse synthese et animation de clones dans un contexte de telereunion virtuelle*. PhD thesis, Ecole Polytechnique federal de Lausanne, 1999.

[66] Kristina Scherbaum, Martin Sunkel, H-P Seidel, and Volker Blanz. Prediction of individual non-linear aging trajectories of faces. In *Computer Graphics Forum*, volume 26, pages 285–294. Wiley Online Library, 2007.

[67] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models&mdash ;their training and application. *Comput. Vis. Image Underst.*, 61(1) :38–59, January 1995.

[68] J.C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1) :33–51, 1975.

[69] Gary Bradski and Adrian Kaehler. *Learning OpenCV : Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[70] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, June 2001.

[71] Gene H Golub, Per Christian Hansen, and Dianne P O'Leary. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1) :185–194, 1999.

[72] Jin Huang, Xiaohan Shi, Xinguo Liu, Kun Zhou, Li-Yi Wei, Shang-Hua Teng, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Subspace gradient domain mesh deformation. *ACM Transactions on Graphics (TOG)*, 25(3) :1126–1134, 2006.

[73] Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on Graphics (TOG)*, 23(3) :399–405, 2004.

[74] David A Leopold, Alice J O'Toole, Thomas Vetter, and Volker Blanz. Prototype-referenced shape encoding revealed by high-level aftereffects. *Nature neuroscience*, 4(1) :89–94, 2001.

[75] Junyi Zhang and Shuqian Luo. image-based texture mapping method in 3d face modeling. In *Complex Medical Engineering, 2007. CME 2007. IEEE/ICME International Conference on*, pages 147–150, May 2007.

[76] Lin Xu, E. Li, Jianguo Li, Yurong Chen, and Yimin Zhang. A general texture mapping framework for image-based 3d modeling. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2713–2716, Sept 2010.

[77] Jinkyu Hwang, Sunjin Yu, Joongrock Kim, and Sangyoun Lee. 3d face modeling using the multi-deformable method. 2012.

[78] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[79] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6) :681–685, 2001.

[80] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 35–42. ACM, 1992.

[81] W.-S Lee and N Magnenat-Thalmann. Fast head modeling for animation. *Image and Vision Computing*, 18(4) :355 – 364, 2000.

[82] V. Lempitsky and D. Ivanov. Seamless mosaicing of image-based texture maps. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–6, June 2007.

[83] Todor Georgiev. Covariant derivatives and vision. In *Computer Vision–ECCV 2006*, pages 56–69. Springer, 2006.

[84] Peter J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. Graph.*, 2(4) :217–236, October 1983.

[85] Peter J Burt and Edward H Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)*, 2(4) :217–236, 1983.

[86] Shamik Sural, Gang Qian, and Sakti Pramanik. Segmentation and histogram generation using the hsv color space for image retrieval. In *in International Conference on Image Processing (ICIP). 2002 : p. 589-592. VIIth Digital Image Computing : Techniques and Applications, Sun C., Talbot H., Ourselin*, pages 589–592, 2002.

[87] Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss. Seamless image stitching in the gradient domain. In *In Proceedings of the European Conference on Computer Vision*, 2006.

[88] Yun Ge, Baicai Yin, Yanfeng Sun, and Hengliang Tang. 3d face texture stitching based on poisson equation. In *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on*, volume 2, pages 809–813, Oct 2010.

[89] Arnaud Desssein, William A.P. Smith, Richard C. Wilson, and Edwin R. Hancock. *Seamless texture stitching on a 3D mesh by Poisson blending in patches*, pages 2031–2035. IEEE, 2014.

[90] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Comput. Graph. Appl.*, 21(5) :34–41, September 2001.

[91] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. *ACM Trans. Graph.*, 21(3) :277–280, July 2002.

[92] G. Peyré and L. D. Cohen. Geodesic remeshing using front propagation. *International Journal of Computer Vision*, 69(1) :145–156, 2006.

[93] Nobuyuki Bannai, Robert B Fisher, and Alexander Agathos. Multiple color texture map fusion for 3d models. *Pattern Recognition Letters*, 28(6) :748–758, 2007.

[94] Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression (pie) database. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 46–51. IEEE, 2002.

[95] Thibaut Weise, Bastian Leibe, and Luc Van Gool. Accurate and robust registration for in-hand modeling. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[96] Hu Han and Anil K Jain. 3d face texture modeling from uncalibrated frontal and profile images. In *Biometrics : Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pages 223–230. IEEE, 2012.

[97] David C. Schneider and Peter Eisert. Fitting a morphable model to pose and shape of a point cloud. In Marcus A. Magnor, Bodo Rosenhahn, and Holger Theisel, editors, *VMV*, pages 93–100. DNB, 2009.

[98] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[99] Frédéric Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, and David H. Salesin. Synthesizing realistic facial expressions from photographs. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 75–84, New York, NY, USA, 1998. ACM.

[100] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. *ACM Trans. Graph.*, 22(3) :950–953, July 2003.

[101] Kok lim Low. Linear least-squares optimization for point-toplane icp surface registration. Technical report, 2004.

[102] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[103] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2) :137–154, 2004.

[104] Lounis Douadi, Marie-José Aldon, and André Crosnier. Pair-wise registration of 3d/color data sets with ICP. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2006, October 9-15, 2006, Beijing, China*, pages 663–668, 2006.

[105] Takeshi Masuda and Naokazu Yokoya. A robust method for registration and segmentation of multiple range images. *Computer Vision and Image Understanding*, 61(3) :295–307, 1995.

[106] Seung-Hwan Kim, Dong-O Kim, Sang Wook Lee, and Rae-Hong Park. Reducing computation time for range image registration using radial-distance down-sampling. *FCV2004*, 2004.

[107] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

[108] Timothée Jost. Fast geometric matching for shape registration. *PhD report, Faculté des Sciences de l'Université de Neuchâtel*, 2002.

[109] Yung Chen and Gérard Medioni. Object modeling by registration of multiple range images. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 2724–2729. IEEE, 1991.

[110] G.C. Sharp, S.W. Lee, and D.K. Wehe. Icp registration using invariant features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(1) :90–102, Jan 2002.

[111] Jason Luck, Charles Little, and William Hoff. Registration of range data using a hybrid simulated annealing and iterative closest point algorithm. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 4, pages 3739–3744. IEEE, 2000.

[112] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (5) :698–700, 1987.

[113] Olivier D Faugeras and Martial Hebert. The representation, recognition, and locating of 3-d objects. *The international journal of robotics research*, 5(3) :27–52, 1986.

[114] Michael W Walker, Lejun Shao, and Richard A Volz. Estimating 3-d location parameters using dual number quaternions. *CVGIP : image understanding*, 54(3) :358–367, 1991.

[115] Andrew W Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(13) :1145–1153, 2003.

[116] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam.

[117] Agisoft photoscan : an advanced image-based 3d modeling solution for creating professional quality 3d content from still images.

[118] Sebastian Schuon, Christian Theobalt, James Davis, and Sebastian Thrun. Lidarboost : Depth superresolution for tof 3d shape scanning. *In Proc. of IEEE CVPR 2009*, 2009.

[119] B. Langmann, K. Hartmann, and O. Loffeld. Comparison of depth super-resolution methods for 2d/3d images. *International Journal of Computer Information Systems and Industrial Management Applications*, 3 :635 –645, 2011.

[120] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

Table des figures

Liste des tableaux

Résumé

Aujourd'hui, de nombreuses applications utilisent des clones 3D de visage. En effet, ils peuvent être utilisés comme prétraitements dans de nombreuses méthodes de synthèse (immersion...) et d'analyse d'un visage (analyse d'émotion...). Cette thèse porte sur le clonage réaliste de visages à partir d'une caméra RVB-Z basse-résolution.

Pour pouvoir être utilisés dans ces applications, les clones doivent modéliser avec précision la forme du visage, tout en conservant les spécificités des individus. Ils doivent aussi être sémantiques, c'est-à-dire que la position des différentes parties du visage (yeux, nez ...) sur le maillage 3D est connue. La texture du visage doit être précise, sans flou et doit contenir un maximum de spécificités de la personne. Nos travaux portent sur une solution permettant d'obtenir un clone 3D sémantique réaliste.

Nous proposons une approche qui utilise des patchs de forme et de texture pour **préserver les caractéristiques du visage de la personne** et un modèle déformable de visage 3D pour obtenir **des clones sémantiques**. Les patchs sont les parties adéquates de données de profondeur et de texture. Ils sont détectés à partir d'une distance d'erreur et de la direction des vecteurs normaux en chaque point du maillage 3D. Ensuite ces patchs, qui mettent l'accent sur les spécificités de l'individu, sont fusionnés pour reconstruire la forme et la texture complète du visage.

Nous avons comparé notre méthode de reconstruction de la forme 3D et de la texture du visage avec les méthodes de l'état de l'art. Ces tests ont montré que notre approche de reconstruction de la forme est plus performante que les méthodes classiques de *fitting*. En effet, elle permet **d'être plus précise** et **de retrouver plus de spécificités des personnes**. Les tests qualitatifs réalisés avec les méthodes de reconstruction de texture de l'état de l'art ont montré que notre méthode de reconstruction de texture est **robuste** et qu'elle permet de **reconstruire une texture précise, sans couture gardant les spécificités des individus**.

Mots clef Clonage de visage, Maillage sémantique, Détection de patchs, Fusion de patchs, Carte de profondeur, *Fitting*, *Warping* de la texture