

UBO

MASTER SIIA

RAPPORT DE STAGE

**Scan 3D de personnes et
génération d'avatar animé dans
un environnement de réalité
virtuelle**

Auteur

Erwan LE CHEVOIR

Encadrant

Cédric BUCHE

UBO
Université de Bretagne Occidentale



Table des matières

1	Introduction	2
2	Etat de l'art	3
2.1	Anorexie mentale	3
2.2	Projet REVAM	3
2.3	L'immersion via scan 3D	4
2.4	Les techniques d'auto-rig	5
2.4.1	Rig guidé	6
2.4.2	Rig automatique par mesh discrétisation	6
2.4.3	Rig automatique par raycast	7
3	Contribution	8
3.1	le scanner 3D	8
3.2	L'auto-rig	8
3.2.1	La solution du site Mixamo	9
3.2.2	l'algorithme Pinocchio	9
3.2.3	Le logiciel Smart body	9
3.3	L'importateur de fichier .dae	11
3.4	Morphing et utilisation du scanner 3D	12
3.4.1	Le Morphing par interpolation linéaire des points	12
3.4.2	Les blendshapes: la techniques des jeux vidéo	12
3.4.3	Multirésolution Interpolation	13
3.4.4	MAPS (Multiresolution Adaptive Parameterization of Surfaces)	14
3.4.5	MIMesh (Multiresolution Interpolation Mesh)	14
3.4.6	Bilan	16
3.5	Gestion des textures	17
4	Conclusion	18
5	Annexe	19
6	Lexique	20
	Références	20

1 Introduction

Ce document retrace le stage de master SIIA effectué au Centre Européen de réalité Virtuelle). J'ai travaillé sur le projet REVAM (Réalité Virtuelle et anorexie Mentale) qui est une collaboration entre le CERV et le CHU de Bohars (BREST). Il a débuté en 2015 à l'initiative de Nathalie LE BIGOT (enseignante-chercheuse en psychologie cognitive), Cédric BUCHE (enseignant-chercheur en informatique) et Stéphanie MONTAVON-LE MEAUX (Dr psychiatre spécialiste de l'anorexie mentale).Le projet est une alliance de la psychologie et la réalité virtuelle afin d'aider les patients atteints de trouble de l'anorexie mentale.

Cette entreprise a pour but de délivrer une application utilisable par des psychiatres. Pour ce faire nous utilisons le moteur Unity ainsi qu'un casque de réalité virtuelle.

Le but de ce stage était d'implanter un avatar à partir d'un scan 3D dans l'application REVAM en fonction des contraintes du projet. Dans un premier temps nous exposerons le processus actuel du projet REVAM et les améliorations que nous allons y apporter.

Par la suite nous présenterons la réalisation de la mise en place de ces dernières , les différentes recherches que nous avons effectuées compte tenu des contraintes ci-avant. Et enfin, nous étudierons les pistes d'évolution pour de futures mise à jour.

2 Etat de l'art

2.1 Anorexie mentale

Ce projet de réalité virtuelle a pour but d'aider les personnes atteintes d'anorexie mentale. L'anorexie mentale fait partie de la catégorie des troubles du comportement alimentaire qui entraîne une privation alimentaire stricte et volontaire. En effet, les personnes souffrant d'anorexie ont tendance à surévaluer leur silhouette corporelle les amenant souvent à adopter un régime alimentaire très réduit. De ce fait, si il n'y a pas de prise en charge du patient, le pronostic vitale des malades peut être engagé. Selon l'INSERM [4], 5% des patients décèdent des suites de ce trouble.

L'anorexie mentale touche particulièrement les jeunes filles lors de l'adolescence. Il n'y aurait pas de cause déterminante au déclenchement, ce serait plutôt un ensemble de facteurs: génétiques, psychologiques, environnementaux et/ou familiaux. Selon les sujets, une phase anorexique peut aller d'un an et demi à trois ans et dans le pire des cas cinq ans. Afin d'éviter une quelconque complication, il est conseillé de prendre en charge le patient. Cette prise en charge a pour but de faire retrouver un poids correct et d'aider psychologiquement le souffrant. Cependant, les rechutes sont assez fréquentes. Un suivi est donc recommandé.

C'est de cette volonté d'aider ces personnes en souffrance que le projet REVAM est né

2.2 Projet REVAM

Pour cela, REVAM s'appuie sur le phénomène psychologique de sortie de corps évoqué par Lopez.C [3] Le principe de sortie de corps consiste à placer un individu devant un corps virtuel qu'il va s'approprier comme étant le sien grâce à des mécanismes psychologiques couplés à une stimulation physique. En se basant sur l'expérience de B. Lenggenhage et al [6] nous plaçons le sujet dans un environnement virtuel où se trouve devant lui un avatar.

Nous proposons à l'utilisateur de paramétrer cet avatar afin que la silhouette qu'il perçoit corresponde à la vision de son corps. Par la suite, nous appliquons un morphing très lent sur l'avatar afin de l'amaigrir. Pendant ce temps, l'utilisateur peut bouger et l'avatar suivra les mouvements capturés par un kinect. Afin de réaliser une sortie de corps, nous stimulons le sujet

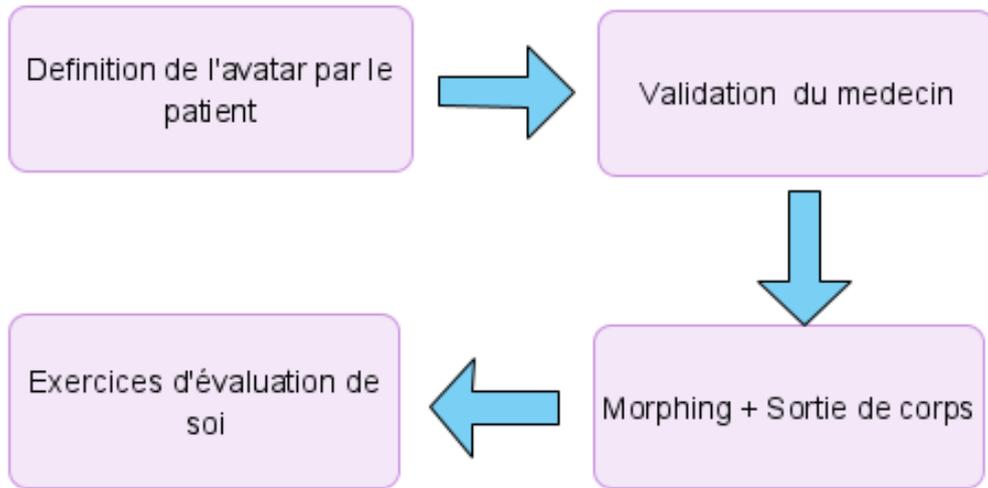


FIG. 2 – *Déroulement de l'application*

grâce à un bâton dont les mouvements sont synchronisés dans l'environnement virtuel. Grâce à la sortie de corps et l'appropriation de l'avatar, l'utilisateur se percevra plus mince que l'image qu'il se représente de lui-même.

Cette application étant destinée à être utilisée par des médecins il est important qu'elle soit assez simple d'utilisation et compréhensive. De ce fait, nous verrons par la suite que certaines options ne sont pas envisageables car elles mobiliseront trop de ressources ou de temps pour être réalisables.

2.3 L'immersion via scan 3D

L'immersion est une partie importante du processus de sortie de corps. En effet, plus l'appropriation de l'avatar sera grande, mieux le phénomène psychologique fonctionnera. Pour renforcer ce sentiment d'appropriation; nous avons décidé d'utiliser un scanner 3D afin de pouvoir modéliser la personne dans l'application. Bien entendu l'utilisation de cette technologie possède quelques contraintes et défis que nous allons devoir résoudre.

Un premier soucis connu sous le nom de phénomène de l'uncanny valley [10] peut survenir. Ce phénomène tiré de la robotique est traduit par la gêne qu'un individu possède devant une entité robotique ou virtuelle atteignant un

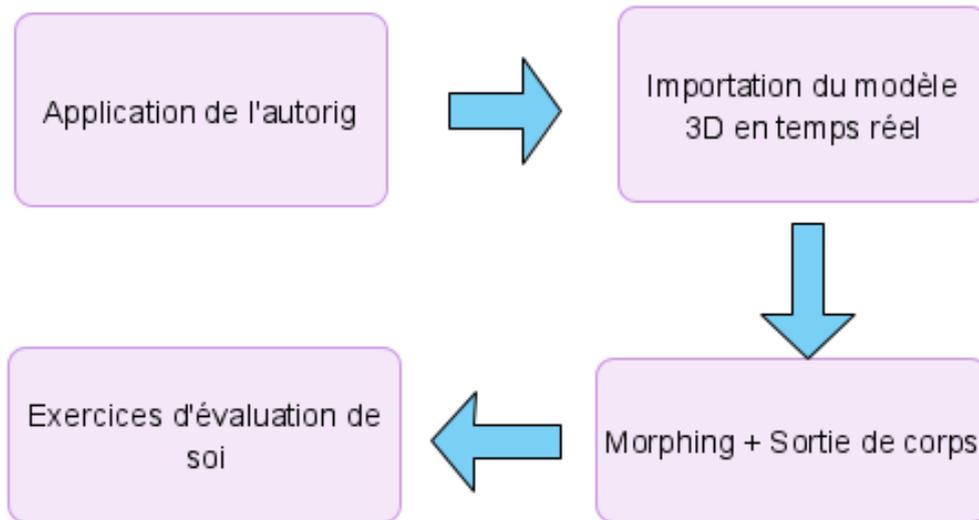


FIG. 3 – Déroulement souhaité de l'application

certain degré de ressemblance anthropomorphe. L'utilisateur remarquera alors tous les défauts de cet entité, lui paraissant aberrante.

Dans notre cas, cet effet pourrait être ressenti au niveau des visages ou encore de certains autres détails ne rendant pas crédible la scène, par exemple des vêtements trop amples sur lesquels la gravité n'aurait aucun effet. Pour pallier cela les avatars sont présentés de dos et nous demanderons aux participants de prendre des vêtements près du corps.

L'une des principales difficultés est d'obtenir un squelette sur le scan au vu de l'utilisation du Kinect. Le scanner 3D ne le génère pas et comme nous l'avons exposé précédemment il n'est pas possible de devoir modifier chaque scan un par un et de leur appliquer un squelette à partir d'un logiciel de 3D du fait de la mobilisation de ressources humaines spécialisées dans le domaine. Nous allons donc devoir trouver un moyen pour générer un squelette automatiquement.

2.4 Les techniques d'auto-rig

Un auto-rig consiste à générer automatiquement un squelette sur un mesh lui permettant de se mouvoir dans un environnement 3D avec des contraintes

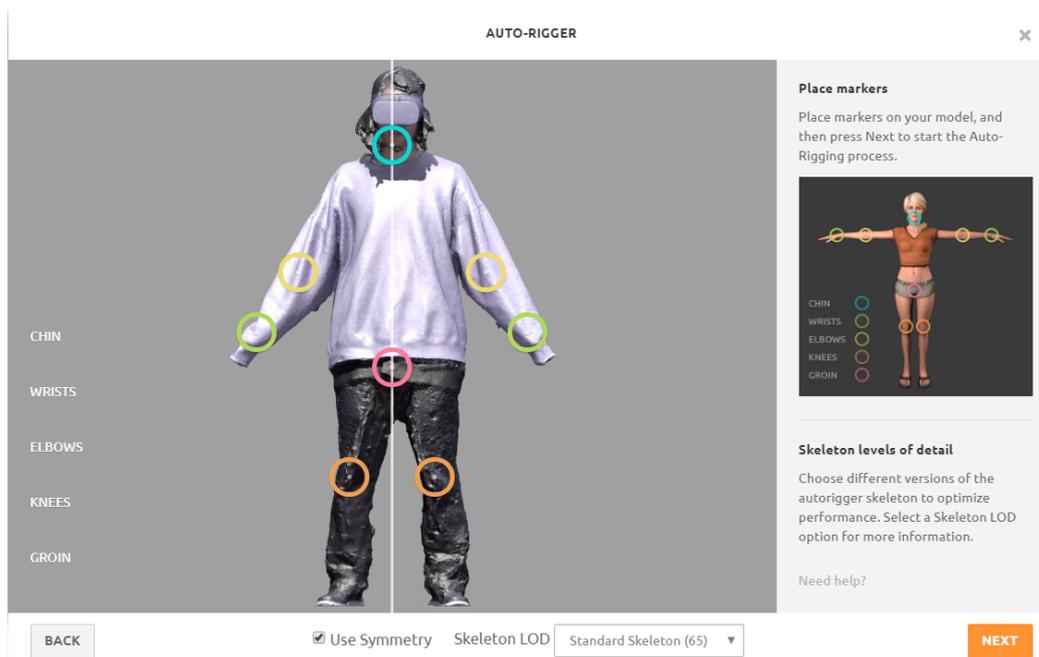


FIG. 4 – Exemple d'utilisation de l'autorig de Mixamo

réalistes. Dans une utilisation normale, les squelettes sont construits “à la main” dans des logiciels 3D tels que Maya ou encore Blender.

Dans notre cas il est impossible de procéder de cette manière, l'application étant destinée à être utilisée par des médecins n'étant pas formés en terme de construction 3D. Il nous faut donc trouver plusieurs manières d'opérer afin de le réaliser.

2.4.1 Rig guidé

Le but de cette technique est de placer des points à des endroits clés (cou, genoux, coudes, mains) qui permettent à l'algorithme de construire le squelette. Elle a pour avantage de rendre un squelette fiable grâce à l'aide humaine apportée. Cette technique est notamment utilisée par le site Mixamo [9]

2.4.2 Rig automatique par mesh discrétisation

C'est une méthode mise en valeur par Ilya Baran et Jovan Popovic [2] connue sous le nom de Pinocchio. Cet algorithme permet de construire un squelette à un mesh complet humanoïde ou quadrupède.

Cette technique se déroule en plusieurs étapes : Une approximation de la surface médiane du mesh Grâce à l'étape précédente, les distances des points vers la surface du mesh sont calculées puis triées du plus loin au plus proche afin de créer des paquets de sphères dont le radius est la distance des points à la surface. Un graph est ensuite construit à partir des sphères en les connectant les unes aux autres si elles possèdent une intersection. Un algorithme de type A* vient ensuite trouver le chemin le plus court sur le graph ce qui nous donne un premier squelette Un affinage est ensuite appliqué pour présenter un meilleur rendu Cette technique donne en général de bons résultats.

Toutefois, il est impératif que le mesh soit unique et complet. En effet, si le mesh donné possède des trous, cet algorithme ne fonctionnera pas. Il conviendra donc porter une attention particulière, car il arrive que les scanner 3D aient des petites imperfections et fournissent un résultat présentant des trous.

2.4.3 Rig automatique par raycast

Le but de cette méthode décrite par Lopez.R et Poirel.C [7] est de réduire la complexité du mesh en envoyant des raycast frontaux et espacés de manière régulière afin d'en extraire une silhouette.

Grâce à cette silhouette, des raycast à 360 sont utilisés pour trouver le barycentre de la partie concernée. Ainsi les points obtenus permettent de créer le squelette.

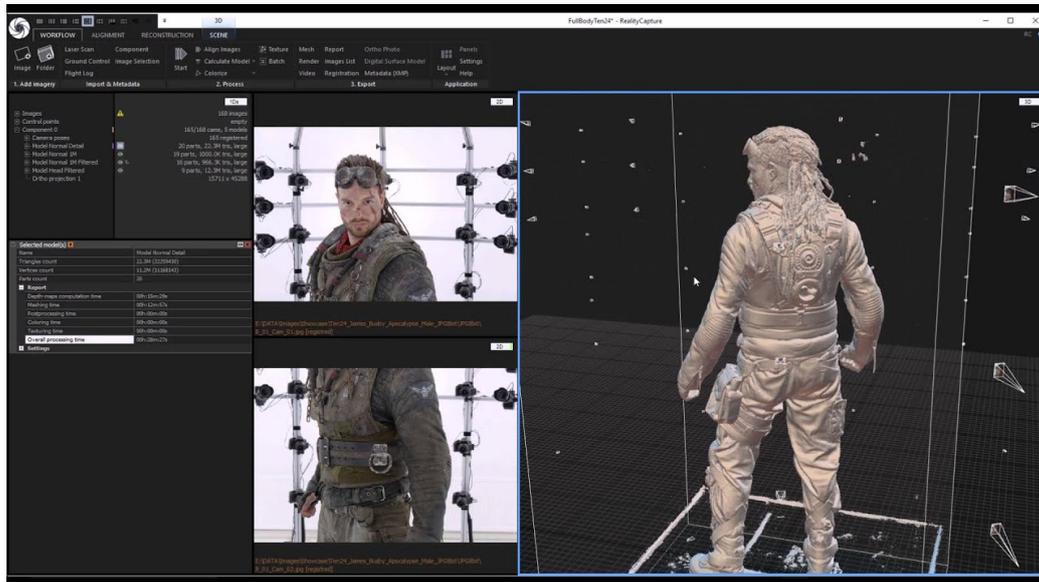


FIG. 5 – *RealityCapture, logiciel de photogrammetrie*

3 Contribution

3.1 le scanner 3D

Le scanner utilisé pour les tests est un scanner à photogrammétrie. Grâce à une centaine d'appareils photos reliés en eux par des Raspberrys et un serveur, ils prennent en photo simultanément le sujet. Ces photos sont ensuite importées dans le logiciel Reality Capture qui construira le mesh du sujet. Cependant nous avons rencontré des problèmes avec quelques appareils photos qui ne photographiaient pas, ce qui génère des trous ou des imperfections dans le mesh reconstruit.

3.2 L'auto-rig

Les recherches sur les méthodes d'auto-rig ont montré que cette technique était réalisable. Dans un premier temps, il convient de déterminer la méthode d'autorité que nous allons implanter dans le projet REVAM eu égard aux contraintes imposées.

Certes, ces techniques ne sont pas aussi précises que si nous utilisions du personnel spécifique pour créer l'avatar dans un logiciel 3D. Mais elles présentent l'avantage de gagner en rapidité..

Parmi mes recherches, trois solutions ont été mises en exergue:

- La solution du site Mixamo,
- Implanter l’algorithme Pinocchio décrit ci-avant
- Le logiciel Smart body qui a déjà l’algorithme Pinocchio intégré, est scriptable et possède un moyen de remodeler le mesh de base grâce à une base de données.

3.2.1 La solution du site Mixamo

Le site mixamo est détenu par Adobe et ne possède pas d’API. Il est donc impossible d’envisager cette option qui nécessite trop de manipulations par l’utilisateur en temps réel .

3.2.2 l’algorithme Pinocchio

L’implémentation de cet algorithme est intéressant, toutefois la solution qui suit permet de réaliser un travail similaire et de meilleure qualité.

3.2.3 Le logiciel Smart body

Après lecture du manuel d’utilisation du logiciel, nous avons choisi d’utiliser smartbody qui a l’avantage d’être scriptable et pourra donc être utilisé facilement à travers l’application unity. Néanmoins ce logiciel nous nous impose la contrainte d’utiliser des fichiers au format .obj. Ce format étant un des standards présente l’avantage d’être open source, il est très souvent disponible lors des rendus sur les scanner 3D, ce qui est le cas du scanner à notre disposition.

L’utilisation de Smartbody est très simple. Il suffit de glisser le fichier .obj dans le logiciel, puis de cliquer sur auto-rig pour que la procédure se lance. Après quelques temps de calcul, le logiciel génère un fichier .dae étant une copie du mesh avec évidemment le squelette approprié.

Cette action devant être réalisée avec le moins de clics possibles pour ne pas perdre l’utilisateur (personnel médical), nous allons devoir la scripter. L’API de l’application stipule pouvoir être scriptable. Cependant la partie qui nous intéresse (l’auto-rig) appartient à un autre module qui n’est pas intégré dans les bibliothèques python de la version de base permettant de scripter. Toutefois, le logiciel Smartbody propose un système d’invite de commandes python intégré qui lui permet d’utiliser l’auto-rig de manière scriptable.



FIG. 6 – *Squelette généré par Smartbody*

D'un point de vue ergonomique écrire dans une console n'est pas intuitif. Nous avons donc trouvé une solution afin de charger un script Python (Annexe) qui sera généré à partir de Unity. L'inconvénient est son chargement manuel dans le logiciel Smart body (voir le schéma des étapes)

L'utilisation de Smartbody est très simple. Il suffit de glisser le fichier .obj dans le logiciel, puis de cliquer sur auto-rig pour que la procédure se lance. Après quelques temps de calcul, le logiciel génère un fichier .dae étant une copie du mesh avec évidemment le squelette approprié.

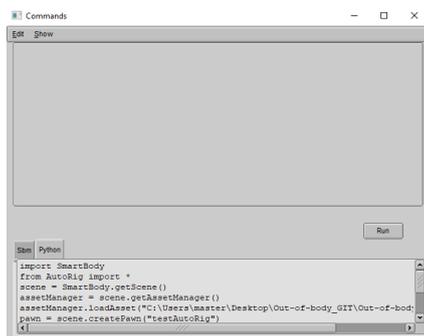


FIG. 7 – *Invite de commande SmartBody*

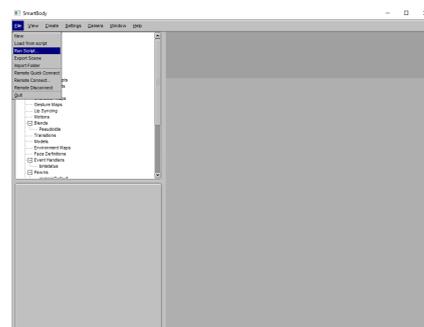


FIG. 8 – *SmartBody*

3.3 L'importateur de fichier .dae

Une des contraintes imposées à ce projet est de pouvoir tout contrôler via une fenêtre de navigation. Un nouvel obstacle se présente : l'importation des futurs scan 3D. En effet, dans Unity il est impossible d'importer des fichiers durant le lancement d'une application. Unity utilise un système d'asset qui oblige de passer par lui afin d'y importer directement les items souhaités dans le projet, puis de le compiler pour en avoir l'application. Ce schéma de fonctionnement nous oblige à avoir tous nos assets préchargés dans le projet. Dans notre cas il n'est pas concevable d'avoir à recopier un projet à chaque nouveau scan de patient.

Nous avons donc dû créer un importateur de fichiers en temps réel. Nous nous sommes inspirés d'un importateur de fichier .OBJ disponible sur le wiki unity (Annexe) pour créer un importateur de fichiers .dae. (Annexe)

En plus de l'importation des différents composants habituels d'un mesh (triangles, vertices, uv), nous avons dû gérer l'importation du squelette qui n'est pas prise en compte par les fichiers .OBJ. Pour la création de cet importateur nous nous sommes basés sur les fichiers générés par l'auto-rig de Smartbody. Cet importateur est donc spécifique à l'écriture des fichiers générés par smartbody et n'est pas universel à tous les fichiers .dae.

En plus de la fonctionnalité de base permettant l'import de n'importe quel scan 3D, nous avons décidé d'implémenter une option permettant simplement d'importer la silhouette du scan.

En validant, cela ouvre le logiciel Smartbody où il suffit de cliquer sur "File,Run Script..." puis de sélectionner le script "AutoRig.py".

L'algorithme de l'auto-rig se lance (ce qui peut prendre de quelques secondes à quelques minutes selon le nombre de triangles du modèle 3D). Une fois l'algorithme terminé, il convient de fermer la console SmartBody pour constater l'import du modèle et de ses textures dans l'application.

Malgré tout, nous avons constaté certaines difficultés dus aux méthodes employées.

3.4 Morphing et utilisation du scanner 3D

L'utilisation d'un scanner 3D vient fait apparaître un nouveau problème et remet en cause le fonctionnement de base de l'application. En effet, comme nous avons pu le voir précédemment sur le déroulement de l'application, le moment du morphing est le coeur de l'application car étroitement lié au phénomène de sortie de corps.

En effet, pour réaliser un morphing, nous avons deux prérequis :

- un modèle de départ
- un modèle d'arrivée

Cependant dans notre cas nous ne possédons qu'un seul modèle 3D: le scan 3D du patient.

3.4.1 Le Morphing par interpolation linéaire des points

Dans l'optique de notre morphing particulier, qui simule un amaigrissement, le même nombre de triangles organisés de la même manière pour chaque modèle est requis afin d'éviter au morphing de ressembler à une transformation. (voir figure) Une même organisation des points et le même nombre de triangles permettent d'appliquer une interpolation linéaire des points [11], l'une des méthodes de morphing les plus utilisées. On calcul le morphing comme ceci:

$$N = \alpha T_i + (1 - \alpha)T_j, 0 < \alpha \leq 1$$

Où N est le nouveau mesh, α le facteur de modification, T_i et T_j respectivement les modèles de début et de fin.

Nous précisons que dans notre cas nous ne possédons qu'un seul modèle qui pourrait être considéré comme le modèle d'arrivée, cette silhouette étant celle de l'utilisateur et que l'objectif souhaité est de faire prendre conscience au patient de sa morphologie.

Il nous manque donc un modèle 3D si nous voulons procéder par un système d'interpolation des points pour le morphing.

3.4.2 Les blendshapes: la techniques des jeux vidéo

Une autre solution pour réaliser un morphing amaigrissant est de se servir d'un système utilisant des blendshapes. Les blendshapes sont souvent utilisés dans les jeux vidéo notamment pour les expressions faciales. Elles permettent

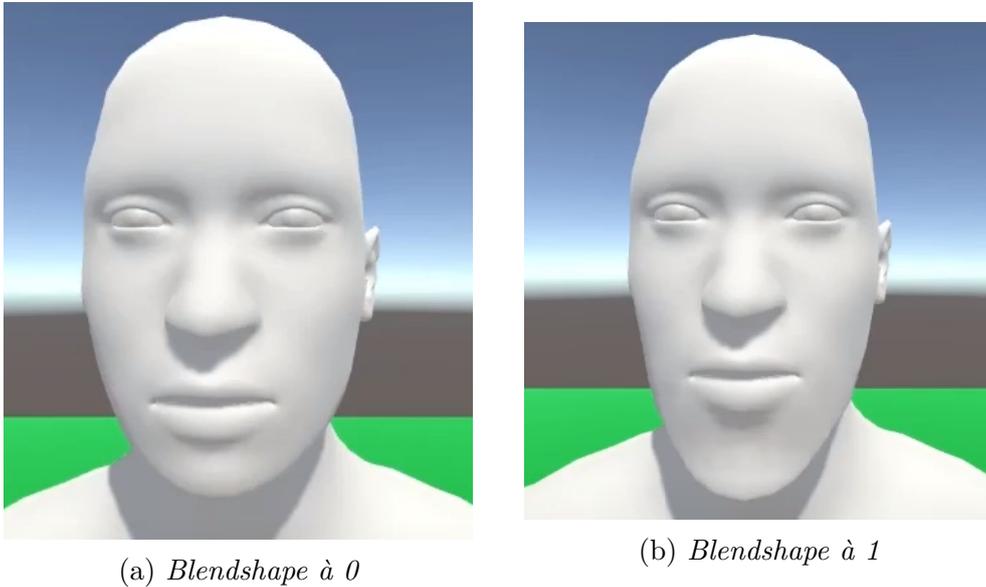


FIG. 9 – *Blendshape au minimum et maximum*

de définir différentes déformations sur un modèle 3D à sa modélisation. Ces déformations peuvent être réglées à différentes intensités entre deux extrêmes. Notons qu'il est nécessaire d'avoir le même nombre de points entre les deux extrêmes

Les blendshapes étant définis sur un seul mesh, il est intéressant de noter que nous n'avons pas besoin de deuxième modèle pour réaliser un morphing. Dans notre cas, les extrêmes étant le modèle maigre et le modèle plus gros, cette méthode serait totalement applicable. Cependant, pour créer des blendshape, il faut pouvoir éditer dans un logiciel de modélisation 3D et définir manuellement les extrêmes en modifiant le mesh. Hélas, l'une des contraintes exposées précédemment est de réduire les interventions humaines longues et fastidieuses afin d'automatiser et accélérer le processus, cette solution n'est donc pas envisageable.

3.4.3 Multirésolution Interpolation

Comme vu précédemment, la technique de l'interpolation linéaire est très efficace pour réaliser un morphing qui reproduit un amaigrissement. Cependant, elle nécessite d'avoir deux modèles de même structure et de même nombre de points. Dans notre cas nous n'en possédons qu'un seul modèle. Pour appliquer cette technique, il est donc été nécessaire de déterminer si un deuxième modèle d'arrivée unique servant à tous les morphing pourrait être envisageable psychologiquement pour les patients.

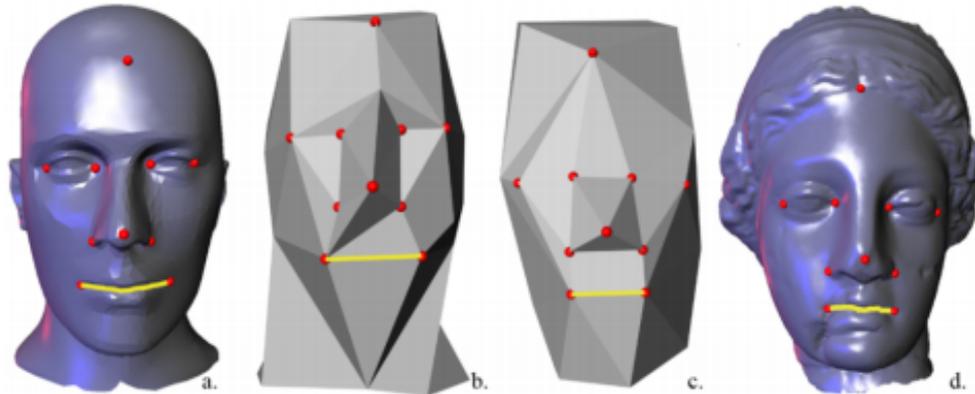


FIG. 10 – (a) *Modèle de départ*, (b) *Modèle de départ simplifié*, (c) *Modèle d'arrivée simplifié*, (d) *Modèle d'arrivée*

Une des principales question lors d'un morphing entre différentes résolutions est la correspondance des points du modèle de départ au modèle d'arrivée et d'identifier les triangles modifiés, supprimés ou ajoutés.

3.4.4 MAPS (Multiresolution Adaptive Parameterization of Surfaces)

Pour répondre à la question Lee et al.[5] ont utilisé des modèles simplifiés au niveau du maillage des deux mesh (Voir figure 10)

L'algorithme MAPS construit un mesh grâce à la suppression successives des vertices puis à une re-triangulation des trous résultants. Pour obtenir le calcul de cette correspondance, des points sont néanmoins à placer manuellement sur le modèle par l'utilisateur. Ces points ne seront pas supprimés par le calcul de simplification ce qui permet au mesh de garder une forme globale. Une fois la correspondance faite, la carte du modèle de départ est projetée sur le modèle d'arrivée. Une détente (une modification de la tailles des triangles) du mesh est ensuite appliquée de manière itérative afin que les points correspondent lors de la transition. Enfin à partir du mesh source un méta-mesh est construit comprenant tous les vertices. Le méta-mesh peut ensuite être utilisé pour le morphing classique, c'est à dire en utilisant la méthode d'interpolation des points.

3.4.5 MiMesh (Multiresolution Interpolation Mesh)

MiMesh est une technique proposée par Michikawa [8]. Elle consiste à créer un mesh M^0 manuellement par l'utilisateur à partir des mesh source et d'arrivée. Le mesh est ensuite subdivisé successivement grâce à un système

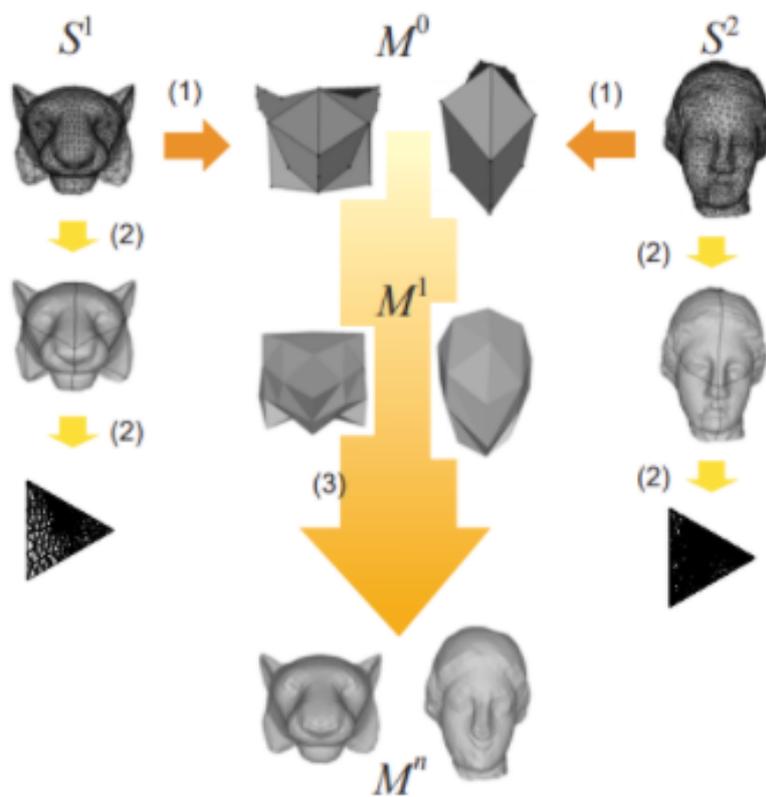


FIG. 11 – (1) Modèle de base simplifié, (2) Paramétrage et interpolation des mesh, (3) Subdivision

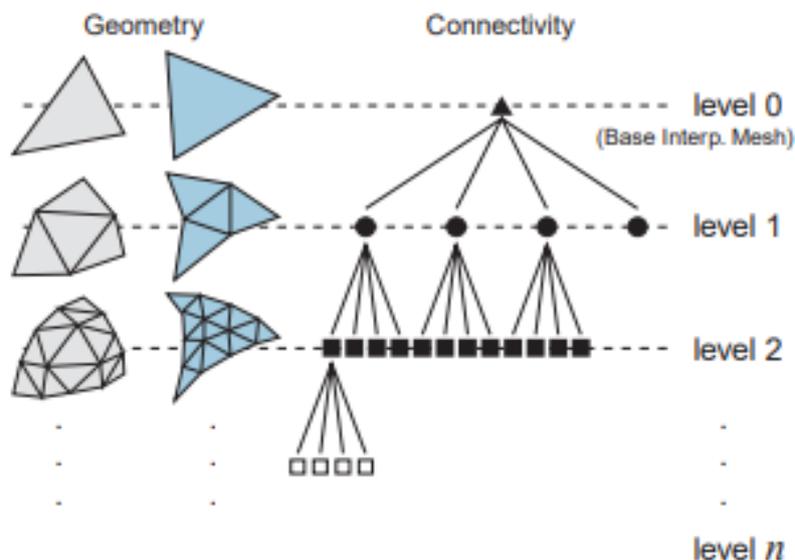


FIG. 12 – *Algorithme 4-to-1 de MIMesh*

de 4-to-1(Figure 12) afin d’approximer le modèle de base.

Les modèles M^0 étant subdivisés n fois chacun, ils possèdent le même nombre de face. A partir des deux modèles M^0 , il est donc possible de faire une interpolation linéaire des points afin de réaliser le morphing(Voir Figure 11).

3.4.6 Bilan

Au vu des contraintes imposées par le projet, tel que l’avatar unique et l’impossibilité de passer par de logiciel tiers pour modifier manuellement les avatars, les techniques de morphing multirésolution sembleraient être un début de solution. Cependant des problèmes subsistent. En effet, ces techniques demandent que l’utilisateur paramétrer le morphing t en plaçant des points ou en créant des modèles 3D manuellement. Placer des points sur chaque modèle est une solution possible. Toutefois de traitement l’application étant utilisable que du côté patient en VR, cette solution en temps réel semble peu envisageable. De plus, créer des modèles 3D pour chaque nouveau scan n’est pas une solution satisfaisante.

De surcroît, sans réorganisation de la structure des mesh, le morphing opéré ne ressemblerait non pas à un amaigrissement mais à une transformation car l’organisation des triangles entre deux modèles est souvent différente,

ce qui cause des difficultés de correspondance entre les points et donc un mesh totalement déconstruit et non un amaigrissement comme voulu. Une réorganisation est de ce fait obligatoire.

Yu et Chuang ont essayé de réduire le nombre d'interventions manuel [1]. Cependant l'utilisateur doit toujours placer des points pour paramétrer le morphing. Si nous regardons plus loin, d'autres problèmes peuvent se profiler comme par exemple le morphing du squelette qui doit lui aussi s'adapter à la nouvelle forme. Ici avoir une structure similaire est importante.

Actuellement il semble difficile de procéder à un morphing entre deux modèles ne correspondant pas au niveau du nombre de points sans un minimum de travail en amont.

3.5 Gestion des textures

Dans une optique de morphing classique (par interpolation linéaire avec deux modèles possédant la même structure et le même nombre de points) il est possible d'appliquer un morphing sur les textures en utilisant une interpolation linéaire 2D et des techniques de triangulation.

Cependant le morphing multirésolution pose des soucis avec la gestion des textures. En effet, nous avons vu que le but de ces techniques était de ramener le mesh de départ et le mesh d'arrivée aux mêmes nombres de points. Toutefois les textures générées par le scanner 3D sont prévues pour des modèles avec une certaine organisation et un certain nombre de points.

Lors de la suppression ou de l'ajout de points, les uv liées à ces points ne correspondent plus, provoquant ainsi des aberrations visuelles au niveau de l'avatar. Pour palier à ce problème nous avons décidé d'appliquer une texture noire unie sur l'avatar. Ainsi il n'y a que la forme du mesh qui doit être modifiée lors du morphing

4 Conclusion

Au cours de ce projet nous avons souhaité impliquer les participants afin que le phénomène de sortie de corps se voit renforcé. Cette illusion sera grandement accentuée lors de l'utilisation de scan 3D. Il y a toutefois certaines difficultés qui empêchent la totale utilisation d'un modèle scanné en 3D.

Nous avons vu qu'il était possible de réaliser un squelette d'un modèle 3D automatiquement de qualité correcte en utilisant le logiciel SmartBody. Ceci était l'un des principaux problèmes qu'imposent le scan 3D et les contraintes du projet. L'import en temps réel du scan 3D était aussi obligatoire pour éviter l'import de nouveau modèle puis de recompiler le projet à chaque patient scanné. Cependant les difficultés apparaissent lors de l'étape du morphing où, dans l'état actuel de la science, aucune solution n'existe sans intervention manuelle au préalable.

Notons que le morphing demandé doit se rapprocher d'un effet d'amai-grissement. Le problème de ce type de morphing vient de la structure du mesh qui se doit d'être la même dans les mesh de départ et d'arrivée en plus d'avoir le même nombre de points. Une piste de recherche sur le sujet se trouve au niveau de la retopologie. Il est possible de faire des retopologies des mesh afin qu'ils aient un nombre de points définis. Si il est possible de projeter la structure d'un mesh sur un autre sans le modifier, alors le morphing sera réalisable. Le logiciel Zbrush permet de faire des retopologies et des projections. Cependant dans ce logiciel les projections se voient modifier les mesh ce qui dans le cas d'un morphing n'a pas d'intérêt.

Si intervention une intervention manuelle est possible, la méthode des blendshapes est à privilégier. Elle est la plus rapide et la plus précise et adapté au moteur Unity. Qui plus ai, elle permet d'utiliser les textures des modèles 3D.

5 Annexe

Exemple de code généré avec Unity pour l'auto-rig

```
import SmartBody
from AutoRig import *
scene = SmartBody.getScene()
assetManager = scene.getAssetManager()
assetManager.loadAsset("C:\Users\PC\Desktop\Out-of-
body_GIT\NoRig.obj")
pawn = scene.createPawn("testAutoRig")
pawn.setStringAttribute("mesh", "NoRig.obj")
autoRigManager = SBAutoRigManager.getAutoRigManager()
autoRigManager.buildAutoRiggingFromPawnMesh('
testAutoRig', 0, 'testAutoRig.sk', 'testAutoRig.dae')
saveDeformableMesh('testAutoRig.dae', 'testAutoRig.sk',
"C:\Users\PC\Desktop\Out-of-body_GIT\SmartBody\
Autorig")
```

Lien pour le code source de DaeImporter.cs

```
git clone https://Braksou@bitbucket.org/Braksou/scipt-
stage.git
```

Importeur temps réel .obj pour Unity

<https://wiki.unity3d.com/index.php/FastObjImporter>

6 Lexique

Mesh: Définit un ensemble de points formant un maillage pour un objet 3D. Un mesh est composé de faces, arêtes(edges) et sommet(vertices)

Uv: Coordonnées sur l'image de textures

Raycast: Un rayon virtuel permettant de mesurer des distances par exemple

Méta-Mesh: Un mesh identique au précédent sans le remplacer

Références

- [1] *Consistent Mesh Parameterizations and Its Application in Mesh Morphing*.
- [2] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3D characters. *ACM Transactions on Graphics*, 26(3):72, July 2007.
- [3] Par Christophe. 4 • Quand l'esprit met le corps à distance. page 4.
- [4] Inserm. <https://www.inserm.fr/information-en-sante/dossiers-information/anorexie-mentale>.
- [5] Aaron W. F. Lee, Wim Sweldens, Peter Schröder, Lawrence Cowsar, and David Dobkin. MAPS: multiresolution adaptive parameterization of surfaces. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*, pages 95–104, Not Known, 1998. ACM Press.
- [6] B. Lenggenhager, T. Tadi, T. Metzinger, and O. Blanke. Video Ergo Sum: Manipulating Bodily Self-Consciousness. *Science*, 317(5841):1096–1099, August 2007.
- [7] Romain Lopez and Christophe Poirel. Raycast based auto-rigging method for humanoid meshes. In *ACM SIGGRAPH 2013 Posters on - SIGGRAPH '13*, page 1, Anaheim, California, 2013. ACM Press.
- [8] T. Michikawa, T. Kanai, M. Fujita, and H. Chiyokura. Multiresolution interpolation meshes. In *Proceedings Ninth Pacific Conference on Computer Graphics and Applications. Pacific Graphics 2001*, pages 60–69, Tokyo, Japan, 2001. IEEE Comput. Soc.
- [9] Mixamo. Mixamo.
- [10] Masahiro Mori, Karl F. MacDorman, and Norri Kageki. The Uncanny Valley [From the Field]. *IEEE Robotics Automation Magazine*, 19(2):98–100, June 2012. Conference Name: IEEE Robotics Automation Magazine.
- [11] Yueqi Zhong, Hongyan Liu, Juanfen Jiang, and Li Liu. 3D Human Body Morphing Based on Shape Interpolation. In *2009 First International Conference on Information Science and Engineering*, pages 1027–1030, December 2009. ISSN: 2160-1291.