Chapter 2

Related Work

The first section of this chapter discusses the classification of different approaches to avatar, and the distinct differences between them. The second section gives an overview of a few applications that make use of avatars. Section 2.3 discusses the avatar generation process from a broad perspective, highlights how different researchers generated the avatars, from creating the avatar to animating the avatars motion, and how do these processes differ in implementation. Section 2.4 discusses how facial expressions are created in avatars and various facial animation approaches over the years.

2.1. Approach to Avatars

The term "Avatar" in Hindu mythology means the descent of god or goddess to earth in bodily form, this is a human body represents god on earth [102].

In Virtual environments, "Avatar" means the user's bodily incarnation in the virtual world. This means an avatar is an entity in a virtual world that represent a human user or a software agent capable of interacting with other entities and the virtual environment [103][104]. When participants enter the virtual environment, they want to interact with the objects and other participants in this virtual world. However, interaction is only possible if the users can view other users or themselves in this virtual world knowing where they are and what they are looking at in the virtual environment. Therefore, avatars represented the users and the virtual characters they interacted within this virtual world. The avatar can be implemented as virtual human (computer generated human

like characters in the virtual world), synthetic avatars (computer generated entity in the virtual world), and video avatars (video generated entity in the virtual world). A Virtual human is a form of synthetic avatar, which has a detailed geometric model that closely resembled the human body.

Avatars can be divided into two distinct types by their implementation and controls, namely video and synthetic avatars. Implementation involved obtaining input data, how the avatar is represented in the application, in terms of modelling, texturing, etc. Controls involved how can the user controlled this avatar in the application.

2.1.1 Video Avatar

As the term "Video Avatar" indicated, the core build up of this avatar is based on video input. They use video capture as the input for building the avatar, therefore the representation of this avatar will be the users themselves and human-like. To acquire the images, the person stands on a turntable and place in front of a blue screen at a distance from the camera. Alternatively, a video camera can be placed in front of the user in the immersive projection display to record the user's image [46].

The image taken by the video camera is captured by the graphics workstation. The user's figure is segmented from the background by comparing the background image without the user to the captured image, or use "Chroma keying" to segment a clear image of the user. "Chroma keying" involved setting up a blue screen behind the user, so that the background of the video images becomes transparent when the chroma key values are set up correctly.

In order to obtain the user's position, electromagnetic sensor is used to track the user's position in the immersive projection display. The user's segmented video image is superimposed onto the three-dimension position in the virtual world. The video image is rendered as video texture in real-time.
The video avatar can be placed at the correct three-dimensional position, however it cannot represented three-dimensional body motion, because it is a two-dimensional

plane image. For example, if the participant points at an object in the virtual world, the accurate information about the index fingertip cannot be transmitted to the other user.

In order to represent the three-dimension body movements, it is necessary to make the video avatar using a three-dimension model. A stereo camera will be required to generate a video avatar that incorporates a three-dimension model. Distance of the user from the camera position can be measured using the triangulation algorithm. So resolution and depth of the image can be calculated based on the distance of the user from the camera.

Video avatars represented the users or participants in the virtual world, allowed the users to communicated and interacted with other users in the virtual environment.
A video avatar usually appears, as a 3D-wax character in the virtual environment, their action and position is dependent to the participant's action and position.
It provides high quality, easily recognisable people for virtual environments.
Because the video images of people are generally static when they are recorded from the camera, they appear rather like statue, but provide significant information about where the other users are located in the virtual environment and where they are looking at.

The control and input methods for different types of video avatars are very similar, but their display/representation differs. Therefore, video avatar can be classified according to their dimension:

- ♦ 2D video avatars
- ♦ 2.5D video avatars
- ♦ 3D video avatars

2.1.1.1. 2D video avatars

They are implemented by capturing a remote virtual reality participant from a single real-time video. Then sent these video images to the system, so that the position and rotation of the user can be obtained by analysing the images and tracking devices [46].

The virtual-reality application used these images as texture, and mapped it onto a polygon, which is placed at the right translation and rotation according to the tracker information of the remote user. The image of the user is separated from the rest of the image by chroma keying.

This kind of video avatar is a texture mapped on a flat polygon; the background-cut-out made the avatar appeared reasonably immersed in the virtual environment. The limitation of this kind of avatar is that when viewing from avatar at angle the avatar appeared to be flat from the view-point of other users.

2.1.1.2. 2.5D video avatars

2.5D video avatars (Figure 2.1.) are not flat like 2D video avatars, they have depth, which is limited, and required a more complex texturing algorithm than 2D avatar.
Unlike 2D video avatars, 2.5D video avatars required one or more cameras to captured the video images of the user, this will depended on the approach taken. The 2.5D video avatars can be implemented in varied methods [46].

The first approach is that the video image is projected onto a static head model using a projective texture matrix. This allows the view from the camera to be mapped back onto the head model from the sensor position and orientation. Using this approach, only the head of the avatar can be displayed onto the screen output.

The second approach is to use two two-dimension images to represent the avatar; but not all the depth cues are supported by this approach. The depth cues supported are convergence, binocular disparity, horizontal motion parallax, and occlusion.

The third approach used the image from the video, and calculated the distance of all the pixels of the captured image to generate the image depth. The three-dimension position of each pixel is calculated in the virtual world co-ordinate system from the depth image. Connecting each pixel with the triangular meshes can generate a surface model. This creates a video avatar in 2.5 dimension that has a surface model for the front side, which

is generated from the texture mapping of the segmented user image onto the surface model [46]. Although the video avatar used a surface model for the front side, there is no shape at its other aspects. The image of the avatar also becomes distorted as the participant moved away from the camera position or is viewed at a different angle. By placing more cameras around the user's area of movement, the camera closest to the user will be used to capture the texture image of the user. This method of camera switching represents a three-dimensional shape according to the other user's viewpoint. This approach was created from a developed method that constructed a three-dimension structure using multiple cameras [47][48].

The limitation of this approach is that it cannot be used in real-time, as large amount of calculations is required for camera switching between several cameras.

Figure 2.1. A 2.5D video avatar interacting with the user in the shared virtual world [46].

2.1.1.3 3D video avatars

This approach was developed by Takaai Akimoto et al. [39], which involved more processing and calculations, than the other types of video avatars mentioned previously. Therefore, only the head of the user can be generated using this approach, but the body of the avatar can be computer generated.

By obtaining the two images of the user's head a front and side view, 3D video avatars are generated. From these two images, the facial features are extracted and calculated according to the facial profile. This determined the position of the facial features (eyes, nose, and mouth) within the head image. Then the values obtained are use to modified the vertices of the generic head mesh and it's facial features to match the head of the

user. The front and side-view images of the user are texture mapped on to the modified head mesh of the user.

The result of this approach is very realistic, but it cannot extract the facial features for some users' head correctly. Processing the extracted data require large amount of time (recognition, transmission and synthesised the facial expression) and the frame-rate is low, which made this approach not feasible for real-time collaborative environment.

2.1.2 Synthetic Avatar

The term "Synthetic Avatar" referred to a computer-generated entity or character in a virtual world. These characters are rendered in real time and performed actions without direct user control. They often interacted with the user who is exploring the Virtual Environment or represented the user's presence in the virtual world.

It is not necessarily that all synthetic avatars are represented in a human form, some avatars appear in computer-generated autonomy, which included animals, insects, robots, and machines, and even characters that can only existed in our creative imagination. They are not necessarily in a virtual human form (human-like) and take independent actions, which include interaction and reaction to the user presence or response. Alternatively, they take dependent actions when representing the user. The user drives the actions and interaction of the synthetic avatars.

Since the form that synthetic avatars may appear varies in different applications, the most complex form of synthetic avatars is virtual human representation. They demanded tracking and other high level mechanisms to animate with maximal facility and minimal input. Therefore, classifications can be made based on their motion control methods and their interactions as proposed by Magnenat-Thalmann and Thalmann [26][4][6].

The participant should be able to animated his/her virtual-avatar representation in real-time, however the avatar control is not straightforward, the complexity of virtual human representation needed a large number of degrees of freedom to be tracked. In addition, interaction with the environment increases this difficulty even more.

Therefore, the human control should use higher level mechanisms for animating the representation with maximal facility and with minimal input. The motion control of avatars increased in complexity, it will be more appropriate to used a classification that base on body motion and facial animation:

- Pure avatars
- Autonomous characters
- Guided avatars
- Interactive-perceptive avatars
- Network real-time synthetic avatars

2.1.2.1 Pure avatars

They are in the form of virtual human, which required a technique call, "real-time rotoscopy" [5]. This consisted of recording input data from a virtual reality (VR) device in real-time and applied these same data to the avatar in the virtual environment concurrently.

The body can be animated using sensors like Polhemus Fastrack or Flock of Birds (sensors attached to the user's arms, legs and body to feed the user's motion as input values to the system). Therefore the body and face of the avatar would look natural and the animation would correlate to the actual body and face.

For example, when the user/animator opens his/her fingers about 2cm, then the avatar should also open its fingers by 2cm. When immersive interaction is required by an application, then the complete body of the avatar should have the same movements as the user's body. This can be achieved by using many sensors to track the every degree of freedom. Molet et al [40] suggested that a minimum of 14 sensors are required to managed a correct and natural-looking posture, but with Semwal et al. [38] 's close-form algorithm only 10 sensors are required to approximated the body posture.

The video sequence of the user's face can be texture mapped onto the avatar's face continuously. This will require the user to be in front of the camera, so that the image of the user from the head to shoulders, or even the whole body can be captured by the camera. The users are allowed to move freely in front of the camera without distorting

or losing the facial images required for texturing the avatars. A simple and fast algorithm is needed to determine the bounding box of the user's face within the image. The algorithm requires a view between the head to shoulders and a static background. Position detection occurs when the algorithm compares each image with the background's original image. Since the background is static, the user's movement causes changes in the image. The position changed in the image can be obtained by calculations from the algorithm. It is possible to analyse the images and extract the changes on the set of facial parameters (Eyes, eye brows and lips positions) that represent a facial expression. The camera in front of the user digitises the video images of the user. Detail measurements of the changes in the facial feature are required for the accurate recognition and analysis process, to determine facial expressions from the video sequences.

2.1.2.2 Autonomous characters

Autonomous characters (agent-based) are synthetic actors that embody the capabilities of the agents [32]. They are driven by agent technology and therefore have their own internal goals and ambitions, and they are able to demonstrated behaviour. An agent is an autonomous software object that is capable of making decisions and act to satisfy it's internal goals, based upon its perceived environment.

Autonomous characters interact with other participants can increase the real-time interaction with the environment and this will increase the sense of presence for the real-people (users).

The users do not guide autonomous characters. Autonomous characters require sufficient behaviours to act autonomously to complete their objective. This requires the appropriate mechanisms for interaction and building behaviours for motion.
Agent-based autonomous characters have the ability to sense their environment, perceive the objects and other avatars in the environment. They can react on that environment based on their perceived information in order to fulfil their objective,

change their own behaviour, and communicate to other avatars in the same virtual environment.

If more complex coding and AI are used in the system, then these virtual agents can be provided with the ability to adapt and evolve in the virtual environment.
Natural environment is modelled in the virtual environment with autonomous characters modelling the entity that researchers want to study. This enables the researchers to simulate the behaviour of people, animals, plants, insects and bacteria, which gives a better understanding of the environment they lived in.

Renault et al. [29], first introduced the concept of virtual vision as main information channel between the avatar and their virtual environment. Avatars perceived their environment through a small window of the rendered environment simulated their point of view. By accessing the pixel depth values and their own position in the virtual environment, the avatar can locate the objects in their virtual world, which are visible to them. Autonomous characters' facial expressions can be animated spontaneously depending on the perception and emotional state of the avatars.

Agent technology is the only method to animate and empower a Synthetic Avatar. Some methods like pre-scripting and programmed behaviours are used, but interactions are predictable and repetitive. Other methods utilise the aspects of agent relating to internal goals and decision-making, but the autonomous character has complete access to virtual environment database (knowledge of the character). Therefore, the agent can made decisions on the aspects of the environment, which it cannot sensed using it's own knowledge. This is commonly used in computer games where the computer opponent can know the position where the player is in the game environment.

The advantages of the agent-based Synthetic avatars is that they are unpredictable and their interactions are rich cause the virtual environment to be more realistic. If the design is adaptive, then the behaviour of the avatar can be changed based on the users' actions or a change in the virtual environment, creating more interesting, unique and

memorable experiences. This increases computations and processing, as decision cycles of the avatar increase when it determines its option based on the sensed information. Better algorithms are currently been developed to overcome these problems and improve the performance of the virtual system.

2.1.2.3 Guided Avatars

These are users driven but guided avatar's motions do not corresponded directly to the users' motions. An example is when an animator is animating an avatar that is not in the form of a virtual human. The animator can use Dataglove to give input values for an alien avatar's jaws movement, so the animator's movement does not correspond to the alien avatar's hand movement. Guided Avatars are very common in Computer Puppetry [66].

Guided avatars are based on the concept of real-time direct metaphor. The participants updated avatars' positions using the input devices. The input device computes the incremental change in the avatars' position. This enables the users to select a set of pre-defined facial expressions or movements from the menu or the graphical interface. In this way, the facial feature changes can be stored as values in the application, which animated a specific facial expression. When simulating a walking avatar, instead of using a real-person with sensors attached to feed input values of the walking motion to the avatar. Sensory information of the walking motion can be obtained using the instantaneous velocity of motion, to computed the joint angles for the whole body, then used gesture with a Dataglove or SpaceBall as the input device.

2.1.2.4 Interactive perceptive avatars

Interactive perceptive avatars are autonomous characters that perceive and communicate with of other avatars and real people interactively. Using postures and other indications of how people feel can do non-verbal communication. Postures provide a mean of communication by defining the positions of the arm and leg, and body angle.

As for communication between avatars, behaviour of the avatars would depend on the emotional state of the avatar, and the facial expression and speech depend on the type of avatars used in the virtual environment. Communication between real people and virtual avatars requires a means for the avatar to sense the presents of real people and their environment. Real people are aware of the presence and actions of the avatar through displays or VR – tools (head-mounted displays) but the difficulty is to allow an avatar to sense or feel the behaviour of real people. This requires some form of gestures and facial expression recognition to enable the interaction between real people and avatars.

Emering et al. [18] produced a combat engagement example between a real person and an autonomous character. This demonstrated gesture recognition by the avatar. The motion of the real person is captured using the flock of birds. The system identified the gestures and transmitted the information to the autonomous avatar, which decided and reacted according to the information of the real person's attitude that was received from the virtual reality sensors.

The same principle can be applied to facial communication between real person and avatar, but facial recognition based upon the video sequence captured by the camera. Mase and Pentland [15] used optical flow and principal direction analysis for reading lips. This model was further refined by Essan et al. [13]. Waters and Terzopoulos [17] animated faces by estimating the muscle contraction parameters from video sequence using "snakes"[5]. This is first developed by Kass et al. [68], active contour method that fitted a contour on a given image. This allowed the fitting of the boundary points of maximum contrast close to the pre-defined rough contour. To get correspondence between points from pictures and points on a generic model, this has a defined number. Some people have used external markers and lipstick on the real face to obtained a more robust recognition of facial movements and expressions [8][9]. Li et al.[11] used a Candid model for 3D-motion estimation for model-based image coding. Azarbayejani et al.[1] used a Kalman filter to retrieved motion parameters restricted to head-motion and orientation. Real-time performances are not featured in many of these methods, but Pandzic et al.[12] developed a fast method that used on a "soft mask" (a set of points on

the facial image that the user adjusted interactively). The method allowed the avatar to imitate a real person's expressions through recognition with real-time performance. Alternatively, this method can be further implemented to enable the avatar perceived the real person's facial expressions.

2.1.2.5 Networked real-time synthetic avatars

Avatars are important in a networked virtual environment, because they represented the user in the virtual environment, enabled the user to interact with other users, avatars and the surrounding environment in this networked virtual environment. For example, the VLNet (Virtual Life Network) system [42] supports a networked-shared VE (Virtual Environment) that allows multiple users to interact with each other and their surroundings in real-time. In this network, it can also include other types of avatars that have been mentioned earlier, using autonomous characters to represent users that are not connected to the network or to represent a background character. This allows asynchronous co-operation between distant partners.

2.2. Applications of Avatars

There are various applications for different types of avatars that we might not even noticed in our daily life. More recently, large number of TV advertising companies use 3D avatars to advertised a product. We can see these kinds of avatars every day, and this is one of the applications where avatars are implemented for marketing products.
The applications for avatars can be divided into applications that implement video avatars and applications that implemented synthetic avatars.

2.2.1 Applications of Video avatars

Video avatars allow distant remote users to communicate and perform non-sophisticate interactions to each other, by representing the users in the same virtual world.

Apart from presenting a participant in networked-virtual environments, video avatar can also be used to improvise acting. This was demonstrated with the experiments done by M. Hirose et al. [46], in which video avatar was used to create an improvised acting

scenario between remote users. This experiment was performed at VR Culture Forum held in Yakushima; the conference hall in Yakushima and the immersive projection display CABIN [78] at the University of Tokyo were connected via two 128 kbps ISDN lines. One line transmitted the video image, and the other line was used for telephone voice and the computer data transmission.

In the conference hall, the dancer played out her part on the stage without the stage setting. Her performance was filmed by the video camera and the image was transmitted to the CABIN at the University of Tokyo. The video avatar was generated and superimposed on the virtual world displayed in the CABIN. The user in the CABIN communicated with the video avatar of the dancer in the three-dimensional virtual world, and created improvised acting created by remote users.

Video avatar's faces are also used in video conferencing, where distant users communicated to each other [94].

2.2.2 Application of Synthetic avatars

The roles of Synthetic actors in virtual environment may include teacher in an educational virtual environment [64], tour guide in the virtual world [23], companion, assistant, entertainer, opponent in computer games [90], presenter [67][33], and various other roles.

Beside the above mentioned roles for synthetic avatars, they are also found in TV advertising, films [34][35][36][69] or films as they are merged into captured video of the real world [25], and even helped to stimulated many other types of interactive applications [65].

The Virtual Shopping Mall example (business application):

For an application described an interactive shopping experience enhancement in dynamic interactive virtual mall environments.

In traditional retail outlets, the goal is to present to the visitor an aesthetic environment that displayed the product in an inviting, stimulating and cost-effective manner.

In shopping online via web page, there is a lack of interaction, the user only sees the image of products and the given information. This form of shopping is cost-effective for the company, but it makes shopping experiences dull and boring.
In a physical store, the environment is static; as such, broad common denominators of decor and theme needed to be factored into the floor presentation of merchandise.
By contrast, in a dynamic virtual mall synthetic environment, the aesthetics of presentation and the attributes of the interactive experience are all dynamic variables that can be directly driven by the personality attributes of a shopper's intelligent avatar. As a representation of the shopper's personality traits, demographic features, and aesthetic and emotional preferences, the avatar's personality matrix dynamically shaped the appearance and affordances of the synthetic-shopping environment. The user is represented by the synthetic avatar, while the shop assistant can represented a participant working for the company selling the products, or using an autonomous character as the virtual shop assistant. Therefore, users can shopped interactively online to various shops in the mall.

With the improvement of computer hardware and networks, and approaches that strive to achieve realism in avatars, this would expand the number of possible applications for avatars in future.

2.3. Creation of Avatars

Over the years there are various attempts and approaches developed to generate highly realistic avatars, some approaches are complex in implementation and require specialised hardware, but can achieve highly realistic results, while other approaches generate fairly realistic avatar with simple implementation and standard hardware. This
Section discusses the avatar generation process in general and some approaches implemented by other researchers in more detail.

When modelling/building the avatar, it is better to separate the creation process to manageable parts:

- Constructing the mesh of the avatar.
- Texture mapping of the avatar

- The body movement and kinematics of the avatar.

2.3.1. Constructing the body and head mesh of the avatar

The body mesh of the avatar can be constructed differently according to the form of the avatar. For example, the body mesh of a comical/non-humanoid avatar [64][65] is simpler than that of a virtual human [26][27].
Firstly, this is because the comical/non-humanoid avatar has a simplified body and limbs, which can be constructed by deforming basic geometric shapes and group them together. Secondly, the comical avatar would more likely to have less joints than a virtual human. Therefore, when using hierarchical modelling the avatar would have less leaf nodes, the complexity of each joint and the movement of the body are lower than the virtual human model. Thirdly, these kinds of avatars mostly have simple facial features created on the head mesh, which allowed facial animation to be fast and simple.

The body structure of the avatar must be modelled using the hierarchical approach. This means that the avatar's fingers are modelled first then the rest of the hand and so on.
Then linked the fingers to the hand, to the arm, to the body. The same applies to the other limbs. The reason for linking the body parts in this manner is, because when the arm is rotated the hand should be rotated with it. Therefore a hierarchical movement is established, which the base object (e.g. hand) inherits the movement and rotation from the upper object (e.g. arm). The other advantages of modelling the body using hierarchical approach is that restricted movement (angles between joints) applied to the upper object will automatically be applied to the inherited object. Therefore, animating the avatar's body motion would be less problematic as the body parts will remain in the correct position and not moved out of place during the animation of the avatar.

When constructing a virtual human and if realism is not the main issue or speed of the application is important, then one can created the body of the avatar using 3D geometric shapes to represented the limbs and body [65]. Triangle meshes can be used to model the head of the avatar. In this way, the avatar had a detail head with a simplified body, which required less system resources.

In constructing realistic avatars, T. K. Capin et al [41], defined an articulated structure that simulated the human skeleton. A 3D articulated hierarchy of joints represented the skeleton, each with realistic maximum and minimum limits. The skeleton is encapsulated with geometrical, topological, and inertial characteristics of different body limbs.
The body structure had a fixed topology template of joints and different body instances. These are created by scaling body limbs globally, as well as applying frontal, high and low lateral scaling, or specifying spine origin ratio between lower and upper body parts [61].

Magnenat-Thalmann et al. [27] attached this skeleton, with a second layer that consisted of blobs called "metaballs" to represented muscle and skin. This method's main advantage is that it permitted the entire human body to be covered with only a small number of blobs. Then the body is divided into 17 parts: head, neck, upper torso, lower torso, hip, left and right upper arm, lower arm, hand, upper leg, lower leg, and foot. Because of their complexity, head, hands and feet are not represented with blobs, but with triangle meshes instead [49].

For the other parts, a cross-sectional table is used for deformation. This cross-sectional table is created only once for each body by dividing each body part into a number of cross-sections and computing the outermost intersection points with the blobs.
These points represented the skin contour and are stored in the body description file. During runtime the skin contour is attached to the skeleton, and at each step is interpolated around the link depending on the joint angles. From this interpolated, skin contour the deformation component created the new body triangle mesh.

There are different parameter sets for defining virtual human postures and faces:

Global Positioning Domain Parameters:

These are the global position and orientation values of particular observable points on the body, in the body co-ordinate system. Possible choices are: top of head, back of neck, mid-clavicle, shoulders, elbow, wrist, hip, knee, ankle, bottom of mid-toe.

Joint Angle Domain Parameters:
These parameters comprise the joint angles defined above, connecting different body parts.

Hand and Finger Parameters:
The hand is capable of performing complicated motions and there are at least fifteen joints in the hand, not counting the carpal part [44]. Using hand joints almost doubles the total number of degrees of freedom and therefore separated the hand parameters from those of other body parts.

Face Parameters:
The face is generally represented differently than the other parts of the body. It is a polygon mesh model with defined regions and Free Form Deformations modelling the muscle action [60]. It can be controlled on several levels. On the lowest level, an extensive set of Minimal Perceptible Actions (MPAs), closely related to muscle actions and similar to FACS Action Units, can be directly controlled. There are 65 MPAs, and they described the facial expression completely. On a higher level, phonemes and/or facial expressions can be controlled spatially and temporally. On the highest level, complete animation scripts can be the input defining speech and emotion over time. Algorithms existed to mapped texture on such facial model.

Depending on the type of virtual reality application and the types of avatars used, there are various approaches to generate the avatars' head to display expressions.

In an immersive environment (e.g. CAVE), stereo cameras (Figure 2.2.) are placed around the participant to captured the images of the participant. These images are mapped to a flat polygon or projected to a generic mesh, but these images are static.

This indicated that the face of the avatar is simple and expressions are created by capturing the emotions of the participant and display them onto the avatar [46].

Figure 2.2. An example of a Stereo camera that is used in immersive environment (Triclops stereo camera, developed by Point Grey Research Inc.) [46].

This approach required some tracking devices or video analysis algorithm to tracked the orientation of the participant's face and it continuously texture mapped the video sequence of the user's face onto the face mesh of the avatar [2]. The user must be positioned in front of the camera, so that the camera can captured the image from his/her head to shoulders and possibly the entire body of the user. A fast, simple image-analysis algorithm is used to find the bounding box of the user's face within the image. The algorithm performed image analysis separated the head of the user from the background and projected the user's face to a simple face mesh. Only frontal projection is possible if one camera is used, as the front image of the user is available.

Deformation can be noticed when the facial images are projected onto a generic face-mesh [42]. For example, if the user's face is elongated and the image captured is mapped to a flat face mesh, then the user's face image will be stretched during the mapping process to cover contours along the lower jaw. To prevent less or no deformation on the images, the alternative approach is to create a generic face mesh at run time to fit the captured facial image [39][43]. This involved more processing, because the shape of the face and facial features are extracted from the image first, the shape is reconstructed and then the generic mesh is modified to match the information extracted from the image.

There are various methods for shape reconstruction to get a detail shape, but this is time-consuming, required a sophisticated equipment or complex algorithm. It can be divided into *Shape reconstruction* and *Structured Shape Reconstruction*.

There are a few methods in *Shape reconstruction*, which can get a detailed range data for a face.

Stripe Generator, is a form of structured light-camera range digitizer. A light striper with a camera and stripe pattern generator can be used for face reconstruction. The advantage of this compared to laser scanners is that it is cheaper. Stripe pattern is taken from the camera and projected on the three-dimensional object's surface. Three-dimensional shape can be calculated with information of the camera and projector positions and stripe pattern. Proesmans et al. [70] displayed a good dynamic 3D shape using a slide projector by a frame-by-frame reconstruction of the video.

Plaster Model, Magnenat-Thalmann et al. [69] used plaster models in the real world and selected vertices and facets that are marked on the models, which are digitised by taking photographs from various angles. From this method, high resolution can be obtained from any regions of the model, but the reconstruction process required a mesh drawn on the face, which made it time consuming.

Laser Scanning, In range image vision system, scanners produced range images. The range to the visible surface of the object in the scene is known for each pixel in the image. The spatial location is determined for a large number of points on this surface. Lee et al. [45] digitised facial geometry, using scanning range sensors, but this method based on 3D digitising required a powerful workstation and specialised, expensive hardware. In addition to geometric 3D information, the textural information can also be obtained to created a realistic model of the view face [85][86]. It is important to remove any markers on the actor's face during scanner, as the result of the face mesh would have an irregular surface due to the markers. Although the models are very realistic, these models cannot be animated as they are, because the vertices of their mesh of

There are various methods for shape reconstruction to get a detail shape, but this is time-consuming, required a sophisticated equipment or complex algorithm. It can be divided into *Shape reconstruction* and *Structured Shape Reconstruction*.

There are a few methods in *Shape reconstruction*, which can get a detailed range data for a face.

Stripe Generator, is a form of structured light-camera range digitizer. A light striper with a camera and stripe pattern generator can be used for face reconstruction. The advantage of this compared to laser scanners is that it is cheaper. Stripe pattern is taken from the camera and projected on the three-dimensional object's surface. Three-dimensional shape can be calculated with information of the camera and projector positions and stripe pattern. Proesmans et al. [70] displayed a good dynamic 3D shape using a slide projector by a frame-by-frame reconstruction of the video.

Plaster Model, Magnenat-Thalmann et al. [69] used plaster models in the real world and selected vertices and facets that are marked on the models, which are digitised by taking photographs from various angles. From this method, high resolution can be obtained from any regions of the model, but the reconstruction process required a mesh drawn on the face, which made it time consuming.

Laser Scanning, In range image vision system, scanners produced range images. The range to the visible surface of the object in the scene is known for each pixel in the image. The spatial location is determined for a large number of points on this surface. Lee et al. [45] digitised facial geometry, using scanning range sensors, but this method based on 3D digitising required a powerful workstation and specialised, expensive hardware. In addition to geometric 3D information, the textural information can also be obtained to created a realistic model of the view face [85][86]. It is important to remove any markers on the actor's face during scanner, as the result of the face mesh would have an irregular surface due to the markers. Although the models are very realistic, these models cannot be animated as they are, because the vertices of their mesh of

triangles do not corresponded to the physical ones of the face. In fact, the distribution of the triangles depended on the 3D data acquisition/estimation process.

One possible solution to this problem could be the adjustment of a sub-set of the vertices, chosen as the closest ones to be used as the key-points of the face. Once all such points are optimally relocated, their animation rules, and the consequent motion of all the other vertices, can be defined based on the displacements of the essential features. This will required the definition of different animation rules for each model, as the mesh geometry may vary from model to model. An example of commercial 3D digitizer based on laser-light scanning, is Cyberware Colour Digitizer.

Lighting Switch Photometry, computed the normal vectors for extracting shapes of static objects [71] or a human face in motion [72], by using three or more light sources. This method assumed that the reflectance map is lambertian. The normal vector can be computed at the points where three incident light sources illuminate using Lighting Switch Photometry. The limitation for this method is that computing accurately the normal vector is difficult, particularly at the point where the intensity of the radiance is small. One example is shadowed regions.

In *Stereoscopy, the* correspondence at certain characteristic points can be established by a distance measurement method. This method used the geometric relation over stereo images to recover the surface depth, which resulted in sparse spatial data. Fua and Leclerc [73] used it in texture areas by weighting the stereo component most strongly for textured image areas and the shading component most strongly for texture-less areas.

Most of the methods in Shape Reconstruction are focused on recovering a good shape, the limitation is that structured information is missing, and only shape can be obtained from these methods. Structured Shape Reconstruction involved getting a structured shape for animation. The common approach is modified the generic mesh with structured information such as the eyes, mouth, eyebrows, nose, etc.

These can be classified into methods using range data and without using range data.

With Range Data

In these methods, to made the model suitable for animation, structural information must be added to a set of three-dimensional points.

Using photogrammetric techniques, precise geometry of the head mesh can be created by the image [7]. Grids are drawn on the actor's face to marked positions on the face for modelling and animation. However these images used to construct the head/face mesh can no longer be used as a valid texture map for the subject. To overcome the grid problems, several methods have been proposed for modelling the face photogrammetrically without the used of grid [74][57]. These approaches used a small predetermined set of features to deformed the generic face mesh to the particular face being modelled, and offered no mechanism to further improved the fit. The result from this approach performed poorly on faces with unusual features or other significant deviations from the normal face.

Warping Kernels, is method by Williams [55]. This method used a Cyberware digitizer to reconstruct the head and applied warpware to animated the model. A set of warping kernels is distributed around the face, each of which is a Hanning (cosine) window. This is scaled to 1.0 in the centre, and diminishing smoothly to 0.0 at the edge.

In Mesh Adaptation, Lee et al. [45] started with a structured facial mesh and developed algorithms that reconstructed automatically the functional models of the heads of people from laser-scanned range and reflection data. After the large arrays of data acquired by the scanner are obtained, these data is reduced into a parsimonious geometric model of the face that can be animated efficiently. The generic face is adapted according to the data. When the feature based matching technique completed the mesh fitting process, the algorithm samples the range image at the location of the nodes of the face mesh to

captured the facial geometry. The node positions also provided the texture-mapping co-ordinates that allowed the full resolution colour image to be mapped onto the triangles.

Without Range Data

Methods that are based on the three-dimension digitisation to obtain a range data often required a specialised high-cost hardware. The better and low cost approach is to reconstruct the two-dimension information to generate a three-dimensional object. Two commonly used methods are reconstruction method with feature points which modified a generic mesh model after feature detection method, and interactive deformation method which modified or generated a surface employing deformation.

The performances of methods that reconstructed a face shape from few pictures of a face [39][57][74] are faster in general. In this method, a generic model in 3D is provided in advance, and a limited number of feature points are detected automatically or interactively on two or more orthogonal pictures. The other points on the generic model are modified by a special function. Then 3D points are calculated by just combining several 2D co-ordinates.

An interactive method obtained a few points and Delaunay triangulation for the conformation of the face and texture mapping, in Kurihara and Arai 's approach [57]. The result is good, but the trade-off is that only few points are available, to modified the generic mesh model. If the shape of the generic mesh model is very different from the person's head, the results from few modification points would not looked similar to the person's head, which made texture mapping deformed or stretched the image slightly. To increase accuracy, more input points for modifying the generic mesh model must be available.

Ip and Yin [74] developed a similar approach as Akimoto et al. [39]. Both of these approaches detected the feature points automatically using dynamic template matching or LMCT (Local Maximum-Curvature Tracking). This checked for concave and convex points on the side profile of the face, and a simple filtering method obtained the interior

points. The method is automatic, but not very robust. For some people who have Mongoloid face it works well, but not for others.

In interactive deformation, by using an interactive tool, Magnenat-Thalmann et al. [27] generated a polygon mesh surface for creating figures. The operations performed included creation of primitives, selection, local deformations and global deformations. This method was time-consuming, but it is the only possible way to digitise a historical personage, whose pictorial or other source is not available and is useful when creating new characters.

Once the face mesh generation is completed, the image with the user's expression is mapped to the face mesh [39]. Instead of transmitting the series of image frames containing the facial expression, the later approach is to developed the head mesh with facial features (eyes, mouth, lips, teeth, tongue, ear) created in the head. An analysis from the video input determined the change in facial features, and extracted the set of parameters that described the facial expression. The facial features in the head mesh are modified according to this change. For example, if the eye in the video input changed its orientation (instead of looking straight, the eye looks up), the eye in the head mesh rotated upwards to match the orientation.
T. K. Capin et al. [42] described a "soft mask"- a set of points on the image of the face adjusted interactively by the user for the recognition process. Detailed measurement is required for accurate recognition and analysis of facial expressions from the video sequences, but it is computation expensive to perform these measurements. Therefore, decreasing the number of facial features to be extracted from the video reduced the computations and recognition of the facial features that relied mainly on colour-sample identification and edge detection.

The set of extracted parameters includes:

- Horizontal head rotation
- Vertical head rotation
- Head inclination

- Aperture of the eyes
- Horizontal position of the iris
- Eyebrow elevation
- Distance between the eyebrows
- Jaw rotation
- Mouth aperture
- Mouth stretch/squeeze

The extracted parameters are translated into minimal perceptible actions, which is sent to a facial animation engine that performed the facial animation. Many techniques allowed the parameterisation of expressions and expression feature components (eyes, lips, eyebrows, etc), deformation of facial models. The parameters are set in a way that each feature-component-animation is constraint by these parameters.
For example, the eyebrows can slide up, down, arch at the end of the eyebrow, etc. Parameters are set, so that if the eyebrows slide up the value is '+1', at neutral position is zero and slide down the value is '–1'.
Early work on this field was done by Ekman [101] who developed the Facial Action Coding System, FACS. Magnenat-Thalmann et al [56] have developed the Abstract Muscle Action System, AMA. Recently, an important development is ISO backed protocol: the MPEG-4 Synthetic/Natural Hybrid Coding (SNHC) scheme. This protocol defined the parameters for facial definition (FDP) and animation (FAP).

In the previous approaches, a generic face mesh is adapted to an individual's face from the images. Then constructing facial animation is simple as they are built in directly in the generic head model by defining the head mesh deformations. These approaches have trade-off between real-time rendition capabilities and realism. The face model may end up being an oversimplified, unrealistic head model for the avatar.

Instead of making the generic mesh specified to a given person, Valente and Dugelay [37] begin from person-dependent data (range and texture image) that corresponded to a neutral facial expression. These data is processed to made them suitable for a general

analysis framework. The head mesh has no separated primitives for the eyeballs, the image used for texturing the face is static initially and the face model is a plane surface. Although the head mesh did not have facial features pre-created in the model, it is still possible to animate the facial expressions. Facial expressions can be achieved from different levels of implementation (vertices, texture co-ordinates, and texture image) and simple deformations on the wireframe vertices and other image manipulation techniques. This method will be further discusses in more details under the section "Facial expressions".

The most difficult aspect of facial animation is constructing a believable 3D facial model that is realistic and flexible. Physically based muscle models, developed by Lee [45], Terzopolulos [50][51], and Waters [16], are fairly realistic, but the trade-off is they required large amount of processing for generating the face mesh and they are computations intensive. Deformations of geometry and texturing [52] are commonly applied when the avatar is used in a virtual environment over the network, because their speed performances are better.

Many systems tried to construct a facial model that resembled the user. Guenter et al [52] used Cyberware scans with complex texture mapping, while Escher and Magnenat-Thalmann [21] fitted a generic mesh model to a specific face via control points obtained from the two camera views.

The limitations of scanning the actor's face to generated a face mesh are:

- If the markers on the actors face is not removed during the scan, dumps will be created on the face mesh.
- The mesh did not have an opening for the mouth.
- The face mesh resulted from a scan has too many polygons, which made animation more difficult as the number of control points is large.

The approaches discussed above are a few methods for creating an avatar's head mesh.

In simplified synthetic avatars, the clothes of the avatar are modelled as part of the body mesh and texture with different images of clothing materials.

In realistic synthetic avatars, the clothes are modelled separately [62]. P. Volino et al. [63] had developed a system for creating clothes for synthetic avatars. The creation process for clothes of avatar is similar to a real tailor designing clothes in real-life. The clothing is mostly created using the B-splines, because folds can be ceased on clothing and simulating clothing can be done easily. The cloth panels are design in two-dimension initially, and the texture is defined for each one. Then the seams are defined. The clothes are placed around the avatar's body in third dimension and the seaming lines are closed, wrapping around the avatar's body. Seaming is performed on the clothing model and the panels are merged together to form the garment. Collision detection algorithms and law of physics are also applied to simulated clothes as in real-life (clothes flowing due to wind and tear due to stretching).

The avatar generation process is complex and the cost of realism is processing speed, therefore many researches are done to improve this process to be more efficient and realistic.

2.3.2. Texture mapping of the avatar

Images captured by the video input are used as textures for the avatars. The texture can be an image of the user's face, or from the head to shoulders of the user, or the whole body. This depends on the type of avatar been implemented in the application. Apart from using video images as texture for the avatar, textures can be created by an artist and texture mapped onto the avatar.

2.3.2.1 Texture extraction

Before these images can be mapped onto the avatar, they are separated from the background, so that the image contained only the user. This can be achieved by two approaches, the image taken by the video camera is captured by the system, and the user's image is segmented from the background by comparing a background image without the user. This requires more analysis and processing, and a background image

without the user given to the system, but no additional set-up (putting up the background screen) is required for the system. Therefore, the images can be captured using the standard cameras.

The other approach is implemented using a blue screen and "chroma-keying" to segment a clear image of the user. This approach is fast and does not require image analysis and comparison computations, but needs to pre-set-up at the capturing area. Therefore, this approach is more common in immersive environments, where the whole body of the user can be captured easily by the system.

In the immersive environment, the extracted image of the user is texture mapped onto a flat polygon plane or projected onto a 3D object that represented the user's head or body.

2.3.2.2 Real-time texture-fitting

G. Sannier and N. Magnenat-Thalmann [10][59] proposed an approach for a real-time texture-fitting interface that fitted a texture interactively to the 3D object. The texture mapping co-ordinates of the features on the mesh are marked on the texture, and they corresponded to the vertices on the mesh, allowing it to fall in the right position. Therefore, the user can see the effects of his/her texture manipulation directly on the mesh. The nearest 3D vertices to the marked-point are found automatically as the user added or removed a marker on the 3D model. This made texture mapping for the avatars faster and simpler, which can be implemented on any complex mesh surface (The face and body of synthetic avatars).

2.3.3. The body movement of the avatar

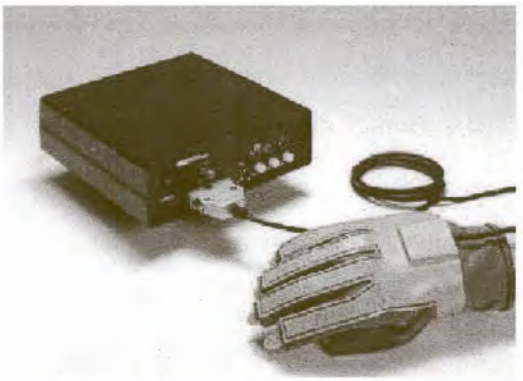
Body movement of the avatar involves animating the avatar's motion. When it interacts with other objects or avatars and when the avatar performs actions in the virtual world. The emergence of techniques like artificial intelligence and object-oriented programming and the increase in computer speed and new virtual reality interacting

devices, makes it possible to classify the movements of the avatar into: avatars with environment, avatars with other avatars and avatars with animator interactions.

2.3.3.1. Motion sensory devices

Body movement data of the avatar can be obtained by a method called, "Motion Capture". Motion capture is a process of capturing the motion of a (human) actor, which used physical devices to control animation of a virtual character [40].

Physical devices may differ in resolution/range of motion, cost, calibration, accuracy and data capturing speed. A few examples that are commonly used are the datagloves (Figure 2.3.), exoskeletons (Figure 2.4.), flock of birds, and other Virtual Reality (VR) tracking sensors. Optical sensors e.g. Ortho Trak developed by Motion Analysis Corporation (Figure 2.5), are used for full body motion capture, it is lighter and less rigid than the exoskeleton, which can captured various body motions. It is widely used in capturing motion for the avatars in computer games and films. The limitation is that it do not supported motion capture in real time, and it required a pre-set-up area to captured the sensor input via a digital camera (Figure 2.5.). Electromagnetic sensors captured body motions in real-time and it do not required a pre-set-up area for motion capture e.g. Polhemus Fastrack, the new types of electromagnetic sensors are wireless e.g. Star Trak developed by Polhemus (Figure 2.6.). These devices' sensors are placed near the joints of the body, which sensed the joints at the body limbs under motion.



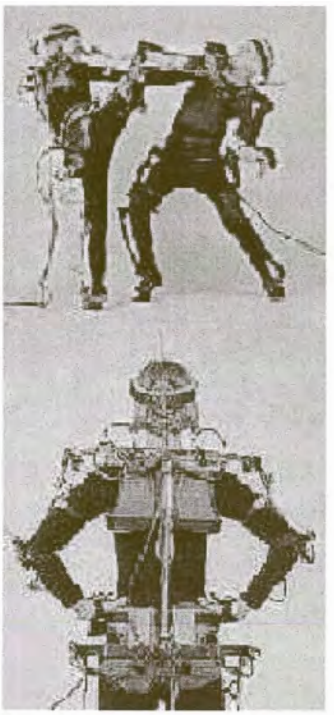


Figure 2.3. (Top) The datagloves are used in capturing motion of the actor's hands [95].

Figure 2.4. (Right) The exoskeleton is used to capture full body motion, but it is rigid and has limited body movement for actors. Data is captured in real time [96].

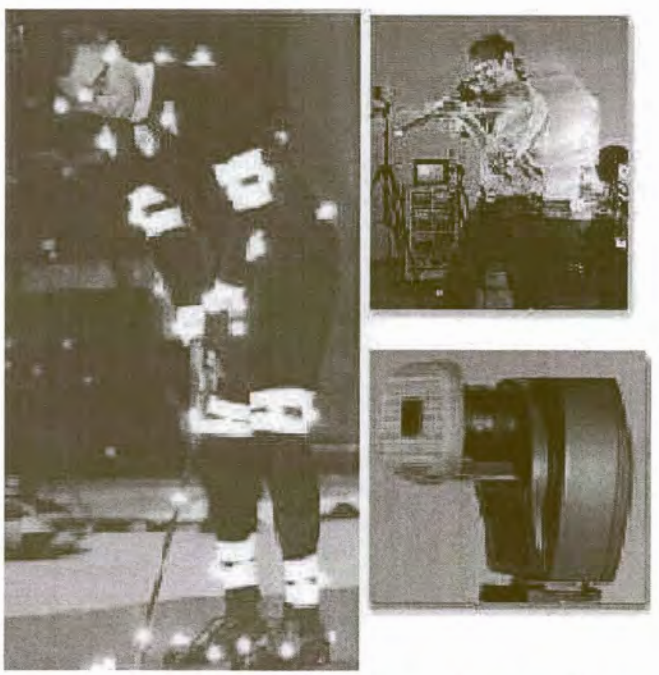


Figure 2.5. The left image showed optical sensors on Mr. Tiger Woods' body, tracking his golfing body motion for a computer game.

The right top image shows the pre-set-up area for capturing body motion [97].

The right bottom image is an example of a digital camera, Eagle, developed by

Motion Analysis Corporation [98].

Figure 2.6. Star Trak, wireless electromagnetic sensors [99].

Apart from body motion, facial movement data can also be obtained using "motion capture". The "motion capture" data played an important role in realistic avatar development. The data obtained by the motion capture process can be used off-line or on-line. During off-line, these data served as filtering for the avatar motion animation, or as Inverse Kinematics (IK). Inverse Kinematics modelled the forces that caused the motion to occur, and this is a technique for positioning end-effectors of the avatar in individual frames of the animation.

While in on-line, these data drove the avatar's motion directly based on the motion of the actor in real-time.

The body motion of the user can be captured by the video and detected via electromagnetic sensors, or virtual reality devices used by the user. Depending on the

type of input devices selected for the application and the type of VR application, the motion of the avatar can be animated differently in the system. For example, in an immersive collaborative environment, the input/output devices used are stereo cameras and projection displays. The video avatars used in the application and their motion will be based on the movements of the user in front of the camera [46]. Therefore, the kinematics of the video avatar is simply the direct video capture of the user.

When the VR application involved using synthetic avatars, the input devices used for the system are the virtual reality tracking sensor (optic or electromagnetic) [66]. Then the avatar must be constructed such that simulating the motion of the user can be done directly from the captured motion data or animated the motion directed by the animator. This required motion detection, which sensed the user's movement and obtained enough data of the user's motion to animate the avatar.

The input values for the motion of the avatar can be received from the VR devices and sensors. If the user moved his/her arm, the sensors sent the input values to the system and computed the change in body motions and then applied these same values to animated the avatar's arm.

2.3.3.2. Motion Control

Magnenat-Thalmann and Thalmann [49][19] classified the computer animation scenes involving synthetic avatars according to the types of avatar interaction and method of controlling motion. Motion control methods specified how an avatar is animated in the application. Motion control methods can be classified according to the types of information belonging to the synthetic avatar that has been animated in the system, into forward dynamics and key-frame system. In a forward-dynamics-based system, when a force acted on the parent object, any child objects are affected with the parent. E.g. If the body is pushed moving backwards, the arms will also move backwards. The information stored in the system is a set of forces and torques. In a key-frame system for an articulated body, the body is positioned at the critical frames with the current or target motion. The animation frames between the critical frames are filled with in-

between motion progressively. The information stored in the system is the angles of the joints. Motion control methods required handling geometric information like the angles of the joints.

The information for the motion control of the synthetic avatars can be separated into three categories: geometric, physical, and behavioural.

In geometric motion-control-methods, the information is of geometrical nature. Motion is defined in terms of angles, co-ordinates, and other shape characteristics. They are applied to calculated the deformations on the bodies and faces, and determined the skeletal motion.

In physical motion-control-methods, the physical characteristics and laws served as the basis for calculating motion. The information used for physical motion-control-methods included mass, moments of inertia, and stiffness. Physical laws helped controlling the skeletal motions, and the face and body deformation calculations. These deformations usually applied to the muscles in the virtual bodies and faces.

Behavioural motion-control-methods defined an avatar's motion in terms of behaviour, which is referred to as the way that animals and human act [14].
It is complex to displayed behaviour in avatars, because they vary among avatars, depending on the type and form of avatars. For example, if the avatar is feminine, then the walking motion and sitting posture will be different to an avatar representing a male user. Therefore, behaviour motion-control-methods would be less formal than geometric and physical motion-control-methods.

2.3.3.3. Avatar motion and Virtual world

Magnenat-Thalmann and Thalmann [49] described the relationship of the synthetic avatar with the virtual world as actor interfaces, which has their own sets of motion-control-methods.
Actors interface contained four basic cases:

- Single avatar situation: The synthetic avatar does not interact with other objects. The motion of the avatar is simple.
- Avatar-environment interface: The synthetic avatar is moving in the environment and it is awaken by its environment. The environment will affect the motion of the avatar. The motion of the avatar depended on the forces that exist in the environment. For example, if a steep path exists in the environment the motion speed of the avatars walking movement will changed and the physics needed to be taken into account.
- Avatar-avatar interface: The synthetic avatar responds to the action performed by the other avatar. The motion of the avatar is dependent on the motion of the other avatar. For example, if avatar A pushes avatar B, the possible motion of avatar B can be falling, moved slightly or nothing happens depending on the physics of avatar B (size, weight, etc.).
- Animator-avatar interface: The synthetic avatar responded to the action performed by the animator. Therefore, the motion of the avatar depended on the action or motion of the animator.

2.3.3.4. Retargetting motion

Although we have defined the avatar's motion control, data capture used for animation and, determined the body structure to aided the animation of avatar motion, it will be more efficient, if we can reused the motion controls and other information that we had determined from one avatar onto another avatar. This is known as "Retargetting motion", which adapted the motion animation from one character to others that might differed in geometric size and structure appearance. For example, we first developed the tall adult avatar walking motion, but we want to extend it for a small child avatar. If this motion is applied directly to the small child without retargetting, the feet of the child avatar during walking motion will not touched the floor, as the constraint of the feet touching the floor is not re-adapted in the body motion animation.

Apart from adapting the motion for one avatar onto another avatar, retargetting motion is required when motion capture is done. Without retargetting motion, the motion

capture data cannot be applied to the avatars that had different sizes or proportions than the actor from whom the motion data was obtained using the motion capturing devices. There are few techniques tried to tackle the retargetting problem. In general, users are restricted to adapted motions using the same tools that created the motion which each frame is manually tuned in the tool. Kinetix's Character Studio [79], is a commercial system that supported retargetting motion animation. In Character Studio, the keyframes are adjusted to maintain the feet-steps and the balance of the motion, when it is re-applied to a new character. (Figure 2.7.)

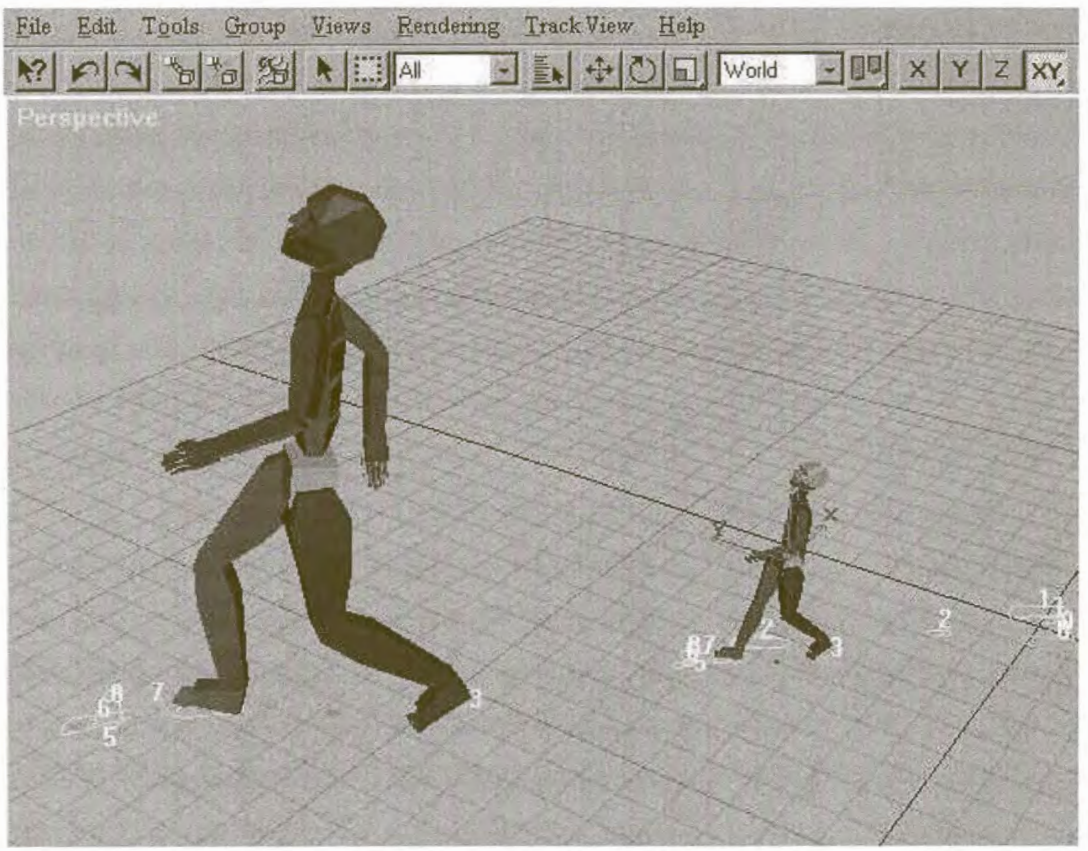


Figure 2.7. Retargetting process showed under Kinetix's character Studio R2.1, when the same motion is applied to characters of different sizes and the motion is re-adapted to each character without distorting the walking motion [79].

Hodgins and *Pollard* [81] addressed a variant of the motion re-used problem and adjusting the parameters of a physical simulation to adapted a controller used for a new character or a character that is changing it's size and shape. The procedural and

simulation based approaches for animation offered representations that are independent from the character, which generated new motions for new characters. Procedural and simulation controllers are able to adjusted different characters easily. However, these methods do not addressed the problem of retargetting, they can generated new motions for new characters, but do not reused existing motions. Re-generation of motion risked losing qualities in the original. The goal of these approaches are to created methods that adapted existing motions obtained from various sources, including the motion capture and keyframing as well as simulation and procedural generation.

In recent years, there is an interest in tools that allowed motion to be altered in ways that are independent of how it was created in the system. At their core, these tools treated animated motions as time-varying signals and applied signal-processing techniques to these signals. Litwinowicz's Inkwell system [80] demonstrated the utility of applying signal processing methods to animation data. Perlin [82] showed how existing motions could be blended together, and how the addition of noise to a motion could be used to transform it. While Gleicher [83], solved the retargetting problem by finding the adaptations required and the setting constraints for each motion that is applied to a new character. If these constraints are violated, re-adaptation process will occurred for the new character. Motions therefore can be reused on other characters independent to how they were created for the first character.

2.4. Facial expression and emotions in Avatars

Facial expressions are the best indicator of a person's mood, emotion and general "state" in the real world.

The most important facial-feature that indicates variety of emotions without other features, are the eyes. As seen in cartoons and comic strips, the simple characters have only eyes to show their facial expressions. Eyes on its own can communicate many basic emotions, but together with other facial-features more information is conveying to the respondent.

2.4.1. Designing facial expressions in avatars

Humans seek meaning through conversation and interaction. In a virtual environment, people interacted with other people through avatar interaction. Therefore, avatar interaction is not complete without facial expression and body motion. Some avatars in the virtual environment represent real people. During the interaction with other users in the virtual world, we want to know the emotion of the other user while interacting with them. Therefore, the avatar must be able to show emotion through facial expressions.

Facial expression analysis

Before facial expressions can be animated, expressions must be analysed, a head mesh must be generated, and texture mapped correctly, to allow expressions to be displayed. Face analysis can be divided into face detection, facial feature extraction, and facial recognition.

Face detection involves finding the position of the face of the actor. Facial feature extraction is concentrated around the facial features such as eyes, mouth, or eyebrows, which identified their position within the face of the actor [84].

Recognition, analysis and synthesis of expressions are often achieved using the user's expressions to drive the animations (tracking of the user's face) [53][54][55].

The expressions on the avatar's face can be determined by tracking the expression of the user's face and then the system determined the facial expression of user, this process is known as facial recognition.

There are two approaches to perform facial recognition.

Firstly, the facial feature markers are set on the video input. Capin et al. [42] defined a "Soft Mask" to manipulate the changes in the markers as the user's facial feature changed positions and orientation.

The alternative approach used markers (paint mark or beads on the face), that are on the actor/user's face and tracked them [77][22] (Figure 2.8.). The markers are usually in brighter and distinct-colours, which are easily identified by chroma keying. These markers simplified the expression recognition process and they are placed around the important facial features, eyes, eyebrows, nose, and the mouth. The limitation of this

approach is that these markers are irritating for the users, as they markers are stuck or painted around the facial features of the users.

Figure 2.8. Markers are placed on the actor's face for facial recognition [22].

Guenter et al. [52] created a system for capturing human facial expressions and replayed them as a highly realistic 3D "talking head" consisted of a deformable 3D polygonal face model with a changing texture map onto the model. In their system, fluorescent coloured paper fiducial are used as markers on the actor's face, which is used to tracked the facial motion as input data for the system.

Functional control can be used to interpolate and replay expressions that are predefined in the system. This allows expression to be created through the control panel interface manually. Expressions of the user's image from the video can also be texture mapped onto the face of the avatar directly [42].
This is unfeasible, because it required more bandwidth to stream video images and mapped them to the face mesh of the avatar continuously, than sending the data value of the position of each facial feature to the application. In case of error in transmission, the facial animation will not be smooth and the head movement will be rigid.

Waters [16] defined a facial expression of a person by muscle movements in the face. He simulated the muscle movements to generated facial expressions on the avatar's face. Therefore, facial feature values are sent to the system and the fast algorithm implemented the changes affecting the expression of the avatar. Expression is therefore broken into virtual muscle movements. Similar to Waters' approach Magnenat-

Thalmann et al. [24] described the concept of abstract muscle action procedure (AMA procedure), which simulated the specific action of a face muscle. Each AMA procedure corresponded roughly to a muscle or bone structure. AMA procedures are not independent of each other, so the order of action is relevant. Since human muscles are complex, the AMA procedures must simulated the same motion without imitating the complexity of the actual muscles. It is possible to animate the low-level expressions by manipulating the facial parameters using AMA procedures. By combining different AMA procedures, complex expressions can be animated in the application.

Mesh predefined expression

Facial expressions of the avatar generated by moving, or deforming the specific facial features created on the head mesh [42]. If jaws, teeth, hard palate and tongue is created with the head mesh, lips and speech simulation can also be implemented in the application [75][76]. The amount of movement or deformation on a specific facial feature can be obtained via capturing the user's facial movements, or predefined these expressions in the application. This enables the user to select an expression from the keyboard, which similar to the ones used in mail messages and chat-rooms, which is less complex and do not needed processing like facial recognition.

In lip movement simulation, Lavagetto [3] showed that by analysing the audio signals of speech, it is possible to extract from audio the visual parameters of lip movement. An application doing this recognition and generating motion parameters for controlling the face is connected to the VE program through the facial expression driver.
The facial representation engine developed by Lavagetto, then synthesising the face with the appropriate lip movements. A primitive version of such a system would just open and close the mouth during speech. A sophisticated system would actually synthesised realistic lip movement, which is an important aid for understanding speech.

2.4.2. Texture Mapping the face mesh

Before facial expressions are animated in the application, the face mesh must be texture mapped correctly. The face consisted of detailed facial features, if the texture is not

mapped correctly it will result in an unrealistic face and caused deviation in the facial expressions.

2.4.2.1 Face Texture extraction

Images of the user's head are captured by the video input and used as textures for the face mesh. The image of the user's head is separated from the background using "Chroma-keying", or position the camera close to the user's face to avoid the background appeared on the face texture.

2.4.2.2 Markers removal from texture

If the texture of the actor's face is captured while the markers are still on the user's face, a marker removal process is needed before the texture can be mapped onto the mesh. Guenter et al. [52] described a process for removing the markers and their associated illumination effects from the camera images of the actor. The markers are removed from the camera image sequences by substituting each pixel that is covered by the colour markers with the skin texture. Inter-reflection effects are noticeable at some parts of the face fold, as the reflective surface of some markers come into close proximity with the skin. The diffused inter-reflection effects and any remaining colour cast from stray pixels that have not been properly substituted are removed from the image.

The skin texture substitution begins by finding the pixels that correspond to coloured dots. A marker mask is generated by applying the classifier to each pixel in the image, which marked the pixels that required to be substituted by the skin texture. The skin texture is divided into low spatial frequency and high spatial frequency components. The low spatial frequency components of the skin texture are interpolated using a directional low pass filter oriented parallel to features that might introduced intensity discontinuities. This prevents smudging of colours across sharp intensity boundaries (e.g. boundary between the lips and the lighter coloured regions around the mouth. The directionality of the filter is controlled by a two dimensional mask which is the projection into the image plane of a three-dimensional polygon mask lying on the 3D face model.

Since the polygon mask is fixed on the 3D mesh, the 2D projection of the polygon mask stayed in registration with the texture as deformation is applied to the face model.

All of the important intensity gradients have their own polygon mask (the eyes, the eyebrows, the lips and naso-labial furrows). The 2D polygon masks are filled with white and the region of the image outside the masks is filled with black to create an image with a low pass filter. The intensity of the resulting image is used to control the filter's directional. The filter is circular symmetric where the image is black, far from the intensity discontinuities, and it is directional where the image is white. The directional filter is oriented so that its long axis is orthogonal to the gradient of the image.

The high frequency skin texture is created from a rectangular sample of the skin texture taken from a part of the face that has no markers. The skin sample is high-pass filtered to eliminated the low frequency components. At each marker mask's pixel location the high-pass filtered skin texture is first registered to the centre of the 2D bounding box of the connected marker region and then added to the low frequency interpolated skin texture.

The remaining diffused inter-reflection effects are removed by clamping the hue of the skin colour to a narrow range determined from the actual skin colours. First, the pixel values are converted from RGB to HSV space and then any hue outside the legal range is clamped to the extremes of the range. Pixels in the eyes and the mouth are found using the eye and lip masks. Using this marker removal process, the camera image can be used as texture for the face-model even if the markers on the actor's face are not removed from the face.

2.4.2.3. Determining texture mapping

When texture mapping the face of the avatar, mapping co-ordinates must be determined, so that the textures are mapped onto the face mesh in the correct position.

View- independent texture mapping

In order to support rapid display of the textured face model from any viewpoint, it is desirable to blend the individual photographs together into a single texture map. This texture is constructed on a virtual cylinder enclosing the face model.

Won-Sook Lee, P. Kalra and N. Magnenat-Thalmann[43] used the following approach to mapped textures on the avatar's head. The standard cylindrical texture map (Figure 2.9.), two images of the user (front and side view) can be jointed to form one texture image of the whole head, allowing the texture to be viewed at all angles. By using a cylindrical projection on these images, the front view covered from –90 degree to 90 degree, right view from 0 degree to 180 degree and left one from –180 degree to 0 degree. The front view is best for a certain range, the same for the other views of the head mesh (Figure 2.10.). By using the conventional blending method for wider range and using variation [57], it is easy to blur certain shape of features. The images on the front and side are cropped for specific points.

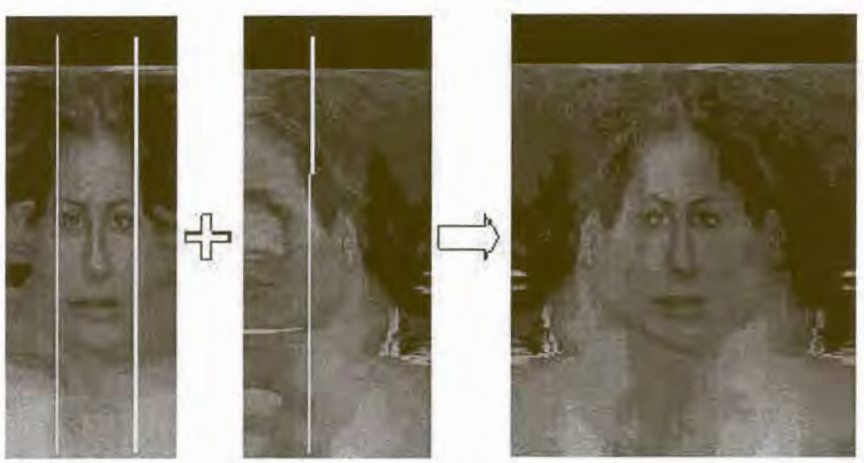


Figure 2.9. A standard cylindrical texture maps the front and side view of the user's face texture together [43].

With the information of the positioning of the eyes, the front view can be cropped automatically, the repetition of the same process for the left and right views and

assembled them to form a facial texture in 360 degrees. Won-Sook Lee, P. Kalra and N. Magnenat-Thalmann used a multi-resolution spline method to assembled two images [58]. Then texture mapped with a composed image onto the 3D-head model by projecting 3D mesh of control points on the image and calculating Voronoi and Delaunay triangulation on the 2D points. The local barycentric co-ordinates of the non-feature points with a surrounding triangle of feature points are calculated in the process, and then determined the texture co-ordinates on each vertex on the 3D surface from a 2D-texture image.

Figure 2.10. The left image indicates the head mesh before mapping. The right image shows the result from mapping the standard cylindrical texture onto the head mesh [43].

View-dependent texture mapping

The main disadvantage of the view-independent cylindrical texture mapping is that its construction involved blending together re-sampled versions of the original images of the face. Due to this re-sampling, the resulting texture is slightly blurry. This problem can be alleviated to a large degree by using a view-dependent texture mapping [93] in which the blending weights are adjusted dynamically, according to the view. For view-dependent texture mapping, the model is rendered several times, each time using a different input photograph as a texture map and blending the results.

View-dependent texture maps have many advantages over the cylindrical texture maps. Firstly, the texture map covered the lack of detail in the model. Secondly, if the model is projected onto a cylinder with overlapping, the cylindrical texture map will not contained data for some parts of the model. View-dependent texture map will contained all the data of the model as the geometry of the mesh matches the photograph.
One disadvantage of the view-dependent texture mapping is its higher memory requirements and slower speed due to the multi-pass rendering. Another limitation is that the resulting images are much more sensitive to any variations in exposure or lighting conditions in the original photographs.

The parts of the mesh that correspond to the eyes, teeth, ears and hair are textured in a separated process. The face usually occludes the eyes and teeth, and it is complex to extract a texture map for these parts in every facial expression. The ears have an intricate geometry with many folds and it cannot be projected without a cylinder that has overlapping. The hair has fine-detailed texture that is difficult to registered across facial expression. Therefore, each of these facial features and hair has an individual texture for realistic head models.
Texture mapping can be applied to a complex and detail head achieved higher level of realism. However, the process of finding the texture co-ordinates is more difficult as the number of vertices becomes numerous.

2.4.3. Facial Animation in Avatars

Over the years, there are various facial animation approaches developed and a distinct generalisation can be made from them. The one is texture manipulation based and the other is face mesh manipulation based.

Texture manipulation of facial animation is achieved by morphing the texture of a neutral face that is pasted on a model of a head or a face mesh (moving pixels in the texture) to a texture with facial expression. The alternative approach is adjusting the texture co-ordinates of the facial features in the face mesh, using texture co-ordinate displacement approach as proposed by Valente and Dugelay [37]. This approach

generated the facial expressions by displacing the texture co-ordinates of the face texture that is texture mapped onto a face or head mesh to simulated the facial expression, which is less in computation and complexity. Since facial animation is done at the texture level, unlike the other approaches where facial animation is done at the mesh level.

For sheer photo-realism, one of the most effective approaches used 2D morphing between the photographic image [91]. The only limitation of this approach is that it requires the animators to specified a few correspondences between physical features of the actor in every frame, and it does not correctly account for changes in the viewpoint or object pose. These approaches use few system resources, low computations and simpler than the approaches that involved morphing the face mesh.

Face mesh manipulation approaches for facial animation can be further divided into low-level muscle motion simulator, known as action units, abstract muscle action procedures, or minimum perceptible actions.
The following are face mesh manipulation approaches for facial animation:

- *Key-framing*, one of the earliest approaches taken, this involved linear transformations from one face mesh to another. This approach involves extensive computation and requires large data set. This approach is inflexible as the number of expressions that can be generated are restricted by the key-frames that were already digitised and it is also difficult to generalised the work on one face mesh to another.

- *Parametric Deformations*, which models the human face as parametric surface and recorded the transformations as control points' movements, in order to minimise the data storage requirements [21]. The limitation of this approach is that it is difficult to generalise over different face meshes.

One attempt is to utilise the B-spline patches that were defined manually on an actual digitised face mesh and the control points of the B-spline patches are moved to effect

the distortion of the face mesh. This method is powerful, but there is no automatic way of defining the relevant control points for the B-spline patch [28].

The other attempt used rational free-form deformation [60] to move points inside a defined volume with respect to the control points placed at the edges of the volume. The volume can be distorted by changing the position of the control points. This approach has the same limitation as the above approach.

- *Anatomically correct muscles*, this approach simulates the actual human face muscles underneath the skin (Figure 10.). This approach can simulate large number of facial expressions, but it is complex and difficult to understand [16][45].

- *Pseudo-muscles*, this hybrid approach simulated muscles that can be anatomically incorrect. Only the muscles that are related to facial animation are taken into account. It is easier than the face meshes manipulation approaches mention previously and it only required small data set [21].

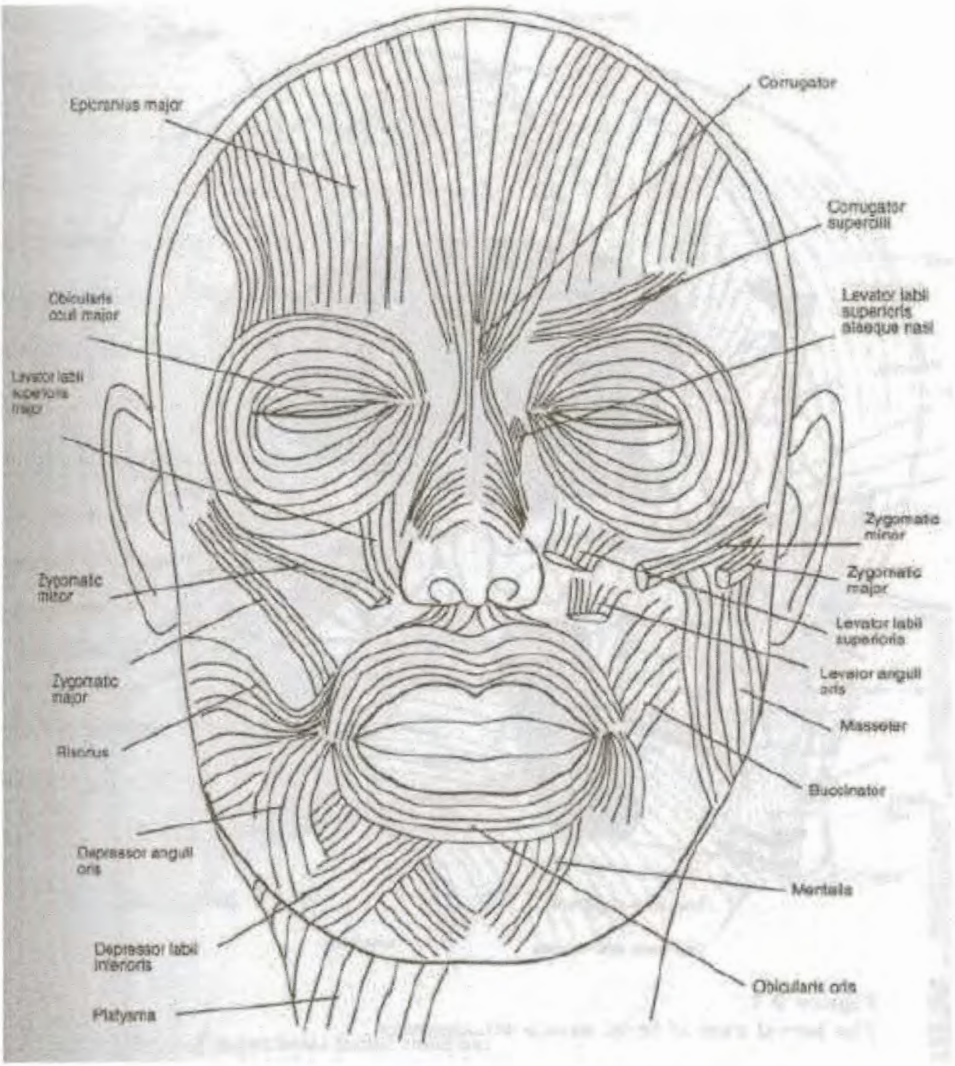


Figure 2.11. The underlying human face muscles are simulated under Water's Anatomically correct muscles approach [16].

MPEG-4 *Facial Animation Parameters*:

The multimedia MPEG-4 standard describes one of the primitive objects defined in this standard is the "face model", which can be displayed and animated according to a number of predefined rules [20]. In particular, FAPs (Facial Animation Parameters) are used to model and encode facial movements. Therefore, the face object is treated as a pseudo-muscular model [87]. Facial Animation Parameters (FAPs) are utilised in the framework of MPEG-4 for facial animation purposes. This enables efficient hybrid

coding for synthetic avatars with natural video and enables the animators to focus on local or global actions on the face, by means of "scripting" the animation sequence. For example, the animator can instruct the synthetic model of a human face to "open mouth" or "close eyes". This instruction is passed to the MPEG-4 decoder, which deforms the model by translating the vertices that correspond to the area of the facial feature to be animated in the animation. The standard does provide for the abstract definition of expressions and emotions as a collection of FAPs, and their subsequent interpolation into intermediate expression. This does not necessarily mean that all possible expressions and emotions can be modelled using this approach [88]. The definition parameters defined by the MPEG Group allow a detailed definition of body/face shape, size and texture, while the animation parameters facilitating the definition of facial expressions and body postures [89]. These parameters are designed to accommodate all natural possible expressions and motions, thus covering not only representation purposes, but also entertainment. This approach provides realistic results, however the limitation of this approach is that not all expression can be modelled using FAPs and it requires expensive hardware to computed complex manipulations of mesh points.

In 3D deformation, the facial expressions are captured from the video and calculated to obtained deformation data for the face mesh to animated facial expression. The result achieved from this approach is very realistic with life-liked face meshes, however the analysis video stream to determined the deformation is a time-consuming process, which required specialise and expensive hardware [52].

Pighin et al. [92] showed how 2D morphing techniques could be combined with 3D transformation of a geometric model to automatically produced 3D facial expression with a high degree of realism. The cameras capture multiple views of the actor with a facial expression; these photographs are digitised to generate the head mesh of the actor from a generic mesh. The texture of the mesh is extracted from the photos. While facial animation is produced by interpolating between two or more different 3D head models, and blended the textures together simultaneously. Since all the 3D models are

constructed from the same generic mesh, there is a natural correspondence between all geometric points for performing the morph.

After the different facial expression models have been generated, transition between expression can be produced automatically without specifying the correspondences between any expressions. This approach is attractive, because interpolating different basic facial expressions can animate complex facial expression. However, creating a set of different face meshes with different facial expression is high computation and time-consuming, while the eyes' gaze is fixed in all face models, unless the user gazed in the other direction in front of the camera.

Self-Evolving Personalities

As Capin et al. [42] suggested, it is possible for the avatar to change their emotions as the avatar's personality changes through time in the system using the object-oriented personality components.

The object-oriented personality components, with their characteristic behaviours, can be re-arranged and interconnected to form a personality matrix specified by the user. Personality component libraries would be available to a user who wishes to construct a "root personality" for their avatar, over which they would have initial total control. The facial expression of the avatar is predefined and they are linked to the avatar personality. However, just as with evolutionary personality growth in humans, the emotional components of the avatar personality would tend to generate their own nuance predilections over time.

2.5. Discussion

Apart from the implementation mentioned above, it is possible to combine the implementation of video avatar and synthetic avatar to produce faster and realistic avatars for collaborative virtual environment. For example, the video camera can capture the facial image of the user, then project the image onto a 2.5D avatar's head, and use a synthetic body for the avatar. In this way, the user can be identified through

his/her face by other users in the virtual environment. Alternatively, the user can use a Pure Avatar's synthetic body and tracking devices for better interaction and communication to other users in the collaborative virtual environment.

The selection of an appropriate approach for creating avatars depends on the type of interaction required for the application, the speed of system processing and the realism required. The methods for implementation can be selected based on their trade-off and advantages, or combination of the approaches. Currently, one of the fastest growing fields for avatars is in the entertainment industries. In the gaming video sequence, guided avatars act the introduction or the in-game cut-scene video through the guidance of the animator. While in the game, the autonomous characters/avatar represented the background characters or opponents, and the users are represented by guided avatars. The users can also interacted with other users using guide avatars in the gaming environment in multi-player mode.

Besides the academic organisations and gaming industries, research regarding avatars is conducted by many institutions from around the world.

The *IMPROV project* by NYU Media Research lab. This research project builds the technologies to produce distributed 3D virtual environments in which human-controlled avatars interacted with computer-directed avatars in real-time, through procedural animation and behaviour scripting techniques.

VET (Virtual Environment for Training), and the Steve avatar. At the Educational Technology Group, USC Information Sciences Institute is another example of research in this area

OZ project (CMU) (Avatar for Entertainment) Carnegie Mellon University (CMU). The Oz Project at CMU is developing technology that combines art to help an artist create high quality interactive drama, based partly on AI and Avatar Technologies. Creating avatars in an interesting dramatic virtual world.

MIT Synthetic Characters Group.

The goal of the Synthetic Characters Group at the MIT Media Laboratory is to understand how to build interactive characters that come alive in the eyes of the users who interact with them. They combined the ideas and disciplines of animal/human behaviour, traditional character animation, AI, robotics, avatar modelling and computer graphics.

Avatars aided in many applications, simulating events in the real world or the virtual environment, enabled distant users to participated in meetings and communicated in the same virtual environment. However, without the existence of avatars, interactions in the virtual environment will be impossible or non-stimulating.

In the Coven experiment [65], the users communicated in a virtual meeting environment. The results indicated that the avatars cannot fully represent people, if facial expressions and gestures are not available, then interaction and communicate becomes unnatural, because facial expressions and gestures allowed people to expressed themselves in a meeting or social environment.

Therefore, the avatar development process discussed in this thesis will concentrate on creating fairly realistic facial expressions using the expressive texture approach and generating simple, upper body movement for avatars, that is suitable for interacting in a synthetic social environment. The next chapter describes the expressive texture approach, in creating facial expressions by manipulation of the face texture.