Chapter 3

Expressive Textures

Expressive Textures approach uses textures of images of both the synthetic faces and the real faces captured from video and a simple low-polygon face/head model. It provides an interactive way of fine-tuning and adjusting the underlined model, which allows a more realistic mapping for a special facial image. Further more, it concentrates in creating facial expressions by texture manipulation. The first section of this chapter discusses the process of face model generation in Expressive Textures approach. Section 3.2 discusses how texture images are texture mapped onto the face model and the process of texture mapping adjustment. Later, Section 3.3 discusses how facial expression is created using Expressive Textures. Section 3.4 looks in-depth at the eye animation and discusses how this can be achieved using Expressive Textures. Finally Section 3.5 gives a short summary of the results of the expressive textures approach.

3.1. Face model

There are many approaches developed to generate the face model as discussed in the previous chapter. Different face mesh generation method can be applied depending on the application and the realism required in the application. In expressive textures, the amount of system resources (System Memory, CPU time, etc) used should be low, while providing a fair amount of realism with low computations. This allowed the possibility of extending the approach to be used over low bandwidth networks for distributed collaborative virtual environments, which is improved by a sense of mutual awareness using facial expressions. Therefore, keeping the resources used to a minimum allows the application to support more face models in a single virtual environment. If the whole avatar body is implemented in the virtual environment, extra system resources can be used to focus on computing body animation of the avatar.

First the face model will be designed with low polygons and simple geometry, so that texture mapping face images on the face model can be done more easily, and allow animated facial expressions, while keeping the system resources usage and computations lowest.

In this thesis a generic, low complexity face mesh is manually created, which different images of real or synthetic human faces can be mapped onto it. The reasons for creating a generic face mesh than using scanning equipment generated a face model from a human subject is:

- Scanning equipment is expensive.
- The face model result from a scan have too many polygons/vertices, which increased face model complexity, computations (e.g. texture mapping computation) and usage of system resources.
- The face model is difficult to map texture onto it, because there are numerous texture-mapping co-ordinates and the amount of texture-mapping computations is large.
- The face models generated from a scan are usually difficult to animated facial expressions, due to large number of control points.
- By creating a generic face mesh, we can avoid the need for creating more specialised face mesh. This also avoids the face mesh morphing computations required re-adaptation from the default face mesh before applied to another face mesh that do not resembled the face of the actor/user.

However, there is a disadvantage by manually creating a generic simple face mesh, the face mesh is less realistic than the face mesh generated from scanning equipment.

To simplify the face mesh, eye sockets and eyeballs are not created in the face mesh. This differs to other face models implemented by other researchers. Since a face model with eye sockets and eyeballs simulates real eye movement by rotation of the eyeballs, but this increases the number of polygons in the face model (Figure 3.1). Therefore in this thesis real eye rotation is simulated at the texture level instead of the face mesh level reducing the number of polygons; this process is further discussed under eye animation at section 3.4.

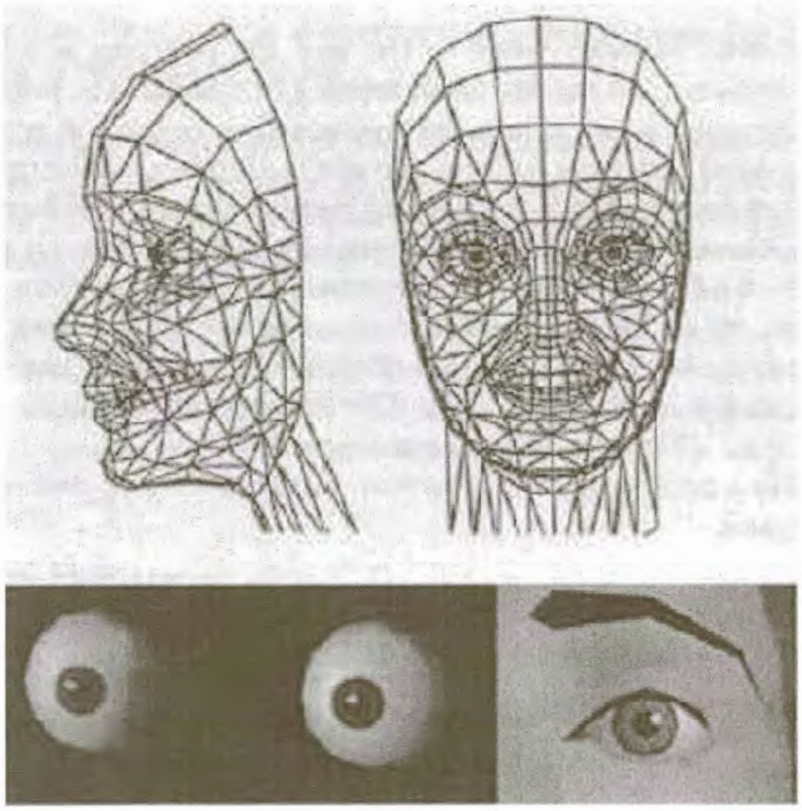
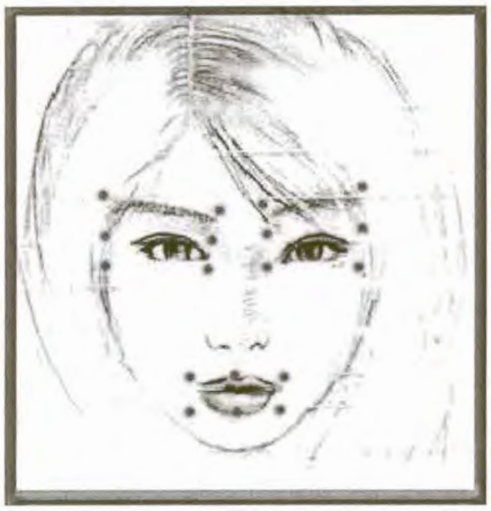


Figure 3.1 A synthetic face mesh with eye sockets and eyeballs, which can simulate the eye animation by rotating the eyeballs [87].

Some facial parts that are not affected under the orientation of the face and can be viewed from different angles – such as the mouth and cheeks are modelled as slightly round surfaces, while the eyes and eyebrows are modelled as slant planes. The nose is modelled so that it protruded from the face mesh, giving the face mesh a non-flat appearance.

Once the face mesh is designed in the application environment, a few vertices in the face mesh are defined as control points (Figure 3.2). Control points are the points in the face mesh, which controlled the facial animation. The control points are set only around the important facial features (Figure 3.3), because the facial expressions are created by the changes at those facial features. In other face models, the control points are usually the vertices that are translated to simulate the movement in the face. In our face model, the control points remain at the same position, only the texture mapping co-ordinates at the control points are translated during facial animation.



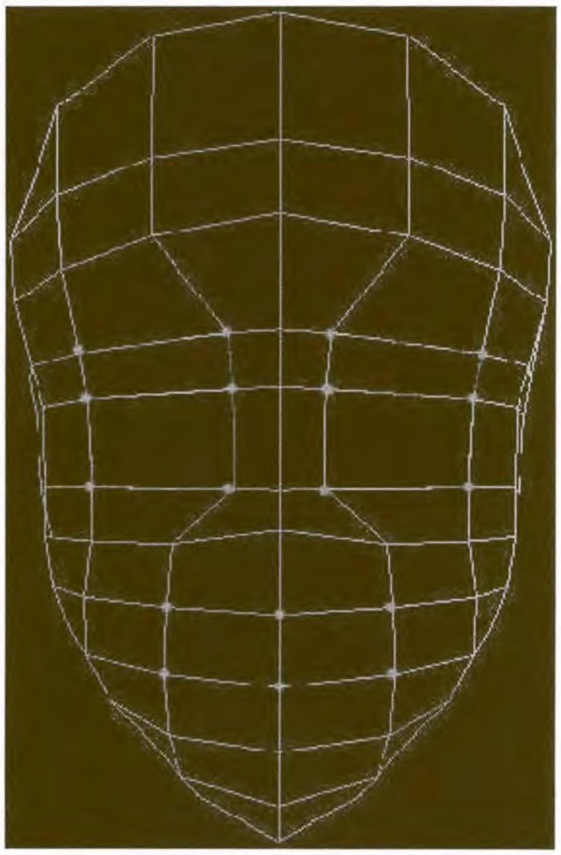


Figure 3.2 (Top)
Indicating the positions of the control points around the facial features, this marks the facial features to be mapped in the same corresponding region in the face mesh.

Figure 3.3 (Right)
The face mesh with the dots indicating the control points.

Although we have defined the control points for the face mesh, animating the face expression is only possible if the texture of the facial features is texture mapped around the control points.

It is difficult to texture map the whole face image onto the face mesh when the number of texture mapping co-ordinates are numerous, so instead of mapping the whole face image at once, the texture mapping process is broken down according to each part in the face mesh. This should facilitate texture mapping of the face image and we can pay more attention to areas around the facial features during texture mapping. Therefore, the face mesh is divided into strips of polygons, which separate the face mesh into regions, so texture mapping the facial features can be done more easily. Since faces vary in size and length, by dividing the face into regions, texture mapping different face images onto the same generic face mesh can be fine-tuned in the application environment.
After generation, the face mesh consists of 138 vertices and 238 polygons with 18 control points. Now we can determine texture mapping for the face mesh.

3.2. Texture Mapping

Texture mapping can be seen as a process where an image is pasted onto objects, but if the position of the image is not determined, the image pasted on the object can be distorted on the object.
Since we have divided the face mesh into strips of polygons regions, so we can determined the texture mapping co-ordinates for each polygon strip regions instead of the whole face mesh. In an image (Figure 3.4), a pixel lies in the pixel co-ordinate system in a square between the value (0,0) and the respective size of the image (width of the image, height of the image). In a texture (Figure 3.5), in OpenGL for example, a Texel lies in the texture co-ordinate system between the values (0,0) and (1,1). Any texture mapping value larger than value one will result in tiling the texture repetitively or clamping the texture. Therefore it is important to establish a relation between the pixel co-ordinate system and the actual texture co-ordinate system.

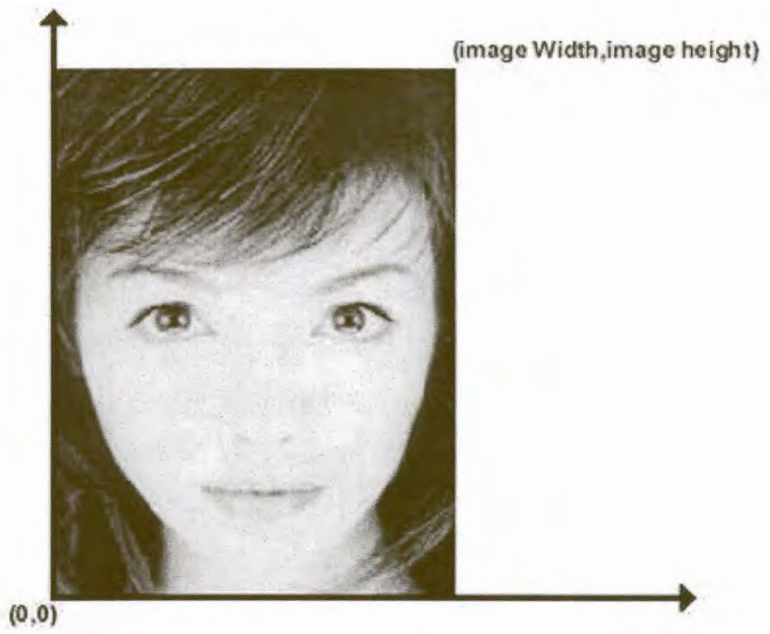


Figure 3.4 The Face Image in pixel co-ordinate system

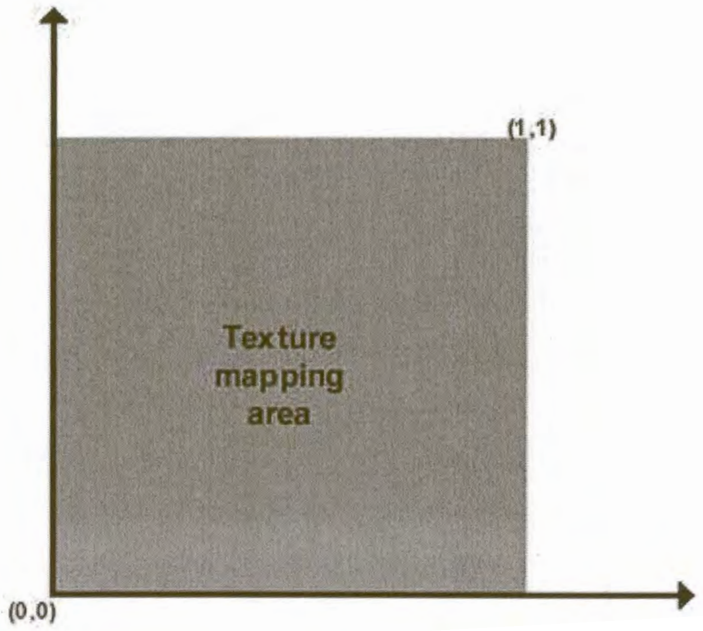


Figure 3.5 Texture mapping co-ordinate system, the texture co-ordinates lie in the texture mapping area will not cause the texture to be tiled or clamped.

Let $\lambda(x, y)$ be a pixel in the image and x and y are the co-ordinates of the pixel in terms of image size. Let $\mu(x, y)$ be the size of the image with x and y as the width and height of the image.

Then the texture mapping position m(x, y) is expressed by:

$$mx = \lambda x/\mu x \quad , \qquad my = \lambda y/\mu y$$

This can also be expressed as:

$$m(x, y) = \lambda(x, y)/\ \mu(x, y)$$

Many image applications can help us resize an image, according to a ratio, so that the face images fitted into the generic face mesh. Unfortunately, we cannot simply resize the image by comparing the image size of the image that fitted the mesh to the one we want to texture mapped onto the face model, so that the image is not from a shoulder height and that it includes only the face.

In Expressive Textures, texture scaling is achieved by first finding the interpupillary distance (distance between the centre of the iris) in the facial image that correctly maps on to the mesh. Then dividing that by the interpupillary distance in the facial image that is texture mapped onto the face mesh. It can also be done, by drawing a perpendicular line between the eyes and the tip of the nose, then use this line as distance for comparing the two images. The distance of the texture image mapped onto the face mesh divides this distance of the new texture image.

These results give the ratio required for scaling the image. Then any portions that are not required in the image are cut away from the final face texture, e.g. shoulders.

Once the still image of the face is scaled to the correct size, it can be mapped to the face mesh using the default texture mapping co-ordinates or adjusted texture mapping co-ordinates (Figure 3.6). The adjusted texture mapping process is discussed later in this chapter.

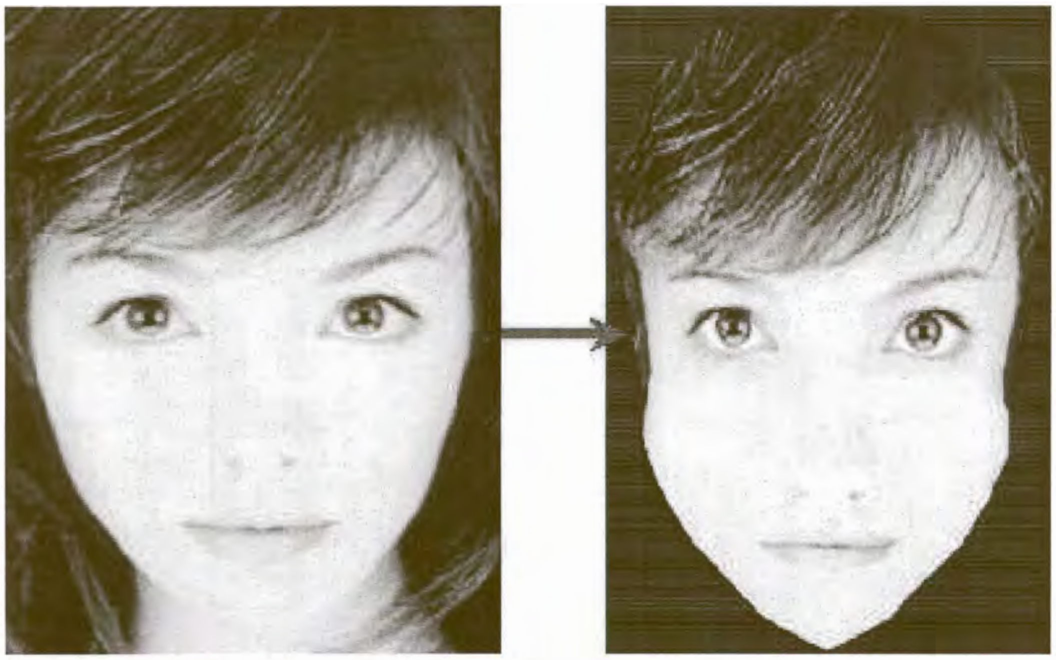


Figure 3.6 A still Photo image (left) is first scaled to the correct size, and then texture mapped onto the face mesh using the adjusted texture co-ordinates (right).

Alternatively, the user can create synthetic images and texture mapped them onto the face mesh using the same texture mapping process.

3.2.1 Video Images

Apart from using still photo images as textures for the face mesh, video images can be captured and texture mapped correctly to the face mesh using the same method as with the still photo images. The video camera must capture the image of the user's face; to avoid unwanted portions map onto the face mesh. After the video image is captured, the video image is scaled to the correct size before it is mapped to the generic face mesh with the default mapping co-ordinates. The default mapping co-ordinates distorted the video face image on the generic face mesh, because the faces are different in length and width. Therefore, tuning is required to adjust the texture co-ordinates for mapping the video image (Figure 3.7).

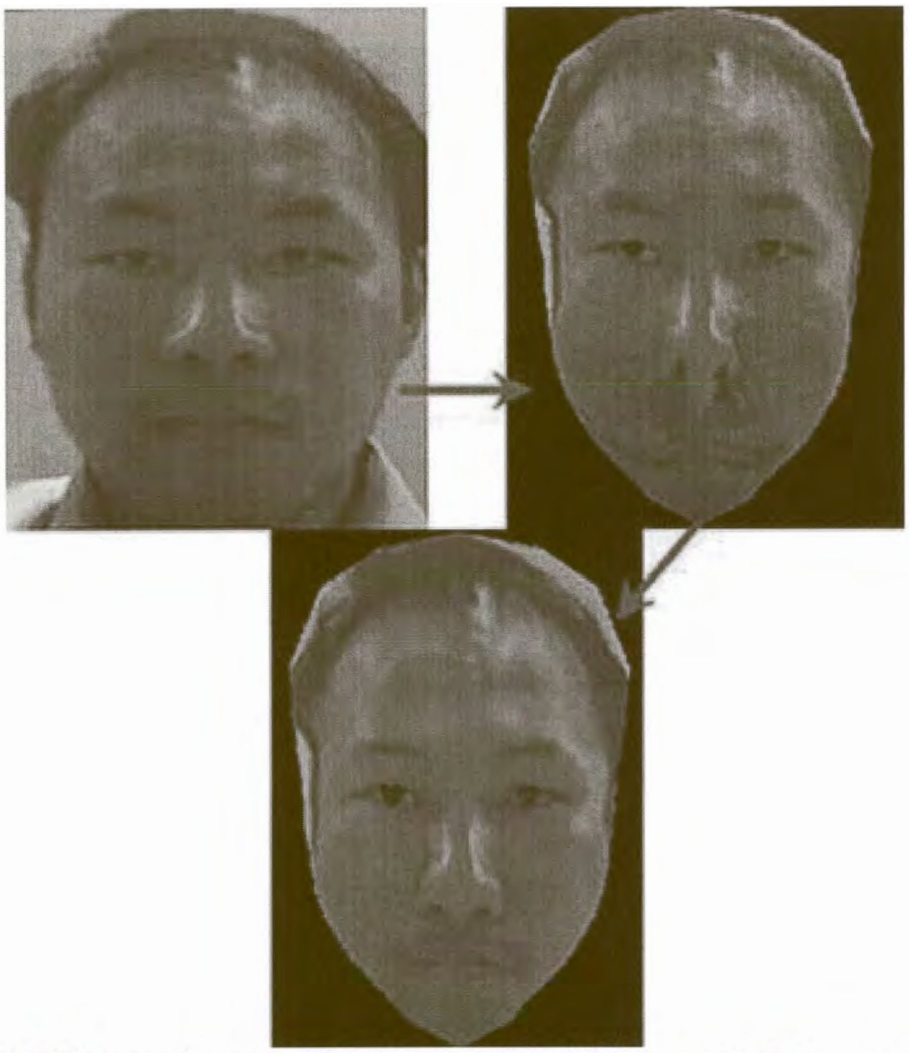


Figure 3.7 The overall process of texture mapping video images to the face mesh, first the captured video image is mapped onto the face mesh using default texture co-ordinates values, then the texture co-ordinates are adjusted to eliminate the distortion.

3.2.2 Texture Mapping Adjustment

Texture mapping co-ordinates can be adjusted at each vertex, but this is very time consuming even if the face mesh is simplified, because there are numerous vertices in the face mesh. Therefore, the Texture mapping adjustment process is broken down into easier steps of texture mapping adjustment. Initially, the default texture mapping co-ordinates map the face texture image's edge onto the face mesh's edge and approximate the texture mapping co-ordinates for the facial features. In some cases, the face image texture might not be in the centre of the image. To accommodated this problem, the texture adjustment allows the user to position the face image's nose at the centre of the face mesh's nose position by shifting the texture mapping co-ordinates moving left, right, top and bottom directions. As the nose is a good indication for the centre of the face, this helps the user to position the face texture before tuning the texture mapping at the other facial features. The face is not flat and varies in length and width, thus it does not correspond to the positions in the generic face mesh. Therefore, some facial features (eyes, eyebrows and mouth) will be distorted or be at the wrong position. The next step in texture mapping adjustment is to position the facial features that are not mapped correctly, because the face mesh is divided into polygon strips texture mapping co-ordinates for the facial features can shifted horizontally. This will speeds up the texture mapping for the facial features. The reason is the eyebrows; eyes and mouth are usually level. Therefore when shifting the entire strip of texture mapping co-ordinates horizontally, we will found the texture mapping co-ordinates for both eyes or eyebrows at once, instead of mapping the one eye first and the other later. Once the facial features are mapped in correct position horizontally, the user can fine-tuned at texture mapping co-ordinates vertically (Figure 3.8).

This texture mapping process allows the user to texture map the overall face image and the main facial features quickly and then apply the fine adjustments at each texture co-ordinate when needed, at a later stage. Since the face image and the mask (an image usually in two colours, e.g. black and white, which marks a portion in the texture image to be transparent or act as filters) have the same size, texture-mapping adjustment is applied to the texture image and automatically to the mask too. This avoids the texture mapping co-ordinates to be re-determine for the mask, and prevents the masking area to differ in positioning at the face image from the mask and mask the wrong area of the face image. Texture mapping adjustment provides a simple texture mapping method with low computation.

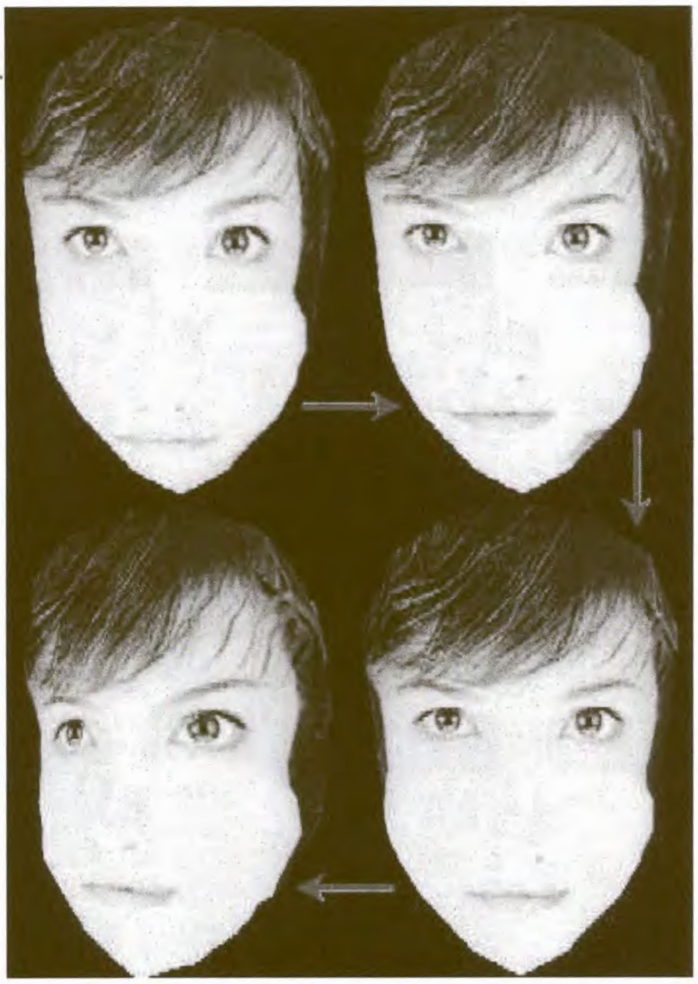


Figure 3.8 The texture mapping adjustment process:
The face texture is mapped using the default texture mapping co-ordinates initially.

Then, the nose is positioned at correctly first and the other facial features are later adjusted based on the nose position. The fine tune adjustment can be applied if necessary. The result will give us the adjusted texture mapping.

3.3. Facial Expressions

Facial expressions result from facial feature movement caused by the underlying muscles beneath the skin contracting and relaxing. To simplify the facial expressions, facial features are only taken into account and the underlying muscle movements are ignored, this makes facial expression easier to animate based on the movement of the facial features. Simple Facial expressions are build up from simple movements of the facial parts namely: eyes, eyebrows, and the corners of the mouth. Psychological research had classified six facial expressions that relate to distinct universal emotions: disgust,

melancholy, fear, happy, anger and surprise [16]. Notice that four of them are negative emotions: disgust, despair, fear, and anger. Only happy and surprise are classified as positive emotions. It is possible to group a few facial expressions with the common facial features, so during animation merging different facial feature movements may expand the number of facial expressions to be animated in the application.

In order to create a facial expression, each facial expression must be generalised into a set of facial feature movements (motion cues). From these motion cues, movement of each facial feature is known for a specific facial expression during animation and they can be applied to different faces. In Table 3.1, is a classification of each facial expression with their corresponding motion cues, it is also possible to add other facial expressions and classify their motion cues.

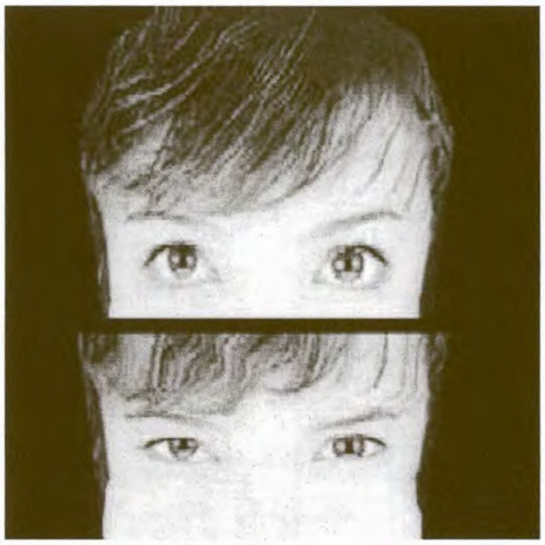
Expression	**Motion cues**
Melancholy	Lowering the mouth corners, raise the inner portions of the eyebrows.
Surprise	Eyebrows slide up, eyes wide open.
Fear	Inner eyebrows raise, eyes open, and mouth lowers slightly.
Happy	Raising mouth corners, lower outer eyebrows, eyes close slightly and upper cheeks expand slightly.
Disgust	Upper-lip raised, eyes close slightly and lower-eyebrows.
Anger	Inner-eyebrows lowered, outer-eyebrows raise and lips press against each other or lower slightly.
Cunning	Eye lids nearly close the eyes, and mouth corners raise slightly

Table 3.1 Motion cues of facial expressions.

Once the face mesh is correctly textured, we can start animating the facial expressions that are classified in the table above (Table 1.). In the face mesh manipulation approach, the control points around the facial feature are moving to simulated a facial feature movement, so the change in control points positions must be determined for the face mesh. Alternatively in Expressive textures, the texture co-ordinates around the facial feature are moved to simulate a facial feature movement. Therefore, the texture co-ordinates for each facial feature under a specific facial expression must be determined and group according to each facial feature. For example, all the texture co-ordinates for the mouth are grouped together. After the texture co-ordinates data of all facial features are grouped, they are stored in the system as different arrays. Consequently, the user can select from the facial cue's movements defined previously and created new, different facial expressions. Additionally, the implementation is simplified and changes in any facial feature's movement will not affect other facial features.

A particular facial expression is animated by moving the texture co-ordinates of all facial features with the current facial expression towards the target facial expression's facial feature texture co-ordinates progressively. For instance, to simulated the closing eyes, the two upper texture co-ordinates can be moved closer to the two lower texture co-ordinates of the eyelid texture (Figure 3.9). To simulate eyes completely closed; this cannot be achieved by moving the two upper texture co-ordinates lapping over the two lower texture co-ordinates of the eyelid texture of an open eye. This is because the eyes texture will be extremely distorted and make the eye look unrealistic. Therefore, another still image of the same face with the eyes closed is required, since both face images can be merged to simulate eyes blinking or eyes closed (Figure 3.10).

Although displacing the texture co-ordinates simulated fairly realistic facial expressions, but in some cases, this is not sufficient. When a person smiles, the face muscles around the cheek contracted causing the cheek to expand slightly. Morphing of the face mesh is required to achieve this effect. When the happy expression is animated, the vertices around the cheek region in the face mesh will expanded slightly, to simulated the face muscle contraction during a smile. The application can now simulate the six universal expressions and other expressions by combining the motion cues defined in the system (Figure 3.11).



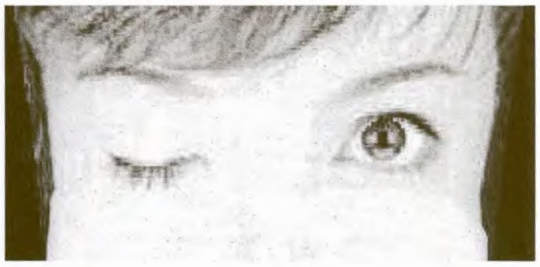


Figure 3.9 (Left image)
The eyes on top are at neutral position, while the eyes at the bottom shows the effect of moving the upper texture co-ordinates closer to the lower texture co-ordinates that map the eyelid texture to simulate eye close.

Figure 3.10 (Top image)
The face image's right eye is merged with the image of the face with closed eyes.

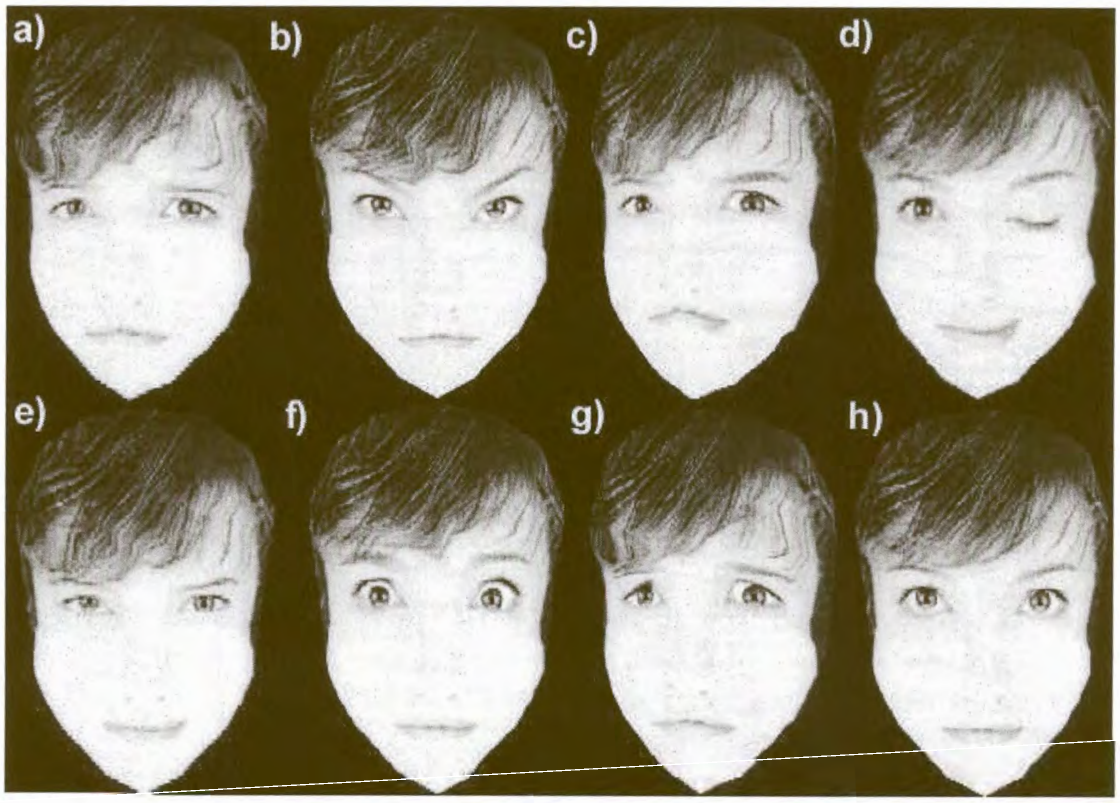


Figure 3.11 The facial expressions animated using expressive textures: a)Melancholy b)angry c)disgust d)happy with left eye closed e)cunning f)surprise g)fear h)neutral

3.4. Eye Animation

The human eyes are considered one of the most important features in the face for facial animation, because at times of extreme anger, showing a sign of boredom or a shy expression, we will usually avoid eye contact. Especially in virtual conferencing [30] the eyes can convey the attention of the user to the collaborating partner. If the user's eyes are directed away from the partner, this shows a lack of gaze awareness and general disinterest. If the user's eyes are directed at the partner, it creates eye contact and an attentive facial expression [94].

In order to show eye gaze, expressive texture approach must use a technique to simulated eye rotations even when still images are used in the application. This is more complex compared to the approaches that uses a face mesh with eyeballs and eye sockets, since simulating the real eye movements can be achieved by simply rotating the two eyeballs.

In expressive textures, although the eyeballs are flat in the face mesh, the eye socket must receded into the face mesh. This is important for creating a realistic look when the face mesh rotates up, down, left and right. First, we will need to extract the pupil texture from the face image and create a hole on the face image. This is achieved by using two masks; one mask is applied onto the face image to generate a pupil texture with only the pupils visible in the original face texture. The other texture-mask creates a hole at the eyeball position in the original face texture using the transparency channel. After both textures are generated, the face texture is mapped on top of the pupil texture with the pupils appeared under this opening (Figure 3.12). To simulate rotation of the eyes, the texture co-ordinates of the pupil texture are displaced on the face mesh.

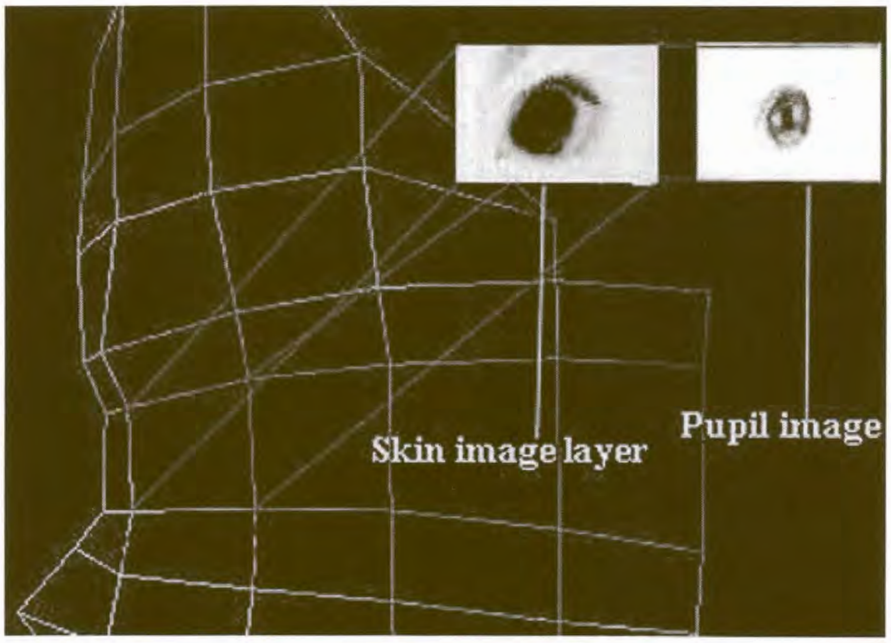


Figure 3.12 Image of the skin layer with a hole created by the masking that appears over the pupil image

The problem arising from this approach is that the speed of the eye movement cannot be adjusted in the application, so this approach will required to calculated the eye movement before the pupil texture co-ordinate is moved towards the target co-ordinate. The eye movement is calculated by:

Let "a" be the pupil image X or Y texture co-ordinate.

Let "b" be the target position of the pupil's X or Y texture co-ordinate, "ob" the current position of the pupil's X or Y texture co-ordinate, and "c" be any small value to control the amount of eye movement.

Then the equation is expressed as: $a = a - c^*(b - ob)$;

When animating a facial expression the eyeballs should be able to rotate freely as real eyes, because the facial expressions are independent from the eyeball rotation. With masking, the underlying pupil texture is independent from the image of the skin layer. When a facial expression is animated, only the texture skin layer is morphed according to the motion cues and this morphing process (Figure 3.13) does not affect the pupil texture.

Both masking mode and non-masking mode are provided. In the non-masking mode, only the face texture is enabled and we can simulate facial expressions without providing the masks, but the pupils will be slightly distorted compared to the pupils at neutral expression.

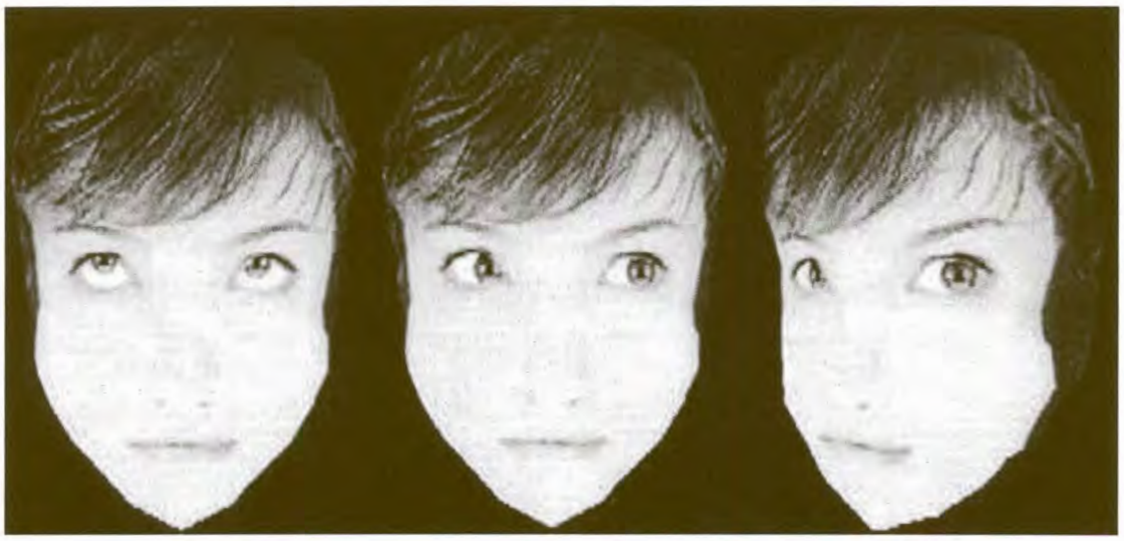


Figure 3.13 The shifting of the pupil texture simulates the rotation of the real eye.

3.5. Summary

The Expressive texture approach for facial animation achieves fairly realistic results with low computations and simple implementation. If an image with skin fold (wrinkles) is available, blending can be applied to simulate movement of the skin on the face. The advantage of using this approach to animate the faces of avatars is that the facial expressions defined in the system can be applied to all the avatars once the face texture is correctly texture mapped onto the avatar's face. The other advantage is that synthetic face images can be mapped onto the avatar to create synthetic avatars and mapped video or photo face images to generate an avatar that represent the users.

The trade-off in this approach is that the hair on the still image will be distorted during animation, if the hair is lying close to the eyes or eyebrows. Another limitation is that mapping an open mouth onto a closed mouth is not realistic.

The next chapter will further discuss how full body avatars are designed in this thesis by combining the Expressive Textures approach, and design upper body animation for these avatars. The next chapter also discusses in detail, the theoretical approach and design of these avatars in a synthetic social-interaction environment.