Customer-Dependent Storytelling Tool with Authoring and Viewing Functions

Sunhee Won[1], Miyoung Choi[2], Gyeyoung Kim[1], and Hyungil Choi[2]

[1] Department of Computer, Graduate school of Soongsil Univ. , Korea
[2] Department of Media, Graduate school of Soongsil Univ. , Korea
{nifty12,utopialove,gykim11,hic}@ssu.ac.kr

Abstract. The animation is the main content of the digital storytelling. It usually has the fixed number of characters. We want a customer to appear in the animation as a main character. For this purpose, we have developed the tool that helps to automatically implants facial shapes of a customer into the existing animation images. Our tool first takes an image of a customer and extracts out a face region and some valuable features that depicts the shape and facial expression of the customer. Our tool has the module that changes the existing character's face with that of the customer. This module employs the facial expression recognition and warping functions so that the customer's face fits into the confined region with the similar facial expression. Our tool also has the module that shows the sequence of images in the form of animation. This module employs the data compression function and produces the AVI format files and throws them into the graphic board.

Keywords: facial expression recognition, warping functions**.**

1 Introduction

Generally we divide to face recognition, facial feature extraction, face expression recognition work with human face. Facial feature extraction consists of face region extraction and face components (eyes, nose and mouth) extraction, facial feature extraction is earlier step of face recognition and face expression recognition, and facial feature extraction is very important part of research for human face. Human face is very sensitive to morphological changes as direction of human face (degree of frontal, either side), size of face image according to distance between camera and human face, and external changes as differences of face region's intensity for illumination, complex background or occlusion for other object. So research of face extraction has many difficulties and problems, and last until these days continuously make research in this field.

Method of analysis projection to extract the facial feature[4] that one way of the basic facial feature extraction divide into region using analysis of each element feature for x, y coordinate. This method has constraints as simple background and sensitive to rotation, so these constraints must be satisfied to get fine results. Second, method using deformable templates that make a template of become a target face and

J. Jacko (Ed.): Human-Computer Interaction, Part III, HCII 2007, LNCS 4552, pp. 1000–1009, 2007.

detect face as compare with input image. This method is robust for illumination change and background, but it has a weak point that sensitive to gradient, rotation angle in the change size of face or face direction along to distance and difficult to produce average templates include information of each person. Third, a method used in this paper that estimate energy of image, it defines that contour is minimum energy and energy function is spline curves. This measure is active contour model (snake) converged on contour of object through process of minimizing the energy from initial position of contour. It has merit that detect with activity for deformable object but it must be given initial function to detect a minimize direction and it is heavily influenced by object around noise. But background is very simple and initial position of snake point is previously given in our system, so we use active contour model (snake) that optimal algorithm to detect contour of human face and character face in animation.

After procedure of facial feature extraction, there is another very important step of image processing that warping for face expression. We can expression of character reflect in human face through control eyes and mouth, therefore apply warping function to eyes and mouth extracted from user image. Client can directly extract source line of eyes and mouth to execute the line warping used in our system using MER region of eyes and mouth. We use control points on source line to apply more accurate warping, each point is corresponding to point of destination line.

The system in this paper is implemented that extracted user face to apply warping function proper for character's expression using expression parameter, and then, automatically implants animation for multiple user's face through equal procedure.

The rest of the paper is organized as follows. In section 2, the system architecture is produced. In section 3, the algorithm used in this system (Snake and Line warping) is explained. The experiment results are analyzed in section 4. Finally, section 5 concludes the paper and points out the next research aspects.

2 System Architecture

This system is edit tool that helps to automatically implants facial shapes of a customer into the existing animation images. It is consists of total three modules as FaceExtracter, AniMaker, AviMaker. In next section, define that these modules.

2.1 FaceExtracter Module

This module consists of two modes that extract mode and edit mode follows by figure 2. First step, extract the face, eyes, and mouth components from input user image in extract mode. In our system, we use snake algorithm that contour detection algorithm for extract user face, and it defined at section 3.1. User directly select the snake initial point by click for extract face contour, maximally close to user face contour cause a defect of snake that it react sensitively by object around noise. Second step, the snake algorithm search for user face contour use this position of initial snake point, we make second mode that edit mode for case that search for unacceptable face contour. Customer can select the edit mode for color image or binary image, this mode consists of three functions as Move, Erase, Draw. Move function used in binary image, in case of snake point go astray cause snake energy function is influenced by noise, it function is moving that point by mouse controlled for right direction. We can obtained clearly and completely face contour by this function use. Erase function is a function that removes the unnecessary edges cause that if user image has too many edges in binary image, searching for inside or outside of actuality contour. Draw function is function that draw new edge for case of edge of actuality contour is weak. We can obtain more clearly face contour by these three functions, and can extract user face to use this contour.

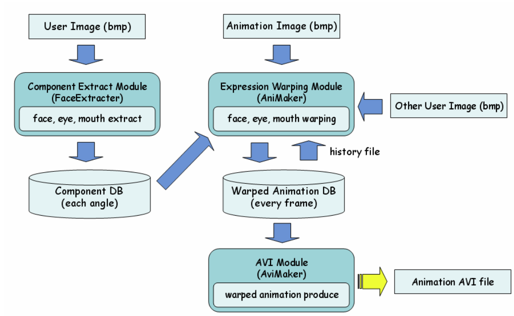

Fig. 1. System architecture – this system consist of total 3 modules as FaceExtracter, AniMaker, and AviMaker

After extract to user face, extract eyes and mouth components by use MER. Customer should draw MER maximally close to eyes and mouse for obtain natural warping result. And then, specify the angle of user face for tilt and panning. Next step for saving, saved that four components image files (.bmp) of extracted face, two eyes and mouth, information file of direction (.inf) and project file (.prj) at once. These files are important data for warping of user image in AniMaker.

Fig. 2. FaceExtracter

2.2 AniMaker Module

This module is user face image extracted from FaceExtracter overlapping with each frame from animation. First, extract character's face using snake algorithm. But, at this time we need rather position of character's face than completely contour of character's face, so estimate a center of gravity from information of contour using snake algorithm. And then, user project file (.prj) accesses from user database and load user face model adjust to character's face angle. When customer want to change that user facial expression adjust to character's facial expression, customer can select to expression type of eyes and mouth. Eyes type can selected from five general expression type – "open", "close", "smile", "cry", "anger" , and mouth type can selected from four type – "Um", "A", "E", "U".

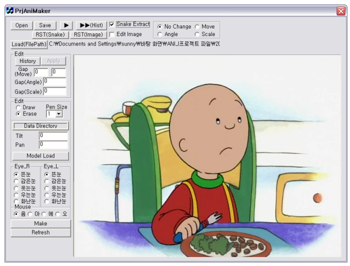

Fig. 3. AniMaker

User facial expression can make using these types and apply warp to extracted face, eyes, and mouth component image from FaceExtracter module, details of warping is explained in section 3.2.

After overlapping face and making facial expression, customer can control user face image adjust to size and angle of character's face for three functions – Move, Angle, Scale. First, customer can move user face using Move functions when use face out of line with position of character's face. Second, angle can controlled using Angle function when need to adjust the angle of user face. Third, customer can control the scale of user face using Scale function.

After apply warping with all steps, saved that transformed animation file (.bmp) and history file (.hist) for automate produce function at once in saving step. Apply warping with each frame go through previous all steps, if customer want that apply other user face image to same animation, then we make a skill that transformed animation is automatically produce function using user's history file without repetition of same stage. A history file for this skill has the information of snake point conclude to angle, size, and position, the information of type for expression of eyes and mouth, the information of user image file. Using this information can produce transformed animation for other user face image equal to previous user face image. If user face image is not fitted to animation image, customer can modify to position, size, angle in total animation with input the gap value in Move, Angle, Scale.

2.3 AviMaker Module

It is a module that make to actuality animation format of transformed animation image using AniMaker and general animation format is AVI using OpenCV library.

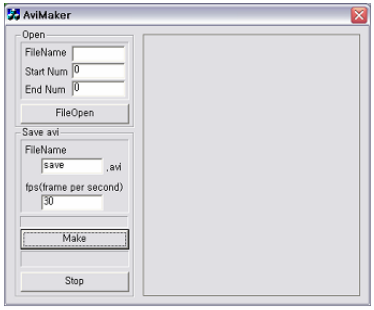

Fig. 3. AviMaker – input Last frame number, FileName, fps

This module produce to transformed animation file of AVI format with information of last frame number, filename, fps(frame per second). Follows figure 4. is a screen captured of AniMaker.

3 The Algorithm

In this section, explain implemented algorithm for our system. First, snake algorithm for extract to face contour explained in section 3.1 and warping algorithm for facial expression explained in section 3.2.

3.1 Active Contour Model (Snake) Algorithm

The active contour model algorithm, first introduced by Kass et al., deforms a contour to lock onto features of interest within in an image [3]. Usually the features are lines, edges, and/or object boundaries. Kass et al. named their algorithm, ``Snakes" because the deformable contours resemble snakes as they move. A snake is defined as an energy function. To find the best fit between a snake and an object's shape, we minimize the energy follows by equation (1).

$$E^*_{snake} = \int_0^1 E_{snake}(v(s))ds = \int_0^1 \left[E_{\text{int}}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))\right]ds \ . \tag{1}$$

Where the snake is parametrically defined as $v(s) = (x(s), y(s))$. $E_{\text{int}\,ernal}$ is a internal spline energy caused by stretching and bending, E_{image} measure of the attraction of image features such as contours, and E_{con} measure of external constraints either from higher level shape information or user applied energy. First, the internal energy provides a smoothness constraint. This can be further defined as equation (2).

$$E_{\text{int}} = \alpha(s)\left|\frac{dv}{ds}\right|^2 + \beta(s)\left|\frac{dv^2}{ds^2}\right|^2 \ . \tag{2}$$

$\alpha(s)$ is a measure of the elasticity and $\beta(s)$ is a measure of stiffness of the snake. The first order term makes the snake act like a membrane; the constant $\alpha(s)$ controls the tension along the spine (stretching a balloon or elastic band). The second order term makes the snake act like a thin plate; the constant β controls the rigidity of the spine (bending a thin plate or wire). If $\beta(s)=0$ then the function is discontinuous in its tangent, i.e. it may develop a corner at that point. If $\alpha(s)=\beta(s)=0$ then this also allows a break in the contour, a positional discontinuity.

The image energy is derived from the image data follows by equation (3). Considering a two dimensional image, the snake may be attracted to lines, edges or terminations.

$$E_{image} = \omega_{line}E_{line} + \omega_{edge}E_{edge} + \omega_{term}E_{term} \ . \tag{3}$$

ω_i is an appropriate weighting function. Commonly, the line functional is defined simply by the image function follows equation (4).

$$E_{line} = f(x, y) \ . \tag{4}$$

So that if ω_{line} is large positive the spline is attracted to light lines (or areas) and if large negative then it is attracted to dark lines (or areas). The use of the terminology ``line'' is probably misleading.

The edge functional is defined by equation (5).

$$E_{edge} = \left|\nabla f(x, y)\right|^2 \ . \tag{5}$$

Hence, the spline is attracted to large image gradients. i.e. parts of the image with strong edges. Finally, the termination functional allows terminations (i.e. free ends of lines) or corners to attract the snake. The constraint energy is determined by external constraints. This energy my come in the form of a spring attached by the user. Or, the constraint energy may come from higher knowledge about the images in question.

3.2 Line Warping

In this section, explain the execution of warping using control line. First, estimate vertical crossing point between pixel and control line, use displacement information of between pixel and vertical crossing point and position of vertical crossing point on control line, and then execute warping using reverse mapping. Figure 4, control line PQ in destination image is corresponding to control line P'Q', and pixel V in destination image copy to pixel V' in source image. For estimate the position of pixel V' by reverse mapping, search for position of C' in P'Q and search for pixel V' far from C' as displacement of between C and V.

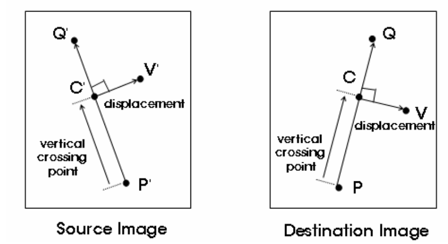

Fig. 4. Information for execute warping using control line

In case of multiple control line, follows algorithm of warping function to estimate each pixel in destination image corresponding to pixel in source image.

```
warping(){
        each pixel v(x, y) of output image,{
            tx = 0           // sum of x coordinate displacement is initialized.
            ty = 0           // sum of y coordinate displacement is initialized.
           each control line L_i ,{
              estimate the vertical crossing point u of V and L_i .
              estimate the vertical displacement of V and L_i .
              get a corresponding position v'(x', y') of input image using u and h .
              estimate the distance d of V and L_i .
                Error! Objects cannot be created from editing field codes.
                 Error! Objects cannot be created from editing field codes.
             }
              Error! Objects cannot be created from editing field codes.
           make a copy of pixel v'(x', y') to pixel v(x, y) .
        }
}
```

First step in warping function, estimate the position of vertical crossing point with each control line L_i for each pixel $v(x, y)$ in destination image. Suppose to u is distance of vertical crossing point $C(x_c, y_c)$ from end point $P(x_i, y_i)$ of L_i , value of u can estimated by equation (6).

$$u = \frac{(x - x_i)(x_{i+1} - x_i) + (y - y_i)(y_{i+1} - y_i)}{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \; . \tag{6}$$

Second step, estimate the vertical displacement h from each control line L_i for pixel $v(x, y)$ in destination image. Formulation of displacement h follows by equation (7).

$$h = \frac{(y - y_i)(x_{i+1} - x_i) - (x - x_i)(y_{i+1} - y_i)}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \; . \tag{7}$$

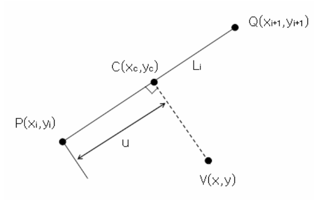

Fig. 5. Vertical crossing point between pixel *V* and control line *Li*

Third step, search for pixel $v'(x', y')$ in source image corresponding to $v(x, y)$ in destination image using u and h. Suppose to either side end point of control line L_i' in source image corresponding to control line L_i in destination image is (x_i', y_i') and (x_{i+1}', y_{i+1}'), $v'(x', y')$ can estimated by equation (8).

$$x' = x_i' + u \times (x_{i+1}' - x_i') - \frac{h \times (y_{i+1}' - y_i')}{\sqrt{(x_{i+1}' - x_i')^2 + (y_{i+1}' - y_i')^2}} \quad . \tag{8}$$
$$y' = y_i' + u \times (y_{i+1}' - y_i') + \frac{h \times (x_{i+1}' - x_i')}{\sqrt{(x_{i+1}' - x_i')^2 + (y_{i+1}' - y_i')^2}}$$

Final step, distance d between pixel and control line can estimated follows by equation (9).

$$d = \begin{cases} \sqrt{(x - x_i)^2 + (y - y_i)^2} \\ \sqrt{(x - x_{i+1})^2 + (y - y_{i+1})^2} \\ h \end{cases} \quad . \tag{9}$$

4 Experiment Result

In this paper, extract user's face, eyes, and mouth of facial feature from animation image and transform user's facial expression adjust to character's facial expression. We use animation that has simply character face and no great change of expression. Also, we experiment with the style of user's hair is very simple, background is not complex and filmed at various direction. User's face direction is left, right, up, and down in the range of ±45 degrees of an angle, and all images are opened eyes and closed mouth.

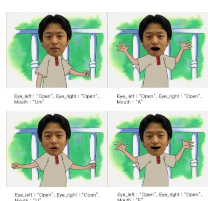

Fig. 6. Change of mouth for frontal view – "Um", "A", "U", "E"

Fig. 7. Change of eyes for frontal view – close, open

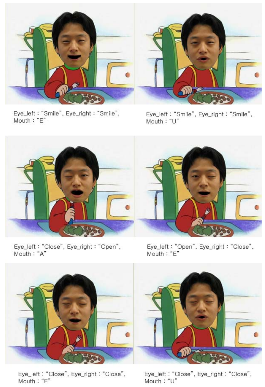

Fig. 8. Change of eyes and mouth for frontal view – various facial expressions

Figure 6 is experiment result that selected opened eyes and changed mouth. Mouth expression type is "Um", "A", "U", "E" as Korean pronunciation, we can see that facial expression is presented only use changing shape of mouth. On the contrary, figure 7 is experiment result that presented only use changing shape of eyes and also figure 8 is presented by changing shape of eyes and mouth.

5 Conclusion

In this paper basically extract to user's face and character's face using snake algorithm and automatically produce transformed user facial expression adjust to character's facial expression in animation. We read up on completely extract of facial feature for more natural facial expression and robust about facial warping at various direction.

Acknowledgments. This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Centerm (AItrc).

References

1. Cootes, T.F., Taylor, C.J.: Active Shape Models - Smart Snakes. In: Proc. British Machine Vision Conference, pp. 266–275 (1992)
2. Cootes, T.F., Taylor, C.J., Cooper, D.H., GraHam, J.: Active Shape Models - Theirs Training and Application. Computer Vision and Image Understanding 61(1), 38–59 (1995)
3. Kass, M., Witkin, A., Terzopoulus, D.: Snakes: Active Contour Models. Internation Journal of Computer Vision 1(4), 321–331 (1987)

4. Garcia, G., Vicente, C.: Face Detection on Still Images Using HIT Maps. In: Bigun, J., Smeraldi, F. (eds.) AVBPA 2001. LNCS, vol. 2091, pp. 102–107. Springer, Heidelberg (2001)
5. Sun, D., Wu, L.: Face Boundary Extraction by Statistical Constraint Active Contour Model. In: IEEE Int. Conf. Neural Networks and Signal Processing, China (December 14-17, 2003)
6. Wan, K-W.: An accurate active shape model for facial feature extraction. Pattern Recognition Letters 26, 2409–2423 (2005)
7. Du, Y., Lin, X.: Emotional facial expression model building. Pattern Recognition Letters 24, 2923–2934 (2003)
8. Mukaigawa, Y., Nakamura, Y., Ohta, Y.: Face synthesis with arbitrary pose and expression from several images – an integration of image-based and model-based approaches. In: Proc. of Asian Conference on Computer Vision, pp. 680–687 (1998)