Multimed Tools Appl (2010) 47:163–187
DOI 10.1007/s11042-009-0412-7

A nonparametric regression model for virtual humans generation

Yun-Feng Chou · Zen-Chung Shih

Published online: 30 October 2009

Abstract In this paper, we propose a novel nonparametric regression model to generate virtual humans from still images for the applications of next generation environments (NG). This model automatically synthesizes deformed shapes of characters by using *kernel regression* with *elliptic radial basis functions* (ERBFs) and *locally weighted regression* (LOESS). *Kernel regression* with ERBFs is used for representing the deformed character shapes and creating lively animated talking faces. For preserving patterns within the shapes, LOESS is applied to fit the details with local control. The results show that our method effectively simulates plausible movements for character animation, including body movement simulation, novel views synthesis, and expressive facial animation synchronized with input speech. Therefore, the proposed model is especially suitable for intelligent multimedia applications in virtual humans generation.

Keywords Image deformation · Nonparametric regression · Elliptic radial basis functions · Functional approximation · Locally weighted regression

1 Introduction

Deforming characters in a 2D image has received lots of interests. Besides, it is very useful for advanced intelligent multimedia applications for next generation environments (NG) utilization, such as character animation [21], real-time live performance [30], and enhancing graphical interface [7]. Based on reanimating still pictures, it has become solvable. For example, Chuang et al. [12] deformed pictures using stochastic motion textures. They animated passive elements which are subject to natural forces like wind. Hornung et al. [20] achieved the motion of photographed persons by projecting them to 3D motion data.

Y.-F. Chou · Z.-C. Shih (✉)
Department of Computer Science, National Chiao Tung University, Hsinchu City, Taiwan
e-mail: zcshih@cs.nctu.edu.tw

Y.-F. Chou
e-mail: yfchou@cs.nctu.edu.tw

In our work, we would take the idea of creating deformations directly in image space one step further by making characters move and creating virtual humans. In practice, a virtual human can be the spokesman or substitute in various services of the NG applications domain, such as remote education, remote diagnosis, network gaming, virtual shopping, digital photo frame, video conference, and so on. In this paper, we propose a nonparametric regression model to generate virtual humans by animating characters from still images, as shown in Fig. 1. Note that the model is trained to fit the shape and detail of the character between two key-poses or moving templates while minimizing unnatural distortion. For instance, animating the character in a comic could be carried out by the creation of a novel view, as shown in Fig. 2. It shows two continuous frames in the original comic that can be regarded as two different scenes and the synthesized frames from a single input frame. Furthermore, the model can be applied to generate virtual humans, who are constructed by different multimedia technologies, such as body movements, novel views synthesis, and expressive facial animation with lips movements from speech.

For body movements and novel views synthesis in virtual humans generation, the proposed nonparametric regression model is trained from two key-poses of a character.

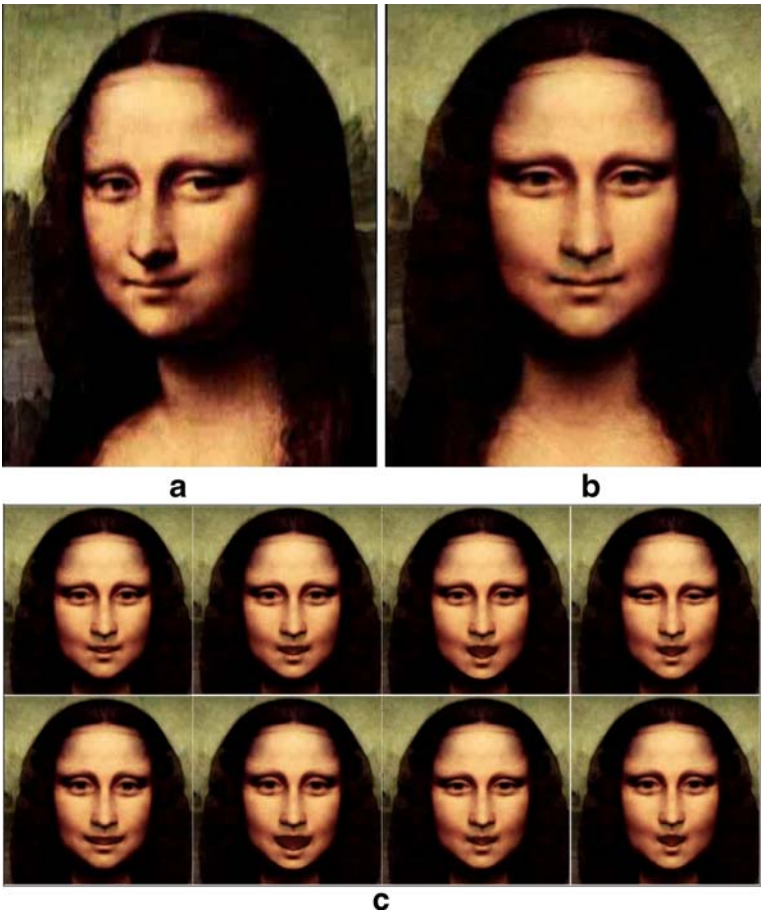

Fig. 1 Virtual humans generation. **a** The picture of Mona Lisa. **b** Novel views synthesis. **c** The lively talking face simulation

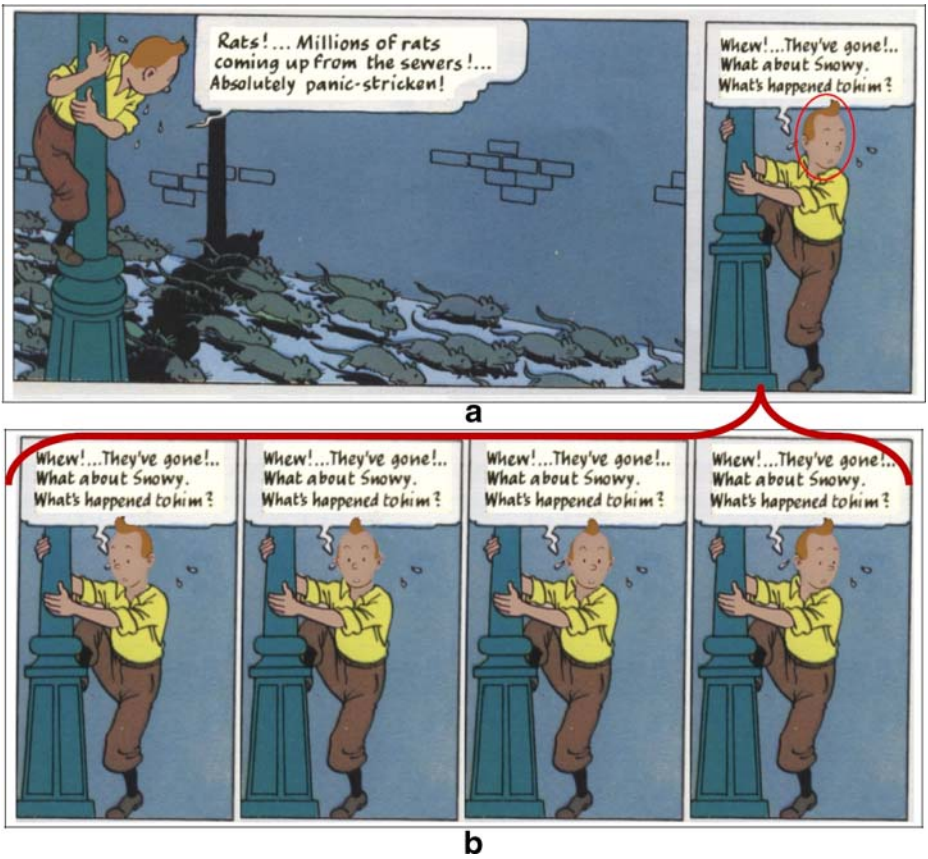

Fig. 2 Novel views synthesis in a comic. **a** Two consecutive frames in a comic. **b** The frames synthesized from a single frame. © Georges Remi (Hergé)

Then the trained model is applied to synthesize the smooth transition between key-poses. For facial animation, the trained nonparametric model is employed to generate synchronized lip movements and drive the stochastic process for facial expression movements by giving any speech data and the relative moving templates.

The proposed model is based on the prediction abilities of both *kernel regression* and *locally weighted regression* [19, 28]. *Kernel regression* approximates the deformed shape of character between two key-poses or moving templates by the prior use of a set of kernel functions. Previously, researchers [48] presented image morphing techniques using *radial basis functions* (RBFs) for the kernel. RBFs are based on spatially-limited circular Gaussian distribution functions. In contrast, circular Gaussian is not an appropriate choice to fit contours, which have noncircular structures, as shown in Fig. 3. Figure 3a is the original character, Fig. 3b using the circular Gaussians needs five kernels to fit the contour of the right arm of the character, and Fig. 3c using the arbitrary directional elliptic Gaussians can fit the right arm and left leg with the same number of kernels. Using too many circular Gaussians increases the learning and fitting time.

In this work, *kernel regression* using *elliptic radial basis functions* (ERBFs), specifically elliptic Gaussians which provide less learning time, is applied for contours fitting during

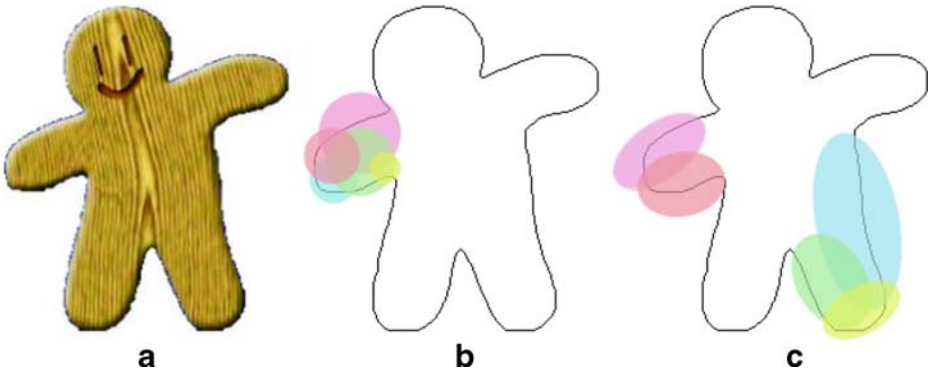

Fig. 3 Comparison of the number of basis functions using Gaussians. **a** The original image. **b** Using RBFs to fit right arm with five kernels. **c** Using ERBFs to fit right arm and left leg with the same number of kernels

shape deformation (shape fitting). Although ERBFs require more computation during optimization, better quality is obtained with smaller number of basis functions.

Except the globally smooth shape fitting mentioned above, the local-fitting methodology is also applied to preserve important features within the contour (that is filling in the color and texture information obtained from the original character in the given image). For example, the wood grain of the character in Fig. 3a. *Locally weighted regression*, or LOESS, is used to preserve the features of details. LOESS is based on the minimized weighted sum of squared residuals. It is a way of estimating the regression surface through a multivariate smoothing procedure by fitting a function of independent variables locally.

In summary, this paper makes the following contributions:

- A novel approach for shape fitting based on *kernel regression* with ERBFs is proposed, which is suited to the natural shape of characters such as the human's head or body.
- By using a closed-form solution of LOESS, a new method for detail preserving is presented, which maintains features invariant during deformations while minimizing unnatural distortion.
- The proposed nonparametric model composed of *kernel regression* with ERBFs and LOESS would be further applied to body movements and novel views synthesis. Besides, it is further used to create lively animated talking faces and synthesize the stylistic variations and mood of facial expression for virtual humans generation.

The rest of this paper is organized as follows. Section 2 presents related works. Section 3 then summarizes the virtual humans generation process. Next, Section 4 introduces the process of character extraction. Section 5 describes the nonparametric regression model for virtual humans generation. The results are shown in Section 6. Conclusions are finally drawn in Section 7, along with recommendations for future research. Additionally, Appendix 1 describes *hyper radial basis functions* (HRBFs) for the inference of ERBFs in more detail.

2 Related work

Various techniques have been applied to animate characters for virtual humans generation in image morphing, image interpolation, view interpolation, shape deformation, and viseme synthesis.

Image morphing In image morphing, several studies [18, 33, 48] referred to as shape blending have been conducted. For example, Sederberg and Greenwood [38] employed an interpolation scheme that can interpolate the length of edges and angles between two keyframes. Besides, RBFs have also been used for image morphing. RBF is a weighted sum of the translation of a radially symmetric basic function augmented by a polynomial term. It is suitable for fitting smooth functions and is used to warp facial expressions and animate images or drawings [2, 25, 36].

In contrast, circular Gaussian is not an appropriate choice to fit noncircular structures. In this paper, we adopt ERBFs to fit contours of characters instead of RBFs. ERBF has the advantage of RBF-like smoothness and is applicable to more general shapes than RBF. Computer graphics researchers may be unfamiliar with ERBFs. Nonlinear approximation of functions in certain general spaces with ERBF networks (referred to as *elliptic basis function* networks in [31]) was proposed. Furthermore, a volumetric approximation and visualization system was developed with ellipsoidal Gaussian functions for a 3D volume (referred to as *ellipsoidal basis functions* in [22]).

Image interpolation Optical flow techniques [17, 23, 24, 34, 42, 43, 49] can be widely adopted for image interpolation. Baker et al. [3] created a collection of optical flow datasets with ground truth. They measured the flow accuracy and the interpolation quality of these optical flow algorithms adopted for image interpolation. While the primary focus of the optical flow algorithms was on evaluating the flow itself. Ghosting and blurring artifacts were visible in their interpolated images even though there were minor errors in the flows. Mahajan et al. [26] proposed an inverse optical flow method. They traced out the path of each pixel between two given images. Then the pixel in the interpolated frame was obtained by moving gradients along the corresponding path and using Poisson reconstruction. Note that they need to determine the flow of each pixel for constructing the path framework. Since these optical flow techniques are based on the disparity of two given images, most of them can only handle two similar images (the disparity or the motion between two images is limited).

View interpolation Besides, several approaches [11, 16, 39, 44] for view interpolation can be applied to generate virtual humans. Seitz and Dyer [39] proposed a method known as view morphing. The input image was prewarped with the image points through the fundamental matrix computed by computer vision or predefined. Then images were transformed onto the same plane such that their scan lines were aligned. Two views were then morphed, and the interpolated images were postwarped with the user-specified parameters to achieve better morphing quality. However, the quality depends on the number of line correspondences made by users.

Shape deformation Recently, Skeleton-based techniques [15, 50] have been used to deform the shapes by manipulating the space in which they are embedded. They are very efficient in computation and easy to be implemented. However, they do not provide convenient or meaningful interaction tools for the user. Note that the weight tuning for rigging is a painful process for users. Besides, shape matching techniques have been used to shape deformation. Wang et al. [46] utilized uniform grids for 2D shapes and maintained the rigidity of each square in the grid by using shape matching during deformations. They implemented pure rotational transformation for each square. Note that the global area cannot be preserved. Botsch and Sorkine [5] deformed a 2D shape by discretizing the shape into finite elements. However, the computation time is dominated by the complexity of the discretization, and not by the intrinsic complexity of the shape itself.

As mentioned previously, Hornung et al. [20] accomplished the motion of photographed persons by projecting them to 3D motion data. However, they stipulated extra 3D information, including 3D model construction or a 3D motion database, thus increasing the overloads which do not belong to shape deformation. Although they can be applied to animate 2D images of arbitrary characters, their system does not work for motions where the character changes its moving direction, or where it turns its head. Furthermore, Alexa et al. [1] considered that the shape deformation of an image should be as rigid as possible. Such deformations would minimize the amount of local scaling and shearing. Igarashi et al. [21] triangulated the input image and minimized the distortion of these triangles in the deformation process by solving a linear system of equations. Schaefer et al. [37] proposed a rigid transformation method by moving least squares. They focused on specifying deformation by using user-specified handles. In order to deform the image, users should set the next pose by manipulating control vertices. Then the method deformed the entire image plane. Since it ignored the geometry of the shape, unnatural distortions or serious artifacts would be generated when the range of controlling handles were exceeded because the locally influencing extent using moving least squares is limited. Weber et al. [47] generalized the concept of barycentric coordinates and provided a few examples of known coordinates which can be used for planar shape deformations. Note that the inputs of these works are images and the outputs are also the edited and deformed images. In comparison, our input is just an image and the output is the whole sequence of interpolated frames.

Viseme synthesis There is a strong correlation between lip movements and speech [27], and a great deal of studies have been conducted on facial animation involve lip-synching. There have been multiple attempts at generating an animated face to match some given speech realistically [4, 6, 13, 14]. Incorporating speech therefore seems crucial to the generation of true-to-life animated faces. Our synthetic faces of virtual humans are also driven by input speech. Furthermore, we reproduce small variations in facial expressions that convey the affective states, mood, and personality of the virtual human. Moreover, the strong interrelation between facial gestures and prosodic features has been reported in the speech processing literatures [8, 9]. However, the interrelation between facial gestures and individual phonemes is not obvious. Our main focus is to synthesize facial animation possibly driven by analyzing phonemes from input speech.

3 Algorithm overview

In Fig. 4, the outline reflects the structure of our proposed method for virtual humans generation. Considering Fig. 4, we briefly describe our method in the following paragraphs.

1. **Character and features extraction:** In order to reduce the effects of the background upon deformations, we first extract characters from the input image. We use level-set-based *GrabCut* to extract characters and features, as described in Section 4. Similar regions are extracted by the level set method. The bounding box of all regions is then used by *GrabCut* [35]. The boundaries of regions corresponding to the matte produced automatically are further applied to obtain the final character matte. The foreground and background are separated successfully. Besides, the facial features are extracted simultaneously by the level set method. As shown in Fig. 4, Mona Lisa is extracted, which is described by the similar parts found by level-set-based *GrabCut*, and the contours are applied to build the nonparametric regression model for shape fitting and detail preserving.

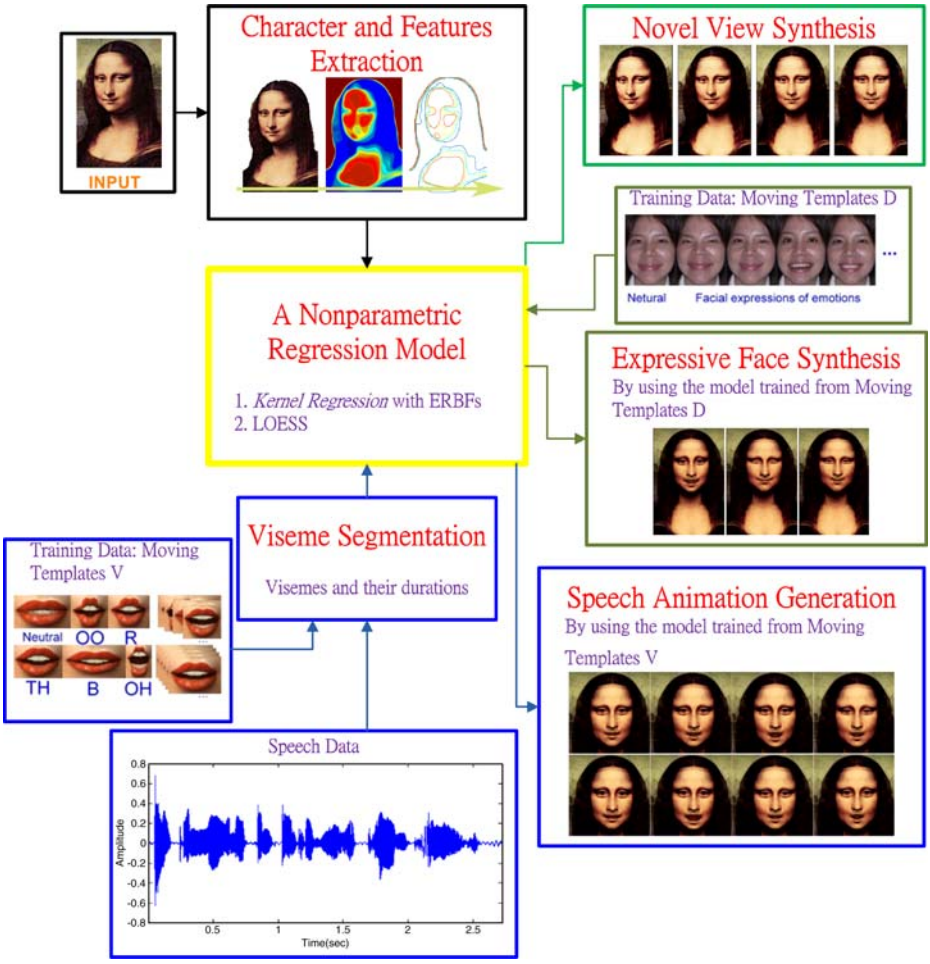

Fig. 4 The overview of virtual humans generation with the picture of Mona Lisa

2. **Nonparametric regression model:** We propose a nonparametric regression model to fit the movements of virtual humans. Movement fitting can be divided into two models: shape fitting and detail preserving. In the shape fitting, the correspondence in training data set is constructed first. *Kernel regression* with ERBFs is employed to train the first model to fit the contour of deformed shape, as described in Section 5.3. In the detail preserving step, the second model is trained by LOESS, as described in Section 5.5. Then we would fit the details of deformed shape. LOESS is suitable for detail preserving in accordance with the previously fitted contours.
3. **Training data:** For model learning, according to different multimedia applications, the nonparametric regression model is trained from different training data sets. As mentioned before, the model is trained from two key-poses of a character for body movements and novel views synthesis. To generate facial animation, the training data set consists of two extreme templates in moving templates V for visemes synthesis, which are the neutral mouth shape and the mouth shape with

vowel /o/, and two emotional states in moving templates *D* for facial expression simulation, as described in Section 5.6. The moving templates are the base target positions of the facial features to be animated. The positions of mouth shape are recorded in moving templates *V*. Moving templates *D* focuses on the positions of all facial features, as shown in Fig. 4.

4. **Body movements and novel views synthesis:** After model learning, we would fit the movements of virtual humans. Given two key-poses of a character, we would deform the shape of the character to synthesize body movements by using the nonparametric model trained from the key-poses. Moreover, we could apply one key-pose and its reverse to train the model for generating novel views of the character.
5. **Viseme segmentation:** In viseme synthesis, the goal is to model the correspondence between facial motion and speech. Viseme segmentation, which means to determine all visemes and their durations, is done from the speech data. Note that it is to align phoneme labels to the audio stream, and use this information to label the corresponding lips movement, as described in Section 5.6.
6. **Expressive face with speech animation:** Given moving templates *D*, we can use the trained nonparametric regression model to synthesize facial expression of any emotional state in moving templates *D*. In addition, we collect moving templates *V* and speech data, whereas the speech data is the voice data. After viseme segmentation, we would convert to the corresponding mouth shape and synthesize speech animation by using the trained nonparametric regression model.

4 Character and features extraction

The level set method, proposed by Osher and Sethian [40, 41], is an approach for approximating the dynamics of moving curves and surfaces. Chan [10] developed the level set method to detect objects in a given image. We adopt his method to extract regions with a similar color distribution in an image. Note that we choose HSV color space, it is not only close to the people understanding of colors, but also is regarded as the best option in judgment on the color changes. It consists of three components, namely representatives of hue H (hue), saturation S (saturation), and brightness V (value). We introduce the concept of color gradient information of images, instead of using gray gradient to update the curve evolution function of the level set method. Furthermore, these regions representing the facial features of a character are found simultaneously.

After feature extraction, *GrabCut* is then applied to separate foreground (characters including humans) and background. However, it requires an initial trimap constructed by users which represents the seeds of foreground and background in *GrabCut*. We construct a bounding box of all these regions extracted by using the level set method. Then we use the bounding box for *GrabCut* instead of the initial trimap. Note that the extracted regions correspond to the regions of a character matte with the similar color distribution. The pixels inside the contours of the regions are considered the foreground distribution replacing users' refinement in *GrabCut*. Subsequently, the entire energy minimization process would be performed again with the updated foreground distribution. After the iterative process is completed, the character matte is extracted successfully.

Note that we choose HSV color space. Due to the hue, saturation, and brightness of the three components to determine changes in color, the level set method with color gradient enrich the way only use gray gradient to judge whether at the border. Since joining the color factor, the character and feature extraction is robust for the images, which the gray level of the background is close to the gray level of the foreground. The final character and features matte is shown in Fig. 4.

5 Virtual humans generation

After extracting characters in two key-poses or features in moving templates, for virtual humans generation, we train a statistical model by using nonparametric regression and specific training data sets. Note that the model consists of two phases: *kernel regression* with ERBFs trained for shape fitting and LOESS trained for detail fitting. Then the trained model is applied to generate a virtual human by deforming the shape of character to synthesize body movements, a novel view, or expressive face with speech animation. First, we focus on the nonparametric regression model trained from two key-poses for body movements and novel views synthesis. We introduce ERBFs in Section 5.1. Then, in Section 5.2, an initial solution to regression parameters is obtained. We discuss the *kernel regression* model with ERBFs trained to fit the character's deformed shape in Section 5.3. After synthesizing the contours of the character's deformed shape, LOESS is applied to preserve the details or features of characters (that is filling in the color and texture information obtained from the original character in the given image). LOESS is described in Section 5.4. In Section 5.5, the detail is preserved within the deformed shape by using LOESS. Then body movements and novel views are synthesized. Besides, we further focus on the face of the character and create an animated talking or expressive face by using the model trained by moving templates, as described in Section 5.6.

5.1 Elliptic radial basis functions

For body movements and novel views synthesis, the nonparametric regression model with *kernel regression* is trained for the prediction of deformed character shape. Because the initial regions used to predict deformations between two key-poses are achieved using the level set method, the distribution of data values (pixels) in each region is assumed to be normal. RBFs are chosen to fit a smooth surface. However, RBF, which is a circularly shaped basis function, has a limitation in fitting long, high-gradient shapes such as cylindrical shapes. The radius might reach the shortest boundary of the area and might require numerous small RBFs to fit one long shape, which would be matched to the shape of the character such as the body or head of the human. Therefore, we use ERBFs instead of RBFs.

Note that there are two kinds of ERBFs: axis-aligned and arbitrary directional ERBFs. A comparison of these two basis functions is shown in Fig. 5. This figure shows a long diagonal data distribution (pixels along contours) and the influences of the two basis functions are drawn overlaid on the data. The data is approximated by two basis functions: axis aligned ERBF in Fig. 5a and arbitrary directional ERBF in Fig. 5b. Note that the major axis of the ellipse with arbitrary directional ERBFs is aligned along the contour of the character which is a long diagonal data distribution (gray region). For achieving more accurate quality with smaller number of basis functions, we train the *kernel regression* model with arbitrary directional ERBFs.

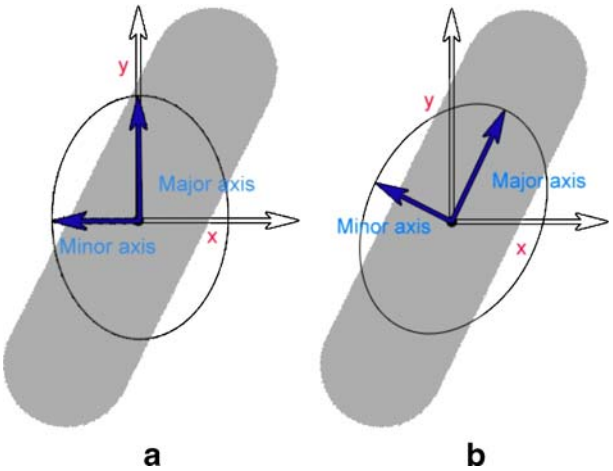

Fig. 5 Comparison of ERBFs. **a** Axis aligned ERBF. **b** Arbitrary directional ERBF. The influence range of each basis function is shown as blue arrows and black curve

Let $\vec{u} = (x, y)$ be a coordinate vector and $\vec{v} = \left(\mu_x, \mu_y\right)$ be a center vector of an elliptic Gaussian. An arbitrary directional ERBF can be represented in a matrix form as follows:

$$k\left(\vec{u}, \vec{v}\right) = \exp\left\{ -\frac{\left(\vec{u}_x - \vec{v}_x\right)^T A_{\theta_x, a_x}\left(\vec{u}_x - \vec{v}_x\right)}{2\sigma_x^2} - \frac{\left(\vec{u}_y - \vec{v}_y\right)^T A_{\theta_y, a_y}\left(\vec{u}_y - \vec{v}_y\right)}{2\sigma_y^2} \right\},$$
$$\vec{u} = \vec{u}_x + \vec{u}_y = [x\ 0]^T + [0\ y]^T,$$
$$\vec{v} = \vec{v}_x + \vec{v}_y = [\mu_x\ 0]^T + \left[0\ \mu_y\right]^T, \tag{1}$$

$$A_{\theta_i, a_i} = \begin{bmatrix} \cos\theta_i / a_i & \sin\theta_i / a_i \\ -a_i \sin\theta_i & a_i \cos\theta_i \end{bmatrix}, \text{for } i \in \{x, y\}, \tag{2}$$

where σ_i^2 for $i \in \{x, y\}$ is the covariance of Gaussian along i-axis. The orientation θ_i (the angle between the major axis of ellipse and i-axis) and the aspect ratio a_i^2 are used to transfer to an arbitrary directional ERBF, as shown in Fig. 6. The transformation matrix A_{θ_i, a_i}, which contains a rotation and scaling component, is applied for alignment along the

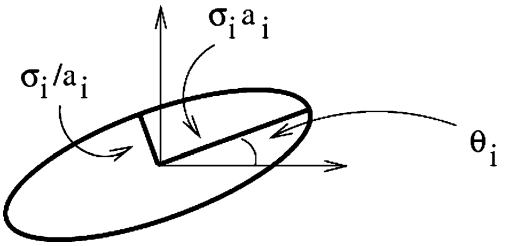

Fig. 6 Schematic diagram of an elliptic Gaussian basis function (arbitrary directional ERBF)

data distribution. In our work, the major axis of ellipse is aligned along the contour of the character, as shown in Fig. 5b. For the mathematical details of (1), it can be inferred from *hyper radial basis functions* [19, 45]. Please refer to Appendix 1.

5.2 The determination of initial values

The initial guesses are important for further optimization convergence in model learning. Before setting the initial value of center and covariance, the correspondence with regard to feature alignment should be done. First we create a window and use it to compute the curvature along each region boundary in the face. Note that these regions in the face are obtained by using the level set method. We choose the top five curvatures from the window interiors and sample points along these contours. The five bounding boxes of these sets of sample points are the feature blocks shown in Fig. 7a and b. The structure of these feature blocks (that is the order of the feature blocks) is constructed to maintain the spatial relationship among these features, as shown in Fig. 7c. Note that the structure is similar to the tree structure. There are no the root node and the leaf node in our work. We only use the link between two nodes (feature blocks) to record the spatial relationship or the order of two nodes. Subsequently, *Tchebichef moments* (TMs) [29] of these blocks are used to determine the correspondence with the spatial constraints of the other key-pose, which is obtained by reversing the original input image, as shown in Fig. 7d and e.

TMs are translation, scale, and rotation invariant functions and useful for image retrieval and pattern recognition. For each feature block, we compute TMs of the block and compare with the other key-pose by using a window with the same size of the block. Since the minimal difference is found, the correspondence can be obtained. Moreover, the hard constraint is used to refine the correspondences found by TMs. According to the spatial relationship of the feature blocks, some correspondences are interchanged. For example, the correspondences of right eye and left eye found by TMs are interchanged. The correspondences based on the structure of the spatial relationship are constructed.

Owing to predicting the contours of deformed character by the nonparametric regression model, we sample the contours and obtain the correspondences shown in Fig. 7f and g as red dots by using TMs. The contour samples, feature blocks mentioned above, and their correspondences are defined as n pairs of anchor points in the space $U = \left(\vec{u}_1, \vec{u}'_1, \ldots, \vec{u}_n, \vec{u}'_n\right)$ for the training stage. K-means clustering is used to set the starting center values to the means of the training anchor points. In addition, the covariance for each k-means group shown in Fig. 7f as the block is computed.

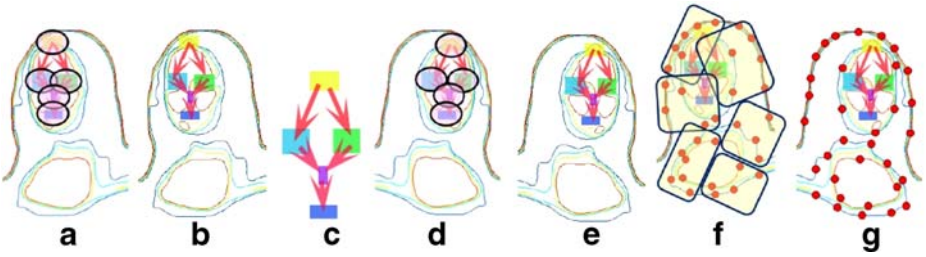

Fig. 7 Correspondences and initial value determination. **a** Top five features are selected. **b** The structure is constructed from feature blocks. **c** The spatial relation is obtained from first key-pose. **d e** The correspondences in the other key-pose are extracted based on the structure of spatial relationship. **f g** The samples and correspondences are shown as red dots, and k-means clustering is employed to determine initial value of regression parameters

5.3 Kernel regression with ERBFs trained for shape fitting

Given n pairs of anchor points and m means of these points we use arbitrary directional ERBFs to predict the contours by interpolating a smooth function. The resulting ERBF interpolating function is defined as a transformation function $F : \Re^2 \rightarrow \Re$. For m pairs of anchor points in input space U, F contains the radial part R and the affine part P as follows:

$$F(\vec{u}) = R(\vec{u}) + P(\vec{u}), \tag{3}$$

$$R(\vec{u}) = \sum_{i=1}^{m} \alpha_i k(\vec{u}, \vec{v}_i), \tag{4}$$

$$P(\vec{u}) = M\vec{u} + \varepsilon, \tag{5}$$

where α_i is the corresponding weight and $F(.)$ is the displacement of either the x-coordinate or the y-coordinate between the correspondences (the given m pairs of anchor points). $P(.)$ is a 2D affine transformation, where M is a 2×2 real matrix and ε is the error term. It can be computed according to the correspondences of anchor points in feature blocks and determined by a least-squares approximation procedure. After the affine component has been computed, the radial component satisfies the following equation:

$$R(\vec{u}) = F(\vec{u}) - P(\vec{u}). \tag{6}$$

The estimated weight $\widehat{\alpha}_i$ is determined by solving the linear system.

$$\begin{aligned} &\widehat{\alpha}_1, \ldots, \widehat{\alpha}_m \\ &= \arg\min_{\alpha_1 \ldots \alpha_m} \sum_{j=1}^{n} \left\| \sum_{i=1}^{m} \alpha_i k\left(\overrightarrow{u}_j, \overrightarrow{v}_i\right) - \left(F\left(\overrightarrow{u}_j\right) - P\left(\overrightarrow{u}_j\right)\right) \right\|^2. \end{aligned} \tag{7}$$

This can be solved by the least-squares normal equations to minimize the sum of the square difference in the matrix form:

$$A = \left(K^T K\right)^{-1} K^T \overline{(F(\vec{u}) - P(\vec{u}))}, \tag{8}$$

where A is the matrix form of the vector α_i, K is the matrix form of the vector $k\left(\overrightarrow{u}, \overrightarrow{v}\right)$, and $\overline{\left(F\left(\overrightarrow{u}\right) - P\left(\overrightarrow{u}\right)\right)}$ is the matrix form of the vector $\left(F\left(\overrightarrow{u}\right) - P\left(\overrightarrow{u}\right)\right)$.

After the weights $(\widehat{\alpha}_1, \ldots, \widehat{\alpha}_m)$ are computed in the initial loop, we can compute the residual for nonlinear optimization. Since residuals are recomputed, the residuals update these parameters in the next iteration, which are centers, covariances, and weights, with a gradient descent. Optimization convergence is achieved when the residual is sufficiently small. The whole process is converged completely soon after in several iterative loops. Then the *kernel regression* model with ERBFs is trained. Note that we can use the model to fit the complete contours of the deformed shape. We can make predictions of the displacement for each contour point using Equation (3). Furthermore, we use *Catmull-Rom splines* to connect new positions of the contour points. For each in-between frame in temporal domain, the contours of the deformed shape are synthesized by the model.

5.4 Locally weighted regression

After synthesizing the contours of the character's deformed shape, a local-fitting methodology called LOESS is applied to preserve the details or features of characters (that is filling in the color and texture information obtained from the original character in the given image). Like *kernel regression*, LOESS [28] is a procedure for fitting a regression surface to data through multivariate smoothing. LOESS uses the data from the neighborhood around a specific location. In other words, LOESS performs a linear regression on points in the data set, weighted by a kernel centered at that pre-defined location. It is much more strongly influenced by the data points that lie close to the location pre-defined according to some scaled Euclidean distance metric. This is achieved by weighting each data point according to its distance to the pre-defined location: a point very close to it is given a weight of one and a point far away is given a weight of zero.

Note that the shape of the kernel is a design parameter for which many possible choices exist. The original LOESS uses the tri-cube weighting function. Nonetheless, we have used the Gaussian kernel to estimate the weights in the range of unit circle, as shown in Fig. 8b. Let x_0 be the specific location (red dot), which would be filled color or texture information. LOESS performs a linear regression on the sampled contour points weighted by a kernel centered at x_0. Given n pairs of points sampled along the contour (purple dots) of the character in the input image and the corresponding new locations of these points, the weight of the i-th sampled contour point x_i with Gaussian kernel is

$$w_i(x_0) = w(x_i - x_0) = \exp\left(-s\|(x_i - x_0)\|^2\right), \tag{9}$$

where $s = 1/2k^2$ and $n = \sum_i w_i(x_0)$ for n data points. s is a smoothing parameter that determines how quickly weights decline in value as one moves away from x_0, k is the kernel width or bandwidth.

5.5 Detail preserving with LOESS

In addition to shape fitting for the whole animation process, the details from the character interiors have to be preserved by filling in the color and texture information obtained from the original character in the given image. To implement detail preserving, we sample the original image with a uniform grid (50×50), as shown in Fig. 8a. Given a grid point x_0, its

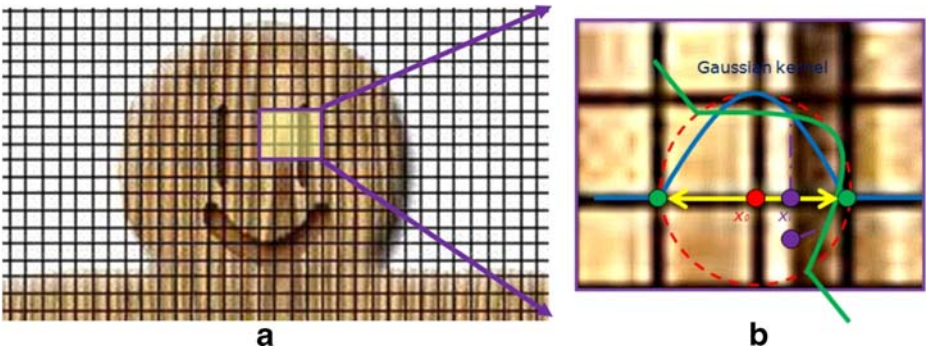

Fig. 8 LOESS analysis. **a** Original image with a uniform grid. **b** The zoom-in view of the image. LOESS with Gaussian kernel is applied to estimate the weights

color or texture information would be controlled by LOESS. Let $x_i = (x_{i,x}, x_{i,y})$ be the i-th point sampled along the contours of the character in the given image, as shown in Fig. 8b. Let $y_i = (y_{i,x}, y_{i,y})$ be the measurements of the dependent variables representing the new position of the sample point x_i in a synthesized frame in temporal domain.

Suppose that the target coordinate $\widehat{y}_i$ is estimated by an estimated local multivariate polynomial as follows:

$$\widehat{y}_i = \beta_1 t_1(x_i) + \beta_2 t_2(x_i) + \ldots + \beta_M t_M(x_i), \tag{10}$$

where $t_j(.)$ is a function that produces the j-th term in the polynomial, and $\beta = (\beta_1, \ldots, \beta_M)$ is a vector of parameters to be estimated. In our transformation model, we have $t_1(x_i) = 1$ for β_1 which is a translation coefficient and $t_2(x_i) = x_i$ for β_2 which is a rotation coefficient. (10) can be written for matrix manipulation, which can be easily extended to datasets with many inputs:

$$\widehat{y}_i = \beta^T t(x_i), \tag{11}$$

where $t(x_i) = (t_1(x_i), t_2(x_i), \ldots, t_M(x_i))$ is the vector of the polynomial terms. Given n pairs of (x_i, y_i), the general way to estimate $\widehat{\beta}$ is by minimizing

$$\widehat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} w_i(x_0)^2 \left(y_i - \beta^T t_i\right)^2, \tag{12}$$

where $t_i = t(x_i)$ and $w_i(.)$ is defined in (9). The minimization can be obtained by the least-squares normal equations. Then target position $\widehat{y}_j$ of the new sample x_j in details within the contours can be approximated from (11) or directly from the closed-form solution as follows. For brevity, we drop the argument x_0 for $w_i(x_0)$ and denote the estimated means and covariances in the following manner:

$$\mu_x = \frac{\sum_i w_i x_i}{n}, \tag{13}$$

$$\sigma_x^2 = \frac{\sum_i w_i (x_i - \mu_x)^2}{n}, \tag{14}$$

$$\sigma_{xy} = \frac{\sum_i w_i (x_i - \mu_x)\left(y_i - \mu_y\right)}{n}, \tag{15}$$

$$\mu_y = \frac{\sum_i w_i y_i}{n}. \tag{16}$$

Then the estimated target coordinate $\widehat{y}_j$ of the new sample x_j can be computed as follows:

$$\widehat{y}_j = \mu_y + \frac{\sigma_{xy}}{\sigma_x^2}\left(x_j - \mu_x\right). \tag{17}$$

Therefore, we can use (17) to find the new location of the grid point x_0 and obtain the pixel values for filling in the color and texture information. In practice, we find the new

location of each vertex in the grid (each grid point). Then we fill the resulting quad using bilinear interpolation. Note that we would reconstruct the details within the contours fitted with ERBFs via a simple closed-form solution. After shape fitting and detail preserving, body movement or a novel view synthesis is carried out. In order to maintain the 3D effect of the new view, it is sometimes combined with backward deformation by using color blending.

5.6 Viseme synthesis with expressive face

For viseme synthesis, viseme segmentation of the speech data is performed to determine all visemes and their durations. First of all, we employ a *Hidden Markov Model* (HMM [32]) speech recognizer, which is the high-precision speech recognition software in noisy environments, to analyze the given speech data. In practice, HMM speech recognizer is used to obtain the phoneme segmentations called phoneme samples, as shown in Fig. 9. Besides, we design fifteen templates in moving templates V for visemes synthesis, which are fourteen common mouth shapes with their relative visemes and a neutral mouth shape for all other visemes, as shown in Fig. 4. The templates in moving templates V are employed to record the positions of anchor points sampled on the contour of the lips. These anchor points are obtained from the extracted features found by using the level set method, as mentioned before. Then we construct a phoneme-viseme mapping table by using a simple table lookup method [51] to find the relative moving template (viseme) of a phoneme sample in moving templates V directly, as shown in Fig. 9. Table 1 shows the conversion from the phoneme to the mouth shape. For two continuous phoneme samples with the same neutral mouth shapes, one of the samples is redefined as the mouth shape with vowel /o/.

In addition, we collect the moving templates D for facial expression simulation, which consist of a neutral expression and several common expressions, such as happy, angry, sad, fear, surprise, and wink, as shown in Fig. 4. The templates in moving templates D are employed to record the positions of anchor points sampled and grouped from the contours of the extracted features, such as facial shape, eyebrows, eyes, nose, and lips, as shown in Fig. 10. Given moving templates D, moving templates V with their relative visemes, and the input phoneme samples, we may create expressive lip-synch animation.

Now, the nonparametric regression model consisting of *kernel regression* with ERBFs and LOESS is trained, as mentioned in Section 5.3 and Section 5.5. In the shape fitting stage, the *kernel regression* model with ERBFs is trained. For viseme synthesis, we would train the model with the training data, that is, n pairs of anchor points recorded in two extreme moving templates which are the neutral mouth shape and the mouth shape with vowel /o/ in V. Then the trained model is applied to fit the variations of mouth shape

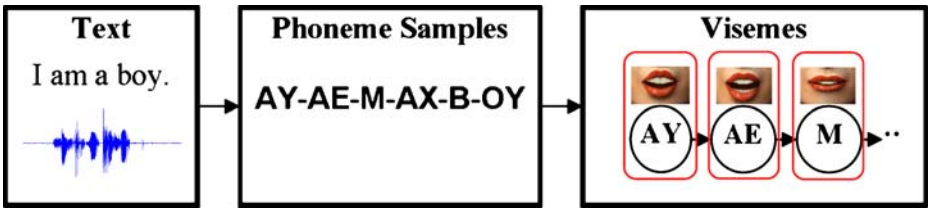

Fig. 9 Viseme segmentation of the given speech data

Table 1 Conversion table from phoneme to mouth shape and the corresponding phonetic alphabet

Mouth shape (Viseme) No.	Phoneme samples	Phonetic alphabet
1	AA, AE, AX, HH, SIL, and Y	/a/, /æ/, /ə/, /h/, /sil/, and /j/
2	B, M, and P	/b/, /m/, and /p/
3	CH, J, and SH	/tʃ/, /dʒ/, and /ʃ/
4	OO, OY, W, and UH	/u/, /ɔɪ/, /w/, and /U/
5	AY, EY, and ER	/aɪ/, /e/, and /ɝ/
6	F and V	/f/ and /v/
7	IH and IY	/I/ and /i/
8	G and K	/g/ and /k/
9	N and NG	/n/, and /ŋ/
10	OH	/o/
11	R	/r/
12	S, TS, and Z	/s/, /ts/, and /z/
13	D, L, and T	/d/, /l/, and /t/
14	TH	/θ/

between arbitrary two visemes of the phoneme samples using the corresponding moving templates in V for lip-synch animation generation. For facial expression simulation, note that we would train the model with a different training data, that is, n pairs of anchor points recorded in two moving templates which are the neutral expression and the specific emotional state in D. Then the trained model is applied to fit the movements of facial features between these two emotional states.

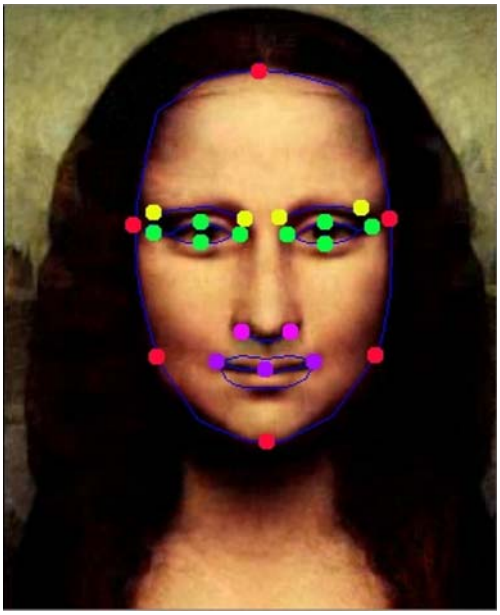

Fig. 10 Groupings of facial shape and features labeled as anchor points and relative curves

Next, we would find the positions of anchor points for facial features in the target expressive face composed of specific emotion and visemes before the model with LOESS is trained for detail preserving. Actually, we would like to identify facial expression simulation independently of the content (utterance of sentences and the corresponding lip movements). The target animated expressive face with lip-synch *FE* can be represented as follows:

$$FE = P(N + F_{nl} + F_l) = P\left(N + \sum_i D_i + \gamma D_l + (1-\gamma)L\right), \tag{18}$$

where $P(.)$ is a 2D rigid transformation of the head movements. The head movements are specified by users. N is a neutral expression. F_l (lips) and F_{nl} (facial features except lips) are the movements of facial features. Note that F_l and F_{nl} are displacements from the neutral expression. So $N + F_{nl} + F_l$ represents an individual facial expression in a certain emotional state and viseme.

Instead of using F_l and F_{nl}, D_l is applied for the detailed movements of lips in a specific emotional state, and D_i for each i represents an individual facial feature except lips. Note that five non-overlapping features are identified for a specific emotional state, such as left eyebrow, right eyebrow, left eye, right eye, and nose. D_l and D_i are obtained through the fitted movements of facial features. Besides, L is lip movement. L is obtained through the fitted variations of mouth shape. For the final mouth shape, a blend weight γ is considered to generate lip synchronization.

After finding the positions of facial features in the target expressive face, the model with LOESS is trained by these facial features. The trained model is employed to preserve details within the target expressive face. For example, after the mouth shape is obtained, the mouth cavity, which is the region between the upper lip and lower lip, is filled in the color and texture information obtained from the original character in the given image by using the trained model. Note that we use the color and texture information inside the mouth of the character to make the virtual human appear realistic while talking. Thus the virtual human with a lively animated talking face is created.

6 Results

In the shape fitting stage, the number of basis functions of all the examples fitting the contours is decided by residual analysis. The default setting is eighteen basis functions with better fitting results. Moreover, we apply only our ERBF model on the top five regions in contours to align the significant features in two key-poses or moving templates instead of the entire character because the prediction of unimportant features leads to redundancy.

The proposed nonparametric model was implemented on an Intel Pentium M 1.5 GHz processor that allows efficient generation of virtual humans. The complete generation process consists of two independent steps: shape fitting and detail preserving. Table 2 lists the resolutions and executions for the figures shown. Execution time was measured in each step. After training the proposed model, it took us only one minute to generate the 20 sec virtual human animation.

The synthesized results of body movements are shown in Figs. 11 and 17. Fig. 11a shows the original image of a cat representing one key-pose. Another key-pose is shown in Fig. 11e. The two key-poses are employed to train the proposed nonparametric regression model consisting of *kernel regression* with ERBFs and LOESS. Then body movements are fitting from the model, as shown in Fig. 11b, c, and d. Besides, Fig. 17a shows one key-

Table 2 Running times for figures

Figure name	Body movements		Novel views synthesis			Viseme synthesis		Facial expression	
	Cat	Object with wood	TinTin (head fitting)	Captain (head fitting)	Snowy (head fitting)	Mona Lisa	Lips	Mona Lisa	Oil painting
Figure no.	11	17	2	12	12	1	15	13	14
Resolution	193× 280	189× 216	240× 502	519× 446	169× 117	182× 268	565× 281	182× 268	505× 582
Shape fitting ~ training	9366	14366	16435	14788	23599	26167	30639	27331	35458
~ fitting (millisecond)	1535	1835	2031	1755	3499	5049	8356	5223	10854
Detail preserving ~ training	6997	8997	8899	8590	9565	8567	10967	8499	12199
~ fitting (millisecond)	856	1065	1101	999	1890	2432	4288	2511	4805

pose of a human-like object. Another key-pose is the reverse of Fig. 17a. Body movements of the object are synthesized and shown in Fig. 17d. Note that the pattern of fur in Fig. 11 and the pattern of wood grain in Fig. 17 are preserved with LOESS.

Our experiments were also performed on digitized images obtained from "The Adventures of TinTin: The Shooting Star" which was originally produced by Georges Remi (Hergé). The results are presented in Figs. 2 and 12. They show different frames in the original comic, several synthesized frames of character's motion, and the zoom-in views. They are only head movement. A user specification exists by which the head and body can be segmented. For fitting the contours of the head component, the second key-pose involves reversing the contours of the head component and concatenating with the other contours. A novel view would then be synthesized using the trained model. Another example of Mona Lisa is shown in Fig. 1.

Besides, we are interested in extending our concept to facial expression and viseme synthesis. With the exception of body movements and a novel view interpolation, several facial effects observed in virtual humans, such as eye, nose, and mouth movements, could be created, as shown in Figs. 1, 13, 14, and 15. By moving the facial features obtained from the structure of spatial relationship, which we constructed before, we simulate the dynamics

Fig. 11 Body movements synthesis. **a e** Two key-poses of the cat. **b c d** The synthesized results

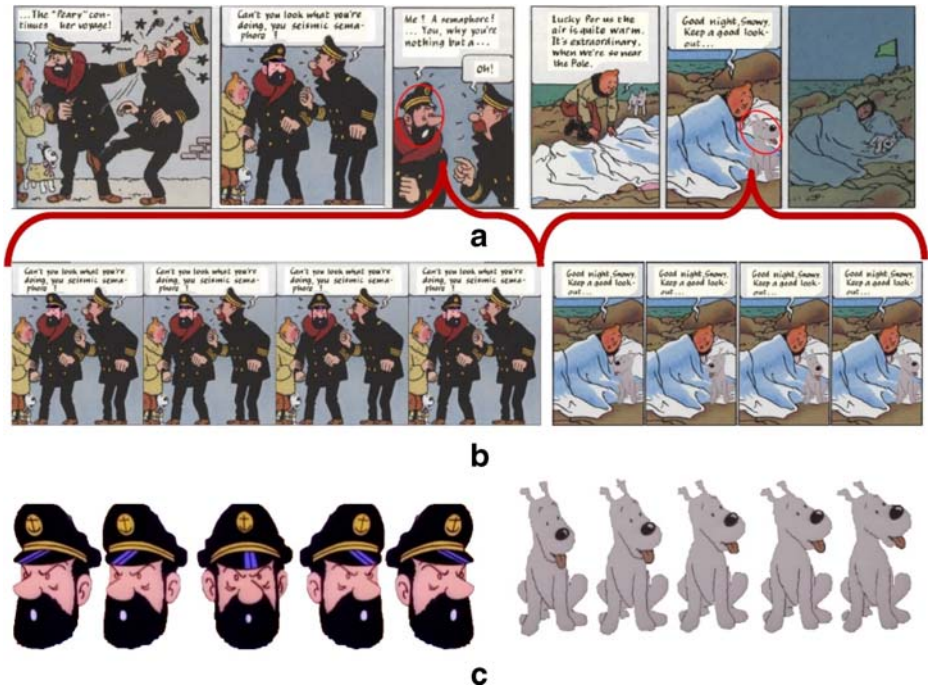

Fig. 12 Novel views synthesis in a comic. **a** The frames in the comic. **b** The frames synthesized from a single frame. **c** The zoom-in views of the results. © Georges Remi (Hergé)

of the features to synthesize different expressions, such as blink, anger, or happy. We could enhance the expression by shaking the shoulders or wagging the human's head. Figure 13a is the original picture of Mona Lisa. Figure 13b, c, d, and e are the synthesized facial expressions. Note that the proposed model is trained from moving templates of an emotional state (smiley). Furthermore, we use the model to predict other emotional states of another character in Vincent van Gogh' Self-Portrait and as shown in Fig. 14. For viseme synthesis, Fig. 1c shows several frames in the synthesized speech animation of Mona Lisa. Another lip-synch example is shown in Fig. 15 for five vowels.

Since our goal is to do visually plausible character animation for virtual humans generation, we focus on the qualitative analysis. We provide the results obtained by using

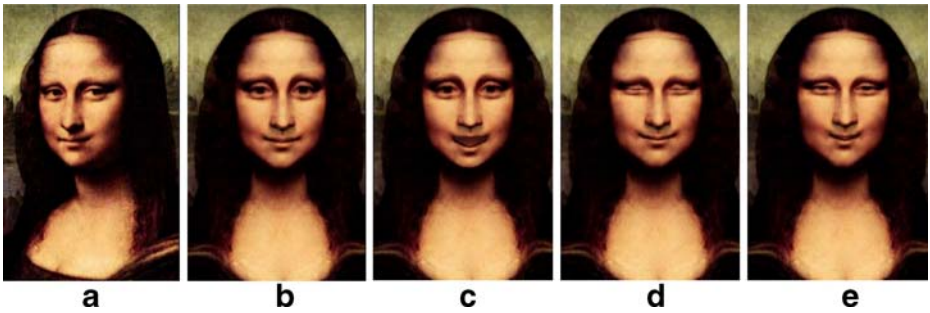

Fig. 13 Virtual human with expression synthesis. **a** The picture of Mona Lisa. **b c d e** The synthesized expressions

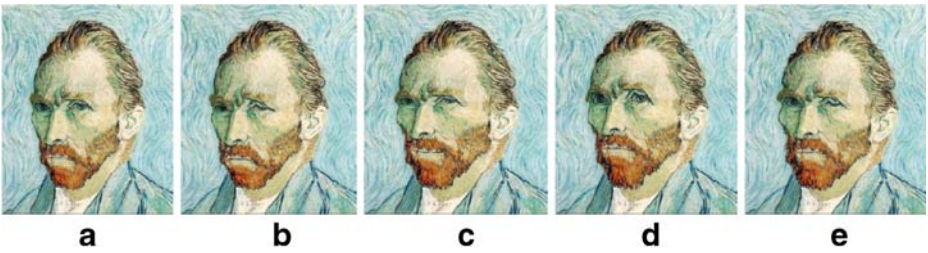

Fig. 14 Virtual human with expression synthesis. **a** The original expression in Vincent van Gogh's Self-Portrait. **b** Sad. **c** Smiley. **d** Staring. **e** Winking

kernel regression with RBFs, view morphing proposed by Seitz and Dyer [39], image deformation using the moving least squares method proposed by Schaefer et al. [37], and animation of still images using the path-based method proposed by Mahajan et al. [26], that suffices for a direct visual comparison. Figure 16 shows a comparison of novel views synthesis. Figure 16b is obtained by using *kernel regression* with RBFs, and Fig. 16c is obtained by our model with ERBFs respectively. Figure 16d is our results combined with the original background. Note that the number of basis functions is the same. Since shape fitting with RBFs contains more unnatural distortions in forward deformation, ghost effects are observed in the final result with color blending even though feature alignment is achieved in contours fitting. The quality of the final blending result with ERBFs is better.

As mentioned before, previous techniques such as view interpolation and shape deformation may be able to produce good quality results for body movements and novel views synthesis. However, both techniques needed user intervention. Figures 16e and 17 provide the comparisons with the view morphing technique proposed by Seitz and Dyer [39], and the image deformation using the moving least squares method proposed by Schaefer et al. [37]. In view morphing, it is necessary to compute an additional estimated fundamental matrix for camera calibration. Further, many users' specifications are required for correspondences. Figure 16e shows that lacked users' specification would create ghost effects because of nonalignment. There were seventeen control lines on the face specified by users. A better result was obtained when more than thirty or forty control lines were specified. Besides, the method of Schaefer et al. preserved the details of characters, such as wood grain. This property may lead to an undesired result and unnatural distortions when users specify the moving handles which exceed the control extent because of the constraint using moving least squares, as shown in Fig. 17. This man-made situation or interference would not occur in the proposed model. Our model would be automatic in shape deformation process of body movements synthesis.

Moreover, Fig. 18 shows a comparison with the path-based method proposed by Mahajan et al. [26]. Their method is based on an inverse optical flow and preserves the

Fig. 15 Viseme synthesis for five vowels. **a** The original mouth shape. **b** /a/. **c** /e/. **d** /i/. **e** /o/. **f** /u/

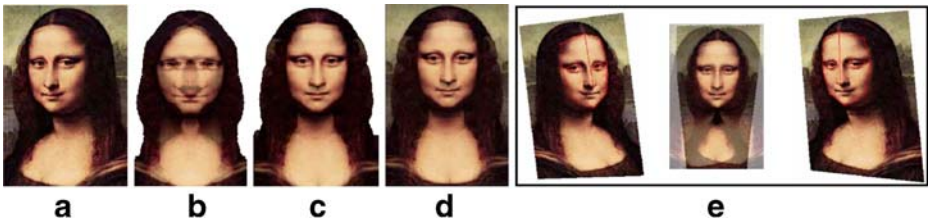

Fig. 16 Comparison of a novel view obtained by using *kernel regression* with RBFs, *kernel regression* with ERBFs, and view morphing proposed by Seitz and Dyer [39]. **a** The picture of Mona Lisa. **b** The result obtained by using *kernel regression* with RBFs. **c d** The result obtained by using *kernel regression* with ERBFs (our model). **e** Ghost occurrence in view morphing without enough correspondences (red lines are specified by users)

spatial frequencies of the input images. However, as mentioned before, the disparity or the motion that they can handle between the images is limited. In [26], the disparity or motion between the images is about 30 pixels. The novel views shown in Fig. 18d and f are synthesized by using our method from Fig. 18a and the reverse of Fig. 18a. Note that the maximum disparity or motion between two images is 70 pixels. A few shades on the face are due to the fact that the inconsistent illumination or brightness on the face in the input image and the aforementioned color blending we used to maintain the 3D effect of the synthesized view. Besides, a few blurring artifacts near the whiskers and tongue are observed in another example of the yawning cat synthesized by their method, as shown in Fig. 18k. Note that the details are preserved explicitly by using our method, such as fur, whiskers, tongue, and wood shown in Fig. 18i and j.

In general, we find that our method provides visually superior shape fitting or details preserving with minimal artifacts in most cases. On the other hand, our method does not suffer from serious ghosting, blurring artifact, or unnatural warping, which exists in other methods. Moreover, our proposed shape fitting and detail preserving method does not require user-specified correspondences. Considering view morphing and image deformation using the moving least squares method, they require users' intervention for correspondences or handling the deformation.

In addition, there are many possible applications of our proposed nonparametric regression model consisting of *kernel regression* with ERBFs and LOESS. For example, we

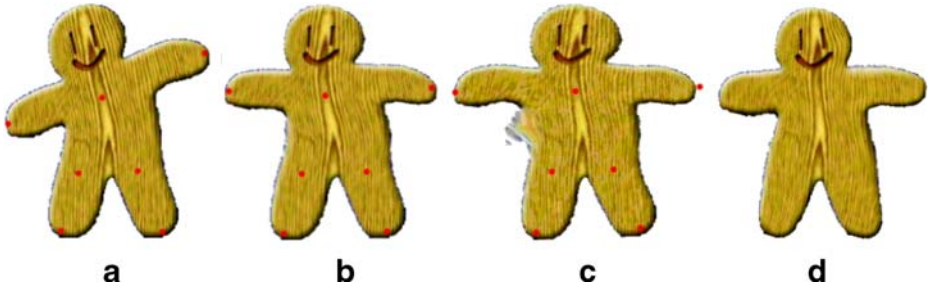

Fig. 17 Comparison with image deformation using the moving least squares method proposed by Schaefer et al. [37]. **a** The character with handles (red dots). **b** The results created by using moving least squares with distortions. **c** The undesired warp occurrence (moving handles exceeds the control extent). **d** The same pose with **b** created by using our method

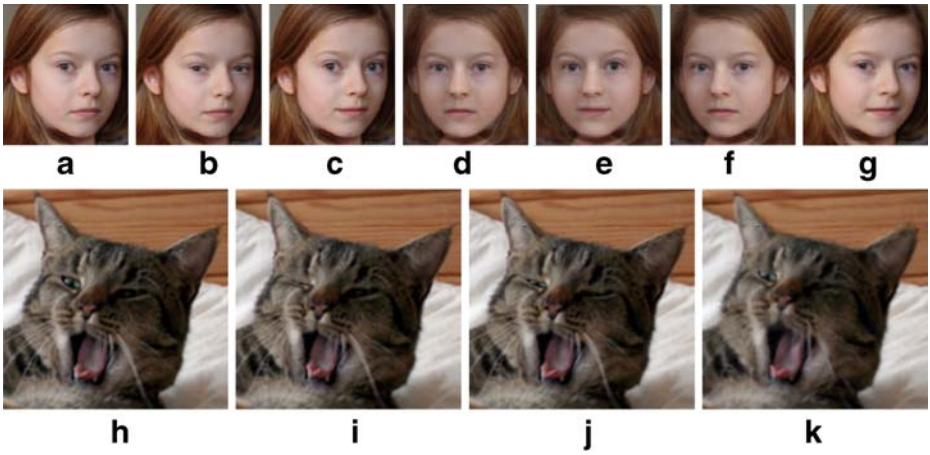

Fig. 18 Comparison with animation of still images using the path-based method proposed by Mahajan et al. [26]. **a h** The input images obtained by [26]. **b c** The face emotions synthesized by using our method (staring and smiley). **d e** Different face emotions in a novel view synthesized by using our method (neutral and smiley). **f** Another novel view synthesized by using our method. **g k** The results obtained by [26] (smiley and yawning). **i j** Two frames of a yawning cat synthesized by using our method

may adopt our model to learn realistic passive elements movements for synthesizing a rippling pond in the scene, which is subject to natural forces like wind. Thus, we could reanimate arbitrary still images without limiting our domain, including both active virtual humans and passive elements for advanced intelligent multimedia applications.

7 Conclusion

In this paper, we have proposed a novel nonparametric regression model by using *kernel regression* with ERBFs and LOESS, which allows virtual humans generation. The virtual humans, who are constructed by body movements, novel views synthesis, and expressive facial animation with lips movements from speech, are generated and animated smoothly while minimizing unnatural distortion. Just like an agent system, the behaviors that are commonly observed for humans would be simulated and forecasted by using the trained model. Furthermore, we have shown visual results for the purpose of comparison. The results also reveal that the generated virtual humans are suitable to be the spokesman or substitute in the NG applications domain. Moreover, generating animated faces of virtual humans from speech has immediate applications to the games and movie industries.

However, the prediction performance of our model is considerably limited by the structure of the input image. The proposed model may fail in case of overlapping regions such as an arm overlapping the body. Each region may be applied to deform separately with users' interaction. In future, we intend to improve the performance and quality of the scattered ERBFs and LOESS fitting algorithm and synthesize the smooth transition between two motions. Furthermore, we can predict the time series model of a moving virtual human with the nonparametric model using *kernel regression* with ERBFs and LOESS. The time series model would be applied to retarget the motion onto any similar humans or human-like characters.

Acknowledgements This work is supported partially by the National Science Council, Republic of China, under grant NSC 98-2221-E-009-123-MY3. We would like to thank Prof. Sang-Soo Yeo and reviewers for their helpful suggestions.

Appendix 1 Hyper radial basis functions (HRBFs)

HRBF is computed by using the *Mahalanobis distance*, which is defined in the matrix form as follows:

$$k(\vec{u},\vec{v}) = \exp\Big(-(\vec{u}-\vec{v})^T \sum (\vec{u}-\vec{v})\Big), \quad \text{for } \Sigma = diag\big(\sigma_1^{-2},\ldots,\sigma_N^{-2}\big) \text{ and } \sigma_1,\ldots,\sigma_N \in \Re^+, \tag{19}$$

where σ_N^2 should be the covariance of the multidimensional Gaussians rather than the single variance. HRBF differs from a standard RBF insofar each axis of the input space $\chi \subseteq \ell_2^N$ (the space of square summable sequences of length N) has a separate smoothing parameter, i.e., a separate scale onto which the differences on this axis are viewed. It is worth mentioning that RBF kernels map the input space onto the surface of an infinite dimensional hyperspace. Note that N=2 in arbitrary directional ERBF kernel represents the analysis of data distribution along the major axis and the minor axis in an ellipse. Along the orientation of arbitrary directional ERBF (the major axis and the minor axis), (1) is constructed.

References

1. Alexa M, Cohen-Or D, Levin D (2000) As-rigid-as-possible shape interpolation. In SIGGRAPH '00 157–164
2. Arad N, Dyn N, Reisfeld D, Yeshurun Y (1994) Image warping by radial basis functions: applications to facial expressions. CVGIP Graph Models Image Process 56(2):161–172
3. Baker S, Scharstein D, Lewis JP, Roth S, Black MJ, Szeliski R (2007) A database and evaluation methodology for optical flow. In IEEE International Conference on Computer Vision 1–8
4. Blanz V, Basso C, Poggio T, Vetter T (2003) Reanimating faces in images and video. Comput Graph Forum 22(3):641–650
5. Botsch M, Sorkine O (2008) On linear variational surface deformation methods. IEEE Trans Vis Comput Graph 14(1):213–230
6. Brand M (1999) Voice puppetry. In SIGGRAPH '99 21–28
7. Bruce HT, Calder P (1995) Animating direct manipulation interfaces. In the 8th ACM Symposium on User Interface Software and Technology 3–12
8. Busso C, Deng Z, Grimm M, Neumann U, Narayanan SS (2007) Rigid head motion in expressive speech animation: analysis and synthesis. IEEE Trans Audio Speech Lang Process 15(8):1075–1086
9. Busso C, Narayanan SS (2007) Interrelation between speech and facial gestures in emotional utterances: a single subject study. IEEE Trans Audio Speech Lang Process 15(8):2331–2347
10. Chan TF (2001) Active contours without edges. IEEE Trans Image Process 10(2):266–277
11. Chen SE, William L (1993) View interpolation for image synthesis. In SIGGRAPH '93 279–288
12. Chuang Y-Y, Goldman DB, Zheng KC, Curless B, Salesin D, Szeliski R (2005) Animating pictures with stochastic motion textures. ACM Trans Graph 24(3):853–860
13. Deng Z, Neumann U (2006) efase: expressive facial animation synthesis and editing with phoneme-isomap controls. In SIGGRAPH/Eurographics Symposium on Computer Animation 251–260
14. Ezzat TF, Geiger G, Poggio T (2002) Trainable video realistic speech animation. ACM Trans Graph 21 (3):388–398
15. Forstmann S, Ohya J, Krohn-Grimberghe A, McDougall R (2007) Deformation styles for spline-based skeletal animation. In SIGGRAPH/Eurographics Symposium on Computer Animation 141–150
16. Fu T, Foroosh H (2004) Expression morphing from distant viewpoints. In International Conference on Image Processing 3519–3522
17. Glocker B, Paragios N, Komodakis K, Tziritas G, Navab N (2008) Optical flow estimation with uncertainties through dynamic MRFs. In IEEE Conference on Computer Vision and Pattern Recognition
18. Goldstein E, Gotsman C (1995) Polygon morphing using a multiresolution representation. In Graphics Interface '95 247–254

19. Herbrich R (2002) Learning kernel classifiers theory and algorithms. The MIT Press
20. Hornung A, Dekkers E, Kobbelt L (2007) Character animation from 2D pictures and 3D motion data. ACM Transaction on Graphics 26(1) Article No. 1
21. Igarashi T, Moscovich T, Hughes JF (2005) As-rigid-as-possible shape manipulation. ACM Trans Graph 24(3):1134–1141
22. Jang Y, Botchen RP, Lauser A, Ebert DS, Gaither KP, Ertl T (2006) Enhancing the interactive visualization of procedurally encoded multifield data with ellipsoidal basis functions. Comput Graph Forum 25(3):587–596
23. Lempitsky L, Roth S, Rother C (2008) FusionFlow: discrete-continuous optimization for optical flow estimation. In IEEE Conference on Computer Vision and Pattern Recognition
24. Li Y, Huttenlocher D (2008) Learning for optical flow using stochastic optimization. In the 10th European Conference on Computer Vision 2:379–391
25. Litwinowicz P, Williams L (1994) Animating images with drawings. In SIGGRAPH '94 409–412
26. Mahajan D, Huang F-C, Matusik W, Ramamoorthi R, Belhumeur P (2009) Moving gradients: a path-based method for plausible image interpolation. ACM Transaction on Graphics 28(3) Article No. 42
27. McGurk H, MacDonald J (1976) Hearing lips and seeing voices. Nature 746–748
28. Montgomery DC, Peck EA, Vining GG (2006) Introduction to linear regression analysis. Wiley
29. Mukundan R, Ong SH, Lee PA (2001) Image analysis by tchebichef moments. IEEE Trans Image Process 10(9):1357–1364
30. Ngo T, Cutrell D, Dan J, Donald B, Loeb L, Zhu S (2000) Accessible animation and customizable graphics via simplicial configuration modeling. In SIGGRAPH '00 403–410
31. Park J, Sandberg WI (1993) Nonlinear approximations using elliptic basis function networks. In 32nd Conference on Decision and Control 3700–3705
32. Rabiner LR (1990) A tutorial on hidden markov models and selected applications in speech recognition. Readings in speech recognition 267–296
33. Ranjan V, Fournier A (1996) Matching and interpolation of shapes using unions of circles. Comput Graph Forum 15(3):129–142
34. Ren X (2008) Local grouping for optical flow. In IEEE Conference on Computer Vision and Pattern Recognition
35. Rother C, Kolmogorov V, Blake A (2004) "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Trans Graph 23(3):309–314
36. Ruprecht D, Müller H (1995) Image warping with scattered data interpolation. IEEE Comput Graph Appl 15(2):37–43
37. Schaefer S, Mcphail T, Warren J (2006) Image deformation using moving least squares. ACM Trans Graph 25(3):533–540
38. Sederberg T, Greenwood E (1992) A physically based approach to 2D shape blending. In SIGGRAPH '92 25–34
39. Seitz SM, Dyer CR (1996) View morphing. In SIGGRAPH '96 21–30
40. Sethian JA (1996) Level set methods. Cambridge University Press
41. Sethian JA (1999) Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge University Press
42. Sun D, Roth S, Lewis JP, Black MJ (2008) Learning optical flow. In the 10th European Conference on Computer Vision 3:83–97
43. Trobin W, Pock T, Cremers D, Bischof H (2008) Continuous energy minimization via repeated binary fusion. In the 10th European Conference on Computer Vision 4:677–690
44. Vedula S, Baker S, Kanade T (2005) Image-based spatio-temporal modeling and view interpolation of dynamic events. ACM Trans Graph 24(2):240–261
45. Vorobyov SA, Cichocki A (2001) Hyper radial basis function neural networks for interference cancellation with nonlinear processing of reference signal. Digit Signal Process 11(3):204–221
46. Wang Y, Xu K, Xiong Y, Cheng Z-Q (2008) 2D shape deformation based on rigid square matching. Computer Animation and Virtual Worlds 19(3–4):411–420
47. Weber O, Ben-Chen M, Gotsman C (2009) Complex barycentric coordinates with applications to planar shape deformation. Comput Graph Forum 28(2):587–397
48. Wolberg G (1998) Image morphing: a survey. Vis Comput 14(8):360–372
49. Xu L, Chen J, Jia J (2008) Segmentation based variational model for accurate optical flow estimation. In the 10th European Conference on Computer Vision 1:671–684
50. Yan H-B, Hu S-M, Martin RR, Yang Y-L (2008) Shape deformation using a skeleton to drive simplex transformations. IEEE Trans Vis Comput Graph 14(3):693–706
51. Yotsukura T, Morishima S, Nakamura S (2003) Model-based talking face synthesis for anthropomorphic spoken dialog agent system. In the 11th ACM International Conference on Multimedia 351–354

Yun-Feng Chou is a Ph.D. candidate in the Department of Computer Science at National Chiao Tung University (NCTU), Hsinchu, Taiwan, Republic of China. He is a member of the Computer Graphics Laboratory at NCTU. He received his B.S. and M.S. in Computer Science from Soochow University in 2003 and 2005. His research interest includes computer graphics, image processing, and content-based multimedia indexing and retrieval algorithm.

Zen-Chung Shih is a professor in the Department of Computer Science at National Chiao Tung University (NCTU), Hsinchu, Taiwan, Republic of China. He received his Ph.D. in Computer Science from National Tsing Hua University. His research interest includes computer graphics, virtual reality, and scientific visualization.