Animating Characters Using Nonparametric Regression

Yun-Feng Chou
Department of Computer Science,
National Chiao Tung University, Taiwan
yfchou@cs.nctu.edu.tw

Zen-Chung Shih
Department of Computer Science,
National Chiao Tung University, Taiwan
zcshih@cs.nctu.edu.tw

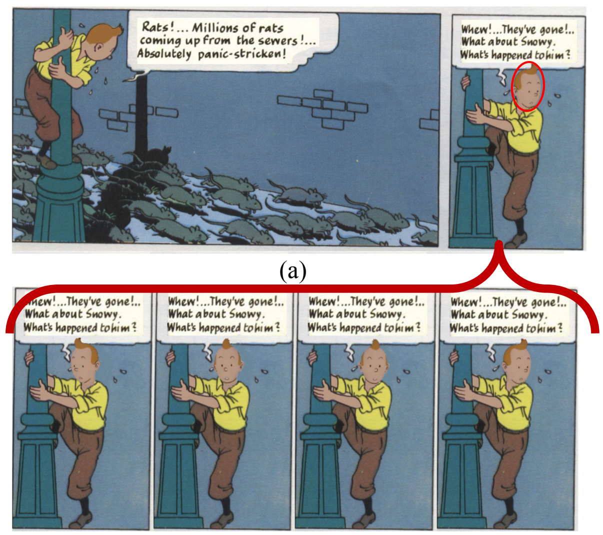
Abstract

In this paper, we propose an image deformation technique using nonparametric regression to animate characters in a still image for multimedia applications. The proposed approach effectively produces a sequence of contiguous frames in an animation. It automatically generates deformed shapes by using elliptic radial basis functions (ERBFs) and locally weighted regression (LOESS). ERBFs are used for representing the deformed character's shapes in synthesized frames. For preserving the pattern within a shape, LOESS is applied to fit the detail with local control. Furthermore, the results show the synthesized frames without unnatural distortion.

1. Introduction

In multimedia applications, deforming characters in a 2D image has received lots of interests. Based on reanimating a still picture, it has become solvable. For example, Chuang et al. [5] deformed pictures using stochastic motion textures. They animated passive elements which are subject to natural forces like wind. Hornung et al. [8] achieved the motion of photographed persons by projecting them to 3D motion data. By contrast, we would take the idea of creating deformations directly in image space one step further by making characters move. We propose a novel application based on image deformation to animate characters, such as shape deformation and different view generation of the character in an image. Then we extend our concept to create virtual humans.

Animating the character in a comic could be carried out by the creation of a new view, as shown in Figure 1. It shows two consecutive frames in the original comic that can be regarded as two different scenes and the synthesized frames from a single input frame. This paper involves a novel technique for building a nonparametric regression model for character



(b)

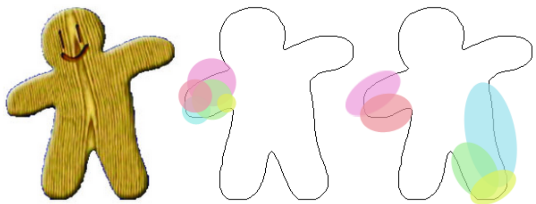
Figure 1. Character deformation in a comic. (a) Two consecutive frames in the comic. (b) The synthesized frames from a single input frame.

deformation, which is used to fit the shape and detail of the character between two key-poses (the input and its reverse) without unnatural distortion.

Our proposed approach is based on the prediction abilities of both kernel regression and *locally weighted regression* [9, 13]. Kernel regression approximates the contours of the deformed character between two key-poses by the prior use of a set of kernel functions. Previously, researchers [26] presented image morphing techniques using *radial basis functions* (RBFs) with spatially-limited circular Gaussian distribution functions for the kernel.

In contrast, circular Gaussian is not an appropriate choice to fit contours, which have noncircular structures, as shown in Figure 2. Figure 2 (a) is the original image, (b) using the circular Gaussians needs five kernels to fit the contour of the right arm of the character, and (c) using the arbitrary directional elliptic Gaussians can fit the right arm and left leg with the

978-0-7695-3428-2/08 $25.00 © 2008 IEEE
DOI 10.1109/CSA.2008.45



(a) (b) (c)

Figure 2. Comparison of the number of basis functions using Gaussians. (a) The original image. (b) Using RBFs to fit the contour of right arm with five kernels, and (c) using ERBFs to fit right arm and left leg with the same number of kernels.

same number of kernels. Using too many circular Gaussians increases the learning and fitting time. In this paper, we develop character deformation using *elliptic radial basis functions* (ERBFs), specifically elliptic Gaussians, which provide less fitting time. Although ERBFs require more computation during optimization, better quality is obtained with fewer number of basis functions.

Except the globally smooth shape deformation with contour fitting mentioned above, the local-fitting methodology is also applied to preserve important features within the contour. For example, the wood grain of the character in Figure 2 (a). *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 investigation makes the following contributions:

- A novel approach for deformed shape fitting based on 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 without unnatural distortion.

2. Related work

Various techniques have been applied to animate characters in image morphing, view interpolation, and shape deformation.

Image morphing. Several studies [7, 16, 22, 26] referred to as image morphing have been conducted. For example, RBF is suitable for fitting smooth functions and is used to warp facial expressions and animate images or drawings [2, 12, 18]. 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. Nonlinear approximation of functions in general spaces with ERBF networks (referred to as *elliptic basis function* networks [15]) was proposed. Furthermore, a volumetric approximation system was developed with ellipsoidal Gaussian functions for 3D volumes (referred to as *ellipsoidal basis functions* [11]).

View interpolation. Besides, several approaches for view interpolation can be applied to character deformation [4, 6, 19, 24]. Seitz and Dyer [19] 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, 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. [10] triangulated the input image and minimized the distortion of these triangles in the deformation process by solving a linear system of equations. Furthermore, Schaefer et al. [23] 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. Unnatural distortions would be generated when the range of controlling handles were exceeded because the locally influencing extent using moving least squares is limited.

3. Algorithm overview

Our proposed method consists of two stages: character extraction and morphing between two key-poses, as shown in Figure 3. The outline below reflects further structure of this paper:

a. Character 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 in Section 4. Similar regions are extracted by the level set method, as shown in Figure 3 (c). The bounding box of all

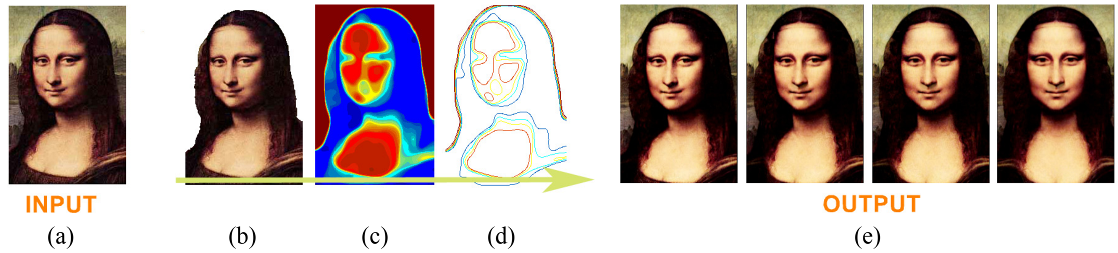


Figure 3. This example shows the picture of Mona Lisa. (a) The original input image. (b) The character is extracted, (c) who is described by the similar parts found by level-set-based GrabCut, and (d) the contours are applied to build the nonparametric regression model for contour fitting and detail preserving. After contour fitting and detail preserving, (e) shows several resulting frames in the synthesized Mona Lisa's deformation.

regions is then used by GrabCut [17]. The boundaries of regions corresponding to the matte produced automatically are further applied to obtain the final character matte, as shown in Figure 3 (d). The foreground and background are separated successfully.

b. Morphing between two key-poses using nonparametric regression. We commence the character deformation process consisting two steps: the shape fitting and the detail fitting step. In the shape fitting step, the correspondences between two key-poses is constructed first. We use kernel regression with ERBFs to fit the contour of deformed character in Section 5.3. In the detail preserving step, we fit the details of the character by LOESS in Section 5.5. Note that it is suitable for detail preserving in accordance with the previously fitted contour.

4. Character extraction

The level set method, proposed by Osher and Sethian [20, 21], is an approach for approximating the dynamics of moving curves and surfaces. Chan [3] developed the active contours with the level set method to detect objects in a given image for image segmentation. We adopt his method using the curve evolution based on the *Mumford-Shah function* to segment regions with a similar color distribution, as shown in Figure 3 (c).

After image segmentation, the regions containing similar color distribution are obtained. GrabCut is then applied to segment foreground and background. However, it requires an initial trimap constructed by users which represents the seeds of the foreground and background in GrabCut. We construct a bounding box of all these regions. Instead of the initial trimap, we use the bounding box for GrabCut, which would proceed without user intervention. Note that the contours of the regions corresponding to the regions of the character matte with similar color distribution are extracted. The contours constrain the pixels such that they form either the foreground or the background replacing users' refinement in GrabCut. Subsequently, the entire iterative energy minimization process would be performed again with the updated foreground and background distribution. The final character matte is shown in Figure 3 (b).

5. Two key-poses morphing

After extracting characters in the original image and its reverse, we obtain two key-poses of the character. To animate the character, we build a statistical model by using nonparametric regression. The statistical model consists of two phases: kernel regression with ERBFs for shape fitting and LOESS for detail fitting. First, we describe ERBFs in Section 5.1. Then, in Section 5.2, an initial solution to regression parameters is obtained. Next, we discuss the fitting of morphed shape of the character with ERBFs in Section 5.3. LOESS is introduced in Section 5.4. Finally, in Section 5.5, the detail is preserved in the morphed shape by using LOESS.

5.1. Elliptic radial basis functions

We construct a kernel regression model for the prediction of deformed character contours. Because 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 chose 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 many small RBFs to fit one long shape, which would be matched to the shape of a character such as the body or head of a human. In order to obtain better quality with fewer number of basis functions, we use arbitrary directional ERBFs instead of RBFs. Let $\vec{u} = (x, y)$ be a coordinate vector and $\vec{v} = (\mu_x, \mu_y)$ be a center vector of an elliptic Gaussian. An arbitrary

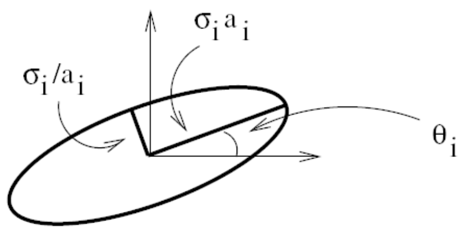


Figure 4. Schematic diagram of an elliptic Gaussian basis function (arbitrary directional ERBF).

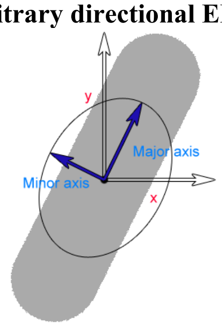
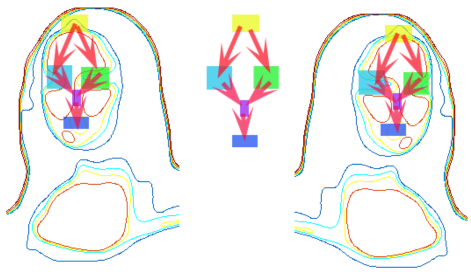


Figure 5. The major axis of the ellipse with arbitrary directional elliptic radial basis functions is aligned along the contour of the character which is a long diagonal data distribution (gray region).



(a) (b) (c)

Figure 6. Correspondences based on the structure of spatial relationship. (a) The spatial relation in the first key-pose. (b) The structure constructed from the first one. (c) The correspondences in the other key-pose using the structure of the spatial relation.

directional ERBF can be represented in a matrix form.

$$k(\vec{u},\vec{v}) = \exp\left\{-\frac{(\vec{u}_x-\vec{v}_x)^T A_{\theta_x,a_x}(\vec{u}_x-\vec{v}_x)}{2\sigma_x^2} - \frac{(\vec{u}_y-\vec{v}_y)^T A_{\theta_y,a_y}(\vec{u}_y-\vec{v}_y)}{2\sigma_y^2}\right\}, \quad (1)$$

$$\vec{u} = \vec{u}_x + \vec{u}_y = [x \quad 0]^T + [0 \quad y]^T,$$

$$\vec{v} = \vec{v}_x + \vec{v}_y = [\mu_x \quad 0]^T + [0 \quad \mu_y]^T,$$

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

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 Figure 4. The transformation matrix A_{θ_i,a_i}, which contains a rotation and scaling component, is applied for alignment along the data distribution. In our work, the major axis of ellipse is aligned along the contour of the character, as shown in Figure 5. For the mathematical details of Equation (1), it can be inferred from *hyper radial basis functions* [9, 25].

5.2. The determination of initial values

The initial guesses are important for further optimization convergence. Before setting the initial value of center and covariance, the correspondences with regard to feature alignment should be done. We choose the top five feature blocks, which are sets of sample points along the contours for each region obtained from the level set method. The criterion is defined as the curvature of the region boundary and the dissimilarity to neighbors. The structure of these five feature blocks is constructed to maintain the spatial relationship among these features, as shown in Figure 6 (a). Subsequently, *Tchebichef moments* (TM) [14] of these blocks are used to determine the correspondences with the spatial constraints of the other key-pose, which is obtained by reversing the original input image. The correspondences based on the structure of the spatial relation are shown in Figure 6.

5.3. Contour fitting with ERBFs

Given n pairs of anchor 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 $F:\Re^2 \to \Re$. For m pairs of anchor points in input space U, F contains the radial part R and the affine part P as follows:

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

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

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

where α_i is the corresponding weight and $F(.)$ is the displacement of either the x-coordinate or the y-coordinate between the correspondences. $P(.)$ 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 $\hat{\alpha}_i$ is determined by solving the following linear system.

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

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

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

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

After the weights ($\hat{\alpha}_1,...,\hat{\alpha}_m$) 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. We can use the model to fit the complete contours of the deformed character.

5.4. Locally weighted regression

Like kernel regression, LOESS [13] 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. Let x_0 be the specific location of interest. x_i represents the location of data points, where $1 \le i \le n$. The weight of data point x_i with Gaussian weight function is

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

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

To implement detail preserving, we sample the original image with a uniform grid (50×50). The vertices in the grid, which belong to the character's contours, would be the specific locations (x_0) controlled by LOESS. Let $x_i = \left(x_{i,1},...,x_{i,p}\right)$ for $i = 1,...,n$ be n measurements of p independent variables. In our work, x_i is the *i-th* sample point along the contours, which is at the coordinate $\left(x_{i,x}, x_{i,y}\right)$ around the specific point x_0 determined by the nearest vertex in the grid. Let $y_i = \left(y_{i,x}, y_{i,y}\right)$ be the measurements of the dependent variables representing the new position of x_i in the contours of the deformed character.

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

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

around the specific point x_0, where $t_j(.)$ is a function that produces the *j-th* term in the polynomial, and $\beta = \left(\beta_1,...,\beta_M\right)$ 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. Equation (10) can be written for matrix manipulation, which can be easily extended to datasets with many inputs:

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

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

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

where $t_i = t(x_i)$ and $w_i(.)$ is defined in Equation (9). The minimization can be obtained by the least-squares normal equations. Then a target coordinate $\hat{y}_j$ of a new sample x_j in details within the contours can be approximated from Equation (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:

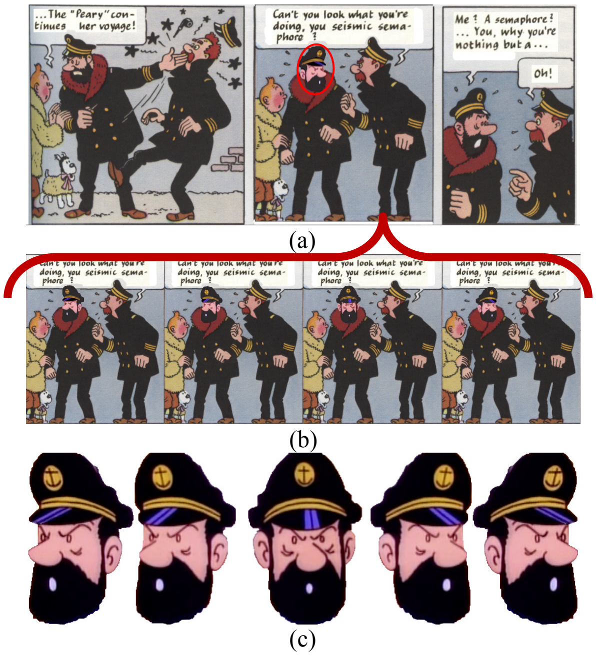


Figure 7. Character deformation in a comic with ERBFs and LOESS. (a) The frames of the comic. (b) The synthesized frames from a single input frame. (c) The zoom-in views of the results. © Georges Remi (Hergé) / Moulinsart

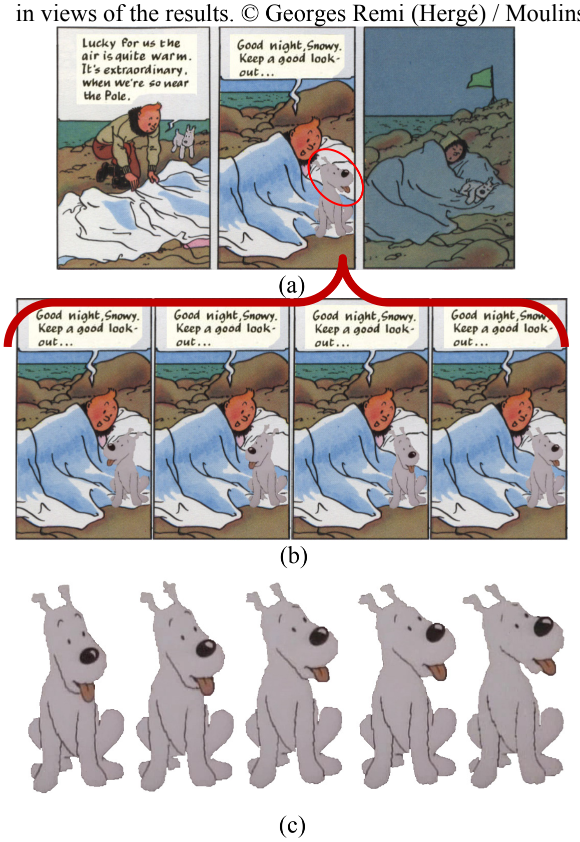


Figure 8. Character deformation in a comic with ERBFs and LOESS. (a) The frames of the comic. (b) The synthesized frames from a single input frame. (c) The zoom-in views of the results. © Georges Remi (Hergé) / Moulinsart

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

$$\mu_y = \frac{\sum_i w_i y_i}{n}, \tag{14}$$

$$\sigma_x^2 = \frac{\sum_i w_i \left(x_i - \mu_x\right)^2}{n}, \tag{15}$$

$$\sigma_{xy} = \frac{\sum_i w_i \left(x_i - \mu_x\right)\left(y_i - \mu_y\right)}{n}. \tag{16}$$

Then, an estimated target coordinate $\hat{y}_j$ of a new sample x_j within the contours can be computed as follows:

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

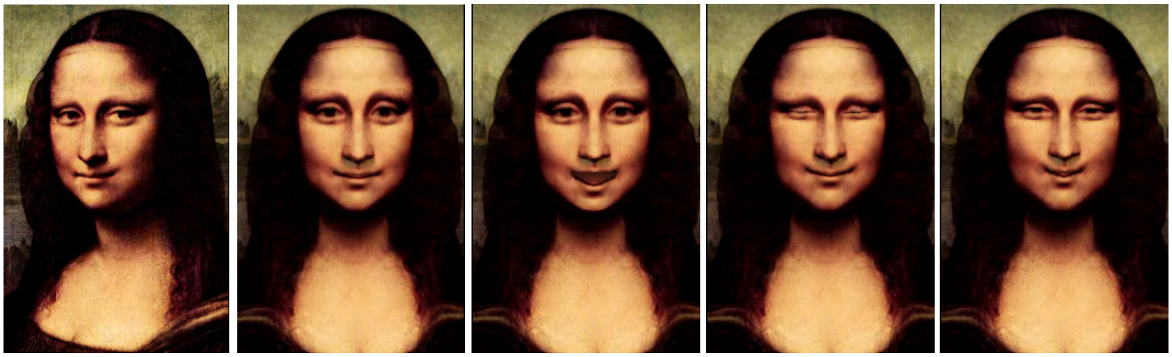
Therefore, we would reconstruct the details within the contours fitted with ERBFs via a simple closed-form solution. After contour fitting and detail preserving, character deformation is carried out. In order to maintain the 3D effect of the new view, it is needed to combine with backward deformation by using color blending.

6. Results

The proposed nonparametric regression model was implemented on an Intel Pentium M 1.5 GHz processor that allows efficient animations of still characters for multimedia applications. The results could be referred to the accompanying video (Please refer to http://cg.cis.nctu.edu.tw/csa08/85_demo.zip).

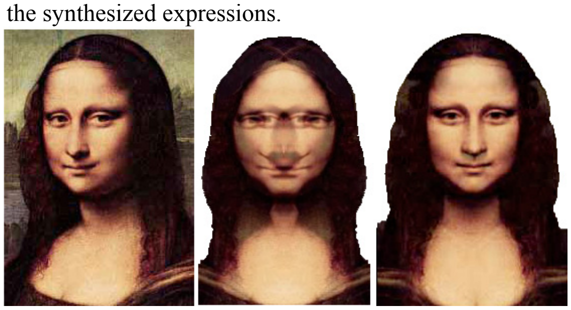
The proposed model is based on image deformation. The complete deformation process consists of three independent stages: character extraction, contour fitting, and detail preserving. In the contour fitting stage, the number of ERBFs of all examples fitting contours is decided by residual analysis. We found that an appropriate number of ERBFs is about eighteen for better fitting results in our experiments.

Our experiments were performed on digitized images obtained from "The Adventures of TinTin: The Shooting Star," which was originally produced by Georges Remi (Hergé) (Moulinsart owns exploitation rights pertaining to Hergé's work). The results are presented in Figure 1, Figure 7, and Figure 8. They show several synthesized frames of character's motion and the zoom-in views. They are only head movements. For fitting the contours, the second key-pose involves inversing the contours of the head component and concatenating with the other contours. The character would then be deformed with ERBFs and LOESS.



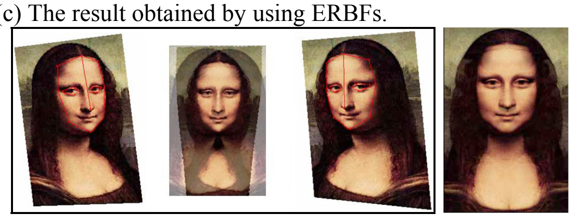
(a) (b) (c) (d) (e)

Figure 9. Character deformation with expression synthesis. (a) The original picture of Mona Lisa. (b), (c), (d), and (e) are the synthesized expressions.



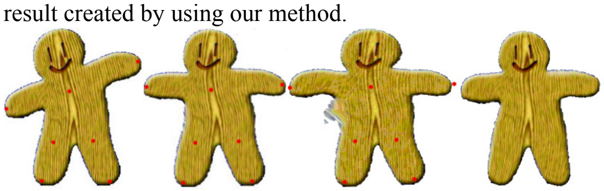
(a) (b) (c)

Figure 10. Comparison of the deformed result obtained by using RBFs and ERBFs. (a) The original picture of Mona Lisa. (b) The result obtained by using RBFs. (c) The result obtained by using ERBFs.



(a) (b)

Figure 11. Comparison with view morphing [19]. (a) Ghost occurrence in view morphing without enough correspondences (red lines are specified by users). (b) The result created by using our method.



(a) (b) (c) (d)

Figure 12. Comparison with image deformation using moving least squares [23]. (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.

Besides, we are interested in extending our deformations to facial expression and viseme synthesis for virtual human creation. With the exception of a novel view interpolation and shape deformation, several facial effects observed in character deformations, such as eye, nose, and mouth movements, could be created, as shown in Figure 9. Figure 9 (a) is the original picture of Mona Lisa. Figure 9 (b), (c), (d), and (e) are the synthesized facial expressions. We develop a module to record the input character's facial expression and viseme. By moving the facial features obtained from the structure of the spatial relation, which we constructed before, we simulate the dynamics 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 character's head. We would further retarget the expression onto another character.

Figure 10 shows a comparison of deformed characters obtained by using RBFs and ERBFs respectively. The number of basis functions is the same. Ghost effects are observed in the final result using RBFs even though feature alignment is achieved in contours fitting. The quality of the final blending result with ERBFs is better.

As mentioned above, previous techniques such as view interpolation and shape deformation may be able to produce good quality results. However, both techniques needed user intervention. Figures 11 and 12 provide comparisons with view morphing proposed by Seitz and Dyer [19] and image deformation using moving least squares proposed by Schaefer et al. [23]. In view morphing, it is necessary to compute an additional fundamental matrix for camera calibration. Further, many users' specifications are required for correspondences. Figure 11 (a) 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 Figure 12. This man-made situation or interference would not occur in the proposed method. Our method would be automatic in character deformation process.

7. Conclusion

We propose a novel multimedia application based on image deformation with a nonparametric regression model using ERBFs and LOESS for character animation and virtual human creation. By using the model, deformation problems that are commonly observed for characters in nonspherical structures would be solved and animated without unnatural distortion. We have shown visual results for the purpose of comparison. The prediction performance of our algorithm is considerably limited by the structure of the input image. The proposed algorithm 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 the 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 character with the nonparametric model using ERBFs and LOESS. The time series model would be applied to retarget the motion onto any similar characters for advanced multimedia applications.

8. References

[1] M. Alexa, D. Cohen-Or, and D. Levin. As-rigid-as-possible shape interpolation. In SIGGRAPH '00, pages 157–164, 2000.

[2] N. Arad, N. Dyn, D. Reisfeld, and Y. Yeshurun. Image warping by radial basis functions: applications to facial expressions. CVGIP Graph Models Image Processing, 56, 2, pages 161–172, 1994.

[3] T. F. Chan. Active contours without edges. IEEE Transactions on Image Processing 10, 2, pages 266–277, 2001.

[4] S. E. Chen and L. William. View interpolation for image synthesis. In SIGGRAPH '93, pages 279–288, 1993.

[5] Y.-Y. Chuang, D. B. Goldman, K. C. Zheng, B. Curless, D. Salesin, and R. Szeliski. Animating pictures with stochastic motion textures. ACM Transaction on Graphics, 24, 3, pages 853–860, 2005.

[6] T. Fu and H. Foroosh. Expression morphing from distant viewpoints. In Proceedings of International Conference on Image Processing 2004, pages 3519- 3522, 2004.

[7] E. Goldstein and C. Gotsman. Polygon morphing using a multiresolution representation. In Graphics Interface '95, pages 247-254, 1995.

[8] A. Hornung, E. Dekkers, and L. Kobbelt. Character animation from 2D pictures and 3D motion data. ACM Transaction on Graphics, 26, 1, Article No. 1, 2007.

[9] R. Herbrich. Learning kernel classifiers theory and algorithms, The MIT Press, 2002.

[10] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. ACM Transaction on Graphics, 24, 3, pages 1134–1141, 2005.

[11] Y. Jang, R. P. Botchen, A. Lauser, D. S. Ebert, K. P. Gaither, and T. Ertl. Enhancing the interactive visualization of procedurally encoded multifield data with ellipsoidal basis functions. Computer Graphics Forum 25, 3, pages 587-596, 2006.

[12] P. Litwinowicz and L. Williams. Animating images with drawings. In SIGGRAPH '94, pages 409–412, 1994.

[13] D. C. Montgomery, E. A. Peck, G. G. Vining. Introduction to linear regression analysis, Wiley, 2006.

[14] R. Mukundan, S. H. Ong, and P. A. Lee. Image analysis by tchebichef moments. IEEE Transactions on Image Processing 10, 9, pages 1357–1364, 2001.

[15] J. Park and I. W. Sandberg. Nonlinear approximations using elliptic basis function networks. In 32nd Conference on Decision and Control '93, pages 3700-3705, 1993.

[16] V. Ranjan and A. Fournier. Matching and interpolation of shapes using unions of circles. Computer Graphics Forum 15, 3, pages 129-142, 1996.

[17] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Transaction on Graphics, 23, 3, pages 309–314, 2004.

[18] D. Ruprecht and H. Müller. Image warping with scattered data interpolation. IEEE Computer Graphics and Applications 15, 2, pages 37–43, 1995.

[19] S. M. Seitz and C. R. Dyer. View morphing. In SIGGRAPH '96, pages 21-30, 1996.

[20] J. A. Sethian. Level set methods, Cambridge University Press, 1996.

[21] J. A. Sethian. Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge University Press, 1999.

[22] T. Sederberg and E. Greenwood. A physically based approach to 2D shape blending. In SIGGRAPH '92, pages 25-34, 1992.

[23] S. Schaefer, T. Mcphail, and J. Warren. Image deformation using moving least squares. ACM Transaction on Graphics, 25, 3, pages 533–540, 2006.

[24] S. Vedula, S. Baker, and T. Kanade. Image-based spatio-temporal modeling and view interpolation of dynamic events. ACM Transaction on Graphics, 24, 2, pages 240–261, 2005.

[25] S. A. Vorobyov and A. Cichocki. Hyper radial basis function neural networks for interference cancellation with nonlinear processing of reference signal. Digital Signal Processing 11, 3, pages 204-221, 2001.

[26] G. Wolberg. Image morphing: a survey. The Visual Computer 14, 8/9, pages 360–372, 1998.