COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2011; **22**:457–470
Published online 18 March 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cav.330

RESEARCH ARTICLE

Comic character animation using Bayesian estimation

Yun-Feng Chou* and Zen-Chung Shih

Department of Computer Science, National Chiao Tung University, Taiwan, ROC

ABSTRACT

The motion of comic characters includes different types of movements, such as walking or running. In a comic, a movement may be described by a series of non-continuous poses in a sequence of contiguous frames. Each pose exists in a frame. We synthesize an animation according to still comic frames. In this paper, we propose a model to analyze time series of a character's motion using the non-parametric Bayesian approach. Then we can automatically generate a sequence of motions by using the estimated time series. Experimental results show that the built time series model best matches the given frames. Furthermore, unnatural distortions of the results are minimized. Copyright © 2011 John Wiley & Sons, Ltd.

KEYWORDS

image deformation; functional approximation; Bayesian inference; elliptic radial basis functions; locally weighted regression; time series

***Correspondence**

Yun-Feng Chou, Department of Computer Science, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 30010, Taiwan, ROC.
E-mail: gis92819@cis.nctu.edu.tw

1. INTRODUCTION

This paper presents a method for synthesizing the animation of characters in a comic. In a comic, no apparent continuous motions of characters exist in every two consecutive frames. While these two consecutive frames represent two time-slice poses in a continuous movement only, they cannot represent the movement completely. Generating a natural-looking animation is still a major challenge in computer graphics. Basically, a motion is an essential 3D transformation problem which consists of a 2D spatial displacement and a 1D shift in time. So we construct a time series model to synthesize a character's motion. The proposed method involves a Bayesian approach for building the time series model, which is adopted to fit the motion trajectories of a character.

An approach of human motion synthesis is the constraint-based motion synthesis [1]. It is formulated in a maximum-a-posterior (MAP) framework. The statistical framework is approximated by only using the likelihood and prior terms, which is equivalent to the minimization of an error function. However, the framework only correlates with the training data. It does not necessarily give a small error with respect to new data.

We also adopt a statistical method to synthesize motions. First, we simulate key-motions of a character between two contiguous frames in a comic by using kernel regression with elliptic radial basis functions (ERBFs). A key-motion is defined as the contour of an in-between pose between the poses of a character in two contiguous frames of a comic. Note that ERBF kernel is suitable to perform scattered-data interpolation and is applicable to fit the human-like shape. ERBF kernel was proposed for nonlinear approximation of functions in certain general spaces (referred to as elliptic basis function networks [2]). Moreover, a volumetric approximation and visualization system was developed with ellipsoidal Gaussian functions for a 3D volume (referred to as ellipsoidal basis functions [3]).

Besides, we obtain the regression parameters suitable for the current motion of a character by using Bayesian inference, which is based on the reversible jump Markov chains Monte Carlo (RJMCMC) method [4]. RJMCMC has advantages for this task. Note that RJMCMC generates a sequence or a chain of samples. Apart from the initial sample, each sample is derived from the previous sample, which allows the algorithm to find parameters that satisfy the situation of current regression model. We do not use least-square error to find the parameters because the least-square method leads to local minimization.

However, these simulated key-motions are described discretely in the temporal domain. For generating a smooth and continuous character animation, we synthesize the contours of a character's motion following the motion trajectory that is obtained by the proposed time series model called by the Bayesian version of autoregressive moving average (BARMA). BARMA integrates autoregressive moving average (ARMA) with a non-parametric Bayesian approach that is based on kernel regression with ERBFs and RJMCMC-based model estimation. Note that the model is trained from the key-motions.

After generating a sequence of motion, a local-fitting methodology is applied to preserve important features within contours (that is filling in the color and texture information obtained from the original character in the given frame). Locally weighted regression (LOESS) is a way of estimating the regression surface through a multivariate smoothing procedure by fitting a function of independent variables locally, which maintains features invariant during deformations without unnatural distortion. The Bayesian version of LOESS (BLOESS) is proposed to improve meaningful regressions by using Bayesian inference to infer regression coefficients in LOESS.

In summary, this paper makes the following contributions:

- Given two contiguous frames in a comic, the contours of a character's key-motions are synthesized by using a Bayesian estimation of kernel regression, which combines ERBF kernel with RJMCMC. This approach is suitable to fit the natural shape of a character, such as a human body or an essentially human-like subject.
- BARMA is proposed to analyze the motion trajectory of a character's motion through a non-parametric Bayesian approach. The Bayesian approach is constructed for adding the smooth variety of the time series data by using ERBF kernel and RJMCMC described above.
- BARMA is applied to synthesize the contours of the whole character's motion. Furthermore, BLOESS is applied to preserve the details or features of characters. The Bayesian approach improves meaningful regressions even with fewer data points than regression coefficients.

The rest of this paper is organized as follows. The Related Work section reviews related literatures about character animation. The Overview section then gives an overview of the character animation process. Next, the Bayesian-based Character Animation section describes the proposed time series model through a non-parametric Bayesian approach for character animation. Implementation results are shown in the Results section. Finally, the detailed conclusions are drawn in the Conclusion section, along with recommendations for future research.

2. RELATED WORK

Various techniques have been applied to animate characters by using image morphing, motion capture, shape deformation, or time series.

2.1. Image morphing approaches

Several methods [5] have extracted properties of given key-poses, and used it to generate characters' motions in image morphing community. Chuang *et al*. [6] adopted a wavelet curve descriptor combined with Lagrangian dynamics to implement the animation and the morphing. The wavelet coefficients can represent the shapes of images in different resolutions. Lagrangian dynamic equation can be applied to simulate periodic motions. They utilized a non-self-intersecting contour morphing to produce the motion of a similar nature by generating in-betweens. Shutler and Nixon [7] derived Zernike velocity moments from the video about a character's locomotion. Then they used the Zernike velocity moments to reconstruct the silhouette of an occluded character's locomotion which preserves a smooth transition. Our method only employs the correspondence of a character in consecutive frames of a comic to synthesize the character's motion.

Besides, several studies [8,9] referred to character motions synthesis have been conducted using radial basis functions (RBFs). DeJuan and Bodenheimer [10] synthesized in-between contours and textures by given two key frames in an animation. Contour points and corresponding normals of a character in a key frame were used in RBF method to interpolate an implicit surface. Then a 3D mesh describing the surface was generated. The mesh was sliced in the middle to create in-between contours. In-between textures were synthesized by using an elastic registration. Our approach fits contours with ERBF kernel in image space directly. ERBF has the advantage of RBF-like smoothness and is applicable to more general shapes than RBF. Besides, in-between textures they created would be distorted in complex patterns made up of a few solid colors. We use BLOESS to preserve the details without undesired distortion.

2.2. Motion capture approaches

Conversely, motion capture technology has enabled users to accumulate large database of a human motion which makes the construction of empirical models of a motion feasible. Hornung *et al*. [11] accomplished the motion of photographed persons by projecting them to 3D motion data. However, their method stipulated extra 3D information, including 3D model construction or a 3D motion database, thus increasing the overloads which do not belong to image reanimation. Although the approach of Hornung *et al.* 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. The time series model based on a non-parametric Bayesian approach does not have this limitation.

2.3. Shape deformation approaches

Researches on image-based animation [12,13] have recently been carried out based on the shape deformation of a single image. Schaefer *et al*. [14] proposed a rigid transformation approach with moving least squares. The study of Schaefer *et al.* concentrated on specifying deformation by using user-specified handles. To generate animation, users needed to set the next pose by manipulating the 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.

2.4. Time series approaches

Time series has been popularly applied in statistics to forecast the trends in finance and marketing [15,16]. They have also been used in control system, pattern recognition, and artificial intelligence [17,18]. In computer graphics, they are adopted for aging trajectories prediction or character motion synthesis. For example, Scherbaum *et al*. [19] applied aging prediction to images of faces with 3D model reconstruction and support vector regression based on RBF kernel. Cai and Hodgins [1] generated an animation from various user-defined constraints. Their system learned a state space model from motion capture data. This state space model was based on the deformed time series model, and was constructed from the concept of autoregressive model. They transferred constraint-based motion synthesis to the MAP problem, and developed an optimization framework which generated a natural motion.

Furthermore, variants of hidden Markov models (HMMs) [20,21] have been widely used to create the time series data of motion trajectories representing a character's motion. HMMs learned from human motion data have been employed to interpolate key frames, and to synthesize a new style of motion. However, these statistical schemes require full information about the character motion to train the initial statistical model. For example, a large motion capture database of human body, or a large amount of user intervention for constraints, is necessary. In a general comic, the information in any two time-sliced poses cannot completely convey the movement. Therefore, our proposed approach learns a statistical dynamic model based on time series. Moreover, the dynamic behavior of our proposed model is predicted by time series. More significantly, in contrast to previous methods, the proposed model allows the user to animate characters smoothly without additional 3D information.

3. OVERVIEW

The proposed approach for generating character animation consists of the following four components: shape structure, Bayesian regression, time series, and details preservation.

3.1. Shape structure

A hard character matte is obtained by using level-set-based *GrabCut*, as shown in Figure 1(b). The foreground and background are adequately separated. The moving components are defined simultaneously. Note that a moving component denotes the basic unit of a character's motion. To create convincing animations of a 2D character, its shape needs to be deformed plausibly, while maintaining the effort

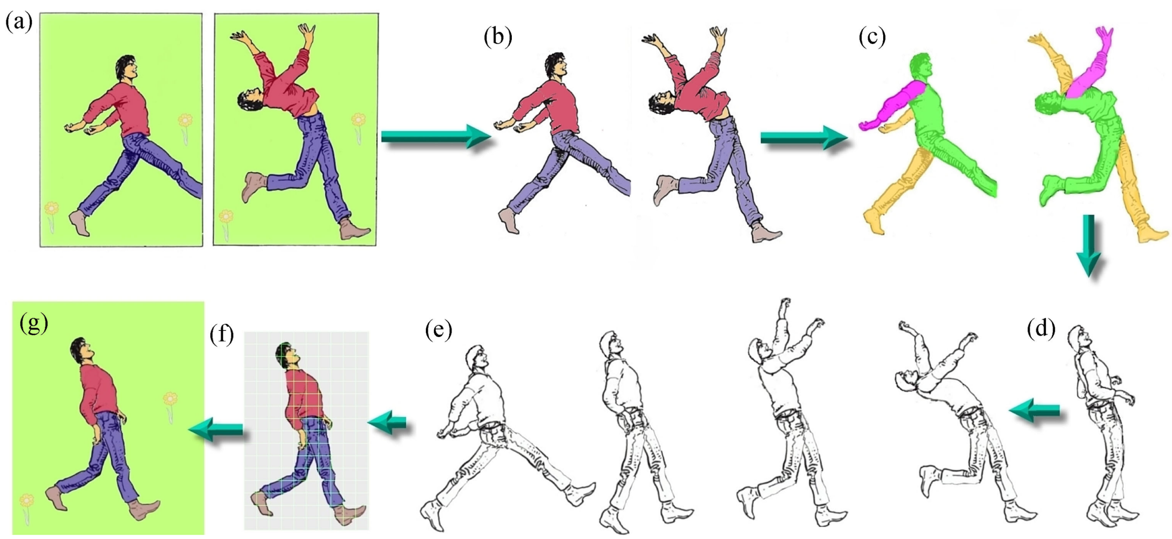

Figure 1. The overview of our approach for generating characters animation. (a) Considering two poses in consecutive frames of the source comic, (b) the character is extracted by level-set-based *GrabCut*. (c) We construct the shape structure and refine it (here: the same color represents as the same level in the shape structure). (d) The key-motion is synthesized by Bayesian regression. (e) Then the time series is estimated to synthesize the whole motion. (f) The intermediate color is overlaid on the morphed contour by BLOESS. (g) The character animation in a comic is generated by using our method.

for generating animations on three generic body layers. These body layers denote topological changes of a character model for different camera viewpoints, as shown in Figure 1(c). We abstract the character in order to construct the shape structure by using toon shading. The skeleton of a character is specified manually by using a predefined 2D skeleton structure.

3.2. Bayesian regression

Anchor points are first sampled along the contours of a character in a frame. The shape deformation function between these samples and correspondences in another frame is trained by using Bayesian regression, which is a Bayesian inference of kernel regression. Note that the ERBF kernel is adopted for regression analysis, and RJMCMC is applied to infer the optimized regression parameters. The deformation function is used to fit the morphed contours for interpolating key-motions. For instance, Figure 1(d) shows the key-motion obtained by using the deformation function. The function is trained from two poses of a character shown in Figure 1(c). Key-motions are applied to construct the time series model further.

3.3. Time series

ARMA is a useful time series model for human motion or stable time series data. Given the key-motions, the entire motion is synthesized by using BARMA, which integrates Bayesian regression [22] estimated above with ARMA. BARMA is applied to predict the motion trajectories between the key-motions.

3.4. Details preservation

The trajectories are applied to fit contours for synthesizing a series of motions, as shown in Figure 1(e). Then the details of character are preserved by using BLOESS. In other words, BLOESS is employed to fill in the color and texture information obtained from the original character in the given frame, as shown in Figure 1(f). The motions are synthesized in accordance with the previously fitted contours and details, as shown in Figure 1(g).

4. BAYESIAN-BASED CHARACTER ANIMATION

In this section, we explain our method in detail. The shape structure is constructed first. Then Bayesian regression with ERBF kernel and RJMCMC is applied for key-motions generations. Next, the time series model BARMA is constructed and used to estimate the motion trajectory. Finally, the animation of a character is synthesized by using BARMA and BLOESS.

4.1. Shape structure

In this stage, similar regions are extracted by approximating the dynamics of moving curves. This method is known as the level set method [23,24] proposed by Osher and Sethian. Chan [25] developed the level set method to detect objects in a given image. After abstracting the character by using toon shading, we apply his method to segment regions with the similar color distribution. 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. 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. Next, the bounding box of these regions is applied for *GrabCut* [26] to separate the foreground and background. Note that the extracted regions correspond to the regions of a character matte with the similar color distribution. The pixels inside the contours of 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 boundaries of a character matte are extracted automatically. Furthermore, the moving components of a character are found simultaneously. Besides, the skeleton of each moving component is obtained by using morphology-based operations [27]. Given a predefined human skeleton structure, the skeleton of a character is specified by moving the bones of that predefined skeleton to align to the bones of the obtained skeletons of moving components. Furthermore, we can refine the skeletal bones and joints in occluded regions manually.

These moving components are further partitioned into three layers manually when animating characters from a side view. For instance, an animation might involve one layer for the foremost arm, one for the body and the foremost leg, and one for the remaining arm and leg. Moreover, these layers cannot move independently. They should be stitched together to convey the impression of a connected body when animating the character. Hence, every layer is composed of moving components, skeletal bones, and joints. Different layers are linked by the shared skeletal joints.

4.2. Bayesian regression with ERBF kernel

Basically, a shape deformation of a character is constructed for motion synthesis. Before defining the deformation, the point-to-point correspondences of anchor points are obtained. Figure 2 illustrates how to construct point-to-point correspondences. We can construct the point-to-point correspondences from these bones (blue lines) and joints (purple dots). The anchor points are sampled along the contours of the character randomly. For example, there is a point u_i sampled along the contour of a right arm randomly. u_i is called an anchor point. First, we find the projection J_m of the anchor point u_i on the line segment $\overline{J_1 J_2}$ or the extended

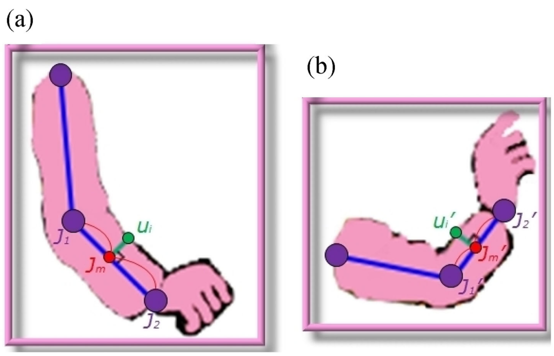

Figure 2. Point-to-point correspondences. (a) An arm of a comic character in a frame. (b) The same arm of that character in another frame.

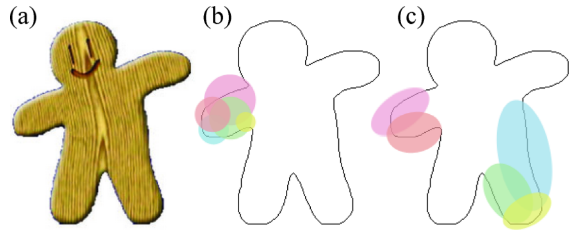

Figure 3. Comparison of the number of basis functions using Gaussians. (a) The original character. (b) Using the circular Gaussians (RBFs) needs five kernels to fit the contour of the right arm, and (c) using the arbitrary directional elliptic Gaussians (ERBFs) can fit the right arm and left leg with the same number of kernels.

line of $\overline{J_1 J_2}$. Based on the predefined skeleton structure, the skeleton correspondence between two frames is obtained. Note that the corresponding joints $J_1{}'$ and $J_2{}'$ of J_1 and J_2 are known. According to the ratio of $\frac{\overline{J_1 J_m}}{\overline{J_1 J_2}}$, we can find the point $J_m{}'$ which satisfies the constraint that $\frac{\overline{J_1{}' J_m{}'}}{\overline{J_1{}' J_2{}'}} = \frac{\overline{J_1 J_m}}{\overline{J_1 J_2}}$. Then we compute the normal vector on the point $J_m{}'$ and find the intersection of the normal vector and the contour of the right arm. The intersection point $u_i{}'$ is the correspondence of the anchor point u_i. Thus, we can obtain n anchor points sampled along the contours in a frame and their correspondences in another frame.

Then we define the deformation based on the mapping of n anchor points sampled along the contour of a character and the relative correspondences. We formulate this problem as regression analysis. In general, the response $\vec{r} = \left(r_{\Delta x}, r_{\Delta y}\right)$ is the displacement of an anchor point, and the predictor $\vec{u} = \left(u_x, u_y\right)$ is the coordinate vector of the anchor point. The relationship can be described as

$$\vec{r}_d = f(\vec{u}_d) + \varepsilon \ \text{ for } \ d = 1, \ldots, n \tag{1}$$

where $\vec{r}_d$ denotes the displacement of the dth anchor point $\vec{u}_d$. Considering the above equation, $f(.)$ denotes an unknown and smooth surface indicating the true relationship between $\vec{r}$ and $\vec{u}$, commonly termed the regression surface representing the shape deformation. Additionally, the error ε is assumed to come from a *Normal distribution* $N\left(0, \tau^2\right)$ in Equation (1), where τ^2 denotes the noise variance.

Previously, Gaussian function was chosen as the basic function of RBF kernel for image morphing. It is suitable to fit smooth functions for the form of $f(.)$. However, RBF, which is a circularly shaped basis function, has a limitation in fitting long, high-gradient shapes such as characters' 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 characters' shape such as the body or head of a human. Therefore, instead of a general RBF kernel, the regression surface is estimated using kernel regression with ERBFs. ERBF has the advantage of RBF-like smoothness and is applicable to fit more general shapes than RBF.

In contrast, elliptic Gaussian (ERBF) is an appropriate choice to fit contours, which have non-circular structures, as demonstrated in Figure 3. Using too many circular Gaussians (RBFs) to fit contours takes more learning time. Although ERBFs require more computation time than RBFs in the optimization phase, they obtain better quality with fewer basis functions. Furthermore, this paper develops a Bayesian estimation of ERBFs, which has the merits over the conventional ERBFs, to make an inference by RJMCMC. Bayesian estimation frequently gives better predictions from the posterior mean of the generated samples of ERBFs.

On the following, ERBF kernel is briefly summarized. Let $\vec{v} = \left(\mu_x, \mu_y\right)$ be the center vector of elliptic Gaussian. The regression model is developed as a linear combination of a set of basis functions and their corresponding coefficients as follows:

$$f(\vec{u}) = \sum_{j=1}^{k} \beta_j \eta\left(\vec{u}, \vec{v}_j\right) + T(\vec{u}) \tag{2}$$

where β_j denotes the suitably chosen coefficient of the jth elliptic Gaussian $\eta(.)$, $\vec{v}_j$ is the related center vector, and k is the number of basis functions in the model. In our work, $T(.)$ represents 2D affine transformation. The parameters related to basis functions are determined by the data. According to the correspondences of anchor points between two frames, controlling the affine component $T(.)$ is carried out by a least-squares approximation procedure, perhaps using matrix pseudo-inverse techniques.

In this work, the contour of a character for motion synthesis can be fitted by ERBF kernel. ERBFs predict well, and are extremely interpretable, especially the arbitrary directional ERBFs. Hence, this investigation develops a generalization of the arbitrary directional ERBFs. Moreover, ERBF is derived from a hyper radial basis function (HRBF) using the *Mahalanobis distance* [28]. For the arbi-

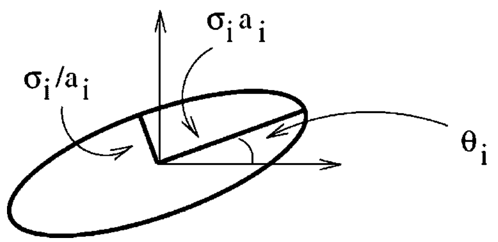

Figure 4. Schematic diagram of an arbitrary directional elliptic radial basis function (ERBF).

trary directional ERBF, $\eta(.)$ is given in a matrix form by

$$\eta(\vec{u},\vec{v}) = \exp\left\{-\frac{(\vec{u}_x-\vec{v}_x)^{\mathrm{T}}A_{\theta_x,a_x}(\vec{u}_x-\vec{v}_x)}{2\sigma_x^2}-\frac{(\vec{u}_y-\vec{v}_y)^{\mathrm{T}}A_{\theta_y,a_y}(\vec{u}_y-\vec{v}_y)}{2\sigma_y^2}\right\},$$
$$\vec{u}=\vec{u}_x+\vec{u}_y=[x \quad 0]^{\mathrm{T}}+[0 \quad y]^{\mathrm{T}},$$
$$\vec{v}=\vec{v}_x+\vec{v}_y=[\mu_x \quad 0]^{\mathrm{T}}+[0 \quad \mu_y]^{\mathrm{T}} \tag{3}$$

$$A_{\theta_i,a_i}=\begin{bmatrix}\cos\theta_i/a_i & \sin\theta_i/a_i\\ -a_i\sin\theta_i & a_i\cos\theta_i\end{bmatrix},\ \text{where } i\in\{x,y\} \tag{4}$$

where ${\sigma_i}^2$ for $i\in\{x, y\}$ is the covariance of elliptic Gaussians along i-axis. Note that we use k-means clustering to set the center vector $\vec{v}_j$ for each k-means group of anchor points. In addition, the covariance is computed for each group. Besides, the orientation θ_i, which is the angle between the major axis of ellipse and i-axis, and the aspect ratio ${a_i}^2$ are applied to transfer to the arbitrary directional ERBFs, as illustrated in Figure 4. Furthermore, the transformation matrix A_{θ_i,a_i}, which contains a rotation and scaling component, is added for alignment along the data distribution. In our work, the major axis of ellipse is aligned along the contour of a character. We would estimate the most suitable parameters to fit the contour of character by Bayesian inference.

The central process of Bayesian inference is the calculation of probability distributions on the unknown parameter vectors. First, let k, $\vec{v}_j$, and ${\sigma_j}^2$ be variables and ${a_i}^2$ term be fixed. Note that k is the number of elliptic Gaussians. $\vec{v}_j$ and ${\sigma_j}^2$ denote the mean and the covariance of the jth elliptic Gaussian. θ_i and A_{θ_i,a_i} are set up according to the principal component of anchor points sampled from contours. Note that the parameter space Θ can be written as $\Theta=\left\{\beta_j,\ \vec{v}_j,\ {\sigma_j}^2\right\}$ for $j=1,...,k$. Taking D to represent our training data, which are the anchor points and their displacements, we are interested in inference about the posterior probability of parameters Θ conditional on the data, i.e. $p(\Theta\,|D)$. Recalling Equation (1), given a new contour point $\vec{u}_{\text{new}}$ (any point on the contour of character), the target response $\vec{r}_{\text{new}}$ (the displacement of that point) can be given as an expectation.

$$E\left[\vec{r}_{\text{new}}\,|\vec{u}_{\text{new}},D\right]=\int\hat{f}\,(\vec{u}_{\text{new}},\Theta)\,p\,(\Theta\,|D)\,\mathrm{d}\Theta \tag{5}$$

where $\hat{f}(.)$ is the estimation of our ERBF model. However, the integral is intractable and untenable for asymptotic methods. We propose a Bayesian estimation of ERBFs. The proposed method imitates the ERBF procedure by RJMCMC which can approximate integrals of Equation (5), as described in Appendix. RJMCMC proceeds by drawing samples of Θ in direct proportion to $p\,(\Theta\,|D)$ and then approximates Equation (5) by

$$E\left[\vec{r}_{\text{new}}\,|\vec{u}_{\text{new}},D\right]\approx\frac{\sum\limits_{t=n_0+1}^{N}\hat{f}\,(\vec{u}_{\text{new}},\Theta_t)}{N-n_0} \tag{6}$$

where N is the number of samples generated, called the Markov chain length, and n_0 is the burn-in period. Θ_t denotes the current parameter space while there are t samples. The burn-in stage discards the samples generated by Markov chain with unstable distribution of interest $p\,(\Theta\,|D)$. If the number of states is greater than the discarded portion, then compute $\hat{f}$ (.) by the recorded parameters of the current model, ERBF term, and affine term. All the simulations are run with a burn-in period of 5000 iterations of RJMCMC followed by 10 000 samples. Finally, we use Equation (6) to generalize the displacement of the character's contour. We can make predictions of the displacement $\vec{r}_{\text{new}}$ of new contour point $\vec{u}_{\text{new}}$ using Equation (6). Furthermore, we use *Catmull-Rom splines* to connect new positions of the contour points in in-betweens. So the contours of key-motions are synthesized by the model. In our implementation, we synthesize 10 key-motions between two contiguous frames in a comic.

4.3. The time series model

As mentioned above, these key-motions are described discretely in the temporal domain. For generating a smooth and continuous character animation, we synthesize the motion following the motion trajectory that is obtained by time series. We propose a non-parametric Bayesian approach to analyze time series representing the motion trajectory. The general form of a time series model considered in this work is

$$d_t=f_{\mathrm{TS}}\left(d_{t-1},...,d_{t-p};\alpha_{t-1},...,\alpha_{t-q}\right)+\alpha_t \tag{7}$$

$$\alpha_t=C\nu_t \tag{8}$$

where d_t denotes a univariate time series and $f_{\mathrm{TS}}(.)$ indicates an unknown function of time series. p and q represent non-negative integers. ν_t is a sequence of random variables assumed to come from a *Normal distribution* with mean zero and variance one. C is assumed to be a constant. Based on this general form, ARMA is formulated as follows:

$$f_{\mathrm{TS}}\left(d_{t-1},...,d_{t-p};\alpha_{t-1},...,\alpha_{t-q}\right)=\sum\nolimits_{i=1}^{p}\phi_i d_{t-i}-\sum\nolimits_{i=1}^{q}\kappa_i\alpha_{t-i} \tag{9}$$

Comp. Anim. Virtual Worlds 2011; **22**:457–470 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/cav

where ϕ_i and κ_i are the coefficients of parameters in this model. It is similar to the time series model proposed by Chen *et al*. [29], except that they assumed the functional form of $f_{TS}(.)$ was a known linear function whereas we assumes $f_{TS}(.)$ is estimated non-parametrically along with the Bayesian estimation of ERBFs already described in order to add smooth variety of the time series data. Equation (9) can be re-written as follows:

$$f_{TS}\left(d_{t-1}, ..., d_{t-p}; \alpha_{t-1}, ..., \alpha_{t-q}\right) = \sum_{i=1}^{p} \phi_i \eta(d_{t-i}) - \sum_{i=1}^{q} \kappa_i \alpha_{t-i} \tag{10}$$

where $\eta(.)$ defined in Equation (3) is applied to estimate $f_{TS}(.)$.

This paper develops a Bayesian version of ARMA (BARMA) by combining ERBF kernel with RJMCMC. In our work, d_i denotes the corresponding displacements of the contour points from the ith key-motion. α_i is obtained by drawing a random variable. Give n contour points sampled from the character and the corresponding displacements, we perform Bayesian inference to obtain the optimal coefficients ϕ_i and κ_i. In our implementation, we already synthesize 10 key-motions between two contiguous frames in a comic using Bayesian regression. We find that an appropriate number of frames is about 10 in our experiments (Note that $p = 10$ and $q = 10$). These key-motions are sufficient to predict the motion trajectory of key-motions effectively. Then we use BARMA to obtain the motion trajectories. Note that the entire movement of a character is synthesized by fitting contours through the trajectories. For each frame in the temporal domain, we use Equation (7) to find the displacements of contour points in that frame.

4.4. Details preservation

In addition to contour 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 frame. The filled color and texture information would be controlled by a local-fitting methodology called LOESS [30]. In this work, let $x_i = \left(x_{i,x}, x_{i,y}\right)$ be the ith sampled contour point of the character in the given frame. $y_i = \left(y_{i,x}, y_{i,y}\right)$ denotes the measurement of the dependent variable representing the new location of x_i in a synthesized frame in the temporal domain. Suppose that the target coordinate $\hat{y}_i$ is generated using an estimated local multivariate polynomial for transformation.

$$\hat{y}_i = \zeta_1 t_1(x_i) + \zeta_2 t_2(x_i) \tag{11}$$

We have $t_1(x_i) = 1$ for a translation coefficient ζ_1 and $t_2(x_i) = x_i$ for a rotation coefficient ζ_2. Equation (11) can be rewritten in matrix form, as follows:

$$\hat{y}_i = \zeta^{\mathrm{T}} t(x_i) \tag{12}$$

where ζ is the matrix form of the coefficients vector (ζ_1, ζ_2), and $t(.)$ is the matrix form of the vector $(t_1(.), t_2(.))$.

During a LOESS prediction, the specific location x_0 within the morphed contour, which would be filled in color and texture information, is supplied. LOESS performs a linear regression on the sampled contour points weighted by a kernel centered at x_0. The regression is strongly influenced by the sampled points that lie close to x_0 according to some scaled Euclidean distance metric. This is achieved by weighting each sampled point according to its distance to x_0: a point very close to it is given a weight of one and a point far away is given a weight of zero. Given m pairs of the sampled contour points and the corresponding new locations of these points, the weight of the ith sampled contour point x_i is defined by Gaussian, as shown in Figure 5.

$$w_i(x_0) = w(x_i - x_0) = \exp\left(-s\, \|(x_i - x_0)\|^2\right) \tag{13}$$

where $1 \le i \le m$, $s = \frac{1}{2k^2}$, and $m = \sum_i w_i(x_0)$. s is a smoothing parameter that determines how fast weights decline in value as one moves away from x_0. k is the kernel width or bandwidth which controls the amount of localness in the regression. Recalling Equation (12), $\hat{\zeta}$ is chosen by minimizing locally weighted sum of squared residuals:

$$\hat{\zeta} = \underset{\zeta}{\operatorname{argmin}} \sum_{i=1}^{m} w_i(x_0)^2 \left(y_i - \zeta^{\mathrm{T}} t_i\right)^2 \tag{14}$$

where $t_i = t(x_i)$. Actually, we use a special form of LOESS called BLOESS [31] to build a model from the data. BLOESS allows meaningful regressions even with fewer data points than regression coefficients.

Note that we assume a wide Gaussian prior on the coefficient vector $\zeta \sim N\left(0, \frac{\tau^2}{p}\right)$ of the regression model in Equation (12) and a Gamma prior on the inverse of the noise variance $\frac{1}{\tau^2} \sim Gamma(0.1, 0.1)$ in common with RJMCMC sampler described in Appendix. p is the precision of the coefficient prior. Let X be the polynomial terms of data samples in the matrix form, such as t_i in Equation (14). Y denotes the response representing the matrix form of the corresponding new locations. P represents the matrix form of the precision p. W represents the diagonal matrix form of $w_i(x_0)$ for $1 \le i \le m$. $\hat{\zeta}$ is obtained from the marginal posterior distribution with posterior mean $\bar{\zeta} = \left(X^{\mathrm{T}} W^2 X + P\right)^{-1} X^{\mathrm{T}} W^2 Y$ and modified standard deviation. Note that the initial standard deviation is drawn from the noise variance τ^2 and modified to be the upper triangle of posterior variance matrix $\left(X^{\mathrm{T}} W^2 X + P\right)^{-1}$ obtained by using *Cholesky decomposition*. According to the estimated regression coefficient vector $\hat{\zeta}$, we can use Equation (12) to find the new location of x_0 and obtain the pixel values for filling in the color and texture information. In practice, we approximate the character with a uniform grid, as shown in Figure 5(a). We find the new location of each vertex

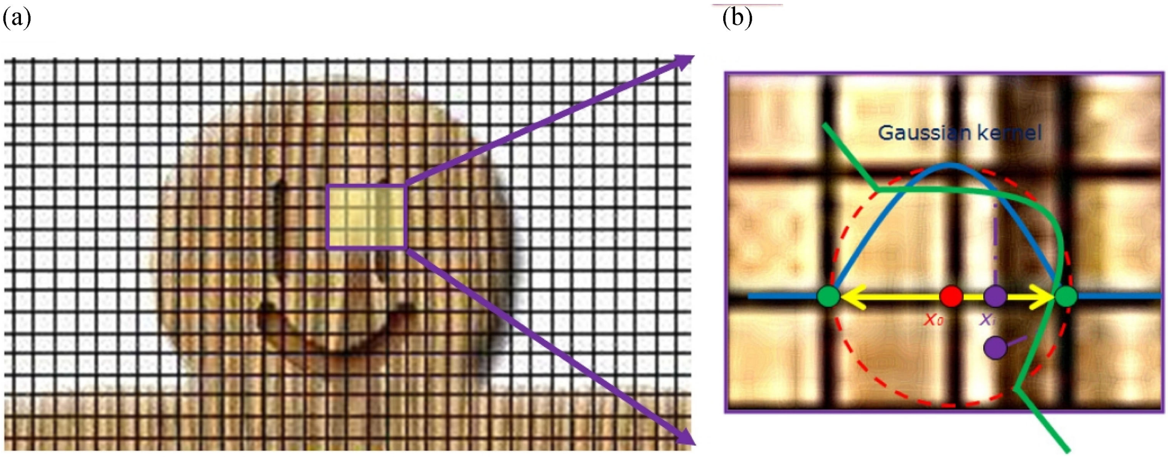

Figure 5. LOESS analysis. (a) The original character with a uniform grid (50 × 50). (b) The zoom-in view of the image. LOESS with Gaussian kernel is applied to estimate the weights.

in the grid. Then we fill the resulting quad using bilinear interpolation.

4.5. Animation generation

In summary, the complete animation generation process consists of the following phases. The shape structure of the character is built first. Each level of the shape structure of characters consists of several moving components, for instance, head, body, arm, and leg. The indications of moving components are then refined manually by the predefined 2D skeleton structure. The morphed contour of each moving component is synthesized through Bayesian regression. Key-motions are synthesized by combining all moving components using the painter's algorithm and connective constraints formed from shape structures. As mentioned before, we actually create 10 key-motions between two contiguous frames in a comic using ERBFs with the parameters estimated by RJMCMC. Then we construct the time series model BARMA to track the motion trajectory, which best matches key-motions and generates the entire character motion in contours. Bayesian regression and time series simulation are both constrained to the connection topology in the shape structure. Furthermore, BLOESS is applied to reconstruct the details within the morphed contours fitted from Bayesian regression and BARMA. The entire character animation is synthesized after contour fitting and detail preserving. Actually, we forecast 300 frames to generate the 10 seconds character animation between two contiguous frames in a comic.

Besides simulating the motion between several key-poses, we propose an additional module to make a more natural appearance of an animated character. Given other frames of a subject which focuses on a special motion of some moving component, we can simulate this motion which cannot find any prior information in those input key-poses by applying the motion trajectory of that subject.

5. RESULTS

Our approach could generate a smooth and natural-looking character animation by using the time series estimated from Bayesian inference with ERBF kernel. Moreover, the time series model is applied to represent the trajectory of a character's motion. It should be noted that, the motion trajectory could be further used to predict a character's movement by nonlinearly extrapolating reasonable deformations without the restriction of a purely interpolation method. The proposed time series model with non-parametric Bayesian inference is implemented on an Intel Pentium M 1.5 GHz

Table 1. Running times for figures

	Cat	Old Person	Ball	Man	Flowers	Tree
Figure no.	6	7	9	10	10	11
Resolution	156 × 101	311×278	100 × 75	368 × 583	120 × 213	467 × 599
Shape structure (second)	10	38	5	32	8	0
Bayesian regression (second)	2889	4898	113	5213	143	87
Time series (second)	167	244	58	274	97	32
UI (minute)	1	1	0	1	1	1

Comp. Anim. Virtual Worlds 2011; **22**:457–470 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/cav

(a)

(b)

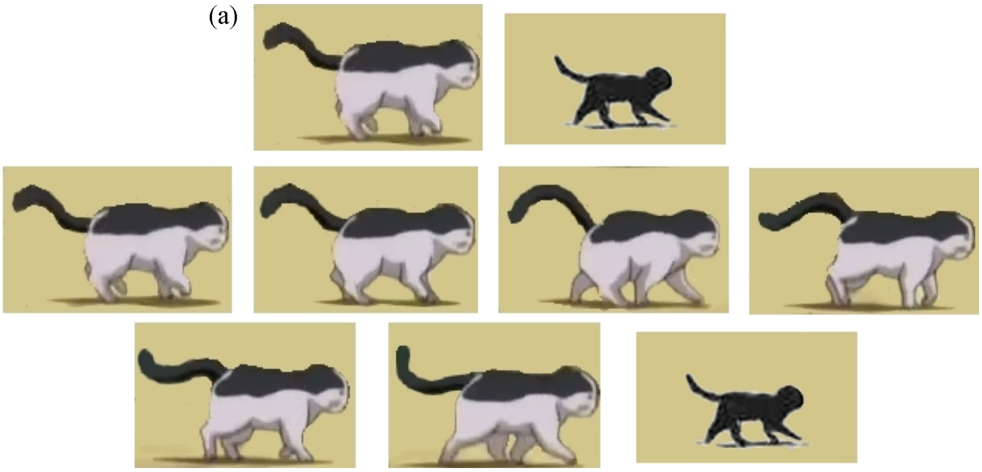

Figure 6. (a) Given two frames in the comic, (b) motion synthesis is carried out using the estimated time series for character animation.

processor, which enables smooth animations of the comic. A user specification exists by which the shape structure construction is made.

Table 1 lists the resolution and execution time for the figure shown. Execution time was measured in each step. Shape Structure consists of the time of segmentation and the whole shape structure generation. Bayesian Regression comprises the time of ERBF kernel training and RJMCMC sampling. RJMCMC sampling takes a lot longer. All the simulations are run with a burn-in period of 5000 iterations of RJMCMC followed by 10 000 samples. Time Series indicates the time to construct the time series model.

(a)

(b)

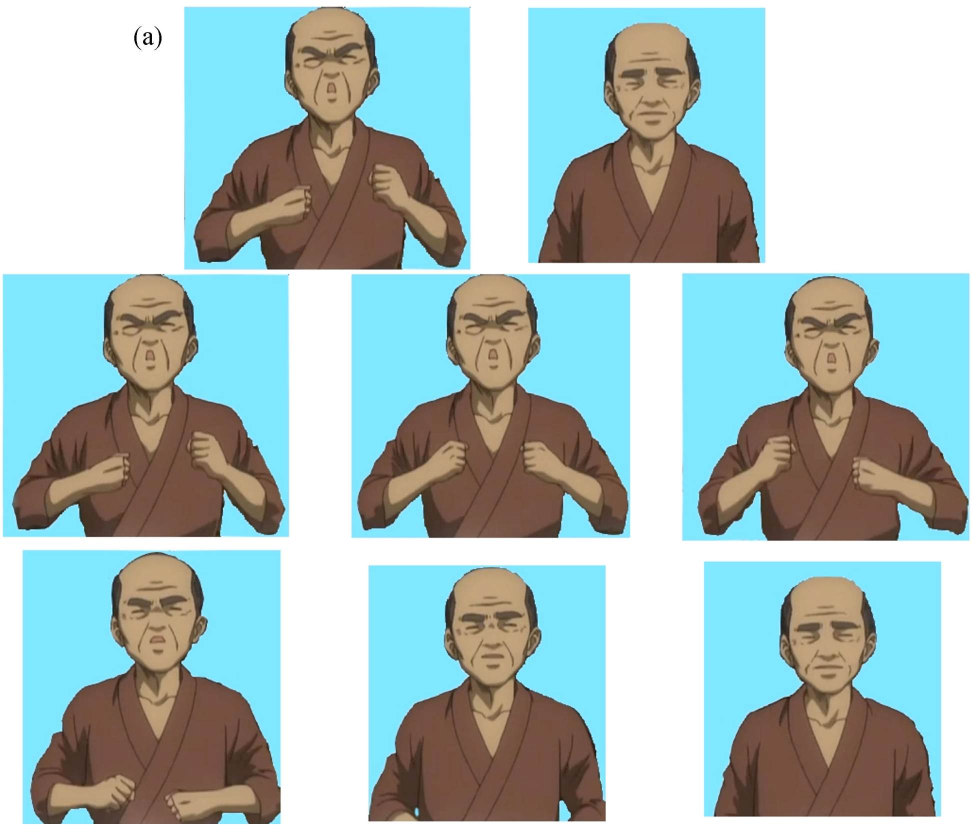

Figure 7. (a) In this example, the inputs are two consecutive frames in the source comic. (b) Motion synthesis is carried out using the estimated time series for character animation.

DOI: 10.1002/cav

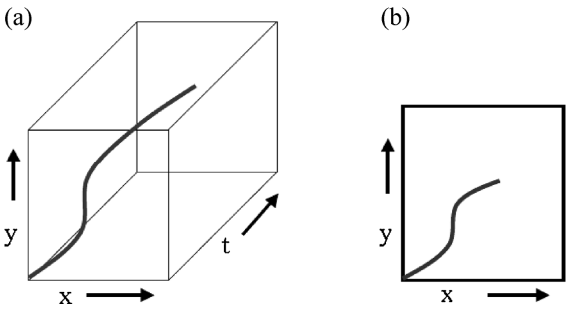

Figure 8. Two major diagrams of a motion trajectory: (a) spatio-temporal and (b) spatial.

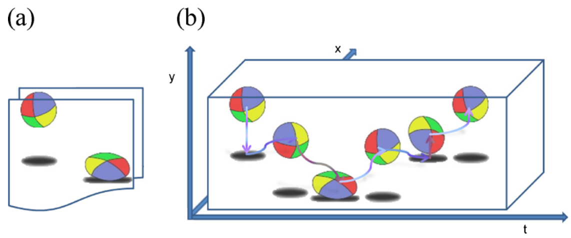

Figure 9. (a) Given two consecutive frames of a bouncing ball and (b) the synthesized frames of animation are shown with the motion trajectory.

Note RJMCMC sampling is carried out once to obtain the regression coefficients during Time Series step because the number of ERBFs is known (recalling $p = 10$ and $q = 10$ in Equation (10)). UI represents as the time of users' intervention to refine shape structure further. Besides, Details Preservation is not demonstrated since it takes less than 20 milliseconds for each frame and is quite fast. The time of animation generation is not shown because it varies significantly due to different numbers of frames generated.

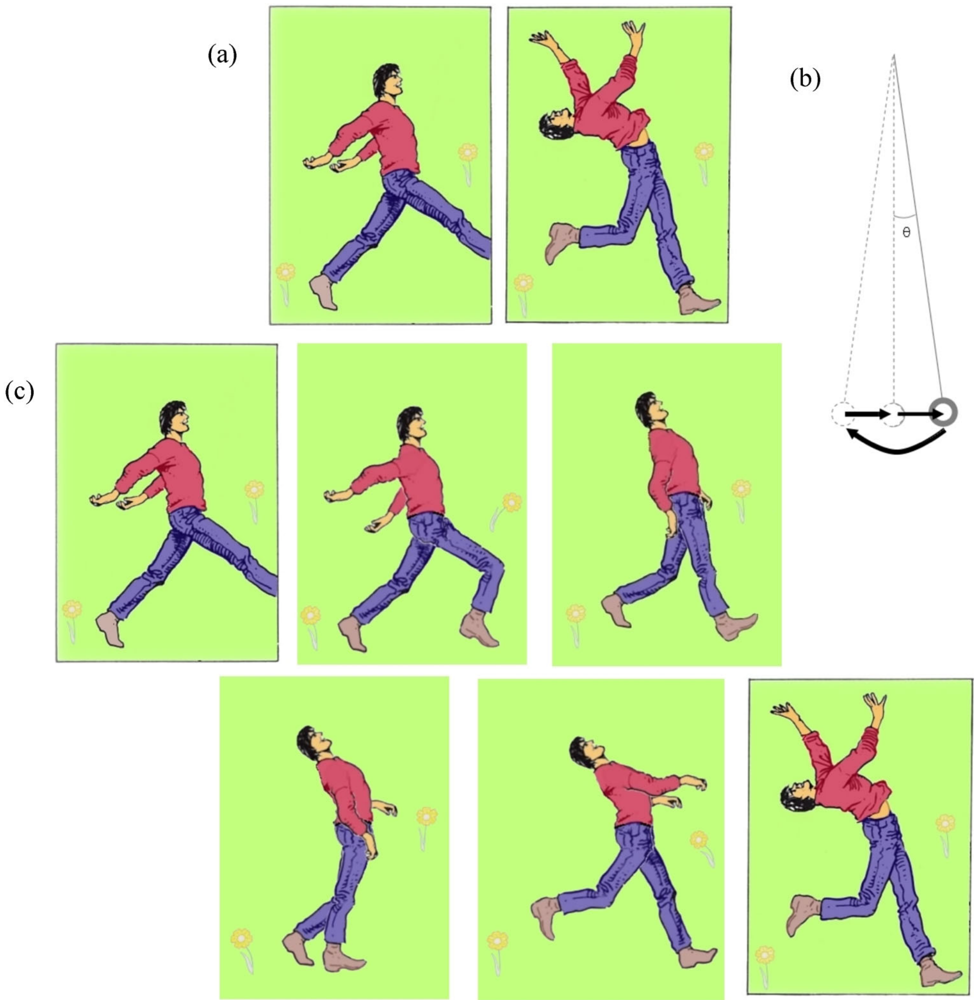

Figure 10. (a) Given two frames in the comic and (b) the motion trajectory obtained from the motion of the simple pendulum ($\theta > 5$), (c) motion synthesis is carried out using the estimated time series for character animation.

Comp. Anim. Virtual Worlds 2011; **22**:457–470 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/cav

The results which are selected frames of character animations generated by our method are presented in Figures 6 and 7. Figure 6(a) shows the original digitized frames of a comic. There are 7 moving components representing the cat, such as torso including head, right foreleg, right hind leg, left foreleg, left hind leg, tail, and shadow. The cat consists of three layers in the proposed shape structure. One layer consists of left foreleg and left hind leg. Another layer consists of right foreleg and right hind leg. Note that shadow belongs to the same layer with torso and tail. Three motion trajectories are computed by BARMA from these three layers respectively and used to simulate the motion of the cat. In Figure 7(a), there are 14 moving components representing the old person, such as head, eyebrows, eyes, nose, mouth, right arm, right hand, right long sleeve, torso, left arm, left hand, and left long sleeve. Two layers are constructed in the proposed shape structure. One layer consists of right arm, right hand, right long sleeve, left arm, left hand, and left long sleeve. Another layer consists of the other 8 moving components. The motion trajectories are also computed by BARMA for synthesizing character animation. The results reveal the strength of our method as the possibility of convincingly animating or posing any kind of comic character, as shown in Figure 6(b) and Figure 7(b) respectively.

The motion trajectory obtained by the proposed time series model is described in Figure 8. The spatio-temporal diagram, which is illustrated in Figure 8(a), captures the movement of a character in time. Then it can be mathematically represented as a curve (x, y, t) in 3D space or equivalently as a parametric curve ($x(t)$, $y(t)$) in 2D space. The spatial diagram can be mathematically represented as a 1D function $y = f(x)$. As illustrated in Figure 8(b), the spatial diagram is a projection of the spatio-temporal trajectory onto the image plane.

Figure 9(a) shows two consecutive key-poses of a bouncing ball. There is one moving component only. The motion of the bouncing ball is synthesized with its motion trajectory, as shown in Figure 9(b). The trajectory is described by the movement of the ball's barycenter. The ball is animated along with the trajectory.

(a) (b)

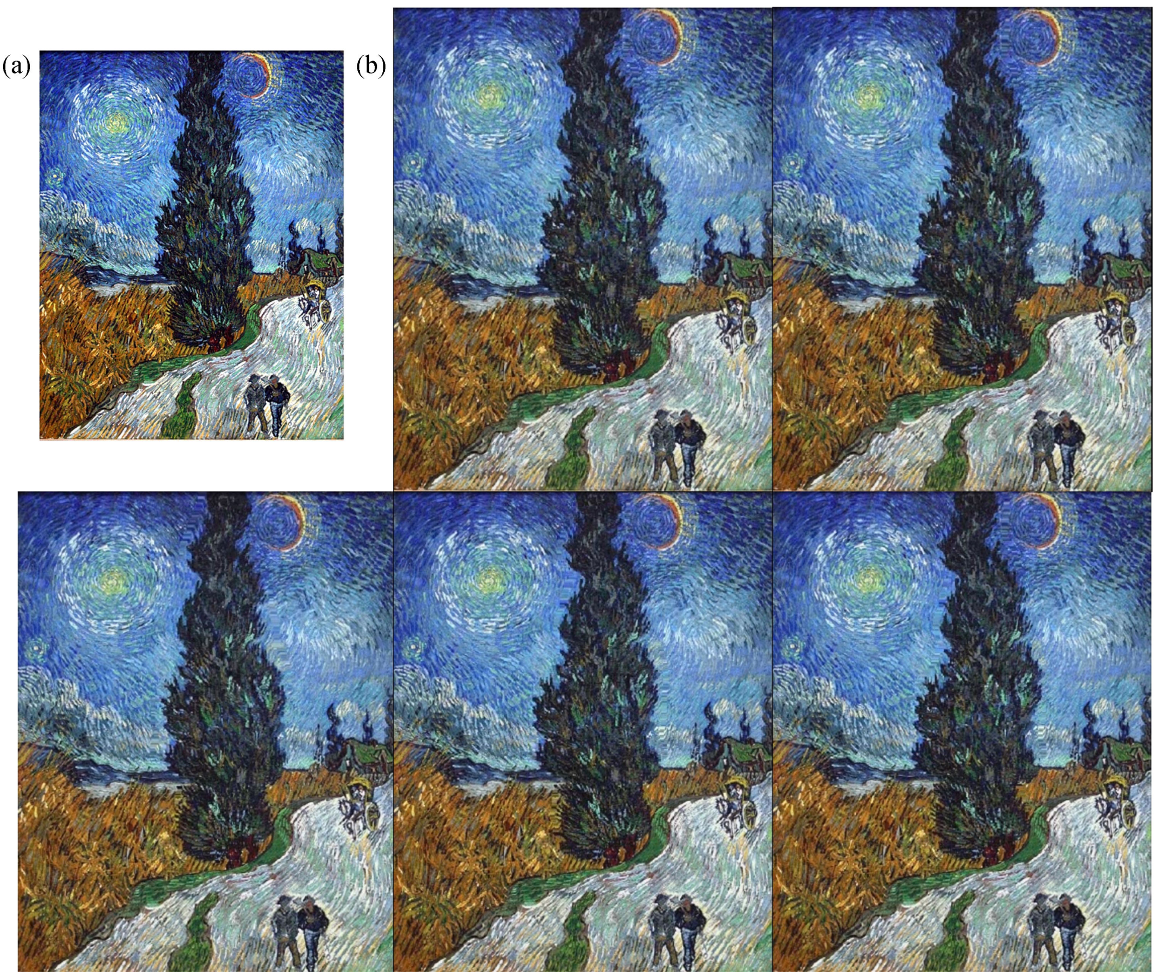

Figure 11. The input is the original painting of Vincent van Gogh's "Country Road in Provence by Night". (b) Motion synthesis is carried out using the motion trajectory obtained from the simple pendulum shown in Figure 10 (b) ($\theta < 5$).

As mentioned previously, we provide an additional retargeting module to make a natural appearance of an animated character efficiently by retargeting the motion trajectory to other characters. For example, the motion trajectory of a simple pendulum shown in Figure 10(b) can be applied to estimate the motions of those flowers, as shown in Figure 10(a). We only need to compute the motion trajectories about the character. The motions of flowers could be simulated directly from the motion trajectory of a simple pendulum.

Furthermore, Figure 10(a) shows two digitized frames of a comic. There are 12 moving components representing the scene, such as head, right arm, right hand, torso, left arm, left hand, right leg, right foot, left leg, left foot, and flowers. Moreover, three motion trajectories of the man are obtained by BARMA. Two motion trajectories of the flowers can be approximated by using the motion trajectory of that simple pendulum shown in Figure 10(b) respectively. According to the different degree of moving orientation θ, the different motion trajectory of the simple pendulum can be estimated by BARMA. Note that the motion of a simple pendulum is simple harmonic motion. The motion trajectory of the simple pendulum that we estimate is also periodic and sinusoidal-like in time. The extracted frames of the animation synthesized by using these motion trajectories are shown in Figure 10(c). The locomotion of the man is smooth. The motion of the flower is similar to harmonic oscillation driven by the wind.Moreover, Figure 11 provides another example for retargeting the motion trajectory to other characters. We would apply the harmonic oscillation of the same simple pendulum to make a tree sway. However, we estimate the motion trajectory of the simple pendulum with $\theta < 5$ by BARMA. Note that the original circular motion of the pendulum can be considered as the horizontal motion since the orientation of the motion θ shown in Figure 10(b) is smaller enough. Figure 11(a) shows the original painting of Vincent van Gogh's "Country Road in Provence by Night". Then the motion trajectory of that simple pendulum is applied to synthesize the motion of the tree. The results for motion synthesis of the tree are shown in Figure 11 (b). Furthermore, the pattern of the tree is preserved by using BLOESS.

6. CONCLUSION

In this paper, we present a time series model to animate comic characters with non-parametric Bayesian estimation. The time series model BARMA can overcome character animation generation problems in a comic. The Bayesian estimation is formed using ERBF kernel and RJMCMC. The extension to ERBFs decreases fitting time involved in alleviating the motion synthesis problems that are commonly observed for characters in non-circular structures, and RJMCMC can be applied to choose the suitable parameters for Bayesian inference. As shown before, the proposed model BARMA is flexible and appropriate for any data distribution for data prediction in the temporal domain.

The prediction performance of the proposed algorithm is considered strongly by correspondences of the input frames. The proposed algorithm will not produce a reasonable result for the lower degree of similarity between the input frames. For example, one is the back-face view, and the other is the front-face view. It may be solved through the extra information needed to handle the motion of rotation with users' interaction. In the future, we focus on enhancing the performance and quality of Bayesian inference with ERBF kernel and RJMCMC. This would enable prediction of the movement of any character in real images to a succeeding frame. Furthermore, this work could be extended to include an interactive system. We are interested in extending our Bayesian estimation to facial expression synthesis. Several facial effects are observed in character animations, such as eye, nose, or mouth movements. Given a suitable motion trajectory of facial component designed from users, the expressive facial animation synchronized with input speech could be simulated.

ACKNOWLEDGEMENTS

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

7. APPENDIX: REVERSIBLE JUMP MARKOV CHAIN MONTE CARLO (RJMCMC) SAMPLER

RJMCMC sampler is applied to estimate the optimized Bayesian regression parameters. This method consists of three steps as follows:

7.1. Set up proper priors

Recalling Equation (2), we begin with a fairly flat Gaussian prior on the basis coefficients $\beta_j \sim N\left(0, \frac{\tau^2}{p}\right)$, where p is the precision of the coefficient prior. τ^2 is the noise variance, and $\frac{1}{\tau^2} \sim Gamma\ (0.1, 0.1)$. A vague but proper Gamma prior distribution represents ignorance of the noise process and avoids inverting large matrices within each iteration of RJMCMC. We set $p = 0.01$ and $\tau^2 = 1$ initially and they would be updated during RJMCMC process.

7.2. Determine initial parameter value

Set the initial dimension k of model equal to 3, that is intercept term plus the number of predictors. Then we use k-means clustering to set the starting center vector $\vec{v}_j$ for each k-means group of anchor points. In addition, the covariance ${\sigma_j}^2$ is computed for each group in Equation (3). Besides, calculate β_j in Equation (2) using a least-squares fitting.

7.3. Iterate RJMCMC sampler until sufficient samples

In the RJMCMC algorithm, we first propose the next state of chain representing a new basis function according to the following criteria. First, draw a uniform random variable $u \sim U(0, 1)$. If $u < 0.33$, then perform the Birth step. In the Birth step, we would add a basis function (ERBF) in the model. Then the corresponding parameters Θ are updated by k-means clustering simultaneously. Recalling Figure 4, for each k-means group, the transformation matrix A_{θ_i,a_i} is computed to add this basis function. If $0.33 \le u \le 0.66$, then perform the Death step. In the Death step, we would lose a basis function. We just select one basis at random and remove it. If $u > 0.66$, then perform the Move step. In the Move step, we choose a basis function from the model at random and reset its position vector to another random point. Then the corresponding parameters are updated.

Then we would compute the marginal log-likelihood of the model and draw k new coefficients β_j. Given n pairs of anchor points $\vec{u}_d$ and corresponding responses $\vec{r}_d$, which are the displacements of anchor points, we would compute the marginal log-likelihood for the creditable change of state as follows:

$$L(\vec{r}\,|\Theta) = -n\log\tau - \frac{1}{2\tau^2}\sum_{d=1}^{n}\left\{\vec{r}_d - \hat{f}(\vec{u}_d)\right\}^2 \tag{15}$$

where $\vec{r}$ is a general representation for the response of regression.

Let X be the responses of basis functions in the matrix form. Y denotes the corresponding responses of the regression model in the matrix form. P represents the matrix form of prior precision p. β represents the matrix form of k coefficients β_j defined in Equation (2). Furthermore, β is obtained from the marginal posterior distribution with posterior mean $\bar{\beta} = \left(X^{\mathrm{T}}X + P\right)^{-1}X^{\mathrm{T}}Y$ and modified standard deviation. Note that the initial standard deviation is drawn from the noise variance τ^2 and modified to be the upper triangle of posterior variance matrix $\left(X^{\mathrm{T}}X + P\right)^{-1}$ obtained by using *Cholesky decomposition*.

Next, consider to accept the proposed change of next state. First, draw a uniform random variable $u \sim U(0, 1)$. If u is less than the ratio of the marginal likelihood of proposed next state to the marginal likelihood of original one, then accept the proposed change to the model and update the state of the *Markov chain*. Otherwise set the next state to be the current state. Then update prior precision p by drawing a random variable from a *Gamma distribution* and is modified by the sum of squares of β_j every 10 iterations. Recalculate the coefficients β_j from the marginal posterior distribution with the updated prior precision p. Furthermore, draw a random variable τ^2 from a *Gamma distribution* for a new noise variance. Given response $\vec{r}$ defined in Equation (15), τ^2 is modified by posterior sum of squares error for the next iteration.

Repeat RJMCMC process and record the number of states. An initial portion of the chain is discarded to ensure stability. If the number of states is greater than the discarded portion, then compute $\hat{f}(.)$ defined in Equation (2) by the recorded parameters of the current model, ERBF term, and affine term. All the simulations are run with a burn-in period of 5000 iterations of RJMCMC followed by 10 000 samples.

REFERENCES

1. Chai J, Hodgins JK. Constraint-based motion optimization using a statistical dynamic model. *ACM Transactions on Graphics* 2007; **26**(3). Article No. 8.
2. Park J, Sandberg IW. Nonlinear approximations using elliptic basis function networks. In *The 32nd Conference on Decision and Control*, 1993; 3700–3705.
3. Jang Y, Botchen RP, Lauser A, Ebert DS, Gaither KP, Ertl T. Enhancing the interactive visualization of procedurally encoded multifield data with ellipsoidal basis functions. *Computer Graphics Forum* 2006; **25**(3): 587–596.
4. Chen M-H, Shao Q-M, Ibrahimj G. Monte Carlo Methods in Bayesian Computation. Springer-Verlag (Springer-Verlag Inc.): New York, USA, 2000.
5. Wolberg G. Image morphing: a survey. *The Visual Computer* 1998; **14**(8): 360–372.
6. Chuang C-H, Tsai S-F, Kuo C-J. Cartoon animation and morphing by using the wavelet curve descriptor. In *International Conference Image Processing*, 1994; 666–670.
7. Shutler JD, Nixon MS. Zernike velocity moments for sequence-based description of moving features. *Image and Vision Computing* 2006; **24**(4): 343–356.
8. Ruprecht D, Müller H. Image warping with scattered data interpolation. *IEEE Computer Graphics and Applications* 1995; **15**(2): 37–43.
9. Baxter W, Anjy K-I. Latent doodle space. *Computer Graphics Forum* 2006; **25**(3): 477–485.
10. DeJuan CN, Bodenheimer B, Re-using traditional animation: methods for semi-automatic segmentation and inbetweening. In *The ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006; 223–232.
11. Hornung A, Dekkers E, Kobbelt L. Character animation from 2D pictures and 3D motion data. *ACM Transactions of Graphics* 2007; **26**(1). Article No. 1.
12. Alexa M, Cohen-or D, Levin D. As-rigid-as-possible shape interpolation. In *Proceedings of ACM SIGGRAPH*, 2000; 157–164.
13. Igarashi T, Moscovich T, Hughes JF. As-rigid-as-possible shape manipulation. *ACM Transactions of Graphics* 2005; **24**(3): 1134–1141.
14. Schaefer S, Mcphail T, Warren J. Image deformation using moving least squares. *ACM Transactions of Graphics* 2006; **25**(3): 533–540.

15. Fan J, Yao Q. Nonlinear Time Series: Nonparametric and Parametric Methods. Springer-Verlag (Springer-Verlag Inc.): New York, USA, 2005.
16. Shumwayr RH, Stoffer DS. Time Series Analysis and Its Applications: With R Examples. Springer-Verlag (Springer-Verlag Inc.): New York, USA, 2006.
17. Bishop CM. Neural Networks for Pattern Recognition. Oxford University Press (Oxford University Press Inc.): New York, USA, 1995.
18. Palm WJ. Modeling, Analysis, and Control of Dynamic Systems. John Wiley & Sons: New York, USA, 1998.
19. Scherbaum K, Sunkel M, Seidel H-P, Blanz V. Prediction of individual non-linear aging trajectories of faces. *Computer Graphics Forum* 2007; **26**(3): 285–294.
20. Brand M, Hertzmann A. Style machines. In *Proceedings of ACM SIGGRAPH*, 2000; 183–192.
21. Galata A, Johnson N, Hogg D. Learning variable length Markov models of behavior. *Computer Vision and Image Understanding*, 2001; **81**(3): 398–413.
22. Young GA, Smith RL. Essentials of Statistical Inference. Cambridge University Press: Cambridge, UK, 2005.
23. Sethian JA. Level Set Methods. Cambridge University Press: Cambridge, UK, 1996.
24. Sethian JA. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press: Cambridge, UK, 1999.
25. Chan TF. Active contours without edges. *IEEE Transactions on Image Processing* 2001; **10**(2): 266–277.
26. Rother C, Kolmogorov V, Blake A. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions of Graphics* 2004; **23**(3): 309–314.
27. Jain AK. Fundamentals of Digital Image Processing. Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1989.
28. Herbrich R. Learning Kernel Classifiers Theory and Algorithms. The MIT Press: 2002.
29. Chen CWS, Mcculloch RE, Tsay RS. A unified approach to estimating and modeling linear and nonlinear time series. *Technical Report*, Gradute School of Business, University of Chicago, IL, 1996.
30. Montgomery DC, Peck EA, Vining GG. Introduction to Linear Regression Analysis. John Wiley & Sons: New Jersey, USA, 2006.
31. Moore AW, Schneider J. Memory-based stochastic optimization. *Advances in Neural Information Processing Systems* 1995; **8**: 1066–1072.

AUTHORS' BIOGRAPHIES

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.

DOI: 10.1002/cav