Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Synthesizing style-preserving cartoons via non-negative style factorization*

Zhang LIANG, Jun XIAO‡, Yue-ting ZHUANG

(*Institute of Artificial Intelligence, School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)

E-mail: {liangzhang, junx, yzhuang}@zju.edu.cn

Received July 8, 2011; Revision accepted Nov. 7, 2011; Crosschecked Feb. 8, 2012

Abstract: We present a complete framework for synthesizing style-preserving 2D cartoons by learning from traditional Chinese cartoons. In contrast to reusing-based approaches which rely on rearranging or retrieving existing cartoon sequences, we aim to generate stylized cartoons with the idea of style factorization. Specifically, starting with 2D skeleton features of cartoon characters extracted by an improved rotoscoping system, we present a non-negative style factorization (NNSF) algorithm to obtain style basis and weights and simultaneously preserve class separability. Thus, factorized style basis can be combined with heterogeneous weights to re-synthesize style-preserving features, and then these features are used as the driving source in the character reshaping process via our proposed subkey-driving strategy. Extensive experiments and examples demonstrate the effectiveness of the proposed framework.

Key words: Character cartoon, Machine learning, Cartoon synthesis
doi:10.1631/jzus.C1100202 **Document code:** A **CLC number:** TP391

1 Introduction

Some famous Chinese characters in traditional 2D cartoon films, drawn manually by animators in the period of 1950–1980s in the last century, were carefully designed and rendered to refine the appearance. These well-known characters influenced several generations of children and adults by their impressive characteristics, which are called 'styles' (Chenney *et al*., 2002; Wang *et al*., 2006; Lau *et al*., 2009). Typically, the style factor always remains stable among one character's various movements. However, it varies from one character to another. As defined by Bregler *et al*. (2002), all the characteristics embedded in the visual expression of a cartoon character fall into the group of the visual or the motion style. Most character cartoons have both highly exaggerated drawings and highly exaggerated motions.

Traditional Chinese cartoon characters share some common attributes inherited from a special drawing skill called 'Chinese realistic painting'. The main expressions of such skill are that, the shape outlines are smooth and the characters always exhibit human-like patterns. Given these observations, we aim to design a framework for synthesizing style-preserving Chinese cartoons with the idea of 'style translation'. Style translation was defined by Tenenbaum and Freeman (2000), and in our case it is creating new cartoons with character A's style while reusing motions belonging to character B. In the following sections, characters are short for traditional Chinese cartoon characters.

Then, there follows the problem of how to quantitatively represent the style in a formalized process. Character styles mostly depend on animators' personal experiences, e.g., how the frame is rendered, the amount of exaggeration, and how the tone goes with the character's mood. This experience-

‡ Corresponding author

* Project supported by the National Basic Research Program (973) of China (No. 2012CB316400), the National Natural Science Foundation of China (No. 60903134), and the Natural Science Foundation of Zhejiang Province, China (No. Y1101129)

dependent work makes it difficult to express the style explicitly. Inspired by research conducted in the machine learning community, we consider the style as a subspace and try to reconstruct the style space using basis vectors.

In this paper, we aim to design a learning based framework with a workflow as shown in Fig. 1, which synthesizes new style-preserving cartoons by cartoon resources reuse. Inspired by works focusing on sparse representation (Hoyer, 2002; Aharon *et al*., 2006), provided with a database containing 2D skeleton features derived by means of our improved rotoscoping system from characters' images, we present the non-negative style factorization (NNSF) algorithm to obtain a style basis and simultaneously preserve class separability in a content-based understanding layer. As the visual style usually combines with the motion style, we consider using both together as an unsplit one. Thus, our algorithm separates the skeleton features of 2D characters into style basis shared among characters as the style-independent factor, and corresponding weights controlling the variance of character's personality as the style-dependent factor. Then, style translation can be undertaken based on the combination of heterogeneous style basis and weights to re-synthesize new skeleton features. Finally, our work extends character animation techniques and there are three key contributions:

1. A 2D skeleton feature extraction method is proposed based on the existing rotoscoping system, in which the original optimization process is improved for key point determination.

2. We present a non-negative style factorization algorithm to solve the style presentation and translation problem based on skeleton features.

3. To preserve Chinese cartoon characteristics, a subkey-driving strategy based on a flexible limb model is presented as a hybrid method to handle the reshaping task.

2 Related work

2.1 Motion feature extraction

Motion feature extraction provides scalable methods to capture motions of characters in various scenes. There are many research works (Kuo *et al*., 2008; Rogez *et al*., 2008) dealing with recovering a 2D human pose from monocular video in the computer vision community. Usually, these methods firstly segment body parts from the background, and then discover the spatio-temporal relationships between the parts using explicit probabilistic models. However, the training processes of these models are time-consuming and demand much manual work.

As discussed by Yang *et al*. (2009), the outlines of characters are easily extracted as the sketches are painted before colorization in cartoon drawing. To this end, we favor contour-based tracking methods here. There is a vast literature on tracking objects based on contour detection in the vision literature. A classic approach is the Snakes system (Jonker and Volgenant, 1987) to find curves in an image, which was then adapted by Hoch and Litwinowicz (1996) for tracking. Freifeld *et al.* (2010) defined a 2D articulated contour person (CP) which is taken from examples to represent the naked body shape. The CP model realistically represents the human form but does not model clothing. To this end, Guan *et al*. (2010) presented the dressed contour person (DCP) to create a low-dimensional dressed model by computing how clothing deviates from the naked body. Wang and Li (2002) proposed a method to capture cartoon motion by shape matching. They modeled cartoon motions as a global affine transformation and local non-affine deformation, where thin-plate splines (TPS) are introduced.

2.2 Style translation

Providing users with efficient and intuitive style capturing and editing tools has been a long-standing

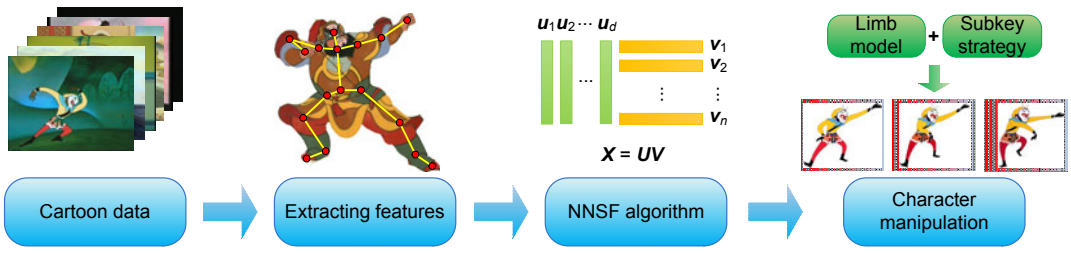

Fig. 1 The outline of the proposed non-negative style factorization (NNSF) algorithm

challenging issue, and has been the focus of much research (Hsu *et al*., 2005; Torresani *et al*., 2007; Lau *et al*., 2009). Brand and Hertzmann (2000) applied the hidden Markov model (HMM) to generate new highly nonlinear and nontrivial behaviors such as choreographic sequences in a broad range of styles. Hsu *et al*. (2005) established a linear time-invariant (LTI) model to describe the input and output styles, and proposed an iterative motion warping (IMW) algorithm to align stylistically different motions. In the above mentioned works, various 3D motion styles were clearly modeled and transferred, such as locomotion and complex dancing styles. Bregler *et al*. (2002) constructed a bridge between realistic capture data and expressive cartoon animation. They used a combination of affine transformation and key-shape interpolation, where the non-rigid shape changes in source cartoon data are captured and then retargeted into different output layers. However, it still needs skilled artists to define the corresponding output key-shapes manually. Li *et al*. (2003) proposed a hybrid framework to make rigid 3D capture animation more expressive. The hybrid modification consists of skeletal and geometric alteration, which can incorporate stylization in 3D animation.

Using factorization techniques to decouple the interaction between various factors has received extensive study in recent years. Tenenbaum and Freeman (2000) detailed a classical approach separating content and style of facial motion using the bilinear models (BM). The constraint-based Gaussian process model (CBGPM) was presented by Ma *et al*. (2009) for learning facial editing styles. Both methods intensively study non-intrinsic factors such as illuminating or editing influences. Wang *et al.* (2007) proposed a multifactor Gaussian process model (MGPM) to create models for human locomotion data, which can be viewed as a special class of the Gaussian process latent variable model (GPLVM). Lau *et al*. (2009) detailed kernel-based nonlinear factorization and took human identity and facial expression as two influencing factors of facial appearance. In their experiments, the face recognition yields competitive results under variable lighting conditions.

2.3 Character deformation

When a driving source is provided, deforming characters in a 2D image has received much interest (Wang *et al*., 2006; Yan *et al*., 2008). Alexa *et al*. (2000) presented the idea that the shape deformation in an image should be as rigid as possible to minimize the impacts of local scaling and shearing. Igarashi *et al*. (2005) triangulated the character and minimized the distortion of these triangles in the deformation process by solving a linear system of equations. In Schaefer *et al*. (2006), another rigid transformation method was proposed based on moving least squares. Both methods focus on specifying deformation by user-specified handles. Weng *et al*. (2006) presented a 2D shape deformation method based on nonlinear least squares optimization. They designed a non-quadratic energy function to preserve the properties of both boundary spline and shape interior iteratively.

At a higher layer, Hornung *et al*. (2007) presented a semi-automatic approach for the animation of driving 2D characters that is based on image warping under the control of projected 3D motion capture data. Zhou *et al*. (2010) altered body attributes in photographs and changed the shape of a person in a semi-automatic way by combining monocular pose inference with the morphable model. However, the aforementioned methods depend on the rigid morphable model as the intermedia, which conflicts with the property of non-rigid exaggerative deformation in 2D cartoons.

3 Two-dimensional skeleton feature extraction

In this section, we present the details of skeleton feature extraction from cartoon images. Inspired by 3D motion data (Pullen and Bregler, 2002; Moeslund *et al*., 2006), we try to capture 2D skeleton features in the time domain. Without loss of generality, each character posture can be represented by a feature vector $\boldsymbol{X}_i=[x_i^1, y_i^1, x_i^2, y_i^2, \ldots, x_i^{17}, y_i^{17}]$, where x_i^j and y_i^j denote the coordinates of the jth key point in image I_i.

However, the key points coordinates $\boldsymbol{X}_i$ cannot be obtained directly via optical or dynamic marker tracking techniques as for 3D motion data. They can be estimated only approximately based on interactive information. As mentioned above, the character outlines detected using our improved rotoscoping system can be used to determine key points. In the original rotoscoping system presented by Agarwala *et al*. (2004), the widespread and non-rigid deformations in

traditional Chinese cartoons make the original method fail to lock onto some limb outlines in the iteration process. For adaption, we add a sketch constraint term to the objective with extended bidirectional sampling.

In Fig. 2, for the curve $\boldsymbol{c}_t(\boldsymbol{s})$ being tracked, t is the temporal index over frames and $\boldsymbol{s}$ is the spatial parameter of the curve. For $\boldsymbol{s}_i$ spreading across the curve, $\boldsymbol{n}_t(\boldsymbol{s}_{ij})$ is the jth and one of the unit normals that are located on two sides of $\boldsymbol{s}_i$ at time t. All these normals on curve $\boldsymbol{c}_t(\boldsymbol{s})$ in frame t compare with those in the next frame t+1, and the new constraint term is defined as follows to minimize the displacement:

$$E_I = \sum_{i,t,\alpha} \| I_t(\boldsymbol{c}_t(\boldsymbol{s}_i) + \sum_j \alpha \boldsymbol{n}_t(\boldsymbol{s}_{ij})) - I_{t+1}(\boldsymbol{c}_{t+1}(\boldsymbol{s}_i) + \sum_j \alpha \boldsymbol{n}_{t+1}(\boldsymbol{s}_{ij})) \|^2, \quad (1)$$

where $I_t(\boldsymbol{p})$ is the RGB color vector of image I_t at point $\boldsymbol{p}$, and α varies over a user-defined window. This term extends the searching range to the whole ribbon, for full scale sampling. Furthermore, in a different way from the original system, a color penalty is weighted here in the overall objective function to bound the searching space. It can be explained by the color configuration of each character in a sequence remaining unchanged for relieving drawing manual burdens.

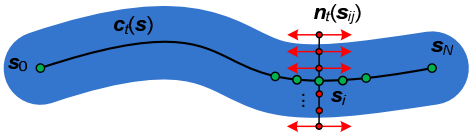

Fig. 2 A thin ribbon around a curve $\boldsymbol{c}_t(\boldsymbol{s})$ which forms the tracking window

A discrete point $\boldsymbol{s}_i \in [\boldsymbol{s}_0, \boldsymbol{s}_N]$ runs along the curve with bidirectional samples on both sides

After outline tracking, we implement a simple way to determine the key points accordingly. Let us define calibration points $\{\boldsymbol{p}_i^1, \boldsymbol{p}_i^2, \boldsymbol{p}_i^3, \boldsymbol{p}_i^4\}$ at both ends of curves $\boldsymbol{c}_t(\boldsymbol{s})$ and $\boldsymbol{c}_t(\boldsymbol{s}')$ belonging to the ith limb (Fig. 3). They are manually annotated in initial and end frames in a sequence. Then, it is assumed that all these points obtain one-to-one correspondence in the remaining frames as they stay relatively stable even under non-rigid deformations. Following this assumption, coordinates of key points $\boldsymbol{p}_j$ and $\boldsymbol{p}_{j+1}$ can be approximately estimated as follows:

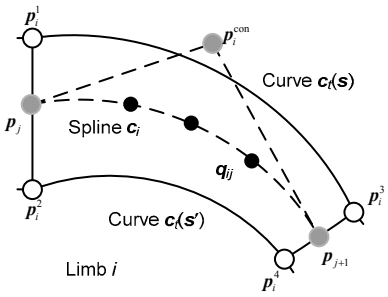

Fig. 3 Parameterization of limb $\boldsymbol{i}$

$$\boldsymbol{p}_j = \omega_j \boldsymbol{p}_i^1 + (1-\omega_j)\boldsymbol{p}_i^2, \quad (2)$$

$$\boldsymbol{p}_{j+1} = \omega_{j+1} \boldsymbol{p}_i^3 + (1-\omega_{j+1})\boldsymbol{p}_i^4, \quad (3)$$

where weights ω_j and ω_{j+1} control the relative positions between key points and calibration points.

These raw coordinates of key points need to be unified because different characters vary in body proportions. For the purpose of reuse, we define a template proportion with fixed ratios of limb and torso lengths. The nonunified data will be mapped onto the template proportion by rectifying data locally. For coordinate matrix $\boldsymbol{P}_i$ of all key points estimated from Eqs. (2) and (3) in each cartoon image I_i, we can obtain a unified feature $\boldsymbol{X}_i$ as follows:

$$\boldsymbol{X}_i = \boldsymbol{W}_{\text{char}} \boldsymbol{P}_i, \quad (4)$$

where $\boldsymbol{W}_{\text{char}}$ is a weight matrix eliminating the proportion diversity between the template and the local character, and may vary according to character changing. Then we have constructed our database of 2D skeleton features for the following analysis.

4 Non-negative style factorization

Now we detail our proposed style translation algorithm, namely non-negative style factorization (NNSF). Define the matrix $\boldsymbol{X}=[\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n]^{\mathrm{T}} \in \mathbb{R}^{n \times d}$, where each image I_i is represented by its feature vector $\boldsymbol{X}_i$, d is the feature dimensionality, and n is the number of samples ($n>>d$ in this study). Note that, in our database, each sample corresponds to the feature vector of a character posture obtained from an image, where the posture may belong to any kind of character type and motion class; e.g., a character posture reflects that the character Wukong Sun takes the motion

of beating towards a monster. Furthermore, all these samples are classified manually into 15 common motion classes according to the semantic meanings of motions; e.g., any samples that contain postures belonging to motion 'walk' are classified into class 'walk'. Obviously, a class may include samples described by more than one character, which can be regarded as insensitive to character types.

We abandon traditional mechanism of 'content' and 'style' split, which makes our work intrinsically different from previous ones. As is known, traditional content-style separations result in larger covariance between each motion content $\{c_i\}$ and mean content $\tilde{c}$ in cartoon applications, and the resulting covariance will interfere with the next style translation dramatically.

Let us describe the matrix $\boldsymbol{X}$ using linear decompositions as $\boldsymbol{U}$ and $\boldsymbol{V}$. The matrix $\boldsymbol{U}=[\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n]^{\mathrm{T}}\in\mathbb{R}^{n\times d}$ contains as its rows the style basis vectors of the decomposition. The matrix $\boldsymbol{V}=[\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d]\in\mathbb{R}^{d\times d}$ contains the corresponding hidden components that give the contribution of each style basis vector in the input vectors. As studied in linear sparse coding, the hidden components reveal probability densities which are highly peaked at zero. Therefore, our 2D motion features can be well represented using a few non-zero hidden coefficients as sparse issues. Combining the goal of a small reconstruction error with that of sparseness, the objective function can be written as

$$C(\boldsymbol{U},\boldsymbol{V})=\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}\|_{\mathrm{F}}^2+\alpha\sum_{ij}f(V_{ij}), \tag{5}$$

with the non-negative constraints $\forall ij$: $U_{ij}\geq 0$, $V_{ij}\geq 0$, and $\|\cdot\|_{\mathrm{F}}$ denotes Frobenius norm. The last term at the right-hand side of the objective function defines how sparseness is measured, and α controls the trade-off between sparseness and accurate reconstruction.

Note that we choose a linear activation penalty (i.e., $f(\boldsymbol{V})=\boldsymbol{V}$) to measure sparseness in Eq. (5), as the solution process demands f to be a strictly increasing function. This typical choice is primarily motivated by the fact that this makes the objective quadratic in $\boldsymbol{V}$. The full convergence proof can be found in Hoyer (2002).

However, for a content-based understanding requirement, providing only an exact reconstruction is insufficient here. It is supposed that after projecting weights $\boldsymbol{V}$ onto the directions $\boldsymbol{W}$, the resulted lower dimensional samples $\boldsymbol{V}'$ should preserve original class separability. Thus, Eq. (5) will add a class-constraint term $\|\boldsymbol{V}'-\boldsymbol{Y}\|_{\mathrm{F}}^2$ which binds the projected results to class label ground truth denoted as the matrix $\boldsymbol{Y}=[\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n]^{\mathrm{T}}\in\{0, 1\}^{n\times r}$, where r is the size of the class set. In matrix $\boldsymbol{Y}$, if $\boldsymbol{X}_i$ belongs to the lth class, then $Y_{il}=1$; otherwise, $Y_{il}=0$. We assign each $\boldsymbol{X}_i$ in samples with an identical label according to the semantic meanings of the motion, and samples with the same labels belong to one class.

Furthermore, we must define the modeling of the matrix $\boldsymbol{V}$ to simplify the computation. It can be inferred that $\boldsymbol{V}$ containing zero or closely near zero terms is considered to be a sparse matrix, which means re-modeling $\boldsymbol{V}$ using the eigenspace of the samples. Let $\boldsymbol{M}=\frac{1}{d}\sum_{i,j=1}^{d}\boldsymbol{v}_i\boldsymbol{v}_j^{\mathrm{T}}$ be the pairwise correlation matrix, and $\{a_i\}_{i=1}^{r}$ be the top r ($r\leq d$) eigenvectors of matrix $\boldsymbol{M}$. Then the reconstructed $\tilde{\boldsymbol{V}}$ is assumed to be a linear combination of the top r eigenvectors:

$$\tilde{\boldsymbol{V}}=\sum_{i=1}^{r}\psi_i\boldsymbol{a}_i\boldsymbol{a}_i^{\mathrm{T}}, \tag{6}$$

where $\psi_1, \psi_2, \ldots, \psi_r$ are the non-negative weights in the linear combination. The set of optimal $\{\psi_i^*\}$ is determined experimentally, which we do not detail here for simplification.

Then, we propose to combine style factorization and class preservation into the non-negative style factorization:

Definition 1 Non-negative style factorization (NNSF) of a non-negative 2D skeleton feature matrix $\boldsymbol{X}$ (i.e., $\forall ij$, $X_{ij}\geq 0$) is given by the minimization of

$$C(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})=\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}\|_{\mathrm{F}}^2+\alpha\sum_{ij}\boldsymbol{V}_{ij}+\beta\|\boldsymbol{W}\tilde{\boldsymbol{V}}-\boldsymbol{Y}\|_{\mathrm{F}}^2, \tag{7}$$

subject to the constraints $\forall ij$: $U_{ij}\geq 0$, $V_{ij}\geq 0$, and $\forall i$: $\|\boldsymbol{u}_i\|=1$. It is also assumed that $\alpha\geq 0$, $\beta\geq 0$.

The minimization of Eq. (7) can be divided as two independent parts: optimizing the objective with respect to style basis $\boldsymbol{U}$ and weights $\boldsymbol{V}$ successively under non-negative constraints, and then yielding optimal $\boldsymbol{W}^*$.

For first optimizing $\boldsymbol{U}$ and $\boldsymbol{V}$, we take different strategies due to the characteristics of convex. Firstly, we optimize $\boldsymbol{V}$ under the given basis $\boldsymbol{U}$. Since the objective function is quadratic with respect to $\boldsymbol{V}$ and $\boldsymbol{V}$ is convex, the optimal $\boldsymbol{V}$ can be obtained based on the multiplicative algorithm presented by Hoyer (2002), by which the global minimum can be found iteratively by following the update rules. Meanwhile, keeping $\boldsymbol{V}$ fixed, minimizing with respect to $\boldsymbol{U}$ can follow the instructions detailed by Lee and Seung (2001), using the idea of the multiplicative update rule. To obtain optimal $\boldsymbol{U}^*$ and $\boldsymbol{V}^*$, we can update both alternately and this yields the following algorithm:

Algorithm 1 Non-negative style factorization

Initialization: Initialize $\boldsymbol{U}^0$ and $\boldsymbol{V}^0$ according to non-negative definitions randomly, and rescale each $\|\boldsymbol{u}_i\|=1$.

Iterations: Iterate lines 1–5 until both $\boldsymbol{U}^t$ and $\boldsymbol{V}^t$ come into convergence:

1 $\boldsymbol{U}' = \boldsymbol{U}^t - \xi(\boldsymbol{U}^t\boldsymbol{V}^t - \boldsymbol{X})\boldsymbol{V}'^{\mathrm{T}}$

2 If any $U'_{ij} < 0$, then set all U'_{ij} to 0

3 Rescale $\|\boldsymbol{u}'_i\| = 1$, and then set $\boldsymbol{U}^{t+1}=\boldsymbol{U}'$

4 $\boldsymbol{V}^{t+1} = \boldsymbol{V}^t \cdot *((\boldsymbol{U}^{t+1})^{\mathrm{T}}\boldsymbol{X}) \cdot /((\boldsymbol{U}^{t+1})^{\mathrm{T}}\boldsymbol{U}^{t+1}\boldsymbol{V}^t + \xi)$

5 Increase t by 1

// Here '.*' and './' denote elementwise multiplication and
// division, respectively. The deduction and proof can be
// found in Hoyer (2002)

Then we can set the derivative of Eq. (7) with respect to $\boldsymbol{W}$ to zero and have

$$\boldsymbol{W}^* = \boldsymbol{Y}\tilde{\boldsymbol{V}}^{*\mathrm{T}}(\tilde{\boldsymbol{V}}^*\tilde{\boldsymbol{V}}^{*\mathrm{T}})^{-1}. \tag{8}$$

Let us provide an interpretive example to explain how the algorithm synthesizes style-preserving motion features. For example, we want to synthesize motion features $\tilde{\boldsymbol{X}}_{\mathrm{ws}}$ belonging to class 'jump' in character Wukong Sun's style, while the database contains only character Nezha's motion features $\boldsymbol{X}_{\mathrm{nz}}$ belonging to the same class 'jump' and Wukong Sun's non-jump motion features $\boldsymbol{X}_{\mathrm{ws}}$. Note that both are classical Chinese cartoon characters.

After following the iterative algorithm for NNSF decomposition, the factorized $\boldsymbol{U}_{\mathrm{nz}}$, $\boldsymbol{U}_{\mathrm{ws}}$, $\boldsymbol{V}_{\mathrm{nz}}$, and $\boldsymbol{V}_{\mathrm{ws}}$ correspond to the style basis and weights of features $\boldsymbol{X}_{\mathrm{nz}}$ and $\boldsymbol{X}_{\mathrm{ws}}$, respectively. Then the target features $\tilde{\boldsymbol{X}}_{\mathrm{ws}}$ can be achieved by recombining the heterogeneous basis $\boldsymbol{U}_{\mathrm{nz}}$ belonging to Nezha and weights $\boldsymbol{V}_{\mathrm{ws}}$ belonging to Wukong Sun. In the recombination, the change of weights from $\boldsymbol{V}_{\mathrm{nz}}$ to $\boldsymbol{V}_{\mathrm{ws}}$ brings style diversities between Nezha and Wukong Sun to the original basis $\boldsymbol{U}_{\mathrm{nz}}$, and incorporates Wukong Sun's special style into the original jump motion. However, the primitive characteristics of jump motion will remain stable, regardless of the character type.

5 Character shape manipulation

In this section, we are trying to combine the self-defined limb model and subkey-driving strategy as a hybrid mechanism to make the reshaping results satisfy the characteristic requirements of Chinese realistic painting.

5.1 Limb model and interpolation

As Chinese realistic painting always values smooth and aesthetic outlines of characters, we design a special limb model for preserving the characteristics of the original outlines after shape manipulation.

As shown in Fig. 3, $\boldsymbol{\varphi}_i=(\boldsymbol{p}_j, \boldsymbol{p}_{j+1}, \boldsymbol{p}_i^{\mathrm{con}})$ corresponds to the coordinates of two key points and the quadratic curve control point in the ith limb as we apply quadratic Bëzier spline $\boldsymbol{B}_i(\lambda)$ to model the spline $\boldsymbol{c}_i$, which is approximately parallel with two-sided limb contours:

$$\boldsymbol{B}_i(\lambda) = (1-\lambda)^2\boldsymbol{p}_j + 2(1-\lambda)\lambda\boldsymbol{p}_i^{\mathrm{con}} + \lambda^2\boldsymbol{p}_j, \tag{9}$$

where λ is the control parameter. As presented by Hornung *et al*. (2007), $(\boldsymbol{p}_j, \boldsymbol{p}_{j+1}, \boldsymbol{p}_i^{\mathrm{con}})$ can be redefined with relative coordinates $\boldsymbol{\omega}_i = (\omega_i^{\mathrm{con\_x}}, \omega_i^{\mathrm{con\_y}})$ as

$$\boldsymbol{p}_i^{\mathrm{con}} = \boldsymbol{p}_j + \omega_i^{\mathrm{con\_x}}(\boldsymbol{p}_{j+1} - \boldsymbol{p}_j) + \omega_i^{\mathrm{con\_y}}\boldsymbol{R}_{\mathrm{quar}}(\boldsymbol{p}_{j+1} - \boldsymbol{p}_j), \tag{10}$$

where the matrix $\boldsymbol{R}_{\mathrm{quar}}$ denotes the counterclockwise rotation by 90°. Then the limb model can be written as $\boldsymbol{\varphi}_i=(\boldsymbol{p}_j, \boldsymbol{p}_{j+1}, \boldsymbol{\omega}_i)$. Until now, we have obtained the matrix $\boldsymbol{\phi}_t=[\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_m]^{\mathrm{T}}$ denoting the pose parameter in a frame, where m is the number of character limb models.

In practice, it is impossible to automatically fit the parameters via video tracking and regression

techniques. Therefore, interactively determining each $\boldsymbol{\phi}_i$ in a sequence $\Phi=\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_n\}$ seems reasonable. As manual work often creates a tiresome burden, we only annotate keyframes as a tradeoff, and extend the results to all frames by interpolation. Based on the previous rotoscoping system, users need only to rectify Bëzier splines predicted by our system in keyframes, and corresponding parameters will be calculated inversely and interpolated.

5.2 Subkey points tracking

Similar to the shape manipulation method addressed by Igarashi *et al*. (2005), we adopt a two-step linear mesh deformation algorithm. We found that, although motion features $\tilde{\boldsymbol{X}}$ achieved by the NNSF algorithm are set as constrained handles, the deformation results are not guaranteed to retain the aesthetic properties of the original drawings. To this end, except for traditional key points, we introduce 'subkey points' inspired by Kwon and Lee (2008). The subkey points (black points in Fig. 3) are distributed along Bëzier spline $\boldsymbol{c}_i$ to provide assisted morphing constraints.

For $\boldsymbol{q}_{ij}$ in Fig. 3, which denotes the jth subkey point on limb i, provided with triangulated character shape, the initial coordinates of $\boldsymbol{q}_{ij}^0$ can be determined as

$$\boldsymbol{q}_{ij}^0 = \arg\min_k \left\| \boldsymbol{V}(k) - \boldsymbol{B}_i(\lambda_j) \right\|^2, \tag{11}$$

where $\boldsymbol{B}_i(\lambda_j)$ calculated using Eq. (9) denotes a point on the spline, and $\boldsymbol{V}(k)$ is the coordinate function of the kth vertex in the character mesh. Eq. (11) assigns the coordinates of $\boldsymbol{q}_{ij}^0$ with the nearest one among all mesh vertices near the spline point $\boldsymbol{B}_i(\lambda_j)$.

Accordingly, a new estimation $\boldsymbol{q}_{ij}^{t+1}$ at frame t+1 can be updated approximately based on $\boldsymbol{q}_{ij}^t$ at frame t according to the following rule:

$$\boldsymbol{q}_{ij}^{t+1} = \boldsymbol{q}_{ij}^t + \frac{\left\| \boldsymbol{p}_j - \boldsymbol{q}_{ij}^t \right\|}{\left\| \boldsymbol{p}_j - \boldsymbol{p}_{j+1} \right\|} \Delta \boldsymbol{p}_j + \frac{\left\| \boldsymbol{p}_{j+1} - \boldsymbol{q}_{ij}^t \right\|}{\left\| \boldsymbol{p}_j - \boldsymbol{p}_{j+1} \right\|} \Delta \boldsymbol{p}_{j+1}, \tag{12}$$

where $\Delta \boldsymbol{p}_j$ and $\Delta \boldsymbol{p}_{j+1}$ are driving increments of key points between adjacent frames, and their varying weights reflect the geometric relationships between $\boldsymbol{q}_{ij}$ and nearby key points in each frame.

5.3 Fitting for subkey points

The limb model presented in Section 5.1 guarantees the smoothness of outlines. However, the generated spline might be insufficient to express the details. Conversely, the subkey coordinates are determined accurately and locally based on nearby key points, and they may suffer from error accumulation due to insufficient global constraints. Here we construct a hybrid method by combining techniques detailed in Sections 5.1 and 5.2. Specifically, we use the splines to constrain incremented coordinates of subkey points for both accurate and smooth results in the temporal domain.

For subkey point $\boldsymbol{q}_{ij}^t$, we have the following objective function:

$$\lambda^* = \arg\min_\lambda \left\| \boldsymbol{B}_i^t(\lambda) - \boldsymbol{q}_{ij}^t \right\|^2, \tag{13}$$

by which the optimal λ^* is obtained. The objective function is trying to find a point on $\boldsymbol{B}_i(\lambda)$ which is closest to $\boldsymbol{q}_{ij}$ at the tth frame. Function $\boldsymbol{B}_i$ is in the quadratic form, making the objective in the right-hand side of Eq. (13) in the 4th order with respect to λ. Therefore, the optimization of Eq. (13) could be performed analytically by setting the derivative with respect to λ to zero:

$$\frac{\partial \left\| \boldsymbol{B}_i^t(\lambda) - \boldsymbol{q}_{ij}^t \right\|^2}{\partial \lambda} = 0. \tag{14}$$

It is easy to justify that Eq. (14) is in the cubic form with respect to λ. We can easily derive the candidate λ values from Eq. (14) and select the optimal one with the closest distance for the final coordinates of subkey point $\boldsymbol{q}_{ij}^t$.

6 Experiments and examples

To verify the effectiveness of the proposed framework extensively, we have collected 2455 traditional Chinese 2D cartoon images containing character postures of various motions from cartoon videos of nine classical characters, e.g., Wukong Sun and Nezha. Then, all these cartoon images were processed using the feature extraction method described in

Section 3 to obtain feature samples for further experiments. We applied each experiment with a 10-fold cross validation to estimate the average accuracy and possible variance.

Considering that subjective evaluations (e.g., high dynamic range) have been widely used in many practical applications, we invited 60 volunteers who are familiar with 2D cartoons, 30 males and 30 females, aged from 20 to 45. Some of them were professional cartoonists and the others amateurs. They were asked to determine whether each synthesized style-preserving result is good or bad, and their judgments were calculated to evaluate the performance.

Our experiments focused on three issues: skeleton feature extraction, NNSF algorithm evaluation, and the reshaping result. First, as detailed in Section 3, we need to evaluate the effectiveness of our improved rotoscoping system in comparison with the original one in terms of subjective measures. One acceptable approach designed by Agarwala *et al*. (2004), as shown in Fig. 4a, is to count the number of user-edited control points, compared with the total number of control points. For achieving acceptable results, 20 professional and 20 non-professional cartoonists were assigned to edit the control points for 10 rounds to average the mean ratios for both methods on the same tracking videos. Note that the ratios slightly fluctuated among images containing different characters due to the variation of tracking difficulties. Furthermore, dramatically transforming parts always resulted in dense probabilities of tracking failures in comparison with smooth ones, neglect of characteristics. It is obvious that our method can save more manual interactions than the previous one due to lower edited ratios.

To illustrate the comparison intuitively, a comparative example on a clip of cartoon directed by Wukong Sun from *Monkey Makes Havoc in Heaven* is shown in Fig. 5. Obviously, the original method failed to track the details of the limb distal, and these

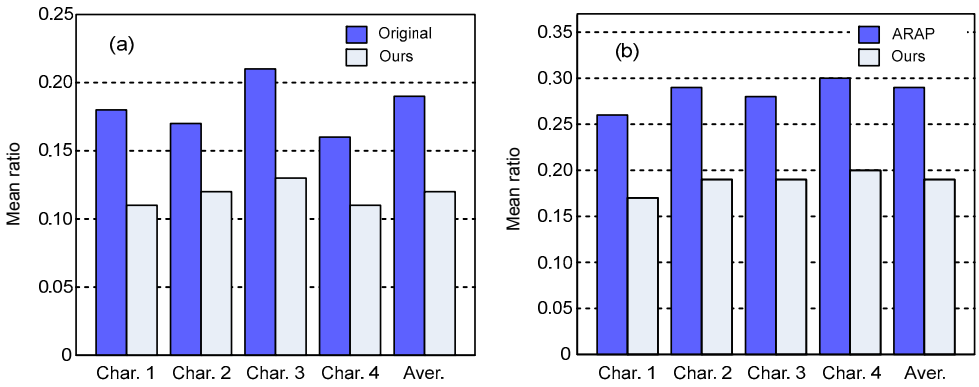

Fig. 4 Mean ratio of the number of user-edited control points to the total number of points in the whole sequence (a) and the ratio of the number of driving failures to the number of the synthesized frames (b)
The horizontal axis denotes data extracted from four selective characters and the average for this experiment

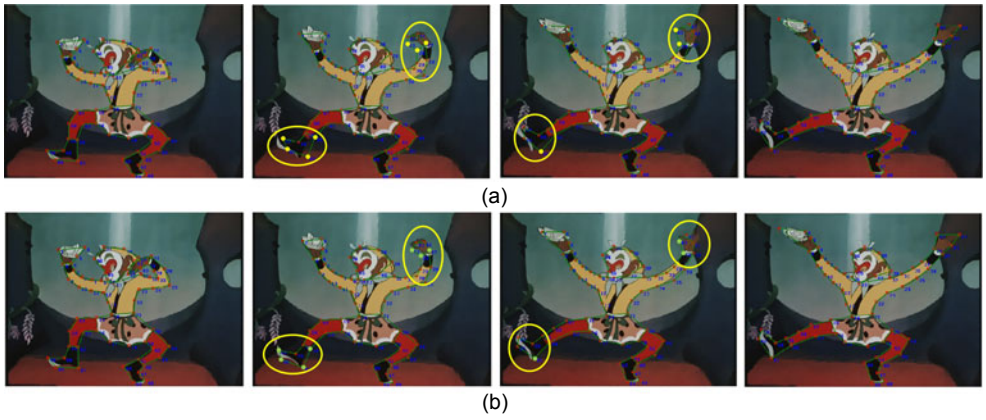

Fig. 5 Tracking results derived by the approach proposed by Agarwala *et al.* (2004) (a) and our approach (b)
The tracked contour of the character is outlined by mark points and their linkages for both methods

failures are emphasized with light markers (Fig. 5a). Meanwhile, our improved method achieved better performance (Fig. 5b).

Before the evaluation of the NNSF algorithm, the dimensionality reduction results of ISOMAP (Tenenbaum *et al*., 2000) on weights matrix $\boldsymbol{V}$ are given (Fig. 6). It can be inferred that the intrinsic dimensionality of $\boldsymbol{V}$'s embedding space was under 10, which coincided with the character types.

Then, we compared the proposed NNSF algorithm against two established approaches and a baseline. The first is the constraint-based Gaussian process model (CBGPM) presented by Ma *et al*. (2009) for facial animation editing. The second is the multifactor Gaussian process model (MGPM) detailed by Wang *et al*. (2007) for stylized human locomotion synthesis. The baseline was a bilinear model (BM) detailed by Tenenbaum and Freeman (2000) for texture and face recognition. All these approaches have been demonstrated as effective ways to deal with style-content separation problems in the cartoon community. The subjective results on our collected data are shown in Fig. 7. It was observed that: (1) Our algorithm outperformed the others after convergence, due to the combination of style factorization and class maintenance. (2) For the few ranks (ranks 1 to 6), the precision of our algorithm was slightly worse than that of BM. Only after rank 6 did our algorithm start to outperform BM in rapid increments. Since our algorithm tries to simultaneously satisfy factorization and class separability, it may sacrifice the lower reconstruction precision for improved class separability. This tradeoff leads to NNSF's poor performance for the few ranks. (3) After convergence, both Gaussian based approaches performed better than BM, and their results remained close.

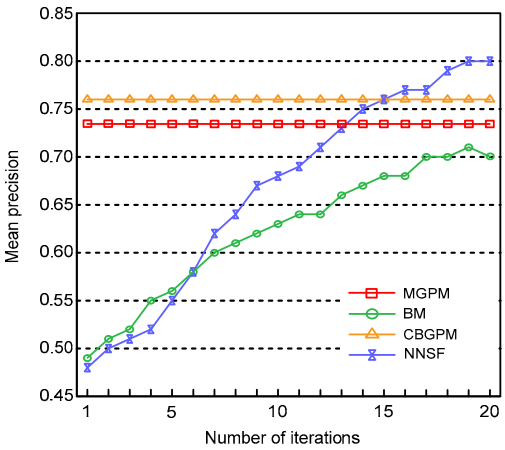

Fig. 7 Subjective results for collected data

MGPM: multifactor Gaussian process model; BM: bilinear model; CBGPM: constraint-based Gaussian process model; NNSF: non-negative style factorization

Finally, to demonstrate the performance of our hybrid subkey-driving strategy, Fig. 8 shows the comparison between the as-rigid-as-possible (ARAP) deformation approach and ours. The deformed results marked by rectangles show that our method yielded thinner and smoother limbs in comparison with ARAP. The smoothness conformed more to the original drawing style. Furthermore, we applied subjective evaluations detailed above to obtain a statistical report of failures as shown in Fig. 4b.

At last, we showed some final examples. In the style factorization process, we combined style basis matrix $\tilde{\boldsymbol{U}}_{\mathrm{nz}}$ factorized from a clip directed by character Nezha from *Nezha Conquers The Dragon King* (Fig. 9a) with heterogeneous style weights $\tilde{\boldsymbol{V}}_{\mathrm{ws}}$ belonging to character Wukong Sun to yield new motion features $\tilde{\boldsymbol{X}}_{\mathrm{ws}}$, which exhibit both Nezha's content and Wukong Sun's style. Before reshaping, a manually

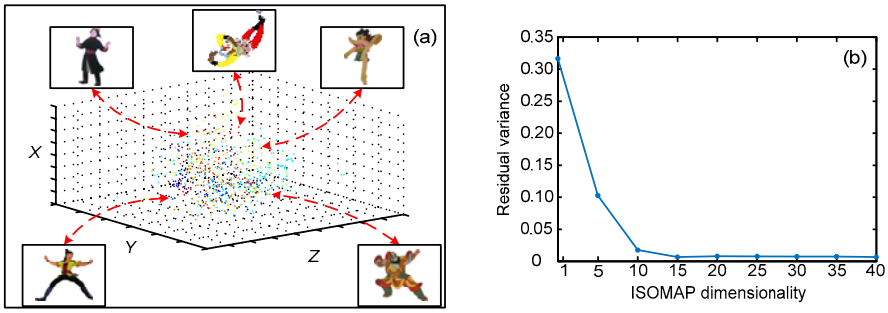

Fig. 6 ISOMAP training: (a) the embedding of character features on a 3D manifold; (b) the dimensionality residual of dimension from 1 to 40

drawn character shape should be provided, and its pose should be similar to that of the source character, i.e., Nezha in this example. According to the reshaping results (Fig. 9b), smooth style was preserved effectively in the existing cartoons of Wukong Sun.

Additionally, the inherent 2D-3D ambiguity will impact some motions, especially the ones involving overlapping parts directed by body parts, and this problem needs to be carefully addressed. Here we dynamically assigned depth values to overlapping parts using the sparse depth (in)equalities algorithm detailed by Sýkora *et al*. (2010) (Fig. 10). Wukong Sun's right arm was moving across the torso with overlapping regions detection and depths being adjusted when necessary. The depth order can be modified quickly to enforce the desired visibility (Fig. 10). The arm and body reflected right geometric relationship throughout the deformation process.

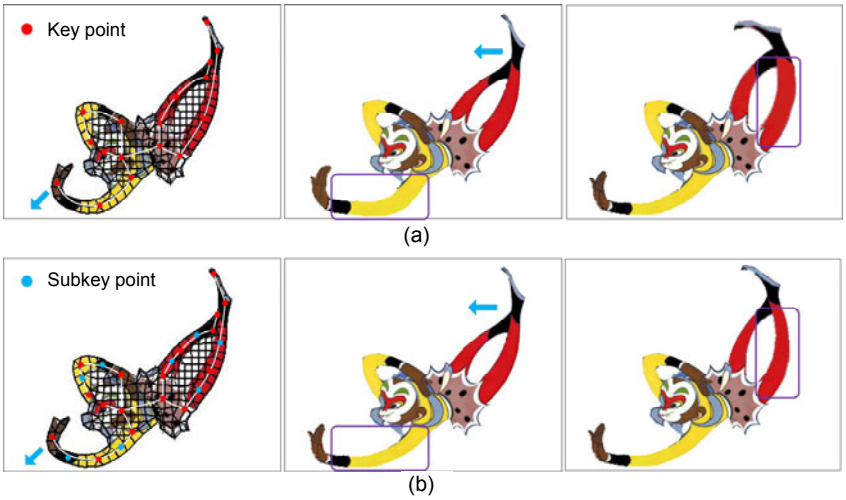

Fig. 8 Comparison of deformation results on character Wukong Sun derived by as-rigid-as-possible (ARAP) deformation (a) and our method (b) in the form of keyframes

In (a), the character mesh was driven only by key points; meanwhile, the same mesh was driven by the combination of key and subkey points in (b). Arrows denote the transforming direction

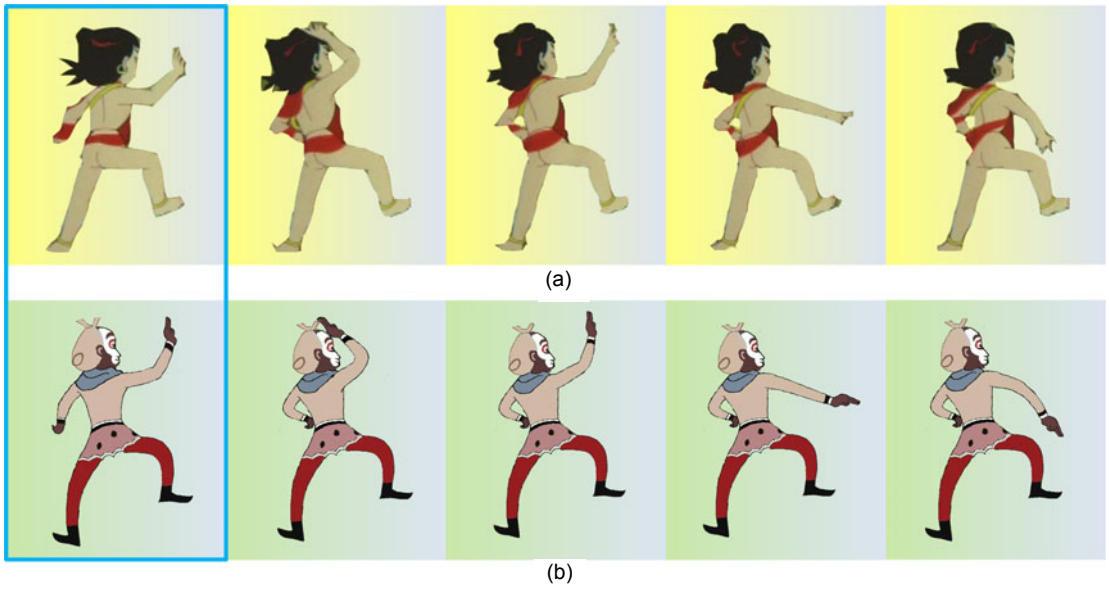

Fig. 9 An example of style-preserving results in the form of keyframes

Source motion (a) was extracted from a real cartoon video, and target results (b) were synthesized by the proposed framework. The pose of Wukong Sun in the initial frame was drawn according to that of Nezha as marked with a rectangle

Fig. 10 An example of style-preserving results in the form of keyframes

7 Conclusions and future work

In this paper, we have presented a novel framework for synthesizing style-preserving cartoons based on 2D cartoon skeleton features, in which the NNSF algorithm is designed to factorize the style basis while preserving class separation. Our approach extends the style translation from 3D motion editing to traditional 2D cartoons, which makes our work intrinsically different from previous ones. Three key issues are 2D skeleton feature extraction, deduction of the NNSF algorithm, and the subkey-driving strategy, which receive detailed analysis in previous sections. Experiments and examples demonstrate the effectiveness of the proposed framework in comparison with state-of-the-art methods.

Finally, there is still space for improvement deserving further exploration, e.g., how to solve the self-occlusion problem efficiently with fewest manual interactions in extracting 2D skeleton features. Also, the NNSF algorithm can be extended to other applications, in which features can be explained as twofold composition.

References

Agarwala, A., Hertzmann, A., Salesin, D.H., Seitz, S.M., 2004. Keyframe-based tracking for rotoscoping and animation. *ACM Trans. Graph.*, **23**(3):584-591. [doi:10.1145/1015706.1015764]

Aharon, M., Elad, M., Bruckstein, A., 2006. K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, **54**(11):4311-4322. [doi:10.1109/TSP.2006.881199]

Alexa, M., Cohen-Or, D., Levin, D., 2000. As-Rigid-as-Possible Shape Interpolation. Proc. SIGGRAPH, p.157-164. [doi:10.1145/344779.344859]

Brand, M., Hertzmann, A., 2000. Style Machines. Proc. SIGGRAPH, p.183-192. [doi:10.1145/344779.344865]

Bregler, C., Loeb, L., Chuang, E., Deshpande, H., 2002. Turning to the masters: motion capturing cartoons. *ACM Trans. Graph.*, **21**(3):399-407. [doi:10.1145/566654.566595]

Chenney, S., Pingel, M., Iverson, R., Szymanski, M., 2002. Simulating Cartoon Style Animation. Proc. NPAR, p.133-138. [doi:10.1145/508530.508553]

Freifeld, O., Weiss, A., Zuffi, S., Black, M.J., 2010. Contour People: a Parameterized Model of 2D Articulated Human Shape. Proc. CVPR, p.639-646. [doi:10.1109/CVPR.2010.5540154]

Guan, P., Freifeld, O., Black, M., 2010. A 2D Human Body Model Dressed in Eigen Clothing. Proc. ECCV, p.285-298.

Hoch, M., Litwinowicz, P.C., 1996. A semi-automatic system for edge tracking with snakes. *Vis. Comput.*, **12**(2):75-83. [doi:10.1007/s003710050049]

Hornung, A., Dekkers, E., Kobbelt, L., 2007. Character animation from 2D pictures and 3D motion data. *ACM Trans. Graph.*, **26**(1):1-es. [doi:10.1145/1189762.1189763]

Hoyer, P.O., 2002. Non-negative Sparse Coding. Proc. 12th IEEE Workshop on Neural Networks for Signal Processing, p.557-565. [doi:10.1109/NNSP.2002.1030067]

Hsu, E., Pulli, K., Popović, J., 2005. Style translation for human motion. *ACM Trans. Graph.*, **24**(3):1082-1089. [doi:10.1145/1073204.1073315]

Igarashi, T., Moscovich, T., Hughes, J.F., 2005. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, **24**(3):1134-1141. [doi:10.1145/1073204.1073323]

Jonker, R., Volgenant, A., 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, **38**(4):325-340. [doi:10.1007/BF02278710]

Kuo, P., Makris, D., Megherbi, N., Nebel, J.C., 2008. Integration of local image cues for probabilistic 2D pose recovery. *LNCS*, **5359**:214-223. [doi:10.1007/978-3-540-89646-3_21]

Kwon, J., Lee, I.K., 2008. Exaggerating character motions using sub-joint hierarchy. *Comput. Graph. Forum*, **27**(6):1677-1686. [doi:10.1111/j.1467-8659.2008.01177.x]

Lau, M., Chai, J., Xu, Y.Q., Shum, H.Y., 2009. Face poser: interactive modeling of 3D facial expressions using facial priors. *ACM Trans. Graph.*, **29**(1):1-17. [doi:10.1145/1640443.1640446]

Lee, D.D., Seung, H.S., 2001. Algorithms for Non-negative Matrix Factorization. Proc. NIPS, **13**:556-562.

Li, Y., Gleicher, M., Xu, Y.Q., Shum, H.Y., 2003. Stylizing Motion with Drawings. Proc. SCA, p.309-319.

Ma, X., Le, B.H., Deng, Z., 2009. Style Learning and Transferring for Facial Animation Editing. Proc. SCA, p.123-132. [doi:10.1145/1599470.1599486]

Moeslund, T.B., Hilton, A., Krüger, V., 2006. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Understand.*, **104**(2-3):90-126. [doi:10.1016/j.cviu.2006.08.002]

Pullen, K., Bregler, C., 2002. Motion capture assisted animation: texturing and synthesis. *ACM Trans. Graph.*, **21**(3): 501-508. [doi:10.1145/566654.566608]

Rogez, G., Orrite-Uruñuela, C., Martínez-del-Rincón, J., 2008. A spatio-temporal 2D-models framework for human pose recovery in monocular sequences. *Pattern Recogn.*, **41**(9):2926-2944. [doi:10.1016/j.patcog.2008.02.012]

Schaefer, S., McPhail, T., Warren, J., 2006. Image deformation using moving least squares. *ACM Trans. Graph.*, **25**(3): 533-540. [doi:10.1145/1141911.1141920]

Sýkora, D., Sedlacek, D., Jinchao, S., Dingliana, J., Collins, S., 2010. Adding depth to cartoons using sparse depth (in)equalities. *Comput. Graph. Forum*, **29**(2):615-623. [doi:10.1111/j.1467-8659.2009.01631.x]

Tenenbaum, J.B., Freeman, W.T., 2000. Separating style and content with bilinear models. *Neur. Comput.*, **12**(6): 1247-1283. [doi:10.1162/089976600300015349]

Tenenbaum, J.B., Silva, V., Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, **290**(5500):2319-2323. [doi:10.1126/science.290.5500.2319]

Torresani, L., Hackney, P., Bregler, C., 2007. Learning Motion Style Synthesis from Perceptual Observations. Proc. NIPS, **19**:1393-1400.

Wang, H., Li, H., 2002. Cartoon Motion Capture by Shape Matching. Proc. Conf. on Computer Graphics and Applications, p.454-456.

Wang, J., Drucker, S.M., Agrawala, M., Cohen, M.F., 2006. The cartoon animation filter. *ACM Trans. Graph.*, **25**(3): 1169-1173. [doi:10.1145/1141911.1142010]

Wang, J.M., Fleet, D.J., Hertzmann, A., 2007. Multifactor Gaussian Process Models for Style-Content Separation. Proc. ICML, p.975-982. [doi:10.1145/1273496.1273619]

Weng, Y., Xu, W., Wu, Y., Zhou, K., Guo, B., 2006. 2D shape deformation using nonlinear least squares optimization. *Vis. Comput.*, **22**(9-11):653-660. [doi:10.1007/s00371-006-0054-y]

Yan, H.B., Hu, S., Martin, R.R., Yang, Y.L., 2008. Shape deformation using a skeleton to drive simplex transformations. *IEEE Trans. Visual. Comput. Graph.*, **14**(3):693-706. [doi:10.1109/TVCG.2008.28]

Yang, Y., Zhuang, Y., Xu, D., Pan, Y., Tao, D., Maybank, S., 2009. Retrieval Based Interactive Cartoon Synthesis via Unsupervised Bi-distance Metric Learning. Proc. Conf. on Multimedia, p.311-320. [doi:10.1145/1631272.1631316]

Zhou, S., Fu, H., Liu, L., Cohen-Or, D., Han, X., 2010. Parametric Reshaping of Human Bodies in Images. Proc. SIGGRAPH, p.1-10. [doi:10.1145/1778765.1778863]