三維角色模型之SD風格轉換技術

沈亮岑* 羅聖傑* 黃群凱* 陳炳宇†
國立臺灣大學
*{olga, forestking, chinkyell}@cmlab.csie.ntu.edu.tw †robin@ntu.edu.tw

ABSTRACT
SD（super-deformed）源於日本漫畫與動畫，是一種誇張化角色使其看起來更為可愛的特殊風格。SD風格化的角色常被廣泛地應用於各個領域之中，並常見於許多動畫或是遊戲裡。然而，創造一個SD風格化的三維角色模型常需要專業技術與大量的時間及功夫。本論文提出一個將一般的三維角色模型轉換成具有SD風格的三維角色模型的技術。我們觀察到一些SD風格的特性，並且根據這些特性推導出能量函式並予以最佳化。使用者也可以給定一些與身體比例有關的參數與強化角色特徵來客製化角色。透過我們的技術，即使是剛接觸我們技術的使用者也可以在很短的時間之內創造出一個視覺上令人感到滿意的SD風格化結果。

Categories and Subject Descriptors
I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Curve, surface, solid, and object representations*

General Terms
Algorithms

Keywords
SD風格化、三維角色模型、變形、風格轉換

1. INTRODUCTION
Super-deformed, a.k.a. SD or Chibi, is a specific style of Japanese manga and anime art which exaggerates characters in the goal of appearing cute and funny. The SD characters are usually drawn or designed in distorted body proportions to resemble small babies, typically with chubby bodies, stubby limbs and oversized heads. The SD style can be seen everywhere in Japanese culture, from anime, manga to advertising. In many video games and CG movies, the characters are sometimes designed in SD style for comedic effect. The SD style is also applied to manufacture character figures and some organizations' mascots. Therefore, designing a super-deformed counterpart (i.e., SD style) of a normal character model is a common and important task for visual artists and graphic designers.

Although the SD style is not clearly defined, it often consists of a number of characteristics. First, the head length (and also width) of an SD character is normally one-half to one-third of the character's height. In contrast, the average proportion of an adult is about one-seventh. Second, the SD characters lack of the details of their normal counterparts. That is, the details such as folds on a jacket are ignored, and general shapes are favoured. Third, the signature characteristics are emphasized on the SD version to make them much more prominent. In fact, creating an SD model usually takes a professional graphic designers considerable time and effort to carefully study the character and employ a bunch of editing operations in order to achieve a visually pleasing result. In addition, the editing process usually requires a spatially-varying deformation through all the body parts rather than a simple scaling. The time-consuming editing process cannot be reused for alternative design tasks. As a result, the creation of an SD character model is challenging.

In this thesis, we present a novel technique that gives users the ability to semi-automatically create an SD version of a normal 3D character model. Our approach uses an optimization guided by some constraints based on the characteristics of the SD style. A model can be customized by specifying a small set of parameters related to the body proportions and the emphasis of the signature characteristics. In addition, a user can annotate a model by marking a number of vertices to indicate the signature characteristics. To achieve this, our system first embeds a predefined skeleton topology into a given model. Based on the skeleton, the model is deformed through an optimization such that the body proportions satisfy the user-specified parameters, the details are smoothed details, and the user-specified signature characteristics are emphasized. We constraint the body proportions in the range to satisfy the SD style properties, and thus it provides an intuitive and simple manner for users to create an SD model. Although it requires some interaction, in practice we have found it relatively efficient and simple to create an SD model with our technique. Furthermore, our technique can achieve visually pleasing results in seconds, it allows users to interactively and iteratively customize the SD models.

The primary contribution in this thesis is an optimization approach for creating an SD model of a normal character

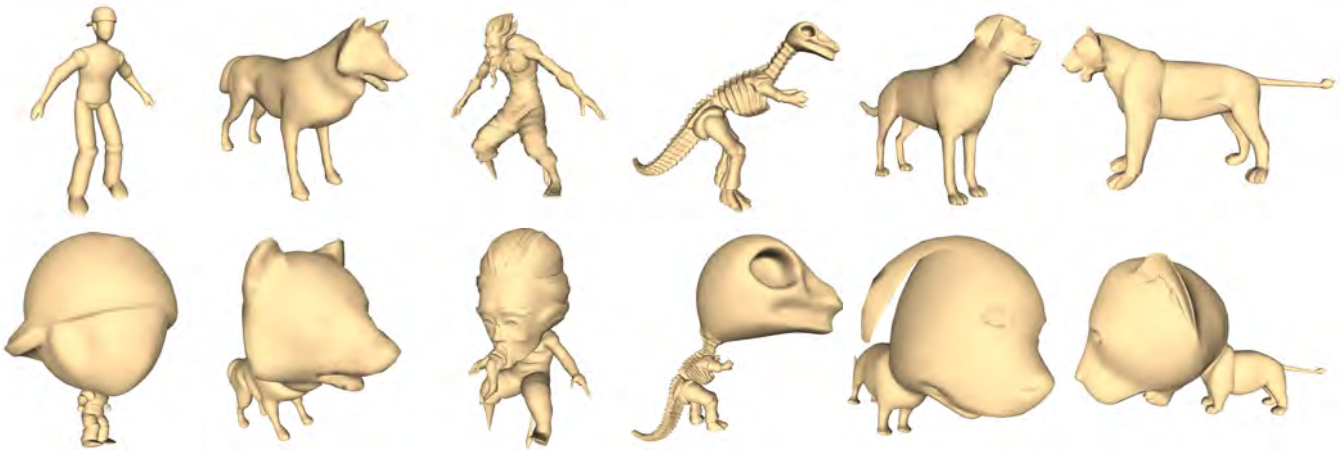


Figure 1: The SD models generated with our method. Upper row: the input models. Lower row: the SD style results. The models form left to right: baseball cap boy, wolf, Milton, dragon, dog, and lion.

that respects the user-specified parameters and constraints while minimizing a set of energy terms that model the characteristics of the SD style. We demonstrate the effectiveness of this approach with a number of results.

2. RELATED WORK

Mesh deformation. High quality mesh deformation is becoming a prominent field in geometric modeling and computer graphics. In recent years, many shape deformation techniques have been introduced. Surface-based deformation techniques [1, 14, 20, 27, 30] regard mesh deformation as an energy minimization problem. Laplacian coordinate constraints are often used to preserve mesh details and manipulate mesh deformation [1, 20, 27]. Huang *et al.* [14] enables the volume and skeleton constrains. Detail preservation can be achieved by multi-resolution techniques [4]. It first decomposes a mesh into a low frequency mesh and high frequency details, and then manipulates the low frequency mesh only while adding the details back for showing final results. In skeleton-based deformation techniques, linear-blend skinning (LBS) [18, 22] is a standard technique for character animation. A skeleton is embedded into a target mesh, and the transformations of each bone are assigned by animators. The deformed mesh is then obtained by linearly blending the bones' transformations.

Geometry processing. Our technique operates on mesh geometry to produce SD models. Many existing geometry processing methods relate to our technique, such as mesh simplification and mesh smoothing. Mesh simplification is used to reduce the number of vertices and faces of an original mesh while approximating the original shape [6, 9, 19, 21]. Mesh smoothing is a common used technique to reduce the geometry details, and also can be adopted to enhance the geometry features. Laplacian smoothing is the simplest method for mesh smoothing, which smooths the mesh geometry by relocating every vertex to the average position of its neighbors [11]. Shontz and Vavasis [26] further improved the results by applying weighted Laplacian. Baker [2] tries to preserve the surface features by estimating normals and principal curvatures. Eigensatz *et al.* [10] proposed a curvature-domain technique to edit the geometry. Although these geometry processing techniques are powerful for surface editing, they cannot be trivially applied to transfer the semantic geometry styles.

Artistic style creation. SD style is a popular artistic style in cartoon production. In practical, many techniques have been proposed to produce a specific artistic style. Non-photorealistic rendering (NPR) techniques are developed to either render a 3D model to an artistic style 2D image [8,13, 17] or convert an image or video to artistic styles [5,7,24,29].

Several 3D model creation techniques allow users to create a 3D model with a number of 2D strokes. Igarashi *et al.* [15] presented a sketching interface that allows users to create and manipulate 3D models. Nealen *et al.* [23] improved the sketching interface by allowing the user-drawn strokes to serve as handles for geometry controlling. Gingold *et al.* [12] presented a system which allows users to create 3D models by placing primitives and annotations on a 2D image. Another technique takes drawings from different views as input, and combines them to generate a 2.5D cartoon which can be used to simulate a rotation in 3D [25].

3. SD STYLIZATION

For stylizing a novel character model to generate an SD model, an optimization approach is designed with respect to a number of user-specified semantic parameters related to body proportions, such as

- Head proportion (ρ_H): the ratio of the height of the character to the head length. Based on the aesthetic rules of the SD style, this is usually constrained such that $\rho_H \in [1.5, 5]$.
- Body-to-feet proportion (ρ_{BF}): the ratio of the body length to the feet length. The range $\rho_{BF} \in [0.3, 5]$ is usually used in SD style design. By default, we maintain the original proportion of the input character model, but we also retain the feasibility for users to specify it.
- Body-to-head width proportion (δ_{WB}): the ratio of the body width to the head length. Because the head of a SD character model is usually bigger than its body, we constraint its range in $[0.3, 1]$.

In addition, the users can also specify the portion of signature characteristics by marking a set of vertices. Based on

the user-specified constraints, the embedded skeleton is first deformed. Then a set of energy functions are developed based on a careful study of the characteristics of the SD style, and a weighted least-squares optimization procedure is adopted to stylize the input model.

In this section, we first describe the definition of the input character model and the embedded skeleton in Section 3.1. We then describe the deformation of the embedded skeleton with respect to user-specified parameters in Section 3.2. Finally, the details of the mesh optimization and the energy formulations based on the deformed skeletal bones are described in Section 3.3.

3.1 Character Model and Embedded Skeleton

The input to our system is a character model's triangular mesh which is represented as $\mathbf{M} = (\mathbf{V}, \mathbf{E})$ with vertices $\mathbf{V}$ and edges $\mathbf{E}$, where $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, ...\mathbf{v}_n\}$, $\mathbf{v}_i \in \mathbb{R}^3$ denotes the vertices' positions of the input model. $\mathcal{N}(\mathbf{v}_i)$ is used to denote the one-ring neighbourhoods of vertex $\mathbf{v}_i$. Then, the input model will be stylized to generate a SD model $\mathbf{M}'$ which has the same connectivity but different geometry , $\mathbf{V}' = \{\mathbf{v}'_1, \mathbf{v}'_2, ...\mathbf{v}'_n\}$, such that the body proportions and the emphasized signature characteristics can satisfy the user-specified constraints.

To manipulate a character model, such as human or animal, professional graphical designers usually tend to adjust its underlying skeleton. Based on the observation, we define a number of skeleton templates with different topologies and embed them into corresponding character models using [3]. Formally, an embedded skeleton is defined as $\mathbf{S} = (\mathbf{J}, \mathbf{B})$ with joints $\mathbf{J}$ and bones $\mathbf{B} = \{B_1, B_2, ...B_k\}$. Each bone is represented as a line segment $B_j = \{(1-t)\mathbf{a}_j + t\mathbf{b}_j \mid t \in [0,1]\}$, where $\mathbf{a}_j, \mathbf{b}_j \in \mathbf{J}$ are two joints the bone connects to. And for each bone, it has bone weight $\omega_{B_j}(\mathbf{v})$ to every vertex $\mathbf{v}$. Corresponding skeletons (e.g., the left upper arm and the right upper arm) are first refined to the same length by scaling them to their average length. In order to achieve semantic adjustment of the character, the bones are pre-annotated with semantic labels such as head, shoulder, body, hand, and feet (Figure 2 (a) and (b)). In addition, we also categorize these bones into two sets manually according to their relevance to the character height. In Figure 2 (a) and (b), the bones coloured in red are the set related to the character height. On the contrary, the bones coloured in green are not related to the character height. These annotations and categories are used to deform the character model.

3.2 Skeleton Deformation

We observed that body reshaping, such as changing body height, heavily depends on changing the underlying skeleton. Therefore, we first deform the skeleton and calculate the target skeleton $\mathbf{S}' = (\mathbf{J}', \mathbf{B}')$ based on the user-specified parameters that are associated with body proportions. Specifically, the bones annotated as head are fixed, and the remaining bones are scaled to their target lengths along their original directions according to the head proportion (ρ_H), body-to-feet proportion (ρ_{BF}), and body-to-head width proportion (δ_{WB}). Let $\mathbf{B}_B$, $\mathbf{B}_S$, and $\mathbf{B}_F$ be the sets of bones annotated as body, shoulder, and feet, respectively. The deformed joint

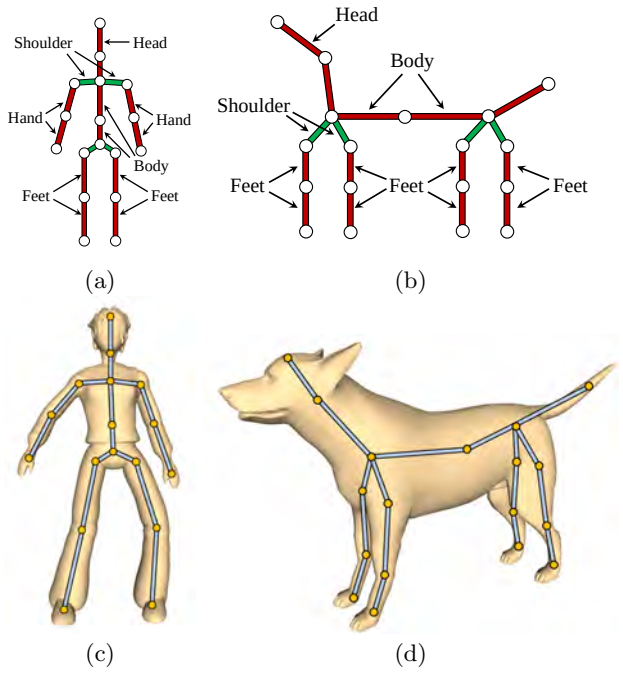


Figure 2: The skeleton templates for (a) humans and (b) quadruped animals with semantic annotations. The skeleton templates of (a) and (b) are embedded into (c) a human model and (d) a dog model, respectively.

positions $\mathbf{J}'$ are decided by solving the following equations:

$$\begin{cases} \rho_H = \frac{L(b'_H)+\sum_{b'\in\mathbf{B}_B} L(b')+\sum_{c'\in\mathbf{B}_F} L(c')}{L(b'_H)}, \\ \rho_{BF} = \frac{\sum_{c'\in\mathbf{B}_B} L(c')}{\sum_{b'\in\mathbf{B}_F} L(b')}, \\ \delta_{WB} = \frac{\sum_{b'\in\mathbf{B}_S} L(b')}{L(b'_H)}, \end{cases} \tag{1}$$

where $L(\cdot)$ refers to the length of a bone, and b'_H is the bone annotated as head.

3.3 Mesh Deformation

After obtaining the deformed skeleton $\mathbf{B}'$ under the user-specified parameters in the previous section. In this section, we introduce our novel SD stylization approach which reliably deform the input character model to an SD style model. The deformation of a character body shape requires the resizing of each body part either along their underlying skeleton axes (e.g., to increase or decrease height) or along their orthogonal directions (e.g., to gain or lose weight). As a result, our technique is based on *skeleton-aware model deformation* which adjusts each body part according to its corresponding skeletal bones. The deformation process is formulated as an optimization to compute the optimal deformed mesh geometry $\mathbf{V}'$. We then describe the deformation constraints in details.

3.3.1 Body Proportion Constraint

The body proportion constraint is designed to deform a model with respect to the deformations of its skeletal bones. A well-known standard real-time skeleton deformation method

is linear blend skinning (LBS), also called skeletal subspace deformation [18,22]. Specifically, given a closed mesh $\mathbf{M}$ and its embedded skeleton $\mathbf{B}$, each skeletal bone B_j is assigned an affine transformation T_{B_j}, which are then propagated to all vertices $\mathbf{v}$ on the mesh with linearly blending. Hence, the skeleton-driven vertices $\mathbf{v}^*$ are computed as

$$\mathbf{v}^* = \sum_{B_j \in \mathbf{B}} \omega_{B_j}(\mathbf{v}) T_{B_j}(\mathbf{v}), \tag{2}$$

where $\omega_{B_j}(\mathbf{v})$ is the weight of the bone B_j for the vertex $\mathbf{v}$. The classic LBS equation works well for animating a character model. However, to generate an SD model whose body proportions are usually distorted, the scaling of bones would result in overly stretching the vertices located beyond an endpoint of a bone, which has been well described in [16]. Therefore, we adopt a modified LBS function to resolve this problem.

The transformation of each bone can be decomposed into translation, scaling, and rotation operators. Because the deformed skeletal bones obtained from the previous section do not perform any rotation, the rotation part can be discarded. Hence, the modified equation becomes:

$$\begin{aligned}\mathbf{v}^* = \sum_{B_j \in \mathbf{B}} \omega_{B_j}(\mathbf{v})\{\mathbf{a_j}' + (\mathbf{v} - \mathbf{a_j}) + \\ (\mathbf{v} - proj_{B_j}(\mathbf{v}))(R_j - 1) + e_{B_j}(\mathbf{v})\mathbf{s}_{B_j}\},\end{aligned} \tag{3}$$

where $\mathbf{a_j}'$ is the transformed position of the endpoint $\mathbf{a_j}$, $\mathbf{s}_{B_j} = (\frac{||\mathbf{b_j}' - \mathbf{a_j}'||}{||\mathbf{b_j} - \mathbf{a_j}||} - 1)(\mathbf{b_j} - \mathbf{a_j})$ is the stretch vector at bone B_j, R_j is the ratio of deformed shoulder length to original shoulder length for all bones exclude the head bone, $R_j = 1$ for the head bone, $\omega_{B_j}(\mathbf{v})$ is computed using the heat equilibrium method presented in [3], and $e_{B_j}(\mathbf{v})$ is the end point weight defined as

$$e_{B_j}(\mathbf{v}) = \frac{||proj_{B_j}(\mathbf{v}) - \mathbf{a_j}||}{||\mathbf{b_j} - \mathbf{a_j}||}, \tag{4}$$

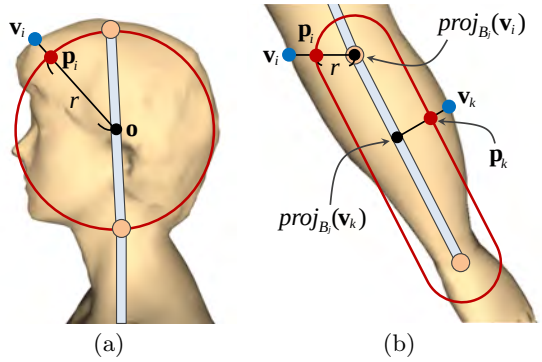
where $proj_{B_j}(\cdot)$ refers to the projection of a vertex to its nearest point on the bone B_j. Notice that the users can specify and emphasize the signature characteristics by marking some vertices. This kind of operation can be achieved by modifying their end point weights $e_{B_j}(\mathbf{v})$ (Eq.(4)), and we will describe this in more details in Section 3.3.4.

Hence, the body proportion energy is formulated by measuring the squared distance between the deformed geometry $\mathbf{v}_i' \in \mathbf{V}'$ and the skeleton-driven geometry $\mathbf{v}_i^* \in \mathbf{V}^*$ obtained from Eq.(3) as

$$E_p = \sum_{i=1}^{n} ||\mathbf{v}_i' - \mathbf{v}_i^*||^2. \tag{5}$$

3.3.2 Primitive Fitting Constraint

As mentioned before, general shapes such as sphere or cylinder are usually used to illustrate SD models. Therefore, each body part could be resembled by a specific 3D primitive for extreme SD illustration. For example, we can fit the character's head to a sphere centred at the midpoint of the head bone. As shown in Figure 3 (a), we first project the vertex $\mathbf{v}_i$ on the sphere to obtain the projected point $\mathbf{p}_i$, and then minimize the distance between them. To avoid



(a) (b)

Figure 3: Each body part of an SD model could be fitted to a basic primitive, such as (a) the head is fitted to a sphere and (b) the limb is fitted to a capsule shape, respectively.

the vertices which are influenced by other bones exclude the head bone fitted to the sphere, we set these vertices to the approximate deformed vertices multiplying associated bone weights without any modification. Figure 4 illustrates the relationship between a marked vertex $\mathbf{v}_i$ and a bone B_j, as well as the relationship between the deformed vertex $\mathbf{v}_i'$ and the deformed bone B_j'. Specifically, denote by $proj_{B_j}(\mathbf{v}_i)$ the projection of the vertex $\mathbf{v}_i$ to its nearest point on the bone B_j, we define $t_{B_j}(\mathbf{v}_i)$ as the parameter of $proj_{B_j}(\mathbf{v}_i)$ in the parametric representation of the bone B_j. That is, $proj_{B_j}(\mathbf{v}) = \mathbf{a}_j + (\mathbf{b}_j - \mathbf{a}_j)t_{B_j}(\mathbf{v}_i)$. The deformed vertex position of head primitive fitting is

$$\begin{aligned}\mathbf{p}_i = \omega_{B_{Head}}\mathbf{v})\left\{\mathbf{a}_j' + (\mathbf{o} - \mathbf{a}_j) + (\mathbf{v}_i - \mathbf{o})\frac{r}{||\mathbf{v}_i - \mathbf{o}||}\right\} + \\ \sum_{B_j \neq B_{Head}} \omega_{B_j}(\mathbf{v})\left\{\mathbf{a}_j' + t_{B_j}(\mathbf{v})(\mathbf{b}_j' - \mathbf{a}_j') + R_j(\mathbf{v} - proj_{B_j}(\mathbf{v}))\right\},\end{aligned} \tag{6}$$

where R_j is the ratio of deformed shoulder length to original shoulder length for all bones exclude the head bone and $R_j = 1$ for the head bone. And $\mathbf{o}$ and r refer to the center and radius of the sphere, respectively. The radius r can be found using least square fitting and yield $\frac{1}{\sum_{i=1}^{n}\omega_{B_{Head}}(\mathbf{v}_i)}\sum_{i=1}^{n}\omega_{B_{Head}}(\mathbf{v}_i)||\mathbf{v}_i - \mathbf{o}||$. Besides, users are also allowed to provide the sphere radius for customization. Then, the energy function is defined as

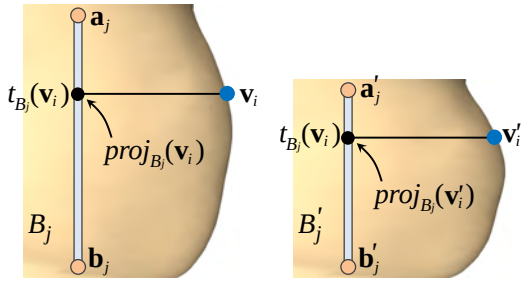
$$E_{fH} = \sum_{i=1}^{n_H} ||\mathbf{v}_i' - \mathbf{p}_i||^2, \tag{7}$$

where n_H is the number of vertices whose $\omega_{B_{Head}}(\mathbf{v}_i) > 0$.

The limbs of a character are fitted to a capsule shape as illustrated in Figure 3 (b). We connect each vertex $\mathbf{v}_i$ to its nearest point $proj_{B_j}(\mathbf{v}_i)$ on the bone B_j and find the intersected point on the capsule surface, and then translate it according to the relationship of original and deformed skeleton then get point

$$\mathbf{p}_i = \sum_{B_j \in \mathbf{B}} \omega_{B_j}(\mathbf{v})\left\{\mathbf{a}_j' + t_{B_j}(\mathbf{v})(\mathbf{b}_j' - \mathbf{a}_j') + R_j J_j(\mathbf{v})(\mathbf{v} - proj_{B_j}(\mathbf{v}))\right\} \tag{8}$$

where $J_j(\mathbf{v}) = \frac{r_j}{||\mathbf{v}_i - proj_{B_j}(\mathbf{v}_i)||}$ for $B_j \in \mathbf{B}_L$, $J_j(\mathbf{v}) = 1$ for



(a) (b)

Figure 4: (a) The relationship between a marked vertex $\mathbf{v}_i$ and a bone B_j in the original mesh. (b) The relationship between the deformed marked vertex $\mathbf{v}'_i$ and a bone B'_j in the deformed mesh.

(a) (b) (c)

Figure 5: The comparison of deforming a character model (a) without (b) and with (c) the primitive fitting constraint.

$B_j \notin \mathbf{B}_L$, and $\mathbf{B}_L$ is the set of bones marked as hand or feet. The squared distance between $\mathbf{v}_i$ and $\mathbf{p}_i$ is minimized as

$$E_{fL} = \sum_{i=1}^{n_L} ||\mathbf{v}'_i - \mathbf{p}_i||^2, \tag{9}$$

where n_L is the number of vertices whose $\omega_{B_j \in \mathbf{B}_L}(\mathbf{v}_i) > 0$.

Figure 5 shows a comparison of the results with and without the head primitive fitting constraint. With the head primitive fitting constraint (Figure 5 (c)), the head of the character becomes more circular and cuter than the result without the constraint (Figure 5 (b)).

3.3.3 *Detail Smoothing Constraint*

The SD model usually lacks of the details. Therefore, the detail smoothing constraint is designed to smooth the surface details. Here we operate on the *Laplacian coordinates* [27] which uses a set of differentials to describe the mesh geometry. To reduce the details of the geometry, we minimize the *Laplacian* through the mesh surface. Users are also allowed to specify the important features that should be preserved by painting on the surface, and the interaction alters the importances of vertices. Formally, the energy is defined as

$$E_s = \sum_{i=1}^{n} w_{\mathbf{v}_i} ||\mathbf{v}'_i - \frac{1}{|\mathcal{N}(\mathbf{v}'_i)|} \sum_{\mathbf{v}'_j \in \mathcal{N}(\mathbf{v}'_i)} \mathbf{v}'_j||^2, \tag{10}$$

where $w_{\mathbf{v}_i}$ refers to the importance of the vertex $\mathbf{v}_i$, and $\mathcal{N}(\mathbf{v}_i)$ is the one-ring neighbourhoods of the vertex $\mathbf{v}_i$.

3.3.4 *Signature Characteristic Constraint*

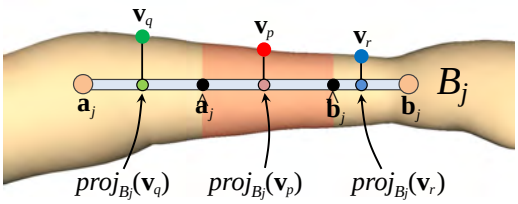


Figure 6: The illustration of the end point weight computation.

The signature characteristics of a character are critical and should be preserved or emphasized during the deformation. However, they are usually related to semantic meanings, and are not easy to be analysed via low-level features. Instead, our approach allows users to specify the signature characteristics by marking a number of vertices of the mesh, and emphasize them either along their underlying skeleton axes or along their orthogonal directions.

Emphasizing along Skeleton Axes. Emphasizing the characteristics along their underlying skeleton axes results in the elongation of the body parts, which highly relates to the scaling factor of the skeletal bones. As discussed in Section 3.3.1, the propagation of each bone's scaling to vertices can be controlled via the end point weights $e_{B_j}(\mathbf{v})$. As a result, rather than developing a new energy function, we modify the computation of the end point weights of Eq.(4), such that the propagation satisfies the user-specified constraints. Figure 6 illustrates the setup of the end point weight computation. The red region indicates the user-specified portion to emphasize along the bone B_j. We first project the marked vertices whose $\omega_{B_j}(\mathbf{v}) > 0$ to the bone B_j, and find $\hat{\mathbf{a}}_j$ and $\hat{\mathbf{b}}_j$ as the nearest points to $\mathbf{a}_j$ and $\mathbf{b}_j$, respectively. For the bone B_j, if its bone weights for the marked vertices all are equal to zero, then set $\hat{\mathbf{a}}_j = \hat{\mathbf{b}}_j$. The end point weight of each vertex is computed according to its projected point on the bone B_j, that is, the projected point may locate on $\overline{\mathbf{a}_j\hat{\mathbf{a}}_j}$ (the green vertex $\mathbf{v}_q$), $\overline{\hat{\mathbf{a}}_j\hat{\mathbf{b}}_j}$ (the red vertex $\mathbf{v}_p$), or $\overline{\hat{\mathbf{b}}_j\mathbf{b}_j}$ (the blue vertex $\mathbf{v}_r$). Formally, we define D_1 as the distance between $\mathbf{a}_j$ and $\hat{\mathbf{a}}_j$, D_2 represents the distance between $\mathbf{a}_j$ and $\hat{\mathbf{b}}_j$, and $D(\mathbf{v})$ refers to the distance between the projected point of the vertex $\mathbf{v}$ on bone B_j and $\mathbf{a}_j$. The modified end point weight of a vertex $\mathbf{v}$ to the bone B_j is then defined as

$$e_{B_j}(\mathbf{v}) = \begin{cases} \frac{D(\mathbf{v}) + ||proj_{B_j}(\mathbf{v}) - \hat{\mathbf{a}}_j||S}{||\mathbf{b}_j - \mathbf{a}_j|| + ||\hat{\mathbf{b}}_j - \hat{\mathbf{a}}_j||S}, & if D_1 \leqslant D(\mathbf{v}) \leqslant D_2 \\ \frac{D(\mathbf{v})}{||\mathbf{b}_j - \mathbf{a}_j|| + ||\hat{\mathbf{b}}_j - \hat{\mathbf{a}}_j||S}, & if D(\mathbf{v}) < D_1 \\ \frac{D(\mathbf{v}) + ||\hat{\mathbf{b}}_j - \hat{\mathbf{a}}_j||S}{||\mathbf{b}_j - \mathbf{a}_j|| + ||\hat{\mathbf{b}}_j - \hat{\mathbf{a}}_j||S}, & if D(\mathbf{v}) > D_2 \end{cases} \tag{11}$$

where S is the stretch factor provided by the users.

Emphasizing along Skeleton Orthogonal Directions. Emphasizing the characteristics along the orthogonal directions of the underlying skeleton axes results in the amplification of the body parts. To achieve this, the users can provide an enlarge factor S_E to indicate how much the marked portion should be emphasized, and the factor are used to control the distance between each marked vertex to the skeleton. We hope that after the deformation, the deformed vertex $\mathbf{v}_i$ can

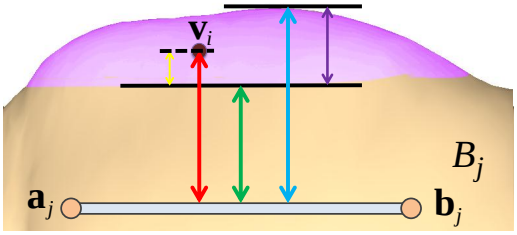


Figure 7: The arrows indicate different distances. The purple arrow is the distance between the maximum (the blue arrow) and the minimum (the green arrow) distances from vertices whose $\omega_{B_j}(\mathbf{v}_i) > 0$ in the marked region to the bone B_j. The yellow arrow is the distance between the distance from vertex $\mathbf{v}_i$ to the bone B_j (the red arrow) and the minimum distance.

be adjusted according to the user-specified factor S_E while maintaining its parameters to all bones. We can obtain the deformed vertex position

$$\dot{\mathbf{v}} = \sum_{B_j \in \mathbf{B}} \omega_{B_j}(\mathbf{v})\{\mathbf{a}'_j + t_{B_j}(\mathbf{v})(\mathbf{b}'_j - \mathbf{a}'_j) + R_j(\mathbf{v} - proj_{B_j}(\mathbf{v}))(S_E d_j(\mathbf{v}) + 1)\}, \quad (12)$$

where R_j is the ratio of deformed shoulder length to original shoulder length for all bones exclude the head bone and $R_j = 1$ for the head bone. And $t_{B_j}(\mathbf{v}_i)$ which has been defined in Section 3.3.2 is the parameter of $proj_{B_j}(\mathbf{v}_i)$ in the parametric representation of the bone B_j. For the bone B_j, if its bone weights for the marked vertices all are equal to zero, then set $d_j(\mathbf{v}) = 0$, otherwise $d_j(\mathbf{v})$ is the ratio of the distance between the distance of vertex $\mathbf{v}_i$ from the bone B_j and the minimum distance to the distance between the maximum and the minimum distances which are the minimum and maximum distances from vertices whose $\omega_{B_j}(\mathbf{v}_i) > 0$ in the marked region to the bone B_j, respectively. Figure 7 shows the distances. Then, the energy function is defined for the marked vertices as

$$E_a = \sum_{i=0}^{m} ||\mathbf{v}'_i - \dot{\mathbf{v}}_i||^2, \quad (13)$$

where m is the number of marked vertice, and $\omega_{B_j}(\mathbf{v}_i)$ is the weight of the bone B_j for the vertex $\mathbf{v}_i$.

3.3.5 *Total Energy and Optimization*

The total energy for the deformation is a weighted sum of the constraint energies defined in the previous sections.

$$E = w_p E_p + w_{fH} E_{fH} + w_{fL} E_{fL} + w_s E_s + w_a E_a. \quad (14)$$

We visually experimented and examined the SD stylization results with different relative weights, and found that a wide range of weights can work well. The weights used to generate the results demonstrated in this thesis are $w_p = 1$, $w_{fH} = 0.3$, $w_{fL} = 0.2$, $w_s = 10$, and $w_a = 1$.

The total energy is a least-square function and is linear, so it can be optimized with a linear system. We use the partial differential equation (PDE) for this energy function and put the equation in the 3n by 3n matrix A and the 3n vectors x and b, and then solve this sparse linear matrix equation with the TAUCS [28] library. Notice that none of the constraints should be satisfied absolutely, because any constraint alone does not illustrate the properties of the SD style completely. These constraints conflict with each other, so we tend to find compromised and optimal solutions among them.

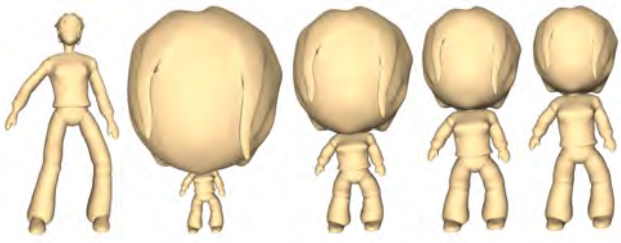


Figure 8: Some curly hair girl model results generated with different head proportion (ρ_H) and body-to-head width proportion (δ_{WB}). (a) Input model. (b) $\rho_H = 1.5$, $\delta_{WB} = 0.2$. (c) $\rho_H = 2$, $\delta_{WB} = 0.4$. (d) $\rho_H = 2.5$, $\delta_{WB} = 0.55$. (e) $\rho_H = 3$ and $\delta_{WB} = 0.65$.

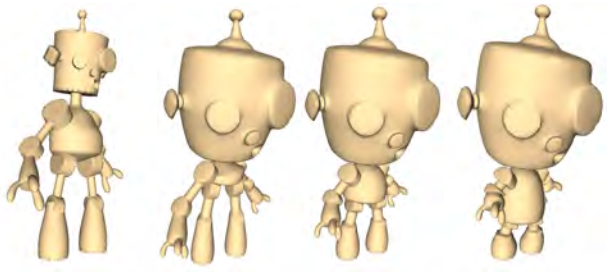
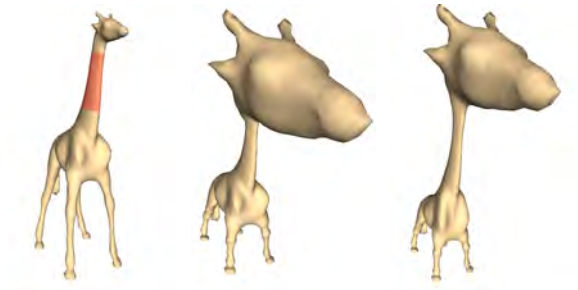


Figure 9: Some results generated with different body-to-feet proportion (ρ_{BF}). (a) Input model. (b) $\rho_{BF} = 0.3$. (c) $\rho_{BF} = 1$. (d) $\rho_{BF} = 3$.

4. RESULTS AND DISCUSSION

In this section, we demonstrate the results of our system on a number of examples. Our system can generate different models by specifying different parameters such as head proportion (ρ_H), body-to-feet proportion (ρ_{BF}), and body-to-head width proportion (δ_{WB}). Figure 8 demonstrates some results generated with different ρ_H and δ_{WB}. The head of the character model would look bigger with lower ρ_H and lower δ_{WB}. Figure 9 demonstrates some results generated with different ρ_{BF}. The body of the character model would be longer with higher ρ_{BF}.

Our technique also allows users to emphasize signature characteristics by marking some vertices. Figure 10 shows the emphasis of the giraffe's neck by stretching it along the bone axis. The result with emphasis would exaggerate the signature characteristics of the giraffe. Figure 11 shows the emphasis of the camel's hump by enlarging its hump. Our technique can be used to customize the SD models by passing above mentioned factors. Figure 12 demonstrates more SD results generated by our system.



(a) (b) (c)

Figure 10: The emphasis of the giraffe's neck. (a) The input model with user-marked portion (red area). (b) The SD result without emphasis. (c) The SD result with emphasis (stretch factor is 1.0).

(a) (b) (c)

Figure 11: The emphasis of the camel's hump. (a) The input model with user-marked portion (red area). (b) The SD result without emphasis. (c) The SD result with emphasis (enlarging factor 0.5).

Performance. We implemented our method with C++ on CPU. Performance reported for a desktop PC which equipped with an Intel i7 3.50GHz CPU and 16GB RAM. The optimization time of some cases presented in the thesis is shown in Table 1. Basically, the processing time is proportional to vertex number, and it usually takes only seconds to obtain the result.

Limitation and future work. Dependence on skeleton can be a limitation, because without correctly embedded skeleton, the constraints will not guide the deformation correctly. In addition, our method do not consider the texture coordinates of the input model. Currently, the texture would be distorted on the SD models. Two research directions are worthy of future exploration. First, we would like to explore the method of handling texture coordinates such that the texture would not be distorted. Second, the combination of image abstraction techniques and SD texture stylization is also important to produce vivid SD results.

Model	Vertex Number	Optimization Time
dog	1,480	0.749s
wolf	4,712	2.48s
lion	5,000	2.527s
giraffe	9,239	4.945s
camel	9,757	4.898s
baseball cap boy	13,336	7.852s
armadillo	25,273	16.287s

Table 1: Timing of several models presented in this thesis.

5. CONCLUSION

We have presented an optimization-based technique that enables users to semi-automatically create a character model with the SD style. Our technique supports the customization of the stylized result by specifying a small set of semantic parameters that are directly associated with the character body proportions. In addition, the users can also mark the parts with signature characteristics, and emphasize them to exaggerate the model. The applications of the proposed technique are manifold. First, it can be used to transfer an existing CG movie to SD style, which could provide a new movie watching experience. Second, it can be efficiently adapted to customize a 3D character in video games for the players. In addition, it can be used to manufacture the SD figures of a normal character. As for future work, we further consider to transfer the texture on a normal character model to the SD model, which is challenging due to the modification of texture coordinates as well as the texture itself.

6. 致謝

本論文感謝國科會經費補助，計畫編號：NSC98-2221-E-002-140-MY2。

7. REFERENCES

[1] M. Alexa. Differential coordinates for local mesh morphing and deformation. *The Visual Computer*, 19(2-3):105–114, 2003.

[2] T. J. Baker. Identification and preservation of surface features. In *Proceedings of 13th International Meshing Roundtable*, pages 299–310, 2004.

[3] I. Baran and J. Popović. Automatic rigging and animation of 3D characters. *ACM Transactions on Graphics*, 26(3):72:1–72:8, 2007.

[4] M. Botsch and L. Kobbelt. Multiresolution surface representation based on displacement volumes. *Computer Graphics Forum*, 22(3):483–492, 2003.

[5] A. Bousseau, F. Neyret, J. Thollot, and D. Salesin. Video watercolorization using bidirectional texture advection. *ACM Transactions on Graphics*, 26(3), 2007.

[6] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. *Computers & Graphics*, 22:37–54, 1997.

[7] D. DeCarlo and A. Santella. Stylization and abstraction of photographs. *ACM Transactions on Graphics*, 21(3):769–776, 2002.

[8] O. Deussen and T. Strothotte. Computer-generated pen-and-ink illustration of trees. In *ACM SIGGRAPH 2000 Conference Proceedings*, pages 13–18, 2000.

[9] S. Dong, S. Kircher, and M. Garland. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Computer Aided Geometric Design*, 22(5):392–423, 2005.

[10] M. Eigensatz, R. W. Sumner, and M. Pauly. Curvature-domain shape processing. *Computer Graphics Forum*, 27(2):241–250, 2008.

[11] D. Field. Laplacian smoothing and Delaunay triangulations. *Communications in Applied Numerical Methods*, 4:709–712, 1988.

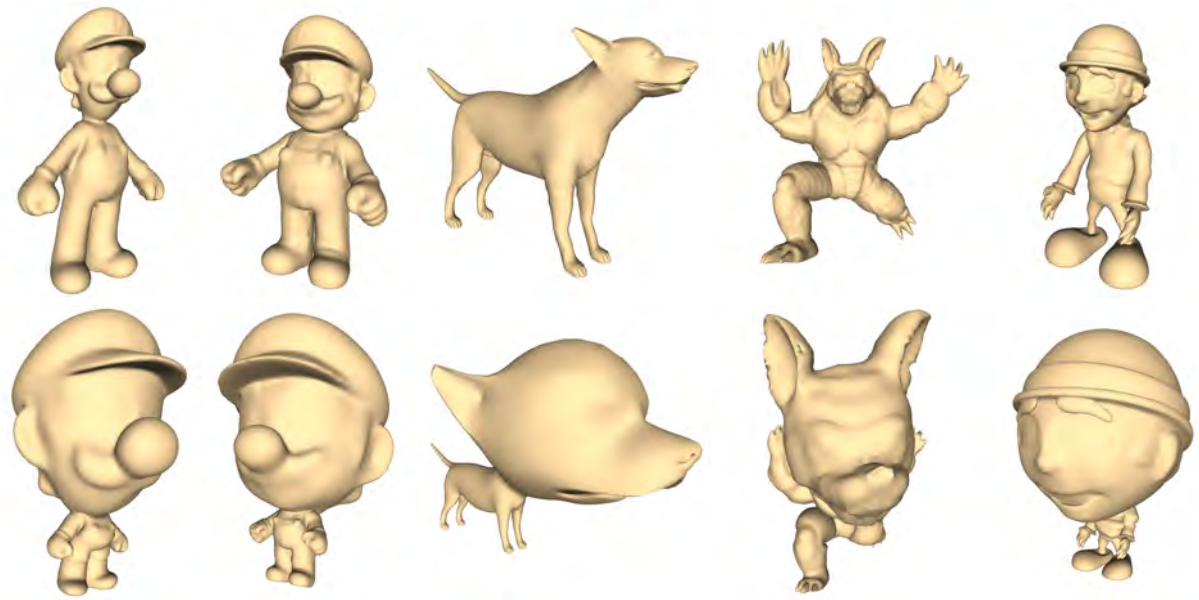


Figure 12: A number of SD results generated with our technique. The cases are Luigi, Mario, dog2, armadillo, and odd guy

[12] Y. Gingold, T. Igarashi, and D. Zorin. Structured annotations for 2D-to-3D modeling. *ACM Transactions on Graphics*, 28(5):148:1–148:9, 2009.

[13] S. Grabli, E. Turquin, F. Durand, and F. X. Sillion. Programmable rendering of line drawing from 3D scenes. *ACM Transactions on Graphics*, 29(2):18:1–18:20, 2010.

[14] J. Huang, X. Shi, X. Liu, K. Zhou, L.-Y. Wei, S.-H. Teng, H. Bao, B. Guo, and H.-Y. Shum. Subspace gradient domain mesh deformation. *ACM Transactions on Graphics*, 25(3):1126–1134, 2006.

[15] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3D freeform design. In *ACM SIGGRAPH 1999 Conference Proceedings*, pages 409–416, 1999.

[16] A. Jacobson, I. Baran, J. Popović, and O. Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Transactions on Graphics*, 30(4):78:1–78:8, 2011.

[17] M. A. Kowalski, L. Markosian, J. D. Northrup, L. Bourdev, R. Barzel, L. S. Holden, and J. F. Hughes. Art-based rendering of fur, grass, and trees. In *ACM SIGGRAPH 1999 Conference Proceedings*, pages 433–438, 1999.

[18] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *ACM SIGGRAPH 2000 Conference Proceedings*, pages 165–172, 2000.

[19] P. Lindstrom. Out-of-core simplification of large polygonal models. In *ACM SIGGRAPH 2000 Conference Proceedings*, pages 259–262, 2000.

[20] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rössl, and H.-P. Seidel. Differential coordinates for interactive mesh editing. In *Proceedings of Shape Modeling International 2004*, pages 181–190, 2004.

[21] D. Luebke, B. Watson, J. D. Cohen, M. Reddy, and A. Varshney. *Level of Detail for 3D Graphics*. Elsevier Science Inc., New York, NY, USA, 2002.

[22] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings of Graphics Interface 1988*, pages 26–33, 1988.

[23] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. FiberMesh: designing freeform surfaces with 3D curves. *ACM Transactions on Graphics*, 26(3):41:1–41:8, 2007.

[24] Y. Qu, W.-M. Pang, T.-T. Wong, and P.-A. Heng. Richness-preserving manga screening. *ACM Transactions on Graphics*, 27(5):155:1–155:8, 2008.

[25] A. Rivers, T. Igarashi, and F. Durand. 2.5D cartoon models. *ACM Transactions on Graphics*, 29:59:1–59:7, 2010.

[26] S. M. Shontz and S. A. Vavasis. A mesh warping algorithm based on weighted Laplacian smoothing. In *Proceedings of the 12th International Meshing Roundtable*, pages 147–158, 2003.

[27] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 175–184, 2004.

[28] S. Toledo. Taucs: A library of sparse linear solvers. http://www.tau.ac.il/ stoledo/taucs/, 2003.

[29] H. Winnemöller, S. C. Olsen, and B. Gooch. Real-time video abstraction. *ACM Transactions on Graphics*, 25(3):1221–1226, 2006.

[30] K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo, and H.-Y. Shum. Large mesh deformation using the volumetric graph Laplacian. *ACM Transactions on Graphics*, 24(3):496–503, 2005.