Mar. 2006, Vol.21, No.2, pp.199–203 *J. Comput. Sci. & Technol.*

Bernoulli Embedding Model and Its Application in Texture Mapping*

Hong-Xin Zhang[1] (张宏鑫), Ying Tang[1] (汤 颖), Hui Zhao[2] (赵 晖), and Hu-Jun Bao[1] (鲍虎军)

[1] *State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310027, P.R. China*

[2] *Automation Department, Xi'an Jiaotong University, Xi'an 710049, P.R. China*

E-mail: {zhx, ytang, bao}@cad.zju.edu.cn; zhaohui@mail.xjtu.edu.cn

Revised January 18, 2006.

Abstract A novel texture mapping technique is proposed based on nonlinear dimension reduction, called Bernoulli logistic embedding (BLE). Our probabilistic embedding model builds texture mapping with minimal shearing effects. A log-likelihood function, related to the Bregman distance, is used to measure the similarity between two related matrices defined over the spaces before and after embedding. Low-dimensional embeddings can then be obtained through minimizing this function by a fast block relaxation algorithm. To achieve better quality of texture mapping, the embedded results are adopted as initial values for mapping enhancement by stretch-minimizing. Our method can be applied to both complex mesh surfaces and dense point clouds.

Keywords dimension reduction, Bernoulli logistic embedding, texture mapping, parameterization

1 Introduction

Texture mapping is an essential task to enhance the rendering quality by using texture images in computer graphics. It can be viewed as an operation to planish a 3D surface onto a 2D domain in low distortions. State-of-the-art methods adopt local adjacent connectivity of meshes, involving solving a large sparse linear system. They suffer the problems of unnatural choice of boundary conditions as well as the unavoidable global stretch and central distortions. We propose a novel texture mapping technique based on a generic nonlinear dimension reduction method, called Bernoulli logistic embedding (BLE). This newly developed method preserves local adjacency configuration and global similarity both with minimal shearing effects.

2 Related Work

Dimension Reduction. The curse of dimensionality is a central difficulty in machine learning and pattern recognition. Instead of using linear approaches like Principal component analysis and multidimensional scaling (MDS)[1], nonlinear dimension reduction techniques[2−7] receive increasing interests in data processing and visualization. Most of these techniques, also referred to as *spectral embedding*[5,7], typically begin with an *affinity matrix* of pairwise relationship between the observations or the variants, and introduce an eigen-decomposition. Recently, Hinton and Roweis[8] propose a nonparametric probabilistic model called *stochastic neighbor embedding* (SNE) for dimension reduction. Inspired by this work, we use Bernoulli logistic random variables and develop a parametric one instead. Rather than utilizing the unstable and slow steepest descent algorithm in SNE[8], our block quadratic lower bound algorithm for parameter estimation adopts a second-order convergent Newton-like method. It is related to the boosting[9], and is a variance of the EM algorithm[10] without missing data.

Mesh Parameterization forms the basis of many geometry processing applications, such as texture mapping and mesh editing. It can be viewed as a simple case of dimensional reduction, which maps 3D meshes onto 2D domains. The dominant methods are based on string energy minimization on vertex neighborhoods, penalizing either large angle shrinkages or area distortions. Using boundary constraints, Harmonic mapping related approaches[11−13] provide stable results, but may cause high distortions. Alternative approaches, e.g., [14, 15] are suitable for arbitrary boundary conditions and result low distortion mappings. Unfortunately, these methods may produce triangle flips on parametric domain.

Zigelman *et al.*[16] recently adopt the Isomap algorithm[3] in mesh parameterization applications. It is well-known that parameterization will lead high distortions when a complex 3D mesh is mapped onto a single chart. Later, Zhou *et al.*[17] propose a multi-chart algorithm by combining the Isomap, mesh segmentation and stretch distortion minimization. However, there is no control factor for the data variance in the classical MDS which acts as the kernel of Isomap.

3 Bernoulli Logistic Embedding Model

In this section, we mainly describe our statistical embedding model. Let $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \boldsymbol{R}^p$ be a given set of points. Our goal is to embed them into a low-dimensional space $\boldsymbol{R}^q$ with $q < p$. Let the embedding result of $\boldsymbol{Y}$ be $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \boldsymbol{R}^q$. For each pair of points $(\boldsymbol{y}_i, \boldsymbol{y}_j)$ $(j \neq i)$, we define an affinity measure $p_{ij} \in [0, 1]$, which is a (generalized) Bernoulli random variable with a parameter q_{ij}, i.e.,

$$P(p_{ij} \mid q_{ij}) = q_{ij}^{p_{ij}}(1 - q_{ij})^{1-p_{ij}}. \quad (1)$$

*A preliminary version of this paper appeared in Proc. the 1st Korea-China Joint Conference on Geometric and Visual Computing.

This work is supported in part by the National Basic Research 973 Program of China (Grant No.2002CB312102) and the National Natural Science Foundation of China (Grant Nos. 60021201, 60505001 and 60133020).

To preserve the relationship between $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ when they are projected into $\boldsymbol{R}^q$, we treat q_{ij} as the affinity of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ which can be defined on the *logistic transformation* as

$$q_{ij} = \frac{1}{1 + e^{(-\boldsymbol{x}_i'\boldsymbol{x}_j)}} = \frac{e^{\boldsymbol{x}_i'\boldsymbol{x}_j}}{1 + e^{\boldsymbol{x}_i'\boldsymbol{x}_j}}. \quad (2)$$

It is clear that q_{ij} is invariant under rotation transformation in $\boldsymbol{R}^q$. Given i.i.d. observations $\{p_{ij}\colon\ i = 1, \ldots, n;\ j = 1, \ldots, n;\ j \neq i\}$ from the Bernoulli distribution, we have the following likelihood function

$$L = \prod_{j \neq i} q_{ij}^{p_{ij}} (1 - q_{ij})^{1 - p_{ij}}, \quad (3)$$

and the log-likelihood:

$$F = \sum_{i \neq j} [p_{ij} \log q_{ij} + (1 - p_{ij}) \log(1 - q_{ij})]. \quad (4)$$

The suitable embedding is obtained by finding the parameters which maximize the log-likelihood. Note that p_{ij} is not necessarily symmetric. In symmetric case, we only need to consider observations $\{p_{ij} : i < j\}$. In this paper, we employ the standard Gaussian kernel to model p_{ij}:

$$p_{ij} = e^{-(\boldsymbol{y}_i - \boldsymbol{y}_j)'(\boldsymbol{y}_i - \boldsymbol{y}_j)/2\sigma} \quad (5)$$

where σ is a width factor concerning the data variance. It is worth mentioning that the truncated version of the Gaussian kernel[4,10] can be adopted here to accelerate the computing processing. We call this embedding model *Bernoulli Logistic Embedding*.

4 Block Quadratic Lower Bound Algorithm

Our proposed embedding model can be efficiently calculated by a fast numerical procedure. Many algorithms in computational statistics, including our algorithm presented below, can be viewed as optimization transfers. For instance, the EM algorithm[18] is an optimization transfer algorithm depending on the notion of incomplete or missing data, and the *iterative majorization*[19] is a variance of EM without missing data[20]. These algorithms are all proceeded by maximizing or minimizing simple surrogates for objective functions. As pointed out in [21], they can be unified as the framework of *block-relaxation*.

4.1 Quadratic Lower Bound

To solve the maximizing problem described by (4), we introduce the quadratic lower bound technique. Given an objective function $F(\theta)$, e.g., (4), let $\nabla \boldsymbol{F}(\theta)$ denote the Fisher score vector and $\nabla^2 \boldsymbol{F}(\theta)$ be the Hessian matrix with $\theta \in \boldsymbol{R}^m$ being the abstract parameter vector. Böhning and Lindsay[22] propose an algorithm under the assumption that a non-negative definite $m \times m$ matrix $\boldsymbol{B}$ can be found such that $\nabla^2 \boldsymbol{F}(\theta) \succeq \boldsymbol{B}$ for all θ. Here $\boldsymbol{X} \succeq \boldsymbol{Y}$ means that $\boldsymbol{X} - \boldsymbol{Y}$ is non-negative definite. Consider the Taylor series expansion of the objective function $F(\theta)$ at ϕ up to the second order term:

$$F(\theta) - F(\phi) = (\theta - \phi)' \nabla \boldsymbol{F}(\phi) + \frac{1}{2}(\theta - \phi)' \nabla^2 \boldsymbol{F}(\phi + \eta(\theta - \phi))(\theta - \phi)$$
$$\geqslant (\theta - \phi)' \nabla \boldsymbol{F}(\phi) + \frac{1}{2}(\theta - \phi)' \boldsymbol{B}(\theta - \phi). \quad (6)$$

Here we utilize the property that $\nabla^2 \boldsymbol{F}(\theta) \succeq \boldsymbol{B}$. In general, $F(\phi) + (\theta - \phi)' \nabla \boldsymbol{F}(\phi) + \frac{1}{2}(\theta - \phi)' \boldsymbol{B}(\theta - \phi)$, denoted as $Q(\theta \mid \phi)$, is referred to as the surrogate function of $F(\theta)$. From (6), it follows that

$$F(\theta) - Q(\theta \mid \phi) = \frac{1}{2}(\theta - \phi)' \boldsymbol{K}(\theta - \phi) \geqslant 0,$$

with $\boldsymbol{K} = \boldsymbol{B} - \nabla^2 \boldsymbol{F}(\phi + \eta(\theta - \phi))$.

It is clear that $F(\theta) - Q(\theta \mid \phi)$ attains its minimum 0 at $\theta = \phi$. Since $Q(\theta \mid \phi)$ is a quadratic function, its convexity shows that it has only one maximum. The so-called optimization transfer algorithm seeks to approximate the maximum of $F(\theta)$ with that of $Q(\theta \mid \phi)$ through an iterative procedure. The quadratic lower bound algorithm is a special case of optimization transfer algorithms. Assume that ϕ is the t-th estimate of θ, i.e., $\phi = \theta^{(t)}$. Then maximizing $Q(\theta \mid \theta^{(t)})$ w.r.t. θ yields

$$\theta^{(t+1)} = \theta^{(t)} - \boldsymbol{B}^{-1} \nabla \boldsymbol{F}(\theta^{(t)}). \quad (7)$$

This iterative procedure shows that the quadratic lower bound algorithm amounts to maximizing $F(\theta)$ by using Newton's method through replacing the Hessian matrix $\nabla^2 \boldsymbol{F}(\theta)$ with $\boldsymbol{B}$. Contrarily, the iterative majorization[23] amounts to maximizing $F(\theta)$ by using the steepest descent method through replacing the fisher score $\nabla \boldsymbol{F}(\theta)$ with $\nabla L(\theta^{(t)})$. It is well-known that the Newton method is faster than the steepest descent method because the former converges at second order while the latter at first order. The following convergent theorem for this algorithm is provided in [22]:

Theorem 4.1.

(i) (*Monotonicity*) $F(\theta^{(t+1)}) \geqslant F(\theta^{(t)})$ *with* "$>$" *if* $\theta^{(t+1)} \neq \theta^{(t)}$.

(ii) (*Convergence*) *If* F *is bounded as above, then* $|\nabla \boldsymbol{F}(\theta^{(t)})| \to 0$ *as* $t \to \infty$.

4.2 Block Relaxation Framework

In this paper, we apply the lower bound principle to our Bernoulli logistic embedding problem. This technique has also been successfully applied to logistic regression, multinomial logistic and mixture models[22,23]. In order to maximize the log function F defined by (4) w.r.t. $\boldsymbol{x}_i$'s, we have the following block relaxation procedure[21], see Fig.1.

In Fig.1, either $\tau = t$ or $\tau = t + 1$ is allowed. When $\tau = t$, we refer to this procedure as *parallel-update scheme*. Otherwise, we refer to it as *serial-update scheme*. For the sake of simplicity, let $F_i = F(\boldsymbol{x}_1^{(\tau)}, \ldots, \boldsymbol{x}_{i-1}^{(\tau)}, \boldsymbol{x}_i, \boldsymbol{x}_{i+1}^{(t)}, \ldots, \boldsymbol{x}_n^{(t)})$. Because of $\frac{\partial q_{ij}}{\partial \boldsymbol{x}_i} = q_{ij}(1 - q_{ij})\boldsymbol{x}_j$ and $\frac{\partial q_{ji}}{\partial \boldsymbol{x}_i} = q_{ji}(1 - q_{ji})\boldsymbol{x}_j$, we have the first

and second derivatives of F_i w.r.t. $\boldsymbol{x}_i$ as:

$$\begin{cases} \dfrac{\partial F_i}{\partial \boldsymbol{x}_i} = \displaystyle\sum_{j\neq i}(p_{ij}+p_{ji}-q_{ij}-q_{ji})\boldsymbol{z}_j, \\ \dfrac{\partial^2 F_i}{\partial \boldsymbol{x}_i^2} = \displaystyle\sum_{j\neq i}[q_{ij}(q_{ij}-1)+q_{ji}(q_{ji}-1)]\boldsymbol{z}_j\boldsymbol{z}_j', \end{cases} \tag{8}$$

where $\boldsymbol{z}_j = \boldsymbol{x}^{(\tau)}$ if $j < i$, and $\boldsymbol{z}_j = \boldsymbol{x}^{(t)}$ otherwise. Because of $q_{ij}(1-q_{ij}) \leqslant \frac{1}{4}$ and $q_{ji}(1-q_{ji}) \leqslant \frac{1}{4}$, letting $\boldsymbol{B}_i = -\frac{1}{2}\sum_{j\neq i}\boldsymbol{z}_j\boldsymbol{z}_j'$, we conclude that $\frac{\partial^2 F_i}{\partial \boldsymbol{x}_i^2} - \boldsymbol{B}_i$ is non-negative definite. According to (7), we can obtain an iterative algorithm to solve $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$ as follows

$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} - \boldsymbol{B}_i^{-1}\nabla \boldsymbol{F}_i(\boldsymbol{x}_i^{(t)}). \tag{9}$$

Begin	Start with $\boldsymbol{x}_i^{(0)} \in \boldsymbol{R}^q$ for $i = 1, \ldots, n$.
Step $t.1$	$\boldsymbol{x}_1^{(t+1)} \in \mathrm{argmin} F(\boldsymbol{x}_1, \boldsymbol{x}_2^{(t)}, \ldots, \boldsymbol{x}_n^{(t)})$;
Step $t.2$	$\boldsymbol{x}_2^{(t+1)} \in \mathrm{argmin} F(\boldsymbol{x}_1^{(\tau)}, \boldsymbol{x}_2, \boldsymbol{x}_3^{(t)}, \ldots, \boldsymbol{x}_n^{(t)})$;
$\cdots$	$\cdots$
Step $t.n$	$\boldsymbol{x}_n^{(t+1)} \in \mathrm{argmin} F(\boldsymbol{x}_1^{(\tau)}, \boldsymbol{x}_2^{(\tau)}, \ldots, \boldsymbol{x}_{n-1}^{(\tau)}, \boldsymbol{x}_n)$.
Motor	$t \leftarrow t+1$ and go to $k.1$.

Fig.1. Basic framework of our block relaxation algorithm.

Since this algorithm employs the quadratic lower bound principle under the block relaxation, we refer to it as *block quadratic lower bound algorithm*. According to the similar spirit of the successive over-relaxation method[25], we introduce a *relaxation parameter* $\omega \in (0, 2)$ giving rise to

$$\boldsymbol{x}_i^{(t+1)} = (1-\omega)\boldsymbol{x}_i^{(t)} - \omega \boldsymbol{B}_i^{-1}\nabla F_i(\boldsymbol{x}_i^{(t)}). \tag{10}$$

4.3 Algorithm Summary

In summary, our BLE algorithm can be presented as the following pseudo code program.

Step 1. Input a point set $\boldsymbol{Y} \in \boldsymbol{R}^p$, the relaxation coefficient ω, the desired embedding dimension q and the maximal iteration step $iter_{\max}$.

Step 2. Calculate p_{ij} by (5) and generate a random point set $\boldsymbol{X}(0) \in \boldsymbol{R}^q$, $iter \leftarrow 0$.

Step 3. Calculate q_{ij} by (2), update $\nabla \boldsymbol{F}_i$ and $\boldsymbol{B}_i$.

Step 4. Block relaxation, i.e., update $\boldsymbol{X}^{(iter+1)}$ by (10).

Step 5. If $iter > iter_{\max}$ or $|\boldsymbol{X}^{(iter+1)} - \boldsymbol{X}^{(iter)}| < \varepsilon$, then output $\boldsymbol{X}^{(iter)}$. Otherwise $iter \leftarrow iter + 1$ and goto Step 3.

4.4 Relationship to the Bregman Distance

Let $F : \Omega \to \boldsymbol{R}$ be a strictly convex function on a convex set $\Omega \subseteq \boldsymbol{R}^m$. The *Bregman distance*[24], induced by F, between $\boldsymbol{p}, \boldsymbol{q} \in \Omega$ is

$$B_F(\boldsymbol{p}|\boldsymbol{q}) \doteq F(\boldsymbol{p}) - F(\boldsymbol{q}) - \nabla \boldsymbol{F}(\boldsymbol{q}) \cdot (\boldsymbol{p}-\boldsymbol{q}). \tag{11}$$

In our BLE model, it is easy to verify that the function F defined in (4) is a strictly convex function. We then have

$$B_F(\{p_{ij}\}|\{q_{ij}\}) = \sum_{j\neq i}\Big[p_{ij}\ln\Big(\frac{p_{ij}}{q_{ij}}\Big) + (1-p_{ij})\ln\Big(\frac{1-p_{ij}}{1-q_{ij}}\Big)\Big]. \tag{12}$$

Therefore, maximizing the log-likelihood function F defined in (4) is equivalent to minimizing the Bregman distance $B_F(\{p_{ij}\}|\{q_{ij}\})$. This shows the relationship between the boosting algorithm and our method. Recently, [26] studies a class of over-relaxed bound optimization algorithms and analyze the convergence in [27]. The relaxation version of our algorithm (refer to (10)) can also be regarded as an over-relaxed bound optimization method in [26].

5 Texture Mapping Based on BLE

In this section, we describe our texture mapping method based on the BLE model. Given a 3D geometry model, we can sample it by a set of points $\boldsymbol{y}_i \in \boldsymbol{R}^3$ $(i = 1, \ldots, n)$. And the BLE model is used to find corresponding points $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$ in $\boldsymbol{R}^2$. To obtain the desired mapping, all relation coefficients p_{ij} are computed using $\boldsymbol{y}_i$ first, and $\boldsymbol{x}_i$ are then obtained by our iteration method. Moreover, we generalize the computing coefficients p_{ij} (refer to (5)) by

$$p_{ij} = p(\boldsymbol{y}_i, \boldsymbol{y}_j, \sigma) := e^{-dist^2(\boldsymbol{y}_i, \boldsymbol{y}_j)/2\sigma} \tag{13}$$

where $dist(\cdot,\cdot)$ denotes a distance measure. Because vertex connectivity is not concerned in BLE model. Thus, there will be few flipped triangles on the result parametric domains. To adjust the parameterization around these flipped triangles, we adopt stretch based parameterization optimization.

5.1 Geodesic Distance Computing

In our early experiments, we adopt Euclidian distance to compute p_{ij}. Obviously, it will not provide satisfying results when $\boldsymbol{y}_i$ are sampled from High curvature surfaces. Similar to Isomap, we use geodesic distance in (13). This strategy minimizes geodesic distance distortions.

In point cloud data cases, we can use Dijkstra's algorithm to compute geodesic distance. To calculate the geodesic distances between surface points of polygonal models more precisely, we use the fast marching method[28]. It is similar to Dijstra's algorithm, except propagating distance along triangles rather than along vertices in the later algorithm. Both algorithms can calculate all the necessary geodesic distances in $O(n^2 \log n)$, where n is the number of vertices of input model.

It is well-known that multi-chart texture mapping methods are more suitable for large complex models than just using one single chart, since later strategy cannot avoid high distortions for complex shapes. Therefore, we exploit a similar spectral segmentation method described in [17] when dealing with complex models. To accelerate the computation, landmarks are selected before computing geodesic distance and segmenting model into pieces. In our current implementation, $k = \alpha n \ll n$ with $\alpha = 0.01$ vertices are selected as landmark points.

Suppose the segmentation step results in s patches. Then the geodesic distance computing step can be processed only among vertices belonging to the same patch. In summary, the geodesic distance computation is reduced to $O(kn \log n) + O(sm^2 \log m)$, where m is the maximal number of vertices of segmented patches.

To adjust those flipped triangles after using BLE, we apply stretch-minimization[29] to optimize parameterizations for all patches created by spectral analysis, similar to [17].

6 Results

We implemented all parts of the algorithm described above. The BLE part is implemented in MATLAB 6.5, and we integrate our method with the iso-charts program by data file communication. To fairly compare our method with iso-charts, we also implemented the Isomap algorithm in MATLAB. In all experiments, our algorithm shows the similar and even better running performance. Figs. 2 and 3 illustrate our experimental results. For all experiments, the stretch threshold is set at $L^2 = 1.1$. To generate the final texture atlases in Fig.3, we use the method proposed by Sander *et al.*[30] to pack all charts together. The results demonstrate that our method produces low-stretch atlases with a small number of charts. Although it is not easy to obtain similar chart configuration for comparing with iso-charts, it still can be observed from our results that our embedding method provides the high-quality mesh flattening for texture mapping application.

In Fig.2(a), a face model with around 1,000 vertices is flattened by 10 iterations of block relaxation within 50ms. Note that the result is unique up to a rotation angle, since the BLE model only concerns relative angles. This problem can be easily avoided by an additional SVD-based axes alignment. Figs. 2(b)–2(e) demonstrate the advantage of using additional stretch-minimization. As we known, the BLE model only minimizes distance distortions, and cannot avoid triangle flips. Within the additional stretch-minimization step, flipped triangles are removed from the final results, the minimization of the distance and stretch distortion are balanced.

Texture atlas of complex models in Fig.3 are generated by our modified iso-charts algorithm. That is, instead of using standard Isomap in [17], we use BLE for the patch flattening part. Our method is a parametric one and can naturally introduce an iterative numerical algorithm. These factors provide user more control power when dealing with complex shapes. Comparing with the results in [17], our method can generate evenly-sized patches for the same models with comparable computational time.

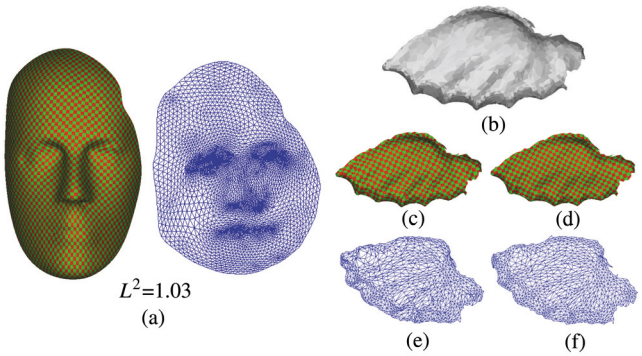

Fig.2. (a) Face model and its flattening result based on BLE. (b) Original patch. (c) and (d) Texture mapping results before and after optimization, respectively. (e) and (f) Mesh flattening effects before and after optimization, respectively. ((b)–(e) BLE optimization of a simple mesh patch.)

7 Conclusions

In this paper, we present a novel texture mapping technique based on BLE. The newly developed probabilistic embedding model uses Bregman distance to measure the similarity between two relational matrices. In texture mapping, this embedding model can be employed for minimizing geodesic distance distortions. Different from previous Isomap based algorithm, a data variance parameter is introduced in the BLE model so as to provide more control. Moreover, an efficient iterative block relaxation algorithm is derived naturally for numerical calculation of BLE. To achieve even better quality of texture mapping, the embedded results are adopted as good initial values for mapping enhancement by stretch-minimizing parameterization. Finally, the BLE model can be used as an alternative gear instead of classical MDS, and can be seamlessly integrated with the iso-charts framework. And we will explore the resent progress of BLE[31] in near future.

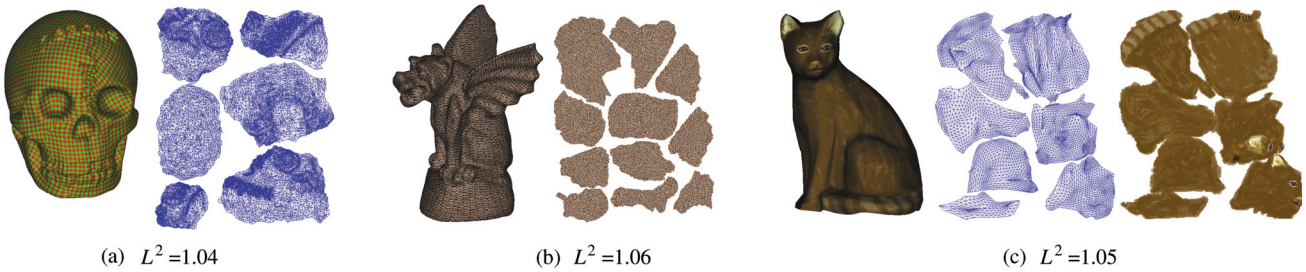

Fig.3. (a) Skull model and its atlas. (b) Textured Goblin model and its texture atlas. (c) Cat model.

Acknowledgement Special thanks to Dr. Kun Zhou for supporting us on the iso-charts implementation. Models are courtesy of Cyberware, Stanford University and Max-Planck-Institut für Informatik.

References

[1] Cox T, Cox M. Multidimensional Scaling. Second Edition, Chapman & Hall/CRC, 2000.

[2] Roweis S, Saul L. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000, 290(22): 2323–2326.

[3] J Tenenbaum, V de Silva, J Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000, 290(22): 2319–2323.

[4] Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Proc. Neural Information Processing Systems Conference*, Vancouver, Canada, 2001, pp.585–591.

[5] Brand M, Huang K. A unifying theorem for spectral embedding and clustering. In *Proc. the 9th Int. Conf. Artificial Intelligence and Statistics*, Key West, Florida, 2003.

[6] Brand M. Continuous nonlinear dimensionality reduction by kernel eigenmaps. In *Proc. 8th International Joint Conf. Artificial Intelligence*, Acapulco, Mexico, 2003, pp.547–554.

[7] Weiss Y. Segmentation using eigenvectors: A unifying view. In *Proc. Int. Conf. Computer Vision*, Kerkyra, Corfu, Greece, 1999, pp.975–982.

[8] Hinton G, Roweis S. Stochastic neighbor embedding. In *Proc. Neural Information Processing Systems Conference*, Vancouver and Whistler, BC, Canada, 2003, pp.833–840.

[9] Dekel O, Singer Y. Multiclass learning by probabilistic embeddings. In *Proc. Neural Information Processing Systems Conference*, Vancouver, BC, Canada, 2002, pp.945–952.

[10] Szummer M, Jaakkola T. Partially labeled classification with Markov random walks. In *Proc. Neural Information Processing Systems Conference*, Vancouver, BC, Canada, 2001, pp.945–952.

[11] Floater M. Parameterization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 1997, 14: 231–250.

[12] Floater M. Mean value coordinates. *Computer Aided Geometric Design*, 2003, 20: 19–27.

[13] Eck M, DeRose T, Duchamp T, Hoppe H, Lounsbery M, Stuetzle W. Multiresolution analysis of arbitrary meshes. In *Proc. SIGGRAPH*, Los Angeles, CA, USA, 1995, pp.173–182.

[14] Levy B, Mallet J-L. Non-distortion texture mapping for sheared triangulated meshes. In *Proc. SIGGRAPH*, Orlando, FL, USA, 1998, pp.343–352.

[15] Desbrun M, Meyer M, Alliez P. Intrinsic parameterization of surface meshes. In *Proc. Eurographics*, Saarbruecken, Germany, 2002, pp.209–218.

[16] Zigelman G, Kimmel R, Kiryati N. Texture mapping using surface flattening via multidimensional scaling. *IEEE Trans. Visualization and Computer Graphics*, 2002, 8: 198–207.

[17] Zhou K, Synder J, Guo B, Shum H-Y. Iso-charts: Stretch-driven mesh parameterization using spectral analysis. In *Proc. Eurographics Symposium on Geometry Processing*, Nice, France, 2004, pp.45–54.

[18] Dempster A, Laird N, Rubin D. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 1977, 39(1): 1–38.

[19] Borg I, Groenen P. Modern Multidimensional Scaling. New York: Springer-Verlag, 1997.

[20] Becker M, Yang I, Lange K. EM algorithms without missing data. *Statistical Methods in Medical Research*, 1997, pp.38–54.

[21] J de Leeuw. Block Relaxation Algorithms in Statistics. Information Systems and Data Analysis, Bock H H, Lenski W, Richter M M (eds.), Berlin: Springer-Verlag, 1994, pp.308–325.

[22] Böhning D, Lindsay B. Monotonicity of quadratic-approximation algorithms. *Annals of the Institute of Statistical Mathematics*, 1988, 40: 641–663.

[23] Böhning D. Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, 1992, 44: 197–200.

[24] Collins M, Schapire R, Singer Y. Logistic regression, Adaboost and Bregman distances. *Machine Learning*, 2002, 48(1-3): 253–285.

[25] Young D. Iterative Solution of Large Linear Systems. Orlando, Florida: Academic Press, Inc., 1971.

[26] Salakhutdinov R, Roweis S. Adaptive overrelaxed bound optimization methods. In *Proc. the 20th Int. Conf. Machine Learning*, Washington DC, USA, 2003, pp.664–671.

[27] Salakhutdinov R, Roweis S, Ghahramani Z. On the convergence of bound optimization algorithms. In *Proc. Uncertainty in Artificial Intelligence*, Acapulco, Mexico, 2003, pp.509–516.

[28] Kimmel R, Sethian J. Computing geodesics on manifolds. In *Proc. National Academy of Sciences*, USA, 1998, pp.8431–8435.

[29] Sander P, Snyder J, Gortler S, Hoppe H. Texture mapping progressive meshes. In *Proc. SIGGRAPH*, Los Angeles, CA, USA, 2001, pp.409–416.

[30] Sander P, Wood Z, Gortler S, Snyder J, Hoppe H. Multi-chart geometry images. In *Proc. Symposium on Geometry Processing*, Aachen, Germany, 2003, pp.146–155.

[31] Wang G, Zhao H, Zhang Z, Lochovsky H. A Bernoulli relational mode for nonlinear embedding. In *Proc. IEEE Int. Conf. Data Mining*, Houston, Texas, USA, 2005, pp.458–465.

Hong-Xin Zhang is an assistant professor of the state key laboratory of CAD&CG at Zhejiang University, P.R. China. He received his BS. and Ph.D. degrees in applied mathematics from Zhejiang University. His research interests include geometric modeling, texture synthesis and machine learning.

Ying Tang is now a post-doctoral researcher in the Computer Science Department at Hong Kong University of Science and technology. Her research interests mainly focus on mesh parameterization, texture synthesis, texture compression and visualization. She received a B.S. degree in 1999, and a Ph.D. degree in 2005, all from Zhejiang University, China.

Hui Zhao is a lecturer of Department of Automatization Science & Technology, Xi'an Jiaotong University. She is also a Ph D. candidate of Xi'an Jiaotong University in control theory. Her research interests are computer control, image processing, data mining, etc.

Hu-Jun Bao is a professor of the State Key Laboratory of CAD&CG at Zhejiang University, P.R. China. He received his M.Sc. and Ph.D. degrees in applied mathematics from Zhejiang University. His research interests include computer graphics, geometric modeling and virtual reality.