Springer

KSME

Journal of Mechanical Science and Technology 28 (7) (2014) 2469~2477

www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-014-0604-6

Upright orientation of 3D shapes via tensor rank minimization†

Weiming Wang[1], Xiuping Liu[1] and Ligang Liu[2,*]

[1]School of Mathematical Sciences, Dalian University of Technology, Dalian, 116-023, China
[2]School of Mathematical Sciences, University of Science and Technology of China, Hefei, 230-026, China

(Manuscript Received October 11, 2013; Revised February 13, 2014; Accepted February 23, 2014)

Abstract

In general, the upright orientation of a model is beneficial for human to recognize this model and is widely used in geometry processing and computer graphics. However, the orientation of the model obtained by existing technologies, such as 3D scanning systems or modeling, may be far away from the right orientation. In order to solve this problem, a robust and efficient upright method is needed. We observe that when the model is aligned with the three axes, the rank of the three-order tensor constructed by this model is the lowest usually. Inspired by this observation, we formulate the alignment of the 3D model with axes as a low-rank tensor optimization problem which is a global and unsupervised method and the genetic algorithm (GA) is used to solve this optimization problem. After the 3D model has been aligned with the three axes, some geometric properties can be used to pick out the best upright orientation from the six candidate supporting bases easily. The three-order tensor is constructed by voxelizing the bounding box of the 3D model, and then filling the voxel element with zero or one based on whether it contains the points of the model or not. The experimental results demonstrate that our method is robust, efficient and effective for all kinds of the models (manifold or non-manifold, man-made or non-artificial, or point cloud).

Keywords: Upright orientation; Tensor; Low-rank; Voxelization; Bounding-box

1. Introduction

Usually, humans are willing to look at things from a good view, because more information can be captured from this view than others and can help them recognize things easily. Under the action of the gravity, almost all of the models have their unique supporting bases which defining the upright orientations of the models. Obviously, the supporting base is the nature property of the model. However, models obtained by existing technologies, such as 3D scanning systems [1] or modeling [2, 3], are often deviation from the upright orientation.

Despite of the views of the human, upright orientation of the model is widely used in geometry processing, such as shape matching [4], shape retrieval [5] and shape registration [6]. The upright orientation of the model is beneficial for these algorithms to find the best matching shapes and parts. The performances of these methods highly rely on the upright orientations of the input models. To make these methods robust and effective, a preprocessing algorithm is needed to align the models in the same coordinate system.

There exist a lot of 3D geometry processing software, such as Maya, 3DS Max, Meshlab and deep exploration, which can be used to obtain the upright orientations of the models. The only operation can be used to get the upright orientation of the 3D model is the rotation. Users have to continue to rotate the model by rotation operation of the software until the upright orientation is obtained. However, this procedure is time-consuming and inaccurate which can affect the performance of the 3D geometry processing algorithms. Especially, if the direction of the model is too random, such as the direction in Fig. 1(a), it is scarcely possible to obtain the upright orientation accurately with the rotation operation of the existing 3D geometry processing software.

As the upright orientation is important for some geometry processing, several researchers have proposed some methods to deal with it. The commonly used technology is to align the model with axes, and then some geometric properties of the 3D model are applied to find the final upright orientation from the six candidate supporting bases. Fu et al. [7] use training set to deal with this problem which is a supervised method. Although their method is great for almost all of the man-made models, the performance of their method relies on the quality of the training set seriously. The low-rank theory of the 2D matrix is used by Jin et al. [8] to align the 3D model with axes. Their method is an unsupervised method and can achieve a great result when the model has perfect symmetries. However,

*Corresponding author. Tel.: +86 551 63600985, Fax.: +86 551 63600985
E-mail address: lgliu@ustc.edu.cn
† Recommended by Associate Editor Gil Ho Yoon

if a model has some ambiguities or asymmetries, their method may fail (see Fig. 8), as their method needs to iteratively rectify the rank of the three images independently which obtained by projecting the 3D model on the three coordinate planes and just uses the local information of the 3D model. Although their method can obtain the minimum of the 2D matrix rank, they cannot guarantee the minimum of the three-order tensor rank. The results in the middle row of Fig. 8 obtained by Jin et al. [8] show that their method can only obtain a local alignment in this situation. Furthermore, their method cannot deal with point cloud, composed models and incomplete models (see Fig. 6).

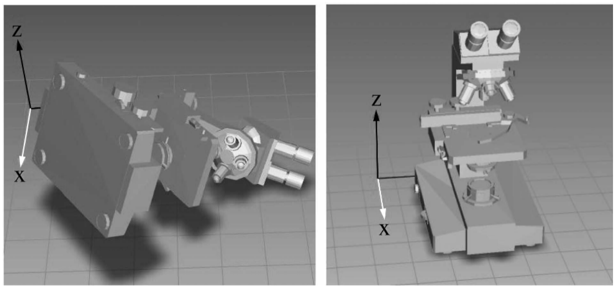
In this paper, we propose a global and unsupervised method to find the upright orientation robustly, automatically, and effectively. The observation of our paper, which is inspired by the recent work of Ref. [9], is that the rank of the three-order tensor constructed by the 3D model is the lowest if the 3D model has been aligned with axes. Fig. 1 shows a model with two different orientations. The rank of the three-order tensor constructed by the model in Fig. 1(a) is 284, while the rank in Fig. 1(b) which has been aligned with axes by our method is 147. As the boundary of the 3D model can reflect the symmetry of the model fully [10], we use the surface points instead of the volume of the 3D model to construct three-order tensor in this paper. The rank of the three-order tensor constructed with the surface points is enough to reflect the global external symmetries of the model. Furthermore, the elements of the three-order tensor constructed in this way are sparser and the memories needed to store these elements are smaller than that constructed with the volume of the 3D model (see Fig. 5). Simultaneously, in order to deal with point cloud, composed models, non-manifold models and incomplete models (see Fig. 6), it is necessary to construct the three-order tensor with surface points of the 3D model, as these kinds of models cannot define their internal. An optimization model is proposed which is used to search an optimal rotation matrix by minimizing the rank of the three-order tensor. The genetic algorithm (GA) [11] is used in our paper to solve this optimization problem. After the optimization, the 3D model has been aligned with axes whose candidate supporting bases are reduced to be the six faces of the bounding box of the model. Then, the best upright orientation can be picked out from the six candidate supporting bases easily by analyzing the geometric properties of the 3D model similar to Refs. [7, 8].

The main contributions of our proposed method are as follows:

(1) The three-order tensor constructed in this paper is very sparse which is beneficial for calculation and storage and it can reflect the original symmetry of the 3D model.

(2) As the rank of the three-order tensor can reflect the global external symmetry of the 3D model, so our proposed method is a global and unsupervised method which do not need training set.

(3) Our algorithm is flexible for all kinds of data including manifold and non-manifold, man-made and non-artificial, or even point cloud, as the three-order tensor constructed just using the surface points of the model.



(a) Tensor rank = 284 (b) Tensor rank = 147

Fig. 1. A model is posed at two different orientations: (a) random direction; (b) upright orientation obtained from the direction in (a) by our method.

2. Related work

Orientation of image. A digitized or scanned photograph can differ from its correct orientation by 0^{o}, 90^{o}, 180^{o}, or 270^{o}. Therefore, the image orientation detection problem is represented as a four-class classification problem [12]. High dimensional feature vector in each possible orientation is extracted, and then the similarity score is obtained by a Support Vector Machine [13]. The higher the score, the better the orientation. However, the high dimensional feature vectors need a very large training dataset of the 3D models.

Upright orientation of the 3D models. In geometry processing and computer graphics, a lot of applications need to align the given models. The commonly used method is the principal component analysis (PCA) [14]. However, this method needs to compute the center of the mass and the principal axes of the model which are not robust and inaccurate for many models, such as the noise models.

In Ref. [7], geometric properties of the model are used to reduce the candidates of the upright orientation, and then the best upright orientation is obtained by a learning method. Their method is great for almost all of the man-made models. However, the robustness and effectiveness of their method extremely relies on the quality of input model and training set.

Inspired by low-rank of 2D matrices theorem, Jin et al. [8] propose an unsupervised upright orientation algorithm. They project model at three axis-aligned orientations to obtain three projection matrices, and the rank of these matrices are optimized for aligning the model with axes. The model is aligned with axes by iterative rectification of axis-aligned projections as low-rank matrices independently. However, their method just considers the local of the 3D model which is a local method.

Viewpoint selection. As the right view may contain more information than any random orientation and can be very useful, so automatically and effectively selecting viewpoint of the 3D model is very important. A lot of methods have been proposed to deal with the viewpoint selection problem. Some different metrics, including viewpoint entropy [15], view saliency [16] and shape distinction [17], are applied to maximize the visibility of interesting content. Refs. [18, 19] minimize the visible redundant information of the symmetry and similarity. Ref. [20] selects the best views automatically by considering the geometry and visual salient features simultaneously. There is no doubt that the viewpoint of the 3D model can be found easily after the model has been posed at its upright orientation.

3D shape matching, retrieval and registration. The aims of the 3D shape matching [4] and retrieval [5] are to find the best similarity models with queries. And the aim of the 3D shape registration [6] is to find the similarity part of two models. The main difficulty of these problems is how to find the way to measure the similarity between the two models or parts. Most of them propose feature descriptors firstly, and then similarities are measured by the distances of the descriptors. However, in order to make the descriptors meaningful and powerful, all of the models in the database and queries must be aligned to the same coordinate system, like the descriptors proposed in Ref. [21]. Commonly used alignment methods include PCA [14] and manual adjustment using 3D geometry processing software. However, these methods are not robust and time-consuming. Our proposed method can help to deal with these problems automatically and efficiently.

Three-order tensor. The low-rank tensor theory has been widely used in computer vision and computer graphics. The details of the tensor decomposition and applications are introduced by Kolda et al. [22]. Liu et al. [23] propose an algorithm to estimate missing values in three-order tensors of visual data. Their algorithm can recover an image with 80% elements missing. Considering that medical image, e.g. CT or MRI, has the natural form of three-order tensor, Guo et al. [24] recognize the problem of medical hole-filling as the low-rank of tensor. As the 3D model can be converted to the three-order tensor easily and naturally, the low-rank tensor theory is used in this paper to deal with the upright orientation problem of the 3D model.

3. Tensor and its rank

3.1 Definition of tensor

A tensor is a multidimensional array. More formally, an n-order tensor is an element of the tensor product of n vector spaces, each of which has its own coordinate system [22]. We use bold upper case letters for matrices, e.g. **X**, and χ denotes tensor. An n-order tensor is defined as $\chi \in \Re^{I_1 \times I_2 \times \ldots I_n}$. Its elements are denoted as $\chi(i_1, i_2, \ldots, i_n) = x_{i_1, i_2, \ldots, i_n}$, where $0 \le i_k \le I_k, 1 \le k \le n$. For example, a vector is a 1-order tensor and a matrix is a 2-order tensor. It is sometimes convenient to unfold a tensor into a matrix [22]. The "unfold" operation along the k-th order on a tensor χ is defined as $unfold_k(\chi) := \mathrm{X}_{(k)} \in \Re^{I_k \times (I_1 \ldots I_{k-1} I_{k+1} \ldots I_n)}$. The opposite operation "fold" is defined as: $fold_k := \chi$. Since the rank of the tensor is difficult to calculate, the trace norm of the tensor is used to approximately calculate the rank of the tensor, as the trace norm of the tensor is the tightest convex envelop for the rank of tensor [25] and is easy to calculate.

3.2 Definition of tensor trace norm

In this paper, the trace norm of the tensor is used to approximately calculate the rank of the tensor which is defined as follows [22]:

$$\| \chi \|_* := \sum_{i=1}^{n} \alpha_i \| \mathrm{X}_{(i)} \|_* \tag{1}$$

where α_i 's are constants satisfying $\alpha_i \ge 0$ and $\sum_{i=1}^{n} \alpha_i = 1$, and $\| \mathrm{X}_{(i)} \|_*$ is the trace norm of the 2D matrix $\mathbf{X}_{(i)}$ which is defined as follows:

$$\| \mathrm{X}_{(i)} \|_* = \sum_i \sigma_i(\mathrm{X}_{(i)}) \tag{2}$$

where $\sigma_i(\mathrm{X}_{(i)})$ denotes the i^{th} largest singular value of $\mathbf{X}_{(i)}$.

3.3 Three-order tensor construction

In Ref. [8], they project 3D model on three axes planes to obtain three images which contain not only the boundary but also the internal of the projected model. It is very natural to generalize their method in 3D space to construct three-order tensor with volume of the 3D model. However, the tensor constructed in this way cannot deal with point cloud, incomplete models, non-manifold models and composed models (see Fig. 6), as these models cannot define the internal of the model, i.e. we cannot construct three-order tensor with the volume of these models. Furthermore, the elements of the three-order tensor constructed with the volume of the 3D model are very huge which will take up very large memory and affect the speed of calculating trace norm of the three-order tensor. As the boundary of the 3D model is enough to reflect the symmetry of the model [10], we just need to use the surface points of the 3D model to construct three-order tensor whose elements are very sparse (see Fig. 5).

It is known to all that the bounding box of the 3D model parallels the coordinate planes and contains the whole model. Therefore, the bounding box of the 3D model is used to construct our three-order tensor in this paper. The details of the three-order tensor construction are introduced in the following.

First of all, we voxelize the bounding box with $m \times n \times l$ voxel elements (see the left of the Fig. 2), where m, n, and l indicate the number of voxel elements along the x, y and z-axes respectively which can be specified by users. Each voxel element can be considered as an element of the three-order tensor and is filled with zero.

Then, for each point **v** of the 3D model, the position of the voxel element that contains this point can be calculated as follows:

$$\begin{aligned} ind_x &= (v_x - min_x) / e_x \\ ind_y &= (v_y - min_y) / e_y \\ ind_z &= (v_z - min_z) / e_z \end{aligned} \tag{3}$$

where (ind_x, ind_y, ind_z) corresponding to the row, column and height of the element in the three-order tensor, (v_x, v_y, v_z) is the coordinate of the surface point **v**, and (min_x, min_y, min_z) are minimum values of the *x*-coordinate, *y*-coordinate and *z*-coordinate of the bounding box of the 3D model. e_x, e_y and e_z are the length, width and height of the voxel element which can be calculated by $e_x = l_x / m$, $e_y = l_y / n$ and $e_z = l_z / l$, where l_x, l_y and l_z denote the length, width and height of the bounding box of the model respectively (see Fig. 2).

At last, the position pointed by (ind_x, ind_y, ind_z) in the three-order tensor is set to one, that is to say $\chi(ind_x, ind_y, ind_z) = 1$. For example the voxel element bounded by the blue line in Fig. 2, this voxel element contains the points of the model, so it is filled with one. After all of the points of the 3D model are traversed, our three-order tensor has been constructed completely.

4. Algorithm

The input model of our algorithm can be a manifold or non-manifold mesh, or even point cloud, denoted by *M*. The point of the model is denoted by **v** and the coordinates of the points of the *M* are denoted by **V**, which is a $n \times 3$ matrix where *n* is the number of the points of the *M*. In our *x*-*y*-*z* coordinate system (green-blue-red in all figures in our paper), the upright orientation is defined by the positive *z*-axis direction in our paper. In order to facilitate the subsequent operations, we translate the barycenter of the input model to the origin of the coordinates of the coordinate system. At this time, we just need to find an optimal rotation matrix **R** so that the 3D model is posed at its upright orientation. It is known to all that any rotation matrix can be decomposed into $\mathrm{R}(\theta_x)$, $\mathrm{R}(\theta_y)$ and $\mathrm{R}(\theta_z)$, i.e. $\mathrm{R} = \mathrm{R}(\theta_x) * \mathrm{R}(\theta_y) * \mathrm{R}(\theta_z)$, which are the rotation matrix around *x*-axis, *y*-axis and *z*-axis respectively, where θ_x, θ_y and θ_z are the rotation angles around three axes. The definition of the $\mathrm{R}(\theta_z)$ is given as follows:

$$\mathrm{R}(\theta_z) = \begin{pmatrix} cos(\theta_z) & sin(\theta_z) & 0 \\ -sin(\theta_z) & cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4}$$

$\mathrm{R}(\theta_x)$ and $\mathrm{R}(\theta_y)$ can be defined similar to $\mathrm{R}(\theta_z)$.

In Ref. [8], Jin et al. project the 3D model on three coordinate planes to obtain three 2D projection images. And then, TILT method [9] is used to obtain an optimal rotation matrix so that the ranks of the three projected matrices are as small as possible. Their method using the rank of the projection matrix is a local method and not robust. In this paper, we use the rank of three-order tensor instead of the rank of 2D matrix which can capture the global information of the 3D models. Similar to Ref. [8], our proposed method is composed of two steps. In the first step, 3D model is aligned with the three axes by an optimization model proposed in the next section. And then, the upright orientation is picked out by analyzing the geometric properties of the 3D model in Sec. 4.3.

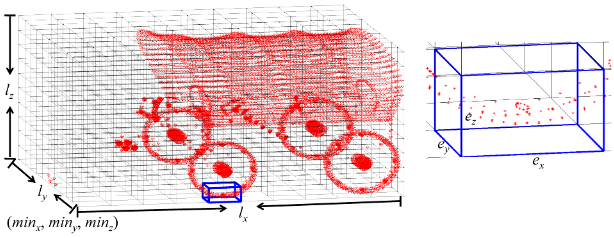


Fig. 2. Voxelization of the bounding box and some notations used in constructing the three-order tensor. The local zooming of the blue cube is shown in the right. The red points are the surface points of the model.

4.1 Objective function

Our observation is that most of the 3D models have symmetry, if the model is aligned with three axes, the model must be in a symmetric form and the rank of the three-order tensor constructed with this model must be the lowest. Inspired by this observation, the alignment of the 3D model with axes is formulated as a low-rank tensor optimization problem to find an optimal rotation matrix $\mathbf{R}^*$ so that the rank of the three-order tensor constructed with the model is the lowest. The optimization model proposed as follows:

$$\mathrm{R}^* = argmin_{\mathrm{R}}(Rank(\chi(\mathrm{V} \circ \mathrm{R}))), \tag{5}$$

where $\mathrm{V} \circ \mathrm{R}$ indicates the coordinates of the new mesh *M'* that obtained by rotating *M* with **R**, and $\chi(\mathrm{V} \circ \mathrm{R})$ is a three-order tensor generated with *M'*. *Rank* (.) denotes the rank of the three-order tensor.

The optimization problem Eq. (5) is non-convex, as the function *Rank* (.) is non-convex. In order to solve Eq. (5) efficiently, one common approach is to use the trace norm $\|.\|_*$, which is defined in Eq. (1), to approximate the rank of three-order tensor [23]. Therefore, the Eq. (5) can be converted into the following optimization model:

$$\mathrm{R}^* = argmin_{\mathrm{R}}(\| \chi(\mathrm{V} \circ \mathrm{R}) \|_*). \tag{6}$$

4.2 Optimization

The Eq. (6) is highly nonlinear as the unknown is the rotation transformation **R** which is difficult to be converted into a linear problem. Furthermore, the three-order tensor and its trace norm must be calculated in each step of the optimization

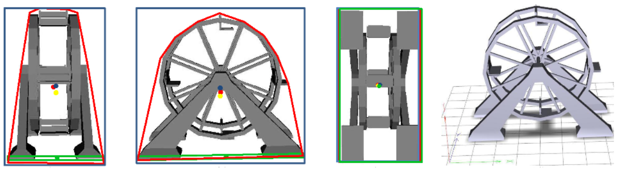

(a) (b) (c) (d)

Fig. 3. Some geometric properties of the 3D model used in this paper and the final selected upright orientation by our method: (a)-(c) projection of the model on the three axes planes; (d) upright orientation and the model has been posed at its upright orientation.

which is very hard to be handled in any optimization method. Therefore, there are no appropriate optimization methods to solve it effectively. In order to solve this problem, the genetic algorithms [11] is used in this paper. The rotation angles θ_x, θ_y and θ_z are encoded to generate our initial population, and fitness function is the trace norm of the three-order tensor. Our algorithm stops when the trace norm of the three-order tensor dose not decreases any more. As the rank of the three-order tensor constructed by the 3D model has minimum value, the genetic algorithms can convergence to a local minimum. At this time, the model has been aligned with axes and the candidate supporting bases have been reduced to be the six faces of the bounding box of the model by our optimization algorithm (see the second column of the Fig. 4). The upright orientation of the model can be picked out easily by analyzing the geometric properties of the model in Sec. 4.3 (see the last column of the Fig. 4).

4.3 Picking out upright orientation

After Eq. (6) is solved, the optimal rotation matrix can be obtained. Now, we can rotate the input model with this rotation matrix to get an axes-aligned model whose supporting base can be any of the six faces of the bounding box. Then, the best upright orientation should be picked out from the six supporting bases.

Similar to Refs. [7, 8], geometric properties of the 3D model are used to calculate some scores for each candidate supporting base of the model, and the best upright orientation is selected based on these scores.

As not all of the points of the supporting base are the supporting points of the model, so we should compute the actual supporting points of the model which is defined as follows:

$$P_S(M) := \{P_S(\mathrm{v}) : d(\mathrm{v}, S) < \varepsilon, \mathrm{v} \in M\} \qquad (7)$$

where $P_S(\mathbf{v})$ denotes the projection of point **v** on the supporting base S, $d(.)$ denotes Euclidean distance, and ε is a constant.

The geometric properties that used in this paper are as follows (see Fig. 3):

(1) Convex hull of the actual supporting points $H(P_S)$ (green polygon) and its barycenter $C(H(P_S))$ (green point).

(2) Projection of the model center mass on the current supporting base C(M) (yellow point).

(3) Convex hull of the model projection on the current supporting base H(MS) (red polygon) and its barycenter C(H(MS)) (red point).

(4) The current supporting base (blue polygon) and its center C(P) (blue point).

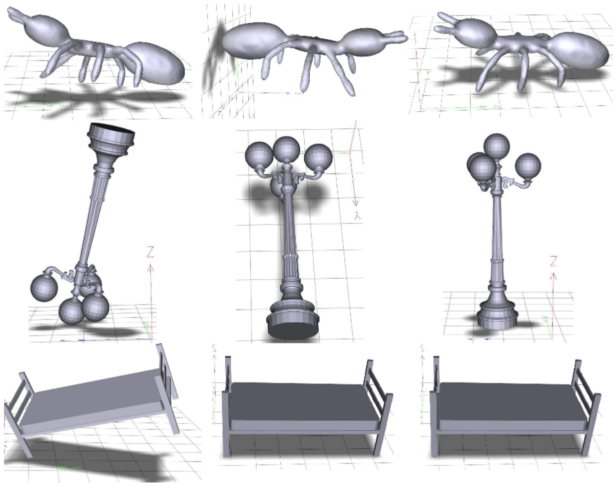


Fig. 4. Three results of our method. Left column shows the input models with random orientations middle column shows the models aligned with axes, and right column shows the final results. All models have been posed at their upright orientations by our method.

Geometric score. Using the geometric properties defined above, we compute stability score S_{sta}, symmetry score S_{vis} and visibility score S_{sym} for each candidate supporting base, like Refs. [7, 8]. As none of the single score can be used to find the best upright orientations for all of the models, so these scores should be combined to calculate a geometric score for the model which is defined as follows:

$$G_S := \alpha S_{sta} + \beta S_{vis} + \gamma S_{sym} \qquad (8)$$

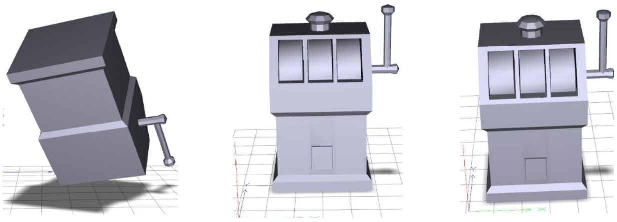
where α, β and γ are combination coefficients.

The candidate supporting base with largest geometric score is selected as the best supporting base. Since the number of candidate supporting bases is six which are the six faces of the bounding box of the 3D model, the optimal supporting base can be found out effectively by our method. We can see from the Fig. 3 that the stability score and the visibility score of the Figs. 3(a) and (b) are lower than that in Fig. 3(c), so the orientation defined by the supporting base in Fig. 3(c) is selected as our final upright orientation. The model is posed at its upright orientation in Fig. 3(d).

5. Experimental results

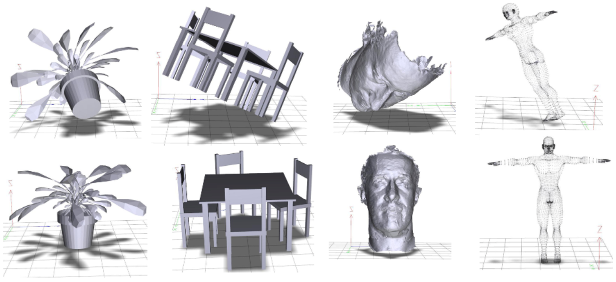
5.1 Implementation

All of our experiments are tested on a PC with Intel(R)



(a) Input model (b) Elements = 87177 (c) Elements = 392

Fig. 5. Results obtained by the three-order tensor constructed with the different ways: (a) input model; (b)-(c) results obtained by the tensor constructed with the volume and surface points of the model.



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

Fig. 6. Different types of models handled by our method: (a) non-manifold model; (b) composed model; (c) incomplete model; (d) point cloud.

Core(Tm) i3-2120 CPU @ 3.30 GHz and 8 GB memory. On average, each model takes about 1-2 minutes in our experiments.

In our paper, α , β , γ and ε are set to 0.30, 0.30, 0.40, and 0.10 respectively. The dimensions of the three-order tensor constructed are set to 100, 100 and 100 respectively. The coefficients of trace norm of the three-order tensor are $\alpha_i = 0.33$, $i = 1,2,3$.

5.2 Our results

We test about 1814 models of the Princeton database [26] which contains furniture, vehicle, humans, buildings, animals and so on, and our method can achieve about 70% prediction accuracy, while Jin et al. [8] just achieve about 55% prediction accuracy. Most of the failed models have big parts whose symmetries are inconsistent with the whole models and the rank of the big parts play the leading roles in the low-rank optimization, so the big parts of the models can be posed at the upright orientation, not the whole models by our method (see Fig. 9).

Some of the test results by our method are shown in Figs. 4-7. The results of the ant, bed and street light are shown in Fig. 4. The first column shows the input models with random directions. From the second column we can clearly see that our method can align the 3D models with axes based on the observation that the rank of the three-order tensor is the lowest if the 3D model is aligned with axes. Our proposed optimization model can align 3D models with axes automatically and efficiently. After the model has been aligned with axes, the upright orientation can be picked out easily by analyzing the geometric properties of the 3D model. All of these models are posed at their upright orientations by our method in the third column. Since the stability score of the supporting base of the ant in the third column is the largest among the six candidate supporting bases, the orientation defined by this supporting base is selected as the final upright orientation. The visibility score of the supporting base of the street light in the third column is the largest among the six candidate supporting bases, so our method selects the direction defined by this supporting base as the final upright orientation.

Fig. 7. More models posed at their upright orientations with our method. From the figure we can see that our method can not only deal with man-made models, but also non-artificial models as long as they have some kinds of symmetries.

Results obtained by the three-order tensor constructed with two different ways are shown in Fig. 5. Input model with random orientation is shown in Fig. 5(a). The results obtained by the three-order tensor constructed with volume and surface points of the model are shown in Figs. 5(b) and (c), respectively. These two results are almost the same, but the number of the elements of these two three-order tensors is 87177 and 392. That is to say, the results obtained with these two ways are the same, but the elements of the three-order tensor constructed with the surface points of the model are sparser and the memories needed to store these elements are smaller than that constructed with volume of the model.

Fig. 6 shows that our method is robust and effective for non-manifold models, point cloud, incomplete models and composed models, as we just use the points of the 3D model to construct our tensor. However, the method proposed in Ref. [8] cannot deal with these kind of models, as the projection images of these models may be incomplete or self-occlusion which cannot reflect the symmetric information of the models and can affect the performance of their algorithm seriously.

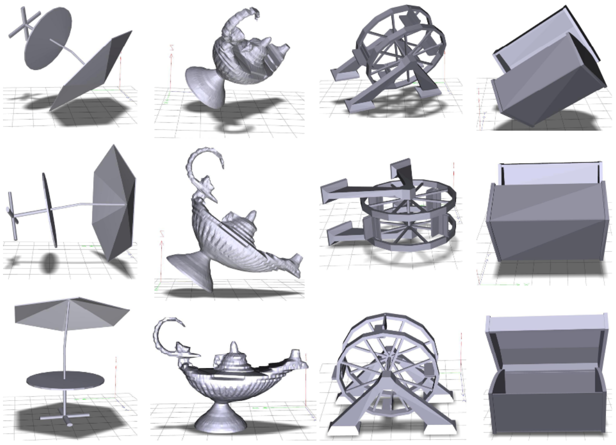


Fig. 8. Comparison results with Jin et al. [8]. The first row shows the input models with random directions, models in middle row are the results obtained by Jin et al. [8], and our method is shown in the last row.

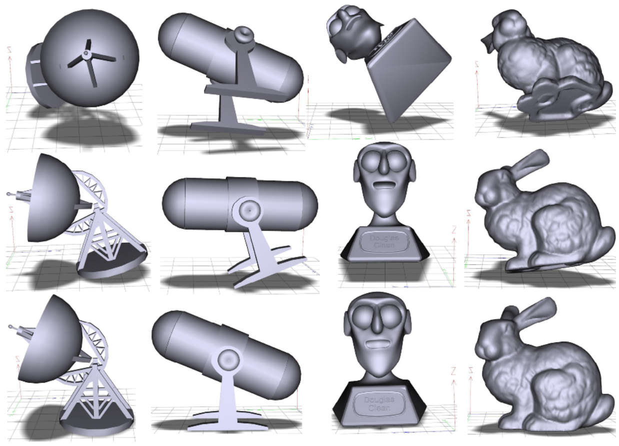


Fig. 9. Failures of our method. The first row shows the models with random orientations, the second row shows the results obtained by our method, and the last row shows the correct orientations we wanted.

More results obtained by our method are shown in Fig. 7. From this figure we can see that, our method is very flexible and it is robust and effective for all kind of models which including manifold or non-manifold models, man-made or non-artificial models, and point cloud, as long as the model have some kinds of external symmetries.

5.3 Comparisons

Since the method proposed in Ref. [7] using training set to find the upright orientation of the model, their method can handle the model as long as it has no ambiguity. However, their method is a supervised method and the performance of their method relies on the quality of the training set seriously and it is hard to implement and use.

Jin et al. [8] project the 3D model on three axes planes and then the low-rank theory of the 2D matrix is used to align the model with axes. Although their proposed method is similar with ours, their method is a local method. They need to iteratively rectify the rank of the three projected images independently which cannot guarantee the global minimum of the rank of the three-order tensor. While, we use the rank of the three-order tensor which can capture the global external symmetry of the 3D model, so our proposed method can find the better solution than using 2D matrix rank. Fig. 8 shows the comparison results. The first row shows the input models with random directions. The models shown in second row of the Fig. 8 indicate that the method proposed in Ref. [8] can only align the local of the models with axes when asymmetries or ambiguities occur. Maybe the rank of each projection image is almost the lowest, but the rank of the tensor constructed with this model is not the lowest, i.e. the whole model is not posed at its upright orientation. However, using the rank of the tensor, we can find the global solutions which are shown in third row of the Fig. 8.

6. Conclusions

In this paper, low-rank tensor theory is used to align the model with axes and then some geometric properties of the 3D model are applied to find the upright orientation from the six candidate supporting bases. We have achieved some impressive results. The experimental results demonstrate that our method is efficient and robust for manifold or non-manifold models, man-made or non-artificial models and point cloud.

Since the rank of the three-order tensor can capture the global external symmetries of the 3D models, our method is a global and unsupervised method. The proposed method is easy to implement, so it can be used as initial step of some geometry processing algorithms conveniently, such as 3D shape retrieval, 3D shape matching and 3D shape registration.

However, our method still has some limitations. First of all, when a model has a big part whose symmetry is inconsistent with the whole model and the rank of the big part plays the leading role in the low-rank optimization, the big part of the model can be posed at the upright orientation, not the whole model by our method (see the first three columns of the Fig. 9). Secondly, if a model (man-made or non-artificial) has not any external symmetry, our method may fail (see the last column of the Fig. 9). In these situations, the results obtained by the method proposed in Ref. [8] are the same as ours, i.e. their method has the same limitations.

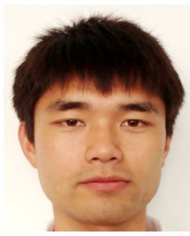
As using the low-rank tensor theory cannot deal with the model with big parts, we should try other ways to handle this problem in the future. Finding upright orientation of the models without any external symmetry will be a challenge, we will try our best to deal with this problem in our subsequent work. In this paper, genetic algorithm (GA) is used to solve our optimization problem which is not very effective sometimes. Exploring a more effective optimization method for this nonlinear optimization problem is the goal of our future research.

Acknowledgment

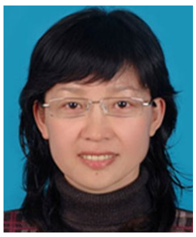
This work supported by National Natural Science Foundation of China (61222206, 61173102, U0935004) and the One Hundred Talent Project of the Chinese Academy of Sciences.

References

[1] 3D Scanning System, *http://zcorp.com/en/3d-scanning-system/spage.aspx*.
[2] K. Xu, H. L. Zheng, H. Zhang, D. Cohen-Or, L. G. Liu and Y. S. Xiong, Photo-inspired model-driven 3D object modeling, *ACM Transactions on Graphics (TOG)*, 30 (4) (2011) 80.
[3] D. W. Lee, J. K. Kim, S. R. Kim and K.-H. Lee, Shape design of a tire contour based on approximation model, *Journal of Mechanical Science and Technology*, 25 (1) (2011) 149-155.
[4] S. Biasotti, S. Marini, M. Mortara, G. Patane, M. Spagnuolo and B. Falcidieno, 3D shape matching through topological structures, *Discrete Geometry for Computer Imagery* (2003) 194-203.
[5] J. W. H. Tangelder and R. C. Veltkamp, A survey of content based 3D shape retrieval methods, *Multimedia tools and applications*, 39 (3) (2008) 441-471.
[6] P. J. Besl and N. D. Mckay, A method for registration of 3D shapes, *IEEE Trans. Pattern Anal. Mach. Intell*, 14 (2) (1992) 239-256.
[7] H. B. Fu, D. Cohen-Or, G. Dror and A. Sheffer, Upright orientation of man-made objects, *ACM Trans. Graph*, 27 (3) (2008) 42.
[8] Y. Jin, Q. B. Wu, and L. G. Liu, Unsupervised upright orientation of man-made models, *Graph. Models*, 74(4) (2012) 99-108.
[9] Z. D. Zhang, X. Liang, A. Ganesh and Y. Ma, TILT: transform invariant low-rank textures, *International Journal of Computer Vision*, 99 (1) (2012) 1-24.
[10] K. Xu, H. Zhang, A. Tagliasacchi, L. G. Liu, G. Li, M. Meng and Y. Xiong, Partial intrinsic reflection symmetry of 3D shapes, *ACM Transaction on Graphics (TOG)*, 28 (5) (2009) 138.
[11] A. D. Belegundu and T. R. Chandrupatla, *Optimization Concepts and Applications in Engineering*, Prentice Hall (1999).
[12] J. B. Luo and M. Boutell, Automatic image orientation detection via confidence-based integration of low-level and semantic cues, *IEEE Trans. Pattern Anal. Mach. Intell*, 27 (5) (2005) 715-726.
[13] S. Tong and E. Chang, Support vector machine active learning for image retrieval, *Proceedings of the ninth ACM International Conference on Multimedia* (2001) 107-118.
[14] E. Paquet, M. Rioux, A. Murching, T. Naveen and A. Tabatabai, Description of shape information for 2D and 3D objects, *Signal Processing: Image Communication*, 16 (1) (2000) 103-122.
[15] P. P. Vázquez, M. Feixas, M. Sbert and W. Heidrich, Automatic view selection using viewpoint entropy and its application to image-based modelling, *Computer Graphics Form*, 22 (4) (2003) 689-700.
[16] J. L. Wu, X. Y. Shen, W. Zhu and L. G. Liu, Mesh saliency with global rarity, *Graphical Models*, to appear (2013).
[17] P. Shilane and T. Funkhouser, Distinctive regions of 3D surfaces, *ACM Trans. Graph*, 26 (2) (2007).
[18] J. Podolak, P, Shilane, A. Golovinskiy, S. Rusinkiewicz and T. Funkhouser, A planar-reflective symmetry transform for 3D shapes, *ACM Trans. Graph*, 25 (3) (2006) 549-559.
[19] H. Yamauchi, W. Saleem, S. Yoshizawa, Z. Karni, A. Belyaev and H. P. Seidel, Towards stable and salient multi-view representation of 3D shapes, *Proceedings of the IEEE International Conference on Shape Modeling and Applications* (2006) 40-45.
[20] K. Hu, B. Wang, B. Yuan and J. Yong, Automatic generation of canonical views for CAD models, In Computer-Aided Design and Computer Graphics (CAD/Graphics), 2011 12th International Conference on (2011) 17-24.
[21] M. Ankerst, G. Kastenmüller, H. P. Kriegel and T. Seidl, 3D shape histograms for similarity search and classification in spatial database, *In Advances in Spatial Databases* (1999) 207-226.
[22] T. G. Kolda and B. W. Bader, Tensor decompositions and applications, *SIAM review*, 51 (3) (2009) 455-500.
[23] J. Liu, P. Musialski, P. Wonka and J. P. Ye, Tensor completion for estimating missing values in visual data, *Pattern Analysis and Machine Intelligence IEEE Transaction on*, 94 (2) (2009) 208-220.
[24] L. Guo, Y. Li, Y. H. Gao and J. Yang, Hole-filling by rank sparsity tensor decomposition for medical imaging, *IEICE Transactions on International and Systems*, 94 (2) (2011) 396-399.
[25] F. R. Bach, Consistency of trace norm minimization, *J. Mach Learn. Res*, (9) (2008) 1019-1048.
[26] P. Shilane, P. Min, M. Kazhdan and T. Funkhouser, The princeton shape benchmark, *Shape Modeling Application* (2004) 167-178.



Weiming Wang received a B.S. degree in Information and Computational science, Dalian University of Technology in Dalian, China in 2010. Now, he is a Ph.D. student in School of mathematical Sciences, Dalian University of Technology in Dalian, China. His research interests include digital geometric Processing and computer graphics.



Xiuping Liu received a B.S. and an M.S. degrees in Department of Mathematics, Jilin University in Jilin, China in 1987 and 1990 respectively. She received a Ph.D. in School of mathematical Sciences, Dalian University of Technology in Dalian, China in 1999. Her research interests include computational geometry, computer graphics and image.

Ligang Liu is a professor at the School of Mathematical Sciences, University of Science and Technology of China. He received his B.Sc. in applied mathematics (1996) and his Ph.D. in computer aided geometric design and computer graphics (2001) from Zhejiang University, China. Between 2001 and 2004, he worked at Microsoft Research Asia as an associate researcher. Then he worked at Zhejiang University as an associate professor and professor during 2004 and 2012. He paid an academic visit to Harvard University during 2009 and 2011. His research interests include digital geometric processing, computer graphics, and image processing. His research works could be found at his research website: http://staff.ustc.edu.cn/~lgliu/.