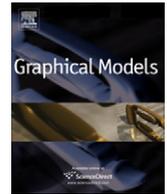




ELSEVIER

Contents lists available at SciVerse ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

Unsupervised upright orientation of man-made models

Yong Jin, Qingbiao Wu, Ligang Liu*

Department of Mathematics, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:

Received 1 March 2012

Accepted 19 March 2012

Available online xxx

Keywords:

Man-made models

Upright orientation

Low-rank matrix

Geometric analysis

ABSTRACT

Most man-made models can be posed at a unique upright orientation which is consistent to human sense. However, since produced by various techniques, digital man-made models, such as polygon meshes, might be sloped far from the upright orientation. We present a novel unsupervised approach for finding the upright orientation of man-made models by using a low-rank matrix theorem based technique. We propose that projections of the models could be regarded as low-rank matrices when they have been posed at axis-aligned orientations. The models are to be iteratively rotated by using the recently presented TILT technique, in order to ensure that their projections have optimal low-rank observations. After that, the upright orientation can be easily picked up from the six axis-aligned candidate orientations by analysis on geometric properties of the model. The approach does not require any other training set of models and should be regardless of the model quality. A number of experiments will be shown to illustrate the effectiveness and robustness of the proposed approach.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

Human usually prefer to recognize objects and models at a certain upright orientation which is determined by their eyes. Contemporary, most objects and models have their own unique support base which guarantees their standing pose against the gravity in daily life. The upright orientation is usually defined by the supporting base of the model, i.e., it can be regarded as a natural property of the model. However, 3D digital models could be posed in arbitrary orientations, since they are always produced by various modeling techniques and scanning systems which might be in customized coordinate systems. We try to provide a novel approach for automatically finding the upright orientation of the given model.

Given a 3D digital model, finding its upright orientation and posing it at the right orientation is vital for users to recognize it. In any commercial 3D geometry processing systems, such as MAYA or 3Ds Max [1], one of the basic manipulations in the systems is the rotation operation.

For input models which are not at their upright orientation, users have to rotate the models manually if they want to find the upright orientation. The procedure is usually time-consuming and inaccurate. Thus, accurate automatic techniques for finding the upright orientation of models are necessary which have been ignored by most 3D geometry processing systems, since it is not their focus point and they usually assume that input models have been already posed at their upright orientations.

Finding the upright orientation is not only profitable for human's natural views, but also useful for a number of digital geometry processing algorithms, such as shape retrieval and shape registration (Section 2). Robust performance of such algorithms relies on the consistence of orientations of the corresponding models. Our algorithm can be adopted as a pre-process of their algorithms to enhance their effectiveness and robustness.

In this paper, we focus on standing man-made models which are designed to stand on a flat supporting surface. Man-made models include most objects and models in our daily life, such as buildings, furniture, and vehicles. Such models have their certain upright orientation with respect to both human experience and natural properties of models.

* Corresponding author. Fax: +86 571 87953668.

E-mail address: ligangliu@zju.edu.cn (L. Liu).

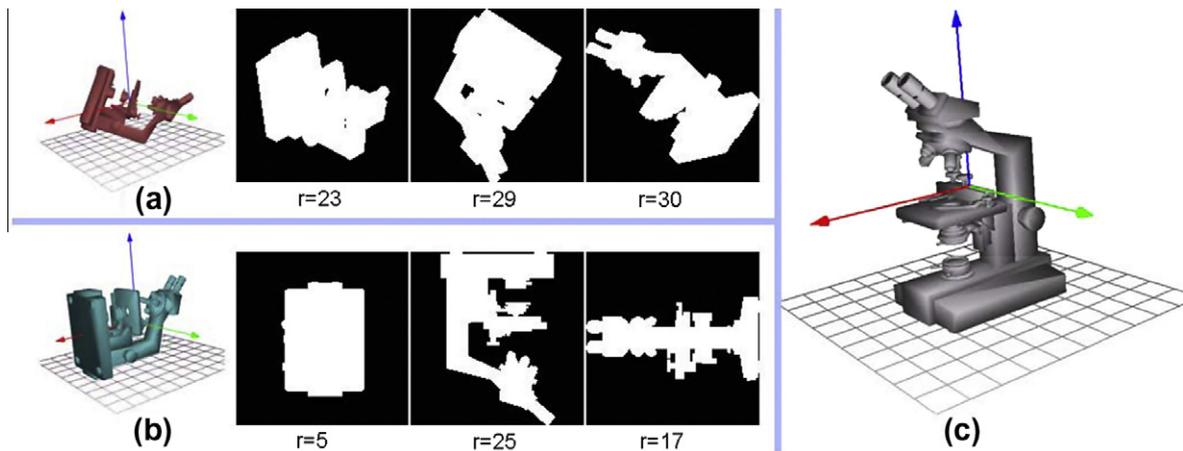


Fig. 1. (a) Input model (pink) at arbitrary orientation and its axis-aligned projections in the x - y - z (red-green-blue) coordinate system. Projections from left to right: y - z , z - x and x - y plane projection, accompanied with their corresponding matrix ranks r ; (b) axis-aligned model (cyan) and its projections with matrix ranks; (c) final model (gray) posed at the upright orientation.

We design our algorithm with respect to the following observation: If we regard axis-aligned projections of man-made models as two-dimensional matrices, when the models have been posed at axis-aligned orientations in current coordinate system, their projections could be modeled as “low-rank” matrices. In other words, ranks of projection matrices at axis-aligned orientations are lower than their counterparts at other orientations, since man-made models are mainly composed by horizontal and vertical edges and shapes. As shown in Fig. 1, ranks of projection matrices of the model posed at axis-aligned orientation in (b) are significantly lower than those of the input model at a sloped orientation in (a). Moreover, once the model has been aligned with the axes, the upright orientation as shown in Fig. 1c should be one of the six orientations determined by the six axis-aligned candidate bases, i.e., top, bottom, left, right, front and back surface of the bounding box of the model. This observation is partly inspired by the recent work of *Transform Invariant Low-rank Textures (TILT)* [23] which proposed a texture rectification technique based on low-rank matrices theorem. With this observation, we propose to iteratively rotate the model in order to make the ranks of its axis-aligned projection matrices as low as possible by performing the TILT technique on model’s projections. After that, the model will be aligned with axes in the canonical coordinate system and the two-dimensional (spherical) orientation space will be reduced to as the set of six candidate orientations. The supporting base of the upright orientation could be easily picked up from the six axis-aligned candidate bases by analysis on geometry properties of the model similar to [6].

The main contributions of our approach are summarized in the following:

- We propose an unsupervised approach for finding the upright orientation of man-made models. Different from [6], our approach does not need any training set of models.
- Our approach works for any types of models including non-manifold meshes and point clouds as only the 2D projections of the models are used.

- Our approach is simple and intuitive which can be easily used for users.

2. Related work

Several kinds of techniques in computer graphics which are relevant to our work will be briefly reviewed in this section.

2.1. Upright orientation of images

Since photos might be usually taken from capture devices at four orientations, i.e., 0° , 90° , 180° , and 270° , one kind of techniques for finding upright orientation of images attempt to classify the upright orientation from the four candidate orientations [12,20]. Such techniques transform the problem as a four-class classification which projects each orientation as a high-dimensional feature vector and computes a confidence score for each orientation by a Support Vector Machine [7], with the highest score corresponding to the best orientation.

Zhang et al. [23] aim to recognize regions in a 2D image that corresponds to a very rich class of regular patterns on a planar surface in 3D, whose appearance can be modeled as a “low-rank” matrix. By utilizing advanced convex optimization tools from matrix rank minimization, their approach is able to find an “upright” version of the image texture by computing an optimal transformation in a certain Lie Group \mathbb{G} . This work could be utilized to significantly enhance the performance of face recognition algorithms [24,21] and optical character recognition (OCR) algorithms [5,14].

2.2. Upright orientation of 3D models

Kazhdan et al. [9] propose to align 3D models into the canonical coordinate system by Principal Component Analysis (PCA). Moreover, Podolak et al. [16] consider symmetry analysis when choosing the principal axis. Symmetry detection methods [13,16] focus on how to detect meaningful symmetries in digital 3D shapes for geometric

purpose rather than how to align models into canonical coordinate system. Given the symmetry information, the model could be cut into two similar parts by the symmetry plane, however, it is hard to identify the support base from all candidate planes perpendicular to the symmetry plane.

Fu et al. [6] try to reduce the two-dimensional orientation space to a small set of orientation candidates using analysis on geometry properties of models. After that, the best orientation will be determined by a learning based approach. Nonetheless, the example-based technique requires additional training set of models, while computation for the set of orientation candidates is tedious and its robustness extremely relies on model quality. Our unsupervised approach is able to find the upright orientation of models without any additional training set of models.

2.3. View selection

Automatic viewpoint selection has also been widely researched since it is important to quickly and effectively view 3D models. Abbasi et al. [2] proposed a new method for automatic selection of optimal views of models by curvature scale space representation. Vázquez et al. [19] try to maximize the visibility of models by using viewpoint entropy theorem. Other metrics as mesh saliency [11] and shape distinction [17] have also been used to determine the optimal viewpoint. We believe that it will be easier to find the optimal viewpoint if the model has been posed at the upright orientation by using our technique.

2.4. 3D model retrieval and registration

3D model retrieval techniques [18,8] attempt to find similar shapes from databases of models and 3D model registration techniques [4,22] try to solve the problem of finding corresponding parts of multiple models. One of the main challenge of both the two techniques is to design a robust and effective method for finding the similarity between two models over the whole space of all transformations. To cover this challenge, most techniques adopt typically PCA based or manual adjustment in order to align all of the models into the same coordinate system before matching them. Since our algorithm finds the upright orientation of models while aligning the model along the coordinate axes, it is able to reduce the orientation alignment problem from the whole space of all transformations to a set of only four candidate orientations (Left, Right, Front and Back).

3. Algorithm

Since most man-made models can be represented as polygon meshes, with no loss of generality, we describe our algorithm based on polygon mesh, which can be easily extended to other digital geometry representations, such as CSG or BREP models. For clarification, we define a polygon mesh as set of its facets: $\mathcal{M} = \{f, f \in \mathcal{M}\}$. In our x - y - z coordinate system (red-green-blue¹ in all

figures in the paper), we define the positive z -axis as the upright orientation.

Following the analysis in Section 1, firstly, we will introduce how to utilize the TILT [23] technique to align input models (Fig. 1a) with axes in order to obtain axis-aligned models (Fig. 1b) in Section 3.1. After that, we will describe the algorithm of picking up the upright orientation (Fig. 1c) from set of the six candidate orientations in Section 3.2.

3.1. Align models with axes

Projections of axis-aligned models can be modeled as “low-rank” matrices, since man-made models are mainly composed by horizontal and vertical edges and shapes. The model can also be voxelized into a representation of third-order tensor matrix, consequently, the third-order tensor matrix ought to have a “low-rank” behavior as well. However, rank of third-order tensor matrix is not well defined [10]. Fortunately, we are able to utilize the TILT technique [23] to find the “low-rank” version of projection image (i.e., 2-dimensional matrix) by computing an optimal rotation transformation.

3.1.1. Rectification of projections as low-rank matrices

Consider the x - y plane projection of the model \mathcal{M} , we can binarize the projection as black and white in order to generate the projection image I with fixed resolution $N \times N$, which can also be referred as a two-dimensional matrix. To avoid affect of noise, suppose $I = U\Sigma V$ is the SVD decomposition of I , we can model I as a low-rank version L with sparse-error matrix E :

$$I = L + E, L = UT_{\delta}(\Sigma)V, \quad (1)$$

where $T_{\delta}(\cdot)$ is an operator that set all small elements less than δ in the matrix as zero. If we fix the parameter δ as δ_0 , we regard the rank of L as the no noise rank of I (all rank appeared in this paper is the no noise rank).

If the model has been posed at axis-aligned orientation, an appropriate assumption is that the projection matrix can be modeled approximately as low-rank matrix after some small imperfect parts being removed because of the axis-aligned property of man-made models. Thus, no matter which orientation the model has been posed, we are able to recover an optimal low-rank representation of x - y plane projection after applying a rigid rotation transformation around the image center:

$$I \circ R = L + E, \quad (2)$$

where $R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$ which is a counter-clockwise rotation transformation with the rotation angle θ . Refer to the TILT technique [23], this can be solved by the following problem:

$$\min_{L,E,R} \|L\|_* + \lambda \|E\|_1, \text{ s.t. } I \circ R = L + E, \quad (3)$$

where $\|\cdot\|_*$ and $\|\cdot\|_1$ are the nuclear norm (sum of all singular values) and the l_1 -norm, which are closely related to rank of matrix and sparsity of matrix respectively. $\lambda > 0$ is the weight.

¹ For interpretation of color in Figs. 1, 2–9, the reader is referred to the web version of this article.

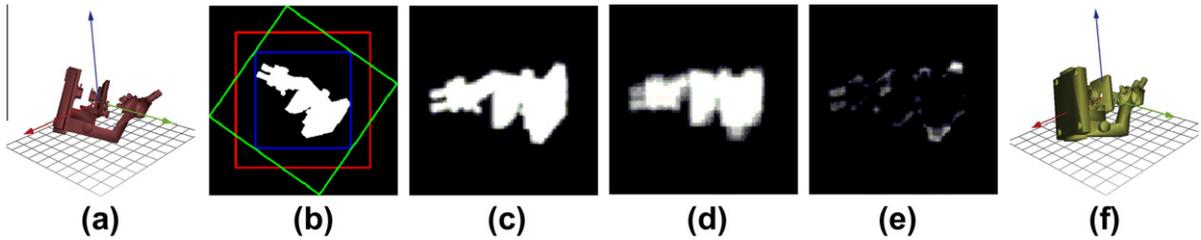


Fig. 2. Rotate the model corresponding to its rectification of x - y plane projection as low-rank matrices: (a) input model; (b) its x - y plane projection image I (Blue square). Red square: Enlarged window centered at I with rank $r = 30$. Green square: window transformed by the rotation matrix R ; (c) rectified image $I \circ R$ with rank $r = 24$; (d) The low-rank part L ; (e) The sparse error part E ; (f) The model has been rotated around the z -axis corresponding to the rotation matrix R .

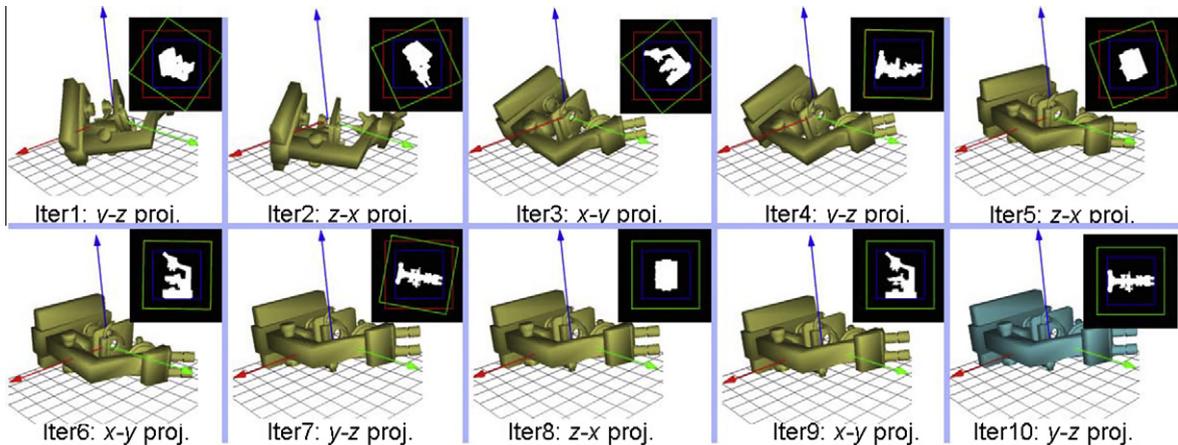


Fig. 3. Straightforward strategy: Align the model with axes by iterative rectification of axis-aligned projections as low-rank matrices in turn of the ordinary sequence iteratively. The corresponding rectification indication is also shown at the right-top of each iteration. The right bottom model has also been aligned with axes.

The optimal rotation matrix R can be computed iteratively by computing its increment ΔR from the linear discrete version of (3):

$$\min_{L, E, \Delta R} \|L\|_* + \lambda \|E\|_1, \quad \text{s.t. } I \circ R + \nabla I \Delta R = L + E, \quad (4)$$

where ∇I represents the derivatives of image w.r.t the transform parameter θ . This can be solved effectively by the Augmented Lagrangian Multiplier (ALM) method, whose details can be found in the TILT paper [23,3].

Fig. 2 has shown the procedure of rectification. Given the x - y plane projection image I (blue window in (b)), we enlarge the image to the resolution of $\sqrt{2}N \times \sqrt{2}N$ (red window in (b)) with additional black backgrounds which does not change the rank of I , in order not to discard any content of the original projection in the window which will be transformed by the rotation matrix (green window in (b)). The rectified projection image $I \circ R$ (c) is composed by its low-rank part L (d) and the sparse error part E (e). The rectified image $I \circ R$ could be modeled as low-rank matrix L with rank $r = 24$, while the original projection image I could be modeled as low-rank matrix with rank $r = 30$. Obviously, the rectified image $I \circ R$ has a better axis-aligned appearance than the original image I . We can rotate the model around the z -axis clockwise in the x - y plane with

θ corresponding to the optimal rotation matrix R , in order to obtain the rotated model (f) whose projection should be $I \circ R$. As a result, the rotated model (f) should have a better axis-aligned appearance than the input model (a).

It is similar if we want to rotate the model by rectifications of y - z and z - x projections. However, it is impossible to rectify the three projections simultaneously. Thus, we have to design an iterative rectification algorithm to decide which projection should be rectified at first and when the iteration could be terminated.

3.1.2. Iterative rectification

Our goal is to align the model with axes by iterative rectification of axis-aligned projections as low-rank matrices. Since it is impossible to rectify the three projections simultaneously, one straightforward strategy is that implement rectification based on the three projections in turn of ordinary sequence circularly. For instance, the ordinary sequence has been set as y - z plane, z - x plane, x - y plane circularly as shown in Fig. 3. However, this procedure can be optimized.

As shown in Fig. 1a, we can imagine that it is most effective to implement rectification based on x - y plane projection at first, which has the maximum rank among the three projections $D = \{x - y, y - z, z - x\}$. Moreover,

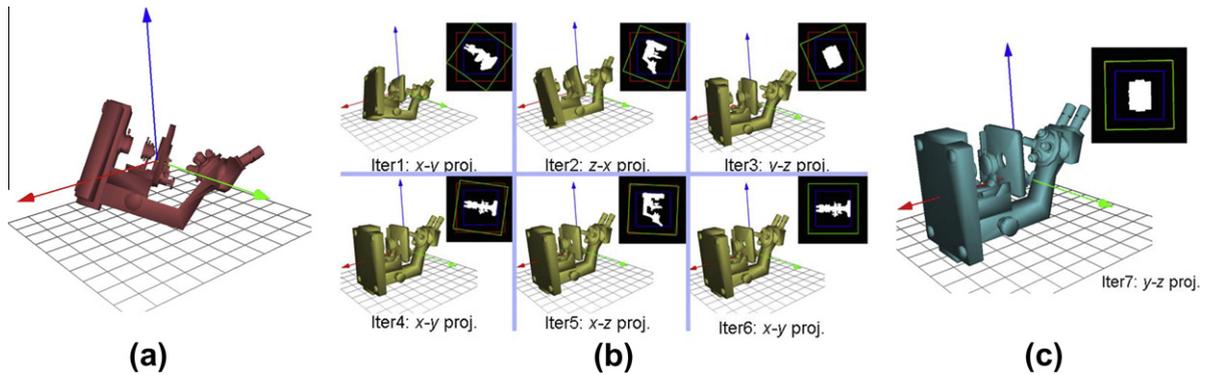


Fig. 4. Rank priority strategy: Align the model with axes by iterative rectification of axis-aligned projections as low-rank matrices with rank priority. (a) Input model; (b) iteratively rotated models by rectification; (c) final iteration and axis-aligned model.

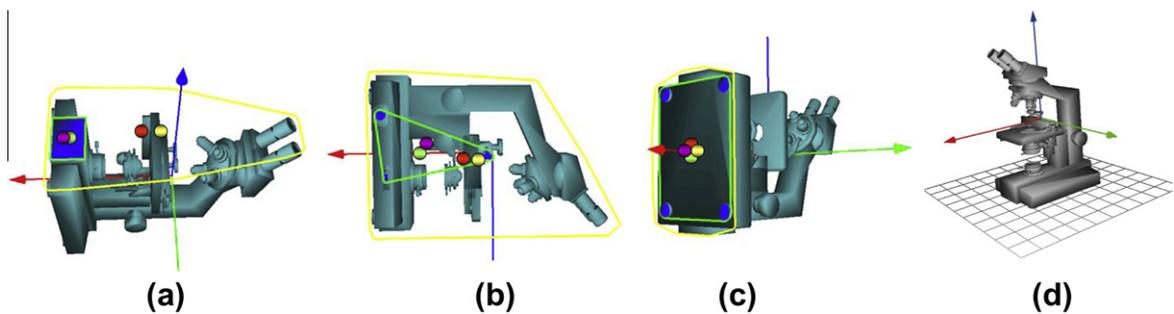


Fig. 5. Final upright orientation selection. The green polygon is the convex hull of actual supporting base (blue facets), with its center (green point). The yellow polygon is the convex hull of model projection on current candidate base, with its center (yellow point). The red point is the projected center of model mass and the purple point is the area-weighted center of actual supporting base. In (a), the projected center of model mass is out of the convex hull of actual supporting base. Current supporting base is unstable which could be pre-discarded. Supporting base (c) has better stability and symmetry than (b). (d) Model posed on supporting base (c), which has been posed at the upright orientation.

during iteration, it will be more effective to implement rectification based on the projection whose rank has been changed most.

We proposed our rank priority strategy as follows: Record history rank \bar{r}_i for each projection $i \in D$. Denote r_i as the current rank of the projection $i \in D$. We design the iteration procedure as follows:

1. Initialize the history ranks of projections $\bar{r}_i = 0, \forall i \in D$
2. Implement rectification based on the three projections respectively in turn of the priority $\Delta r_i = |r_i - \bar{r}_i|, i \in D$ (Larger value, higher priority) as following: For current projection i , compute the optimal θ , update the history rank \bar{r}_i only for current projection and rotate the model. If $\theta \geq \theta_{th}$, skip to the very beginning of step 2.

θ_{th} is the degree tolerance which has been set as 2° in our implementation. Iteration will be terminated if θ of all the 3 final rectifications in step 2 are smaller than θ_{th} .

Fig. 4 has shown detail steps of the iterative rectifications of the rank priority strategy of the microscope model. The straightforward strategy in Fig. 3 requires three more steps than the rank priority strategy, since the 4th and 6th iteration step are unnecessary which have also been implemented. The rank priority strategy has omitted unnecessary steps.

3.2. Final upright orientation selection

Once the model has been aligned with axes, the upright orientation ought to be one of the six orientations determined by the six axis-aligned candidate bases, i.e., top, bottom, left, right, front and back surface of the bounding box of the model. Moreover, most of the six candidates can be discarded by simple geometry analysis on the model similar to the work [6].

The actual supporting facets of model are not the surface of bounding box itself. Thus, for each candidate surface S of the bounding box of the model. We compute the actual base of the model as projections of part of the model facets, which are closely enough to the candidate surface and have the similar normals to the candidate surface normal, on the candidate surface:

$$P_S(\mathcal{M}) = \{P_S(f); \text{dist}(f, S) < \text{Dist}_{th}, N(f) \cdot N(S) < \text{Cos}_{th}, f \in \mathcal{M}\},$$

where $P_S(f)$ is the projection of facet f on the candidate surface S ; $N(f)$ and $N(S)$ represent the normal of f and S , respectively; $\text{dist}(f, S)$ denotes the distance between f and S ; Dist_{th} and Cos_{th} are tolerance parameters, which have been set as 0.05 (Unified Model) and 0.985 in our implementation. Since some parts of the model might be composed of “sharp” surfaces, we can discard the

candidate surfaces whose actual base $P_S(\mathcal{M})$ is an empty set. Moreover, we compute the convex hull $H(S)$ of the actual base $P_S(\mathcal{M})$ and the projection of model mass on the candidate surface $C(\mathcal{M})$. To keep the model stable against the gravity, one of the condition is that the projection of the model mass $C(\mathcal{M})$ is inside the convex hull $H(S)$ of the actual base. Thus, we check this condition for each remaining candidates and discard the candidates in which the projection is outside, such as in Fig. 5a.

After that, we compute several confidence scores based on three geometry features of the model described in [6] for the very few survived candidate bases. As shown in Fig. 5, we compute the following geometric information for each survived candidate bases for preparation:

- Convex hull of the actual base $H(S)$ (green polygon) and its center $C(H(S))$ (green point).
- Convex hull of the model projection on current candidate base $\bar{H}(S)$ (yellow polygon) and its center $C(\bar{H}(S))$ (yellow point)
- Projection of the model mass on the supporting plane $C(\mathcal{M})$ (red point)
- Area weighted center of the actual base $C(P_S(\mathcal{M}))$ (purple point)

3.2.1. Stability score

Since stability is usually better if the projected center of model mass is far away from the boundary of the convex hull of the actual base $H(S)$. Moreover, stability will be better if $H(S)$ covers most part of $\bar{H}(S)$. Thus, stability score is defined as:

$$E_a = 1.0 - \min_{0 < \theta < 2\pi} d_{in}(\theta)/d_{out}(\theta),$$

where d_{in} and d_{out} are distances from projection of the model mass $C(\mathcal{M})$ to the boundary of convex hull of the actual base $H(S)$ and convex hull of the model projection on current candidate base $\bar{H}(S)$, along a direction determined by θ .

3.2.2. Symmetry score

The four points $C(H(S))$, $C(\bar{H}(S))$, $C(\mathcal{M})$ and $C(P_S(\mathcal{M}))$ should be collinear or even consistent if the model has a

better symmetry. We compute two symmetry-related distance:

Consistent Distance: the average distance of the four points to their centroid.

Collinearity Distance: the average distance of the four points to their least-squares best-fit line.

The symmetry score E_s is defined as the average value of the Consistent distance and Collinearity Distance.

3.2.3. Visibility score

To measure visibility, we employ a similar technique to [15]. We put model on the current supporting base and render the model from five uniformly sampled view directions on the upper semi-sphere with respect to the supporting base (as bottom). We compute the ratio of visible facets as the visible score E_v .

E_a and E_v are ranged from 0 to 1, while E_s can also be normalized by the radius of the bounding sphere of the model. Higher E_a , E_s and E_v represents better stability, symmetry and visibility respectively. Unlike the learning based method [6], our candidate base set contains very few candidates, since the candidate base generates from the axis-aligned model. Thus, it is effective enough to select the candidate base which has the highest composite score:

$$E = \alpha E_a + \beta E_s + \gamma E_v, \quad (5)$$

where α , β , γ are weights.

As shown in Fig. 5, the projection of the model mass $C(\mathcal{M})$ is outside the convex hull $H(S)$ of the actual base in (a), which has been pre-discarded. The candidate base in (c) has better stability and symmetry than that in (b). Consequently, the candidate base in (c) has been selected as the supporting base of the upright orientation in (d). Fig. 6 has shown an example of importance of visibility. The cup in upright orientation in (b) should have a better visibility.

4. Experimental results and discussions

4.1. Implementation

There are a few parameters in the proposed algorithm. In our experience, the whole algorithm

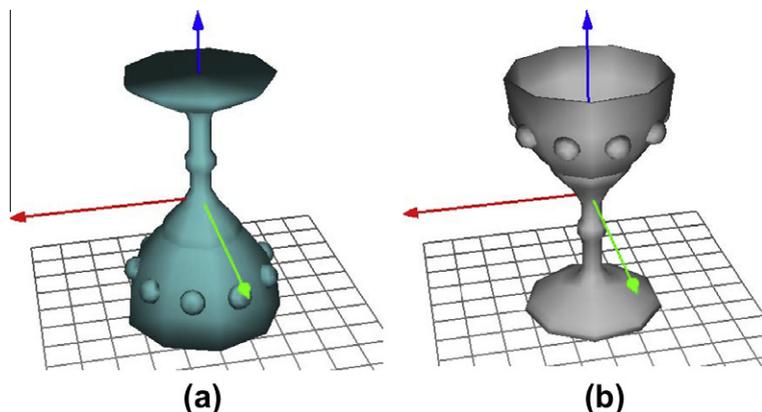


Fig. 6. An example of importance of visibility. Candidate bases in (a) and (b) have similar stability score and symmetry score. Candidate bases in (b) has a higher visible score and has been selected as the supporting base of the upright orientation.

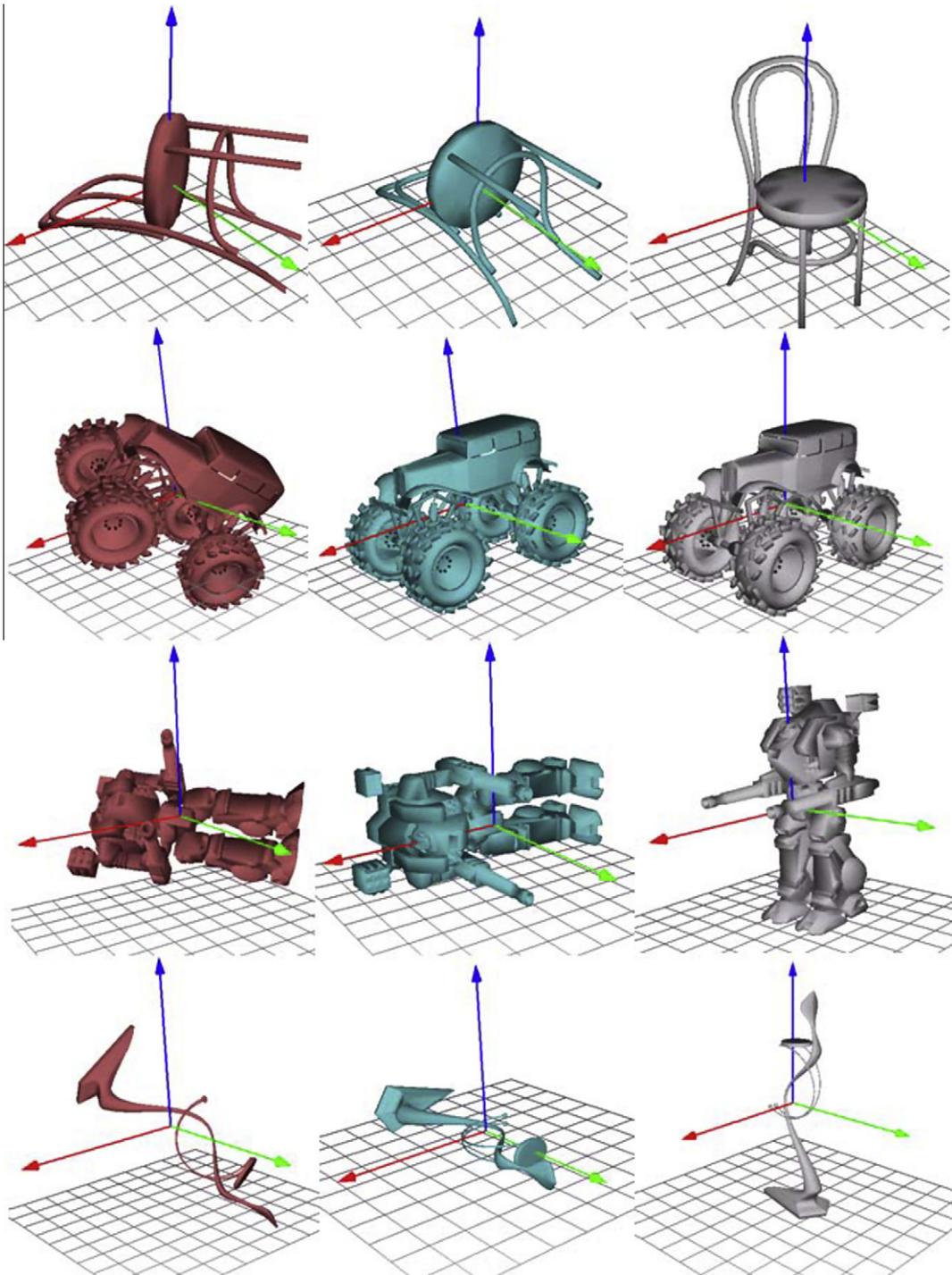


Fig. 7. Four examples of finding the upright orientation. **Left:** Input model posed at a randomized orientation; **Middle:** Model aligned with axes; **Right:** Model posed at its upright orientation.

performs effective over a large range of all the parameters. As a rule of thumb, we set the resolution parameter $N \times N$ of projections as 150×150 . We set white color as 1.0 and set black color as 0.0 in the projection matrix. Thus, the noise tolerance parameter δ_0 as 2.0, which is

able to discard most noise part during rank computation. The weight parameters in computation of confidence score E are averagely set as $\alpha = 0.3$, $\beta = 0.3$, $\gamma = 0.4$ in (5), since all the three parameters has been normalized from 0 to 1.

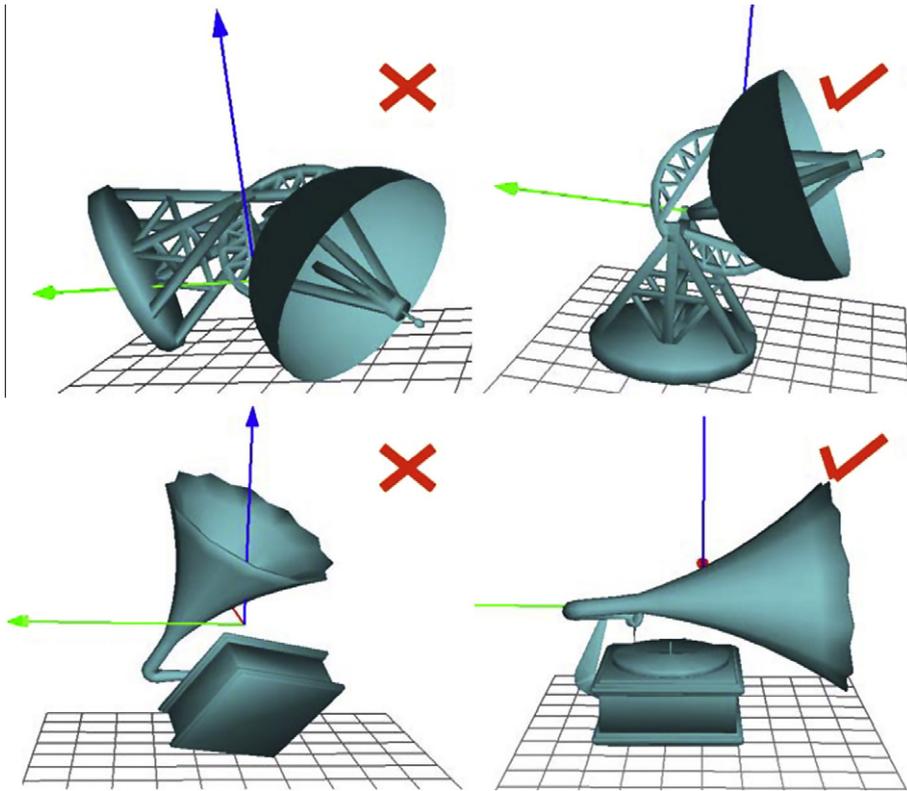


Fig. 8. Failure case: Mis-aligned model. **Left:** Models mis-aligned by using our algorithm; **Right:** Correct orientation.

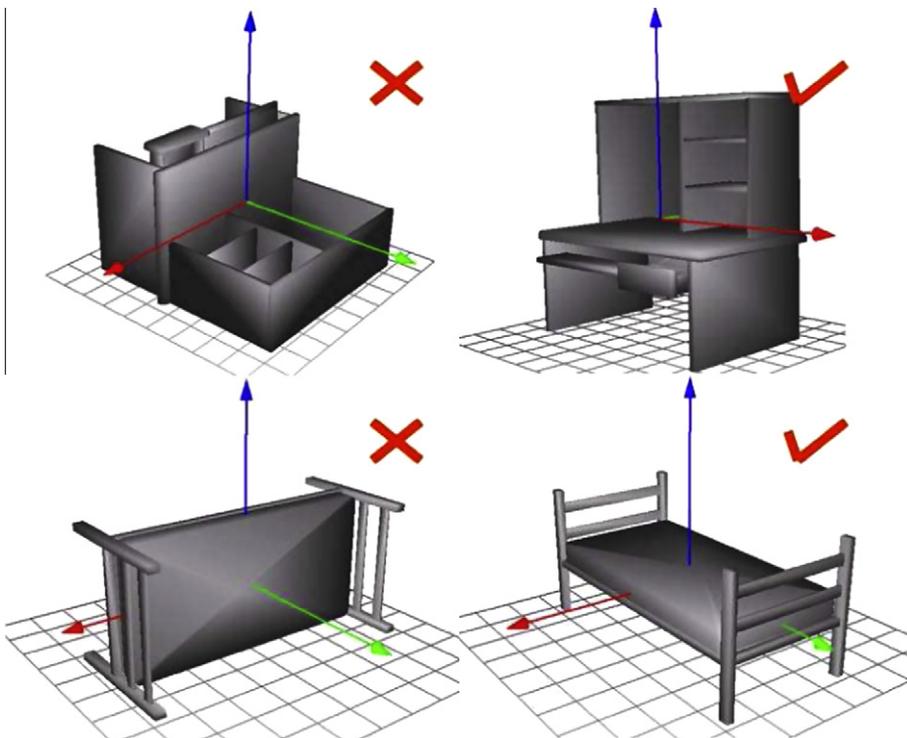


Fig. 9. Failure case: Ambiguity. **Left:** Orientation selected by our algorithm. **Right:** Correct orientation.



Fig. 10. More models which have been successfully tested through our algorithm.

All experimental results presented in this paper have been tested on a PC with DUO CPU 2.5 GHz and 2 GB memory. Computation is mainly costed during the iterative rectification. Computation of confidence score E is quickly enough to be ignored. Each step of the iterative rectification requires about 1–2 s and it requires about 5–7 iterations for most models. Consequently, it requires around 10 s to find the upright orientation of one model. The performance on timing is close to the performance of the similar work [6].

4.2. Results

In Fig. 7, four input models have been posed at a randomized orientation. We show both their axes-aligned model and model posed at its upright orientation, which has been found by our algorithm. It can be found that although the robot model in the third row is composed of multiple parts and the artwork model in the last row contains curved parts, the algorithms also works only if their projections fulfill the low-rank property. More models which have been tested successfully through our algorithm has been shown in Fig. 10.

Our algorithm succeeds to align 3D models with coordinate axes based on the observation that projections of models could be modeled as “low-rank” matrices when they have been posed at axis-aligned orientation. Models that contain main parts which are “parallel” to its support base especially fit for our algorithm. For instance (in Fig. 7), seat of the chair model is parallel to its support base. Alternatively, the four tires of the car model can be regarded as an integrated part of the model which is parallel to the support base of the car, although it is hard to assert whether an individual tyre is parallel to the support base or not. Such property severely penalties the projection

matrix rank of the model projection if the model is not aligned with coordinate axes, which lead the model to its axis-aligned orientation during iterative rectification of projections as low-rank matrices.

After aligning model with axes, the two-dimensional (spherical) orientation space for searching the support base will be reduced to as the set of six candidate orientations. It is much easier to find the support base from the set of six candidate orientations by using geometric analysis than from set of all planes of convex hull as adopted by [6].

4.3. Limitation

Although our algorithm performs quite effectively and robustly, there also exists failure cases. As shown in Fig. 8, the top left model has been mis-aligned by our algorithm since the three main part of the receptor have their own low-rank orientation which conflicts. Similarly, speaker of the bottom left model is not parallel to the supporting base, has also been mis-aligned. The two models have passed the test by technique of [6], since their learning method does not contain the align-model step. Our algorithm will fail if the model is composed by several equivalently main parts which have low-rank observation in different orientations or which are not parallel to the supporting base.

Similar to technique of [6], our algorithm is also not able to identify ambiguity, such as the two examples in Fig. 9. The algorithm is able to align such kind of models with axes, but the six-candidate orientation have similar confidence scores. In such case, our system provides an optional operation for users to select the upright orientation manually from the six candidate orientations.

4.4. Comparison

To compare our technique with [6], the advantage of [6] is that any model without ambiguity ought to fit for their technique theoretically if the training set has been properly set. On the contrary, our unsupervised algorithm does not require any additional training set of models and can be implemented independently, while the precision of their algorithm relies on the quality of the training set more or less. Our unsupervised technique might be preferred for models without priori information.

Moreover, whether a model fits for our algorithm depends on if the model contains dominant parts parallel to the supporting base, which is able to model a 'low-rank' projection matrix. On the other hand, whether a model fits for the technique of [6] mainly depends on the training set of models. It is really hard to assert which one performs better.

5. Conclusion and future work

This paper has presented an effective and robust algorithm for finding the upright orientation of man-made models. We align the model with axes by iteratively rotating it by using the recently presented TILT technique, in order to ensure that axis-aligned projections of the model have optimal low-rank observations. After that, the upright orientation could be easily selected from the six axis-aligned candidate orientations by analysis on geometric properties of the model. Our unsupervised technique does not require additional training set of models and performs effectively and relatively robust, which is valuable to be integrated into popular software systems of digital geometry processing.

It will be interesting to utilize suitable low-rank theorem of third-order tensors, which has not been widely developed yet, to better align the model with axes in our future work. Since it is easier to do symmetry analysis when models have been posed in upright orientation, it is also reasonable to make effort to utilize our algorithm framework to survey symmetry information of 3D models.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (10871178, 61070071), the Important National Science & Technology Specific Projects (2009ZX07424-001), and the 973 Program (2011CB302400, 2011JB105000).

References

- [1] Autodesk <<http://www.autodesk.com/>>, 2011.
- [2] S. Abbasi, F. Mokhtarian, Automatic view selection in multi-view object recognition, in: *Proceeding of ICPR*, 2000.
- [3] D.P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 2004.
- [4] P.J. Besl, N.D. McKay, A method for registration of 3-d shapes, *IEEE Transactions of PAMI* 14 (2) (1992) 239–256.
- [5] M. Cheriet, N. Khama, C.L. Liu, C. Suen, *Character Recognition Systems: A Guide for Students and Practitioners*, Wiley-Interscience, 2007.
- [6] H. Fu, D. Cohen-Or, G. Dror, A. Sheffer, Upright orientation of man-made objects, *ACM Transactions on Graphics* 27 (3) (2008) 1–7.
- [7] T. Hastie, R. Tibshirani, J.H. Friedman, *The Elements of Statistical Learning*, second ed., Springer, 2003.
- [8] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Three-dimensional shape searching: state-of-the-art review and future trends, *Computer-Aided Design* 37 (5) (2005) 509–530.
- [9] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3d shape descriptors, in: *Proceeding of SGP*, 2003, pp. 156–164.
- [10] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, *SIAM Review* 51 (3) (2009) 455–550.
- [11] C.H. Lee, A. Varshney, D.W. Jacobs, Mesh saliency, *ACM Transactions on Graphics* 24 (3) (2005) 659–666.
- [12] A. Lumini, L. Nanni, Detector of image orientation based on borda count, *Pattern Recognition Letters* 27 (3) (2006) 180–186.
- [13] N.J. Mitra, L. Guibas, M. Pauly, Partial and approximate symmetry detection for 3d geometry, *ACM Transactions on Graphics (SIGGRAPH)* 25 (3) (2006) 560–568.
- [14] S. Mori, C.Y. Suen, K. Yamamoto, Historical review of ocr research and development, in: *Proceedings of the IEEE*, vol. 80, 1992, pp. 1029–1058.
- [15] F.S. Nooruddin, G. Turk, Interior/exterior classification of polygonal models, in: *Proceedings IEEE Visualization*, 2000.
- [16] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, T. Funkhouser, A planar-reflective symmetry transform for 3d shapes, *ACM Transactions on Graphics* 25 (3) (2006) 549–559.
- [17] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The princeton shape benchmark, in: *Proceeding of SMI*, 2004, pp. 167–178.
- [18] J. Tangelder, R. Veltkamp, A survey of content based 3d shape retrieval methods, in: *Shape Modeling Applications*, 2004, pp. 145–156.
- [19] P.P. Vázquez, M. Feixas, M. Sbert, W. Heidrich, Automatic view selection using viewpoint entropy and its application to image-based modelling, *Computer Graphics Forum* 22 (4) (2003) 689–700.
- [20] Y.M. Wang, H. Zhang, Detecting image orientation based on low-level visual content, *Computer Vision and Image Understanding* 93 (3) (2004) 328–346.
- [21] X. Zhang, Y. Gao, Face recognition across pose: a review, *Pattern Recognition* 42 (11) (2009) 2876–2896.
- [22] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, *International Journal of Computer Vision* 13 (2) (1994) 119–152.
- [23] Z. Zhang, A. Ganesh, X. Liang, Y. Ma, Tilt: transform invariant low-rank textures, in: *The Tenth Asian Conference on Computer Vision*, 2010.
- [24] W. Zhao, R. Chellappa, A. Rosenfeld, P.J. Phillips, Face recognition: a literature survey, *ACM Computing Surveys* 27 (3) (2003) 399–458.