Context-aware Geometric Object Reconstruction for Mobile Education

Jinxin Zheng, Yongtao Wang*, Zhi Tang
Institute of Computer Science and Technology, Peking University
No.5 Yiheyuan Road, Haidian District
Beijing, China, 100871
zhengjinxin,wangyongtao,tangzhi@pku.edu.cn

ABSTRACT

The solid geometric objects in the educational geometric books are usually illustrated as 2D line drawings accompanied with description text. In this paper, we present a method to recover the geometric objects from 2D to 3D. Unlike the previous methods, we not only use the geometric information from the line drawing itself, but also the textual information extracted from its context. The essential of our method is a cost function to mix the two types of information, and we optimize the cost function to identify the geometric object and recover its 3D information. Our method can recover various types of solid geometric objects including straight-edge manifolds and curved objects such as cone, cylinder and sphere. We show that our method performs significantly better compared to the previous ones.

Keywords

3D reconstruction; geometric object; line drawing

1. INTRODUCTION

Many educational paper books, especially the textbooks for primary and secondary education, often contain a lot of illustrations of the three-dimensional (3D) geometric objects. Because these 3D geometric objects are illustrated as the two-dimensional (2D) line drawings with the loss of 3D information, sometimes it's too difficult to quickly understand the geometric objects by observing the 2D line drawings in the paper books.

This paper aims at helping the students understand the geometric objects in their books. Specifically, we recover the 3D geometric object from the 2D line drawing and its surrounding description text. Our algorithm is designed to be efficient enough to run on modern mobile devices (e.g. smart phones or tablets). By using just a mobile phone with a built-in camera, a student can view the recovered geometric object by photographing the line drawing on the book. The application of our method improves the student's experience in the mobile educational environment.

Some existing methods try to recover the 3D object from a line drawing by using its line configurations and lift the 2D line drawing into the 3D space by enforcing a variety of geometric regularities (parallelism, orthogonality, face planarity, minimal standard deviation of angles) [3, 9, 10, 11, 12, 1, 6, 7, 13]. These methods work quite well for the clear and accurate input line drawings. However, they only take into account the line drawing itself and try to recover the lost 3D information by merely guessing from the 2D positions of the vertices and the lines. Hence, the previous methods would perform very poorly when the input line drawing contains errors [18, 19].

Obviously, the inaccuracy of the line drawings is the main difficulty for the 3D reconstruction algorithms. Since it is so difficult to successfully recover an inaccurate line drawing, we seek for additional information that might help in identifying the correct object in the line drawing. Particularly, the geometric line drawings in the geometric books are usually accompanied with description text, which provides us with a hint of what the geometric object in the line drawing might be.

In this paper, we propose an algorithm to recover the 3D information of the solid geometric objects from single line drawing images taken from the geometric books. The core technical contribution of our work is the strategy to combine the geometrical information and the textual information. Moreover, we propose a solution to represent and recover some primitive curved objects (cones, cylinders and spheres). The rest of this paper is organized as follows. The related work is briefly reviewed in Section 2. An overview of the proposed method is given in Section 3, and details are introduced in Section 4. Experimental results are reported in Section 5 and conclusions are drawn in Section 6.

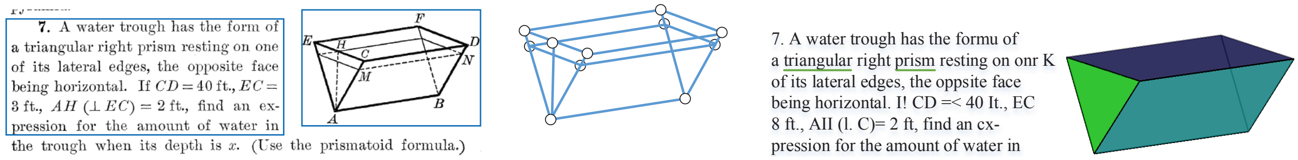
2. RELATED WORK

In the past two decades, a lot of researchers have made efforts to resolve the single line drawing-based 3D reconstruction problem. These methods include: (1) regularity-based methods which use some geometric rules as constraints to construct a cost function, and then minimize this function to obtain the 3D object [3, 9, 10, 11, 12, 1, 6, 7, 13]; (2) deduction-based methods which impose some assumptions over the input line drawing and deduce the reconstruction result based on these assumptions [4, 5, 19]; (3) divide-and-conquer-based methods which decomposes a complex line drawing into some simpler parts and conduct reconstruction

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '16, October 15–19, 2016, Amsterdam, The Netherlands.
© 2016 ACM. ISBN 978-1-4503-3603-1/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2964284.2967244



(a) Input (b) Extracted sketch (c) OCR text detection (d) Output

Figure 1: Overview of our method. (a) We take as input a line drawing image and an accompanied paragraph of description text. (b) We extract the sketch from the line drawing image and represent it as an undirected graph. The sketch might be under-complete or over-complete. The example shown is an over-complete one because of the extra lines. (c) We use OCR to detect the text from the input, and find some keywords from the text. The keywords "triangular" and "prism" are detected in this example. (d) The sketch and the keywords are combined in a formulation to estimate the pose of the object.

over each part and then combine all the partial results as the final result [2, 15, 16, 8, 17, 21, 20].

Among the methods listed above, our work is mostly related to Xue et al. [16]'s method (E3D) and our previous method (SGOR) [18]. Both of the methods have pre-defined a 3D model database from which to derive the most possible models for the input line drawing. The E3D method can handle complex objects based on divide-and-conquer approach. However, it requires the input to be a perfect line drawing. The SGOR method is more robust in that it can handle various inaccurate input line drawings. Both of them can not handle curved objects, such as cones, cylinders and spheres. In this paper, we propose a method which can handle both straight-edge manifolds and curved objects. Moreover, the context information is used to identify the correct geometric object the image contains.

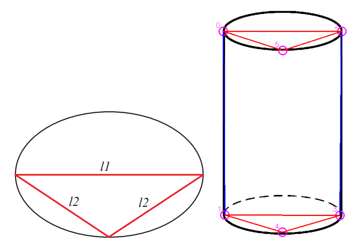
3. OVERVIEW

Our algorithm takes a line drawing image and an optional image that contains a paragraph of text which describes the geometric object as input. Currently the user needs to manually select the area of the text which is related to the geometric object in the image.

Figure 1 shows the main steps of our method. For the input line drawing image, we extract the lines from it and convert them to an undirected graph, which is called the sketch. The details of the sketch extraction can be found in [18]. Note that due to the limitation of the sketch extraction algorithm, the extracted sketch is more likely to be inaccurate (i.e. under-complete or over-complete). In addition, we detect the ellipses (corresponding to curved objects) from the image and integrate them to the sketch. For the input text image, we perform OCR to extract the text information in it. We then find the keywords that has potential relationship with the geometric object in the extracted text. Finally, we mix the sketch and the keywords in an objective formulation to obtain the most possible geometric object and its pose, based on which we render the recovered object in 3D style.

4. 3D RECONSTRUCTION

The goal of our method is to convert the line drawing images that contain the primitive geometric objects to the vectorized line drawings and present them in 3D style. We follow our previous work [18] by extracting a 2D sketch of the image firstly, and then searching for the most possible



(a) (b)

Figure 2: (a) An ellipse is represented as an isosceles triangle. The base side of the triangle is the major axis of the ellipse and is labeled as $l1$. The other two equal sides connect the two endpoints of the major axis and one endpoint of the minor axis, and they are labeled as $l2$. (b) The sketch graph of the curved object contains unlabeled edges which are the normal straight lines and the labeled edges which come from the triangle of the ellipse.

3D model in a pre-built 3D model database and compute the corresponding pose of the 3D model based on the extracted sketch.

4.1 Curved Object Models

If the line drawing contains curved objects, we use the algorithm in [14] to detect the ellipses in the image. Then each detected ellipse is represented as a triangle as shown in Figure 2(a). The triangle consists of three labeled sides: one side is the major axis of the ellipse, labeled as $l1$; each of the other two sides connects an endpoint of the major axis and an endpoint of the minor axis, labeled as $l2$. The labeled triangle is integrated into the sketch graph, as shown in Figure 2(b).

A curved object model is mostly the same as the 3D object model in [18], except that it has labeled triangles which represent the circles in the curved object model. For example, in Figure 3(a), the bottom circle is represented as the inscribed triangle whose base side is the axis of the circle and is isosceles. The three sides of the triangle are labeled as follows: the base side is labeled as $l1$, the two legs are labeled as $l2$. The labels in the curved object model are the same as the labels of the sketch graph shown in Figure 2(b). Besides the triangle, the cone model also has two unlabeled edges which represent the side face, corresponding to the 2D projection of the edges in the sketch graph. The

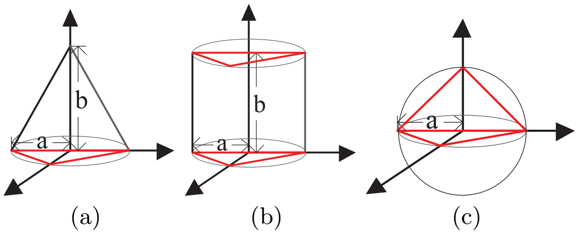


Figure 3: Examples of the curved object models. (a) Cone. (b) Cylinder. (c) Sphere.

cylinder object in Figure 3(b) is likewise. The sphere is just a bit different – as shown in Figure 3(c), it has two triangles sharing the same base side (the axis of the sphere), and the two triangles represent two perpendicular circles on the surface of the sphere, so they are perpendicular to each other in 3D space.

We impose strict labeling correspondence rule to the graph matching process: only unlabeled edges can be matched to unlabeled edges, and only the edges with the same labels can be matched(unlabeled matches unlabeled, $l1$ matches $l1$, $l2$ matches $l2$). This ensures that only curve object models can be selected if the image contains curve objects and the ellipses are detected from their 2D projections.

After the graph matching process [18], the candidate models are selected for the extracted sketch of the input image. With the triangular representation of the ellipses, the curved object models contain only straight edges (labeled and unlabeled). So the 3D reconstruction process in [18] can be easily applied to the curved objects in this paper. Figure 4 shows examples of the 3D reconstructions of curved objects.

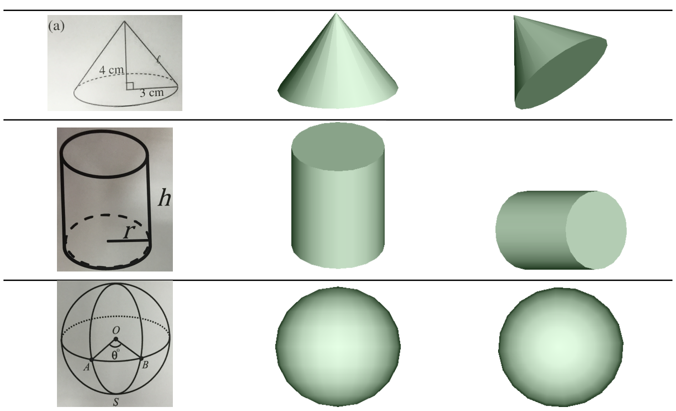


Figure 4: Examples of reconstruction results for the curved objects.

4.2 Keyword Identification

In this paper, we use the Ocropus[1] OCR algorithm to extract the text information from the images. We only identify the keywords that are related to the geometric objects, in English only. The interested keywords are listed as follows: `cuboid`, `cube`, `pyramid`, `tetrahedron`, `triangular`, `trihedral`, `quadrangular`, `parallelogram`, `parallelepiped`, `trapezoid`, `prism`, `oblique`, `frustum`, `pentagon`, `dodecahedron`, `cone`, `cylinder`, `sphere`. The extracted text information by OCR is not always correct due to the quality of the input image and the limitations of OCR algorithm. We calculate the edit distance d_{ij} between each of the interested keyword i and each word j in the space-delimited text. The keyword with the smallest edit distance and that satisfies $d_{ij} \leq 3$ is selected as the accepted keyword. We denote by k_i for each keyword identified from the input text and $\mathbf{k} = \{k_i\}$.

4.3 Formulation

Candidate models are first selected by using sub-graph isomorphism algorithm. Specifically, for a given sketch, subgraph isomorphism are performed twice in order to handle both under-complete and over-complete sketches [18]. Moreover, if a keyword k_i is detected from the input text, we add the corresponding model to the candidate models list if it is not in the list.

The objective function for a candidate model m is defined as follows:

$$F_m = G(A, R, t) + \omega_1 P(A) + \omega_2 L(m, \mathbf{k}), \tag{1}$$

where $G(\cdot)$ evaluates the geometric coherence of the candidate model and the sketch, $P(\cdot)$ favors simpler model based on the number of model parameters, and $L(\cdot)$ is an add-on prior which incorporates the keyword detection. A denotes the set of model parameters, R and t are the rotation and transpose of the model in 3D space, k is the identified keywords, ω_1 and ω_2 balances these three terms.

Geometric term.

The geometric term is defined as the projection error as follows:

$$G(A, R, t) = \frac{1}{N} \sum_{k=1}^{N} \|K(RX_{i_k} + t) - x_{j_k}\|^2, \tag{2}$$

where N denotes the number of corresponding pairs of vertices in the mapping between the model and the sketch, K is the parallel projection matrix, X_{i_k} and x_{j_k} are the coordinates of the corresponding pairs of vertices in the model and the sketch respectively.

Parametric term.

We expect the selected model to be as simple as possible among the candidate models. Obviously, the fewer parameters a model has, the simpler it is. So we define the parametric term as the number of the model parameters as follows:

$$P(A) = |A|, \tag{3}$$

Keyword prior.

The keyword prior term should favor the model that matches the detected keyword.

$$L(m, \mathbf{k}) = \sum_{k_i \in \mathbf{k}} c(m, k_i), \tag{4}$$

where

$$c(m, k_i) = \begin{cases} \sigma_1 & m \text{ matches } k_i \\ \sigma_2 & \text{otherwise} \end{cases}, \tag{5}$$

[1]https://github.com/tmbdev/ocropy

where $\sigma_2 > \sigma_1 \geq 0$ and they are constant values that encourage the models matching the keywords. In this paper we set $\sigma_1 = 0$ and $\sigma_2 = 1$ for the whole experimental data set.

Let $\mathbf{M}$ denote the set of the candidate models. We minimize the objective function for each candidate model, and select the model with the smallest optimization value, as well as the optimal model parameters, rotation matrix and transpose vector as the final result, that is:

$$\tilde{A}, \tilde{R}, \tilde{t} = \arg\min_{m \in \mathbf{M}} \{F_m\}. \tag{6}$$

4.4 Optimization

Given the coordinates of the 2D sketch $\{\mathrm{x}_1 = (x_1, y_1)^{\mathrm{T}}, \mathrm{x}_2 = (x_2, y_2)^{\mathrm{T}}, \ldots, \mathrm{x}_n = (x_n, y_n)^{\mathrm{T}}\}$, and the correspondences between the vertices of the 2D sketch and the 3D model, we find the 3D coordinates $\{\mathrm{X}_1 = (x_1, y_1, z_1)^{\mathrm{T}}, \mathrm{X}_2 = (x_2, y_2, z_2)^{\mathrm{T}}, \ldots, \mathrm{X}_n = (x_n, y_n, z_n)^{\mathrm{T}}\}$ of the vertices of the model. Specifically, the objective function is a quadratic function with an orthogonal constraint of R. We use the algorithm in [16] to obtain the value of R, A and t. With R, A and t, we are able to generate the final result in 3D style. More details of the optimization algorithm can be found in [18].

5. EXPERIMENTS

Our experimental data set consists of the traditional printed geometric paper books and online documents that contain geometric objects including books, papers, teaching materials, slide shows and other types of documents. We use different brands of smart phones under different light environments to photograph the line drawings and text paragraphs. As a result, we capture 925 line drawing images as our testing data set, among which 679 contain normal geometric objects (containing only straight lines), and 246 contain the curved geometric objects.

5.1 Experiments Setup

The matching accuracy metric used in [18] is also used in this paper to evaluate the performances of our algorithm and previous ones. The matching accuracy is defined as

$$f_a = \frac{|\mathbf{F}_{correct}|}{|\mathbf{F}|}, \tag{7}$$

where $\mathbf{F}$ denotes the test image set, and $\mathbf{F}_{correct}$ denotes the set of correctly matched images.

We evaluate the parameters ω_1 and ω_2 with our testing data set. The ω_1 parameter controls the complexity of the selected model. A low value of ω_1 favors more complicated models (with more parameters) which produce lower values of the projection error, while a high value of ω_1 favors less complicated models (with fewer parameters) which produce higher values of the projection error. Lower value of the projection error means better fitting between the model and the sketch, but is more likely to be over fitting (e.g. by setting ω_1 to 0, for a cube, the cuboid model is always wrongly selected, because it can better fit the sketch no matter the object is in fact a cube or a cuboid). In practice we set $\omega_1 = 20.0$ for the whole data set.

The ω_2 parameter controls the impact of the matched keywords. If no keyword is found, the value of ω_2 has no difference between the models. By setting $\omega_2 = 0$, we completely discard the keyword detection results. A too high value of ω_2 may increase the negative impact of mis-detected keywords. In the experiments, we set $\omega_2 = 100.0$ for the whole data set.

5.2 Comparison

A strict comparison with previous works is not possible due to the different scenario of our method. Most of the existing methods take input as a perfect line drawing with only straight lines, while our method takes input as a roughly extracted line drawing which may contain some primitive curved objects and is possibly inaccurate, together with an image containing the context information. To provide a rough comparison (please note that this comparison is not strictly conducted due to the previous reasons. It can be seen as a reference to the performance of our method.), Table 1 shows the results of our method (with/without the keyword detections) and two previous methods (E3D [16] and SGOR [18]) on our testing data set. As we can see that the matching accuracy of our method is significantly higher than that of E3D and SGOR. The reason why E3D performs so poorly is that, as we have known, it can only handle complete sketches while our method can also handle incomplete or over-complete sketches. And in our experiment, we find that, for most of the testing line drawing images, the extracted sketch is inaccurate (i.e. incomplete or over-complete). Therefore, our method outperforms E3D significantly. The SGOR performs much better in terms of matching accuracy as it can handle inaccurate sketches. However, it can only handle the straight line drawing sketches (so does E3D) while our method can also handle the curved object sketches, plus that our method employs the contextual information, therefore it's still inferior to our method.

Table 1: Comparison between our method and previous ones

Method	Correct match	Incorrect match	Accuracy
E3D	126	799	13.6%
SGOR	537	398	57.4%
Ours w.o. keywords	675	250	72.9%
Ours	752	173	81.2%

6. CONCLUSION

We have presented a method to recover the 3D information of the solid geometric object from single line drawing image in the presence of contextual information. The core of our method is a formulation that incorporates the geometric information and the textual information. Extensive experimental results demonstrate that our method can achieve significantly better performance than the previous methods. In the future we plan to recover more complex line drawings, not just limited to the primitive geometric objects.

Acknowledgments

We'd like to thank the anonymous reviewers for providing encouraging and valuable comments to our work. This work is supported by National Natural Science Foundation of China under Grant 61300061 and Beijing Natural Science Foundation under Grant 4132033.

7. REFERENCES

[1] E. Brown and P. S. Wang. Three-dimensional object recovery from two-dimensional images: a new approach. In *Photonics East'96*, pages 138–147. International Society for Optics and Photonics, 1996.

[2] Y. Chen, J. Liu, and X. Tang. A divide-and-conquer approach to 3d object reconstruction from line drawings. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[3] Y. G. Leclerc and M. A. Fischler. An optimization-based approach to the interpretation of single line drawings as 3d wire frames. *International Journal of Computer Vision*, 9(2):113–136, 1992.

[4] Y. T. Lee and F. Fang. 3d reconstruction of polyhedral objects from single parallel projections using cubic corner. *Computer-Aided Design*, 43(8):1025–1034, 2011.

[5] Y. T. Lee and F. Fang. A new hybrid method for 3d object recovery from 2d drawings and its validation against the cubic corner method and the optimisation-based method. *Computer-Aided Design*, 44(11):1090–1102, 2012.

[6] H. Lipson and M. Shpitalni. Optimization-based reconstruction of a 3d object from a single freehand line drawing. *Computer-Aided Design*, 28(8):651–663, 1996.

[7] J. Liu, L. Cao, Z. Li, and X. Tang. Plane-based optimization for 3d object reconstruction from single line drawings. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):315–327, 2008.

[8] J. Liu, Y. Chen, and X. Tang. Decomposition of complex line drawings with hidden lines for 3d planar-faced manifold object reconstruction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1):3–15, 2011.

[9] J. Liu and Y. T. Lee. Graph-based method for face identification from a single 2d line drawing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(10):1106–1119, 2001.

[10] J. Liu, Y. T. Lee, and W.-K. Cham. Identifying faces in a 2d line drawing representing a manifold object. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(12):1579–1593, 2002.

[11] J. Liu and X. Tang. Evolutionary search for faces from line drawings. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6):861–872, 2005.

[12] T. Marill. Emulating the human interpretation of line-drawings as three-dimensional objects. *International Journal of Computer Vision*, 6(2):147–161, 1991.

[13] C. Tian, M. Masry, and H. Lipson. Physical sketching: Reconstruction and analysis of 3d objects from freehand sketches. *Computer-Aided Design*, 41(3):147–158, 2009.

[14] Y. Wang, Z. He, X. Liu, Z. Tang, and L. Li. A fast and robust ellipse detector based on top-down least-square fitting. *Computers & Electrical Engineering*, 40(4):1415–1428, 2014.

[15] T. Xue, J. Liu, and X. Tang. Object cut: Complex 3d object reconstruction through line drawing separation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1149–1156. IEEE, 2010.

[16] T. Xue, J. Liu, and X. Tang. Example-based 3d object reconstruction from line drawings. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 302–309. IEEE, 2012.

[17] L. Yang, J. Liu, and X. Tang. Complex 3d general object reconstruction from line drawings. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1433–1440, Dec 2013.

[18] J. Zheng, Y. Wang, and Z. Tang. Recovering solid geometric object from single line drawing image. *Multimedia Tools and Applications*, pages 1–22, 2015.

[19] J. Zheng, Y. Wang, and Z. Tang. Solid geometric object reconstruction from single line drawing image. In *International Conference on Computer Graphics Theory and Applications (GRAPP 2015)*, 2015.

[20] C. Zou, S. Chen, H. Fu, and J. Liu. Progressive 3d reconstruction of planar-faced manifold objects with drf-based line drawing decomposition. *Visualization and Computer Graphics, IEEE Transactions on*, PP(99):1–1, 2014.

[21] C. Zou, H. Yang, and J. Liu. Separation of line drawings based on split faces for 3d object reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 692–699, June 2014.