Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Database-assisted 3D reconstruction and printing of a mechanical key from a single photograph

Yicang Huang[1, †], (Member, IEEE), Yikai Zhang[2, †], Hui Li[1, 2], (Senior Member, IEEE), Shengnan Shen[1, 2], (Senior Member, IEEE), and Feng Li[3]

[1]School of Power and Mechanical Engineering, Wuhan University, Wuhan 430072, China
[2]Institute of Technological Sciences, Wuhan University, Wuhan 430072, China
[3]School of Electrical and Electronic Engineering, Wuhan Polytechnic University, Wuhan 430023, China

Corresponding author: Hui Li (li_hui@whu.edu.cn)

[†]These authors contributed equally to this article.

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1704600, and in part by the Hubei Provincial Natural Science Foundation of China under Grant 2020CFA032.

ABSTRACT Reverse engineering is a powerful tool for duplicating real objects from data acquired by tracking systems, industrial X-ray computed tomography (CT), laser scanners, and image captures. This work proposed a novel method using the three-dimensional (3D) reconstruction and 3D printing to duplicate a mechanical key based on only one single naturally-captured photograph. A reverse engineering system of the mechanical key was self-developed which includes functions of projective transformation, curve extraction, point clouds generation, and 3D geometry reconstruction. The database technology was introduced into the image-based 3D reconstruction process for the first time to store basic dimensional information of key blanks, which provided supplementary information during the 3D key reconstruction process. Five different types of duplicated keys verified the effectiveness of the proposed method. The maximum geometric deviation was less than 0.39 mm. The duplicating time cost was 5 minutes for 3D geometry reconstruction and 18.3 minutes for 3D printing.

INDEX TERMS Database technology, image processing, key duplicating, 3D reconstruction

I. INTRODUCTION

Mechanical lock is one of the oldest tools that secure our properties and homes. The possession of the associated keys ensures that only the holder will be able to unlock it, but it is sometimes necessary to get duplicates of a key. The most common way to do this is to cut a key blank in a duplicator with an original key as the reference. With the rapid development of image-based three-dimensional (3D) reconstruction technology, it is practical to create a 3D model of the key from a single photograph with computer-vision and image processing technologies.

There are many investigations for designing 3D models from images or sketches. Gingold *et al.* [1] proposed a system for 3D modeling of free-form surfaces from two-dimensional (2D) sketches. Their system was able to create a variety of models with existing primitives and annotations. Chen *et al.* [2] introduced an interactive technique for manipulating simple 3D shapes, by extracting them from a single photograph and combining human cognitive abilities with the computational accuracy of the machine. Lau *et al.* [3] created a system that provided a 2D interface for the user to design an object from a single photo by sketching the outline of the new object. Igarashi *et al.* [4] presented a fast sketching interface for easily designing free-form models, such as stuffed animals and other rotund objects. The user drew several 2D freeform strokes interactively on the screen and the system

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3049996, IEEE Access

automatically constructed plausible 3D polygonal surfaces. Several other systems design 3D models by converting 2D sketches [5]–[7] or 3D curves [8], [9] to 3D models. Liu *et al.* [10] proposed a 3D image reconstruction method for spinning cone-shaped targets, which by taking full advantage of the similarity of the 2D imaging result with the target mapping on the 2D imaging plane. Shalaby *et al.* [11] introduced a new model to reconstruct the 3D image from any 2D image by using the marching cubes algorithm. And Chen *et al.* [12] proposed a novel 3D shape reconstruction method based on maximum correntropy Kalman filtering. In recent years, neural networks are introduced into 3D reconstruction. For example, Cai *et al.* [13] proposed an end-to-end approach to reconstruct a 3D model from a multi-view image set. Mandikal *et al.* [14] proposed the 3D-LMNet being mainly used for latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image. And Henderson *et al.* [15] solved the single-image-based 3D reconstruction by a unified framework which can be trained only by 2D images without annotations. And Hou *et al.* [16] realized the function of 3D reconstruction in canonical co-ordinate space from arbitrarily oriented 2D images. In terms of the effectiveness of 3D reconstruction, Chibane *et al.* [17] introduced implicit functions into 3D image reconstruction. Yu *et al.* [18] presented a data-driven point cloud upsampling technique to generate a denser and uniform set of points from a sparser set of points.

Some efforts reconstructed the 3D model of mechanical keys from 2D images. Laxton *et al.* [19] proposed the prototype system ("sneaky") for duplicating a key from an image by extracting the complete and precise biting code via optical decoding and the cutting of precise duplicates in a code-cutting machine. However, the biting code is not applicable to other types of keys. Itthisupornrat and Kiatsunthorn [20], [21] duplicated keys in a mini CNC machine with CNC G-code associated with key grooves from the cross-sectional image and key cuts from the top-sectional image. However, their images were captured in a specially designed image capture equipment, which is equipped with complex fixtures and light source with adjustable direction and illuminance. In the above systems, the key duplicator machine and key blanks were required to manufacture the actual key. Smith and Burghardt [22] proposed a DeepKey architecture, which can duplicate a key from a single photograph taken in daily scenes. However, DeepKey was only applicable for the zigzag key due to that all basic dimensions are fixed except for the biting segmentation.

This work proposes a novel method to reconstruct a 3D mechanical key based on only one single naturally-captured photograph. The main works and contributions are as follows:

1) A novel image-based key duplication system with functions of 3D geometry reconstruction and 3D printing is proposed, which is more robust than existing methods due to its wide applicability to various types of keys and low requirements for devices and shooting conditions.
2) Database technology is introduced into the single-image-based 3D key reconstruction process, which enables the system capable to reconstruct a 3D key using one single 2D photograph which lacks the depth information. And its extensibility makes the system capable to handle various types of keys.
3) A high-accuracy 3D key geometry reconstruction method is presented based on the database, projection transformation, curve extraction, and point clouds generation methods. Its effectiveness is validated by experimental tests.

A description of the proposed method is provided in *Section 2*. In *Section 3*, experimental tests are performed to evaluate the effectiveness of the proposed method. The conclusion is drawn in *Section 4*.

II. SYSTEM IMPLEMENTATION

Figure 1 shows the workflow of the proposed reverse engineering system for duplicating a mechanical key from a single naturally-captured photograph. An SQL database is established to store the basic dimensions of the key blank and the reference key photograph corresponding to each type of the key. The whole process from loading a single key photograph to obtaining a printed key is divided into two main stages: image processing and 3D reconstructing. In the image processing stage, a photograph of the target key is projectively transformed by referring to the image in the database using a projection transformation method. Sample points on the key curves are then extracted from the transformed image and stored as coordinates in float format. In the 3D reconstructing stage, the sample points are refined to generate point clouds of the key through a self-developed point clouds generation code. 3D geometry of the key is eventually created based on the point clouds and printed by a 3D printer. The details of the process are described in the following sections.

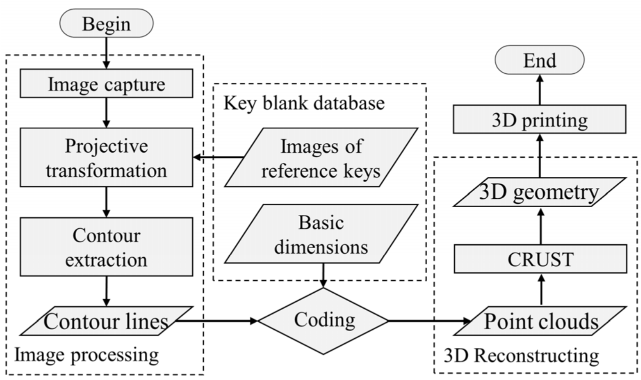


FIGURE 1. The workflow of the reverse engineering system.

A. KEY BLANK DATABASE

Many conventional 3D reconstruction methods, such as Multi-View Stereo (MVS), require multiple images from enough viewpoints to reconstruct accurate 3D models, which means that an image acquisition system with specially designed illumination modules is needed [23], [24]. Hata *et al.* [25] improved the Patch-based MVS (PMVS) to reconstruct the 3D model from a limited number of viewpoints. However, the high requirement of technical apparatus remains.

To reconstruct the 3D geometry of the mechanical key based on only one single naturally-captured photograph, this work introduces an SQL database to store the reference images and fixed basic dimensions of different keys. It is known that the difference among the keys of the same type lies in the curve shapes of key cuts. Therefore, the SQL database stores the basic dimensions of key blanks, which will be used in the point cloud generation process. As shown in Figure 2, the stored basic dimensions include the length (l), width (w), thickness (t), and rib thickness (t_r) of the function part. The basic dimensions are stored in float format. The reference images are stored in JPG format and their file paths are stored in ntext format. Those images will be used as the reference in the projective transformation process. The names of key types are stored in ntext format and will be used to help the system match the target key photograph with the stored reference image and call basic dimensions from the SQL database.

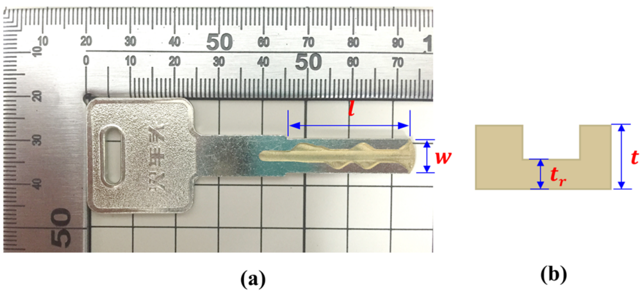


FIGURE 2. The reference key photograph and basic dimensions stored in the database (a) reference photograph and (b) the cross-section of a key blank.

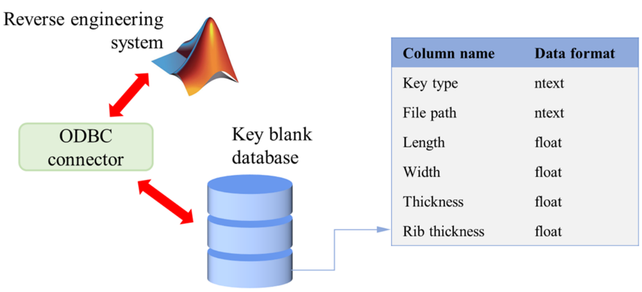


FIGURE 3. Communications between the reverse engineering system and the key blank database.

As shown in Figure 3, the database in the system is handled by the MATLAB Database Toolbox (Philadelphia, PA, USA), and the communication between the reverse engineering system and the key blank database is done via an ODBC connector.

Considering the flexibility and extensibility of the database, the system is capable to handle various types of keys. When a new type of key needs to be duplicated, the database can be extended by adding a relevant reference image and basic dimensions of the associated key blank.

B. PROJECTIVE TRANSFORMATION

To duplicate a key, the precise curves of key cuts must be extracted. The projective transformation method is adopted to correct the distortion and scale in the naturally-captured key photograph. It transforms points into points and lines into lines to preserve the cross ratio. Given a set of control points $p_i\,(x_i,\, y_i, 1)$ in the key photograph and corresponding points $p_i'(x_i,\, y_i, 1)$ in the reference image, the mapping of p_i to p_i' is described by:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{1}$$

where $\boldsymbol{T}$ is the projectivity with eight unknowns $A, B, \ldots, H$. According to equation (1), it can be obtained that

$$Ax_i + By_i + C = u_i(Gx_i + Hy_i + 1) \tag{2}$$

$$Dx_i + Ey_i + F = v_i(Gx_i + Hy_i + 1) \tag{3}$$

Then, two linear equations with the eight unknowns are obtained as:

$$Ax_i + By_i + C - Gx_iu_i - Hy_iu_i = u_i \tag{4}$$

$$Dx_i + Ey_i + F - Gx_iv_i - Hy_iv_i = v_i \tag{5}$$

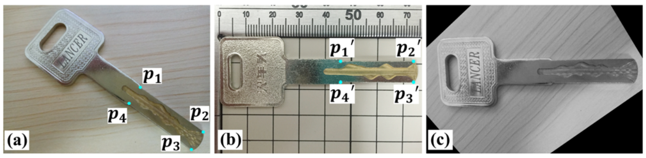


FIGURE 4. Process of projective transformation (a) key photograph labeled with control points, (b) reference image labeled with corresponding points, and (c) the transformed photograph.

Therefore, the computation of projectivity $\boldsymbol{T}$ requires four pairs of points, p_i and p_i', $i = 1, \ldots, 4$, to yield eight linear equations for solving the eight unknowns $A, B, \ldots, H$. To obtain the precise key curves, the four pairs of control points are located at the corners of the function part, as shown in Figure 4 (a) and (b). With these points, the projective transformation method is performed by solving $\boldsymbol{T}$ to obtain the transformed image, as shown in Figure 4 (c).

C. CURVE EXTRACTION

As many tousled curves will be produced with current curve extraction algorithms due to the complicated background in a naturally-captured photograph [26]. The reflective metal surface, bad key-surface quality, and low-light conditions will increase the difficulty for key-curve extraction. Therefore, we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3049996, IEEE Access

manually extract key curves from key photographs. Consequently, such manual operation lowers the requirement for the photo acquisition device.

As shown in Figure 5, three points (O, P, and Q) at the corners of the function part are selected as reference points to create the invisible plane coordinate system. The X coordinate of P is the length of the function part (l) while the Y coordinate of Q is the width of the function part (w). The key curves are extracted by selecting sample points along the cut lines, and the local density of sampling is properly chosen to ensure the precision of the extracted key curves. High-density sampling is required for curves with high curvature, while low-density sampling can be used for straight lines. The sample point coordinates are stored as $(x_m, y_m, m = 1, 2, 3, \dots, n)$.

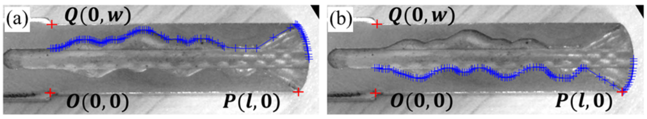


FIGURE 5. Contour extraction of (a) the upper curve and (b) the lower curve of key cuts.

Considering the non-uniformity of the selected sample points, a linear interpolation method is adopted to refine the density of these samples. The X coordinate of the ith refined point is

$$x_{r_i} = (i-1) * s,\ i = 1, 2, 3, \dots, (\frac{l}{s} + 1) \tag{6}$$

where s is the intervals of the refined points and its value determines the density of the generated point clouds. Considering both the computing resources and the error caused by the projective transformation, curve extraction, and 3D printing, s is set to 0.02 mm to meet the accuracy requirement of key duplication and save calculation time.

The Y coordinates of the ith point on the curves are calculated by:

$$\begin{cases} y_{r_i} = y_\alpha, & \text{if } \exists x_\alpha \in \{x_m\}, x_{r_i} = x_\alpha \\ y_{r_i} = \left(\frac{y_{\beta+1} - y_\beta}{x_{\beta+1} - x_\beta}\right) * (x_{r_i} - x_\beta) + y_\beta, & \\ & \text{if } \forall x \in \{x_m\}, x_{r_i} \neq x \end{cases} \tag{7}$$

where β makes $(x_{\beta+1} - x_{r_i}) * (x_\beta - x_{r_i}) < 0$, which indicates that the $(\beta + 1)$th and βth sample points are closest on both sides of the ith refined point.

The coordinates of the first and last refined points are calculated by the following formulas because the first and last refined points are not always in the middle of two sample points. The first refined point on each key curve is calculated by:

$$\begin{cases} y_{r_1} = y_1, & \text{if } \forall x \in \{x_m\},\ x_{r_1} \le x, \\ y_{r_1} = \frac{y_{\beta+1} - y_\beta}{x_{\beta+1} - x_\beta} + y_\beta, & \text{if } \exists x \in \{x_m\}, x_{r_1} > x, \end{cases} \tag{8}$$

The last refined point on each key curve is calculated by:

$$\begin{cases} y_{r_{\frac{l}{s}+1}} = y_n, & \text{if } \forall x \in \{x_m\},\ x_{r_{\frac{l}{s}+1}} \ge x, \\ y_{r_{\frac{l}{s}+1}} = \frac{y_{\beta+1} - y_\beta}{x_{\beta+1} - x_\beta} + y_\beta, & \text{if } \exists x \in \{x_m\}, x_{r_{\frac{l}{s}+1}} < x, \end{cases} \tag{9}$$

Figure 6 shows the comparison between the extracted key curves with the original key shape, which is extracted from the top-view photograph of the key. Most of the extracted curves agree well with the key shape, and the deviations mainly occur around curves with high curvature. The maximum deviation is only 0.26 mm, which meets the precise requirement for duplicating keys. Therefore, it is proved that the projective transformation and curve extraction methods are practical to extract accurate key curves from photographs.

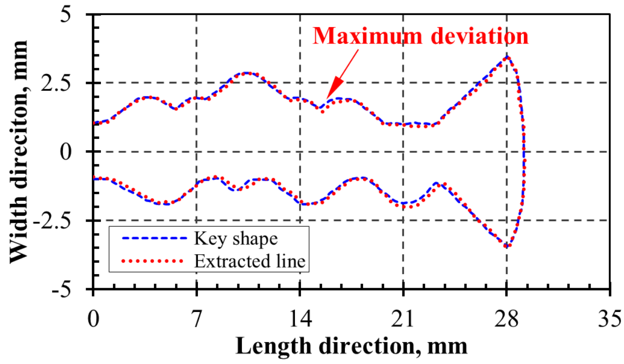


FIGURE 6. Comparison of the extracted and original key curves.

D. POINT CLOUDS GENERATION

A traditional reverse engineering process reconstructs a 3D geometry based on point clouds obtained from the 3D scanning process, which requires professional 3D scanning devices [27]. In this work, the point clouds of the key are generated using a self-developed MATLAB code based on coordinates of refined points and basic dimensions of the key blank stored in the database including the length (l), width (w), thickness (t), and rib thickness (t_r) of the function part of the key.

As shown in Figure 7, a mechanical key consists of the handle and function part. The handle part is simplified into a cuboid. Figure 8 shows the schematic of the proposed method for generating point clouds of the target key. 2D polygonal sections consisted of equal interval points are generated one by one to create the 3D point clouds of the key.

In Figure 7, the coordinates of refined points on the upper curve (RP_u) and lower curve (RP_l) are used to generate related edges of the polygon with the width (w), thickness (t), and rib thickness (t_r) of the function part. For the ith polygonal section of part 1, the coordinates of the jth point on related edges are given as:

$$\text{Edge } A\text{:}\ x = x_{r_i},\ y = 0,\ z = \frac{t}{s} j \tag{10a}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3049996, IEEE Access

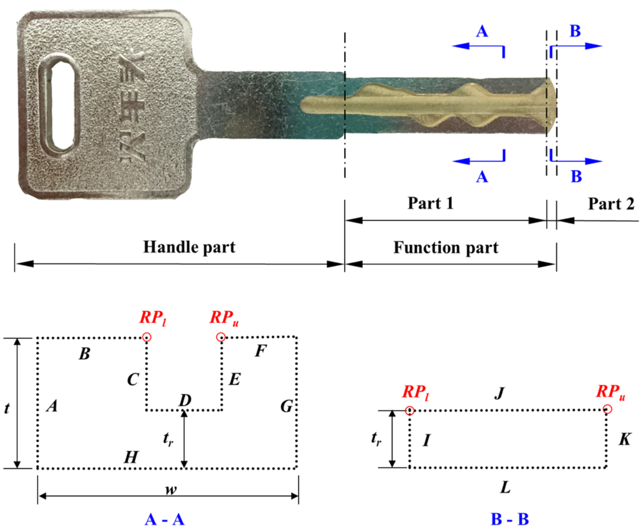


FIGURE 7. Parts of the mechanical key with different polygonal sections.

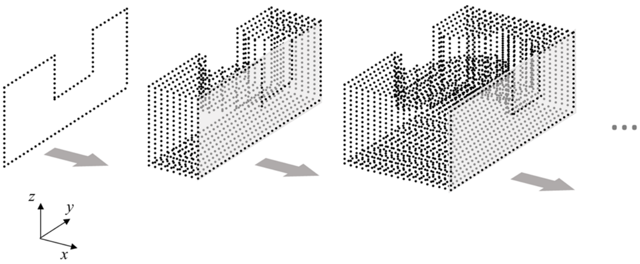


FIGURE 8. The schematic of the process for generating point clouds.

$$\text{Edge } B\text{: } x = x_{r_i},\ y = \frac{w}{s} j,\ z = t \tag{10b}$$

$$\text{Edge } C\text{: } x = x_{r_i},\ y = y_{lr_i},\ z = t_r + \frac{t - t_r}{s} j \tag{10c}$$

$$\text{Edge } D\text{: } x = x_{r_i},\ y = y_{lr_i} + \frac{y_{ur_i} - y_{lr_i}}{s} j,\ z = t_r \tag{10d}$$

$$\text{Edge } E\text{: } x = x_{r_i},\ y = y_{ur_i},\ z = t_r + \frac{t - t_r}{s} j \tag{10e}$$

$$\text{Edge } F\text{: } x = x_{r_i},\ y = y_{ur_i} + \frac{w - y_{ur_i}}{s} j,\ z = t \tag{10f}$$

$$\text{Edge } G\text{: } x = x_{r_i},\ y = w,\ z = \frac{t}{s} j \tag{10g}$$

$$\text{Edge } H\text{: } x = x_{r_i},\ y = \frac{w}{s} j,\ z = 0 \tag{10h}$$

Part 2 has different cross-sections from part 1 because its material on the rib has been milled out. For the ith polygonal section of part 2, the coordinates of the jth point on the related edges are presented by:

$$\text{Edge } I\text{: } x = x_{r_{i+d}},\ y = y_{lr_i},\ z = \frac{t}{s} j \tag{10i}$$

$$\text{Edge } J\text{: } x = x_{r_{i+d}},\ y = y_{lr_i} + \frac{y_{ur_i} - y_{lr_i}}{s} j,\ z = t_r \tag{10j}$$

$$\text{Edge } K\text{: } x = x_{r_{i+d}},\ y = y_{ur_i},\ z = t_r + \frac{t - t_r}{s} j \tag{10k}$$

$$\text{Edge } L\text{: } x = x_{r_{i+d}},\ y = y_{lr_i} + \frac{y_{ur_i} - y_{lr_i}}{s} j,\ z = 0 \tag{10l}$$

where d is the ordinal number of the division line of parts 1 and 2. Using the above point clouds generation algorithms, the point clouds of a typical key are created when all sections are generated, as shown in Figure 9 (a). With the point clouds, the well-known CRUST algorithm [28, 29], from the Delaunay triangulation [30] and Voronoi diagrams [31], [32], is adopted to reconstruct surfaces of the target key (in Figure 9 (b)). The STL file of the key is then obtained from the surface data using the MATLAB function (Surf2stl.m) [33].

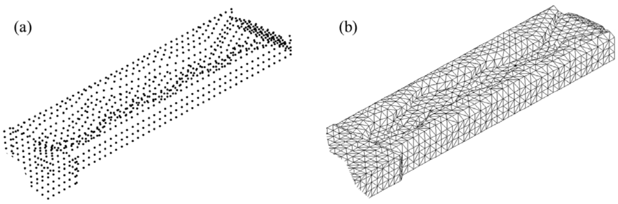


FIGURE 9. (a) Point clouds and (b) triangulated-surface mesh of the function part of a test key.

As shown in Table I, the proposed method of this work is compared with previous methods [19], [20], [21], [22]. The proposed method is applicable for various types of keys under a wide range of shooting angles and low requirement for shooting environment. The other four methods could only duplicate the key in the fixed zigzag type with fixed basic dimensions. Therefore, it can be concluded that our method is the most robust.

TABLE I
KEY-DUPLICATING METHODS COMPARISON

<table>
<tr><th>Methods</th><th>Shooting angle</th><th>Shooting environment</th><th>Images required</th><th>Automation level</th><th>Key types</th></tr>
<tr><td>Proposed</td><td rowspan="2">wide</td><td rowspan="2">daily scene</td><td rowspan="2">single</td><td rowspan="2">medium</td><td>various types</td></tr>
<tr><td>[19]</td><td rowspan="4">only zigzag type</td></tr>
<tr><td>[20]</td><td rowspan="2">fixed</td><td rowspan="2">specially-designed capture equipment</td><td rowspan="2">two</td><td rowspan="2">low</td></tr>
<tr><td>[21]</td></tr>
<tr><td>[22]</td><td>wide</td><td>daily scene</td><td>single</td><td>high</td></tr>
</table>

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3049996, IEEE Access

III. EXPERIMENTAL RESULTS

As shown in Figure 10, twelve keys of the type with two key curves on one side were photographed by a digital camera with a resolution of 72 dpi under daily scenes with different shooting angles. For each target key, the 3D geometry is obtained based on the proposed method using only one single photograph. Figure 11 shows the computer and 3D printer used in the system. The software program based on MATLAB is running on a computer with a CPU of 3.2 GHz and a memory capacity of 8 GB. The Doogell D300 3D printer (Doogell Co., LTD., Hangzhou, China) loads the geometry of the target key and produces the target key with the material of polylactic acid (PLA) filament. Considering the precision requirement, mechanical strength, and printing time, the layer thickness in 3D printing is set to 0.1 mm. It is also found that when the PLA infill density is 100% and the key surface with the largest cross-sectional area is placed on the print bed (Figure 11 (c)), the printed key can obtain the best mechanical properties and the highest surface quality, which is well agreed with the findings of Ranganathan *et al.* [34].

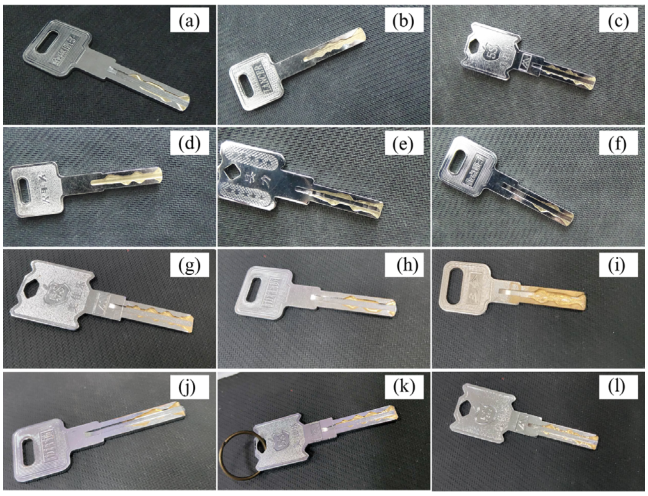


FIGURE 10. Photographs of keys tested in experiments (a) LC, (b) JDYZ-1, (c) MJL-1, (d) HCT, (e) SL, (f) JDYZ-2, (g) MJL-2, (h) JDYZ-3, (i) SJ, (j) MJL-3, (k) GU, and (l) MJL-4.

The printed keys are shown in Figure 12 and the process results are listed in Table II. The reconstructing process from loading the key photograph to obtaining the 3D key geometry takes 4.2 to 5.0 minutes. The time cost deviations occur in the curve extraction process since the different number of points are sampled for key curves with different curvatures. The 3D printing process takes 13.2 to 18.3 minutes for different keys. The keys in Figure 12 (g), (h), and (i) cost more time compared to other keys due to their larger handle parts. The variation of printing time among the rest nine keys is mainly induced by the different basic dimensions and complexity of key curves. The maximum geometric deviation between the printed key and the original key ranges from 0.20 mm to 0.39 mm, which meets the precision requirement for unlocking the lock cylinder.

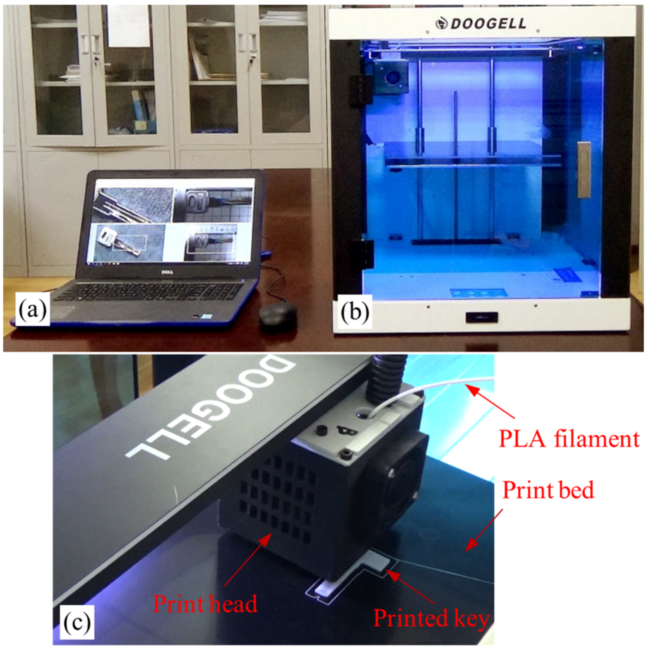


FIGURE 11. Equipment devices (a) the computer, (b) the Doogell D300 3D printer, and (c) the printing process.

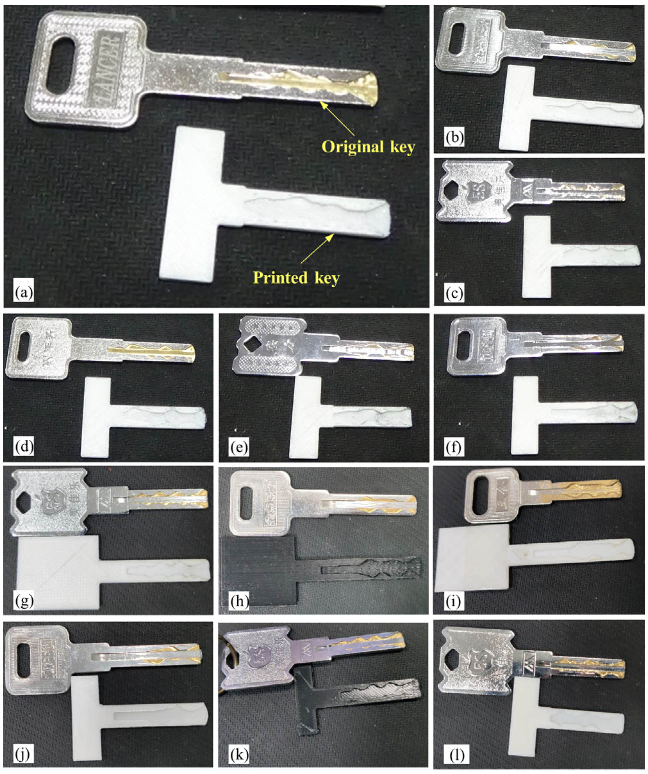


FIGURE 12. Original keys and printed keys (a) LC, (b) JDYZ-1, (c) MJL-1, (d) HCT, (e) SL, (f) JDYZ-2, (g) MJL-2, (h) JDYZ-3, (i) SJ, (j) MJL-3, (k) GU, and (l) MJL-4.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3049996, IEEE Access

TABLE II
EXPERIMENTAL RESULTS

Key	Time for 3D reconstruction (min)	Time for 3D printing (min)	Maximum geometric deviation (mm)
LC	4.2	14.7	0.28
JDYZ-1	4.6	13.2	0.31
MJL-1	4.8	13.8	0.34
HCT	4.8	14.5	0.39
SL	4.7	14.3	0.20
JDYZ-2	4.6	13.4	0.28
MJL-2	4.9	18.3	0.19
JDYZ-3	5.0	17.7	0.23
SJ	4.8	17.6	0.27
MJL-3	4.5	12.8	0.16
GU	4.9	14.1	0.21
MJL-4	5.0	14.3	0.29

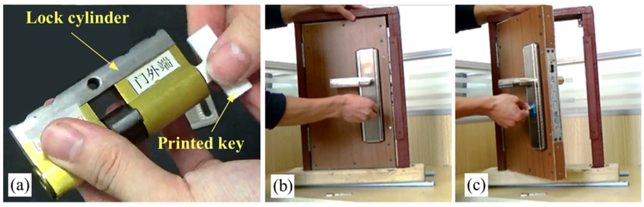
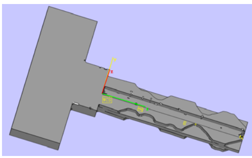
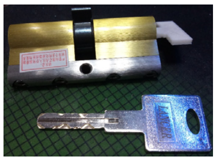
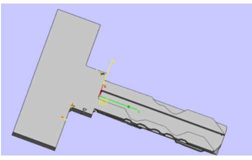
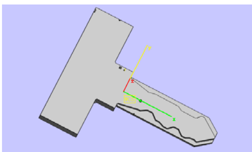
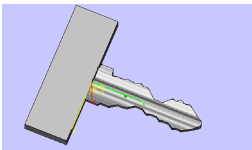


FIGURE 13. Unlocking tests for printed keys.

Figure 13 shows the unlocking test conducted with the lock cylinder (Figure 13 (a)) and a mini door (Figure 13 (b)). All printed keys successfully unlock the associated cylinders, which proves the effectiveness of the system for duplicating a mechanical key based on only one single photograph. However, during the process of unlocking an actual door, the torsional force on the printed key should be increased. Therefore, a mini door equipped with a lock cylinder is used to simulate the actual opening process to test whether the printed key has enough strength to overcome the torsional force. As shown in Figure 13 (b) and (c), the mini door is successfully unlocked by the printed keys, which proves that the material of PLA is practical for printing the key and opening a real door.

Four other types of keys with different cross-sections are used to further evaluate the robustness of the proposed method. As shown in Table III, based on the photographs of those keys, the proposed method reconstructs their 3D geometries and prints them using the 3D printer. All these printed keys successfully unlock the cylinders, which proves the ability of the proposed method in handling various types of keys.

TABLE III
EXPERIMENTAL RESULTS FOR DIFFERENT TYPES OF KEYS

Key type	Key photograph	3D reconstructed model	Printed key	Unlocking test
Two curves on both sides				
Three curves on both sides				
Triangle head with two curves on both sides				
Two curves with arc cross-section				

IV. CONCLUSION

This work proposed an easy and fast key duplicating method based on only one single naturally-captured photograph. A reverse engineering system of the mechanical key was self-developed which includes functions of projective transformation, curve extraction, point clouds generation, and 3D geometry reconstruction. Creatively, an SQL database was established to store basic dimensional information of key blanks, which provides supplementary information during the 3D geometry reconstruction. Five different types of duplicated keys are tested in the unlocking experiment to evaluate the effectiveness of the proposed method. The main results are as follows:

(1) All the duplicated keys by the proposed method succeeded in unlocking tests.

(2) The maximum geometric deviation between the printed key and the original key is less than 0.39 mm, which meets the precision requirement for key duplication.

(3) The 3D geometry reconstruction of a key finished within 5 minutes and the 3D printer took less than 18.3 minutes to manufacture an actual key.

The experimental tests proved that the proposed system is feasible for key duplication. The future work will focus on developing a fully-automated key reconstruction method based on artificial intelligence (AI) for various types of keys, especially for the dimple key whose dimple depth is difficult to be obtained from a single photograph.

REFERENCES

[1] Y. Gingold, T. Igarashi, and D. Zorin, “Structured annotations for 2D-to-3D modeling,” *ACM Trans. Graphics*, vol. 28, no. 5, Art. no. 148, 2009.

[2] T. Chen, Z. Zhu, A. Shamir, S. Hu, and D. Cohen-Or, “3-sweep: Extracting editable objects from a single photo,” *ACM Trans. Graphics*, vol. 32, no. 6, Art. no. 195, 2013.

[3] M. Lau, G. Saul, J. Mitani, and T. Igarashi, “Modeling-in-context: user design of complementary objects with a single photo,” in *Proc. SBIM*, 2010, pp. 17–24.

[4] T. Igarashi, S. Matsuoka, and H. Tanaka, “Teddy: a sketching interface for 3D freeform design,” in *Proc. of ACM SIGGRAPH*, Jan. 1999, pp. 409–416.

[5] Y. Liu, C. Ma, and D. Zhang, “EasyToy: plush toy design using editable sketching curves,” *IEEE Comput. Graph. Applicat.*, vol. 31, no. 2, pp. 49–57, 2011.

[6] R. Schmidt, B. Wyvill, M. C. Sousa, and J. A. Jorge, “Shapeshop: Sketch-based solid modeling with blobtrees,” in *Proc. SBIM* Aug. 2005, pp. 53–62.

[7] A. Shtof, A. Agathos, Y. Gingold, A. Shamir, and D. Cohen-Or, “Geosemantic snapping for sketch-based modeling,” *Comp. Graph. Forum*, vol. 32, no. 2, pp. 245–253, 2013.

[8] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, “FiberMesh: designing freeform surfaces with 3D curves,” *ACM Trans. Graphics*, vol. 26, no. 3, Art. no. 41, 2007.

[9] R. Schmidt, A. Khan, K. Singh, and G. Kurtenbach, “Analytic drawing of 3D scaffolds,” *ACM Trans. Graphics*, vol. 28, no. 5, Art. no. 149, 2009.

[10] X. W. Liu, Y. Li, J. Liu, Q. Zhang, and Y. Luo, “3D image reconstruction of spinning cone-shaped target,” *J. Eng.*, vol. 2019, no. 20, pp. 6461–6465, 2019.

[11] A. Shalaby, M. Elmogy, and A. A. Elfetouh, “3D image reconstruction from different image formats using marching cubes technique,” *Int. J. Comput. Vision Robot*, vol. 9, no. 3, pp. 293–309, 2019.

[12] M. Chen, Y. Zhou, Z. D. Li, and J. Wu, “A novel 3D shape reconstruction method based on maximum correntropy Kalman filtering,” *Sens. Rev.*, vol. 39, no. 3, pp. 332–340, 2018.

[13] Y. Cai, M. Cao, L. Li, and X. Liu, "An end-to-end approach to reconstructing 3D model from image Set," *IEEE Access*, vol. 8, pp. 193268–193284, 2020.

[14] P. Mandikal, N Murthy, M. Agarwal, and R. V. Babu, "3D-LMNet: latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image," in *Proc. 29th Br. Mach. Vis. Conf.*, Sep. 2019.

[15] P. Henderson and V. Ferrari, "Learning single-image 3D reconstruction by generative modelling of shape, pose and shading," *Int. J. Comput. Vision.*, vol. 128, no. 4, pp. 835–854, 2020

[16] B. Hou, B. Khanal, A. Alansary, S. McDonagh, A. Davidson, M. Rutherford, J. Hajnal, D. Rueckert, B. Glocker, and B. Kainz, "3-D reconstruction in canonical co-ordinate space from arbitrarily oriented 2-D images," *IEEE Trans. Med. Imaging*, vol. 37, no. 8, pp. 1737–1750, 2018.

[17] J. Chibane, T. Alldieck, and G. Pons-Moll, "Implicit functions in feature space for 3D shape reconstruction and completion," in *Proc. CVPR*, 2020, pp. 6968–6979.

[18] L. Yu, X. Li, and C. Fu, "PU-Net: point cloud upsampling network," in *Proc. CVPR*, Dec. 2018, pp. 2790-2799.

[19] B. Laxton, K. Wang, and S. Savage, “Reconsidering physical key secrecy: teleduplication via optical decoding,” in *Proc. ACM Conf. Computer Commun. Secur.*, Oct. 2008, pp. 469–478.

[20] V. Itthisupornrat and S. Kiatsunthorn, “Duplicate a blank key using image processing,” in *Proc. SICE-ICASE Int. Joint Conf.*, 2006, pp. 5715–5718.

[21] M. Cheewanantakul and S. Kiatsunthorn, “Duplicate a key using image processing,” in *Proc. SICE Annu. Conf.*, 2008, pp. 40–44.

[22] R. Smith and T. Burghardt, “DeepKey: towards end-to-end physical key replication from a single photograph,” in *Proc. GCPR 2018*, 2018, pp. 487–502.

[23] Y. Wang, N. Deng, and B. Xin, “Investigation of 3d surface profile reconstruction technology for automatic evaluation of fabric smoothness appearance,” *Measurement*, vol. 166, Art. no. 108264, 2020.

[24] X. Wang, L. Shi, and H. Xu, “A simple image acquisition system and its calibration for image-based 3D reconstruction,” in *Proc. ICVIP 2018*, Dec. 2018, pp. 103–107.

[25] J. Hata, K. Ito, and T. Aoki, “A 3D reconstruction method using PMVS for a limited number of view points,” in *Proc. IWAIT 2019*, Jan. 2019, Art. no. 11049:1104942.

[26] M. M. Cheng, “Curve Structure Extraction for Cartoon Images,” in *Proc. HHME*, 2009, pp. 13–20.

[27] T. Várady, R. Martin, and J. Cox, “Reverse engineering of geometric models–an introduction,” *Comput. Aided Des.*, vol. 29, no. 4, pp. 255–268, 1997.

[28] N. Amenta, M. Bern, and M. Kamvysselis, “A new Voronoi-based surface reconstruction algorithm,” in *Proc. SIGGRAPH '98 Computer Graphics Conf.*, 1998, pp. 415–421.

[29] Available at: https://github.com/sylqiu/GPToolBox/blob/7ed535fa56c0dce22128047f708580465c525ad6/mesh/crust.m.

[30] D. F. Watson, “Computing the n-dimensional delaunay tessellation with application to voronoi polytopes,” *Comput. J.*, vol. 24, no. 2, pp. 167–172, 1981.

[31] G. Voronoi, “Nouvelles applications des paramètres continus à la théorie des formes quadratiques,” *J. Reine. Angew. Math.*, vol. 133, pp. 97–178, 1907.

[32] N. Amenta and M. Bern, “Surface reconstruction by Voronoi filtering,” *In Symp. on Comp. Geom.*, 1998, pp. 39–48.

[33] Available at: https://www. mathworks.com/matlabcentral/fileexchange/4512-surf2stl?s_tid=srchtitle%20.

[34] S. Ranganathan, K. S. Kumar, S. Gopal, and C. Pradeep, “The effect of print orientation and infill density for 3D printing on mechanical and tribological properties,” in *Proc. ADMMS 2020*, 2020, Art. no. 2020-28-0411.

Yicang Huang received his B.S. degree in Mechanical Engineering from Wuhan University, Wuhan, China, in 2015, where he is currently pursuing the Ph.D. degree with the School of Power and Mechanical Engineering. His current research interests

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3049996, IEEE Access

include image processing, additive manufacturing, and computer-aided design.

Yikai Zhang received his B.S. degree in electronic information engineering from Hefei University of Technology in 2020 and is currently studying for a master's degree at the Institute of Industrial Science of Wuhan University. His current research interests include image processing, deep learning, and additive manufacturing.

Hui Li received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree in electrical & computer engineering from National University of Singapore, Singapore, in 1999 and 2007, respectively.

He is currently a Professor at Wuhan University, China. His research interests include design of micro/nanoelectromechanical systems and reliability of electronic product.

Shengnan Shen received the B.S. degree from Harbin Institute of Technology, Weihai, China, and the Ph.D. degree in mechanical & aerospace engineering from Nanyang Technological University, Singapore, in 2005 and 2013, respectively.

She is currently an Associate Professor at Wuhan University, China. Her research interests include precision mechanical design, and design & application of high precision positioning control.

Feng Li is currently pursuing the bachelor’s degree with the School of Electrical and Electronic Engineering, Wuhan Polytechnic University, Wuhan, China. His research interests include computer-aided design, computer vision, image processing, and machine learning.