

## Shape completion for depth image via repeated objects registration

Ming Zeng <sup>a</sup>, Liujuan Cao <sup>c,\*</sup>, Xuan Cheng <sup>b</sup>, Kunhui Lin <sup>a</sup>, Huailin Dong <sup>a</sup>, Cheng Wang <sup>c</sup>

<sup>a</sup> Software School of Xiamen University, Xiamen, China

<sup>b</sup> College of Computer Science, Zhejiang University, Hangzhou, China

<sup>c</sup> Department of Computer Science, Xiamen University, Xiamen, China

### ARTICLE INFO

#### Article history:

Received 6 May 2014

Received in revised form

15 November 2014

Accepted 16 November 2014

Available online 20 May 2015

#### Keywords:

Depth image

Repeated objects detection

Image inpainting

Non-rigid registration

Shape completion

### ABSTRACT

With the recent advances of low-cost depth sensors, there are increasing researches which focus on applications using such devices. A critical issue involved is the processing of the low-quality depth images. This paper presents an example-based method for filling the holes in the depth images. The method utilizes the redundant geometrical information contained in the repeated occurrences of some similar objects, so as to mutually infill these incomplete objects. We first introduce a user-assisted object segmentation to select the objects of interest. Subsequently, the method employs an object recognition procedure to detect the occurrences of the selected objects. Given these detected object instances, we complete each object using the detected partial range data in a registration way. Finally, the completed objects are synthesized back to the original depth image, thus to infill the holes. We demonstrate the effectiveness of the proposed method and the therein algorithms by experiments on several real scenes, which contains repeated objects either of one pattern or multiple patterns.

© 2015 Elsevier B.V. All rights reserved.

## 1. Introduction

Recently, commodity depth cameras (e.g. Kinect [1]) are becoming prevalent in computer vision community due to its several advantages such as low price and convenience in use. However, the low cost on the other hand limits the quality of the acquired depth map. Therefore, processing such kind of depth images is currently a fundamental problem deserving investigation for other research topics which are based on these depth images.

The common problems concerning the quality of the depth images are the high degree of noise and the substantial hole existences. Compared to the data noisy, which can be reduced by kinds of filters [2], the hole issue is a more touchy problem, since it entirely misses data in the hole regions. The missing depth can be filled by linear interpolation from nearby depth data, but this way will lead to losing geometry properties, taking corners and edges for example. Another insight borrows the image inpainting spirit, which has been studied over decades. The basic idea behind image inpainting is to repair the lost region by pasting some selected patches onto the missing regions [3]. This infills the complex structure inside the missing region by utilizing information redundancy in repeated patterns, which is a good candidate for repairing depth images. However, the depth images apparently contain geometry information, which inherently is a 3D problem, while the 2D inpainting methods only consider how to copy and paste patches in the image plane. Therefore, directly moving the

2D method to depth image cannot achieve 3D-space coherence, and the reconstructed results will lose inherent geometry information. In this paper, we extend the 2D example-based image inpainting spirit to 3D depth scenario, especially investigate how to complete the data while preserving the geometry structure.

In this paper, we aim at completing the missing regions in the depth image by similar objects cropped from different regions, while preserving the three-dimensional structure. Besides, we also consider the shape deformation when pasting the objects, so as to account for potential shape differences between two similar objects. To these ends, we adopt a detection-registration-synthesis paradigm, where we first detect the repeated objects, and then register similar objects onto hole regions in 3D space, and finally synthesize the scene back to the depth image.

In summary, this paper proposes a novel method to infill the holes in depth images by copying and registering similar objects. The method consists of two ingredient algorithms. The first one is repeated objects detection and the second one is the hole-filling algorithm which is based on 3D shape registration. To the best of our knowledge, this is the first attempt to solve the depth image inpainting in such a manner which employs object detection and registration in 3D space.

## 2. Related work

**Image inpainting:** Image inpainting (also called image restoration) is an ancient topic in image processing. There are great works

\* Corresponding author.

in this field. But the basic idea is intuitive, i.e. interpolate the missing region via its surrounding pixels or synthesize consistent patches from different regions of the image [4]. The former methods will lead to flat region in the interpolated regions thus losing the necessary texture. The latter methods, called exemplar-based methods, however, are able to generate abundant texture in the missing regions. Past decade saw many success of exemplar-based inpainting methods. For example, Sun et al. [3] fill missing regions by propagating neighboring patches into the missing regions along with the specified structures; Inspired by [3], recently, Yu et al. [5] restored hole regions in a RGBD image, guided by the structure extracted from its RGB image. These works inpaint the image via pixel-level or patch-level infilling, which are general but are hard to deal with complex scenes. Cheng et al. [6] parse and detect repeated patterns in the image, and use the extracted template to edit the image. The work solves the problem in a high-level manner, which brings great flexibilities for its potential applications.

**3D object detection:** To detect or recognize an object, one category of work trained object-specific classifiers offline use a large dataset. For example, Silberman et al. [7] and Shao et al. [8] train random regression forests to recognize indoor objects, Karpathy et al. [9] also train several kinds of classifiers (SVM, Naive Bayes, Nearest Neighbor, etc.) for this task. Another category methods detect object by evaluating the quality of feature matching and object alignment [10,11]. In contrast to the classifier-based methods, this kind of methods do not need expensive training stage, and are very suitable for finding repetitive objects.

**Shape registration:** Shape registration aligns partial scans of the same object to match each other. The basic iterative closest point algorithm (ICP) [12] compute rigid transformation between two shapes. For objects undergoing non-rigid deformation, the registration needs to estimate non-rigid motions of its local geometry. To this end, several non-rigid registration framework are proposed, such as embedded deformation [13] and Laplacian deformation [14]. Based on these frameworks, many works complete the whole shape using partial scans from different views [15–17].

### 3. Overview

As shown in Fig. 1, our proposed method consists of four main steps. The input of our method is a depth image of a scene containing repeated objects. Given the depth image, the first step needs users to select a reference object for each repeated pattern (Section 4). To this end, we design a user-assisted object segmentation to select the reference objects in an easy way. The second step is object recognition, which identifies all occurrences that are similar to the reference objects (Section 5). Consequently, given the detected repeated objects, the shape registration separately completes objects which belong to each pattern (Section 6.1). Finally, the depth re-synthesis step fills the holes of the depth image using the completed objects (Section 6.2).

### 4. User-assisted reference object selection

Given a depth image, our goal of this step is to select reference objects as repeated patterns in this image. To this end, we design a user-assisted method to efficiently fulfill this task. For a scene

especially which is indoor, it is very common that almost all objects stand on their corresponding supporting planes (Fig. 2).

Based on this observation, we design a simple-yet-efficient algorithm to select interest objects, as is shown in Algorithm 1. The algorithm requires users to draw strokes on the target objects and their supporting planes, respectively. We denote the point set of the stroke on the  $i$ th object as  $R_i$ , and the point set of the stroke on its associated supporting plane as  $P_i$ .

Given  $R_i$ , the selected parts are propagated to their neighboring parts according to these stroked points. The supporting planes serve as divisions between points of different objects. The equation of the supporting plane containing  $P_i$  can be estimated by fitting the following plane function with least square linear regression:

$$n_i \cdot p_k + D_i = 0, \quad p_k \in P_i \quad (1)$$

where  $n_i$  is the normal of the supporting plane of  $P_i$ , and  $D_i$  is the offset.

It should be noted that, in order to avoid ambiguities in the 2D space when computing neighbors, the above-mentioned steps but the user-selection are operated on the point clouds generated from the depth image.

#### Algorithm 1. User-Assisted Reference Object Selection.

```

1: Draw Strokes on reference objects  $RO = \{R_1, R_2, \dots, R_n\}$  and
   their supporting planes  $P = \{P_1, P_2, \dots, P_n\}$ 
2: for each object-plane pair  $\{R_i, P_i\}$  do
3:   Step 1: compute the plane function  $plane_i$  according to
      points in  $P_i$ ,
4:   Step 2: propagate points in  $R_i$  to neighbors form the
      expanded point set  $\tilde{R}_i$ , the propagation stops when
      encountering points on the plane  $f_i$ .
5: end for
```

### 5. Repeated objects recognition

After selection of the interest objects, the next step is to find the repeated instances of these objects. There are many existing methods for object matching based on classifiers to recognize objects' categories [7]. Since there are several instances of a pattern in our application scenario, it is not necessary to train classifiers offline. Instead, the object recognition can be performed by finding matching between the reference and candidates' objects.

#### 5.1. Feature representation

To describe the geometry characteristics of an object  $\tilde{R}_i$ , we uniformly sample the points in  $\tilde{R}_i$  and points on the depth images. We denote the sampled location sets by  $L_{ref_i}^{sample}$  and  $L_{scene}^{sample}$ , respectively. In our implementation, the sample space  $\epsilon$  is set to 1% of the length of the scene's diagonal line. Other than the uniformly sampled locations, we also detect SIFT feature points on depth image, denoting them as  $L_{ref_i}^{sift}$  and  $L_{scene}^{sift}$ .

Then we compute feature descriptors given these sampled and detected locations. Since the depth image may be noisy, the single feature will fail to robustly describe the local geometry. Therefore, we further combine two informative features (VFH [18] and SHOT [19]) together to improve the discriminative ability of the feature.

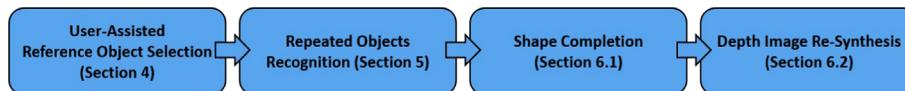
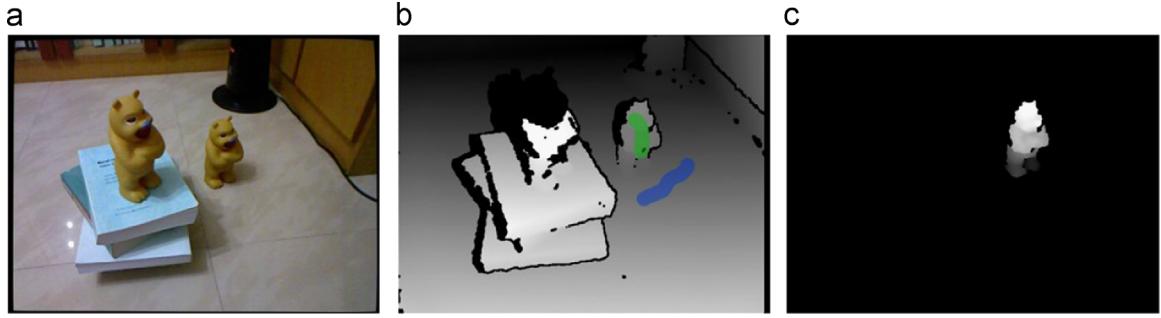


Fig. 1. The pipeline of the proposed infilling algorithm.



**Fig. 2.** Illustration of the User-Assisted Object Selection. (a) The reference RGB image. (b) The depth image and the user strokes over it. The green indicates the object, and the blue represents its supporting plane. (c) The selected reference object. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

## 5.2. Feature matching and repeated object detection

**Feature matching:** After computing the feature descriptor, for each point in  $L_{\text{sample}}^{\text{ref}}$  and  $L_{\text{ref}}^{\text{sift}}$ , we find its nearest neighbors in the interest point sets  $L_{\text{scene}}^{\text{sample}}$  and  $L_{\text{scene}}^{\text{sift}}$  of the scene. Then we need to group correspondences and prune wrong matchings. Our method follows the classic RANSAC scheme. We assume the repeated objects are nearly rigid, therefore it is reasonable to approximate transformation between the reference object and other objects in the scene using a 6-DOF rigid motion. For every 3 pairs of matchings, the procedure computes the rigid transformation, and then verify this transformation using other matchings.

**Repeated object detection:** After the RANSAC filtering, now we are able to estimate a rough rigid transformation between the reference object and the detected object in the scene. Then, a ICP operation is applied onto the reference object so as it can match to the target object. After which, we check the proportion of the correspondences whose distances are small enough (e.g. below 5 mm). If the proportion is larger than a threshold (we use 50%), the target object is recognized as a repeated instance of the reference object.

**Detected object extraction** After identifying the target object, we apply the object selection algorithm for the target object, but at this time there is no need to draw strokes. The corresponding points in the target object can be used as the seed points for propagation, and the supporting plane can be also calculated by transforming the supporting plane of the reference object onto the target object's position (see Fig. 3(a)). Now, we obtain all repeated instances for each reference object  $R_i$ , and these instances are denoted by  $I_i^1, I_i^2, \dots, I_i^{N(i)}$ , where  $N(i)$  is the number of the detected instances for  $R_i$ .

## 6. Shape completion and depth image re-synthesis

### 6.1. Shape completion

The task of this step is to complete each shape. For the  $i$ th shape, we use the reference object  $R_i$  and its other instances  $I_i^1, I_i^2, \dots, I_i^{N(i)}$  to achieve this task (for convenience, we also use  $I_i^0$  to refer to the reference object  $R_i$ ). The illustration of this step can see Fig. 3(c, d). To account for both rigid transformation and slight non-rigid deformation, we employ a two-stage registration to complete the shape.

**Stage 1: Rigid shape registration:** This stage transforms all instances  $I_i^1, I_i^2, \dots, I_i^{N(i)}$  to the local coordinate of the reference object  $R_i$ . Ideally, the transformations computed from previous ICP step can be directly used. However, to distribute registration among these instances, we adopt a global registration [20] which jointly registers these instances together, refer to [20] for details.

**Stage 2: Non-rigid shape registration:** After rigid registration, we further deform the detected repeated instances  $I_i^1, I_i^2, \dots, I_i^{N(i)}$  to fit the geometry of  $I_i^0$ , so as to infill the hole of  $I_i^0$  more precisely. Here,

we employ embedded deformation graph [13] to present non-rigid deformation, where the deformation favors as-rigid-as-possible motions of the objects. At this framework, we adopt the non-rigid ICP to pairwisely register instances to  $I_i^0$ . To this end, we are to find an affine transformation for each point in deforming parts. The transformation can be represented by a  $3 \times 3$  matrix  $H$  and a  $3 \times 1$  vector  $t : H \cdot p + t$ .

The entire registration algorithm for  $i$ th reference object  $I_i^0$  can be formulated into the following energy minimization problem:

$$E = w_{\text{feat}} E_{\text{feat}} + w_{\text{closest}} E_{\text{closest}} + w_{\text{rigid}} E_{\text{rigid}} + w_{\text{smooth}} E_{\text{smooth}} \quad (2)$$

The  $E_{\text{feat}}$  term constrains the matched feature points between the reference object and detected instances to be near. The  $E_{\text{closest}}$  term requires the closest point between the reference object and detected instances to fit well together. These two terms adopt the same L2 Euclidean distance as their energies

$$E_{\text{feat}} = \sum_{k \in N(i)} \sum_{p_j \in I_i^0} \|p_j - H_{kj} \cdot q_{kj}^* + t_{kj}\|_2^2 \quad (3)$$

and

$$E_{\text{closest}} = \sum_{k \in N(i)} \sum_{p_j \in I_i^0} \|p_j - H_{kj} \cdot q_{kj}^* + t_{kj}\|_2^2 \quad (4)$$

where  $p_j$  is the  $j$ th feature point ( $E_{\text{closest}}$  takes all point) of the reference object  $I_i^0$ , and  $q_{kj}^*$  is  $p_j$ 's correspondence in the  $k$ th repeated instance  $I_i^k$ .

The rigid term  $E_{\text{rigid}}$  constrains the transformation matrix  $H_i$  to be rotational:

$$E_{\text{rigid}} = \sum_{k \in N(i)} \sum_{j=1}^{|I_i^k|} \text{Rot}(H_{kj}) \quad (5)$$

where  $\text{Rot}(H) = \|H' H - I\|_F^2$ .

The smooth term considers the smoothness of the neighboring deformation, which measures the difference of the nearby points' transformations:

$$E_{\text{smooth}} = \sum_{k \in N(i)} \sum_{j=1}^{|I_i^k|} \sum_{p_{k,l} \in \text{Nb}(p_{k,j})} \|H_{kj}(p_{k,l} - p_{k,j}) + p_{k,j} + t_{kj} - (p_{k,l} + t_{k,l})\|_2^2 \quad (6)$$

where  $\text{Nb}(p_{k,j})$  is the neighbor points of  $p_{k,j}$ .

The  $E_{\text{rigid}}$  and  $E_{\text{smooth}}$  together impose as-rigid-as-possible constraints for the deformation, for more details refer to [17].

### 6.2. Depth image re-synthesis

After shape registration, we simply re-render the completed objects back to the depth image. We first place the completed object onto the locations of the instances, under each instance's orientation and pose (deformation). Then, we render the completed scene using

a modified z-buffer algorithm. In our method, if the current depth value is very close to the value recorded in the buffer, we instead average the new and recorded value and update the buffer, rather than simply keep the nearer value. This strategy can efficiently reduce discontinuity due to registration error.

## 7. Experimental results

### 7.1. Experimental setup

The depth data used in this experiments is captured from the depth stream of a Kinect. The resolution of the depth data is 640\*480. Due to the minimal and maximal working distances required by Kinect, and the surface reflection of the infrared light, some depth data are missing (see Fig. 4(d)).

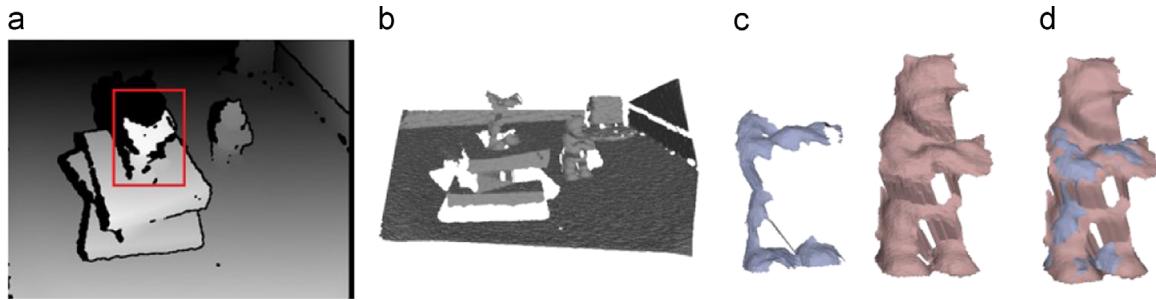
### 7.2. With repeated objects in one pattern

We first validate our method by a scene which contains one kind of repeated objects. There are three bears in this scene (Fig. 4(d)).

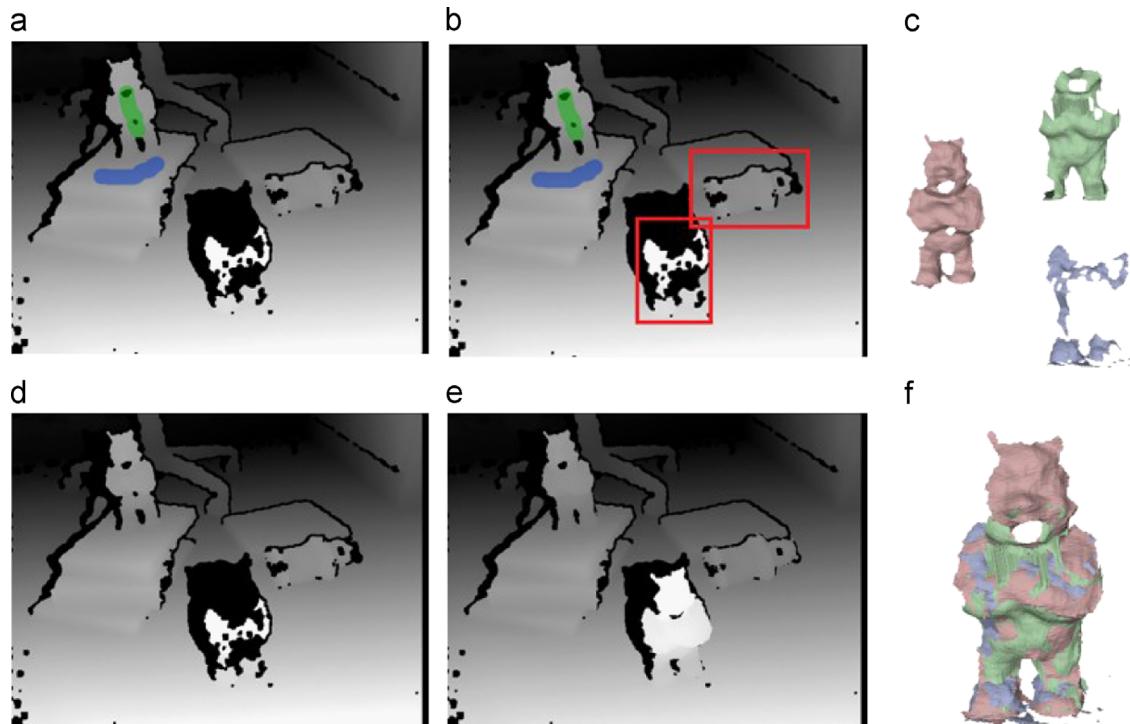
We select the reference bear by simply drawing two strokes (one on the bear and the other on its supporting plane, the green and the blue strokes in Fig. 4(a)). Then the object recognition detects the repeated object (the rectangles in Fig. 4(b)). Fig. 4(c) shows the selected and detected objects in 3D. Registering these interest parts together yields a more complete 3D model (Fig. 4(f)). Finally, the repaired object is rendered back to the original depth image, and thus infills some missing regions. By comparing the original depth data (Fig. 4(d)) with the repaired depth data (Fig. 4(e)), we can observe that the head of the frontal standing bear and the hole of the other standing bear has been infilled, where the head is a large and complex region, it is very challenging to infill such kind of structures.

### 7.3. With repeated objects in multiple patterns

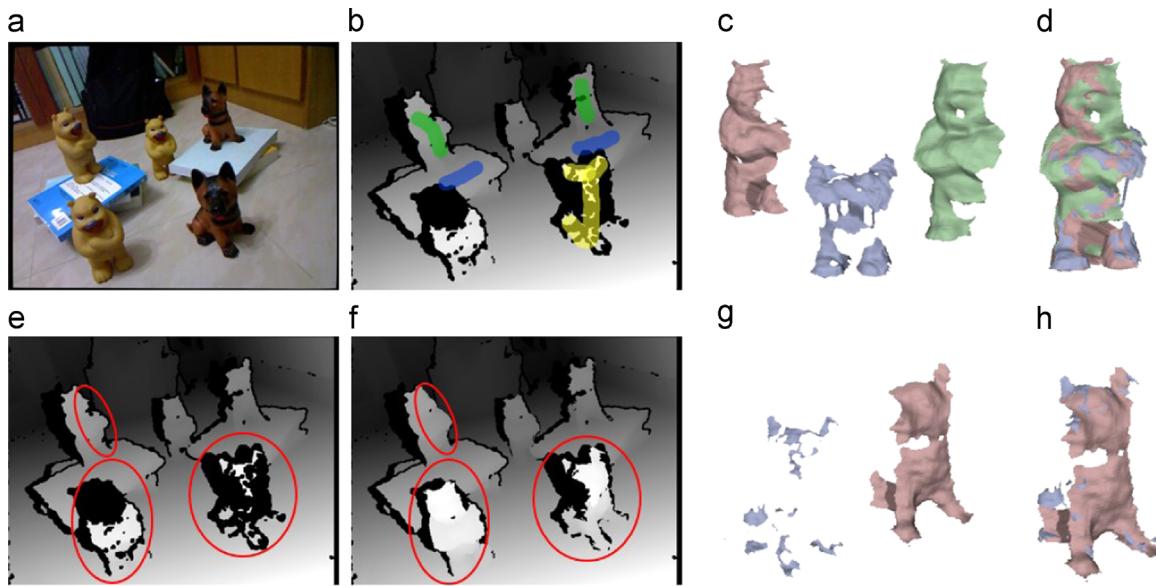
Our method is able to handle more complex scenes which may contain multiple kinds of repeated objects. We capture depth data of a scene with three bears and two dogs (Fig. 5(a)). References of each kind of objects are selected, respectively (Fig. 5(b)). Then each kind of objects is detected. In this example, our detection algorithm fails to



**Fig. 3.** Steps of object detection and completion. (a) Repeated objects detection. (b) The 3D scene of the depth image. (c) The 3D models of the detected (blue) and the reference (red) object. (d) The completed objects. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)



**Fig. 4.** Example with one single kind of object. (a) The user strokes on the depth image. (b) The detected objects (red rectangles). (c) The 3D models of the reference object and the detected objects. (f) The completed objects. (d) The original depth image. (e) The repaired depth image. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)



**Fig. 5.** Example with multiple kinds of objects. (a) Corresponding RGB image for reference. (b) The user strokes on the depth image. (c,g) The 3D models of the reference objects and the detected objects. (d,h) The completed objects. (e) The original depth image. (f) The repaired depth image. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

detect the frontal dog, because its depth data is severely corrupted. We draw another stroke on the dog to guide the detection algorithm (the yellow stroke in Fig. 5(b)). The detected objects are completed (Fig. 5(c, d, g, h)) and rendered to the depth map. See the highlighted regions on the depth maps before (Fig. 5(e)) and after infilling (Fig. 5(f)), which shows the significant effects of the proposed method.

## 8. Conclusion and future work

In this paper we present a novel method to fill holes in the depth images. Our method extends the copy-and-paste fashion into 3D space, and devises associated algorithms for repeated objects detection and shape completion. All operations of the algorithms are conducted in 3D space, which allows more complete transformation space than 2D fashions, leading to improvement both in robustness and accuracy. We conduct extensive experiments with different scenes, all containing repeated patterns. The results validate the feasibility and effectiveness of our proposed method.

The proposed method can benefit at least three application scenarios. First, it can be used to preprocess the RGB-D data, expanding the application field of the low-cost depth sensor. For example, the quality-improved depth data is a potential extra information to visual search [21–25]. Second, the repeated object extraction method can be employed to collect depth data of the same object, being further used to learn object-specific classifiers [26,27]. Third, standing on the stone of this work, more applications regarding to depth image editing can be explored, e.g. scene rearrangements [6].

The current method still has limitations to be further investigated in the future work. First, considering substantial non-rigid deformation within objects will largely improve the versatility and robustness of the method. To this end, object detection should be designed to recognize deformed shapes [28], also the shape registration should be adapted to more general deformations [29]. Second, current method merely considers the severe occlusion problem due to clustered object arrangements. The future work may investigate the scene segmentation and object detection under this situation.

## Acknowledgments

We would like to thank the reviewers for their valuable comments. This work was partially supported by NSFC (Nos. 61402387 and 61402388) and the Open Project Program of State Key Lab of CAD&CG (Grant no. A1419), Zhejiang University.

## References

- [1] Microsoft, <http://www.microsoft.com/>, 2011.
- [2] C. Richardt, C. Stoll, N.A. Dodgson, H.P. Seidel, C. Theobalt, Coherent spatio-temporal filtering, upsampling and rendering of RGBZ videos, *Comput. Graph. Forum* 31 (2) (2012) 247–256.
- [3] J. Sun, L. Yuan, J. Jia, H.Y. Shum, Image completion with structure propagation, *ACM Trans. Graph.* 24 (3) (2005) 861–868.
- [4] A. Criminisi, P. Pérez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, *IEEE Trans. Image Process.* 13 (9) (2004) 1200–1212.
- [5] L.F. Yu, S.K. Yeung, Y.-W. Tai, S. Lin, Shading-based shape refinement of RGB-D images, in: CVPR, 2013, pp. 1415–1422.
- [6] M.M. Cheng, F.L. Zhang, N.J. Mitra, X. Huang, S.M. Hu, RepFinder: finding approximately repeated scene elements for image editing, *ACM Trans. Graph.* 29 (4) (2010).
- [7] N. Silberman, R. Fergus, Indoor scene segmentation using a structured light sensor, in: Proceedings of the International Conference on Computer Vision—Workshop on 3D Representation and Recognition, 2011.
- [8] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, B. Guo, An interactive approach to semantic modeling of indoor scenes with an RGBD camera, *ACM Trans. Graph.* 31 (6) (2012) 136.
- [9] A. Karpathy, S. Miller, L. Fei-Fei, Object discovery in 3D scenes via shape analysis, in: ICRA, 2013, pp. 2088–2095.
- [10] J. Shin, R. Triebel, R. Siegwart, Unsupervised discovery of repetitive objects, in: ICRA, 2010, pp. 5041–5046.
- [11] H. Chen, B. Bhambhani, 3D free-form object recognition in range images using local surface patches, *Pattern Recognit. Lett.* 28 (10) (2007) 1252–1262.
- [12] P.J. Besl, N.D. McKay, A method for registration of 3-D shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (2) (1992) 239–256.
- [13] R.W. Sumner, J. Schmid, M. Pauly, Embedded deformation for shape manipulation, *ACM Trans. Graph.* 26 (3) (2007) 80.
- [14] M. Botsch, O. Sorkine, On linear variational surface deformation methods, *IEEE Trans. Vis. Comput. Graph.* 14 (1) (2008) 213–230.
- [15] H. Li, L. Luo, D. Vlasic, P. Peers, J. Popović, M. Pauly, S. Rusinkiewicz, Temporally coherent completion of dynamic shapes, *ACM Trans. Graph.* 31 (1) (2012).
- [16] H. Li, E. Vouga, A. Gudym, L. Luo, J.T. Barron, G. Gusev, 3D self-portraits, *ACM Trans. Graph.* 32 (6) (2013) 187.
- [17] M. Zeng, J. Zheng, X. Cheng, X. Liu, Templateless quasi-rigid shape modeling with implicit loop-closure, in: Proceedings of CVPR 2013, IEEE, Portland, 2013.

- [18] R.B. Rusu, G.R. Bradski, R. Thibaux, J. Hsu, Fast 3D recognition and pose using the viewpoint feature histogram, in: IROS, 2010, pp. 2155–2162.
- [19] S.S. Federico Tombari, L.D. Stefano (Eds.), Unique signatures of histograms for local surface description, in: Proceedings of ECCV 2010, 2010.
- [20] K. Pulli, Multiview registration for large data sets, in: Proceedings of 3DIM, 1999, pp. 160–168.
- [21] R. Datta, D. Joshi, J. Li, J.Z. Wang, Image retrieval: Ideas, influences, and trends of the new age, *ACM Comput. Surv.* 40 (2) (2008).
- [22] R. Ji, L.Y. Duan, J. Chen, H. Yao, J. Yuan, Y. Rui, W. Gao, Location discriminative vocabulary coding for mobile landmark search, *Int. J. Comput. Vis.* 96 (3) (2012) 290–314.
- [23] R. Ji, H. Yao, W. Liu, X. Sun, Q. Tian, Task-dependent visual-codebook compression, *IEEE Trans. Image Process.* 21 (4) (2012) 2282–2293.
- [24] R. Ji, L.Y. Duan, J. Chen, L. Xie, H. Yao, W. Gao, Learning to distribute vocabulary indexing for scalable visual search, *IEEE Trans. Multimed.* 15 (1) (2013) 153–166.
- [25] R. Ji, Y. Gao, B. Zhong, H. Yao, Q. Tian, Mining flickr landmarks by modeling reconstruction sparsity, *ACM Trans. Multimed. Comput. Commun. Appl.* 7S (1) (2011) 31:1–31:22.
- [26] B. Browatzki, J. Fischer, B. Graf, H.H. Bulthoff, C. Wallraven, Going into depth: Evaluating 2D and 3D cues for object classification on a new, large-scale object dataset, in: ICCV Workshops, 2011, pp. 1189–1195.
- [27] J. Tang, S. Miller, A. Singh, P. Abbeel, A textured object recognition pipeline for color and depth image data, in: Proceedings of the International Conference on Robotics and Automation (ICRA), 2012.
- [28] D. Smeets, J. Hermans, D. Vandermeulen, P. Suetens, Isometric deformation invariant 3D shape recognition, *Pattern Recognit.* 45 (7) (2012) 2817–2831.
- [29] Q.X. Huang, B. Adams, M. Wicke, L.J. Guibas, Non-rigid registration under isometric deformations, *Comput. Graph. Forum* 27 (5) (2008) 1449–1457.



**Ming Zeng** is currently an Assistant Professor in the Software School of Xiamen University, China. He obtained his Ph.D. degree in computer science from Zhejiang University, in 2013. His main research interests include interactive computer vision, geometry acquisition, and 3D reconstruction.



**Liujuan Cao** is currently an Assistant Professor at the Department of Computer Science, Xiamen University. Before that, she obtained her Ph.D. degree from Harbin Engineering University. Her research is in the field of multimedia analysis, geo-science and remote sensing, and computer vision. She has published extensively at CVPR, Neurocomputing, Signal Processing, ICIP, VCIP, etc.



**Xuan Cheng** is currently a Ph.D. candidate in the State Key Lab of CAD&CG, Zhejiang University, China. He obtained his B.S. degree from South China University of Technology, in 2010. His research interests mainly include object tracking, motion capture and 3D reconstruction.



**Kunhui Lin** is a Professor in the Software School of Xiamen University, China. His main research interests include multimedia, intelligent information processing, and cloud computing.



**Huailin Dong** is a Professor in the software school of Xiamen University, China. His main research interests include digital media processing, intelligent information processing, data mining, and software engineering.



**Cheng Wang** is a Professor at the Department of Computer Science, School of Information Science and Technology (SISE), Xiamen University, China. His research interests include 3D point cloud and remote sensing image analysis, video enhancement, information fusion and mobile mapping data processing.