

# Stripes-based Object Matching

Oliver Tiebe, Cong Yang\*, H. Hassan Khan and Marcin Grzegorzek

Institute for Vision and Graphics  
University of Siegen  
Hoelderlinstr. 3, Siegen, Germany

Dominik Scarpin

Institute of Automatic Control Engineering  
University of Siegen  
Hoelderlinstr. 3, Siegen, Germany

**Abstract**—We propose a novel and fast 3D object matching framework that is able to fully utilise the geometry of objects without any object reconstruction process. Traditionally, 3D object matching methods are mostly applied based on 3D models. In order to generate accurate and proper 3D models, object reconstruction methods are used for the collected data from laser or time-of-flight sensors. Although those methods are naturally appealing, heavy computations are required for segmentation as well as transformation estimation. Moreover, some useful features could be filtered out during the reconstruction process. On the contrary, the proposed method is applied without any reconstruction process. Building on stripes generated from laser scanning lines, we represent an object by a set of stripes. To capture the full geometry, we describe each stripe by the proposed robust point context descriptor. After representing all stripes, we perform a flexible and fast matching over all collected stripes. We show that the proposed method achieves promising results on some challenging real-life objects.

## I. INTRODUCTION

Determining the similarity between 3D objects is a fundamental task for many robotic and industrial applications [1], [2] such as 3D shape retrieval, face morphing, and object recognition [3]. A challenging aspect of this task is to find suitable object signatures that can be constructed and compared quickly, while still discriminating between similar and dissimilar objects. With traditional approaches [4], [5], [6], accurate and proper 3D models [7], [8], [9] are firstly reconstructed to feature the geometrical and textural properties of objects. Specifically, as shown in Figure 1, the first step is the object scanning via time-of-flight cameras [10] or laser scanning systems [11]. After that, the scanned object is represented by some special formats (e.g. point cloud) [12] with a filtering process to remove outliers and noise. Finally, original surfaces from 3D scans are reconstructed into an object model using meshes or other formats. For object matching, 3D features are generated at a certain 3D point or position in space, which describe geometrical patterns based on the information available around the point. Finally, the similarity between two 3D objects is calculated based on object matching.

However, it shares some challenges with the pipeline of traditional approaches. For object reconstruction, some heavy computation costs may be required depending on the outliers in point clouds and tasks at hand. Specifically, Due to the background clutter and measurement errors, certain objects present a large number of shadow points. This complicates

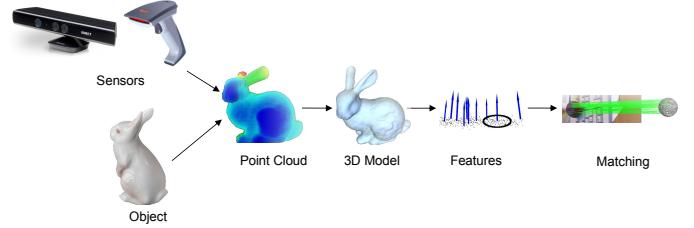


Fig. 1. Pipeline of the traditional object representation and matching.

the estimation of local point cloud 3D features. Thus, some of these outliers should be filtered by performing a statistical analysis on each point's neighbourhood, and trimming those which do not meet a certain criteria [12]. This filtering process normally calls for a large number of calculations. Moreover, if the point cloud is composed of multiple scans that are not aligned perfectly, a smoothing and re-sampling process is also required. In addition, during the outlier removing process, some fine-grained features could be filtered out.

For object matching, in order to find reliable feature point correspondences, some high-order graph matching frameworks are employed to establish feature correspondences, combining both appearance similarity and geometric compatibility [13], [14], [15]. Although those methods have been successfully applied in 2D image features, limited prior art refers 3D surfaces. The main reason is that a 3D surface is not represented in the Euclidean 2D domain, and therefore distances between two points on the surface cannot be computed in a closed form [16]. Moreover, computing object similarity using correspondences normally requires high computational complexity (e.g. the most commonly used Hungarian algorithm [17] needs  $O(n^3)$  time complexity, where  $n$  is the number of feature points) since each feature point in one object should be assigned to a point in another object<sup>1</sup>. Thus, it is hard to be applied in real time.

In order to solve the above problems, we present a method that is able to efficiently represent a 3D object using scanning stripes without any reconstruction process. With this, similarity between objects can be calculated directly using vector distance methods [19], [20] with low computational complexity. Specifically, as shown in Figure 2, instead of object reconstruction using point clouds, we represent an object into a

<sup>1</sup>In some partial matching tasks, partial feature points could be jumped using dummy points like in [18].

\*Corresponding author: cong.yang@uni-siegen.de (Cong Yang).

set of stripes which are collected from laser scanning lines. In order to capture the full geometry of an object, we process and describe each stripe by the proposed point context descriptor. After representing all stripes, we perform a flexible and fast matching over all collected stripes. Since the collected scripts are naturally ordered by the moving direction of a scanning laser, stripes can be easily matched based on their relative locations. With this property, our matching task is applied in real time without any heavy stripe corresponding process.

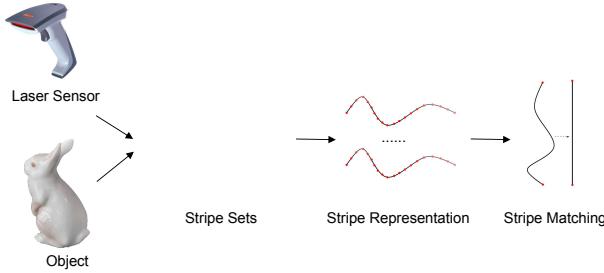


Fig. 2. Pipeline of the proposed representation and matching methods.

The most significant scientific contributions of this paper include: (1) We propose a novel and efficient stripe-based object representation method without any 3D object reconstruction process. (2) In order to fully capture the geometrical properties of each stripe, we introduce an intuitive and robust point context descriptor. (3) We introduce a fast matching method for calculating the similarity between two stripe sets. This approach is applied without any corresponding process and the matching complexity can be reduced dramatically. (4) The experiments show that some objects with similar shapes can be classified accurately using the proposed method.

## II. RELATED WORK

3D object reconstruction with range data is widely covered in literatures together with a good overview given in [21]. In order to capture the geometrical and surface features of 3D objects, two types of approaches are proposed. The first one is to preserve the fine-grained features of objects by combinations of atomic shapes, generalised cones and superquadrics [22], [23]. However, such approaches could not robustly handle real world imagery, and largely failed outside controlled lab environments. In order to solve these problems, researchers introduce more and more geometric structure in object class models and improve their performance [24], [25]. Moreover, objects can also be represented as collections of planar segments using CAD models and lifted to 3D with non-rigid structure-from-motion [26]. The second one is to preserve the coarse-grained features of objects by combining multiple simple shapes to obtain object models [27]. This idea is further improved to the level of plane- and box-type models [28], [29]. Though most works [30], [28], [29] indicate that both fine- and coarse-grained models can help one to better guess the 3D layout of an object while at the same time improving 2D recognition, those methods normally require high computing time for processing and analysing 3D surfaces

since most surfaces rarely have simple parametrisations. In addition, since 3D surfaces can have arbitrary topologies, many useful methods for analysing other media have no obvious equivalent for surface models. On the contrary, as we directly employ the scanning stripes for object representation, the proposed method is applied without any 3D reconstruction process.

For object matching, the biggest challenge is the large non-rigid deformations of object surfaces. In applications such as facial expression recognition, there are localised, high-degree of freedom deformations. To tackle this problem, two types of approaches are normally employed [16], [31]. The first one obtains dense feature point correspondences by embedding the surfaces to a canonical domain which preserves the geodesics or angles [32], [33]. Such embedding requires an initial set of feature correspondences or boundary conditions. However, it is difficult to find reliable feature point correspondences and consistent boundary conditions. Furthermore, since most surface deformations are not perfectly isometric, solely considering intrinsic embedding information may introduce approximation errors to the matching results. Therefore, Zeng et al. propose an approach to achieve robust dense surface matching via high-order graph matching in the embedding manifold [16], [13]. Specifically, they use multiple measurements to capture the appearance and geometric similarity between deformed surfaces and high-order graph interaction to model the implicit embedding energy. As these approaches require high computational complexity for optimising high-order graphs, they are hard to be applied in real time. The second type is to represent 3D models using skeletons and then skeleton matching approaches are employed for matching objects [18], [34]. However, as skeletonisation methods [31], [35] are normally sensitive to noise, the generated skeletons require an extra skeleton pruning process [36]. Moreover, similar to the feature point matching approaches, skeleton matching is built on skeleton graphs which require expensive computational time for search correspondences. Different from the aforementioned approaches, the proposed method calculates the similarity between objects without any corresponding process since the generated stripes are naturally ordered. Therefore, our method can be applied in real time.

## III. STRIPE GENERATION

In this section, the stripe generation method is introduced. The stripe generation is done with a robot guided 2D laser scanner (see Figure 3(a)), which was developed for the mod-iCAS [37] project at the University of Siegen. The original purpose of this system is to assist medical personnel in a surgical environment to acquire the patients face anatomy, to perform intra-operative patient registration, surgical navigation and placement of medial tools.

As shown in Figure 3(b), the laser scanner is build up from a commercial 3D stereo-vision system, developed by the company Point Grey, which is normally used for marker based tracking. To extend this system to a high resolution laser scanner, a line laser module is rigidly attached to one of the

two integrated cameras of the stereo-vision system. Afterwards a camera calibration is done to calculate the intrinsic camera parameters, which are needed to correct distortions caused by the camera lens. At least the laser scanner is calibrated with a special calibration device for correct distance measurement. The achieved measurement accuracy of the laser scanner is less than 0.3 mm.

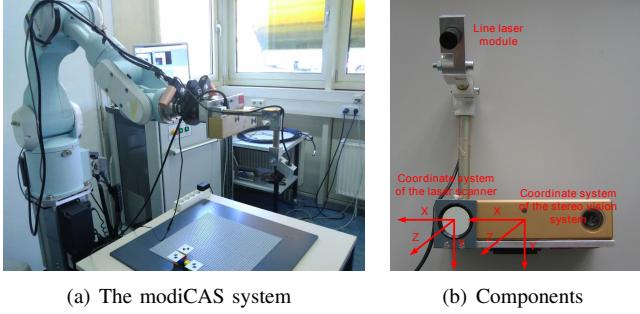


Fig. 3. The laser scanning system and its components.

For a measurement with the laser scanner the line laser module is used to project a laser line to the object surface and the reflection of the laser then is detected by the camera. Due to the characteristic of the objects surface, the laser line is deformed in the acquired camera image. From this deformation, and the knowledge of the camera parameters and the triangulation angle between the camera and the line laser module, the distance between the camera and the object surface can be calculated.

To ensure a solid detection of the reflected laser line in the camera image, it is necessary to avoid any influences from background light. To suppress any background light, an optical bandpass filter, which is designed to only let light around a wavelength of 650nm pass through, is mounted in front of the camera lens used. With the help of such a filter the laser line is the brightest object in the camera image and can be detected easily. For this, in every column of the image array the start and the end of the laser line are detected by an adjustable threshold. Afterwards, the correct position of the laser line in each column is calculated using a weighted average of the intensity values of the laser between the threshold borders.

For the acquisition of the laser lines, the laser scanner is mounted to the robot arm with the help of a rigid fast coupling and a hand-eye calibration is done to calculate the transformation matrix between the coordinate systems of the optical system and the robot flange. Afterwards, the objects are placed one after another on a table, with a distance of approximately 60 cm between the table and the laser scanner, and the laser scanner is moved with the robot arm along the object's surface with a homogenous speed, so the distance between the laser lines is constant over the whole object's surface. The density of the laser lines on the object surface can be affected by the moving speed of the robot arm.

After completion of the data acquisition the detected laser lines are saved in an array, which is transferred from the camera control PC to the user PC. For further processing, the

found line positions are recalculated to black/white images. The calculated distance values are not needed for this project. Figure 4 shows an example of the original object (Figure 4(a)) and its collected stripes (Figure 4(c)). For comparison, a 3D model of the original object is illustrated in Figure 4(b). We can observe that the proposed stripe descriptor is more simple than the 3D model. Moreover, considering the generation speed, the proposed descriptor is much faster than 3D models. Here we only illustrate the stripes from one side of the surface. To capture the full geometry, we can conduct multiple scans while changing the object's pose. However, experiments in Section VI show that even with the single-side stripes, matching performances are still promising. Thus, the complete scanning strategy is applied based on different applications.

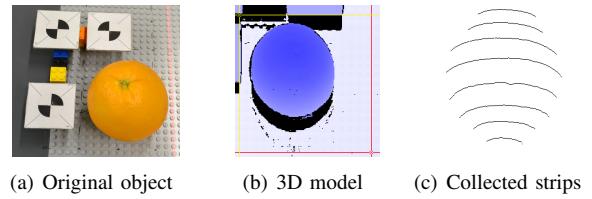


Fig. 4. An apple and its 3D model and generated stripes (partly).

#### IV. STRIPE REPRESENTATION

For a given stripe, we describe its geometrical and topological properties by the point context descriptor. Specifically, given a stripe  $C$  with  $H$  points, for every point  $p_i \in C, i = 1, 2, \dots, H$ , we consider both the distance and direction of the vector form  $p_i$  to other points in  $C$ . Then, the mean distance and direction are calculated for stripe representation. Moreover, in order to distinguish the straight line-similar stripes, the normalised<sup>2</sup> stripe length  $l$  is also employed for stripe description. Thus, a stripe is represented by a three-dimensional feature vector. The proposed descriptor has many characteristics: (1) It is simple and intuitive. (2) It integrates both geometrical and topological features of a stripe. (3) It is flexible for stripe matching since it can be adopted to different matching algorithms.

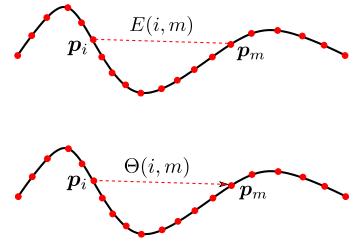


Fig. 5. The proposed descriptor for a stripe  $C$ .

More specifically, as shown in Figure 5, given a stripe  $C = p_1, p_2, \dots, p_H$ , we compute two matrices, one presenting all distances and the second one

<sup>2</sup>Here we normalise a stripe length  $H$  by the mean length of all stripes in an object.

representing all pairwise orientations of vectors from each  $\mathbf{p}_i$  to each  $\mathbf{p}_m \in \mathbf{C}, m = 1, 2, \dots, H$ . The distance  $E(i, m)$  from  $\mathbf{p}_i$  to  $\mathbf{p}_m$  is defined as the Euclidean distance in the log space:

$$E(i, m) = \log(1 + \|\vec{\mathbf{p}}_i - \vec{\mathbf{p}}_m\|_2) . \quad (1)$$

We add one to the Euclidean distance to make the  $E(i, m)$  positive. The orientation  $\Theta(i, m)$  from  $\mathbf{p}_i$  to  $\mathbf{p}_m$  is defined as the orientation of vector  $\vec{\mathbf{p}}_i - \vec{\mathbf{p}}_m$ :

$$\Theta(i, m) = \angle(\vec{\mathbf{p}}_i - \vec{\mathbf{p}}_m) \in [-\pi, \pi] . \quad (2)$$

Based on Eq. 1 and Eq. 2, a stripe  $\mathbf{C}$  is encoded in two  $H \times H$  matrices  $E$  and  $\Theta$ . Since the matrix  $E$  is symmetrical, we only extract its lower triangular part for calculating the mean distance  $d$ . For the matrix  $\Theta$ , as its absolute values are symmetrical, we extract the upper or the lower triangular part which has more positive values and calculate the mean orientation  $o$ . Finally, together with the normalised stripe length  $l$ , a stripe  $\mathbf{C}$  is represented by:

$$\mathbf{C} = [l, d, o] . \quad (3)$$

## V. STRIPE MATCHING

Let  $\mathbf{A}_1$  and  $\mathbf{A}_2$  denote sets of stripes from two objects  $O_1$  and  $O_2$ , respectively.  $\mathbf{C}_i$  and  $\mathbf{C}'_j$  denote a single stripe in  $\mathbf{A}_1$  and  $\mathbf{A}_2$ ,  $i = 1, 2, \dots, N, j = 1, 2, \dots, M$ . For notational simplicity we assume that  $N \leq M$ . Our aim is to calculate the similarity between  $O_1$  and  $O_2$  using their stripe sets. As each stripe is represented by a three-dimensional feature vector, we calculate the similarity between  $O_1$  and  $O_2$  using the properties of each feature distribution.

Specifically, for two objects  $O_1$  and  $O_2$ , we first remove some redundant stripes from  $\mathbf{A}_2$  to ensure they have the same number of stripes  $N$ . In order to do so, we remove the rounding number  $(M - N)/2$  stripes from two ends of  $\mathbf{A}_2$ . For example, we remove the stripes  $\{\mathbf{C}'_1, \dots, \mathbf{C}'_{(M-N)/2}\}$  and the stripes  $\{\mathbf{C}'_{N-((M-N)/2)+1}, \dots, \mathbf{C}'_N\}$ . The rationale behind this is (1) Stripes which are close to the boundary have less influence on the global structure of an object. (2) Most objects have symmetrical structures. (3) This strategy can avoid the removing of some crucial stripes which have major contribution for object distinction. With the above step,  $\mathbf{A}_1$  and  $\mathbf{A}_2$  have the same number of stripes  $N$ . With the original order, we renumber the index of stripes in  $\mathbf{A}_2$  as:

$$\mathbf{A}_2 = \{\mathbf{C}'_1, \mathbf{C}'_2, \dots, \mathbf{C}'_N\} . \quad (4)$$

As each stripe can be represented by a three-dimensional feature vector  $\mathbf{C}_i = [l_i, d_i, o_i]$ ,  $\mathbf{C}'_i = [l'_i, d'_i, o'_i]$ , we capture the distribution of each feature by its feature values in all stripes. Then the distance between  $O_1$  and  $O_2$  can be calculated by Bhattacharyya distance [20]. Specifically, let  $\mathbf{l}$ ,  $\mathbf{d}$  and  $\mathbf{o}$  denote the distributions of all feature values in  $\mathbf{A}_1$ .  $\mathbf{l}'$ ,  $\mathbf{d}'$  and  $\mathbf{o}'$  denote the distributions of all feature values in  $\mathbf{A}_2$ . For example,  $\mathbf{l} = [l_1, l_2, \dots, l_N]$  and  $\mathbf{l}' = [l'_1, l'_2, \dots, l'_N]$ . Assume

$\mathbf{K} = \{\mathbf{l}, \mathbf{d}, \mathbf{o}\}$  and  $\mathbf{K}' = \{\mathbf{l}', \mathbf{d}', \mathbf{o}'\}$ , the distance between  $O_1$  and  $O_2$  is calculated by

$$s(O_1, O_2) = \frac{1}{3} \sum_{t=1}^3 \lambda_t D_B(\mathbf{K}(t), \mathbf{K}'(t)) \quad (5)$$

where  $\lambda$  is the weight for fusing three features and  $D_B(\mathbf{K}(t), \mathbf{K}'(t))$  denotes the Bhattacharyya distance between two feature distributions. For example,  $D_B(\mathbf{K}(1), \mathbf{K}'(1)) = D_B(\mathbf{l}, \mathbf{l}')$ . In practice,  $\lambda$  can be searched using the heuristic method of Gradient Hill Climbing integrated with Simulated Annealing [38]. Specifically, the Gradient Hill Climbing [39] method starts with randomly selected parameters. Then it changes single parameters iteratively to find a better set of parameters. A fitness function then evaluates whether the new set of parameters performs better or worse. The Simulated Annealing strategy [40] impacts the degree of the changes. In later iterations, the changes to the parameters are becoming smaller. With our preliminary experiments, we set  $\lambda_1 = 0.6$ ,  $\lambda_2 = 0.3$  and  $\lambda_3 = 0.1$  for three features. Furthermore, the Bhattacharyya distance  $D_B(\mathbf{l}, \mathbf{l}')$  is calculated by:

$$D_B(\mathbf{l}, \mathbf{l}') = \frac{1}{4} \ln\left(\frac{1}{4}\left(\frac{\sigma_l^2}{\sigma_{l'}^2} + \frac{\sigma_{l'}^2}{\sigma_l^2} + 2\right)\right) + \frac{1}{4}\left(\frac{(\mu_l - \mu_{l'})^2}{\sigma_l^2 + \sigma_{l'}^2}\right) \quad (6)$$

where  $\sigma$  and  $\mu$  are the variance and mean of a feature distribution, respectively. Similar to  $D_B(\mathbf{K}(1), \mathbf{K}'(1))$ ,  $D_B(\mathbf{K}(2), \mathbf{K}'(2)) = D_B(\mathbf{d}, \mathbf{d}')$  and  $D_B(\mathbf{K}(3), \mathbf{K}'(3)) = D_B(\mathbf{o}, \mathbf{o}')$  can also be calculated with Eq. 6.

## VI. EXPERIMENTS

In this section we first introduce the dataset we used for the experiments. After that, we evaluate and compare the performance of the proposed method with some traditional methods to illustrate our advantages. Lastly, we analyse the computational complexity of the proposed method. The experiments in this paper are performed on a laptop with Intel Core i7 2.2GHz CPU, 8.00GB memory and 64-bit Windows 8.1 OS. All methods in our experiments are implemented in Matlab R2015a.

### A. Dataset

To validate the idea of our proposed method on real-life objects, we organised a dataset namely Daily100 from daily life. The Daily100 database includes 100 objects with 10 classes, such as apple, banana, book, chips, chopstick, egg, orange, pear, pen and bottle cap (the first row in Figure 6). For each object, we also generated and collected its correlated stripes and shape for the experiment. From Figure 6 we can see that some objects (e.g. apple and orange) are really difficult to be distinguished using only their shapes.

### B. Experiment Performance

Based on the Daily100 dataset, we perform two experiments within the object retrieval frame work. Specifically, in the first experiment, we compare the global object retrieval performances between stripes and shapes. In the second experiment, more detailed comparisons in each object class are illustrated and discussed.

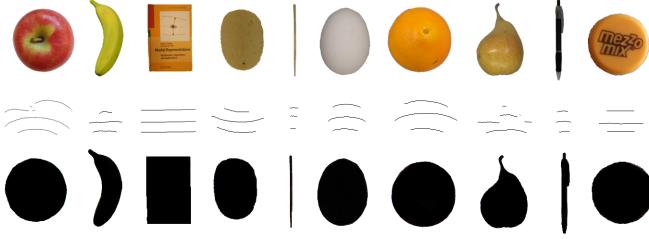


Fig. 6. Sample objects of the proposed Daily100 dataset. The first row illustrates the original objects. The second and third row show the sample stripes and shape of each object, respectively.

Table I depicts the matching performance of the proposed method and other shape-based approaches. We use each object as a query and retrieve the 10 most similar objects among the whole dataset. The final value in each position is counter values that are obtained by checking retrieval results using all the 100 objects as queries. For example, the fourth position in the row of our method shows that from 100 retrieval results in this position, 89 objects have the same class as the query objects. We can clearly observe that the proposed method achieves the best results among all the other methods. The main reason is that since most objects have similar shapes (e.g. apple, orange and bottle cap), it is hard to distinguish them using only their shape features. Different from the shape features, the proposed method captures and preserves deformations on object surfaces using stripe sets. Thus, the proposed method can distinguish objects even if their shapes are similar to each other.

TABLE I  
OBJECT RETRIEVAL COMPARISON BETWEEN THE PROPOSED METHOD AND SHAPE-BASED APPROACHES: SHAPE CONTEXT (SC), INNER DISTANCE (ID) AND PATH SIMILARITY (PS).

	1	2	3	4	5	6	7	8	9	10
SC [41]	100	79	84	71	75	74	64	67	58	48
ID [42]	100	80	77	70	68	68	64	66	54	41
PS [18]	100	81	79	78	76	71	65	60	55	53
HF [43]	100	80	71	67	64	54	52	43	49	51
<b>Our</b>	<b>100</b>	<b>95</b>	<b>91</b>	<b>89</b>	<b>86</b>	<b>77</b>	<b>73</b>	<b>72</b>	<b>66</b>	<b>63</b>

In order to perform a more detailed analysis of object matching, we report the mean matching accuracy in each object class in Table II. In this table, the mean accuracy is calculated by the matching accuracy on each query object. Specifically, we use an object as the query and retrieve the 10 most similar objects among the whole dataset. Within these 10 objects, we count how many objects have the same class as the query object. The matching accuracy of the query object is then calculated by the ratio between the matched objects and 10. We use all the 100 objects as queries and calculate the mean accuracy for each class. In Table II, we can observe that for most classes, our method achieves the best accuracy. For example, since apple, egg, orange and bottle cap have a very similar shape, the performances of shape-based methods in these classes are not promising. As the collected stripe sets have different geometrical properties in these classes, our

method achieves the best performance. However, considering the classes of banana and pen, our method performs worse than the shape-based approaches. The main reason is that since both banana and pen have very similar surface deformation, the proposed stripe-based approach cannot robustly distinguish them using only stripes. In order to improve the accuracy, a proper combination of shape descriptors and the proposed stripe descriptor can improve the matching accuracy over the individual descriptor.

TABLE II  
COMPARISON OF MEAN MATCHING ACCURACY (%) IN EACH CLASS.

	apple	banana	book	chips	chopstick	egg	orange	pear	pen	cap
SC [41]	46	100	79	73	87	77	51	79	100	28
HF [43]	53	82	48	61	51	43	54	97	100	30
<b>Our</b>	<b>100</b>	<b>78</b>	<b>48</b>	<b>100</b>	<b>100</b>	<b>96</b>	<b>84</b>	<b>100</b>	<b>42</b>	<b>64</b>

### C. Computational Complexity and Runtime

We now analyse the computational complexity of the proposed hierarchical skeleton generation and matching approaches. (1) For stripe set generation, the time complexity is in the order of  $O(Nl)$ , where  $N$  is the number of stripes on each object and  $l$  is the mean stripe length. This is because our stripe can be directly generated from laser scanning lines. Thus, for each stripe, we only need  $O(l)$  for thinning and noise removing. (2) For stripe representation, since we generated the point context descriptor using sample points along the stripe path, the time complexity is  $O(H^2)$  where  $H$  is the number of sample points. Considering there are multiple stripes for each object, the global complexity for object representation is  $O(NH^2)$ . (3) For object matching, as we directly employ the Bhattacharyya distance on three-dimensional vectors, the complexity is  $O(3)$ . However, for each dimension, we need to calculate the mean value on the distance and orientation matrix, the global complexity for object matching is  $O(3H)$ . Thus, the total complexity of our method is  $O(Nl) + O(NH^2) + O(3N)$ . By dropping the constant number, our time complexity is bounded by  $O(Nl) + O(NH^2)$ .

Here we report the computation time based on the Daily100 dataset with the experimental environment introduced above. On average, the shape resolution in this dataset is  $600 \times 712$ . For each object, the mean stripe number is 140. Together with object representation and matching, the proposed method takes 0.0375 hours while the Shape Context [41] method takes 1.6223 hours. Thus, our method can dramatically reduce the runtime while achieving promising matching accuracy. However, please notice that our code is not optimised, and its faster implementation is possible by optimising loops, settings and programming language, etc. Thus, there are still plenty of opportunities to reduce the running time.

## VII. CONCLUSION AND FUTURE WORK

A novel 3D object matching method based on the similarity of stripes is presented. The most significant contribution of this paper is the novel approach to 3D object matching. We

represent an object as a set of stripes which are directly collected from laser scanning lines. The distance between objects is computed using the Bhattacharyya distance based on stripe features. The proposed approach does not require any complicated strategies for 3D object reconstruction as well as the feature point corresponding. Thus, our method can dramatically reduce computational complexity for 3D object matching. In addition to low time costs, our method achieves a promising performance on some challenging objects compared to the traditional 2D shape-based approaches. In the future, we will try to optimise our stripe collection strategy for adapting hand-held laser scanning devices. Moreover, we will compare our approach to other 3D object matching methods.

## REFERENCES

- [1] B. Drost and S. Ilic, "Graph-based deformable 3d object matching," in *Pattern Recognition*, 2015, vol. 9358, pp. 222–233.
- [2] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3d models with shape distributions," in *International Conference on Shape Modeling and Applications*, 2001, pp. 154–166.
- [3] R. J. Campbell and P. J. Flynn, "A survey of free-form object representation and recognition techniques," *Computer Vision and Image Understanding*, vol. 81, no. 2, pp. 166–210, 2001.
- [4] C. Hong, J. Yu, J. You, X. Chen, and D. Tao, "Multi-view ensemble manifold regularization for 3d object recognition," *Information Sciences*, vol. 320, pp. 395–405, 2015.
- [5] B. Leng, J. Zeng, M. Yao, and Z. Xiong, "3d object retrieval with multitopic model combining relevance feedback and lda model," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 94–105, 2015.
- [6] Y. Yu, J. Li, H. Guan, F. Jia, and C. Wang, "Three-dimensional object matching in mobile laser scanning point clouds," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 3, pp. 492–496, 2015.
- [7] F. Bernardini, C. L. Bajaj, J. Chen, and D. R. Schikore, "Automatic reconstruction of 3d cad models from digital scans," *International Journal of Computational Geometry and Applications*, vol. 9, pp. 327–369, 1999.
- [8] R. Maier, J. Sturm, and D. Cremers, "Submap-based bundle adjustment for 3d reconstruction from rgbd data," in *Pattern Recognition*, 2014, vol. 8753, pp. 54–65.
- [9] W. Kehl, N. Navab, and S. Ilic, "Coloured signed distance fields for full 3d object reconstruction," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.
- [10] D. Lefloch, R. Nair, F. Lenzen, H. Schifer, L. Streeter, M. Cree, R. Koch, and A. Kolb, "Technical foundation and calibration methods for time-of-flight cameras," in *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*. Springer Berlin Heidelberg, 2013, vol. 8200, pp. 3–24.
- [11] M. Kedzierski and A. Fryskowska, "Methods of laser scanning point clouds integration in precise 3d building modelling," *Measurement*, vol. 74, pp. 221–232, 2015.
- [12] R. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4.
- [13] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce, "A tensor-based algorithm for high-order graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2383–2395, 2011.
- [14] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1482–1489.
- [15] C. Yang, C. Feinen, O. Tiebe, K. Shirahama, and M. Grzegorzek, "Shape-based object matching using point context," in *International Conference on Multimedia Retrieval*, 2015, pp. 519–522.
- [16] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios, "Dense non-rigid surface registration using high-order graph matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 382–389.
- [17] R. Jonker and T. Volgenant, "Improving the hungarian assignment algorithm," *Operations Research Letters*, vol. 5, no. 4, pp. 171–175, 1986.
- [18] X. Bai and L. Latecki, "Path similarity skeleton graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1282–1292, 2008.
- [19] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman, "Linear time euclidean distance transform algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 529–533, 1995.
- [20] S. Dubuisson, "The computation of the bhattacharyya distance between histograms without histograms," in *International Conference on Image Processing Theory Tools and Applications*, 2010, pp. 373–378.
- [21] F. Bernardini and H. Rushmeier, "The 3d model acquisition pipeline," *Computer Graphics Forum*, vol. 21, no. 2, pp. 149–172, 2002.
- [22] R. A. Brooks, "Symbolic reasoning among 3-d models and 2-d images," *Artificial Intelligence*, vol. 17, no. 1, pp. 285–348, 1981.
- [23] A. P. Pentland, "Perceptual organization and the representation of natural form," *Artificial Intelligence*, vol. 28, no. 3, pp. 293–331, 1986.
- [24] B. Leibe, A. Leonardis, and B. Schiele, "An implicit shape model for combined object categorization and segmentation," in *Towards Category-Level Object Recognition*, 2006, pp. 496–510.
- [25] M. Stark, M. Goesele, and B. Schiele, "Back to the future: Learning shape models from 3d cad data," in *British Machine Vision Conference*, 2010, pp. 1–11.
- [26] S. Savarese, "Estimating the aspect layout of object categories," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3410–3417.
- [27] L. G. Roberts, "Machine perception of three-dimensional soups," Ph.D. dissertation, Massachusetts Institute of Technology, 1963.
- [28] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*, 2012, pp. 746–760.
- [29] X. Liu, Y. Zhao, and S.-c. Zhu, "Single-view 3d scene parsing by attributed grammar," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 684–691.
- [30] M. Z. Zia, M. Stark, and K. Schindler, "Towards scene understanding with detailed 3d object representations," *International Journal of Computer Vision*, vol. 112, no. 2, pp. 188–203, 2014.
- [31] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 1–9, 2009.
- [32] S. Wang, Y. Wang, M. Jin, X. D. Gu, and D. Samaras, "Conformal geometry and its applications on 3d shape matching, recognition, and stitching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 7, pp. 1209–1220, 2007.
- [33] W. Zeng, Y. Zeng, Y. Wang, X. Yin, X. Gu, and D. Samaras, "3d non-rigid surface matching and registration based on holomorphic differentials," in *European Conference on Computer Vision*, 2008, pp. 1–14.
- [34] C. Yang, O. Tiebe, K. Shirahama, and M. Grzegorzek, "Object matching with hierarchical skeletons," *Pattern Recognition*, 2016.
- [35] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, and B. Chen, "L1-medial skeleton of point cloud," *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 1–8, 2013.
- [36] X. Bai, L. J. Latecki, and W.-Y. Liu, "Skeleton pruning by contour partitioning with discrete curve evolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 449–462, 2007.
- [37] R. A. C. Cruces, H. C. Schneider, and J. Wahrburg, *Cooperative robotic system to support surgical interventions*. INTECH Open Access Publisher, 2008, pp. 481–490.
- [38] C. Yang, O. Tiebe, P. Pietsch, C. Feinen, U. Kelter, and M. Grzegorzek, "Shape-based object retrieval and classification with supervised optimisation," in *International Conference on Pattern Recognition Applications and Methods*, 2015, pp. 204–211.
- [39] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall Press, 2009.
- [40] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, pp. 671–680, 1983.
- [41] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.
- [42] H. Ling and D. Jacobs, "Shape classification using the inner-distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 286–299, 2007.
- [43] J. Wang, X. Bai, X. You, W. Liu, and L. J. Latecki, "Shape matching and classification using height functions," *Pattern Recognition Letters*, vol. 33, no. 2, pp. 134–143, 2012.