IEEE ICIP 2014

3D OBJECT RETRIEVAL BY 3D CURVE MATCHING

Christian Feinen, Joanna Czajkowska, Marcin Grzegorzek

Pattern Recognition Group, University of Siegen
Department of Electrical Engineering and Computer Science
Hoelderlinstrasse 3, D-57076 Siegen, Germany

Longin Jan Latecki

Temple University
Department of Computer and Information Sciences
1805 N. Broad St. Philadelphia, PA 19122, USA

ABSTRACT

In this paper, we introduce a novel approach to 3D object retrieval by 3D curve matching. First, we project 2D object edges obtained from a depth image into 3D space. Second, we find distinctive feature points on the object. Third, we represent the shortest paths between the features by robust descriptors invariant to rotation, scaling, and translation. Finally, we match two 3D objects using the Maximum Weight Subgraph search. The most important contribution of this paper is the powerful object representation by 3D curves together with the corresponding matching algorithm. Excellent retrieval results achieved with our method show its benefits compared to the state-of-the-art.

Index Terms— 3D Object Retrieval, 3D Curve Matching

1. INTRODUCTION

Researchers early realised that accuracy and robustness of the object segmentation, detection, and recognition can be remarkably increased, if 2D data is enriched with 2.5D or 3D information. In the most related work [1] to our approach the same 3D line segment data structure to represent 3D objects is employed. However, instead of using a correspondence graph of all pairs of 3D line segments, we first detect feature points and subsequently use them for computing the Maximum Weight Subgraph. Nguyen et al. [2] describe an algorithm that combines 2D images and 3D point clouds. In our approach, 2D lines are first extracted from a 2D image and then back-projected to get a set of 3D points for each line. Based on these point sets, 3D lines are estimated. In another relevant method [3], the authors use range data for reliable silhouette extraction that represents an object for recognition. Payet and Todorovic [4] address view-invariant object detection and pose estimation from a single image by using contours as basis features. In this approach, a few view-dependent shape templates are jointly used for detecting object occurrences and estimating their 3D poses. However, their work requires training examples of arbitrary views of an object to learn a sparse object model.

A highly related 2D skeleton matching approach was proposed by Bai and Latecki [5]. They match skeletons based on dissimilarities between the shortest paths connecting their endpoints. For this, each shortest path is sampled by a fixed number of points. Each of these points is represented by a radius of a maximum disc that has been determined for it during the skeletonisation process. In this way, every shortest path is represented by a vector of radii. Subsequently, matching costs for all pairs of skeleton endpoints are calculated. This is done by an approach called Optimal Subsequence Bijection (OSB). Afterwards, all output values of the OSB are rearranged and given as input to the Hungarian algorithm, where the matching problem is reduced to a classical assignment problem of a bipartite graph.

The input to our approach is a depth image preprocessed by the methods described in [6] and [1]. First, the Canny edge detector is applied and the obtained object contours are projected to 3D. Second, a local coordinate system (LCS) is determined. Third, representative feature points are identified and shortest paths between them are computed (Section 2). Since some of these paths are ambiguous, a modification of the Dijkstra algorithm is needed (Section 3). In the next step a shortest path descriptor based on relative angles is generated (Section 4). Finally, we perform the matching (Section 5) by transforming the problem into the search of Maximum Weight Subgraphs (MWS). At the end, the comparison of the user generated model and the object captured by the depth device is conducted.

2. OBJECT FEATURE POINTS

Our proposed matching procedure utilises the concept of shortest paths introduced in [5]. Consequently, object features are necessary, because they act like the skeleton end nodes described in the original method. To simplify their computation, we assume that the objects are symmetric and are approximable by cube-similar geometries. The feature detection process is depicted and explained in Figure 1. In

The research activity leading to this article has been supported by the German Research Foundation within the Research Training Group 1564 “Imaging New Modalities”.

We would like to acknowledge the NSF grants IIS-1302164 and OIA-1027897.

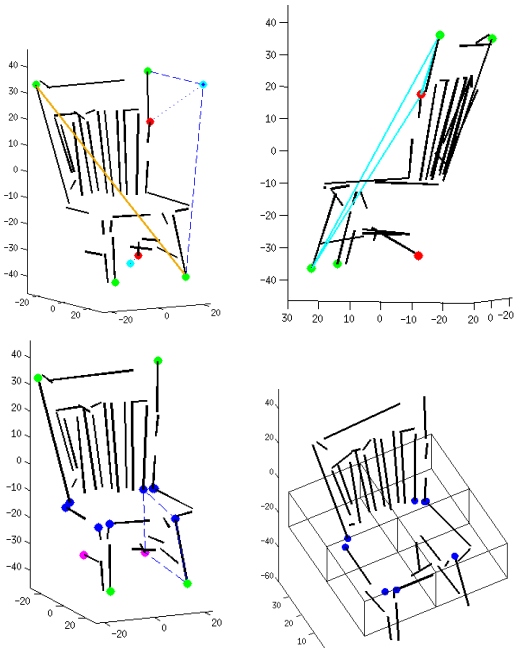

Fig. 1: Top left: initial feature points (green); virtual feature points (cyan); closest object correspondences (red); imaginary diagonal (orange). Top right: false positive removal. Bottom left: valid feature points after false positive removal (pink); corner point candidates after rectangular fitting (blue). Bottom right: 2D cluster scheme for mid-level features according to their position inside the LCS.

order to perform the shortest path analysis, feature connections need to be established. Therefore, we search for 3D curves, which can be interpreted as possible links between two points. Thus, all 3D curves have to be taken into account for each feature point pair, with the aim to find at least one segment satisfying the following constraints. Firstly, the length of the curve has to be greater than the average of all line lengths. Secondly, the ratio of lengths between the 3D curve and the virtual link has to be greater than $T^{(R)}$. Thirdly, the distance of the curve's orientation to the one of the virtual segment has to be below $T^{(W)}$. Fourthly, the closest distance from the currently observed feature point to either start or endpoint of the curve (used for verifying the virtual link), has to be below $T^{(C)}$. If all these constraints are fulfilled, the observed feature pair is marked as connected. Even if the result is suffering from a small amount of wrong connections, it does not decrease the robustness of our methodology due to our modified shortest path algorithm.

3. MODIFIED DIJKSTRA ALGORITHM

Even if we compare objects of the same class, it can happen that paths are not similar or ambiguous, although all features are found and connections are established correctly. For example, in case of the user-generated chair model, there are at least two paths from the lower left front leg to the upper right back with identical lengths. If there is no appropriate criterion, the algorithm has to choose non-deterministically one of these paths. Other kinds of path irritations are caused by inaccuracies occurring during the depth acquisition, the line approximation or the curve back-projection. Consequently, there is a high risk to retrieve another result as the actual shortest path. In this case, the dissimilarity would increase drastically and the whole matching algorithm would fail. To tackle this problem, we propose two additional mechanisms to improve the result of the Dijkstra algorithm. Firstly, all shortest paths of the query object are monitored. This is done by a simple description in terms of left/right, up/down and front/back with the objective to depict the paths taken during the computation. These path descriptors are then given as input to the shortest path computation of the target object. However, if connections are missing, it is not possible to follow the instructions of the path descriptor anymore. For these situations, we formulated a second rule: The algorithm is allowed to establish one missing feature connection based on the information given by the query path descriptor. If this new connection does not support the further path computation, the algorithm terminates.

4. SHORTEST PATH REPRESENTATION

The shortest path representation is a crucial factor regarding the actual matching process (Section 5). It constitutes the only way to compute the cost values that are required to match feature pairs. For this, a feature vector is introduced consisting of K' tuples $\boldsymbol{r}_m = ((\alpha,\beta)_{m,1},\ldots,(\alpha,\beta)_{m,K'})^{\mathrm{T}}$, where α and β denote angles and m is the corresponding path. By employing these two angles, we are able to uniquely describe a 3D point. Therefore, we firstly compute the angle α between the point vector and $\boldsymbol{z}$-axis of our LCS. Secondly, the point is projected onto the $\boldsymbol{x}$,$\boldsymbol{y}$-plane, where we calculate β as shown in Figure 2. In order to describe the whole path m based on this angle constellation, it is sampled by K' equidistantly distributed points (see Figure 2). Afterwards, each sample point is linked with the feature point from which the path is emanating with the result of K' sample vectors, which get finally described by α and β. Attributable to the use of relative angles based on the object's LCS, our path descriptor is invariant to rotation, translation and scaling (RTS). The actual dissimilarity between the paths is calculated based on their representation vectors. Therefore, the vector $\boldsymbol{r}_m$ is separated into two sub-vectors $\boldsymbol{r}_m^{(\alpha)}$ and $\boldsymbol{r}_m^{(\beta)}$. The sub-vectors, in turn, are given as input to a distance measurement approach, e.g., DTW or EMD.

5. FEATURE POINT MATCHING

The proposed matching principle works in analogy to the Path Similarity Skeleton Graph Matching algorithm proposed in [5], but we adapted it to depth sensory data and a completely

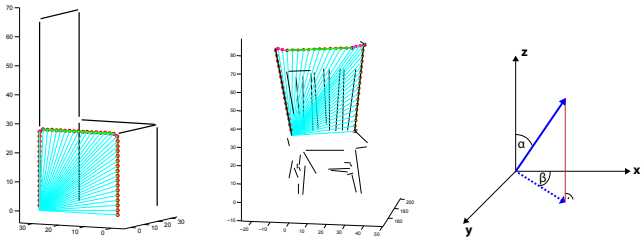

Fig. 2: Demonstration of our path representation. The red circles indicate the position of the sample points, the lines drawn in cyan are the sample vectors ($|\boldsymbol{g}_i|$) between the currently observed feature and each sample point on the path used for the descriptor.

different object representation. Our matching strategy also involves the Optimal Subsequence Bijection (OSB) procedure proposed by Latecki et al. [7]. OSB can be used for elastic matching of sequences of different lengths. It is similar to the Dynamic Time Warping and the Longest Common Subsequence (LCSS) algorithms, but outperformed these methods during a comprehensive evaluation. Its key property is its capability to exclude outliers from the matching. Moreover, it is suitable for partial matching and it preserves the order of points during traversing the graph.

Attributed to the use of the OSB, our method requires a deterministic and reproducible scheme regarding the order of feature points; a problem that can easily be solved in 2D, but not in 3D. In order to do so, first, the z coordinates p_z of all features are extracted $\boldsymbol{p}_{i=1,\ldots,|\Theta|} = (p_{i,x}, p_{i,y}, p_{i,z})^{\mathrm{T}}$ and arranged in descending order inside a vector $\boldsymbol{s}$. By subtracting neighbouring elements in $\boldsymbol{s}$, we retrieve a new vector $\boldsymbol{s'} = (s_1 - s_2, s_2 - s_3, \ldots, s_{|\Theta|-1} - s_{|\Theta|})^{\mathrm{T}}$, where the difference values are analysed in terms of peak occurrences as shown in Figure 3. For each peak a z value is computed based on the two elements in $\boldsymbol{s}$ leading to this peak. Afterwards, these z values are used to cluster the features along the $\boldsymbol{z}$-axis. Finally, the points inside each cluster are ordered counter-clockwise in 2D. The technique completes with recombining all clusters. Therefore, the currently processed feature point, accommodated in one of the z clusters, is projected in the 2D space of the remaining ones, respectively (Figure 3). The closest point to this projection identifies the start for the counting operation. The final feature matching and the overall object similarity is computed with the Maximum Weight Subgraphs [8] considering mutual exclusion (mutex) constraints on weighted graphs.

6. EXPERIMENTS

To be comparable to other methods, we evaluated our methodology on the same chair database as used in [1] consisting of 213 objects. Regarding the set of *stands* with 40 instances, we decided to extend it by further 67 objects. Additionally, we introduced a third dataset consisting of images of tables that encompasses 70 objects. All objects have been recorded with

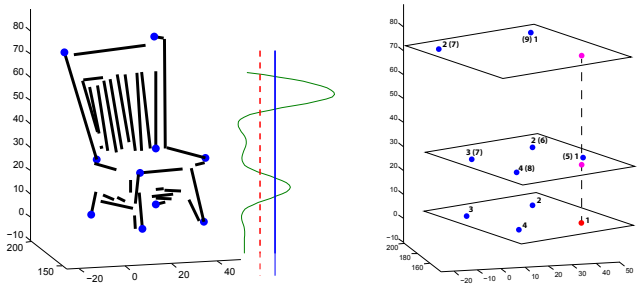

Fig. 3: Left: chair object and its feature points (blue). The plot next to it visualises the values stored in $\boldsymbol{s'}$; peaks can be detected by using the average (red dashed line) or the standard derivation (blue line) of $\boldsymbol{s'}$. Right: feature points (blue) clustered according to the peaks in $\boldsymbol{s'}$. The points in each cluster are ordered counter-clockwise in relation to the currently processed point (red), that is projected (pink) into the remaining clusters.

a RGB-D device (Kinect) from different viewpoints within complex real-world scenes. Hence, some parts of the objects are occluded, missing or distorted by outliers. Example images can be seen in Figure 4. Our ground truth data

Fig. 4: Example images used for evaluation.

was generated manually and each entry corresponds to one scene object that is described in terms of "chair", "stand", "table" and "undefined". The similarity values obtained with our method used for object retrieval and precision-recall graphs were generated for quantitative evaluation. Furthermore, we also assessed the average precision (AP) as introduced in [9]. All thresholds used in our experiments are based on the mean and on the standard deviation derived by incorporating all 3D curves. During our experiments we used the Dynamic Time Warping as well as the Earth Mover's Distance to calculate the distances between the paths (Section 4).

Point Detection: To evaluate the robustness and accuracy of our feature detection method, a feature ground truth has been created for the most challenging object type in our dataset, namely the chair. As presented in Figure 5, we obtained excellent results, especially, if one considers the quality of our data. The threshold T1 is responsible for the triangular and T2 for the rectangular fitting. All configurations led to good results, except T1=0.3 and T2=0.1, where the rectangular fitting threshold is too restrictive.

Chair Database: First, we evaluated our approach on the

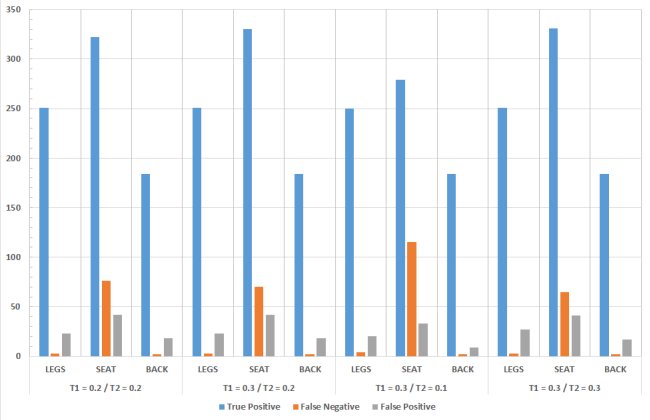

Fig. 5: Overview about four different threshold combinations (horizontally plotted) regarding their power in terms of detected features (vertically plotted). All of them perform as expected and lead to good results, expect the configuration T1=0.3 and T2=0.1.

chair database used in [1]. This database is a real challenge for our method, since many chairs are of poor quality and accommodate heavy outliers. Moreover, the dataset includes objects which principally cannot be recognised by our method, e.g., office chairs. However, even in presence of these negative factors, we have been able to obtain very good results as shown in Figure 6. Moreover, the proposed method

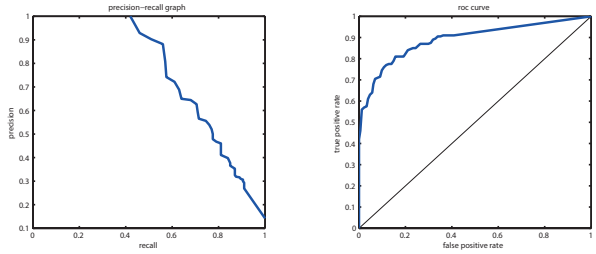

Fig. 6: Recall-Precision Graph (left) and ROC Curve (right) for the chair database used in [1].

outperforms all of its competitors. Table 1 shows the AP of our approach compared to several state-of-the-art methods.

Stand Database: Figure 7 shows the results of our stand database evaluation. Even in this case, we obtained excellent results, although we expected more irritations caused by our path descriptor. Nevertheless, despite the fact that we extended the database with new objects, it only includes half of the amount of true positives compared to the previous one.

Table Database: This evaluation was operating on our own database. It is the smallest database in our evaluation. However, it has been chosen deliberately, since this object class also provides a high risk of being ambiguous to the other ones. This makes it even more attractable as we retrieve very

Method		**Average Precision**
Our Method		0.760
Ma et al.	(2012)	0.714
Janoch et al.	(2011)	0.438
Felzenswalb et al.	(2010)	0.419
Ferrari et al.	(2010)	0.351

Table 1: The table shows the average precision of our method compared to other state-of-the-art approaches. The most interesting information is its comparison to *Ma et al.* since their approach is the most related one to ours.

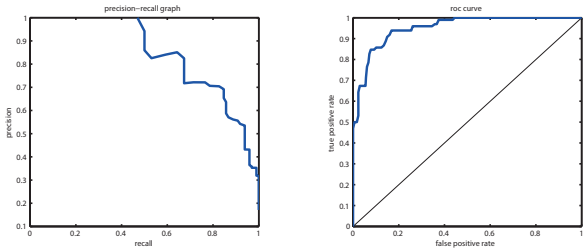

Fig. 7: Recall-Precision Graph (left) and ROC Curve (right) for the stand database, AP = 0.8484.

promising results as shown in Figure 8. It seems, that the different object proportions can be represented adequately by our path descriptor.

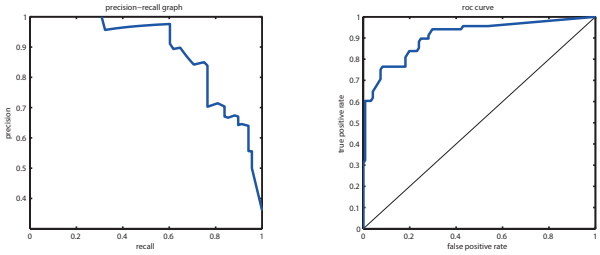

Fig. 8: Recall-Precision Graph (left) and ROC Curve (right) for the table database, AP = 0.8840.

7. CONCLUSION

In this paper we presented a robust and generic 3D object recognition algorithm based on 3D curves. To underline the generic aspect, a user-generated and, hence, strongly abstracted model is introduced as a query object. Moreover, since this model is processed equally compared to the target one, the query can be easily replaced by other objects. Furthermore, we proposed an intelligent method to localise feature points by incorporating a local coordinate system. Finally, we demonstrated how these steps are combined to recognise 3D objects. In a comprehensive setup of experiments we outperformed all related state-of-the-art methods.

8. REFERENCES

[1] T. Ma, M. Yi, and L. J. Latecki, "View-Invariant Object Detection by Matching 3D Contours," in *ACCV Workshops*, Jong-Il Park and Junmo Kim, Eds. 2013, vol. 7729 of *Lecture Notes in Computer Science*, pp. 183–196, Springer Berlin Heidelberg.

[2] T. B. Nguyen and L. Sukhan, "Accurate 3D Lines Detection Using Stereo Camera," in *Int. Sym. on Assembly and Manufacturing, ISAM.*, 2009, pp. 304–309.

[3] S. Stiene, K. Lingemann, A. Nuchter, and J. Hertzberg, "Contour-Based Object Detection in Range Images," in *Int. Sym. on 3D Data Processing, Visualization, and Transmission*, 2006, pp. 168–175.

[4] N. Payet and S. Todorovic, "From Contours to 3D Object Detection and Pose Estimation," in *ICCV*, 2011, pp. 983–990.

[5] X. Bai and L. J. Latecki, "Path Similarity Skeleton Graph Matching," *IEEE PAMI*, vol. 30, no. 7, pp. 1282–1292, 2008.

[6] M. Yi, Y. Yang, W. Qi, Y. Zhou, Y. Li, Z. Pizlo, and L. J. Latecki, "Navigation toward Non-static Target Object Using Footprint Detection Based Tracking," in *Proceedings of the 11th ACCV*. 2013, ACCV, pp. 389–400, Springer-Verlag.

[7] L. J. Latecki, Q. Wang, S. Koknar-Tezel, and V. Megalooikonomou, "Optimal Subsequence Bijection," in *Int. Conf. on Data Mining, ICDM*, 2007, pp. 565–570.

[8] T. Ma and L. J. Latecki, "Maximum Weight Cliques with Mutex Constraints for Video Object Segmentation," in *CVPR*, 2012, pp. 670–677.

[9] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, June 2010.