Towards an adaptive system for lifelong object modelling

Johan Ekekrantz*, Nils Bore*, Rares Ambrus*, John Folkesson* and Patric Jensfelt*

Abstract— **In this paper, a system for incrementally building and maintaining a database of 3D objects for robots with long run times is presented. The system is a step towards lifelong autonomous object modelling using a mobile robot. The proposed solution iteratively fuses observations as they arrive into better and better models. By greedily allowing the system to fuse data, mistakes can be made. The system continuously seek to detect and remove such errors, without the need for batch updates using all known data at once.**

I. INTRODUCTION

Robots operating in dynamic unstructured environments, without expert assistance and for extended periods of time is slowly becoming reality. In such situations robots cannot rely solely on prior knowledge of the environment. Instead, robots are required to perceive and learn from gathered data autonomously, continuously improving their understanding of the world. Many important robotic techniques that enable robots to perform meaningful behaviours, such as manipulation, grasping and localization are highly dependent on the ability to detect and model objects in the environment during execution. Autonomous object modelling is therefore an important field of study.

Unfortunately, because of occlusions, object models created from sensors such as cameras or laser scanners, taken from a single point of view cannot provide coverage of the whole object. The solution is to use multiple points of view of the object. These views can be acquired by actively moving the robot around the object or by having the robot passively observing as objects move around in the environment. Both methods provide different advantages.

For most applications, building a model from multiple views requires the data to be put into a single shared coordinate system. This is equivalent to computing the relative positions of the sensor to the object at all available views.

In the case where the robot acquires multiple views of the object by actively moving around the object, the relative positions of the views relative to the object can be found using the robot localization. Robot localization can nowadays be accurately and robustly performed using a variety of sensors. The obvious disadvantage of the active approach is that some of the views required to create a complete model are likely to be inaccessible due to physical obstacles in the environment. Navigating a robot around an object can also be impractical because of the time it takes to acquire the required data.

*The authors are with the Centre for Autonomous System at KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden {ekz,nbore,raambrus,johnf,patric}@csc.kth.se

The passive approach can eventually get around the problems of the active approach by waiting for the object to be visible from otherwise inaccessible viewpoints. The passive approach on the other hand requires a database of object models in which a new object can be re-recognized. Once recognized in the database, the new view of the object must be aligned to the object in the database. For pointclouds or images, aligning the data is usually referred to as registering the data. Both pointcloud registration and object recognition techniques are prone to errors. It is therefore important that a passive system is robust to such issues. A byproduct of running a passive system is that the robot becomes aware of when and where a specified object were be found, something that can be leveraged by the robot to perform tasks such as for example fetching a specific object.

In this paper we propose a hybrid system able of incorporating views from both the active and and passive approaches into the same system. The system is set up like a passive system but can incorporate groups of aligned views from the active component into the models maintained by the passive component, providing the advantages of both approaches.

The proposed system autonomously and incrementally builds a database of object models. Using the MetaRooms [1] framework to gather segmented data autonomously, an autonomous robot navigates around an a priori created map of the environment. Once the robot arrives at some specified location, an Asus Primesense camera mounted on a pan tilt unit takes a set of RGBD images of the area surrounding the robot. Object segmentation is performed by detecting data in the new pictures that occlude data captured at the same location at some point earlier in time. This allows the robot to autonomously detect and segment moving objects from the static background.

Using the system presented in [2], the robot is able to plan views and navigate around a detected object to gather data.

Once segmented RGBD images are available, the segments are inserted into a database of object models. If possible, the new segments are integrated into previously created object models. By continuously incorporating data into previously found models, the database size is reduced. Given that the database contains fewer and more complete models, the recognition task is simplified, improving the scalability of the system.

By incrementally fusing multiple views as data comes in, the system runs the risk of incorporating and propagating errors into the database of objects. For a system aimed at lifelong learning, such flaws can be catastrophic. We therefore introduce a method for detecting and correcting such errors as additional data is provided, improving the robustness of the system.

The contribution of this paper is a robust framework for autonomous object modelling. The framework is naturally able to detect and recover from previous errors as more data becomes available to the robot.

II. RELATED WORK

Object reconstruction is an interesting problem with a long history. A good survey paper of the field can be found in [3]. Object reconstruction can be performed with different levels of automation. Our system differs from the vast majority of papers on the fact that we perform fully autonomous, online, object detection and reconstruction using a mobile robot.

In [4][5][6] an object is placed on a flat textured surface. Camera tracking is performed as a RGB-D camera is moved around the object, which is then segmented from the flat surface using plane segmentation. The data is then fused by the systems to create a final object model.

In [7][8] object modelling is performed using a static camera and a handheld object. Since the background cannot be used for registration, estimating the positions of the camera relative to the the object model gets a lot harder. The advantage lies in the fact that the method of capturing requires less setup because it is often more convenient for the user to move the object than the camera.

In [2] a system for fully automated detection and object modelling is presented. A robot autonomously detects objects from the static background using the MetaRooms approach presented in [1] and [9]. Once an object has been detected, the robot navigates around the detected object, as best as it can, acquiring multiple views of the object. The relative poses of the camera are found using the camera tracker from [5]. Our system uses the same segmentation and viewplanning engine as [2]. However our system is able to fuse observations taken at multiple points in time. This reduces the risk of modelling the same object several times and potentially improves the models in the vast majority of real world cases where the robot is unable to plan views from all angles of the object.

In [10] a batch processing system for unsupervised object discovery is presented. The system uses the change detection algorithm of [11] to detect and segment objects from the static background. This work is similar to [1] and [9] in that the objects are segmented by detecting occlusions in the registered sensor data. [10] differs from our work in that it is a batch-processing system, whereas our system is an iterative system where data is incrementally added. The presented system in [10] also significantly differs from our system in that all detections are exhaustively aligned and compared in a pairwise fashion, whereas our system only registers new data to a smaller set of known models. Once all detections have been registered, the system performs clustering on all detections simultaneously.

In [12] a system that uses global shape and color features for matching segments discovered from change detection or segmentation is presented. The system is able to merge and update the feature representations that represents objects online. In contrast to [12], our system maintains fused 3D models on which it performs merging. Further, because we maintain and work directly on the 3D models, our system allows for a principled way of detecting and correcting model with erroneously merged observations.

III. AUTOMATIC OBJECT MODELLING FRAMEWORK

The proposed system performs long term object reconstruction by maintaining and updating a database of models. A model is represented by a set of range images, segmentation masks defining which pixels in the corresponding range image that are a part of the model and estimated relative poses of the object in the range images and segmentation masks.

The key principle of the system is to incrementally merge incomplete models in order to improve the completeness of the models. The system is able to, as more data becomes available from the robot, change the grouping of views to better represent the data, in case mistakes were made because the of incomplete data or poorly aligned views.

In fig.(1) a sequence of views taken at different times of a single physical object is input into the system. The models found in the database are shown at different times in the sequence.

Using the system of [9], one can acquire single image models of moved objects autonomously on the robot. The single image models can then be fed into the database of models.

The system then perform pairwise registration between the added model and the other models in the database.

After the registration has been performed, the system seeks to determine if the registration was successful and if the two models are actually the same underlying object. Rather than using just the registration fitness score, we propose that the system take advantage of what would normally be a drawback of cameras and range sensors: self occlusion. Using graph partitioning algorithms, we seek to find the partition of frames into groups which results in the maximal surface overlap and minimal self occlusion. Details can be found in Section IV.

If a new partitioning is found, the registered models are removed from the database and the found partitions form new models which are sequentially added to the database. By adding the new models to the database, the system is triggered again. This enables the system to use the new and improved models to increase the chance to find even more models to merge.

Using the system presented in [2], the robot is able to detect and plan a path around objects moving around in the environment. Since the objects are unlikely to move over very small time scales, the entire data of the depth sensor can be used to register the views. Naturally this makes the problem simpler since a good initial guess can be provided from the robot localization. The grouping of the views can also be easily solved since the position of the object in the views is well known.

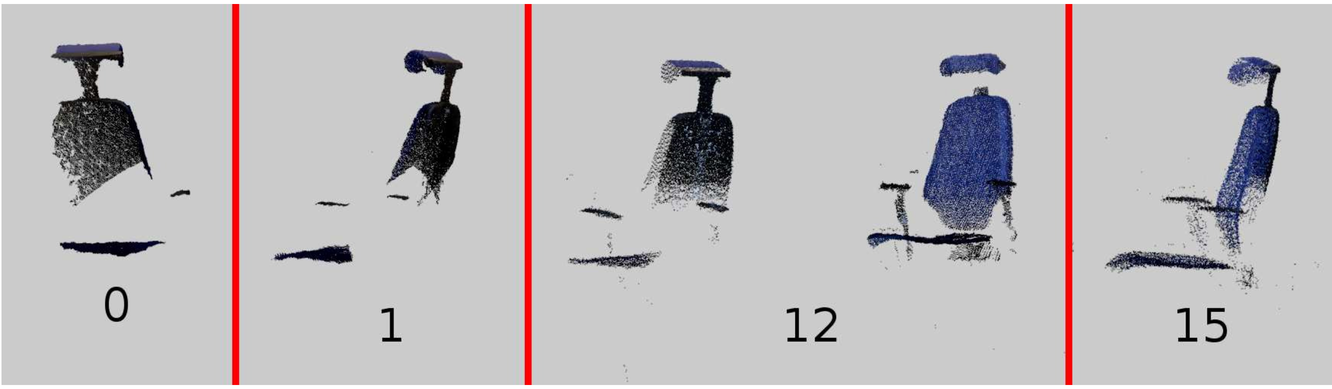


Fig. 1: The proposed system incrementally builds models of multiple observations by merging data into existing models when possible. The image above shows the contents of the model database with different number of observations added to the database for a test case of an office chair. At time 0 the algorithm has one model. At time 1 the system is able to fuse the new data into the existing model. At time 12 the system has received observations from two vastly different viewpoints(front and back) of the chair with very little overlap. The system is therefore unable to merge the new data into a previous model, resulting in a database containing two models. At time 15 the system receives an observation which allows the merging of both previous models.

IV. Finding coherent models

We propose a system that maximizes the total overlap amongst surfaces obtained from range images and minimizes occlusions between them. Given a pointcloud and a relative transform to a range image, each point in the pointcloud can quickly be reprojected into the image plane of the range sensor. A reprojected point can then be classified into one of three categories: overlapping the range image measurement, occluding the range image measurement or being occluded by the range image measurement.

A measurement is determined to be overlapping if the difference between the measurement value in the range image and the reprojected point is less then some threshold value. If a reprojected point is in between the sensor and the measured surface and not overlapping the surface, the point is considered to be occluding and vice versa if the point is behind the measured surface of the range image, the point is considered to be occluded.

The registration score of the pointcloud to a range image can then be computed as the sum of the number of overlapping points minus a constant times the number of occluding points. The occluded points are ignored as they cannot be observed in the range image. The value of the constant controls the trade-off between seeking a large overlap and minimizing the number of occluding points. We found that a value between 4 and 10 provided the best results.

The registration score between two aligned range images with segmentation masks can then be computed as the bidirectional sum of the registration scores computed from the segmentation masks and range images projected into the other range image.

Partitioning the views into groups that maximize overlap and minimize occlusions can then be reformulated as a graph partitioning problem where a range image, a relative pose and a segmentation mask form a node and the edge weights are set to be the registration score between the nodes. Once formulated as a graph partitioning problem, any standard graph partitioning technique can be applied. We found that recursively applying [13] worked well on our data.

If one already knows that two nodes are part of the same object and are accurately registered, one can set the edge weights between these nodes to infinity. This forces the graph partitioning to keep the two nodes in the same group.

V. Experiments

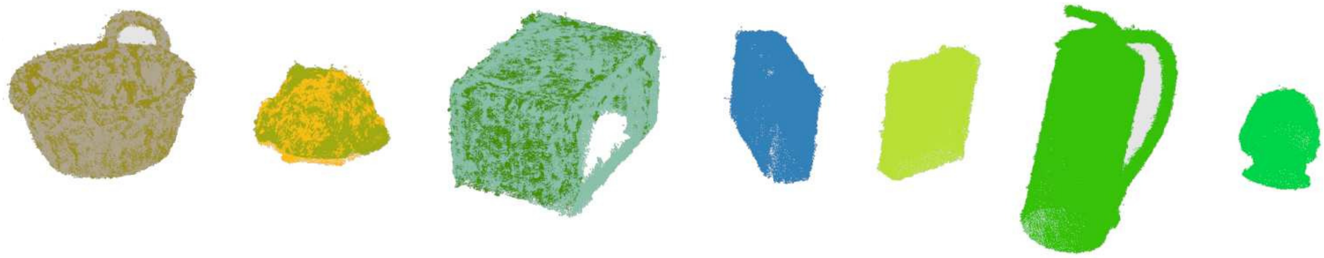
Using [2], we feed our system a set of observations where each observation contains approximately 10 views of an object. Fig. (2) shows a visual rendition of the database with different amounts of data. We see that no two observations are incorrectly fused. Because of the cylindrical shape, the fire extinguisher proved too hard for the registration algorithm to create an accurate model. The system correctly identifies the under-segmented fire extinguisher and avoids fusing it into the other views of the fire extinguisher. The system was unable to fuse all observations of the small box after 20 observations, after 30 observations the system manages to fuse all observations of the small box. The performance of the system is summarized in table I.

VI. Summary, conclusions and future work

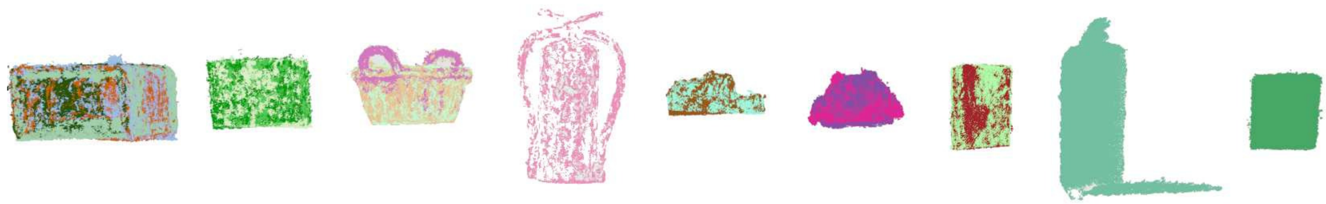
In this paper we proposed a robust and flexible system for autonomously building a database of object models on a mobile robot. We believe that this is an important step towards improving the creating a robot which learn and improve throughout its lifespan. The system can be used with any unconstrained registration algorithm, it would therefore be interesting to test which registration algorithm is most suitable for the framework. It would also be interesting to use a pointclouds searching framework similar to [14] to limit registration to only be performed on models with similar appearance in the database.

Rendering of database	Number of observations	Number of views	Number of models in database	Correct number of models	Correctly registered models	Incorrectly registered models
Fig.(2a)	10	98	7	7	7	0
Fig.(2b)	20	197	9	8	8	1
Fig.(2c)	30	311	9	8	8	1

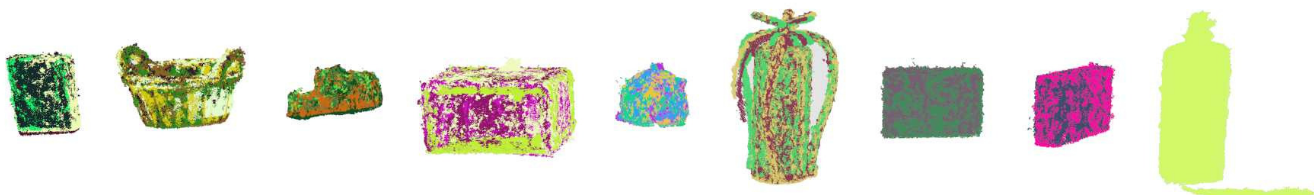
TABLE I: Performance of the object modelling system for different ammounts of data.



(a) Database after 10 observations, containing 98 different views.



(b) Database after 20 observations, containing 197 different views.



(c) Database after 30 observations, containing 311 different views.

Fig. 2: Database of created models. Each set of views captured using [2] is shown with an individual random color. The grouping of views at different observation times are all correct and appear to be well registered. The models are sorted from left to right based on the number of views used to create the model, where the leftmost model contains the most views.

ACKNOWLEDGMENT

The work presented in this papers has been funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No 600623 ("STRANDS").

REFERENCES

[1] R. Ambrus, N. Bore, J. Folkesson, and P. Jensfelt, "Meta-rooms: Building and maintaining long term spatial models in a dynamic world," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pp. 1854–1861, IEEE, 2014.

[2] T. Faeulhammer, R. Ambrus, C. Burbridge, M. Zillich, J. Folkesson, N. Hawes, P. Jensfelt, and M. Vincze, "Autonomous learning of object models on a mobile robot," Jan. 2016.

[3] F. Bernardini and H. E. Rushmeier, "The 3d model acquisition pipeline.," *Comput. Graph. Forum*, vol. 21, no. 2, pp. 149–172, 2002.

[4] M. Dimashova, I. Lysenkov, V. Rabaud, and V. Eruhimov, "Tabletop object scanning with an rgb-d sensor," SPME, 2013.

[5] J. Prankl, A. A. Buchaca, A. Svejda, and M. Vincze, "Rgb-d object modelling for object recognition and tracking," in *RGB-D Object Modelling for Object Recognition and Tracking*, p. 8, 2015.

[6] 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.

[7] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-time 3D model acquisition," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 21, pp. 438–446, July 2002.

[8] T. Weise, T. Wismer, B. Leibe, and L. J. V. Gool, "Online loop closure for real-time interactive 3d scanning.," *Computer Vision and Image Understanding*, vol. 115, no. 5, pp. 635–648, 2011.

[9] R. Ambrus, J. Ekekrantz, J. Folkesson, and P. Jensfelt, "Unsupervised learning of spatial-temporal models of objects in a long-term autonomy scenario," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015*, pp. 5678–5685, IEEE, 2015.

[10] E. Herbst, X. Ren, and D. Fox, "Rgb-d object discovery via multi-scene analysis," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 4850–4856, IEEE, 2011.

[11] E. Herbst, P. Henry, X. Ren, and D. Fox, "Toward object discovery and modeling via 3-d scene comparison.," in *ICRA*, pp. 2623–2629, IEEE, 2011.

[12] R. Finman, T. Whelan, M. Kaess, and J. J. Leonard, "Toward lifelong object segmentation from change detection in dense rgb-d maps," in *Mobile Robots (ECMR), 2013 European Conference on*, pp. 178–185, IEEE, 2013.

[13] M. Stoer and F. Wagner, "A simple min-cut algorithm," *J. ACM*, vol. 44, pp. 585–591, July 1997.

[14] N. Bore, P. Jensfelt, and J. Folkesson, "Retrieval of arbitrary 3d objects from robot observations," in *Mobile Robots (ECMR), 2015 European Conference on*, pp. 1–8, IEEE, 2015.