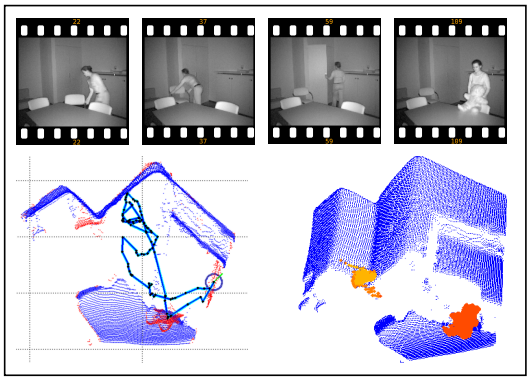
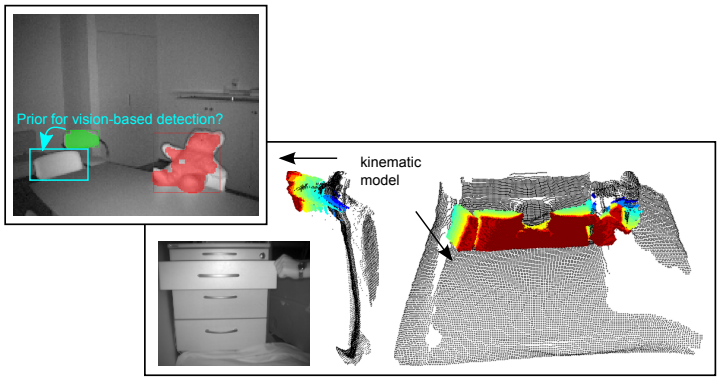
The Articulated Scene Model: Model-less Priors for Robot Object Learning?

Agnes Swadzba, Niklas Beuter, Sven Wachsmuth, and Franz Kummert
Applied Informatics, Bielefeld University
aswadzba@techfak.uni-bielefeld.de

Abstract

Human analysis of dynamic scenes consists of two parallel processing chains [2]. The first one concentrates on *motion* which is defined as variation of location while the second one processes *change* which is the variation of structure. The detection of a scene change is realized phenomenologically by comparing currently visible structures with a representation in memory. These psychological findings have motivated us to design an *articulated scene modeling* approach [1] which enables a robot to extract articulated scene parts through observing the spatial changes caused by their manipulation. This approach processes a sequence of 3D scans taken from a fixed view point which captures a dynamic scene where a human moves around and manipulates the environment by, e.g., replacing chairs or opening doors. It estimates per frame $\mathcal{F}_t$ the moving entities $\mathcal{E}_t$, the so far static scene background $\mathcal{S}_t$, and movable objects $\mathcal{O}_t$. The moving entities are tracked using a particle filter with a weak cylinder model. Static and movable scene parts are computed by a comparison of the current frame with the background model $\mathcal{S}_{t-t}$ estimated from the previous frames. For dense depth sensors, like the SwissRanger camera or the Kinect camera, such a comparison can be implemented as pixel-wise subtraction of $\mathcal{S}_{t-t}$ from $\mathcal{F}_t$. Using the fact that per pixel the farthest static depth measurements define the static background, arbitrary movable objects (like a replaced chair or an opened cupboard door) can be extracted model-less from depth measurements as they emerge in front of a known static background. The video [1] [2] shows for an Swissranger sequence the emerging of the static background (in blue), the movable objects (in orange), and the trajectories of an entity (in cyan and green) for two view points. The scene modeling part of our approach can also be presented on site in real-time on Kinect data. The development of cameras like the Kinect camera which combine dense depth measurement with a normal color camera in an elegant way, opens up new possibilities for interactive object learning. Future work could concentrate on the question whether the extracted movable objects (like chair) can be used to compute suitable features that can be used to detect, for example, other chairs in the scene which have not been moved, so far. Further, a history of positions of an articulated object like a drawer can be used to learn its kinematic model [3].





References

[1] N. Beuter, A. Swadzba, F. Kummert, and S. Wachsmuth. Using Articulated Scene Models for Dynamic 3D Scene Analysis in Vista Spaces. *3D Research*, 3(4), 2010. http://dx.doi.org/10.1007/3DRes.03(2010)04.

[2] R. A. Rensink. Change Detection. *Annual Review of Psychology*, 53:245–277, 2002.

[3] J. Sturm, V. Predeep, C. Stachniss, C. Plagemann, K. Konolige, and W. Burgard. Learning Kinematic Models for Articulated Objects. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1851–1856, Pasadena, CA, USA, 2009. AAAI Press.

[1] http://www.youtube.com/watch?v=VVc9bngjB2s
[2] http://aiweb.techfak.uni-bielefeld.de/content/6d-scene-analysis

2.5D Local Feature Matching System

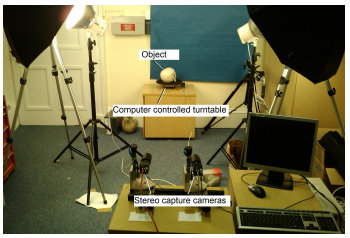
Euan Strachan, Dr J.Paul Siebert

March 2011

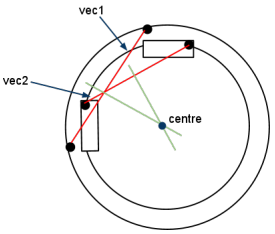
1 Introduction

A 3D visual tracking system was developed to investigate pose corrected surface sample patches for creating pose invariant SIFT features. The system comprises of a binocular stereo vision system and computer controlled turntable, Figure 1a. Rangemaps and images were captured of the object situated on the turntable, these were then aligned using the ground truth of the turntable motion. The system can capture both 3D surface structure and 2D surface texture simultaneously, without suffering from texture alignment issues affecting Lidar systems, such as seen in FRGC 2.0 [1].

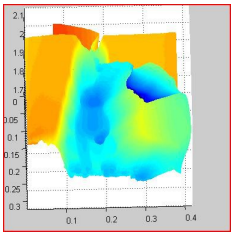
The system was used to evaluate projective corrected salient features for 3D invariant sample patches. This work was intended to extend affine invariant SIFT features [3] by using the partial 3D information from rangemaps to correct the viewing angle to the normal of the surface at the position of the interest point.



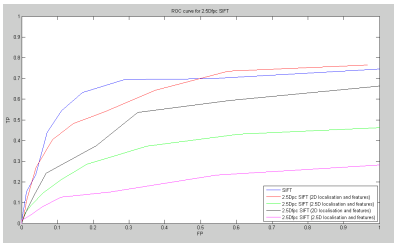
(a) Setup



(b) Turntable Center



(c) Aligned Rangemaps



(d) ROC curve

2 Data Capture Setup

The data capture system comprises of a stereo camera setup and calibrated turntable, rangemaps were built using the software C3D [4]. To calibrate the turntable, points on a calibration target placed on the turntable were tracked. Subsequent positions of the points were compared to create tangents to the rotation of the turntable, these were then used to find a least means square fit for the center of rotation of the turntable, Figure 1b.

The axis of rotation was found by taking the quaternion of the rotation matrix between 2 turntable rotations, and finding the axis of rotation from the quaternion. This gives the vector for the axis of rotation for the comparison of two rotations, to include all measurements the median of this vector from all observations was used.

3 2.5D Local features

Features were extracted from the 2D texture image using SIFT[2], SIFT with an affine pose corrected sample patch, and SIFT with a projective corrected sample patch. The affine corrected patch was formed by fitting a plane to the range surface of the local feature, and sampling evenly on the plane to form the 16x16 sample patch used to create the SIFT feature descriptor.

To create the projective corrected sample patch the rangemap was treated as a 3D point cloud which could be rotated so that the surface normal at a local feature could be set to zero. Hidden points were then removed and the surface was evenly resampled. Each local feature was extracted in this way, so that the view point was normalised for all SIFT keypoints.

4 Results and Conclusions

The local features were tested on their ability to match the same location on the object in successive rotations. The calibrated turntable was used as ground truth for the position changes in 3D. The results of this experiment are shown in Figure 1d. Keypoint localisation was also investigated, ie localising keypoints in texture and in range.

It was found that standard SIFT outperformed both the affine corrected SIFT and projective corrected SIFT. Affine corrected SIFT preformed better than projective corrected SIFT. Closer examination of the cases where the two proposed systems failed showed that there were 3 causes for this.

In some cases errors in the estimate of the surface normal caused the patch to warp to fit an incorrect surface, this caused no change in the original SIFT sample patch, small changes in affine corrected SIFT and large changes in projective corrected SIFT. For projective corrected SIFT the assumption that the rangemap can be treated as 3D and that rotating the points and resampling, should produce pose invariant features does not hold. This caused the resampled surface to be fit to invalid data, and the sample patch to differ further from between differing object views. Finally, for projective corrected SIFT when a keypoint is taken at a depth discontinuity, part of the sample patch will project to sample some point in the background, as this part of the sample patch varies the feature varies and loses its descriptability.

Future work will be based on learning the feature descriptor space for varying views

References

[1] C. Boehnen and P. Flynn. Impact of involuntary subject movement on 3d face scans. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 1 –6, 2009.

[2] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[3] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004.

[4] J.P Siebert X. Ju, T. Boyling. A high resolution stereo imaging system. In *Proceedings of 3D Mod03*, 2003.

A test bench to improve registration using RGB-D sensors

François Pomerleau and Stéphane Magnenat and Francis Colas and Ming Liu and Roland Siegwart*

Depth cameras (RGB-D) can provide dense 3D point clouds at a high frequency. Using the ICP algorithm, these point clouds can be matched to deduce the transformation between them and consequently the 6 degrees-of-freedom motion of the camera. This allows to build a tracker that can be used, for instance, as a front-end to a SLAM system.

However, as shown by the large amount of papers recently published on ICP, this algorithm has many variations, each of them depending on several parameters. To explore this large possibility space, we provide two contributions:

- A real-time tracker of the pose of a RGB-D sensor[1]. The latter takes as input a stream of point clouds and outputs an estimation of the 3D pose of the sensor. To avoid drift due to the high frame rate of the input (30 Hz), this tracker holds a single reference and matches every incoming point cloud against it. If the ratio of matching points drops below a pre-defined threshold, the tracker creates a new reference with the current cloud.

  ICP consists of performing several sequential steps, both inside and outside a main iteration loop. For each step, there exist several strategies, and each strategy demands specific parameters. Current works provide no easy way to compare these strategies. To enable such a comparison, the tracker employs a modular ICP chain, which is fully configurable through text-based parameters[2].

- A dataset of 27 runs containing point clouds produced by a Kinect along ground truth from a Vicon[3]. These runs cover 3 environments of increasing complexity. For each complexity, an operator performs 3 types of motion: translations on the three axis, rotations on the three axis and a free fly motion over the scene. For each environment, the operator performs each type of motion at 3 different speeds.

We expect these contributions to enable the community to improve the understanding of what are the critical parameters to produce a precise and fast tracking using RGB-D sensors[4].

*Autonomous Systems Lab. – ETH Zurich, Tannenstr 3, 8092 Zürich, Switzerland `firstname.lastname@mavt.ethz.ch`

[1]The tracker is available from `http://www.ros.org/wiki/modular_cloud_matcher`

[2]The ICP-chain library is available at `http://github.com/ethz-asl/libpointmatcher`

[3]This dataset is available at `http://www.asl.ethz.ch/research/datasets`

[4]A video of our work is available at `http://www.youtube.com/watch?v=McxpJGOZTPs`

RGB-D object recognition and localization with clutter and occlusions

Federico Tombari, Samuele Salti, Luigi Di Stefano
Computer Vision Lab
DEIS, University of Bologna
Bologna, Italy

This work demonstrates object recognition robust to clutter and occlusions in 3D data represented by range maps enriched with color information, also allowing for recognition of multiple instances of the model to be found. Our approach compares each scene to a library of models, each model being represented by a set of different views of the model itself. For each model and scene, the stages sketched in the diagram of Figure 1 are applied, which will be now briefly described.

Given a model and a scene, first features are extracted and described on each model view (offline) and on the scene (at run-time) by means of, respectively, random sampling of the range map and Color-SHOT (CSHOT) [1], a recently proposed 3D descriptor, which represents an extension to RGB-D data of the SHOT descriptor [2]. These proposals will be now briefly outlined.

The SHOT descriptor [2] relies on the definition of a repeatable local Reference Frame (RF) based on the Eigenvalue Decomposition of the scatter matrix of the neighborhood of a point. Given the local RF, an isotropic spherical grid is defined to encode spatially well localized information, i.e. to define a signature structure. For each sector of the grid, the angles among the normal of the central point and that of each point falling in that sector are accumulated in a histogram. The final descriptor results from the juxtaposition of these histograms. Motivated by the increasing availability of 3D sensors capable of delivering both shape and color information, in [1] an extension to the SHOT descriptor is proposed. This novel descriptor, dubbed CSHOT and aimed at feature matching in 3D data enriched with color, adds to geometrical information (angles between normals of the features, as in SHOT) also color information (color differences computed in the CIELab space among points).

Following the description stage, each model view is efficiently matched against the scene by means of *kd-trees*, determining scene-to-model correspondences. By applying the Hough-based 3D Object Recognition scheme proposed in [3] a subset of geometrically-coherent correspondences is selected on each view. As for this approach, each model feature point is associated with its relative position with respect to the centroid of the model, so that each corresponding scene feature can cast a vote in a 3D Hough space to accumulate evidence for possible centroid position(s) in the current scene. This enables simultaneous voting of all feature correspondences within a single 3-dimensional Hough space. To correctly cast votes according to the actual pose(s) of the object(s) being sought, we rely on the local RFs associated with each pair of corresponding features.

The view with the highest number of selected correspondences is chosen as the best one: if this number is higher than a threshold, the object is detected on the scene, and its pose determined by applying a final RANSAC and Absolute Orientation stage [4] on the remaining correspondences, so as to yield a Rotation matrix and Translation vector that aligns the best model view to the scene.

With our current implementation, processing RGB-D data acquired at run-time with a Microsoft *Kinect* sensor, around 10 seconds are required to go through all stages of the algorithm, from scene acquisition to object recognition. A demo video can be found online [1].

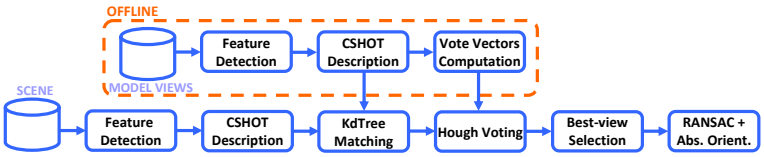


Figure 1: Stages of the proposed algorithm.

References

[1] F. Tombari, S. Salti, and L. Di Stefano, "A combined intensity-shape descriptor for texture-enhanced 3d feature matching," in *ICIP, submitted*, 2011.

[2] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. ECCV*, 2010.

[3] F. Tombari and L. Di Stefano, "Object recognition in 3d scenes with occlusions and clutter by hough voting," in *Proc. PSIVT*, 2010.

[4] B. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Optical Society of America A*, vol. 4, no. 4, pp. 629–642, 1987.

[1] `www.vision.deis.unibo.it/demos/rgbdDemo.avi`

Outdoor Terrain Traversability Analysis for Robot Navigation using a Time-Of-Flight Camera

G. De Cubber* D. Doroftei* H. Sahli** Y. Baudoin*

Autonomous robotic systems operating in unstructured outdoor environments need to estimate the traversability of the terrain in order to navigate safely. Traversability estimation is a challenging problem, as the traversability is a complex function of both the terrain characteristics, such as slopes, vegetation, rocks, etc and the robot mobility characteristics, i.e. locomotion method, wheels, etc. It is thus required to analyze in real-time the 3D characteristics of the terrain and pair this data to the robot capabilities.

Stereo cameras or 3D laser range finders are generally used as input devices for traversability analysis and two main approaches can be distinguished. There are the methods as the ones advocated by Labayrade (1) and Mufti (2) who assume a (piecewise) planar ground plane. They estimate the ground plane, set a threshold and consider objects with distances to the ground plane further than this threshold as obstacles. Other methods, as the ones proposed by Birk (3) and Helmick (4) search for specific types of objects (rocks, canyons) and classify the image based on this data.

To our knowledge, time-of-flight cameras have until now not been used for these kind of applications, simply because there were no sensors capable of coping with outdoor conditions, especially due to the interference of solar irradiation. This situation is changing now, with the advent of outdoor-capable sensors. Therefore, we present in this paper an approach for outdoor terrain traversability which mixes 2D and 3D information for terrain classification.

The methodology towards time-of-flight-based terrain traversability analysis extends our previous work on stereo-based terrain classification approaches (5). Following this strategy, the *RGB* data stream is segmented to group pixels belonging to the same physical objects. From the *Depth* data stream, the $v-disparity$ (1) is calculated to estimate the ground plane, which leads to a first estimation of the terrain traversability. From this estimation, a number of pixels are selected which have a high probability of belonging to the ground plane (low distance to the estimated ground plane). The mean a and b color values in the *Lab* color space of these pixels are recorded as $\mathbf{c}$. The result of both data streams is then combined to optimize the classification result. For each pixel i in the image, the color difference $\|\mathbf{c}_i - \mathbf{c}\|$ and the obstacle density in the region where the pixel belongs to are calculated. The obstacle density δ_i is here defined as: $\delta_i = \frac{\langle o \in A_i \rangle}{\langle A_i \rangle}$, where o denotes the pixels marked as obstacles (high distance to the estimated ground plane) and A_i denotes the segment where pixel i belongs to. This allows us to define a traversability score as $\tau_i = \delta_i \|\mathbf{c}_i - \mathbf{c}\|$, which is used for classification. This is done by setting up a dynamic threshold, as a function of the distance measured. Indeed, as the error on the depth measurement increases with the distance, it is required to increase the tolerance on the terrain classification as a function of the distance. An important issue when dealing with data from a time-of-flight sensor is the correct assessment of erroneous input data and noise. Therefore, the algorithms combines information from the 2D intensity image (indicating the intensity of the measured irradiation) with 3D distance data. Regions with low intensities and a large variance in distance measurements are therefore automatically detected and marked as "suspicious".

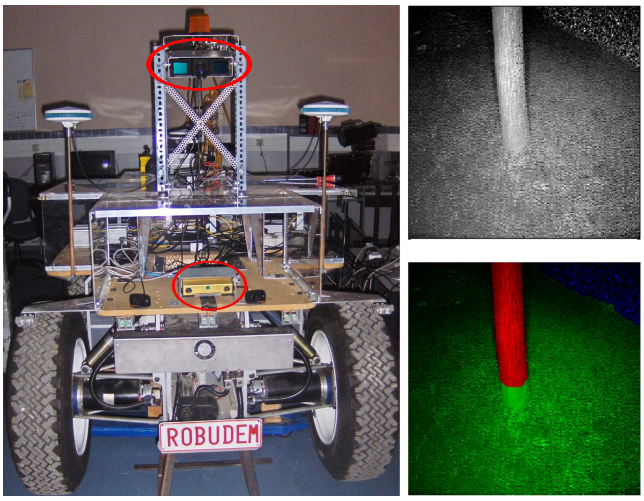


Figure 1: Left: System setup - Outdoor robot with a Time-Of-Flight Camera (on top) and a Stereo Camera (in the middle); Top Right: Amplitude Image; Bottom Right: Traversability Estimation (green: traversable; red: obstacle; blue: suspicious / not enough data)

Figure 1 shows the Robudem platform which was used as a testbed for the presented algorithms. It is a heavy outdoor robot equipped with a PMDTec CamCube time-of-flight sensor (on top) and a Point Grey Bumblebee stereo camera (in the middle). The time-of-flight camera is mounted in a tilted angle to avoid typical signal modulation problems. The top right image shows the amplitude input image, whereas the bottom left image shows the terrain classification result. Obstacles are red, well traversable terrain is green and "suspicious" areas (not enough data) are blue. It can be noticed that the classification is correct, as the obstacle (the tree) is well-detected. In the upper left corner, there are some problems with foliage giving erroneous reflections (blue area), which is due to the sensor.

A video demonstration of the presented algorithm is available on http://www.youtube.com/watch?v=CNFc5qPvnB0

References

[1] R. Labayrade and D. Aubert, "In - vehicle obstacles detection and characterization by stereovision," in *Int. Workshop on In-Vehicle Cognitive Comp. Vision Systems*, 2003.

[2] F. Mufti and Al., "Spatio-temporal ransac for robust estimation of ground plane in video range images for automotive applications," in *Int. conf. on Intelligent Transportation Systems*, 2008.

[3] A. Birk and Al., "Terrain classification for autonomous robot mobility: from safety, security rescue robotics to planetary exploration," in *ICRA08*, 2008.

[4] D. Helmick and Al., "Terrain adaptive navigation for planetary rovers," *J. of Field Robotics*, vol. 26, no. 4, pp. 391–410, 2010.

[5] G. DeCubber, "Multimodal terrain analysis for an all-terrain crisis management robot," in *Humanitarian Demining*, 2011.

*Unmanned Vehicle Centre of the Belgian Royal Military Academy. email: geert.de.cubber@rma.ac.be

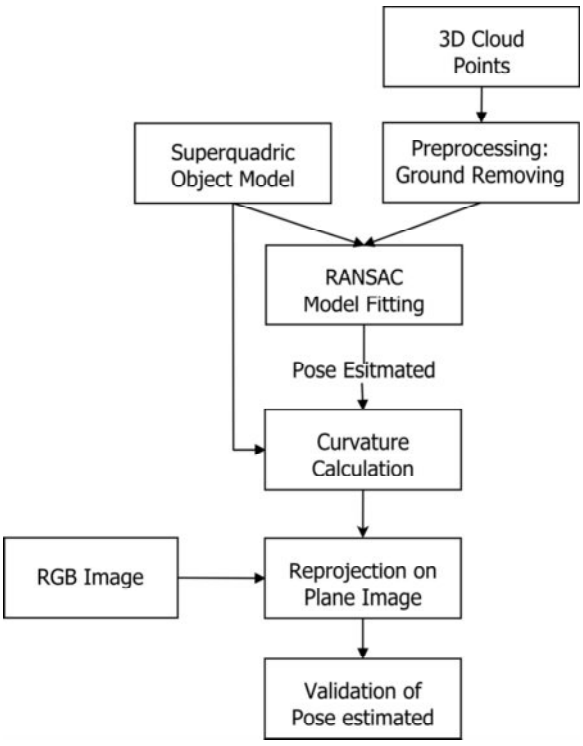
**Department of Electronics and Informatics, Vrije Universiteit Brussel, Belgium. email: hichem.sahli@etro.vub.ac.be

3d object localization using superquadric models with a Kinect sensor.

Nicolo' Biasi, Ilya Afanasyev, Alberto Fornaser, Luca Baglivo, and Mariolino De Cecco
(University of Trento, Italy)

In this abstract is presented a new method for 3D object recognition and pose estimation by using RGB-D sensor, like Microsoft Kinect. The pose is estimated by a robust least square fitting of the 3D points with a SuperQuadric (SQ) model of the searched object. The solution is verified by evaluating the matching score between the projected edges of the object model and the real edges extracted from RGB image. This method can concurrently be used for the refining of the camera and 3D sensor extrinsic parameters.

Details: Object pose estimation starts with a preprocessing phase of cloud of points captured by 3D sensor. In this preprocessing stage, the points relative to the ground are identified with a RANSAC plane-fitting technique and then removed from data set.
The next step consists in fitting a SuperQuadric model of the object of interest with 3D data. SQ models permit to describe complex-geometry object with few parameters and generate simple minimization function to estimate object pose.
SQ modelling allow also to represent geometrical characteristic of the object, as surface normal and curvature, in closed form. Such information are further exploited to identify position of model's edges. To face measurement noise and outliers, object pose estimation problem is approached with RANSAC-based technique.
Pose verification phase consists in reprojecting the SQ-model of the object on RGB image and comparing the position of SQ edges with the ones identified in color image. The comparison exploits a standard pattern matching approach as distance transform techniques.
Using reference objects, the projection of the object identified with the 3D sensor onto the image plane can give a refined estimation of extrinsic parameters between the two sensors while concurrently estimate their extrinsic parameters.



Results: The algorithm has been developed in MATLAB. The RGB-D information were obtained with Microsoft Kinect device and then processed offline. The pose estimation technique described has been tested with simple geometry objects (cube, cylinder, …) giving encouraging results.

Acting and Interacting in the Real World

Jeannette Bohg, Niklas Bergström, Mårten Björkman and Danica Kragic

A characteristic of robots that distinguishes them from other intelligent agents like the chess playing *Deep Blue* [1] or any 2D image classifier [2] is its physical embodiment in the real world. A robot can act in or interact with the environment and thereby change it. To do this in any meaningful way, it needs to be able to perceive and represent the 3D structure of its surrounding. This facilitates processes like motion planning of actuators or grasping and manipulation.

Recently, through the release of devices like the *kinect* [3], sensing of dense and high quality 3D data became cheap, fast and easy. Certain constraints introduced by other 3D sensing devices are removed. However, the basic challenges of *processing* and *understanding* 3D data remain. These are (i) the high dimensionality of the data, (ii) noise, (iii) occlusions and most importantly (iv) how to get from the low-level set of 3D points to high-level semantic information.

In the following, we will outline how we deal with these challenges given the specific task of grasping and manipulation in a table top scenario.

I. Segmentation

For picking up objects from a table, segmenting them from the table plane and each other facilitates grasp planning but also helps processes like object recognition, categorization and pose estimation that attach high-level semantic information to the low-level data. In our previous work [4], we posed segmentation as an optimization problem that maintains three hypotheses: figure, ground and a supporting plane. The proposed approach is an iterative two-stage method that first performs pixel-wise labeling using a set of model parameters and then optimise's these parameters in the second stage until convergence. The model parameters are describing distributions in hue, saturation and disparity space. The approach is robust to noise and occlusions. Recently, this approach got extended to maintain not just one but several foreground hypotheses. Furthermore, we showed that this approach works well for 3D data of very different quality coming from different sensing devices (see Fig 1).

Compared to the popular plane removal approach [5], there are several advantages. First, the segmentation is not purely geometrical but uses multimodal information for optimizing the pixel labeling. Therefore even object parts close to the table will be detected as foreground. Second, the 2D ordering of the 3D data in the pixel plane is used to efficiently represent the data and solve for the neighborhood relationships. However, it needs some top-down information for initializing the foreground hypotheses. We show that this can either come from attention in which salient scene point serve as initial foreground points or from user input.

This work was supported by the EU through the project GRASP, IST-FP7-IP-215821, the Swedish Foundation for Strategic Research and the Swedish Research Council. The authors are with the Centre for Autonomous Systems and Computational Vision and Active Perception Lab, School of Computer Science, KTH in Stockholm, Sweden. bohg,nbergst,celle,danik@csc.kth.se

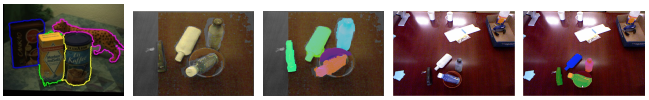


Fig. 1. Segmentation results given input data from different sensing devices. From Left to Right: (i) KTH vision system [4]. (ii) PR2 narrow stereo-image overlayed with co lour information transferred from wide-field. [6] (iii) PR2 Segmentation Result. (iv) Kinect input image mounted on the PR2. [3] (v) Kinect Segmentation Result.

II. Prediction of Occluded Object Shape

Given segmented object hypotheses from the segmentation described above, the robot now might want to know whether it has seen this or similar objects before and what their pose is. Due to occlusions, even dense 3D data will only provide a partial object reconstruction. Estimating the occluded and unknown part of an object can support 3D object or classification approaches but has also advantages for collision detection and grasp planning. Psychological studies suggest that humans are able to predict the portions of a scene that are not visible to them through *controlled scene continuation* [7]. The expected structure of unobserved object parts are governed i) visual evidence and ii) completion rules gained through prior visual experience. A very strong prior that exists in especially man-made objects is symmetry.

In our previous work [8], we show how we can use this prior to successfully predict whole object shape and evaluate their plausibility based on visibility constraints. Furthermore, we reconstruct a mesh based on which grasps can be planned in simulation using traditional grasp quality measures. However in our previous work, we have not taken the plausibility rank of each point explicitly into account. Here, we will show how this can be useful in a probabilistic grasp planning framework like the one presented in [9].

References

[1] IBM, "Deepblue," http://www.research.ibm.com/deepblue/.
[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, Jun. 2010.
[3] Microsoft, "Kinect," http://www.xbox.com/en-US/kinect.
[4] M. Björkman and D. Kragic, "Active 3D scene segmentation and detection of unknown objects," in *ICRA*, 2010.
[5] R. Rusu, A. Holzbach, R. Diankov, G. Bradski, and M. Beetz, "Perception for mobile manipulation and grasping using active stereo," in *Humanoids*, Paris, 2009.
[6] W. Garage, "Pr2 robot," http://www.willowgarage.com.
[7] T. P. Breckon and R. B. Fisher, "Amodal volume completion: 3d visual completion," *CVIU*, 2005.
[8] J. Bohg, M. Johnson-Roberson, B. León, J. Felip, X. Gratal, N. Bergström, D. Kragic, and A. Morales, "Mind the gap - robotic grasping under incomplete observation," in *ICRA*, May 2011, to appear.
[9] P. Brook, M. Ciocarlie, and K. Hsiao, "Collaborative grasp planning with multiple object representations," in *ICRA*, 2011, to appear.

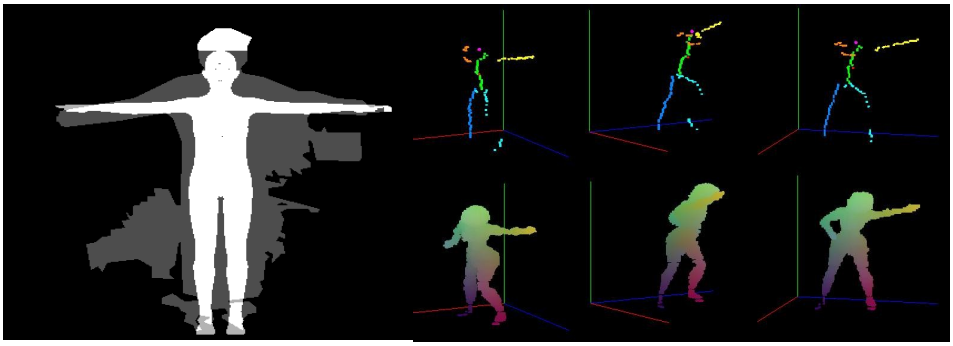
RGB-D Presentation @ EURON – Koen Buys
koen.buys@mech.kuleuven.be

Interaction between pointclouds and camera images for human motion capture segmentation.

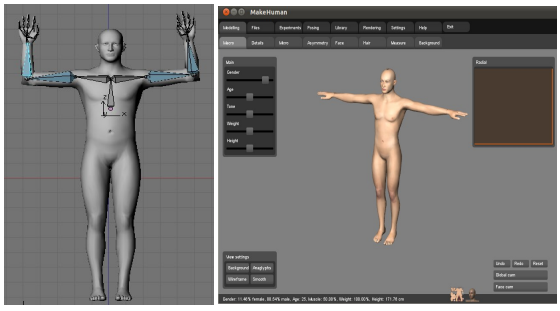
Abstract

This presentation will present two main research fields in which I currently use combination of RGB and Depth information. The goal of the presentation is to discuss approaches and ideas of possible collaborations and feature descriptors.

The first part handles about marker-less human motion capture where RGBD data is not only used to influence segmentation in 2D but also do PF model validation. The goal is the estimate body poses of people lying in bed sleeping.
First an automatic avatar creation algorithm was implemented which uses combined 3D and 2D segmentation.



A segmentation on the depth image helps to avoid shadows, removes background and ground influences, this is a combination of a pass through filter and a plane estimation and subtraction. The remaining pointcloud is an aid in verifying correct postures. This pointcloud is backprojected into the camera image and serves as an input to the GrabCut algorithm which segments out the human.
This segmented image serves to estimate the body poses and to adapt a MakeHuman avatar to the subject.



For calibration purposes 5 known poses were asked to the subjects. Once a customized avatar is created, the sleeping process of the human will be monitored. This is done by an active mattress which adapts to the human body pose and

generates a 170 point pointcloud of the current pose. In an OpenGL environment a PF simulates the physics of a human sleeping and generates a corresponding pointcloud. These pointclouds are compared to do particle evaluation.

In the second part I will discuss a number of combinations of pointcloud and RGB data segmentation that were discussed for the object and person detection and recognition in the Robocup@home contest.

2D/3D Object Categorization for Task Based Grasping

Marianna Madry, Dan Song and Danica Kragic

We present an object categorization system integrated with a grasp planning and reasoning system (see Figure 1). The main motivation for the work is to equip robots with the ability of transferring grasping knowledge between objects that belong to the same category. The categories are defined based on their geometric properties and functionality, relating to the idea of affordances.

In the heart of the system, there is the *Object Categorization System* (OCS) using camera images as input. The system employs both 2D (RGB image) and 3D (point cloud representing the visible part of an object) information about an object. In designing the 2D/3D Object Categorization System a two-fold approach was adapted. First, we build the classic single cue OCS for each of the descriptors capturing the following object properties: (a) color (opponentSIFT), (b) 2D shape (HoG) and (c) 3D shape (FPFH), and then these systems are integrated to provide the final decision based on all cues. We present and test several 2D/3D integration strategies. The system is evaluated on real data collected using an active stereo head, capable of vergence and foveation. The data is generated in natural scenes, for a number of household object categories. The results showed that the proposed system achieved high object recognition rate (up to 91%), significantly better than the classic single cue OCS in the same task. The system is built upon an active scene segmentation module, able of generating object hypotheses thus segmenting them from the background in real-time [1]. We integrated the OCS with a task-constraint model for robot grasping [2], [3]. The results showed that the object categorization is very useful for reasoning and planning of goal-directed grasps in natural scenes with multiple objects.

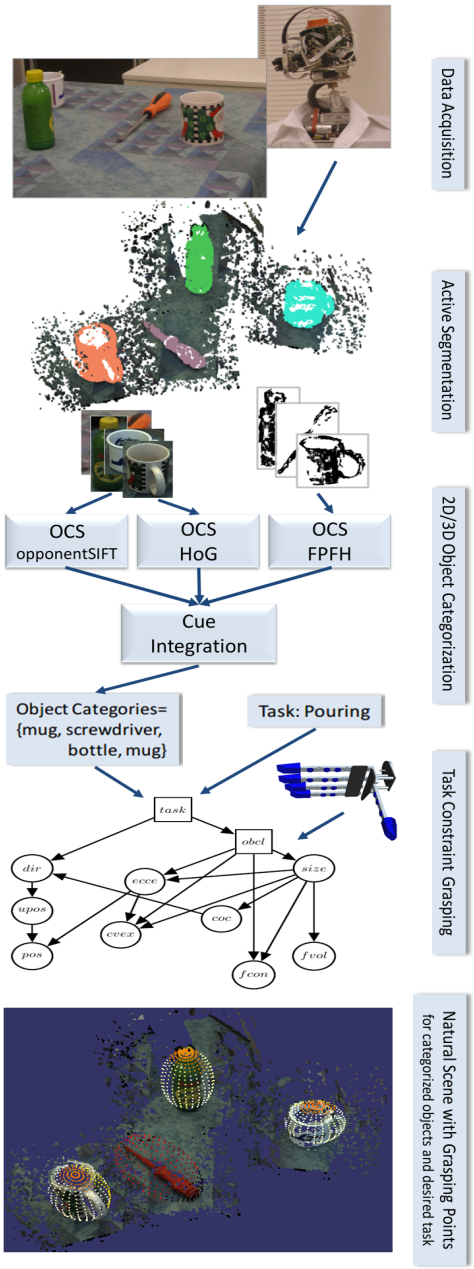


Fig. 1. System outline. First row: Data acquisition using ARMAR III robot head and view of a typical experimental scene. Second row: Segmented objects in the same scene. Third row: Integrated 2D and 3D Object Categorization Systems (OCSs). Fourth row: Generation of grasping points by Bayesian network. Fifth row: Experimental scene with grasping points for categorized objects and desired task (the lighter is the point the higher is the probability). For the accurate color information we kindly direct to the electronic version of the abstract.

REFERENCES

[1] M. Bjorkman and D. Kragic, "Active 3D scene segmentation and detection of unknown objects," in *ICRA*, 2010.

[2] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning Task Constraints for Robot Grasping using Graphical Models," in *IROS*, 2010.

[3] D. Song, C.-H. Ek, K. Huebner, and D. Kragic, "Multivariate discretization for bayesian network structure learning in robot grasping," in *ICRA*, 2011, to appear.

The authors are with KTH – Royal Institute of Technology, Stockholm, Sweden, as members of the Computer Vision & Active Perception Lab., Centre for Autonomous Systems, e-mail addresses: madry,dsong,danik@csc.kth.se. This work was supported by EU through the project GRASP, IST-FP7-IP-215821 and Swedish Foundation for Strategic Research.

Real-time 3D visual SLAM with a hand-held RGB-D camera

Nikolas Engelhard[a] Felix Endres[a] Jürgen Hess[a] Jürgen Sturm[b] Wolfram Burgard[a]

The practical applications of 3D model acquisition are manifold. In this paper, we present our RGB-D SLAM system, i.e., an approach to generate colored 3D models of objects and indoor scenes using the hand-held Microsoft Kinect sensor. Our approach consists of four processing steps as illustrated in Figure 1. First, we extract SURF features from the incoming color images. Then we match these features against features from the previous images. By evaluating the depth images at the locations of these feature points, we obtain a set of point-wise 3D correspondences between any two frames. Based on these correspondences, we estimate the relative transformation between the frames using RANSAC. The third step is to improve this initial estimate using a variant of the ICP algorithm [1]. As the pair-wise pose estimates between frames are not necessarily globally consistent, we optimize the resulting pose graph in the fourth step using a pose graph solver [4]. The output of our algorithm is a globally consistent 3D model of the perceived environment, represented as a colored point cloud. The full source code of our system is available as open source [2]. With an earlier version of our system, we participated in the ROS 3D challenge organized by Willow Garage and won the first prize in the category "most useful".

Our approach is similar to the recent work of Henry et. al [5]. Our approach applies SURF instead of SIFT features. Additionally, our source code is available online.

Figures 2 and 3 illustrate the quality of the resulting 3D models. For both experiments, we slowly moved the Kinect around the object and acquired around 12 RGB-D frames. Computing the model took approximately 2 seconds per frame on an Intel i7 with 2 GHz. We applied our approach also to a large variety of other objects. Videos with more results are available online [3]. We plan to give a live demonstration with an online version of our system during the RGB-D workshop at the Eurobotics Forum 2011.

Our approach enables a robot to generate 3D models of the objects in the scene. But also applications outside of robotics are possible. For example, our system could be used by interior designers to generate models of flats and to digitally refurbish them and show them to potential customers. At the moment, we do not deal with the problem of automatic view point selection but assume instead that the user is moving the camera through the scene.

[a] N. Engelhard, F. Endres, J. Hess, and W. Burgard are with the Autonomous Intelligent Systems Lab, Computer Science Department, University of Freiburg, Germany. {engelhar,endres,hess,burgard}@informatik.uni-freiburg.de

[b] J. Sturm is with the Computer Vision and Pattern Recognition Group, Computer Science Department, Technical University of Munich, Germany. sturmju@in.tum.de

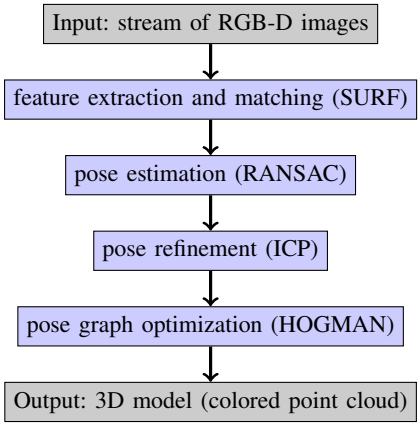
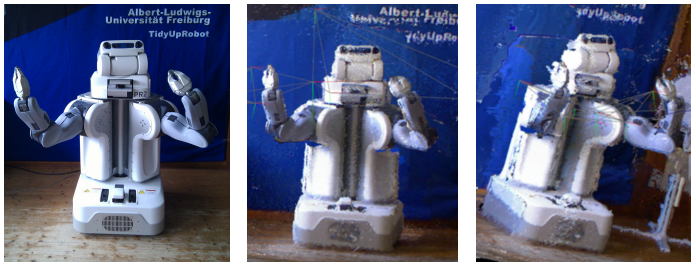


Fig. 1: The four processing steps of our approach. Our approach generates colored 3D environment models from the images acquired with a hand-held Kinect sensor.



(a) (b) (c)

Fig. 2: (a) Image of the PR2 robot in our lab. (b) and (c) Resulting model, visualized from two different perspectives. As can be seen from these images, the individual point clouds have accurately been integrated into the map.

(a) (b) (c)

Fig. 3: (a) Image of a teddy bear. (b) and (c) Resulting model, visualized from two different perspectives.

REFERENCES

[1] A. Segal D. Haehnel, S. Thrun. Generalized ICP. In *Proc. of Robotics: Science and Systems (RSS)*, 2009.

[2] F. Endres, J. Hess, N. Engelhard, J. Sturm, and W. Burgard. http://www.ros.org/wiki/openni/Contests/ROS 3D/RGBD-6D-SLAM, Jan. 2011.

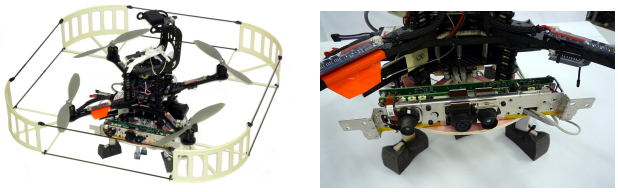
[3] F. Endres, J. Hess, N. Engelhard, J. Sturm, and W. Burgard. http://www.youtube.com/watch?v=XejNctt2Fcs, ?v=5qrBEPfEPaY, and ?v=NR-ycTNcQu0, 2011.

[4] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2D and 3D mapping. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Anchorage, AK, USA, 2010.

[5] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Proc. of the Intl. Symp. on Experimental Robotics (ISER)*, Delhi, India, 2010.

Autonomous Corridor Flight of a UAV Using an RGB-D Camera

Sven Lange Niko Sünderhauf Peer Neubert Sebastian Drews Peter Protzel



(a) (b)

Fig. 1. (a) Our modified *Pelican* quadrotor system. (b) The dismantled Kinect sensor mounted on the UAV.

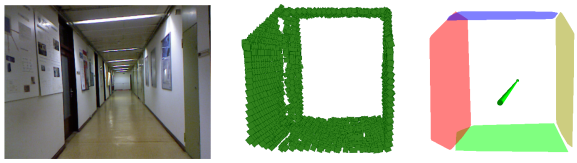


Fig. 2. *(left)* RGB image of the corridor. *(mid)* Downsampled point cloud containing about 3,000 points. *(right)* Extracted and classified planes of the walls. The green arrow shows the intended motion direction computed from the position and orientation relative to the walls.

I. Introduction

We describe the first application of the novel Kinect RGB-D sensor on a fully autonomous quadrotor UAV. We apply the UAV in an indoor corridor scenario. The position and orientation of the UAV inside the corridor is extracted from the RGB-D data. Subsequent controllers for altitude, position, velocity, and heading enable the UAV to autonomously operate in this indoor environment.

II. System Overview

The UAV we use in our project is a "Pelican" system (see Fig. 1(a)) that is manufactured by Ascending Technologies. We extended the UAV's configuration and equipped the quadrocopter with additional hardware: An SRF10 sonar sensor measures the altitude, an ADNS-3080 optical flow sensor board provides information on the current velocity. Altitude, velocity and position are controlled using cascaded PID controllers that are implemented on an ATmega644P. The Kinect RGB-D device (Fig. 1(b)) is connected to the onboard embedded PC system and interfaced using ROS.

III. Autonomous Corridor Flight Using the Kinect RGB-D Device

The Kinect driver of ROS provides a 640×480 3D point cloud that is downsampled (thinned) to approximately 3,000 points for further processing. For performance reasons, we implemented a specialized downsampling algorithm that runs five times faster compared to the function provided by the Point Cloud Library.

After downsampling, large planar sections are found in the remaining points by applying a sample consensus (MLESAC) based parameter estimation algorithm. Fig. 2 visualizes the results. The Point Cloud Library already provides convenient algorithms that extract the planes in their parameter form $ax + by + cz + d = 0$.

The authors are with the Department of Electrical Engineering and Information Technology, Chemnitz University of Technology, 09111 Chemnitz, Germany. firstname.lastname@etit.tu-chemnitz.de

Given these parameters, the extracted planes are assigned to one of the following wall classes: *floor, ceiling, left, right, front*. Distances from the UAV to the walls Δ_i and their orientations ϕ_i are calculated as well.

To keep the UAV aligned with the corridor and in the center of the corridor, motion commands $(dx, dy, d\phi)$ are calculated from the plane distances Δ_i and the yaw estimates ϕ_i. Weight factors w_i are used to ensure that those walls that were supported by more scan points during the plane extraction have a stronger influence on the resulting motion command.

IV. Results

A video that shows an example flight is available at our website www.tu-chemnitz.de/etit/proaut/forschung/quadrocopter.html.en. Fig. 3 shows the position estimates Δ_{left} and Δ_{right} inside the corridor while performing autonomous flight. According to these internal measurements, the maximum deviation from the corridor center was 35 cm. Mean velocity was $0.36\frac{m}{s}$. The sourcecode for efficient point cloud downsampling and trajectory generation is available to the community as part of our ROS repository at http://www.ros.org/wiki/tuc-ros-pkg.

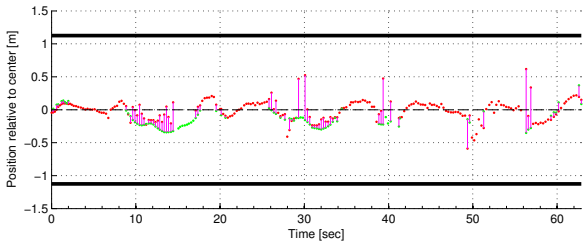


Fig. 3. Position estimates of the quadrotor within the corridor during autonomous flight, based on the Kinect measurements. The red points represent the calculated distances based on the left wall and the green points are based on the right wall. If two walls are visible, measurements are connected through a line. The corridor walls are shown as thick black horizontal lines. Maximum deviation from the corridor center was 35 cm.

FAB-MAP 3D: Topological Mapping with Spatial and Visual Appearance

Rohan Paul and Paul Newman
Oxford University Mobile Robotics Research Group. {rohanp,pnewman}@robots.ox.ac.uk

***Abstract*—We present a probabilistic framework for appearance based navigation and mapping using spatial and visual appearance data. We adopt a bag-of-words approach in which positive or negative observations of visual words in a scene are used to discriminate between already visited and new places. Additionally, we explicitly model the spatial distribution of visual words as a 3D random graph in which nodes are visual words and edges are distributions over distances. The spatial model captures the multi-modal distributions of inter-word spacing and incorporates a probabilistic sensor model for word detection and distances. Crucially, the inter-word distances in 3D are viewpoint invariant and collectively constitute strong place signatures for appearance based navigation. Results illustrate a tremendous increase in precision-recall area compared to a state-of-the-art visual appearance only systems.**

The goal of this work is non-metric topological navigation and mapping in appearance space - a by-product of which is loop closure detection. We provide and test a formulation which uses not only the visual appearance of scenes but also aspects of its geometry. Our approach, called FAB-MAP 3D, has its roots in the FAB-MAP algorithm which in essence learns a probabilistic model of scene appearance online using a generative model of visual word observations and a sensor model which explains missed observations of visual words. FAB-MAP 3D takes the same approach but incorporates the observation of spatial ranges between words coupled to the observation of pairs of visual words, Figure 1. This interaction is captured via a random graph which models a distribution over word occurrences as well as their pairwise distances. Using non-parametric Kernel Density Estimation we learn complex multi-modal distributions over inter-word distances and also accelerate inference by executing a Delaunay tessellation of the perceived 3D graph. The system shows improved performance over vision only sensing in an outdoor setting.

Our motivation for incorporating range information is two fold. Firstly, prior to this the work, the FAB-MAP framework only modeled the presence or absence of a word at a location and did not incorporate the spatial arrangement of visual words. Secondly, FAB-MAP currently discards the number of times a word appears in a scene - there is information being neglected here. This is addressed in FAB-MAP 3D because by using the range between occurrences of visual words we are implicitly counting word occurrence. Note also that we are in the business of robotics where range information is ubiquitous be it from lidar, stereo or structure from motion - we should use it if we can. Finally, there is also an important prima facia advantage of using distances because they are invariant under rigid transformation and that is precisely what we require of a place descriptor in topological navigation. The system only needs intra-scene distances which can be derived in a local frame without requiring a global metric map.

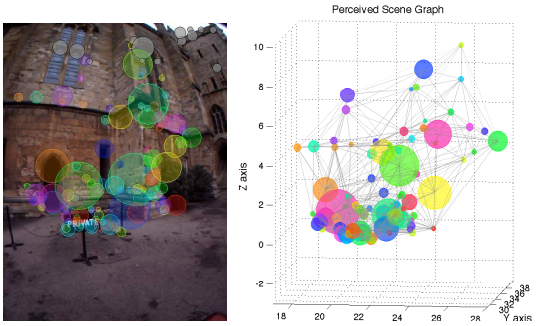


Figure 1. Random graph location model. Robot view (left) and perceived 3D constellation of visual features (right). 3D distances from lidar, stereo or structure from motion.

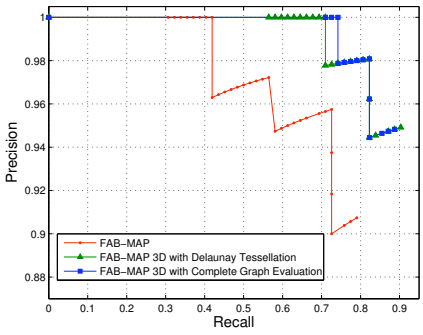


Figure 2. Precision-recall curves for the New College data set. FAB-MAP 3D has a higher recall of 74% at 100% than FAB-MAP that has 42% recall at 100% precision. The accelerated approach has marginally lower recall of 71% but still performs better than FAB-MAP.

FAB-MAP 3D provides substantial and compelling improvement in precision-recall performance over the existing FAB-MAP system, Figure 2. By capturing spatial information, the algorithm reduces the number of false positives and shows a dramatic decrease in false negative rate, particularly in scenes possessing a large number of common words where a loop closure decision hinges on spatial information. The framework shows robustness to perceptual aliasing as well as scene change. The system scales linearly with the number of places in the map. Graph inference can be accelerated by executing a Delaunay tessellation of the observed graph scaling log-linearly with scene complexity.

Efficient Surface and Feature Estimation in RGBD

Zoltan-Csaba Marton, Dejan Pangercic, Michael Beetz
Intelligent Autonomus Systems Group, Technische Universität München, {marton, pangercic, beetz}@cs.tum.edu

***Abstract*— Extracting useful information of RGBD images at high frame-rates requires fast algorithms that are more than just individual steps in a pipeline. Ideally, they should be doing as much as possible in a single step, and re-use information from previous computations. We will present our first experiences with such a pipeline, consisting of open-source algorithms from the Point Cloud Library (PCL) of ROS, and detail the modifications and parameters used for easy reproduction and extension.**

I. Motivation

Our approach for efficient object classification is based on a database of object scans using the kinect sensor. We perform smoothing, normal estimation, surface radius estimation and feature extraction using PCL, sped up by the use of voxelization. The locally labeled voxels are then used to calculate two features in parallel for each object hypotheses, and classified using SVM. While groups of points can be formed in an organized dataset by extracting pixel neighborhoods as well, using voxelization ensures spacial closeness and similar volume.

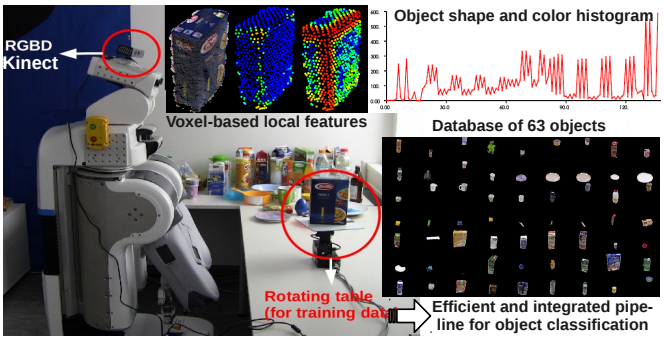


Fig. 1: We perform object classification based on RGBD data, and present the efficient processing algorithms that are used to achieve it, as well as our freely available object database.

A. Smoothing and Normal Estimation

In each voxel, the surface defined by the points can be estimated robustly using an MLS approach [1], and the cross product of two tangents of the fitted polynomial in a given point define the normal. Computing the points and normals once per voxel (by sampling multiple points on the surface) avoids nearest neighbor searches for each point and provides enough data to estimate the geometry, as detailed below.

B. Descriptive Local Geometric Features

The radius of the curves with the minimum and maximum curvature can be estimated as in [2], but based on the sampled points for each voxel. As these are metric values, they can be used intuitively to categorize the surfaces in the cells, which can be further used to build object-level features [3].

C. RGBD Feature For Object Classification

Having voxels with both RGB and geometric features enables the simultaneous use of two features operating on voxel neighborhoods: GRSD [3] and Color-CHLAC [4]. The former computes transitions between the surface types, while the latter color correlations. The features can be computed at once, on average in 0.26 s per object (consisting of 4632 points on average). Initial classification results on 63 objects using SVM show 76.4% accuracy, with a classification time of 0.05 s per object. This can be improved further by marking each view as a separate class during training, but considering only the object type when interpreting the results. Initial tests using the less accurate Hokuyo improved the success rate by 12%.

There are other features in PCL that can be used as well, and some of them can be even adapted to work on voxels to achieve a speed-up and to combine them with the above two.

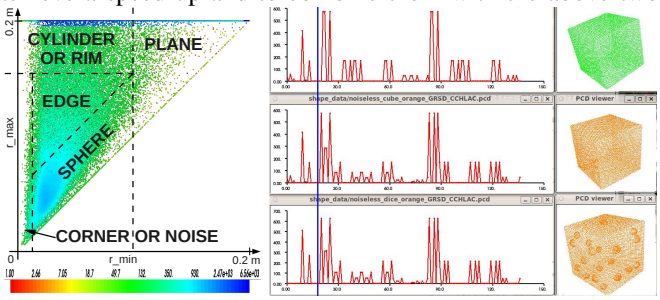


Fig. 2: Left: surface curve radius log-distribution for objects and main surface type associated with local feature ranges. Right: GRSD can not differentiate colors (identical histogram bins on the left), while ColorCHLAC can not differentiate the die from the cube (bins on the right). Their combination produces distinct signatures for them.

II. Results

Our results will be presented in a demo and the object database released for free use. The unreleased parts of the code will be published as part of PCL (`pointclouds.org`).

Acknowledgement: This work was supported by the CoTeSys (Cognition for Technical Systems) cluster of excellence at the Technische Universität München and Willow Garage, Menlo Park, CA. We would also like to thank Asako Kanezaki, as well as all PCL developers, contributors and supporters.

References

[1] Z. C. Marton, R. B. Rusu, and M. Beetz, "On Fast Surface Reconstruction Methods for Large and Noisy Datasets," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, May 12-17*, 2009.

[2] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, "General 3D Modelling of Novel Objects from a Single View," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 18-22 2010.

[3] Z.-C. Marton, D. Pangercic, R. B. Rusu, A. Holzbach, and M. Beetz, "Hierarchical object geometric categorization and appearance classification for mobile manipulation," in *Proceedings of 2010 IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA, December 6-8 2010.

[4] A. Kanezaki, T. Harada, and Y. Kuniyoshi, "Partial matching of real textured 3D objects using color cubic higher-order local auto-correlation features", in *The Visual Computer*, 26(10):12691281, 2010.