Multi-View Probabilistic Segmentation of Pome Fruit with a Low-Cost RGB-D Camera

Pablo Ramon Soria[1], Fouad Sukkar[2], Wolfram Martens[3],
B.C. Arrue[1], Robert Fitch[2]

[1] University of Seville, Seville 41092, Spain,
{prs, barrue}@us.es
[2] University of Technology Sydney, Ultimo NSW 2007, Australia
fouad.sukkar@student.uts.edu.au, rfitch@uts.edu.au
[3] The University of Sydney, Sydney NSW 2006, Australia
w.martens@acfr.usyd.edu.au

Abstract. Fruit harvesting is a topic of intereset in agricultural industries. In order to perform this task, robots should be able to recognize and segment fruit in their perceptual environment. Particularly, apple trees are often arranged as planar trellis structures in commercial orchards. The vine-like branches have leaves that can occlude fruit and produce noise in typical depth sensor data that also populates the scene with objects that are not of interest. In this paper, we present a method that uses a Dirichlet mixture of Gaussian processes and a Gibbs-Sampler for segmenting clusters of apples to support selective autonomous harvesting. Furthermore, the model provides probabilistic reconstruction of the entire apple which can be used for better grasping of the fruit.

Keywords: Probabilistic segmentation, RGB-D, agricultural robotics

1 Introduction

The emerging field of agricultural robotics has gained increasing interest in recent years. The growing demand for high-quality food worldwide, combined with inherently limited natural resources, has forced the agriculture industry to modernize assets in order to increase overall efficiency. The task of collecting fruit from trees is typically characterized by large quantities, but comparably low levels of selectivity, and an enormous requirement for manual labor. As a result, large quantities of fruit must be sold at low value or wasted, as it cannot be harvested at the right time (being either too mature or not ripe) or cannot be harvested at all due to inavailability of workers. Increasing the level of automation allows for continuous harvesting over wide time windows, and the intelligent selection of fruit to be picked at just the right time with lower labor cost.

Automation of agricultural processes using robots requires high levels of robustness compared to processes that are performed by humans or closely monitored by humans. In uncontrolled environments, such as typically encountered in outdoor fruit harvest scenarios, a robot needs to cope with high levels of

uncertainty in the perception of its environment, and the need for representations of uncertainty is twofold. On the one hand, for a robot to operate safely, it should not be steered into regions with high uncertainty, so that accidents are prevented. On the other hand, large scale automation of agricultural processes can only be achieved with increased levels of autonomy, where a robot actively explores its environment to reduce uncertainty.

This paper discusses several crucial components of an automized pome fruit harvesting pipeline using an RGB-D camera. We present an experimental setup where an Intel RealSense SR300 camera is mounted to a commercial robot arm, and collects multiple scans of an artificial apple trellis in order to increase the accuracy of perception. Accurate knowledge of the location and shape of the fruit reduces potential damage to the fruit and surrounding parts of the plant at the time of harvesting. Harvesting with robots is challenging because of the varying light conditions of outdoor environment and the large amount of occlusions by the leaves and branches.

Authors in [1] used a RGB-D sensor for the detection of apples. They use single scans from the camera and process them to quantify the amount of apples and their size. They first apply a color filter to remove the non-apple points, and then use Euclidean clustering for initial segmentation of the apples. Finally, apples that are close to the camera are segmented using a RANSAC algorithm with a model of a sphere.

Authors in [2] develop a classification algorithm for detection of peduncles of sweet peppers. They use a Support Vector Machine classifier for 3D points in a reconstructed scene fed with color features and point feature histograms.

In [3] the authors use simple color cameras for the detection and 2D segmentation of tomatoes using a set of filters and an adaptive threshold algorithm. They achieve good results for the segmentation of tomatoes, but their work is limited to 2D perception, and further steps are required for automatic harvesting of the tomatoes.

In [4] authors develop a multisensory system consisting of a multi-espectral camera, a TOF camera, an RGB-D camera and an artificial illumination system. They combine a set of filters for the input data combined with a pixel-wise classification algorithm. They achieve good results for different scenarios and fruits, but the cost of the entire system is large considering the need for mass employment as required in pome fruit harvesting.

In this paper we use probabilistic modeling of the fruit surface based on Gaussian Process Implicit Surfaces (GPIS) [5]. State-of-the-art GPIS algorithms typically assume a constant mean level function, which contains no prior knowledge about the shape of an object. This restriction is limiting the applicability of GPIS, especially when the data points for the surface reconstruction are not evenly distributed or large fractions of the objects have not been observed. In [6] authors propose the use of *a priori* learned shapes of objects to integrate prior knowledge in the GPIS process. This improves the surface reconstruction and can be used to improve segmentation of the object when significant parts of the scene have not been observed.

Here, we use a Dirichlet mixture of GPIS for probabilistic segmentation of the scene into distinct pieces of fruit and non-fruit components. Dirichlet mixture models are a systematic way of describing data association problems where latent components, such as an unknown number of objects in a scene, need to be inferred. The resulting probability space increases super-exponentially, and Markov-Chain Monte Carlo methods (such as Gibbs-sampling) are typically employed.

The remainder of the paper is organized as follows. Section 2 introduces the probabilistic segmentation model, and Section 3 presents the procedure of the multiple view selection for reducing the uncertainty on the segmentation and improving the object segmentation. In Section 4, the hardware system used for testing the algorithm is shown, and results are presented for a real apple vine. In Section 5, the paper concludes with a discussion of the presented results and further steps to be investigated.

2 Dirichlet segmentation

Segmentation is a common preprocessing step to higher-level tasks such as object detection, classification, or manipulation. Probabilistic representations of pointcloud segmentation are essential for active methods such as targeted exploration [7], decision making in grasping contexts [8] or probabilistic object detection [9]. We formulate segmentation as a data association problem, where each point n is assigned a label a_n, associating it to one of a set of objects present in the scene. With common pointcloud sizes, data association on a point level is generally impractical. Instead, we perform probabilistic segmentation on an oversegmented scene, as obtained using the voxel cloud segmentation algorithm presented in [10], with source code distributed as part of the Point Cloud Library (PCL) [11].

The surfaces of fruit are modelled by Gaussian process implicit surfaces [5], where observations of object surfaces are interpreted as points in the zero-level set of an underlying GP. As we are interested in scenes consisting of multiple objects, we consider mixtures of GPs, where the association of data points to different GPs is modelled by a Dirichlet process (DP). DP mixture models have the advantage that they readily deal with data sets where the number of latent clusters is not known. Each GP is then characterised by a set of hyperparameters, which can be used to represent the surface properties as well as prior shape and location [6].

Analytical representations of the resulting probability space are intractable, even for reasonably small problems, as it consists of all possible combinations of part associations. As a result, we employ a Markov-Chain Monte Carlo method that sequentially explores the probability space by producing samples according to a Gibbs-sampler. It is known that samples from a DP can be generated by sampling from a Chinese Restaurant process (CRP): the prior distribution of a data point's association is conditioned on the association of all other data points

and is given by

$$p(a_n = k|\mathbf{a}_{\setminus n}, \alpha) = \frac{N_k}{N - 1 + \alpha} \tag{1}$$

$$p(a_n = K + 1|\mathbf{a}_{\setminus n}, \alpha) = \frac{\alpha}{N - 1 + \alpha}, \tag{2}$$

where a_n denotes the association of point n, $\mathbf{a}_{\setminus n}$ denotes the association vector for all other points, N denotes the total number of points (including point n), N_k denotes the number of points associated to component K, and α denotes the Dirichlet process concentration parameter. The second expression $p(a_n = K+1)$ represents the probability that point n is associated to a new object without any points associated. The algorithm randomly selects a data point $\{\mathbf{x}_n, y_n\} \in \mathcal{D}_M$ and removes it from its current GP, before computing the likelihoods with respect to all GPs currently present in the scene, i.e. with number of associated data points $N_k > 0$. The GP likelihoods are weighted by N_k according to (1) and (2) and used to compute posterior association probabilities $p(a_n = k)$ by normalisation. Our algorithm produces a desired number of samples for the scene segmentation, which can be used to draw probabilistic conclusions about the scene segmentation and the location and shape of objects.

3 Multiple view segmentation

3.1 Pose generation and cloud registration

The vine-like structure of apple trees on a trellis comprises leaves, branches and fruit. The leaves can occlude the fruit and also induce noise in the observed point clouds. For these reasons the input point clouds are firstly filtered with a standard noise removal. At this stage, the clouds have fragments of the apple but morphologically these are identical to leaves since both are small planar surfaces.

In order to achieve better results in the probabilistic segmentation we perform registration of point clouds taken from multiple points of view. A pose planner generates camera poses such that the scene is observed from different angles. Figure 1 summarizes the multiview probabilistic segmentation process. It is assumed that the robot is positioned in front of the apple branches. First, a snapshot is taken and used to infer the relative angle of the apple "vine" with respect to the arm. Using this angle, we compute candidate poses reachable by the arm arranged on a portion of a sphere. Then, at each step of the algorithm a pose is selected that maximizes the distance from the history of past poses. From each viewpoint a new cloud is taken, which is registered using ICP plus information from the arm controller that provides an initial estimate of the point cloud's true pose.

Finally, the map of the scene is filtered by color using Euclidean distance between colors and a model value in HSV. Then, it is simplified using the voxel cloud segmentation algorithm described in [10] before being introduced into the Dirichlet segmentation process.

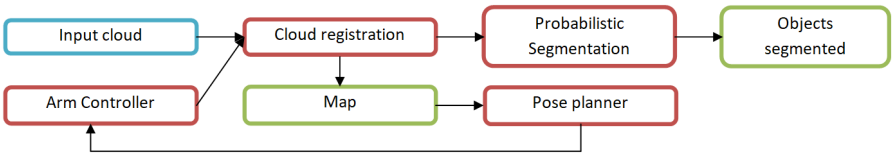


Fig. 1. System block diagram.

3.2 Arm controller

Path planning and control The arm path planning framework leverages recent breakthroughs in trajectory optimization algorithms that perform very well in high-dimensional configuration space. The algorithms used in our system include: TrajOpt [12], GPMP2 [13] and CHOMP [14]. In preliminary experiments we found that none of the algorithms performed consistently better than any other for all scenarios. CHOMP had the highest success rate, however, TrajOpt had significantly better computation time.

The algorithms can fail for several reasons, most often because they are inherently prone to getting trapped in local minima and there is no guarantee of ever finding a solution. Sampling based planners, such as RRT*, are another effective path planning method used in high-dimensional configuration spaces. A key property is that these planners are probabilistically complete, that is as the planner continues running, the probability of not finding a solution (if one exists) asymptotically approaches zero. Bi-directional RRT, a variant of RRT, exhibits a high planning success rate at the cost of sub-optimal paths [15].

We performed comparison experiments with 20 random end-effector positions on the sphere. As can be seen from results in Table 1 the Ranked planner (which runs all optimizers in parallel) performed the best, however still had a 2% failure rate. Therefore, the Ranked planner is used in our system and supplemented by Bi-directional RRT, which is used as a last resort if all others fail. The paths returned are geometric and often sparse, hence they are first time parameterised and up-sampled before being sent to the arm for execution.

Planner	Computation Time (s)	Success (%)
TrajOpt	0.09	94
GPMP2	0.09	68
Chomp	1.15	96
Ranked	0.088	98

Table 1. Average computation time and success rates per planner. Ranked planner runs all three optimisers in parallel and, unless all planners fail, returns first successful solution.

Fig. 2. Arm hardware setup and simulation in OpenRAVE

Arm Hardware In order for a robot arm to achieve an arbitrary end effector position and orientation in 3D space, i.e. a 6D pose, six degrees of freedom (DOF) are necessary [16]. Hence, manipulators suited for dexterous tasks, such as active sensing, should have at least six DOF. Redundant manipulators have seven or more DOF and offer greater dexterity and flexibility for maneuvering around obstacles. Rethink Robotics' Sawyer, a 7DOF robot arm, is used in our system. An open source SDK for the Sawyer arm provides a convenient API for executing trajectories and requesting state information.

To control the robot arm, its SDK provides a built-in Joint Trajectory Action Server (JTAS) which facilitates commanding the arm through multiple waypoints [17]. JTAS takes as input a list of timestamped waypoints and then determines appropriate joint velocity commands, through interpolation, to send to the arm so that the given trajectory is followed.

Simulation Environment The Sawyer arm and its environment are shown in Figure 2. The arm is simulated in OpenRAVE, an Open Robotics Automation Virtual Environment for developing and testing motion planners [18]. OpenRAVE can import standard robot models, such as COLLADA, allowing seamless integration with all of its interfaces, including motion planning libraries, inverse kinematics solvers and collision checkers. The motion planners are implemeneted as OpenRAVE planner plugins. Further, one of the strengths of OpenRAVE is its ability to robustly run multiple environments simultaneously in the same process. Hence, parallel path planning is well supported.

Function With Rest of the System The Arm Controller provides a TCP server front end which acts as the interface to the rest of the system. It listens on a TCP socket for messages, containing serialised commands, and once received deserialises and parses them for execution. The sequence of commands is as follows. First the Cloud Registration sends a request to the Arm Controller for the arm's current pose. Then the Pose Planner computes a target pose based on

the current pose and map information, which is then sent to the Arm Controller for planning and execution. The Arm Controller then lets the Cloud Registration node know whether the execution succeeded or failed. This process is repeated until no more viewpoints are needed.

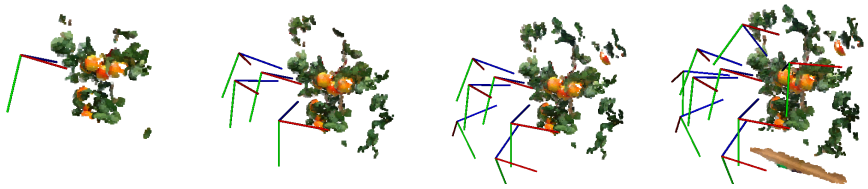
4 Experimental validation

In this section we discuss the experimental setup used for testing our system. We use a 1.5m by 1.5m section of an artificial apple trellis with the same characteristics as typical orchard trellises. The initial position of the arm with respect to the trellis is unknown at the beginning of each experiment. The only assumption we make is that the trellis is in the range of the camera (the Intel RealSense SR300 has a maximum range of 1.5m).

Figure 3 illustrates the map building result of an experiment using the registration procedure described in Section 3.

The first row in Figure 4 shows the input cloud that is fed into the Diritchlet segmentation algorithm. In the second row the result of the segmentation process is shown, where different colors are used to show the association to the different objects. Finally, the last row shows the centroids of the resulting objects. As the sampler tends to create new candidates of groups with single parts at random samples, a threshold has been set for a minimum number of parts for a group to be considered an object. As the number of viewpoints increases, more parts are added to the scene and an increasing number of apples are correctly located.

The computation time of the Dirichlet segmentation scales super-exponentially with the number of parts in the scene. The supervoxelisation algorithm in Section 2 over-segments the input clouds into supervoxels (which we refer to as parts). This algorithm depends on two input parameters parameters: seed size and voxel resolution [10]. Subfigure 5 (a) shows the effect of these parameters on the number of parts fed into the segmentation process for the same dataset. Subfigure 5 (b) shows the average time per sample in the Gibbs sampler. Finer parameters (lower voxel resolution and seed size) produce more parts, which increases the computation time.



(a) (b) (c) (d)

Fig. 3. Example of map building according to Section 3. The coordinate frames illustrate the inferred poses from where the point clouds were taken, and each cloud shows the reconstructed RGB-D pointcloud of the apple scene.

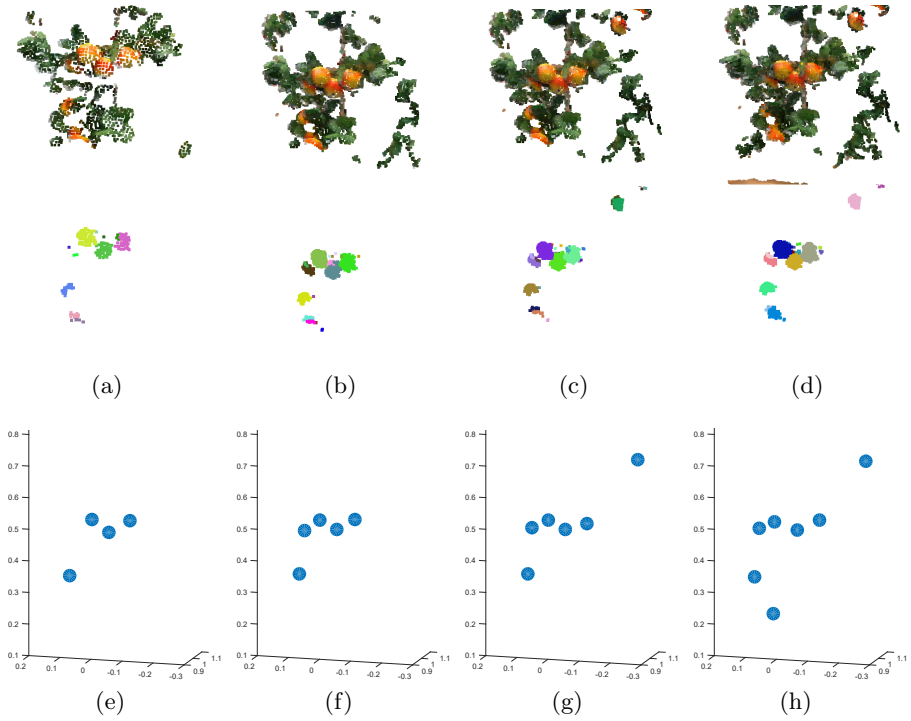


Fig. 4. Segmentation results for an increasing number of viewpoints from left to right. The first row shows the registered point clouds for multiple views, the second row shows the labeled groups representing the segmented apples, and the last row displays the centroids of apples that were segmented with a sufficiently large number of parts associated.

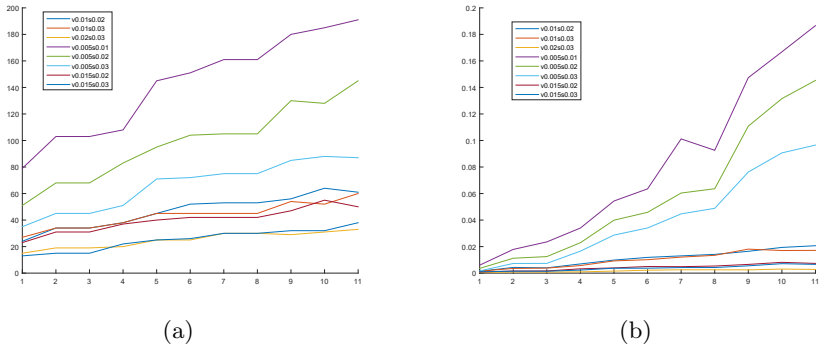


Fig. 5. Effect of the seed size and voxel resolution on the number of parts of the input cloud for the Dirichlet segmentation and the effect on the speed of the sampler. Note: the legend describes the voxel resolution (number after $'v'$) and seed size (number after $'s'$).

5 Discussion

We have presented a system for probabilistic segmentation of pome fruit, including hardware realization consisting of an Intel RealSense SR300 camera mounted on a commercial robotic arm, robust path planning algorithms, point cloud acquisition and registration, preprocessing (color-based presegmentation and cloud-based supervoxelisation) and MCMC-based proabilistic segmentation and fruit detection. We have investigated the robustness of the path planning and point cloud registration algorithms in hardware experiments and demonstrated the feasibility of the probabilistic segmentation algorithm.

The results of the GPIS- and DP-based probabilistic segmentation algorithm are promising in that they reliably detect apples and their exact location in the point cloud. Further, the probabilistic surface reconstruction using GPIS can be used for robust grasping algorithms [19] that take into account uncertainty in the fruit surface. Furthermore, probabilistic representations of the environment are essential for active methods to improve perception.

Figure 6 shows the pointcloud and segmentation results at an intermediate step of 5 registered scans. As was observed before, apples are correctly located

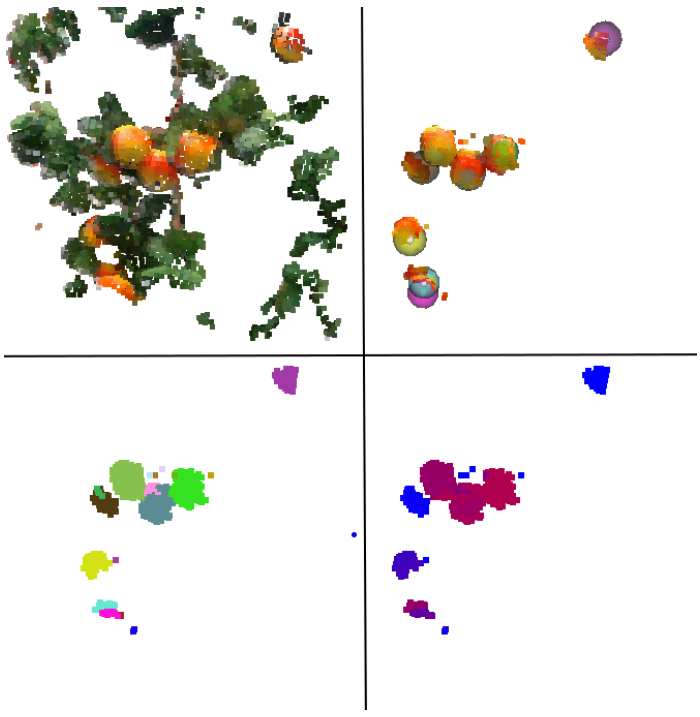


Fig. 6. Results of the probabilistic segmentation. The bottom right figure shows the association entropy map, where blue indicates low entropy and red indicates high entropy.

and segmented. Furthermore, the bottom right subplot shows the association entropy map, where blue indicates low entropy and red indicates high entropy. Those parts of the scene with high association entropy indicate that parts have been less consistently associated to the same objects compared to regions of lower entropy. Fruit that are densely cluttered (like in the center of the scene) generally result in larger segmentation uncertainty, whereas the single apple at the top right is confidently identified as one object. Such results can immediately be employed for grasping (start by picking the apple that is confidently segmented) and active perception (acquire more data in cluttered regions of the scene).

In future work we intend to increase the robustness of the segmentation algorithm. We have already employed a noise model that rejects parts that do not belong to objects, and in a more involved approach planar prior shapes can be used to model the shape of leaves. Our general goal is to close the loop around perception and provide an end-to-end system that makes observations with a low cost camera, registers point clouds, chooses viewpoints based on information theoretic measures and robustly moves the arm accordingly.

6 Acknowledgments

This work has been carried out in the framework of the AEROARMS (SI-1439/2015) EU-funded projects and the Australian Research Council's Discovery Projects funding scheme (DP140104203).

References

1. Nguyen, T.T., Vandevoorde, K., Kayacan, E., De Baerdemaeker, J., Saeys, W.: Apple detection algorithm for robotic harvesting using an RGB-D camera. In: Proc. of International Conference of Agricultural Engineering, Zurich, Switzerland. (2014)
2. Sa, I., Lehnert, C., English, A., McCool, C., Dayoub, F., Upcroft, B., Perez, T.: Peduncle detection of sweet pepper for autonomous crop harvesting combined color and 3-D information. IEEE Robotics and Automation Letters **2**(2) (April 2017) 765–772
3. Zhao, Y., Gong, L., Huang, Y., Liu, C.: Robust tomato recognition for robotic harvesting using feature images fusion. Sensors **16**(2) (2016) 173
4. Fernández, R., Salinas, C., Montes, H., Sarria, J.: Multisensory system for fruit harvesting robots: Experimental testing in natural scenarios and with different kinds of crops. Sensors **14**(12) (2014) 23885–23904
5. Williams, O., Fitzgibbon, A.: Gaussian process implicit surfaces. In: Gaussian Processes In Practice. (2006)
6. Martens, W., Poffet, Y., Soria, P.R., Fitch, R., Sukkarieh, S.: Geometric priors for Gaussian process implicit surfaces. IEEE Robotics and Automation Letters **2**(2) (April 2017) 373–380
7. van Hoof, H., Kroemer, O., Peters, J.: Probabilistic segmentation and targeted exploration of objects in cluttered environments. IEEE Trans. Robot. **30**(5) (2014) 1198–1209

8. Pajarinen, J., Kyrki, V.: Decision making under uncertain segmentations. In: Proc. of IEEE ICRA. (2015) 1303–1309
9. Cadena, C., Kosecká, J.: Semantic parsing for priming object detection in indoors RGB-D scenes. Int. J. Robot. Res. **34**(4-5) (2015) 582–597
10. Papon, J., Abramov, A., Schoeler, M., Wörgötter, F.: Voxel cloud connectivity segmentation - supervoxels for point clouds. In: Proc. of CVPR. (2013) 2027–2034
11. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: Proc. of IEEE ICRA. (2011) 1–4
12. Schulman, J., Ho, J., Lee, A., Awwal, I., Bradlow, H., Abbeel, P.: Finding locally optimal, collision-free trajectories with sequential convex optimization. Robotics: Science and Systems IX (2013)
13. Dong, J., Mukadam, M., Dellaert, F., Boots, B.: Motion planning as probabilistic inference using Gaussian processes and factor graphs. Robotics: Science and Systems XII
14. Zucker, M., Ratliff, N., Dragan, A.D., Pivtoraiko, M., Klingensmith, M., Dellin, C.M., Bagnell, J.A., Srinivasa, S.S.: Chomp: Covariant hamiltonian optimization for motion planning. Int. J. Robot. Res. **32**(9-10) (Jan 2013) 11641193
15. Kuffner, J., Lavalle, S.: RRT-connect: An efficient approach to single-query path planning. In: Proc. of IEEE ICRA. (2000) 995–1001
16. Hayashi, A.: Geometrical Motion Planning for Highly Redundant Manipulators Using a Continuous Model. PhD thesis, Austin, TX, USA (1991)
17. InternaSDK: *http* : *//sdk.rethinkrobotics.com/intera/arm_control_systems*
18. Diankov, R., Kuffner, J.: Introduction to the openrave architecture 0.9.0 *http* : *//openrave.org/docs/latest_stable/coreapihtml/architecture_concepts.html*
19. Dragiev, S., Toussaint, M., Gienger, M.: Gaussian process implicit surfaces for shape estimation and grasping. In: Proc. of IEEE ICRA. (2011) 2845–2850