

Importance Sampling based on Adaptive Principal Component Analysis

Jan Rosell, Luis Cruz, Raúl Suárez and Alexander Pérez
Institute of Industrial and Control Engineering (IOC)
Technical University of Catalonia (UPC)
Barcelona, Spain
e-mail: jan.rosell@upc.edu

Abstract—Sampling-based approaches are currently the most efficient ones to solve path planning problems, being their performance dependant on the ability to generate samples in those areas of the configuration space relevant to the problem. This paper introduces a novel importance sampling method that uses Principal Component Analysis to focalize the region where to sample in order to increase the probability of finding collision-free configurations. The proposal is illustrated with a 2D configuration space with a narrow passage and compared to the uniform random sampling method.

I. INTRODUCTION

Robot motion planning is already a mature discipline in robotics that deals with the problem of finding collision-free paths to move a robot from an initial to a goal configuration [1]. Planning is usually done in the Configuration Space (\mathcal{C} -space), where the robot is mapped to a point and the obstacles in the workspace are enlarged accordingly (\mathcal{C} -obstacles). Among the planning methods, the sampling-based ones are the best alternative to cope with problems with a high number of degrees of freedom. These methods avoid the explicit characterization of the \mathcal{C} -obstacles, requiring only the collision evaluation of a discrete set of sample configurations and the interconnection of the free ones in either roadmaps (PRM [2]) or trees (RRT [3]). These approaches were demonstrated to be probabilistic complete (e.g. Kavraki et al. [4] determined, for the basic PRM method, the number of samples necessary to achieve a probability of failure below a given threshold). Since this number may be quite large, the key issue for obtaining a good performance is the ability to generate samples in those areas of \mathcal{C} -space relevant to the problem, either by using importance sampling or dimension-reduction techniques [5], i.e. the generation of samples is one of the crucial factors in the performance of sampling-based planners.

Importance sampling strategies, on one hand, increase the density of samples in critical areas of the \mathcal{C} -space. These strategies have been classified by Hsu et al. [6] into: a) those that bias samples using workspace information (e.g. [7], [8]); b) those that over-sample the \mathcal{C} -space but quickly filter any not-promising configuration (e.g. [9], [10]); c) those that bias the sampling using the information gathered during the

construction of the roadmap or tree (e.g. [11], [12]); and d) those that deform (dilate) the free \mathcal{C} -space to make it more expansive to easily capture its connectivity (e.g. [13], [14]).

Dimension-reduction techniques, on the other hand, focus on defining the submanifolds of \mathcal{C} -space where the solution lies (or where a solution is more easily found), and where samples are to be obtained, like for instance submanifolds defined by those configurations that satisfy kinematic closure constraints [15], dynamic constraints [16], or a given set of task-dependant geometric constraints [17], [18], [19]. Among the dimension-reduction techniques, Principal Component Analysis (PCA) has been used to synthesize human-like motion in graphic applications [20], and also to capture the coupling between finger joints of anthropomorphic mechanical hands, either for the search of grasping configurations [21], or for the planning of collision-free paths for the hand-arm system [22].

The present paper proposes the use of PCA as an importance sampling method, i.e. to use PCA to focalize the region where to sample in order to increase the probability of finding collision-free configurations. The paper is structured as follows. Section II reviews the Principal Component Analysis Method; Section III presents the proposed approach, including the formal procedure, implementation issues and examples; Section IV discusses and evaluates the contributions and, finally, Section V presents the conclusions of the work.

II. PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) is a statistical technique used to process a set of vectorial samples with the aim of looking for a new base of the vectorial space whose axis indicate, in a decreasing order, the directions of the space with more information to discriminate the samples, i.e. the dispersion of the samples is maximal along the first direction of the new base and decreases along the remaining ones. This technique is frequently used to reduce the dimension n of the initial working space, using instead a subspace of dimension $m < n$ defined by the first m directions of the new base and neglecting the others. PCA is a common preprocessing step used to simplify the problem in pattern recognition and classification applications as well as in compression schemes and, in the field of motion and path planning, it is frequently used

This work was partially supported by the Spanish Government through the projects PI09/90088, DPI2008-02448 and DPI2010-15446.

A. Pérez is also with the Escuela Colombiana de Ingeniería “Julio Garavito”, Bogotá D.C.

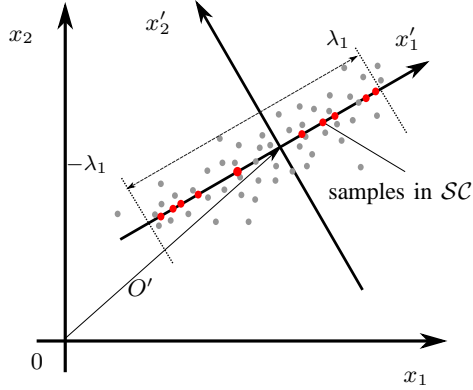


Fig. 1. Illustration of the use of PCA to reduce the dimension of the workspace. The original 2-dimensional space \mathcal{C} is defined by x_1 and x_2 , the mean of the set of samples is O' , the direction with maximal dispersion of samples is given by x'_1 , and the new base is defined by x'_1 and x'_2 . Then, the new working subspace is defined by x'_1 , and the valid portion is constrained to the range $[-\lambda_1, \lambda_1]$.

to reduce the dimension of the searching space and therefore decreasing the running time of the planning procedures.

There are different ways of performing the PCA [23]. Basically, it can be done by computing the eigenvalue decomposition of a data covariance matrix or the singular value decomposition of a data matrix, usually after mean centering the data for each attribute. The larger the eigenvalues or the singular values the larger the dispersion of the data along the corresponding eigenvector direction; the eigenvectors are directly used to define the directions of the new base.

Figure 1 shows a simple illustrative example of the use of PCA to reduce the dimension of the working space \mathcal{C} . The grey dots represent samples \vec{x} in a 2-dimensional space defined by the original variables x_1 and x_2 (which may represent two real features of the problem). O' represents the mean of the set of samples, so the samples are first modified as $\vec{x}' = \vec{x} - O'$. Then, using PCA, a new base defined by x'_1 and x'_2 (which could be considered 2 virtual features) determines a new reference system with the origin at O' . Now, since the dispersion of the samples is larger along x'_1 , the component x'_2 is neglected, which is equivalent to consider the subspace $\mathcal{SC} \subset \mathcal{C}$ defined only by x'_1 as the working space instead of \mathcal{C} , so the dimension of the working space was reduced from 2 to 1. Finally, the actual workspace for the generation of new samples is constrained to a portion of \mathcal{SC} defined by the range $[-\lambda_1, \lambda_1]$, such that it includes a desired percentage of the original samples in the former reference system.

III. PROPOSED APPROACH

A. The key idea

The key idea of the proposed approach is twofold: a) the use of PCA to define a new basis for the sampling space able to generate with a greater probability collision-free configurations in difficult areas of the \mathcal{C} -space (the whole new base is considered, i.e. no reduction of dimensionality is pursued); and b) the periodic recomputation of this basis as

Algorithm 1 PCA-based Sampling

Require:

- R_S : region of \mathcal{C} -space
- V_S : sampling volume (oriented hyper-box)
- S : set of at least d collision-free samples from R_S
- α : value in the range $[0, 1]$
- k : number of configurations sampled at a time

Ensure:

- S enlarged with up to k new collision-free samples
- V_S updated using the new collision-free samples

$r = \text{RAND}()$

if $r \geq \alpha$ **then**

$S_k = \text{SAMPLE-FROM}(R_S, k)$

else

$S_k = \text{SAMPLE-FROM}(V_S, k)$

end if

if $\text{CARD}(S_k) > 0$ **then**

$V_S = \text{PCA}(S \cup S_k)$

end if

$S = S \cup S_k$

return S, V_S

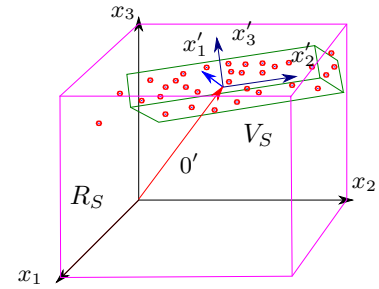


Fig. 2. Region R_S and sampling volume V_S obtained using PCA for a three dimensional \mathcal{C} -space.

new collision-free configurations are obtained, i.e. make the process adaptive to obtain a continuous improvement of the sampling performance. The sampling procedure proposed is conceived as a local method, i.e. it is to be applied to a region R_S of the \mathcal{C} -space where the area of interest is known to be located (e.g. a narrow passage).

The principal component analysis requires a minimum number of samples equal to the dimension of the space. Therefore, these samples are first obtained from R_S . Then, PCA is applied and the new basis is used to define a sampling volume V_S where to obtain new samples (Fig. 2). It may be the case, however, that V_S do not cover the whole area of interest due to the particular set of samples used for the principal component analysis. Therefore, V_S will be used to generate new collision-free configurations, but it will be recomputed again to try to effectively cover it. This recomputation will be performed with new collision-free configurations sampled from both V_S and R_S . In summary, the sampling process proposed is based on the sampling of configurations from both the region R_S of the \mathcal{C} -space as well as from the sampling volume V_S , that is periodically recomputed using the new sampled collision-free configurations.

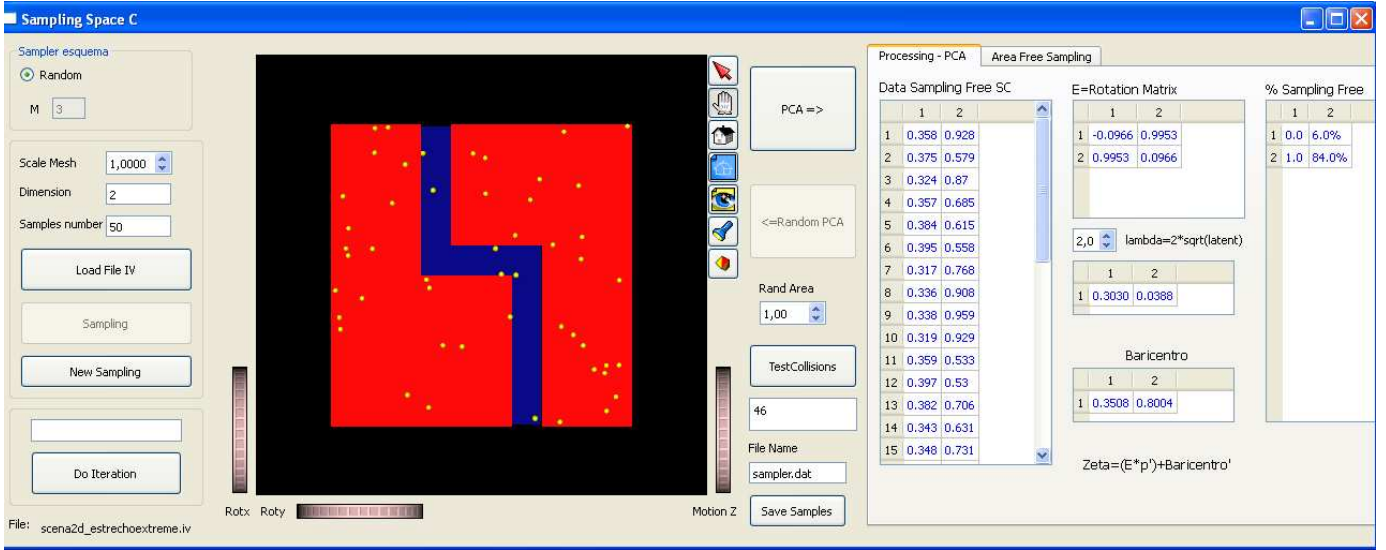


Fig. 3. Graphical user interface with the workspace used as test.

B. Procedure

Algorithm 1 describes the sampling procedure. It uses the following nomenclature and functions:

- d : Dimension of the configuration space.
- $CARD(S)$: Returns the cardinality of set S .
- $RAND()$: Returns a random value in the range $[0, 1]$.
- $SAMPLE-FROM(B, n)$: Samples n configurations from region B (a hyper-box of dimension d) and returns those collision-free.
- $PCA(S)$: Performs the Principal Component Analysis over the samples of the set S and returns an hyper-box aligned with the resulting new base, centered at the mean value of S , and with the length of each side equal to two times the deviation of the data in the corresponding axis.

The sampling procedure proposed has five parameters:

- R_S : The region of C -space where the PCA-based sampling is to be performed.
- S : A set of at least d collision-free samples from R_S .
- V_S : The sampling volume obtained by performing the Principal Component Analysis to the set S .
- α : A value in the range $[0, 1]$ expressing the probability to sample from R_S or from V_S .
- k : The number of configurations sampled per call to the sampling function $SAMPLE-FROM$.

The algorithm computes a random value r in the range $[0, 1]$ and compares it to the threshold α . If r is greater than α a set of k samples are obtained from R_S and those collision-free are returned; otherwise the samples are obtained from V_S . The algorithm returns the set S enlarged with the new collision-free configurations, and the new sampling volume V_S (when no new collision-free samples are found V_S is not recomputed).

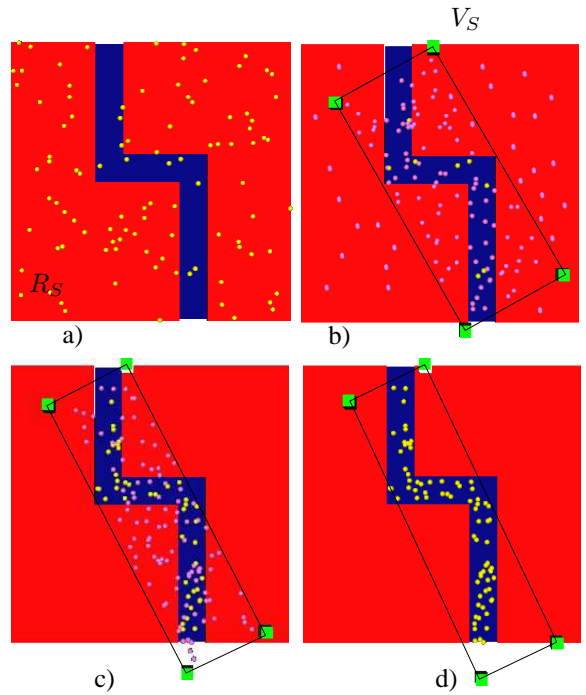


Fig. 4. Snapshots of the sampling procedure with $\alpha = 0.5$ and $k = 80$.

C. An Example

The proposed approach is illustrated using the S-shaped narrow passage shown in Fig. 3, where the whole GUI developed to test the algorithms is visible. Fig. 4 and 5 show some snapshots of the sampling procedure, that have been run with $\alpha = 0.5$ and different values of k :

- In the example of Fig. 4 a value of $k = 80$ has been used. Snapshot **a** shows the first configurations sampled from R_S . Snapshots **b** and **c** illustrate the volume V_S and new configurations obtained after calling the sampling

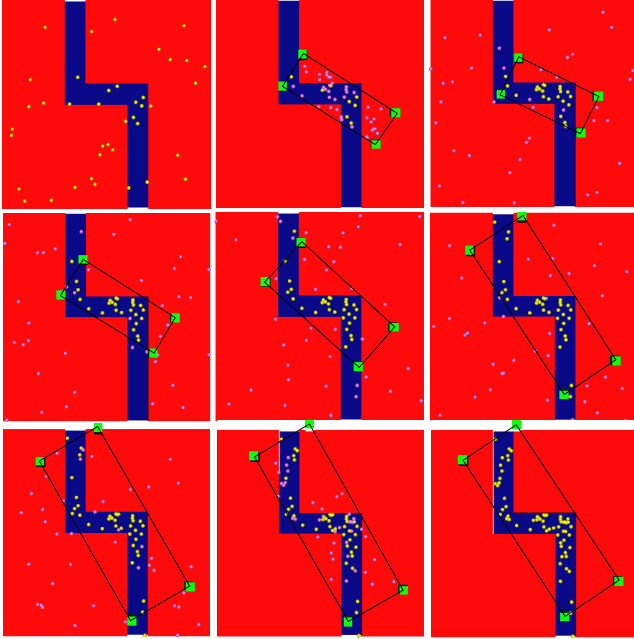


Fig. 5. Snapshots of the sampling procedure with $\alpha = 0.5$ and $k = 40$.

procedure using R_S and V_S , respectively. Snapshot **d** shows the final result. Using a high value of k , the first sampling results in collision-free configurations all along the passage and therefore the first computation of V_S is reasonably good. After two calls to the sampling procedure, i.e. after using a total of 300 samples, the narrow passage is covered with the dispersion required.

- In the example of Fig. 5 a value of $k = 40$ has been used. In this case the first computation of V_S is not so good and more iterations are required. V_S is recomputed each time the sampling results in new collision-free configurations. The final snapshot (bottom-right) shows the resulting volume V_S and the narrow passage properly covered.

D. Implementation

The proposed approach has been implemented in C++ using open-source cross-platform libraries such as Qt [24] for the user interface, Coin3D [25] for the graphical rendering and PQP [26] for the collision detection. For the computation of the PCA one possibility is to use Octave [27] or R [28] together with a package like the RCPP [29] to connect them to the application. However, since our application requires to perform PCA within the sampling loop, the performance criteria is considered a key factor, and for this reason the Armadillo C++ Linear Algebra Library has been used [30]. This library is open-source and has a good performance in response time for large volumes of data, as shown in the experiments reported in Table I.

IV. EVALUATION AND DISCUSSION

This section evaluates the proposal and compares it to the uniform random sampling. The evaluation is done on

TABLE I
PERFORMANCE OF THE ARMADILLO LIBRARY FOR THE EXECUTION OF PCA ON SETS OF 10,000 SAMPLES OF DIFFERENT NUMBER OF DEGREES OF FREEDOM.

d.o.f	2	3	5	7	9	11	13
PCA time (ms)	0.14	0.26	0.55	1.04	1.51	2.20	3.01

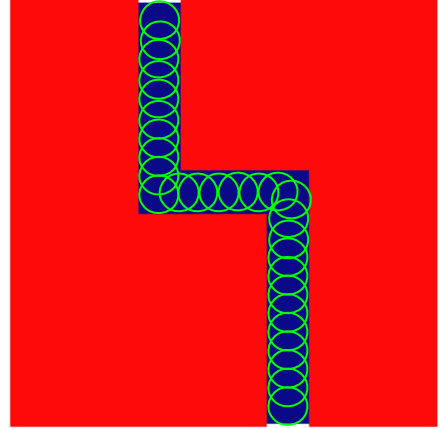


Fig. 6. Set of circumferences used to test the coverage of the narrow passage.

the example shown in Section III-C. The narrow passage has been populated with circumferences of radius r equal to the passage clearance, whose centers are located along the passage central axis and separated r (see Fig. 6). Then, the parameter to evaluate a sampling procedure is the number of samples N to be generated in order to populate each circumference with at least one sample, i.e. the number of samples required to achieve a given dispersion of the samples on the narrow passage (with those samples in the narrow passage a probabilistic roadmap can for sure connect the entrance to the exit, although it may be achieved with less samples).

Using uniform random sampling, the number of samples required to solve this problem has been $N = 484 \pm 29$. this value is compared to the value required when using the proposed PCA-based sampling algorithm. The value of N has been computed for different values of α and k . The evaluation procedure is shown in Algorithm 2, where the function EVALUATE(S) returns TRUE when all the test circumferences are populated, or FALSE otherwise. The cases of $\alpha = 0$ and $\alpha = 1$ are not considered since the first one corresponds to the uniform random sampling, and the second one only samples in V_S , and therefore if the first computation of V_S does not cover well the narrow passage it is completely useless (i.e. the algorithm always fails).

The results are shown in Table II and represented graphically in Fig. 7. It can be seen that the use of the proposed sampling method always reduces the number of samples required, being the maximum reduction of 57%, obtained for $k = 80$ and $\alpha = 0.8$. Even using a small ratio of sampling

Algorithm 2 Evaluation

Require:

R_S : region of \mathcal{C} -space
 α : value in the range $[0, 1]$
 k : number of configurations sampled at a time

Ensure:

N : total number of samples generated

$S = \text{SAMPLE-FROM}(R_S, k)$

if $\text{CARD}(S) < d$ **then**

repeat

$S = \text{SAMPLE-FROM}(R_S, 1)$

until $\text{CARD}(S) = d$

end if

$V_S = \text{PCA}(S)$

$N = d$

repeat

$(S, V_S) = \text{PCA-based Sampling}(R_S, V_S, S, \alpha, k)$

$N = N + k$

until $\text{EVALUATE}(S) = \text{TRUE}$

return N

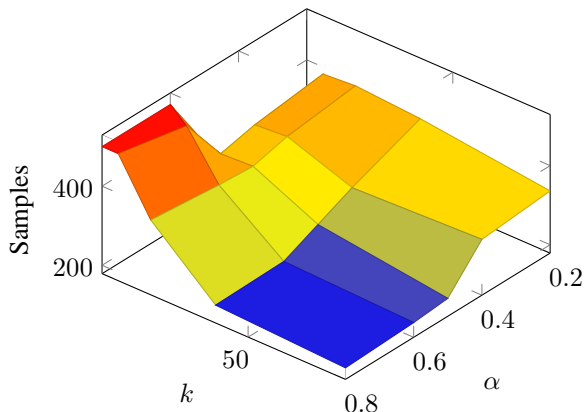


Fig. 7. Mean number N of samples required as a function of k and α .

within V_S , e.g. using $\alpha = 0.2$, the reduction of N is significant for any value of k . Similar results have been obtained in other 2D scenarios. It can be noted that:

- Using large values of k the number of required samples decreases for increasing values of α . The reason is that when using large values of k , the first sampling volume V_S obtained with the PCA already covers quite well the whole narrow passage because a representative set of configurations has been sampled, and therefore better results are obtained when the probability of sampling in V_S is higher (i.e. for larger values of α).
- Using small values of k , the lowest value of N is obtained for values of α equal or below 0.5. In this case, the first approximations of V_S may not be good because they are obtained with a set of configurations not representative enough, and therefore the search of more configurations outside V_S is required. This can be achieved with a lower value of α in order to increase the probability of sampling in R_S . The value of α must be, however, not too small since then the information captured by the PCA is hardly

TABLE II

MEAN NUMBER N OF SAMPLES GENERATED AS A FUNCTION OF α AND k .

k	α				
	0.2	0.4	0.5	0.6	0.8
5	280±40	239±21	351±25	—	—
10	382±63	360±56	340±65	464±70	—
20	388±82	366±53	344±55	350±42	368±50
40	376±49	308±49	268±43	228±47	224±26
80	336±73	320±51	224±36	216±35	208±33

used and the results tend to approach those obtained with pure uniform sampling on the whole region R_S . For large values of α , on the other hand, the method could not achieve the given dispersion using a reasonable amount of samples, and hence no values appear in the table.

Considering $k = m^d$, the values of k classified as small in the 2D example correspond approximately to $m \leq 5$. For high-dimensional configuration spaces, due to the exponential growth of the volume to be sampled, reasonable values of k will always be small (i.e. from a practical point of view m will surely be in that range), and therefore in this case it is advisable, as a general rule, to use $\alpha \leq 0.5$. As a future work we are planning to make α adaptive, starting with a low value and making it increase as V_S improves the coverage of the difficult region of the \mathcal{C} -space.

V. CONCLUSIONS

In order to improve the efficiency of sampling-based methods, this paper has proposed a method to bias samples towards difficult regions of configuration space, like narrow passages, based on the use of Principal Component Analysis applied to the collision-free configurations. The sampling procedure proposed is conceived as a local method, i.e. it is to be applied to a region R_S of the \mathcal{C} -space where the area of interest is known to be located. The use of PCA results in a new basis of the configuration space where a sampling volume can be defined. Configurations sampled from that volume are more prone to be collision-free, thus improving the sampling efficiency. This sampling volume is iteratively updated to assure the coverage of the region of interest, as new collision-free configurations are sampled.

The proposal has been implemented and evaluated satisfactorily, and results from a 2D example have been reported. Its use in high-dimensional configuration spaces is now under development, in particular to solve the problem of finding collision-free samples of a hand-arm robotic system, with up to 19 degrees of freedom, near the cluttered grasping configurations.

REFERENCES

- [1] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion*. The MIT Press, 2005.
- [2] L. E. Kavraki and J.-C. Latombe, "Randomized preprocessing of configuration for fast path planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 3, 1994, pp. 2138–2145.

- [3] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 995–1001.
- [4] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 1, pp. 166–171, Feb. 1998.
- [5] R. Geraerts and M. H. Overmars, "Sampling and node adding in probabilistic roadmap planners," *Robotics and Autonomous Systems*, vol. 54, p. 165173, 2006.
- [6] D. Hsu, J.-C. Latombe, and H. Kurniawati, "On the probabilistic foundations of probabilistic roadmap planning," *The International Journal of Robotics Research*, vol. 25, no. 7, pp. 627–643, 2006.
- [7] J. P. van der Berg and M. H. Overmars, "Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners," *The International Journal of Robotics Research*, vol. 24 (12), pp. 1055–1071, 2005.
- [8] H. Kurniawati and D. Hsu, "Workspace-based connectivity oracle: An adaptive sampling strategy for PRM planning," in *Algorithmic Foundations of Robotics VII*, S. Akella and et.al., Eds. Springer-Verlag, 2006.
- [9] V. Boor, M. H. Overmars, and A. F. van der Stappen, "The Gaussian sampling strategy for probabilistic roadmap planners," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1999, pp. 1018–1023.
- [10] D. Hsu, T. Jiang, J. Reif, and Z. Sun, "The bridge test for sampling narrow passages with probabilistic roadmap planners," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 4420–4426.
- [11] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. K. Overmars, "Probabilistic roadmaps for path planning in high - dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, August 1996.
- [12] D. Hsu, G. Sanchez-Ante, and Z. Sun, "Hybrid PRM sampling with a cost-sensitive adaptive strategy," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 3874–3880.
- [13] M. Saha, J. C. Latombe, Y. C. Chang, and F. Prinz, "Finding narrow passages with probabilistic roadmaps: The small-step retraction method," *Autonomous robots*, vol. 19(3), pp. 301–319, 2005.
- [14] H. L. Cheng, D. Hsu, J. C. Latombe, and G. Sanchez-Ante, "Multi-level free space dilation for sampling narrow passages in prm planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 1255–1260.
- [15] J. Cortés and T. Siméon, "Sampling-based motion planning under kinematic loop closure constraints," in *Proc. of the 6th Int. Workshop on the Algorithmic Foundations of Robotics*, 2004, pp. 59–74.
- [16] J. J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, "Dynamically-stable motion planning for humanoid robots," *Autonomous Robots*, vol. 12, no. 1, pp. 285–300, 2002.
- [17] A. Rodríguez, A. Pérez, J. Rosell, and L. Basañez, "Sampling-based path planning for geometrically-constrained objects," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 2074–2079.
- [18] M. Stilman, "Task constrained motion planning in robot joint space," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007, pp. 3074–3081.
- [19] D. Berenson, S. Srinivasa, D. Ferguson, and J. Kuffner, "Manipulation planning on constraint manifolds," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 625–632.
- [20] A. Safonova, J. K. Hodgins, and N. S. Pollard, "Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces," *ACM Transactions Graph.*, vol. 23, no. 3, pp. 514–521, 2004.
- [21] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 851–867, July 2009. [Online]. Available: <http://dx.doi.org/10.1177/0278364909105606>
- [22] R. Suárez, J. Rosell, A. Pérez, and C. Rosales, "Efficient search of obstacle-free paths for anthropomorphic hands," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 1773–1778.
- [23] I. Jolliffe, *Principal Component Analysis*. Upper Saddle River, NJ, USA: Springer Series in Statistics, 2002.
- [24] J. Blanchette and M. Summerfield, *C++ GUI Programming with Qt 4*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2006.
- [25] Kongsberg Oil & Gas Technologies, "Coin3D - 3d graphics development tools," www.coin3d.org, December 2010.
- [26] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, "Fast proximity queries with swept sphere volumes," in *Proc. of Int. Conf. on Robotics and Automation*, 2000, pp. 3719–3726.
- [27] J. W. Eaton, *GNU Octave Manual*. Network Theory Limited, 2002.
- [28] T. R. Foundation, "The R project for statistical computing," www.r-project.org, February 2011.
- [29] D. Eddelbuettel, "Rcpp: Seamless R and C++ integration," dirk.eddelbuettel.com/code/rcpp.html, February 2011.
- [30] C. Sanderson, "Armadillo: An open source c++ linear algebra library for fast prototyping and computationally intensive experiments," NICTA, Tech. Rep., 2010.