

Semester
Wintersemester

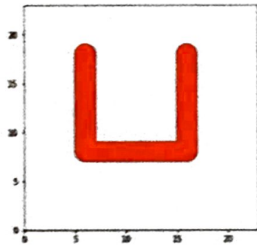
Jahr
21/22

Datum
14.12.2021

Projektaufgabe 7 – Sampling Strategien

Untersuchen Sie den Einsatz von unterschiedlichen Sampling-Strategien, um eventuell die Planungsergebnisse zu verbessern, d.h.

- 1.) Implementieren Sie das Gauß-Sampling und das Bridge-Sampling, das als Eingangsparameter die Anzahl der zu generierenden Punkte in der Nähe von Hindernissen erwartet und nicht eher stoppt, bis diese Anzahl von Punkten generiert wurde.
Stellen sie die erzeugten Punkte graphisch dar, d.h.



- 2.) Stellen Sie für mind. 10 interessante Benchmarkumgebungen das Ergebnis für jeweils das Gauss-Sampling und das Bridgesampling dar. Stellen Sie auch jeweils die benötigten Kollisionstests dar, die notwendig waren die vorgegebene Anzahl an Punkten zu generieren.
- 3.) Erweitern Sie die Roadmap Generierung des BasicPRM geeignet, so dass neben der gleichverteilten Roadmapgenerierung auch das Gauß-Sampling eingesetzt werden kann. Gehen Sie beim Aufbau der Roadmap wie folgt vor:
 - a. Erzeugen Sie einen Graphen mit einer gegebenen Anzahl von gleichverteilten kollisionsfreien Punkten
 - b. Fügen Sie diesem Graphen entsprechend die aus dem Gauß-Sampling erzeugten Punkte hinzu
 - c. Laufen Sie nun durch jeden Knoten und suchen Sie die Nachbarknoten im gegebenen Radius und fügen Sie eine Kante hinzu, wenn diese kollisionsfrei ist bzw. der Nachbarknoten nicht bereits in der selben Connected-Komponente war.

Prof. Dr. Björn Hein
Intelligente Produktion &
Cloud-Robotics

Hochschule Karlsruhe
Moltkestraße 30
76133 Karlsruhe

Tel.: +49 (0)721 925-1942
Fax: +49 (0)721 925-1947
bjoern.hein@h-ka.de
www.h-ka.de

Folgend ein Beispiel wie es ungefähr aussehen könnte (Anmerkung: Code ist nicht getestet):

```
def _learnRoadmapNearestNeighbour(self, radius, numNodes):
    # set up number of collision free nodes
    for i in range(numNodes):
        self.graph.add_node(i, pos=self._getRandomFreePosition())

    #*****
    # Insert here Gauss-Sampling
    #*****

    for node in self.graph.nodes(data=True):
        # Find set of candidates to connect to sorted by distance
        result = self._nearestNeighbours(node[1]['pos'], radius)

        for data in result:
            if self._inSameConnectedComponent(node[0], data[0]):
                break

        if not self._collisionChecker.lineInCollision(node[1]['pos'], data[1]['pos']):
            self.graph.add_edge(node[0], data[0])
```

- 4.) Evaluieren sie den so erweiterten BasicPRM mit dem ursprünglichen BasicPRM anhand von mind. 10 Benchmarks. Stellen Sie die Ergebnisse grafisch dar und diskutieren Sie diese.

Viel Erfolg!

14.12.2021

Error! Reference source
not found.

Seite 2