

# Data Visualization: Potential Fields Method

Till-Julius Krüger  
Freie Universität Berlin  
Institut für Informatik

Michael Pluhatsch  
Freie Universität Berlin  
Institut für Informatik

Anahid Roshandel  
Freie Universität Berlin  
Institut für Informatik

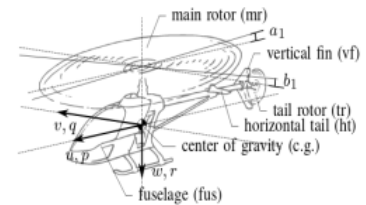
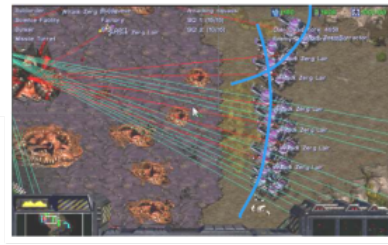
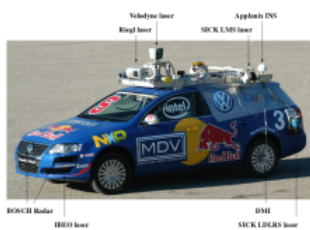


Figure 1: Anwendungsmöglichkeiten der *Potential Field Method* [5][7][10]

## ABSTRACT

Dieses Paper beschäftigt sich mit der Analyse und Planung einer Visualisierung der *Potential Fields Method* für Studierende des ersten Semesters und Interessierte nach dem Abitur. Dafür haben wir verschiedene vorhandene Ansätze betrachtet und analysiert. Aus den resultierenden Erkenntnissen haben wir einen eigenen Prototyp erstellt. Dieser erschafft zunächst mit dem Blackbox-Verfahren ein Verständnis für das Problem des Algorithmus. Danach sollen mit Hilfe des Whitebox-Verfahrens Einblicke in die Funktionsweise des Algorithmus gewährt werden. Nach der Implementierung soll die Visualisierung evaluiert werden.

## KEYWORDS

Potential Fields Method, Data Visualization, Analyse, Algorithmus

## ACM Reference Format:

Till-Julius Krüger, Michael Pluhatsch, and Anahid Roshandel. . Data Visualization: Potential Fields Method. In *Proceedings of Data Visualization (DataViz'18)*. Berlin, Germany, 7 pages.

## 1 PROBLEMBESCHREIBUNG UND EIGNUNG DES ALGORITHMUS

Der *Potential Fields Method* Algorithmus befasst sich grundlegend mit der Pfadplanung eines Roboters. Das Problem besteht dabei aus einem Roboter, der sich zu einem Ziel bewegen soll. Die Pfadplanung versucht einen Pfad vom Startpunkt des Roboters zum Ziel zu planen. Dabei sollen Kollisionen mit Hindernissen vermieden werden. Der Pfad sollte außerdem so kurz wie möglich sein.

*DataViz'18, Dezember 2018, Berlin, Germany*  
2018.

Im Falle der *Potential Fields Method* werden hierfür einfache physikalische Gesetze verwendet. Bei der Pfadplanung werden künstliche Kräfte bzw. Potentiale berechnet, die den Roboter zum Ziel weisen. Der Roboter wird vom Potentialfeld des Ziels angezogen, aber vom Potentialfeld des Hindernisses abgestoßen. Die Kräfte werden mit einigen Gleichungen berechnet, die in Abschnitt 2 erläutert werden.

Der Algorithmus behandelt ein abstraktes Problem, das auch ohne jegliche Vorkenntnisse verständlich ist. Daher ist er für Erstsemesterstudierende ohne fundiertes Wissen möglich zu verstehen. Man braucht lediglich mathematisches und physikalisches Verständnis aus dem Abitur. Für Erstsemesterstudierende könnte der Algorithmus für die Mathematik im Studium motivieren, da es ein mathematisches Konstrukt entmystifiziert. Scheinen die mathematischen Gleichungen im ersten Moment erschlagend, können sie durch eine gute Visualisierung Schritt für Schritt erschlossen werden.

Durch die variierbare Komplexität und mögliche Erweiterungen ist der Algorithmus aber auch für fortgeschrittene Studenten von Interesse. Der Algorithmus wird im Modul *Robotik* gelehrt und wird auch in der modernen Forschung noch verwendet. Beispiele sind hier autonome Automobile [3], Computerspiele wie *Starcraft* [5], Drohnen [9] und Mars-Rover [6]. Die ersten drei Beispiele sind in Bild 1 abgebildet.

## 2 ALGORITHMUS

Der Algorithmus *Potential Fields Method* nutzt einfache physikalische Elemente zur Vermeidung von Kollisionen mit einem Hindernis.

## Beschreibung

Es wird eine Kraft  $F$  beschrieben, welche die Bewegung des Roboters beeinflusst um das Ziel zu erreichen und Hindernissen auszuweichen. Diese Kraft besteht dabei aus zwei Komponenten, die für jede Konfiguration  $q$  berechnet werden kann. Hierbei stellt  $q$  eine Konfiguration im Konfigurationsraum dar (zum Beispiel eine  $xy$  Koordinate). Eine Konfiguration ist eine Position, die der Roboter anfahren kann. Der Konfigurationsraum enthält alle anfahrbaren Konfigurationen. Der Roboter hat dabei die Konfiguration  $q_r$ . In jeder Iteration des Algorithmus wird eine neue Position des Roboters berechnet:

$$q_r = \sigma \cdot \frac{F(q_r)}{|F(q_r)|} \quad (1)$$

Hierbei bezeichnet  $\sigma$  die Schrittweite des Roboters. Das bedeutet, dass dieser Parameter entscheidet, wie weit sich der Roboter in Richtung der berechneten Kraft  $F(q_r)$  bewegt. Aus diesem Grunde wird die  $F(q_r)$  normalisiert, sodass nur  $\sigma$  die Schrittweite beeinflusst.

$F(q_r)$  wird aus zwei Termen berechnet:

$$F(q_r) = F_{\text{anziehend}}(q_r) + F_{\text{abstoßend}}(q_r) \quad (2)$$

Unter  $F_{\text{anziehend}}(q_r)$  versteht sich eine anziehende Kraft, die den Roboter in Richtung einer Position leitet. Dies ist das Ziel des Roboters. Dadurch wird dafür gesorgt, dass der Roboter sich in Richtung des Ziels  $q_z$  bewegt. Hierfür gibt es verschiedene Variationen, die hier beschriebene Variante nennt sich *konisch* [1].

$$F_{\text{anziehend}}(q_r) = -\epsilon \cdot \frac{q_r - q_z}{|q_r - q_z|} \quad (3)$$

Es wird ein Vektor in Richtung des Ziels bestimmt und gewichtet. Dabei gibt es den Parameter  $\epsilon$ , mit dem die anziehende Kraft verstärkt oder abgeschwächt werden kann.

Nutzt man nur die anziehende Kraft  $F_{\text{anziehend}}(q_r)$ , bewegt sich der Roboter zwar zum Ziel, ignoriert auf seinem Weg allerdings jegliche Hindernisse. Um dem entgegen zu wirken, geht auch von den Hindernissen eine Kraft aus:  $F_{\text{abstoßend}}(q_r)$ . Diese wird für das Hindernis mit dem kleinsten Abstand  $p(q_r)$  mit Konfiguration  $q_h$  berechnet:

$$F_{\text{abstoßend}}(q_r) = \begin{cases} [0, 0], & \text{if } \phi > p(q_r) \\ \theta \cdot F_{\text{Erkennung}} \cdot F_{\text{Entfernung}} \cdot \frac{q_r - q_o}{|q_r - q_o|}, & \text{otherwise} \end{cases} \quad (4)$$

$$F_{\text{Erkennung}} = \left( \frac{1}{|p(q_r)|} - \frac{1}{\phi} \right) \quad (5)$$

$$F_{\text{Entfernung}} = \frac{1}{p(q_r)^2} \quad (6)$$

Das bedeutet, dass die abstoßende Kraft nur angewandt wird, wenn  $p(q_r) \leq \phi$  ist. Dadurch beginnt der Roboter dem Hindernis erst auszuweichen, wenn der Grenzwert  $\phi$  überschritten wird. Der Roboter erkennt also Hindernisse

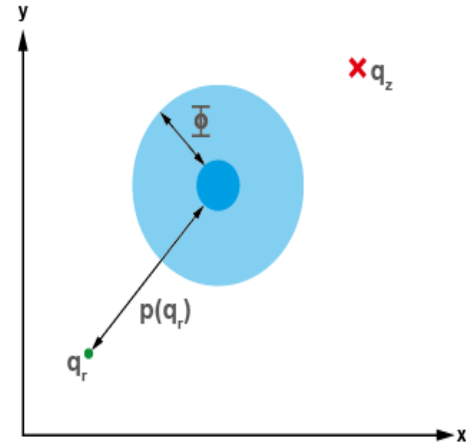


Figure 2: Darstellung der Parameter des Algorithmus.

in einem Abstand von  $\phi$ . Durch dieses Verhalten soll ein kürzerer Weg zum Ziel erreicht werden.

$\theta$  ist hier das Äquivalent von  $\epsilon$  für Hindernisse. Es dient zur allgemeinen Veränderung der abstoßenden Kraft. Mit  $F_{\text{Erkennung}}$  wird die Stärke der Kraft relativ zu einer Grenze  $\phi$  bestimmt. Der letzte Term  $F_{\text{Entfernung}}$  skaliert die Kraft im Verhältnis zur Entfernung des Hindernisses. Je näher es ist, desto stärker wird die abstoßende Kraft.

Die einzelnen Parameter sind in Bild 2 verdeutlicht.

## Verständnisschwerpunkte

Aus persönlicher Erfahrung liegt die Schwierigkeit beim Verständnis des Algorithmus bei den mathematischen Gleichungen, die das Zusammenspiel der Kräfte beschreiben. Werden nur die Gleichungen ohne ein anschauliches Beispiel betrachtet, erscheinen die Gleichungen abstrakt und komplex. Daher sollen die Gleichungen in die einzelnen Terme aufgebrochen werden. Durch eine Visualisierung sollen die Auswirkungen der Veränderung einzelner Terme sichtbar gemacht werden. Auf diesem Wege wird immer nur ein Teil der Gleichung betrachtet und sich dessen Veränderung bewusst gemacht. Durch diese Möglichkeit zum Experimentieren soll die Mathematik interaktiv verstanden werden.

## 3 VERWANDTE ARBEITEN

Wir haben verschiedene Visualisierungsmöglichkeiten gefunden.

### Beschreibung

- Bild 3 auf der nächsten Seite ist ein Ausschnitt einer Animation, die den Weg des Roboters zeigt. Die Darstellung verwendet eine Farbkarte zur Visualisierung der Kraft. Durch die Farbintensität wird verdeutlicht, welche Kraft an welcher Konfiguration herrscht. Ein dunkler

Farbwert steht für ein hoch abstoßenden Kraftwert, helle Werte sind hingegen anziehend. Daher sind die Hindernisse sowie Bereiche hinter dem Roboter sehr dunkel, das Ziel jedoch sehr hell. Durch den Farbverlauf kann eine Richtung erkannt werden, in die der Roboter forciert wird. Der Roboter bewegt sich entlang der roten Linie vom Startpunkt zum Ziel. Die Eingabedaten und Parameter des Algorithmus sind hier nicht veränderbar.

- Bei Bild 4 wird die Kraft jeder Konfiguration als Vektor in jeder möglichen Konfiguration dargestellt. Aus diesem Grund zeigen die Pfeile (Vektoren) vom Hindernis fort und in Richtung des Ziels. Der Kreis in Magenta stellt das Ziel dar, während das Hindernis durch einen grünen Kreis symbolisiert wird. In dieser Visualisierung ist der Einfluss von  $\phi$  gut erkennbar, da der Einflussbereich des Hindernisses sich kreisförmig um das Hindernis ausbreitet.
- Das letzte Beispiel (Bild 5 auf der nächsten Seite) zeigt ein verwandtes Thema. Es behandelt elektronische Felder. Diese entwickeln auch ein Kräfteverhältnis und visualisieren somit ein ähnliches Problem. Der grüne Kreis wird von den roten Kreisen angezogen. Die roten Kreise stoßen sich hingegen ab. Daher könnte der Roboter als grüner Kreis verstanden werden und rote Kreise als Ziele.

## Analyse

Wir haben auf den beschriebenen Beispielen eine morphologische Analyse (Tabelle 1 auf Seite 7) durchgeführt. Daraus ließen sich die folgenden Erkenntnisse ableiten.

- Bild 3: Die Visualisierung versucht, den Weg des Roboters zu erklären, indem die Kräfte der möglichen Konfigurationen zu jedem Zeitpunkt durch die Farbkarte sichtbar sind. Dadurch erhält der Nutzer ein Verständnis, warum der Roboter den zurückgelegten Weg gewählt hat. Der Nutzer muss dafür aber die gesamte Welt des Roboters im Blick haben, was im ersten Moment überfordernd wirkt. Das Verständnis wird daher nicht Schritt für Schritt aufgebaut und ist eher für fortgeschrittene Betrachter geeignet.
- Bild 4: Auch bei dieser Visualisierung wird das globale Kräftewirken verdeutlicht. Allerdings wird hier ein Vektorfeld benutzt. Die Vor- und Nachteile des vorherigen Bildes sind jedoch auch hier zu finden.
- Bild 5 auf der nächsten Seite: Der Vorteil dieser Visualisierung ist ein sehr klarer, einfacher Aufbau. Es wird nicht versucht, die wirkenden Kräfte des ganzen Feldes an jedem Punkt zu visualisieren. Dadurch ist die Visualisierung weniger überladen. Es bleibt überschaubarer und überfordert den Betrachter nicht. Allerdings stellt

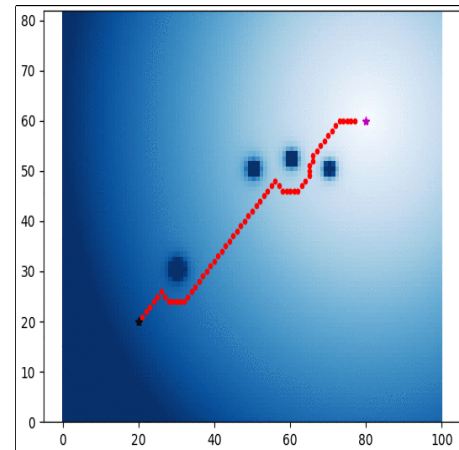


Figure 3: Beispielhafte Visualisierung mit einer Farbkarte [12].

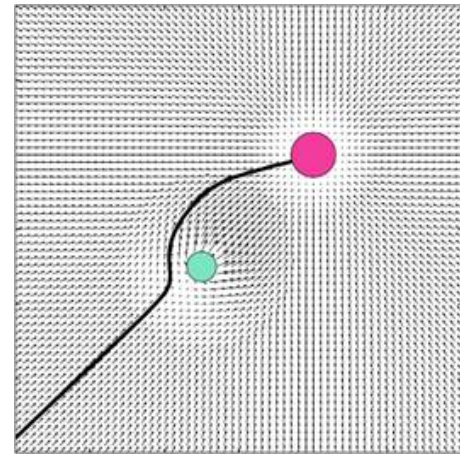


Figure 4: Beispielhafte Visualisierung als Vektorfeld [11].

es unseren Algorithmus nicht exakt dar, es gibt kein Äquivalent für Hindernisse.

Als Folge daraus werden wir versuchen die globale Farbkarte aus Bild 3, eine vereinfachte, lokale Vektordarstellung aus Bild 4 und die klare Einfachheit und hohe Interaktivität aus Bild 5 auf der nächsten Seite für unsere Visualisierung zu benutzen.

## 4 ANSATZ/PROTOTYP

Zunächst wird dem Nutzer in einer Einleitung das Problem der Pfadplanung erläutert. Konkret, wie kommt zum Beispiel ein Roboter von seinem Startpunkt zu einem Ziel. Die Studenten und Studentinnen bekommen ein Verständnis wofür die Pfadplanung dient. Gleichzeitig wird ihnen vermittelt, welche Probleme bei einer Pfadplanung auftreten können. So müssen zum Beispiel kürzeste Wege gefunden, Hindernisse

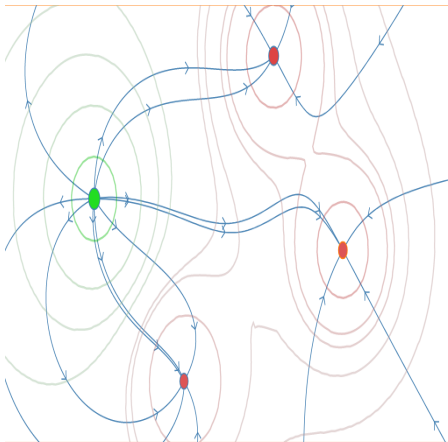


Figure 5: Interessante Visualisierung für elektronische Felder [2].

umgangen und Sackgassen gemieden werden. Anschließend wird mit der sogenannten *Potential Fields Method* eine Lösung vorgestellt. Die Hauptzielgruppe in diesem Projekt sind Erstsemesterstudierende im Fach Informatik. Jedoch wird der Algorithmus auf unterschiedliche Art und Schwierigkeit erklärt, so dass auch fachfremde Personen ein einfacher Einstieg in die Thematik gegeben werden soll.

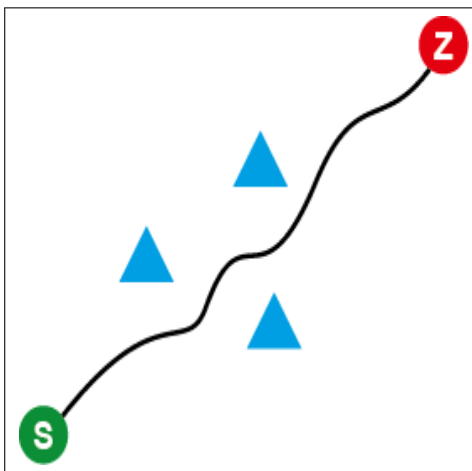


Figure 6: Abstrakte, statische Darstellung des Problems

Mit der ersten Visualisierung wird der Algorithmus allgemein vorgestellt und anhand einer Animation gezeigt. Bild 6 stellt dafür ein Beispiel dar. Dabei wird ein Roboter (bzw. ein Startobjekt) gezeigt, der sich in Richtung Ziel bewegt und die Hindernisse umgeht. Der zurückgelegte Weg wird nach und nach eingeblendet. Die Animation kann nicht verändert werden. Um die Animation zu steuern, gibt es einen Start-

und Stoppknopf. Neben der Animation wird das Problem mit Text vorgestellt.

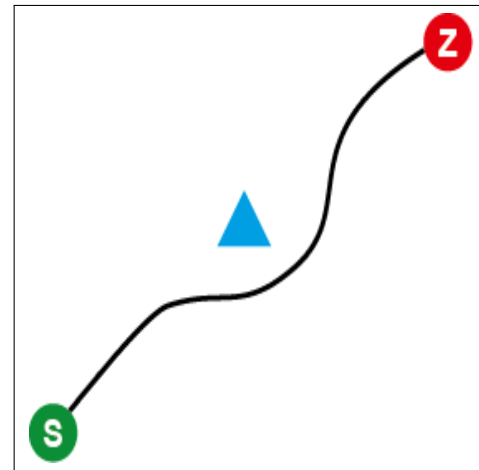


Figure 7: Interaktiv veränderbares Beispiel

Für die Erklärung des Algorithmus benutzen wir zwei Ansätze. Der erste Ansatz beruht auf dem sogenannten Black-box Verfahren. Das heißt, dass der Nutzer Eingaben verändern kann und ein entsprechendes Ergebnis erhält ohne die genauen Abläufe des Algorithmus zu kennen. Wir möchten den Nutzer spielerisch an die *Potential Fields Method* heranführen und ihn für den Algorithmus begeistern. Der Nutzer bekommt die Möglichkeit interaktiv mit den Objekten zu spielen. Das Bild enthält ein Start- und Zielobjekt und ein Hindernis, wie in Bild 7 dargestellt. Durch das Verschieben des Start- und Zielpunkts und das Setzen von Hindernissen wird dem Nutzer visuell und interaktiv vermittelt, wie sich der Algorithmus verhält und wie die Auswirkungen auf die Pfadplanung sind. Der Fokus im ersten Teil des Ansatzes liegt darin, allen Personengruppen die Thematik visuell zu erklären und eine Einführung zu geben.

Mit dem zweiten Ansatz, unter Zuhilfenahme des sogenannten Whitebox Verfahrens, verfolgen wir die Vermittlung eines tiefgründigen Verständnisses des Algorithmus. Der Fokus dieser Erklärung richtet sich an fachnahe Personen, speziell Erstsemesterstudierende. Hier wird dem Nutzer mit vorgegebenen Eingaben detailliert der Algorithmus erklärt. Es können Parameter mithilfe von Reglern verändert werden. In Bild 8 auf der nächsten Seite wird ein Weg in gestrichelter Linie gezeigt, welcher einem Standardweg entspricht. Werden Parameter verändert, so wird im Vergleich zum Standardweg ein neuer Weg mit den veränderten Parametern berechnet und gezeigt. Die Positionen der Objekte kann jedoch nicht verschoben werden, da es bei dieser Erklärung in erster Linie um die Auswirkungen der Parameter geht. Mit den Parametern werden die anziehenden und abstoßenden Kräfte verändert und visuell dargestellt. Dabei haben



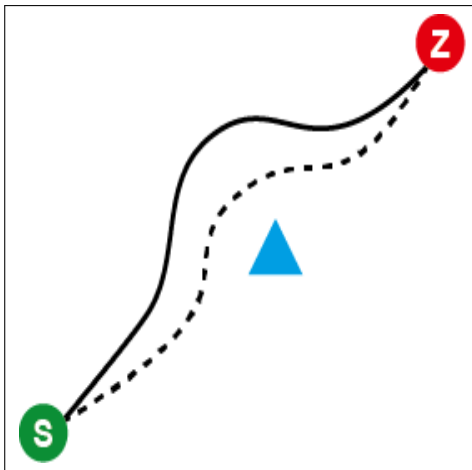


Figure 8: Whitebox Beispiel mit veränderbaren Parametern

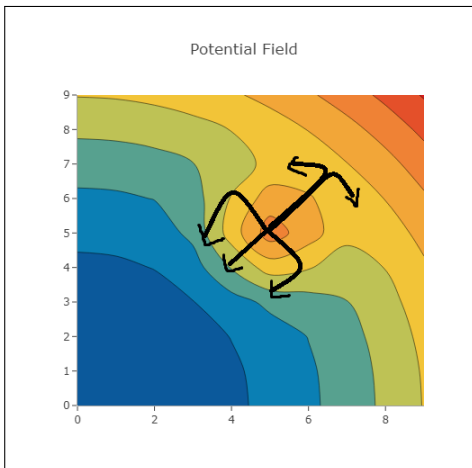


Figure 9: Kräfte dargestellt als Isokonturen und lokalen Flussgraphen

wir uns gegen die klassische Darstellung mit Vektorfeldern entschieden (Bild 4 auf Seite 3). Ungeübte Betrachter können nur schwer erkennen wie nicht zirkulierende Senken und Quellen [8] interagieren. Es ist nicht intuitiv ersichtlich wie bzw. wo ein Weg für die Pfadfindung entsteht. Ebenso verhält es sich mit den in Bild 5 auf der vorherigen Seite dargestellten elektronischen Feldern. Für ein besseres Verständnis haben wir einerseits eine aktivierbare und deaktivierbare Darstellung mit Isokonturen [8] und lokale Vektoren gewählt. Dabei stellen die unterschiedlichen Farben die anziehenden bzw. abstoßenden Kräfte dar, welche mit den lokalen Vektoren hervorgehoben werden (Bild 9). Andererseits wird diese Darstellung mit einer 3D Karte anschaulich erklärt. Die unterschiedlichen Farben repräsentieren hierbei verschiedene Höhenmeter. Objekte mit abstoßenden Kräften

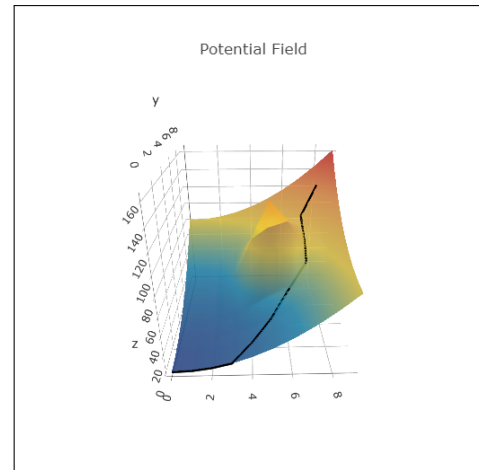


Figure 10: 3D Visualisierung der Kräfte mit Pfad

werden als Berge visualisiert und das Ziel mit der anziehenden Kraft als Tal. Ziel des Roboters ist ein möglichst schneller und einfacher Weg ins Tal (Bild 10). Dies ist vergleichbar mit einem Ball der ins Tal rollt.

Nachdem der Nutzer ein grundlegendes Verständnis über den Algorithmus vermittelt bekommen hat, möchten wir auf potenzielle Probleme hinweisen, wie z. B. in Bild 11. Dort sieht man, dass der Roboter in ein lokales Minimum geraten ist und dieses nicht mehr verlassen kann. Zum Schluss zeigen wir, dass der vorgestellte Algorithmus nicht nur ein theoretisches Konstrukt ist. Anhand von realen Beispielen wird der praktische Nutzen belegt (siehe Abschnitt 1 auf Seite 1).

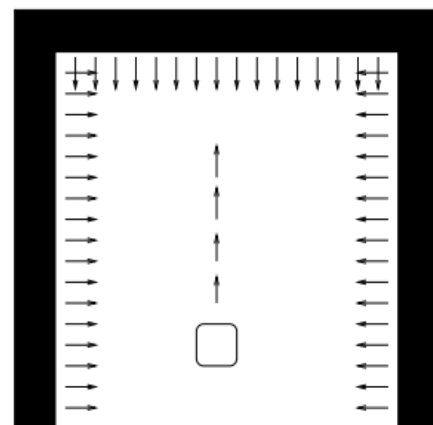


Figure 11: Box Canyon Problem [4].

## 5 ZIEL

Das Ziel des Projektes war bis zum 14.02.2019 die visuelle Darstellung und Erklärung einer Pfadplanung mit der Potential Fields Methode. Hauptzielgruppe sind Erstsemesterstudierende in der Informatik und Interessenten. Mehr als 50 Prozent der Tester sollten die Visualisierung und Erklärung verstehen. Wir möchten eine Basisversion der Visualisierung in der gegebenen Zeit erreichen.

## 6 EVALUATION

### REFERENCES

- [1] Nancy Amato. [n. d.]. Potential Field Methods. [http://portal.ku.edu.tr/~cbasdogan/Courses/Robotics/projects/algorithm\\_poten\\_field.pdf](http://portal.ku.edu.tr/~cbasdogan/Courses/Robotics/projects/algorithm_poten_field.pdf).
- [2] Pance Cavkovski. [n. d.]. Electrical field lines and Equipotential surfaces simulation. <http://hsilomedus.me/wp-content/uploads/d3electricField/electricField.html>.
- [3] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. 2009. Path planning for autonomous driving in unknown environments. In *Experimental Robotics*. Springer, 55–64.
- [4] Michael A. Goodrich. 2002. Potential Fields Tutorial.
- [5] Johan Hagelbäck. 2012. Potential-field based navigation in starcraft. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*. IEEE, 388–393.
- [6] Mauro Massari, Giovanni Giardini, and Franco Bernelli-Zazzera. 2004. Autonomous navigation system for planetary exploration rover based on artificial potential fields. In *Proceedings of Dynamics and Control of Systems and Structures in Space (DCSSS) 6th Conference*. 153–162.
- [7] Michael Montemerlo, Jan Becker, Suhrud Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, et al. 2008. Junior: The stanford entry in the urban challenge. *Journal of field Robotics* 25, 9 (2008), 569–597.
- [8] T. Munzner. 2015. *Visualization Analysis and Design*. CRC Press. <https://books.google.de/books?id=NfkYCwAAQBAJ>
- [9] Tobias Paul, Thomas R Krogstad, and Jan Tommy Gravdahl. 2008. Modelling of UAV formation flight using 3D potential field. *Simulation Modelling Practice and Theory* 16, 9 (2008), 1453–1462.
- [10] Tobias Paul, Thomas R Krogstad, and Jan Tommy Gravdahl. 2008. UAV formation flight using 3D potential field. In *Control and Automation, 2008 16th Mediterranean Conference on*. IEEE, 1240–1245.
- [11] Hani Safadi. 2007. Local Path Planning Using Virtual Potential Field. <https://www.cs.mcgill.ca/~hsafad/robotics/>.
- [12] Atsushi Sakai. [n. d.]. PythonRobotics. <https://github.com/AtsushiSakai/PythonRobotics>.

Kategorie	Kriterium	[12]	[11]	[2]
Interaktivität	Objekte verschiebbar	↯	↯	✓
Interaktivität	Verhalten änderbar	↯	↯	✓
Erklärbarkeit	Erklärender Text vorhanden	✓↯wenig	✓	Link Wikipedia
Erklärbarkeit	Überschrift / Unterschrift vorhanden	✓	✓	✓
Erklärbarkeit	Achsenbeschriftung vorhanden	(✓) ↯	(✓) ↯	↯
Erklärbarkeit	Legende vorhanden	↯	↯	✓
Erklärbarkeit	Selbsterklärend	✓	✓↯	✓↯
Erklärbarkeit	Statisch/Dynamisch	Dynamisch	Statisch	Dynamisch
White/Blackbox	Einblick in Algorithmus			✓
White/Blackbox	Algorithmus verständlich	✓	✓	↯
White/Blackbox	Probleminformation		↯	↯

Table 1: Morphologische Analyse