

Erschienen in:

Tagungsband zur Konferenz Go-3D 2017 „Mit 3D Richtung Maritim 4.0“,
Rostock, Fraunhofer Verlag, S. 135-146.

3D-Visualisierung von Über- und Unterwasserfahrzeugen zur Evaluation von Steuerungsalgorithmen mithilfe einer Game-Engine

Tobias Theuerkauff, Tobias Werner, Frank Wallhoff, Thomas Brinkhoff

Abteilungen Geoinformation und Technik und Gesundheit für Menschen
Institut für Technische Assistenzsysteme (ITAS), Institut für Angewandte
Photogrammetrie und Geoinformatik (IAPG)
Ofener Straße 16/19
26121 Oldenburg, Deutschland
tobias.theuerkauff@jade-hs.de
tobias.werner@jade-hs.de
frank.wallhoff@jade-hs.de
thomas.brinkhoff@jade-hs.de

Abstract: Simulationen von Über- und Unterwasserfahrzeugen erfolgen bislang durch eine Kombination unterschiedlicher Softwarebibliotheken. Game-Engines schaffen unter möglichst realen Bedingungen eine virtuelle Welt und bündeln physikalische Modelle und Visualisierung in einem Softwareprodukt. Sie bieten somit die Grundfunktionalitäten zur Durchführung von Simulationen. Mit dem Ziel eine Evaluation von Steuerungsalgorithmen vorzunehmen, fokussiert sich dieser Beitrag auf den Aspekt der Visualisierung von Über- und Unterwasserfahrzeugen in einer möglichst realitätsnahen Umgebung. Hierfür werden verschiedene Geländemodelle eingebunden, in denen ein Fahrzeug durch eine Netzwerkschnittstelle positioniert und gesteuert werden kann.

1 Motivation

Mit autonomen Unterwasserfahrzeugen (AUV) lassen sich Anwendungsfelder abdecken, bei denen bislang typischerweise ferngesteuerte Unterwasserroboter (ROV) zum Einsatz kommen. Ein Beispiel umfasst die Inspektion von Pipelines unter Wasser [FNR+15]. Unbemannte Überwasserfahrzeuge (USV) unterstützen dabei u. a. bei der Lokalisierung von Unterwasserfahrzeugen. Besonders die Weiterentwicklung im Bereich der dreidimensionalen Rekonstruktion von Objekten mithilfe von optischen Sensoren erlaubt die Erschließung neuer Anwendungsfelder [LLS17]. Dabei wird eine Software zur Steuerung von

Fahrzeugen benötigt, die unter Berücksichtigung der Umgebungseinflüsse auf die speziellen Anforderungen einer Mission angepasst ist. Vor dem praktischen Einsatz sollte frühzeitig eine Erprobung der Steuerungssoftware innerhalb einer virtuellen Simulationsumgebung erfolgen. Die visuelle Repräsentation einer Mission unterstützt dabei die Evaluation des Fahrzeugverhaltens. Die spezifischen Softwareprodukte stellen in der Regel eine Kombination aus mehreren unterschiedlichen Softwarebibliotheken dar, die entweder einen einzelnen Teilaspekt der Simulation abbilden oder die Anwendung um zusätzliche Funktionalitäten ergänzen (siehe Kapitel 2).

Ziel dieser Arbeit ist es, eine Simulationsmöglichkeit auf Basis einer Game-Engine zu entwickeln. Der Fokus liegt auf dem Teilaspekt der Visualisierung von Über- und Unterwasserfahrzeugen. Es sind Szenarien, wie das Testen von Steuerungsalgorithmen einzelner Fahrzeuge, als auch Missionen mit dem kombinierten Einsatz mehrerer Fahrzeuge (z. B. Leader-Follower-Mission) evaluierbar. Als Game-Engine wird die Unreal-Engine verwendet. Die Arbeit ist eine Ausgliederung aus dem Projekt „Entwicklung innovativer Technologien für autonome maritime Systeme“ (EITAMS)¹.

Bei einer Game-Engine handelt es sich um ein umfangreiches Softwaresystem, das ursprünglich als Grundlage zur Umsetzung von Computerspielen entwickelt wurde. Häufig eingesetzte Engines wie Unity, Unreal oder Cryengine stellen ein Gesamtprodukt dar, das u. a. physikalische Modelle, Schnittstellensysteme und einheitliche Programmierstrukturen in einem homogenen System vereint und gebündelt zur Verfügung stellt [CK17].

Zu Beginn werden bestehende Simulationsumgebungen und verwandte Arbeiten vorgestellt. In Abschnitt 3 erfolgt eine Definition der Funktionen, über die das Zielsystem verfügen soll. Abschnitt 4 befasst sich mit der Visualisierung von Geländemodellen auf Basis von drei unterschiedlichen Datengrundlagen. Nachdem anschließend auf die Umsetzung von Über- und Unterwasserfahrzeugen eingegangen wird, erfolgt in Abschnitt 6 eine Zusammenfassung mit Ausblick auf zukünftige Arbeiten.

¹ <http://eitams.de/>

2 Verwandte Arbeiten

Watanabe et al. entwickeln auf Basis des Roboter-Simulators Gazebo² und der Softwarebibliothek Rock³ eine Möglichkeit zur echtzeitbasierten Simulation und Visualisierung von AUVs. Sie ermöglichen dabei u. a. die Integration physikalischer Wassermodelle und eine Echtzeitvisualisierung auf Basis der Bibliothek OpenSceneGraph⁴. Innerhalb der physikalischen Modelle der Wasserbeschaffenheit wurden zunächst Strömung und Trägheit sowie der Auftrieb von Körpern modelliert. Die Anbindung von Sensoren, wie einer RGB-Kamera und einem Sonargerät, ist geplant. Die Arbeit ist ein Teil des FlatFish⁵-Projektes, das die Entwicklung eines autonomen Unterwasserfahrzeuges zum Ziel hat. Mit ihm soll die Inspektion von Unterwasserstrukturen im Öl- und Gas-Segment ermöglicht werden [WNC+15].

Prats et al. verwenden als Grundlage zur Visualisierung von Unterwassermissionen ebenfalls die OpenSceneGraph-Bibliothek. Mit dem quelloffenen UWSim⁶ stellen sie eine Möglichkeit zur Simulation und grafischen Darstellung von Unterwassermissionen bereit. Für die Steuerung virtueller Unterwasserfahrzeuge und die Verteilung simulierter Sensordaten erfolgt der Einsatz einer netzwerkbasierter Schnittstelle. Das Unterwassergeländemodell kann der Nutzer per XML-Datei konfigurieren. Es können Parameter wie z. B. Wasseroberfläche, Sicht unter Wasser und Partikel im Wasser definiert werden. Der Unterwasserroboter ist ebenfalls über eine XML-Datei anpassbar. Standardmäßig wird ein Unterwasserfahrzeug mit sechs Freiheitsgraden in der Bewegung bereitgestellt (6 DOF). In der Software stehen zudem einzeln konfigurierbare Sensoren zur Verfügung, die an die Fahrzeuge gebunden werden können. Es besteht die Möglichkeit mehrere Roboter innerhalb einer Simulation zu steuern. Durch die eingebundene Physik-Engine osgBullet besteht die Möglichkeit, stark vereinfachte, physikalische Aspekte in der Unterwasserwelt zu simulieren [PPF+12].

In dieser Arbeit soll eine Lösung der Problemstellung über eine einzelne Softwarekomponente erfolgen, die alle erforderlichen Aspekte integriert.

² <http://gazebosim.org/>

³ <http://rock-robotics.org>

⁴ <http://www.openscenegraph.org/>

⁵ <http://robotik.dfki-bremen.de/en/research/projects/flatfish.html>

⁶ <http://www.irs.uji.es/uwsim/>

3 Zielsetzung

Das Ziel dieser Arbeit umfasst den Entwurf und die Umsetzung eines Systems, das eine dreidimensionale Visualisierung von Unter- und Überwasserfahrzeugen ermöglicht. Die Evaluation der Steuerungsalgorithmik eines AUVs, ROVs oder USV anhand eines realen Terrains setzt eine möglichst realitätsnahe Modellierung des Geländes unter Wasser voraus. Neben Umgebungseinflüssen wie bspw. Strömungen und Auftrieb stellt dieses Gelände einen wichtigen Parameter für die Unter- und Überwassernavigation dar. Zur Berücksichtigung der Geländesituation in situ erfolgt die Integration virtueller Sensoren, die anhand ihrer modellierten Umgebung entsprechende Messdaten generieren und der Steuerungseinheit bereitstellen. Spezielle Missionen erfordern den Einsatz eines Schwarms von mehreren AUVs, ROVs und USVs. Daher ist das zu entwickelnde System auf die simultane Einbindung mehrerer Instanzen eines Fahrzeugtyps auszurichten. Jedes virtuell abgebildete Vehikel benötigt somit einen eigenen Kommunikationskanal zum Austausch von Steuerbefehlen und gemessenen Sensordaten. Um den Zugriff auf das Visualisierungssystem möglichst interoperabel zu gestalten, wird der Einsatz einer flexiblen und plattformunabhängigen Schnittstelle erforderlich.

In dieser Arbeit wird mit der Unreal-Engine eine der leistungsstärksten Game-Engines eingesetzt. Sie steht komplett quelloffen zur Verfügung und wird laufend weiterentwickelt. Somit können eigene Anpassungen vorgenommen werden. Die Entwicklung der Unreal-Engine⁷ erfolgt durch Epic Games, Inc.. Sie setzen die typischen Kernelemente einer Game-Engine (Sound-, Grafik-, Netzwerk- und Physik-Engine) mithilfe von C++ um. Mitgelieferte Editoren erlauben eine Anpassung von Spielabläufen, graphischen Oberflächen und Gestaltung der virtuellen Welt [CK17]. Für den räumlichen Bezug steht ein metrisches Koordinatensystem zur Verfügung, das als Grundlage zur Georeferenzierung verwendet werden kann.

⁷ <https://www.unrealengine.com/>

4 Visualisierung digitaler Geländemodelle

Die Positionierung von Über-, bzw. Unterwasserfahrzeugen innerhalb einer virtuellen 3D-Welt erfordert als Grundlage die Abbildung eines Geländes. Während ein digitales Geländemodell (DGM) innerhalb von Geoinformationssystemen oftmals durch ein Triangulated Irregular Network (TIN) repräsentiert und verarbeitet wird [BML15], erlaubt die Unreal-Engine lediglich eine Abbildung von Gelände durch Terrains und Meshes.

- **Meshes** stellen eine Standardmöglichkeit zur Integration von 3D-Elementen dar und können bspw. auf Basis von 3D-Datenformaten wie FBX oder 3DS erzeugt werden.
- **Terrain** benötigt die Eingabe eines Graustufenbilds und ist somit nur in der Lage eine 2,5-dimensionale Landschaft abzubilden. Das Gelände wird in Form einer Interpolation von Höhenwerten visuell aufbereitet.

Beide Möglichkeiten benötigen zur späteren Positionierung von weiteren Elementen eine Georeferenzierung des Modells in einem vereinheitlichten Koordinatensystem. Sowohl die automatische Interpolation von Höhenwerten als auch die Limitierung auf 2,5D schränken die realitätsnahe Abbildung eines DGMs stark ein. Markante Bodenprofile mit herausragenden Elementen sind nicht modellierbar. Daher erfolgt der Einsatz eines Meshes, für die als Grundlage sowohl die elektronische Seekarte als auch 3D-Punktwolken in verwendet werden kann.

4.1 Elektronische Seekarte als Basis

Die Electronic Navigational Chart (ENC) dient als digitale Karte und unter Verwendung des S-57 Standards zur computergestützten Navigation auf Schiffen [SLJ+16]. Eine manuelle Datenvorverarbeitung mithilfe der Software *3ds Max* erlaubt die Überführung der Tiefeninformationen einer ENC in das von Unreal akzeptierte 3D-Format FBX. Abbildung 1 zeigt das in Unreal überführte Geländemodell auf Basis von ENC-Tiefeninformationen. Dabei wird besonders die schichtweise Darstellung von Solltiefen deutlich, die zu scharf abfallenden Kanten im Übergang zweier Ebenen führen. Das Ziel, eine realitätsnahe Modellierung des Unterwassergeländes, kann mithilfe von ENC-Inhalten aufgrund der Klassifizierung von Höheninformation in Schichten nur eingeschränkt erreicht werden.



Abbildung 1: Visualisierung des Geländes in der Unreal-Engine auf Basis der elektronischen Seekarte

4.2 Punktwolke als Basis

Ein anderer Ansatz verfolgt die Abbildung des Geländes unter Wasser auf Basis von 2,5D-Punktwolken. Das Wasserstraßen- und Schifffahrtsamt Bremerhaven generiert im Rahmen seiner Seevermessung ein digitales Oberflächenmodell. Ein Teilbereich der mit einem Fächerecholot erzeugten Daten wurde dieser Ausarbeitung als regelmäßiges Raster mit einer Gitterweite von 0,5 m zur Verfügung gestellt. Die als ASCII-codierte Punktwolke verwendet als Referenzsystem UTM und bezieht sich auf Normalhöhennull. Ihre Vermaschung und Überführung in das 3D-Datenformat FBX erfolgt ebenfalls als manueller Vorverarbeitungsschritt mithilfe von *3ds Max*.

Das metrische Unreal-Koordinatenreferenzsystem erlaubt für die Georeferenzierung eine direkte Übernahme des Geländemodells ohne eine zusätzliche Transformation zwischen Quell- und Zielkoordinatensystem. Lediglich eine Verschiebung des digitalen Modells zum Koordinatenursprung kann eine Translation notwendig machen. Das Geländemodell auf Basis realer Punktwolkendaten muss in der ersten Implementierung der Simulationssoftware innerhalb eines Preprocessing-Schrittes in die Visualisierung eingefügt werden.

Abbildung 2 zeigt das in Unreal überführte Geländemodell auf Basis einer Punktwolke. Im Gegensatz zur abstrakten ENC ist das Bodenprofil des Geländes unter Wasser deutlich erkennbar und ermöglicht eine realitätsnahe Abbildung des Unterwassergeländes.



Abbildung 2: Visualisierung des Geländes auf Basis einer Punktwolke

4.3 Fiktive Daten als Basis

Zur Evaluation der Steuerungsalgorithmen ist auch eine Berücksichtigung von Extremsituationen erforderlich. Hierfür dient eine virtuelle Welt auf Basis von fiktiven Daten. Ihre Generierung erfordert die Eingabe von Parametern zu denen die Gebietsausdehnung in Form eines MUR⁸ gehören. Der Nutzer muss eine Ecke des Gebietes anhand von X- und Y-Koordinaten im UTM-Koordinatensystem mit zugehöriger UTM-Zone angeben. Die Ausdehnung des Gebietes wird auf $X + 1000$ m und $Y + 1000$ m festgesetzt. Das fiktive Gebiet umfasst somit 1 km^2 . Für das Unterwassergeländemodell ist zudem eine minimale Höhe, maximale Höhe, Höhe der Wassersäule (vom minimalen DGM-Wert bis zur Wasseroberfläche) einzugeben. Auf Basis dieser Parameter wird eine fiktive Unterwasserwelt mit der zugehörigen Wasseroberfläche generiert.

5 Implementierung von Über- und Unterwasserfahrzeugen

Die Modelle der Fahrzeuge wurden mit der Software Bentley Microstation sowie der Software 3dsMax von Autodesk erstellt. Für das Überwasserfahrzeug wurde ein Prototyp mit einfachen Geometrien erstellt. Das Unterwasserfahrzeug wurde auf Basis von Originalplänen des OpenROVs konstruiert. Beide Modelle wurden in ein geeignetes Dateiformat (FBX) für den Import in die Unreal-Engine exportiert.

⁸ MUR - Minimal umgebendes Rechteck

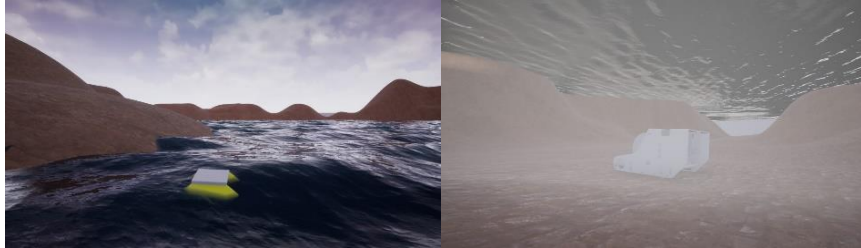


Abbildung 3: Fahrzeuge in der Simulation; Links: Überwasserfahrzeug, Rechts: Unterwasserfahrzeug

Die Positionierung der Fahrzeuge wird durch drei verschiedene Ansteuerungsmöglichkeiten erlaubt. Überwasserfahrzeuge können durch GPS-Koordinaten und Beschleunigungsvektoren gesteuert werden. Des Weiteren können einfache Steuerbefehle (vorwärts-, rückwärts- und seitwärtsfahren) an das virtuelle Fahrzeug übergeben werden. Für die Orientierung der Fahrzeuge ist für beide Positionierungsarten zudem die Angabe der Werte von Roll, Pitch und Heading implementiert. Somit können die Fahrzeuge über sechs Freiheitsgrade gesteuert werden. Die Positionierung der Unterwasserfahrzeuge wurde zunächst auf die Angabe von Beschleunigungsdaten sowie auf die Angabe der einfachen Steuerbefehle beschränkt, da die Auswertung von GPS-Koordinaten unter Wasser nicht möglich ist. Die einfachen Steuerbefehle wurden für die Unterwasserfahrzeuge um das Auf- und Abtauchen erweitert. Die Orientierung der Unterwasserfahrzeuge ist wiederum über die Angabe der Roll-, Pitch- und Headingwerte möglich.

Die Unreal-Engine verfügt über Funktionen, die zur Abbildung virtueller Sensorik verwendet werden können. Bspw. lässt sich auf Basis von Line Tracing der virtuelle Abstand zwischen Unterwasserfahrzeug und Gelände bestimmen und somit ein Echolot simulieren. Der Austausch virtueller Sensormesswerte und Steuerbefehlen zwischen Engine und Steuerungseinheit geschieht mithilfe einer netzwerkbasierter Schnittstelle. Jedes Fahrzeug verfügt über eine separate Schnittstelle. Somit können mehrere Fahrzeuge parallel in der gleichen Simulationsumgebung mit unterschiedlicher Steuerungssoftware erprobt werden.

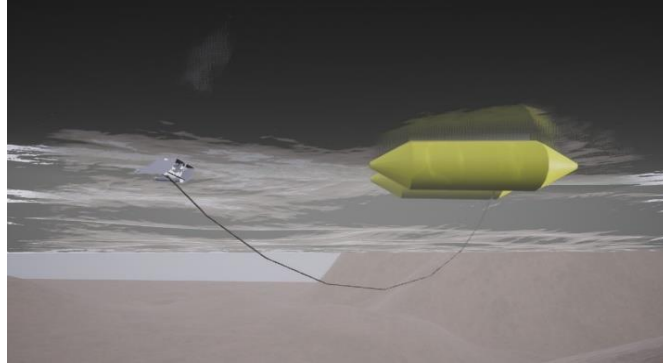


Abbildung 4: Über- und Unterwasserfahrzeug mit Kabelverbindung

6 Zusammenfassung und Ausblick

Während die Simulation von AUV-Missionen typischerweise eine Integration vieler verschiedener Softwarekomponenten benötigt, bündeln Game-Engines die Grundfunktionen zur Durchführung einer Simulation in einem Gesamtprodukt. Speziell die Unreal-Engine ermöglicht eine Visualisierung und Steuerung von Über- und Unterwasserfahrzeugen in einer virtuellen Welt. Als Geländemodell unter Wasser kann entweder die elektronische Seekarte, eine Punktwolke oder eine fiktive Landschaft verwendet werden. Neben einer realen Umgebung können somit auch Extremsituationen produziert werden, auf die eine Steuerungseinheit reagieren können muss.

Die Visualisierung von Über- und Unterwasserfahrzeugen mithilfe der Unreal-Engine stellt einen ersten Schritt in Richtung Simulation dar. Mit Bezug auf das Projekt EITAMS untersuchen zukünftige Arbeiten die Fähigkeiten der integrierten Physik-Engine Unreals zur Modellierung von Einflussfaktoren unter Wasser, die zur Evaluation von Steuerungsalgorithmen benötigt werden. Insbesondere soll die Integration von Strömungsmodellen, Trägheits- und Auftriebsfaktoren untersucht werden. Die Abbildung weiterer Sensorik, die Einbindung einer simulierten akustischen Unterwasserkommunikation und die Positionierung über akustische Modems von Unterwasserfahrzeugen stellen die Hauptaspekte von weiterführenden Arbeiten dar. Der Datenvorverarbeitungsschritt zur Integration digitaler Geländemodelle erfolgt derzeit noch manuell. Dieser Prozessschritt ist für eine direkte Einbindung von Geländemodellen zu automatisieren.

Acknowledgement

Diese Arbeit entstand im Zusammenhang mit dem Projekt EITAMS, das von der „Niedersächsischen Vorab“ Stiftung gefördert wird. Des Weiteren wird das Projekt von internen Forschungsmitteln der Jade Hochschule finanziert. Wir danken dem „Wasserstraßen- und Schifffahrtsamt Bremerhaven“ für die Bereitstellung der digitalen Geländedaten.

Literatur

- [BML15] Burrough, Peter A.; MacDonnell, Rachael A.; Lloyd, Christopher D.: Principles of geographical information systems. Oxford Univ. Press., Oxford, 2015; S. 231 - 248.
- [CK17] Cowan, B.; Kapralos, B.: An overview of serious game engines and frameworks. In: Intelligent Systems Reference Library 119. Springer International Publishing, Cham, 2017, S. 15–38.
- [FNR+15] Fernandes, Victor Hugo; Neto, Arthur Ayres; Rodrigues, Dalto Domingues: Pipeline inspection with AUV. In: 2015 IEEE/OES Acoustics in Underwater Geosciences Symposium (RIO Acoustics). Rio de Janeiro, 2015; S. 1–5.
- [LLS17] Lu, Huimin; Li, Yujie; Serikawa, Seiichi (2017): Computer Vision for Ocean Observing. In (Huimin Lu und Yujie Li): Artificial Intelligence and Computer Vision. Springer International Publishing, Cham, 2017; S. 1–16.
- [MA16] Matej, Jaroslav: Virtual reality and vehicle dynamics in unreal engine environment. In: Modern Machinery Science Journal, Volume 4. Prag, 2016; S. 1141–1144.
- [PPF+12] Prats, Mario; Perez, Javier; Fernandez, J. Javier; Sanz, Pedro J.: An open source tool for simulation and supervision of underwater intervention missions. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. Vilamoura-Algarve, Portugal, 2012; S. 2577–2582.
- [SLJ+16] Song, Moonsub; Lee, Kwangil; Jang, Byungtae; Ro, Soonghwan: A Design and Implementation of Lightweight ENC for Android App. In: Advanced Multimedia and Ubiquitous Engineering, Bd. 354. Heidelberg, 2016; S. 137–143.
- [WNC+15] Watanabe, Thomio; Neves, Gustavo; Cerqueira, Romulo; Trocoli, Tiago; Reis, Marco; Joyeux, Sylvain; Albiez: The Rock-Gazebo Integration and a Real-Time AUV Simulation. In: 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR). Uberlandia, 2015; S. 132–138.