
Nutzung von Unterwassergeländemodellen in virtuellen 3D-Spielewelten

Bernhard Vockner, Manfred Mittlböck und Michael Andorfer

RSA iSPACE, Salzburg · bernhard.vockner@researchstudio.at

Short paper

Zusammenfassung

Der Beitrag fokussiert sich auf die Entwicklung einer nachhaltigen Workflowstrategie zur Integration und Nutzung von Unterwassergeländemodellen in virtuellen 3D-Spielewelten. Ausgehend von der Aufnahme mittels Sonar bzw. Laserscan wird ein Transformationsprozess vorgestellt, mit dem Unterwassergeländemodelle (semi-)automatisiert in 3D-Umgebungen in Spiele-Engines integriert werden können. In einem Nachbearbeitungsschritt werden die Unterwassergeländemodelle mit Texturen sowie Orten von Interesse (POIs) versehen, damit z. B. Tauchgänge vorab virtuell simuliert werden können.

1 Einleitung

Die Computerspielindustrie setzt gerade in den letzten Jahren verstärkt auf den Einsatz von Realumgebungen anstelle von rein virtuellen. Der Realitätsgrad ist dabei in vielen Fällen bereits sehr nahe an der Wirklichkeit dran (THE TELEGRAPH 2015). Neben dem spielerischen Charakter wird durch den Einsatz von Echtwelten vor allem der Trainingseffekt bei Sport- und Freizeitaktivitäten erhöht. Ähnlich wie Formel-1-Fahrer aufgrund des strengen Regelwerks nur wenige Testfahrten auf Rennstrecken unternehmen dürfen (FIA 2015) und dementsprechend auf äußerst exakte Simulatoren angewiesen sind, lassen sich auch Tauchgänge vorab in Form von Unterwasserspielewelten simulieren. Das hat für Taucher mitunter den Vorteil, dass die Welt, die sie unter Wasser erwartet, ähnlich den Kurven einer Rennstrecke vom Taucher bereits im Vorfeld verinnerlicht werden kann. Des Weiteren können Taucher vorab auf Besonderheiten – z. B. wo sich bestimmte Tierarten besonders gerne aufhalten, die maximale Tauchtiefe, Strömungen etc. – hingewiesen werden.

2 Primärdatenaufnahme

Im Gegensatz zur Aufnahme der Erdoberfläche können Unterwassergeländedaten nur bedingt mit klassischen Methoden der Fernerkundung erhoben werden (PALMER et al. 2015). Gebräuchlich sind in diesem Kontext vor allem Laserscan- und Sonaraufnahmen. Der Beitrag konzentriert sich vorrangig auf Sonardaten, lässt sich aber genauso auf Laserscandaten übertragen.

Die Sonardaten als Basis für die Erstellung von 3D-Unterwasserwelten werden mit Booten in Form von mäanderartigen Schleifen (ähnlich wie bei Luftbildflügen) erhoben (Abb. 1). Bei der Aufnahme ist vorrangig darauf zu achten, dass diese mit einer möglichst konstanten Geschwindigkeit erfolgt, weite Kurvenradien sofern möglich einzuhalten sind sowie die doppelte Aufnahme von Gebieten zu vermeiden, da dies das Interpolationsergebnis deutlich verschlechtert. Es ist jedoch anzumerken, dass das tatsächliche Aufnahmeergebnis auch von Parametern wie z. B. Seegang und Zugänglichkeit des Gebietes mit Boot sowie (rechtlich, organisatorisch) eingeschränkt freie Wahl der Fahrtstrecke. Zusätzlich zu den Sonardaten wird Bildmaterial von Riffen in Form von Video- und Fotoaufnahmen erhoben, um das Gelände mit Texturen zu versehen und feiner modellieren zu können.

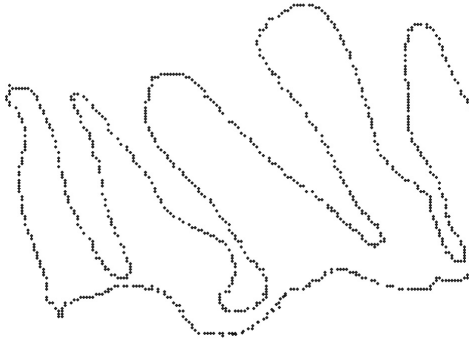


Abb. 1: Aufnahmepunkte von Unterwassergeländedaten mittels Sonar (Ausschnitt eines GPS-Tracks) (Quelle: eigene Darstellung 2015)

3 Workflow zur Erstellung des Unterwassergeländemodells

Zur Erstellung des Basismodells der Unterwasserwelt kommen primär Open-Source-Produkte zum Einsatz. Diese umfassen PYTHON (2015), GDAL (2015) und POSTGRESQL/POSTGIS (2015). Im Rahmen des ETL-Prozesses (ETL = Extract, Transform, Load) kommt auch FME Desktop von SAFE SOFTWARE (2015) für Transfer- und Schnittstellenfunktionalitäten zum Einsatz. Diese Software bietet durch ihren modulhaften Aufbau zugleich die Möglichkeit, einzelne Parameter (z. B. Größe des bei der Interpolation zu verwendenden Buffers) rasch und einfach ersetzen zu können.

Die Ausgangsdatensätze bestehen aus Sonardaten, welche im .son-Format vorgehalten sind. Aus diesen Sonardaten werden Python-gestützt die x-,y- und z-Koordinaten extrahiert sowie in csv-Files als auch als Punkt-Shapefiles abgespeichert (Abb. 2). Als Vorbereitungsschritt für die Interpolation werden die WGS84-Koordinaten (EPSG: 4326) in ein metrisches Koordinatensystem überführt, das die Anforderung erfüllt, weltweit gut passende und einfach zu integrierende Ergebnisse zu liefern (im vorliegenden Fall: Web Mercator; EPSG: 3857). Des Weiteren wird ein durch den Nutzer parametrisierbarer Buffer um die Aufnahmepunkte festgelegt, damit die Interpolation innerhalb einer Maske durchgeführt werden kann und sich an den Außenbereichen weniger Verzerrungen ergeben.

Die eigentliche Interpolation erfolgt mit IDW (Inverse Distanzgewichtung; SHEPARD 1968) mit der Open Source Software GDAL. Diese externe Komponente wird in den FME ETL-Prozess mit einem Python-Skript am Ende des Workflows (Abb. 2) ausgeführt. Die Python-Skripts lassen sich direkt aus FME Desktop durch sogenannte ‚Startup-‘ bzw. ‚Shutdown Python Scripts‘ aufrufen, wodurch kein manueller Zwischenschritt im Arbeitsablauf erforderlich ist. Die einzigen manuell durchzuführenden Schritte sind die Parametrisierungen von Buffergröße und Feinjustierungen der Interpolationsparameter (z. B. Setzen von Suchradien, Erhöhung/Vermindern des Power-Faktors).

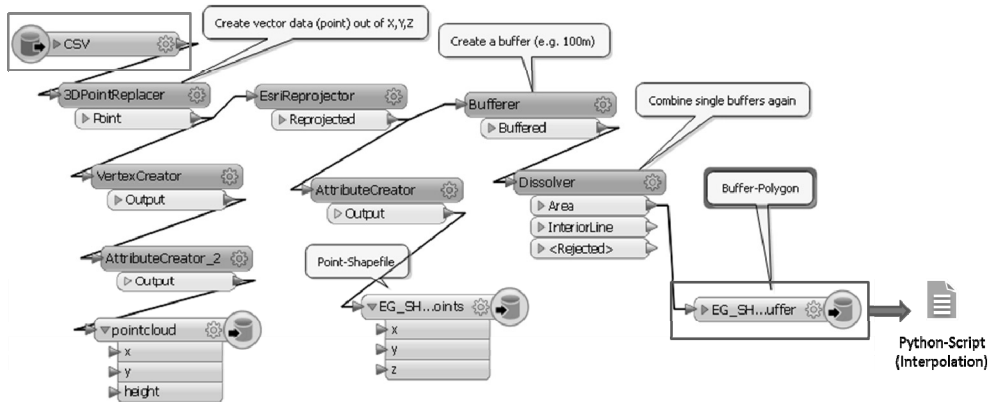


Abb. 2: Transformationsworkflow von Sonardaten (Teil 1 – vereinfacht)
(Quelle: eigene Darstellung 2015)

Die abschließende Triangulierung („TINGenerator“) wird wiederum mit FME Transformatoren durchgeführt (Abb. 3), da sich TINs besser für die Verwendung in 3D-Softwareprogrammen eignen. Das Ergebnis wird in unterschiedlichen Formaten abgespeichert. Diese umfassen u. a. 3ds, DWG, SHP sowie PDF. Zu Archivierungszwecken werden die Ergebnisse sowie Zwischenergebnisse allesamt zusätzlich in einem Datenbanksystem (PostgreSQL/PostGIS) vorgehalten.

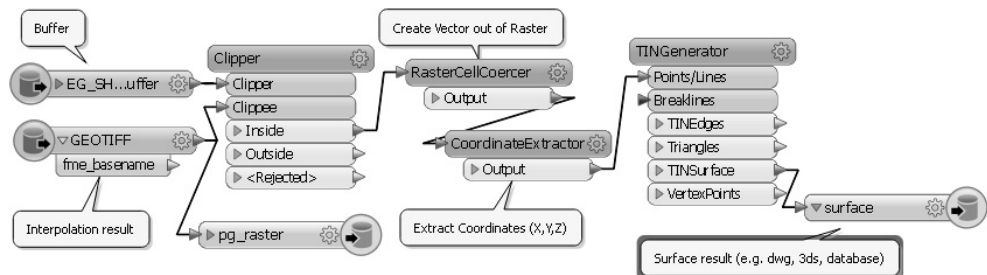


Abb. 3: Transformationsworkflow von Sonardaten (Teil 2 – vereinfacht)
(Quelle: eigene Darstellung 2015)

4 Resultate

Die Ergebnisse des Transformationsworkflows sind TINs. Im Vergleich zu anderen Ausgabedaten lassen sich TINs unkompliziert und exakt mit 3D-Software in 3D-Meshes konvertieren. Die Interpolationsergebnisse selbst weisen einen hohen Detailgrad auf, wobei die Aufnahmestrecke (Tracks) durch Verwendung der Interpolationsmethode IDW bewusst herausgeglättet wurde. Es werden automatisiert mehrere Interpolationen mit leicht nuancierten Parametrisierungen durchgeführt, um mit einfacher visueller Kontrolle das am besten für die Weiterverarbeitung geeignete Modell zu identifizieren. Das Interpolationsergebnis weist einen sehr hohen Detailgrad auf, um der Anforderung zu genügen, markante Punkte herauszustreichen, da die anschließende Texturierung (siehe Abschnitt 5) dadurch leichter von der Hand geht.

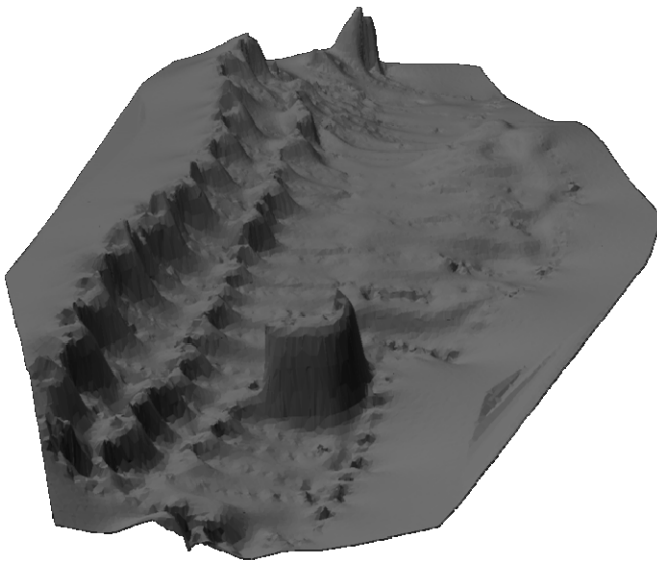


Abb. 4: Exemplarisches Ergebnis des Transformationsworkflows: TIN mit 0,02 m Auflösung (Quelle: eigene Darstellung 2015)

Eine Validierung der Ergebnisse erfolgt durch ‚Ground Truthing‘, also mit Tauchexperten vor Ort, die diese Interpolationen bewerten. Im Anschluss werden die Ergebnisse des Transformationsworkflows an ein 3D-Grafikprogramm zur weiteren Nachbearbeitung und Texturierung übergeben.

5 Nachbearbeitung und Texturierung

Die Outputs werden in ein 3D-Grafikprogramm (z. B. Maya von AUTODESK 2015) übertragen und dort mit Texturen und feineren Details, die im Zuge der räumlichen Interpolation eventuell zu stark weggeglättet wurden, ergänzt. Des Weiteren werden Orten von besonde-

rem Interesse (POIs; z. B. Aufenthaltsort bestimmter sehenswerter Fischarten) und Vegetation als 3D-Objekte ergänzt.

Mithilfe von Laufzeit- und Entwicklungsumgebungen für Computerspiele wie z. B. Unity 3D von UNITY TECHNOLOGIES (2015) kann das Unterwassermodell im virtuellen Tauchgang auf Desktop-PCs, Tablets und Smartphones eingesetzt werden (Abb. 3).

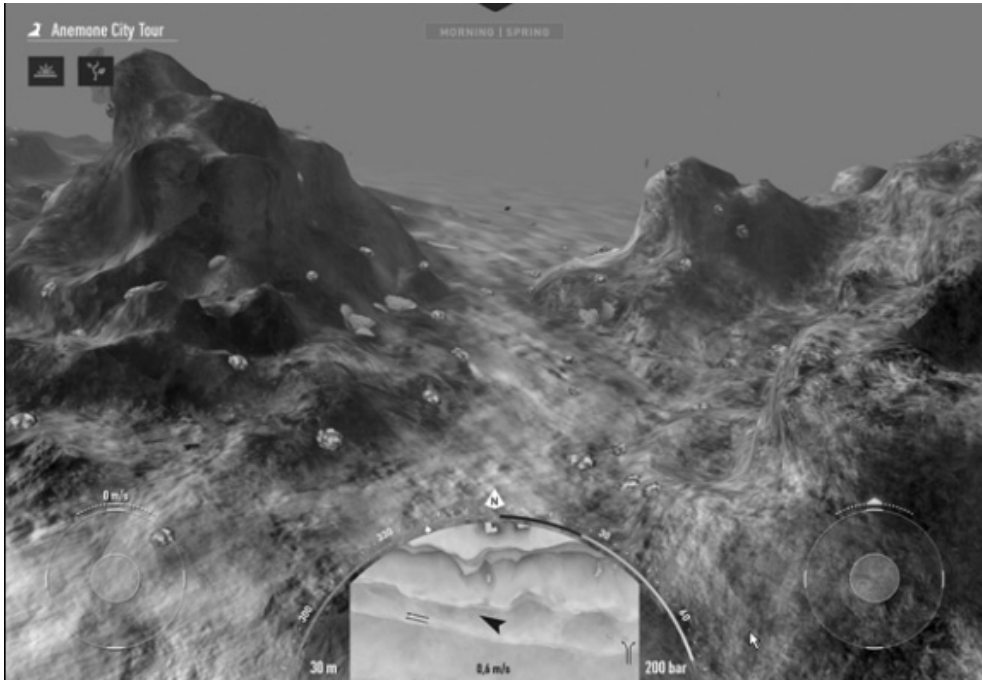


Abb. 5: Texturiertes Unterwassergeländemodell im virtuellen Tauchgang
(Quelle: <http://www.reef-interactive.com>, 2015)

6 Fazit

In diesem Beitrag wurde die Entwicklung eines nachhaltigen Transformationsworkflows zur Nutzung von Unterwassergeländemodellen in virtuellen 3D-Spielwelten aufgezeigt. Dieser Arbeitsablauf ermöglicht es, dass Unterwassergeländemodelle (semi-)automatisiert in 3D-Umgebungen in Laufzeit- und Entwicklungsumgebungen in 3D-Spielen integriert werden können, ohne dass die Bearbeiter/Designer spezielle Geoinformatik-Kenntnisse benötigen. Die Ergebnisse wurden anhand von ‚Ground Truthing‘ Methoden von Tauchexperten an den jeweiligen Riffen validiert.

Danksagung

Die Entwicklung des angesprochenen Transformationsworkflows erfolgte in enger Zusammenarbeit mit Reef Interactive (<http://www.reef-interactive.com>).

Literatur

- AUTODESK (2015), Maya. Umfassende Software für 3D-Animation.
<http://www.autodesk.de/products/maya/overview> (10.04.2015).
- FIA (2015), 2015 Formula One Sporting Regulations.
<http://www.fia.com/sites/default/files/regulation/file/1-2015%20SPORTING%20REGULATIONS%202014-06-29.doc> (01.02.2015).
- GDAL (2015), GDAL – Geospatial Data Abstraction Library. <http://www.gdal.org/> (10.04.2015).
- PALMER, S., KUTSER, T. & HUNTER, P. (2015), Remote sensing of inland waters: Challenges, progress and future directions. *Remote Sensing of Environment*, 157, 1-8.
- POSTGIS (2015), About PostGIS. <http://postgis.net/> (10.04.2015).
- POSTGRESQL (2015), About PostgreSQL. <http://www.postgresql.org/about/> (10.04.2015).
- PYTHON (2015), About Python. <https://www.python.org/about/> (10.04.2015).
- SAFE SOFTWARE (2015), FME Desktop. <http://www.safe.com/fme/fme-desktop/> (10.04.2015).
- SHEPARD, D. (1968), A two-dimensional interpolation function for irregularly-spaced data. In: *Proceedings of the 1968 23rd ACM national conference (ACM '68)*. ACM, New York, NY, USA, 517-524.
- THE TELEGRAPH (2015), 15 best real world locations in video games.
<http://www.telegraph.co.uk/technology/video-games/Picture-galleries/8757867/15-best-real-world-locations-in-video-games.html> (08.04.2015).
- UNITY TECHNOLOGIES (2015), Unity 3D Editor: A feature-rich and highly flexible editor.
<http://unity3d.com/unity/editor> (10.04.2015).