Fraunhofer
IGD

FRAUNHOFER-INSTITUT FÜR GRAPHISCHE DATENVERARBEITUNG IGD

VISUAL ENGINEERING AS A SERVICE

Abschlussbericht

MV tut gut.

EUROPÄISCHE UNION
Europäischer Fonds für
regionale Entwicklung

IGD_Berichte_de_A4_Vorlage.dotm | 1.0 hd 14.03.2012

ABSCHLUSSBERICHT (IGD)

Visual Engineering-as-a-Service (VEaaS)

Matthias Vahl und Tom Krause

Fraunhofer-Institut für Graphische Datenverarbeitung IGD Rostock

Rostock –24.10.2023

Version: 1.00

Projektnummern (intern): 10-13530-2780-00002, 10-13530-2780-00003

Bearbeitungsnummer: TBI-V-1-421-VBW-143

Projektpartner: SKM Informatik GmbH

IGD_Berichte_de_A4_Vorlage.dotm | 1.0 hd 14.03.2012

Inhalt

1 Ziel der Projektarbeiten 2
2 Beschreibung der durchgeführten Arbeiten (Arbeitspakete) 3
2.1 Anforderungsanalyse und Spezifikation von Mockups und Demonstratoren 3
2.2 Architektur und Infrastruktur 4
2.3 Point-Cloud-Viewer als Alternative zum instant3dhub 5
2.3.1 Arbeiten am Point-Cloud-Viewer 5
2.4 App zum Use-Case „Datenbankabfrage mit AR-/VR-fähigem Gerät“ 7
2.5 Services 8
2.5.1 Konvertierung von E57 Punktwolken 8
2.5.2 Erkennung von einfachen Geometrischen Objekten in Punktwolken (KI basiert) 8
2.5.2.1 Ergebnisse 11
2.5.2.2 Literatur 12
2.5.3 Erkennen von Zylindern in Punktwolken (klassische Verfahren) 13

3 Zusammenfassung 16

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

1
Ziel der Projektarbeiten

Im Mittelpunkt des Verbundprojekts stand die Entwicklung der cloudfähigen Dienst-Plattform »Visual Engineering-as-a-Service«. Das Konzept basiert auf der im Fraunhofer IGD entwickelten Technologie für Visual Computing-as-a-Service und knüpft an die bei der SKM Informatik GmbH vorhandenen Kompetenzen und Software-Bausteine an. Wichtige Merkmale des Architekturkonzepts sind die Entkopplung von Frontend und dynamisch in der Cloud orchestrierte Services zum Arbeiten auf den Engineering-Daten sowie einer bestehenden Datenhaltung. Auf diese Weise lassen sich sowohl neue Software-Module online vermarkten als auch kundenspezifische Lösungen mit einem hohen Grad an Wiederverwendung realisieren.

Im Zentrum aller Arbeiten standen Laserscan- und CAD-Daten. Auf diesen Daten setzte zum einen die SKM Informatik GmbH durch eigene Tools (z.B. TT-PLANT) und zum anderen der instant3Dhub auf. Während der instant3Dhub als reine Viewer-App auf anzuzeigende Daten nur lesend Zugriff hat, hat die Anwendung der SKM Informatik GmbH Vollzugriff und somit auch die Möglichkeit Daten zu manipulieren.
Die Arbeiten des Fraunhofer IGD mit dem Teilvorhaben der »Maritimen Toolbox« setzen an den Schnittstellen des instant3Dhub an, auf deren Basis eigene Services und Apps zur Datenbetrachtung und Manipulation entwickelt wurden. Daten-Manipulationen können dabei nicht direkt über den instant3Dhub zurückgeführt, sondern müssen über eine definierte Schnittstelle an die Anwendung der SKM Informatik GmbH übertragen werden. Die Anwendung der SKM Informatik GmbH wiederum überführt die Anfragen zur Daten-Manipulation in ihre eigenen Strukturen und setzt diese dann entsprechend um.

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

2 Beschreibung der durchgeführten Arbeiten (Arbeitspakete)

2.1 Anforderungsanalyse und Spezifikation von Mockups und Demonstratoren

Im ersten Schritt wurde die Anwendungsdomäne genauer analysiert. Eine Zielsetzung, die daraus entstanden ist, ist dass die entstandenen Services und Apps möglichst domänenunabhängig gestaltet werden. Daher wurden für das Vorhaben Use-Cases aus verschiedenen Anwendungsdomänen ausgewählt.

Bei der Analyse der Anwendungsdomänen wurde zusammen mit dem Projektpartner SKM Informatik eine Liste allgemeiner Anforderungen erarbeitet. Diese umfasst u.a. die folgenden Fragestellungen:

- Welche Probleme gibt es bei aktueller Software/Technik?
- Was sind die häufigsten Anfragen von Kunden?
- Was sind die häufigsten Anforderungen von Kunden?
- Wo wollen wir noch Kompetenz aufbauen?
- Was sind gängige Datenformate, die wir unterstützen wollen?

Auf Basis der gesammelten Anforderungen wurden zwei Services bzw. Anwendungen abgeleitet, die wiederum in verschiedenen Use-Cases einsetzbar sind:

Service: Punktwolken Analyse

Dieser Service stellt die Grundlage für verschiedene Use-Cases dar. Im Bereich Architektur sowie Schiff- und Anlagenbau erfolgt die Planung i. d. R. durch CAD-Modellen. Die Erfassung des Ist-Zustandes erfolgt häufig mittels Laserscan. Die hieraus gewonnen Daten lassen sich nur durch hohen Aufwand in weiterverwendbare CAD-Daten konvertieren bspw. zum Vergleich mit Planungsdaten.
Der Punktwolken-Analyse-Service soll einen Laserscan-Datensatz und eine Liste möglicher Bauteile (inklusive der CAD-Modelle) entgegennehmen, um daraus automatisiert den Ist-Zustand als CAD-Modell abzuleiten.
Use-Cases sind z.B.:

- die Planung der Nachrüstung von Ballastwassertanks auf Schiffen (Retrofit)
- der Soll-Ist-Vergleich im Bereich Architektur und Anlagenbau
- die Erfassung des Ist-Zustands für Assistenzsysteme (z.B. für Wartungsaufgaben)

Anwendung: Datenbankabfrage mit AR-/VR-fähigem Gerät

In modernen Produktionsbetrieben oder chemischen Großanlagen sind viele Geräte mit Sensorik ausgestattet, deren Daten üblicherweise kontinuierlich in einer Kundendatenbank abgelegt werden. Die Idee ist, an relevanten Punkten optische Marker zu hinterlegen, die mit einem AR- oder VR- fähigem Gerät erfasst werden können. Über den Marker identifizierte Einträge aus der Kundendatenbank sollen abgerufen und dynamisch visualisiert werden können. Eine wichtige Anforderung besteht darin, dass die beim Kunden existierende Datenbankstruktur nicht verändert werden darf. Vielmehr soll die Anwendung so gestaltet werden, dass die Adaption an

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

die Kundendatenbank durch das Hinterlegen einer Konfigurationsdatei geschehen kann. Hierdurch wird die Anwendung flexibel einsetzbar.

Durch das Einblenden relevanter Zusatzinformationen wären zugehörige Use-Cases bspw.:

- Assistenzsystem für Wartung- und Inspektion
- Assistenzsystem zur Bedienung der Geräte
- Assistenzsystem zur Navigation in Schiffen und großen Anlagen (Der Anwender möchte eine bestimmte Maschine aufsuchen.)

2.2 Architektur und Infrastruktur

Die Architektur der VEaaS-Cloud wurde so gestaltet, dass die einzelnen Komponenten unabhängig voneinander auf verschiedenen Systemen lauffähig sind (siehe Abbildung 1). Die Kommunikation der Komponenten untereinander geschieht dabei über eine REST-Schnittstelle. Das System ist beliebig skalierbar und somit auch für größere Infrastruktur geeignet.

Um die Architektur in kleinerem Maßstab zu testen, wurde die Architektur prototypisch wie folgt aufgesetzt:

- Im Netzt des Fraunhofer IGD läuft:
    - ein Server auf dem der instant3dhub läuft
    - auf dem gleichen PC läuft ein Dockerimage mit dem Main-Service, den Datenbanken, den Service- und den Dataprovidern
    - separat läuft eine VM mit einem Point-Cloud-Viewer
- Für SKM wurde eine Freigabe eingerichtet, die es ermöglicht sowohl auf die API des Instant3dhub als auch auf die API des Main-Service zuzugreifen und dort verschiedene Aktionen zu triggern

Per API können so Daten hochgeladen oder gelöscht werden. Es ist möglich verschiedene Services auf den Daten anzuwenden. Die Ergebnisse werden dann mit den ursprünglichen Datensatz verknüpft und können je nach Service auch als separater Datensatz betrachtet werden. Sofern die Daten und Ergebnisse der Services kompatibel sind, können diese an einen der Provider-Dienste übergeben und so entweder im instant3dhub oder im Point-Cloud-Viewer angezeigt werden. Die Viewer-Komponente des instant3Dhub lässt sich ebenfalls per API ansprechen und somit auch in eigene Anwendungen integrieren.

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

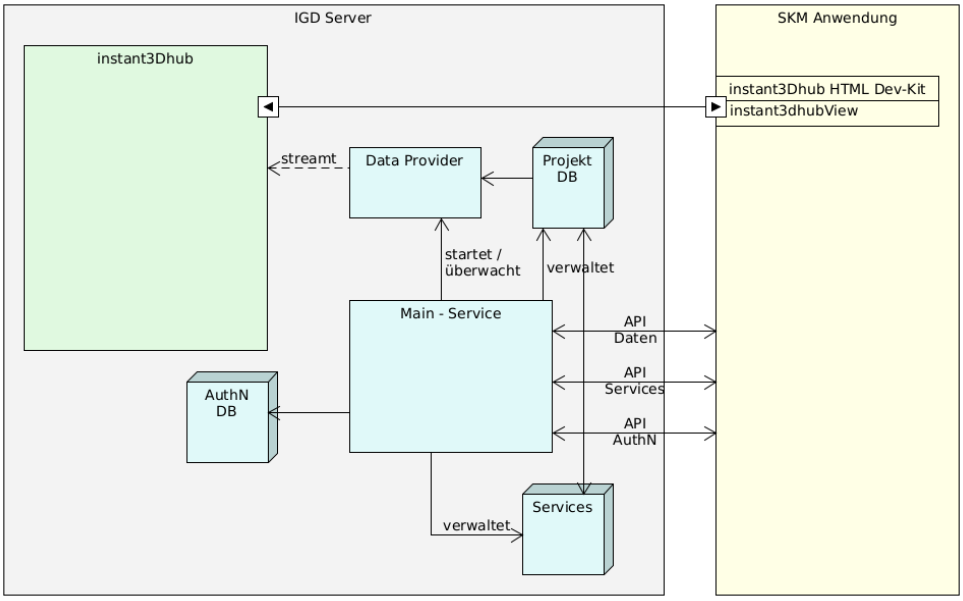

Abbildung 1- Architekturbild VEaaS

2.3 Point-Cloud-Viewer als Alternative zum instant3dhub

Während der Projektlaufzeit gab es einige Bedenken bezüglich des instant3Dhub, da dieser nicht alle Kernfeatures (insbesondere die Darstellung von Punkwolken/Laserscandaten) unterstützt. Zudem gab es zwischenzeitlich ein Problem mit der Lizensierung durch threedy. Daher wurde entschieden, ein alternative Anwendung in Form eines Point-Cloud-Viewers aufzusetzen, die am Ende auch ohne den instant3dhub auskommt und den Anwendungsfall vollständig abdeckt.

2.3.1 Arbeiten am Point-Cloud-Viewer

Der Point-Cloud-Viewer ist eine Webanwendung zur Darstellung von Punktwolken. Sie ist in der Lage Ergebnisse von Laserscans zu verarbeiten und ermöglicht die Darstellung der Punktewolke inklusive der Farbwerte des Scans. Die Daten werden auf dem Server analysiert und passende Bauteile identifiziert, die im Point-Cloud-Viewer zusätzlich zur Punktwolke angezeigt werden können.

Im Rahmen des Teilprojekts *Industrielle Forschung* wurde die Anwendung zur Visualisierung von Punktwolken konzipiert. Es wurde entschieden, nicht nur eine Webanwendung zu erstellen, sondern gleichzeitig eine PC-basierte Anwendung bereitzustellen. Dies war durch die Verwendung der Entwicklungsumgebung Unity 3D möglich, da dort auf Basis des gleichen Quellcodes gleichzeitig sowohl die Webanwendung als auch die PC-Anwendung erstellt werden kann.

Um der Anwendung die sehr großen Datenmengen zur Verfügung zu stellen, wurde außerdem ein Konzept für einen Bereitstellungsservice von Punktwolken-Daten ausgearbeitet.

Im Rahmen des Teilprojekts *Experimentelle Entwicklung* erfolgte die Umsetzung der beiden Konzepte für die Visualisierungsanwendung und den Bereitstellungsservice.

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

Während verschiedene Varianten der Umsetzung der Visualisierung evaluiert wurden (Verwendung von Bibliotheken, Plugins oder eigene Entwicklung), wurde parallel dazu bereits eine multiplattformfähige ShaderFactory entworfen und entwickelt, um die Unterschiede von WebGL und Direct3D im Shader Handling berücksichtigen zu können. Verschiedene Tests zeigten die Brauchbarkeit dieses Ansatzes. Die ShaderFactory wurde im Verlauf des Projektes aufgrund neuer Anforderungen an einigen Stellen angepasst.

Auch für die Umsetzung des Bereitstellungsservice wurden unterschiedliche Varianten evaluiert. Dazu wurden verschiedene Ansätze zum Download von Punktwolken entwickelt, implementiert und evaluiert. Schließlich wurde ein Streaming der Daten als beste Variante ausgewählt und weiter getestet. Hierzu wurden verschiedene Datensets generiert und mehrfach erweitert, mit denen die Downloads getestet wurden. Wichtig war dabei vor allem, dass die Downloads regelmäßig unterbrochen werden konnten, damit stets genügend Verarbeitungszeit für die Interaktivität mit der Anwendung vorhanden war. Durch regelmäßige Tests konnte dies auch beim Download großer Datensätze erreicht werden.

Zeitgleich wurden erste Entwürfe des Userinterface entwickelt und umgesetzt sowie die Steuerung des Bereitstellungsservice in das Userinterface integriert. Das Userinterface verhalten wurde nach und nach an das Streamingverhalten angepasst und die Performance und die Interaktivität insbesondere für größere Datensätze getestet.

Um beim Laden auf die Struktur der Datensätze eingehen zu können, wurde der Bereitstellungsservice erneut angepasst. Die Gesamtdaten konnten nun in einzelne Teile in separate Dateien aufgeteilt werden. Mit Hilfe von Metadateien, die durch den Bereitstellungsservice zur Verfügung gestellt werde, konnten verschiedene Teile der Punktwolke bevorzugt geladen werden. Die vorhandenen Testdaten wurden angepasst und die Umsetzung des neuen Service erfolgreich evaluiert.

Um den Bereitstellungsservice serverseitig an unterschiedliche Server anpassen zu können, wurde ein Konfigurationsdatei erstellt, die es den Betreibern eines solchen Service ermöglicht, die Bandbreite, Größe der Teildateien etc. zu konfigurieren. Das Lesen dieser Testdatei musste für die Webanwendung angepasst werden, da dort besondere Rechte für das Lesen notwendig waren.

Die Umsetzung des Userinterface erforderte schließlich noch eine Anpassung, um eine einheitliche Darstellung in der Webanwendung und der PC-Anwendung zu erreichen. Die Besonderheiten der Visualisierung von Punktwolken (freie Rotation, Streaming der Daten) erforderte eine Anpassung der Standard-Kamerasteuerung, um Probleme mit dem Gimbal Lock zu beseitigen. Beides konnte erfolgreich umgesetzt werden.

Zum Ende des Projektes wurde die Bereitstellung der Daten auf dem Anwendungsserver und die notwendigen Schnittstellen für den Bereitstellungsservice diskutiert und beidseitig umgesetzt. Das dies erst am Ende des Projektes notwendig war, lag an der Umsetzung, die bereits im Projektverlauf gegen eine Schnittstelle implementierte. Diese musste nach der gemeinsamen Festlegung nur geringfügig angepasst werden.

Schließlich wurde der WebBuild von Unity auf den zentralen Anwendungsserver übertragen und erneut mehrfach getestet.

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

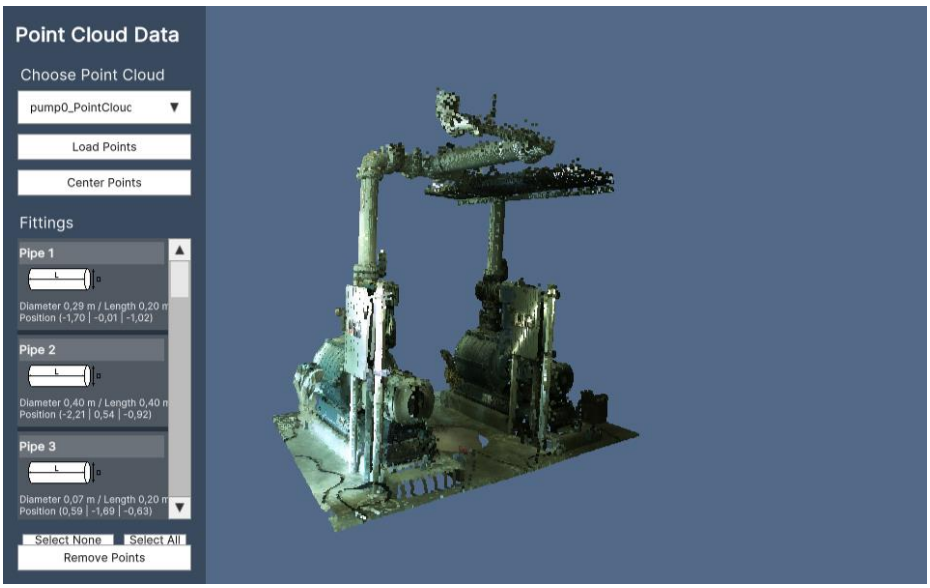

Abbildung 2 – Anwendung Point-Cloud-Viewer

Alle Implementierung im Bereitstellungsservice und in der Anwendungsentwicklung wurden während der Projektlaufzeit durch Codereviews und Tests begleitet. Die abschließenden Nutzertests wurden ausgewertet und das Feedback der Nutzer in die Anwendung eingearbeitet.

2.4 App zum Use-Case „Datenbankabfrage mit AR-/VR-fähigem Gerät“

Für die Anwendung „Datenbankabfrage mit AR-/VR-fähigem Gerät“ konnte bereits eine erste prototypische Umsetzung erfolgen. In einem kleinen Versuchsaufbau wurde ein Schiffsmodell und ein Marker durch eine Aufgestellte Kamera betrachtet (siehe Abbildung 3).

Abbildung 3: Versuchsaufbau AR App

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

In dem Versuchsaufbau wird durch die Kamera ein Marker erkannt, der sich auf ein Schiffsmodell bezieht. Passend dazu wird das reale Schiffsmodell getrackt und mit dem zugehörigen CAD-Modell überlagert. Auf diese Weise können immersiv Zusatzinformationen zu dem Schiffsmodell eingeblendet werden.

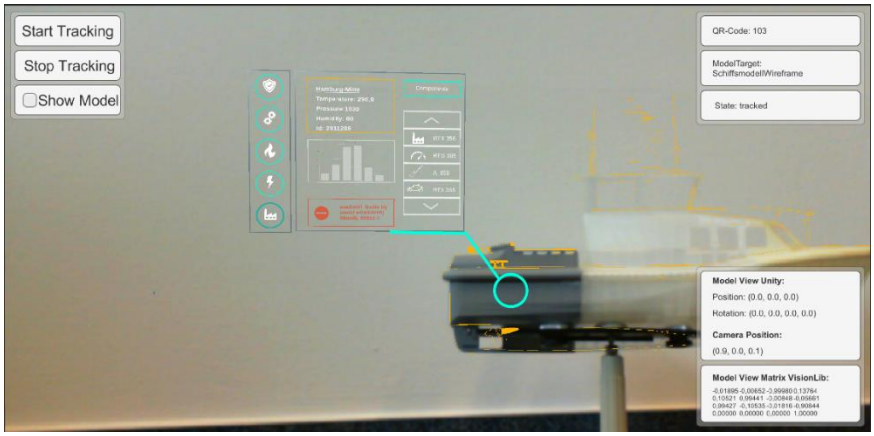

Abbildung 4: Prototyp der Anwendung

2.5 Services

2.5.1 Konvertierung von E57 Punktwolken

Da es sich bei dem häufig im Einsatz befindlichen E57-Format um ein Format handelt, welches nicht mit allen im Einsatz befindlichen Tools kompatibel ist, wurde zusätzlich ein Service entwickelt, mit dem sich diese Konvertierung durchführen lässt.

2.5.2 Erkennung von einfachen Geometrischen Objekten in Punktwolken (KI basiert)

Der KI-basierte Service hat die Aufgabe, Rohre und Rohrstücke in Punktwolken zu erkennen. Die dafür genutzten Punktwolken wurden mithilfe eines LiDAR-Sensors aufgenommen. Zur Lösung dieser Aufgabe wurde der Ansatz der Segmentierung gewählt, bei dem jedem Punkt in der Punktwolke eine Klasse zugewiesen wird, entweder "Rohr" oder "Nicht-Rohr".

Für die Segmentierung wurde ein Neuronales Netzwerk eingesetzt. Im Verlauf des Projekts wurden verschiedene Netzwerkarchitekturen untersucht, implementiert, modifiziert und verschiedene Experimente durchgeführt, um die beste Lösung zu finden.

Um das Problem der langen Lernphase bei Neuronalen Netzen zu bewältigen, wurden neben den mit einem echten LIDAR aufgenommenen Datensätzen auch synthetische Datensätze erzeugt. Dabei wurde ein Abtastmuster verwendet, das dem von echten LIDARs ähnelt.

Die Implementierung der Neuronalen Netze erfolgte mit Tensorflow[1], einer weit verbreiteten Open-Source-Bibliothek für maschinelles Lernen und künstliche Intelligenz.

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

Zur synthetischen Generierung von Punktwolken wurde Open3D [2] genutzt, eine leistungsstarke Bibliothek für 3D-Datenverarbeitung und Visualisierung. Die gesamte Software wurde in Python geschrieben.
Für die Anzeige und Analyse der Punktwolken wurde CloudCompare [3] verwendet, eine leistungsstarke Software, die speziell für die Verarbeitung und Visualisierung von 3D-Punktwolken entwickelt wurde.

Zunächst wurden verschiedene Netzarchitekturen zur Verarbeitung von Punktwolken untersucht. Insbesondere wurden die ersten Vorreiter auf diesem Gebiet betrachtet. Eine bekannte Architektur namens PointNet [4] wurde als bevorzugte Wahl für die Implementierung ausgewählt. PointNet zeichnet sich durch eine kompakte Größe und einfache Struktur aus und liefert dabei trotzdem sehr gute Ergebnisse.

Die Untersuchungen konzentrierten sich zunächst auf Verfahren zur Klassifizierung von Punktwolken und später wurde auch Segmentierungsverfahren mit einbezogen. Bei der Klassifizierung wird jeder (Teil-)Punktwolke eine Klasse zugewiesen, wie z. B. "Würfel", "Torus" oder "Zylinder". Für die Experimente mussten synthetische Datensätze verwendet werden, da die möglichen Szenarien begrenzt waren.

Für die synthetischen Daten wurden verschiedene dreidimensionale Objekte erzeugt, zufällig rotiert und skaliert. Anschließend wurden sie aus verschiedenen Blickwinkeln mit einem virtuellen LiDARs abgetastet. In Abbildung 1 sind einige Beispiele dargestellt. Zusätzlich wurde synthetisches Rauschen zu den Daten hinzugefügt, um die Robustheit der Netzarchitekturen gegenüber Rauschen im Allgemeinen zu testen. Ursprünglich wurde angenommen, dass reale Daten oft stark verrauscht sind. Später stellte sich jedoch heraus, dass die bereitgestellten Daten von unserem Projektpartner äußerst präzise waren und nur ein minimales Rauschen aufwiesen. Aus Gründen der Geheimhaltung können diese Daten hier nicht gezeigt werden.

Die Netzarchitekturen PointNet und PointNet++ erwiesen sich auf den synthetischen Daten für die Klassifizierung als effektiv. Insbesondere erreichte PointNet++ mit einem F1-Score von über 0,97 das beste Ergebnis. Diese Ergebnisse deuten darauf hin, dass die gewählten Architekturen vielversprechend sind und gute Leistungen bei der Verarbeitung von Punktwolken erzielen können.

Der nächste Schritt war dann die Segmentierung. Das Erzeugen von synthetischen Daten für diese Aufgabe erwies sich als sehr komplex, da die Anzahl der möglichen Szenarien sehr groß ist. Trotzdem konnten einige passende Szenen aus dreidimensionalen Objekten erzeugt und mit einem virtuellen LiDAR abgetastet werden. Die Ergebnisse in Abbildung 3 zeigen Teile einer solchen Szene. Auch PointNet erzielte hier bereits zufriedenstellende Ergebnisse. Um die Ergebnisse weiter zu verbessern, wurden anschließend komplexere Architekturen implementiert. Begonnen wurde mit dem Nachfolger PointNet++ [5]. Dieses lieferte insgesamt etwas bessere Ergebnisse bei der Segmentierung auf den synthetischen Datensatz.

Um nicht nur auf synthetischen Datensätzen zu trainieren, wurde zusätzlich ein echter Datensatz mit Punktwolken von Unterwasserpipelines genutzt [6]. Die Punktwolke wurde in diesem Fall jedoch aus einem Stereo-Kamera-System generiert, wodurch sich die Fehlercharakteristik von der des LIDAR unterscheidet. Sowohl PointNet als auch PointNet++ lieferten auf diesen Datensatz gute Ergebnisse, wie in Abbildung 2 zu sehen ist. Hierbei erzielte PointNet++ das beste Ergebnis mit einem kategorischen F1-Score von über 0,92.

Im weiteren Verlauf wurden noch andere komplexere Architekturen und Ansätze wie DGCNN [7], FoldingNet [8] und VoxNet [9] ausprobiert. Der Ansatz vom FoldingNet erwies sich als sehr interessant, da auch die generierte Punktwolke

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

Nachbarschaftsbeziehungen hat, die man im Voraus selbst festlegen kann, und durch den Faltungsvorgang nicht verloren gehen. VoxNet funktioniert mit einem dreidimensionalen Gitternetz, also einer anderen Datenstruktur als die der Punktwolke. Diese funktioniert gut für die Klassifizierung, skaliert allerdings weniger gut mit größeren Punktwolken.

Die Unterschiede in den Ergebnissen waren dabei allerdings sehr gering und der Trainingsaufwand wesentlich höher. Daher wurde sich im weiteren Projektverlauf wieder auf die Architekturen PointNet und PointNet++ konzentriert.
Eine weitere Herausforderung im Projekt, war die Größe der Punktwolken. Betrachtet man die Klassifizierungsverfahren, so reichen oft einige Tausend Punkte, um die Objekte sehr gut zu klassifizieren. Bei den Segmentierungsverfahren deckt die gleiche Menge an Punkten nur einen sehr kleinen Ausschnitt (grob 1 m^3) aus der realen Welt ab (erfasst mit einem LiDAR-Scanner). Diese Problematik wird dadurch verstärkt, dass die Netze eine feste Anzahl an Punkten für das Training und die Inferenz benötigen. Daher dürfen die zu segmentierenden Punktwolken auch nicht beliebig groß sein.
Im Projektverlauf wurde mit verschiedenen Ansätzen experimentiert, welche die Netze dynamisch aufbauen, sodass auch größere Punktwolken segmentiert werden können. Dieser Ansatz konnte jedoch nur konzeptionell fertiggestellt werden, da das Verwendete Framework Tensorflow es nicht ermöglicht hat, die gesamte Pipeline, innerhalb eines verhältnismäßigen Zeitraums, auf eine dynamische Eingabe und sogar dynamische Ausgabegröße umzustellen.

Aus diesem Grund wurde ein weiterer Verarbeitungsschritt notwendig, welcher die Punktwolke erstmal in kleinere Punktwolken unterteilt hat. Diese wurden dann einzeln segmentiert und im Nachgang wieder kombiniert. Die besten Ergebnisse konnten dabei erzielt werden, wenn sich die Teilwolken leicht überlappt haben.

In der Praxis wurden die Experimente stark durch den begrenzten Grafikspeicher unserer Grafikkarten eingeschränkt, so dass nicht alle notwendigen Architekturen ausprobiert werden konnten. Komplexere Netze, die viele Schichten haben, benötigen auch hier wieder wesentlich mehr Grafikspeicher, was die Experimente häufig eingeschränkt hat. Ein häufiges Problem hierbei war die Suche nach den Nachbarn eines Punktes, welches auf einer CPU zwar mit heuristischen Verfahren funktioniert, auf der Grafikkarte jedoch mit einem quadratisch anwachsenden Speicherverbrauch einhergeht.

Zusammenfassend lässt sich feststellen, dass es bereits erste Architekturen gibt, die speziell für die Verarbeitung von Punktwolken entwickelt wurden. Zudem können auch synthetische Datensätze erzeugt werden, mit denen die Neuronalen Netze trainiert werden können. Allerdings besteht ein großes Problem darin, synthetische Daten zu generieren, die das reale Anwendungsgebiet realistisch abbilden. Ein weiteres Hindernis stellt die Varianz in der Größe der Punktwolken dar, die aufgrund ihrer teilweise enormen Dimensionen zusätzlich verstärkt wird.

Das Hauptproblem lag jedoch darin, einen realen Datensatz zu finden, der für das Projekt geeignet war. In diesem Bereich scheint es noch viel Potenzial für die Erstellung großer und qualitativ hochwertiger Datensätze zu geben.

Wir sind zu dem Ergebnis gekommen, dass die entwickelten Netzwerke grundsätzlich gut funktionieren und in der Lage sind, die gestellte Aufgabe erfolgreich zu bewältigen. Allerdings ist dies nur möglich, wenn ausreichende Trainingsdaten vorhanden sind oder das Szenario gut definiert ist.

Im Rahmen des Projekts konnte jedoch noch keine allgemeine Lösung für das Problem der Rohrerkennung in Punktwolken gefunden werden.

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

2.5.2.1 Ergebnisse

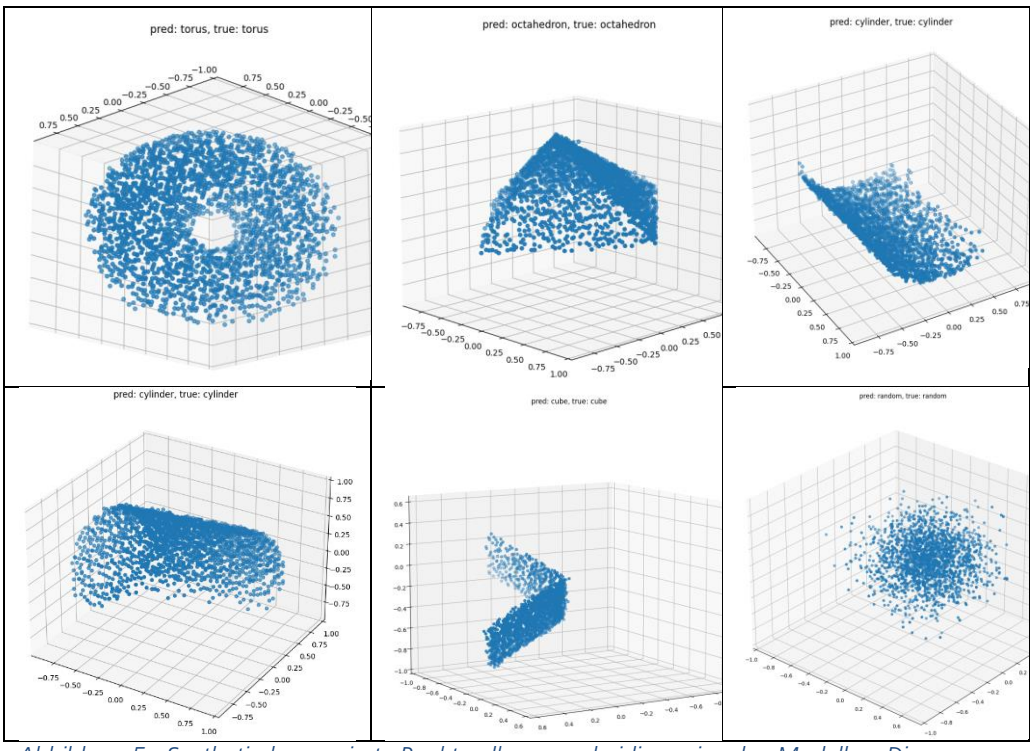

Abbildung 5 - Synthetisch generierte Punktwolken von dreidimensionalen Modellen. Die Überschriften zeigen jeweils die eigentliche Klasse „true“ und die Klasse, die das Netz (PointNet [4]) errechnet hat „pred“.

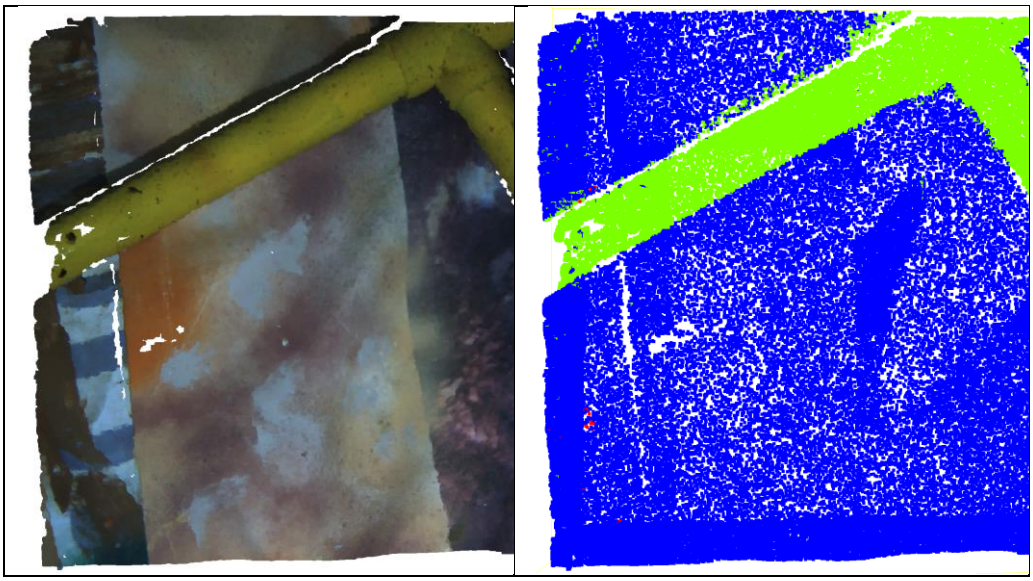

Abbildung 6 - Punktwolke aus dem Unterwasser Pipeline Datensatz [6] (links), segmentiert mit der PointNet++ [5] Architektur (rechts). Blau ist „Nicht-Rohr“, in diesem Fall also der Meeresboden und Grün ist „Rohr“, in diesem Fall also die Pipeline

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

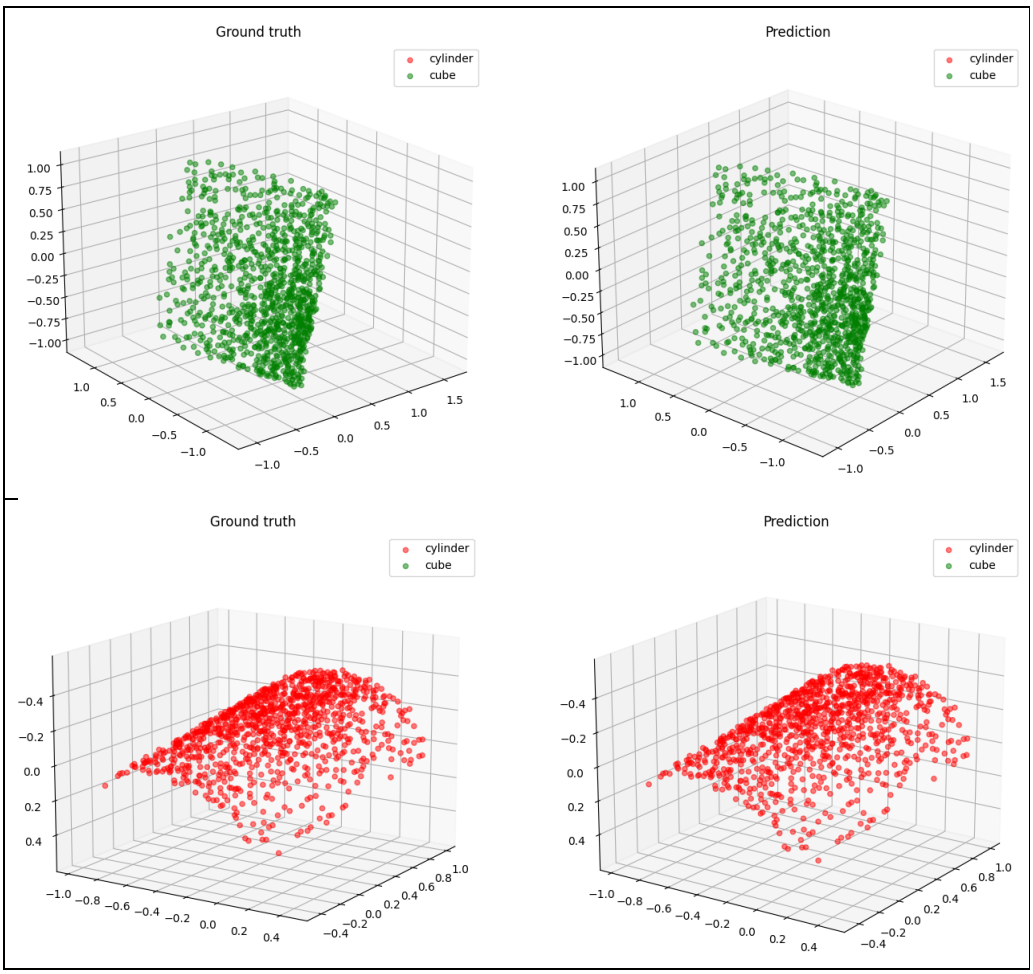

Abbildung 7 - Synthetisch generierte Punktwolken in denen jeder Punkt einer Klasse zugeordnet wurde. Links sind jeweils die Trainingsdaten („ground truth") und rechts dann die vom Netz errechneten Klassen („prediction")

2.5.2.2 Literatur

[1] Tensorflow. https://www.tensorflow.org/. Accessed: 30.08.2023

[2] Open3D. http://www.open3d.org/. Accessed: 30.08.2023

[3] CloudCompare. https://www.cloudcompare.org/. Accessed: 30.08.2023

[4] Qi, Charles R., et al. "PointNet: Deep learning on point sets for 3d classification and segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[5] Qi, Charles Ruizhongtai, et al. "PointNet++: Deep hierarchical feature learning on point sets in a metric space." *Advances in neural information processing systems 30*. 2017.

[6] Martin-Abadal, Miguel, et al. "Underwater pipe and valve 3D recognition using deep learning segmentation." *Journal of Marine Science and Engineering 9.1: 5*. 2020.

[7] Wang, Yue, et al. "Dynamic graph CNN for learning on point clouds." *ACM Transactions on Graphics (tog) 38.5: 1-12*. 2019.

[8] Yang, Yaoqing, et al. "FoldingNet: Point cloud auto-encoder via deep grid deformation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

[9] Maturana, Daniel, and Sebastian Scherer. "VoxNet: A 3d convolutional neural network for real-time object recognition." *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. 2015.

2.5.3 Erkennen von Zylindern in Punktwolken (klassische Verfahren)

Neben der Rohrerkennung/Zylinderdetektion in 3D-Punktwolken durch den Ansatz der Segmentierung mit Hilfe von Neuronalen Netzwerken wurde ein weiter Ansatz implementiert und getestet. Dieser Ansatz verzichtet auf den Einsatz von Methoden aus dem Bereich des maschinellen Lernens.

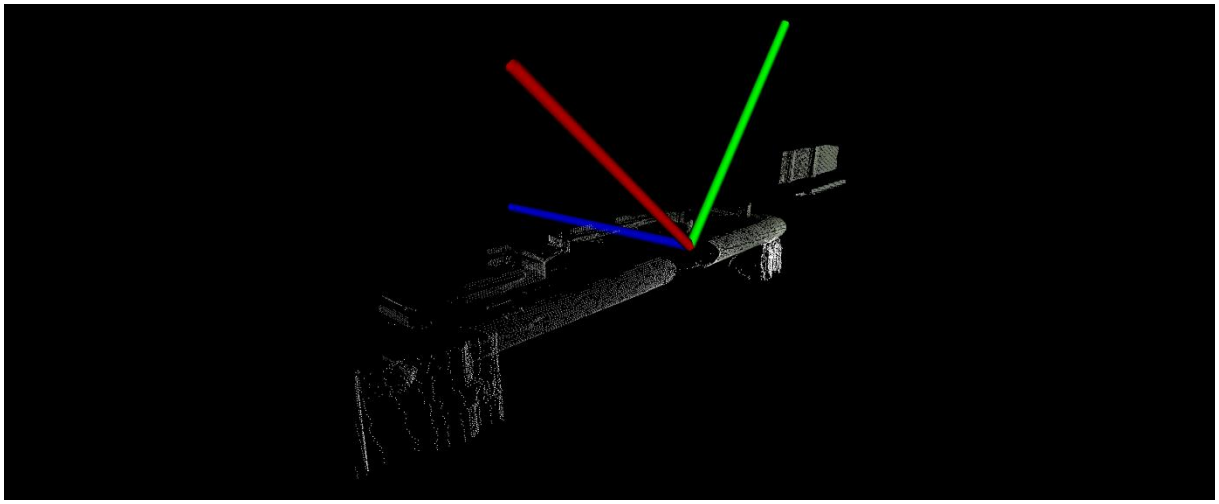

Abbildung 8: 3D-Punktwolke aus einem Laserscan ohne Ausrichtung am Koordinatensystem.

Den Ausgangspunkt bildet eine 3D-Punktwolke. Zur Entwicklung und zum Testen wurde eine 3D-Punktwolke aus einem Laserscan genutzt. Eine Annahme ist, dass eine Punktwolke typischerweise nicht ausgerichtet ist und beliebig in einem gegebenen Koordinatensystem liegt, wie in Abbildung 8 zu sehen.

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

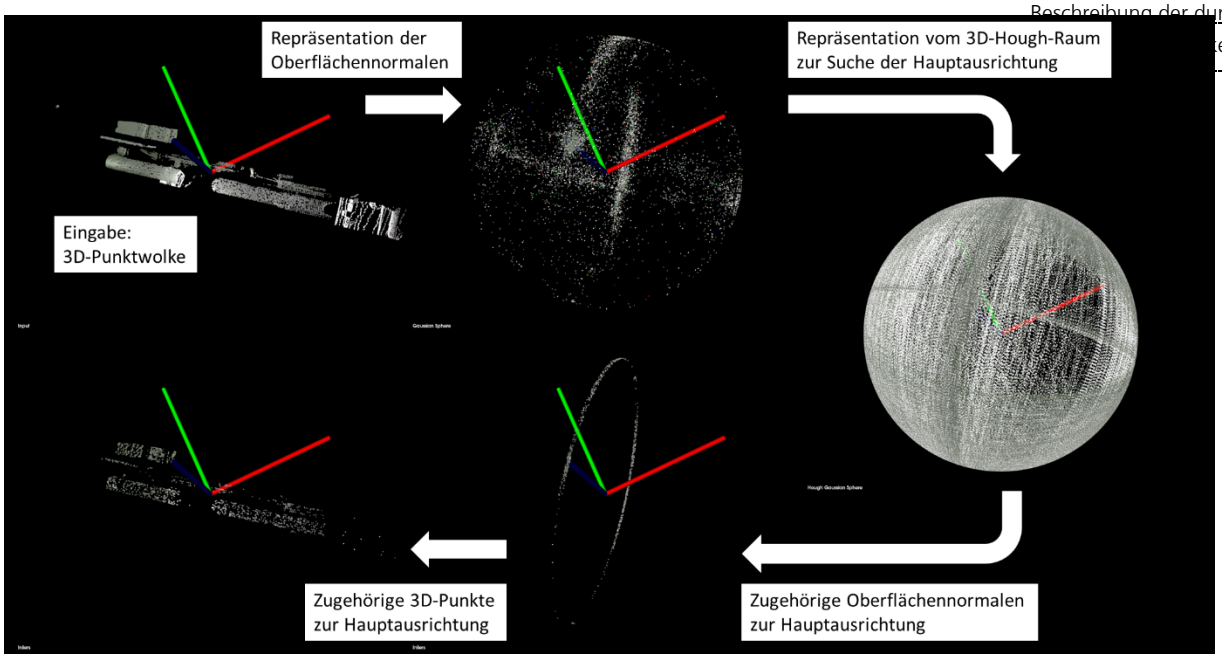

Abbildung 9: Bestimmung der Hauptausrichtung der 3D-Punktwolke und Extraktion der zugehörigen 3D-Punkte für die Zylinderdetektion.

Der implementierte Ansatz zeichnet sich dadurch aus, dass in einem iterativen Prozess jeweils die Hauptausrichtung der 3D-Punktwolke bestimmt wird. Dieser Vorgang ist beispielhaft in Abbildung 9 dargestellt. Die Hauptausrichtung der 3D-Punktwolke entspricht in diesem Fall der dominantesten Richtung in der 3D-Punktewolke, die unter Ausnutzung der Oberflächennormalen mit Hilfe einer Hough-Transformation bestimmt werden kann. Die Oberflächennormalen können dabei entweder bereits aus dem Scanprozess bekannt sein, oder sie werden in einem ersten Verarbeitungsschritt für jeden 3D-Punkt berechnet. Die Repräsentation der Oberflächennormalen für jeden 3D-Punkt erfolgt in Abbildung 9 auf einem Einheitskreis. Aus der Repräsentation der Oberflächennormalen wurde anschließend eine Parametrisierung für einen dreidimensionalen Hough-Raum abgeleitet, der zur Suche der Hauptausrichtung der 3D-Punktwolke anhand eines Voting-Schemas dient. Für die algorithmische Implementierung dieses Hough-Raums wurden Datenstrukturen aus der freien Programmbibliothek Point Cloud Library (PCL) genutzt. Aus diesem Voting-Prozess geht die dominanteste Richtung der 3D-Punktwolke hervor und es können Rückschlüsse auf die dafür verantwortlichen Oberflächennormalen und demzufolge auch auf die zugehörigen 3D-Punkte gezogen werden. Diese 3D-Punkte werden aus der ursprünglichen 3D-Punktwolke extrahiert und weiterverarbeitet, um zylinderförmige Strukturen zu detektieren. Nach der Zylinderdetektion wird die verbleibende 3D-Punktwolke dem ganzen Prozess erneut unterzogen, um die nächste Hauptrichtung zu ermitteln und auf zylindrische Strukturen zu untersuchen. Typischerweise kann dieser Prozess bei der Analyse von menschgemachten Strukturen, wie zum Beispiel der Verrohrung, nach 3 Iterationen gestoppt werden, da sich die meisten Rohre an den drei Raumrichtungen (Koordinatenachsen des 3D-Raums) ausrichten.

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

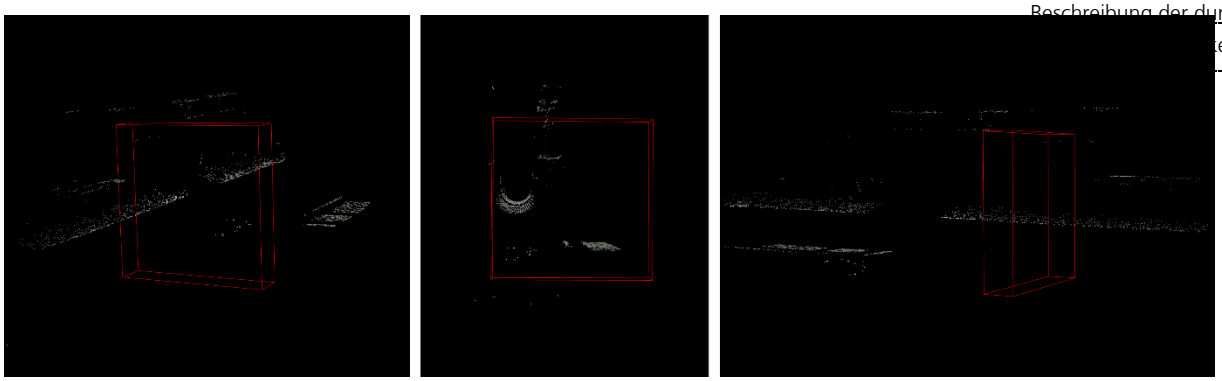

Abbildung 10: Scanvorgang der Teilpunktwolke zur Detektion von Zylindersegmenten.

Die Zylinderdetektion in den Teilpunktwolken erfolgt durch einen Scanvorgang mit einem 3D-Box-Filter. Ein Scanschritt ist beispielhaft in Abbildung 10 dargestellt. Die 3D-Punktwolke wird dabei durch eine teilweise überlappende Bewegung des 3D-Box-Filters in alle drei Richtungen gescannt. In jedem Scanschritt werden alle 3D-Punkte innerhalb des 3D-Box-Filers analysiert, um Zylindersegmente zu finden.

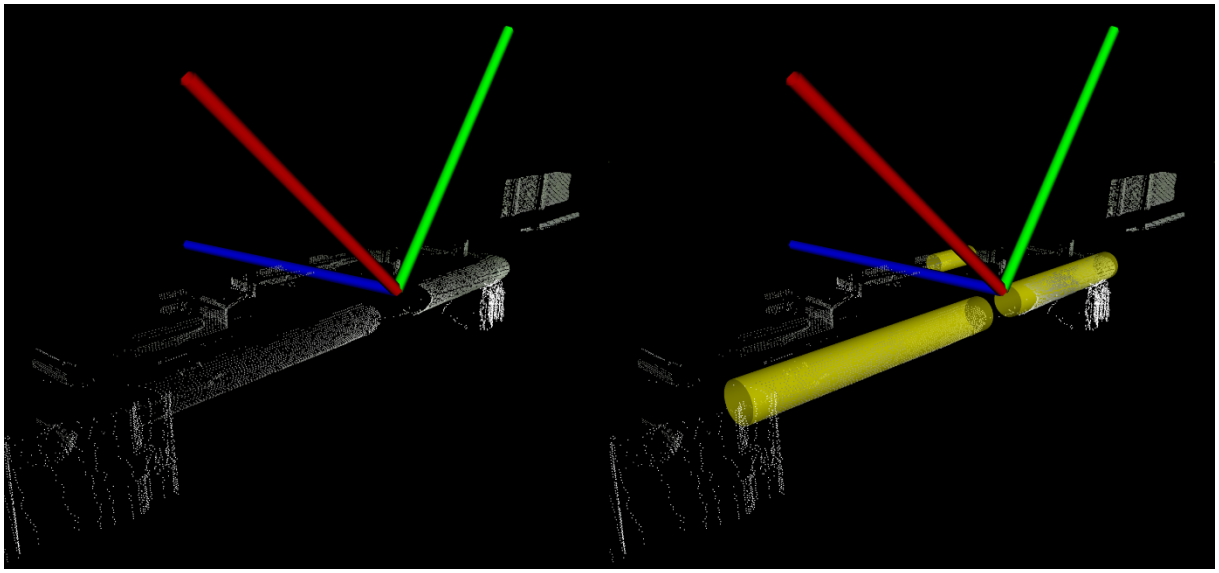

Abbildung 11: Detektierte Zylinder/Rohrabschnitte.

Nach diesem Scanvorgang erfolgt eine Plausibilisierung und das Zusammenfügen einzelner Zylindersegmente zu Zylindern, und somit zu Rohrabschnitten. Ein beispielhaftes Resultat ist in Abbildung 11 dargestellt.
Das entwickelte Verfahren liefert plausible Ergebnisse, arbeitet auf großen 3D-Punktwolken jedoch noch nicht sehr performant und weist somit noch einen großen Optimierungsbedarf auf. Insbesondere stellt hierbei die iterative Hough-Transformation zur Bestimmung der jeweiligen Hauptausrichtung einen sehr rechenintensiven Arbeitsschritt dar. Ein vielversprechender Ansatz wäre die Kombination mit der zuvor beschriebenen Segmentierung der 3D-Punktwolke mit Hilfe von Neuronalen Netzwerken. Die Plausibilisierung und das Zusammenfügen einzelner Zylindersegmente zu Zylindern erfolgt in rauscharmen 3D-Punktwolken, mit klar abgrenzbaren Objekten in den 3D-Punktwolken zu zufriedenstellenden Ergebnissen, kommt jedoch bei Scans von komplexen Anlagen mit dicht gelagerten Rohren schnell an die Grenzen. Im Rahmen des Projekts konnte somit noch keine allgemein-gültige Lösung für das Problem der Rohrerkennung in 3D-Punktwolken gefunden werden.

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013

3 Zusammenfassung

Im Laufe des Projekts konnte das IGD eine eigene Cloud-Infrastruktur erstellen, die sowohl im Zusammenspiel mit als auch ohne den instant3dhub genutzt werden kann. Es konnten verschiedene Services und Anwendungen entwickelt werden, welche in der Infrastruktur nutzbar sind. Zudem konnten unsere Kompetenzen im Bereich der KI basierten Punktwolkenanalyse deutlich ausgebaute werden.

Die Infrastruktur und Services wurden unserem Projektpartner der SKM Informatik GmbH per API zur Verfügung gestellt, welche dadurch in der Lage waren, auf Basis der Fraunhofer Technologie eigene an ihre Nutzer zugeschnittene Anwendungen zu erstellen.

Vorlage_IGD_Berichte_de_A4.dotm | 0.95 jv 22.01.2013