

## **Schlussbericht für das Vorhaben**

AI-Cube – Verbundvorhaben: Mehrwert aus Big Earth Data durch Kombination von KI und DIAS-föderierten Datenwürfeln

Laufzeit des Vorhabens: 01.09.2021 bis 31.08.2023

Peter Baumann<sup>1</sup>, Dimitar Misev<sup>2</sup>, Vlad Merticariu<sup>2</sup>, Heike Hoenig<sup>2</sup>, Otoniel Campos<sup>2</sup>, Mohit Basak<sup>1</sup>

<sup>1</sup> Constructor (formerly: Jacobs) University

<sup>2</sup> rasdaman GmbH

*Zuwendungsempfänger:* Constructor University,  
rasdaman GmbH  
  
Prof. Dr. Peter Baumann (Leitung)

*Förderung* BMWK - Bundesministerium für Wirtschaft und Klimaschutz  
BMWK-Ausschreibung  
„Entwicklung und Nutzung von Methoden der Künstlichen  
Intelligenz für den anwendungsorientierten Einsatz in der  
Satellitenerdbeobachtung“

*Förderkennzeichen:* 50EE2012A, 50EE2012C

*Vorhabensbezeichnung:* Verbundvorhaben:  
Mehrwert aus Big Earth Data durch Kombination von KI und  
DIAS-förderierten Datenwürfeln (AI-Cube)

*Laufzeit des Vorhabens:* 01.09.2021- 31.08.2023

*Berichtszeitraum:* 01.09.2021 - 31.08.2023

## Inhaltsverzeichnis

Teil I: Kurzbericht	3
I 1. Aufgabenstellung und Ziele des Verbundprojektes	3
I 2. Stand der Wissenschaft und Technik an den angeknüpft wurde	3
I 3. Ablauf des Vorhabens (Planung und Projektdurchführung)	5
I 4. Wesentliche Ergebnisse und Zusammenarbeit	5
Teil II: Darstellung zur Verwendung der Zuwendung und der erzielten Ergebnisse	6
II 1. Erzielte Ergebnisse in den einzelnen Arbeitspaketen	6
II 1.1 AP2: Etablierung Infrastruktur	6
II 1.2 AP3: ML-Unterstützung in Array-Datenbanken	7
II 1.3 AP4: Multi/Cross-modale Datenwürfel-gestützte KI-Methoden	12
II 1.4 AP5: Evaluierung	12
II 1.5 AP6: Öffentlichkeitsarbeit	14
II 2. Wichtigsten Positionen des zahlenmäßigen Nachweises	15
II 3. Notwendigkeit und Angemessenheit der geleisteten Projektarbeiten	15
II 4. Voraussichtlicher Nutzen, Verwertbarkeit und Perspektiven	15
II 5. Fortschritt bei anderen Stellen	16
II 6. Vorträge und Publikationen	16
Teil III: Erfolgskontrollbericht	19
III 1. Aufgabenstellung und Beitrag des Ergebnisses zu den förderpolitischen Zielen	19
III 2. Wissenschaftlich-technische Ergebnis des Vorhabens, erreichte Nebenergebnisse und gesammelte wesentliche Erfahrungen	19
III 3. Fortschreibung des Verwertungsplanes	19
III 3a. Erfindungen/Schutzrechtsanmeldungen und erteilte Schutzrechte	19
III 3b. Wirtschaftliche Erfolgsaussichten nach Projektende	19
III 3c. Wissenschaftliche und/oder technische Erfolgsaussichten nach Projektende	20
III 3d. Wissenschaftliche und wirtschaftliche Anschlussfähigkeit	20
IV. Berichtsblatt	21

# Teil I: Kurzbericht

## I 1. Aufgabenstellung und Ziele des Verbundprojektes

Im Rahmen des AI-Cube Projekts hatten Constructor University und rasdaman GmbH – in Zusammenarbeit mit dem Verbundpartner Technische Universität Berlin – das Ziel, KI-Methoden mit Datenwürfeln zu integrieren. Speziell im Fokus standen dabei raum-zeitliche Geo-Datenwürfel und deren Analyse mittels der RSVQA-Modelle der TU Berlin. Im RSVQA-Ansatz werden Neuronale Netze auf Pärchen von multi-modalen Radar- und optischen Satellitendaten für eine Land-Klassifikation auf Basis der europäischen CORINE-Einteilung trainiert; die Big Earth Net Kollektion der TU Berlin hält hierfür über 500,000 annotierte Sentinel-1 und Sentinel-2 Patches vor.

Derartig trainierte Modelle waren von Constructor University und rasdaman GmbH auf Datenwürfeln einzusetzen, indem ein Nutzer über eine Datenwürfel-Anfragesprache beliebige Regionen und Zeiten selektieren kann, auf welche das Modell angewendet werden soll. Dies erlaubt (i) erheblich vereinfachte Nutzung der ML-Modelle und (ii) ihre effiziente Auswertung im Server, also nahe an den Daten. Die dafür benötigten Sentinel-1 und -2 Datenwürfel waren auf Basis von EU DIAS-Archiven aufzubauen. Zusätzlich war die Einbindung in föderierte Datenwürfel-Archive zu untersuchen.

Die Ergebnisse waren über wissenschaftliche Veröffentlichungen sowie Kanäle der Wirtschaft (soziale Medien etc.) der Allgemeinheit zur Verfügung zu stellen.

**Arbeitsaufteilung.** Von den beiden im Projekt kombinierten Ansätzen, KI und Datenwürfel, wird der KI-Teil im Bericht der TU Berlin beleuchtet, während der vorliegende Bericht den Datenwürfel-Teil adressiert, welcher von Constructor University und rasdaman GmbH bearbeitet wurde.

## I 2. Stand der Wissenschaft und Technik, an den angeknüpft wurde

Die Fortschritte in der Satellitentechnologie haben einen bedeutenden Zuwachs an Erdbeobachtungsdaten (EO) in den Datenarchiven bewirkt. Diese wachsenden Datenarchive erfordern skalierbare, zuverlässige Methoden und Tools für den raschen Zugriff und die Verwaltung der EO-Daten. Zu den zentralen Methoden zählen:

- **Datenwürfel:** anstatt einer unübersehbaeren Zahl von Einzeldateien wird pro Sensor ein einziges homogenisiertes, raum-zeitliches Objekt angeboten, durch das der Nutzer ohne Hindernis traversieren und raum-zeitliche Analysen anwenden kann. Im Verlauf des Projekts ist der Begriff „Analysis-Ready Data“ (ARD) international immer bedeutsamer geworden, und Datenwürfel bilden einen wichtigen Baustein für ARD.
- **KI-Methoden:** Da es auf diesen „Big Data“ nicht möglich ist, die ganzen Daten analytisch oder gar von Hand zu charakterisieren (z.B. durch eine Verschlagwortung von Regionen), hat sich Maschinelles Lernen (ML) etabliert. Mit solcherart auf repräsentativen Daten trainierten Modellen lassen sich schnell Aussagen auf beliebigen Regionen gewinnen. Siehe den Bericht des Verbundpartners TU Berlin für Details.

**Datenwürfel, Array-Datenbanken.** Mit rasdaman wurden abfragbare Datenwürfel und Array-Datenbanken begründet, dokumentiert durch Patente und [über 160 Publikationen](#) u.a. auf VLDB und SIGMOD, den bedeutendsten Datenbank-Konferenzen. Inzwischen bilden Datenwürfel, wie oben diskutiert, einen Eckpfeiler für analysebereite Daten (ARD), ermöglichen sie doch durch die homogenisierte Integration von tausenden von Einzelszenen in ein einheitliches raum-zeitliches Objekt eine einfache Navigation in Raum und Zeit sowie vereinfachte Analyse, Visualisierung und Fusion.

Seit einiger Zeit wird die rasdaman-Idee server-seitig prozessierbarer Datenwürfel weltweit von diversen Gruppen aufgenommen. Technologisch lassen sich diese Ansätze wie folgt klassifizieren:

- **Array-DBMS**e wie SciDB, TileDB, SciQL, PostGIS Raster, Oracle GeoRaster, EXTASCID, Teradata. Mit Ausnahme von PostGIS Raster und Oracle GeoRaster (welche bekannt sind für mangelnde Skalierbarkeit und Performance) bieten sie keine EO-Unterstützung. SciDB/USA ist ein Array-DBMS realisiert auf Basis eines modifizierten Ingres-Kerns plus User-Defined Functions (UDFs).
- **Funktionsbibliotheken und Kommandozeilen-Werkzeuge** wie OPeNDAP (Programmierkenntnisse erforderlich, keine Föderation, mangelhafte Geo-Unterstützung), Ophidia, boost::geometry, Wendelin. core, TileDB; oftmals auch Erweiterungen von Python APIs (typisch: xarray) wie ODC (Open Data Cube, nur für Programmierer geeignet, mangelnde Parallelisierung und keine Föderation), Pangeo (Low-Level Zugriffsroutinen, keine Föderation), Brockmann CubeX (Prototyp-Stadium, keine Details publiziert) und ADAM (Prototyp, keine wirkliche Datenwürfel-Funktionalität). Manche dieser Werkzeuge werden mit einem Web-Frontend ausgestattet, z.B. für Ausschnittsbildung, jedoch ohne die Mächtigkeit und Flexibilität, welche die modernen Standards bieten. Andere erlauben das Hochladen und Ausführen beliebigen Python-Codes durch die Nutzer, was aus Betreibersicht ein beträchtliches Sicherheitsrisiko bedeutet und aus Nutzersicht spezielle Programmierkenntnisse erfordert. Weiterhin bietet keiner der Ansätze Sicherheitskonzepte auf Datenwürfeln.
- **Hadoop/Spark-Erweiterungen um Array-Unterstützung** wie SciHadoop, SciSpark, GeoTrellis, Calvalus). Sie sind generell schwierig in der Handhabung und ineffizient im Sinne von "Green Computing". So hat sich beispielsweise Calvalus auf CODE-DE nicht bewährt, nur wenige Anwender haben sich auf Hadoop eingelassen (im Gegensatz zur großen Zahl von CODE-DE Datacube Besuchern, 7000+ seit 2018).

Diese Ansätze sind für geringere Funktionalität, Performance und Skalierbarkeit als rasdaman mit seiner „full-stack“ Implementierung bekannt. Aus wissenschaftlicher Sicht fehlt den Ansätzen eine formale Grundlage vergleichbar der rasdaman Array Algebra. Andere Array-Formalisierungen, wie etwa RAM, haben keine Implementierung gefunden.

Im Feld der **Standardisierung** bilden die Coverage-Standards von OGC, ISO und INSPIRE den akzeptierten Industrie-Standard<sup>1</sup>, bestehend aus *Coverage Implementation Schema* (CIS) als Datenmodell mit einer Reihe von unterstützten Formaten sowie *Web Coverage Service* (WCS) Suite als Service-Modell mit dem einfachen Core (Ausschnittsbildung, Format-Codierung) und der *Web Coverage Processing Service* (WCPS) Datenwürfel-Analyse-

---

<sup>1</sup> [http://external.opengeospatial.org/twiki\\_public/CoveragesDWG/WebHome#Known\\_Implementations](http://external.opengeospatial.org/twiki_public/CoveragesDWG/WebHome#Known_Implementations)

sprache. Die WCS-Suite ist auf Petabyte-großen Diensten, z.B. DIASen, praktisch bewährt. Eine Reihe von offenen und proprietären Clients erlaubt den Zugriff auf solche Dienste.

Während im Bereich **Daten-Dienste** bei Metadaten COTS-Technologie (z.B. relationale Datenbanken) Standard ist, dominieren proprietäre, nicht-standardisierte Silo-Lösungen. Google Earth Engine ist ein Spezialfall: eine proprietäre Lösung, welche i.w. auf die schier unbegrenzten Hardware-Ressourcen von Google aufbaut, jedoch eine recht einfache Architektur benutzt<sup>2</sup>, ohne die Möglichkeiten der Anfrageoptimierung, Föderation etc. wie rasdaman. Von Nutzern wird jedoch das breite Datenangebot gelobt. Initiativen wie Earth-Server.eu arbeiten an standard-konformen, mächtigeren Dienste-Schnittstellen.

**Maschinelles Lernen auf Datenwürfeln.** Die Verbindung ist in der Theorie naheliegend: Datenwürfel repräsentieren Tensoren, die Anfragesprachen von Array-Datenbanken lassen sich in natürlicher Weise durch Tensor-Algebra beschreiben, Maschinelles Lernen basiert auf Tensor-Algebra. In der Praxis erlauben Array-Anfragesprachen allerdings derzeit nicht die Formulierung von Tensor-Algebra in voller Allgemeinheit. Typischerweise bieten Datenwürfel-Schnittstellen programmatische APIs in einer prozeduralen Programmiersprache (z.B. [ODC ML mit python](#)). Dies bringt eine Reihe von Nachteilen mit sich, u.a.: Sicherheitsbedenken (unsupervidiertes Ausführen beliebigen auf den Server hochgeladenen python-Codes), mangelnde serverseitige Optimierbarkeit, keine Terminierungsgarantie, komplizierte prozedurale Programmierung. Oft, wie bei ODC, werden spezialisierte Funktionen bereitgestellt, so dass sich nicht die generelle Mächtigkeit von Tensor-Algebren ergibt.

### I 3. Ablauf des Vorhabens (Planung und Projektdurchführung)

Der Ablauf des Verbundprojekts in Form von Arbeitspaketen entspricht den Planungen wie im Antrag formuliert. Änderungen wurden jeweils zeitnah in den jeweiligen Zwischenberichten dokumentiert. Die zentralen Veranstaltungen und Aktivitäten des zweijährigen Projektes wurden durch Prof. Dr. Peter Baumann koordinierend durchgeführt. Dazu zählten vor allem monatliche Meetings im Online-Format.

### I 4. Wesentliche Ergebnisse und Zusammenarbeit

Alle wissenschaftlichen Ergebnisse wurden bis zum Abschluss des Projekts in Form von Präsentationen auf internationalen Konferenzen und/oder durch wissenschaftliche Publikationen der breiten Fachöffentlichkeit zugänglich gemacht. Dies trägt zur Transparenz und Nachvollziehbarkeit der Forschungsergebnisse bei und ermöglicht es anderen Forschenden, auf den gewonnenen Erkenntnissen aufzubauen und diese weiterzuentwickeln.

---

<sup>2</sup> <https://www.sciencedirect.com/science/article/pii/S0034425717302900>

# Teil II: Darstellung zur Verwendung der Zuwendung und der erzielten Ergebnisse

## II 1. Erzielte Ergebnisse in den einzelnen Arbeitspaketen

### II 1.1 AP1: Projektmanagement

**Beteiligte Projektpartner:** Constructor U

**Geplante Tätigkeiten:** allgemeine Projektkoordination

**Erfolgte Tätigkeiten und Ergebnisse:** Berichte, Abschlußpräsentation, etc.

Begründet durch Verzögerungen bei der Einstellung eines Doktoranden an der Constructor University wurde abgesprochen, dass für AP2 rasdaman GmbH in 2021 verstärkt das Aufsetzen der Infrastruktur übernommen hat, während im Gegenzug Constructor University ab 2022 verstärkt Training etc. für TU Berlin übernommen hat. Danach wurde die Arbeitsaufteilung plangemäß fortgesetzt.

### II 1.2 AP2: Etablierung Infrastruktur

**Beteiligte Projektpartner:** alle

**Geplante Tätigkeiten:**

- Installation der bereits vorhandenen Komponenten der Projektpartner auf der CODE-DE (später: EO-Lab) Plattform.
- Wechselseitige Schulungen.

**Erfolgte Tätigkeiten und Ergebnisse:**

**rasdaman GmbH:**

- Parallel zur Aktualisierung des Datenwürfel-Dienstes auf CODE-DE, welcher prototypisch im Vorgängereprojekt BigDataCube erstellt worden war, durch den Projektpartner rasdaman GmbH wurde mit der DLR die im Projektantrag vorgesehene Nutzung von CODE-DE verhandelt.

**Anmerkung:** Nachdem die DLR den Zugang zu CODE-DE nicht weiterhin zur Verfügung stellen wollte, musste eine Ersatzlösung geschaffen werden. Dazu wurde zwischen dem Ende des Berichtszeitraums und dem Zeitpunkt der Abfassung dieses Berichts ein Antrag zur Nutzung von DLR EO-Lab eingereicht und genehmigt. Seitens DLR wurde der Zugang zu EO-Lab bereitgestellt, so dass auf dieser Basis – mit entsprechendem Zusatzaufwand für das erneute Aufsetzen – Copernicus-Daten für das Projekt zur Verfügung gestellt werden konnten.

Weiterhin wurden Trainings- und Testdaten in dieser rasdaman-Instanz angelegt, um eine geeignete Arbeitsbasis für die ML-Erweiterungen in WP3 zu erhalten. Zusätzlich wurde diese Instanz mit einem DIAS-Datenwürfeldienst föderiert, so daß dessen Copernicus-Daten ebenfalls zur Verfügung standen. Schließlich wurden im Rahmen

der Arbeitsvorbereitung Daten (z.B. von BigEarthNet) und Code von TU Berlin übernommen und auf dem Server bereitgestellt.

### Constructor U:

- Für die TU Berlin wurde eine Einführung in standard-basierte Datenwürfel-Dienste gegeben, und es haben initiale Detailgespräche zur Kopplung der Datenwürfel- und AI-Komponenten stattgefunden.
- Für alle Projektpartner wurde eine gemeinsame (interne) [Wiki-Seite](#) für Projektmanagement, Code-Sharing, Ideen, Diskussionen, etc., angelegt, zusammen mit einem (ebenfalls internen) [Version Control Repository](#) für die Verwaltung von Projektdokumenten, Code, Forschungsfortschritt, Publikationen und Sitzungsprotokollen.

## II 1.2 AP3: ML-Unterstützung in Array-Datenbanken

**Beteiligte Projektpartner:** Constructor U, rasdaman

### Geplante Tätigkeiten:

- **Task 3.1:** Sprachkonzepte für ML-Anfragen in Array-Datenbanken (Constructor U);
- **Task 3.2:** Prototypische Implementierung der ML-Erweiterungen in rasdaman; Dashboard; alles auf CODE-DE installiert und zur Evaluierung in AP5 bereit;
- **Task 3.3:** Server-seitige Optimierung von ML-Anfragen (rasdaman).

### Erfolgte Tätigkeiten und Ergebnisse:

- **Task 3.1** (Constructor U): In mehreren Treffen mit allen Partnern wurden Dienste und API-Anforderungen der ML-Komponente der TU Berlin herausgearbeitet. Die unterliegende Methodik der von TU Berlin benutzten ML-Werkzeuge wurde analysiert, um allgemeine Basisoperationen zu extrahieren. Dies führte zu einer ersten Liste von Operationen der Tensoralgebra, insbesondere Transposition, Multiplikation und Inversion von Matrizen; diese Operationen sind als Kandidaten für die QL-Spracherweiterungen festzuhalten. Weiterhin ist explizite Iteration („Schleifen“) erforderlich, da viele gängige Methoden inhärent iterativ sind und algorithmische Kriterien (z.B. erforderliche Genauigkeit) als Terminierungskriterium haben. Der vorhandene Array-Iterator ist hierfür nicht ausreichend, eine Erweiterung ist in Arbeit. Die Implementierung aller diskutierten Operatoren wird jedoch erschwert durch die gekachelte Ablage der Arrays, welche durch naive Iteration (z.B. row-major oder column-major) zu sehr ineffizienten Implementierungen führen würde.

Auch das ML-Austauschformat ONNX wurde untersucht. Im Ergebnis stellte sich heraus, dass Modelle über die verschiedenen gängigen Werkzeuge wie tensorflow und das von TUB verwendete pytorch nicht kompatibel sind. Daher wurde vorerst (auch bei den im ständigen Austausch durchgeführten Arbeiten von rasdaman GmbH) auf pytorch fokussiert, welches von RSVQA der TU Berlin verwendet wird.

Darauf aufbauend wurden die drei im Antrag gelisteten Varianten untersucht:

- (i) Die KI-Anwendung delegiert Teile der Bearbeitung an die Datenbank. Diese Alternative schied schnell aus, da bei den üblichen Werkzeugen (pytorch, TensorFlow, etc.) entsprechenden Codeänderungen abschreckend komplex wären.
- (ii) Die Anfragesprache wird um fehlende Tensoralgebra-Konstrukte erweitert, um ML-Algorithmen zu formulieren.

Diese Untersuchungen haben zu interessanten Ergebnissen geführt, jedoch ist ML nur in der Theorie durch einfache Algebra-Operationen auszudrücken; in der Praxis sind zu viele algorithmische (und oft unzureichend dokumentierte) Details eingewoben, um diesen Ansatz durchführen zu können.

Die Erweiterungen der Anfragesprache sind im Anhang beschrieben.

- (iii) Der Erweiterungsmechanismus „User-Defined Function“ (UDF) wird genutzt, um ML-Funktionalität in die Datenbank zu integrieren („ML as a Service“).

In SQL-Datenbanken erlauben benutzerdefinierte Funktionen (UDFs) eine Erweiterung der Funktionalität des Datenbankservers durch Hinzufügen einer Funktion, die in SQL-Anweisungen ausgewertet werden kann. Dieser Mechanismus ist auch in rasdaman enterprise verfügbar: Funktionen, die nicht in der Datenbank-Engine implementiert sind, sondern separat bereitgestellt werden, können als Teil von Ausdrücken in regulären Abfragen aufgerufen werden; dazu lädt der Server den Code dynamisch bei Bedarf.

Während der SQL-Standard zwischen skalaren und Tabellenfunktionen unterscheidet, kennt rasdaman stattdessen skalare und Array-Funktionen. Eine skalare Funktion gibt nur einen einzigen Wert (oder NULL) zurück, während eine Array-Funktion ein rasdaman-Array zurückgibt.

UDFs können in vielfältigen Situationen eingesetzt werden: Für ad-hoc Funktionserweiterungen, wo der im Server hinterlegte Quellcode bei Bedarf kompiliert und gelinkt wird bzw. eine Shared Library eingebunden wird. Dies ermöglicht nicht nur die Bereitstellung spezieller Funktionen, sondern auch eine Schnittstelle zu einer Vielzahl bestehender Tools dar wie z. B. Matlab, R oder ScalaPack.

Beispiel: Ein Paket `mat` könnte Matrixoperationen anbieten, u.a. eine Operation `transpose(mddExp)`; diese Funktion kann wie folgt aufgerufen werden:

```
select encode( mat.transpose( m ), "json")
from MyMatrices as m
```

In AI-Cube stellte sich nach einigen Überlegungen und Experimenten heraus, daß dies der vielversprechendste Weg zur ML-Integration ist. Er wird im folgenden beschrieben.

Nach eingehenden Untersuchungen – auch unter Beteiligung der rasdaman GmbH in Task T3.2 – ergab sich ein ernüchterndes Bild, was das Training von Modellen angeht:

- Es gibt nicht eine oder wenige stringent definierte Methoden zum Training, vielmehr benutzt jede Methode eigene Schritte, eigene Parameter, eigene Selektion von Lern/Testdaten, etc. Diese sind hochparametrisiert, und zwar wiederum individuell. Daher zeichnet sich keine allgemeine Trainings-Methode ab, welche sinnvoll durch Datenbanken unterstützt werden könnte. Weiterhin ist das Training von Modellen oftmals ein interaktiver Prozeß, bei dem der menschliche Supervisor Zwischenzustände abfragt und ggf. Trainingsparameter während des Laufes verändert. Dies ist inkompatibel mit dem üblichen synchronen Datenbank-Ansatz „sende Anfrage, erhalte Antwort“.
- Ähnliches gilt für die Modell-Auswertung: Wiewohl die Konzepte von ML auf Tensor-Algebra beruhen, werden in der Praxis viele – undokumentierte – algorithmische Schritte verwendet, welche sich nicht durch einfache Algebra beschreiben lassen.

Zur Sicherheit wurden weitere KI-Experten konsultiert, u.a. Wageningen Research, das Gesamtbild blieb jedoch dasselbe.

Daher wurde, nach eingehenden Überlegungen, der Ansatz der expliziten Formulierung von ML-Operationen in der Anfragesprache verworfen und statt dessen der UDF-Ansatz weiterverfolgt. Ein weiterer Grund für diese Entscheidung ist die Nutzerfreundlichkeit: Auch Nicht-Experten sollen AI-Cubes nutzen können, und dabei werden sie in der Regel vorab bereitgestellte Modelle als „Black Box“ nutzen, ohne Interesse an dem Auswerteprozess selbst.

- **Task 3.2** (rasdaman): Prototypische Implementierung der ML-Erweiterungen
  - **UDF-Implementierung:** Ziel war, in WCPS-Anfragen ML-Aufrufe zuzulassen, welche der Datenbank-Kern ausführt, indem er über eine UDF-Bridge pytorch aufruft, also beispielsweise:
 

```
for $s2 in (Sentinel-2),
    $m in (CropModel)
  return encode( nn.predict( $s2[ Lat(...), Lon(...), time(...) ], $m ), "tiff" )
```

Abb. 1 zeigt links schematisch den Einsatz von UDFs im rasdaman-Stack. In der (domänen-neutralen) unteren Schicht können die Arbeiter-Prozesse (rasserver) externen Code einbinden. Da sie in C++ implementiert sind, war die bei Projektstart verfügbare Schnittstelle zur Anbindung ebenfalls C++. Oberhalb des domänenneutralen Datenbankkerns sitzt eine Schicht, welche die Semantik von Raum und Zeit versteht (und damit auch reguläre und irreguläre Gitter) und diese dem Nutzer über die OGC-Standards WMS, WMTS, WCS und WCPS anbietet. Auch hier lassen sich UDFs einklinken, welche entweder direkt Funktionalität bereitstellen oder aber nur die Übersetzung in die rasql-Sprache des Kerns vornehmen. Die Zielsprache ist hier Java, da die Schicht in dieser Sprache implementiert ist. Im vorliegenden Fall wurde ein WCPS-UDF programmiert, welche einen Modell-Aufruf in einer WCPS-Anfrage in die korrespondierende rasql-Anfrage übersetzt, welche dann der Kern über einen C++-UDF ausführt.

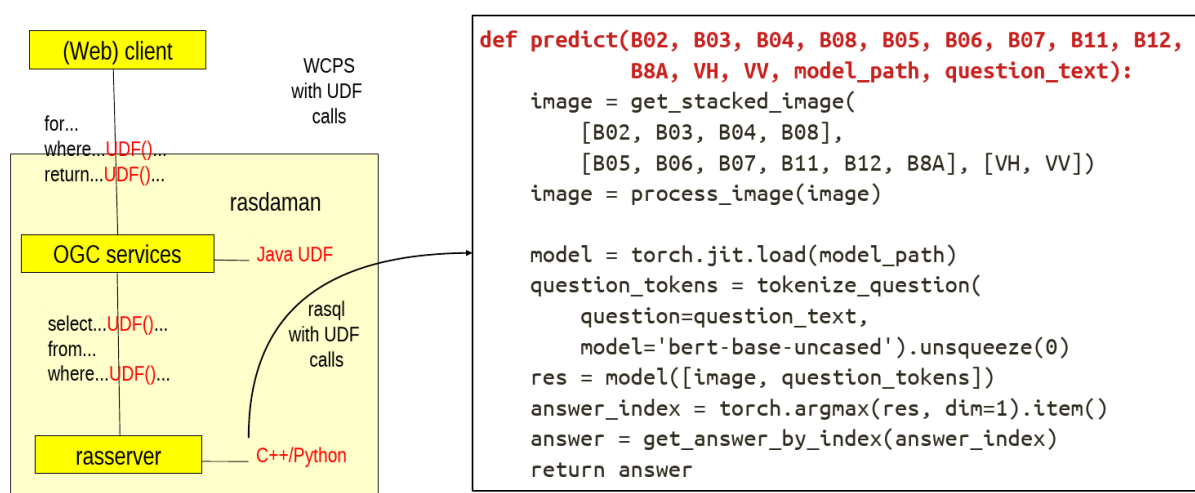


Abb. 1: pytorch-UDF in rasdaman zur Modellauswertung in WCPS-Anfragen

Für die Realisierung dieses C++ ML-UDFs im Datenbank-Kern bestand der erste Ansatz darin, den in C++ programmierten torch-Kern von pytorch anzubinden –

prinzipiell die effizienteste Lösung, da kein Marshalling C++ / python / C++ erforderlich wird. Die Ergebnisse wichen jedoch immer von den Referenzergebnissen der TU Berlin ab. Nach längeren Untersuchungen stellte sich heraus, daß der python-Wrapper von torch nicht nur eine Übersetzung der Parameter (Marshalling) bewerkstelligt, sondern bereits eigene Algorithmen in undokumentierter Weise einsetzt. Damit war dieser Weg hinfällig. Als nächster Ansatz wurden daher python-UDFs implementiert, welche beliebige python-Programme an den C++-Servercode anbinden können. Damit konnte Code von der TU Berlin eingebunden werden, der (wie meistens in ML) in Python geschrieben ist und ausgiebig Gebrauch von Python-Bibliotheken wie NumPy, PyTorch, Transformers, SciKit, etc. macht. Als Ergebnistypen kann dieser UDF NumPy-Arrays, Strings (wichtig für RSVQA) oder skalare Zahlenwerte zurückgeben. Nachdem dies hinreichend ausgetestet war, wurde zusätzlich eine spezielle, optimierte pytorch-UDF-Anbindung entwickelt. Damit konnten Modelle via pytorch direkt in ihrer Normal-Umgebung ausgeführt werden und lieferten, wie in AP5 nachgewiesen, korrekte Ergebnisse. Der Aufbau wird im folgenden schrittweise erläutert.

Die UDF-Funktion "rsvqa.predict()" erlaubt folgende Ein- und Ausgaben (Einschränkungen sind durch RSVQA gegeben):

Eingaben:

- (array) Sentinel-2 Band B02 120x120 (10 Meter Auflösung)
- (array) Sentinel-2 Band B03 120x120 (10 Meter Auflösung)
- (Array) Sentinel-2-Band B04 120x120 (10 Meter Auflösung)
- (Array) Sentinel-2-Band B08 120x120 (10 Meter Auflösung)
- (Array) Sentinel-2-Band B05 60x60 (20 Meter Auflösung)
- (Array) Sentinel-2-Band B06 60x60 (20 Meter Auflösung)
- (Array) Sentinel-2-Band B07 60x60 (20 Meter Auflösung)
- (Array) Sentinel-2-Band B11 60x60 (20 Meter Auflösung)
- (Array) Sentinel-2-Band B8A 60x60 (20 Meter Auflösung)
- (Array) Sentinel-1-Band VH 120x120 (10 Meter Auflösung)
- (array) Sentinel-1 Band VV 120x120 (10 Meter Auflösung)
- (string) Pfad zu der trainierten Modelldatei auf dem Server
- (string) textuelle (englische) Frage zum bereitgestellten Bild

Alle Array-Parameter müssen das gleiche geografische Gebiet abdecken.

Ausgabe:

- (string) Die Antwort auf die Eingabefrage als Klartext

Beispiel:

```
SELECT
  rsvqa.predict(
    B02, B03, B04, B08, B05, B06, B07, B11, B12, B8A, VH, VH,
    "rsvqa_trained_model.pt",
    "What L1 land cover classes are there in the image?")
FROM S2_L2A_B02_10m as B02, S2_L2A_B03_10m as B03,
     S2_L2A_B04_10m as B04, S2_L2A_B08_10m as B08,
     S2_L2A_B05_20m as B05, S2_L2A_B06_20m as B06,
     S2_L2A_B07_20m as B07, S2_L2A_B11_20m as B11,
     S2_L2A_B12_20m as B12, S2_L2A_B8A_20m as B8A,
     S1_GRDH_IW_DV_VH as VH, S1_GRDH_IW_DV_VV as VV
```

Ergebnis: „water bodies“

Die WCPS UDF hat die gleiche Signatur wie die rasql UDF:

Beispiel:

```

for $B02 in (S2_L2A_B02_10m),
  $B03 in (S2_L2A_B03_10m),
  $B04 in (S2_L2A_B04_10m),
  $B08 in (S2_L2A_B08_10m),
  $B05 in (S2_L2A_B05_20m),
  $B06 in (S2_L2A_B06_20m),
  $B07 in (S2_L2A_B07_20m),
  $B11 in (S2_L2A_B11_20m),
  $B12 in (S2_L2A_B12_20m),
  $B8A in (S2_L2A_B8A_20m),
  $VH in (S1_GRDH_IW_DV_VH),
  $VV in (S1_GRDH_IW_DV_VV)
let subs2 := [ansi("2023-06-08"), Lat(53.045:53.057), Lon(8.7800:8.7920)],
  subs1 := [ansi("2023-06-04"), Lat(53.045:53.057), Lon(8.7800:8.7920)]
return rsvqa.predict2($B02[subs2], $B03[subs2], $B04[subs2], $B08[subs2],
  $B05[subs2], $B06[subs2], $B07[subs2], $B11[subs2], $B12[subs2],
  $B8A[subs2], $VH[subs1], $VV[subs1],
  "rsvqa_trained_model.pt",
  "What L1 land cover classes are there in the image?"
)

```

Antwort: „water bodies“

Die Realisierung des UDFs muß nicht nur das Marshalling (dies ist weitgehend automatisiert in rasdaman), sondern auch eine Daten-Vorprozessierung leisten. Im Detail bewerkstelligt dies der Code aus B.. 1 (rechts) wie folgt:

- Die Sentinel-1- und Sentinel-2-Daten werden vorprozessiert, um sie für das Vorhersagemodell geeignet zu machen:

- Reskalierung der S-2 20m Bänder auf eine Auflösung von 120x120 mit bikubischer Interpolation; alle Bänder sollten nun die Größe 120x120 haben.
- Stapeln der Bänder von S2\_10m, S2\_20m und S1 in der Reihenfolge [S2\_B02, S2\_B03, S2\_B04, S2\_B08] (10m), [S2\_B05, S2\_B06, S2\_B07, S2\_B11, S2\_B12, S2\_B8A] (20m) und [S1\_VH, S1\_VV] (10m).
- Konkatenieren der gestapelten Bänder, so dass ein multispektrales Bild mit 12 Bändern entsteht.
- Normalisierung des Bildes mit diesen Werten (von der TU Berlin bereitgestellt und im UDF hartcodiert; sie sind empirisch gewonnen, ihre Gültigkeit bzw. Übertragbarkeit ist unklar):

mean = [ 429.9136,	std = [ 213.2436,	# B02
614.1826,	252.0842,	# B03
590.1931,	316.4347,	# B04
2219.0117,	675.8503,	# B08
950.5765,	334.6534,	# B05
1792.4370,	517.9016,	# B06
2075.4717,	603.1708,	# B07
2266.4849,	632.5003,	# B8A
1594.4441,	458.6363,	# B11

1009.3064,	382.7222,	# B12
19.2280,	3.1637,	# VV
-12.5923 ]	2.9608 ]	# VH

- Zerlegen der natürlichsprachlichen Anfrage mit dem [BERT-Tokenizer](#) in Wörter und Wortteile, Umwandeln in Zahlen mit [bert-base-uncased](#).
- Übergabe des Multi-Band Stacks (Schritt 1) und der tokenisierten Frage (Schritt 2) an das Modell.
- Bestimmung des Antwortindex aus der Ausgabe mittels argmax und dem Index der höchsten Aktivierung.
- Konvertieren vom Index in eine textuelle Antwort.

Diese UDFs wurde im Projektserver für die projektinterne Diskussion, Evaluierung in WP5 sowie öffentliche Demos (vgl. AP5) bereitgestellt.

- **Dashboard-Frontend:** Während rasdaman durch seine OGC-konformen APIs eine breite Vielfalt an Dritt-Clients unterstützt, sind viele dieser Clients noch sehr an 2D-Karten gebunden und erlauben keinen adäquaten Zugang zu Datenwürfeln. Daher hat rasdaman GmbH schon vor längerem die Entwicklung eines eigenen Web-basierten Dashboard-Clients begonnen, welcher die einfache Konfiguration von kundenspezifischen Clients, von Kiosk-Modus bis zum Expertenwerkzeug, erlaubt. AI-Cube ist auf einer ersten Version dieses Dashboards, auf dem Level eines „Minimum Viable Products“, gestartet und hat diese Version um Funktionalität angereichert, u.a. Diagramme, verbesserte Kontrolle der Layer-Darstellung, mehr Interaktionsmöglichkeiten, etc. Das verbesserte Dashboard wurde u.a. im AI-Cube Demonstrator eingesetzt (siehe WP5).
- **Task 3.3:** Server-seitige Optimierung von ML-Anfragen (rasdaman)  
Die gezeigte UDF-Implementierung reicht die Patches 1:1 durch zu pytorch. Damit muß der Benutzer (d.h. Verfasser der Anfrage) die übliche Patch-Größe von 120x120 (bei 10m Bodenauflösung) einhalten. Eine Erweiterung des UDFs könnte beliebig große Patches akzeptieren und diese intern in die pytorch-genehme Größe kacheln. Dies bietet gleichzeitig eine gute Möglichkeit, die Modellauswertung zu parallelisieren.

## II 1.3 AP4: Multi/Cross-modale Datenwürfel-gestützte KI-Methoden

**Beteiligte Projektpartner:** TU Berlin

Siehe dortigen Bericht.

## II 1.4 AP5: Evaluierung

**Beteiligte Projektpartner:** rasdaman, TU Berlin

**Geplante Tätigkeiten:**

- Technische Validierung auf CODE-DE (rasdaman)
- Anwendungs-Validierung auf CODE-DE (rasdaman, TU-Berlin)

**Erfolgte Tätigkeiten und Ergebnisse:**

**Technische Validierung.** Die prinzipielle Machbarkeit des UDF-Ansatzes wurde anhand von Original-Modellen der TU Berlin demonstriert. Die Korrektheit der Ergebnisse wurde im direkten Vergleich mit den Trainingsdaten validiert (mehr dazu weiter unten). Hinsichtlich der Performance ergaben sich nur kleine Unterschiede zwischen einer Modellauswertung via Kommandozeile und einem UDF-Aufruf. Der Zusatzaufwand im Server, bedingt durch Anfrageübersetzung und Marshalling der Parameter über die UDF-Schnittstelle bewegt sich in der Größenordnung von 10ms. Auch das Optimierungspotential wurde untersucht, siehe hierzu die Ergebnisse in T3.3.

**Anwendungs-Validierung.** Im Labor-Prototyp wird noch eine Reihe von Einschränkungen des Modells an den Nutzer durchgereicht, von deren Einhaltung das Funktionieren des Modells abhängt: alle Patches über derselben Region, gleiche Größe, festgelegte Auflösung, etc. In einer industriellen Anwendung wird man diese Komplexitäten und Fehlerquellen hinter einem komfortablen, sicheren UDF verbergen.

Ein dergestalt vereinfachter öffentlich zugänglicher Demonstrator ist unter <https://ai-cu.be> verfügbar. Er basiert auf dem rasdaman-UDF, dem rasdaman Dashboard und einem RSVQA-Modell der TU Berlin (Abb. 2).

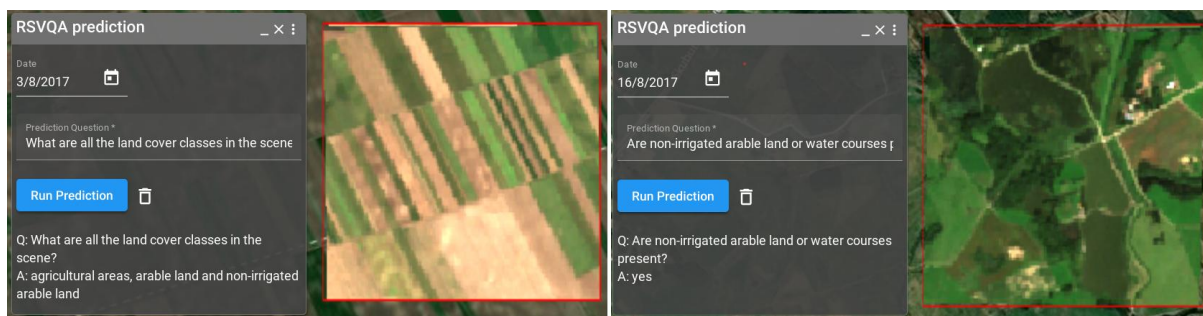


Abb. 2: AI-Cube Demo-Service, auf Basis von RSVQA

Ein generelles Problem ergab sich hinsichtlich der Antwort-Qualität. Bei Regionen in Raum und Zeit (trivial im Dashboard zu selektieren) kollabierte die Zuverlässigkeit dramatisch, sobald die Regionen außerhalb der antrainierten Wohlfühlzone des Modells lagen. Dieses Phänomen wird von der TU Berlin bestätigt (siehe dort): gegenüber einer gutartigen Trefferwahrscheinlichkeit von 75% - 85% fällt diese Wahrscheinlichkeit gerne auf 20% - 25% (also schlechter als Würfeln). In Gesprächen mit anderen KI-Forschern wurde dieses Phänomen bestätigt. Im o.g. Demonstrator wurden daher Szenen in Raum und Zeit handverlesen, auf denen RSVQA gute Ergebnisse abliefern.

Im Bericht der TU Berlin wird detailliert ausgeführt, welche Herausforderungen das Modelltraining insbesondere hinsichtlich einer Reproduzierbarkeit des Trainingsergebnisses beinhaltet. Zur Veranschaulichung: TU Berlin berichtet, daß das beste Modell ungefähr 210 Millionen lernbare Parameter besitzt.

Als Reaktion u.a. auf diese Diskussion hat TU Berlin einen neuen Trainings-Test-Split entwickelt, bei dem die Trainingsdaten weniger mit den Testdaten korrelieren und geografisch weiter voneinander entfernt sind. Dies sollte dazu beitragen, dass die Evaluation von Modellen auf BigEarthNet aussagekräftiger wird.

Als Konsequenz sieht rasdaman GmbH – vor jeglichem Angebot an Nutzer – die Notwendigkeit, Modelle zu überwachen. Da in dem angestrebten Marktplatz von Modellen die Genesis der Modelle nicht beeinflusst werden kann (und das Unternehmen weder Ressourcen noch Kompetenz hätte) und üblicherweise auch nicht hinreichend dokumentiert

ist, bleibt nur, aufgrund von Heuristiken die Modelle vor einer Anwendung abzuschätzen und ggf. Maßnahmen zu ergreifen. Wir nennen diesen Ansatz „model fencing“. Damit hat sich ein neues, umfangreiches Forschungsthema ergeben - siehe hierzu auch II.4.

**Externe Probleme.** Eine weitere kritische Problemquelle bilden Überarbeitungen der Daten durch deren Anbieter. Mit verbessertem Verständnis der EO-Pipelines werden Softwareversionen aktualisiert, Parameter nachjustiert (beispielsweise bei Sen2Cor von der ESA) oder gar Werkzeuge ausgetauscht. Dies bedeutet einen Bruch in der Zeitreihe – für den Benutzer macht es keinen Unterschied, „2023-05-30“ oder „2023-06-01“ aufzurufen, jedoch können die unterliegenden Daten signifikant unterschiedliche Merkmale aufweisen – Abb. 2 gibt ein Beispiel. Insbesondere erweisen sich trainierte Modelle als extrem empfindlich.

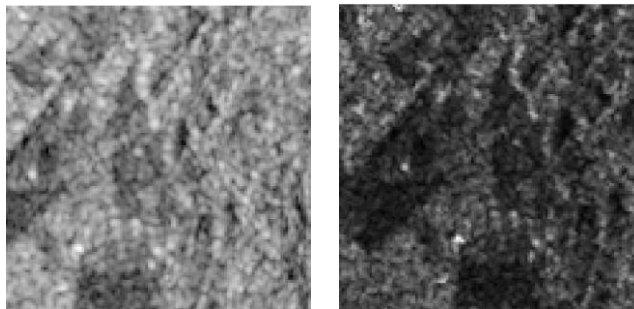


Abb. 2: Sentinel-1 Szene, von der ESA zu unetrschiedlichen Zeiten ausgeliefert

Die ESA versucht, wieder Einheitlichkeit zu schaffen, in dem auch die historischen Daten nachprozessiert werden. Dies erhöht jedoch die Verwirrung weiter, da nun z.B. die trainierten Modelle nicht mehr zu den (nunmehr modifizierten) Sentinel-Trainingsdaten passen.

Aber auch andere Daten werden verändert, wie z.B. i Projekt FAIRiCUBE beobachtet wurde. CORINE hat im Lauf der Zeit weitere Klassen hinzugefügt, EEA hat die Auflösung der Daten geändert, etc.

Diese Problematik wird aus KI-Sicht auch im Bericht der TU Berlin thematisiert. Aus unserer Sicht wurde das Thema, durchgeführt u.a. im Kontext OGC Testbed-19, inzwischen in einem eigenen Papier zusammengefaßt<sup>3</sup>.

## II 1.5 AP6: Öffentlichkeitsarbeit

### Beteiligte Projektpartner:

- JacobsU
- TU Berlin
- rasdaman

### Geplante Tätigkeiten:

- **Task 6.1:** Pflege Webauftritt, Projektflyer (Constructor U)
- **Task 6.2:** Internationale wiss. Veröffentlichungen (Constructor U, TU Berlin)
- **Task 6.3:** kontinuierliche Medien-Präsenz (rasdaman GmbH)

---

<sup>3</sup> P. Baumann: On the Analysis-Readiness of Spatio-Temporal Earth Data and Suggestions for Its Enhancement. Environmental Modelling and Software (under review)

## Erfolgte Tätigkeiten und Ergebnisse:

### Constructor U:

- Zeitnah zu Projektbeginn wurde der Web-Auftritt des Projekts realisiert und ist – auch nach Projektende – online unter <https://ai-cu.be>.
- Anstelle des Flyers wurde ein Rollup entwickelt, welches oftmals auf Messen etc. zum Einsatz kommen konnte (siehe Messe-Impressionen weiter unten)
- Wissenschaftliche Publikationen und Konferenzbeiträge, siehe Kapitel II.6

### rasdaman GmbH:

- Symposiums- und Messeauftritte, siehe Kapitel II.6
- Social Media: *Greencube Data Analytics*. Published in GISresources + dissemination in 9 geo-relevant LinkedIn groups: Earth Observation Group, Geomatics, GIS, GIS Professional & Networking, Horizon Framework Program Research & Innovation, OGC, GIS & Geospatial Technology Leadership Group, GIS Expert, Database Experts



## II 2. Wichtigsten Positionen des zahlenmäßigen Nachweises

Siehe profi-online Tabellen

## II 3. Notwendigkeit und Angemessenheit der geleisteten Projektarbeiten

Der Verlauf der Arbeit im Projekt erfolgte in großen Teilen in der im Projektantrag formulierten Planung. Der größte Teil der formulierten Aufgaben wurde erfolgreich bearbeitet und in angemessenem Umfang verbreitet.

## II 4. Voraussichtlicher Nutzen, Verwertbarkeit und Perspektiven

In AI-Cube wurde von den beiden Datenwürfel-Partnern die Grundlage für die Anwendung trainierter Modelle in Datenwürfel-Anfragen gelegt und die prinzipielle Machbarkeit an einem speziellen Modell (RSVQA) demonstriert. Auf dieser Basis sind weitere Forschungsfragen entstanden, welche künftig weiter behandelt werden sollen:

- Wie können ML-Modelle als „first-class citizens“ in der Datenbank verwaltet werden? Dies betrifft Ablageformat, notwendige Metadaten, Import und Export, Verwaltung von Zugriffsrechten und vieles mehr. Dies ist insbesondere für eine künftige Kommerzialisierung der AI-Cubes unabdingbar.

- Derzeit wird nur pytorch als ML-Engine unterstützt. Ist es erforderlich, weitere Werkzeuge (etwa TensorFlow) zu unterstützen, würden die Ergebnisse bei ein und demselben Modell und Analysegebiet abweichen? Idealerweise wird der Nutzer nicht mit der Entscheidung über pytorch oder TensorFlow belastet, da kaum jemand die Konsequenzen abschätzen kann. Auch dies ist für einen kommerziellen Einsatz wichtig zu beforschen.
- Die Zuverlässigkeit der Modelle nimmt offenbar rapide ab, sobald sich die Analyse-Region aus der raum/zeitlichen Wohlfühlzone der Trainingsdaten des Modells bewegt. Die neue einfache Zugriffsmöglichkeit auf jeglichen Ort, jegliche Zeit, welche die Datenwürfel mit sich bringen, führt damit zu erheblicher Gefahr von Falschergebnissen. Wir wollen daher ein Konzept des „model fencing“ erforschen, welches vor Anwendung der Modelle die Anwendbarkeit in der konkreten Anfrage prüft und gegebenenfalls vor der Anwendung warnt. Ein einfaches Beispiel wäre eine Landmaske, welche vor Anwendung eines Buschfeuer-Modells prüft, ob die Region der Anwendung Land oder Wasser adressiert. Pixel über Wasser können z.B. vor der Modellauswertung auf Nullwerte gesetzt werden, sodaß das Modell diese nicht interpretiert.

Nach Projektende werden die Arbeiten von AI-Cube bereits aktiv weitergeführt:

- EU FAIRiCUBE (aktiv), Constructor University ist Partner;
- AgriCube: gemeinsamer DFG/MOST-Antrag, eingereicht von Constructor University & Feng Chia University Taiwan;
- EU WorldCube: Föderierte AI-Cubes, Constructor University & rasdaman GmbH im Konsortium zum breiten Einsatz von AI-Cubes für Digital Twins.

In allen Projekten ist eine enge Zusammenarbeit sowohl mit KI-Experten als auch mit Anwendern vorgesehen bzw. realisiert.

## II 5. Fortschritt bei anderen Stellen

Projektrelevante Ergebnisse wurden auf Fachtagungen ausgetauscht und auf entsprechenden Vorträgen/Workshops präsentiert.

## II 6. Vorträge und Publikationen

### Publikationen

- S. Villarroya, P. Baumann: *A Survey on Machine Learning in Array Databases*. Applied Intelligence 53, pp. 9799–9822, 2022-08-12
- P. Baumann: *Federated AI-Cubes: Insight from Petabytes in Space and Time*. Poster, EuroGEO, Athens, Greece, 2022-12-08
- O.J. Campos Escobar, P. Baumann: *Implementation Roadmap for Neural Networks in Array Databases*. Proc. Computational Science and Computational Intelligence (CSCI), Las Vegas, USA, 2022-12-14

- P. Baumann, O. Campos, D. Misev: *AI-Enabled Analysis-Ready Datacubes: Towards a Roadmap for More Human-Centric Services*. Proc. IEEE Geoscience and Remote Sensing Society (IGARSS), Pasadena, USA, 2023-07-16
- P. Baumann, D. Misev: *AI and Datacubes: Towards a Happy Marriage*. Poster, ESA Big Data from Space (BiDS), Vienna, Austria, 2023-11-06

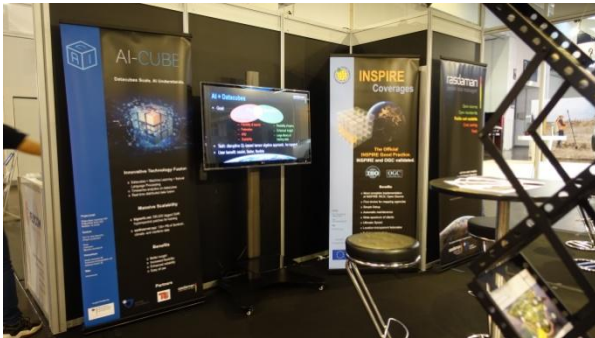
### Vorträge und Konferenzbeiträge

- "DWKI'21 Symposium – Datenwürfel: stark im Team, stark mit KI", half-day side event to Intergeo (hybrid), 2021-09-22
- P. Baumann: ESA PHI-Week Side Event, PHI Week, virtual event, 2021-10-14
- P. Baumann, Kai Clasen: *AI-Cube: AI + Federated Datacubes = Added Value*. DLR KI-Kickoff, virtual, 2022-03-31
- P. Baumann: *AI-Cube: AI + Federated Datacubes = Added Value*. AI, Automation, and Robotics session, GWF, Amsterdam, NL, 2022-05-11
- P. Baumann: *AI-Cube: Combining Datacube Scalability with AI Intelligence*. Poster, ESA Living Planet Symposium, Bonn, Germany, 2022-05-23
- *Earth Datacubes: From Simplified Access to ML Analytics*. Tutorial, IEEE Geoscience and Remote Sensing Society (IGARSS), hybrid, 2022-07-17
- P. Baumann: *AI-Datacubes in a European Spirit: Federations in Cloud, Edge, and Orbit*. ORDIN SCBD Workshop, virtual, 2022-11-08
- P. Baumann: „rasdaman“ – vom Forschungsprojekt zum Erfolgsmodell für multi-temporale Rasterdaten! Raumfahrt Daten als Schlüsseltechnologie der Zukunft – wie beherrschen wir die Komplexität? 2023-03-30, Darmstadt, DE
- P. Baumann: *New EO-based Climate Services Tailored to the Needs of European Public Authorities*. EXPANDEO (European Association of Remote Sensing Companies Annual Conference), virtual, 2023-06-14
- P. Baumann: *Analysis-Ready AI-Datacubes: How They Support the Green Deal*. Data Week Leipzig, European Open Science Cloud (EOSC) session, 2023-06-26
- P. Baumann: *Advances in Spatio-Temporal Datacube Management, Visualization, Analytics and AI*. Session, IEEE Geoscience and Remote Sensing Society (IGARSS), Pasadena, USA, 2023-07-16
- P. Baumann, O. Campos, D. Misev: *AI-Enabled Analysis-Ready Datacubes: Towards a Roadmap for More Human-Centric Services*. Talk, IEEE Geoscience and Remote Sensing Society (IGARSS), Pasadena, USA, 2023-07-16

### Messestände:

- Intergeo, Hannover, DE, 2021-09-21..23
- Techconnect, Washington DC, USA, 2021-10-18..20
- Geospatial World Forum, Amsterdam, NL, 2021-10-20..22
- ESA Industry Space Days, virtual, 2021-12-07..08
- Industry Space Days, ESA-ESTEC, Noordwijk, Netherlands, 2022-09-28..29
- Intergeo 2022, Essen, DE, 2022-10-18..20

**Impressionen von Veranstaltungen, auf denen AI-Cube prominent vorgestellt wurde:**



Bremen, den 04.03.2024

Bremen, den 04.03.2024

Prof. Dr. Peter Baumann  
Lead, Large-Scale Scientific Information Systems Working Group  
Constructor University

Dr. Peter Baumann  
Geschäftsführer  
Rasdaman GmbH

## Teil III: Erfolgskontrollbericht

### III 1. Aufgabenstellung und Beitrag des Ergebnisses zu den förderpolitischen Zielen

### III 2. Wissenschaftlich-technische Ergebnis des Vorhabens, erreichte Nebenergebnisse und gesammelte wesentliche Erfahrungen

#### **Constructor U:**

- Erweiterungen für Array-Anfragesprachen, um ein breites Spektrum von wissenschaftlichen Berechnungen (insbesondere Iterationen) in der Datenbank auszuführen
- Vertieftes Verständnis um Wege und Herausforderungen, ML dem Nutzer direkt als Datenbank-Operationen auf beliebigen Regionen anzubieten
- Hindernisse auf dem Weg zu Analyse-Bereiten Daten, bedingt durch ML und weitere Faktoren
- Neues Forschungsfeld „model fencing“ zur Verbesserung der Zuverlässigkeit von ML auf Datenwürfeln

#### **rasdaman GmbH:**

- Enge Datenbank-Anbindung für ML und – allgemein – python-Programme zur Auswertung im Server
- Aufmerksamkeit im Markt für kommende AI-Cube Angebote des Unternehmens

### III 3. Fortschreibung des Verwertungsplanes

#### III 3a. Erfindungen/Schutzrechtsanmeldungen und erteilte Schutzrechte

Keine.

#### III 3b. Wirtschaftliche Erfolgsaussichten nach Projektende

Als einziger Wirtschafts-Partner betreibt rasdaman GmbH kommerzielle Umsetzung.

Aus den im Bericht genannten Gründen sind noch gravierende Fragen zu lösen welche sich im Projektverlauf aus der gemeinschaftlichen Forschung ergeben haben. Erst nach Lösung dieser Anschlußfragen plant rasdaman GmbH ein Produktangebot. Um die relevante Forschung voranzutreiben, haben Constructor U und das Unternehmen zwischenzeitlich zusammen mit weiteren internationalen Partnern ein EU-Projekt beantragt (siehe III 3d).

### III 3c. Wissenschaftliche und/oder technische Erfolgsaussichten nach Projektende

In AI-Cube wurde von den beiden Datenwürfel-Partnern die Grundlage für die Anwendung trainierter Modelle in Datenwürfel-Anfragen gelegt und die prinzipielle Machbarkeit an einem speziellen Modell (RSVQA) demonstriert. Auf dieser Basis sind weitere Forschungsfragen entstanden, welche künftig weiter behandelt werden sollen:

- Wie können ML-Modelle als „first-class citizens“ in der Datenbank verwaltet werden? Dies betrifft Ablageformat, notwendige Metadaten, Import und Export, Verwaltung von Zugriffsrechten und vieles mehr. Dies ist insbesondere für eine künftige Kommerzialisierung der AI-Cubes unabdingbar.
- Derzeit wird nur pytorch als ML-Engine unterstützt. Ist es erforderlich, weitere Werkzeuge (etwa TensorFlow) zu unterstützen, würden die Ergebnisse bei ein und demselben Modell und Analysegebiet abweichen? Idealerweise wird der Nutzer nicht mit der Entscheidung über pytorch oder TensorFlow belastet, da kaum jemand die Konsequenzen abschätzen kann. Auch dies ist für einen kommerziellen Einsatz wichtig zu beforschen.
- Die Zuverlässigkeit der Modelle nimmt offenbar rapide ab, sobald sich die Analyse-Region aus der raum/zeitlichen Wohlfühlzone der Modell-Trainingsdaten wegbewegt. Die neue einfache Zugriffsmöglichkeit auf jeglichen Ort, jegliche Zeit, welche die Datenwürfel mit sich bringen, führt damit zu einer erheblichen Gefahr von Falschergebnissen. Wir wollen daher ein Konzept des „model fencing“ erforschen, welches vor Anwendung der Modelle die Anwendbarkeit in der konkreten Anfrage prüft und gegebenenfalls vor der Anwendung warnt. Ein einfaches Beispiel wäre eine Landmaske, welche vor Anwendung eines Buschfeuer-Modells prüft, ob die Region der Anwendung Land oder Wasser adressiert. Pixel über Wasser können z.B. vor der Modellauswertung auf Nullwerte gesetzt werden, sodaß das Modell diese nicht (miß-)interpretiert.

### III 3d. Wissenschaftliche und wirtschaftliche Anschlussfähigkeit

Nach Projektende werden die Arbeiten von AI-Cube bereits aktiv weitergeführt:

- EU FAIRiCUBE (aktiv), Constructor University ist Partner;
- AgriCube: gemeinsamer DFG/MOST-Antrag, eingereicht von Constructor University & Feng Chia University Taiwan;
- EU WorldCube: Förderierte AI-Cubes, Constructor University & rasdaman GmbH im Konsortium zum breiten Einsatz von AI-Cubes für Digital Twins.

In allen Projekten ist eine enge Zusammenarbeit sowohl mit KI-Experten als auch mit Anwendern vorgesehen bzw. realisiert.

## IV. Berichtsblatt:

Ausführende Stelle(n): Constructor University, rasdaman GmbH	
Förderkennzeichen: 50EE2012A, 50EE2012C	Kassenzeichen: keines
Bearbeitung: ja	
<p><b>Ziel</b></p> <p>Im Rahmen des AI-Cube Projekts hatten Constructor University und rasdaman GmbH – in Zusammenarbeit mit dem Verbundpartner Technische Universität Berlin – das Ziel, KI-Methoden mit Datenwürfeln zu integrieren. Speziell im Fokus standen dabei raum-zeitliche Geo-Datenwürfel und deren Analyse mittels der RSVQA-Modelle der TU Berlin.</p>	
<p><b>Inhalt</b></p> <p>Im RSVQA-Ansatz werden Neuronale Netze auf Pärchen von multi-modalen Radar- und optischen Satellitendaten für eine Land-Klassifikation auf Basis der europäischen CORINE-Einteilung trainiert; die Big Earth Net Kollektion der TU Berlin hält hierfür über 500,000 annotierte Sentinel-1 und Sentinel-2 Patches vor.</p> <p>Derartig trainierte Modelle waren von Constructor University und rasdaman GmbH auf Datenwürfeln einzusetzen, indem ein Nutzer über eine Datenwürfel-Anfragesprache beliebige Regionen und Zeiten selektieren kann, auf welche das Modell angewendet werden soll. Dies erlaubt (i) erheblich vereinfachte Nutzung der ML-Modelle und (ii) ihre effiziente Auswertung im Server, also nahe an den Daten. Die dafür benötigten Sentinel-1 und -2 Datenwürfel waren auf Basis von EU DIAS-Archiven aufzubauen. Zusätzlich war die Einbindung in föderierte Datenwürfel-Archive zu untersuchen.</p>	
<p><b>Ausblick</b></p> <p>Nach Projektende werden die Arbeiten von AI-Cube bereits aktiv weitergeführt:</p> <ul style="list-style-type: none"> <li>• EU FAIRiCUBE (aktiv), Constructor University ist Partner;</li> <li>• AgriCube: gemeinsamer DFG/MOST-Antrag, eingereicht von Constructor University &amp; Feng Chia University Taiwan;</li> <li>• EU WorldCube: Föderierte AI-Cubes, Constructor University &amp; rasdaman GmbH im Konsortium zum breiten Einsatz von AI-Cubes für Digital Twins.</li> </ul> <p>In allen Projekten ist eine enge Zusammenarbeit sowohl mit KI-Experten als auch mit Anwendern vorgesehen bzw. realisiert.</p>	

**Aussagen zum konkreten Nutzen**

AI-Cube hat Möglichkeiten, aber auch weiteren Forschungsbedarf bei der Nutzung von ML auf massiven EO-Zeitreihen aufgezeigt. Insbesondere die Einfachheit des Datenzugriffs via Datenwürfel läßt Nutzer von ML in einfacher Weise, ohne es zu wissen, in Regionen geringer Zuverlässigkeit vorstoßen.

**Anwendungsmöglichkeiten des Projekts/ der Projektergebnisse**

- Nahtlose Integration von klassischer Analytik und ML
- Perspektive von ML-Marktplätzen mit einfacher Zero-Coding Anwendung durch Datenwürfel
- Anreicherung der Funktionalität des Datenbankservers um Code, welchen der Nutzer ausführen, jedoch nicht einsehen kann (IP-Schutz)

# Anhang:

## Erweiterungen der Datenwürfel-Anfragesprache

Dieser Anhang ist Teil des Abschlußberichts zu Aufgabe 3.1: Sprachkonzepte für ML-Abfragen in Array-Datenbanken.

### A.1 Hintergrund

Wir benutzen für die folgenden Überlegungen der Einfachheit halber die rasdaman Array-Anfragesprache, *rasql*, welche auch die Basis für ISO IS 9075:2023 SQL/MDA (Multi-dimensional Arrays) bildet; gleichwohl lassen sich die Prinzipien auch auf andere Array-Anfragesprachen anwenden. In diesem Abschnitt wird eine kurze Einführung zu *rasql* gegeben.

Ein *n*-dimensionales *Array* besteht aus *Zellen*, deren jede einen Wert enthält. Jede Zelle wird adressiert durch eindeutige *Koordinaten*, ausgedrückt durch einen integer-Vektor der Länge *n*. Die zulässigen Koordinaten (und damit Zellen) sind gegeben durch die Unter- und Obergrenze eines Intervalls [lo:hi] für jede der *n* Dimensionen.

Alle Operationen der *rasql*-Sprache lassen sich durch zwei Basis-Operationen beschreiben, einen *Array-Konstruktor* und einen *Array-Kondenser*.

**Array-Konstruktor:** Der Array-Konstruktor erlaubt es, *n*-dimensionale Arrays zu erzeugen, deren Inhalt durch einen allgemeinen Ausdruck definiert ist. Damit lassen sich in einer Anfrage neue Arrays aus bestehenden ableiten, z.B. zur die Berechnung eines Histogramms.

Die Grundform des Array-Konstruktors ist wie folgt:

```
MARRAY var IN mintervalExp
VALUES cellExp
```

Der Ausdruck `cellExp` beschreibt, wie das resultierende Array an jedem Punkt des durch `mintervalExp` (für "multi-dimensional interval") definierten Bereichs erzeugt wird; der Ausdruck kann Referenzen auf die Koordinate der aktuellen Array-Zelle enthalten.

Beispiele:

- ein Array mit Ausdehnung [1:100,-50:200] über dem Zelltyp *char*, wobei jede Zelle mit dem Wert 1 initialisiert wird:

```
MARRAY x in [ 1:100, -50:200 ]
VALUES 1c
```

- Ein horizontaler Grauwert-Gradient von Schwarz nach Weiß:

```
MARRAY x in [ 0:255 ], y in [ 0:100 ]
VALUES x
```

- Histogramm für Array *a*:

```
MARRAY bucket in [ 0 : 255 ]
VALUES count_cells( a = bucket )
```

**Array-Kondenser:** Die Operation Condense konsolidiert die Zellenwerte eines mehrdimensionalen Arrays zu einem skalaren Wert auf der Grundlage der angegebenen Kondenser-Operation. Die allgemeine Struktur ist:

```
CONDENSE condenserOp
OVER var IN mintervalExp
[ WHERE booleanExp ]
USING cellExp
```

Sie durchläuft den durch `mintervalExp` angegebenen Bereich und kombiniert dabei die vorgefundenen Werte unter Anwendung `cellExp`. Die `condenserOp` ist eine von `count_cells`, `add_cells`, `avg_cells`, `min_cells`, `max_cells`, `some_cells`, `all_cells`. Die `cellExp` kann Verweise auf `var` enthalten, z.B. die aktuelle Koordinate, die verarbeitet wird.

Kondenser werden häufig in zwei Situationen verwendet: Entweder, um boolesche Arrays in skalare boolesche Werte zu kollabieren, so dass sie in der where-Klausel verwendet werden können, oder aber in Verbindung mit dem marray-Konstruktor, um komplexere Operationen aus Signalverarbeitung und Statistik zu formulieren.

Beispiel:

- Summe über alle Array-Werte, multipliziert mit 2:

```
CONDENSE +
OVER x IN sdom(a)
USING 2 * a[ x ]
```

## A.2 rasql-Erweiterungen für ML

**Dedizierte Funktionen:** Für die Formulierung von ML und andere statistischen Operationen sind einige Funktionen nötig, die entweder nur umständlich in rasql formulierbar sind (z.B. Matrixmultiplikation und -Transposition) oder überhaupt nicht (z.B. Matrix-Inversion, da dies eine inhärent iterative Operation ist und rasql absichtlich keine explizite Iteration enthält, um "sicher in der Auswertung" zu bleiben).

Nach einer Analyse der meistgenutzten Anwendungsfelder für Tensor-Algebra, u.a. Lineare/logistische Regression, Neuronale Feed-Forward Netze, Convolutional Neural Networks (CNN), k-Means Support Vector Machines (SVM) ergab sich eine relative kurze Liste von Kern-Operationen, welche in rasql für Anfragen in diesen Feldern benötigt werden:

- Produkt: `matmul(A, B)`
- Transposition: `transpose(A)`
- Inversion: `inverse(A)`
- Determinante: `det(A)`
- Rang: `rank(A)`
- Spur: `trace(A)`

Weitere weit verbreitete Funktionen sind:

- Sigmoid: `sigmoid(A)`
- Argmax / Argmin: `argmax(A)`, `argmin(A)`
- Softmax: `softmax(A)`
- Sign: `sgn(A)`

Daher ist für die Zukunft geplant, effiziente Implementierungen im Server zu finden, welche insbesondere die gekachelte Speicherung von Arrays berücksichtigt.

### Direktionale Iteration:

Die deklarative Natur von Array-Konstruktor und Kondenser sehen eine Auswertung von Array-Regionen in beliebiger Reihenfolge vor. Dies erlaubt dem Server eine kachelbasierte Auswertung sowie eine Reihe von hochwirksamen Optimierungen. In vielen Algorithmen ist jedoch eine deterministische Reihenfolge erforderlich, um zu garantieren, dass Werte an früher adressierten Koordinaten berechnet wurden und in der aktuellen Iteration verfügbar sind.

Dazu werden die Schlüsselwörter `ASC` und `DESC` eingeführt, welche aufsteigende bzw. absteigende Iteration entlang der Dimension erzwingen, für die sie spezifiziert sind:

```
MARRAY var IN mintervalExp [ASC|DESC]
VALUES cellExp
```

Hiermit läßt sich für Operationen die Iteration in einer bestimmten Reihenfolge erzwingen; beispielsweise könnte ein Array eine Dimension der Verarbeitung haben, durch welche die nächste "Scheibe" aus einer oder mehreren vorhergehenden Scheiben berechnet wird. Beispiele hierfür werden weiter unten angegeben.

Anzumerken ist, daß damit der Konstruktor bzw. Kondenser zu einer "blockierenden Operation" wird, was intern im Server einige Auswirkungen auf die Auswertung des Ausdrucks hat, z.B. hinsichtlich Parallelisierung, Optimierung, etc.

**Rekursiver Zugriff auf das Ergebnis-Array:** Array-Konstruktor und Kondenser iterieren durch den räumlichen Bereich und erzeugen ein Ergebnis, das in der klassischen Form nicht rekursiv von sich selbst abhängig sein kann. Eine solche Rückbezüglichkeit ist jedoch in ML-Algorithmen weit verbreitet. Um dies in der Praxis zu unterstützen, müssen mehrere Aspekte berücksichtigt werden.

- Das Ergebnisarray muß benannt sein, so daß es in `cellExp` selbstbezüglich referenziert werden kann; dazu wird ein `AS` Statement eingeführt:

```
MARRAY var IN mintervalExp ASC AS resultAlias
CONDENSE condenserOp ASC AS resultAlias
```

- Festlegen der Iterationsreihenfolge, in der das Ergebnisfeld mit den im vorherigen Abschnitt eingeführten Schlüsselwörtern `ASC` / `DESC` erzeugt wird.
- Festlegung eines Standardwertes, der zurückgegeben wird, wenn der Zellausdruck auf das Ergebnisfeld außerhalb der Iterationsreihenfolge zugreift, etwa bei einem nicht initialisierten Zellenzugriff:

```
MARRAY var IN mintervalExp ASC AS resultAlias
DEFAULT 0
```

Diese Erweiterung ist in dem Sinne "sicher in der Auswertung" ("safe in evaluation"), dass keine Endlosschleifen entstehen können, da alle Iterationen innerhalb fester Grenzen im Ausdruck stattfinden.

Beispiele:

- Fibonacci-Folge für die ersten `N` Zahlen:

```

MARRAY i IN [0:N] AS result
DEFAULT 0
VALUES CASE
    WHEN i = 1 THEN 1
    ELSE result[i-1] + result[i-2]
END

```

- Lineare Regression unter Verwendung des Gradientenabstiegs; `X`, `y` und `initial\_thetas` sind benutzerdefiniert:

```

MARRAY i IN [1:N] AS thetas
VALUES CASE
    WHEN i = 0 THEN initial_thetas
    ELSE thetas[i-1] -
        matmul(alpha/m,
            matmul(transpose(X),
                matmul(X, thetas[i-1]) - y
            )
        )
END

```

- Logistische Regression unter Verwendung des Gradientenabstiegs; `X`, `y` und `initial\_thetas` sind benutzerdefiniert:

```

MARRAY i IN [1:N] AS thetas
VALUES CASE
    WHEN i = 1 THEN initial_thetas
    ELSE thetas[i-1] -
        matmul(alpha/m,
            matmul(transpose(X),
                sigmoid(matmul(X, thetas[i-1])) - y
            )
        )
END

```

- Feed-forward Netz mit Gradientenabstieg (beachte die Selbstbezüglichkeit von layers sowie die ASC und DESC Angaben):

```

MARRAY i IN [0:10] AS iters
VALUES
(
    -- forward pass
    MARRAY j IN [0:9] ASC AS layers
    VALUES {
        weights:
        CASE
            WHEN i = 0 then initial_weights[j]          -- user-specified
            ELSE iters.weights[i-1,j] -
                matmul(alpha/m,
                    matmul(iters.activations[i-1,j-1], iters.errors[i-1,j]))
        END,

```

```

biases:
CASE
  WHEN i = 0 then initial_biases[j]          -- user-specified
  ELSE iters.biases[i-1,j] -
      matmul(alpha/m, reduceByRow(iters.errors[i-1,j]))
END,
activations:
CASE
  WHEN j = 0 then X                          -- X is user-specified
  ELSE sigmoid(matmul(iters.activations[i,j-1],
                      iters.weights[i,j]) + iters.biases[i,j])
END,
errors: null
}

OVERLAY

-- backward pass
MARRAY j IN [0:9] DESC AS layers
VALUES {
  weights:  null,
  biases:   null,
  activations: null,
  errors:
  CASE
    WHEN j = 9 then iters.activations[i,j] - y    -- y is user-specified
    ELSE matmul(iters.errors[i,j+1], transpose(iters.weights[i,j+1])) *
        sigmoidDerivative(iters.activations[i,j])
  END,
}
)

```

Funktionen, die in der Beispielsanfrage verwendet werden, um sie besser lesbar zu machen:

- ``sigmoid(x)`` ist die Aktivierungsfunktion, die im Netz verwendet wird, um Aktivierungswerte im Bereich  $[0,1]$ , inaktiv-aktiv zu erhalten; entspricht  $1/(1+\exp(-x))$
- ``sigmoidDerivative(x)`` ist die Ableitung der Aktivierungsfunktion; entspricht ``matmul(sigmoid(x), 1 - sigmoid(x))``
- ``reduceByRow(A)`` bildet die induzierte Summe der Zeilen von A; entspricht ``CONDENSE + OVER i in sdom(A)[0] USING A[i, :*]``