Liminal Space Dataset Specification – Version 1.0

1. Einleitung

Die *Liminal Space Dataset Specification v1.0* definiert die vollständige Datenstruktur, Speicherformate, Metadatenstandards, Sampling-Architektur und Organisation der Messdaten, die im Rahmen der experimentellen Performance *Liminal Space* erhoben werden. Das Ziel ist die langfristige wissenschaftliche Nutzbarkeit, Reproduzierbarkeit und Erweiterbarkeit des Datensatzes für Physik, Medienwissenschaft, Sozialforschung, KI und Systemtheorie.

Die Spezifikation ist so ausgelegt, dass sie sowohl für kleine experimentelle Analysen als auch für großskalige Machine-Learning-, Simulation- oder Hybrid-Modelle (z. B. Resonanzmodell / orthogonale Kopplung) geeignet ist.

---

2. Gesamtarchitektur des Datensatzes

Der Datensatz ist modular aufgebaut und trennt zwischen *raw data* (Rohdaten), *processed data* (verarbeitete Daten) und *meta data* (Metadaten & Manifest-Struktur).

2.1 Verzeichnisstruktur

```
/liminal_space_dataset/
│
├── raw/                    # Unveränderte Messdaten (Time Series, Video,
Audio)
│   ├── fusion/             # 600D Multimodaler Fusionsvektor (Parquet,
gechunkt)
│   ├── biosignals/         # Herzrate, HRV, EDA, Atmung
│   ├── motion/             # Realsense Skeleton, IMU, Pose, Position
│   ├── audio/              # Original-Audio der Performance
│   ├── video/              # Kamerafeeds in Zeitstempel-Synchronisierung
│   └── logs/               # Geräte-Logs, Drift-Informationen, Zeitmarker
│
├── processed/              # Daraus abgeleitete Messgrößen
│   ├── M_vectors/          # Reelle Komponente (Struktur)
│   ├── I_vectors/          # Imaginäre Komponente (Antistruktur)
│   ├── features/           # Feature-Vektoren (Entropie, Peaks, RR-
Intervalle, etc.)
│   └── embeddings/         # NLP-, Motion- & Social-Field-Embeddings
│
├── analysis/               # Python/Julia/Matlab-Notebooks & Scripts
│   ├── notebooks/
│   └── scripts/
│
├── docs/                   # Dokumentation
│   └── dataset_specification.pdf
```

```
│
└── dataset_manifest.json    # Zentrale Übersicht & Zeitindexierung
```

---

3. Datenformate

3.1 Parquet (empfohlen für große Zeitreihen)

- Spaltenspeicherformat
- LZ4 oder Snappy-Kompression
- extrem schnelle I/O
- unterstützt fehlende Werte
- maschinenlesbar und ML-optimiert

Formatvorgabe:

```
parquet_format = {
  "compression": "snappy",
  "data_page_version": "2.0",
  "use_dictionary": true
}
```

3.2 CSV (nur für Beispiel- und Debug-Daten)

- UTF-8 ohne BOM
- Dezimalpunkt
- kein Separator in Headern
- maximal 100k Zeilen empfohlen

3.3 Videoformate

- **Master**: ProRes422 oder DNxHR HQ
- **Proxy**: H.264/H.265 (YUV 4:2:0)
- **Timecode**: Framerate-stabil mit Drop-Frame Indikator

3.4 Audio

- WAV, 48kHz, 24bit
- Mono + Binauraler Referenzmix

---

4. Der multimodale Fusionsvektor (600 Dimensionen)

Der zentrale Messvektor besteht aus 600 synchronisierten Dimensionen und ist das Herzstück des Datensatzes.

4.1 Struktur des Fusionsvektors

```
Fusion(t) = [ Motion(0..99), Bio(0..49), Geo(0..49),
              Pose(0..149), Context(0..49), NLP(0..199) ]
```

Eine mögliche Aufteilung:

Block	Dim	Beschreibung
Motion	100	IMU + Realsense Vektoren (XYZ, Winkel, Gelenkpfade)
Biosignals	50	HR, HRV, EDA, Respiration, Temperatur
Geometry	50	Raumkoordinaten der Tänzerin + Objektdistanz
Pose	150	Pose-Estimation + Skeleton Joints
Context	50	Lichtwerte, Lautstärke, Szenenindex
NLP/LLM	200	Embedding der gesprochenen oder relevanten Texte

---

5. Chunking-System

Zur effizienten Analyse wird der Datensatz in gleich lange Pakete unterteilt.

5.1 Chunk-Spezifikation

- **Größe pro Chunk:** 1000 Samples (bei 50 Hz = 20 Sekunden)
- **Dateinamen:**
- `fusion_000.parquet`
- `fusion_001.parquet`
- `fusion_002.parquet`

5.2 Warum Chunking?

- speicherschonend
- schnelle Random-Access-Operationen
- Training von ML-Modellen ohne Out-of-Memory
- parallele Verarbeitung auf CPU/GPU

---

6. Zeitstempel-Synchronisation

Alle Messgeräte müssen auf eine gemeinsame Zeitbasis synchronisiert sein.

Synchronisationspipeline:

1. NTP-Sync aller Geräte
2. Audio-Clap oder Timecode-Signal als physischer Marker
3. Drift-Korrektur über lineare Regression der Offsets

4. Normalisierung aller Daten auf T in Millisekunden

Format:

```
"t": UNIX-time (ms)
```

---

7. Der dataset_manifest.json Standard

Ein maschinenlesbares Manifest führt alle Dateien und Parameter zusammen.

Beispiel:

```
{
  "dataset": "Liminal Space v1.0",
  "sampling_rate": 50,
  "dimensions": 600,
  "chunks": [
    {"file": "fusion_000.parquet", "start": 0, "end": 999},
    {"file": "fusion_001.parquet", "start": 1000, "end": 1999}
  ],
  "video": {
    "cam1": "video/cam1_master.mov",
    "cam1_proxy": "video/cam1_proxy.mp4"
  },
  "sensors": {
    "imu": "biosignals/imu_stream.parquet",
    "hr": "biosignals/hr_stream.parquet"
  }
}
```

---

8. Metadaten & Logging

Zu jedem Datensatz gehören:

- Sensor-IDs
- Firmware-Versionen
- Fehlerlogs
- Raumtemperatur
- Licht-Szenen
- Operator-Notizen
- Szenenwechselmarker

Metadatenformat:

- JSON oder YAML

- flach strukturiert
- maschinenlesbar

---

9. Qualitätskontrolle (QC)

Jeder Chunk erhält QC-Kennzahlen:

- Missing-Rate pro Dimension
- Standardabweichung vs. Normbereich
- Drift-Score
- Synchronisationsgüte
- Signal-Rausch-Verhältnis

QC wird im Ordner:

```
processed/features/QC/
```

abgelegt.

---

10. Schlussbemerkung

Diese Spezifikation stellt die erste vollständige Version der Datenarchitektur für das *Liminal Space Social Field Measurement* dar. Sie ermöglicht die wissenschaftliche Weiterverwendung, Reproduzierbarkeit und interdisziplinäre Anschlussfähigkeit in Physik, Medienwissenschaft, KI, Soziologie und Performance Studies.

Version 1.1 wird in Zukunft um konkrete Messgeräte, Formate für Video-Depth-Fusion und operatorische Anleitungen ergänzt.