Phase 2 — Kinematic-Constrained Multi-Camera Solver

1 Ziel

Nach Phase 1 existiert:

- ein gemeinsames Weltkoordinatensystem
- Kameraposen aller Kameras
- bekannte absolute Markerpositionen
- fusionierte Beobachtungen mehrerer Kameras

Phase 2 erweitert das System um:

- Rigid-Body Constraints
- mechanische Zusammenhänge
- Gelenke
- relative Marker-Geometrien (`relPos`)
- Stabilisierung bei wenigen sichtbaren Markern

2 Ausgangslage

Aktuell wird jeder Marker unabhängig behandelt.
Das ist suboptimal, weil:

- viele Marker nur kurz sichtbar sind
- oft nur 1–2 Marker eines Bauteils sichtbar sind
- solvePnP bei wenigen Markern instabil wird
- Markerrauschen direkt in die Weltkoordinaten eingeht

Mechanisch sind die Marker jedoch nicht unabhängig.
Mehrere Marker gehören jeweils zu:

- Arm1
- Arm2
- Joint1
- Base
- Finger1
- Finger2

und bilden jeweils starre Körper (Rigid Bodies).

3 Grundidee

Statt einzelne Marker zu lösen:

```
Marker -> Welt
```

wird gelöst:

```
RigidBody -> Welt
```

und daraus:

```
Marker = RigidBody * relTransform
```

4 Vorteil

Schon ein einzelner sichtbarer Marker kann:

- einen ganzen Körper stabilisieren
- andere unsichtbare Marker indirekt bestimmen

Beispiel:
Wenn Marker 198 sichtbar ist und Marker 229 relativ dazu bekannt ist, dann kann Marker 229 geschätzt werden, auch wenn er aktuell unsichtbar ist.

5 Erwartete Verbesserungen

5.1 Stabilität

Deutlich stabilere Pose-Schätzung bei:

- Motion Blur
- wenigen sichtbaren Markern
- schlechten Blickwinkeln
- Teilverdeckungen

5.2 Konsistenz

Marker eines Bauteils bleiben:

- geometrisch korrekt
- starr
- ohne unrealistische Verzerrungen

5.3 Multi-Camera-Verkettung

Kameras können indirekt gekoppelt werden.
Beispiel:
Cam1 sieht:

- Marker 1,2,3

Cam2 sieht:

- Marker 3,198

Cam3 sieht:

- Marker 198,229

Dadurch wird:

- Arm1 relativ zur Welt bestimmbar
- obwohl keine einzelne Kamera alles sieht

6 Benötigte Daten

6.1 Bereits vorhanden

6.1.1 Absolute Marker

`"position":[x,y,z]`

für Board-Marker.

6.1.2 Relative Markerpositionen

`"relPos":[x,y,z]`

für Marker auf einem Rigid Body.

6.1.3 Body-Zuordnung

`"on":"Arm1"`

6.1.4 Gelenke

```
"type":"revolute"
"axis":[1,0,0]
```

7 Noch fehlende Daten

7.1 Marker-Orientierung relativ zum Body

Aktuell existiert nur:

`"relPos"`

Empfohlen wird zusätzlich:

`"relRot":[rx,ry,rz]`

oder alternativ:

```
"normal":[x,y,z]
"up":[x,y,z]
```

Denn Marker besitzen nicht nur Position, sondern auch Orientierung.
Das verbessert spätere Pose-Fits deutlich.

8 Geplante Solver-Strategie

8.1 Phase 2A — Rigid Body Fit

Zunächst:

- pro Element einen starren Körper fitten
- noch keine Gelenkoptimierung

Beispiel:

`T_world_arm1`

wird geschätzt.
Alle Marker von Arm1 folgen daraus automatisch.

8.2 Phase 2B — Joint Constraints

Danach:

- Gelenkachsen erzwingen
- Rotationen einschränken
- mechanische Grenzen verwenden

Beispiel:

`Arm2 darf sich nur um Y drehen`

8.3 Phase 2C — Global Optimization

Später optional:

- vollständiges Bundle Adjustment
- gleichzeitige Optimierung aller:
  - Kameras
  - Marker
  - Bodies
  - Gelenkwinkel

9 Wichtige Architekturentscheidung

NICHT:

```
Marker einzeln lösen
```

SONDERN:

```
Bodies + Constraints lösen
```

Marker werden damit Beobachtungen, nicht mehr primäre Zustände.

10 Geplante Datenstruktur

10.1 Weltpose eines Körpers

```
{
  "body":"Arm1",
  "worldPose":{
    "position":[x,y,z],
    "rotationMatrix":[...]
  }
}
```

10.2 Relative Markerdefinition

```
{
  "id":198,
  "on":"Arm1",
  "relPos":[x,y,z],
  "relRot":[rx,ry,rz]
}
```

11 Phase 1 Reminder

Phase 1 bleibt weiterhin wichtig:

- alle Kameras finden
- alle Detection-Dateien laden
- gemeinsame Marker fusionieren
- Weltkoordinaten berechnen
- Qualitätsmetriken speichern

- auch schlechte / unvollständige Beobachtungen abspeichern

Die Ergebnisse von Phase 1 dienen als Eingang für Phase 2.

12 Zielbild

Langfristig entsteht:

- ein globales Robotermodell
- mit mehreren Kameras
- mehreren Rigid Bodies
- Gelenken
- Unsicherheiten
- Qualitätsmetriken
- temporaler Stabilisierung

basierend auf:

- ArUco-Beobachtungen
- Mechanik
- Kinematik
- Multi-View-Geometrie