

Mathematische Beschreibung von `3_fuse_markers_world.py`

1 Ziel

Das Script fusioniert mehrere ArUco-Detektionen aus mehreren Kameras zu einem gemeinsamen Weltmodell.

Gegeben sind:

- Kameraposen im Weltkoordinatensystem
- 2D Marker-Detektionen pro Kamera
- Kameraintrinsics
- Markergröße

Gesucht sind:

- Weltposition aller Marker
- Markerorientierungen
- Qualitätsmetriken

2 Koordinatensysteme

Verwendete Systeme:

- Weltkoordinatensystem W
- Kamerakoordinatensystem C
- Markerkoordinatensystem M

3 solvePnP

Für jeden Marker wird mittels OpenCV solvePnP berechnet:

$$X_C = R_{CM}X_M + t_{CM}$$

Dabei gilt:

- R_{CM} = Rotation Marker \rightarrow Kamera
- t_{CM} = Translation Marker \rightarrow Kamera

4 Kamerapose

Aus der vorher berechneten Kamerapose:

$$X_W = R_{WC}X_C + t_{WC}$$

mit:

$$t_{WC} = C_W$$

(Kameraposition in Weltkoordinaten)

5 Markerposition in Weltkoordinaten

Markerzentrum:

$$p_M = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Markerposition im Kamerasystem:

$$p_C = R_{CM}p_M + t_{CM}$$

Da $p_M = 0$:

$$p_C = t_{CM}$$

Transformation ins Weltkoordinatensystem:

$$p_W = R_{WC}p_C + t_{WC}$$

Somit:

$$p_W = R_{WC}t_{CM} + t_{WC}$$

6 Markerrotation in Weltkoordinaten

Die Markerrotation relativ zur Kamera:

$$R_{CM}$$

Die Kamerarotation relativ zur Welt:

$$R_{WC}$$

Markerrotation relativ zur Welt:

$$R_{WM} = R_{WC}R_{CM}$$

Dies ist die zentrale Rotationsgleichung des Scripts.

7 Gewichtung der Beobachtungen

Mehrere Kameras können denselben Marker beobachten.

Für jede Beobachtung wird ein Gewicht berechnet:

$$w_i = w_{\text{confidence}} \cdot w_{\text{area}} \cdot w_{\text{view}} \cdot w_{\text{reprojection}}$$

Typische Faktoren:

- Marker Confidence
- Markergröße in Pixel
- Sichtwinkel
- Distanz zum Bildrand
- Reprojektionsfehler

8 Gewichtete Positionsfusion

Die endgültige Markerposition:

$$p = \frac{\sum_i w_i p_i}{\sum_i w_i}$$

Dies entspricht einem gewichteten Mittelwert.

9 Rotationsfusion

Rotationen werden gesammelt:

$$R_1, R_2, \dots, R_n$$

Eine einfache erste Näherung:

- Eulerwinkel mitteln
- oder Quaternionen mitteln

Später empfohlen:

- SVD-basierte Rotationsmittelung
- Lie-Group Mittelung auf $SO(3)$

10 Qualitätsmetriken

Das Script berechnet:

- Anzahl beobachtender Kameras
- Positionsstreuung
- Reprojektionsfehler
- Gesamtgewicht

- Sichtwinkel

Beispiel:

$$\sigma = \sqrt{\frac{1}{N} \sum_i \|p_i - \bar{p}\|^2}$$

11 Rigid-Body Erweiterung

Später können Marker über bekannte Relativpositionen gekoppelt werden.

Beispiel:

$$p_{M2} = p_{M1} + R_{Body} \Delta p$$

Dadurch können Marker rekonstruiert werden, selbst wenn sie nicht direkt sichtbar sind.

12 Spätere Erweiterungen

Geplant:

- Bundle Adjustment
- Kinematic Constraints
- Joint Solver
- Graph Optimization
- Temporal Tracking