

## **main\_App.mlapp**

Diese MATLAB-Code definiert eine App `main_App`, die für die Bildverarbeitung verwendet wird. Die App-Oberfläche enthält verschiedene UI-Komponenten wie Schaltflächen und Achsen, um unterschiedliche Operationen auszuführen.

### **Eingabe:**

- **Bildauswahl:** Der Benutzer wählt ein Bild aus, das in der GUI angezeigt und verarbeitet wird.
- **Punktdefinition:** Der Benutzer definiert fünf Punkte auf dem ausgewählten Bild.
- **Reset:** Der Benutzer kann die definierten Punkte zurücksetzen.
- **Transformation starten:** Der Benutzer kann die Transformation basierend auf den definierten Punkten starten.

### **Ausgabe:**

- **Bildanzeige:** Das ausgewählte Bild wird in der GUI angezeigt.
- **Punkte und Linien:** Die definierten Punkte und die daraus resultierenden Linien werden auf dem Bild angezeigt.
- **3D-Transformation:** Die Transformation des Bildes wird als 3D-Szene angezeigt.

## **CalculatePointCoordinates.m**

Diese MATLAB-Funktion `CalculatePointCoordinates` berechnet acht Nicht-Kontrollpunkte, die später verwendet werden können, um das gegebene Bild zu schneiden.

### **Eingabe:**

- `image`: Ein RGB- oder Graustufenbild.
- `control_pts`: Eine Matrix, die fünf Kontrollpunkte enthält:
  - `cpt_1`: Kontrollpunkt oben links.
  - `cpt_2`: Kontrollpunkt oben rechts.
  - `cpt_3`: Kontrollpunkt unten rechts.
  - `cpt_4`: Kontrollpunkt unten links.
  - `vp`: Fluchtpunkt in der Mitte.

### **Ausgabe:**

- pts: Eine Matrix, die acht berechnete Nicht-Kontrollpunkte enthält.

## ImageCropping.m

Diese MATLAB-Funktion `ImageCropping` schneidet ein Bild basierend auf einem gegebenen Polygon aus.

### Eingabe:

- image: Das Originalbild (kann RGB oder Graustufen sein).
- polygon: Eine Matrix, die die x- und y-Koordinaten der Eckpunkte des Polygons enthält.

### Ausgabe:

- img\_cropped: Das zugeschnittene Bild, das nur den Bereich innerhalb des Polygons enthält.

## ProjectiveRectification.m

Diese MATLAB-Funktion `ProjectiveRectification` führt eine projektive Transformation auf fünf Bildern basierend auf angegebenen Kontroll- und Nicht-Kontrollpunkten durch. Ziel ist es, die Bilder so zu transformieren, dass sie in einem 3D-Raum richtig ausgerichtet sind.

### Eingaben:

- control\_pts: Eine 2x5-Matrix von Kontrollpunkten, wobei jede Spalte einen Kontrollpunkt darstellt.
- non\_control\_pts: Eine 2x8-Matrix von Nicht-Kontrollpunkten, wobei jede Spalte einen Nicht-Kontrollpunkt darstellt.
- img\_front, img\_left, img\_right, img\_top, img\_bottom: Die fünf zu transformierenden Bilder.
- f: Die Brennweite.

### Ausgaben:

- img\_front\_rectified, img\_left\_rectified, img\_right\_rectified, img\_top\_rectified, img\_bottom\_rectified: Die transformierten Bilder.

- geo: Ein Vektor, der die berechnete Tiefe, Höhe und Breite des 3D-Bildes enthält.

## **Transform3D.m**

Diese MATLAB-Funktion Transform3D erstellt eine 3D-Szene und platziert ein Zielbild sowie sechs Flächenbilder (vorne, links, rechts, oben, unten) auf den entsprechenden 3D-Ebenen.

### **Eingaben:**

- x, y: Koordinaten zur Platzierung des Zielbildes.
- img\_target: Zielbild.
- img\_front: Bild der Vorderseite.
- img\_left: Bild der linken Seite.
- img\_right: Bild der rechten Seite.
- img\_top: Bild der Oberseite.
- img\_bottom: Bild der Unterseite.

### **Augaben:**

- Erstellung eines 3D-Fensters
- Platzierung des Zielbildes in der 3D-Szene
- Platzierung der Gesichterbilder in der 3D-Szene
- Achsen- und Gittereinstellungen
- Maus-Scroll-Rad-Ereignis
- Aktualisierung der Ansicht