

kiegészíti a Matplotlib, melynek segítségével különböző ábrákat készíthetünk a tömbjeinkről. Egy harmadik rendkívül hasznos könyvtárcsalád a scikit, ami számos tudományos számításához szükséges alkönyvtárt foglal össze. A scikit-image képek kezelésére, a scikit-learn gépi tanulás algoritmusok használatára, míg a scikit-fuzzy fuzzy logika használatára ad lehetőséget. Ezek a könyvtárak tulajdonképpen együttesen kiadják a Matlab funkcionalitásának jelentős részét.

3. Mérési feladatok

A mérés folyamán az alábbi feladatokat kell elvégezni:

1. Készítsen eljárást egy adott objektum 2D-ben történő követésére egy RGB-D kamera mélységképe alapján!
2. Készítsen eljárást egy adott objektum szín alapú követésére a HSV színtérben!
3. Határozza meg az adott objektum koordinátáit a 3D térben!
4. Ellenőrizze az algoritmus helyes működését előre felvett videófelvételek, valamint élő videó segítségével!

4. Hasznos kódrészletek

Részlet kivágása a képből

```
array[y1:y2, y1:y2]
```

Küszöbözés adott tartományban

```
roiMask = cv2.inRange(depthRoi, np.array([minval]), np.array([maxval]))
```

Egy másikkal egyező méretű csupa nulla kép készítése

```
mask = np.zeros_like(binary)
```

Kontúrok keresése

```
_, contours, _ = cv2.findContours(binary, cv2.RETR_EXTERNAL,  
cv2.CHAIN_APPROX_SIMPLE)
```

Kontúr területének számítása

```
area = cv2.contourArea(cont)
```

Kontúr rajzolása:

```
cv2.drawContours(mask, contours, maxInd, 255, -1)
```

Momentumok számítása

```
moments = cv2.moments(contours[maxInd])
```

HSV konverzió

```
imgHsv= cv2.cvtColor(imgRoi, cv2.COLOR_BGR2HSV)
```

Kép maszkolása

```
maskedImgHsv = cv2.bitwise_and(image, image, mask=mask)
```