

Structure from Motion from Two Views

1. fejezet

Algoritmus

A Structure from motion (SfM) folyamat segítségével 3D rekonstrukciót hajthatunk végre egy képpár segítségével.

1. Két kép közötti ritka ponthalmazok megfeleltetése (pontmegfeleltetés): az első kép sarkainak azonosítása a *detectMinEigenFeatures* függvénnyel, majd azok követése a második képre a *vision.PointTracker* segítségével.
2. Az esszenciális mátrix becslése *estimateEssentialMatrix* használatával.
3. Kamera elmozdulásának kiszámítása *estrelpose* függvénnyel.
4. Két kép közötti sűrű ponthalmazok megfeleltetése (pontmegfeleltetés): több pont kinyeréséhez újra kell detektálni a pontokat a *detectMinEigenFeatures* függvény segítségével a '*MinQuality*' opciót használva. Ezt követi a sűrű ponthalmaz követése a második képre a *vision.PointTracker* használatával.
5. Az illeszkedő pontok 3D helyzeteinek meghatározása a *triangulate* segítségével (háromszögelés).

2. fejezet

Kód magyarázata

2.1. Képpár betöltése

1. *fullfile(string1, string2, ...)* = az argumentumként kapott stringekből összeállít egy elérési útvonalat, pl.:

```
path = fullfile('myfolder', 'mysubfolder')  
path = 'myfolder\mysubfolder\'
```

toolboxdir(toolbox) = visszaadja az argumentumként kapott toolbox abszolút elérési útvonalát.

2. *imageDatastore(path)* = létrehoz egy ImageDatastore objektumot a kapott elérési útvonallal meghatározott képekből. Az ImageDatastore objektum segítségével egy mappában található összes képet össze lehet gyűjteni egy változóba (de alapból nem lesz az összes kép egyszerre betöltve).
3. *readimage(datastore, n)* = betölti az n. képet a megadott datastore-ból.
4. *figure* = létrehoz egy új, üres ábra ablakot.
5. *imshowpair(image1, image2, 'montage')* = a meghatározott két képet egymás mellé helyezi a legutolsó ábrán.
6. *title('string')* = hozzáad egy címet a legutolsó ábrához.

2.2. A Camera Calibrator alkalmazás segítségével előre kiszámolt kamera paraméterek betöltése.

1. `load(file_name.mat)` = betölti egy korábban elmentett workspace adatait a jelenlegi workspace-be. A workspace egy ideiglenes tároló amely a MATLAB elindítása óta létrehozott változókat tárolja. Alapértelmezetten a MATLAB ablak jobb oldalán látható. A workspace-t el lehet menteni, így a benne tárolt változókat később vissza lehet tölteni a MATLAB-ba.

2.3. Lencse által okozott torzítás eltávolítása.

1. `undistortImage(image, intrinsics)` = a második argumentumként megadott kamera paramétereket felhasználva eltünteti a kamera lencséje által okozott torzítást a megadott képről.

A kamera kalibrációja során kapott kamera paramétereket és a torzítási együtthatókat felhasználva kiszámítjuk a bemeneti kép minden pixelének eredeti pozícióját. Az egyes pixelek pozícióját az alábbi torzítások módosítják:

- **Radiális torzítás** = kiváltó oka, hogy a lencse szélén áthaladó fény jobban törik, mint a lencse közepén környezetében áthaladó fény. Ez kiszámolható:

$$x = x_d(1 + k_1r^2 + k_2r^4)$$

$$y = y_d(1 + k_1r^2 + k_2r^4)$$

Ahol x, y = korrigált koordináták; x_d, y_d = torzított koordináták; k_1, k_2 = radiális torzítási együtthatók; $r^2 = x_d^2 + y_d^2$

- **Tangenciális fordítás** = előfordul, ha a kameraszensor és a lencse nem állnak tökéletesen párhuzamosan. Ez kiszámolható:

$$x = 2p_1x_dy_d + p_2(r^2 + 2x_d^2)$$

$$y = 2p_2x_dy_d + p_1(r^2 + 2y_d^2)$$

Ahol x, y = korrigált koordináták; x_d, y_d = torzított koordináták;
 p_1, p_2 = tangenciális torzítási együtthatók; $r^2 = x_d^2 + y_d^2$

3. fejezet

Forrás

- <https://www.mathworks.com/help/vision/ug/structure-from-motion-from-two-views.html>
- <https://www.mathworks.com/help/visionhdl/ug/image-undistort.html>