

# MOZGÁSVIZSGÁLAT VIDEÓ FELVÉTELEK ÉS FÉNYKÉPEK ALAPJÁN

*Siki Zoltán*

2022.10.25.



BUDAPESTI MŰSZAKI  
ÉS GAZDASÁGTUDOMÁNYI EGYETEM  
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Általános és Felsőgeodézia Tanszék



**OSGeo Budapest**  
Your Open Source Compass

# Áttekintés



- Ulyxes rendszer bemutatása
- Mozgásvizsgálat digitális képekből
- Kamera kalibráció
- Alkalmazási példák
- Telepítés, beállítás
- Programok kipróbálása



*And the best part is, it's open source!*

# Ulyxes



- Három az egyben
  - Kutatás és fejlesztés (kávé)
  - Ipari alkalmazások (tej)
  - Oktatás (cukor)
- Részei (alkönyvtárai)
  - pyapi építőkövek (osztályok)
  - pyapps mint alkalmazások
  - camera videó és képfeldolgozás
- <https://github.com/zsiki/ulyxes>
- <http://www.agt.bme.hu/ulyxes>



## Contributors 7



## Languages



● Python 67.5%	● Tcl 15.2%
● C++ 13.0%	● PHP 2.1%
● Makefile 0.6%	● Clarion 0.6%
● Other 1.0%	

# Mozgásvizsgálat digitális képekből



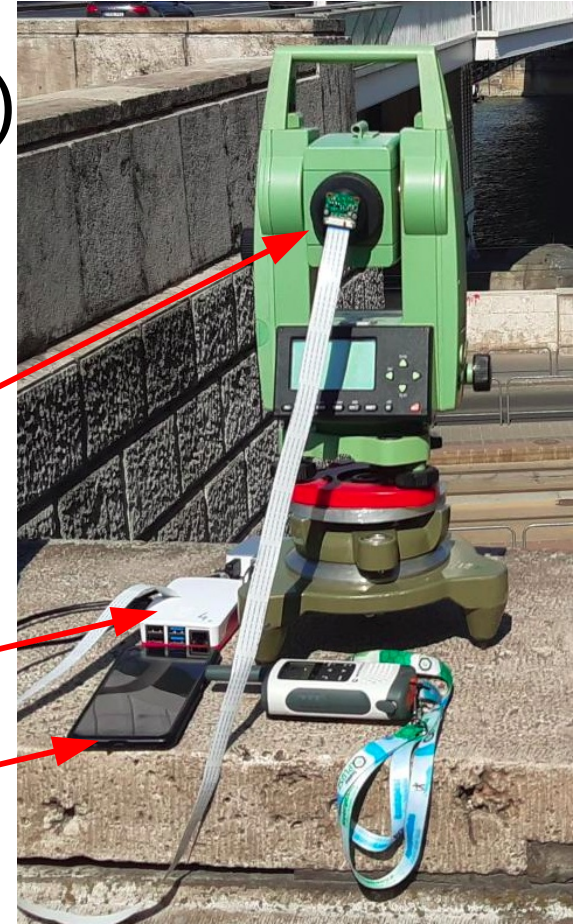
- Gyors mozgások követése (10-30 fps)
- Költséghatékony
- Geodéziai távcső közbeiktatása



Raspberry Pi  
kamera modul 8 MP  
fénykép 3280 x 2464  
videó 1632 x 1216

Raspberry Pi 4

Mobiltelefon  
(WiFi router)



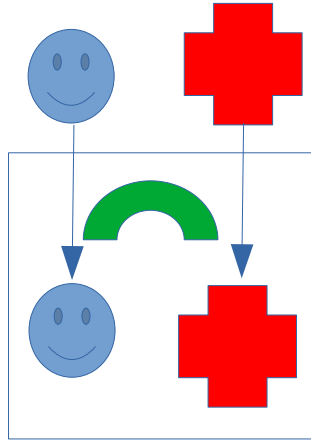
# Mintakeresési módszerek



## Mintaillesztés (template matching)

+

Mindig van találat  
Egyszerű algoritmus  
Nem kell speciális jelet feltenni



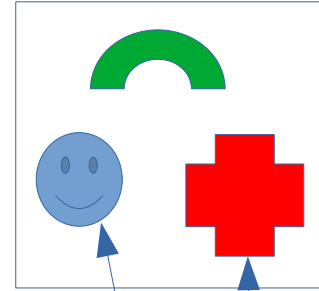
-

Hamis találat esélye nagyobb  
Csak minimális elfordulás  
Csak minimális méret változás

## Alakfelismerés (pattern recognition)

+

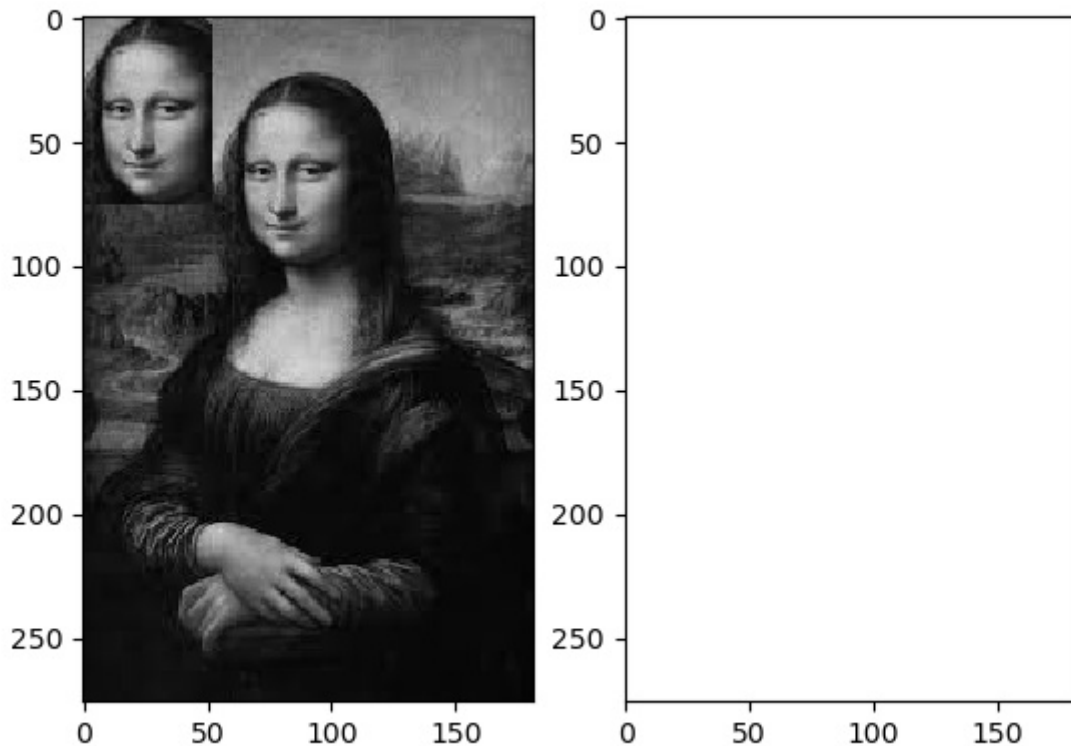
Minta elfordulhat  
Minta mérete megváltozhat  
Írány becsülhető (3D)



-

Fényviszonyokra érzékenyebb  
Speciális jelet kell elhelyezni

# Mintaillesztés (template matching)



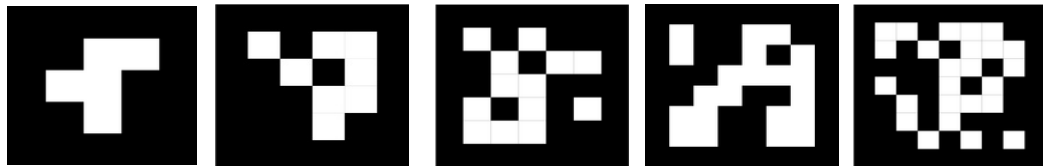
$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2 \quad \min$$

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}} \quad \min$$

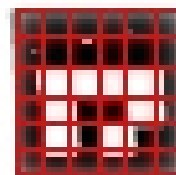
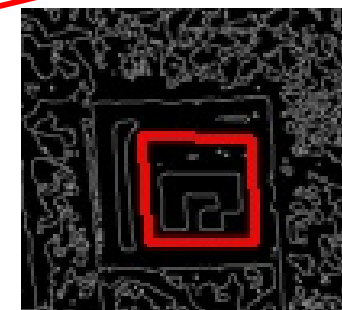
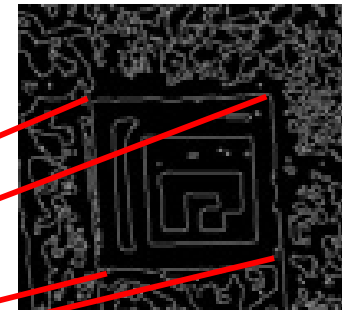
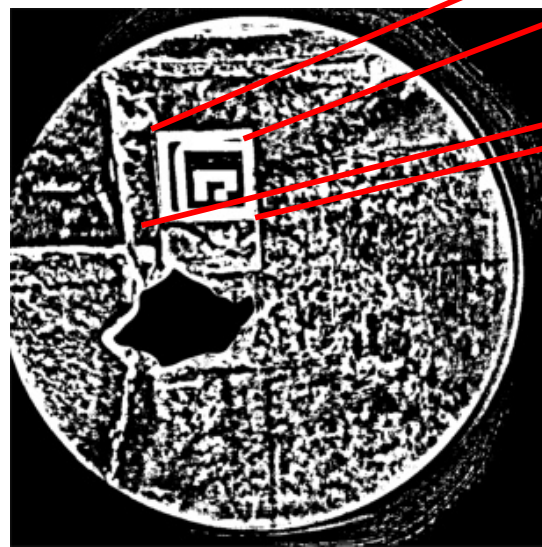
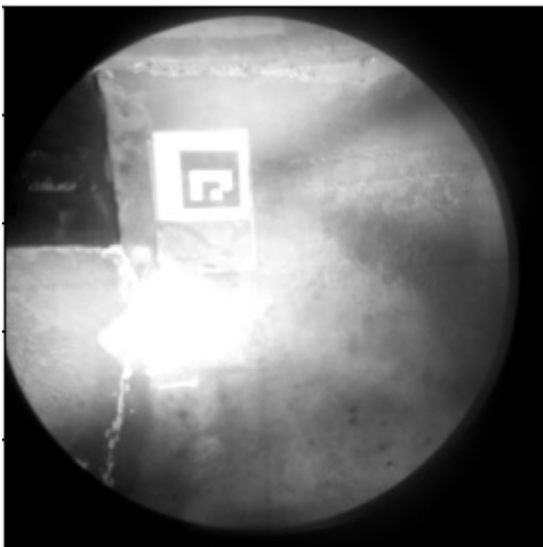
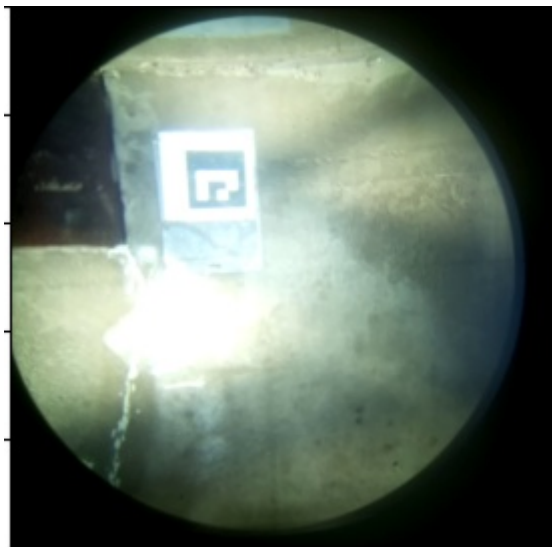
$$R(x, y) = \sum_{x', y'} (T(x', y') \cdot I(x + x', y + y')) \quad \max$$

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}} \quad \max$$

# Alakfelismerés



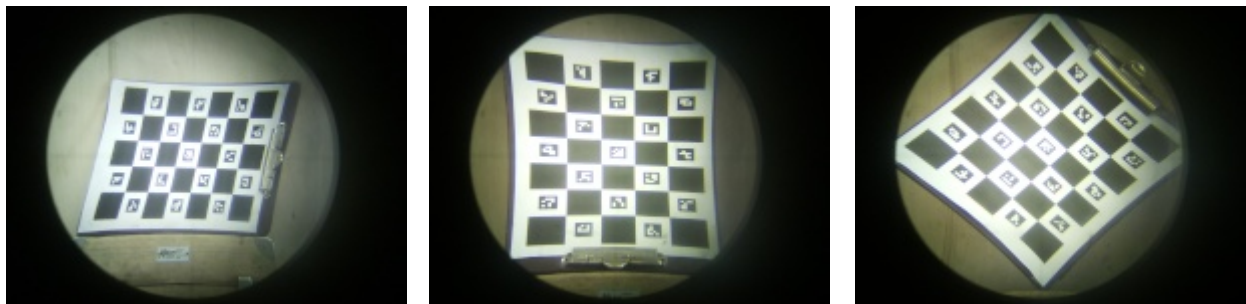
ArUco kódok



# Kamera kalibráció



ChArUco tábla (12-20 felvétel)



$$\begin{bmatrix} 27785.30 & 0.0 & 1397.28 \\ 0.0 & 26618.89 & 1458.82 \\ 0.0 & 0.0 & 1 \end{bmatrix}$$

$$k = [75.00 \quad -605.21 \quad 5.79]$$

$$p = [0.61 \quad -0.64]$$

Radiális torzítás

$$x_{\text{torzított}} = x \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6)$$

$$y_{\text{torzított}} = y \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6)$$

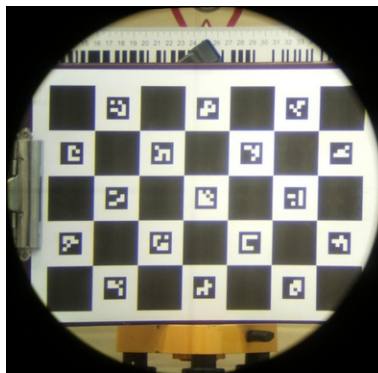
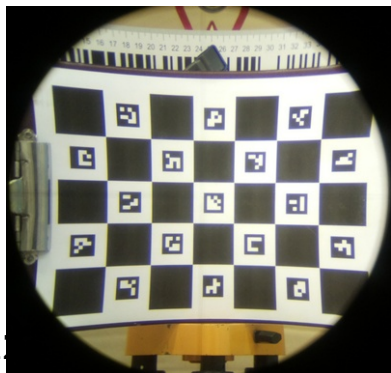
Tangenciális torzítás

$$x_{\text{torzított}} = x + (2 \cdot p_1 \cdot x \cdot y + p_2 \cdot (r^2 + 2 \cdot x^2))$$

$$y_{\text{torzított}} = y + (p_1 \cdot (r^2 + 2 \cdot y^2) + 2 \cdot p_2 \cdot x \cdot y)$$

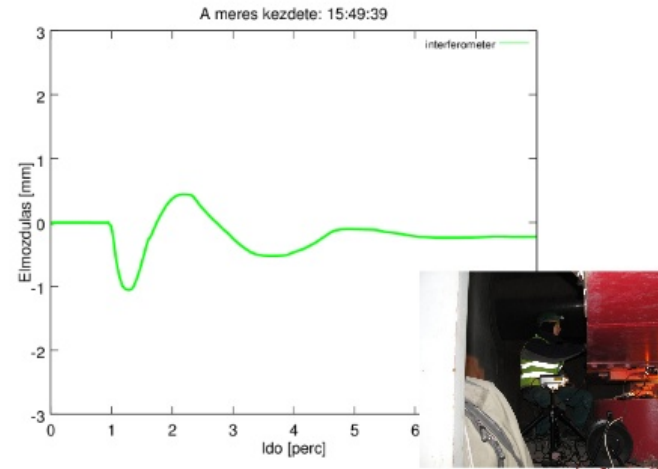
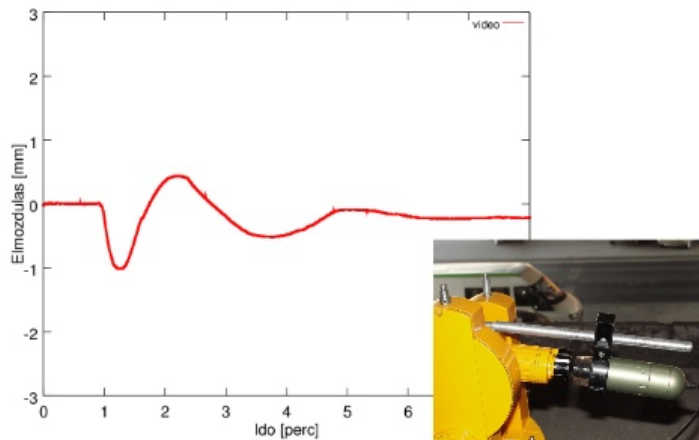
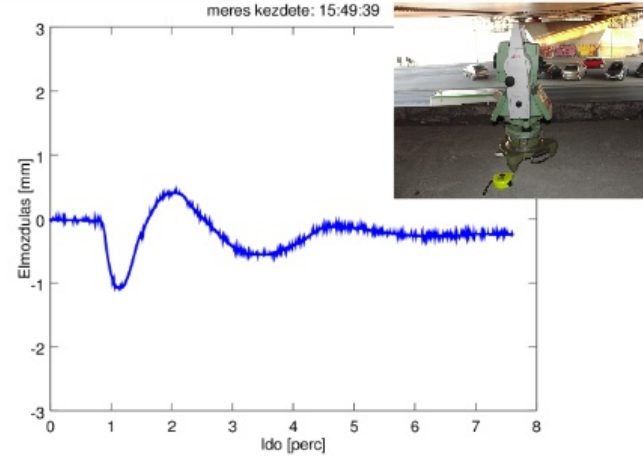
Kamera mátrix

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

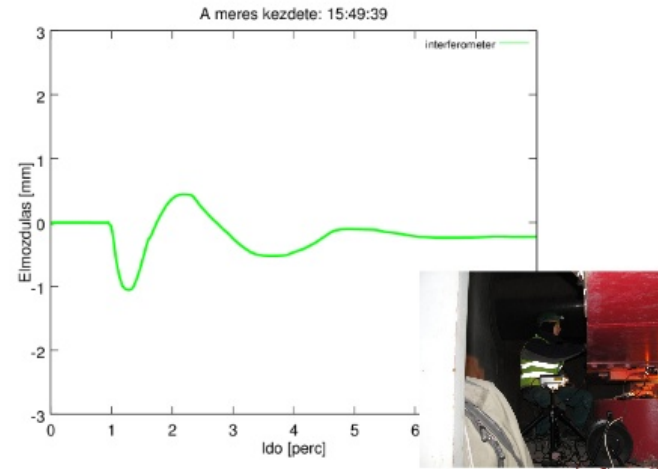
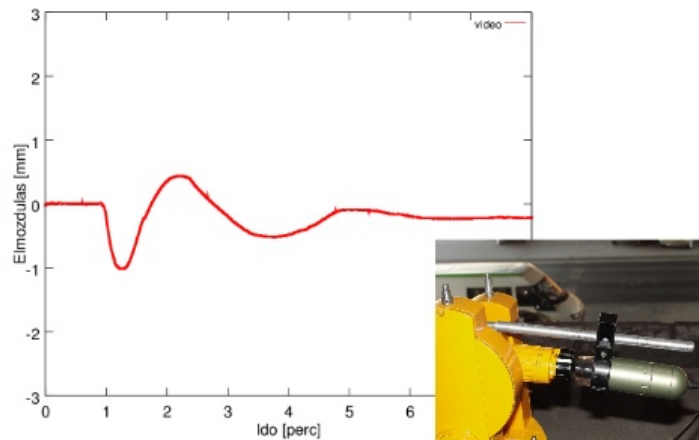
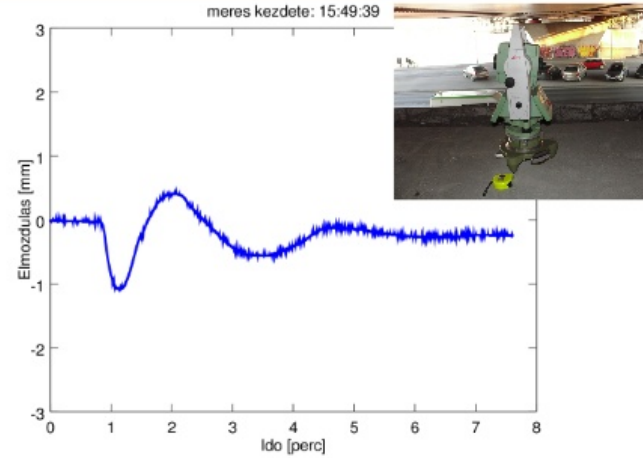




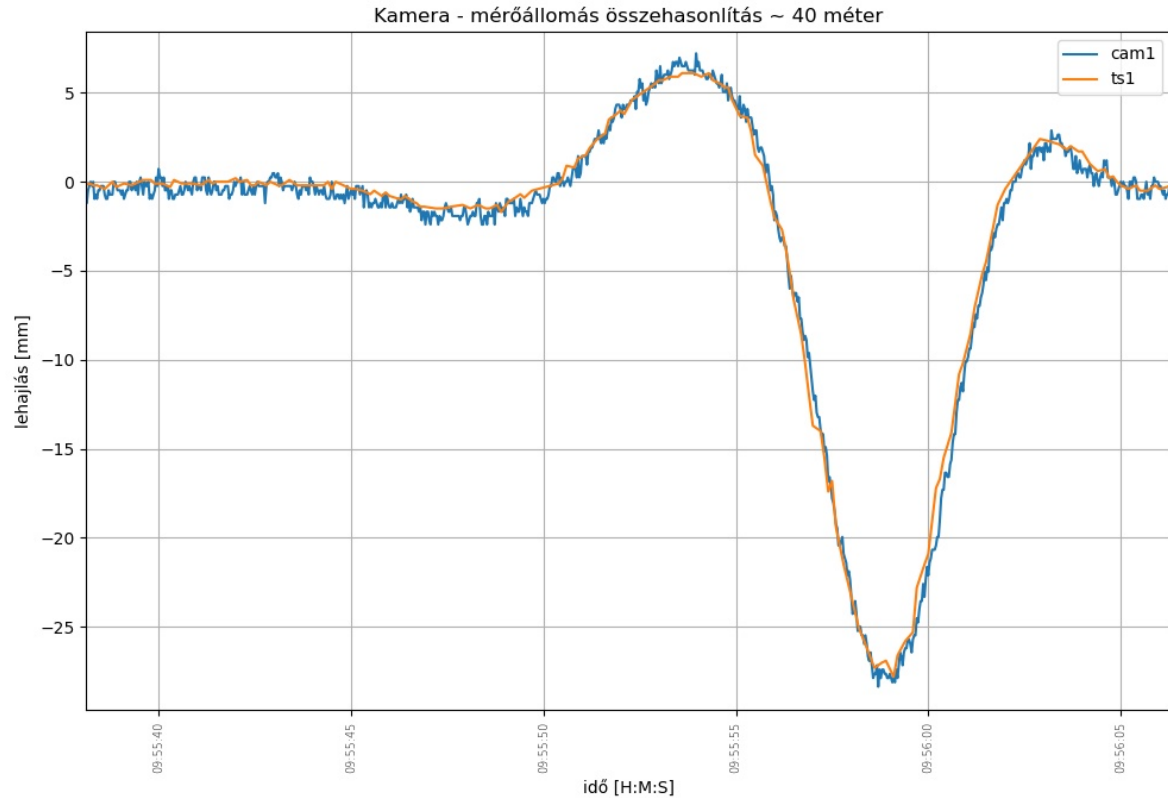
# Alkalmazási példa



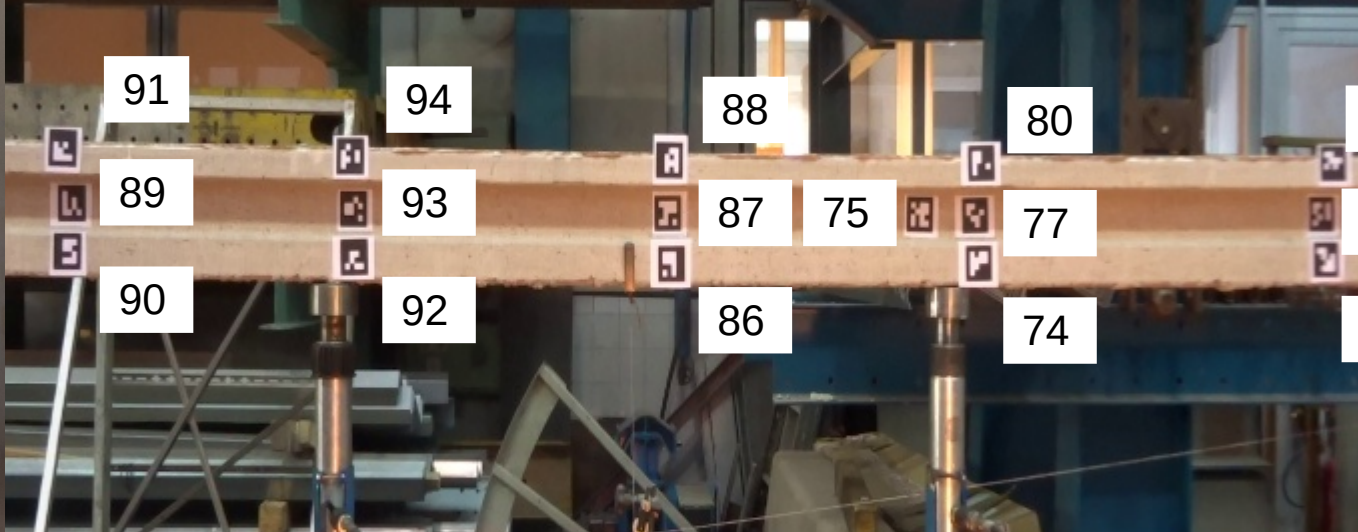
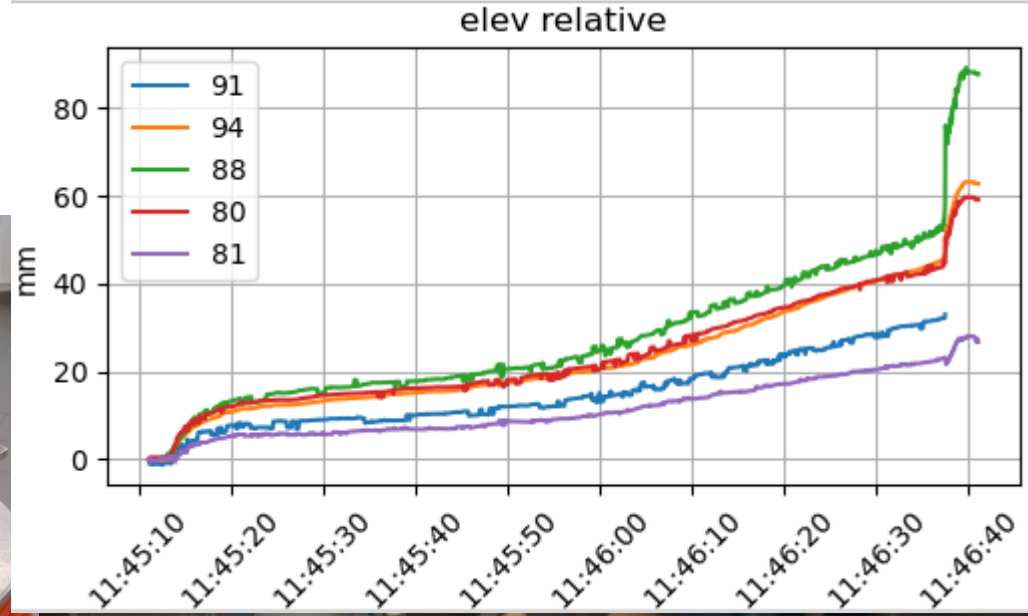
# Alkalmazási példa



# Alkalmazási példa



# Alkalmazási példa





2022.10.25.

13/20

# Python telepítés



+

MAX\_PATH limit  
kikapcsolása

# Python modulok telepítése



```
pip install numpy
pip install pyyaml
pip install python3-matplotlib
pip install opencv-python
pip install opencv-contrib-python
```

**pip – Package Installer for Python**

## Telepítés ellenőrzése

```
python
>>> import numpy
>>> import pyyaml
>>> import cv2
>>> cv2.aruco
<module 'cv2.aruco'>
>>> import matplotlib.pyplot
```

**Hibaüzenet nélkül kell lefutnia**

# Ulyxes telepítés



zsiki / ulyxes Public

Pin

Unwatch 3

Fork 11

Star 13

<> Code Issues 14 Pull requests Discussions Actions Projects Wiki Security Insights Settings

master

3 branches 0 tags

Go to file

Add file

Code

About

Surveyor's sensor data processing system

Readme

GPL-2.0 license

13 stars

3 watching

11 forks

- zsiki better handling for non-existent input file(s)
- camera better handling for non-exi
- doc camera applications added
- pyapi better handling for non-exi
- pyapps missing signal modul impo
- robot initial upload of robot project

Clone

HTTPS SSH GitHub CLI

https://github.com/zsiki/ulyxes.git

Use Git or checkout with SVN using the web URL.

Download ZIP

Kitömörítés alkönyvtárakkal

2022.10.25.

16/20



# Környezet beállítása



PYTHONPATH környezeti változó beállítása  
ulyxes\pyapi és ulyxes\camera könyvtárakra

## Ellenőrzés:

```
python
```

```
>>> import sys
```

```
>>> sys.path
```

```
['', '/home/siki/ulyxes/pyapi', '/home/siki/ulyxes/camera',  
'/usr/lib/python310.zip', '/usr/lib/python3.10',  
'/usr/lib/python3.10/lib-dynload',  
'/usr/local/lib/python3.10/dist-packages',  
'/usr/lib/python3/dist-packages']
```



# Programok

- **charuco.py** kamera kalibráció **charuco táblával**
- **video\_aruco.py** videó felvételtől vagy közvetlenül a kamerából jövő képekből ArUco jelek felismerése
- **video\_nth.py** videó felvételtől képek kivétele
- **video\_correlation.py** minta illesztés videó felvételtől vagy közvetlenül a kamerából jövő képekből
- **imgs\_aruco.py** mint video\_aruco, de önálló képekből
- **imgs\_correlation.py** mint video\_correlation, de önálló képekből

# Gyakorlati feladatok



- Kamera kalibrálás (12-20 kép a Charuco tábláról, különböző irányokból) **charuco.py**
- Videó felvétel mozgó ArUco jelről
- Videó feldolgozás **video\_aruco.py** és **video\_correlation.py**
- Eredmények grafikus megjelenítése **graph\_plot.py**

