

# Vasúti infrastruktúra automatizált hibadetektálása mobil lézerszkenneléssel

Cserép Máté, Hudoba Péter  
Eötvös Loránd Tudományegyetem  
{mcserep, peter.hudoba}@inf.elte.hu

FOSS4G-HU, Budapest  
2018. november 30.



**SZÉCHENYI** 2020

Európai Unió  
Európai Szociális  
Alap



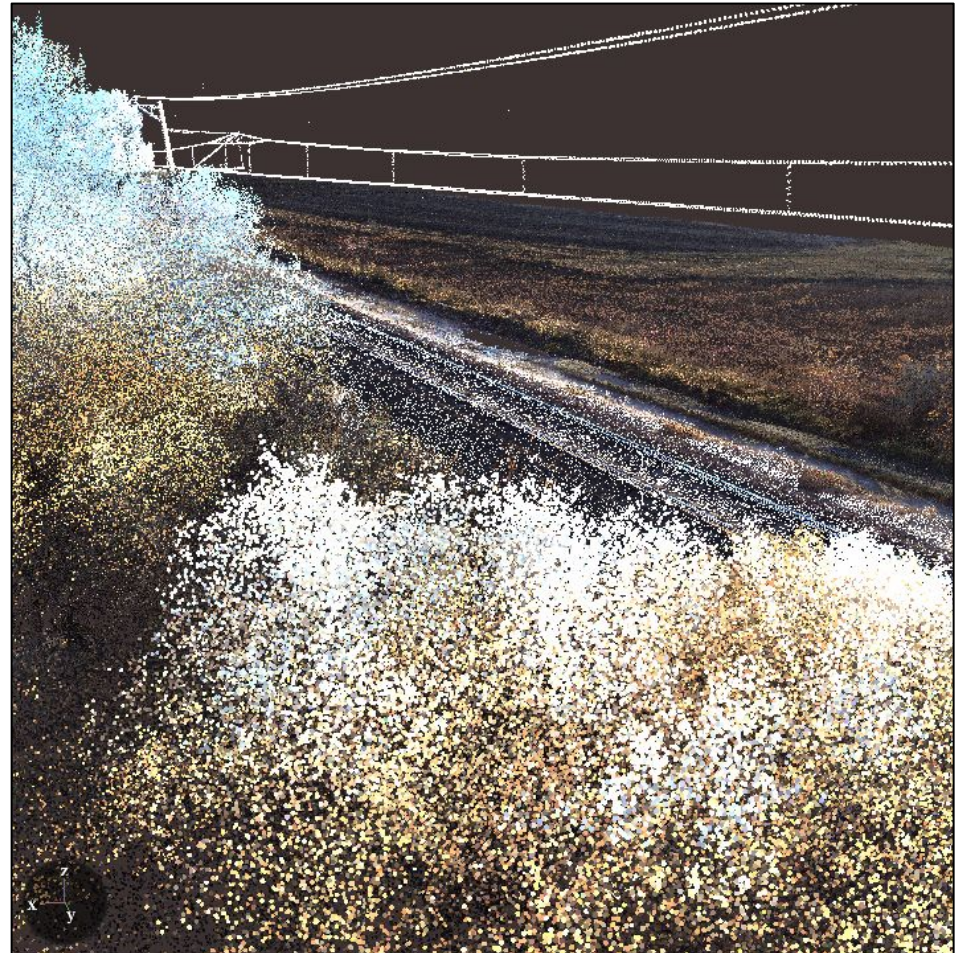
**BEFEKTETÉS A JÖVŐBE**

# MOTIVÁCIÓ

Megfigyelő és monitoring  
rendszer fejlesztése

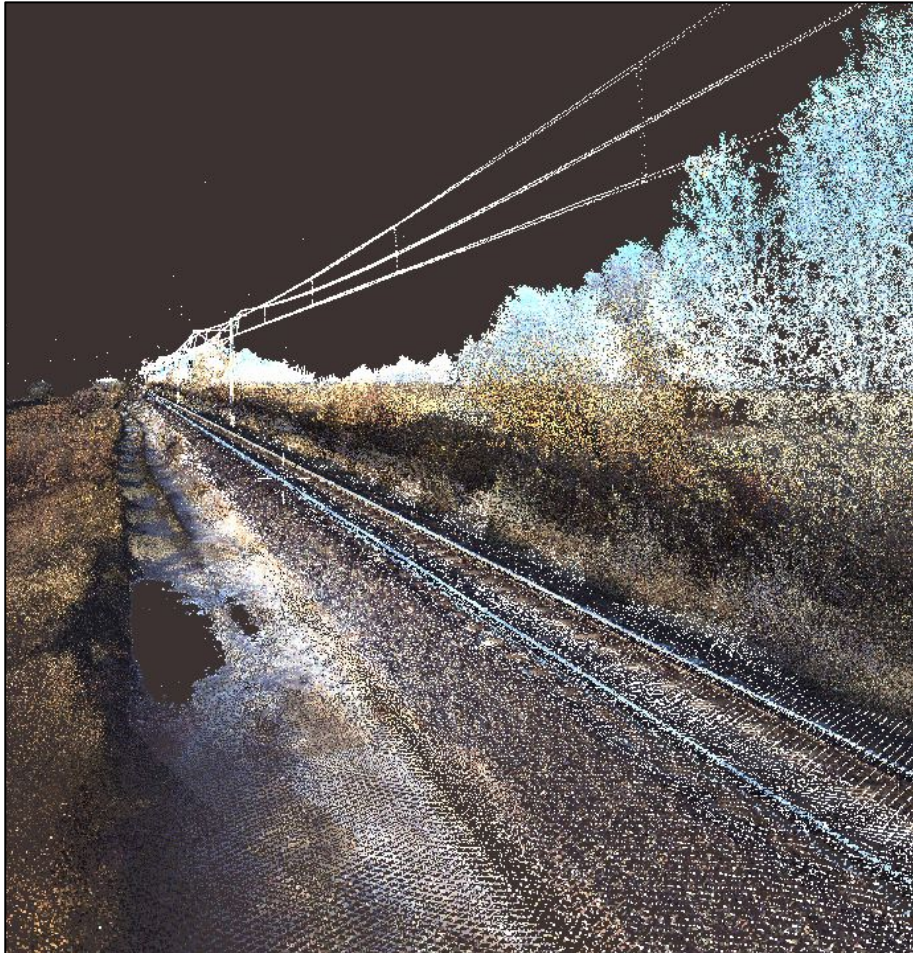
## **Automatizált ellenőrzés:**

- Felsővezeték magassága  
sínkorona felett
- Felsővezeték kigyózása
- Űrszelvény közeli  
(veszélyes) növényzet
- Stb.





# TESZT LIDAR PONTFELHŐ



## Méret:

- $1,54 * 10^9$  pont
- 18,5 km hosszú
- 130 m széles

## Pontosság:

- Mérési pontosság: 3-7 mm
- Helyzeti pontosság: 3-5 cm

# FELSŐVEZETÉK DETEKTÁLÁSA

- **Probléma:** a precíz state-of-the-art algoritmusok nem alkalmazhatóak nagy méretű pontfelhőkön
- **Megoldás:** pontok számának csökkentése durva, de hatékony algoritmusok által:
  - Elemi, lineáris előszűrők (1D, 2D)
  - Középvonal általi szűrés (2D)
  - Kábel objektum detektálása (2D, 3D)

# LINEÁRIS SZŰRŐK

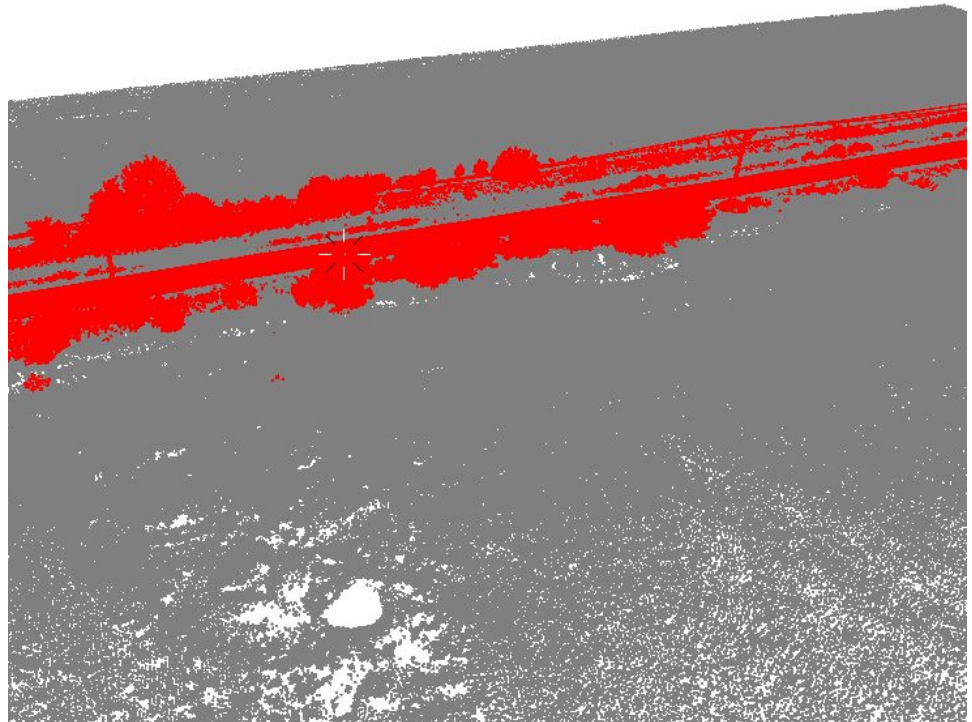
Egyszerű előszűrések

## **GroundFilter:**

- Pontok eltávolítása adott Z koordináta érték alatt.

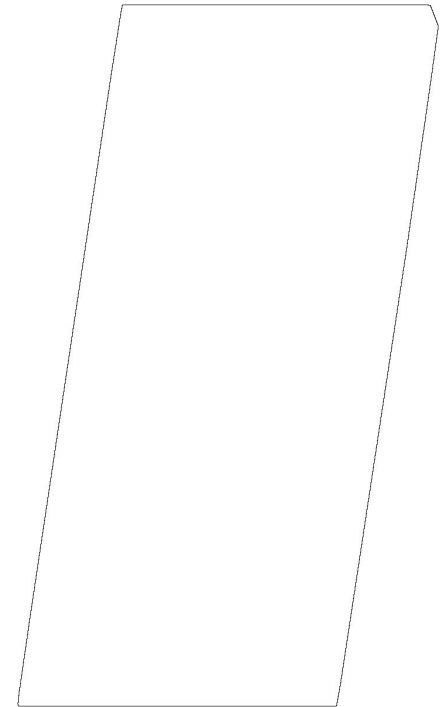
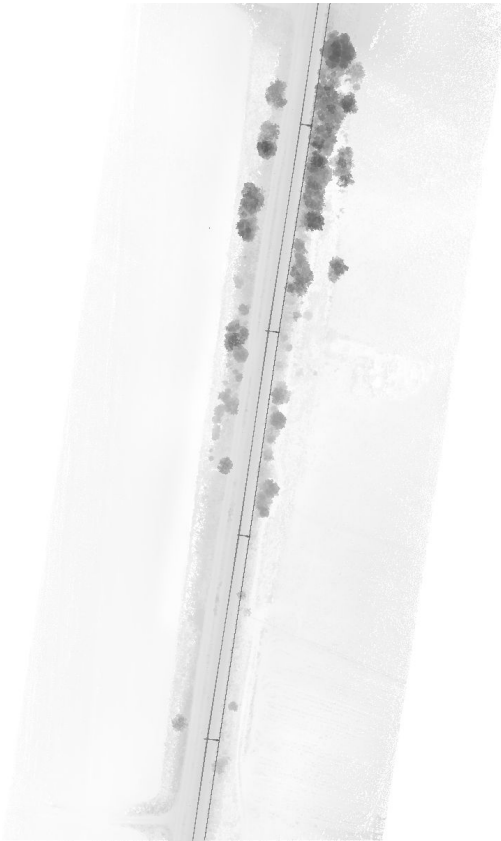
## **LimitFilter:**

- Pontfelhő szegmentálása adott tengely mentén, majd a ritka szegmensek eltávolítása.



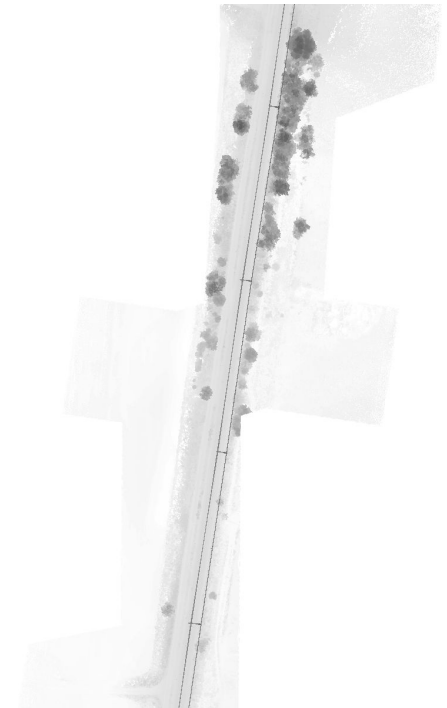
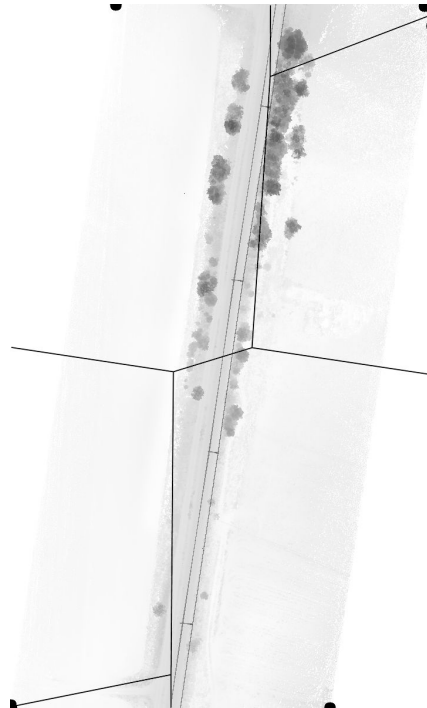
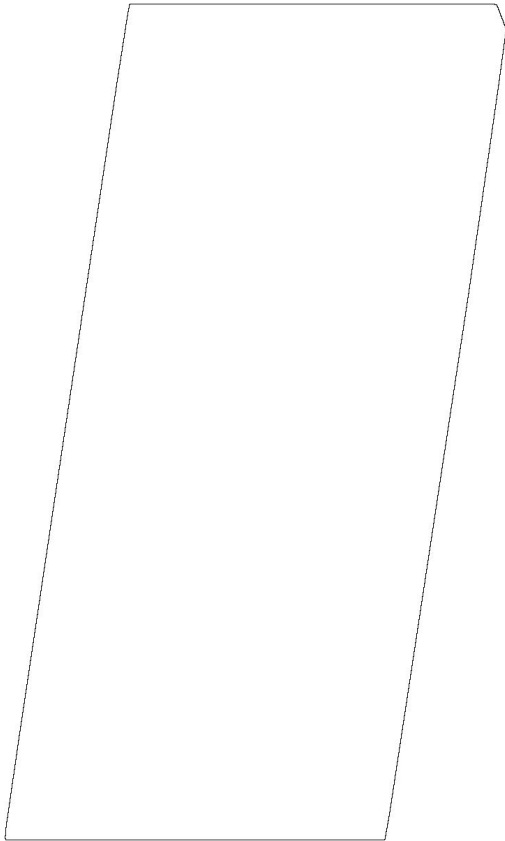
# KÖZÉPVONAL KIVÁGÁSA - Voronoi

Konvex burok létrehozása



# KÖZÉPVONAL KIVÁGÁSA - Voronoi

Voronoi poligon kiszámítása és a pontok megtartása kizárólag az élek mentén





# KÖZÉPVONAL KIVÁGÁSA - Skeleton

- A kitöltött konvex burok redukálása morfológiai vázkijelöléssel
- Pontok megtartása kizárólag a váz körül

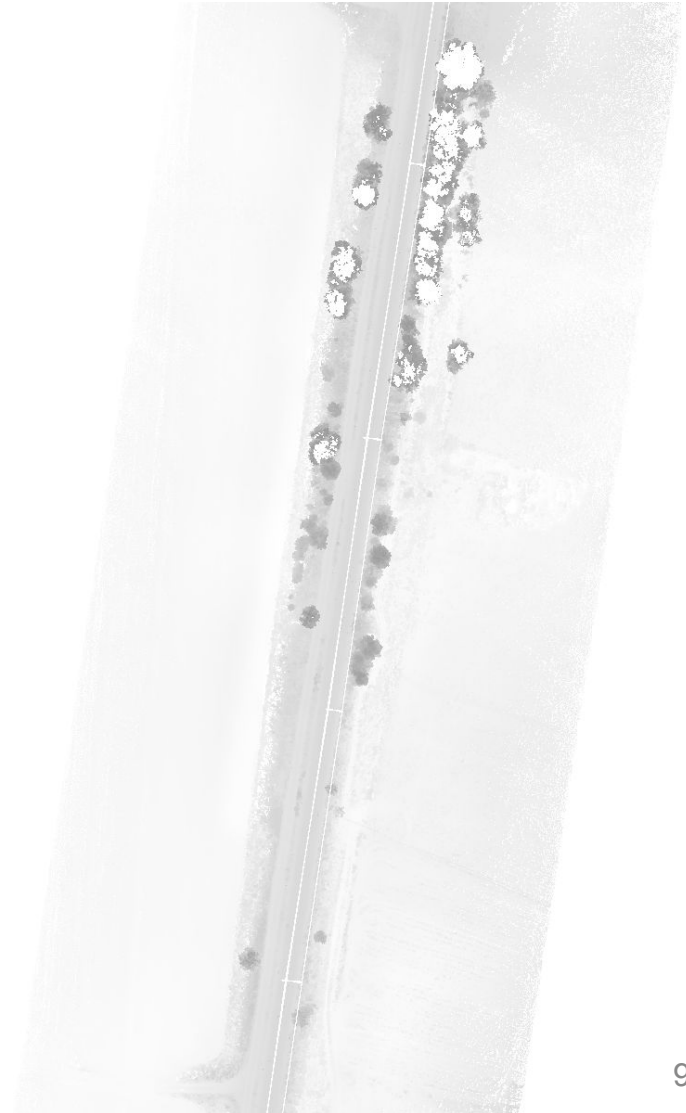
*Megjegyzés: a módszer hamis negatív területet is produkál*





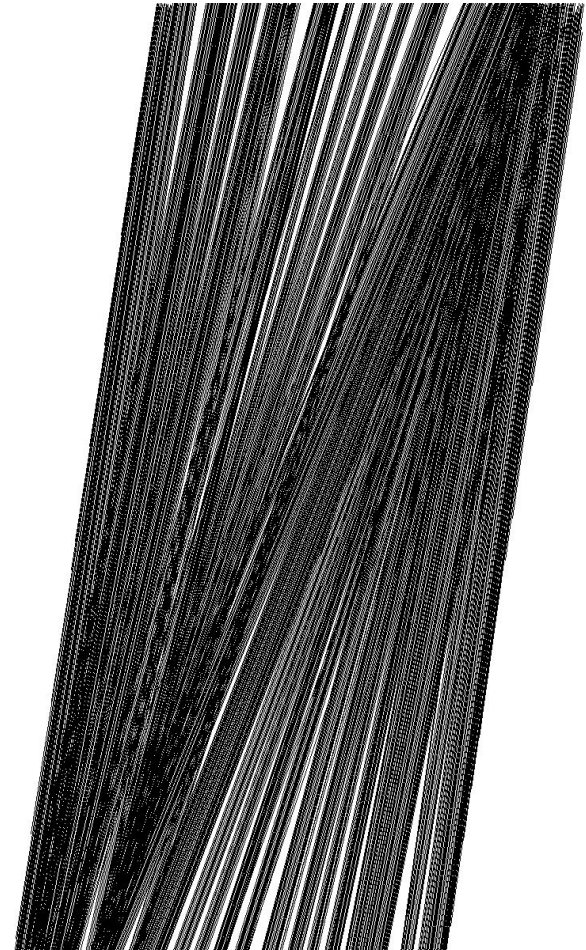
# KÖZÉPVONAL KIVÁGÁSA - Angle

- Pontfelhő rétegekre osztása (Z tengely)



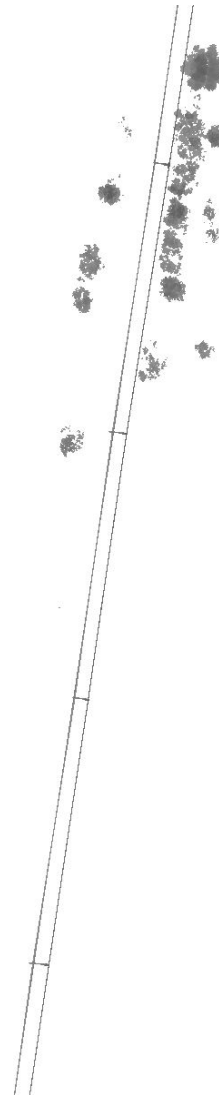
# KÖZÉPVONAL KIVÁGÁSA - Angle

- Pontfelhő rétegekre osztása (Z tengely)
- Keresési vonalak kijelölése Hough transzformációval



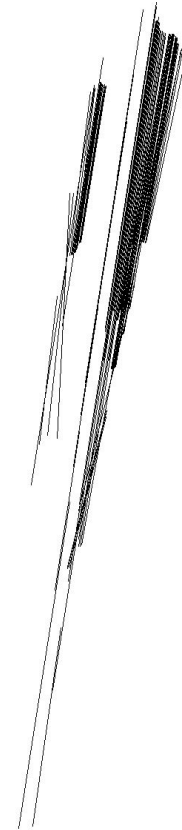
# KÖZÉPVONAL KIVÁGÁSA - Angle

- Pontfelhő rétegekre osztása (Z tengely)
- Keresési vonalak kijelölése Hough transzformációval



# KÖZÉPVONAL KIVÁGÁSA - Angle

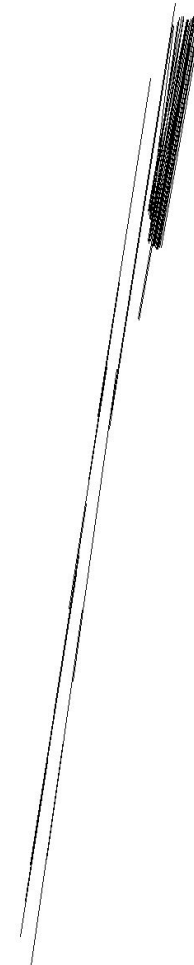
- Pontfelhő rétegekre osztása (Z tengely)
- Keresési vonalak kijelölése Hough transzformációval





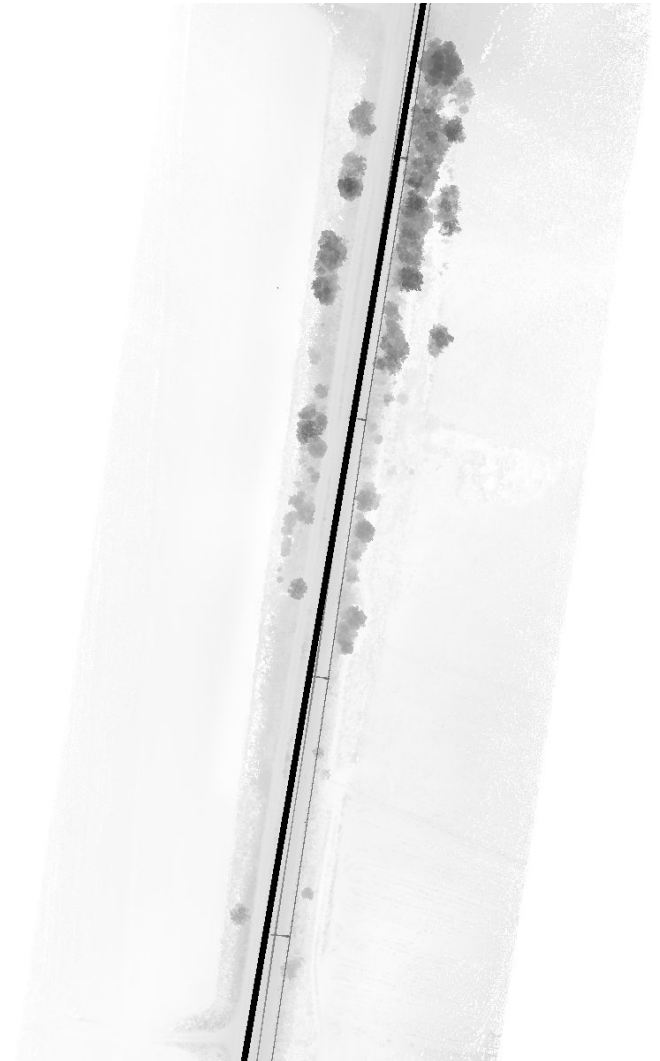
# KÖZÉPVONAL KIVÁGÁSA - Angle

- Pontfelhő rétegekre osztása (Z tengely)
- Keresési vonalak kijelölése Hough transzformációval



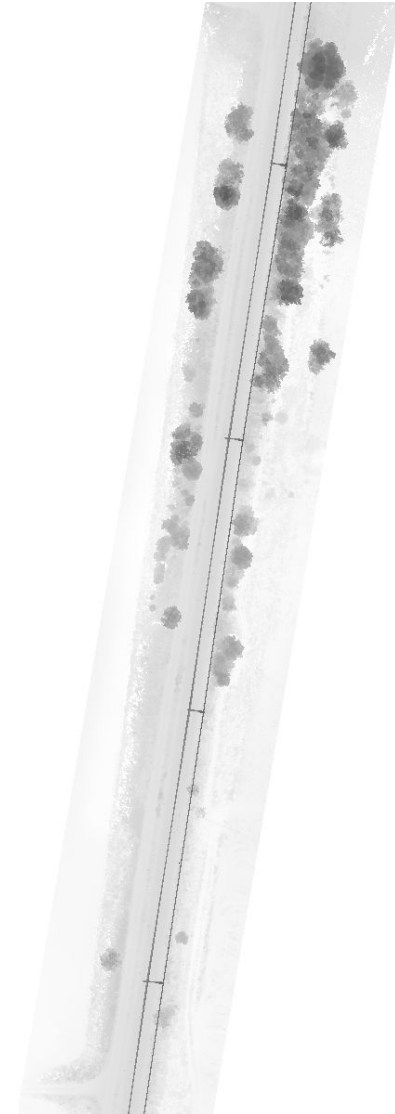
# KÖZÉPVONAL KIVÁGÁSA - Angle

- Vonalak átlagos szögének kiszámítása rétegenként
- Középvonal számítása a rétegek szögei alapján
- A réteg súlya fordítottan arányos a rétegben talált keresési vonalak számával

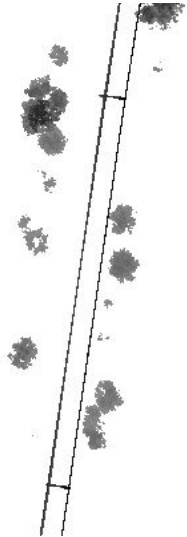
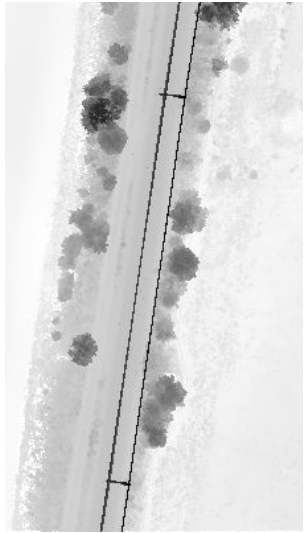


# KÖZÉPVONAL KIVÁGÁSA - Angle

- Vonalak átlagos szögének kiszámítása rétegenként
- Középvonal számítása a rétegek szögei alapján
- A réteg súlya fordítottan arányos a rétegben talált keresési vonalak számával
- Pontok megtartása a középvonal közelében

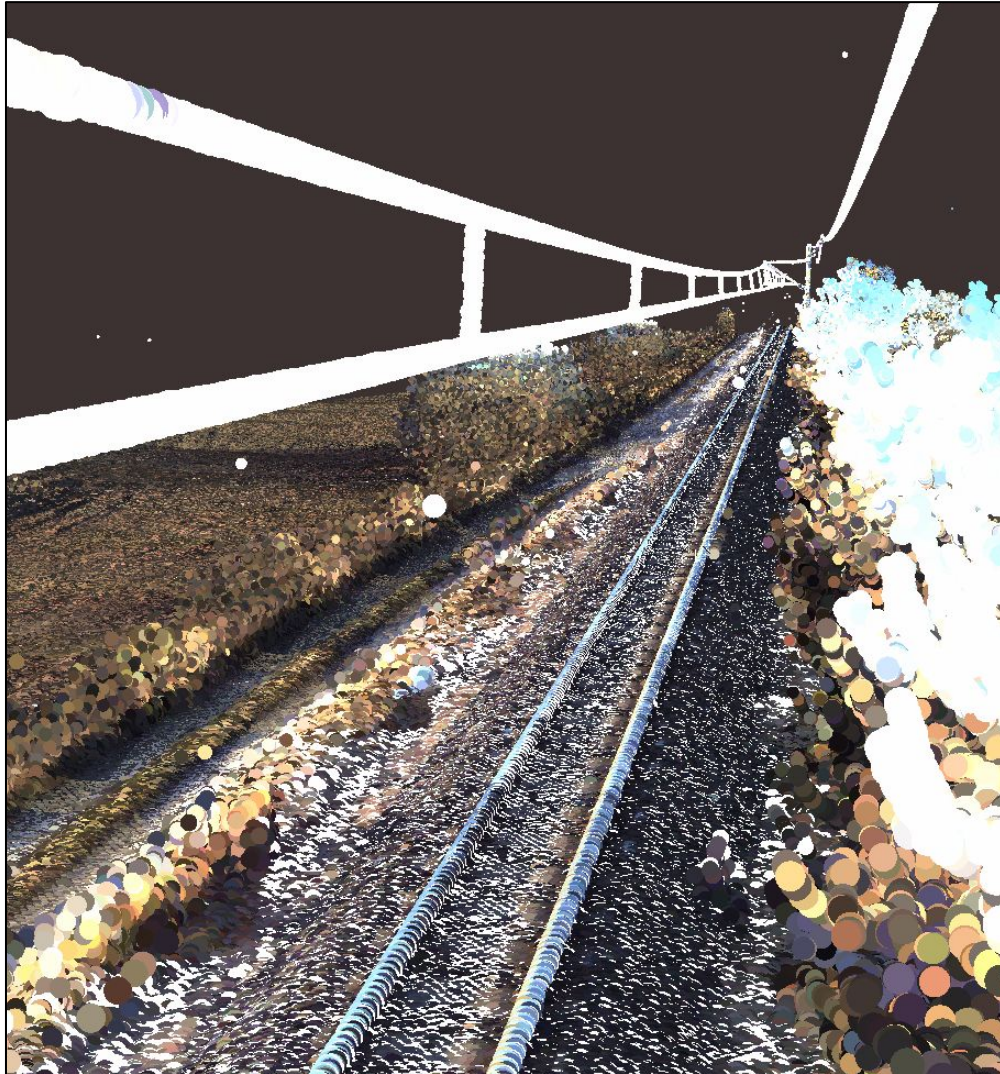


# KERESÉS FELÜLRŐL - Above



- DEM konstruálása
- Minden cell a legnagyobb Z értékkel kerül reprezentálásra
- Küszöbérték alatti cellák törlése
- Pontok szűrése vonal kereséssel
- Pontok szűrése környezet alapján (hasonló Z értékek vártak a kábelek mentén)

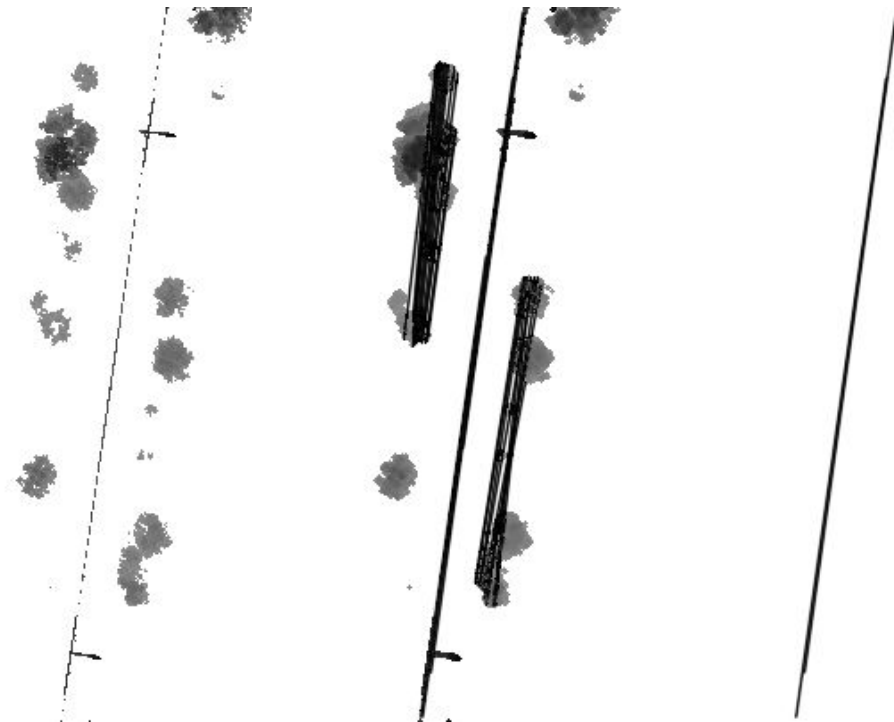
# KERESÉS FELÜLRŐL - Above



Egymás alatti kábelek  
detektálása?



# KERESÉS FELÜLRŐL - Above



Egymás alatti kábelek  
detektálása?

- Első futtatás  
eredményének  
eltávolítása
- Algoritmus újbóli  
kiértékelése
- Egyesítés

# KERESÉS PONTSŰRŰSÉGGEL - Density

- Fixpontos 3D grid konstruálása a pontfelhőn, amelyben minden cellát a tartalmazott pontok száma, azaz **sűrűsége** reprezentál.
- Módszer #1: cellák eltávolítása, amelyek körül túl kevés vagy sok sűrű cella van. (A kábelek folytonosak és nem érintkeznek közvetlenül más objektummal.)
- Módszer #2: a felsővezeték lehetséges hajlásának figyelembe vétele a szomszédos cellák kiválasztásakor.

# EREDMÉNYEK

|          | Futásidő (mp) | Kiszűrt pontok |
|----------|---------------|----------------|
| Ground   | 0,22          | 3.099.515      |
| Voronoi  | 1,94          | 239.701        |
| Angle    | 1,14          | 560.805        |
| Skeleton | 1,04          | 3.192.666      |
| Above    | 3,29          | 7.292.901      |
| Density  | 2,07          | 6.425.803      |

Minta mérete: 7,316,298 pont

# EREDMÉNYEK

|          | Hamis negatív | Hamis pozitív |
|----------|---------------|---------------|
| Ground   | 0.00 %        | 99.41 %       |
| Voronoi  | 0.00 %        | 99.65 %       |
| Skeleton | 52.54 %       | 99.71 %       |
| Angle    | 0.00 %        | 99.63 %       |
| Above    | 7.77 %        | 2.24 %        |
| Density  | 21.79 %       | 97.82 %       |

Hamis negatív: a validációs pontfelhő méretéhez viszonyítva

Hamis pozitív: az eredmény pontfelhő méretéhez viszonyítva

# EREDMÉNYEK

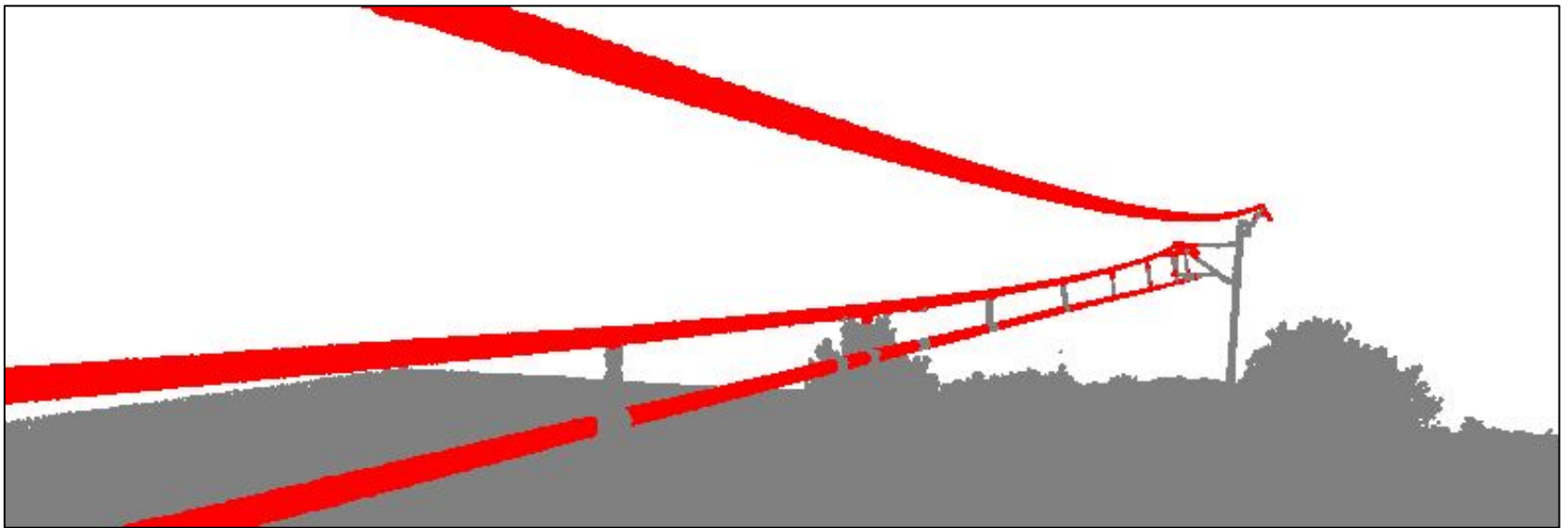
A módszerek kombinálása algoritmus pipelineok révén:

|                            | Futásidő<br>(sec) | Hamis<br>negatív | Hamis<br>pozitív |
|----------------------------|-------------------|------------------|------------------|
| Voronoi -> Above           | 3.06              | 7.77 %           | 2.24 %           |
| Angle -> Above             | 3.01              | 8.29 %           | 2.32 %           |
| Angle -> Ground -> Above   | 2.53              | 8.91 %           | 0.93 %           |
| Voronoi -> Ground -> Above | 2.24              | 8.36 %           | 1.08 %           |
| Ground -> Density -> Above | 0.70              | 16.69 %          | 0.98 %           |



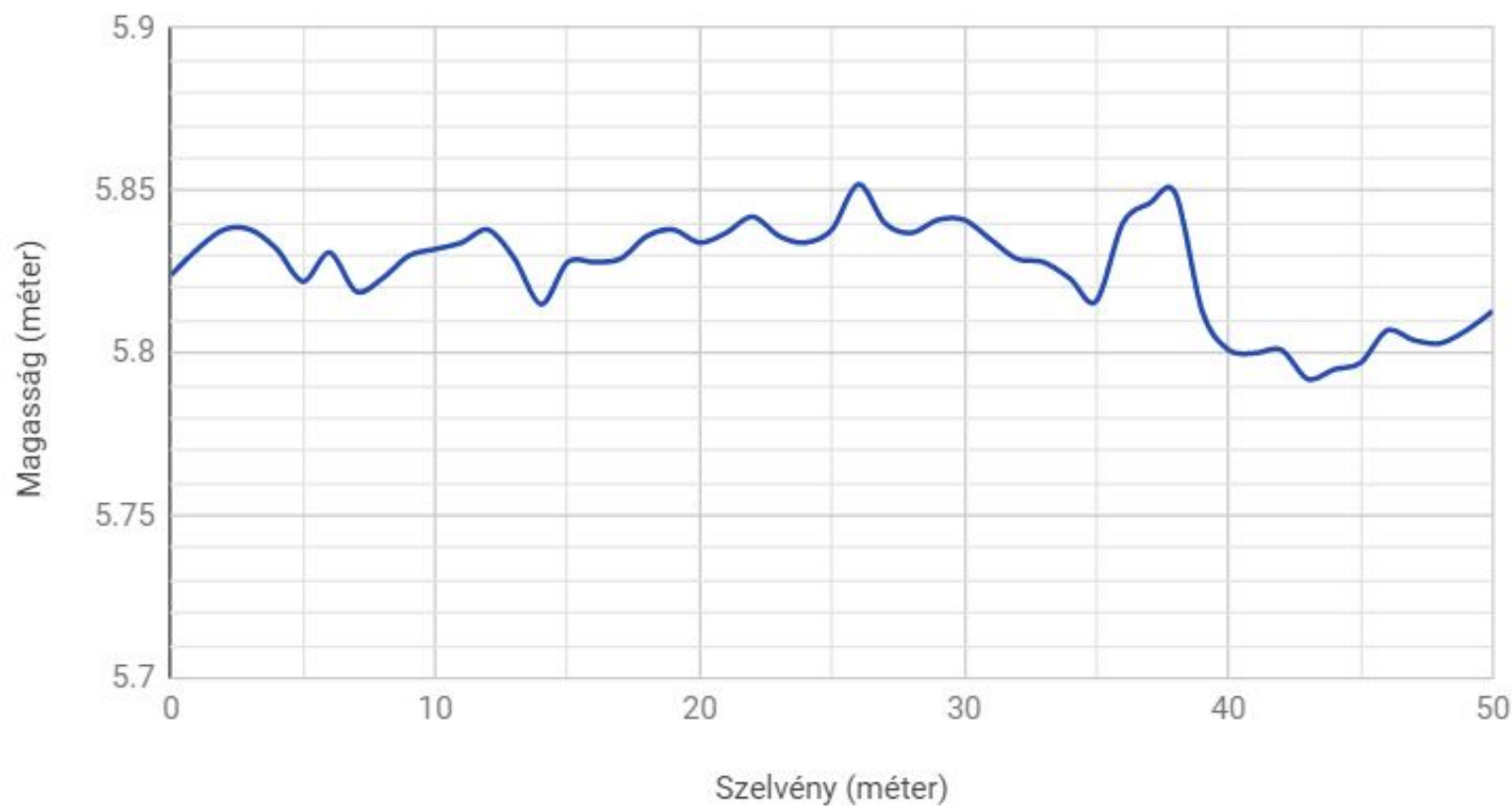
# EREDMÉNYEK

*A Voronoi -> Search From Above* pipeline eredmény vizualizálva:



# EREDMÉNYEK

Felsővezeték magassága a sínkorona felett



# IMPLEMENTÁCIÓ

Implementáció C++ nyelven, nyílt forráskódú  
szoftverkönyvtárak felhasználásával:

- LASlib, LASzip
- Point Cloud Library (PCL)
- openCV



A forráskód elérhető GitHub-on:

<https://github.com/mcserep/railroad>

# ÖSSZEGZÉS

## Összegzés:

- Az elemzett (elsődlegesen 2D) eljárások jelentősen jobbak a state-of-the-art algoritmusok futásidejénél
- Kisebb pontosságuk mellett jó előszűrők lehetnek

## További munka:

- Futtatás a teljes adathalmazon, párhuzamos feldolgozás lehetőségének elemzése
- További ellenőrző és hibakereső automatizmusok implementációja

# KÖSZÖNÖM A FIGYELMET!

**Cserép Máté**, Hudoba Péter  
Eötvös Loránd Tudományegyetem  
{mcserep, peter.hudoba}@inf.elte.hu

<https://github.com/mcserep/railroad>

EFOP-3.6.3-VEKOP-16-2017-00001:  
A projekt a Magyar Állam és az Európai Unió  
támogatásával, az Európai Szociális Alap  
társfinanszírozásával valósul meg.

**SZÉCHENYI** 2020



MAGYARORSZÁG  
KORMÁNYA

Európai Unió  
Európai Szociális  
Alap



**BEFEKTETÉS A JÖVŐBE**