



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Építőmérnöki Kar
Általános és Felsőgeodézia Tanszék

Ingatlan-nyilvántartási térkép felújítása pontfelhő segítségével

Készítette: Hrutka Bence Péter
Konzulensek: Dr. Takács Bence
Dr. Siki Zoltán

2020.

Tartalomjegyzék

1.	Bevezetés.....	3
2.	Kataszteri térképek felújítása modern eszközök segítségével.....	4
2.1.	Holland ingatlan-nyilvántartási térkép létrehozása UAV-vel.....	4
2.2.	3D épületmodellezés Digitális Felület Modellek segítségével.....	5
2.3.	Lengyel példák UAV félmérések használatára a kataszteri térképezés során.....	6
2.4.	Kataszteri térkép előállítása katasztrófa sújtotta területen.....	9
2.5.	Egy kínai LiDAR példa.....	10
3.	Egy magyarországi példa.....	12
3.1.	Barnag.....	12
3.2.	Digitális alaptérkép és állapota.....	12
3.3.	Felmérés és feldolgozás.....	16
4.	Pontfelhő tartalmának egyszerűsítése.....	17
4.1.	Előkészítés.....	17
4.2.	Osztályozás.....	17
4.2.1.	LAStools.....	17
4.2.2.	Feldolgozás.....	17
4.3.	Normalizált pontfelhő előállítása.....	21
5.	Manuális felhasználási lehetőségek.....	26
5.1.	Előkészítés és betöltés AutoCAD Civil 3D-be.....	26
5.2.	Metszet készítés.....	26
5.3.	Egyéb AutoCAD eszközök.....	27
6.	Automatizálási lehetőség vizsgálata.....	32
6.1.	2D metszet előállítása.....	32
6.2.	RANSAC – sorozatos egyenes illesztés.....	34
6.3.	Sorozatos egyenes illesztésen alapuló program készítése Matlab-ban.....	36
6.4.	A feldolgozás során kapott végtermék összehasonlítása az AutoCAD Civil3D Section Lines parancsának eredményeivel.....	40
6.5.	Az összehasonlítás eredményei számokkal kifejezve.....	43
7.	Összefoglalás.....	45
8.	Irodalomjegyzék.....	46
9.	Mellékletek.....	49

1. Bevezetés

Az ingatlan-nyilvántartási térképek minőségének ellenőrzése, fenntartása és esetleges felújítása a földmérő szakma egyik kiemelt fontosságú kérdései közé tartozik.

A napjainkban használt digitális földmérési alaptérképek pontossága területenként igencsak változó a digitális változat eltérő előállítási módjának köszönhetően. Azokon a területeken, ahol a kataszteri program keretében részleges vagy teljes újfelmérést végeztek, a birtokhatárpontok koordinátái jóval pontosabbak. Egy másik példa Budapest, ahol az 1930-as években végeztek szabatos városmérést, amelynek eredményeként a birtokhatárpontokat cm pontossággal határozták meg. A mérés céljára egy egységes háromszögelési hálózatot is kifejlesztettek, a budapesti városi sztereografikus rendszert (BÖV). Az egységes országos vetületi rendszerre (EOV) való áttéréskor az egységes átszámítási módszernek köszönhetően a városmérés eredményeit a pontosságvesztés nélkül fel lehetett használni. Így ezeken a helyeken megmaradt a cm-es pontosság.

Azonban nem mindenhol készült ilyen szabatos mérés. Az ország nagy részében a digitális alaptérkép előállítására grafikus térképeket használtak fel. Az érintett területeken a grafikus térkép készítésekor jellemző távolságokat mérőszalaggal dm élesen ellenőrizték le. Így a pontosság elmaradt a numerikus térképektől. Az ellenőrzés természetéből adódóan a különböző méretek hiányosak voltak, hiszen nem lehetett mindent ellenőrizni (pl. telek hossz) és ezek a méretek relatív és nem abszolút adatokat szolgáltatottak. A geometriai alap így nem volt egyértelmű. (Boda 2008)

A kialakult probléma megoldását az jelentette volna, hogy a szükséges területeken új felmérést végeznek el. Az időhiány, a nagy költségek miatt azonban végül ez elmaradt. Helyette jogszabályok segítségével próbálták a kialakult helyzetet orvosolni, amivel megmutatták, hogy miként kell kezelni gyakorlatban a különböző adatforrásokból származó digitális alaptérképeket. A szakma gyakorlatilag megtanult együtt élni hibákkal. A legtöbb területen a relatív pontosság megtartása vált céllá. Azonban ez nem mindenhol valósulhatott meg. Ugyanis előfordulnak olyan települések, területek, ahol olyan szintű eltérések alakultak ki a valódi helyzethez képest, amelyek már a hétköznapi földmérési és számos más mérnöki tevékenységet (pl. közműnyilvántartás, tervezési feladatok) is erősen hátráltatták. A térképek a változások eltérő módon történő változásvezetése miatt elkezdtek inhomogénné, nehezen használhatóvá válni. Felvetődött újra a kérdés; hogy a digitális alaptérkép tartalmát korrigálni kellene egy új felmérés segítségével.

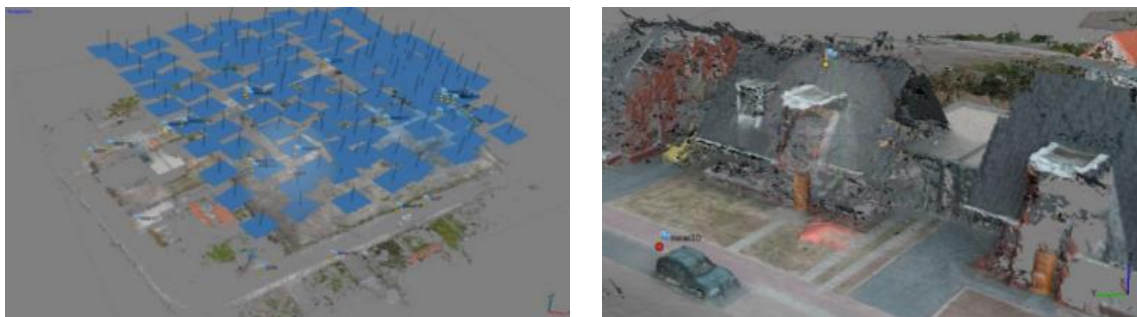
Korábban ez a munka országos léptékben gyakorlatilag esélytelennek tűnt a jelentős mennyiségű munka és költségek miatt. Azonban ma már rendelkezésre állnak olyan technológiák (drónról végzett fotogrammetria, földi, mobil és légi lézerszkennelés), amelyek a korábban elképzelhetetlennek tűnő gyorsaság mellett, megfelelő pontossággal és részletességgel teszik lehetővé nagy területek felmérését.

2. Kataszteri térképek felújítása modern eszközök segítségével

A kataszteri térképek felújítása, karbantartása korszerű technológiákkal nem csak hazánkban kutatott terület. Számos külföldi kísérlet, kutatás és tanulmány született a témában, amelyekben a rendelkezésre álló új technológiák térképezésben történő alkalmazhatóságának vizsgálata mellett, nagyobb területek térképezéséről, az előállítható termékekről és azok pontossági vizsgálatáról szólnak. A következő részben ezek közül szeretnék párat ismertetni.

2.1. Holland ingatlan-nyilvántartási térkép létrehozása UAV-vel

2013-ban Hollandiában már kísérleteztek az UAV (Unmanned Aerial Vehicles) segítségével készített nagy felbontású fényképekből előállított pontfelhő kiértékeléséből levezetett birtokhatárok megállapításával. A technológia bevethetőségének alapját az adta, hogy az ingatlan-nyilvántartási térképek előállításához a technológia segítségével elérhető a szükséges pontosság. További előnynek könyvelték el azt, hogy a szükséges mérések elvégzéséhez nem kell az érintett telkek tulajdonosaival időpontegyeztetés, ami egy elég erős hátráltató tényező volt a korábbi munkavégzés során. Az éles felmérés előtt két tesztrepülést és feldolgozást is végeztek. Különböző beállítások mellett (repülési magasság, átfedés), tesztelték a feldolgozószoftvereket, kamera kalibrációt is végrehajtottak, majd a szerzett tapasztalatok alapján megállapították a megfelelő paramétereket a munkavégzéshez. Így a vizsgált területen 7 darab 1.7 cm középhibával jellemezhető illesztőpont felhasználásával, 50 méteres repülés magasság mellett, 80% átfedéssel végezték el a felmérést. A begyűjtött 360 fotóból georeferált pontfelhőt és ortofotót állítottak elő. (*Rijsdijk et al. 2013*)



1. ábra: A készített fotók elhelyezkedés és a feldolgozás eredménye (*Rijsdijk et al. 2013*)

Az előállított pontfelhőt 15 db a helyszínen GNSS technológiával bemért pont segítségével ellenőrizték, ami alapján a jellemző középhibát 6.3 cm értékre határozták meg. Összeségében megállapították, hogy ha megfelelő mennyiségű illesztőponttal, repülési beállítással és feldolgozási módszerrel az UAV-s mérésekből kapott végtermék alkalmazható az ingatlan-nyilvántartási munkálatokhoz szükséges térképezésre.



2. ábra: Az előállított ortofotó és a kiértékelt telekhatár (*Rijsdijk et al. 2013*)

2.2. 3D épületmodellezés Digitális Felület Modellek segítségével

A UAV-val végzett felmérés csak egy módja az információszerzésnek. Számos egyéb technológia segítségével is gyűjthetünk adatokat, mint például a műholdas felvételek, légifotók vagy akár légi lézerekkel készített adatok feldolgozásával. Az ily módon végzett felméréssel akár nagyobb városok modellezését is el lehet végezni különböző felbontásokban. Svéd, olasz és spanyol együttműködésnek köszönhetően Trentó városára (Olaszország) végeztek vizsgálatot a fentiekben említett három technológia segítségével megalkotható 3D városmodellek pontossági vizsgálatáról. Épület alaprajzokat három esetben hasonlították össze; a GeoEye műhold képeiből alkotott DSM (Digital Surface Model), repülés során készített légifelvételekből előállított DSM és LiDAR eredményekből előállított DSM-ből. A három végtermékből automatizáltan állítottak elő LoD1 és LoD2-es városmodelleket. (*Moreira et al. 2013*)

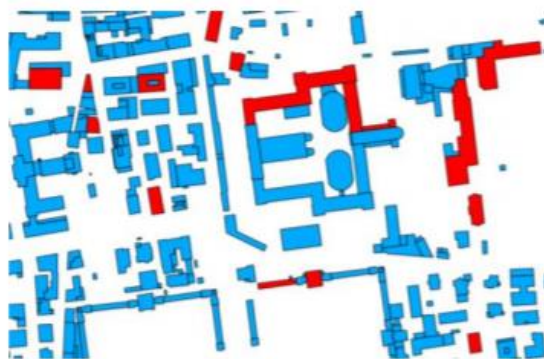


3. ábra: Az 1:10 000-es topográfia térképből generált LoD1-es épület modellek (a: GeoEye; b: légifotó; c: Lidar) (*Moreira et al. 2013*)



4. ábra: Az 1:1000-es domborzati térképből generált LoD2-es épület modellek (a: GeoEye; b: légifotó; c: Lidar) (*Moreira et al. 2013*)

Végeredményként sikerült meghatározni a különböző modellek átlagos ellentmondásainak értékét két különböző méretarányú kiindulási adatok esetében. Az 1:1000 kiindulási alapú adatokból az épületeket a GeoEye műholdfelvételek kiértékelésével 41%-os, légifotóból 59% és LiDAR adatokból 67%-os mértékben sikerült rekonstruálni. Az 1:10.000-es alapnál ezek az értékek mind a három kategóriában 10%-al elmaradtak a fenti értékekhez képest. (*Moreira et al. 2013*)



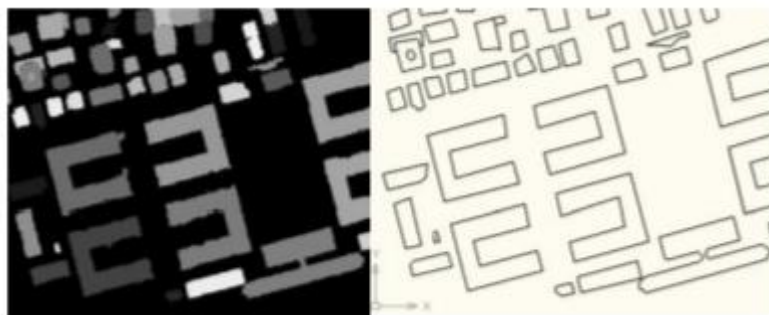
5. ábra: Tartalmi eltérések a valós állapothoz képest

A területen végzett geodéziai mérésekhez viszonyított maradék ellentmondások átlagai a három különböző adatforrás esetén a LiDAR-on alapuló felmérések esetén adódott a legjobbra; 0.51 m értékben a 1:10.000 méretarányú topográfiai térképi alpnál. (Moreira et al. 2013)

1. táblázat: A maradék ellentmondások átlagai

		2D DATA	
		1:1,000	1:10,000
3D DATA	GeoEye	1.94 m	2.06 m
	Airborne	1.82 m	1.98 m
	LiDAR	0.64 m	0.51 m

A kutatással kimutatták, hogy a műholdfelvételek nem ideálisak az automatikusan előállított épületmodellezéshez. A legjobb eredményeket a légifotók és a LiDAR-os felmérésekből lehet kihozni. Az eredmények minősége erőteljes korrelációban van a topográfiai térképek és a DTM állapotával. Ahol nem érhető el a domborzati adatok, ott az előállított DSM termékekből vezethetők le az épületkontúrok. Ez az információ akár felhasználható az ingatlan-nyilvántartás számára. (Moreira et al. 2013)



6. ábra: Eredeti DSM és levezetett épület kontúrok

2.3. Lengyel példák UAV felmérések használatára a kataszteri térképezés során

Ingatlan-nyilvántartási térképek UAV alapú felméréssel készült felújítására Lengyelországban is tettek kísérletet. Itt Chesne városában végeztek méréseket merev szárnyú platformra felszerelt fényképezőgép segítségével. A fotókat 200 méteres repülési magasságból készítették 15 darab illesztőpontot és 5 ellenőrző pont RTK-GNSS technikával történő bemérése mellett.

A mérési eredményekből pontfelhőt állítottak elő, amit 0.10-0.13 m középhibával jellemeztek. A pontfelhőből DTM-et és ortofotót is előállítottak. Az ortofotót kiértékeltek. A kiértékelés eredményét a következő táblázat (2. táblázat) mutatja. (Kedzieriski et al. 2016)

2. táblázat: A kiértékelés eredményei (Kedzieriski et al. 2016)

Object type	The accuracy of object identification on the map [%]	RMSE [m]
Building	94	0.09
Boundaries of the plots	78	0.19
Boundaries of roads	92	0.23
Other elements of terrain infrastructure	76	0.13

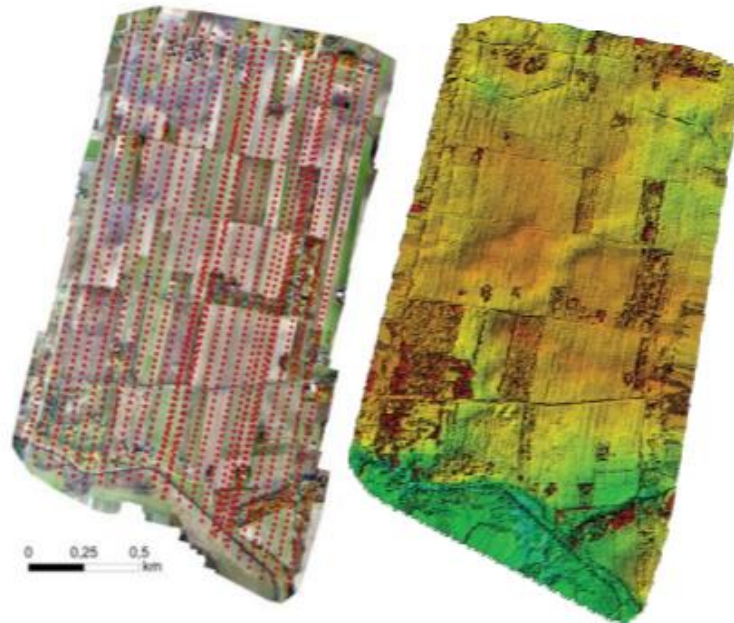
A közölt eredményeknél látható, hogy az épület (94%) és telekhatárokat (92%) elég nagy megbízhatósággal sikerült detektálni. A legnagyobb pontosságot az épületek esetén sikerült elérni a 9 cm-es átlagos négyzetes hiba gyök értékkel. A telekhatárok esetén az RMSE érték: 23 cm. (Kedzieriski et al. 2016)



7. ábra: UAV ortofotó alapján felújított alaptérkép (Kedzieriski et al. 2016)

A tanulmányban alapul vett mérések, kiértékelésük szerint a UAV-n alapuló térképezés jó kiindulásul szolgálhat térképek aktualizálásához. (Kedzieriski et al. 2016)

Lengyelországban további kutatás is született, ahol szintén UAV segítségével történő felmérés lehetőségeit vizsgálták a kataszteri térképezés során. A feltételezés szerint a technikai és gazdasági előnyök miatt érdemes lehet ezt a már hétköznapi földmérési gyakorlatban előforduló módszert használni. A bizonyításra Kampinos környékén végeztek méréseket. 1660 fotót készítettek és 21 illesztőpontot mértek be GNSS technológiával.



8. ábra: A felmért terület: bal oldalt az ortofotó (piros pontokkal a felhasznált fényképek középpontját jelölik); jobb oldalt a DSM (Kurczynski et al. 2016)

A drónos felmérés mellett 100 épület sarokpontjait is bemérték RTK-GNSS és lézeres távmérő segítségével. (Kurczynski et al. 2016)

A fényképekből 20 pont/m²-ből álló pontfelhőt és egy 3 cm felbontású ortofotót is generáltak. Ezekből vektoros térképet vezettek le.



9. ábra: Levezetett vektoros térkép (Kurczynski et al. 2016)

A kapott eredményeket összehasonlították a nyilvántartásban rögzített állapottal. Az eredményt a következő táblázat tartalmazza:

3. táblázat: A kiértékelt adatok összehasonlítása az ingatlan-nyilvántartásban rögzített állapottal (Kurczynski et al. 2016)

	Stereoscopic (n = 25) [m]			Multi-photo (n =112) [m]		
	dX	dY	dp	dX	dY	dp
MEAN	-0.03	0.02	0.15	0.01	-0.03	0.14
STD	0.14	0.10	0.08	0.12	0.12	0.08
RMS	0.14	0.10	0.17	0.12	0.12	0.17

A kiértékelt eredményeket a helyszínen végzett GNSS mérésekkel is összehasonlították:

4. táblázat: A kiértékelt adatok összehasonlítása a helyszíni méréssel (Kurczynski et al. 2016)

	Stereoscopic (n =25) [m]			Multi-photo (n =30) [m]		
	dX	dY	dp	dX	dY	dp
MEAN	0.03	0.00	0.16	-0.01	0.00	0.09
STD	0.13	0.14	0.11	0.06	0.09	0.05
RMS	0.13	0.14	0.19	0.06	0.09	0.10

Az eredményekből megállapították, hogy az drónos mérésből levezetett eredmények pontossága 10 cm az épületek esetében. Ortofotó segítségével a telekhatárokat is megállapították.



10. ábra: A valós földhasználat és a kataszteri térkép tartalma közötti eltérések (Kurczynski et al. 2016)

2.4. Kataszteri térkép előállítása katasztrófa sújtotta területen

Argentínában, a Jujuy tartományban található Volcán városát 2017-ben áradás és földcsuszamlás miatt a várost jelentős károk érték. A katasztrófa kiváltó oka az éves csapadékátlag 50%-nak megfelelő mennyiségű eső hirtelen lehullása volt. (Gutiérrez et al. 2019)

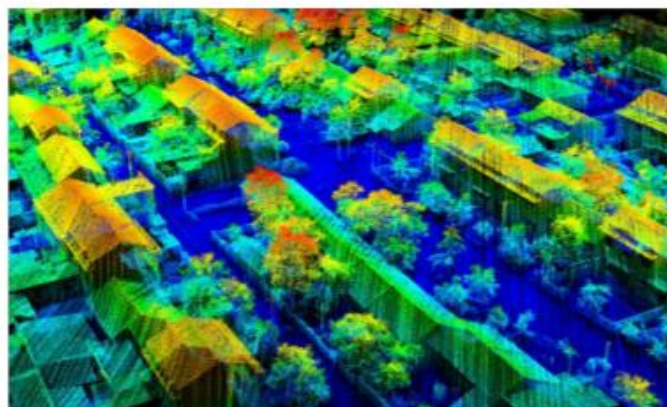
A rekonstrukciós tervek elkészültével az érintett területen feltűnt, hogy jelentős eltérések vannak a természetbeni és az ingatlan-nyilvántartási térképen rögzített állapotok között. Probléma megoldására térkép felújítást rendeltek el az. Mivel a felmérés viszonylag nagyobb területet érintett, így a hagyományos geodéziai felmérések az időhiány és költségek miatt nem jöhettek szóba. Az érintett területet UAV segítségével készítették fotókkal és kihelyezett illesztőpontok segítségével tervezték felmérni. A repülés eredményeként egy ortomozaikot állítottak elő. Későbbiekben ezt manuálisan dolgozták fel. (Gutiérrez et al. 2019)



11. ábra: Az ortofotó és a levezetett ingatlan-nyilvántartási térkép (Gutiérrez et al. 2019)

2.5. Egy kínai LiDAR példa

Az utóbbi évtizedben Kína a világ vezető nagyhatalmává vált. Ez a fejlődés a földmérési tevékenységben is megfigyelhető, ugyanis egyre fontosabbá kezd válni a tulajdoni viszonyok tisztázása szolgáló ingatlan-nyilvántartási térképek létrehozása. Mivel Kínában több százezer falu van így első sorban olyan eszközökben gondolkodnak, melyek kellő pontosságot biztosítanak, nagy területen, gyorsan képesek előállítani az adatokat. A korábbi tanulmányokhoz hasonlóan ők is UAV-t használtak, azonban kicsit tovább fejlesztve a mérési módszert azt LiDAR szenzorral szerelték fel. (He – Li 2020)



12. ábra: A felmért terület pontfelhője (He – Li 2020)

A méréseket 50 méteres repülési magasság mellett, 30% átfedéssel oldották meg. Georeferáláshoz GNSS segítségével bemért pontokat használnak fel. A technológia pontosságáról a kiértékelt pontfelhőn visszamért pontok árulkodnak:

5. táblázat: A bemért pontok és a pontfelhő közti eltérések (*He – Li 2020*)

Serial Number	Plane Error	Error Attribute	Serial Number	Plane Error	Error Attribute
1	0.045	<5cm	14	0.079	5-10 cm
2	0.033	<5cm	15	0.049	<5cm
3	0.051	5-10 cm	16	0.059	5-10 cm
4	0.042	<5cm	17	0.073	5-10 cm
5	0.051	5-10 cm	18	0.013	<5cm
6	0.057	5-10 cm	19	0.074	5-10 cm
7	0.016	<5cm	20	0.024	<5cm
8	0.039	<5cm	21	0.030	<5cm
9	0.007	<5cm	22	0.048	<5cm
10	0.047	<5cm	23	0.021	<5cm
11	0.026	<5cm	24	0.019	<5cm
12	0.078	5-10 cm	25	0.034	<5cm
13	0.049	<5cm			

Mean square error of plane position: $\pm 0.047m$

Megállapítható, hogy a számukra szükséges 5 cm-es hiba határt a LiDAR-os felmérés segítségével el tudták érni. Az elvégzett felméréssel nagy pontossággal, hatékonyan lehet az ingatlan-nyilvántartási térképeket létrehozni.

6. táblázat: Kimutatás a felmérés hatékonyságára a hagyományos felméréshez képest (*He – Li 2020*)

Contents of Work	Traditional Method (GNSS + Total Station)	LiDAR Measurement	Efficiency Ratio of LiDAR VS Traditional Method
Field survey	About 7 days	0.5days	About 14 times
Housekeeping	About 8 days	2 days	About 4 times
Staff input	About 20 days	8 people	About 2.5 times
Comprehensive per capita efficiency	13 parcels / day	101 parcels / day	About 7.8 times

Overall operation efficiency is improved by more than 7 times

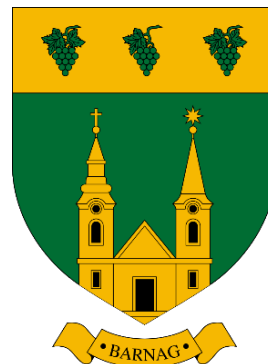
A szakirodalmi példákat látva megállapítható, hogy a rendelkezésre álló új technológiákkal a korábban elképzelhetetlennek vélt ingatlan-nyilvántartási térkép létrehozás és felújítás nagy területen jóval hatékonyabban, gyorsabban tehetjük meg a szükséges pontosság megtartása mellett.

3. Egy magyarországi példa

A következő részben egy magyarországi helyszínt név szerint Barangot mutatom be, ahol az előző részben már bemutatott UAV segítségével végeztek felmérést a Magyar Mérnöki Kamara Feladat Alapú Pályázata keretében. Az ismertetés során a területen elérhető ingatlan-nyilvántartási térkép jellemző adataira is kitérek.

3.1. Barnag

A falu a Balaton-felvidéken és a Veszprémi-fennsík találkozásánál található, Veszprém és Nagyvázsony között. A középkorban a magyar királynék szakácsai laktak itt. A török uralom idején elnéptelenedett. A 18. században azonban új magyar és német lakók települtek ide és emiatt a faluban egyfajta kettőség alakult ki. A településen két temető van a falu keleti és nyugati végén, illetve két templom. Korábban két külön álló településként létezett: Magyarbarnag és Németbarnag néven. A két település lakosait alapvetően az anyanyelv, a nemzetiség és a különböző vallási felekezet választotta el egymástól. A két kis település egyesült és a maga 1201 hektáros területével, 125 fős lakónépességével alkotja napjainkban a falut; Barnagot. (*Barnag 2020*), (*KSH 202*)



13. ábra: Barnag címere



14. ábra: Barnag elhelyezkedése

3.2. Digitális alaptérkép és állapota

Jelenleg a 8/2018. (VI. 29.) AM rendelet (az ingatlan-nyilvántartási célú földmérési és térképészeti tevékenység részletes szabályairól) ad iránymutatást a különböző forrásból származó térképek gyakorlatban történő alkalmazására. A rendelet a térképek keletkezésének megfelelően ad információt a térkép-terepazonos pontok megengedett eltérésére annak függvényében, hogy az adott területen milyen kiindulási adatot vettek alapul a ma elérhető digitális alaptérkép előállításához. (*Takarnet 2020*), (*8/2018 (VI.29.) AM rendelet*)

Földhivatali Portálon elérhető egy úgynevezett *településkereső* szolgáltatás. Ezen belül településekre szűrve kikereshető az adott helység statisztikai adatai mellett a térképi ellátottság is. Itt olyan adatokat találunk, amely az alapként szolgáló térkép keletkezésének körülményeit foglalja magába (felmérés éve, vetület, térkép méretaránya). A metaadatokat ismerve 8/2018. (VI. 29.) AM rendelet 7. mellékletében kikereshető, hogy az adott településen végzendő földmérési tevékenység során milyen hibahatárokkal számíthatunk, dolgozhatunk.

A Barnagról rendelkezésre álló digitális alaptérkép a Fasching Antal (*Varga 2018*) által bevezetett hengervetületben készült. Az országot lefedő 3 darab hengervetület közül Barnag a

középső rendszerben (HKR) 1908-ban mérték fel, amelynek eredménye egy az 1:2880 méretarányban készült grafikus térkép volt. A későbbiekben ezt a térképet vették alapul a digitális átalakítás során.

7. táblázat: Megengedett eltérések a határvonalak jogi és természetbeni állapotának összehasonlítására (8/2018 (VI.29.) AM rendelet)

A) EOTR földmérési alaptérkép alapú adatbázis

Méretarány	Megengedett eltérés egy-egy határvonal helyzetében	Megengedett eltérés a földrészlet szélességében
1:1 000	± 0,35 m	± 0,45 m
1:2 000	± 0,65 m	± 0,90 m
1:4 000	± 1,30 m	± 1,80 m

B) Régi vetületi rendszerű alaptérkép alapú adatbázis

	Méretarány	Megengedett eltérés egy-egy határvonal helyzetében	Megengedett eltérés a földrészlet szélességében
1	1:1 000	± 0,45 m	± 0,60 m
2	1:2 000	± 0,90 m	± 1,20 m
3	1:2 880	± 1,30 m	± 1,70 m
4	1:4 000	± 1,75 m	± 2,40 m
5	1:2 880*		± 3,78 m

* 207/1962(T.6.) ÁFTH (Állami Földmérési és Térképészeti Hivatal) utasítása alapján készült alaptérkép

8. táblázat Megengedett eltérések a határvonalak jogi (térképi) és természetbeni állapotának összehasonlítására a digitális alaptérképek esetén (8/2018 (VI.29.) AM rendelet)

F) A részletpontok meghatározott helyzete és az ellenőrző mérésből számított helyzet közötti megengedett eltérés

Tűrés osztály		
Részletpontok rendősége	T11	T21
R1	± 9 cm	± 15 cm
R2	± 15 cm	± 21 cm
R3	± 18 cm	± 30 cm
R4	± 24 cm	± 57 cm

G) A méretek megengedett eltérése belterületen

T11 tűrés osztály				
Részletpontok rendősége	R1	R2	R3	R4
R1	± 12 cm	± 18 cm	± 21 cm	± 27 cm
R2		± 21 cm	± 24 cm	± 27 cm
R3			± 24 cm	± 30 cm
R4				± 33 cm

A rendelet 7. mellékletében található táblázatok (7. és 8. táblázat) alapján Barnagon a megengedett eltérés egy-egy határvonal helyzetében akár ± 1.3 m is lehet. A földrészletek szélessége esetében ez az érték ± 1.7 m.

Szerencsés esetben ezek az eltérések azonosak egy ingatlan vagy akár egy egész tömb pontjaira. Ez azt jelenti, hogy a birtokhatárok, a feltüntetett épületek egymáshoz viszonyított helyzete a valós állapotnak megfelel, relatív értelemben helyes. Az ingatlan-nyilvántartási célú földmérési tevékenységek során ilyen esetekben a cél a térkép homogenitásának megtartása, vagyis a relatív méretek a mértékadóak a különböző munkafolyamatok során.

Előfordul azonban, hogy egy-egy munkarész elvégzésekor (pl.: épületfeltüntetés) a földmérő nem veszi figyelembe a jelenséget és a direktbe mért pontok koordinátáit használja fel az egyes munkarészek elkészítése során. Így az adott területen elérhető térkép relatív pontosságát elvesztheti és így az inhomogéné válhat. Ha járási földhivatalok nem figyelnek erre a változások

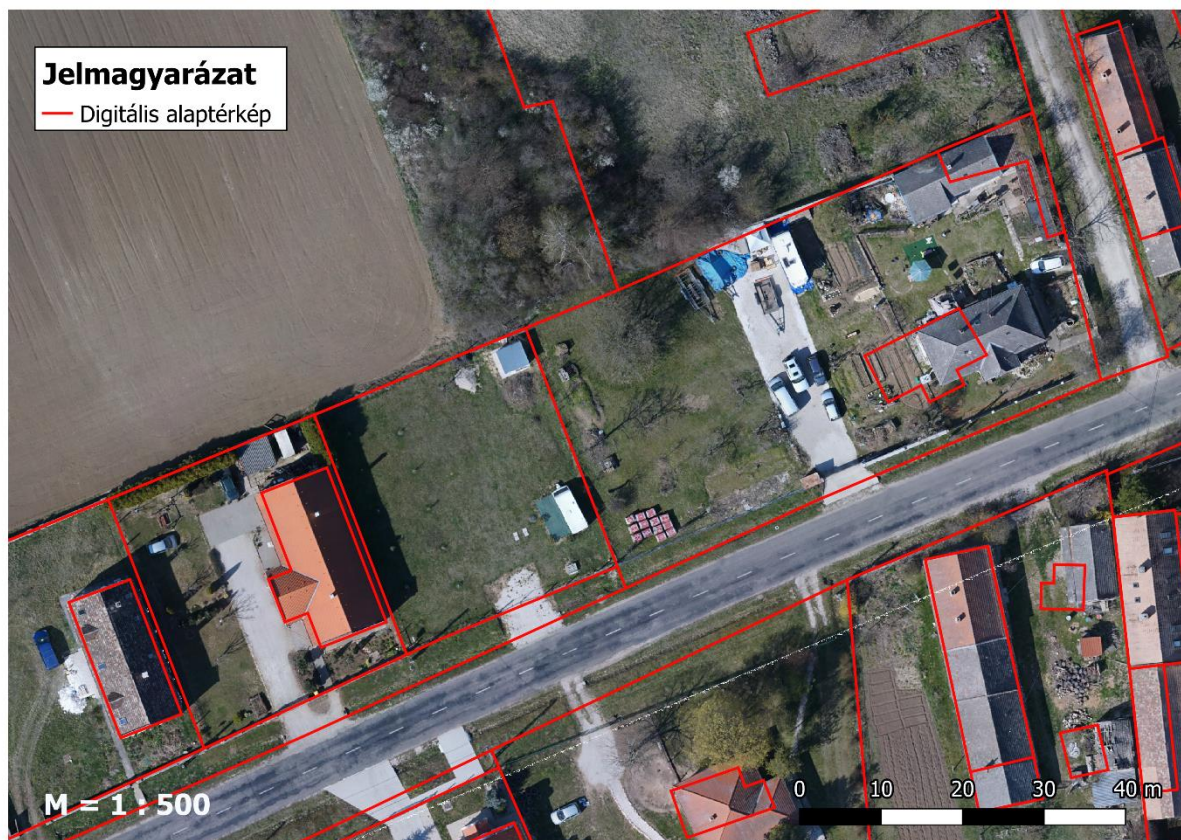
átvezetés során, akkor az alaptérkép idővel elkezd használhatatlanná válni. Tovább fokozza a problémát az is, hogy ha a digitális alaptérkép tartalma nem tükrözi a valós állapotokat. Bizonyos épületek hiányoznak róla, vagy esetleg teljesen más geometriával vannak feltüntetve.

Barnagon is hasonló helyzet alakult ki. A relatív értelemben vett pontosság nem mindenhol van jelen. Szabálytalan az egyes pontokban tapasztalható hiba, és akár 1-2 méteres eltérések is előfordulnak az alaptérkép és valós állapot között.



15. ábra: A digitális alaptérkép és ortofotó a település közepén

A település bizonyos részein, elsősorban a szélén megfigyelhető, hogy vannak olyan épületek, amelyek a valós helyükön vannak, miközben a közvetlen szomszédságukban ez a kijelentés már nem állja meg a helyét (16. ábra).



16. ábra: Digitális alaptérkép és ortofotó a település szélén kialakult helyzet

Fontos megemlíteni, hogy nem csak a napjainkban végzett földmérési tevékenység okoz problémát nyilvántartási térképek minőségi állapotának esetleges romlásában. Vannak területek, ahol már a térképi alapként felhasznált grafikus térkép készítésére vonatkozó engedélyes jogszabályok miatt alakult ki olyan helyzet, ami a napjainkban okoz fejtörést bizonyos területeken.

Ilyen helyzetben vannak például azok a települések is, ahol a grafikus térképeket a 207/1962. (T.6.) ÁFTH és 209/1962. (T.9.) ÁFTH utasítások alapján készítették. 207/1962. (T.6.) ÁFTH utasítás a földrendezéssel nem érintett területek földnyilvántartási adatainak rendezéséről szól. Ebben a térképről lemerített és valós állapotban mért méretek közötti legnagyobb eltérést a földrészlet határvonalai vagy épületek között 2 ölben (~ 3.792m!) és a művelési ágak határvonalai vagy épületek között 3 ölben (~ 5.689m!) adták meg. (Boda 2018), (207/1962. (T.6.) ÁFTH utasítás), (209/1962. (T.9.) ÁFTH utasítás)

A 209/1962. (T.9.) ÁFTH a belterületek és zártkertek kataszteri térképeinek felújítására készült utasítása.

9. táblázat: 209/1962. (T.9.) ÁFTH utasításba foglalt megengedett eltérések a valós és térképezett állapotok között (209/1962. (T.9.) ÁFTH utasítás)

	I. r.	II. r.	III. r.
földrészlet közterületi határvonalnál	1,0 d	1,5 d	–
belső pontoknál	1,5 d	2,0 d	2,5 d
ahol d a vizsgált távolság értéke m egységben			
	40 m	100 m	200 m
M = 1 : 1000	0,25	0,4	0,5
M = 1 : 2000	0,5	0,8	1,0

Az utasításhoz tartozó mellékletben (9. táblázat) látható, hogy 40 m-en belüli I. rendű részletpont (pl.: belterületi határvonal, földrészletek és közterületek határvonalai és a kisajátított területek állandósított pontjai) közötti távolság terepi és térképi mérete közötti eltérés 25 cm, ugyanez 200 m alatti III. rendű részletpontok (belterületi határvonal, földrészletek, nyomvonalas létesítmények nem állandósított, de jól azonosítható pontjai között 2,5 m lehet. (Boda 2008), (209/1962. (T.9.) ÁFTH utasítás)

Az ilyen és ezekhez hasonló problémákkal küzdő területeken, ahol a digitális alaptérkép a fentiekben felsorolt okok miatt gyakorlatilag használhatatlanná vált, érdemes elgondolkodni az alaptérképek felújításán. A [2. fejezetben](#) nemzetközi szakirodalmi példák alapján ismertettem több példát a leghatékonyabb módszerek közül, amelyeket a felmérésre lehet használni. A következő fejezetben nézzük át, hogy Magyarországon miért és hogyan végeztek el hasonló felmérést.

3.3. Felmérés és feldolgozás

A 2020-ban Magyar Mérnöki Kamara által kiírt Feladat Alapú Pályázatára a Geodéziai és Geoinformatikai Tagozat a „Módszertani útmutató az elavult ingatlan-nyilvántartási térképek korszerű technológiákkal végzett felújításához” című projekttel indult. A munka keretein belül pontatlan térképi alapadatokkal rendelkező területek ellentmondásainak feloldásához szükséges térképfelújítási módszertan kidolgozásával kezdtek el foglalkozni. Az elsődleges kérdés az volt, hogy a ma elérhető korszerű adatnyerési módszerekkel (légi fotogrammetria, lézerszkennelés), hogyan lehetne egy költséghatékony, gyors felmérést elvégezni. A kérdés megválaszolására a projekt keretein belül megvizsgálták a különböző módokon nyert mérési adatok pontosságát, ezek beilleszkedését a jogi szabályozásba. A vizsgálatokra korábbiakban bemutatott Barnagon végzett felméréseket a BME Általános- és Felsőgeodézia Tanszéke, az ÓE Alba Regia Műszaki Kar Geoinformatikai Intézete, a Geodézia Kft. és a Pannon Geodézia Kft. (FAP 2020)

A BME ÁFGT drón segítségével készített fényképek és RTK-GNSS-el bemért illesztőpontok segítségével mérte fel Barnagot. A mérés során 1000 darab fotót készítettek ferde, illetve nadír irányból. A repülési magasság 50-70 méter között volt. A feldolgozás során ortofotót és pontfelhőt is előállítottak. Utóbbit 5-6 cm-es pontossággal határozták meg. A dolgozatom elkészítéséhez ezt a pontfelhőt kaptam meg és használtam fel.

4. Pontfelhő tartalmának egyszerűsítése

A következő részben a felmérés és feldolgozás eredményeként kapott pontfelhő előkészítését mutatom be. A célom az volt, hogy az eredetileg 450 millió pontot tartalmazó pontfelhőt az ingatlan-nyilvántartási térkép létrehozásához szükség információk (épületek, kerítések) megtartásával kezelhetőbb méretűvé tegyem. Erre két módszert használtam fel; a pontfelhő osztályozást és normalizált pontfelhő létrehozást.

4.1. Előkészítés

A munkát egy a pontfelhők kezelésére fejlesztett nyílt forrású programban a *CloudCompare*-ben kezdtem el. A szoftverről érdemes megemlíteni, hogy különböző formátumban tárolt pontfelhők beolvasása mellett, 3D objektumokat is képes kezelni. A beolvasott adatokkal számos műveletet végre lehet hajtani a beépített eszközök segítségével.

A feldolgozást Barnag felmérésének eredményeként kapott pontfelhő beolvasása után a zajsűrűssel kezdtem. Ezt követően a pontfelhőt újramintavételeztem. A feldolgozást kétféleképpen is elvégeztem; 5 és 25 cm-es ponttávolságokkal. Ezzel a lépéssel a pontfelhő méretét az első esetben 140 millió pontra sikerült lecsökkentenem, a második esetben 8.5 millió pontra. A kérdés az, hogy miért volt szükség arra, hogy kétféle módon is elvégezzem az újramintavételezést.

4.2. Osztályozás

Az ingatlan-nyilvántartási térkép pontfelhő alapú levezetéséhez első sorban az épületekhez, kerítéshez tartozó pontokat tartalmazó adatokra lenne szükségünk. A pontfelhő tartalmának szétválogatását a pontfelhő osztályozásával lehetne megoldani. Erre a feladatra vannak különböző programok, az egyik ilyen például a *LAStools*.

4.2.1. LAStools

A rapidlasso GmbH által fejlesztett szoftver (*LAStools 2020*) alapvetően LiDAR adatok gyors és hatékony feldolgozására fejlesztették ki. A program nagy előnye, hogy parancssorból is futtatható. Így akár előre megírt rutinokat is használhatunk hozzá. A parancssori futtatás mellett lehetőség van grafikus felhasználó felület használatára is, amelyet akár a QGIS-el is összekapcsolhatunk.

Alapvetően nyílt- és zárt forráskódú részekből áll. A zárt részek korlátozva vannak ugyan, de használhatóak oktatási és tesztelési célokra. Ez azt jelenti, hogy az ingyenes változat, csak 1.5 millió pontig kezelik tökéletesen az állományokat. Ha az egész falura szeretnénk osztályozást végezni, akkor korlátok miatt szükséges a pontfelhő méretének csökkentése. Az előző részben ezért végeztem el kétféle beállítással az újramintavételezést, hogy a szoftver alkalmazhatóságát nagyobb területek esetében is meg tudjam vizsgálni. Az osztályozáshoz így a 25 cm újramintavételezésű pontfelhőt vettem alapul. (*Neuberger 2020*)

4.2.2. Feldolgozás

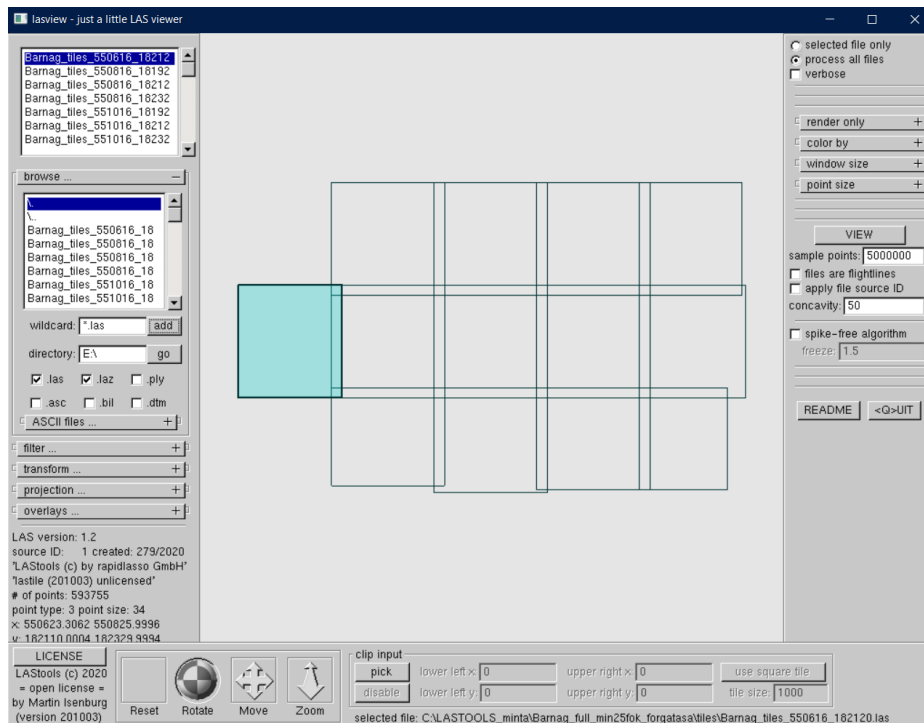
A pontok mennyiségére vonatkozó 1.5 milliós korlátozást a *LAStools*-ban csempékre osztással tudjuk kezelni. A szoftver nyílt forráskódú részében található a *lastile* modul. Segítségével a betöltött területet kisebb részekre, úgynevezett „csempékre” bonthatjuk és menthetjük el.

A feldolgozást egy nulladik lépéssel kezdtem. Az állományt *CloudCompare*-ben töltöttem be majd +26°-al elforgattam azt. A feldolgozás szempontjából így optimálisabb helyzetbe tudtam transzformálni a pontfelhőt.



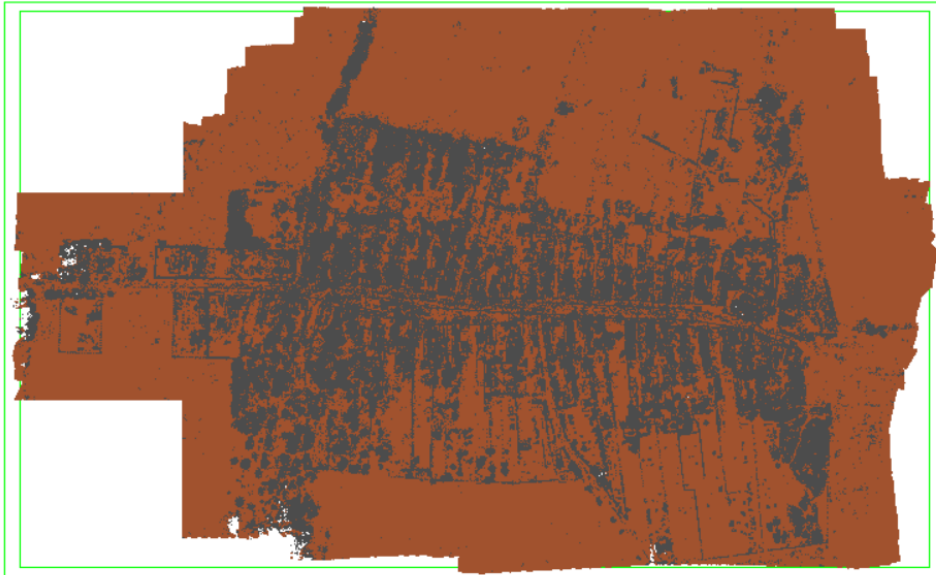
17. ábra: A LAStools-ba betöltött, elforgatott pontfelhő

A forgatás eredményét LAS kiterjesztésben mentettem el, majd a már ismertett *lastiles* modulba töltöttem be. A programban a paramétereket úgy állítottam be, hogy az átfedésekkel együtt az egyes csempéken található pontok száma ne haladja meg a 1.5 milliót. A feldolgozás eredményeit egy külön mappába mentettem ki LAS formátumba.



18. ábra: A pontfelhő felosztása a 1.5 milliós pontszám limitnek megfelelően

A következő lépés a talajpontok leválogatása volt. Ez elengedhetetlen az osztályozás végrehajtásához. A talajszint definiálását a *lasground* modul végzi a pontok leképződésének helye alapján.



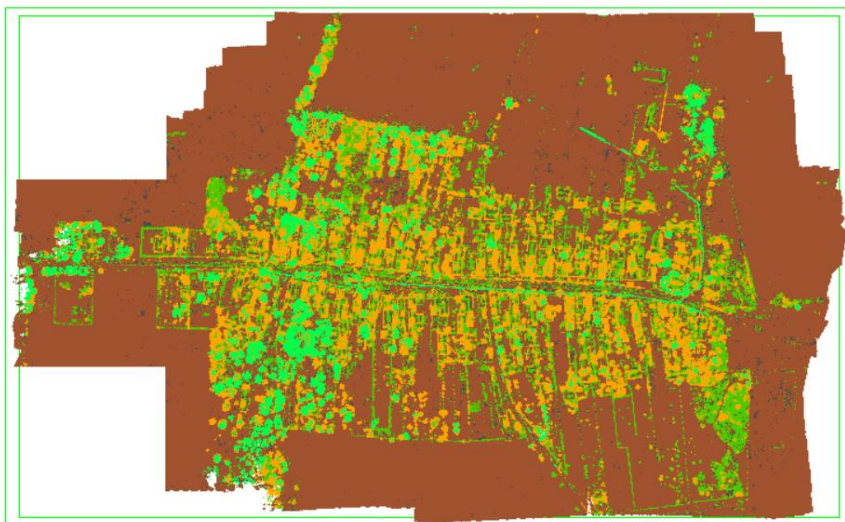
19. ábra: A leválogatott talajpontok (barna)

A *lasground*-on belül lehetőségünk van a talajpontokhoz viszonyított magasságok számítására is. Erre az osztályozás miatt van szükség. Ha kihagyjuk a beállítást, akkor a *lasheight* modulban is van lehetőségünk utólag kiszámítani az egyes pontokhoz tartozó értékeket.

A *lasheight* lehetőséget ad a talajszint feletti pontok távolság alapú osztályozására is. A beépített osztályozó modul használata előtt kipróbáltam ezt a funkciót is. Pár próbálkozás után a végeredmények alapján a következő csoportokat sikerült kialakítanom:

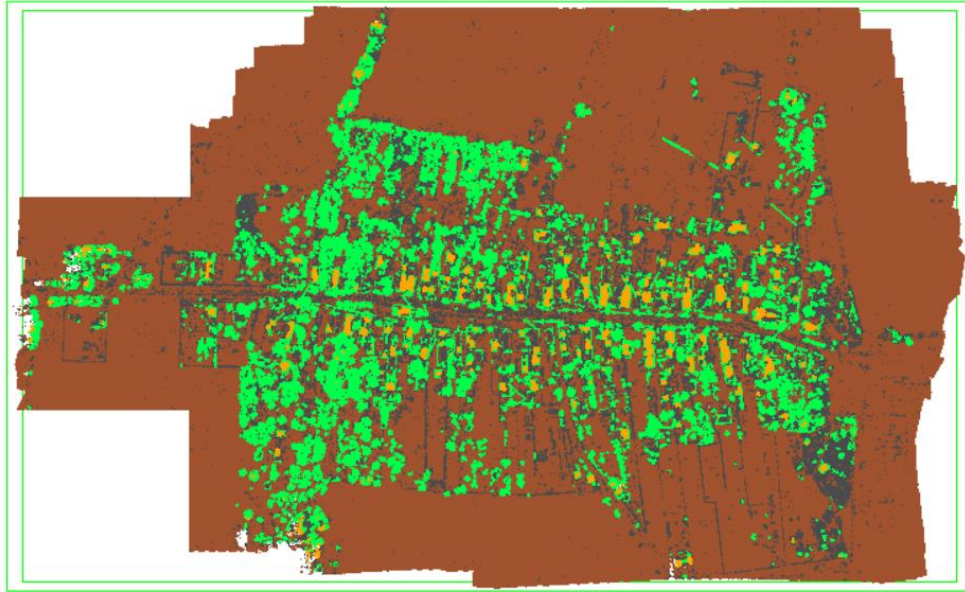
10. táblázat: Az osztályozáshoz felhasznált adatok

Osztályok	Határok [m]
Növényzet (low vegetation)	0 – 1.5
Épület (building)	1.5 – 7
Növényzet (high vegetation)	7 -



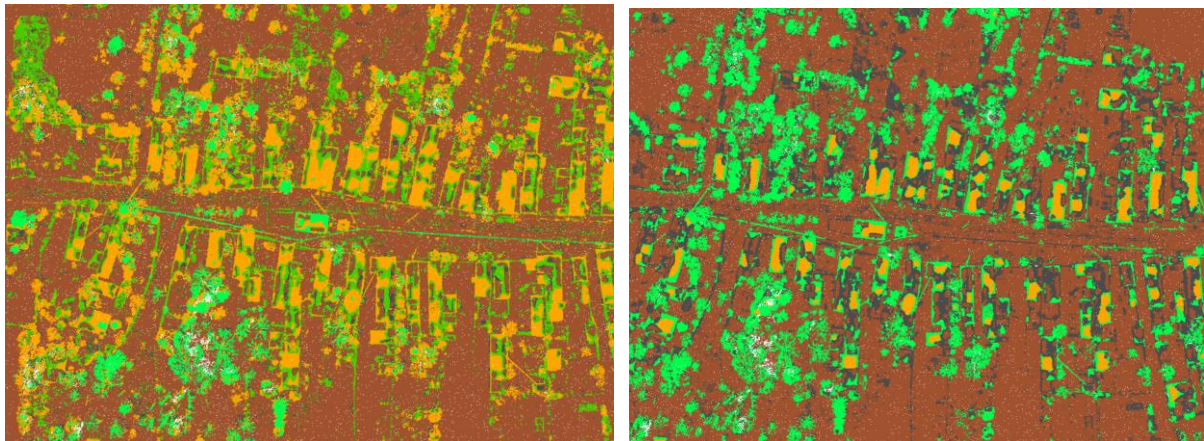
20. ábra: A talajpontokhoz viszonyított magasságok alapján történő osztályozás eredménye

Végül az osztályozást a *lasclassify* modul segítségével is elvégeztem. Sajnos itt a program nem enged túl sok szabadságot a paraméterezésben. A beépített osztályozó modul működésének alapelve tulajdonképpen az „hogya a talajszinthez képest egy bizonyos magasság feletti szomszédos pontok érdekességét vizsgálja” (Neuberger 2020). Ha a szomszédos pontok egyenletes síkot alkotnak épületként kerülnek detektálásra. Ha a síkhoz képest a szórásuk nagyobb, akkor növényzet kategóriába kerülnek a pontok. (Neuberger 2020)



21. ábra: A beépített osztályozó modul eredménye

Vizsgáljuk meg a két módszer eredményeit közelebbről. Alapvetően a magasság alapján történő osztályozás szemmel láthatólag jobb végeredményt adott, mint a beépített osztályozó modul.



22. ábra: A lasheigh (bal oldal) és lasclassify (jobb oldal) eredménye közelebbről

Sajnos a módszer eredményének korrigálása a felhasználhatóság szempontjából további utómunkát igényelne. Ezt a *lasview*-on belül elvégezhetnénk, ugyanis a pontfelhő nézegetőben lehetőség van a már osztályozott pontok kategóriába sorolásának módosítására. Fontos megjegyezni, hogy a módosítások mentését csak 1 milliónál kevesebb pontot tartalmazó pontfelhőn végezhetnénk el. Ehhez a pontfelhőnk méretét tovább kellene csökkenteni mennyiségben vagy terjedelemben. Ezt a lehetőséget elvettem, mert már így is jelentős mértékű adatot kellett eldobni az újramintavételezés során azért, hogy a *LAStools* osztályozási lehetőségeit megvizsgáljam.

4.3. Normalizált pontfelhő előállítása

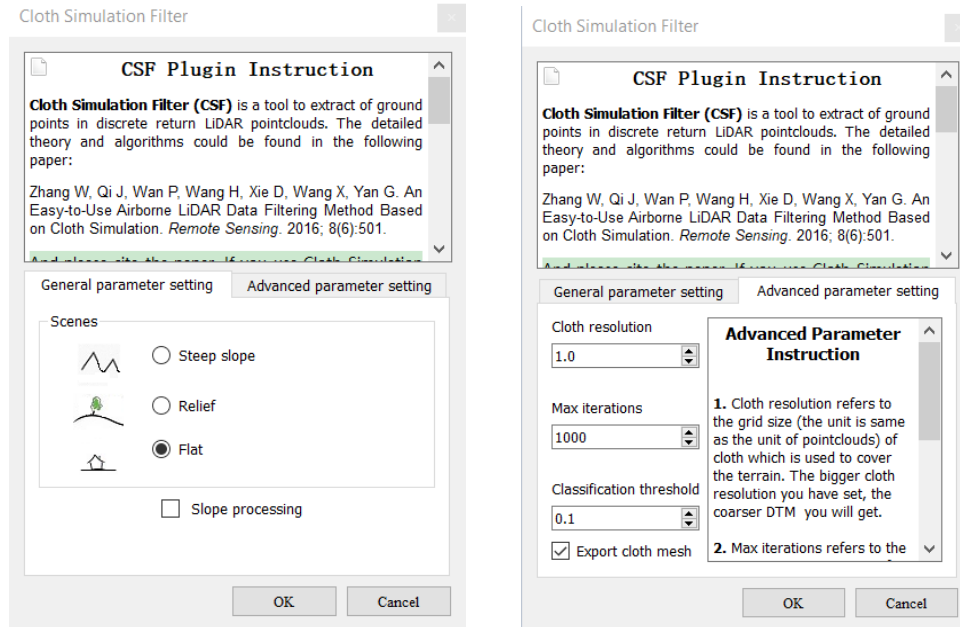
Mivel az osztályozás a feldolgozó szoftver korlátai miatt nehézségekbe, ezért más irányból próbáltam megközelíteni a problémát. A továbbiakban az [előkészítés](#) fejezetben már bemutatott 25 cm-es ponttávolságon alapuló újramintavételezett állomány helyett sűrűbb, 5 cm-es értékkel feldolgozott pontfelhőt vettem elő.

A 140 millió pontból álló pontfelhő alapvető problémája, hogy számos a feldolgozás szempontjából felesleges (talaj, növényzet) pontot tartalmaz. Ingatlan-nyilvántartási térkép levezetéséhez alapvetően egy metszetre lenne szükség. A legtöbb feldolgozó szoftver képes ezek létrehozására, azonban ez most nem járható út, hiszen egy egész településről beszélünk. A nagyobb területen elhelyezkedő falut a domborzati viszonyok miatt nem tudnánk egy metszősíkkal elmetszeni. Ha ezt a domborzati hatást valahogyan ki tudnánk küszöbölni, akkor már a manuális feldolgozásra alkalmas részeredményt tudnánk előállítani.



23. ábra: A CloudCompare-be betöltött pontfelhő

A pontfelhő tartalmának további redukálásához a már újramintavételezett és zajszűrt pontfelhőn, a *CloudCompare*-ben található *Cloth Simulation Filter* plugin-t használtam fel. A modul segítségével a *LAStools*-hoz hasonló módon a talajszinthez tartozó pontokat lehet szűrni. A plugin-ben lehetőség van számos paraméter beállítására. Beállítható, hogy a vizsgált területen a futtatás után milyen sűrű pontfelhőt akarunk, mennyi pontra végezzen iterációt és milyen küszöbértéknél válassza el a talaj pontjait a pontfelhőtől. Az általam használt beállításokat a 24. ábra mutatja.



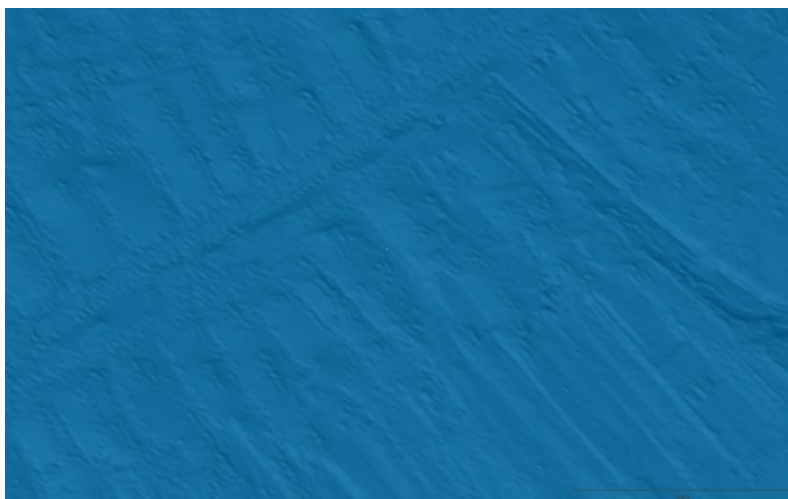
24. ábra: CSP Plugin beállításai a talajpontok leválogatására

Ezekkel a paraméterekkel sikerült előállítanom a talajsínt és az azon kívül eső pontokat tartalmazó pontfelhőket.



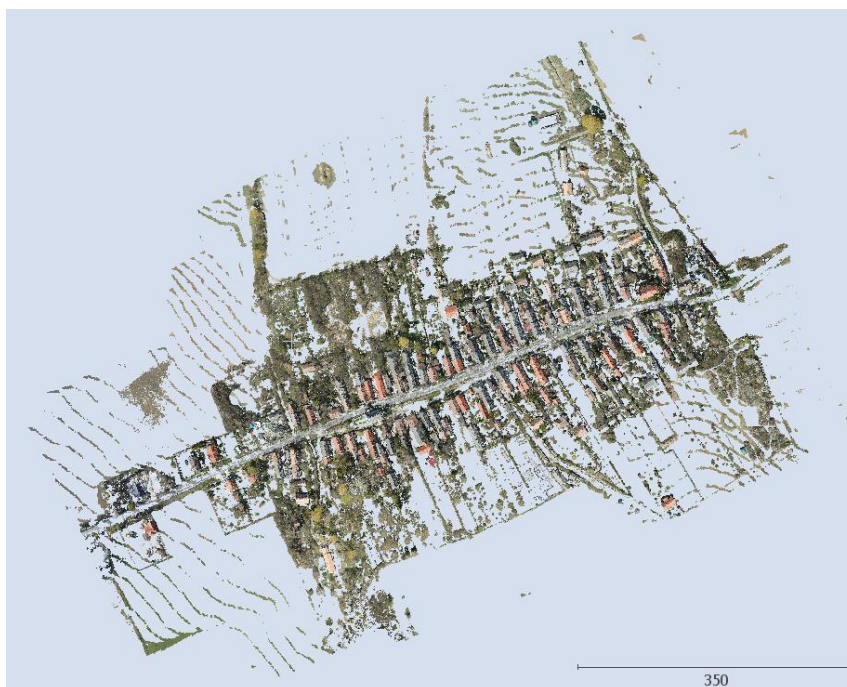
25. ábra: A leválogatott talajpontok

A talajpontok a leválogatás mellett *CSF pluginnal* mesh-t is illeszttem a talajpontokra.



26. ábra: A talajpontokból generált mesh közelebbről

A talaj és egyéb pontokat tartalmazó pontfelhőt két külön fájlba LAS kiterjesztésben mentettem el.



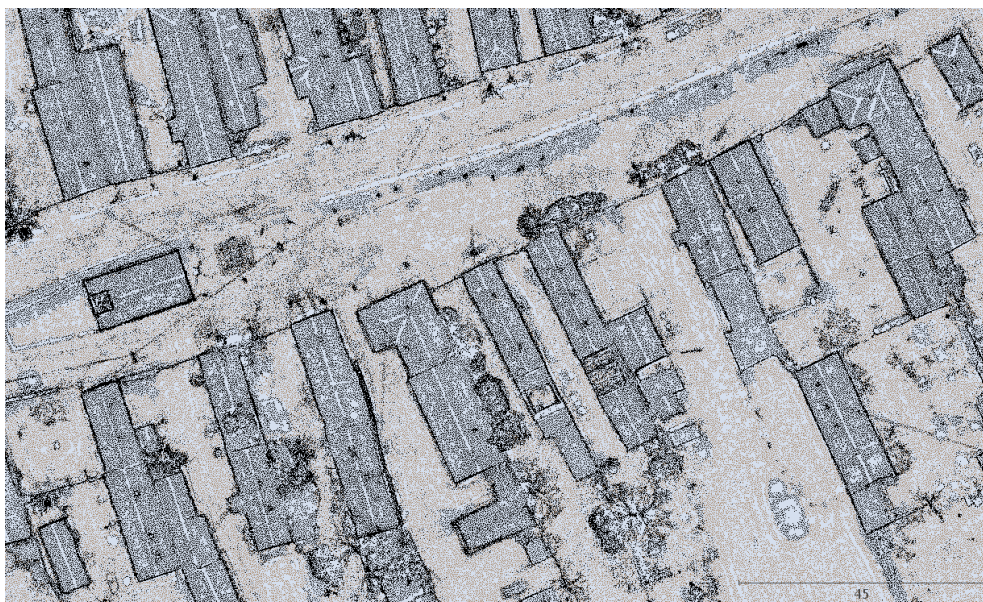
27. ábra: A "nem talajpontok"

Az épületeket, fákat, vezetékeket tartalmazó pontfelhőt a következő (28. ábra) mutatja be közelebbről.



28. ábra: Közelebbről

Miért is jó, ha a talaj és nem-talaj pontokat eltudjuk választani egymástól?



29. ábra: Az osztályozott pontfelhők (barna - talaj; fekete - egyéb)

A *CloudCompare*-ben lehetőségünk van objektum-pontfelhő és pontfelhő-pontfelhő távolságok számítására. A számított távolságokat az adott pontfelhőn belül tárolhatjuk, menthetjük. Az értékeket akár eltérés térkép készítésére is felhasználhatjuk. A program a különbségeket külön értéként tárolja, és ha a pontfelhőt ASCII-ba kimentjük az értékeket külön az X,Y,Z koordináták és RGB adatok mellett egy külön oszlopban találhatjuk meg.

A *CloudCompare*-ben a talajra illesztett mesh és a nem-talaj pontokat tartalmazó pontok távolságát kiszámítottam (*Cloud/Mesh Dist*), előállítva a pontok terepéhez, ha úgy tetszik DTM-hez viszonyított távolságát. A végeredményt mentettem TXT formátumba.

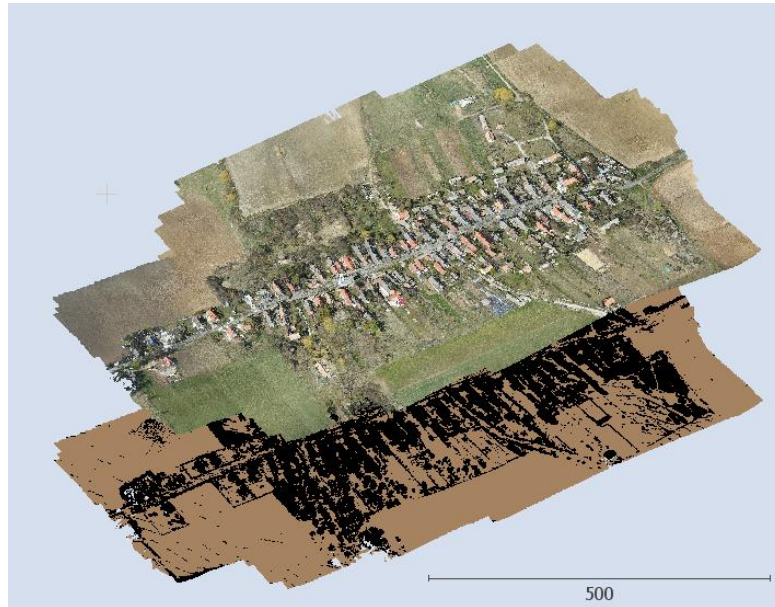
A domborzati viszonyoktól függetlenített pontfelhő előállításához már csak az ASCII fájlban található „Z” koordinátákat kellett megcserélni az eltéréseket tartalmazó oszloppal. Erre feladatra a *gawk* -t használtam fel. Ez a program szabadon elérhető, parancssorban futtatható,

speciális szövegfájl feldolgozó nyelv. Alapvetően elválasztójellel tagolt szövegfájlok feldolgozására ideális. Használata egyszerű; mivel csak az adott sorok feldolgozását kell megadnunk. A sorok és mezők léptetését maga a program végzi. (Robbins 2015)

A *gawk*-ban a vesszővel elválasztott oszlopok felcserélését a következő sorral oldottam meg:

```
gawk -F "," '{ print $1, $2, $7; }' input.txt > output.txt
```

Itt látható, hogy a 3. oszlop helyére a 7. oszlopot raktam. 4-5-6 oszlopban található színre vonatkozó információkat elhagytam ugyanis a későbbi feldolgozás során erre nem lesz szükség. A módosított pontfelhőt szemléltetés az eredeti pontfelhővel együtt betöltöttem a *CloudCompare*-be. A végeredményt a következő két ábra mutatja.



30. ábra: Az eredeti pontfelhő és a normalizált pontfelhő



31. ábra: Az eredeti pontfelhő és a normalizált pontfelhő

5. Manuális felhasználási lehetőségek

Az [előző részben](#) bemutatott lépésekkel a rendelkezésre álló pontfelhőt előkészíthetjük egy esetleges manuális feldolgozásra is. A normalizált pontfelhőt CAD környezetbe betöltve, egy megfelelő magasságban elhelyezkedő metszősikkal teljes mértékben felhasználhatjuk a vektoros állományunk elkészítése során. A következő fejezetekben ennek a munkafolyamatnak mutatom be az egyes lépéseit néhány, az *AutoCAD Civil 3D 2020*-ban is megtalálható kiegészítő funkció segítségével.

5.1. Előkészítés és betöltés AutoCAD Civil 3D-be

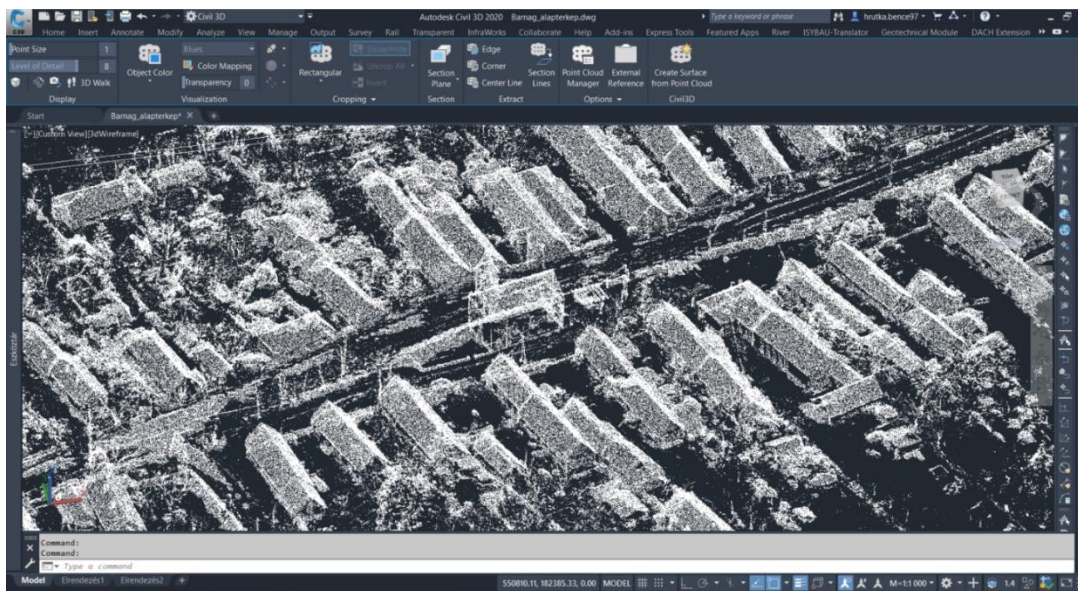


32. ábra: AutoCAD Civil 3D logója

Az *AutoCAD* alapvetően az építőmérnöki gyakorlatban használt szoftver. Számos tervezést igénylő feladat megoldható benne. A tervezés mellett geodéziai munkavégzésre is használható funkciókat is megtalálhatunk a programban. A pontfelhők elterjedésével a *Civil 3D*-t is felkészítették pontfelhők betöltésére és kezelésére. Így *AutoCAD Civil 3D*-ben is lehetőségünk van RCP formátumban tárolt pontfelhőket behívni és felhasználni a különböző rajzok (metszetek),

modellek elkészítése során. Az RCP fájlformátumot alapvetően az *Autodesk* másik szoftverével az *Autodesk Recap* segítségével tudjuk létrehozni. A szoftver lézerszkenneléssel előállított állományok összeillesztését, georeferálását teszi lehetővé.

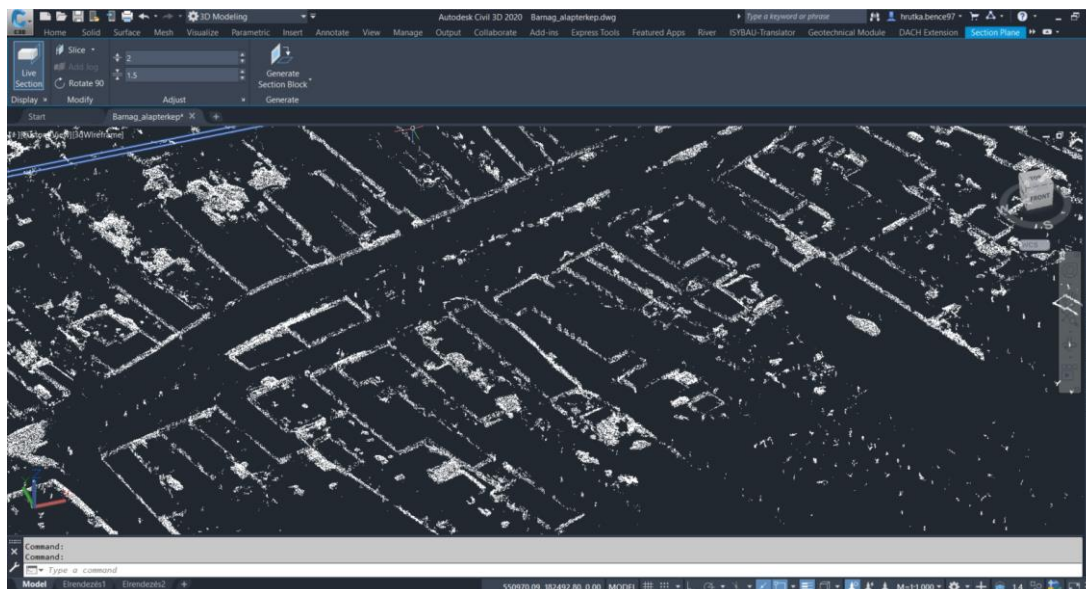
A munkát az RCP fájl létrehozásával kezdtem. A nem-talajpontokból álló LAS kiterjesztésű pontfelhőt Barnagról *Recap*-be töltöttem be, majd elmentettem RCP-be. Ezt követően az új állományt *Civil 3D*-be nyitottam meg.



33. ábra: A betöltött pontfelhő

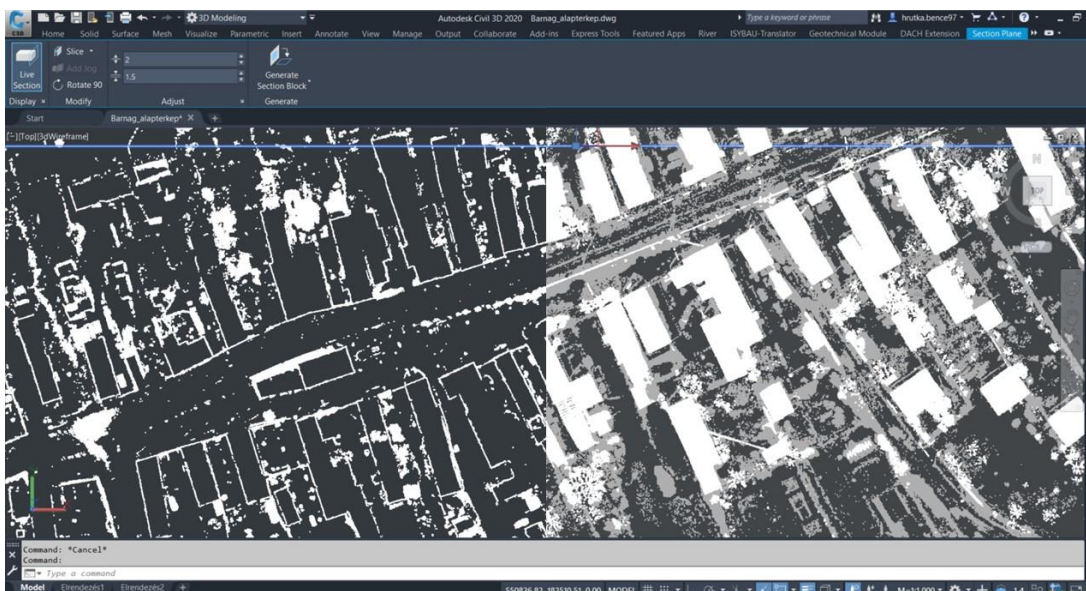
5.2. Metszet készítés

Az *AutoCAD*-ben, az importált pontfelhőre kattintva lehetőség van a *Section plane* fülön belül metszeteket létrehozni. A mostani példában én egy sáv metszetet határoztam meg 1.5 és 2 méter között. Az eredményt a 35. ábra mutatja.



34. ábra: A sáv metszet

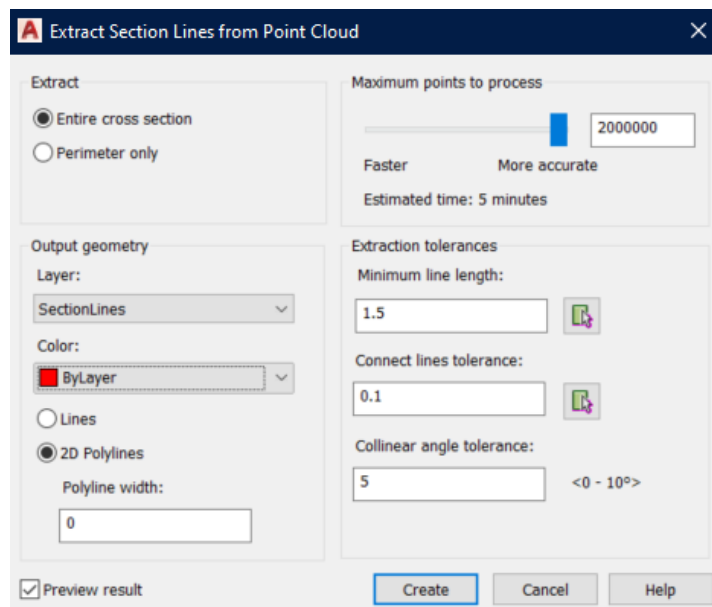
Az így előállított metszetet gyakorlatilag már felhasználhatnánk az alaptérkép elkészítésére. A szerkesztéshez csupán egy olyan helyi koordináta-rendszert kellene létrehozni, amely magával a metszősíkkal egybe esik. A programban található ki- és bekapcsolható snap funkció segítségével az épületek falzatának vonalát a rendelkezésre álló rajzi eszközökkel felülnézetben meg lehetne rajzolni.



35. ábra: Jobbra: a 0.5 méteres sávra redukált pontfelhő; balra: a teljes pontfelhő felülnézetből

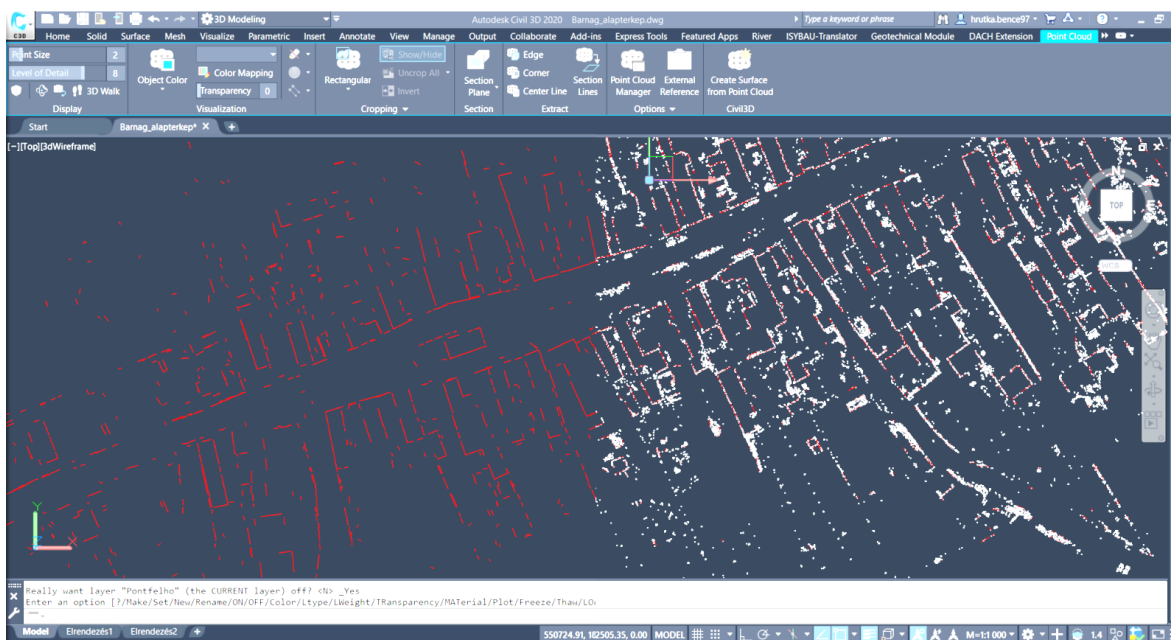
5.3. Egyéb AutoCAD eszközök

Belátható, hogy egy nagyobb település esetén az összes vonal megrajzolása igen fáradságos munka. *AutoCAD Civil 3D*-ben viszont van lehetőség a folyamat felgyorsítására. Amennyiben a megfelelő metszősíkot meghatároztuk, a szoftverben található beépített *Section Lines from Point Cloud* funkció segítségével adott rétegen, minimális vonalhossz és kapcsolódási küszöb beállítása mellett lehetőségünk van a rajzolás felgyorsítására. A vizsgált területen kis tesztelés után én is lefuttattam egy ilyen parancsot a következő paraméterekkel:



36. ábra: A beállított paraméterek

A fenti beállításokkal a következő eredményt kaptam az AutoCAD-ben:



37. ábra: A program által meghatározott egyenesek (piros) az adott metszősíkon

Az elsőre talán biztató eredményt ha közelebről megvizsgáljuk, akkor láthatjuk, hogy nem teljesen tökéletes.



38. ábra: Az eredmény közelebbről

A funkció az épületek falait helyenként egész jól, viszont valahol darabokban találta meg. A falak, kerítések mellett számos egyéb rövidebb hosszúságú egyenest is detektált. A kialakult hibákat korábban az *AutoCAD Map* szoftver segítségével lehetett javítani. Ma már a *AutoCAD Civil 3D*-ben is elérhetjük a *Map*-ben található rajztisztítási eszközöket – gyakorlatilag egy bővítményként hozzá tudjuk adni a programhoz, ha nem telepítettük külön.

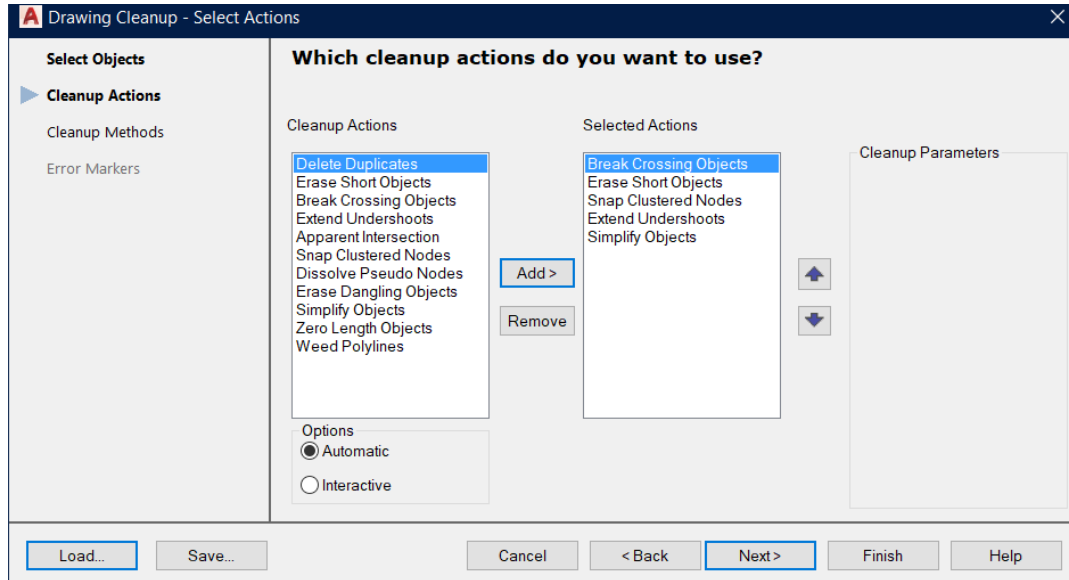
A vektoros felülnézeti rajzok tisztítása során a cél digitális térképek létrehozásakor kialakuló nehezen észrevehető hibák detektálása és javítása. Az előforduló a rajzi hibákat a következő táblázat mutatja be:

11. táblázat: Rajzi hibák típusai (Siki 2016)

	Rajzi hiba	Magyarázat
1.	Duplikált rajzelemek(duplicate objects)	Részben vagy teljesen átfedő rajzelemek. Például két azonos végpontokkal bíró vonalelem.
2.	Túl rövid rajzelemek(short objects)	A rajzban előforduló a megadott toleranciánál rövidebb rajzelemek.
3.	Alul lógás (undershoots)	Egy vonal nem éri el a megadott tolerancián belül eső cél rajzelemet.
4.	Túl lógás (overshoots)	Egy vonal, mely a toleranciát nem meghaladó mértékben nyúlik túl egy cél rajzelemmel alkotott metszésen.
5.	Metsződő rajzelemek (crossing objects)	Két egymást metsző rajzelem.
6.	Pontatlan csatlakozások(clustered nodes)	A megadott toleranciánál közelebb eső kettő vagy több elem végpont.
7.	Álcsomópontok(pseudo nodes)	Egy olyan pont melyben csak két rajzelem végpontjai csatlakoznak.
8.	Lógó vonalak(dangling objects)	Olyan rajzelemek, melyek egyik végéhez nem csatlakozik másik rajzelem (pl. zsákutca).

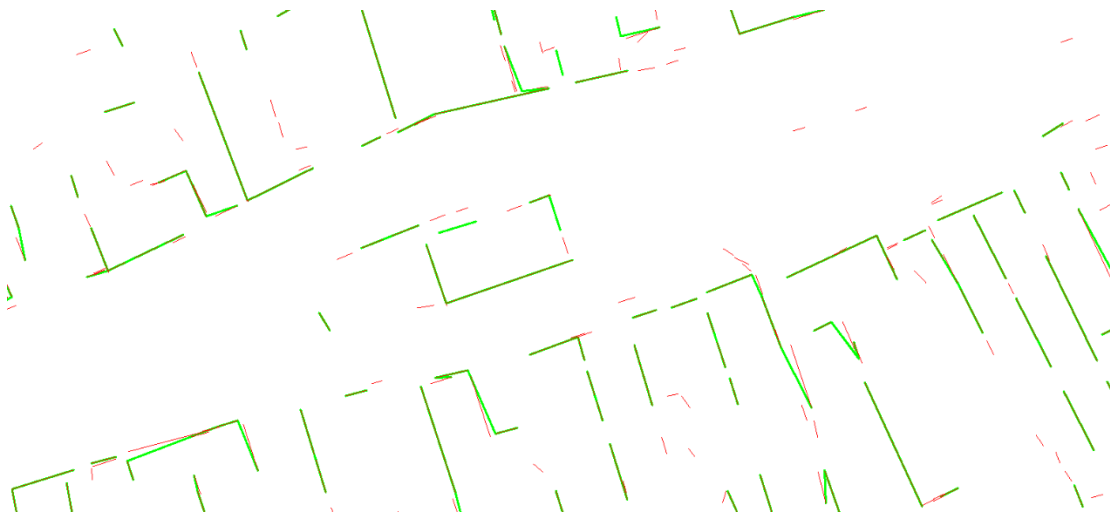
A *Civil 3D*-ben a *Planning and Analysis, Tools* fülön belül található a *Drawing and Cleanup* menüpont. Itt számos beállítás közül választhatunk. A vizsgálandó réteg megadása mellett lehetőségünk van kiválasztani az output réteg nevét, különböző tisztítási módszereket magunk állíthatjuk be és paraméterezhetjük. Megadhatjuk, hogy a vizsgált réteg elemeivel mit csináljon a program (pl.: dobja el) és hogy milyen típusú objektumot hozzon létre az elemzés után.

A 37. ábrán láthattuk, hogy a vizsgált területen a legjelentősebb hiba túl rövid rajzi elemekből adódik. A paraméterezés során ezeket az *Erase Short Object* segítségével 1.5 m tolerancia érték beállításával töröltem. Az állományban kialakult túllövések javítására a *Break Crossing Object* és előző funkció együttes használatával próbáltam megoldást találni. Az esetlegesen kialakult alullövéseket az *Extend Undershoots* megadásával (0.20 cm tolerancia) szüntettem meg. A feleslegesen kialakult töréspontok megszüntetésére egy 20 cm-es határt állítottam be. (Siki 2016)



39. ábra: Az általam használt eszközök

A paraméterezést követően lefuttattam a rajz tisztítást, amelynek eredményt a következő ábra mutatja be.



40. ábra: A rajztisztítás eredménye (zöld) és a kiinduló állomány (piros)



Látható, hogy a rövid rajzi elemeket nagyrészt sikerült eltüntetni. A túl- és alul lógásokat, ahol lehetett ott a program megszüntette. A paraméter beállítások variálásával a rajztisztítás még lehetne finomítani; például nagyobb tolerancia értékkel is kereshetnénk rövid egyeneseket, viszont ebben az esetben számolnunk kell azzal, hogy a már megtalált falsíkokat jelölő egyeneseket is törölhetünk az állományunkból. Belátható, hogy az alaptérkép alapjául szolgáló állományt elő tudjuk állítani, azonban még számos utómunkára lenne szükségünk. Az előzőekben bemutatott *AutoCAD* funkciók segíthetik a feldolgozást, de teljes automatizálást nem tudjuk megvalósítani velük.

6. Automatizálási lehetőség vizsgálata

Az előző részben bemutatam, hogy egy előkészített pontfelhő segítségével az *AutoCAD*-ben, hogyan lehet az alaptérkép előállítás bizonyos lépéseit automatizáltan elvégezni. Összeségében elmondható, hogy eléggé korlátozottak a lehetőségeink, a beépített funkciók segítségével a manuális feldolgozást nem tudjuk kiváltani, csupán csak segíteni tudjuk azt.

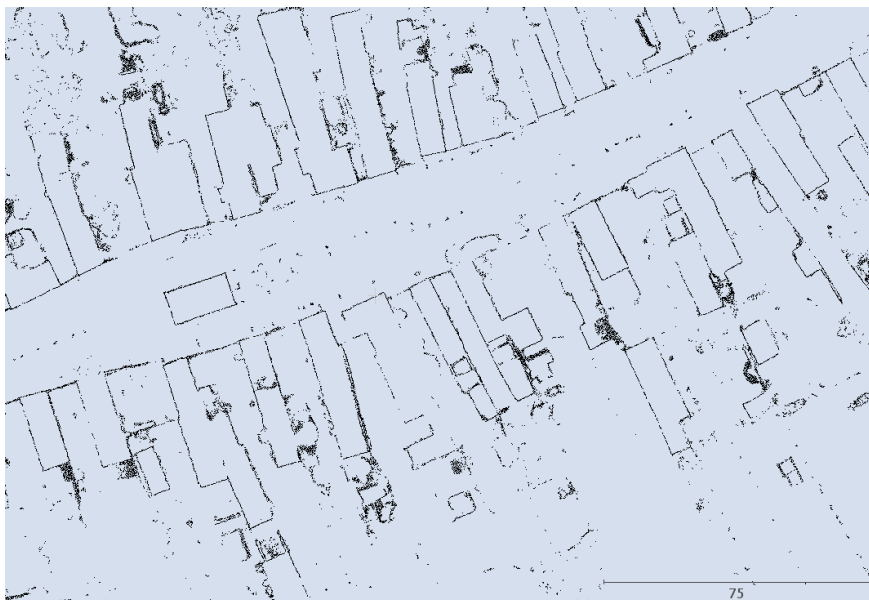
Ebben a részben egy kicsit másik irányba kezdtem el vizsgálni. A sorozatos egyenes illesztés módszerének felhasználását vizsgáltam meg a pontfelhő alapú ingatlan-nyilvántartási térkép előállítása során.

6.1. 2D metszet előállítása

A korábbi fejezetekben láthattuk, hogy a normalizált pontfelhő egy-egy megfelelő magasságban elhelyezett metszősíkjai hogyan tudják segíteni a manuális feldolgozást. A metszetek felülnézetben az egy helyen besűrűsödő pontjaival az épületek falazatát, kerítéseket szinte tökéletesen kirajzolják. A manuális feldolgozás során gyakorlatilag ezt az információt használjuk fel, hiszen az kirajzolódó vonalakat lényegében átrajzoljuk (vektorizáljuk).

A feldolgozás során így csak a 2D információra van szükség, a magasság szerepe a megfelelő metszősík, sáv megválasztása után háttérbe kerül. Így további egyszerűsítésre van lehetőségünk; elegendő ugyanis az általunk megadott magassági értékek között lévő pontokat megtartanunk. Miután a magasságon alapuló leválogatás megtörtént a „Z” koordináták értékeit eldobhatjuk. Így elő lehet állítani egy X,Y koordinátákból álló koordináta listát.

A zajsztűrt, 5 cm pont távolsággal újramintavételezett pontfelhőből korábban már ismertett *gawk* program segítségével kiválogattam azokat a pontokat, amelyek 1.5 és 2 méter közé estek ([1. melléklet](#)). Az így előállított 0.5 méteres sáv „Z” értékeit ezután elhagytam.

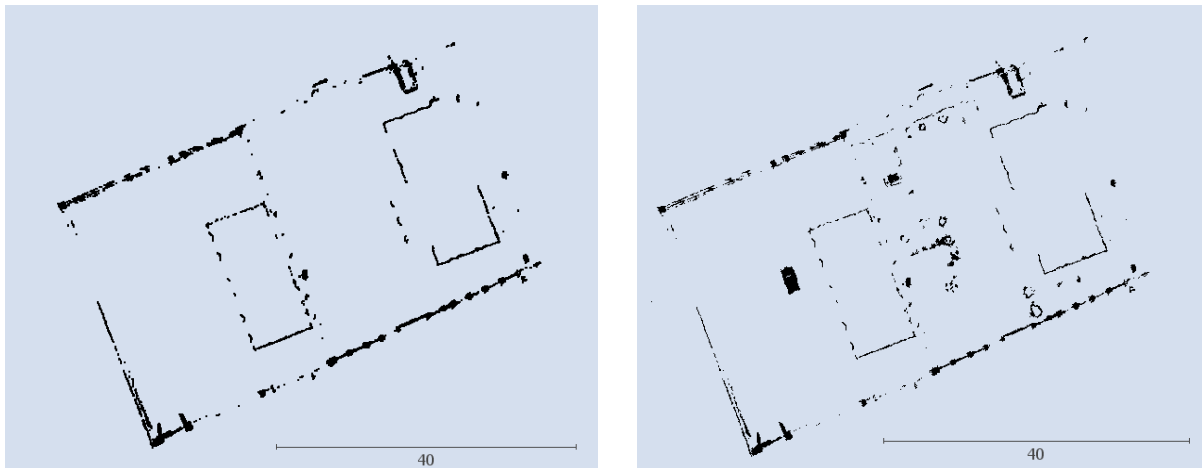


41. ábra: A sávból levezetett 2D koordináták

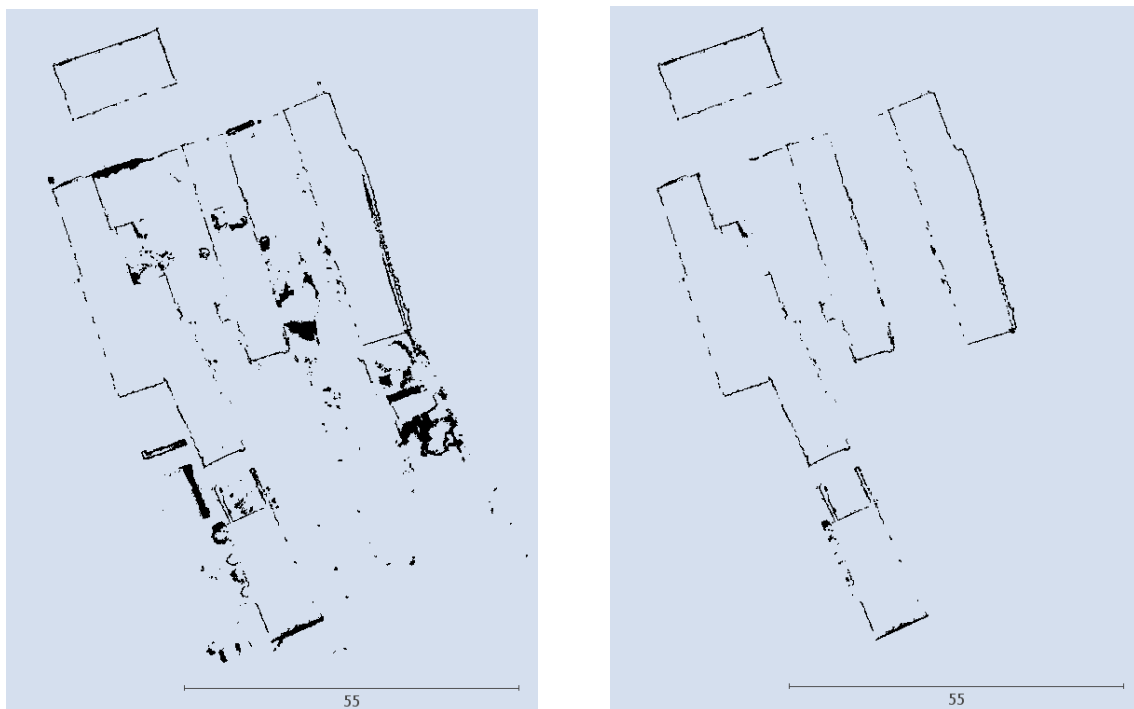
A fenti ábrán látható, hogy az épületek milyen mértékben rajzolódnak ki a metszetből levezetett pontokon. A falak, kerítések vonalát meghatározó pontokon kívül azonban egyéb pont csoportosulásokat is megfigyelhetünk, ami a leképződött növényzet miatt van. Ezeket a pontokat megpróbáltam eltávolítani a *CloudCompare*-ben található statisztikai alapon működő *SOR filter* segítségével. Számos beállítást megnéztem; azonban nem sikerült tökéletes szűrést

végeztem. Legtöbb esetben a problémát az jelentette, hogy a zaj nagyrészenek eltávolítása mellett a vektorizálás szempontjából fontos (épület) pontok is eltűntek.

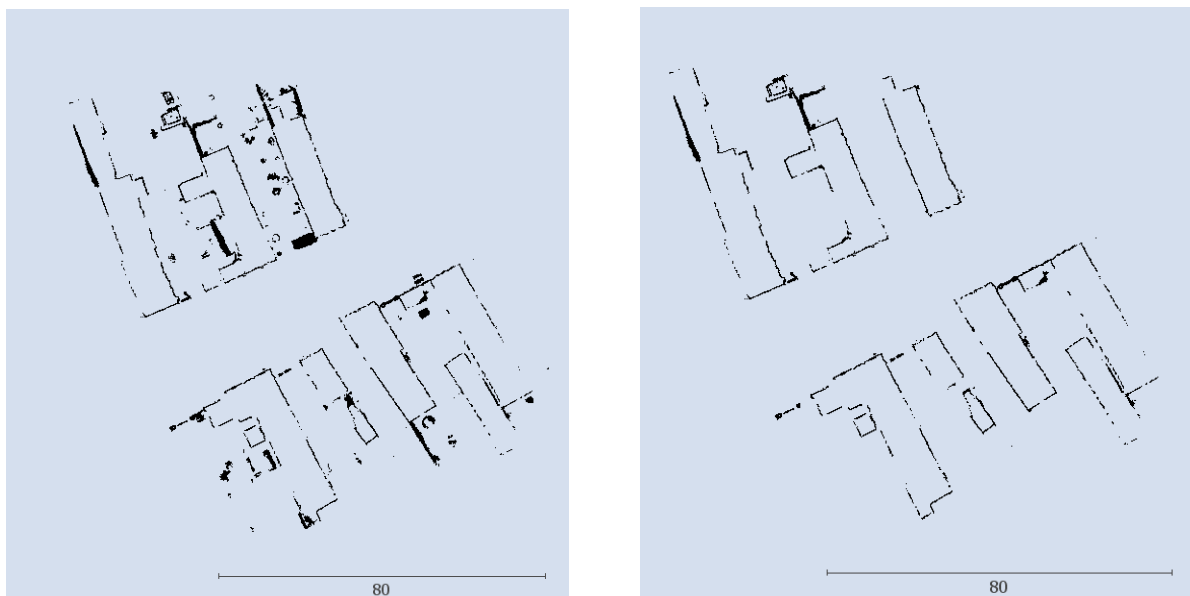
A legjobb a kiindulási alap egy osztályozott pontfelhő lenne, ahol a különböző építmények, vegetáció egymástól elválasztva szerepelnek. A [4. részben](#) már láthattuk, hogy a rendelkezésre álló keretek között az osztályozás ekkora méretű pontfelhőre a *LAStools* segítségével nehézkes lenne a szoftver különböző korlátozásai miatt. A továbbiakban ezért azt feltételezve, hogy az osztályozást sikeresen megvalósult, az építményeket tartalmazó pontfelhő normalizált változatából kiválasztottam 3 darab mintaterületet, ahol levezettem a 2D pontokból álló épületek falazatát tartalmazó koordináta-listát. Mivel ezek a területek a falu méretéhez képest jóval kisebbek voltak, ezért a zajszűrést könnyebben megtudtam oldani úgy, hogy a felesleges növényzetből leképződő pontokat manuálisan levágtam. A továbbiakban az automatizálás lehetőségének vizsgálatához a következő 3 darab teszt területet használtam fel.



42. ábra: I. mintaterület



43. ábra a II. mintaterület



44. ábra: A III. mintaterület

6.2. RANSAC – sorozatos egyenes illesztés

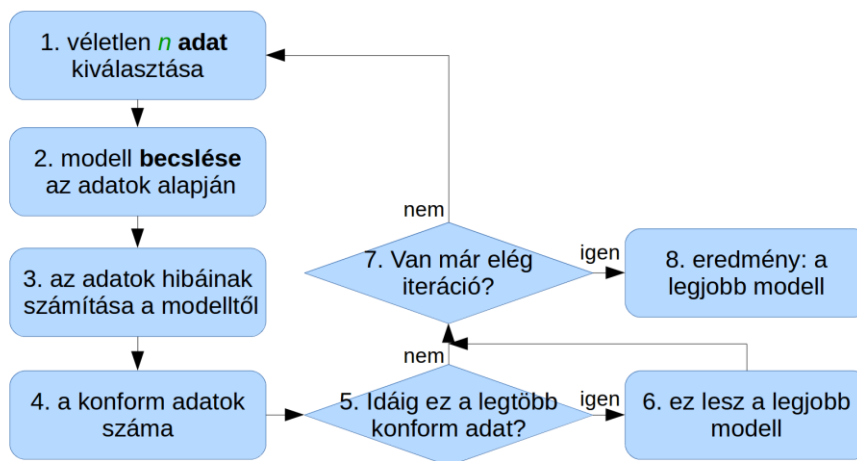
Az [előző fejezetben](#) bemutatott metszeteket látva felvetődhet a kérdés, hogyan lehetne az állományokat felhasználni? Szemmel láthatólag a megmaradt pontok többé-kevésbé kiadják a falazatok, kerítések vonalát. A legjobb az lenne, hogyha az egy egyenesre eső pontokat valahogyan ki tudnánk válogatni, a kigyűjtött pontokra külön-külön egyeneseket illesztենek, majd ezeket az egyeneseket valamilyen módon megjelenítenék egy térkép tervező szoftverben.

Ilyen problémák esetében lehet hatékony az iteratív robosztus becslési eljárás (RANSAC). A módszer lényege, hogy kezdeti modell paramétereit minimális mennyiségű adat esetén határozzuk meg először (most ezek legyenek az egyenesek paraméterei), majd következő lépésben megpróbáljuk bővíteni ezt az információt a többi adatunkra (jelen esetben a metszet pontjaira). A modellek közül (egyenes paraméterek) csak azokat tartjuk meg amelyekre a legtöbb konform adat illeszkedik. (Tóth 2019)

Ez az alapelve a Fischer és Bolles által 1981-ben megalkotott módszernek, a RANdom SAMple Consensus-nak, vagyis RANSAC-nak. Ez egy rendkívül népszerű, általánosan is alkalmazható robosztus becslési eljárás. Csak egy adott valószínűséggel ad helyes eredményt, tehát nem determinisztikus az algoritmus. A valószínűség értékét az iterációk számának emelésével lehet növelni. (Tóth 2019)

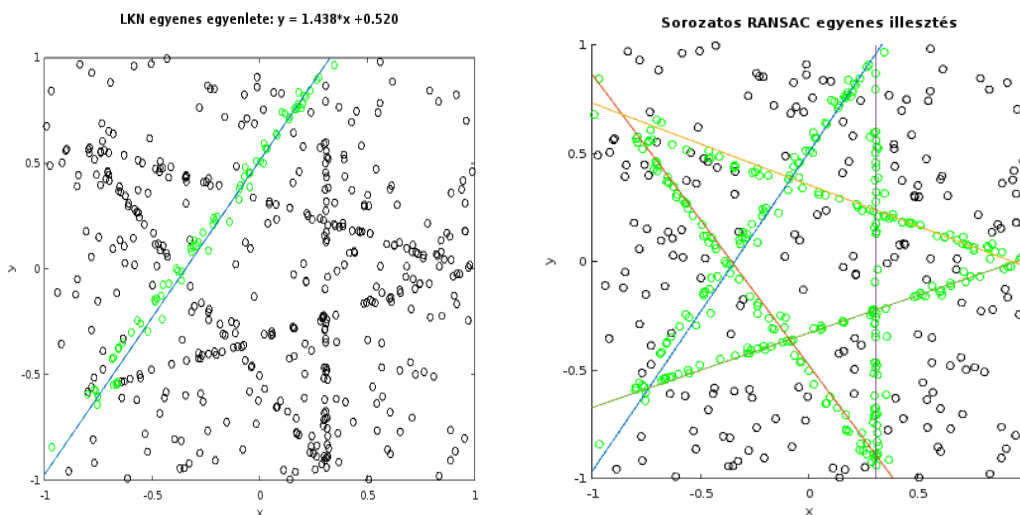
A módszer az elterjedését annak köszönheti, hogy könnyen érthető és hatékony. A durva hibákat alapvetően kevés iteráció segítségével is képes kiszűrni. A végeredmény minősége az iterációk számával van arányban, viszont amint egyezést talál az iteráció leáll. (Tóth 2019)

A RANSAC eljárás során a modellek megalkotásához paraméterek szükségesek. Meg kell adnunk a paraméterek minimális számát, a maximálisan megengedhető iterációk számát, egy küszöbértéket modellhez való illeszkedés eldöntéséhez, valamint egy adat számot, ami alapján el tudjuk dönteni, hogy az adott modell jól illeszkedik-e az adatokhoz vagy sem. A paramétereket ismerve az iteratív robosztus becslés folyamatábrája a következőképpen néz ki (Tóth 2019):



45. ábra: RANSAC folyamatábrája (Tóth 2019)

RANSAC további nagy előnye, hogy lehetőség van több modellre (például több egyenesre) vonatkozó illeszkedés vizsgálatra is. (Tóth 2019) Ilyenkor az alapelv az, hogy iterációt addig végezzük, ameddig az előre megadott számú modellt az adathalmazból meg nem találjuk. Ez azt jelenti, hogyha például egyeneseket keresünk a pontok között, akkor miután megtaláltuk az első legjobb egyenest és meghatároztuk a paramétereit, akkor ezeket a pontokat eldobhatjuk, és a megmaradó pontokon folytatjuk tovább az egyenes keresést a modell szám eléréséig.

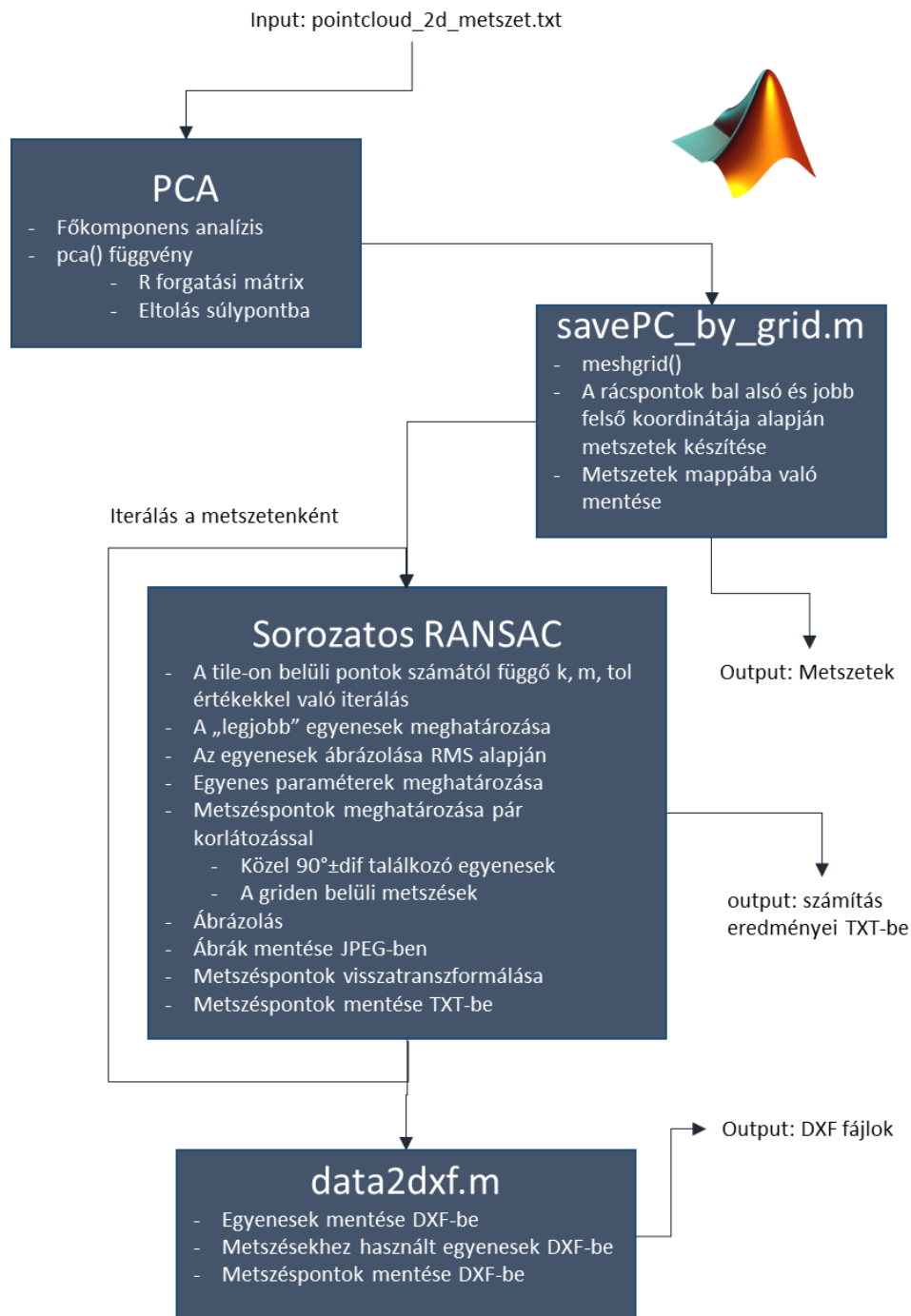


46. ábra: A sorozatos egyenes illesztés elve

Az automatizálási lehetőségek vizsgálata során ezt a módszert használtam fel a továbbiakban *Matlab* környezetben. A munkám során a Kiegyenlítő számítások MSc tantárgy egy mintapéldáját használtam fel és bővítettem ki az algoritmusom megírása során.

6.3. Sorozatos egyenes illesztésen alapuló program készítése Matlab-ban

A sorozatos egyenes illesztés módszerét felhasználva *Matlab*-ban írtam egy szkriptet ([2-melléklet](#)). A programom felépítését, a benne zajló lépéseket a következő folyamatra mutatja be:

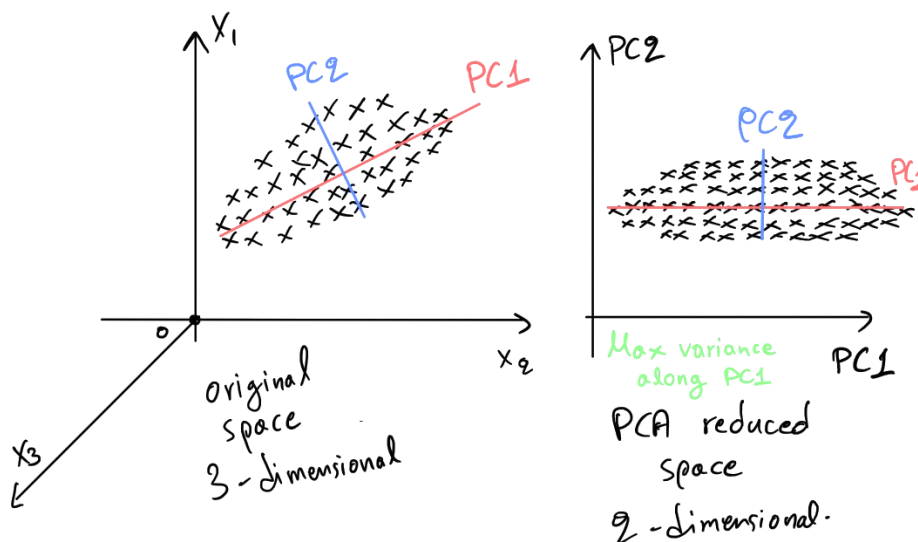


47. ábra: A seqRANSAC_on_pc.m Matlab kód működését bemutató folyamatra

A kiindulási alapja programomnak a 2D-re redukált, zajszűrt metszet. TXT formátumban tárolt koordinátaalista beolvasását követően egy főkomponens analízist (Principal Component Analysis, továbbiakban PCA) hajt végre.

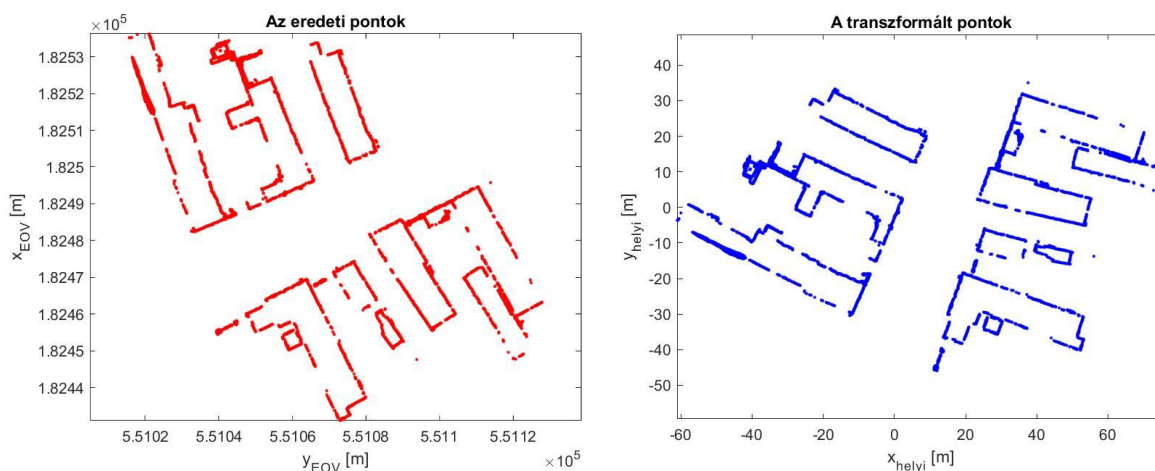
A módszer egy többváltozós statisztikai eljárás. Célja a dimenziók olyan halmazának megkeresése, amely jobban tükrözi az adatok variabilitását. Egyszerűsítve ez azt jelenti, hogy „az adatok által meghatározott pontfelhőt arra az egyenesre vetítjük le, ahol a kapott pontok szóródása a lehető legnagyobb lesz. Ezt követően erre az egyenesre merőleges irányok mentén határozzuk meg a további főkomponenseket.” (Tóth 2011) Ez gyakorlatilag az adatok transzformációját jelenti, amely során forgatási mátrix a kovariancia mátrix sajátértékeinek elemzése alapján tevődik össze. (Pang-Ning et al. 2011)

A főkomponens analízis közeli rokonságban áll a szinguláris érték felbontás (SVD) módszerével. A különbség csupán annyi, hogy PCA-nál eltávolítjuk az adatok átlagát is, vagyis az adatokat eltoljuk a halmaz súlypontjába. (Pang-Ning et al. 2011)



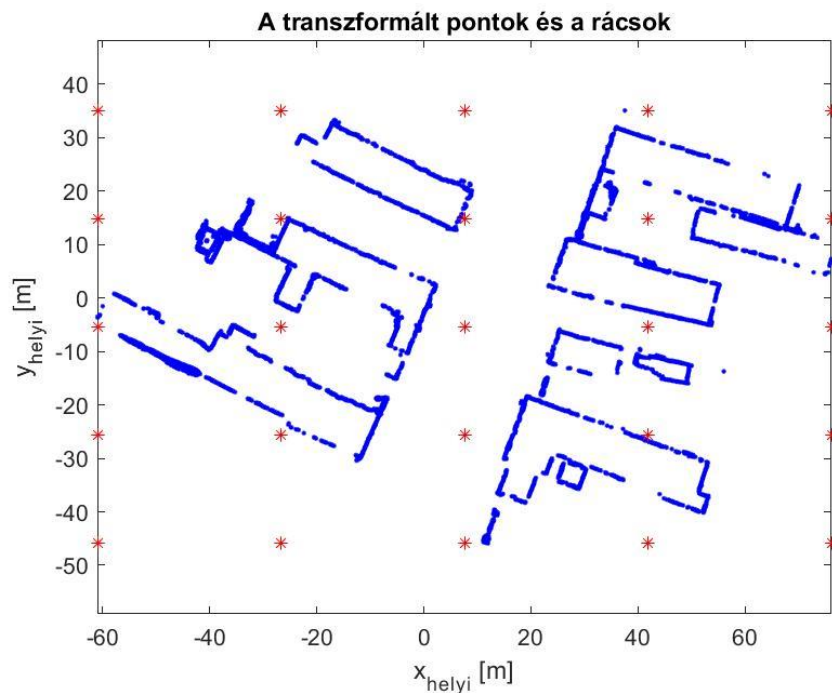
48. ábra: PCA elvi ábrája (Google)

Az általam felhasznált pontfelhőt EOV-ba transzformálták korábban, vagyis az egyes pontok koordinátái igen nagy számértékekből tevődnek össze. A numerikus számítás során ez hibát eredményezhet a különböző számítások elvégzése során. A PCA segítségével viszont a hatást csökkenteni tudjuk. A koordináták átlagával való eltolás jelentősen csökkenti a számítás során felhasznált számok értékét miközben a ponthalmazt a legideálisabb irányba forgatjuk el.



49. ábra: Az eredeti (bal) és elforgatott pontok (jobb)

Az elforgatásnak köszönhetően optimális helyzetben lévő pontokat a [3. mellékletben](#) található *savePC_by_grid.m* nevű függvénnyel szedtem részekre. A függvény a darabokat előre meghatározott rácspontok segítségével tudja csempékre szétbontani. A programomban megadható, hogy a teljes munkaterületet hány darab részre osszuk fel x és y értelemben (dx_db , dy_db). Azoknál a rácsoknál, ahová nem esne pont a felosztás során, ott script a metszetet nem készíti el. A helyi rendszerben lévő részeredményeket külön mappába mentem el.



50. ábra: A metszetek sarokpontjai (piros csillag)

A sorozatos egyenes illesztést az egyes csempékre külön-külön fut le. A módszerhez szükséges paramétereket a beolvasott pontok számának függvényében, határoztam meg. A programban lehetőség van a konszenzus és modellszámok értékének megadására oly módon, hogy a beolvasott pontszámokhoz viszonyított arányokat állíthatunk be a script elején található paraméterek (k_arany , m_arany) segítségével. Itt az illesztések vizsgálatához szükséges tolerancia (tol) értéket is megadhatjuk. A program gerincét adó sorozatos egyenes illesztés algoritmusának működését a következő ábra mutatja be részletesebben:

```

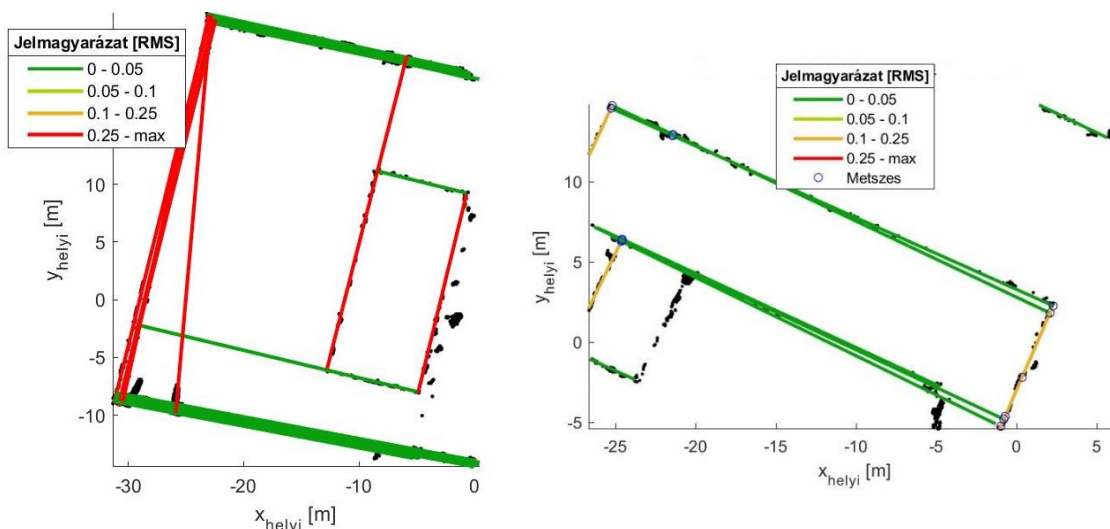
Adott:
adat - a mért adatpontok
model - az adatokra illesztendő modell
n - a modellek megalkotásához szükséges paraméterek minimális száma
k - A maximálisan megengedhető iterációk száma
t - küszöbérték annak eldöntéséhez, hogy egy adat illeszkedik-e a
modellhez
d - adatszám, amely alapján eldöntjük, hogy egy modell jól
illeszkedik-e az adatokhoz
Eredmény:
legjobb - legjobban illeszkedő modellhez tartozó paraméterek (semmi,
ha nincs jó modell)
iterációk = 0
legjobb = semmi
legjobb_hiba = valami nagyon nagy érték
amíg iterációk < k {
lehet_konform = n véletlenszerűen választott adat
lehet_modell = a lehet_konform-hoz illesztett modell paraméterek
szintén_konform = üres halmaz
tedd minden adatpontra amely nincs a lehet_konform-ban {
ha a pont illeszkedik a lehet_modell-re t-nél kisebb hibával
add a pontot a szintén_konform-hoz }
ha az elemek száma a szintén_konform-ban > d {
% ez jelzi, hogy egy jó modellt találtunk
%most ellenőrizzük, hogy mennyire jó
jobb_modell = az összes pontra (a lehet_konform és szintén_konform
pontokra)
végzett illesztés paraméterei
ez_a_hiba = annak mértéke, mennyire illeszkedik a modell a pontokra
ha ez_a_hiba < legjobb_hiba {
legjobb = jobb_modell
legjobb_hiba = ez_a_hiba }
}
következő iteráció
}
eredmény: legjobb

```

51. ábra: RANSAC algoritmus pszeudokódja (Dr. Tóth Gyula 2019)

A program által meghatározott egyenesek paramétereit (meredekség, eltolás, hány pontra illesztette az adott egyenest, RMS, a pontokra illeszkedő egyenes szélső pontjai) külön mappába csempésként TXT formátumban menti el. A megtalált egyeneseket ábrázolja a program RMS érték alapján színezve és a rajzokat a cikluson belül külön mappába menti.

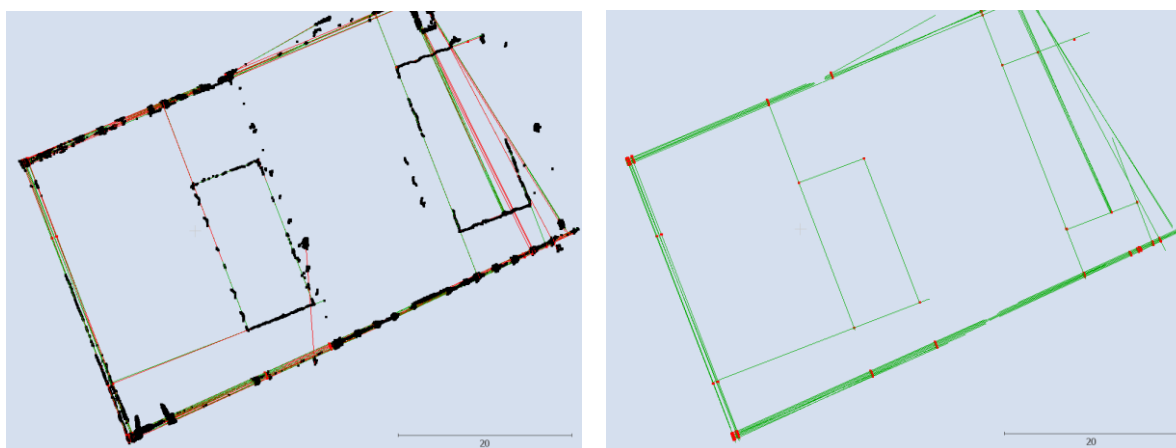
A „legjobb” egyenes paraméterek meghatározása után az egyenesek egyenletének felhasználásával a metszéspontokat is meghatározza a program. A metszések számítása során különböző korlátozásokat iktattam bele a scriptbe. Így azoknál az egyeneseknél, amiknél nem talál metszést, vagy a talált metszés kilógnak a vizsgálati tartományból (jelen esetben rács), vagy $90^\circ \pm \text{dif}^\circ$ eltérnek az egyenesek által bezárt szögek azoknak a metszéseit nem veszi figyelembe a feldolgozás során. A megtalált metszéspontok koordinátáit külön állományba menti el a program. A pontokat az ábrázolt egyeneseket tartalmazó ábrán külön feltünteti és szintén egy külön mappába menti.



52. ábra: A feldolgozás során mentett kép

Fontos megemlíteni, hogy az egyes lépésekben meghatározott pontok koordinátáit visszatranszformálom EOV-ba az egyes mentési lépések előtt.

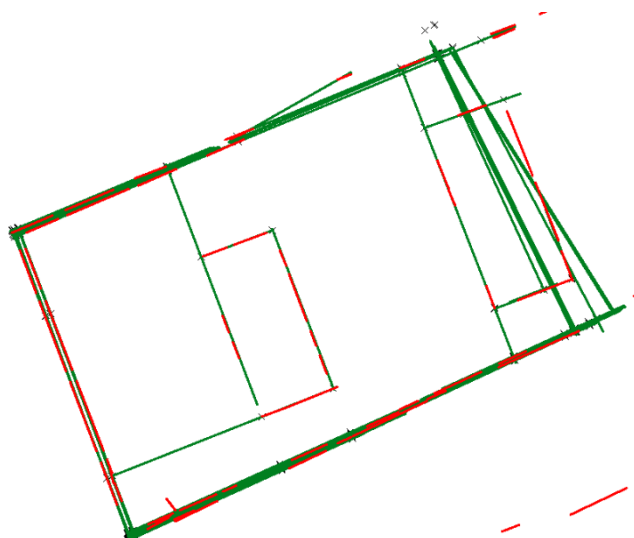
A megtalált egyenesek és pontok további felhasználásához egy másik *Matlab* scriptet is írtam. A *data2dxf.m* (4.melléklet) segítségével a *seqRANSAC_on_pc.m*-ben meghatározott struktúrában mentett adatokat tudjuk kiírni DXF formátumban. Így az egyes egyeneseket, pontokat akár *CloudCompare*-ben, vagy *AutoCAD*-ben is meg lehet tekinteni és további feldolgozásra is fel lehet használni.



53. ábra: Az exportált eredmények CC-ben

6.4. A feldolgozás során kapott végtermék összehasonlítása az AutoCAD Civil3D Section Lines parancsának eredményeivel

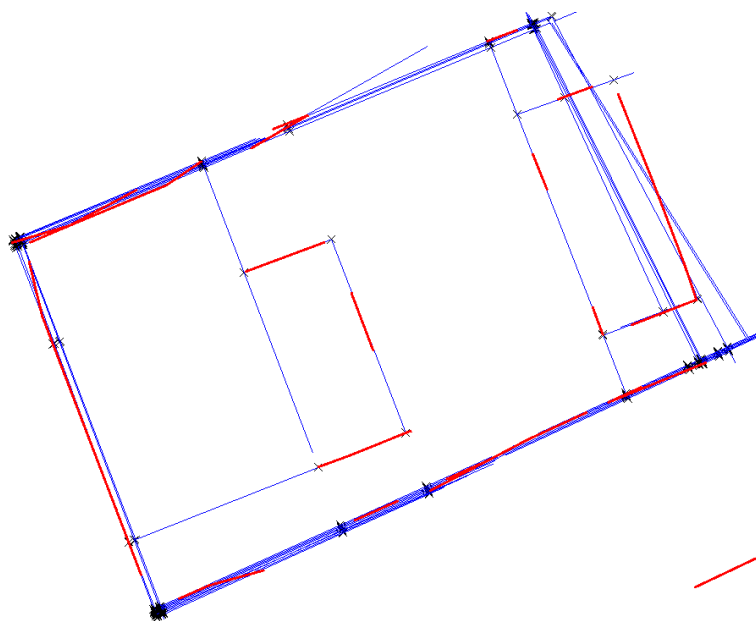
Az előállított termékek összehasonlítására a korábbi 5. részben bemutatott *Section Lines* *AutoCAD* parancs nyers és rajztisztított eredményeit vettem elő. A valós összehasonlíthatóság miatt az osztályozás hiányában megmaradt vonalakat (növényzet) töröltem. A két állomány egybe raktam.



54. ábra: A rajztisztítás előtt az AutoCAD Section Lines által létrehozott (piros) és az általam vezetett (sötétzöld) végtermék az 1. mintaterületen

Jól látható, hogy az *AutoCAD* parancs által kapott eredmény az I. mintaterületen ([5. melléklet](#)) nem adott olyan összefüggő egyeneseket, mint a *Matlab* programmal segítségével meghatározott végtermék. A szkriptből kapott eredmények a megjelenő metszéspontok, felhasznált egyenesek ábrázolásával sokkal több információt szolgáltathatnak. Fontos megemlíteni, hogy a *Matlab* segítségével alkotott eredménynél az egyes egyenesek illeszkedéséről külön információt is tárol (RMS), így amennyiben szükségünk van ezekre az adatokra a dokumentálás során felhasználhatjuk azokat is.

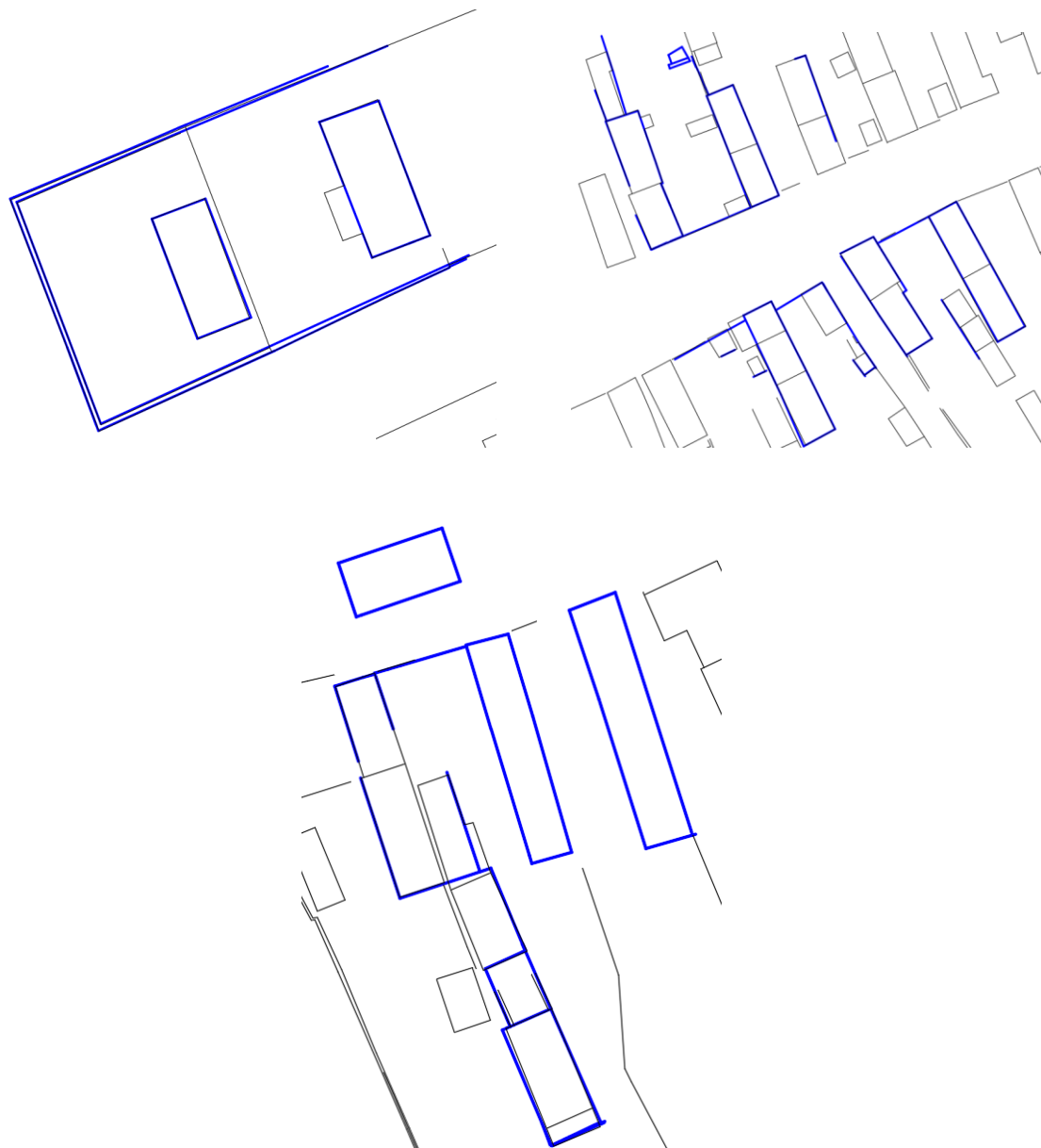
A következőkben megnéztem, hogy mi történik, ha a [4. részben](#) már bemutatott módon a rajztisztítási eszközöket is lefuttatjuk az létrehozott DXF fájlokon.



55. ábra: A rajztisztítás eredményei; a beépített paranccsal létrehozott (piros) és saját (kék) állományokon

Alapvetően látható, hogy a beépített parancs eredménye részletességet tekintve elmarad a *Matlab* script segítségével létrehozott terméktől. A rajztisztítás mindkettő esetben kiküszöbölte a szemmel gyakorlatilag észrevehetetlen túl és alul lógásokat. A megfelelő helyeken összekötötte a vonalakat.

Felhasználva a program segítségével létrehozott, majd rajztisztított DXF állományokat kicsit tovább mentem a vizsgálaton. A három területen a rajztisztítás után megmaradt egyeneseket, metszéspontokat felhasználva (pontfelhő nélkül) megpróbáltam az egyes épületeket reprodukálni a meglévő adatokból. A feldolgozás során feleslegesnek tűnő egyeneseket eldobtam. Ahol nem volt elegendő információ az azonosításhoz, ott az egyes vonalakat meghagytam, vagy nem kötöttem össze. A kapott eredményt (5. melléklet) egy már a pontfelhőből manuálisan kiértékelte állománnyal hasonlítottam össze.



56. ábra: A három munkaterületen a csak a Matlab-ból származó adatok kiértékelésének eredménye (kék) egy már digitalizált állománnyon (fekete)

A három mintaterületet bemutató ábrán jól látható, hogy bizonyos épületeket sikerült megrajzolnom. Azokon a területeken, ahol a beépítés sűrűbb, ott a program nem talált meg minden egyenest. A paraméterek állításával (kisebb rácsméreteket, modell szám, iterációt állításával) feltehetőleg lehetne javítani az eredményeken.

6.5. Az összehasonlítás eredményei számokkal kifejezve

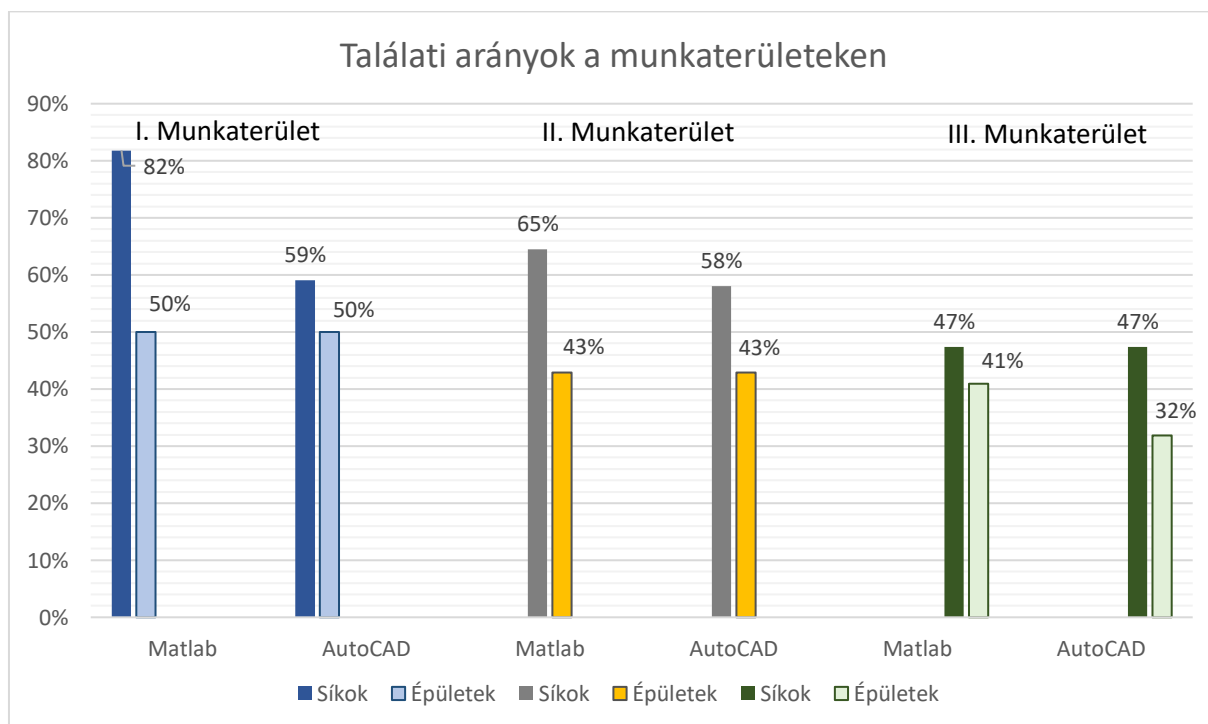
Az előző fejezetben láthattuk, hogy a két módszer eredményei grafikusán hogyan néznek ki. Most vizsgáljuk meg ezeket számszerűen.

A kézzel digitalizált állomány felhasználásával meghatároztam az egyes munkaterületekre eső síkok (épület és kerítés), illetve épületek számát. Ezt követően összeszámoltam a már rajztisztított *AutoCAD* és *Matlab* segítségével kiértékelt állományokon a megtalált síkokat. A szerkesztett eredmények alapján az épületek számát is összegyűjtöttem. Az egyes mennyiségeket a következő táblázat mutatja:

12. táblázat: Az *AutoCAD* és *Matlab* segítségével megtalált elemek

I. mintaterület			
Megtalált	Matlab	AutoCAD	Össz.
Síkok (db)	18	13	22
Épületek (db)	1	1	2
II. mintaterület			
Megtalált	Matlab	AutoCAD	Össz.
Síkok (db)	20	18	31
Épületek (db)	3	3	7
III. mintaterület			
Megtalált	Matlab	AutoCAD	Össz.
Síkok (db)	46	46	97
Épületek (db)	9	7	22

Az mintaterületeken azonosított elemek egymáshoz viszonyított arányát a következő ábra mutatja be:



57. ábra: A megtalált elemek

A *Matlab*-os módszer összességében jobb eredményt adott. Kisebb területen (I. mintaterület) sokkal hatékonyabban ismerte fel az algoritmus az egyes síkokat, mint az *AutoCAD*. Sajnos a nagyobb, bonyolultabb épületeket tartalmazó II. és III. mintaterületeknél ez az arány gyengébb, viszont összességében jobb, mint a beépített parancs esetében. Fontos megjegyezni, hogy a *Matlab* segítségével kapott eredményeknél az egyenesek illeszkedését leíró pontossági mérőszámok (RMS - [6.melléklet](#)) is rendelkezésre állnak ellentétben az *AutoCAD*-es paranccsal. A *Matlab*-os eredményre jellemző részletesség (metszéspontok, hosszabb egyenesek) nem kapott számszerű értéket. Az összehasonlítás szempontjából pedig ez is fontos információ. Az összes épületet egyik módszer segítségével sem sikerült tökéletesen detektálni. A bonyolultabb, több kisebb töréspontból álló épületek esetében csak a főbb síkok detektálását sikerült elérni a tesztelés során használt paraméterekkel.

Összességében megállapítható, hogy az utófeldolgozást egyik módszer segítségével sem lehet elkerülni. A különbség csupán a manuális feldolgozás támogatásának mértékében van. Alapvetően az *AutoCAD* funkció is használható, viszont számolnunk kell azzal, hogy a megfelelő beállítások megtalálása után is jóval több javítást kell elvégezni az állományon a *Matlab*-ban készített végtermékhez képest.

7. Összefoglalás

A dolgozatomban az ingatlan-nyilvántartási térkép pontfelhőből történő levezetésének különböző lehetőségeit vizsgáltam meg. A munkám során egy Barangon készült felmérés eredményét, egy pontfelhőt használtam fel.

A tanulmányomban a pontfelhőnek, mint kiindulási alapnak a különböző feldolgozási lehetőségeit tekintettem át. Az állomány előkészítését követően megvizsgáltam *LAStools*, és *CloudCompare* szoftver környezetekben a pontfelhőosztályozás lehetőségét. A pontfelhők ilyen típusú feldolgozása során ugyanis mindenképpen szükség van valamilyen, a pontfelhő tartalmára vonatkozó egyszerűsítésre. A *LAStools* beolvasott pontok mennyiségére vonatkozó korlátozása miatt végül a *CloudCompare* beépített plugin felhasználásával előállítottam egy a talaj és nem-talaj pontokat tartalmazó pontfelhőt. A talajpontokat felhasználva mesh-t is generáltam. A mesh és nem-talaj pontokból normalizált pontfelhőt vezettem le.

A normalizált pontfelhőből *AutoCAD Civil 3D*-ben és az általam írt sorozatos egyenes illesztésen alapuló *Matlab* program segítségével megvizsgáltam milyen mértékben lehet a térképek alapjául szolgáló vektoros állományok előállítását támogatni, esetleg kiváltani. Mindkettő esetben egyértelműen megállapítottam; hogy a teljes automatizálást nem lehet megoldani. A vizsgált két módszer eredményei közötti különbség az utófeldolgozásának mértékében van. Az általam írt *Matlab* script segítségével több információ (egyenesek, metszéspontok) segítségével lehet támogatni az utófeldolgozást úgy, hogy az automatikusan generált adatok mögött a minőségét leíró jellemzőkhöz is hozzáférésünk van az *AutoCAD*-es funkcióval ellentétben. Egy dolog belátható, hogy utólagos javításokra, ellenőrzésre mindenképpen szükség van.

Természetesen még számos egyéb, a dolgozatban be nem bemutatott automatizáláshoz kapcsolódó vizsgálati lehetőség van. Mindenképpen kellene különböző alakzatok, illetve geometriai elemek (pl. téglalapok, síkok) automatizált felismerésével is foglalkozni, akár pontfelhőből, akár valódi (true) ortofotókból kiindulva.

A rendelkezésre álló technológiák miatt ma már egyre nagyobb területek felmérést tudjuk gyorsan, nagy pontossággal, részleteséggel elvégezni. Így az ingatlan-nyilvántartási térképek minőségének ellenőrzéséhez, fenntartásához és esetleges felújításához szükséges méréseket a korábbiakkal ellentétben jóval hatékonyabban tudjuk elvégezni. Az ilyen típusú felmérések során keletkező nagy mennyiségű adatot viszont fel kell tudnunk dolgozni. Szükség van új feldolgozási módszerek fejlesztésére, tesztelésére és alkalmazására, hogy a jövőben ezt minél hatékonyabban tehesük meg.

8. Irodalomjegyzék

A. Bellakaout - M. Cherkaoui - M. Ettarid - A. Touzani 2016. Automatic 3D Extraction of Buildings, Vegetation and Roads from LIDAR Data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B3, 2016 XXIII ISPRS Congress, 12–19 July 2016, Prague, Czech Republic*

<https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLI-B3/173/2016/isprs-archives-XLI-B3-173-2016.pdf>

Utolsó elérés: 2020. október. 8.

Arnold Robbins 2015. *Effective awk Programming: Universal Text Processing and Pattern Matching.*

<https://books.google.hu/books?id=rIPoBgAAQBAJ&pg=PT758&dq=gawk&hl=hu&sa=X&ved=2ahUKewj8zN-X98fsAhUml4sKHYmjCJwO6AEwBXoECAyQA#v=onepage&q=gawk&f=true>

Utolsó elérés: 2020. október 23.

Boda Géza 2008. *A digitalizált térképek tűréshatárai. Geodézia és Kartográfia 60. évf. 3. sz./2008*

<https://docplayer.hu/7733037-A-digitalizalt-terkepek-tureshatarai.html>

Utolsó elérés: 2020. október 18.

Fischler M.A, Bolles R.C 1981. *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381-395.*

Fouhey, D.F 2011. *Multi-model estimation in the presence of outliers. PhD Thesis, Middlebury College*

G. B. He- L. L. Li 2020. Research and application of LiDAR technology in cadastral surveying and mapping. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLIII-B1-2020, 2020 XXIV ISPRS Congress (2020 edition)*

<https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLIII-B1-2020/33/2020/isprs-archives-XLIII-B1-2020-33-2020.pdf>

Utolsó elérés: 2020. október. 8.

J.M. Macay Moreira - F. Nex - G. Agugiaro - F. Remondino - N.J. Lim 2013. From DSM to 3D building models: a quantitative evaluation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-1/W1, ISPRS Hannover Workshop 2013, 21 – 24 May 2013, Hannover, Germany*

<https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-1-W1/213/2013/isprsarchives-XL-1-W1-213-2013.pdf>

Utolsó elérés: 2020. október. 8.

Laky Piroska 2019. *Numerikus módszerek építőmérnököknek Matlab-bal. Numerikus módszerek MSc tágy. 2019/20 1. félév*

Lehoczky Máté - Siki Zoltán 2020. Fotogrammetriai feldolgozószoftverek. *Geodézia és Kartográfia 72:2 pp. 23-27.*

M. Brédif - O. Tournaire - B. Vallet - N. Champion 2017. *Extracting polygonal building footprints from digital surface models: A fully automatic global optimization framework. ISPRS Journal of Photogrammetry and Remote Sensing, Elsevier, 2013, 77, pp.57-65.*

- M. Gutiérrez Viñuales - E. Puch Sleive - S. Chalabe 2019.** Experiences in cadastral restitution at a town affected by a natural disaster. The case of volcan, Jujuy Province, Argentina. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-2/W13, 2019 ISPRS Geospatial Week 2019, 10–14 June 2019, Enschede, The Netherlands*
<https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W13/343/2019/isprs-archives-XLII-2-W13-343-2019.pdf>
Utolsó elérés: 2020. október. 8.
- M. Kedzierski - A. Fryskowska - D. Wierzbicki - P. Nerc 2016.** Chosen aspects of the production of the basic map using UAV imagery. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B1, 2016 XXIII ISPRS Congress, 12–19 July 2016, Prague, Czech Republic*
<https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLI-B1/873/2016/isprs-archives-XLI-B1-873-2016.pdf>
Utolsó elérés: 2020. október. 8.
- M. Rijdsdijk - W.H.M. van Hinsbergh - W. Witteveen - G.H.M. ten Buuren - G.A. Schakelaar - G. Poppinga - M. van Persie - R. Ladiges 2013.** Unmanned arial systems in the process of juridical verification of cadastral border. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-1/W2, 2013 UAV-g2013, 4 – 6 September 2013, Rostock, Germany*
<https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-1-W2/325/2013/isprsarchives-XL-1-W2-325-2013.pdf>
Utolsó elérés: 2020. október. 8.
- Neuberger Hajnalka 2020.** *Lastools használata QGIS 2.12.1 környezetben*
http://www.agt.bme.hu/gis/qgis/Lastools_in_QGIS_2.12.1.pdf
Utolsó elérés: 2020. október. 23.
- Pang-Ning Tan, Michael Steinbach, Vipin Kumar 2011.** *Bevezetés az adatbányászatba.*
https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0046_adatbanyaszat/apb.html#id775268
Utolsó elérés: 2020. október. 24.
- Siki Zoltán 2016.** *Rajztisztítás az AutoCAD Mappel.*
<https://edu.epito.bme.hu/local/coursepublicity/mod/resource/view.php?id=10476>
Utolsó elérés: 2020. október
- Siki Zoltán 2004.** *AutoCAD MAP képernyő digitalizálás segédlet.*
<https://edu.epito.bme.hu/local/coursepublicity/mod/resource/view.php?id=10475>
Utolsó elérés: 2020. október. 23.
- Tóth Gyula 2019.** *Kiegyenlítő számítások MSc. 12. előadás anyaga. 2019/20 1. félév.*
- Tóthné Parázsó Levente 2011.** *A kutatómódszertan matematikai alapjai.*
https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0005_31_kutato_modszer_tan_scorm_07/737_fkomponensanalizis.html
Utolsó elérés: 2020. október 24.
- Varga József 2018.** *Vetülettan előadás jegyzet*
- Z. Kurczynski - K. Bakula - M. Karabin - M. Kowalczyka - J. S. Markiewicz - W. Ostrowski - P. Podlasiak - D. Zawieska 2016.** The possibility of using images obtained from the UAS in cadastral works. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B1, 2016 XXIII ISPRS Congress, 12–19 July 2016, Prague, Czech Republic*
<https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLI-B1/909/2016/isprs-archives-XLI-B1-909-2016.pdf>
Utolsó elérés: 2020. október. 8.
- Zuliani M. (2014).** *RANSAC for Dummies.vision.ece.ucsb.edu/~zuliani, 101, GNU FDL*

Felhasznált jogszabályok:

207/1962. (T.6.) ÁFTH utasítás a földrendezéssel nem érintett területek földnyilvántartási adatainak rendezéséről

209/1962. (T.9.) ÁFTH utasítás a belterületek és zártkertek kataszteri térképeinek felújítására

8/2018 (VI.29.) AM rendelet az ingatlan-nyilvántartási célú földmérési és térképészeti tevékenység részletes szabályairól.

<https://net.jogtar.hu/jogszabaly?docid=A1800008.AM>.

Utolsó elérés: 2020. október 18.

Egyéb weboldalakhoz tartozó hivatkozások:

Barnag 2020. A település weboldala. <https://www.barnag.hu/> Utolsó elérés: 2020. október 23.

FAP 2020. MMK-GGT pályázata a „Módszertani útmutató az elavult ingatlan-nyilvántartási térképek korszerű technológiákkal végzett felújításához”

<http://mmk-ggt.hu/hirlevel/2020/20200507.pdf>

Utolsó elérés: 2020. október 24.

KSH 2020. Központi Statisztikai Hivatal - Helységnévtár.

http://www.ksh.hu/apps/hntr.telepules?p_lang=HU&p_id=15778 Utolsó elérés: 2020. október 23.

LAStools 2020. A program elérése. https://github.com/LAStools/LAStools/blob/master/bin/lasview_README.txt.

Utolsó elérés: 2020. október 23.

Takarnet 2020. Település kereső. http://www.takarnet.hu/pls/tknet/hivatalok_p.hivatallista.

Utolsó elérés: 2020. október 18.

9. Mellékletek

1. melléklet: *pc_z_section.awk*
2. melléklet: *seqRANSAC_on_pc.m*
3. melléklet: *savePC_by_grid.m*
4. melléklet: *data2dxf.m*
5. melléklet: *Elkészített rajzok (PDF)*
6. melléklet: *Az elkészített munkarészeket és kódokat tartalmazó Google Drive állomány:*

https://drive.google.com/drive/folders/1Ece_LTBbrIos1ezMO2JiNk7re_ktHIjS?usp=sharing

A mappa tartalama:

- *Matlab scriptek*
- *Elkészített rajzok az egyes munkarészekről, területekről (PDF)*
- *Az elkészített rajzok DWG formátumban*
- *Pontfelhő (RCP)*

1. melléklet:

pc_z_section.awk

```
# A program a megadott pontfelho 3. oszlopa (z) es a megadott min es max
#ertekek szerint levalogatja a pontokat
BEGIN {FS = "[,]";
        min = 1.5;          # legkisebb Z ertek
        max = 2;           # legnagyobb Z ertek
    }
    { if(($3 >= min) && ($3 <= max))
      { print $1,$2,$3}
    }
```

2. melléklet:

seqRANSAC_on_pc.m

```
% Ingatlan-nyilvántartási térkép felújítás pontfelhő alapján
% TDK 2020.
% Készítette: Hrutka Bence Péter
% Neptun: FDGJW8
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% A script segítségével pontfelhőből levezetett metszeten határozhatunk meg
% egyeneseket sorozatos egyenes illesztés segítségével.
clear all; close all; clc; format longG;
% inputs: - folder: a mappa, ahova menteni szeretnénk az eredményeket
%          folder = 'Barnag_Mintaterulet/Barnag_Mintater_1';
% - filename: a prefixum a mentett adatok előtt
%          filename = 'Barnag_mintater';
% - pc_metszet: a normalizált pontfelhő metszet betöltése
%          pc_metszet = 'Barnag_mintater.txt';
% - dx_db: megadhatjuk, hogy mennyi részre ossza x irányba az adatokat
%          dx_db = 2 []
% - dy_db: megadhatjuk, hogy mennyi részre ossza y irányba az adatokat
%          dy_db = 1 []
% - k_arany megváltoztatjuk az iterációk számát (k = a beolvasott sorok száma /
k_arany
%          k_arany = 2 []
% - m_arany a modellszám - hány egyenest keressen?
%          m_arany = 100 Megjegyzés: ha kevés pont van az adatoknál ,akkor m = 3
% -tol a tolerancia értéke az egyenes illesztésnél
%          tol = 0.05 [m]
% - dif a metszések keresésénél csak azokat a metszeteket keresi, amik 90 +-
dif körül találkoznak
%          dif = 5 [fok]
% - tul megadhatjuk, hogy az egyes egyenesek szélső pontjaihoz képest
mennyivel lóghatnak túl a metszések számításánál
%          tul = 0.15 [m]
% outputs - mappák:
% - folder + '_egy': megtalált egyenesek ábrája
% - folder + '_egy_par': egyenesek paraméterei (felosztásonként)
% [m,b,db,RMS,x1_helyi,y1_helyi,x2_helyi,y3_helyi, x1_eov,y1_eov,x2_eov,y3_eov]
% - folder + '_metszesek': talált pontok; legjobb egyenesek kezdő és végpontokkal
megadva
% - folder + '_racso': felosztás
% - folder + '_teljes_kep': a feldolgzás elején található lépések mentése
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INPUTS:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tesztelés input adatok:
folder = 'Barnag_Mintaterulet/Barnag_Mintater_1';
filename = 'Barnag_mintater';
pc_metszet = 'Barnag_mintater.txt';
dx_db = 2; % x irányban mennyi felosztás legyen a GRID makerben
dy_db = 2; % y irányban mennyi felosztás legyen a GRID makerben
k_arany = 2; % Konszenzus halmaz száma
m_arany = 100; % Modell szám
tol = 0.05; % [m] Mekkora tolarenciát engedjünk az illesztés során
dif = 5; % [°] - merőlegeshez képest hány fokkal térhet el a metszések vizsgálatánál a program
tul = 0.25; % [m] mekkora mértékben nyúlhat túl az illesztett pontok halmazához képest egy-
egy metszés megkeresésénél

%% 1 mintaterület input adatok:
folder = 'Barnag_mintaterulet_1/Barnag_mintater_1';
```

```

filename = 'Barnag_mintaterulet_1';
pc_metszet = 'Barnag_mintaterulet_1.txt';
dx_db = 2; % x irányban mennyi felosztás legyen a GRID makerben
dy_db = 1; % y irányban mennyi felosztás legyen a GRID makerben
k_arany = 2; % Konszenzus halmaz száma 2
m_arany = 75; % Modell szám 100
dif = 5; % [°] - merőlegeshez képest hány fokkal térhet el a metszések vizsgálatánál a program
tul = 0.25; % [m] mekkora mértékben nyúlhat túl az illesztett pontok halmazához képest egy-egy metszés megkeresésénél

% % % 3mintaterület input adatok:
% % folder = 'Barnag_mintaterulet_2/Barnag_mintater_2';
% % filename = 'Barnag_mintaterulet_2';
% % pc_metszet = 'Barnag_mintaterulet_2.txt';
% % dx_db = 3; % x irányban mennyi felosztás legyen a GRID makerben
% % dy_db = 2; % y irányban mennyi felosztás legyen a GRID makerben
% % k_arany = 2; % Konszenzus halmaz száma 2
% % m_arany = 75; % Modell szám 100
% % dif = 5; % [°] - merőlegeshez képest hány fokkal térhet el a metszések vizsgálatánál a
program
% % tul = 0.25; % [m] mekkora mértékben nyúlhat túl az illesztett pontok halmazához képest
egy-egy metszés megkeresésénél
% %
% %
% % % 3mintaterület input adatok:
% % folder = 'Barnag_mintaterulet_3/Barnag_mintater_3';
% % filename = 'Barnag_mintaterulet_3';
% % pc_metszet = 'Barnag_mintaterulet_3.txt';
% % dx_db = 4; % x irányban mennyi felosztás legyen a GRID makerben
% % dy_db = 4; % y irányban mennyi felosztás legyen a GRID makerben
% % k_arany = 2; % Konszenzus halmaz száma 2
% % m_arany = 75; % Modell szám 100
% % dif = 5; % [°] - merőlegeshez képest hány fokkal térhet el a metszések vizsgálatánál a
program
% % tul = 0.25; % [m] mekkora mértékben nyúlhat túl az illesztett pontok halmazához képest
egy-egy metszés megkeresésénél
% %

% Ábrázolás 3D-ben ellenőrzés
%pcshow(pc_metszet)

%% Principal Component Analysis - ponokra a legoptimálisabb tengely meghatározása
% A PCA elmélete
% https://hu.wikipedia.org/wiki/Főkomponens-analízis
% https://www.youtube.com/watch?v=oBhm2V\_cGZk
% https://www.youtube.com/watch?v=HMOI\_lkzW08
% https://uk.mathworks.com/help/stats/pca.html
pc_metszet = load(pc_metszet);
% PCA
[coeff,score] = pca([pc_metszet(:,1), pc_metszet(:,2)]);
coeff; % Koeficiens mátrix -> FORGATÁS
score; % Eltolt és forgatott ponthalmaz
% Megjelenítés - ellenőrzéshez
figure(1)
plot(pc_metszet(:,1), pc_metszet(:,2), 'r. ');
axis equal
title('Az eredeti pontok')
ylabel('x_E_O_V [m]')
xlabel('y_E_O_V [m]')
% Kép mentése
folder0 = [folder, '_teljes_kep'];
if ~exist(folder0, 'dir')
    mkdir(folder0)
end
picturename = [folder0, '/', filename, '_eredeti_pontok'];
print(picturename, '-djpeg');
% Elforogatott pontfelhő megjelenítése
figure(2);
plot(score(:,1),score(:,2), 'b. ');
title('A transzformált pontok')
axis equal
ylabel('y_h_e_l_y_i [m]')
xlabel('x_h_e_l_y_i [m]')
% Kép mentése
picturename = [folder0, '/', filename, '_trafo_pontok'];
print(picturename, '-djpeg');

```

```

% Transzformálás
% Átlag számítás - eltolás
x_mean = mean(pc_metszet(:,1));
y_mean = mean(pc_metszet(:,2));
% Forgatási mátrix
R = coeff;
eltolt_xy = [pc_metszet(:,1)- x_mean, pc_metszet(:,2)- y_mean];
% Visszatranszformálás az 'R' mátrix és a súlypont ismeretében
ellenorzes = eltolt_xy * R;
% Ellenőrzéshez megjejenítés
% % % figure(3)
% % % plot(ellenorzes(:,1),ellenorzes(:,2), 'b. ');
% % % hold off;
% Visszaszámítás - ellenőrzés
ell_x = rms(score(:,1) - ellenorzes(:,1));%           ELLENŐRÍZHETŐ a trafo
ell_y = rms(score(:,2) - ellenorzes(:,2));

%% Feldarabolás
% score = load('Barnag_OffGroundPoints_1_5__2.txt');
% Bal alsó sarokpont meghatározása
x_min = min(score(:,1));
y_min = min(score(:,2));
B = [x_min,y_min];
% Jobb alsó sarokpont meghatározása
x_max = max(score(:,1));
y_max = max(score(:,2));
C = [x_max, y_max];
x_dist = x_max - x_min; % 66.4584734512565 m
y_dist = y_max - y_min; % 41.5881415461416 m
arany = x_dist/y_dist; % 1.59801498649613

%% GRID létrehozása
% Rácsmérés megadása x_dist és y_max alapján érdemes beállítani
% Default érték           PARAMÉTEREK BEÁLLÍTÁSA - RÁCSMÉRÉS!!!
% dx_db = 2;
% dy_db = 1;
dx = x_dist / dx_db      % 46.1389134715184 m      20
dy = y_dist / dy_db      % 53.8322701001485 m      12
% Rács bal alsó és jobb felső pontjainak koordinátái:
[x_grid y_grid] = meshgrid(x_min:dx:x_max, y_min:dy:y_max);
% Ellenőrzéshez ábra
figure(4)
plot(score(:,1),score(:,2), 'b. ')
hold on;
plot(x_grid,y_grid,'r*')
title('A transzformált pontok és a rácsok')
ylabel('y_h_e_l_y_i [m]')
xlabel('x_h_e_l_y_i [m]')
axis equal
hold off;
picturename = [folder0, '/', filename, '_racs_pontok'];
print(picturename, '-djpeg');

%% Pontfelhő darabolása
P_t = score;
%folder = 'Barnag_Darabolt_PC'
%filename = 'Barnag_tile'
[grid_min, grid_max, n] = savePC_by_grid(x_grid, y_grid, P_t, [folder, '_racsok'],
[filename, '_tile']); % Számított rács pont alaján működik

%           - folder\filename+###: a feldarabolt pontfelhő
%                                           --> Barnag_tile_###.txt
% Visszajelzés - hány db rácsot hozott létre
n;

%% Kigyűjti a txt fájl elérését egy változóba
dirList = dir(fullfile([folder, '_racsok'], '*.txt'));
fnms = fullfile([folder, '_racsok'], {dirList.name});
db = length(fnms); % a GRID fájlok miatt
% Kiinduló paraméterek
intsec = [];
intsec_jo2 = [];
best_egy2 = [];
%intsec_bad = [];
for fnumb = 1:db % fnumb = 2 % %
    % Teszthez betöltés
    d = load([folder, '_racsok', '/', filename, '_', 'tile_', num2str(fnumb), '.txt']);

```

```

% Bal alsó sarokpont meghatározása
grid_x_min = min(d(:,1));
grid_y_min = min(d(:,2));
grid_min_fnumb = [grid_x_min, grid_y_min];
% Jobb alsó sarokpont meghatározása
grid_x_max = max(d(:,1));
grid_y_max = max(d(:,2));
grid_max_fnumb = [grid_x_max, grid_y_max];
x = d(:,1);
y = d(:,2);
tol = 0.05; % Hiba küszöb PARAMÉTEREK BEÁLLÍTÁSA
% k = 100; % iterációk száma 100 - beállítás
% Default k = 2
k = floor(length(x)/k_arany)
% Modellszám a fájlban lévő pontok alapján
if k < 300 % modellek száma ha kevés pont van --> mi számít kevés pontnak?
    m = 3;
else
    %m = 10;
    % Default m = 100
    m = floor(k/m_arany) %/100 --> 24 modellt keres
    % m = 40;
end
% Modellszám a fájlban lévő pontok alapján?? EGY FÁJL ESETÉN ÉRDEMES ÁLLÍTGATNI
% A m = 20 a falu közepén jó
% Adatok az inicializáláshoz
egyenes_par = [];
J_sarok = [];
B_sarok = [];
for n = 1:m
    numb_data = length(x); % Adatsor hossza
    n_max = 0; % Nincs még konszenzus halmaz
    for i = 1:k
        % Két pont véletlenszerű kiválasztása
        is = randperm(numb_data,2);
        % ax + by - 1 = 0 egyenes paramétereinek meghatározása
        A = d(is,:);
        b = [1;1];
        % EGYENES PARAMÉTEREI a KÉT pontra
        par = A\b;
        % Adatok egyenestől mért távolságai
        t = abs(par(1)*x+par(2)*y-1)/sqrt(par(1)^2+par(2)^2);
        x_k = x(t<tol);
        y_k = y(t<tol); % Illeszkedő adatok
        numb_in = length(x_k); % Konszenzus halmaz elemszáma
        if numb_in > n_max; % Jobb mint idáig -> frissít
            x_in = x_k;
            y_in = y_k;
            n_max = numb_in;
            x_out = x(t>=tol);
            y_out = y(t>=tol); % Nem konszenzus halmaz elemeit tartalmazza
        end
    end
    % LKN egyenes illesztés
    best_par = polyfit(x_in,y_in,1);
    % 'Legjobb' egyenes paramétereinek kiszámítása
    line = @(x,y) best_par(1)*x + best_par(2)*y - 1;
    % Eltérések számítása
    t_b = best_par(1).*x_in + best_par(2) - y_in; % VAN - KELL
    best_rms = rms(t_b); % Az illesztés jellemzése - RMS
    % A meredekség előjelének vizsgálata ábrázoláshoz
    if best_par(1) <= 0; % + meredekség
        B_sarok(1,1) = min(x_in);
        B_sarok(1,2) = best_par(1)*B_sarok(1,1) + best_par(2); % Szélső pont koord.
    meghatározása
        J_sarok(1,1) = max(x_in);
        J_sarok(1,2) = best_par(1)*J_sarok(1,1) + best_par(2);
    else % - meredekség
        B_sarok(1,1) = min(x_in);
        B_sarok(1,2) = best_par(1)*B_sarok(1,1) + best_par(2); % Szélső pont koord.
    meghatározása
        J_sarok(1,1) = max(x_in);
        J_sarok(1,2) = best_par(1)*J_sarok(1,1) + best_par(2);
    end
    % A konszenzus halmaz elemeit eltávolítjuk -> ebből megyünk tovább
    x = x_out;
    y = y_out;
end

```

```

d = [x,y];
% Egyenes kezdő végpontjának visszaforgatása
B_sarok_traf = B_sarok*inv(R);
B_sarok_traf = [B_sarok_traf(1) + x_mean, B_sarok_traf(2) + y_mean];
J_sarok_traf = J_sarok*inv(R);
J_sarok_traf = [J_sarok_traf(1) + x_mean, J_sarok_traf(2) + y_mean];
% Egyenes paraméterek kigyűjtése változóba
egyeses_par2(n,1) = best_par(1); % m
egyeses_par2(n,2) = best_par(2); % b
egyeses_par2(n,3) = length(x_in); % db
egyeses_par2(n,4) = best_rms; % rms
egyeses_par2(n,5) = B_sarok(1,1); % bal sarok x helyi koordinátája
egyeses_par2(n,6) = B_sarok(1,2); % bal sarok y helyi koordinátája
egyeses_par2(n,7) = J_sarok(1,1); % jobb sarok x helyi koordinátája
egyeses_par2(n,8) = J_sarok(1,2); % jobb sarok y helyi koordinátája
egyeses_par2(n,9) = B_sarok_traf(1,1); % bal sarok y_eov koordinátája
egyeses_par2(n,10) = B_sarok_traf(1,2); % bal sarok x_eov koordinátája
egyeses_par2(n,11) = J_sarok_traf(1,1); % jobb sarok y_eov koordinátája
egyeses_par2(n,12) = J_sarok_traf(1,2); % jobb sarok x_eov koordinátája
end
% Fájlba mentés - [m,b,db,RMS,x1,y1,x2,y3]
folder1 = [folder, 'egy_par'];
if ~exist(folder1, 'dir');
    mkdir(folder1)
end
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Egyenesek szűrése RMS alapján % Állítható
paraméter
% row = find(abs(egyeses_par2(:,4)) <= 0.5); % tol = 0.3
% egyeses_par = egyeses_par2(row,:);
egyeses_par = egyeses_par2;
% Fájl kiírás
dlmwrite([folder1, '/', filename, '_par_', num2str(fnumb), '.txt'], egyeses_par, ...
    'delimiter', ';', 'precision', '%6.3f');
% Megjelenítés
figure(fnumb+10); hold on;
plot(x,y, "k.");
figname = ['Tile:', num2str(fnumb)];
% Színezés RMS alapján % egyénileg PARAMÉTEREZHETŐ
színezés
rms_class1 = 0;
rms_class2 = 0.05;
rms_class3 = 0.10;
rms_class4 = 0.25;
% Érték alapján színezés beállítása
color1 = '#09A10F'; disp_name1 = [num2str(rms_class1), ' - ', num2str(rms_class2)]; % dark
green
color2 = '#ACD200'; disp_name2 = [num2str(rms_class2), ' - ', num2str(rms_class3)]; % light
green
color3 = '#EDB120'; disp_name3 = [num2str(rms_class3), ' - ', num2str(rms_class4)]; % orange
color4 = 'red'; disp_name4 = [num2str(rms_class4), ' - max']; % red
best_rms;
for l = 1:length(egyeses_par(:,4)) % l - "L" --> betű :)
    best_rms2 = egyeses_par(1,4);
    if best_rms2 <= rms_class2;
        color = color1;
        disp_name = disp_name1;
    elseif rms_class2 < best_rms2 <= rms_class3 ;
        color = color2;
        disp_name = disp_name2;
    elseif rms_class3 < best_rms2 <= rms_class4;
        color = color3;
        disp_name = disp_name3;
    else rms_class4 < best_rms2;
        color = color4;
        disp_name = disp_name4;
    end
    plot([egyeses_par(1,5) egyeses_par(1,7)], [egyeses_par(1,6) egyeses_par(1,8)], ...
        'color', color, 'LineWidth', 2); % 'DisplayName', disp_name);
end
mx = mean(x);
my = mean(y);
plt_1 = plot(mx,my, '-', 'color', color1, 'LineWidth', 2, 'DisplayName', disp_name1);
plt_2 = plot(mx,my, '-', 'color', color2, 'LineWidth', 2, 'DisplayName', disp_name2);
plt_3 = plot(mx,my, '-', 'color', color3, 'LineWidth', 2, 'DisplayName', disp_name3);
plt_4 = plot(mx,my, '-', 'color', color4, 'LineWidth', 2, 'DisplayName', disp_name4);
lgnd = legend([plt_1 plt_2 plt_3 plt_4], 'Location', 'best');

```

```

title(lgnd,'Jelmagyarázat [RMS]');
% Az egyeneshez tartozó pontok kigyűjtése fájlba??
axis equal;
axis([grid_x_min grid_x_max grid_y_min grid_y_max]);
title(["Sorozatos RANSAC egyenes illesztés";filename]);
n; % Ellenőrzéshez kiíratni - lefutott a ciklus
ylabel('y_h_e_l_y_i [m]')
xlabel('x_h_e_l_y_i [m]')
hold off;
% Kép mentése
folder2 = [folder, '_egy'];
if ~exist(folder2, 'dir');
    mkdir(folder2)
end
picturename = sprintf([folder2, '/', filename, '_lines_%03d.jpg'],fnumb);
print(picturename, '-djpeg');
% clc;
% Kiírás fájlba - output:
% - az illesztett egyenesek paraméterei [m,b,db,RMS,x1,y1,x2,y3]
% - a talált egyenesek és azok oszályozása RMS alapján
% %% Metszessünk mindent, mindennel - bizonyos feltételek mellett
%dif = 5; % deg
derékszögtől mennyire térhetünk el
%tul = 0.15; % Mennyire lóghat túl?
loghat túl a metszesi tartomány
% Kezdőértékek a fájlok mentéséhez
i = 1;
j = 1;
n = 1;
intsec = [];
intsec_jo = [];
intsec_bad = [];
best_egy = [];
for i = 1:length(egyenes_par(:,1))
    b_i = egyenes_par(i,2);
    m_i = egyenes_par(i,1);

    for j = 1:length(egyenes_par(:,1))
        n = 1;
        b_j = egyenes_par(j,2);
        m_j = egyenes_par(j,1);
        % Egyenesek által bezárt szögek
        delta_alfa = rad2deg(atan(m_j)) - rad2deg(atan(m_i));
        if delta_alfa < 0;
            delta_alfa = delta_alfa + 360;
            if 270 - 2*dif < delta_alfa < 270 + 2*dif;
                delta_alfa = delta_alfa - 180;
            end
        end
        delta_alfa;
        % Metszéspontok számítása
        if 90 - dif <= delta_alfa <= 90 + dif ;
            x_intsec = (b_j - b_i)/(m_i - m_j);
            y_intsec = egyenes_par(i,1)*x_intsec + egyenes_par(i,2);
        else
            m_i = m_j;
            x_intsec = [];
            y_intsec = [];
        end
        intsec_seged2 = [x_intsec, y_intsec, delta_alfa];
        % A két vonal hosszán kívül eső értékek leszűrése egy "tul"logási értékkel megadva
        if (m_i - m_j == 0); % Ahol nincs metszés
            x_intsec = [];
            y_intsec = [];
            delta_alfa = [];
            best_egy_seged = [];
            intsec_seged2 = [x_intsec, y_intsec, delta_alfa];
        elseif m_i - m_j == 0 || grid_min_fnumb(1,1) >= x_intsec || x_intsec >=
grid_max_fnumb(1,1) || ...
            grid_min_fnumb(1,2) >= y_intsec || y_intsec >= grid_max_fnumb(1,2);
            x_intsec = [];
            y_intsec = [];
            delta_alfa = [];
            best_egy_seged = [];
            intsec_seged2 = [x_intsec, y_intsec, delta_alfa];
        % GRID-en kívüli metszés*ek szűrése
    end
end
end

```

```

elseif
(min([egyenes_par(i,5);egyenes_par(i,7);egyenes_par(j,5);egyenes_par(j,7)]) - tul >=
intsec_seged2(1) ||...
    max([egyenes_par(i,5);egyenes_par(i,7);egyenes_par(j,5);egyenes_par(j,7)])
+ tul <= intsec_seged2(1) ||...
    min([egyenes_par(i,6);egyenes_par(i,8);egyenes_par(j,6);egyenes_par(j,8)])
- tul >= intsec_seged2(2) || ...
    max([egyenes_par(i,6);egyenes_par(i,8);egyenes_par(j,6);egyenes_par(j,8)])
+ tul <= intsec_seged2(2));
    x_intsec = [];
    y_intsec = [];
    delta_alfa = [];
    best_egy_seged = [];
    intsec_seged2 = [x_intsec, y_intsec, delta_alfa];
elseif 90-dif >= intsec_seged2(:,3) || intsec_seged2(:,3) >= 90+dif ;
    intsec_seged2 = [];
else
    intsec_seged2 = intsec_seged2;
    best_egy_seged = [egyenes_par2(i,9),egyenes_par2(i,10)...
        egyenes_par2(i,11),egyenes_par2(i,12),egyenes_par2(j,9),...
        egyenes_par2(j,10), egyenes_par2(j,11),egyenes_par2(j,12)];
end
n = n + 1;
intsec_jo = [intsec_jo;intsec_seged2];
best_egy = [best_egy;best_egy_seged];

end

end
% Kívülre kell kerülnie a fájlba írás miatt
x = d(:,1);
y = d(:,2);
% A metszések megjelenítése az előző ábrán
try
figure(fnumb+10)
% plot(x,y,"k.");
hold on;
plot(intsec_jo(:,1),intsec_jo(:,2),'bo','MarkerSize',5,'DisplayName','Metszes')
axis([grid_x_min grid_x_max grid_y_min grid_y_max]);
ylabel('y_h_e_l_y_i [m]')
xlabel('x_h_e_l_y_i [m]')
% Kép mentése
folder3 = [folder,'_egy_pont'];
if ~exist(folder3, 'dir')
    mkdir(folder3)
end
picturename = sprintf([folder3, '/', filename, '_linepoints_%03d.jpg'],fnumb);
print(picturename, '-djpeg');
clf('reset')
hold off;
close all;
catch
continue
end
% Kiírás fájlba - output:
% - a korábban mentett képen a metszések megjelenítése körrel

best_egy2 = [best_egy2;best_egy];
intsec_jo2 = [intsec_jo2;intsec_jo];
end
%% Export, sarokpontok
%intsec_jo = [intsec_jo;intsec_jo];
R = coeff;
eltolt_xy = [pc_metszet(:,1)- x_mean, pc_metszet(:,2)- y_mean] ;
intsec_jo_traf = intsec_jo2(:, [1,2])*inv(R);
intsec_jo_traf = [intsec_jo_traf(:,1) + x_mean,intsec_jo_traf(:,2) + y_mean];
% Fájlba mentés - [y,x]
folder4 = [folder,'_metszesek'];
if ~exist(folder4, 'dir')
    mkdir(folder4)
end
% Kiírás fájlba - output: koordináta lista a metszések koordinátaival
dlmwrite([folder4, '/', filename, '_metszet_pontok.txt'],intsec_jo_traf,'delimiter',...
    ',', 'precision', '%6.3f')
% Kiírás fájlba - output: a metszések meghatározásához felhasznált egyenesek két pontja
dlmwrite([folder4, '/', filename, '_legjobb_egy.txt'],best_egy2,'delimiter',...
    ',', 'precision', '%6.3f')
close all;

```


3. melléklet:

savePC_by_grid.m

```

function [grid_min, grid_max, n] = savePC_by_grid(x_grid, y_grid, P_t, folder, filename)
% A gridmaker-ben elkészített rács alapján a P_t (transzformált - helyi
% rendszerben lévő) pontfelhő darabolása, majd ezeknek a daraboknak a
% kimentése a grid-ben megadott paraméterek alapján.

%folder = 'PC'
%filename = 'Lanchid'

% Mappa létrehozása, ahova menti a pontfelhőket
if ~exist(folder, 'dir')
    mkdir(folder)
end
% A cilushoz kiindulási értékek
size_grid = size(x_grid);
i = 1;
j = 1;
n = 1;
% Mátrixok az adatok tárolására
Seged_matrix11 = [];
Seged_matrix22 = [];
grid_min = [];
grid_max = [];

% Ciklus
for j = 1:size_grid(2)-1;
    for i = 1:size_grid(1)-1; % Lépkedés a sorok között
        %Sarokpontok (Bal alsó;jobb felső) meghatározása
        x_grid_min = x_grid(i,j);
        y_grid_min = y_grid(i,j);
        x_grid_max = x_grid(i+1,j+1);
        y_grid_max = y_grid(i+1,j+1);
        %Pontok kigyűjtése
        numb = find(P_t(:,1) >= x_grid_min & P_t(:,1) <= x_grid_max & P_t(:,2) >= y_grid_min &
P_t(:,2) <= y_grid_max);
        P_t_kivagat = P_t(numb,:);
        %Ahol nincs pont a rácsban ott lépjen tovább
        if P_t_kivagat ~= 0
            % plot(P_t_kivagat(:,1), P_t_kivagat(:,2), 'b. '); % Ellenőrzéshez
            % Mentse ki a pontokat
            fid = [folder, '/', num2str(n), '.txt'];
            dlmwrite(fid, P_t_kivagat);
            n = n + 1; % A fájlok számozását növelje eggyel
            Seged_matrix1 = [x_grid_min, y_grid_min];
            Seged_matrix11 = [Seged_matrix11; Seged_matrix1];

            Seged_matrix2 = [x_grid_max, y_grid_max];
            Seged_matrix22 = [Seged_matrix22; Seged_matrix2];
            % Törölje a pontok koordinátáit tartalmazó mátrix tartalmát
            clear P_t_kivagat
        else
            n = n; % HA nincs pont a mátrixban, akkor ne növeljen a fájl számozáson
            % NE mentsen ki üres mátrixot
            Seged_matrix1 = [];
            Seged_matrix2 = [];
        end
    end
    % n = n + 1; %Ha befejezte az adott oszlopot -> növeljen a számozáson
    clear P_t_kivagat % Tisztítsa ki a mátrixot
end
n = n - 1; % Ennyi fájl van a mappában
fprintf('A', folder, 'mappába', n, 'fájl mentve!')
grid_min = [grid_min; Seged_matrix11];
grid_max = [grid_max; Seged_matrix22];
% dlmwrite([folder, + '/grid_min.txt'] , grid_min);
% dlmwrite([folder, + '/grid_max.txt'] , grid_max);
end

```

4. melléklet: data2dxf.m

```

%% Ingtalan-nyilvántartási térkép felújítás pontfelhő alapján
% TDK 2020.
% Készítette: Hrutka Bence Péter
% Neptun: FDGJW8
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% A program seqRANSAC_on_pc.m átlal készített adatokat kiírja a metszéspontokat,
% egyenesek DXF fájlba.
% A kódhoz felhasznált anyagok
% https://uk.mathworks.com/matlabcentral/fileexchange/4895-dxf-export-for-3d-line-elements
% https://archive.edu.epito.bme.hu/local/coursepublicity/mod/resource/view.php?id=40166
% inputs: - folder: a mappa, ahova mentettük az eredményeket
%          folder = 'Barnag_Mintaterulet/Barnag_Mintater_1';
%          - filename: a prefixum a mentett adatok előtt
%          filename = 'Barnag_mintater';
% outputs - DXF fájlok:
%           - a metszéseket tartalmazó DXF
%           - a megtalált egyeneseket tartalmazó DXF
%           - a metszések számításához felhasznált egyenesek DXF
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; close all; clc; format longG
% Inputok:
% Mintaterület 1
folder = 'Barnag_mintaterulet_1/Barnag_mintater_1';
filename = 'Barnag_mintaterulet_1';

% Mintaterület 2
% folder = 'Barnag_mintaterulet_2/Barnag_mintater_2';
% filename = 'Barnag_mintaterulet_2';

% Mintaterület 3
% folder = 'Barnag_mintaterulet_3/Barnag_mintater_3';
% filename = 'Barnag_mintaterulet_3';

%% A megtalált egyenesek DXF-be - A kódot nem kell már módosítani
folder0 = [folder, '_egy_par'];
fload = [filename, '_par'];

% Mappában lévő fájlokra végzi el a dxf exportot
dirList = dir(fullfile(folder0, '*.txt'));
fnms = fullfile(folder0, {dirList.name});
fnum = length(fnms);

% A dxf fájl ahova menteni akarjuk
fullname = sprintf('%s.dxf', folder0);
fid = fopen(fullname, 'w');
fprintf(fid, '999\nTDK2020\n0\nSECTION\n2\nENTITIES\n0\n');
% Minden fájlból szedje ki a vonalakat
for i = 1 : fnum % fnum = 103; %
% Adatok kiszedése a megnyitott fájlból
%filename = [fload, num2str(fnum), '.txt'];
%data = load([folder0, '/', filename]);
data = load(char(fnms(i)));
X1 = data(:,9);
Y1 = data(:,10);
Z1 = zeros(length(X1),1);
X2 = data(:,11);
Y2 = data(:,12);
Z2 = Z1;
    for a = 1 :length(X1)
        fprintf(fid, 'LINE\n8\nMatlabVonalak\n');
        % Create new line element
        fprintf(fid, '10\n%.4f\n20\n%.4f\n30\n%.4f\n', X1(a), Y1(a), Z1(a));
        % First coordinate triple - starting point of line-element
        fprintf(fid, '11\n%.4f\n21\n%.4f\n31\n%.4f\n', X2(a), Y2(a), Z2(a));
        % Second coordinate triple - ending point of line-element
        fprintf(fid, '0\n');
    end
end %
fprintf(fid, 'ENDSEC\n0\nEOF\n');
fclose(fid);

%% A 'legjobb' egyenesek mentése DXF-be
% Mappa megadása

```

```

folder1 = folder;
folder2 = [folder1, '_metszesek'];
filename2 = [filename, '_legjobb_egy'];

% Beolvasás
data = load([folder2, '/', filename2, '.txt']);
X1 = data(:,1);
Y1 = data(:,2);
Z1 = zeros(length(X1),1);
X2 = data(:,3);
Y2 = data(:,4);
Z2 = Z1;

% A dxf fájl ahova menteni akarjuk
fid2 = [folder1, '_legjobb_egy'];
fullname2 = sprintf('%s.dxf',fid2);
fid2 = fopen(fullname2,'w');
fprintf(fid2, '999\nTDK2020\n0\nSECTION\n2\nHEADER\n9\n$PDMODE\n70\n3\n0\nENDSEC\n0\nSECTION\n2\nENTITIES\n0\n');
for i = 1 :length(X1)
    fprintf(fid2, 'LINE\n8\nMatlab_VonalJo\n');
    %create new line element
    fprintf(fid2, '10\n%.4f\n20\n%.4f\n30\n%.4f\n',X1(i),Y1(i),Z1(i));
    %first coordinate triple - starting point of line-element
    fprintf(fid2, '11\n%.4f\n21\n%.4f\n31\n%.4f\n',X2(i),Y2(i),Z2(i));
    %second coordinate triple - ending point of line-element
    fprintf(fid2, '0\n');
end
fprintf(fid2, 'ENDSEC\n0\nEOF\n');
fclose(fid2);

%% folder = 'Barnag_Mintaterulet'
% Teszt input
% folder = 'Barnag_mintaterulet_1/Barnag_mintater_1';
% filename = 'Barnag_mintaterulet_1';
% folder1 = 'Barnag_Mintaterulet/Barnag_Mintater_1';
% filename = 'Barnag_mintater';

% Mintaterület 1 input
folder1 = folder;

% A beolvasás innen indul:
folder2 = [folder1, '_metszesek'];
filename2 = [filename, '_metszet_pontok'];
%Barnag_mintater_metszet_pontok.txt

% Beolvasás
data = load([folder2, '/', filename2, '.txt']);
X = data(:,1);
Y = data(:,2);
Z = zeros(length(X),1);

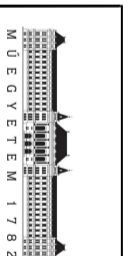
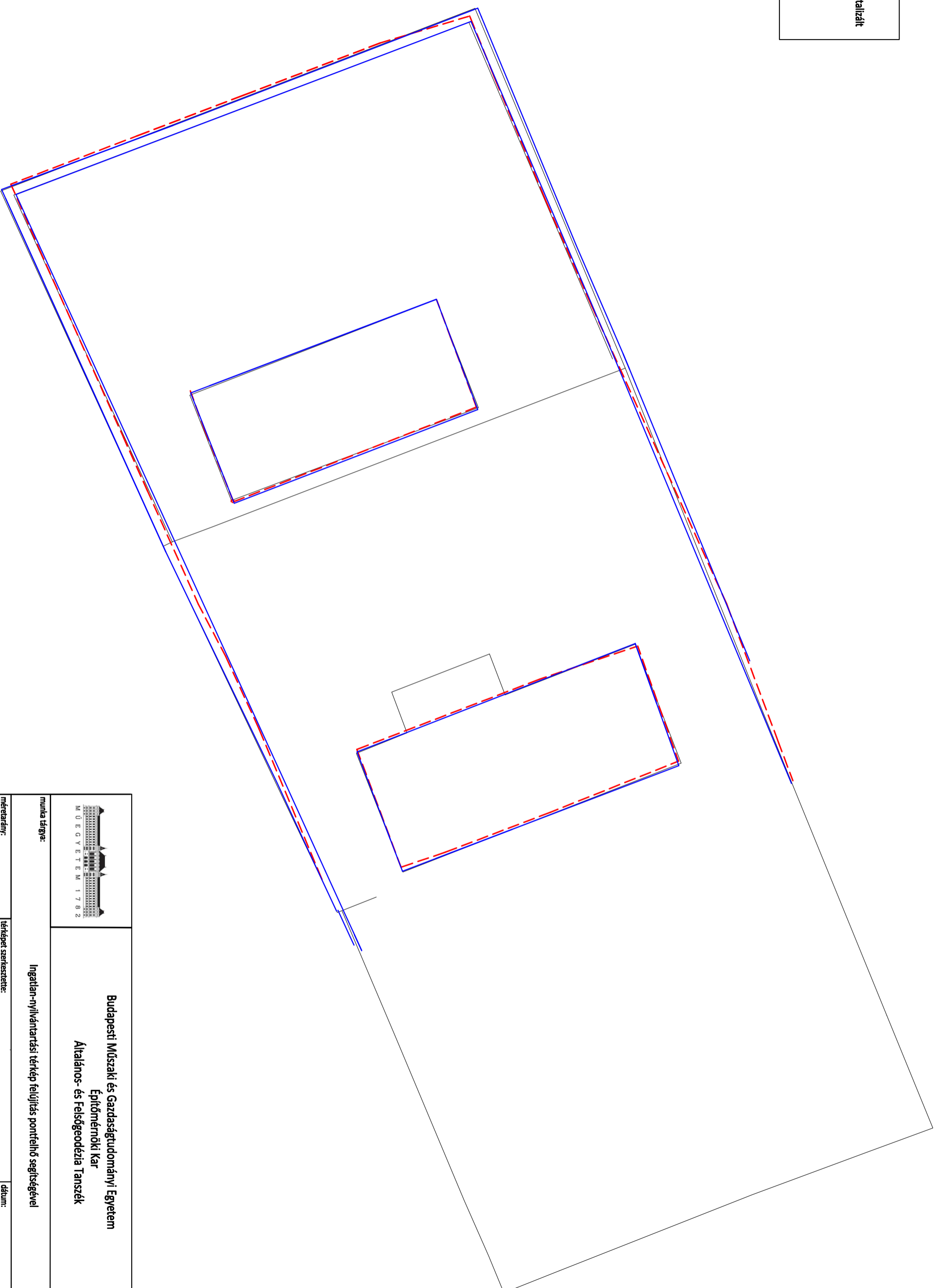
fid3 = [folder2, '_pontok'];
% A dxf fájl ahova menteni akarjuk
fullname3 = sprintf('%s.dxf',fid3);
fid3 = fopen(fullname3,'w');
fprintf(fid3, '999\nTDK2020\n0\nSECTION\n2\nENTITIES\n0\n');
for j = 1 :length(X)
    fprintf(fid3, 'POINT\n8\nMatlab_Pontok\n');
    %create new line element
    fprintf(fid3, '10\n%.4f\n20\n%.4f\n30\n%.4f\n',X(j),Y(j),Z(j));
    %second coordinate triple - ending point of line-element
    fprintf(fid3, '0\n');
end
fprintf(fid3, 'ENDSEC\n0\nEOF\n');
fclose(fid3);

```

5. melléklet:

Az elkészített rajzi részek:

Jelmagyarázat	
—	Kézzel digitalizált
- - -	AutoCAD
—	Matlab



Budapesti Műszaki és Gazdaságtudományi Egyetem
Építőmérnöki Kar
Általános- és Felsőgeodézia Tanszék

Ingatlan-nyilvántartási térkép felújítás pontfelhő segítségével

munka tárgya:

méretarány:
M = 1 : 250

térkép szerkesztette:
Hrutka Bence Péter

dátum:
2020.10.24.

munkaszám:

rajzi állomány neve:
AutoCAD Section lines és Matlab eredményekből
megtárolható elemek - 1. mintaterületen

fájlnev:
1_mintaterulet_levezetett.pdf

AT1

konzulensek:
Dr. Takács Bence Dr. Siki Zoltán

fájlteret:
211 KB

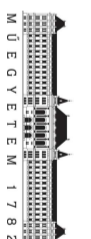
vetület:

EOV





Jelmagyarázat

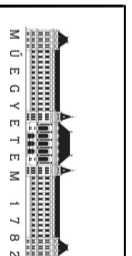
—	Kézzel digitalizált
—	AutoCAD
—	Matlab
×	Metszéspontok



 <p>Budapesti Műszaki és Gazdaságtudományi Egyetem Építőmérnöki Kar Általános- és Felsőgeodézia Tanszék</p>		<p>Inggatlan-nyilvántartási térkép felújítás pontfelhő segítségével</p>	
méretarány: M = 1 : 250	térképért szerkesztette: Hrutka Bence Péter	dátum: 2020.10.24.	
munkaszám: AT1	rajzi állomány neve: AutoCAD Section lines és Matlab segítségével meghatározott nyers eredmények - 1. mintaterületen	fájlneve: _mintaterulet_nyers.pdf	
vetület: EOV	konzulensek: Dr. Takács Bence Dr. Siki Zoltán	fájlnevet: 215 KB	

Jelmagyarázat

	Kézrel digitalizált
	AutoCAD
	Matlab
	Metszéspontok



Budapesti Műszaki és Gazdaságtudományi Egyetem
Építőmérnöki Kar
Általános- és Felsőgeodézia Tanszék

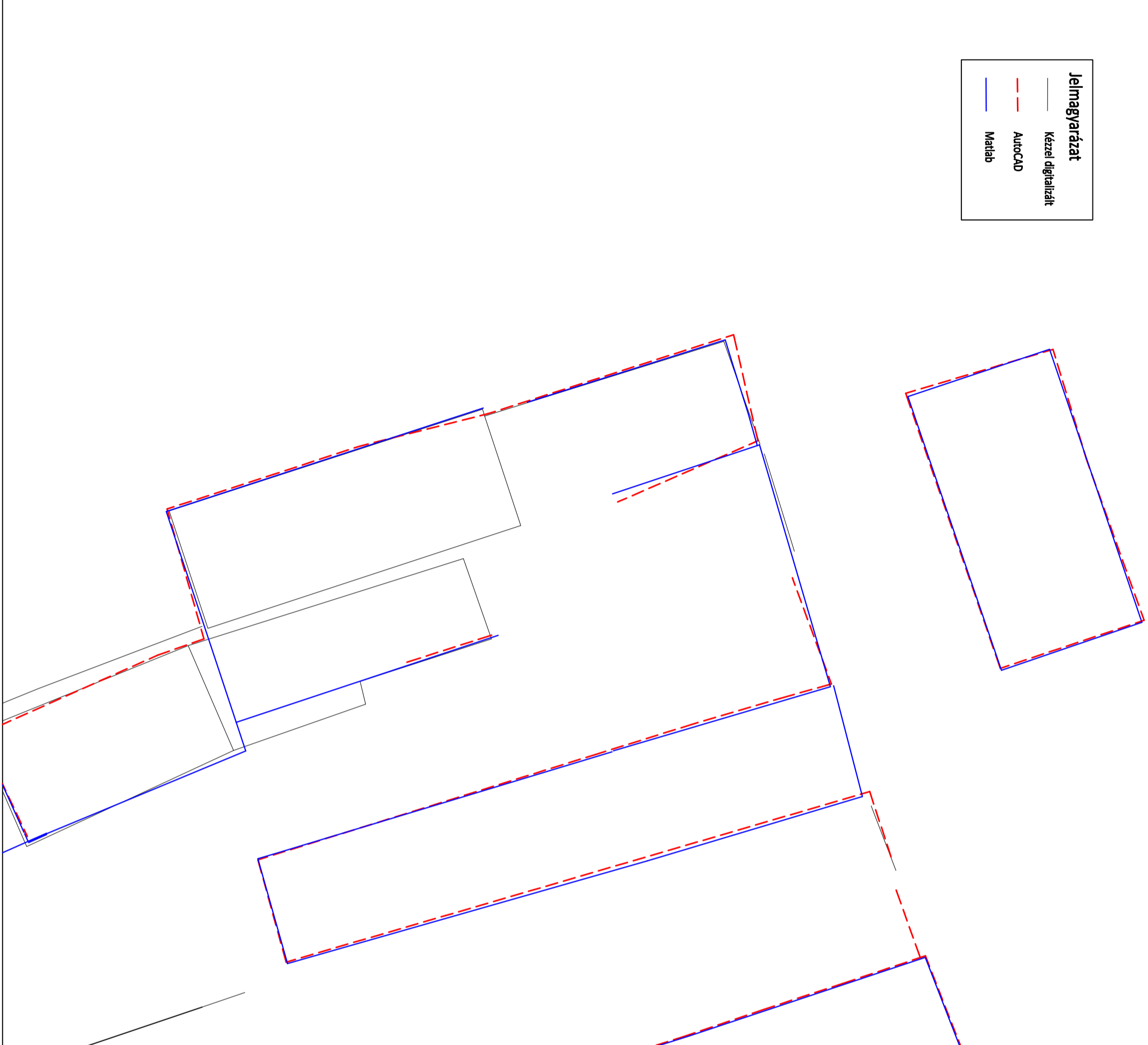
munka tárgya: Ingatlan-nyilvántartási térkép felújítás pontfelhő segítségével

méretarány: M = 1 : 250
térkép szerkesztette: Hrutka Bence Péter
dátum: 2020.10.24.

munkaszám: AT1
rajz állomány neve: AutoCAD Section lines és Matlab rajzított eredményei
rajzítottási eszközök használata után az I. mintaterületen
fájl név: I_mintaterulet_tisztított.pdf

vetület: EO V
konzulensek: Dr. Takács Bence Dr. Siki Zoltán
fájl méret: 215 KB

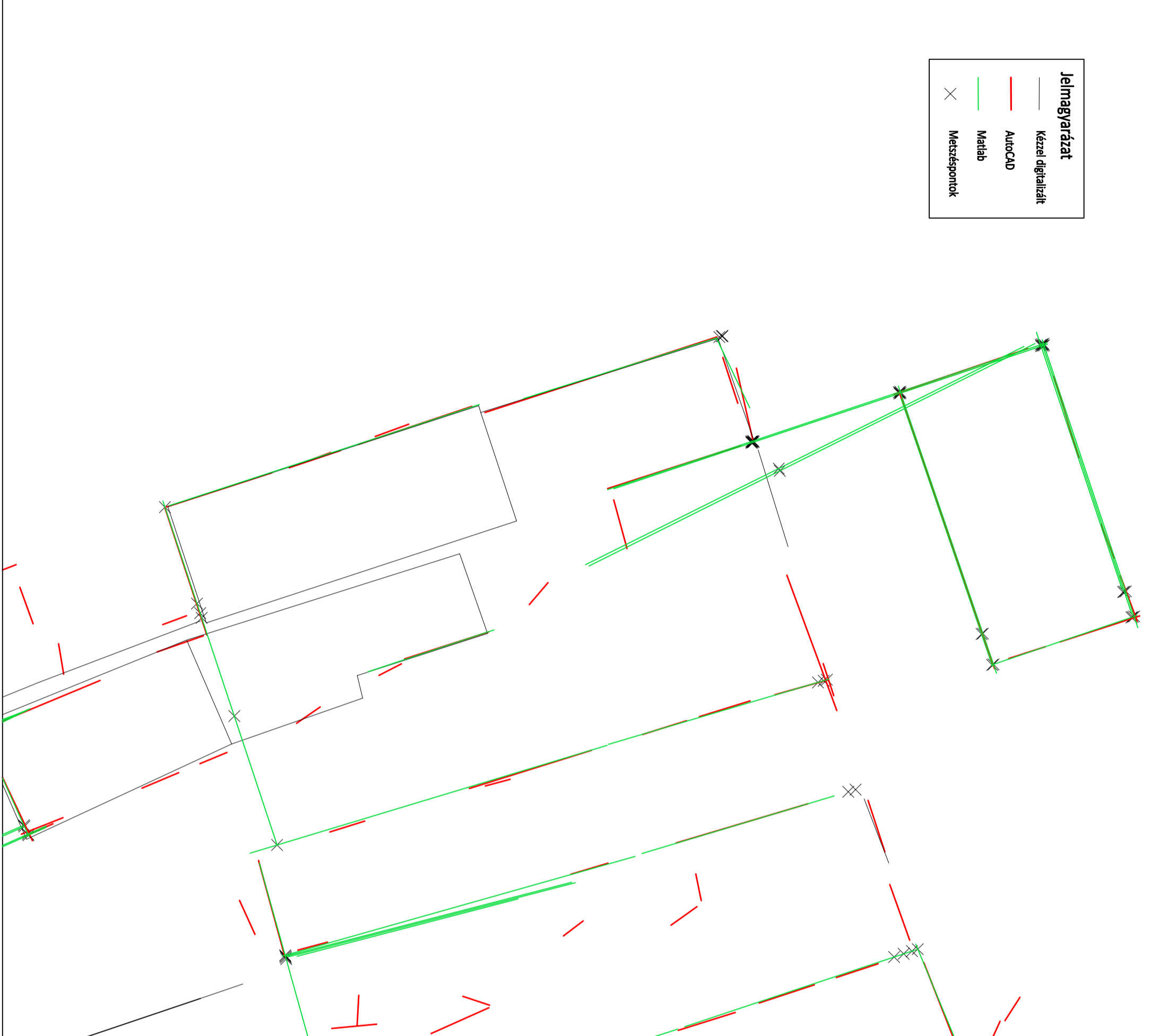
Jelmagyarázat	
	Kézzel digitalizált
	AutoCAD
	Matlab

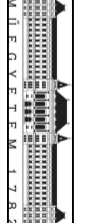






<p>Budapesti Műszaki és Gazdaságtudományi Egyetem Építőmérnöki Kar Általános- és Felsőgeodézia Tanszék</p>	
<p>munka tárgya: Ingatlan-nyilvántartási térkép felújítás pontfelhő segítségével</p>	
<p>méretarány: M = 1 : 250</p>	<p>térképért szerkesztette: Hrutka Bence Péter</p>
<p>munkaszám: AT12</p>	<p>rajzi állomány neve: AutoCAD Section lines és Matlab eredményekből meghatározható elemek - II. mintaterületen</p>
<p>vetület: EOV</p>	<p>konzulensek: Dr. Takács Bence Dr. Siki Zoltán</p>
	<p>dátum: 2020.10.24.</p>
	<p>fájlnev: II_mintaterulet_levezetett.pdf</p>
	<p>fájlméret: 212 KB</p>

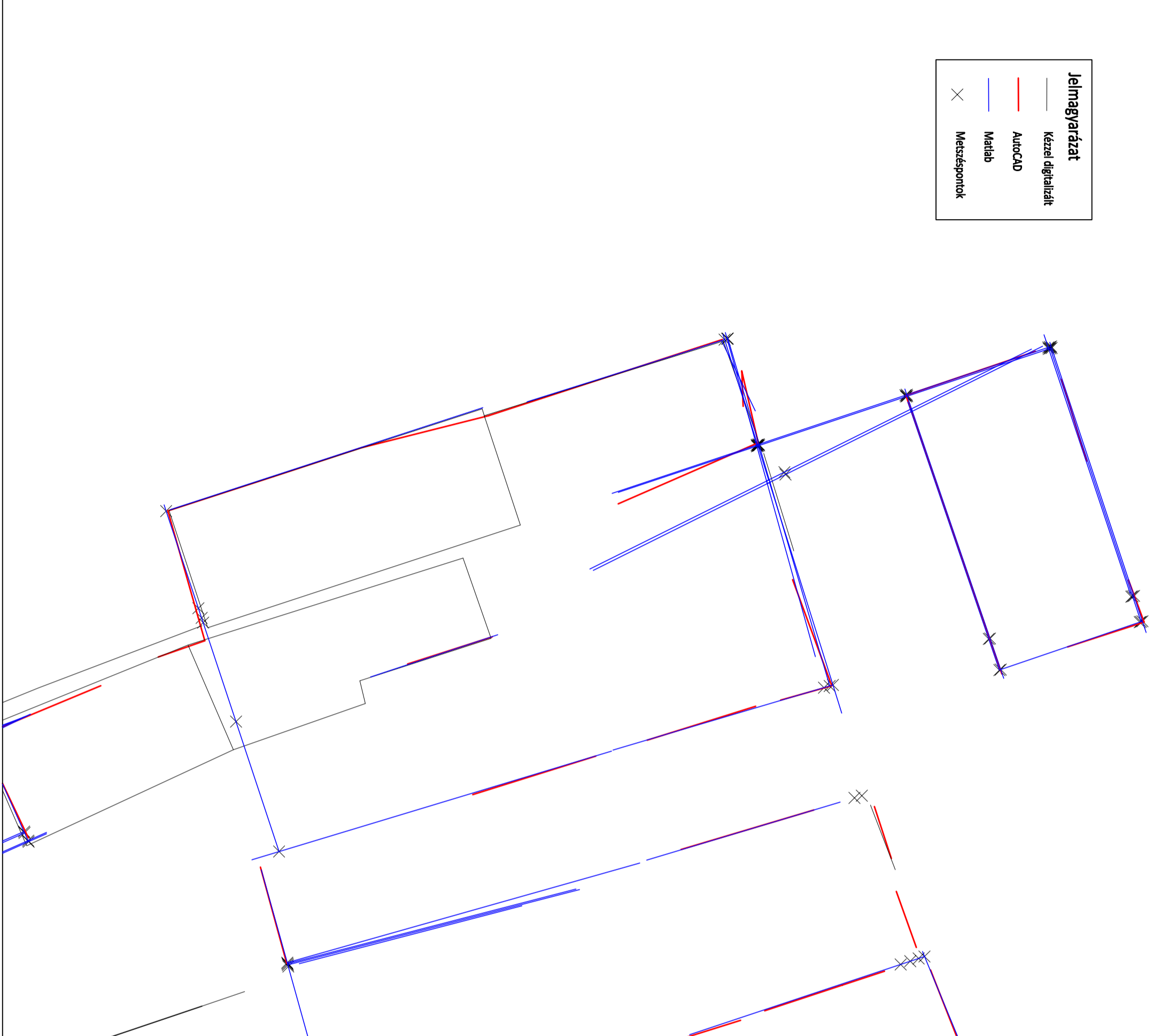
Jelmagyarázat

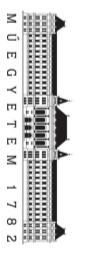
—	Kézzel digitalizált
—	AutoCAD
—	Matlab
×	Metszéspontok



 <p>Budapesti Műszaki és Gazdaságtudományi Egyetem Építőmérnöki Kar Általános- és Felsőgeodézia Tanszék</p>	
<p>munka tárgya: Ingatlan-nyilvántartási térkép felújítás pontfelhő segítségével</p>	
<p>méretarány: M = 1 : 250</p>	<p>térkép szerkesztette: Hrutka Bence Péter</p>
<p>munkaszám: AT12</p>	<p>rajzi állomány neve: Automatizálással meghatározott eredmények összehasonlítása I. munkaterületen</p>
<p>vetület: EOV</p>	<p>konzulensek: Dr. Takács Bence Dr. Siki Zoltán</p>
	<p>dátum: 2020.10.24.</p>
	<p>állomány neve: II_mintaterulet_nyers.pdf</p>
	<p>állomány mérete: 224 KB</p>

Jelmagyarázat	
	Kézzel digitalizált
	AutoCAD
	Matlab
	Metszéspontok

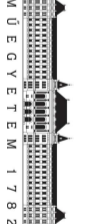


 M Ű E G Y E T E M 1 7 8 2		Budapesti Műszaki és Gazdaságtudományi Egyetem Építőmérnöki Kar Általános- és Felsőgeodézia Tanszék	
Ingatlan-nyilvántartási térkép felújítás pontfelhő segítségével			
művelet:	M = 1 : 250	térképet szerkesztette:	Hrutka Bence Péter
munkaszám:	AT12	rajzi állomány neve:	AutoCAD Section lines és Matlab rajzított eredményei rajztisztítási eszközök használata után a II. miniatűrlelten
vetület:	EOV	konzulensek:	Dr. Takács Bence Dr. Siki Zoltán
		dátum:	2020.10.24.
		fájlnev:	II_miniatűrület_tisztított.pdf
		fájlméret:	222 KB



Jelmagyarázat

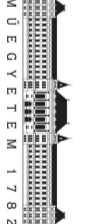
- Kézzel digitalizált
- - - AutoCAD
- Matlab

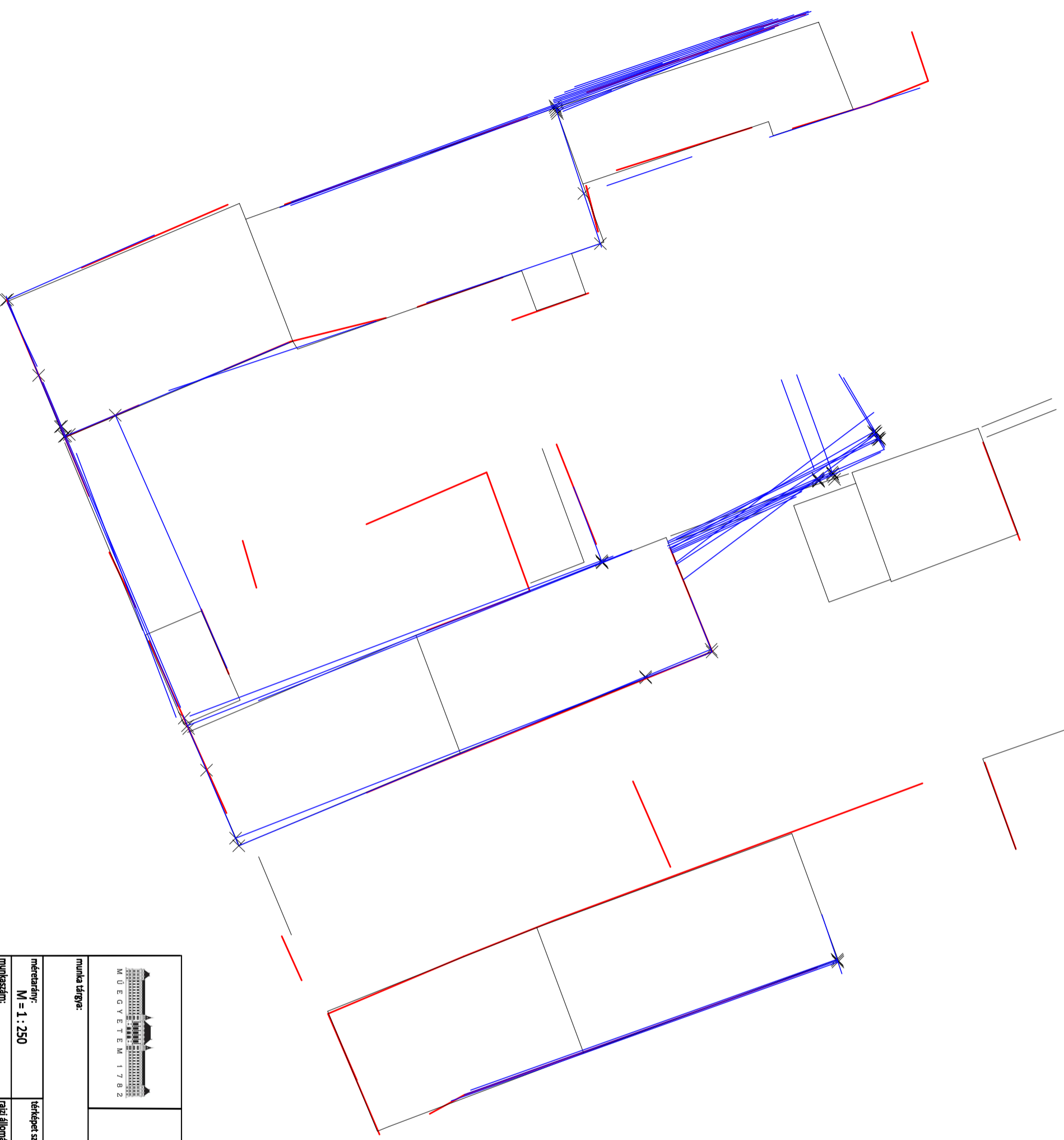
 <p>Budapesti Műszaki és Gazdaságtudományi Egyetem Építőmérnöki Kar Általános- és Felsőgeodézia Tanszék</p>		<p>Ingatlan-nyilvántartási térkép felújítás pontfelhő segítségével</p>	
<p>méretarány: M = 1 : 250</p>	<p>térkép szerkesztette: Hrutka Bence Péter</p>	<p>dátum: 2020.10.24.</p>	
<p>munkaszám: AT3/a</p>	<p>rajz állomány neve: AutoCAD Section lines és Matlab eredményekből meghathározható elemek - III. műtérületen</p>	<p>filjnév: III_a_műtérület_levezetett.pdf</p>	
<p>vetület: EOV</p>	<p>konzulensek: Dr. Takács Bence Dr. Siki Zoltán</p>	<p>filjnév: 213 KB</p>	



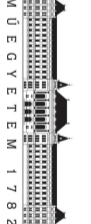
Jelmagyarázat




- Kézzel digitalizált
- AutoCAD
- Matlab
- × Metszéspontok

 <p>Budapesti Műszaki és Gazdaságtudományi Egyetem Építőmérnöki Kar Általános- és Felsőgeodézia Tanszék</p>		<p>Ingatlan-nyilvántartási térkép felújítás pontfelhő segítségével</p>	
méretarány: M = 1 : 250	térképért szerkesztette: Hrutka Bence Péter	dátum: 2020.10.24.	
munkaszám: AT3/a	rajz állomány neve: Automatizálással meghatározott eredmények összehasonlítása III. mintaterületen	állomány: III_a_mintaterulet_nyers.pdf	
vetület: EOV	konzulensek: Dr. Takács Bence Dr. Siki Zoltán	állomány mérete: 224 KB	



Jelmagyarázat	
—	Kézzel digitalizált
—	AutoCAD
—	Matlab
×	Metszéspontok

 <p>Budapesti Műszaki és Gazdaságtudományi Egyetem Építőmérnöki Kar Általános- és Felsőgeodézia Tanszék</p>	
<p>munka tárgya: Ingatlan-nyilvántartási térkép felújítás pontfelhő segítségével</p>	
<p>méretarány: M = 1 : 250</p>	<p>térképért. szerkesztette: Hrutka Bence Péter</p>
<p>munkaszám: AT3/a</p>	<p>rajzi állomány neve: AutoCAD Section lines és Matlab rajzított eredményei rajztisztítási eszközök használata után a III. mintaterületen</p>
<p>vetület: EOV</p>	<p>konzulensek: Dr. Takács Bence Dr. Siki Zoltán</p>
	<p>dátum: 2020.10.24.</p>
	<p>állomány: III_a_mintaterulet_tisztított.pdf</p>
	<p>állomány mérete: 223 KB</p>

Jelmagyarázat	
	Kézzel digitalizált
	AutoCAD
	Matlab







Budapesti Műszaki és Gazdaságtudományi Egyetem
Építőmérnöki Kar
Általános- és Felsőgeodézia Tanszék

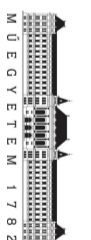
munka tárgya: Ingatlan-nyilvántartási térkép felújítás pontfelhő segítségével

méretarány: M = 1 : 250	térképész szerkesztette: Hrutka Bence Péter	dátum: 2020.10.24.
munkaszám: AT3/b	rajzi állomány neve: AutoCAD Section lines és Matlab eredményekből megtárolható elemek - III. mintaterületen	fájlnev: III_b_mintaterulet_levezetett.pdf
vetület: EOV	konzulensek: Dr. Takács Bence Dr. Siki Zoltán	fájlméret: 213 KB





Jelmagyarázat

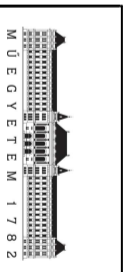
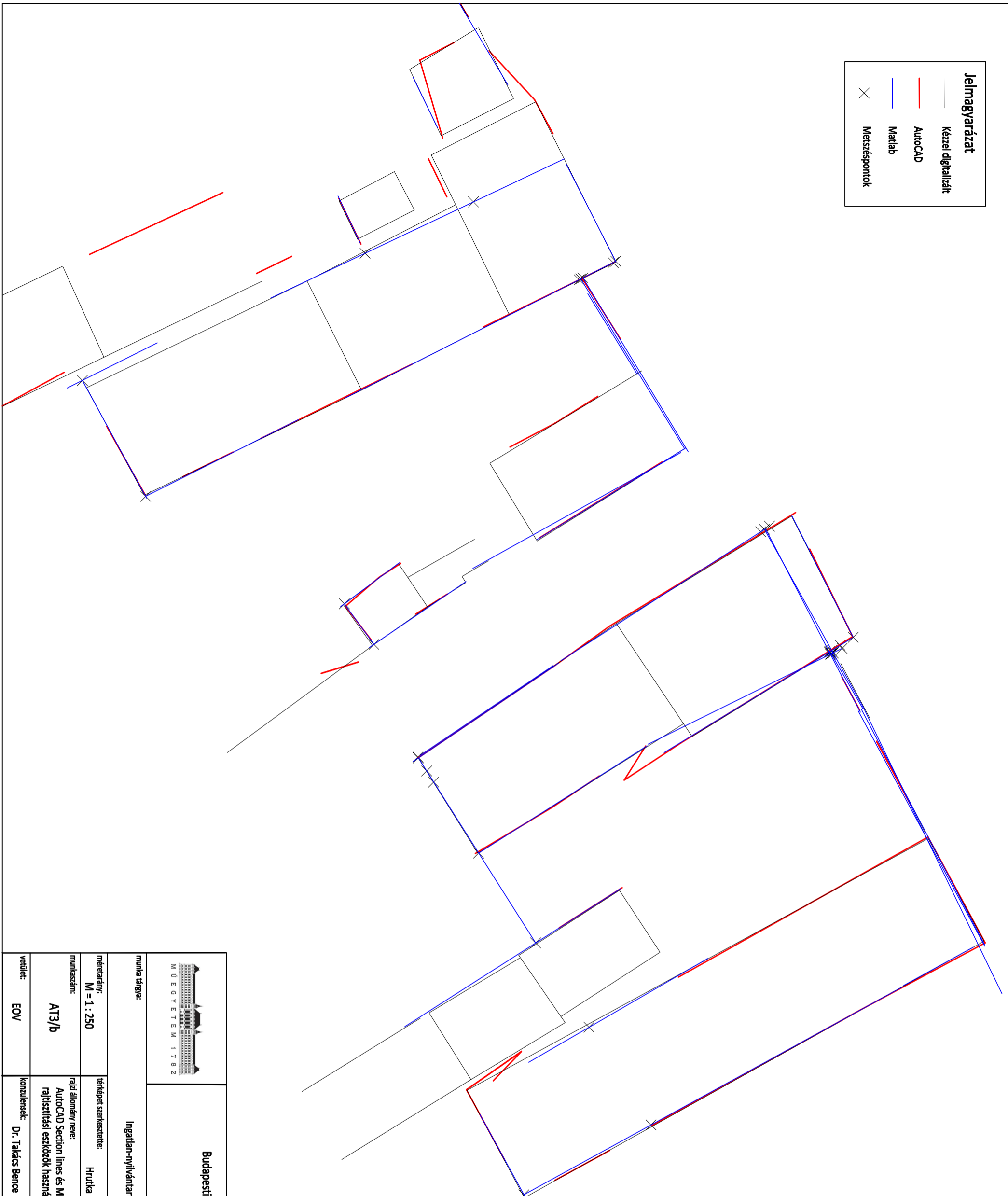
	Kézzel digitalizált
	AutoCAD
	Matlab
	Metszéspontok



 <p>Budapesti Műszaki és Gazdaságtudományi Egyetem Építőmérnöki Kar Általános- és Felsőgeodézia Tanszék</p>	
<p>Munka tárgya: Ingatlan-nyilvántartási térkép felújítás pontfelhő segítségével</p>	
<p>méretarány: M = 1 : 250</p>	<p>térképért. szerkesztette: Hrutka Bence Péter</p>
<p>munkaszám: AT3/b</p>	<p>rajz állomány neve: Automatizálással meghatározott eredmények összehasonlítása III. mintaterületen</p>
<p>vetület: EOV</p>	<p>konzulensek: Dr. Takács Bence Dr. Siki Zoltán</p>
	<p>dátum: 2020.10.24.</p>
	<p>állomány neve: III_b_mintaterulet_nyers.pdf</p>
	<p>állomány mérete: 223 KB</p>

Jelmagyarázat

	Kézzel digitalizált
	AutoCAD
	Matlab
	Metszéspontok



Budapesti Műszaki és Gazdaságtudományi Egyetem
 Építőmérnöki Kar
 Általános- és Felsőgeodézia Tanszék

munka tárgya: Ingatlan-nyilvántartási térkép felújítás pontfelhő segítségével

méretarány: M = 1 : 250	térképért. szerkesztette: Hrutka Bence Péter	dátum: 2020.10.24.
munkaszám: AT3/b	rajzi állomány neve: AutoCAD Section lines és Matlab rajzészítőt eredményei rajzítási eszközök használata után a III. mintaterületen	fájlnev: III_b_mintaterulet_tisztított.pdf
vetület: EOV	konzulensek: Dr. Takács Bence Dr. Siki Zoltán	fájlméret: 224 KB