

**Óbudai Egyetem**  
**Alba Regia Műszaki Kar**  
**Geoinformatikai Intézet**



**TUDOMÁNYOS DIÁKKÖRI DOLGOZAT**

**NYÍLT FORRÁSKÓDÚ PROGRAMOK  
ALKALMAZÁSA A PONTFELHŐ  
FELDOLGOZÁSBAN**

**Szerző: Gáti Bence**

földmérő és földrendező mérnöki  
BSc. szak, III. évf.

**Konzulens: Dr. Tóth Zoltán**

adjunktus

**Székesfehérvár, 2017**

# Tartalomjegyzék

|  |    |
|--|----|
| Bevezetés.....   | 1  |
| 1. Tarnaszentmária temploma .....                          | 2  |
| 2. Felmérési technológiák .....                            | 4  |
| 2.1 Felhasznált technológiák .....                         | 4  |
| 2.2 A földi lézerszkennerek általános áttekintése .....    | 4  |
| 3. Terepi mérés.....                                       | 5  |
| 4. Leica Cyclone.....                                      | 8  |
| 5. Transzformációk.....                                    | 9  |
| 5.1. Térbeli hasonlósági transzformáció.....               | 9  |
| 5.2. Térbeli egybevágósági transzformáció.....             | 10 |
| 6. GNU Octave .....  | 11 |
| 7. CloudCompare .....                                      | 13 |
| 7.1. Pontfelhők transzformálása .....                      | 13 |
| 8. Mérési zaj eltávolítása .....                           | 16 |
| 9. Metszetek készítése .....                               | 17 |
| 10. Lézerszkenneres pontfelhők vizsgálata .....            | 18 |
| 10.1 Felhő-felhő távolság .....                            | 18 |
| 11. Földi fotogrammetria.....                              | 19 |
| 11.1 Pontfelhő építése amatőr felvételekből .....          | 19 |
| 11.2 Pontfelhők durva illesztése.....                      | 20 |
| 11.3 ICP.....  | 21 |
| 12. Fotogrammetriai pontfelhők pontossági vizsgálata:..... | 22 |
| 13. Összegzés .....  | 24 |
| 13.1 A dolgozat végtermékei.....                           | 25 |
| 14. Irodalom jegyzék.....                                  | 27 |

## Bevezetés

A nyílt forráskódú, más szóval szabadszoftverek térhódítása rohamos ütemben növekszik. Az informatika minden területéről fel lehet sorolni legalább egy olyan programot, ami nyílt forráskódú. Mit is jelent a szabadszoftver kifejezés?

Azokat a programokat sorolhatjuk szabadszoftver (free software) kategóriába, amelyek megfelelnek a következő feltételeknek: felhasználói bármilyen céllal futtathatják a programot; tanulmányozhatják a program működését, és azt személyre szabhatják (szabadon elérhető forráskód); másolatokat tehetnek közzé; tökéletesíthetik a programot.

Elnevezésük (free) félreértéshez vezetett, ugyanis a közzétételnek nem kötelező ingyenesnek lennie, sokkal inkább a jogi szabadságra utal. Ezért vezették be a későbbiekben a nyílt forráskód kifejezést.

Szerte a világon programozók milliói segítenek felebarátaiknak azzal, hogy számukra anyagilag nem feltétlenül megtérülő fejlesztéseket a köz javára bocsátanak. A közösségek alakulnak és az összefogás gyümölcseként születnek olyan szoftverek, amelyek szinte kompromisszumok nélkül helyettesítik kereskedelmi forgalomban lévő társaikat. Érdemes rájuk figyelmet fordítani és használni őket, akkor is, ha alkalomadtán több hozzáértést igényelnek.

# 1. Tarnaszentmária temploma

A dolgozatban a tarnaszentmáriai római katolikus műemlék templom földi lézerszkenneres felmérése kapcsán mutatom be a feldolgozásnál használható nyílt forráskódú szoftverek alkalmazási lehetőségeit, továbbá a feldolgozás során kapott alaprajzokat, metszeteket és a földi fotogrammetriai felmérésből nyerhető pontfelhők megbízhatóságát, illetve ezen adatok feldolgozásának előnyeit és hátrányait.

Tarnaszentmárián található hazánk, még működő egyik legrégebbi temploma, melynek további különlegessége, hogy magyar és csehszlovák közlemények szerint is valószínűsíthetően a honfoglalás előtti szláv korszakból származik és fejedelmi temetkezőhelynek készült. (<https://hu.wikipedia.org/wiki/Tarnaszentmária>)

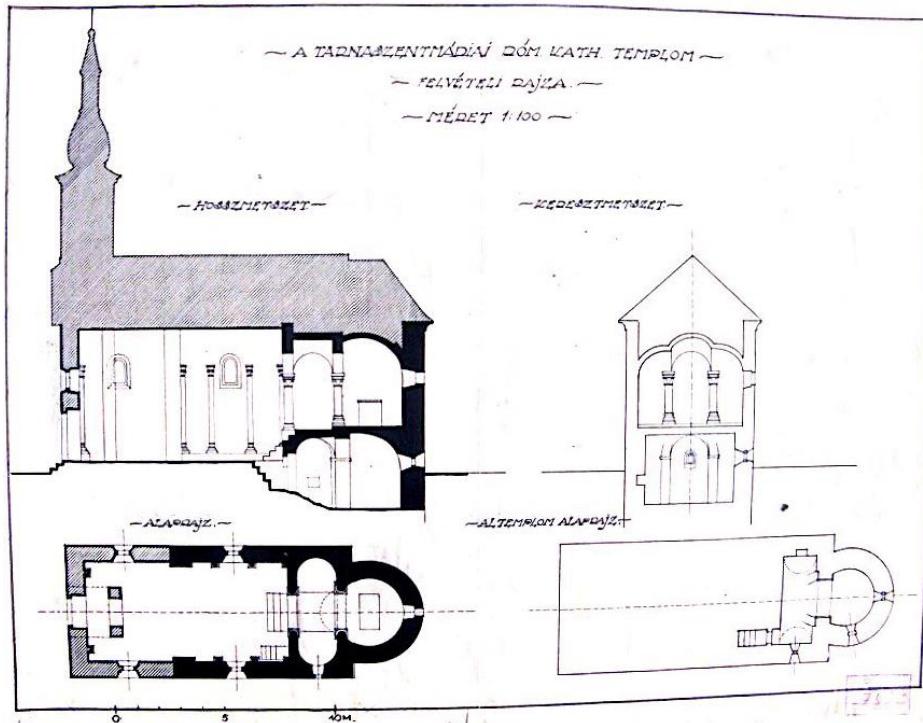
Egyes kutatóink azt állítják, hogy ez a körtemplom az egyik legfontosabb hazánk román kori emlékei közül (XI-XII század). Gerevich Tibor megállapítása szerint a templom a XII. század közepén épülhetett fel. Csemegi József viszont a IX-X századra teszi a hajó részek megépítését, amit az ókeresztény és bizánci jellegzetességek feltűnésével támaszt alá. Ilyen például a vékony oszlopok lábánál futó díszes padka. Véleménye szerint, „a tarnaszentmáriai templom hajóját az Aba nemzetség világi, pogány gyülekezőhelyének tekinti, amelynek díszítőformáit őseink a kazár birodalomból való kiválásunk után (969) hoztak magukkal.” Egy évtized múltán Mencl csehszlovák kutató osztja Csemegi véleményét. Később Major Jenő egy publikációjában a Tarna-völgy korai történetét kutatta. Arra a megállapításra jutott, hogy Szent István király egyik nővére lakhatott ott, aki feleségül ment Sámuel bolgár cár fiú gyermekéhez. Megszületett gyermekük, akit cárrá nevelés céljából szerzetesek neveltek, valószínűsíthetően Feldebrőn. Ebbe a keretbe helyezi a feldebrői és tarnaszentmáriai körtemplom megépülését is. A két települést csak Verpelét község választja el egymástól, amiről szintén úgy tartják, hogy már a honfoglalás előtt is lakott volt, kelta és szkíta leleteket mellet bronzkori tárgyakat is feltártak az ott dolgozó régészek.



1.1 ábra A feldebrői és a tarnaszentmáriai templom  
(<http://mapio.net/pic/p-7624883>, <http://www.skyscrapercity.com/showthread.php?t=1792363>)

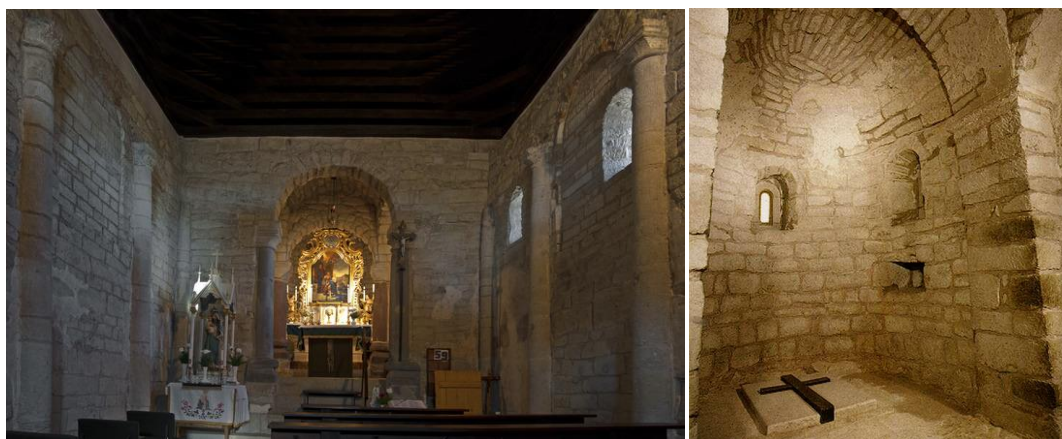
További régészeti feltárások során egyértelműen kiderült, hogy a tarnaszentmáriai templom szentélye és hajó része is egy időben épült. „A kisméretű altemplom hajójában egy, az épülethez igazodó sírt, és annak szentélyében egy egykor jelentős személyt őrző „aknasírt” tártak fel, többször megbolygatott csontokkal. Ez utóbbi adat egyértelművé teszi, hogy ez a nálunk különleges értéket képviselő kis templom eredetileg nem pogány gyülekezőhelyként, hanem

egy igen jelentős család temetkezőhelyeként — legalábbis részben, más rendeltetéssel együtt — épült, még a magyar román kori építészet kialakulása előtt.” Építetőként két személy valószínűsíthető: Géza fejedelem, vagy testvére Mihály. (Kozák, 1984)



1.2 ábra a tarnaszentmáriai templom felvételi rajza M=1:100 méretarányban (Forrás: Forster Központ, Budapest)

Az osztatlan hajó rész majdnem tökéletes négyzet, ahonnan hat lépcső a magasabban elhelyezkedő szentélybe, öt lépcső pedig az altemplomba vezet.



1.3 ábra Ma is Árpád kori állapotát megőrző szentély és altemplom  
(forrás: <http://www.muemlekem.hu/fotopalyazat?id=5687>, <http://mek.oszk.hu/01900/01954/html/index645.html> )

## 2. Felmérési technológiák

### 2.1 Felhasznált technológiák

Háromféle eszközt használtunk: UAV-t, földi fotogrammetriát, lézerszkennert.

A pilóta nélküli repülő eszközzel készített képekből 3D modell és pontfelhő előállítás lett volna a célunk, azonban a viharos szél miatt a felszállás meghiúsult, így ezzel a technológiával nem tudtam foglalkozni.

A földi fotogrammetriához a fényképeket két féle digitális kamerával készítettem, egy Nikon L340 és egy Sony Alpha digitális fényképező géppel. Körbe fotózás során előbbivel 20 megapixeles képeket készítettem autófókusszal, utóbbinál 12 megapixeles képeket úgy, hogy a fókusz távolságot nem változtattam. Azok a felső élek, amik alulról nem látszanak, sajnos lyukasak lettek a modellben. Ez a hiba kiküszöbölhető lett volna, ha a hiányzó részekről felülről, drónnal készített képeket is fel tudom használni a feldolgozás során.

### 2.2 A földi lézerszkennerek általános áttekintése

A legelső lézerszkennerek az 1990-es években jelentek meg, ez a technológia nem rendelkezik nagy múltal. Alapját képezik olyan optikai vívmányok, mint a távmérés. Ez jó alapot ad, hiszen kiforrott technológia, előnyei közé sorolhatók, hogy objektív módon mér, aminek a folyamata nagyon jól automatizálható, illetve aktív energia forrásának köszönhetően nem szükséges, hogy az észlelt tárgy meg legyen világítva, akár ködben, vagy éjjel is dolgozhatunk vele. Bár a levegőben lévő pára és szálló por negatív hatással van a méréseinkre. A pozitív tulajdonságait ellensúlyozza, hogy bár megvilágítás nem szükséges, de az összelátást biztosítani kell a műszer és a mérendő objektum között, továbbá különös figyelmet kell fordítani a kitakarásból eredő árnyékok leképződésének megelőzésére. (Staiger, 2013)

A lézerszkennerek két fő kategóriára oszthatók a tükör mozgása alapján. A nagyobb megbízhatóságú oszcilláló tükrös és a több pontot mérni képes forgó tükörrel ellátott műszerekre. A távolságmérés alapján megkülönböztetünk: idő (ToF, Time of Flight) és fázis mérések (pb, phase-based) műszereket. Előbbinek a magasabb hatótávolság, utóbbinak a nagyobb megbízhatóság az előnye. (Berényi et al., 2012)

Szolgáljon példaként két Leica lézerszkennerek összehasonlítása:

| Műszer típusa                              | <b>Leica HDS7000</b>   | <b>Leica C10</b> |
|--|------------------------|------------------|
| Távmérés technológiája                     | PB                     | ToF              |
| Mérési távolság (min.-max.) [m]            | 0.3-187                | 0.1-300          |
| Pontosság (laboratóriumi körülmények) [mm] | 0.5 → 10m<br>10 → 100m | 4-6              |
| Lézer osztály                              | 1                      | 3R               |

2.2.1 táblázat Földi lézerszkennerek összehasonlítása

Látható, hogy az időmérési technológián alapuló távolság meghatározás közel fele akkora megbízhatóságú, mint a fázis mérésen alapuló távolsághatározásé, de csaknem kétszer nagyobb ható távolságú.

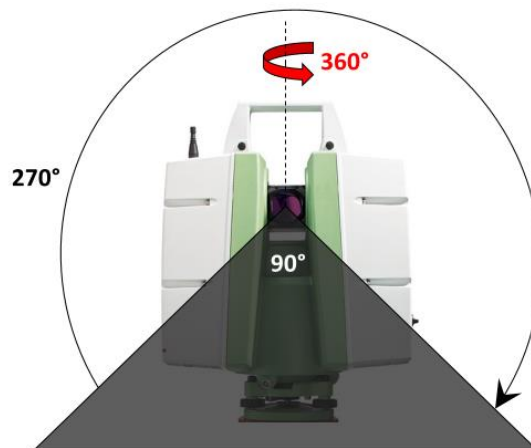


2.2.1 ábra Leica HDS7000 és Leica ScanStation C10

(forrás: <http://www.geotech.sk/Produkty/Laserove-skenery-HDS/Leica-HDS7000.html?id=3123>; <http://leica-geosystems.com/products/laser-scanners/scanners/leica-scanstation-c10>)

### 3. Terepi mérés

A felmérés során az előző példában vizsgált Leica ScanStation C10 típusú műszerrel dolgoztunk. Felbontását úgy állítottam be, hogy a legtávolabbi pontra értelmezve egy centiméteres legyen a felbontás, ami azt jelenti, hogy a tereptárgyról készülő pontfelhő legkisebb felbontása egy cm lesz, míg a közelebbi részeknél a felbontás nő. A műszer fotókat is készített a szkennelt tér egészéről, így színhelyes pontfelhőt lesz képes előállítani a szoftver. Ezeket tömörítés mentes RAW formátumban tárolta. A templom belső tereiben panoráma szkennelést végeztünk, ami azt jelenti, hogy a műszer a teret  $360^\circ$ -ban mérte vízszintes értelemben, magassági értelemben pedig  $+90^\circ$  és  $-45^\circ$ -os magassági szögtartományában képes méréseket végezni. Árnyéka kúp alakban képződik le. Annak érdekében, hogy a kitarítás a lehető legkisebb mértékű legyen, a műszerállás magassága mindig minimális volt.



3.1 ábra A lézerszkennelő mérési tartományai és árnyéka

A külső részeket a gazdaságosabb mérés szempontjából úgynevezett ablakos szkenneléssel mértük be. Ez a lehetőség a Leica esetében QuickScan néven fut. Itt lehetőségünk van négy szögértékkel egy tetszőleges téglalap alakú területet lehatárolni, akár kézzel beírt adatokkal, akár az adott irányra fordítva a műszert, rögzítve az irányértéket. Képeket is csak a tér ezen részéről készített a műszer.

A pontfelhők összekapcsolása érdekében illesztőpontokat kellett elhelyeznünk, ha a pontfelhőkben legalább két közös pontjel van, akkor egyértelműen összekapcsolhatók. A kívülről történő mérés során kapcsoló pontoknak a Leica saját tárcsáit használtuk.

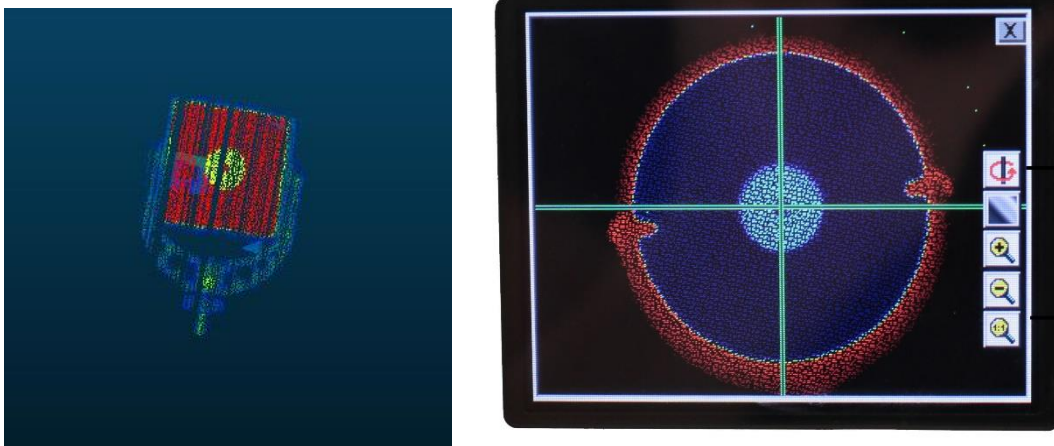


3.2 ábra Leica kapcsoló pontjelölések bal oldalon a három hüvelyk jobb oldalon a hat hüvelyk átmérőjű jel  
(forrás: <https://www.sccsurvey.co.uk/survey-equipment-accessories/targets/scanning-targets.html>)

Kettő darab három hüvelyk, illetve kettő darab hat hüvelyk átmérőjű tárcsa állt rendelkezésünkre. Ezek mágneses talppal vannak ellátva, így könnyen és mozgásmentesen tudtuk elhelyezni őket különböző fém tárgyakon, jellemzően kerítések tartó oszlopain és villámhárító védő burkolatán. A belső mérésekhez a lényegesen költséghatékonyabb papír jeleket használtuk. Ezek mozgásmentes rögzítését ragasztószalaggal biztosítottuk. Az elazonosítás elkerülése érdekében filccel megszámoztuk őket. Három-három darab pontjel került a templom szentélyébe és az altemplomba. Azoknál a méréseknél, ahol ezen jeleket használtuk kapcsolópontnak, ott öt darab fölös adattal rendelkezünk.



A mérés során összesen hét állásponton szkenneltem. Négy álláspontot létesítettem kívül, kettőt a templom szentély részében és egyet az altemplomban. A mérések végeztével a műszer által generált állományokat kiolvastam egy pendrive-ra, így összesen 6.7 GB méretű állományt kaptam. Ezeket az állományokat nem lehet társítani szövegszerkesztőkhöz, vagy egyéb programokhoz, csak a Leica Cyclone képes megnyitni a pontfelhőket és képeket. Én az Óbudai Egyetem Alba Regia Műszaki Kara által írt programot használtam arra, hogy adatokhoz jussak. Ennek a kódnak a használatához hivatkoznunk kell a műszer által létrehozott pontfelhő állományra, aminek ibn kiterjesztése van. Ez a program beolvassa az állományt és a tartalmát kiírja egy csv szöveges fájlba, amit már több alkalmazás is képes kezelni. A létrehozott fájl minden sorában négy adat található, XYZ koordináta és egy intenzitás érték. A műszer a célokat nagy pontsűrűséggel szkennelte be, majd minden célt külön állományban tárolt. A műszer, ezen mérés után, meghatározta a kapcsoló pontjel térbeli középpontját és eltárolta azt egy olyan ini fájlba, amiben még sok egyéb információt találhatunk az álláspontról. Nekem ez a kis szöveges fájl adta a legtöbb segítséget a feldolgozás során.



3.3 ábra Kapcsoló pontjelek pontfelhői 3" és 6" /Bal oldalon az irodai feldolgozás, jobb oldalon a mérés közben -műszeren- látható/  
(forrás: <http://gmv.cast.uark.edu/scanning/hardware/leica-c10/leica-c10-starting-a-scan-2/>)

## 4. Leica Cyclone

A kereskedelmi Leica Cyclone szoftver a Leica cég saját fejlesztésű programja. Nagyon kényelmesen, pár kattintás után, automatikusan elvégezte a közös pontok párosítását, majd a transzformációk kiszámítását. Az így végzett számítás eredményéről egy szöveges fájl lehet megtekinteni, majd elmenteni. Ebből készítettem az alábbi táblázatot:

| Constraints |              |       |   |              |        |         |   |       |   |        |
|-------------|--------------|-------|---|--------------|--------|---------|---|-------|---|--------|
| Constraint  | Type         | Error |   | Error vector |        |         |   | Horz  |   | Vert   |
| t1          | Station-002: | 0.002 | m | (0.001       | 0.001  | -0.001) | m | 0.002 | m | -0.001 |
| t1          | Station-006: | 0.002 | m | (0.001       | 0.001  | -0.001) | m | 0.001 | m | -0.001 |
| t1          | Station-007: | 0.002 | m | (0.001       | 0.001  | 0.001)  | m | 0.002 | m | 0.001  |
| t4          | Station-001: | 0.001 | m | (-0.001      | -0.001 | 0.000)  | m | 0.001 | m | 0.000  |
| t4          | Station-002: | 0.002 | m | (-0.001      | -0.001 | 0.000)  | m | 0.002 | m | 0.000  |
| t4          | Station-007: | 0.002 | m | (-0.001      | -0.001 | 0.000)  | m | 0.002 | m | 0.000  |
| p1          | Station-003: | 0.003 | m | (-0.001      | 0.000  | -0.003) | m | 0.001 | m | -0.003 |
| p1          | Station-008: | 0.001 | m | (0.000       | -0.001 | 0.000)  | m | 0.001 | m | 0.000  |
| p2          | Station-003: | 0.002 | m | (0.001       | 0.001  | 0.002)  | m | 0.001 | m | 0.002  |
| p2          | Station-008: | 0.001 | m | (-0.001      | 0.000  | 0.000)  | m | 0.001 | m | 0.000  |
| p3          | Station-003: | 0.002 | m | (0.000       | -0.001 | 0.002)  | m | 0.001 | m | 0.002  |
| p3          | Station-008: | 0.001 | m | (0.000       | 0.000  | 0.000)  | m | 0.001 | m | 0.000  |
| p4          | Station-005: | 0.002 | m | (-0.001      | -0.001 | -0.001) | m | 0.002 | m | -0.001 |
| p5          | Station-005: | 0.002 | m | (-0.001      | 0.001  | 0.001)  | m | 0.002 | m | 0.001  |
| p6          | Station-005: | 0.002 | m | (0.002       | 0.000  | 0.001)  | m | 0.002 | m | 0.001  |
| t4          | Station-002: | 0.001 | m | (0.000       | 0.000  | 0.001)  | m | 0.000 | m | 0.001  |
| t4          | Station-007: | 0.001 | m | (0.000       | -0.001 | 0.000)  | m | 0.001 | m | 0.000  |
| t3          | Station-002: | 0.001 | m | (0.000       | 0.000  | 0.001)  | m | 0.001 | m | 0.001  |
| t2          | Station-006: | 0.001 | m | (-0.001      | 0.000  | 0.001)  | m | 0.001 | m | 0.001  |
| t2          | Station-007: | 0.002 | m | (0.000       | 0.000  | -0.002) | m | 0.000 | m | -0.002 |
| t1          | Station-006: | 0.001 | m | (0.000       | 0.000  | 0.000)  | m | 0.000 | m | 0.000  |
| t1          | Station-007: | 0.002 | m | (0.000       | 0.000  | 0.002)  | m | 0.000 | m | 0.002  |
| t4          | Station-007: | 0.001 | m | (0.000       | 0.000  | -0.001) | m | 0.000 | m | -0.001 |
| p3          | Station-008: | 0.003 | m | (0.000       | 0.002  | -0.002) | m | 0.002 | m | -0.002 |
| p1          | Station-008: | 0.003 | m | (0.001       | -0.001 | 0.003)  | m | 0.001 | m | 0.003  |
| p2          | Station-008: | 0.002 | m | (-0.001      | -0.001 | -0.002) | m | 0.001 | m | -0.002 |
| t2          | Station-007: | 0.003 | m | (0.000       | 0.000  | -0.003) | m | 0.000 | m | -0.003 |
| t1          | Station-007: | 0.003 | m | (0.000       | 0.001  | 0.002)  | m | 0.001 | m | 0.002  |

4.1 táblázat A Leica Cyclone szoftver által számított transzformációs eredmények maradék ellentmondásai

Tartalmazza a pont és az álláspont azonosítóját, illetve a transzformáció maradék ellentmondásait. Az ellentmondások jellemzésére, a hiba vektor hosszát alkalmazza (melynek jellemző értéke kettő milliméter) és ennek térbeli összetevőit is külön megjeleníti, továbbá külön oszlopban jelöli a hiba vízszintes és magassági mértékét is. Látható, hogy csekély hibák jelentkeztek a számítás során.

Következő lépésben az összeforgatott pontfelhőket egy állományban egyesítettem. Ezután az összegzett pontfelhőből kitöröltem a felesleges pontokat. Végül a képekből nyert színek alapján, kiszíneztem a pontfelhőt.

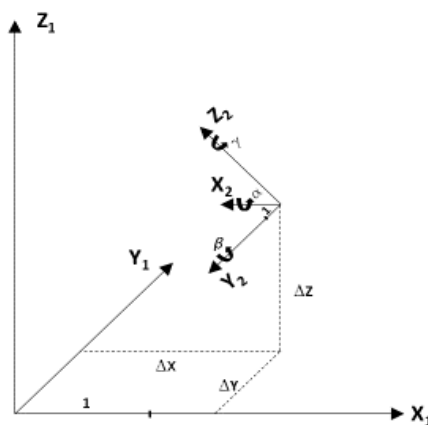
A dolgozatomban ezt, a kereskedelmi forgalomban lévő Leica Cyclone szoftver által készített pontfelhőt vettem a vizsgálatok alapjául, minden pontfelhő-összehasonlításban referenciaként használtam.

## 5. Transzformációk

A műszer által készített pontfelhőt saját test koordináta-rendszerében tárolta el, minden állásponton. Ez hét állásponton hét különböző műszer koordináta-rendszert jelent. Ezek közötti átjárhatóságot a kapcsoló pontok koordinátái biztosítják, mivel több rendszerben is ismertek. Ahhoz, hogy ezeket a pontfelhőket összekapcsoljam, transzformációs paramétereket kellett számítanom.

### 5.1. Térbeli hasonlósági transzformáció

Ennek a transzformációnak rengeteg elnevezését ismerjük, miszerint lehet: hét paraméteres, térbeli Helmert, Bursa-Wolf, klasszikus, illetve térbeli hasonlósági transzformáció. E módszer esetén két olyan térbeli derékszögű koordináta-rendszerrel rendelkezünk, melyeknek origója nem esik egybe, mindhárom tengely körül el vannak fordulva egymáshoz képest és méretarány változással is terhelve vannak.



5.1.1 ábra Hét paraméter ábrázolása:  $\Delta X$ ,  $\Delta Y$ ,  $\Delta Z$ ,  $\alpha$ ,  $\beta$ ,  $\gamma$  és a méretarány tényező

Legalább három közös pont ismeretében, a változások paramétereit (három eltolás, három elfordulás és egy méretarány tényező) számíthatjuk. Ekkor a három pont térbeli koordinátái adják az egyenletek számát, tehát kilenc egyenlettel rendelkezünk. Hét ismeretlen paraméter birtokában ezek kiegyenlítése is lehetséges a legkisebb négyzetek módszerével, két fölös adat alapján.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_u \\ Y_u \\ Z_u \end{bmatrix} + m * \begin{bmatrix} \cos(xX) \cos(yX) \cos(zX) \\ \cos(xY) \cos(yY) \cos(zY) \\ \cos(xZ) \cos(yZ) \cos(zZ) \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

5.1.2. ábra A térbeli Helmert transzformáció általános képlete, ahol:  $x, y, z$  és  $X, Y, Z$  a koordináta-rendszerek jelölése,  $u$  az eltolási értékek jelölése és  $m$  a méretaránytényező

A kiegyenlítés során számítható maradék ellentmondások, a koordináta-rendszerek illeszkedésének pontosságát fejezik ki. Lehetőségünk van a hét paraméter illeszkedésének egy mérőszámon át való kifejezésére a súlyegység középhibáján keresztül, ami a javítások négyzetösszegének és a fölös adatok számának hányadosa a gyök alatt. (Busics 2011)

$$m_0 = \sqrt{\frac{\sum (v_x^2 + v_y^2 + v_z^2)}{3n - 7}}$$

5.1.3 ábra Súlyegység középhibájának számítása

## 5.2. Térbeli egybevágósági transzformáció

Két alakzatot akkor nyilváníthatunk egybevágónak, ha létezik olyan egybevágósági transzformáció, amely a két alakzatot megfelelteti egymásnak. Ilyen transzformációk például a tengelyes tükrözés, eltolás, forgatás, középpontos tükrözés. Tehát elmondható, hogy az olyan transzformációk nevezhetők egybevágóságinak, amelyek esetében egy szakasz képe egy azonosan hosszú szakasz lesz. (Ashburner, Friston 2004)

A dolgozatom elején azt írtam, hogy két pontfelhő összekapcsolásához elegendő két közös pont, amennyivel én rendelkeztem. A háromdimenziós transzformációhoz viszont legalább három közös pont szükséges, ezért a műszer állótengelyét elfogadtam függőlegesnek, amit megtehettem, mert a lézerszkennert állótengelyét minden állásponton gondosan függőlegessé tettem, a nagy érzékenységgű, két tengelyű, 1.5 szögmásodperces digitális kompenzátor segítségével. További feltételezésem volt, hogy a fentebb említett hét pontfelhő, amelyek egyenként eltérő műszer koordináta-rendszerben álltak rendelkezésemre, azonos méretarányúak, mivel a távmérés nem tartalmaz szorzó jellegű hibát. A hét vagy hat paraméteres transzformációk így túl sok paramétert határoztak volna meg számomra, illetve nem is volt három kapcsoló pontom a legtöbb esetben, ezért négy paraméteres transzformációt választottam, amelyben a következő egyszerűsítésekkel éltem: a méretaránytényező értékét egységnyinek vettem fel, illetve  $X$  és  $Y$  tengely körüli forgatást nem számoltam. Így a  $Z$  tengely körüli forgatás paramétere mellé három eltolást kellett számítanom. Erre a célra írtam egy programot a GNU Octave szabad szoftver segítségével. A két közös pontot úgy használtam fel a transzformációhoz, hogy első lépésben az egyik pontfelhőt eltoltam az első közös pontba, majd a másik közös pont irányába fordítottam.

## 6. GNU Octave

A free software kategória egy kiemelkedő alakja a GNU projekt. Alapító atyja Richard M. Stallman 1983-ban indította el a rendszert. Célja az volt, hogy a programozást újra közösségi élménnyé tegye, megadja mindenkinek a lehetőséget a fejlesztésben és megadja a szabadságot, amit az ingyen elérhető szoftverek kínálnak. „Nem 95%-ban és nem is 99.5%-ban, hanem 100%-ban ingyenes” rendszert tervezett létrehozni, amit véghez is vitt. Az alap elképzelése az volt, hogy létre hoz egy rendszert, ami „olyan, mint a Unix, de mégsem olyan”. Ezt azzal magyarázta, hogy a rendszere gyakorlatban hasonlítani fog a Unix-hoz, de emellett megadja felhasználóinak a szabadságot. Az elnevezés is innen származik azaz: **GNU's Not Unix**. (<https://www.gnu.org/gnu/about-gnu.html>)

Olyan rendszert sikerült elindítania, amely számos polgári kezdeményezést valós szoftverre tudott alakítani, mivel, ha valaki érdeklődik a háttérben zajló folyamatok iránt, a nyílt forráskódot bármikor megtekintheti. Ennek hozományaként született a GNU Octave program, ami a MatLab alternatívája ingyenes programként. Olyan szkripteket tudunk vele írni, amit mindkét alkalmazás tud kezelni. Itt megemlítenék még egy közösség által fejlesztett alkalmazást, ami pedig pontfelhők kezelésére képes magas szinten, a CloudCompare-t.

Tehát a paraméterváltozások számítását az Octave nevű programmal végeztem. A transzformáció paraméterezéséhez a következő képleteket alkalmaztam, ami négy paraméterre vonatkozik. Az X és Y tengely körüli forgatás értékét nullának vettem fel.

$$\mathbf{M} = \mathbf{TR} = \begin{bmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(q_6) & \sin(q_6) & 0 & 0 \\ -\sin(q_6) & \cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6.1 ábra Az alkalmazott képlet, ahol  $q_1, q_2, q_3$  az eltolási paramétereket,  $q_6$  az Z tengely körüli elforgatási paramétert jelenti (Ashburner, Friston 2004)

Az egyszerűbb programozás érdekében nem alkalmaztam szögfüggvényeket, azokat lineáris kapcsolattal helyettesítettem. Ennek fényében született a program, aminek eredménye a paraméterváltozások értékei lettek.

```

*4param.m
1  clc; clear all; format long
2  %jelölés: t=tárcsa p=papír h=helyi rendszer
3  %      tij ahol i illesztőpont száma és j az álláspont száma
4  %t4 tárcsa 1 álláspont= t41
5
6  %kapcsoló pontok koordinátái lézerszkennер testkoordináta-rendszerében
7  t41=[-14.422881126,-13.450659752,-0.189214870];
8  t31=[-25.016187668,-11.091095924,-0.347449154];
9  t21=[16.606405258,-15.496079445,-1.938332319];
10
11 t42=[5.796263218,6.644602776,-0.374661893];
12 t32=[11.852534294,-2.360752106,-0.532548606];
13 t12=[14.879729271,12.662933350,-1.994816780];
14
15 %transzformációs paraméterek számítása
16 %l álláspont transzformálása a 2-be
17 A12=[t41(1),t41(2),1,0,0;
18      t41(2),-t41(1),0,1,0;
19      0,0,0,0,1;
20      t31(1),t31(2),1,0,0;
21      t31(2),-t31(1),0,1,0;
22      0,0,0,0,1]
23 l12=[-t42(1);
24      -t42(2);
25      t41(3)-t42(3);
26      -t32(1);
27      -t32(2);
28      t31(3)-t32(3)]
29 x12=-pinv(A12'*A12)*A12'*l12

fileID = fopen('oparam.dat','w');
fprintf(fileID,'x12=\n');
fprintf(fileID,'%f\n',x12);
fprintf(fileID,'\nx67=\n');
fprintf(fileID,'%f\n',x67);
fprintf(fileID,'\nx72=\n');
fprintf(fileID,'%f\n',x72);
fprintf(fileID,'\nx2h=\n');
fprintf(fileID,'%f\n',x2h);
fprintf(fileID,'\nx3h=\n');
fprintf(fileID,'%f\n',x3h);
fprintf(fileID,'\nx8h=\n');
fprintf(fileID,'%f\n',x8h);
fprintf(fileID,'\nx5h=\n');
fprintf(fileID,'%f\n',x5h);
fclose(fileID);

```

6.2 ábra Az általam írt GNU Octave program a paraméterek változásának kiszámítására

A program írása során elő kellett készíteni a közös pontok koordinátáit, úgy, hogy egyértelműen el lehessen különíteni az álláspontokat. Ezután minden egyes transzformációhoz fel kellett építenem két mátrixot. Először az alakmátrixot (A), melynek mindig öt oszlopa volt, illetve három sora közös pontonként, melyek száma általában kettő vagy három, így a mátrix hat vagy kilenc sorból épült fel. Másodszor pedig a tisztatag vektor (l) számítása következett. Oszlop vektor formájában, pontonként három sorral nő a tisztatag vektor mérete, melyből az első két elem a cél koordináta-rendszerbeli pont X és Y koordinátája, harmadik eleme pedig Z koordináták különbsége. Ezen két mátrix ismeretében a kiegyenlítés szabályai szerint számítható a paraméterek változása, melyet x oszlopvektorban tárol a program.

|                    | x12       | x67      | x72      | x2h       | x3h       | x8h       | x5h       |
|--------------------|-----------|----------|----------|-----------|-----------|-----------|-----------|
| Eltolás X [m]      | -0,72509  | 0,96698  | -0,94464 | -0,98556  | 0,03291   | 0,94283   | -0,74795  |
| Eltolás Y [m]      | -0,68859  | 0,25495  | -0,32824 | 0,17103   | -0,99964  | -0,33332  | 0,66281   |
| Eltolás Z [m]      | -13,92358 | 25,05177 | 15,91948 | 491,13114 | 498,67935 | 494,92660 | 497,66635 |
| Elforgatás Z [ ° ] | 6,82322   | -1,28645 | 8,93430  | 210,77402 | 200,11555 | 200,06662 | 200,06768 |
|                    | -0,18527  | -0,16854 | -0,62275 | 11,18324  | 13,10344  | 11,21901  | 8,83639   |

6.1 táblázat A paraméterváltozások eredménye

A fenti ábrán az általam írt program végeredményéből készített táblázat található. Az Octave kiírt egy szöveges fájlt, ami az x vektorokat tartalmazta. Ennek első négy sora volt hasznos számomra. Az első három az eltolási értékeket mutatta, a negyedik sor pedig a Z tengely körüli elfordulás mértékét.

## 7. CloudCompare

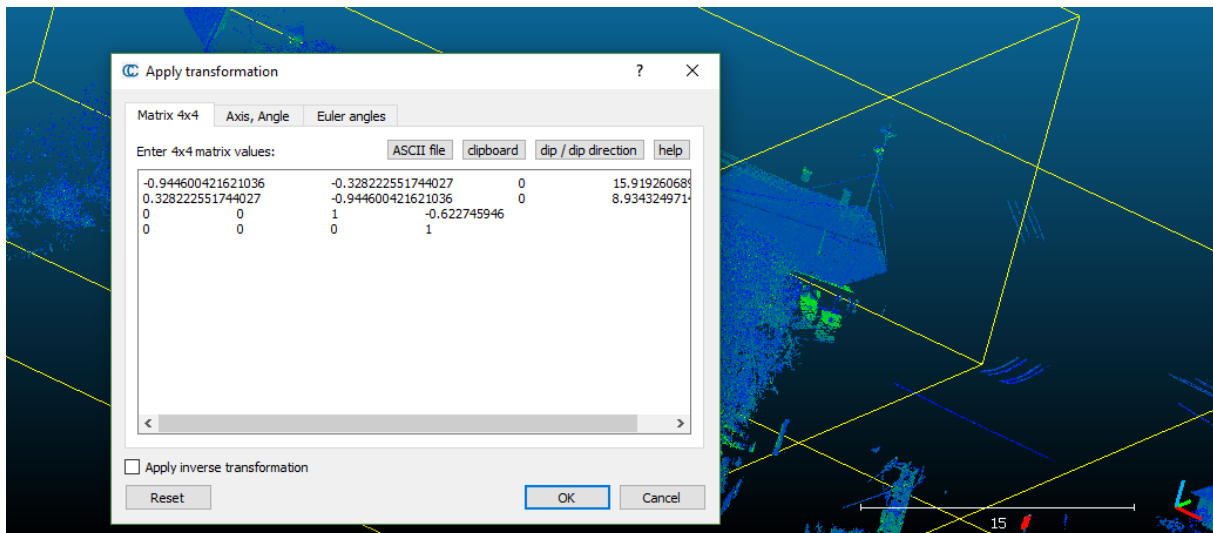
Ezután következett a pontfelhők transzformációja, melyhez a CloudCompare szoftvert alkalmaztam. Bátran állíthatjuk, hogy ez a legismertebb és legnépszerűbb pontfelhő kezelő szabadsoftver a felhasználók körében. Olyan, közösség által fejlesztett megjelenítő és szerkesztő programról van szó, mely számtalan szakterület igényeit kielégíti. Magas színvonalon lehet vele dolgozni, kompromisszumok nélkül, ezért döntöttem én is emellett a szoftver mellett a feldolgozás során. A következőkben tehát bemutatom a pontfelhő feldolgozás és minősítés mentét.

### 7.1. Pontfelhők transzformálása

Én nyers, helyi rendszerben lévő pontfelhőkkel rendelkezem, ezért számomra a szoftver egyik legfontosabb tulajdonsága, hogy képes egy vagy több kijelölt pontfelhőt térbeli hasonlósági transzformációval a modell tér bármely pontjába eltolni és forgatni a megadott paraméterek alapján. Ezeket az adatokat 3 módon adhatjuk meg:

- 4x4 mátrix segítségével,
- forgatási tengely, szög és eltolás megadásával,
- Euler szögek segítségével.

Számomra a legszimpatikusabb az első lehetőség volt. Összegyűjtöttem és felírtam az összes transzformáció eredményeként kapott paraméter változásokat mátrixos alakban, egy szöveges állományba. Amikor aztán szükségem volt valamelyikre egyszerűen be kellett másolnom a szövegszerkesztő mezőbe.

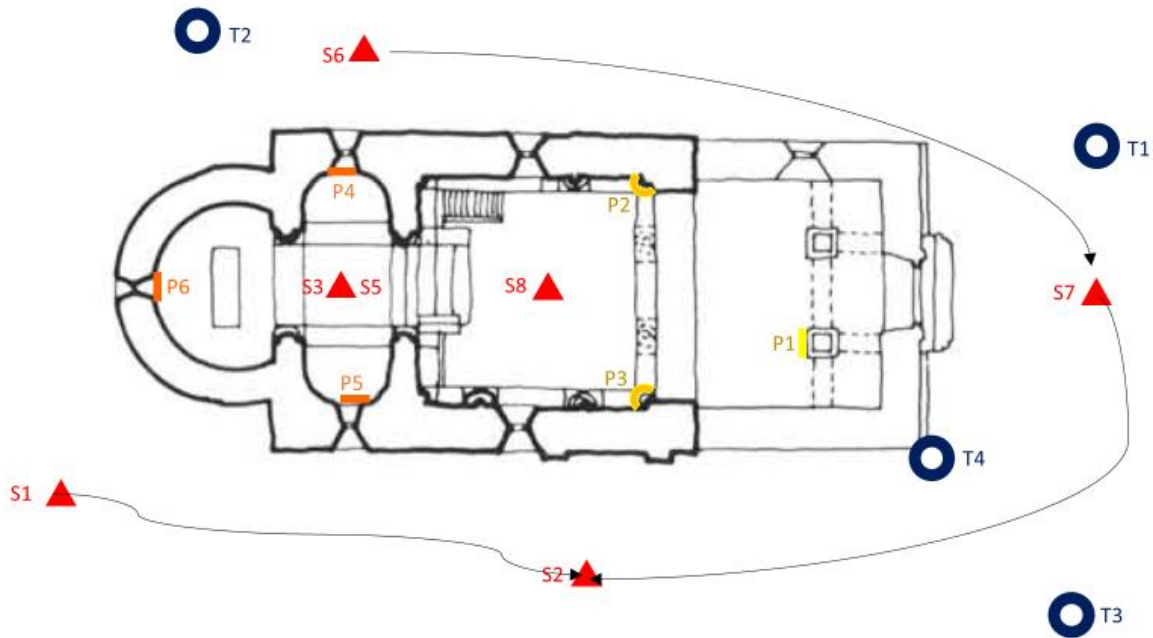


7.1.1 ábra Transzformációs paraméterek megadása 4x4 mátrix segítségével

A feldolgozás következő megoldásra váró problémája, amivel szembe találtam magam, hogy a templomon kívüli, és a templomon belüli pontfelhőket nem kapcsoltam össze közös pontokkal, sőt a szentély és az altemplom sem volt összekapcsolva. Ezt a nehézség úgy került feloldásra, hogy minden kapcsolópont koordinátáját polárisan bemértük, így a végén minden pontfelhőt egy közös, helyi rendszerbe transzformálhattam át. Ezen helyi rendszer kialakításában segített Tóth Sándor József évfolyamtársam is, aki szkenneléssel egyidőben saját kutatása céljából mérte a templomot mérőállomásos technológiával. Mikor éppen a részletmérést végezte, meghatározta az én kapcsoló pontjaimnak a koordinátáit is az ő saját, helyi rendszerében. Meghatározta mind a hat papír jelet, viszont a négy tárcsából sajnos csak kettőt. A kapcsolópontok java csak poláris pontként lett meghatározva, ezért nem szerepelt a hálózat kiegyenlítésben. Maga a hálózat tájolása úgy lett kialakítva, hogy az Y tengely a templom hossz tengelyével párhuzamos és az 500, 200 koordinátapár a körszentély középpontjába esik.

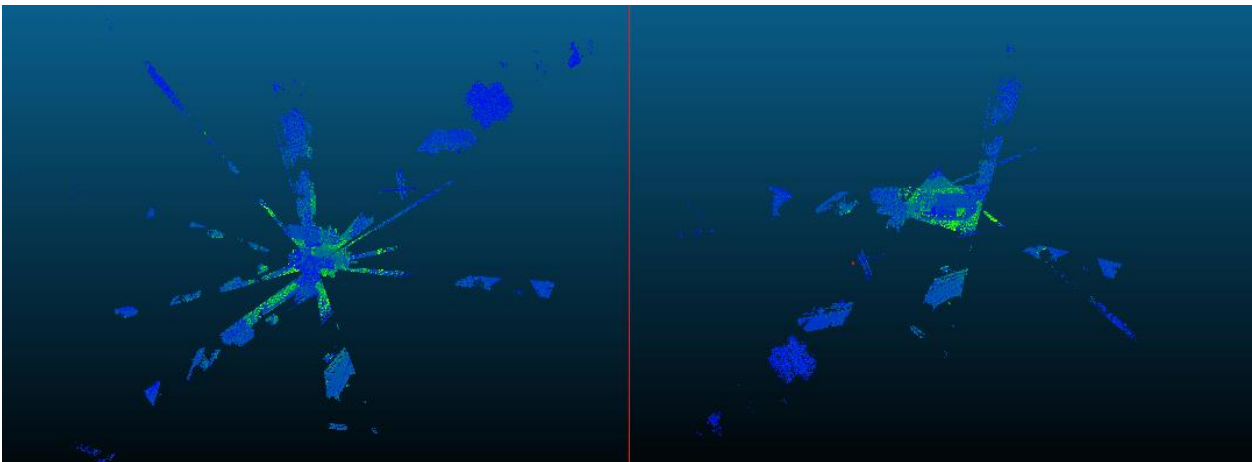
Nem minden kapcsolópontnak mértük a helyi rendszerbeli koordinátáját, ezért nem jöhetett számításba az a lehetőség, hogy minden egyes műszerállás pontfelhőjét egyenként a helyi rendszerbe transzformáljam és így előálljon a végleges pontfelhő. A belső álláspontokkal nem volt probléma, ezeket egyből lehetett a végleges helyükre transzformálni. Ezzel ellentétben a külső álláspontok gondot okoztak, mivel csak a második állásponton mértem olyan kapcsolópontokra amelyeknek meglett határozva a helyi rendszerbeli koordinátája, a többi három állásponton nem.





7.1.2 ábra A külső pontfelhők transzformálásának irányai, ahol S az álláspontokat, T a tárcsákat és P a papír pontokat jelöli

Azt a megoldást találtam ki, hogy az összes külső pontfelhőt ebbe a második álláspontba transzformáltam. Az első és hetedik álláspont pontfelhőjét közvetlenül tudtam a második álláspontba transzformálni, viszont, hogy a hatodik álláspont pontfelhőjével is elvégezhessem, két lépésben kellett transzformálnom, először a hetedik álláspontba, majd a kettőt együtt a második álláspontba. Ekkor az összes külső pontfelhőt kijelöltem, eltoltam és forgattam a helyi rendszerbe. Majd, amikor a belső tér pontfelhőit a helyi rendszerbe transzformáltam, egyből a templom belsejébe helyezkedtek el.

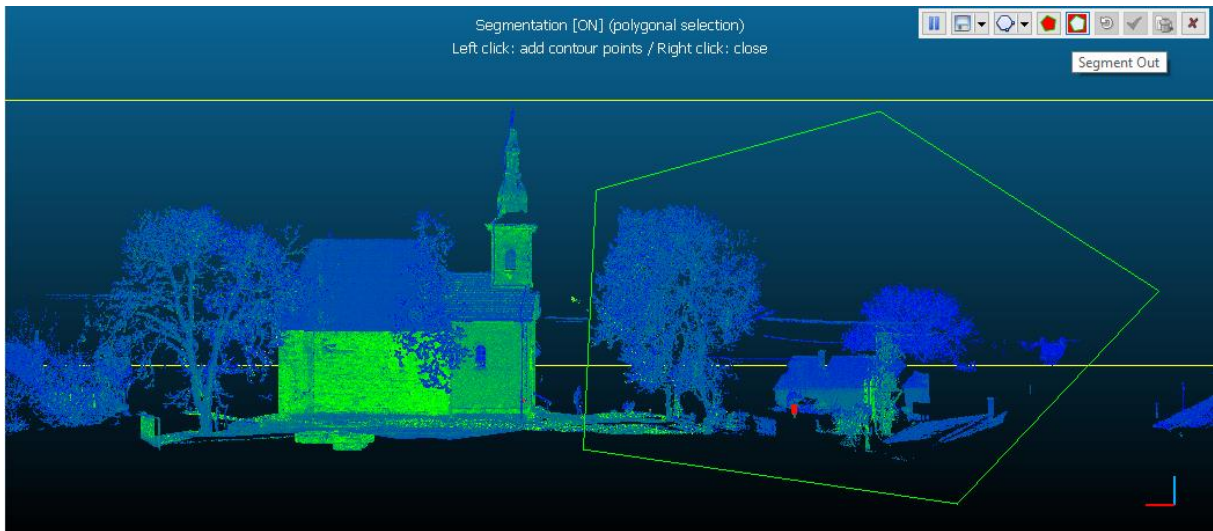


7.1.3 ábra A pontfelhők felülnézetben a transzformációk előtt és után

A fenti ábrán látható, hogy a pontfelhő sikeresen összeállt. Az állományban összesen 77.7 millió pont szerepelt, melynek egy része felesleges volt számomra. Ez a zaj az ablakos mérésekből adódott, mivel túl nagy szkennelési tartományt adtam meg. Ezt biztonsági okokból tettem, nehogy kimaradjon a templom egy kilógó részlete. A modellen így zajként jelentek meg a beszkenelt fák, épületek és emberi alakok.

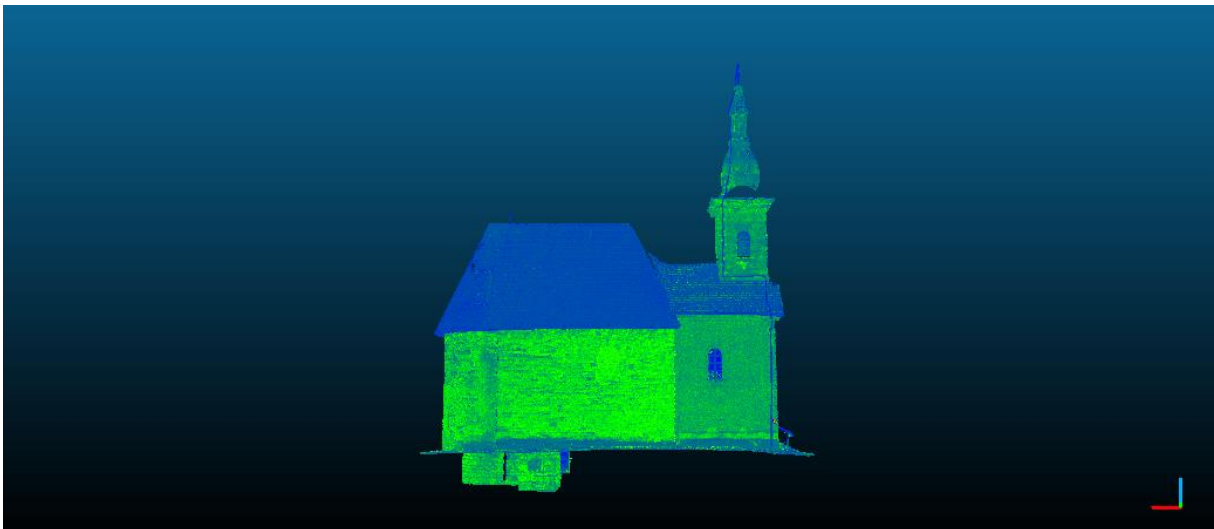
## 8. Mérési zaj eltávolítása

Erre a munkára a szoftver segment parancsát használtam, ami egy olyan funkciót biztosít, amellyel kijelölhető egy tetszőleges tartomány a pontfelhőben (sokszöggel, négyszög ablakkal), majd eldönthetjük, hogy csak a kijelölt pontokat megtartsuk, vagy letöröljük. Az alábbi képen látható, az a terület, amelyből a segment out paranccsal töröltem a pontokat.



8.1 ábra Szegmentálás, a számomra haszontalan pontok eltávolítása

A területek kijelölése sok időt emésztett fel, mert úgy kellett forgatni a pontfelhő nézetét, hogy a mérési zaj kijelölése egyértelmű legyen, ügyelnem kellett, hogy ne töröljek bele véletlenül a templom modelljébe. A templom melletti talaj részből meghagytam egy keskeny sávot, ami segít szemléltetni, hogy az altemplom milyen mértékben nyúlik a föld alá.

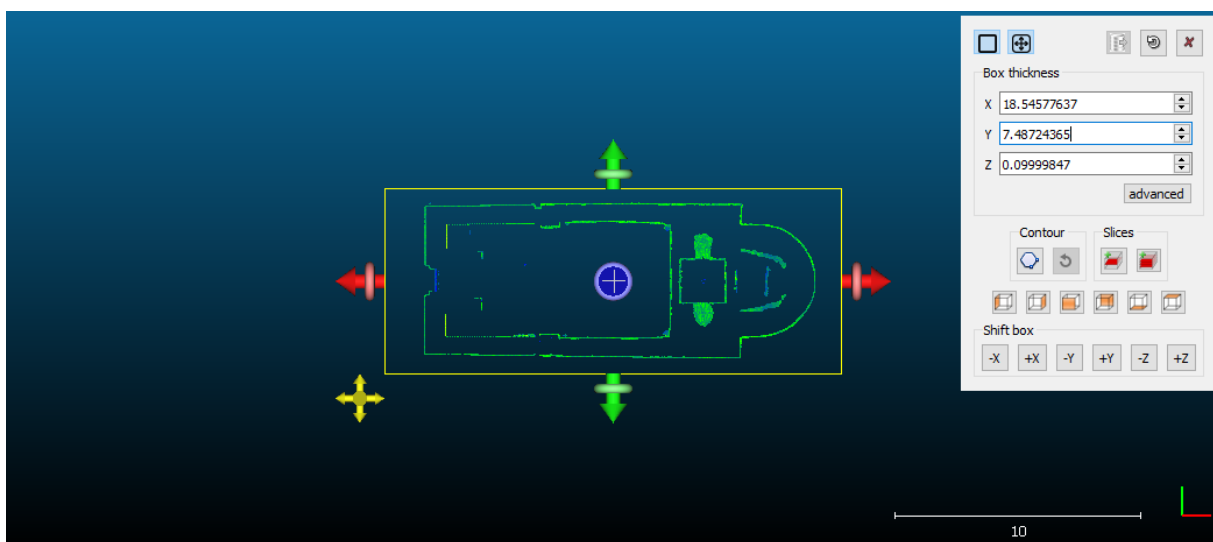


8.2 ábra A mérési zajtól megtisztított pontfelhő

A szegmentálási műveletek után mindössze 53.7 millió pont maradt az állományban, azaz közel 24 millió pontot töröltem le. Meg kell jegyezni, hogy így nem csak egzaktabb, de esztétikusabb is lett a modell, sőt jelentős számítási kapacitást is megtakarított a számítógépnek, ami azért volt hasznos, mert egy hétköznapi felhasználásra szánt konfiguráció véleményem szerint nem rendelkezik ehhez a feladathoz elegendő teljesítménnyel. A metszetek elkészítése során, egy átlagosnál erősebb hardverrel rendelkező eszközt kellett használnom.

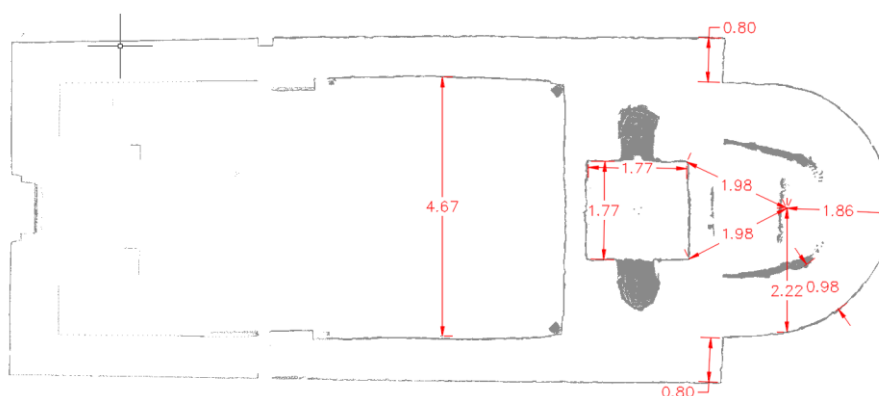
## 9. Metszetek készítése

Metszetek előállítására a Cross Section parancsot használtam, mely arra szolgál, hogy egy adott pontfelhő egy részét kijelölve elmentsük annak pontjait. Az XY tengely által meghatározott síkkal metszettem el a pontfelhőt. Tíz centiméter vastag pontfelhőt mentettem ki, annak érdekében, hogy felülnézetben ne legyen szakadás a falak vonalában.



9.1 ábra Metszetek készítése

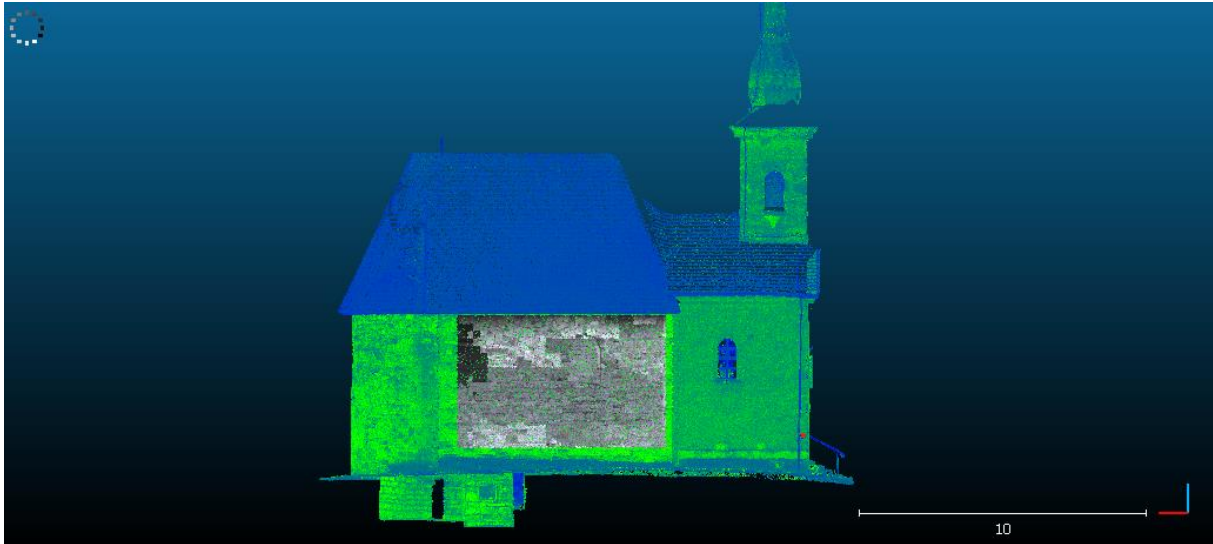
Ezt a részleges pontfelhőt arra használtam fel, hogy alaprajzot készítsék a templomról. Igaz már létezett ilyen alaprajz, de az még a XX. század második felében készült. Az alaprajzot elláttam méretekkkel. A feliratozott méretekből kimutatható, hogy milyen pontossággal mérték fel a templomot az előző alaprajz készítői.



9.2 ábra A körtemplom metszetének méretezése [m]

## 10. Lézerszkenneres pontfelhők vizsgálata

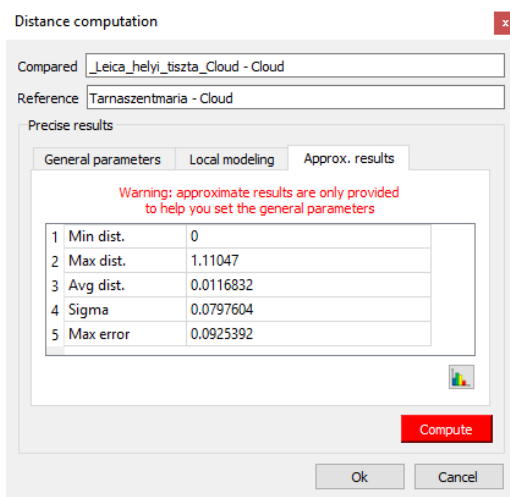
A vizsgálat két pontfelhő összehasonlításán alapszik. Ezt nehezen lehetett elvégezni az egész modell figyelembevételével, mivel a modellek nem egyeztek meg. Az eltérés abban mutatkozott meg, hogy a talajszint kiterjedése nem egyenlő mértékben szerepelt a pontfelhők pontjai között. Ezen rontó hatás kiküszöbölésére a templom északi falából kivágtam egy részt, ami a vizsgálatok alapját képezte. Mérete megközelítőleg 720 cm x 460 cm.



10.1 ábra A kiválasztott falrészlet

A vizsgálatot a két pontfelhő egymáshoz viszonyított távolsága alapján végeztem el. Erre a számításra a CloudCompare Compute cloud/cloud distance beépített modulját használtam.

### 10.1 Felhő-felhő távolság



10.1.1 ábra Előzetes eredmények

A CloudCompare beépített funkciója, hogy két kiválasztott pontfelhő pontjai között távolságot számol, melyekből statisztikát készít és ezt a pontfelhők átlagos távolságával és szórásával fejezi ki. A modul használatához meg kell adni egy vizsgálati (Compared) és egy referencia (Reference) pontfelhőt. A távolság, relatívan a második pontfelhőhöz képest kerül számításra. Különös figyelmet kell fordítani arra, hogy a referencia pontfelhő mérete ne legyen kisebb, mint a vizsgált pontfelhőé. Én pontosan ugyanakkora méretű pontfelhőkkel dolgoztam. A pontfelhők megadása után a program kiszámolja az eredményeket. Vigyázni kell, hogy a párbeszédablakban megjelent eredményeket nem szabad felhasználni a pontfelhő minősítésére. Ezek az

adatok, melyek a nyolcfelé ágazó fa (octree) beállításához alkalmazhatók, számunkra ez esetben irrelevánsok.

Az előzetes eredmények fülön tehát megtekinthető a minimális, maximális átlagos távolság, illetve a szigma érték és a maximális hiba. Ahhoz, hogy ezen előzetes értékek segítségével minősítéshez alkalmas információkat kapjunk, el kell indítanunk a valós számítást a Compute parancs segítségével. Lefutása után az eredményt, a parancssor (Console) ablakban közli a program. Kiolvasható a pontfelhők átlagos távolsága és a szórást. A Leica Cyclone kereskedelmi szoftver által létrehozott, és az általam alkalmazott szabad szoftverek alapján készült pontfelhők átlagos távolsága és szórása:

Mean distance = 0.0006m / std deviation = 0.0007m.

Az átlagos távolság 0.6mm, és a szórás pedig 0.7mm. Ezen értékek alapján megállapítható, hogy a két feldolgozási módszer által kapott pontfelhők különbsége elenyésző. A kereskedelmi szoftver könnyen, gyorsan használható, de ára magas. A szabad szoftverek módszere, viszont meglehetősen költséghatékony, ellenben a feldolgozásra több időt kell szánni. A későbbiekben ezen módszerek automatizálását elvégezve, csökkenthető a feldolgozásra szánt idő.

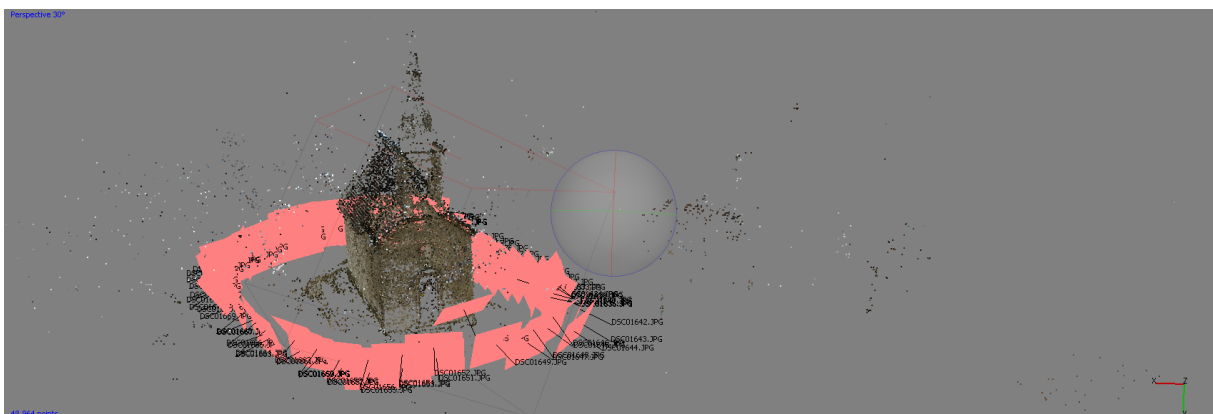
## 11. Földi fotogrammetria

A földi fotogrammetria leggyakrabban a mérnök geodéziában használatos, ide tartozik a külszíni bányák kubatúra számítása, alagutak, barlangok, földalatti hengeres szelvények felmérése és még számtalan szakág által való hasznosítás. (Engler, 2011)

Nem rendelkezem mérőkamerával, ezért amatőr felvételeket készítettem. A templomot körbe fényképeztem kétfajta amatőr kamerával, tekintettel arra, hogy körülbelül 50%-os átfedések legyenek a képekben. A képekből pontfelhőt Agisoft PhotoScan kereskedelmi szoftverrel állítottam elő. A cég 2006-ban alakult, szoftverére jellemző a nagy automatizáltság, aminek köszönhetően a képek feldolgozása nem jelentett nehézséget.

### 11.1 Pontfelhő építése amatőr felvételekből

A PhotoScan egy varázsló segítségével végig vezet a képek előfeldolgozásának lépésein. Első lépésként össze kell kapcsolni a pontokat a párosítás (Align) funkció segítségével. Ez a modul automatikusan megkeresi azokat pontokat, amelyek legalább két képen egyértelműen azonosíthatók és felhasználja őket közös pontként. Ezen pontpárokat felhasználva ki tudja számítani a kamera térbeli helyzetét és a kameratengely irányát az expozíciós pillanatokban. A modelltérből el kell távolítani a számomra felesleges, vagy hibásan észlelt közöspontokat.



11.1.1 ábra Align Points eredménye

Következő lépés már a pontfelhő építése (Build Dense Cloud) volt, ahol a pontfelhő sűrítés típusát magasra (High) állítottam. Ezt több órán keresztül számolta a program, minek eredményeként megkaptam az elkészült, színhelyes pontfelhőmet, amit végül pts formátumban mentettem el. Ezt a folyamatot mindkét kamera képeivel elvégeztem, és ettől fogva már csak CloudCompare program segítségével dolgoztam a pontfelhőn.



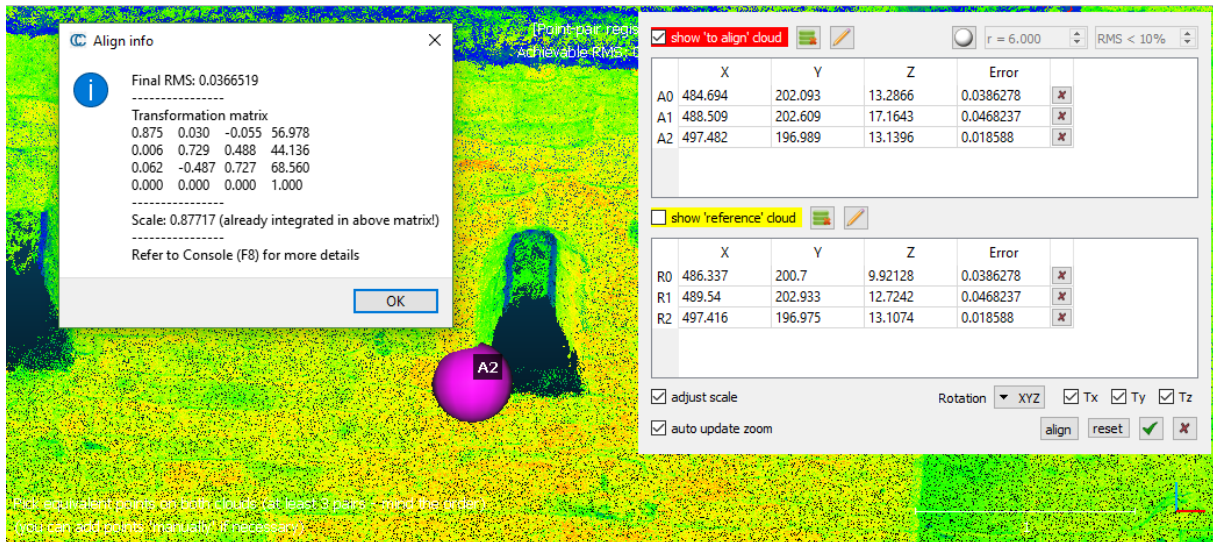
11.1.2 ábra Agisoft PhotoScan által készített pontfelhő

A szoftver még rengeteg lehetőséget kínált volna fel a további feldolgozások végzésére, de ennél a fázisnál befejeztem a PhotoScan szoftver használatát.

## 11.2 Pontfelhők durva illesztése

A földi fotogrammetriából származó pontfelhők még nem voltak tájékozva a modell térben. CloudCompare segítségével hét paraméteres transzformációt kellett rajta alkalmazni, mivel a modell három elfordulás, három eltolás és egy méretaránytényező változással volt terhelve. A felvételek készítése során nem helyeztünk el fotogrammetriai kapcsolópontokat, ezért a szoftver kínálta lehetőséget használtam ki. Ez a funkció volt az Align, melynek segítségével, legalább három közös pont alapján, két pontfelhő összeilleszthető.

A funkció használatához két pontfelhő kijelölésére volt szükség. Ki kell választani, hogy melyik az illesztendő (to align) és melyikhez képest kell illeszteni (reference). Mindkét földi fotogrammetriából származó pontfelhő esetében a Leica Cyclone által előállított pontfelhőt választottam ki referenciának. A közös pontok megadására két lehetőség kínálkozott, bevihettük a közös pontok koordinátáit manuálisan, vagy kijelölhettük ezeket a pontokat kurzor segítségével. Mivel nem rendelkeztem a közös pontokkal, a második módot használtam, mely során, a könnyebb munkavégzés elősegítésére, a két pontfelhő külön-külön kikapcsolható, valamint a modell térben szabadon forgatható és nagyítható. Jól azonosítható pontokat kerestem a pontfelhőkön, melyek jellemzően nyílászárók sarkai voltak. A párbeszéd ablakban a méretarány tényező változtatásának lehetőségét külön engedélyezni kellett. Nemcsak arra kellett nagy figyelmet fordítani, hogy a közös pontok mindkét modellen ugyanazon pontokon helyezkedjenek el, hanem arra is, hogy a megfelelő sorrendben vegyük fel a pontokat.

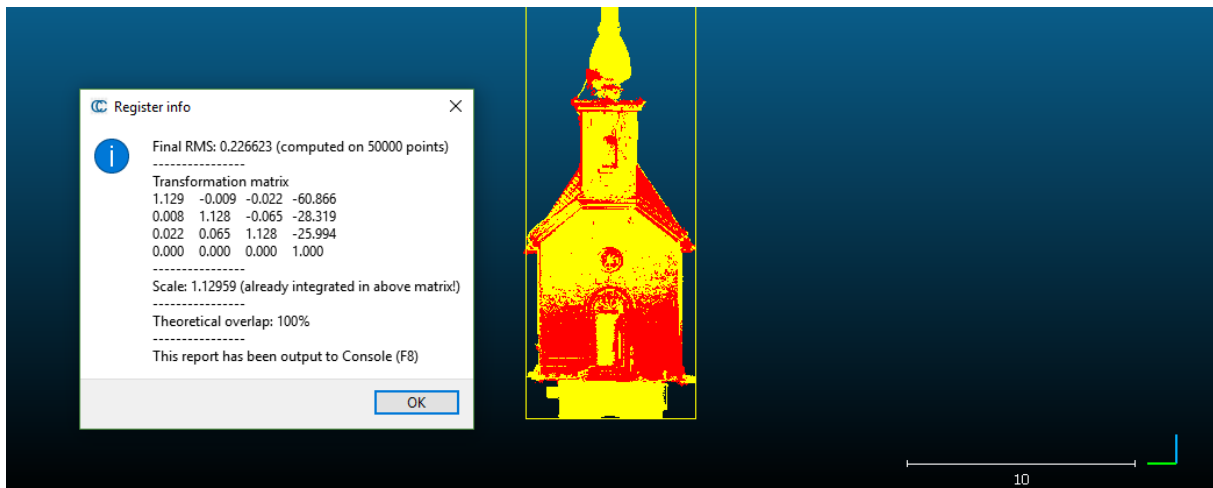


11.2.1 ábra Transzformáció számításának eredménye

Amint a pontokat felvettük, a program a transzformálni kívánt pontfelhőt áthelyezi a kívánt helyre. Ekkor már durván illeszkednek a pontfelhők. Érdeemes a műveletet még párszor végrehajtani, így iterálva a legpontosabb megoldás felé. A legnagyobb gondossággal végezve a közös pontok kijelölésének műveletét sem lehetett tökéletes illeszkedést elérni. A két pontfelhő közel fedő helyzetbe került, áttérhettem az ICP (Iterative Closest Point) algoritmusra.

### 11.3 ICP

A módszer képes, a közelítőleg illesztett pontfelhők finom összehangolására automatikusan, az összes pont bevonásával. Két feltétel teljesülésével működik a módszer: a pontfelhők legyenek durván illesztve és ugyanazt az objektumot ábrázolják (legalább részlegesen). A már ismert kategóriákba kellett besorolni az állományokat. Referencia természetesen itt is a Leica Cyclone pontfelhő volt, a transzformálni kívánt pedig a földi fotogrammetria pontfelhője. A párbeszéd ablakban beállítható volt, hogy bizonyos számú iteráció, vagy bizonyos értéknél kisebb négyzetes középhiba esetén érjen véget a pontfelhők illesztése. Itt is bekapcsolható a méretaránytényező korrigálása.



11.3.1 ábra Iterative Closest Point eredménye

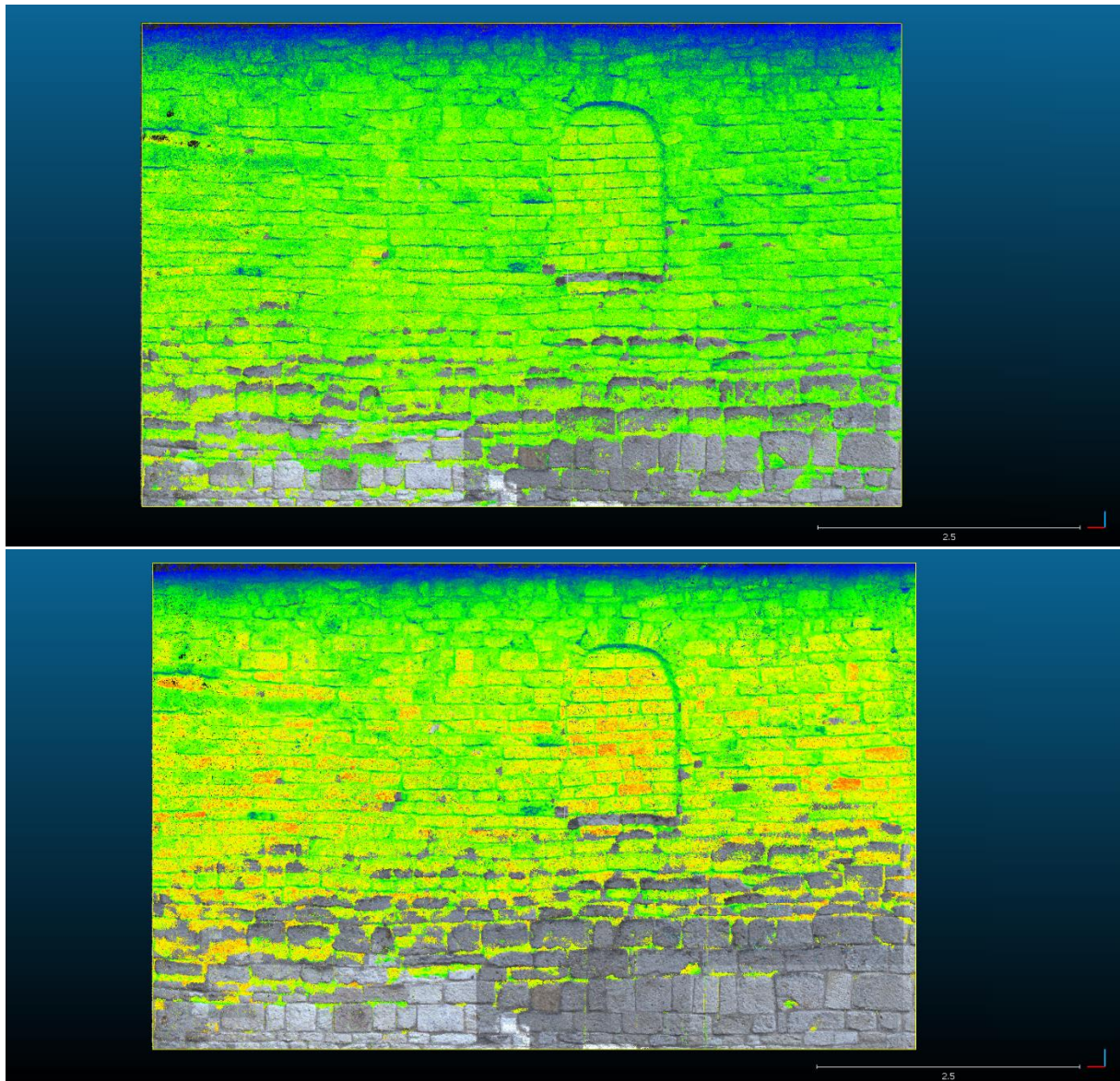
A fenti ábrán látható, a transzformációs mátrix, a méretaránytényező értéke és az elméleti átfedés értéke (100%), illetve a referencia pontfelhő (piros színnel) és az illesztett pontfelhő (sárga színnel). Az átfedő pontfelhőket elmentettem, hogy a vizsgálatokat ezeken végezhessem el.

## 12. Fotogrammetriai pontfelhők pontossági vizsgálata:

A földi fotogrammetria pontfelhőinek pontossági minősítésének alapjául szintén a Leica Cyclone szoftver által számolt pontfelhőből kimetszett fal részletet használtam, mint referenciát, amit a következő ábrákon valódi színekkel jelöltem. A vizsgálati pontfelhők különböző intenzitás értékeit színskálával jelöltem (kék→ zöld→ sárga→ piros), az egyértelműbb szemléltetés végett.

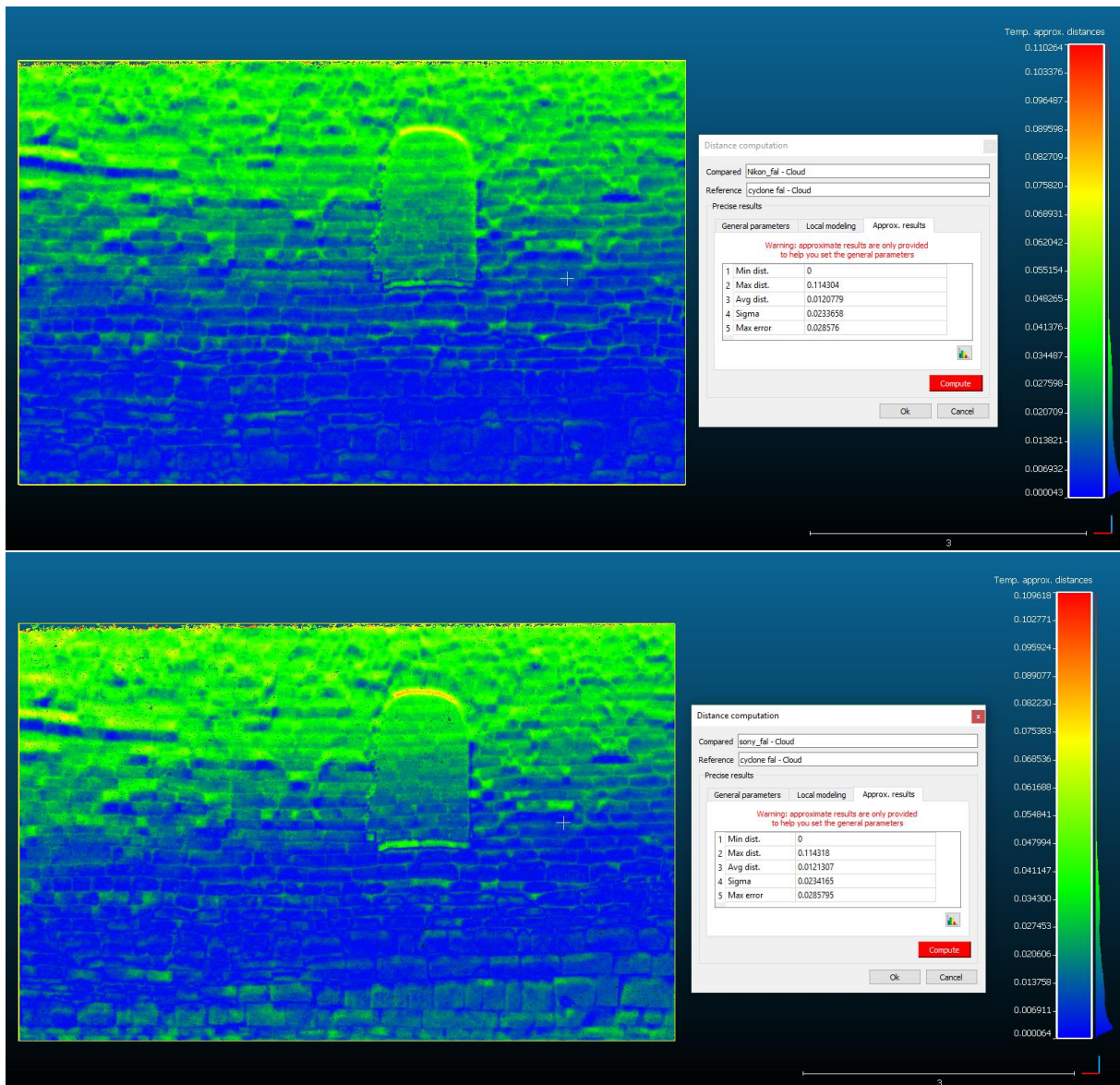
Egyszerre megjelenítve a két pontfelhőt már látszik, hogy hol van eltérés. Ideális esetben, ha a két pontfelhő tökéletesen fedné egymást a színek sakktabla szerűen jelennének meg, helyette a képeken jól látható, hogy a vizsgált pontfelhők alsó része a referencia mögött, míg a felső része a referencia előtt helyezkedik el. Ez a fajta dőlés a földi fotogrammetria szabályos hibáinak tudható be, legnagyobb részt az elrajzolásnak. Kamera kalibráció figyelembe vételével ez a hatás csökkenthető, kiejthető.





12.1 ábra Felül a Nikon, alul a Sony kamera képeiből készített pontfelhő és a referencia fedése

Az egymásra helyezett pontfelhőket a már ismertetett, pontfelhők távolsága modul (Compute Cloud/ Cloud distance) segítségével vizsgáltam. A vizsgálat végén a referencia állomány láthatóságát automatikusan kikapcsolja a szoftver, és a vizsgált pontfelhőt pedig kiszínezi úgy, hogy a referenciától való távolságokat színekkel skálázza. Ez esetben a jellemző színek a kék és a zöld. Ideális esetben az egész felhő homogéneen sötétkék lenne, viszont a hibákból adódóan megjelennek a zöld színnel jelölt távolságok is, bár ezek még mindig csekélynek nevezhetők.



12.2 ábra A pontfelhők színezése a referenciától való távolság alapján

### 13. Összegzés

A vizsgálatok során a földi lézerszkenneres mérési állományok szabad szoftver és kereskedelmi szoftver alapján történő feldolgozásának pontosságát hasonlítottam össze. A földi fotogrammetria esetén az amatőr kamerával készített képek alapján számított, és a földi lézerszkenneres mérésekből származó pontfelhő illesztés utáni távolságát vizsgáltam. Az eredményeket az alábbi táblázat tartalmazza:

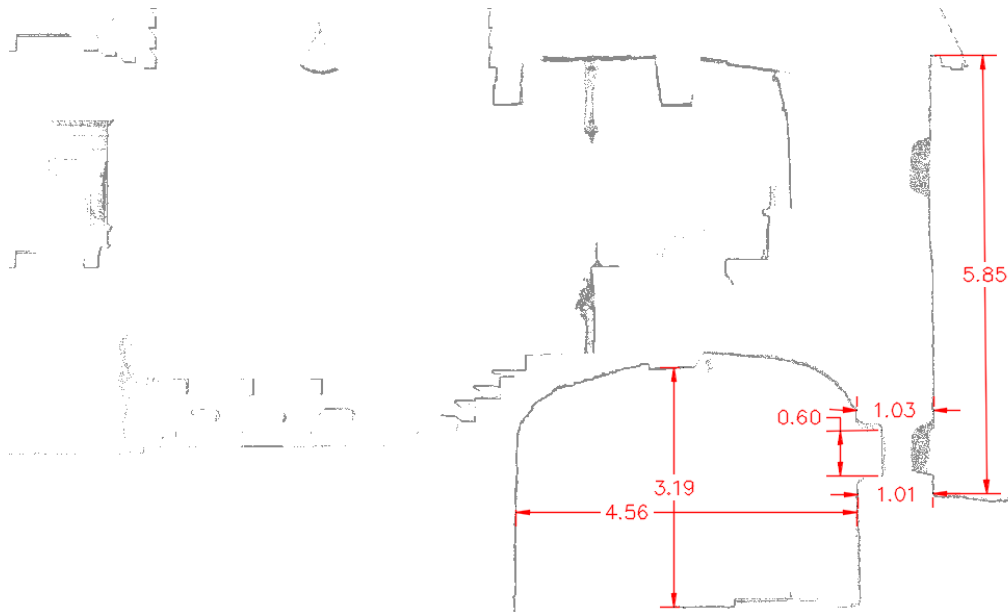
| Felmérés típusa                   | Átlagos távolság [mm] | Szórás [mm] |
|-----------------------------------|-----------------------|-------------|
| Lézerszkennér (szabad szoftverek) | 0.6                   | 0.7         |
| Földi fotogrammetria (Nikon)      | 5                     | 4           |
| Földi fotogrammetria (Sony)       | 6                     | 5           |

13.1 táblázat A pontfelhők vizsgálati eredményeinek táblázata

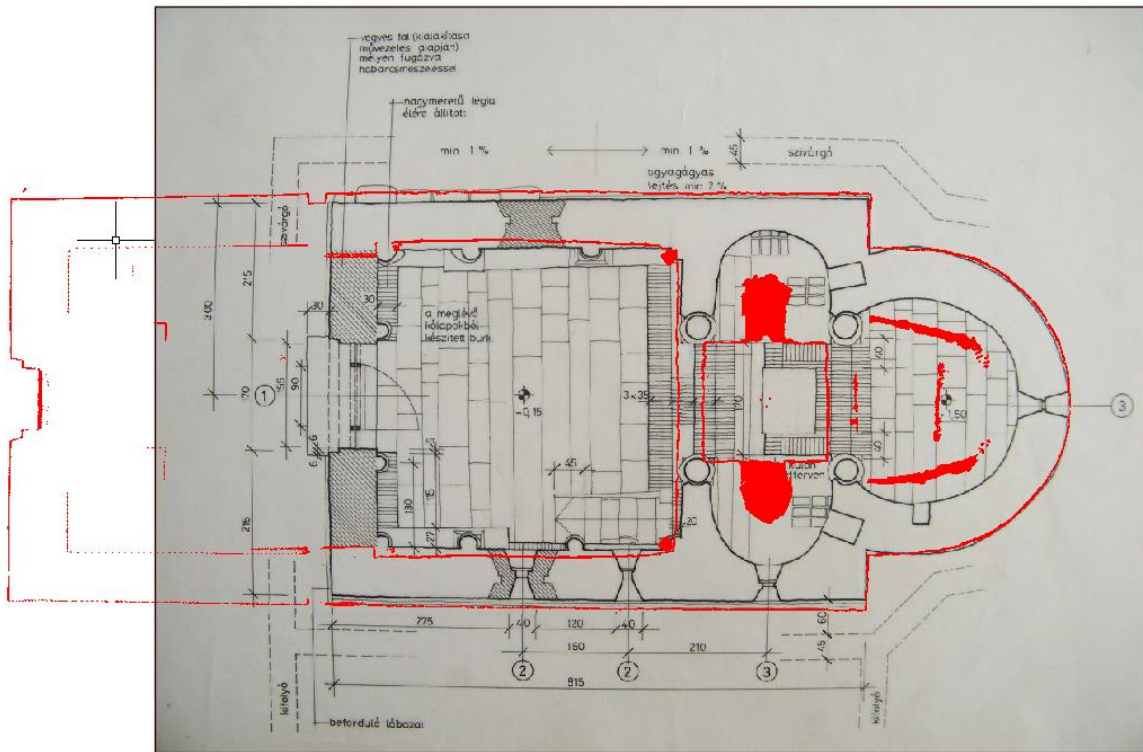
Megállapítható tehát, hogy a Leica típusú földi lézershakennerrel előállított pontfelhő nyers állományai nem csak a kereskedelmi szoftverrel dolgozhatók fel. A felhasználók akár kizárólag szabad szoftverekre támaszkodva is megfelelő pontosságú pontfelhőket érhetnek el. További automatizálásokkal ez a folyamat még nagyobb kényelmet biztosíthat. Továbbá elmondható az is, hogy a földi fotogrammetria, mint adatgyűjtési eljárás, a pontfelhők készítésében, alkalmazási területtől függően, megfelelő megbízhatóságú adatokat ad, továbbá jóval költséghatékonyabb, mint a földi lézershakenneres technológia.

### 13.1 A dolgozat végtermékei

Alaprajzok, és metszetek készültek, melyeken utólag bármilyen méretet meg lehet határozni, akkor is, ha azokat a terepen nem mértük meg.



13.1.1 ábra Templom metszete oldalnézetből, méretekkel [m]



13.1.2 ábra Az áltunk felmért és a korábbi alaprajz összevetése (Forrás: Forster Központ, Budapest)



13.1.3 ábra Metszetek a szentélyről és az altemplomról különböző szemszögekből

TDK dolgozatom során számos pontfelhőt állítottam elő, melyek megőrizték a műemlék templom pillanatnyi állapotát kívül és belül.

## 14. Irodalom jegyzék

Agisoft PhotoScan User Manual, Professional Edition, 2016, Version 1.2, Copyright © 2016 Agisoft LLC

Dr. Busics György (2015): Egy elfeledett hossz mérték, a királyi öl, és kapcsolata középkori építmények méreteivel. Tanulmány, Lánosz-Szekfü Közalapítvány, Székesfehérvár, 2015.

Dr. Busics György, 2011: Műholdas helymeghatározás 4.: GNSS transzformációs eljárások TÁMOP - 4.1.2-08/1/A-2009-0027, Székesfehérvár, 5-7

C.R.Kennedy & Company, 2016: Leica Cyclone Basic User Manual

CloudCompare Version 2.6.1 User manual, 2015

Detrekői Ákos, 1991: Kiegyenlítő Számítások, Budapest, Tankönyv kiadó, 369-396 old.

Dr. Engler Péter, 2011: A földi fotogrammetria, TÁMOP - 4.1.2-08/1/A-2009-0027, Székesfehérvár

John Ashburner & Karl J. Friston, 2004: Rigid Body Registration. London, UK

John Markoff, 1989: One Man's Fight for Free Software. The New York Times. Business Day, January 11, 1989

John W. Eaton, David Bateman, Søren Hauberg, Rik Wehbring, 2017: GNU Octave Free Your Numbers. Boston, USA

Kozák Károly, 1984: Közép-Európa Centrális Templomai (IX—XI. Század). 18-19

Dr. Lovas Tamás, Dr. Berényi Attila, Dr. Barsi Árpád, 2012: Lézerszkennelés. Budapest

Rudolf Staiger, 2003: Terrestrial Laser Scanning Technology, Systems and Applications. 2nd FIG Regional Conference, Marrakech, Morocco

Internetes hivatkozások:

<https://www.gnu.org/philosophy/>