hajder_274_24

MTA doktori értekezés

Geometriai alapú háromdimenziós számítógépes látás digitális kamerák és LiDAR érzékelők segítségével

Hajder Levente

Eötvös Loránd Tudományegyetem
Informatikai Kar

Algoritmusok és Alkalmazásaik Tanszék

Budapest, 2025

hajder_274_24

Köszönetnyilvánítás

Ez a dolgozat a 2008-2024 közötti tudományos munkámat foglalja össze. Visszaemlékezve az elmúlt másfél évtizedre, számtalan ember és szervezet segítette a munkámat.

2002-tól az MTA SZTAKI[1] tudományos segédmunkatársa, munkatársa, majd főmunkatársa lehettem, hálás vagyok az intézetnek, hogy kutatásaimhoz a szükséges felszereléseket és infrastruktúrákat rendelkezésre bocsájtotta. Az Eötvös Loránd Tudományegyetemen 2017 óta vagyok alkalmazásban, ahol különösen az 'EFOP EFOP-3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies' és a TKP2020-NKA-06: 'Alkalmazásiterület-specifikus nagy megbízhatóságú informatikai megoldások' projektek támogatták a munkámat, meglehetősen bőkezűen. A Robert Bosch Kft. támogatását szintén kiemelném 2022-től, a vállalat nemcsak anyagiakban finanszírozta a munkámat, hanem az ipari-egyetemi együttműködés során számomra új távlatokat nyitott ki.

Munkatársaim közül elsőként azokat a PhD hallgatókat sorolnám fel, akikkel a tizenöt év alatt együtt dolgozhattam: Tóth Tekla, Baráth Dániel, Fodor Bálint, Kazó Csaba, Pernek Ákos, Pusztai Zoltán, Tófalvi Tamás. Noha én a témavezetőjük voltam, magam is sokat tanultam Tőlük, amit nagyon köszönök Nekik. Nemcsak doktoranduszok, hanem MSc, sőt BSc hallgatók munkájából is meríthettem, a lista hosszúsága miatt itt sajnos nem sorolhatom fel Őket.

Feletteseim közül kiemelném még PhD-s korszakom két témavezetőjét, Csetverikov Dmitrijt († 2024) és Vajk Istvánt, mindketten alapvetően határozták meg világlátásomat. Ugyanígy kollégáim a SZTAKI-ban és az ELTE-n, akikkel együtt dolgozhattam, közülük külön megemlíteném Molnár Józsefet, Eichhardt Ivánt, Valasek Gábort, Jankó Zsoltot és Lóczi Lajost.

Személyes vonalon elsősorban családomat, feleségemet és öt gyermekemet említeném meg, akik az érzelmi hátteret biztos hátteret. Édesapámat és apósomat szakmai oldalról is ki tudom emelni, ők mérnökként, de rendkívül gyakorlatias gondolkodásukkal segítettek felismerni, hogy a józan paraszti ész még a tudományos munkában is az egyik legfontosabb feltétele a minőségi munkának.

[1] Jelenleg az intézet neve HUN-REN SZTAKI.

Jelölések, nyelvhasználat

A dolgozatban a számítógépes látás szakirodalmában leggyakrabban alkalmazott jelöléseket igyekszem használni. A jelölésrendszer egységesítése komoly kihívások elé állított, igyekeztem felnőni a feladathoz.

A térbeli és síkbeli pontokat általában vektorok formájában tároljuk. Ezeket vastag betűvel jelölöm a szövegben, szemben a skalár értékekkel, amelyet normál betűtípussal igyekeztem mindenhol írni.

A térbeli pontok leírására az $\mathbf{X}$ vektort használom a leggyakrabban, amely a térbeli elhelyezkedés három koordinátáját tartalmazza: $\mathbf{X} = [X \quad Y \quad Z]^T$. A vékony betűs X az első koordinátát, a vastag $\mathbf{X}$ pedig magát a vektort jelöli. A vetítéseknél a homogén koordinátás alakokat is használni fogom: $\mathbf{X} = [X \quad Y \quad Z \quad 1]^T$.

A képen a pontokat általában az $\mathbf{u} = [u \quad v]^T$ vagy $\mathbf{u} = [u \quad v \quad 1]^T$ homogén koordinátás alakban írom. Előfordul a kisbetűs $\mathbf{x} = [x \quad y]$ karakter használata is a síkon történő jelölésre.

Mátrixokkal szintén sűrűn találkozhat a szövegben a kedves Olvasó, hiszen a transzformációk, vetítések paramétereit legtöbbször mátrixokkal reprezentálják a szakirodalomban. Vastag betűs jelöléseket alkalmazok a mátrixokra, az elemeket vékony betűvel jelölöm. Dupla indexelést alkalmazok, az első index a sornak, a második az oszlopnak a számát jelöli. Például:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}.$$

Néha az áttekinthetőség miatt előfordul szimpla indexelés is, ebben az esetben sorfolytonosan írom az elemeket:

$$\mathbf{A} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}.$$

Szóhasználat. A téma szakirodalma döntő többségében angol nyelvű, mégis sokan érezzük kötelességünknek, hogy a saját anyanyelvünkön is műveljük a tudományt, és a szakszavakat megpróbáljuk magyarosítani. A témában magyar nyelven mindenképpen érdemes elolvasni Kató Zoltán és Czúni László könyvét [2], nemcsak magyarosítási szempontból, hanem azért is, mert remek összefoglalása a háromdimenziós látás alapvető elméletének és módszereinek.

Számomra a magyarosításkor két stratégia létezik: vagy fonetikusan átírom a kifejezéseket vagy megpróbálom ténylegesen lefordítani. A változatosság kedvéért mindkét stratégiát követtem. A lentebbi táblázatban a legfontosabb kifejezéseket fordítását írtam le, a teljesség igénye nélkül:

Angol kifejezés	**Magyarosított kifejezés**	**Magyar elnevezés**
Fundamental matrix	Fundamentális mátrix	Alapvető mátrix
Essential matrix	Esszenciális mátrix	Lényegi mátrix
Projection	Projekció	Vetítés
Triangulation	Trianguláció	Háromszögelés
Decomposition	Dekompozíció	Felbontás
Factorisation	Faktorizáció	Szorzattá bontás
Point registration	Pontregisztráció	

Társszerzők. A tudományos kutatás ma már csapatmunka. Az elmúlt másfél évtizedben, miután PhD fokozatomat megszereztem, jó néhány doktorandusz hallgatóval és kollégával dolgozhattam együtt. A munkában, még a tézisek leírásában is, igyekeztem a módszerek társszerzőinek hozzájárulását is megemlíteni. Ezért fordulhat elő, hogy egyes és többes számot egyaránt használtam. Mindenképpen megemlíteném, hogy hálás vagyok a doktori szabályzat megalkotóinak, hogy védett doktoranduszok esetében az egyszerzős folyóiratcikkek kötelessége alól felmentést kaptam, mert tudományos dolgozataimban mindig arra törekedtem, hogy bekerüljön minden résztvevő, a hozzájárulás nagyságától függetlenül. Így nem kellett senkit sem a szerzőség helyett a köszönetnyilvánítások közé száműzni .

Tartalomjegyzék

1. Bevezetés

Immáron több, mint húsz évvel ezelőtt kezdtem el számítógépes látással foglalkozni, amikor egyetemi diplomám megszerzése után megkezdtem doktoranduszi tanulmányaimat a Műegyetemen és tudományos segédmunkatársként az MTA SZTAKI-ban. Bő öt évvel később, 2008. januárjában védtem meg PhD fokozatomat, disszertációmat gyengén perspektív kameramodellekre épülő háromdimenziós rekonstrukciós algoritmusok megalkotásából írtam. Doktoranduszi éveim alatt szerettem meg a számítógépes látás világát, különösen az ragadott meg, hogy az algebra és a geometria határterületén megtapasztalhattam, a gyakorlatban használható módszerek esetében milyen fontos, hogy elméletileg is meg legyenek alapozva.

Disszertációmat gyengén perspektív kamerákkal kapcsolatos kutatásokból írtam, fokozatszerzésem után azonban figyelmem egyre inkább a valódi perspektív kamerák alkalmazása felé fordult, noha pár évig még foglalkoztam gyengén perspektív kamerákkal is: a doktoranduszi munkák során talált nyitott kérdések közül néhányra sikerült megoldást találni.

Korábbi témavezetőm és felettesem, Csertverikov Dmitrij 2010 után ismertetett meg tanítványával, Molnár Józseffel, aki egy SZTAKI-SZTE projekt keretében elkezdett foglalkozni az affin transzformációk alkalmazásával [113]. Bár ő hamar felhagyott a területtel, de számomra és saját doktoranduszaim számára ez a terület egy teljesen új világot nyitott meg, és alapvetően meghatározta a következő évtized kutatási irányait. Az elméleti alapok megismerése után saját algoritmusokat kezdtünk készíteni a tanítványaimmal, később az elméletet is sikerült újabb részekkel kiegészíteni, a perspektív kamerákra újabb törvényeket megfogalmazni. A munkát elsősorban Baráth Dániellel közösen végeztem, aki az elért eredmények alapján 2020-ban sikeresen megvédte doktori munkáját.

A dolgozatban szintén foglalkoztam a pontalapú, lyukkamerát alkalmazó háromdimenziós rekonstrukciós eljárásokra kifejlesztett numerikus algoritmus [167], a "kötegelt behangolás" alkalmazásával újabb problémaosztályokra. A kötegelt behangolás egy olyan eljárás, amelyben sok paramétert – tipikusan ezres nagyságrendben – kell becsülni, de a minimalizálandó költségfüggvényben a paraméterek csak lazán kötődnek. Ezt a területet a 2010-es években sikerült két újabb problémaosztályra kiterjeszteni két munkatársaimmal közösen a SZTAKI-s kutatócsoportunkban.

Az utolsó terület, amelyet ez a dolgozat – külön fejezetben – szintén taglal, LiDAR eszközök és perspektív kamerák kalibrációja. 2015-2016 táján kezdtünk el Pusztai Zoltánnal doktorandusszal közösen dolgozni a területen, és miután 2018-ban az ELTE Informatikai Karán elindult az Autonóm Rendszerinformatikus (MSc) képzés, a járműre rögzített érzékelők feldolgozásával még intenzívebben foglalkoztunk egyetemi kollégáimmal együtt. A kamerák mellett önvezető járművek esetén a LiDAR lézeres letapogatóeszközök is nagyon hasznosak/népszerűek. Azt tapasztaltuk, hogy kamerák és LiDAR-ok közös kalibrációjához a jelenlegi megoldások nem elég pontosak, ezért több új eljárást is javasoltunk. A munkába bekapcsolódott egy új doktorandusz, Tóth Tekla is, így ebből a területből két tanítványom is doktori fokozatot szerzett: Zoltán 2020-ban, Tekla 2024-ben.

A disszertáció felépítése. Ebben a dolgozatban kísérletet teszek arra, hogy az elmúlt másfél évtized során elért tudományos eredményeimet összefoglaljam. A 2. fejezetben rövid elméleti áttekintést adok a tézisek megértéséhez szükséges jelölésekről és törvényszerűségekről. Ezen belül a 2.2. fejezetben a képalapú rekonstrukciós eljárá-

sok alapvető módszereit ismertetem.

Az elméleti áttekintés után következik az egyes téziscsoportok részletes ismertetése. A 3. fejezet a lokális affin transzformációkkal kapcsolatos legfontosabb elméleteket és az azokon belül megfogalmazott újdonságokat tartalmazza. A 4. fejezet konkrét algoritmusokat ismertet, amelyek segítségével az affin transzformációkat a háromdimenziós látásban alkalmazni lehet. Terjedelmileg és újdonság tartalmában is ez a rész a legsűrűbb, hiszen a legtöbb eredményt ezen a területen sikerült közzétennem.

Az 5. fejezet egy teljesen új témával foglalkozik, azt mutatom be, hogyan sikerült a kötegelt behangolás algoritmust kiterjeszteni újabb problémaosztályokra. A rákövetkező, 6. fejezet pedig azt ismerteti, milyen újszerű módszereket sikerült LiDAR-ok és digitális kamerák külső paramétereinek meghatározásához kifejleszteni.

Végezetül a 7. fejezetben foglalom össze a doktori munkámat.

Kutatási együttműködések. A kutatás a mai, modern világban a legtöbbször csapatmunka. Az egyes tézispontoknál igyekeztem a társzerzők szerepvállalását pontosan rögzíteni. Elsősorban saját doktoranduszaimmal dolgoztam, de helyenként MSc és BSc hallgatókkal is sikerült tudományosan értékesnek tekinthető módszereket kifejleszteni. Szerencsésnek mondhatom magamat abban az értelemben, hogy 2021-ben az Eötvös Loránd Tudományegyetemen saját kutatócsoportot alapíthattam, így saját tanítványaim már egy önálló csoportban tevékenykedhetnek. Az együttműködések elsősorban országon belüliek voltak, bár az utóbbi években kutatócsoportommal igyekszünk külföldi együttműködések felé is nyitni, amely a publikációs listámban egy-egy társszerző képében is megmutatkozik. A tézisek megfogalmazásánál igyekeztem az együttműködésekben a saját szerepemet minél pontosabban leírni.

2. Elméleti alapok

A számítógépes látás az esetek többségében középpontos vetítést megvalósító kamerákat alkalmaz, ezen a kategórián belül is a leggyakrabban lyukkamerát szokás alkalmazni [62]. Ilyen leképző eszközök esetén a vetítés geometriai modellje egyszerű: a térbeli pontot látó pixel helyét úgy kapjuk meg, hogy a térbeli pontot összekötjük a kamera fókuszpontjával, és így kapunk egy vetítő egyenest. Ennek az egyenesnek és a képsíknak a metszéspontja jelöli ki azt a pixelt, amelyen a térbeli pont vetülete látszik.

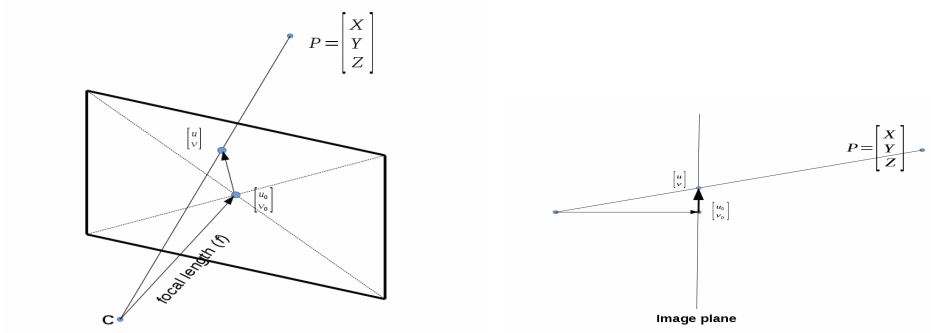
A digitális kamerák szinte kivétel nélkül középpontos vetítést valósítanak meg, de lehetőség van sík helyett más felületekre is vetíteni. Az érdeklődő Olvasók Geyer és Daniilidis munkájában [54] találhatnak kiváló összefoglalást az alternatív kameramodellekre. Munkám során elsősorban lyukkamerával foglalkoztam, bár merőleges [164] és skálázottan merőleges [175] vetítési módszerekkel is értem el eredményeket. Ezek a modellek is felfoghatóak a lyukkamera speciális esetének, amikor a fókusztávolságot végtelen hosszúra megnyújtjuk.

Amennyiben a koordináta-rendszerünk a kamerához van rögzítve, a tengelyek párhuzamosak a vízszintes és függőleges pixeliránnyal, a harmadik tengely (optika tengelynek nevezik) pedig merőleges a képsíkra. A vetítést az 1. ábra mutatja be, a bal oldalon térbeli, a jobb oldalon síkbeli ábrázolással. Ez utóbbit úgy kell elképzelni, hogy oldalról nézünk a kamerára, tehát az ábrára merőleges irány megfelel a vízszintes pixeliránynak.

A perspektív kamerák paramétereit az alábbi felső kalibrációs mátrixban szokás ábrázolni:

$$\mathbf{K} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

ahol f_u és f_v az optika fókusztávolságának és a szenzoron a (vízszintes és függőleges) pixelméretnek a szorzata. Ezt a két paramétert a szakmában fókusztávolságnak szokás nevezni, ebben az alakban a mértékegysége pixelben értendő, azaz a fókusztáv hosszát a szenzor pixeleinek méretéhez viszonyítva fejezzük ki. Ha a pixel alakja négyzet, ami az esetek döntő többségében igaz, akkor $f_u = f_v$. Az $[u_0 \quad v_0]^T$ pixelpozíció az úgynevezett döféspont, amit geometriai megközelítésben a képsík és az optikai tengely metszéspontjaként kaphatunk meg. Ez a pont a perspektíva középpontja a képen. Speciális esetek – például gumioptika ('zoom') alkalmazása esetén –



1. ábra. *Perspektív vetítés általános nézetből (bal oldal) és oldalról (jobb oldal)*

az f_u és f_v skálázási paraméterek arányosan változhatnak a felvételek között.

A hétköznapokban nemperspektív kamerákat is szoktak használni, de ezeket kiegyenesítéses [2] algoritmussal perspektívokká lehet alakítani [190, 149].

A képsíkra vetítés képletekkel így határozható meg:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \mathbf{K} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} f_u X + u_0 Z \\ f_v Y + v_0 Z \\ Z \end{bmatrix}. \tag{2}$$

Az összefüggésben a $\sim$ (hullám) operátor a skálázás erejéig egyenlőséget jelenti. Ha $\mathbf{a} \sim \mathbf{b}$, az azt jelenti, hogy létezik olyan valós α szám, amelyre $\alpha\mathbf{a} = \mathbf{b}$. A skála eltüntetésére az úgynevezett homogén osztás műveletet lehet alkalmazni, mely szerint az α értékét az utolsó koordináta reciproka adja. A fenti vetítési egyenletre így az alábbi két összefüggés adódik, ezek a végső egyenletek, melyek a térbeli $[X \quad Y \quad Z]^T$ pontot és a vetített $[u \quad v]^T$ pixelek közötti kapcsolatot leírják:

$$u = \frac{f_u X + u_0 Z}{Z} = f_u \frac{X}{Z} + u_0, \qquad v = \frac{f_v Y + v_0 Z}{Z} = f_v \frac{Y}{Z} + v_0.$$

Ha a koordináta-rendszer "valahol a világban" van, egy $\mathbf{R}$ elforgatásmátrix és egy $\mathbf{t}$ eltolásvektor segítségével előbb a kamerába kell transzformálni a pontokat [3] . Ekkor írhatjuk, hogy

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \mathbf{K} \, [\mathbf{R}|\mathbf{t}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{3}$$

Jól látható, hogy ebben az esetben már a jobb oldalon elhelyezkedő térbeli pontot is homogén koordinátával kell leírni, hogy a koordináta-rendszerek közötti eltolást is kompakt, mátrixos alakban meg tudjuk adni. Fontos megjegyzés, hogy a "világban" levő pontokat világ koordináta-rendszerben adjuk meg, ha rendszerünk már a kamerához van rögzítve, akkor kamera koordináta-rendszerről beszélünk.

2.1. Visszavetítés és felületmetszés

A 2. összefüggés alapján a visszavetítést is jól meg tudjuk határozni:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \alpha \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \tag{4}$$

ahol α értéke tetszőleges lehet, és a skálázás erejéig egyenlőség operátort váltja ki a 2. összefüggésben. Ez az egyenlőség geometriailag úgy értelmezhető, hogy az adott pixelhez vetítés előtt végtelen sok pont tartozik, ezek a pontok alkotják a vetítőegyenest.

Amennyiben adott egy térbeli test, és arra lennénk kíváncsiak, hogy melyik pontja látszik a pixelen, akkor a felületet el kell metszeni a vetítősugárral. Erre a feladatra egy példát síkmetszésen keresztül mutatok meg.

[2] Angolul: rektifikáció.

[3] Az $\mathbf{R}$ forgatás és a $\mathbf{t}$ eltolás segítségével az elképzelhető összes egybevágósági transzformációt le lehet írni a térben, ha $\mathbf{R}$ ortonormált, azaz $\mathbf{R}^T\mathbf{R} = \mathbf{I}$.

Metszés síkkal.

A visszavetítés alkalmazásához egy egyszerű példát tekintünk, ahol a feladat egy adott pixelhez meghatározni, hogy a sík melyik pontját látja, ha tudjuk, hogy a sík a legközelebbi objektum. Ez a feladat a számítógépes grafikában [106, 41] a sugárvetés (ray-casting), amely a sugárkövetés (ray-tracing) alapproblémája [4].

Ha adott egy sík, amelyet $\mathbf{n}$ felületi normálvektorával és $\mathbf{p}_0$ pontjával adunk meg, egy $\mathbf{p}$ pont akkor van a síkon, ha az $\mathbf{n}^T(\mathbf{p}-\mathbf{p}_0) = 0$ implicit egyenlet teljesül. A visszavetített sugárnak és a síknak a metszéspontját egyszerűen megkaphatjuk, ha a vetítőegyenes kifejezését behelyettesítjük a $\mathbf{p}$ pont helyére, a sík implicit egyenletében:

$$\mathbf{n}^T\left(\alpha\mathbf{K}^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - \mathbf{p}_0\right) = 0. \tag{5}$$

Ebből az összefüggésből a hiányzó α paraméter elemi műveletek segítségével meghatározható:

$$\alpha = \frac{\mathbf{n}^T\mathbf{p}_0}{\mathbf{n}^T\mathbf{K}^{-1}\begin{bmatrix} u & v & 1 \end{bmatrix}^T}. \tag{6}$$

A megkapott α paramétert a sugár megadásába (a 4. egyenlet) visszahelyettesítve megkapjuk a metszéspont helyét a térben.

A későbbiekben, a IV. téziscsoportban fog a kedves Olvasó a visszavetítéssel találkozni, amikor azt mutatom meg, hogyan lehet gömbök kontúrjaiból a gömb középpontját meghatározni, ha a kamera paraméterei ismertek, azaz a kamera kalibrált.

2.2. Képalapú háromdimenziós rekonstrukció

Ebben a fejezetben röviden összefoglalom a két- és sokképes rekonstrukció lépéseit, kiemelve a legfontosabb és leggyakrabban használt fogalmakat és algoritmusokat.

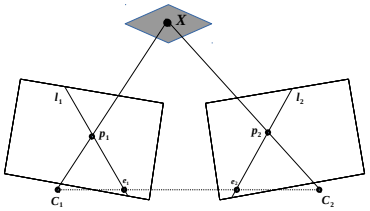
2.2.1. Kétképes (sztereó) rekonstrukció

A sztereó rekonstrukció problémáját a 2. ábrán láthatjuk. Egy térbeli felületdarabka $\mathbf{X}$ vektorral jellemzett pontját levetítjük a két kamerára. Középpontos vetítést alkalmazunk, a két kamera vetítési középpontjait, melyeket fókuszpontnak hívnak, $\mathbf{C}_1$-el és $\mathbf{C}_2$-vel jelöljük. A vetítés legegyszerűbb modellje a síkra vetítés, egy átlagos kamera igen jó közelítéssel megfelel ennek a modellnek. Ahogy azt fentebb már leírtam, a térbeli $\mathbf{X}$ pont képét úgy kapjuk meg, hogy összekötjük a pontot a fókuszpontokkal, így vetítősugarakat kapunk, és a két kamerasíkon az elmetszett pixelek adják a képen a pont helyét.

A számítógépes látás feladata algoritmikusan visszaállítani a térbeli $\mathbf{X}$ pont helyét, ha a két képen ismerjük a vetületi pontokat. A rekonstrukciós folyamatokat az alábbi lépésekre szokták osztani: kamera kalibráció, epipoláris geometria becslése és háromszögelés (trianguláció).

[4] Egy lábjegyzetet érdemes annak a meggondolásnak szánni, miszerint a háromdimenziós számítógépes grafika és látás egymás inverz feladatai. Grafikában a térbeli modellekből kell életszerű képeket előállítani, látásban a képekből próbáljuk rekonstruálni a térbeli információt, többek között a modelleket.

Kamera kalibráció. Amennyiben a kamerának a belső paramétereit, azaz a fókusztáv(ok)at és a döféspontot ismerjük, a rekonstrukciós egyenletek egyszerűsödnek, és az eredmények numerikusan is pontosabbak lesznek. Általában sakktábla segítségével [190] be lehet az alkalmazott kamerákat előzetesen kalibrálni. Sakktáblát azért használnak széleskörűen, mert szabályos a mintázata, és a sakktáblacellák alakját (élek, sarokpontok) nagyon pontosan meg lehet határozni. *Ha a belső paraméterek ismertek, a kamerákat kalibráltaknak nevezik.*



2. ábra. *Sztereó látás alapproblémája. A térbeli objektum* $\mathbf{X}$ *pontját a két képsíkra középpontosan vetítjük. A kamerákat (vetítést) a képsíkok és a* $\mathbf{C}_1$*,* $\mathbf{C}_2$ *fókuszpontok határozzák meg.*

Epipoláris geometria becslése. Az epipoláris geometria a projektív geometria kétképes esete. Kalibrált kamerák esetén egyszerűbb a feladat, ebben az esetben az úgynevezett lényegi (esszenciális) mátrix reprezentálja az epipoláris geometriát. Ismeretlen kameraparaméterek esetén csak az alapvető (fundamentális) mátrix meghatározása lehetséges [5].

Fundamentális mátrix becslése két képpár között: két általános perspektív kamera között megadható egy 3×3-as mátrix, amely a képek között levő geometria kapcsolatot általános esetben is képes leírni. Ezt a mátrixot hívják alapvető (fundamentális) mátrixnak, és a szakirodalomban általában $\mathbf{F}$-fel jelölik. A mátrixnak ugyan kilenc eleme van, de valójában hét paraméter segítségével leírható, mert két megkötést is lehet tenni:

1. A mátrix szinguláris, azaz a determinánsa zérus, ebből egy egyenletet lehet felírni, tehát egy szabadságfokot lehet meghatározni.

2. Másrészt skálázásra invariáns a mátrix, azaz ugyanazt a geometriai kapcsolatot írja le, ha egyazon valós számmal szorozzuk meg az alapvető mátrix minden elemét. Ez a tény még egy szabadságfokon levon a problémából.

Kalibrált kamerák esetén az alapvető mátrixban lévő ismeretlenek számát tovább lehet csökkenteni, mivel csupán a kamerák külső paraméterei határozzák meg azt. Ebben az esetben már lényegi mátrixnak szokás nevezni. A forgatás – melyet az $\mathbf{R}$ ortonormált mátrixszal szokás reprezentálni – három paraméterrel írható le, az eltolásvektor $\mathbf{t} = [t_x \quad t_y \quad t_z]^T$ további hárommal, de képalapú térbeli látás esetén az eltolás nagysága nem rekonstruálható, csupán az iránya, ezért csak két paraméter

[5] A magyar nyelvű szakirodalomban az idegen elnevezések fonetikus átírását alkalmazzák (fundamentális és esszenciális mátrixok), itt megkísérlem a magyar fordítást is használni (alapvető és lényegi mátrixok).

határozza meg az eltolást. Így összesen $3 + 2 = 5$ paramétert kell becsülne, ebből a kettőből lehet a lényegi mátrixot előállítani az alábbi módon:

$$\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}, \tag{7}$$

ahol $[\mathbf{t}]_{\times}$ a $\mathbf{t}$ eltolásvektorból készített, bal oldali vektoriális szorzást reprezentáló 3×3-as mátrix. Az $\mathbf{F}$ alapvető és $\mathbf{E}$ lényegi mátrixokat kalibrált kamerák esetén könnyen át lehet számítani az $\mathbf{E} = \mathbf{K}_2^T \mathbf{F} \mathbf{K}_1$ összefüggéssel, melyben $\mathbf{K}_1$ és $\mathbf{K}_2$ az első és a második kamera belső paramétereit leíró felső háromszög mátrix, melyet az 1. összefüggésben láthattunk először.

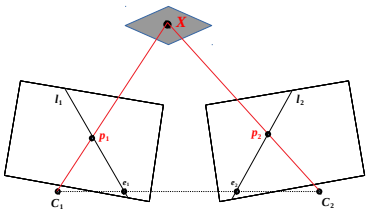
Mind az alapvető, mind a lényegi mátrixok meghatározására a sztereó képpárokon pontmegfeleléseket határoznak meg. Minden pontpár egy egyenletet ad, melyet az alábbi alakokban lehet leírni:

$$\mathbf{p}_2^T \mathbf{F} \mathbf{p}_1 = \mathbf{p}_2^T \mathbf{K}_2^{-T} \mathbf{E} \mathbf{K}_1^{-1} \mathbf{p}_1^T = 0. \tag{8}$$

A lényegi mátrixhoz öt [121], az alapvető mátrixhoz hét pontpárra van szükség, miután minden pontpár egy egyenletet ad. Praktikus okokból az alapvető mátrix becslésére a nyolcpontos algoritmus is igen népszerű [60], még akkor is, ha eggyel kevesebb pontból is meg lehetne határozni a mátrixot.

Háromszögelés (trianguláció)

Ha két képen ismerjük az egymásnak megfelelő pontokat, a pont térbeli helyzete kiszámítható. A két képen a pontok egy-egy vetítőegyenest határoznak meg, az eredeti térbeli pont helye közel lesz az egyeneshez. A feladathoz tartozó rajzot a 3. ábra mutatja meg. Pirossal jelöltem a vetítőegyeneseket, ezekhez legközelebbi pont adja a becsült térbeli pont helyét.



3. ábra. *A triangulációs probléma: a két képen* $\mathbf{p}_1$ *és* $\mathbf{p}_2$ *egymásnak megfelelő pontok meghatároznak egy-egy vetítőegyenest. Ideális esetben ezek metszéspontja adja a vizsgált térbeli pont helyét. A zaj miatt a gyakorlatban az egyenesek kitérőek, ezért a térbeli pont helyét becsülni kell.*

Többféle eljárás létezik háromszögeléshez, van, amelyik algebrai hibát minimalizál [62], másik megoldás a vetítőegyenesekhez legközelebbi pontot találja meg. Létezik optimális trianguláció [61] is kétképes esetre, amely attól optimális, hogy a képtérben minimalizálja a hibát, és a képek közötti geometriai megkötést is rögzíti, melyet az alapvető mátrix határoz meg.

2.2.2. Többképes rekonstrukció

A sztereó rekonstrukció kiterjesztése sokképes rekonstrukcióra többféleképpen képzelhető el. Lehetséges kétképes rekonstrukciók összefűzésével, ekkor az egyes sztereó

(rész)eredményeket a megfelelő hasonlósági transzformáció segítségével lehet egymásba eltolni és elforgatni. A triangulációs feladat megoldása többképes esetben is lehetséges, optimális eljárás – képtérben nézve a hibát – ismereteim szerint erre a feladatra még nem készült el.

Megjegyezném, hogy háromképes projektív geometriát a trifokális tenzor [165] tudja leírni, azonban ennek jelentősége a gyakorlatban marginális.

Akármilyen módszerrel készítjük is el a rekonstrukciós feladatot, a megoldás akkor a leghatékonyabb, ha a képtérben minimalizáljuk a hibát. Az eljárás során az ismeretlen, behangolandó paraméterek a háromdimenziós koordináták és a kameráknak a paraméterei. Ez utóbbiak lehetnek belső és külső (úgynevezett póz) paraméterek.

A hangolás általában numerikus optimalizálás segítségével történik, a Levenberg-Marquardt [96, 108] algoritmust szokás leggyakrabban alkalmazni a feladatra. Miután a probléma meglehetősen ritka, egy térbeli rekonstruált pont csak a hozzá tartozó képpozíciókat befolyásolja, a normálegyenletben – és magában a Jakobi-mátrixban is – sok nulla érték van. Ezért a feladat kisebb részfeladatokra bontható. Ezt a speciális, 3D rekonstrukcióra kifejlesztett eljárást nevezik kötegelt behangolásnak [167].

Ezt a problémakört a harmadik tézispontban részletesebben is be fogom mutatni az 5. fejezetben.

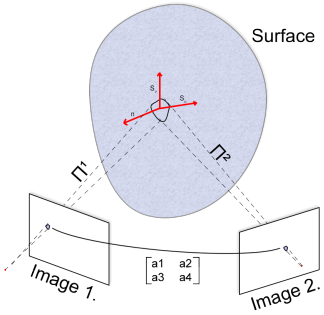
2.3. Affin transzformációk a kétképes látásban

Molnár József és Csetverikov Dmitrij 2014-ben jelentette meg véleményem szerint úttörő munkájukat [113], melyben összekötik a lokális affin transzformációkat és a kétképes projektív geometriát: megmutatják, hogy ha egy térbeli síkfelület egy adott pontját két képre levetítjük, akkor a vetített pontok körüli lokális affin transzformáció hogyan függ a térbeli pont helyétől, a kameraképtől és a felületre merőleges normálvektortól. Eredményüket egyetlen általános összefüggésbe lehet foglalni:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \frac{1}{\nabla\Pi_u^{1\,T} [\mathbf{n}]_\times \nabla\Pi_v^1} \begin{bmatrix} \nabla\Pi_u^{2\,T} [\mathbf{n}]_\times \nabla\Pi_v^1 & \nabla\Pi_u^{1\,T} [\mathbf{n}]_\times \nabla\Pi_u^2 \\ \nabla\Pi_v^{2\,T} [\mathbf{n}]_\times \nabla\Pi_v^1 & \nabla\Pi_u^{1\,T} [\mathbf{n}]_\times \nabla\Pi_v^2 \end{bmatrix}, \tag{9}$$

ahol $\mathbf{n}$ az érintőfelületre merőleges úgynevezett normálvektor, $\mathbf{A}$ a lokális affin transzformáció 2×2-es lineáris része, Π_u és Π_v a vetítő függvények a képsíkon vízszintes és függőleges irányban. Az összefüggésben a felső index a képszámot jelenti, $\nabla\Pi$ pedig a vetítő függvény térbeli koordináták szerinti gradiense, azaz egy háromdimenziós vektor. A geometria összefüggéseket a 4. ábrán mutatom meg.

Magának az összefüggésnek a levezetése az A. függelékben olvasható. Az eredeti munka [113] véleményem szerint kicsit nehezen érthető, ezért a levezetést újraírtuk, és később publikáltuk [15] is.



4. ábra. *Az affin transzformáció a két kép közötti lineáris transzformáció. A térbeli pont, a felületi érintő* $\mathbf{n}$ *normálvektora és a* Π *vetítőfüggvények határozzák meg.*

A kapott formula nagy értéke, hogy bármilyen vetítőfüggvény esetén működik, nem korlátozódik egyetlen kameratípusra sem. A fenti alapegyenlet tehát egy általános összefüggést ad, az alkalmazott kameratípusok közötti különbség a Π vetítőfüggvényekben jelenik meg.

3. I. téziscsoport: új tudományos eredmények affin megfeleltetések felhasználásával

Ahogyan már a bevezetőben is említettem, a 2010-es évek első felétől kezdtünk el kollégáimmal affin transzformációk alkalmazásával foglalkozni. Affin transzformációkat két kép között lehet értelmezni azáltal, hogy az egymásnak megfelelő pontok környezetét kiemeljük, és a környezet pixelei közötti alakváltozást lineáris transzformáció segítségével közelítjük. Egy általános lineáris transzformációt két kép között az alábbi összefüggés segítségével formalizálhatjuk:

$$\mathbf{p}_2 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \mathbf{p}_1 + \begin{bmatrix} a_u \\ a_v \end{bmatrix}, \quad \mathbf{p}_1 = \begin{bmatrix} u_1 \\ v_1 \end{bmatrix}, \quad \mathbf{p}_2 = \begin{bmatrix} u_2 \\ v_2 \end{bmatrix}. \tag{10}$$

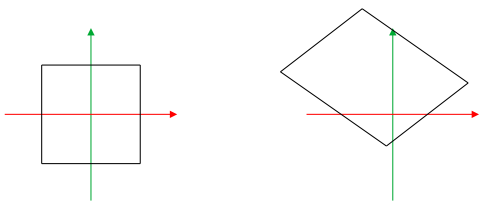
Az affin transzformáció tehát összeköti a $\mathbf{p}_1$ és $\mathbf{p}_2$ képkoordinátákat. Az eltolás két paramétere (a_u és a_v) jól elkülöníthető a maradék négy paramétertől (a_{11}, a_{12}, a_{21} és a_{22}). Ez utóbbi négy paraméter együttesét hívják lineáris paramétereknek, mert ezekkel kell szorozni a koordinátákat, a másik két paramétert egyszerűen hozzá kell adni az eredményhez. Az 5. ábrán láthatjuk, hogyan változtatja meg egy általános affin transzformáció a négyzetet, hogyan kapunk a segítségével paralelogrammát eredményül.

Ahhoz, hogy jobban el tudjuk emberi fantáziával képzelni, kategorizálni érdemes az affin transzformációkat:

- eltolás,
- kétdimenziós (síkban) forgatás,
- skálázás mindkét főirány mentén és
- nyírás.

Ezeket a transzformációkat figyelembe véve fel lehet bontani részekre az affin transzformációt, ahogyan azt az F. függelékben részletesen leírom.

A dekompozíció után a formula így alakul:



5. ábra. *Az affin transzformáció hatása egy négyzetre. Az eredmény az eredeti négyzet eltolt, forgatott, nagyított és nyírt mása. Paralelogrammát tudunk előállítani a négyzetből.*

$$\mathbf{p}_2 = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} s_u & 0 \\ 0 & s_v \end{bmatrix} \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} \mathbf{p}_1 + \begin{bmatrix} a_u \\ a_v \end{bmatrix}.$$

A forgatást tehát az α szög reprezentálja, a vízszintes skálát s_u, míg a függőlegeset s_v. A b valós paraméter a nyírásásért felelős.

Az F. függelékben részletesebb leírást talál a kedves Olvasó a felbontásról, illetve arról, hogyan lehet a felbontáshoz a paramétereket meghatározni. Ezen kívül még egy alternatív, szinguláris érték szerinti felbontást is tartalmaz ez a függelék.

3.1. Szakirodalmi áttekintés

A számítógépes képfeldolgozás körülbelül ötvenéves tudományága az informatikának [57]. A kezdeti módszerek kameraképek feldolgozásával foglalkoztak, de nagyon hamar elmozdultak térbeli látás irányába is, hiszen két képből már térbeli információ is előállítható [101]. A 3D látásnak egyébként nem is annyira a képfeldolgozás, mint inkább a fotogrammetria [154] tekinthető első számú elődjének, hiszen két mérési pontból a földmérők már több, mint egy évszázada képesek kiszámítani a tárgyak térbeli pozícióját.

A jelenkorban, a korszerű gépi látásban alapvetően kétféle megközelítéssel dolgoznak a szakemberek:

1. *Jellegzetes pont alapú megoldások*. [62]. A képeket jellegzetes pontokat [103, 21] detektálunk, majd ezeket egymásnak megfeleltetjük. Ezekből a megfeleltetett koordinátákból aztán ki lehet számítani először a kamerák egymáshoz képesti relatív elhelyezkedését, majd pedig a térbeli pozícójukat is.

2. *Látás gépi tanulás segítségével* [64, 43] Betanított hálózatok képesek az egyes részproblémákat, vagy akár a teljes rekonstrukciós folyamatot (úgynevezett end-to-end tanulással) előállítani. Az elmúlt időszak robbanásszerű fejlődésének egyik oka a gépi tanulás előretörése volt. A járműre szerelhető szenzorok, azon belül is a kameraképek hatékony feldolgozása manapság elképzelhetetlen a mestersége intelligencia nélkül. Azonban sokszor már maga a mesterséges intelligencia a fejlődés gátja: ugyan jóval 90%-os hatékonyság felett teljesít, de a 100%-os hatékonyság még mindig messze van.

A két megközelítést egyesíteni is lehet, például mélyháló segítségével lehet jellegzetes pontokat detektálni [39], majd képek között párosítani [145], végezetül a megfeleltetett pontokból ki lehet a térbeli információt számítani.

Ebben a doktori disszertációban én az első megközelítéssel foglalkozom, de új megközelítésben. A szakirodalomban a pontalapú megközelítések mind a mai napig szinte egyeduralkodóak [62], azaz jellegzetes pontokat keresnek a képeken, és azokat egymásnak megfeleltetik, majd ezek alapján elvégezhető a térbeli információk becslése.

Érdekesség, hogy a különböző képeken a jellegzetes pontokat úgy keresik meg, hogy a pontok körül megvizsgálják a pixeleket, és azokból leírókat készítenek. Tehát nemcsak egy pixelt, hanem annak a környezetét is megvizsgálják. Amennyiben pedig egy ilyen környezetet tekintenek, a környezetek közötti alakváltozást is meg lehet határozni. Elsőrendű közelítésben ezt az alakváltozást egy lineáris transzformációval írhatjuk le, ezt nevezzük lokális affin transzformációnak.

(a) *Pontalapú rekonstrukció. Az egymásnak megfelelő pontok meghatározzák a térveli pontot.*

(b) *Affin transzformációt figyelembe vevő rekonstrukció. A felületi normálvektor is kiszámítható.*

6. ábra. *A pontalapú és az affin transzformáció alapú rekonstrukciók. (A kép színes nyomtatásban könnyebben értelmezhető.)*

A két megközelítés közötti eltérést a 6. ábra szemlélteti. A bal oldalon (6a. ábra) azt láthatjuk, hogy a két képen adottak a $\mathbf{p}_1$ és $\mathbf{p}_2$ összetartozó képpontok, míg a jobb oldalon ((6b. ábra) az affin transzformációk is rendelkezésre állnak. Ekkor lehetőség van kinyerni a térbeli $\mathbf{X}$ pozíción kívül a felületre merőleges $\mathbf{n}$ normálvektort is.

Ahogyan azt korábban, a 2. fejezetben röviden már bemutattam, ismert kamera paraméterek és pózok esetén háromszögeléssel a térbeli pont kiszámítható. Ezért a klasszikusnak mondható pontalapú rekonstrukciós eljárások esetén a kimenet térbeli pontok sorozata, melyet röviden nevezhetünk pontfelhőnek is.

Affin rekonstrukció esetén, ahogyan azt a 2.3. szakaszban ismertettem Molnár József és Csetverikov Dmitrij munkája alapján [113], rendelkezésre áll egy általános összefüggés, amely összeköti az affin transzformációt a kamera paramétereivel, a háromszögelt felületi ponttal és a hozzá tartozó felületi normálvektorral. Rekonstrukció esetén ezért nemcsak térbeli pontokat, hanem hozzájuk tartozó felületi normálvektorokat is kaphatunk, ezért az eredmény egy irányított pontfelhő lesz.

Irányított pontfelhők feldolgozása érdekes és még napjainkban is intenzíven kutatott terület mind a számítógépes látásban, mind a robotikában, akár még a geometriai modellezésben is, számos fontos valós alkalmazás profitál az irányok nyújtotta többletinformációból.

Ilyen alkalmazások lehetnek a 3D rekonstrukció [5, 14, 65, 150, 191, 156, 137], egyidejű lokalizáció és feltérképezés (SLAM) [6] [38, 44, 117], ahol a térbeli pontnormálisok nagyon hasznos információkat szolgáltatnak a felületről. A képalapú vizuális lokalizációs megközelítések [105, 128, 146, 147] szintén profitálhatnak az ismert felületi normálisokból [173]. A széles körben használt Poisson-rekonstrukció [81, 82] mind a pontkoordinátákat, mind azok orientációját képes felhasználni.

Az autonóm járművekhez az egyik fontos algoritmus a jármű mozgásának meghatározása. Ha ez kameraképek segítségével történik, a feladatot vizuális odometriának nevezzük [123, 124]. Ezek a módszerek szintén hasznosíthatják az affin transzformációkat, ahogy ezt a későbbiekben egy saját algoritmuson keresztül be is fogom mutatni.

Az orientált pontfelhő birtokában a geometriai modellek és az egyidejűleg több modellt becslő módszerek nagymértékben egyszerűsödnek [72, 11, 12], mivel olyan

[6] Angolul: Simultaneous Localization and Mapping (SLAM)

algoritmusok készíthetőek, amelyek kevesebb adatpontot igényelnek, mint pontalapú társaik [151]. Ezen túlmenően a normálisok a rekonstruált orientált háromdimenziós pontfelhő numerikus finomításában is felhasználhatók [161].

Magát a normálvektor-becslési problémát ismereteim szerint először a sok évtizeddel ezelőtt bevezetett fotometrikus sztereó (Photometric Stereo – PS) segítségével oldották meg [179, 180]. E módszer legnagyobb hátránya, hogy csak olyan laboratóriumokban használható, ahol a fényviszonyok teljesen ellenőrzöttek. Később, az 5. fejezetben egy konkrét algoritmust is be fogok mutatni. Hétköznapi körülmények között is lehetséges egy képből a felszín normálvektorait rekonstruálni [177], ha tudunk a felületre megkötéseket tenni, pl. a homogén színt és a felületek viszonylagos simaságát. Ezeket az eljárásokat "shape from shading" [7] néven hivatkozzák meg. Az alkalmazások egy jelentős része a bolygók felszínének rekonstrukciójával foglalkozik, hiszen ezekről a felületekről nem könnyű különböző nézőpontú képeket készíteni.

Ismereteim szerint az affin transzformációk alkalmazásával a sztereó látásban először Megyesi Zoltán és Csetverikov Dmitrij foglalkozott először [109]. Egy speciális esetet, az emberi szem működési modelljének is megfelelő standard sztereó kamerabeállításra készítettek egy felületi normálvektorbecslőt. Ebben az esetben az affin transzformáció két szabad paraméterrel leírható, a speciális beállítás két paramétert a háromdimenziós világ tartalmától függetlenül egyértelműen meghatároz.

Pár évvel később, 2009-ben Kevin Köser doktori munkáját [84] szintén az affin transzformációknak szentelte, a szerző több oldalról vizsgálta meg a kérdés, nemcsak a normálvektorok becslésével foglalkozott, hanem a kamerák külső paramétereinek meghatározásával is, bár a munkájában az általános eseteket nem vizsgálta meg, ő is a lehetséges konfigurációknak csak egy részhalmazával foglalkozott.

A mi munkásságunk előtt még egy jelentősebb publikációt emelnék ki: Jacob Bentolila és Joseph M. Francos [23] egy harmadfokú formulát vezetett le, melyek egy affin transzformációt és a fundamentális mátrixot közötti összefüggést írják le.

Az elmúlt évtized során az affin transzformációkkal kapcsolatban kollégáimmal közösen három törvényszerűséget sikerült kidolgozni. Büszkén írhatom, hogy mindhárom magyar kutatókhoz köthető, akikkel volt szerencsém együtt dolgozni.

Ezek a törvényszerűségek a következők:

1. Általános összefüggés írható fel az affin transzformációk, a pontok képbeli és térbeli elhelyezkedése, a projekciós mátrix és a felületi normálvektor között.

2. Kapcsolat létezik az affin transzformáció és a homográfia között. Az affin transzformáció lineáris részét meg lehet kapni egy kiválasztott pontban, ha a homográfiát meghatározó vetítő függvényt deriváljuk az első képen a hely koordinátáival.

3. Lineáris összefüggés adható a fundamentális mátrix és az affin transzformációk között. Bebizonyítottuk, hogy a két mátrix között fel lehet egy kétdimenziós egyenletet írni.

[7] Magyar fordítását nem ismerem, a javaslatom: alakzat árnyalásból.

3.2. Első törvény: kapcsolat a normálvektor, a kameraparaméterek és az affin transzformáció között

Az első törvényszerűség – melyet Molnár József és Csetverikov Dmitrij közös munkájukban publikáltak először 2014-ben [113], – kimondja, hogy az affin paraméterek meghatározhatóak a vetítési függvények gradienseinek és a felületi normálvektornak segítségével. Mindezt az alábbi alakban lehet formálisan leírni:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \frac{1}{\nabla\Pi_u^{1\,T}\left[\mathbf{n}\right]_\times \nabla\Pi_v^1} \begin{bmatrix} \nabla\Pi_u^{2\,T}\left[\mathbf{n}\right]_\times \nabla\Pi_v^1 & \nabla\Pi_u^{1\,T}\left[\mathbf{n}\right]_\times \nabla\Pi_u^2 \\ \nabla\Pi_v^{2\,T}\left[\mathbf{n}\right]_\times \nabla\Pi_v^1 & \nabla\Pi_u^{1\,T}\left[\mathbf{n}\right]_\times \nabla\Pi_v^2 \end{bmatrix}, \tag{11}$$

ahol a felső index jelzi a képszámot, illetve a Π_u vetítőfüggvény a vízszintes, Π_v pedig a függőleges koordinátához tartozik. Ezeknek a vetítőfüggvénynek a gradienseit kell venni a térbeli koordináták szerint, ezért ezek a gradiensek háromdimenziós vektorok.

Noha a 22. egyenlet először [113]-ben jelent meg, a levezetést talán kicsit közérthetőbb alakra módosítva mi is publikáltuk [15]. A fenti egyenletek általános kameramodellek esetén igazak, de munkánkban [15] azt is megmutattuk, hogy a gyakorlat számára legfontosabb lyukkamera esetében hogyan konkretizálódnak az összefüggések. Ilyenkor a vetítést a 3×4-es

$$\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix}$$

projekciós (vetítő) mátrixszal írjuk le, a vetítő függvények pedig ekkor a következőképpen adhatók meg:

$$\Pi_u = \frac{1}{s}\left(P_{11}X + P_{12}Y + P_{13}Z + P_{14}\right), \tag{12}$$

$$\Pi_v = \frac{1}{s}\left(P_{21}X + P_{22}Y + P_{23}Z + P_{24}\right), \tag{13}$$

ahol $s = P_{31}X + P_{32}Y + P_{33}Z + P_{34}$ a projektív mélység, amellyel a homogén osztást el kell végezni.

A gradiensek ebben az esetben így alakulnak:

$$\nabla\Pi_u = \tfrac{1}{s}\begin{bmatrix} P_{11} + uP_{31} \\ P_{12} + uP_{32} \\ P_{13} + uP_{33} \end{bmatrix}, \qquad \nabla\Pi_v = \tfrac{1}{s}\begin{bmatrix} P_{21} + vP_{31} \\ P_{22} + vP_{32} \\ P_{23} + vP_{33} \end{bmatrix}.$$

A későbbiekben ezeket az egyenleteket használjuk normálvektorok becslésére.

Mivel ez a levezetés alapvetően társszerzőnk, Molnár József érdeme, az első törvényt a tézisek között nem szerepeltettem, viszont a normálvektorbecslés alapját adja, ezért itt mindenképpen szerettem volna megismertetni vele az Olvasót.

3.3. Második törvény: kapcsolat a homográfia és az affin transzformáció között

Amennyiben síkfelületeket vizsgálunk a kamerákkal, a síkfelület pontjai között az úgynevezett homográfiával lehet leírni a kapcsolatot [62]. Ebben az esetben, a homogén osztást leszámítva, lineáris az összefüggés:

$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} \sim \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}. \tag{14}$$

A kétdimenziós koordinátákat a homogén osztás elvégzésével fejezhetjük ki:

$$\begin{aligned} u_2 &= \tfrac{H_{11}u_1+H_{12}u_2+H_{13}}{H_{31}u_1+H_{32}u_2+H_{33}}, \\ v_2 &= \tfrac{H_{21}u_1+H_{22}u_2+H_{23}}{H_{31}u_1+H_{32}u_2+H_{33}}. \end{aligned} \tag{15}$$

Az affin paraméterek segítségével a transzformációk elsőrendű közelítését lehet leírni, magukat az affin paramétereket ezért a homográfiából származtatott transzformációk parciális deriváltjaiból lehet meghatározni. Például:

$$a_{11} = \frac{\partial \frac{H_{11}u_1+H_{12}u_2+H_{13}}{H_{31}u_1+H_{32}u_2+H_{33}}}{\partial u_1} = \frac{H_{11}\left(H_{31}u_1+H_{32}u_2+H_{33}\right) - H_{31}\left(H_{11}u_1+H_{12}u_2+H_{13}\right)}{\left(H_{31}u_1+H_{32}u_2+H_{33}\right)^2} = \\ = \frac{H_{11}-H_{31}u_2}{\left(H_{31}u_1+H_{32}u_2+H_{33}\right)} = \frac{H_{11}-H_{31}u_2}{s},$$

ha bevezetjük az $s = H_{31}u_1 + H_{32}v_1 + H_{33}$ jelölést.

Hasonlóan megkaphatjuk a mások három lineáris affin paramétert is. A négy paraméter kompakt formában így írható:

$$\begin{aligned} a_{11} &= \tfrac{\partial u_2}{\partial u_1} = \tfrac{H_{11}-H_{31}u_2}{s}, \quad a_{12} = \tfrac{\partial u_2}{\partial v_1} = \tfrac{H_{12}-H_{32}u_2}{s}, \\ a_{21} &= \tfrac{\partial v_2}{\partial u_1} = \tfrac{H_{21}-H_{31}v_2}{s}, \quad a_{22} = \tfrac{\partial v_2}{\partial v_1} = \tfrac{H_{22}-H_{32}v_2}{s}. \end{aligned} \tag{16}$$

A második törvény azt mondja ki, hogy négy új egyenletet lehet felírni, amelyek a homográfiát és az affin paramétereket kötik össze. A 14. összefüggésből további két egyenletet tudunk felírni a homogén osztás elvégzése után. *Az affin paraméterek alkalmazásának egyértelműen nagy előnye, hogy ha magát a homográfiát szeretnénk kiszámítani, akkor újabb négy egyenlettel bővül a becsléshez alkalmazott rendszer a korábban alkalmazott kettő egyenlethez [62] képest.*

Érdekesség, hogy a 16-es sorszámú egyenletekben a 3×3-as homográfia mátrixnak az összes eleme szerepel az új összefüggésben, kivéve a homográfia mátrixban az utolsó oszlop első két elemét. Ezt az elemet kizárólag a 14. összefüggés figyelembe vételével lehet megbecsülni, tehát legalább egy összefüggést ismerni kell a pontok helyére is, nem elég kizárólag affin transzformációkat használni.

Bár ezt a törvényt 2016-ban közösen publikáltuk [16], a levezetést társszerzőnk, Molnár József készítette, ezért ezt nem tekinthetem saját eredményemnek. A későbbiekben ebből a törvényből homográfia-becslőt készítettünk, amiben már az én hozzájárulásomat is jelentősnek ítélem.

3.4. Harmadik törvény: kapcsolat az alapvető/lényegi mátrixok és az affin transzformáció között

Az affin transzformációk alkalmazására egy harmadik törvényt is ki lehet mondani, amely ezúttal a fundamentális mátrixot köti össze az affin transzformációkkal. A 7. ábra mutatja meg a levezetés alapelvét. Adott két kép, két kiválasztott ponttal és a körülöttük levő minták változását meghatározó lokális affin transzformációval. A két képen a középpontokat $\mathbf{p}_1$-el és $\mathbf{p}_2$-vel jelöljük.

A fundamentális mátrix segítségével meghatározhatjuk az $\mathbf{l}_1$ és $\mathbf{l}_2$ epipoláris egyeneseket. Az ezekre merőleges normálirányok skálázásának felhasznnálásával fel lehet írni az alábbi vektoros egyenletet:

$$\mathbf{A}^{\mathbf{T}}\hat{\mathbf{F}}\begin{bmatrix}\mathbf{p}_1\\1\end{bmatrix} = -\tilde{\mathbf{F}}^T\begin{bmatrix}\mathbf{p}_2\\1\end{bmatrix}, \tag{17}$$

ahol $\tilde{\mathbf{F}}$ és $\hat{\mathbf{F}}$ a fundamentális mátrix "csonkított" változata: az eredeti mátrixból az utolsó sort, illetve oszlopot letöröljük. Ezáltal újabb egyenleteket kapunk, hiszen a 24. egyenlet kétdimenziós, tehát két egyenletre bontható a dimenziók szerint. Ha csak a $\mathbf{p}_1$ és $\mathbf{p}_2$ pozíciókat használjuk, a 8. összefüggés alapján egy egyenletünk van, az affin paraméterek figyelembevételével további két egyenletet tudunk felírni.

Bizonyítás.

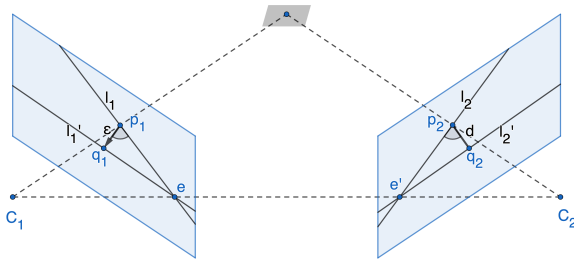
Amennyiben adott az $\mathbf{F}$ fundamentális mátrix, és a két képen ismerjük az egymásnak megfelelő $\mathbf{p}_1$ és $\mathbf{p}_2$ pontokat (homogén alakban), a megfelelő epipoláris egyenesek implicit paramétereit könnyen ki lehet számolni: $\mathbf{l}_2 = \mathbf{F}\mathbf{p}_1$ és $\mathbf{l}_1 = \mathbf{F}^T\mathbf{p}_2$, ahol $\mathbf{l}_1$ és $\mathbf{l}_2$ is háromdimenziós vektorok. Az egyenesekre merőleges normálok irányát a vektorok első két koordinátája adja meg. A két koordinátára itt a $\hat{\mathbf{l}}_1$ és a $\hat{\mathbf{l}}_2$ jelöléseket alkalmazzuk.

A merőlegesek irányvektorait illik egységvektorként kezelni. A két egységhosszúságú normál az alábbi alakban kapható meg:

$$\mathbf{n}_1 = \frac{\hat{\mathbf{l}_1}}{\sqrt{\hat{\mathbf{l}_1}^T\hat{\mathbf{l}_1}}} = \frac{\tilde{\mathbf{F}}^T\begin{bmatrix}\mathbf{p}_2\\1\end{bmatrix}}{\left\|\tilde{\mathbf{F}}^T\begin{bmatrix}\mathbf{p}_2\\1\end{bmatrix}\right\|_2}, \tag{18}$$

$$\mathbf{n}_2 = \frac{\hat{\mathbf{l}_2}}{\sqrt{\hat{\mathbf{l}_2}^T\hat{\mathbf{l}_2}}} = \frac{\hat{\mathbf{F}}\begin{bmatrix}\mathbf{p}_1\\1\end{bmatrix}}{\left\|\hat{\mathbf{F}}\begin{bmatrix}\mathbf{p}_1\\1\end{bmatrix}\right\|_2}. \tag{19}$$

A hullámjeles $\tilde{\mathbf{F}}$ és kalapos $\hat{\mathbf{F}}$ csonkolt mátrixokat a fundamentális mátrix két felső sorának (hullám), illetve két bal oldali oszlopának(kalap) megtartásával kapjuk meg.



7. ábra. *Az affin transzformáció az epipoláris egyenesekre merőleges irányt képes transzformálni. A képen a d/ϵ arány mutatja meg a skálázást a merőleges irányokban.*

A $\mathbf{p}_1$ és a $\mathbf{p}_2$ pontok az $\mathbf{l}_1$ és $\mathbf{l}_2$ egyenesekre illeszkednek, ezért igaz, hogy $\mathbf{l}_1^T\mathbf{p}_1 = 0$ és $\mathbf{l}_2^T\mathbf{p}_2 = 0$.

Hogyha egy másik $\mathbf{q}_1 = \mathbf{p}_1 + \epsilon\mathbf{n_1}$ pontot is veszünk az első képen, ahol $\mathbf{n}_1$ a fentebb kiszámított normálvektora az $\mathbf{l}_1$ egyenesnek, akkor a megfelelő epipoláris egyenest így kaphatjuk meg, homogén koordinátákat alkalmazva:

$$\mathbf{l}'_2 = \mathbf{F}\begin{bmatrix} \mathbf{p}_1 + \epsilon\mathbf{n_1} \\ 1 \end{bmatrix} = \mathbf{F}\begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} + \mathbf{F}\begin{bmatrix} \epsilon\mathbf{n}_1 \\ 0 \end{bmatrix}.$$

A d távolságot az eredeti $\mathbf{p}_2$ pont és a most megkapott $\mathbf{l}'_2$ egyenes között ki tudjuk fejezni:

$$d = \frac{\begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}^T \mathbf{l}'_2}{\sqrt{\hat{\mathbf{l}_2'}^T \hat{\mathbf{l}_2'}}},$$

ahol $\hat{\mathbf{l}'_2}$ vektor az $\mathbf{l}'_2$ epipoláris egyenes első két koordinátáját tartalmazza. Tehát

$$\hat{\mathbf{l}'_2} = \hat{\mathbf{F}}\left(\begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} + \begin{bmatrix} \epsilon\mathbf{n}_1 \\ 0 \end{bmatrix}\right),$$

azaz balról az $\mathbf{F}$ fundamentális mátrix helyett csak annak az első két sorát tartalmazó $\hat{\mathbf{F}}$ mátrixszal kell a pontot szorozni.

A normáló osztótényezőt az alábbi alakba írhatjuk:

$$\sqrt{\hat{\mathbf{l}_2'}^T \hat{\mathbf{l}_2'}} = \left\|\hat{\mathbf{F}}\left(\begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} + \begin{bmatrix} \epsilon\mathbf{n}_1 \\ 0 \end{bmatrix}\right)\right\|_2.$$

Elemi átalakításokat végezve a d távolság alakját módosíthatjuk:

$$d = \frac{\begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}^T \left(\mathbf{F}\begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} + \mathbf{F}\begin{bmatrix} \epsilon\mathbf{n}_1 \\ 0 \end{bmatrix}\right)}{\left\|\hat{\mathbf{F}}\left(\begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} + \begin{bmatrix} \epsilon\mathbf{n}_1 \\ 0 \end{bmatrix}\right)\right\|_2} = \epsilon\frac{\begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}^T \mathbf{F}\begin{bmatrix} \mathbf{n}_1 \\ 0 \end{bmatrix}}{\left\|\hat{\mathbf{F}}\left(\begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} + \begin{bmatrix} \epsilon\mathbf{n}_1 \\ 0 \end{bmatrix}\right)\right\|_2}.$$

Az első tag a számlálóban azért esett ki, mivel $\mathbf{p}_2$ az $\mathbf{l}_2$ egyenesen van, tehát

$$\begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}^T \mathbf{F}\begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} = 0.$$

Ezért írhatjuk, hogy

$$d = \epsilon\frac{\begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}^T \mathbf{F}\mathbf{n_1}}{\left\|\hat{\mathbf{F}}\left(\begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} + \begin{bmatrix} \epsilon\mathbf{n}_1 \\ 0 \end{bmatrix}\right)\right\|_2}.$$

A merőleges irányok megváltozását a d és ϵ távolságok arányával írhatjuk le. Ha a módosított pontot végtelenül közel visszük az egyeneshez, akkor írhatjuk az arányra, hogy

$$s = \lim_{\epsilon \to 0} \frac{d}{\epsilon} = \lim_{\epsilon \to 0} \frac{\begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}^T \mathbf{F}\mathbf{n_1}}{\left\| \hat{\mathbf{F}} \left(\begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} + \begin{bmatrix} \epsilon \mathbf{n}_1 \\ 0 \end{bmatrix} \right) \right\|_2} = \frac{\begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}^T \mathbf{F}\mathbf{n_1}}{\left\| \hat{\mathbf{F}} \left(\begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} \right) \right\|_2}.$$

Az $\mathbf{n}_1$ normálvektorra a 18. összefüggést behelyettesítve ezt a formulát kapjuk:

$$\frac{\begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}^T \tilde{\mathbf{F}} \frac{\tilde{\mathbf{F}}^T \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}}{\left\| \tilde{\mathbf{F}}^T \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix} \right\|_2}}{\left\| \hat{\mathbf{F}} \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} \right\|_2} = \frac{\begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}^T \tilde{\mathbf{F}}\tilde{\mathbf{F}}^T \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}}{\left\| \tilde{\mathbf{F}}^T \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix} \right\|_2 \left\| \hat{\mathbf{F}} \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} \right\|_2} =$$

$$\frac{\left\| \tilde{\mathbf{F}}^T \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix} \right\|_2^2}{\left\| \tilde{\mathbf{F}}^T \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix} \right\|_2 \left\| \hat{\mathbf{F}} \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} \right\|_2} = \frac{\left\| \tilde{\mathbf{F}}^T \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix} \right\|_2}{\left\| \hat{\mathbf{F}} \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} \right\|_2}. \tag{20}$$

Ha egy affin transzformáció az egyenesek irányait jól transzformálja, akkor az egyenesekre merőleges irányra igaz [169], hogy $\mathbf{A}^T \mathbf{n}_2 = s\mathbf{n}_1$, ahol az s skála egy pozitív valós szám. Ez a skála kiküszöbölhető, ha a normálvektorokból egységvektort készítünk. Ekkor a 20 egyenletből a skálát behelyettesítve, és egységvektorokat készítve a normálvektorokból igaz lesz, hogy

$$\mathbf{A^T} \frac{\hat{\mathbf{F}} \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix}}{\left\| \hat{\mathbf{F}} \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} \right\|_2} = \frac{\left\| \tilde{\mathbf{F}}^T \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix} \right\|_2}{\left\| \hat{\mathbf{F}} \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} \right\|_2} \frac{\tilde{\mathbf{F}}^T \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}}{\left\| \tilde{\mathbf{F}}^T \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix} \right\|_2}.$$

A nevezőben és számlálóban is jelen levő tagokkal egyszerűsítve adódik a végső formula:

$$\mathbf{A^T} \hat{\mathbf{F}} \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} = \pm \tilde{\mathbf{F}}^T \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}. \tag{21}$$

Itt az előjel mindössze azért bizonytalan, mert nem tudjuk, hogy a normálvektor az egyenes melyik oldalára mutat. A gyakorlati kamera konfigurációkban ez az előjel negatív.

□

Gyakorlati alkalmazás esetén fontos tény, hogy a fundamentális mátrixnak az összes eleme szerepel az új összefüggésben, kivéve a jobb alsó, azaz $(3,3)$-as indexű elemet. Azt az elemet a 8. egyenletből lehet megkapni., tehát szükség van a pontok helyére is, nem elég csupán a lineáris részét használni az affin transzformációnak.

3.5. Összefoglalás

Ebben a fejezetben megmutattam azt a három törvényt, amelyek a lokális affin transzformációkat a kétképes (sztereó) látással összekötik. A következő fejezetben ezekre a törvényszerűségre alapozott algoritmusokat fogok bemutatni.

– Az első törvény az affin transzformáció lineáris részére ad egy algebrai alakot, amely a felületi normálvektorokból és a vetítési egyenletek térbeli pozíció szerinti gradienseiből számítja ki a transzformációt. Az általános formula tetszőleges vetítési modellre működik.

– A második törvény feltételezi, hogy egy síkfelületről két képet készítünk, és négy egyenletet ad a törvény, amelynek segítségével az affin transzformáció lineáris részét ki lehet számítani a homográfia ismeretében. A homográfia elemeinek segítségével megmutattam azokat a törtkifejezéseket, amelyek a négy lineáris affin paramétert meghatározzák. A homográfia kilenc eleméből hét szerepel ezekben a törtekben.

– A harmadik törvény a fundamentális mátrix és az affin transzformáció közötti kapcsolatra ad egy kétdimenziós egyenletet. A formulát az epipoláris egyenesek közötti távolságok alapján vezettem le. Az összefüggésben a fundamentális mátrix kilenc eleméből nyolc szerepel.

Ismereteim szerint ezt a három törvényszerűséget én foglaltam össze először, erről "Tutorial" előadást tartottam a CVPR2022 [8] és a 3DV2022 konferenciákon. A törvények összefoglalását ezért tudományos eredménynek tekintem, ebből mondom ki az első tézist. A három törvény közül azonban csak az utolsót tekintem saját eredményemnek, ennek alapján fogalmazom meg a második tézisemet.

I.1. tézis. Affin transzformációk törvényészerűségei[18, 34, 19].

Kutatómunkánk és a minket megelőző szakemberek munkássága alapján három törvényben fogalmaztam meg az affin transzformációkkal kapcsolatos összefüggéseket.

1. Az első törvény a felületi normálvektor, a kameraparaméterek, a vetítési függvény és a vetületi középpontok közötti összefüggést írja le:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \frac{1}{\nabla\Pi_u^{1\,T}\left[\mathbf{n}\right]_\times \nabla\Pi_v^1} \begin{bmatrix} \nabla\Pi_u^{2\,T}\left[\mathbf{n}\right]_\times \nabla\Pi_v^1 & \nabla\Pi_u^{1\,T}\left[\mathbf{n}\right]_\times \nabla\Pi_u^2 \\ \nabla\Pi_v^{2\,T}\left[\mathbf{n}\right]_\times \nabla\Pi_v^1 & \nabla\Pi_u^{1\,T}\left[\mathbf{n}\right]_\times \nabla\Pi_v^2 \end{bmatrix}, \quad (22)$$

ahol a gradiens operátor (∇) jelöli a vetítési függvény gradiensét a térbeli koordináta szerint[9], $\mathbf{n}$ pedig a felületi normálvektor. A kapott formula véleményem szerinti rendkívül nagy értéke, hogy bármilyen vetítőfüggvény esetén működik, nem korlátozódik egyetlen kameratípusra sem. A fenti alapegyenlet tehát egy általános összefüggést ad, az alkalmazott kameratípusok közötti különbség a Π vetítőfüggvényekben jelenik meg.

[8] https://cvpr22-affine-tutorial.com/ oldalon elérhető a tutorial, a másfél órás előadásom címe 'Affine Correspondences in Stereo Vision'.

[9] Az alsó index a vízszintes/függőleges irányt, a felső index a kép számát jelöli.

2. A második törvény megmutatja, hogy milyen összefüggés képezhető lyukkamera esetén a $\mathbf{H}$ homográfia és a két képen egymásnak megfelelő $[u_1 \quad v_1]^T$ és $[u_2 \quad v_2]^T$ pixelkoordináták között, ha a lokális felület egy síkról származik. A parciális deriváltak alapján összesen négy összefüggést lehet felírni:

$$\begin{array}{ll} a_{11} = \frac{\partial u_2}{\partial u_1} = \frac{H_{11} - H_{31}u_2}{s}, & a_{12} = \frac{\partial u_2}{\partial v_1} = \frac{H_{12} - H_{32}u_2}{s}, \\ a_{21} = \frac{\partial v_2}{\partial u_1} = \frac{H_{21} - H_{31}v_2}{s}, & a_{22} = \frac{\partial v_2}{\partial v_1} = \frac{H_{22} - H_{32}v_2}{s}. \end{array} \tag{23}$$

3. A harmadik törvény a következő, I.2. altézisben megfogalmazott összefüggés a fundamentális (alapvető) mátrix és a lokális affin transzformáció elemei között.

I.2. tézis. Lineáris kapcsolat az epipoláris geometria és az affin transzformációk között [18, 34, 19].

Amennyiben adott egy színtérről két, lyukkamerával felvett kép, és a két kép között az $\mathbf{F}$ fundamentális mátrix teremt kapcsolatot, akkor az affin transzformáció és a fundamentális mátrix között az alább vektoros (kétdimenziós) egyenletet lehet megadni:

$$\mathbf{A}^{\mathbf{T}}\hat{\mathbf{F}} \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} = -\tilde{\mathbf{F}}^T \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}, \tag{24}$$

ahol $\tilde{\mathbf{F}}$ és $\hat{\mathbf{F}}$ a fundamentális mátrix "csonkított" változatai, melyeket úgy kapunk, hogy az eredeti $\mathbf{F}$ mátrixból az utolsó sort, illetve az utolsó oszlopot töröljük. $\mathbf{p}_1$ és $\mathbf{p}_2$ a két képen az egymásnak megfelelő pontok homogén koordinátákkal megadva, ezen pontok körül értelmezzük az $\mathbf{A}$ affin transzformációt.

Érdekesség, hogy a harmadik törvényt 2016-ban publikáltuk [34], de a 24. összefüggésben a két oldal közötti arányról még nem tudtuk, hogy mínusz egy, ezt a tényt egy évvel később, 2017-ben sikerült bebizonyítani [19]. Ezt a bizonyítást teljes egészében magaménak érzem, azzal a megjegyzéssel, hogy az egyszerűsített alak doktoranduszom, Baráth Dániel sejtése volt, de a bizonyítást nekem sikerült megalkotni először.

4. II. Téziscsoport: algoritmusok lokális affin tanszformáció felhasználásához

Az előző fejezetben az affin transzformációval kapcsolatosan három törvényszerűséget fogalmaztam meg. Ebben a fejezetben ezekre a törvényekre épített algoritmusokat mutatok be, melyek a gemetria alapú számítógépes látás alapproblémáira nyújtanak egy-egy megoldást.

4.1. Optimális normálvektorbecslés

Az első törvény kimondja, hogy az affin paraméterek meghatározhatóak a felületi normálvektor és a vetítőfüggvény gradienseinek ismeretében. A gondolatmenetet meg is lehet fordítani: ha ismert a kamera a külső és a belső paramétereivel egyetemben, akkor az affin transzformációk meghatározzák a felületi normálvektort. A feladat ábrázolását a 8. ábrán lehet megtekinteni.

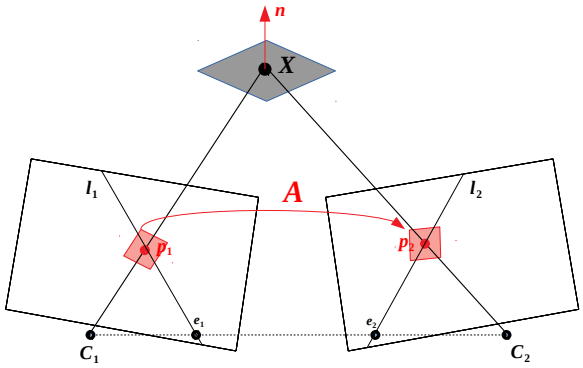
Az első törvényszerűség (22. formula) alapján az affin transzformációk értékére négy egyenletünk van. Ezeknek felhasználásával legkisebb négyzetes értelemben meghatározható egy hibafüggvény, amelynek optimalizálására javaslatot tettünk. A feladat matematikai alakja az alábbi:

$$\mathbf{n} = \arg_{\mathbf{n}} \min \sum_{i=1}^{2} \sum_{j=1}^{2} \left(\frac{\mathbf{n}^T \mathbf{w}_{ij}}{\mathbf{n}^T \mathbf{w}_c} - a_{ij} \right)^2 , \tag{25}$$

ahol

$$\begin{aligned} \mathbf{w}_{11} &= (\nabla \Pi_v^1 \times \nabla \Pi_u^2) , \quad \mathbf{w}_{12} = (\nabla \Pi_u^2 \times \nabla \Pi_u^1) , \\ \mathbf{w}_{21} &= (\nabla \Pi_v^1 \times \nabla \Pi_v^2) , \quad \mathbf{w}_{22} = (\nabla \Pi_v^2 \times \nabla \Pi_u^1) , \\ \mathbf{w}_c &= (\nabla \Pi_u^1 \times \nabla \Pi_v^1) . \end{aligned} \tag{26}$$

Könnyű belátni, hogy ennek a négyzetes hibafüggvénynek a minimuma nem függ a háromdimenziós normálvektor irányától, csak annak nagyságától. Ezért a normálvektor becslése valójában egy kétparaméteres probléma.



8. ábra. *Az affin transzformáció esetén a pirossal jelzett felületi normálvektor meghatározható, amennyiben a kamera kalibrált, azaz ismerjük a vetítőfüggvényt.*

A két paraméter szerinti deriváltak görbéket határoznak meg a síkon, ha a két paramétert tekintjük a kétdimenziós tengelyeknek. A megoldás pedig a görbék metszéspontjaiban van.

Először megmutattuk, hogy síkban kvadratikus görbék metszéspontját kell keresni, melyek egy negyedfokú polinom gyökén keresztül számíthatóak [15, 9]. Később azt is bebizonyítottuk, hogy a másodfokú görbék speciálisak, ezért elég egy harmadfokú polinom gyökeit számolni [110], azaz lényegesen gyorsabban megkapható az eredmény. 2023-ban Lóczi Lajos kollégánk közreműködésével sikerült még egyszerűbb megoldást adni az optimális normálvektorbecslésre, két elsőfokú polinom hányadosaként [56].

4.2. Kamera kalibráció

Az első affin transzformációkra vonatkozó törvény, amelyet a 22. összefüggés ír le, azt is kimondja, hogy a vetítő függvény gradiensei és a normálvektorok hogyan függnek össze. Miután a gradiens a kameraparaméterektől függ, az affin transzformációk ismerete esetén egyenleteket tudunk felírni a paraméterekre. 2016-os munkánkban [42] megmutattuk Eichhardt Iván (akkor még) doktorandusz kolléga közreműködésével [10], hogyan lehet ismert normálvektorok segítségével a kameraparaméterek becslését finomítani. Ennek az eredménynek inkább elméleti jelentősége van, elsősorban a III. tételcsoportban ismertetett kötegelt behangolást használó numerikus finomításnál használtuk a kapott összefüggéseket.

Azért kevésbé jelentős gyakorlati szempontból ez a tézis, mert a felületi normálvektorok csak jól meghatározható kalibrációs objektumok esetén ismertek, szabadabb formájú felületek rekonstrukciójakor ki kell számítani, de ehhez a számításhoz már szükség van a kamera paraméterekre.

Az első törvényből ismert összefüggés:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \frac{1}{\nabla\Pi_u^1 \left[n\right]_\times \nabla\Pi_v^1} \left[\begin{array}{c|c} \nabla\Pi_u^2 \left[n\right]_\times \nabla\Pi_v^1 & \nabla\Pi_u^1 \left[n\right]_\times \nabla\Pi_u^2 \\ \hline \nabla\Pi_v^2 \left[n\right]_\times \nabla\Pi_v^1 & \nabla\Pi_u^1 \left[n\right]_\times \nabla\Pi_v^2 \end{array}\right].$$

A jobb oldalon a kiemelt skalár nevezőjével beszorozhatjuk az egyenletet, és mivel $\mathbf{a}^T[\mathbf{n}]_x\mathbf{b} = \mathbf{n}^T\left(\mathbf{b}\times\mathbf{a}\right)$, írhatjuk, hogy

$$\left(\nabla\Pi_u^1 \left[n\right]_\times \nabla\Pi_v^1\right) \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{n}^T\left(\nabla\Pi_v^1 \times \nabla\Pi_u^2\right) & \mathbf{n}^T\left(\nabla\Pi_u^2 \times \nabla\Pi_u^1\right) \\ \hline \mathbf{n}^T\left(\nabla\Pi_v^1 \times \nabla\Pi_v^2\right) & \mathbf{n}^T\left(\nabla\Pi_v^2 \times \nabla\Pi_u^1\right) \end{array}\right]$$

A bal oldalon a mátrix elemeire bevezethetjük a $m_{ij} = a_{ij}\left(\nabla\Pi_u^1 \left[n\right]_\times \nabla\Pi_v^1\right)$ jelölést. A jobb oldalon pedig a vektoriális szorzást mátrixos alakra módosíthatjuk, így igaz lesz, hogy

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{n}^T \left[\nabla\Pi_v^1\right]_\times \nabla\Pi_u^2 & -\mathbf{n}^T \left[\nabla\Pi_u^1\right]_\times \nabla\Pi_u^2 \\ \hline \mathbf{n}^T \left[\nabla\Pi_v^1\right]_\times \nabla\Pi_v^2 & -\mathbf{n}^T \left[\nabla\Pi_u^1\right]_\times \nabla\Pi_v^2 \end{array}\right].$$

Ez pedig négy egyenletet jelent:

$$\begin{aligned} m_{11} &= \mathbf{n}^T \left[\nabla\Pi_v^1\right]_\times \nabla\Pi_u^2, \quad m_{12} = -\mathbf{n}^T \left[\nabla\Pi_u^1\right]_\times \nabla\Pi_u^2, \\ m_{21} &= \mathbf{n}^T \left[\nabla\Pi_v^1\right]_\times \nabla\Pi_v^2, \quad m_{22} = -\mathbf{n}^T \left[\nabla\Pi_u^1\right]_\times \nabla\Pi_v^2. \end{aligned}$$

[10] Ivánnal egy kutatócsoportban dolgoztunk már akkoriban is, mindkettőnk doktori munkájának témavezetője Csetverikov Dmitrij volt.

A gradiensek:

$$\nabla\Pi_u = \frac{1}{s}\begin{bmatrix} P_{11}+uP_{31} \\ P_{12}+uP_{32} \\ P_{13}+uP_{33} \end{bmatrix} \rightarrow s\nabla\Pi_u = \begin{bmatrix} P_{11}+uP_{31} \\ P_{12}+uP_{32} \\ P_{13}+uP_{33} \end{bmatrix} = \hat{\mathbf{p}}_1 + u\hat{\mathbf{p}}_3,$$

ahol a $\hat{\mathbf{p}}_i$ a $\mathbf{P}$ mátrix i-dik oszlopát jelöli. Teljesen hasonlóan:

$$\nabla\Pi_v = \frac{1}{s}\begin{bmatrix} P_{21}+vP_{31} \\ P_{22}+vP_{32} \\ P_{23}+vP_{33} \end{bmatrix} \rightarrow s\nabla\Pi_v = \begin{bmatrix} P_{21}+vP_{31} \\ P_{22}+vP_{32} \\ P_{23}+vP_{33} \end{bmatrix} = \hat{\mathbf{p}}_2 + v\hat{\mathbf{p}}_3.$$

Az összefüggésekben $s = \tilde{\mathbf{p}}_3^T\mathbf{X}$, ahol a hullám operátor azt jelenti, hogy a $\mathbf{P}$ mátrixnak a harmadik sorát (vektort) kell venni. Az első kamerára az s^1 projektív mélység ismert, hiszen minden adott, de a második kameránál az is ismeretlennek tekinthető.

Tehát behelyettesítve és linearizálva az alábbi négy egyenletet kapjuk:

$$\begin{aligned} m_{11}s^2\tilde{\mathbf{p}}_3^{2T}\mathbf{X} &= \mathbf{n}^T\left[\nabla\Pi_v^1\right]_\times\left(\hat{\mathbf{p}}_1^2 + u^2\hat{\mathbf{p}}_3^2\right), \\ m_{12}s^2\tilde{\mathbf{p}}_3^{2T}\mathbf{X} &= -\mathbf{n}^T\left[\nabla\Pi_u^1\right]_\times\left(\hat{\mathbf{p}}_1^2 + u^2\hat{\mathbf{p}}_3^2\right), \\ m_{21}s^2\hat{\mathbf{p}}_3^{2T}\mathbf{X} &= \mathbf{n}^T\left[\nabla\Pi_v^1\right]_\times\left(\hat{\mathbf{p}}_2^2 + v^2\hat{\mathbf{p}}_3^2\right), \\ m_{22}s^2\tilde{\mathbf{p}}_3^{2T}\mathbf{X} &= -\mathbf{n}^T\left[\nabla\Pi_u^1\right]_\times\left(\hat{\mathbf{p}}_2^2 + v^2\hat{\mathbf{p}}_3^2\right). \end{aligned} \tag{27}$$

Ez pedig egy homogén lineáris egyenletrendszer a második kép $\mathbf{P}^2$ projekciós mátrixának az elemeire nézve, ahogyan az el is várható. Fontos megjegyzés, hogy a projekciós mátrixnak négy oszlopa van összesen, ezek közül a negyedik nem szerepel az affin összefüggésekben, azt az oszlopot a vetületek koordinátáinak a segítségével lehet meghatározni.

Minimális eset. A projektív mátrixnak így összesen kilenc paraméterét lehet megbecsülni. Két affin transzformációból a 27. formulák alapján nyolc egyenletet tudunk felírni, ha ezekhez még a pontok helyei is hozzájönnek, azok tovább $2+2$ egyenletet adnak, tehát összesen 12 egyenletünk van a projekciós mátrix 11 szabad paraméterére.

A korábbi munkánkban [42], amelyben az eredeti módszert ismertettük, kétféle algoritmust javasoltunk:

1. CLOSED. A 27. egyenleteknek megfelelően a homogén lineáris egyenletrendszer sajátérték-feladatként megoldható, ahogyan azt a B. függelékben megmutatom.

2. ALTER. A projektív mélység minden egyes térbeli pontra ismeretlenként bevezethetjük: $s^2 = \tilde{\mathbf{p}}_3^{2T}\mathbf{X}$. Ekkor a 27. egyenlet bal oldalán a projektív mélység szerepel ismeretlenként, a jobb oldalon pedig a kamera paraméterei. Kézenfekvő megoldás, hogy válasszuk szét a projekciós mátrix elemeit és a mélységeket, és felváltva optimalizáljuk, mert két lineáris problémára esik szét. A B. függelékben mutatom meg, hogyan lehet lineáris túlhatározott rendszereket megoldani. Az iteráció elején kezdeti értéket kell beállítani, amit a DLT algoritmus szolgáltat kizárólag pontmegfelelések segítségével.

A fenti eljárásoknak hátránya, hogy algebrai hibát minimalizálnak, nem geometriait, annak érdekében, hogy a probléma lineáris maradjon. Ezért numerikus eljárások

alkalmazására van szükség a geometriai hiba csökkentésére. Ezzel szintén foglalkoztunk kollégámmal, de a hely szűkössége miatt a részleteket itt nem tudom közölni, 2016-os munkánkat [42] elolvasva tájékozódhat az Olvasó.

Szintetikus tesztek.

A szintetikus tesztek során különböző normálvektorokat generáltunk, amelyek egyértelműen meghatározzák a térbeli elhelyezkedéssel együtt az affin paraméterekkel. Először egy sztereó képpárt generáltunk, a megfelelő projekciós mátrixokat véletlenszerűen állítottuk elő. Ezután 3D objektumokat is generáltunk, melynek felületi pontjait igyekeztünk nagyjából egyenletesen mintavételezni. Kétféle tesztobjektummal dolgoztunk:

1. *Gömb:*A szintetikus gömb az alapértelmezett normálisokkal együtt a 9. ábra bal oldali ábráján látható. A vizsgált módszerek bemenetéhez 72 különböző felszíni pontot mintavételeztük. A normálvektor a gömb alakjából triviálisan kiszámítható.

2. *Kocka*: A szintetikus kocka a 9. ábra jobb oldali képén látható. A kocka minden oldalát egyenletesen mintavételeztük: oldalanként $7 \times 7 = 49$ mintavételt vettünk figyelembe. Ezért minden egyes futtatásnál $49 \cdot 6 = 294$ virtuális helyet vettünk figyelembe.

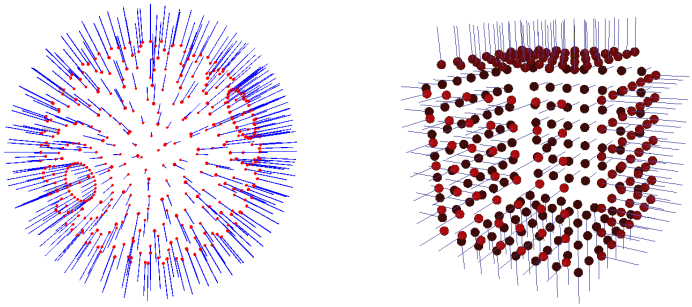
A sztereó képek közötti affin paramétereket a következőképpen számoltuk ki: (1) A gömb/doboz érintő síkját a modell alapján meghatároztuk, ezután (2) a két képre vetítettük a kamera paramétereinek segítségével. (3) A vetítésből kiszámoltuk az érintősíkhoz tartozó homográfiát. (4) Az affin transzformáció az elsőrendű közelítése a homográfiának a megadott adott helyeken. (5) Végül a pontok vetületeihez és az affin transzformációkhoz is zajt adunk.

Három módszert hasonlítottunk össze: a (normalizált) **DLT** algoritmust [62], a javasolt iteratív módszert (**ALTER**) és a zárt alakú megoldás alapján számított eljárást (**CLOSED**). A kalibrálás minősége a becsült vetületi mátrix Frobenius-normája alapján számítjuk ki az alapigazsághoz viszonyítva. (Megjegyzendő, hogy csak a bal oldali 3×3 almátrixot vesszük figyelembe, és a mátrixokat ugyanúgy normalizáljuk.) Minden egyes tesztesetet 20 alkalommal megismételtünk. Az ábrákon majd a hibák átlaga és mediánja látható.

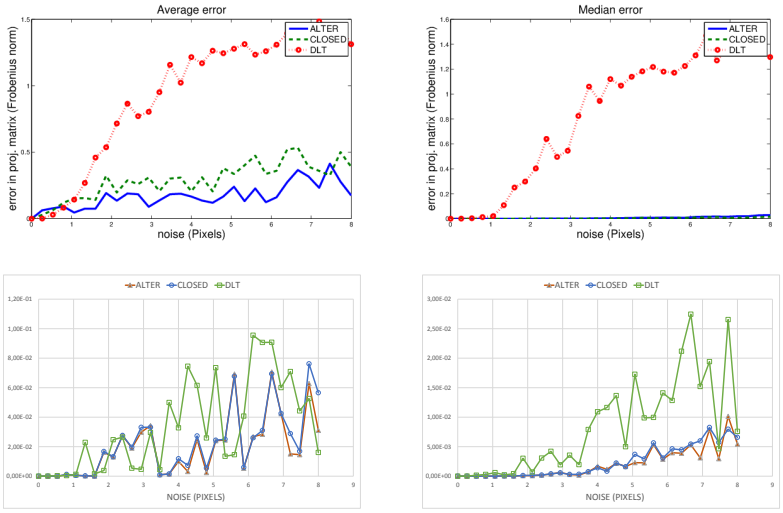
A 10. ábrán látható teszteredmények egyértelműen azt mutatják, hogy a normálvektorokat figyelembe vevő kalibrációs algoritmusok (ALTER és CLOSED) jelentősen felülmúlják a klasszikus DLT-módszert. Ez mind a gömbre, mind a kockára érvényes. A hibák átlagának és mediánjának a görbéje nagyon hasonló, ahogyan az várható is. A grafikonok azt is szemléltetik, hogy az ALTER és a CLOSED módszerek minősége megközelítőleg azonos. Ez a várható viselkedés, mivel a módszerek ugyanazokat a költségfüggvényeket próbálják minimalizálni, csak az eredményvektorok normalizálása különbözik.

A szükséges számítási időt a 11. ábrán láthatja az Olvasó. Egyértelmű, hogy a CLOSED algoritmus használata javasolt, mivel az lényegesen gyorsabb, mint az iteratív algoritmus. A DLT algoritmus ugyanakkor mindkét saját módszernél gyorsabb. Ez nem meglepő, hiszen az együtthatómátrix jóval kevesebb elemet tartalmaz, a sajátérték-felbontás elvégzése pedig gyorsabb kisebb mátrixok esetén.

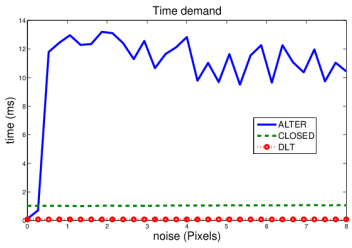
Valós tesztek.



9. ábra. *Szintetikus gömb (bal) és kocka (jobb) felületi normálvektorokkal.*



10. ábra. *A kamera paraméter becslések eredménye szimulált tesztadatokon . A javasolt módszerek lényegesen gyorsabbak az alapnak tekintett DLT módszerhez képest. Bal oldal: átlagos hiba. Jobb oldal: a hibák mediánja. Felső sor: Eredmények gömbfelületeken. Alul: Eredmények a kocka oldalain.*

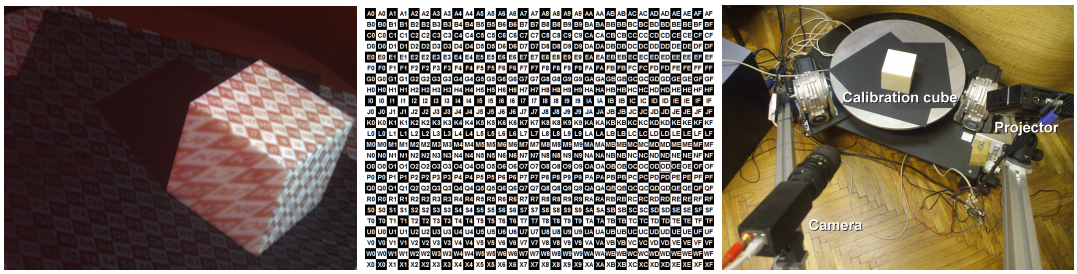


11. ábra. *A módszerek időigénye. A javasolt CLOSED algoritmus lényeges gyorsabb az ALTER módszernél, hiszen nem tartalmaz iterációt. Viszont a javasolt algoritmusok lassabbak a DLT-nél: kompromisszumot kel kötni a számítási idő és a minőség között.*

A javasolt kalibrációs módszert sikeresen alkalmaztuk egy forgótányér-alapú projektor-kamera rendszer kalibrálására. A berendezés képe a 12. ábra alsó képén látható.

A javasolt kalibrációs módszer újdonsága, hogy *a teljes eljáráshoz csak egy kamerára és egy másik vetített képre van szükség, és mindkettőhöz a kamera és a kivetítő tetszőleges perspektivikus vetületeket valósít meg*. A kamera előtt egy kalibrációs kocka van, amelyet a projektor megvilágít, ahogy azt a 12. ábra bal felső képén láthatjuk.
Kamera kalibráció. A kocka sarkait manuálisan határoztuk meg a kameraképeken. A kocka méretét ismerjük, ezért a vetítési mátrixot a DLT-algoritmus [62] segítségével meg tudtuk becsülni. Ezután a projekciós mátrixot felbontottuk belső és külső paraméterekre. A külső paraméterek az elforgatást és eltolást reprezentálják.
Projektor kalibráció. A bevezetett módszer szempontjából a fő újítás a vetítő paramétereinek a megbecslése. A projektor vetítési modellje hagyományos lyukkamerának felel meg, az egyetlen különbség, hogy a kamera a háromdimenziós világot vetíti a képsíkra, a projektor pedig a képet vetíti ki a térbeli világba. De magát a kivetítést ugyanúgy egy 3×4-es vetítő (projektív) mátrix határozza meg.

A javasolt algoritmus képes megbecsülni a bal oldali 3×3-mas részét ennek a vetítő mátrixnak. A módszer végrehajtásához az affin transzformációkat kell ismerni a kivetített minta egyes pontjai és a kamerakép megfelelő része között. A kamera és a vetítő perspektivikus jellege miatt a kivetített minta és a kép között síkhomográfia írja le a pontos kapcsolatot. Az affin paraméterek a homográfia első deriváltjaiból származnak, ahogy azt az előző fejezetben, a második törvény leírásánál megmutattam.



12. ábra. *A javasolt algoritmusunk pontosságát a strukturált fényszkennerünk kalibrálásával ellenőriztük. Balra: Speciális sakktáblamintával megvilágított kalibrációs kocka. Középen: A megvilágításhoz használt minta. Jobbra: A teljes berendezés (kamera+projektor) a kalibrációs kockával.*

Ha a kocka hét sarka ismert a kamerakalibrációból, akkor a három látható sík is ismert a szabályos, egymásra merőleges normálvektorokkal. Ekkor a normálvektorok és a megfelelő affin transzformációk a javasolt kalibrációs egyenletekbe behelyettesíthetőek.

A kivetített minta kódokat tartalmaz, amint az a 12. ábra jobb felső ábráján látható. Ezekre azért volt szükség, mert a sakktábla sarkait kézzel detektáltuk, és a projektor-kamera közötti megfeleltetést a minta segítségével végeztük el.

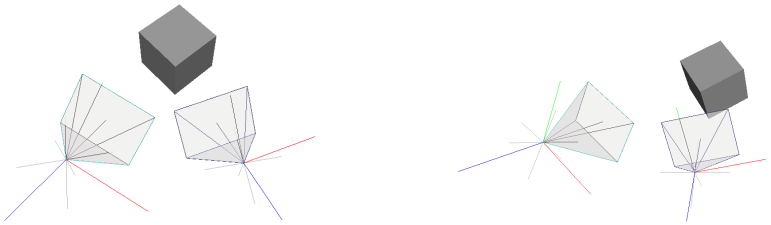
A CLOSED algoritmus megbecsüli a vetítő vetítési mátrixát. A belső paraméterek QR-dekompozícióval [125] nyerhetőek ki. A kapott paraméterek a következőek: vízszintes és függőleges fókusztávolság: $f_u = 1136$, és $f_v = 1068$; nyírás: $s = 59$; döféspont (egészre kerekítve): $[478{,}758]^T$.

Ezek a paraméterek reálisak, mivel a vízszintes és függőleges fókusztávolságok

megközelítőleg azonosak, és a ferdeség lényegesen kisebb, mint a fókusztávolságok. A projektor felbontása 800×600, ezért a döféspont is elfogadható, mivel a vízszintes koordináta a szélesség feléhez közel, a függőleges pedig az alsó határ közelében van.

A projektort szabványos sakktábla-alapú algoritmus [190] segítségével is kalibráltuk, hogy ellenőrizzük az eredmény pontosságát. Az eljárás egy végső numerikus finomítást is tartalmaz. A kalibráláshoz *többször tíz képet* használtunk, ezért az eredménynek lényegesen pontosabbnak kell lennie, mint az előző esetben, ahol egyetlen egy kép szolgáltatott adatot. A sakktáblás kalibráció eredményként a becsült döféspont $[467{,}764]$ volt, míg a vízszintes és függőleges fókusztávolságokra 1454, illetve 1461 jött ki. A kapott eredmények nagyon közel állnak egymáshoz.

A kalibrált elrendezés a 13. ábrán látható. A két gúla a kamerát (balra) és a projektort (jobbra) szimbolizálja. A gúla csúcspontja a fókuszpont, a négyoldalú lapja pedig az érzékelő (szenzor) felületét ábrázolja. A kalibrációs kocka a kép közepén található.



13. ábra. *Rekonstruált (kalibrált) kamera-projektor elrendezés két nézőpontból. A kamerát és a projektort gúlák szemléltetik.*

4.3. Homográfia becslése affin transzformációk felhasználásával

4.3.1. Irodalmi áttekintés

Ahogyan az a sztereó látásban jól ismert[62], két egymásnak megfelelő képrészlet között a homográfia teremt kapcsolatot, ha a megfigyelt térrész pontjai egy síkon vannak, vagy ha a két nézet fókuszpontja ugyanott van. A homográfiát a 3×3-as $\mathbf{H}$ mátrixszal szokás reprezentálni. Ha egy $\mathbf{p}_1 = [u_1 v_1 1]^T$ homogén koordinátás pontot az első képről egy $\mathbf{H}$ homográfia transzformálja a második kép $\mathbf{p}_2 = [u_2 \quad v_2 \quad 1]^T$ pontjára, akkor a kapcsolatot a $\mathbf{p}_2 \sim \mathbf{H}\mathbf{p}_1$ összefüggés írja le.

A megfeleltetés kétdimenziós vonalakra is megadható a következőképpen: $\mathbf{l}' \, \mathbf{H}^{-T} \mathbf{l}$, ahol az egyenesek paramétereit az első és a második képen az $\mathbf{l}$ és $\mathbf{l}'$ háromdimenziós vektorok írják le. Ha egy $\mathbf{p}$ pont a $\mathbf{l}$ egyenesen fekszik, akkor a $\mathbf{p}'$ transzformált helynek a megfelelő $\mathbf{l}'$ transzformált egyenesen kell feküdnie.

Nagyon izgalmas tény, hogy a homográfia fogalma már a múlt század közepén ismert volt [152].

A szakirodalomban számos megoldás létezik a két kép közötti homográfia becslésére, ahogyan azt Agarwalnak és munkatársainak a munkája [4] összefoglalja. Elsőként a legegyszerűbb módszert, a lineáris transzformációra (DLT) visszavezetett [62] eljárást illik megemlítenünk. Ebben az esetben a $\mathbf{H}$ homográfia nyolc ismeretlen paraméterét pontmegfelelések alapján kívánjuk megbecsülni a $\mathbf{p}' \sim \mathbf{H}\mathbf{p}$ alapösszefüg-

gés linearizálásából keletkező túlhatározott egyenletrendszer megoldásával, ahol a $\sim$ operátor a skálázás erejéig vett egyenlőséget jelenti. Maga a linearizálás torzítja a zajt, ezért az eredeti nemlineáris vetületi egyenletekre való optimalizálás pontosabb eredményeket ad. Ez utóbbit numerikus optimalizálási technikákkal lehet elvégezni, mint például a széles körben használt Levenberg-Marquardt [108] optimalizálással. Azonban a lineáris algoritmusok is javíthatók, elsősorban a numerikus problémák elkerülésével, ha először az adatok normalizálását [62] alkalmazzuk.

Kenichi Kanatani[78] javasolt egy módszert a becslési hiba minimalizálására a képtérben, mivel a zaj a képkoordinátákban jelentkezik, nem pedig az absztrakt, magasabb dimenziós algebrai terekben.

A homográfia becslésére természetesen számos más módszer is létezik: vannak vonal alapú [118], kúp alapú [79, 116], kontúr alapú [88] és folt alapú [86] módszerek. Ezeknek a jellemzőknek a megfeleltetése azonban nem olyan egyszerű, mint a pontoké. Manapság már léteznek nagyon hatékony jellegzetes pont detektorok és megfeleltetők [115, 103, 21, 142]. Annak ellenére, hogy nagyon sokféle homográfia-becslési technika áll rendelkezésre a területen, nem találtunk olyat, amely a helyi affin transzformáción alapuló homográfia-becsléssel foglalkozott volna.

Homográfiák alkalmazása. A számítógépes látásban számos olyan eset van, amikor a homográfia ismeretére van szükség. Először is a kamera kalibrálásáról kell írni [190]. Ha a 3D sakktábla koordináták és a vetített koordináták közötti homográfiákat több képre kiszámítjuk, akkor a belső kameraparaméterek kiszámíthatók, ahogy azt Zhang megmutatta [190] széles körben alkalmazott kalibrációs módszerével.

A kamera kalibrálása az eszköz belső és külső paramétereinek meghatározása, ahol a belső paraméterek kamera-specifikusak: ezek a gyújtótávolság, a lencse torzítása és a főpont. A külső paraméterek a kamera tájolását és a térben való elhelyezkedését írják le.

A felületi normálisok becslése szintén fontos alkalmazása a síkbeli homográfiáknak. Ha a homográfia ismert egy sík két perspektivikus kamerával készített képei között, akkor a homográfia felbontható a kamera külső paramétereire, a sík normálisára és a sík távolságára az első kamerától [45, 107]. Molnár és munkatársai [114], illetve jómagam is kollégáimmal [15] megmutattuk, hogy a felületi normális kiszámításához elegendő az affin transzformáció, és az affin transzformáció kiszámítható a homográfiából annak levezetésével, ahogy azt korábban ebben a a fejezetben is megmutattam.

A homográfia-becslés egyik nagyon fontos alkalmazási területe olyan jelenetek 3D modelljeinek készítése, ahol viszonylag nagy síkok vannak jelen. Tipikus példa az ilyen feladatokra a városi jelenetek rekonstrukciója, hiszen az ember alkotta világban sok helyen található kiterjedt síkfelület. A feladat régóta kutatott [119, 158]. Napjainkban a 3D rekonstrukciós csővezetékek pontmegfeleléseket használnak a jelenetek ritka [5, 131] vagy sűrű [51, 174] pontfelhőjének kiszámításához. A közelmúltban azonban foltalapú megközelítéseket is javasoltak [26, 158].

4.3.2. Megoldók homográfia becslésére lokális affin transzformációk felhasználásával

Ennek a résznek a fő hozzájárulása az, hogy különböző módszereket mutatok be a homográfia becslésére, ha lokális affin transzformációk egymásnak megfelelő helyeken ismertek a két kép között. Azt is megmutatom, hogy hatékonyabb becslők

képezhetők, ha az epipoláris geometria is ismert.

Először ismerkedjünk meg a főbb geometriai fogalmakkal és kifejezésekkel!

4.3.3. Elméleti háttér

Ahogyan azt már a bevezetőben is említettem, a $\mathbf{H}$ homográfia kapcsolatot teremt az első kép $\mathbf{p}_i^{(1)} = [x_i^{(1)} y_i^{(1)}]^T$ pontja és a másodikon megfeleltetett: $\mathbf{p}_i^{(2)} = [x_i^{(2)} y_i^{(2)}]^T$ pont között:

$$[x_i^{(2)} y_i^{(2)}, 1]^T \sim \mathbf{H}[x_i^{(1)} y_i^{(1)}, 1]^T,$$

ahol a $\sim$ operátor a skála erejéig egyenlőséget jelenti, melyet a homogén osztással lehet egyenlőséggé alakítani. Itt a felső indexek a képszámot, az alsó index pedig a kiválasztott pontpár számát jelölik. Molnár és mtsai. [113] megmutatták, hogy az

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \tag{28}$$

affin traszformáció első két oszlopa megkapható a 16. összefüggés segítségével. Az $\mathbf{A}$ utolsó sora adja az eltolást, melyet a pontok helye alapján lehet meghatározni.

4.3.4. Homográfia becslése affin transzformációkból (HA algoritmus)

Az $\mathbf{A}_i$ transzformáció felelős az i-edik pontpárért. A 16. formulák alapján írhatjuk fel az alábbi lineáris rendszert, amennyiben az s skálázást beszorzás segítségével eltüntetjük:

$$\begin{aligned}
h_{11} - h_{31}\left(x_i^{(2)} + a_{i,11}x_i^{(1)}\right) - h_{32}a_{i,11}y_i^{(1)} - h_{33}a_{i,11} = 0 \\
h_{12} - h_{32}\left(x_i^{(2)} + a_{i,12}y_i^{(1)}\right) - h_{31}a_{i,12}x_i^{(1)} - h_{33}a_{i,12} = 0 \\
h_{21} - h_{31}\left(y_i^{(2)} + a_{i,21}x_i^{(1)}\right) - h_{32}a_{i,21}y_i^{(1)} - h_{33}a_{i,21} = 0 \\
h_{22} - h_{32}\left(y_i^{(2)} + a_{i,22}y_i^{(1)}\right) - h_{31}a_{i,22}x_i^{(1)} - h_{33}a_{i,22} = 0
\end{aligned} \tag{29}$$

A homográfia elemeire nézve ez egy homogén lineáris egyenletrendszer. Sajnos a kilenc elemből csak hetet tartalmaz, ezért a maradék kettőt – konkrétan a h_{13} és h_{23} elemeket –, az affin transzformációk segítségével nem lehet megbecsülni. Szerencsére a pontmegfeleltetésekből további két egyenletet fel lehet írni [62], melyeket a DLT [11]-nek hívnak a szakirodalomban:

$$\begin{aligned}
h_{11}x_i^{(1)} + h_{12}y_i^{(1)} + h_{13} - h_{31}x_i^{(1)}x_i^{(2)} - h_{32}y_i^{(1)}x_i^{(2)} - h_{33}x_i^{(2)} = 0, \\
h_{21}x_i^{(1)} + h_{22}y_i^{(1)} + h_{23} - h_{31}x_i^{(1)}y_i^{(2)} - h_{32}y_i^{(1)}y_i^{(2)} - h_{33}y_i^{(2)} = 0.
\end{aligned} \tag{30}$$

A 29. és a 30. formulákkal összesen hat homogén lineáris egyenletrendszert kapunk a $\mathbf{H}$ homográfia mátrixának kilenc ismeretlen paraméterére. Azaz fel lehet írni $\mathbf{Bh} = 0$ alakban, ahol a $\mathbf{B}$ együtthatómátrix az ismert elemeket tartalmazza, melyeket a pontkoordinátákból és az affin transzformációk segítségével kaphatunk meg. A $\mathbf{h}$ kilenc dimenziós vektor a homográfia elemeit tartalmazza.

[11] DLT: Direct Linear Transformation

A B.2 függelékben mutatjuk meg, hogy a homogén lineáris egyenletrendszereket legkisebb négyzetes értelemben optimálisan meg lehet határozni, mégpedig a $\mathbf{B}^T\mathbf{B}$ mátrix legkisebb sajátértékéhez tartozó sajátvektorként. Ezt a módszert nevezzük HA [12] algoritmusnak.

Miután kilenc elemet kell megbecsülni, de a sajátvektor számításában a hossz nem számít, nyolc szabad paraméter található a $\mathbf{h}$ vektorban. Ehhez legalább nyolc egyenletet kell megadni. Ha két pontot tudunk affin transzformációkkal együtt megfeleltetni, máris 12 egyenletet kapunk. Tehát a feladat túlhatározottá válik. A minimális nyolc egyenlethez elegendő egy pont affin transzformációkkal, és még egy pont affin transzformáció nélkül.

A klasszikus DLT módszerhez négy pontmegfelelés szükséges. A HA algoritmusnak nagyon nagy előnye a klasszikus DLT módszerrel szemben, hogy elegendő kettő megfeleltetés. Mivel valós alkalmazások esetén a kiugró pontok jelenléte szinte elkerülhetetlen, robusztus eljárásokra van szükség. A legtöbb ilyen algoritmus – mint például Az E.2.1. függelékben ismertetett RANSAC módszer – véletlen mintavételezésre épül. A szükséges mintavételek számát – és emiatt az eljárás sebességét –, alapvetően határozza meg a minimális modellhez szükséges pontszám. Ezért a HA módszer nagyon nagy előnye, hogy sokkal gyorsabban lehet robusztus becslésekben alkalmazni.

4.3.5. Homográfia becslése ismert fundamentális mátrix segítségével (HAF)

Amennyiben az epipoláris geometria ismert a két kép között, a homográfia számítása sokat egyszerűsödik. A 107. egyenlet írja le a kapcsolatot az epipoláris geometria és egy tetszőleges sík-sík homográfia között. Ebbe behelyettesítve a 16 egyenletekben leírt, második törvényünknek megfelelő összefüggésekkel, a homográfia becslésére a következő formulák adódnak:

$$
\begin{aligned}
h_{31}\left(a_{i,11}x_i^{(1)} + x_i^{(2)} - e_x\right) + h_{32}a_{i,11}y_i^{(1)} + h_{33}a_{i,11} &= f_{21},\\
h_{32}\left(a_{i,12}y_i^{(1)} + x_i^{(2)} - e_x\right) + h_{31}a_{i,12}x_i^{(1)} + h_{33}a_{i,12} &= f_{22},\\
h_{31}\left(a_{i,21}x_i^{(1)} + y_i^{(2)} - e_y\right) + h_{32}a_{i,21}y_i^{(1)} + h_{33}a_{i,21} &= -f_{11},\\
h_{32}\left(a_{i,22}y_i^{(1)} + y_i^{(2)} - e_y\right) + h_{31}a_{i,22}x_i^{(1)} + h_{33}a_{i,22} &= -f_{12}.
\end{aligned}
$$

Most is lineáris összefüggéseket kaptunk, de inhomogén rendszer adódott, melyet a $\mathbf{Cy} = \mathbf{d}$ alakba írhatunk. $\mathbf{d} = [f_{21}, f_{22}, -f_{11}, -f_{12}]$ tartalmazza a fundamentális mátrix ismert elemeit, $\mathbf{y} = [h_{31}, h_{32}, h_{33}]^T$ pedig az ismeretleneket tartalmazó négydimenziós vektor. A $\mathbf{C}$ mátrix elemei mind ismert adatokat tartalmaznak:

[12] HA: Homography from Affine

$$
\begin{aligned}
C_{11} &= \left(a_{i,11}x_i^{(1)} + x_i^{(2)} - e_x\right), & C_{12} &= a_{i,11}y_i^{(1)}, \\
C_{13} &= a_{i,11} & C_{21} &= a_{i,12}x_i^{(1)}, \\
C_{22} &= \left(a_{i,12}y_i^{(1)} + x_i^{(2)} - e_x\right), & C_{23} &= a_{i,12}, \\
C_{31} &= \left(a_{i,21}x_i^{(1)} + y_i^{(2)} - e_y\right), & C_{32} &= a_{i,21}y_i^{(1)}, \\
C_{33} &= a_{i,21} & C_{41} &= a_{i,22}x_i^{1}, \\
C_{42} &= \left(a_{i,22}y_i^{(1)} + x_i^{(2)} - e_y\right), & C_{43} &= a_{i,22}.
\end{aligned}
\tag{31}
$$

A legkisebb négyzetes értelemben vett optimális megoldás az $\mathbf{y} = \mathbf{C}^{\dagger}\mathbf{d}$ alakban kapható meg, ahol $\mathbf{C}^{\dagger}$ a Moore-Penrose-féle pszeudonverze a $\mathbf{C}$ együtthatómátrixnak.

A módszer túlhatározott és minimális esetben egyaránt működik [13]. A módszer komoly előnye, hogy egyetlen pontmegfelelés és a hozzá tartozó affin transzformációból meg tudja oldani a feladatot. Mindez összhangban van a 4.1. fejezetben ismertetett normálvektorbecslővel. Hiszen a normálvektort egyetlen affin megfelelésből ki lehetett számítani, sőt, a feladat túlhatározott volt.

Jól ismert [62], hogy a homográfia a két kamera belső paramétereiből és a kamerák egymáshoz viszonyított elhelyezkedéséből[14] határozhatók meg. Ha a kamera (belső) paraméterei ismertek, a homográfiát fel lehet bontani [45, 107] a relatív elhelyezkedésre és a normálvektorra. A normálvektorbecsléshez képest itt további előny, hogy a kamerák elhelyezkedése is megkapható kalibrált kamerák esetében, ha a felbontó algoritmust [107] futtatjuk az eredményül kapott homográfiára.

A futási idő tekintetében is fontos eredmény, hogy csak pontmegfelelést használva három pontpárra van szükségünk a homográfia megbecsléséhez a fundamentális mátrix ismerete esetén, míg az affin transzformáció megléte egyetlen egy megfelelésre szorítja le a minimális modell elkészítéséhez szükséges bemeneteket.

4.3.6. Javítási lehetőségek

Normalizálás. A fenti egyenletekben az együtthatómátrixokat a pontkoordináták és az affin transzformációk alapján határozhatjuk meg. A pontkoordinátáknak nagy problémája, hogy egy kép feldolgozása után sok százas, egy-két ezres értékeket kapunk, ellentétben az affin paraméterek, amelyek tipikusan tíznél kisebbek. Amikor ilyen szélsőséges különbségek vannak egy mátrixban, a sajátérték számítás vagy éppen a pszeudo-inverz kiszámítása komoly numerikus kérdéseket vet fel [15]

A normalizálás esetén a pontkoordináták nagyságrendjét az egyszámjegyű tartományba kell levinni. A szakmában szinte egyeduralkodó [62] megoldás, hogy affin transzformációkkal először a súlypontba toljuk az origót, majd a pontok szórását skálázással $\sqrt{2}$-vé változtatjuk (általában zsugorítjuk). A két képre külön kell a normalizálást elvégezni. Az eltolást és a skálázást összevonva kapjuk a két normalizáló transzformációt, melyeket $\mathbf{T}_1$-el és $\mathbf{T}_2$-vel jelöljük. A normalizálás a $\mathbf{p}_i^{(1)'} \sim \mathbf{T}_1\mathbf{p}_i^{(1)}$ és $\mathbf{p}_i^{(2)'} \sim \mathbf{T}_2\mathbf{p}_i^{(2)}$ összefüggés adja. A normalizálást a homográfia és az alapvető mátrix esetében is ki lehet használni. Az átváltás a $\mathbf{H}' = \mathbf{T}_2\mathbf{H}\mathbf{T}_1^{-1}$. és a $\mathbf{F}' = \mathbf{T}_2^{-T}\mathbf{F}\mathbf{T}_1^{-1}$.

[13] Minimális esetben hagyományos inverz alkalmazható.

[14] A két kamera elhelyezkedését szokás relatív póznak is nevezni.

[15] Például: hány biten tároljuk a lebegőpontos számokat.

formulák felhasználásával történik, ahol a vessző a normalizált homográfiát és fundamentális mátrixot jelöli.

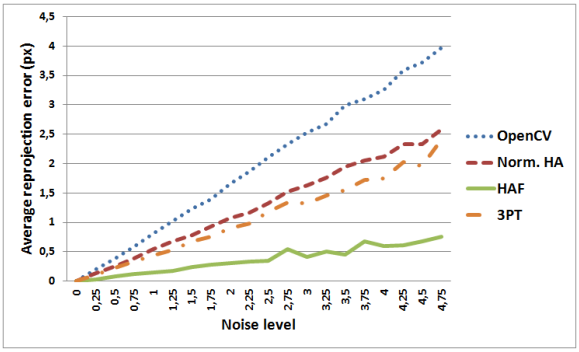
A függelék H. fejezete mutatja meg, hogy az affin transzformációkat is lineáris transzformációkkal lehet normalizálni. Az eltolás, ahogyan az el is várható, nem befolyásolja az affin transzformáció lineáris részét. A vízszintes és függőleges skálázást azonosnak tekintve az első képhez tartozó normalizáló skálát s_1-el, a második képhez tartozót s_2-vel jelölve az affin transzformáció normalizálását így írhatjuk le:

$$\mathbf{A}' = \frac{s_1}{s_2}\mathbf{A}. \tag{32}$$

Numerikus, nemlineáris finomítás. A becsléshez kapott egyenleteket a 16. összefüggések alapozták meg. Első lépésként az s skálával (projektív mélységgel) szoroztuk meg a négy alapegyenletet. Ez a lépés ugyan lineáris formulákat eredményez, melyeket algebrai módszerekkel jól tudunk kezelni, azonban súlyosan beleavatkozunk a mért adatok és a zaj viszonyába. Miután s (projektív mélység) becsült és nem pontos érték, a benne lévő hibával szorozzuk a tagokat, amelyek maguk is zajt tartalmaznak. Így a két zaj szerencsétlen esetben felnagyítja egymást, és ettől a becslésünk pontatlanabb lesz. Ezért az eredeti összefüggéseket numerikusan lehet finomítani. Mi a Levenberg-Marquardt algoritmust [96, 108] javasoljuk .

4.3.7. Teszteredmények

A disszertációban terjedelmi okokból csupán néhány vizsgálati eredményt tudok bemutatni, részletesebb teszteléseket a konferencia [16] és folyóirat publikációnkban [17] lehet olvasni.



14. ábra. *Az átlagos visszavetítési hiba pixelben, amennyiben a homográfia becsléséhez használt pontokat nulla várható értékű normál eloszlású zajjal terheljük. A megvizsgált módszerek: OpenCV: A hagyományos, DLT [62] alapú négy pontos becslés; 3PT: fundamentális mátrixot használó három pontos módszer; Norm HA: normalizált affin transzformációt alkalmazó HA algoritmus; HAF: affin transzformációt és a fundamentális mátrixot egyaránt alkalmazó módszer.*

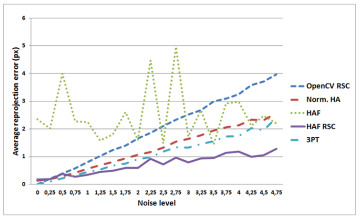
Az 1. táblázat mutatja meg a négy alapmódszert, amelyek az összehasonlítás alapját adják. Először szintetikus adatokon teszteltük, az alapigazság [16] adatokat mind pontmegfelelésekre, mind az affin transzformációkra ismertek.

[16] Az alapigazság az angol ground-truth (GT) magyarra fordítása. A szintetikus tesztek esetén a tökéletes értékeket jelenti, melyekhez minél közelebbi becslést szeretnénk adni.

Affin transzformáció / Alapvető mátrix	Nem	Igen
Nem	OpenCV (DLT)	HA
Igen	3PT	HAF

1. táblázat. *A homográfia becslésére alkalmazott módszerek. A HA és a HAF algoritmusok saját algoritmusok.*

Először azt nézzük meg, hogy kizárólag a pontkoordináták zajosításával milyen eredményeket lehet érni. A 14. ábrán hasonlítjuk össze a kizárólag pontmegfeleléseket (OpenCV, 3PT) algoritmusokat a kombináltakkal (HA,HAF). A becslés végén kapott vetítési hibát ábrázoljuk, azaz azt nézzük meg, hogy a kapott homográfiával áttranszformálva a pontokat mekkora eltérés adódik az alapigazsághoz képest. A használt megfelelések minden módszernél ugyanazok. Jól látható, hogy az elvárt eredményt kapjuk: az alapvető mátrixokat használó módszerek (3PT, HAF) alacsonyabb hibát eredményeznek, tehát jobbak, hiszen többletinformációt alkalmaznak. Ugyanígy az affin módszerek (HA, HAF) megelőzik a simán pont-koordinátákat alkalmazó versenytársaikat.

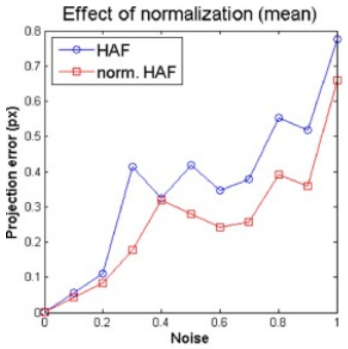


15. ábra. *Az átlagos homográfia vetítési hiba, amennyiben a homográfia becsléséhez használt affin transzformációkat is nulla várható értékű normál eloszlású zajjal terheljük. A megvizsgált módszerek: OpenCV RSC: A hagyományos, DLT [62] alapú négy pontos becslés RANSAC [46] alkalmazásával; 3PT: fundamentális mátrixot használó három pontos módszer; Norm HA: normalizált affin transzformációt alkalmazó HA algoritmus; HAF: affin transzformációt és a fundamentális mátrixot egyaránt alkalmazó módszer. HAF RSC: ugyanez a módszer RANSAC alkalmazásával.*

A 15. ábrán ugyanennek a tesztnek azt a változatát látjuk, ahol az affin transzformációt is zajosítjuk, hasonlóan nulla várható értékű normális eloszlású zajjal. Az affin transzformációk és a pontok zaja ugyanakkora szórású, de azt fontos megjegyezni, hogy a két zaj teljesen eltérő hatással rendelkezik, ezért nem teljesen "tisztességes" az összehasonlítás. Az affin zaj lényegesen többet ront az eredményen, ezért a RANSAC [46] eljárást bevetettük a HAF algoritmus javítására. Ez utóbbi adja egyértelműen a legjobb eredményt. A többi teszt is az elvárható eredményt adja: a 3PT algoritmus jobb, mint a sima OpenCV (DLT) megoldás.

Végezetül a 16. ábrán a normalizálás hatását külön is megvizsgáltuk. Csodát nem szabad várni a normalizálástól, de az eredmények érezhető javulását igen, különösen nagy zajszint esetén.

Miután a harmadik törvény a négy lineáris affin paraméterre négy egyenletet határoz meg, melyek az osztás eliminálásával lineáris egyenletekké alakíthatóak, minden egyes affin transzformácó négy plusz ismeretet ad a homográfia becslésére. A



16. ábra. *A normalizálás hatása növekvő zajszint hazására. Magasabb zajérték mellett nagyobb a normalizálás jelentősége.*

pontmegfelelések további két egyenletet adnak, ezért összesen hat egyenlet adott affin megfeleltetésként. Ezért már két affin megfelelés elég ahhoz, hogy a homográfiát megbecsüljük. Kizárólag pontmegfeleléseket alkalmazva négy párra lenne szükségünk, így a mi a megközelítésünk jelentős újításnak tekinthető a területen: fele annyi affin megfelelés elég a becslés eléréséhez. Ha robusztus becslést akarunk végezni – például a legelterjedtebb, véletlen mintavételezést alkalmazó RANSAC algoritmus [46] alkalmazásával –, jelentős gyorsítás érhető el, hiszen mintavételenként jóval nagyobb az esély, hogy kiugró adat nem szennyezi a mintát.

Speciális eset, ha a két kép között ismert a fundamentális mátrix, ekkor a a homográfia szabad paramétereinek száma háromra csökken. Ebben az esetben egy pontmegfelelés csupán egyetlen egyenletet ad, a lineáris része az affin transzformációnak pedig további kettőt. Ezért egyetlen affin pár elég a két képen, hogy a homográfiát meghatározzuk. Ennek a ténynek elméletileg is nagy jelentősége van: *lokális információból* megbecsülhető a homográfia kalibrált esetben. Ráadásul egy homográfiából az adott ponthoz tartozó felületi érintősíkot is meg lehet határozni [45, 107].

Ez logikus is, hiszen a homográfia térbeli síkok projektív vetületeinek kapcsolatát írja le, és a háromszögelés segítségével (hagyományos szereó látás esetén) a sík egy pontja határozható meg, a korábbi tézisnek megfelelően a lineáris részből megbecsülhető a felületi normálvektor. Egy pont és a normálvektor pedig egyértelműen meghatározza a kapcsolódó térbeli síkot.

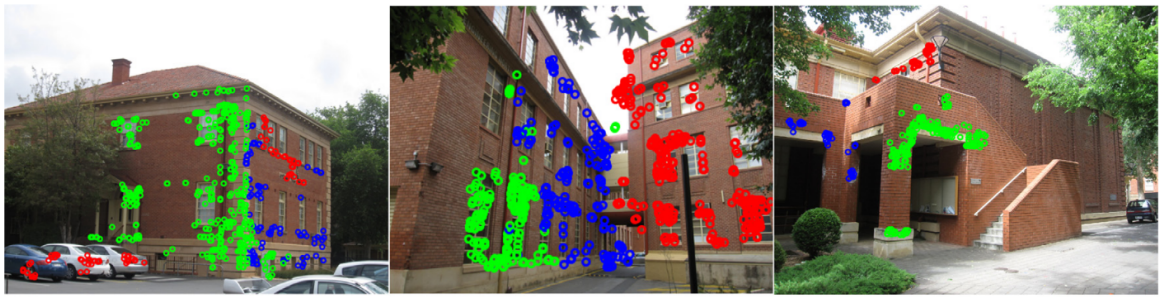
Az itt ismertetett elvekre alapozva dolgoztunk ki Baráth Dániel doktoranduszommal közösen egy algoritmuscsaládot [17, 16], melyben a homográfiát becslő algoritmusok a fenti eseteket fedik le.

4.4. Síkok szegmentációja

Miután a HAF algoritmus egyetlenegy affin transzformációból és a hozzá tartozó pontok koordinátáiból képes a homográfiát kiszámítani, ezért ezt bátran nevezhetjük "egy pontos algoritmusnak". Ezt a homográfiát további affin megfelelésekre illeszteni lehet, és meg lehet határozni, vajon azok megfelelnek-e a homográfiának. Így pedig, véletlen mintavételezéssel kiegészítve, egy nagyon gyors síkillesztőt kapunk, amely elméletileg tetszőleges számú sík detektálására képes. Ezért neveztük el ezt az eljárást MultiH-nak.

A 17. ábrán láthatjuk, hogy az azonos színnel jelzett pontok illeszkednek az azo-

nos homográfiákra.



17. ábra. *Síkok szegmentálása a MultiH eljárással.*

Miután egy affin megfelelésből egy térbeli sík meghatározható (rekonstruálható), minden lehetséges párra egy felületi érintősík adódik. Síkobjektumok esetén a sík pontjaiból ugyanaz az érintősík becsülhető, ezért a síkok detekciója hatékonyan elvégezhető. 2016-os munkánkban [35] javasoltunk egy síkdetektor eljárást, amellyel tetszőleges számú sík meghatározható a képen, kizárólag az affin megfelelések felhasználásával. Ez lényegében egy klaszterező illesztő eljárás, amelynél a síkok reprezentálják a modellt, és minden egyes lokális affin transzformáció szavaz egy modellre. Azokat a síkokat kell megtalálni, amelyekre közeli transzformációk szavaznak.

A 2016-os publikációnk [35] Baráth Dániel doktorandusz és egy neves cseh szakértő, Jiri Matas közreműködésével készült. Utóbbi elsősorban az ötleteivel segítette a közös munkát, Jiri egyik szakterülete a robusztus multi-modell illesztés, akkoriban nálunk lényegesen szélesebb látókörrel rendelkezett a területen.

4.5. Alapvető és lényegi mátrixok becslése

Az affin transzformációkra vonatkozó III. törvény alapján (24. összefüggés) írtuk, hogy

$$\mathbf{A}^{\mathbf{T}}\hat{\mathbf{F}}\begin{bmatrix}\mathbf{p}_1\\1\end{bmatrix}=-\tilde{\mathbf{F}}^T\begin{bmatrix}\mathbf{p}_2\\1\end{bmatrix}, \tag{33}$$

amennyiben a fundamentális mátrix kilenc elemére az

$$\mathbf{F}=\begin{bmatrix}f_{11}&f_{12}&f_{13}\\f_{21}&f_{22}&f_{23}\\f_{31}&f_{32}&f_{33}\end{bmatrix} \tag{34}$$

jelölést alkalmazzuk, és

$$\tilde{\mathbf{F}}=\begin{bmatrix}f_{11}&f_{12}\\f_{21}&f_{22}\\f_{31}&f_{32}\end{bmatrix},\qquad \hat{\mathbf{F}}=\begin{bmatrix}f_{11}&f_{12}&f_{13}\\f_{21}&f_{22}&f_{23}\end{bmatrix}. \tag{35}$$

Ekkor a 33. összefüggést a fundamentális és az affin mátrix elemivel így írhatjuk le:

$$\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = - \begin{bmatrix} f_{11} & f_{21} & f_{31} \\ f_{12} & f_{22} & f_{32} \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}. \tag{36}$$

Egyszerű átalakításokkal a fundamentális mátrix elemeire a

$$\left[\begin{array}{ccc|ccc|c|c|c|c} a_{11}u_1+u_2 & a_{11}v_1 & a_{11} & a_{21}u_1+v_2 & a_{21}v_1 & a_{21} & 1 & 0 & 0 \\ a_{12}u_1 & a_{12}v_1+u_2 & a_{12} & a_{22}u_1 & a_{22}v_1+v_2 & a_{22} & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0 \tag{37}$$

lineáris homogén alakot kaphatjuk meg. Amennyiben a 8. összefüggést is figyelembe vesszük, ahol a két képen az egymásnak megfelelő $\mathbf{p}_1$ és $\mathbf{p}_2$ pontokra igaz, hogy $\mathbf{p}_2^T \mathbf{F} \mathbf{p}_1 = 0$, akkor ennek alapján írhatjuk, hogy

$$\begin{bmatrix} u_2u_1 & u_2v_1 & u_2 & v_2u_1 & v_2v_1 & v_2 & u_1 & v_1 & 1 \end{bmatrix} \mathbf{f} = 0, \tag{38}$$

amennyiben a $\mathbf{f} = [f_{11} \quad f_{12} \quad f_{13} \quad f_{21} \quad f_{22} \quad f_{23} \quad f_{31} \quad f_{32} \quad f_{33}]^T$ vektor tartalmazza a becsülendő elemeit az alapvető mátrixnak. Ezt hívjuk a pontmegfelelésből kapott összefüggésnek. Ezt kiegészítve a 37. egyenletből kapott affin összefüggésekkel, írhatjuk, hogy $\mathbf{C}\mathbf{f} = 0$, ha

$$\mathbf{C} = \left[\begin{array}{ccc|ccc|c|c|c} a_{11}u_1+u_2 & a_{11}v_1 & a_{11} & a_{21}u_1+v_2 & a_{21}v_1 & a_{21} & 1 & 0 & 0 \\ \hline a_{12}u_1 & a_{12}v_1+u_2 & a_{12} & a_{22}u_1 & a_{22}v_1+v_2 & a_{22} & 0 & 1 & 0 \\ \hline u_2u_1 & u_2v_1 & u_2 & v_2u_1 & v_2v_1 & v_2 & u_1 & v_1 & 1 \end{array}\right]. \tag{39}$$

Azaz a harmadik törvényünkből, és a pontokra már korábban meghatározott törvény [62] felhasználásával *három homogén lineáris egyenletrendszert lehet felírni a fundamentális mátrixra, ha egy pontmegfelelést és a hozzá tartozó affin transzformációt ismerjük.*

Egy megfelelésből tehát három egyenletet kapunk. A fundamentális mátrixnak a szabadságfoka hét: ugyan kilenc eleme van, de a determinánsa nulla, tehát van egy megkötésünk; továbbá a skálázásra invariáns, ezért a kilencből a két szabadságfokot levonva hetet kapunk. Ezért legalább három megfelelésre van szükség. Azonban ez is jelentős előrelépés, hiszen a hagyományos, kizárólag pontmegfelelésen alapuló módszer esetében hét megfelelést kell megadni.

Lényegi és alapvető mátrix. Ugyan a fenti levezetést az alapvető (fundamentális) mátrixra végeztük el, kalibrált esetben a lényegi (eszenciális) mátrixokra is alkalmazhatóak. Kalibrált esetben a kamera belső paramétereit tartalmazó felső-háromszög kalibrációs mátrixokat, $\mathbf{K}_1$-et és $\mathbf{K}_2$-t ismerjük. Ha ezekkel elvégezzük a normalizációt:

$$\mathbf{p}_1 \leftarrow \mathbf{K}_1^{-1}\mathbf{p}_1, \quad \mathbf{p}_2 \leftarrow \mathbf{K}_2^{-1}\mathbf{p}_2, \tag{40}$$

akkor a normalizált koordinátákat behelyettesítve a 39. összefüggésbe, a homogén egyenletrendszert megoldva a lényegi mátrixot kapjuk meg.

Érdekesség: Raposo és Barreto [136] 2016-ban ugyanazt az összefüggést megtalálta a MATLAB szimbolikus "toolbox" alkalmazásával, de a III. törvényt analitikusan nem tudták felírni. Ugyanakkor három megfeleléses becslőt publikáltak.

A továbbiakban azt mutatom meg, hogyan lehet két pontos megoldót készíteni

Kétpontos megoldó.

Ha két pontpárt ismerünk, továbbá a hozzá tartozó affin transzformációkat, akkor a 39. összefüggés alapján készíthetünk egy $\mathbf{C}^1$ és egy $\mathbf{C}^2$ 3×9 -es méretű mátrixot. Értelemszerűen a felső index a pár sorszámát mutatja.

A két mátrixot egymás alá tesszük:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}^1 \\ \mathbf{C}^2 \end{bmatrix}. \tag{41}$$

A megoldás, azaz a fundamentális mátrix elemeit tartozó vektor merőleges a mátrix minden sorára, hiszen homogén lineáris egyenletekről van szó. A $\mathbf{C}$ mátrix mérete ebben az esetben 6×9, ezért a nulltere háromdimenziós. Így minden $\mathbf{x}$ nullvektort fel tudunk írni a nulltér bázisainak lineáris kombinációjaként:

$$\mathbf{x} = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}, \tag{42}$$

ahol $\mathbf{a}$, $\mathbf{b}$ és $\mathbf{c}$ a nullvektorok, míg α, β és γ az (általában ismeretlen) súlyok. Az $\mathbf{x}$ vektor $\mathbf{F}$ elemeit tartalmazza, ezért a kettő egymásnak egyértelműen megfeleltethető.

A módszerünk esetében félig kalibrált kamerákat feltételezünk:

$$\mathbf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \sim \mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \tau \end{bmatrix}, \quad \tau = 1/f^2. \tag{43}$$

A trace (spur) megkötés alapján [62]: $2\mathbf{F}\mathbf{Q}\mathbf{F}^T\mathbf{Q}\mathbf{F} - trace(\mathbf{F}\mathbf{Q}\mathbf{F}^T\mathbf{Q})\mathbf{F} = 0$.

Szimbolikus eszközt használva, $\mathbf{x}$-ből $\mathbf{F}$ mátrixot csinálva, majd behelyettesítve harmadfokú, többváltozós polinomiális egyenletrendszert kapunk α-ra, β-ra, γ-ra és τ-ra, ahol $\tau = f^{-2}$ tartalmazza az ismeretlen fókusztávot. A feladat megoldása során "szerencsénk volt", hiszem a trace megkötés alkalmatása során τ-t el tudtuk különíteni, így erre a rejtett változók [17] technikáját alkalmazhattuk. Ha τ-t rejtett változónak tekintjük, meghatározhatjuk $\mathbf{C}(\tau)$ mátrixot. A mátrix soraira τ-tól függő egyváltozós függvényeket kapunk. Bár α, β és γ skálázásra érzéketlen, nem rögzítjük a skálát, hiába tehetnénk meg, hogy ne kelljen inhomogén egyenletekkel foglalkozni. A szimbolikus eszköz után a monomiálisa így alakulnak: $\mathbf{y} = [\alpha^3 \;\; \alpha^2\beta \;\; \alpha^2\gamma \;\; \alpha\beta^2 \;\; \alpha\beta\gamma \;\; \alpha\gamma^2 \;\; \beta^3 \;\; \beta^2\gamma \;\; \beta\gamma^2 \;\; \gamma^3]^T$.

Miután a monomiális $\mathbf{x}$ vektor skálája nincs rögzítve, a nem triviális megoldást a determináns kritérium szerint kaphatjuk meg:

$$\det(\mathbf{C}(\tau)) = 0. \tag{44}$$

A mi esetünkben a szimbolikus megoldás τ-ra egy 15-ödfokú polinomot ad, ennek az egyik gyöke adja a jó megoldás. Nem marad más feladat, mint a nem valóságszerű gyököket ki kell szűrni.

[17] Angolul: hidden variable technique

4.5.1. A jó gyökök megtalálása

Ebben a szakaszban egy új technikát javasolunk a gyökerek elhagyására geometriai megfontolások alapján. Ehhez az alkalmazott digitális fényképezőgépek tulajdonságait vesszük figyelembe, és ennek alapján az érvénytelen gyújtótávolságokat el tudjuk távolítani. Végül bemutatunk egy gyökérkiválasztó algoritmust.

Érvénytelen fókusztávolságok kiszűrése

Itt egy olyan megoldást javasolunk, amely a mögöttes geometrián alapul, hogy kiküszöböljük az érvénytelen fókusztávolságokat. Tegyük fel, hogy adott egy $(\mathbf{p}_1, \mathbf{p}_2)$, pontpár a hozzá tartozó $\mathbf{A}$ lokális affin transzformációval együtt, továbbá az $\mathbf{F}$ alapvető mátrix és az f fókusztávolság. Mivel félig kalibrált kamerákat feltételezünk, $\mathbf{F}$ és f pontosan meghatározza mindkét kamera $\mathbf{P}_1$ és $\mathbf{P}_2$ vetületi mátrixait [62]. Jelöljük a $(\mathbf{p}_1, \mathbf{p}_2)$ pontpárhoz és $\mathbf{A}$ lokális affin transzformációhoz tartozó térbeli 3D pozíciót $\mathbf{q} = [x \quad y \quad z]^T$-val, és a felületi normálvektort $\mathbf{n} = [n_x \quad n_y \quad n_z]^T$-vel! A térbeli $\mathbf{q}$ pontot itt az egyszerűség kedvéért a lineáris háromszögelés [62] (DLT algoritmus) segítségével, a normálvektort pedig a 2015-ben publikált [15], a 4.1. szakaszban ismertetett, negyedfokú polinomok gyökkeresésére visszavezetett módszerünkkel becsültük meg.

Az általánosság megsértése nélkül feltételezzük, hogy egy 3D-s felület egy pontját nem lehet hátulról megfigyelni. Ennek következtében a $\mathbf{c}_i - \mathbf{q}$ és $\mathbf{n}$ vektorok közötti szögnek mindkét kamera esetében kisebbnek kell lennie, mint $90°$, ahol $\mathbf{c}_i$ az i-edik kamera pozíciója ($i \in \{1,2\}$). Ez a következőképpen értelmezhető: minden kamera kiválaszt egy fél egységgömböt a $\mathbf{q}$ megfigyelt pont körül. A $\mathbf{n}$ felületi normálisnak e félgömbök metszetében kell lennie.

A félgömbök felületén a szögek tartományait a gömbi koordináta-rendszerben egy téglalap írja le az alábbiak szerint: $\text{rect}_i = \left[\theta_i - \frac{\pi}{2} \quad \sigma_i - \frac{\pi}{4} \quad \pi \quad \frac{\pi}{2}\right]$, ahol θ_i, σ_i jelöli a gömbi koordinátákat. A két kamera által meghatározott terület metszete a következőképpen alakul:

$$\text{rect}_\cap = \bigcap_{i \in [1,2]} \text{rect}_i.$$

Ha egy kapott érték kilóg a metszetből, akkor érvénytelennek tekinthető.

A $\mathbf{q}$ pont mindkét kamerából megfigyelhető *akkor és csak akkor*, ha az $\mathbf{n}$ felszíni normális, amelyet a Θ és Σ gömbi koordináták ábrázolnak, a metszéspont területén fekszik: $\left[\Theta \quad \Sigma\right] \in \text{rect}_\cap$. A f fókusztávolság által indukált, e kritériumoknak nem megfelelő beállítás érvénytelen, és ezért elhagyható. Megjegyzendő, hogy ez a megkötés egyszerűen kiterjeszthető a több nézetű esetre is, így a metszési terület még szigorúbbá válik.

Fókusztávok technikai korlátai

A becsült fókusztávolságokra további korlátozásokat vezethetünk be, ha figyelembe vesszük a kamerák fizikai korlátait. A $\mathbf{K}$ kameramátrixon belüli fókusztávolság nem egyenértékű a lencsék fókusztávolságával, mivel az az optikai fókusztávolság és a pixelméret [62] szorzata, más szóval a fókuztáv az érzékelő pixelméretében lett megadva. A pixelméret a gyakorlatban néhány mikrométer, míg kereskedelemben kapható optikák esetében az optikai fókusztávolságok $[1 \ldots 500]$ mm-es intervallumban vannak. Ezért f-re durva alsó és felső határként vehetjük a 100 és az 500.000 értékeket. Az ezen intervallumon kívüli fókusztávolságok automatikusan elvetésre kerülnek. Megjegyzendő, hogy ezek a határok könnyen módosíthatók, figyelembe véve a különböző tulajdonságokkal rendelkező kamerákat.

A végső gyök kiválasztása

A többszörös gyökerek feloldására és a zaj hatásának minimalizálására a klasszikus módszer a mérések ismétlése és az ellentmondásos mérések kiküszöbölése. Mivel a 44. egyenlet egy magas fokú polinom, meglehetősen érzékeny a zajra – a koordináták és az affin elemek kis változásai jelentősen eltérő együtthatókat okoznak.

A RANSAC [46] sikeres robusztus technika ennek a problémának a megoldására, például az ötpontos algoritmushoz [122] is lehet alkalmazni. A legújabb módszerek, pl. a kernel-szavazás[18], azt a tulajdonságot használják ki, hogy a gyökök a valós megoldás körül csúcsot alkotnak [99, 98, 87]. A kernel-szavazás egy kernelsűrűségfüggvényt maximalizál, maximum-likelihood becsléként. Tapasztalataink szerint ez a technika pontosan működik, ha a koordináták zaja átlagosan nem haladja meg az $1-2$ pixeles értéket. E fölött a gyökök több erősen támogatott csúcsot is képezhetnek, és nem garantált a valódi megoldás megtalálása.

Mi munkánkban a problémát egy egydimenziós keresésként fogalmazzuk meg: a valós fókusztávolság a leginkább támogatott módusként jelenik meg. Többféle móduszkeresési technika [74] közül a legrobusztusabb a Median-Shift [153] a tapasztalatok szerint. A móduszként Tukey-mediánokat [168] biztosító Median-Shift nem hoz létre új elemeket a tartományban, amelyre alkalmazzák. Különösen a Tukey- [168] és a Weiszfeld-medians [176] eredményei között nincs jelentős különbség, azonban az előbbi számítása valamivel gyorsabb. Végül, a Median-Shift diszkrét jellegének kiküszöbölése érdekében – mivel nem ad hozzá új példányokat, csak az adott példányokkal dolgozik –, a végeredmény finomításához legmeredekebb-lejtő módszert alkalmazó numerikus optimalizálást is lefuttatunk. A minimalizálandó függvény az alábbi:

$$f(x) = \sum_{i=1}^{n} \frac{\kappa(x_i - x)}{h}, \tag{45}$$

ahol n a fókusztávolságok száma, κ egy kernelfüggvény – mi a Gauss-kernelt választottuk –, x_i az i-edik fókusztávolság, h pedig a Median-Shifthez hasonló sávszélesség.

Tesztelési eredmények. A módszerünk nagy erénye, hogy a korábbi hét [62] és öt [121] pontot igénylő algoritmusokat át tudjuk alakítani úgy, hogy csupán három (fundamentális mátrix), illetve két (esszenciális mátrix) lokális affin transzformáció mintavételezése szükséges. A robusztus becslések [46] futási ideje nem polinomiális összefüggésben van a pontok számával, ezért nagyon hasznos, ha kevesebb mintavételből tudjuk a minimális modelleket elkészíteni.

A módszereinket természetesen összehasonlítottuk más megoldókkal is, helyhiány miatt azonban itt az összehasonlításokat nem közöljük. Véleményem szerint ez az elméleti újdonság a módszerek legnagyobb erénye. Az érdeklődő kedves Olvasónak az eredeti nemzetközi publikációk elolvasását ajánlom [18, 19] a futási eredmények kiértékelésének megtekintéséhez.

4.6. Síkbeli mozgás becslése sztereó képpárra

Az elmúlt években, egészen pontosan 2020-tól kezdődően, elsősorban járműre rögzített érzékelők adatainak a feldolgozásával foglalkoztam.

[18] Angolul: kernel voting

Amennyiben egy kamerát úgy rögzítünk a járműhöz, hogy a kamera képsíkja merőleges az úttestre, és az úttest maga is sík, speciális mozgást kapunk, amely a kamera mozgásának meghatározását jelentősen egyszerűsíti.

(Egy zárójeles bekezdés erejéig megemlíteném, hogy perspektív kamera esetén homográfiával a nézeti irányt korrigálni lehet. Ha ismerjük a függőleges irányt, például IMU adatai alapján, akkor azt a korrekciót el tudjuk végezni, ami a képsíkot merőlegessé teszi az úttest síkjához képest. Elméletben ez egy járható út, ám a mai MEMS technológiával készített gyorsulásérzékelők sokszor nem érik el azt a pontosságot, hogy a gyakorlatban is életképes legyen ez az elképzelés.)

A lényegi mátrix speciális esete ez a síkban mozgás. A mozgás ilyenkor egy kétdimenziós iránnyal (első paraméter) és egy függőleges elfordulással (második paraméter) írható le, így a szabadságfokunk összesen kettő, ennyi paraméter kell becsülni. A forgatási mátrix és az eltolás ebben a speciális esetben így alakul:

$$\mathbf{t} = \begin{bmatrix} t_x \\ 0 \\ t_z \end{bmatrix} = \rho \begin{bmatrix} \cos\alpha \\ 0 \\ \sin\alpha \end{bmatrix}, \qquad \mathbf{R} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}.$$

A lényegi mátrix ebben az esetben szintén különleges, ahogyan azt a J. függelékben olvashatjuk, hiszen a fent bevezetett két szögnek a szögfüggvényei segítségével adható meg négy eleme, a többi öt elem pedig zérus:

$$\mathbf{E} \sim \begin{bmatrix} 0 & -\sin\alpha & 0 \\ \sin(\alpha+\beta) & 0 & -\cos(\alpha+\beta) \\ 0 & \cos\alpha & 0 \end{bmatrix}.$$

A 39. formula mutatta meg az összefüggést a pontmegfelelések, az affin transzformációk és a fundamentális mátrix között. Amennyiben normalizáljuk a koordinátákat, tehát a $\mathbf{p} \leftarrow \mathbf{K}^{-1}\mathbf{p}$ összefüggéssel élünk minden egyes pontra, akkor írhatjuk, hogy

$$\begin{bmatrix} a_{11}u_1+u_2 & a_{11}v_1 & a_{11} & a_{21}u_1+v_2 & a_{21}v_1 & a_{21} & 1 & 0 & 0 \\ a_{12}u_1 & a_{12}v_1+u_2 & a_{12} & a_{22}u_1 & a_{22}v_1+v_2 & a_{22} & 0 & 1 & 0 \\ u_2u_1 & u_2v_1 & u_2 & v_2u_1 & v_2v_1 & v_2 & u_1 & v_1 & 1 \end{bmatrix} \mathbf{e} = 0,$$

ahol

$$\mathbf{e} = [e_{11} \quad e_{12} \quad e_{13} \quad e_{21} \quad e_{22} \quad e_{23} \quad e_{31} \quad e_{32} \quad e_{33} \quad]^T$$

a lényegi mátrix elemeit tartalmazó vektor. A 114. összefüggés elemit beírva és a lényegi mátrix zérussal szorzott elemeit elhagyva, illetve az elemeket sorrendjét megváltoztatva az összefüggés sokat egyszerűsödik:

$$\begin{bmatrix} -a_{11}v_1 & 0 & a_{21}u_1+v_2 & -a_{21} \\ -a_{12}v_1-u_2 & 1 & a_{22}u_1 & -a_{22} \\ -u_2v_1 & v_1 & v_2u_1 & -v_2 \end{bmatrix} \begin{bmatrix} \sin\alpha \\ \cos\alpha \\ \sin(\alpha+\beta) \\ \cos(\alpha+\beta) \end{bmatrix} = 0. \tag{46}$$

A jobb oldalon levő $[\sin\alpha \quad \cos\alpha \quad \sin(\alpha+\beta) \quad \cos(\alpha+\beta)]T$ vektor két egységvektor egymás alatt. Ennek megoldására egy érdekes megoldást fejlesztek ki Koreában [31], amely azonban nem optimális. A feladatot legkisebb négyzetes értelemben mi tudtuk optimálisan megoldani, ahogyan azt a K. függelékben ismertetem.

4.6.1. Tesztelési eredmények

A javasolt algoritmust szintetikus és valós teszteknek egyaránt alávetettük. A Ebben a munkában csak röviden ismertetem az eredményeket, az eredeti ICRA2020 cikkben [159] bővebb leírás található ugyanazokról a tesztelési eredményekről.

4.6.2. Szintetikus tesztek

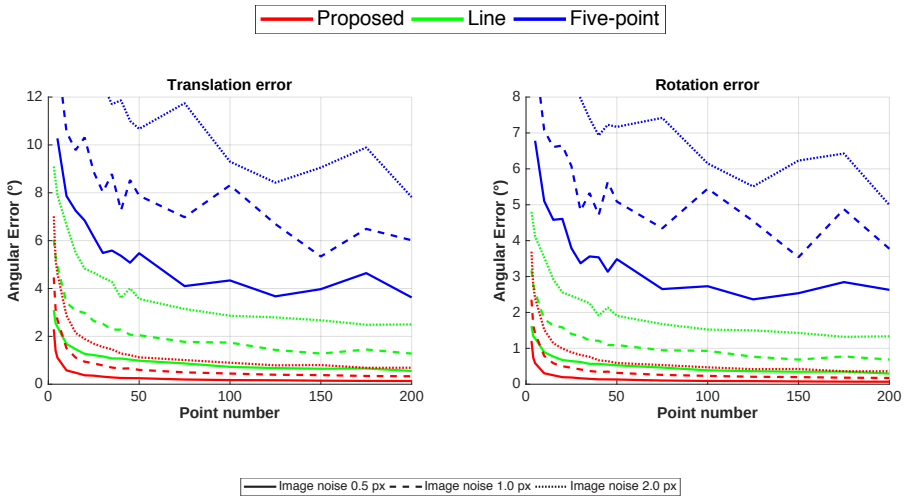
MATLAB-alapú szintetikus tesztelésünkhöz véletlen kameraparamétereket állítottunk elő, melyek megfelelnek a síkban mozgásnak. A kamerák belső paramétereit valóságszerűnek állítottuk me, szintén véletlenszám-generátor segítségével. A háromdimenziós színtér – szintén véletlenülszánokként generált – pontjait a kamera paraméterekkel levetítettük a síkra, majd nulla várható értékű, normális eloszlású zajt adunk, melynek a σ szórása a szimuláció állítható paramétere. Minden egyes értéket ezer futtatás eredményeképpen kaptunk meg.
Általános teszteset. Az összehasonlításhoz egy 'klasszikus', ötpontos algoritmust [157] és a nemrégiben közzétett Choi-módszert [31] vettük alapul. A forgatási mátrixban levő β elforgatás szögének és az eltolás α szögének a zajszinthez való viszonyát a 18. ábrán láthatja a kedves Olvasó, három különböző zajszint esetében. Jól látható, hogy a javasolt módszer minden tekintetben jobb eredményt ad a konkurens módszerekhez képest. Az α és β szögeknél nincsen lényegi eltérés.
Teszt a síkmozgástól eltérés érzékenységére. A 19. ábra tesztelésekor picit elrontjuk a síkbeli mozgást, a kép és az út síkjára egy, illetve három fokos eltérést engedtünk meg a merőlegeshez képest. Ezt úgy kell elképzelni, hogy a vízszintes (X) tengely mentén elforgattuk a háromdimenziós világot. Az eredményeken, mely az eltolás α szögének a hibáját mutatja, jól látszik, hogy a javasolt, optimális módszer a legkevésbé érzékeny a síktól való eltérésre.

Egy bekezdés erejéig megjegyezném, hogy ilyen eltérés a kamerák rögzítése esetén gyakran előfordul. Homográfia segítségével lehet korrigálni az eltérést, ahogyan arra 2022-es munkánkban [80] javaslatot is tettünk.
Futási idő. A 20. ábrán a futási időket kapjuk meg [19]. A Choi-féle módszer (Line) egyszerűsége egy picit jobb futási időt eredményét. Az ötpontos algoritmus igényli a legtöbb számítási kapacitást, hiszen jóval több paramétert becsül. Azonban érdemes észrevenni, hogy a számítási idő a másodperc ezredrészéhez van közel minden esetben, ezért akár valós idejű alkalmazásokon is alkalmazni lehet az algoritmusokat.

4.6.3. Valós teszteredmények

Valós teszteléshez a `Malaga`[20] publikus adatbázist választottuk [25]. Ezt az adathalmazt teljes egészében városi környezetben gyűjtötték olyan autóval, amely számos érzékelővel, többek között egy nagyfelbontású sztereó kamerapárral és öt lézerszkennerrel volt felszerelve. A nagyfelbontású kamera szekvenciáiból mi minden tizedik képkockát használtunk. A javasolt módszert mindig az egymást követő képpárra alkalmaztuk. Az alapigazság (ground truth - GT) útvonalakat az adatkészletben megadott GPS-koordináták felhasználásával állítottuk össze. Minden egyes egymást követő képkockapárt egymástól függetlenül dolgoztunk fel, ezért nem futtattunk sem-

[19] A futási idők értékeléséhez fontos tény, hogy 2018. körüli asztali számítógép-konfiguráción készült a teszt, egy magon futtatva a MATLAB számításokat.
[20] `https://www.mrpt.org/MalagaUrbanDataset`



18. ábra. *Szintetikus teszteredmények külső kamera paraméterek meghatározására, síkmozgás szimulálásával. Az ábrán a becsült eltolás (bal ábra) és forgatás (jobb) szöghibáit látjuk (függőleges tengely; fokban). A pontkoordinátákhoz σ-szórású, nulla várható értékű, normális eloszlású zajt adunk. Az összehasonlított módszerek a javasolt, a Choi és mtsai. [31] (Line) és Stewenius és mtsai. [157] (Five-point) algoritmusai.*

milyen optimalizálást, amely a teljes útvonal hibájának minimalizálását vagy a hurokzárás felismerését célozta volna. Az egymást követő képkockák becsült mozgásait egyszerűen összefűztük. A becsült szögek ($\in (-180, 180]$) egyetlen korrekciója az volt, hogy folyamatos pályát feltételeztünk, és így a szögeket $90°$-os modulussal használtuk. Például, ha a becsült szög $110°$ volt, akkor $20°$-t használtunk, $-110°$ esetén $-20°$-ra cseréltük. Összesen $9\,064$ képpárt használtunk fel az értékeléshez.

Robusztus becslőként Baráth Dániel doktoranduszom Graph-Cut RANSAC [13] (GC-RANSAC) módszerét választottuk, ami egyrészt korszerűnek tekinthető, másrészről a forráskódja nyilvánosan elérhető[21], ezért mások is alkalmazni tudják.

A GC-RANSAC (és más RANSAC [46]-szerű módszerek) két különböző megoldót használnak:

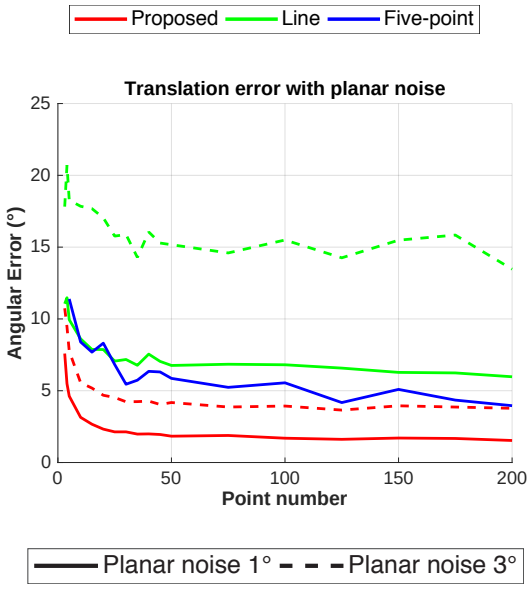
(a) Minimális mintához való illesztésre és

(b) túlhatározott esetre, amikor a modell finomítását végzi az összes kiugró érték kiszűrése után.

Az a) pont esetében a fő cél a probléma megoldása a lehető legkevesebb adatpont felhasználásával, mivel a feldolgozási idő a RANSAC ciklusok számával megszorzódik. Egy esetet kivéve (amikor az ötpontos algoritmust [157] alkalmaztuk), a tesztekben mindig a Choi-féle dolgozat [31] "Line" nevű módszerét választottuk, amely minimálisan két pont megfeleltetéséből oldja meg a problémát, és a 20. ábrán mutatott teszteredmény szerint rendkívül gyors.

A b) pont esetében összehasonlítjuk a javasolt módszert, a síkbeli mozgásproblémát lineáris rendszerként megoldó technikát (DLT algoritmus [62]), a Choi és mtsai. [31] "line" megoldóját és az általános ötpontos algoritmust [157]. Az a) és b) pontok esetében vizsgált kombinációkat a 2. táblázat tartalmazza.

[21] `https://github.com/danini/graph-cut-ransac`



19. ábra. *Az eltolás hibája (függőleges tengely; fokban) a pontszám függvényében, ha a kameramozgás nem teljesen síkbeli. Ebben az esetben a képzaj szórását* 0.5 *pixelre állítottuk be.*

A legtöbb tesztelt algoritmus több jelöltet ad vissza a kameramozgásra. A legjobb modell kiválasztásához nem használtuk fel a pontok legrosszabb 5%-át az illesztés során. (Felfelé kerekítéssel, tehát legalább egy pontot kidobtunk.) Így a legjobb 95%-ból becsültük meg a modellt, és végül azt a jelöltet választottuk ki, amelyik a legkisebb átlagos hibát adja a kihagyott pontokon. Az a síkbeli lineáris módszer egyetlen megoldást ad vissza, ezért nem igényli ezt az eljárást.

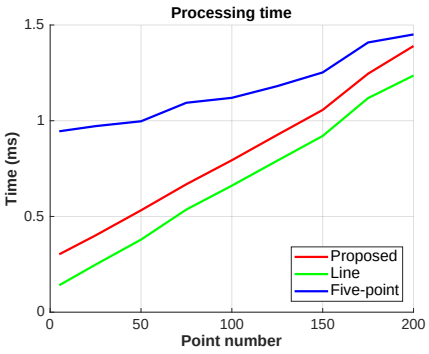
Az összehasonlított módszerek pontosságát a 21. ábra bal oldali és középső diagramjai mutatják. A kumulatív sűrűségfüggvényeket [22] (Cumulative Density Function – CDF) ábrázoltuk. Egy módszer akkor pontossabb, ha a görbéje az ábra bal oldalához közelebb van. A felső diagram a becsült eltolások, a középső pedig a forgatás pontosságát mutatja. Például a javasolt módszer (piros görbe) esetében a $20°$-nél kisebb hibájú fordítás visszaadásának valószínűsége kb. 90%. Az összes többi megoldó esetében ez az érték lényegesen alacsonyabb, körülbelül 75% körül van. Látható, hogy a *javasolt módszer lényegesen pontosabb, mint a versenytársaké*.

A teljes robusztus becslési eljárás feldolgozási ideje az összehasonlított megoldók használatával a 21 ábra jobb oldali diagramján látható. Látható, hogy a lineáris megoldó (átlagos idő $0{,}033$ mp) és a Choi-féle [31] módszer (átlagos idő: $0{,}032$ mp) vezet a leggyorsabb robusztus becsléshez. A GC-RANSAC a javasolt megoldóval kombinálva kis mértékben lassabb, átlagosan $0{,}039$ mp. Azonban még mindig jelentősen gyorsabb, mint az ötpontos algoritmust használó becslő – a "Line + Five-point" és az "Baseline" átlagos ideje 0.043 és 0.065 másodperc.

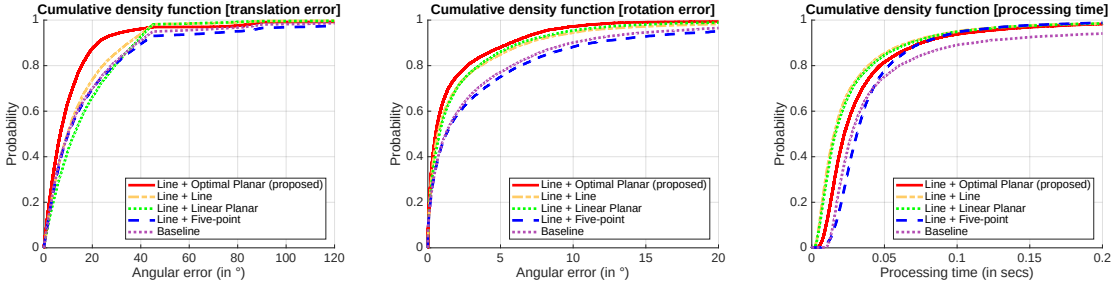
Összefoglalva, a modellparaméterek becslése nem minimális mintából a javasolt módszerrel jobb pontosságot eredményez, a feldolgozási idő jelentős többletköltsége nélkül.

A korábban már bevezett félig kalibrált kamera esetén az egyetlen belső paramé-

[22] https://en.wikipedia.org/wiki/Cumulative_distribution_function



20. ábra. *A feldolgozási idő (milliszekundumban; Matlabban implementálva) a becsléshez használt pontok számának függvényében. Az összehasonlításban a Choi és mtsai. [31] módszere (Line) és az ötpontos l algoritmus [157] (Five-point) szerepel a javasolt (Proposed) algoritmus mellett.*



21. ábra. *A szöghiba (bal - eltolás; közép - forgatás; fokban) és a feldolgozási idő (jobb oldalt, másodpercben) kumulatív sűrűségfüggvényei a Malaga-adatbázis* 15 *jelenetén (összesen* 9 064 *képpár). A valószínűségek (függőleges tengely) a szöghiba és a feldolgozási idő (vízszintes) függvényében vannak ábrázolva. A GC-RANSAC [13] robusztus becslőt használtuk a módszerek kereteként. Egy módszer akkor pontosabb, hogyha a görbéje a grafikon bal oldalához közelebb van. A módszerek elnevezését a 2. táblázat tartalmazza.*

ter a $\mathbf{K} = diag(f, f, 1)$ alakban írható; ilyenkor az alapvető mátrix alakja a következő:

$$\mathbf{F} = \mathbf{K}^{-T}\mathbf{E}\mathbf{K}^{-1} \sim \begin{bmatrix} 0 & -\frac{\sin\alpha}{f^2} & 0 \\ \frac{\sin(\alpha+\beta)}{f^2} & 0 & -\frac{\cos(\alpha+\beta)}{f} \\ 0 & \frac{\cos\alpha}{f} & 0 \end{bmatrix}.$$

Mind a lényegi, mind az alapvető mátrix ritka, a kilenc elemből mindössze négy nem nulla. Ezért a becsléshez elég egyetlen affin megfelelés, hiszen egy megfelelésből három egyenlet adódik, amiből a két mozgásparaméteren kívül még egy harmadik érték is megbecsülhető. Ez lehet például a félig kalibrált kamera fókusztávja. Ha a fenti 46. összefüggésbe a lényegi mátrix helyett a félig kalibrált kamera esetére levezetett elemeket rakjuk, a becslési feladat így módosul:

$$\begin{bmatrix} -a_{11}v_1 & 0 & a_{21}u_1 + v_2 & -a_{21} \\ -a_{12}v_1 - u_2 & 1 & a_{22}u_1 & -a_{22} \\ -u_2v_1 & v_1 & v_2u_1 & -v_2 \end{bmatrix} \begin{bmatrix} \frac{\sin\alpha}{f^2} \\ \frac{\cos\alpha}{f} \\ \frac{\sin(\alpha+\beta)}{f^2} \\ \frac{\cos(\alpha+\beta)}{f} \end{bmatrix} = 0.$$

Elnevezés	Minimális	Túlhatározott
Line + Optimal planar	Choi-módszer [31]	Javasolt optimális megoldó
Line + Line	Choi-módszer [31]	Choi-módszer [31]
Line + Linear planar	Choi-módszer [31]	Choi-féle lineáris módszer [31]
Line + Five-point	Choi-módszer [31]	Ötpontos algoritmus [157]
Baseline	Ötpontos algoritmus [157]	Ötpontos algoritmus [157]

2. táblázat. *A GC-RANSAC [13] megoldók kombinációi, amelyeket a valós kísérletek során használtunk (lásd a 21. ábrát). A 2. oszlopban a póz becsléséhez használt megoldók láthatók minimális mintából. A 3. oszlop a modell paramétereinek finomításához használt megoldókat tartalmazza nem minimális mintán.*

Jelöljük az ismeretleneket tartalmazó vektort $\mathbf{n}$-el! Ekkor

$$\mathbf{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} = \begin{bmatrix} \frac{\sin\alpha}{f^2} \\ \frac{\cos\alpha}{f} \\ \frac{\sin(\alpha+\beta)}{f^2} \\ \frac{\cos(\alpha+\beta)}{f} \end{bmatrix}.$$

Az $\mathbf{n}$ vektor elemeit párosítva írhatjuk, hogy

$$\frac{n_1}{n_3} = \frac{\sin\alpha}{\sin(\alpha+\beta)}, \frac{n_2}{n_4} = \frac{\cos\alpha}{\cos(\alpha+\beta)}.$$

A nevezőkkel átszorozva és az eredményt négyzetre emelve kapjuk, hogy

$$n_1^2 \sin^2(\alpha+\beta) = n_3^2 \sin^2\alpha,$$
$$n_2^2 \cos^2(\alpha+\beta) = n_4^2 \cos^2\alpha.$$

A $\sin^2(.) = 1 - \cos^2(.)$ trigonometrikus alapösszefüggést alkalmazva a koszinuszok négyzeteire lineáris egyenletrendszert kapunk:

$$\begin{bmatrix} -n_3^2 & n_1^2 \\ n_4^2 & -n_2^2 \end{bmatrix} \begin{bmatrix} \cos^2\alpha \\ \cos^2(\alpha+\beta) \end{bmatrix} = \begin{bmatrix} n_1^2 - n_3^2 \\ 0 \end{bmatrix}.$$

Ezzel megoldottuk a feladatot, a szinuszok négyzeteire katunk összefüggéseket. Két fontos dolgot azonban meg kell említeni:

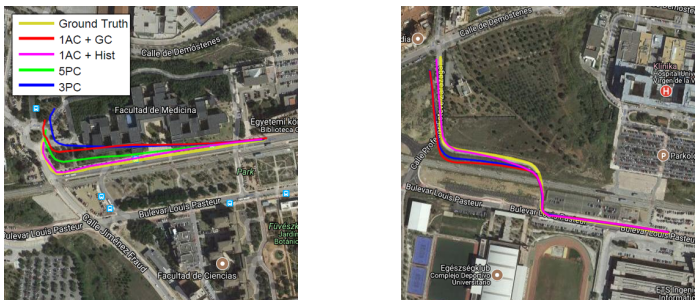
– Miután az elemeknek a négyzetét kapjuk meg, több megoldás is kapunk, hiszen az előjelet a gyökvonás után nem tudjuk egyértelműen meghatározni. A hagyományos sztereó problémához hasonlóan [62] azokat az eseteket is megkapjuk, amikor a térbeli pontok háromszögelés [61] után a kamera mögé kerülnek. A láthatósági feltétel figyelembe vételével a hamis megoldásokat ki lehet szűrni.

– Néha nincsen valós megoldás. A lineáris egyenletrendszer nem tudja garantálni, hogy $\cos^2\alpha$-ra és $\cos^2(\alpha+\beta)$-ra mindig egynél nem nagyobb pozitív érték jöjjön ki, komplex szögekkel pedig a valós feladatok során nem tudunk mit kezdeni. Ilyenkor a negatív értéket nullára cseréljük, ezzel $\pm\pi/2$ kapunk a szögekre.

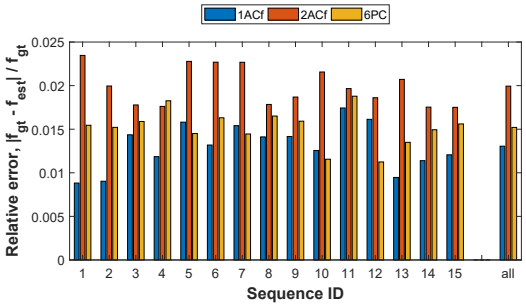
Ezt az algoritmust jelentős elméleti eredménynek tekintem, hiszen félig kalibrált kamerák esetén nem kell két külön megfelelést találni a képpárokon, hanem egyetlen lokális információból a két kép közötti elmozdulás (póz) megbecsülhető.

4.7. Tesztelési eredmények

A javasolt módszereket szintetikus környezetben és nyilvánosan elérhető valós adathalmazokon teszteljük.



22. ábra. *Kapott útvonalak a Malaga-adatbázisból. Az alapigazság (GPS-alapon) útvonal sárga, a javasolt egypontos algoritmus GC-RANSAC-kal pirossal, a hisztogramszavazással lilával, a három- és az ötpontos algoritmusok kékkel és zölddel rajzoltak.*

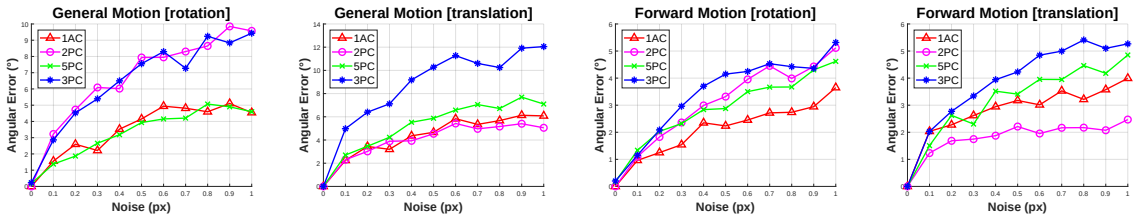


23. ábra. *Az átlagos relatív fókusztávolság-hiba, azaz* $|f_{est} - f_{textgt}|/f_{gt}$*, a javasolt (1ACf), a 2ACf [18] és a 6PC [59] módszerek* 15 *szekvenciáin (összesen* 6111 *képpár) a Malaga adathalmazon. Az utolsó sávhármas az összes szekvenciára vonatkozó átlagos hibát közli.*

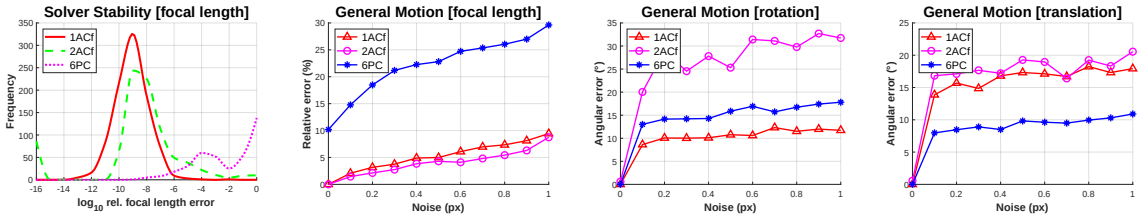
4.7.1. Konkurrens módszerek

A módszerek során a minimális megoldókat részesítettük előnyben. Ezért az összehasonlításban ezek a módszerek vettek részt:

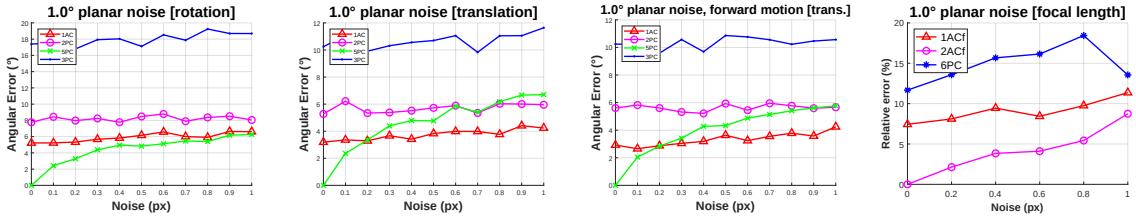
- **5PC** [121]: A klasszikusnak mondható, öt pontot használó módszer, mely egy általános lényegi mátrixot számít ki.
- **6PC** [59]: Hartley és Li speciális, hat pontot használó módszere, mely a közös fókusztávolságot is ki tudja számítani.
- **3PC** [59]: Saját, egyszerűsített algoritmus, ami a 46. egyenlet DLT-szerű [60] megoldása.
- **2PC** [31]: A Choi-féle algoritmus, amely kettő pontból számítja ki a síkmozgás két szabad paraméterét.
- **1AC** [159]: Az előző fejezetben ismertetett megoldó, amely egy affin megfelelésből (1AC) a 46. összefüggésen keresztül kiszámítja a kamera mozgását.



(a) *Kalibrált kamerák.*



(b) *Ismeretlen fókusztáv.*



(c) *Általános mozgás helytelen síkmozgás feltételezésével*

24. ábra. *Diagramok szimulált teszteredményekkel. (a) Kalibrált eset. A kapott forgatások (balra) és eltolások (jobbra) átlagos szöghibája (°) a σ zaj függvényében ábrázolva. A módszereket minden σ esetében 1000 alkalommal futtattuk le. Az összehasonlított módszerek: a javasolt algoritmus(1AC), Kétpontos módszer síkmozgást feltételezbe [31] (2PC), David Nistér [122] 5 pontos algoritmusa (5PC) az általános kameramozgásra és a lineáris hárompontos algoritmus (3PC) síkmozgásra. Az első két ábrához általános síkbeli mozgást vettünk figyelembe. Az utolsó kettőnél a kamerák forgatás nélküli előre irányuló mozgást végeztek. (b) Fókusztávolság becslés. Az összehasonlított módszerek a javasolt (1ACf), a saját általános kamerára fejlesztett algoritmusunk [18] (2ACf) Hartley-Li hatpontos módszere [59] (6PC). A becsült fókusztávolságok relatív hibáinak (vízszintes) $\log_{10}$ gyakoriságát (1000 futás; függőleges tengely) zajmentes esetben a bal oldalon mutatjuk. A következő diagram a relatív fókusztávolság-hiba ($\%$-ban) látható a zaj σ függvényében. A kapott forgatások (3. diagramm) és eltolások (utolsó kép) átlagos szöghibáit (°) a σ zaj függvényében ábrázoljuk. (c) Elrontott síkbeli mozgás: a második kamera függőleges irányú elforgatása történt egy $\sigma = 1.0°$ véletlenszerű forgatással. Az első két diagram az általános síkbeli mozgás esetén a forgások és az eltolások hibáit mutatja. A harmadik oszlopban az előremenő mozgás hibáit láthatjuk. Az utolsó képen a becsült fókusztávolság hibája található. Látható, hogy a javasolt 1AC és 1ACf módszerek a sérült síkbeli kényszer ellenére is meglehetősen jól működnek.*

	1ACf				2PC [31] & 1AC				3PC [23]				5PC [122]			
steps	4×4 EIG				2×2 INV + 2×2 SVD.				3th degree poly. + conic intersect				5×9 SVD + 10×10 Gauss-J + 10×10 EIG			
1 iter	$4^3 = 64$				$2^3 + 2^3 = 16$				$3^2 \log(3) + 12 \approx 22$				$9 \cdot 5^2 + 10^3 + 10^3 = 2{,}225$			
m	1				2				3				5			
1 - μ	0.25	0.50	0.75	0.90	0.25	0.50	0.75	0.90	0.25	0.50	0.75	0.90	0.25	0.50	0.75	0.90
# iters	4	7	17	44	6	17	72	459	9	35	293	4,603	17	146	4,714	$\sim 5 \cdot 10^5$
# comps	256	448	1,088	2,816	96	272	1,152	7,344	198	770	6,446	101,266	37,825	324,850	$\sim 10^7$	$\sim 10^9$

3. táblázat. *Négy különböző megoldó (négy blokk, mindegyik négy oszlopból áll) elméleti számítási komplexitása egy RANSAC [46]-szerű robusztus keretrendszerben. A saját módszer az első oszlop, a módszert **1ACf**-el jelöltük. A felírt jellemzők a következőek: az egyes megoldók műveleteinek száma (#steps; első sor); egy iterációra a becslés számítási bonyolultsága (#iter; második sor); a becsléshez szükséges megfeleltetések száma (m; harmadik sor); a lehetséges kiugró pontok számának arányok ($1 - \mu$; negyedik sor); a RANSAC-hoz szükséges iterációk száma, ha a szükséges megbízhatóságot 99-%-ra állítjuk (ötödik sor); és a teljes eljáráshoz szükséges számítási becsült darabszáma (# comps; hatodik sor).*

- **1ACf** [160]: A jelenlegi fejezetben ismertetett, egy affin megfelelésből (1AC) működő megoldó, mely az f fókusztávolságot is kiszámítja.
- **2ACf** [19]: Két affin megfelelésből (2AC) a lényegi mátrixot és az f fókusztávolságot kiszámító módszerünk, melyet a 4.5. szakaszban ismertettem.

4.7.2. Számítási komplexitás

Ebben a szakaszban az algoritmusok számítási bonyolultságát hasonlítjuk össze. A 3. táblázatban az egyes oszlopok egy-egy minimális megoldó tulajdonságait mutatják. Az első sor a főbb lépéseket tartalmazza. Például az 5×9 SVD + 10×10 Gauss-J + 10×10 EIG azt jelenti, hogy a lépések a következők: egy 5×9 mátrix SVD-felbontása, egy 10×10 mátrix Gauss-Jordan eliminációja és végül egy 10×10-es mátrix sajátérték felbontása. Alatta az elméleti számítási bonyolultságokat összegezzük. A harmadikban az egyes megoldókhoz szükséges megfeleltetések száma (m) szerepel. A negyedik sorban a kiugró pontokra (outlier-ekre) beállított értékeket adjuk meg. Utána a RANSAC [46] algoritmushoz kiszámított, elméleti iterációszáma van felírva minden egyes kiugró arányra, a szükséges konfidenciát 0.99-ra állítva. Az utolsó sor a RANSAC és a minimális módszerek kombinált számítási bonyolultságát mutatja: ez az érték egy iteráció bonyolultsága szorozva az iterációs számmal. Látható, hogy *a javasolt 1ACf és 1AC módszerek vezetnek a legkevesebb iterációhoz* és a legkisebb számítási bonyolultsághoz.

4.7.3. Szintetikus teszteredmények

A javasolt módszer teszteléséhez és a legkorszerűbb eljárásokkal való összehasonlításához teljesen ellenőrzött környezetben két perspektivikus kamerát generáltunk a síkmozgás feltételezésével. A kamerák közös belső paraméterekkel rendelkeztek: $f_x = f_y = 600$ (fókusztávolság) és $[300 \quad 300]^{\mathrm{T}}$ a döféspont. Minden vizsgálatban egy véletlen síkot generáltunk, és a síkból 50 véletlen pontot vetítettünk a kamerákba. A pontkoordinátákhoz zéró várható értékű Gauss-zajt adtunk σ szórással. Az egyes pontmegfeleletések affin paramétereinek meghatározásához a homográfiát a zajos megfeleltetésekből becsültük meg. Ezután a zajos affin paramétereket minden egyes megfeleltetéshez a megfelelő homográfiából számítottuk [9].

A kapott forgatás hibája a $\mathbf{R}_{\text{est}}\mathbf{v}$ és $\mathbf{R}_{\text{gt}}\mathbf{v}^{\text{T}}$ vektorok közötti szög az alábbiak szerint: $\epsilon_{\text{R}} = \cos^{-1}((\mathbf{R}_{\text{est}}\mathbf{v})^{\text{T}}(\mathbf{R}_{\text{gt}}\mathbf{v}))$, ahol $\mathbf{v} = [\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]$, $\mathbf{R}_{\text{est}}$ a becsült és $\mathbf{R}_{\text{gt}}$ az alapigazság forgatási mátrixa. Az eltolás hibáját hasonló módon számoljuk ki – ez a szög az alapigazság és a kapott forgatási vektorok között. A közölt értékek 1000 futtatás átlagai minden egyes σ zajra.

Összehasonlítás kalibrált esetben. Kalibrált kamerák esetén a α és β szögeket kell megbecsülni. Az eredményeket a 24a. ábra szemlélteti. Az összehasonlított algoritmusok a következőek: (i) a javasolt megközelítés (1AC); (ii) David Nistér [121] ötpontos algoritmusa (5PC); (iii) Kétpontos algoritmus [31] (2PC) és a hárompontos lineáris algoritmus (3PC). Az 5PC esetében a lényegi mátrixot öt pont megfeleltetéséből számítjuk ki, általános háromdimenziós mozgást feltételezve.

A 24a. ábrán a szöghiba (fokban) a zaj σ függvényében van ábrázolva. Az első két ábrán az általános síkbeli mozgást vettük figyelembe, amely a kapott elforgatások és eltolások hibáit mutatja. Látható, hogy az 1AC mindkét esetben a második legpontosabb. Az első azonban különbözik ebben a két esetben. Azt is megvizsgáltuk, hogyan viselkednek a módszerek, ha a kamerák tisztán előre irányuló mozgást végeznek (forward motion), és a Y tengely körüli forgás nulla. Lásd a 24a. ábra utolsó két grafikonját. Ebben az esetben az 1AC kapja a legpontosabb forgatásokat és a második legpontosabb eltolásokat.

A 24c. ábra első három grafikonján a síkbeli kényszert úgy rontottuk el, hogy a második kamera függőleges irányát egy 1° szórású véletlen szöggel elforgattuk. Ezért a problémának több (három) szabadságfoka van, mint amit a síkmozgást feltételező módszerek általában figyelembe vesznek. Látható, hogy a síkbeli mozgást feltételező módszerek közül a javasolt módszer a legrobusztusabb az ilyen típusú zajjal szemben. Ha a koordináták zaja $> 0{,}2$ pixel, akkor az 1AC a legpontosabb eltolást és a második legpontosabb elforgatást kapja. Ha a zaj kisebb, mint $0{,}2$ pixel, akkor az általános ötpontos módszer [122] a legpontosabb.

Összehasonlítás félig kalibrált kamerákkal. Ebben a tesztforgatókönyvben a mozgás két szögét és a két kamera közös fókusztávolságát becsüljük meg. Az eredmények a 24b. ábra második sorában láthatók. Három módszert hasonlítunk össze: (i) a javasolt megközelítést (1ACf), (ii) a két affin megfelelést használó módszert [18] (2ACf) általános kameramozgásra; (iii) és egy hatpontos technikát [59] (6PT). A bal oldali ábra a megoldók numerikus stabilitását közli: a $\log_{10}$ relatív hibák (vízszintes) gyakoriságai (függőleges tengely; 1000 futtatásban), azaz $f_{\text{rel}} = (f_{\text{est}} - f_{\text{gt}})\,/\,f_{\text{gt}}$, zajmentes esetben.

Látható, hogy a 2ACf és a 6PC módszerek esetében a 10^{-2} – 10^{0} körüli hibák gyakorisága nem nulla. Ezért vannak olyan esetek, amikor a kívánt fókusztávolságot ezek a módszerek még zajmentes esetben sem állítják vissza, azaz numerikusan nem elég stabilak. A javasolt 1ACf ezzel ellentétben teljesen stabil, a hibák legsűrűbb pontja 10^{-9} relatív hiba környékén van zajmentes esetekben.

A 24b. ábra másik három grafikonja szerint az 1ACf a második legpontosabb fókusztávolságot adja, kissé lemaradva a 2ACf mögött. Az új módszer a legpontosabb forgásbecsléshez vezet, a becsült eltolások esetén a második legjobb eredményt érte el. A 24c. ábra jobb oldali grafikonja azt mutatja, hogy az 1ACf a második legjobb fókusztávolságokat kapja, még a sérült síkviszonyok ellenére is.

	1AC + GC	1AC + Hist	3PC	5PC
Idő (ms)	749 ± 672	$\mathbf{16} \pm 10$	598 ± 530	588 ± 527
Idő/Minta #	$\mathbf{23} \pm 16$	–	31 ± 64	38 ± 32

4. táblázat. *A GC-RANSAC és hisztogramszavazás feldolgozási ideje (milliszekundumban; C++ implementáció) és a minták számának átlaga és szórása a Malaga-adatkészleten. A megfeleltetések átlagos száma* 4070*. Az első sor a teljes időt, a második véletlen mintánként az időt tartalmazza. A GPU-ASIFT [33] által végzett affin jellemzőkivonatolás átlagos feldolgozási ideje* 24 *ms. A teszteléshez egy NVIDIA GeForce GTX 980 GPU-t alkalmaztunk.*

4.7.4. Valós teszteredmények

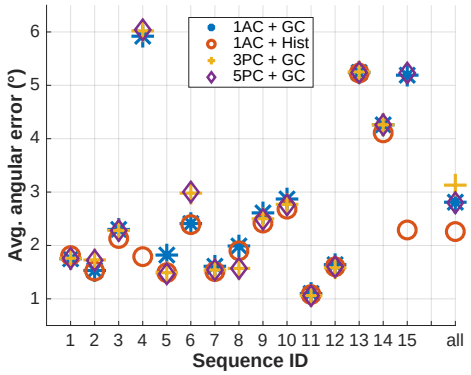
A javasolt módszer valós adatokon való teszteléséhez a Malaga[23] [25] adathalmazt választottuk. Ezt az adathalmazt teljes egészében városi környezetben gyűjtötték össze. Egy kamera felvételeit emeltük ki, és csak minden tizedik képet használtuk fel. 15 különböző útvonalat tartalmaz az adatbázis, így mindösszesen 6 képpáron teszteltük őket. Végül az eredményeket egyszerűen összefűztük. Hogy kizárólag a minimális megoldók hatását mutassuk be, mindegyiküket a teljes útvonal optimalizálása nélkül alkalmaztuk az egymást követő képpárokra. Az alapigazság útvonalát az adatkészletben megadott GPS-koordináták felhasználásával állítottuk össze.
Kalibrált kamerák. Ugyanazokat a módszereket választottuk ki az összehasonlításra, mint szimulált tesztek esetében. Megjegyzendő, hogy Scaramuzza [148] egypontos algoritmusa is versenytárs lehet, azonban ez speciális kamerabeállítást igényel, azaz a kamerának a hátsó tengely felett kell lennie. Ez a feltétel nem áll fenn a Malaga adatbázisnál [25]. Az affin megfeleléseket az Affine-SIFT [183] (ASIFT) módszer segítségével detektáljuk. Az egymást követő képkockák közötti mozgás abszolút hosszát a megadott GPS-koordinátákból számoltuk ki, hiszean a kamera alapú látásnál csak az elmozdulás irányát lehet rekonstruálni, a hosszt magát nem. Robusztus becslőként a hisztogramszavazást és a Graph-Cut RANSAC [13] algoritmust választottuk. Utóbbi doktoranduszom fejlesztése, és nemcsak korszerű, hanem forráskóddal együtt nyilvánosan elérhető[24]. A GC-RANSAC esetében a szükséges konfidenciát 0.99-ra állítottuk be, és a szerzők által javasolt alapértelmezett beállítást használtuk.

A 25. ábra a Malaga-adatkészlet egyes szekvenciáinak átlagos szöghibáját (fokban; függőleges tengely) mutatja. Összesen $6{,}111$ képpáron teszteltük. Látható, hogy a javasolt módszer átlagosan felülmúlja a versenytársak algoritmusát a geometriai pontosság tekintetében.

A 4. táblázat az egyes megoldókkal kombinált GC-RANSAC által igényelt feldolgozási időt és a minták számát mutatja. A GC-RANSAC a javasolt algoritmussal a legkevesebb mintaszámot eredményezi, azonban a feldolgozási idő kissé magasabb, mint a 3PC és 5PC algoritmusoké. A hisztogramszavazás robusztus becslőként való használata nagyságrendi gyorsulást eredményez.

A tulajdonságok kinyerése a GPU-ASIFT [33] implementációnkkal egy NVIDIA GeForce GTX 980 processzoron $24{,}07$ ms időt vesz igénybe minden egyes 1024×768 méretű képpárra. A teljes folyamat képenként hozzávetőlegesen 40 milliszekundumot

[23] `https://www.mrpt.org/MalagaUrbanDataset`
[24] urlhttps://github.com/danini/graph-cut-ransac



25. ábra. *Átlagos szöghiba (fokban; függőleges tengely) a Malaga-adatkészlet egyes szekvenciáin (vízszintes tengely). Az összehasonlított módszerek a javasolt (1AC), a 3 pontos (3PC) és az 5 pontos (5PC) algoritmusok. A „GC" utótag azt jelzi, hogy a GC-RANSAC [13] robusztus becslőt alkalmaztuk, míg a „Hist" a súlyozott hisztogramszavazást [28] jelenti. Az utolsó oszlop az összes jelenet (összesen* 6111 *képpár) átlagát mutatja.*

vesz igénybe, ami *valós idejű feldolgozásként értelmezhető*.

Tesztelés félig kalibrált kamerákkal.

A becsült fókusztávolságok pontosságát szintén a Malaga adathalmazon teszteltük. A megadott belső kalibrációk miatt a fókusztávolságokat ismerjük, tehát az alapigazság rendelkezésre áll. A javasolt 1ACf, a 2ACf [18] és a 6PC [59] módszereket minden egymást követő képpárra alkalmaztuk. A robusztus fókusztávolság-becsléshez hisztogramszavazást használtunk. A 23. ábra a relatív hibákat (függőleges tengely) az adatkészlet minden egyes szekvenciájára (vízszintes tengely) mutatja. A relatív hiba $|f_{\text{est}} - f_{textgt}|/f_{\text{gt}}$, ahol f_{est} a becsült, f_{gt} pedig az alapigazság fókusztávolság. Az algoritmusokat összesen 6111 képpáron teszteltük. Látható, hogy a javasolt 1ACf megoldó a 15 képsorozatból 11 esetében a legpontosabb fókusztávolsághoz vezet. Emellett az 1ACf átlagos hibája az összes szekvenciára vetítve a legalacsonyabb.

4.8. Affin transzformáció becslése egyenesekből

A korábbi algoritmusok mind azt feltételezték, hogy a képrészek közötti affin transzformáció rendelkezésre áll. Ahogyan az a tesztelésekből kiderült, a becslések érzékenyek az affin transzformáció zajára, ezért azok pontos, jó minőségű kinyerése a képpárokból kritikus kérdés. Ez a feladat gyakorlati szempontból azonban meglehetősen nehéz feladat. A jellegzetes pont leírók [103, 21, 142] a skáláról és az orientációról csak elnagyoltan (kvantáltan) képesek értéket tárolni, ezért ezekből nem mindig lehet a pontos affin paramétereket kinyerni.

Ebben a fejezetben azt mutatom meg, hogy amennyiben egyeneseket meg tudunk feleltetni egymásnak, ezek irányát felhasználva hogyan lehet megbecsülni az affin transzformációt.

Ebben a fejezetben két dolgot mutatok meg:

– hogyan lehet felhasználni az egyenesirányokon túlmenően az alapvető mátrixot; és

– optikai áramlás miként tud segíteni a becslésben.

Affin transzformáció meghatározása egyenesekből és az alapvető mátrixból.

Először azt nézzük meg, hogyan határozzák meg az egyenesirányok az affin paramétereket. A pont helyét jelöljük a két képen $\mathbf{p_1} = \begin{bmatrix} u_1 & v_1 \end{bmatrix}^T$-el és $\mathbf{p_2} = \begin{bmatrix} u_2 & v_2 \end{bmatrix}^T$-vel, az irányt pedig $\mathbf{d}_{1i} = \begin{bmatrix} l_{1u} & l_{1v} \end{bmatrix}^T$-vel és $\mathbf{d}_{2i} = \begin{bmatrix} l_{2u} & l_{2v} \end{bmatrix}^T, i \in \{1,2\}$-vel.

Az $\mathbf{A}$ affin transzformációt a megszokott módon egy 2×2 mátrix írja fel.

Affin transzformáció becslése

A pontok helyei, az affin transzformáció és az alapvető mátrix közötti kapcsolatot a 24. egyenlet adja meg. A koordinátákat behelyettesítve a következő képletet kapjuk:

$$\mathbf{A}^T \tilde{\mathbf{F}} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = -\hat{\mathbf{F}}^T \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}.$$

Mivel az alapvető mátrix és a pontok helyének szorzata adja az epipoláris vonalat a második képen, és a mátrix és a második pont helyének transzponálása adja a megfelelő epipoláris vonalat az első képen, a vektoregyenlet átírható a következőképpen:

$$\mathbf{A}^T \begin{bmatrix} l_{2u} \\ l_{2v} \end{bmatrix} = - \begin{bmatrix} l_{1u} \\ l_{1v} \end{bmatrix},$$

ahol az epipoláris egyenesek normálisai az $\mathbf{n}_1 = [l_{1u} \quad l_{1v}]^T$ és $\mathbf{n}_2 = [l_{2u} \quad l_{2v}]^T$ vektorok.

Ha az affin transzformációk elemeit behelyettesítjük, a következő lineáris egyenletrendszert kapjuk:

$$\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} l_{2u} \\ l_{2v} \end{bmatrix} = - \begin{bmatrix} l_{1u} \\ l_{1v} \end{bmatrix}. \tag{47}$$

Most az ismert irányok közötti kapcsolatot az affin paraméterekkel együtt írjuk fel. Az irányokat az affin mátrix helyesen transzformálja, azonban a vektorok hossza nem ismert. Ez a tény a következőképpen fogalmazható meg:

$$\begin{aligned} \mathbf{A} \begin{bmatrix} u_{11} \\ v_{11} \end{bmatrix} &= \alpha_1 \begin{bmatrix} u_{21} \\ v_{21} \end{bmatrix}, \\ \mathbf{A} \begin{bmatrix} u_{12} \\ v_{12} \end{bmatrix} &= \alpha_2 \begin{bmatrix} u_{22} \\ v_{22} \end{bmatrix}, \end{aligned}$$

ahol α_1 és α_2 az ismeretlen hosszok.

Az alak egyszerűen átírható az affin transzformációk elemeinek behelyettesítésével:

$$\begin{aligned} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} u_{11} \\ v_{11} \end{bmatrix} &= \alpha_1 \begin{bmatrix} u_{21} \\ v_{21} \end{bmatrix}, \\ \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} u_{12} \\ v_{12} \end{bmatrix} &= \alpha_2 \begin{bmatrix} u_{22} \\ v_{22} \end{bmatrix}. \end{aligned} \tag{48}$$

A végső probléma a 47. és a 48 egyenletek összevonásával alakítható ki. A kapott

alak lineáris, ezt írja fel a következő hatdimenziós egyenlet:

$$\begin{bmatrix} l_{2u} & 0 & l_{2v} & 0 & 0 & 0 \\ 0 & l_{2u} & 0 & l_{2v} & 0 & 0 \\ u_{11} & v_{11} & 0 & 0 & -u_{21} & 0 \\ 0 & 0 & u_{11} & v_{11} & -v_{21} & 0 \\ u_{12} & v_{12} & 0 & 0 & 0 & -u_{22} \\ 0 & 0 & u_{12} & v_{12} & 0 & -v_{22} \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} -l_{1u} \\ -l_{1v} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{49}$$

A megoldást triviálisan úgy kapjuk meg, hogy a jobb oldali vektort megszorozzuk a bal oldali együtthatómátrix inverzével.

4.8.1. Affin transzformációk becslése optikai áramlásból

Az affin transzformációk más technikákkal is becsülhetők, például affin-invariáns megfeleltetők [184] segítségével. Tapasztalataink szerint azonban ezek minősége nem kielégítő, mert a becsült transzformációk zajjal erősen szennyezettek.

Ezért az affin transzformációk kinyerésének egy másik módját javasoljuk: az affin transzformációkat akkor becsüljük meg, ha a képek közötti optikai áramlás rendelkezésre áll. A becslési probléma egy inhomogén lineáris probléma, ahogyan azt itt tárgyaljuk. Ezért a becslés nagyon gyors, hiszen egy hatdimenziós problémának megfelelő mátrix pszeudo-inverzét kell kiszámítani.

Feladat leírása.

Az optikai áramlás adott, így az egyes kamerák relatív eltolódása ismert. Ezért írhatjuk, hogy

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix},$$

ahol a $[x_i \quad y_i]^T$ és a $[x_i' \quad y_i']^T$ vektorok az első és a második kép pixelkoordinátáit jelölik. Magát az áramlást az $[\Delta x_i \quad \Delta y_i]^T$ eltolási vektorok jelölik.

A feladat az affin transzformáció becslése az adott pont $\mathbf{x}_0 = [x_0 \quad y_0]^T$ helyének környezetében. A 'környezete' ebben az esetben azt jelenti, hogy a szomszédos képpontokat is figyelembe kell venni. A pontokat mi egy R sugarú korongban választjuk ki, ahol R az algoritmus statikus paramétere.

Javasolt becslés. Az affin transzformáció a megfelelő szomszédos pontok közötti kapcsolatokat a következőképpen írja le:

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_u \\ a_{21} & a_{22} & a_v \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}.$$

Írhatjuk, hogy

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} a_u \\ a_v \end{bmatrix}.$$

Másik alakba is írhatjuk a tömörség kedvéért:

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_i & y_i & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_u \\ a_{21} \\ a_{22} \\ a_v \end{bmatrix} = \begin{bmatrix} x'_i \\ y'_i \end{bmatrix}.$$

A bal oldali mátrixot jelöljük $\mathbf{C}_i$-vel. Ekkor

$$\mathbf{C}_i \begin{bmatrix} a_{11} & a_{12} & a_u & a_{21} & a_{22} & a_v \end{bmatrix}^T = \mathbf{x'_i}.$$

Ha N különböző ponton ismerjük az áramlást, a probléma túlhatározottá válik:

$$\begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \vdots \\ \mathbf{C}_N \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_u \\ a_{21} \\ a_{22} \\ a_v \end{bmatrix} = \begin{bmatrix} \mathbf{x'}_1 \\ \mathbf{x'}_2 \\ \vdots \\ \mathbf{x'}_N \end{bmatrix}.$$

A feladat így lineárissá válik:

$$\mathbf{C} \begin{bmatrix} a_{11} & a_{12} & a_u & a_{21} & a_{22} & a_v \end{bmatrix}^T = \mathbf{x'},$$

ahol $\mathbf{C} = [\mathbf{C_1^T} \dots \mathbf{C_N^T}]^T$ és $\mathbf{x} = [\mathbf{x_1^T} \dots \mathbf{x_N^T}]^T$.

Az affin paraméterek legkisebb értelemben vett optimális becslését a következőképpen kapjuk meg:

$$\begin{bmatrix} a_{11} & a_{12} & a_u & a_{21} & a_{22} & a_v \end{bmatrix}^T = \left(\mathbf{C^T C}\right)^{-1} \mathbf{C}^T \mathbf{x'}.$$

Megjegyzendő, hogy legalább három különböző pontra ismerni kell az áramlást. Háromnál több pont esetén a probléma túlhatározottá válik.

4.8.2. Vizsgálati eredmények

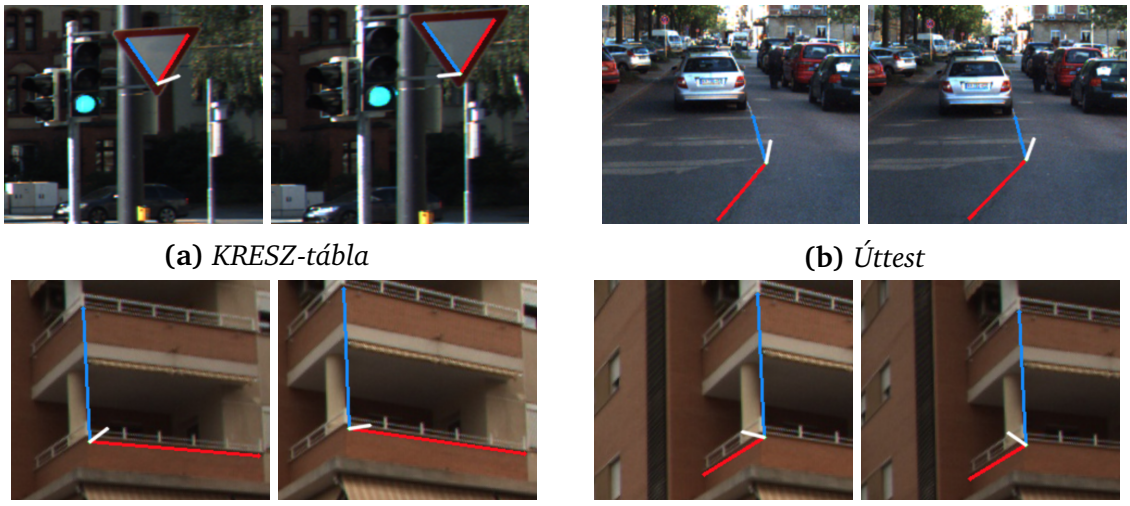
A tesztelés során elsősorban valós esetekre koncentráltunk, mivel a munkánk fő célja a felületi normális becslés beillesztése egy 3D rekonstrukciós csővezetékbe.

Szintetikus tesztelés

A szintetikus tesztet azért készítettük, hogy igazolni tudjuk, hogy a 49. egyenletben megadott képlet helyes. Ehhez egy egyszerű szintetikus tesztkörnyezetet implementáltunk az Octave [25] programcsomag segítségével. A kamera paramétereit, valamint a térbeli jelenet geometriáját, ami tesztünkben egy szabályos gömb, véletlenszerűen generáltuk. A pontok helyét a kameraképen általában perspektív vetítés [62] segítségével adtuk meg. Az alapigazság szerinti affin transzformációkat a gömb érintő síkjain keresztül generáltuk. Az affin paraméterek a megfelelő ponthelyeken az érintő síkokhoz kapcsolódó homográfiák levezetésével határozhatók meg, ahogyan azt 2015-ös munkánkban [15] részletesen tárgyaljuk.

A szintetikus tesztek fő tanulsága. Sikeresen igazoltuk, hogy a 49. egyenlet helyes, az alapigazság affin transzformációit mindig pontosan megkaptuk.

[25] Octave egy nyílt forráskódú MATLAB-klón. Lásd: http://www.octave.org.



(a) *KRESZ-tábla* **(b)** *Úttest*

(c) *Erkély (1)* **(d)** *Erkély (2)*

26. ábra. *Eredmények valós képpárokon. A kézzel kiválasztott irányok piros és kék színnel vannak jelölve. A becsült felületi normálisok fehér színnel rajzoltak.*

27. ábra. *Becsült felületi normálisok az egyik KITTI [52] szekvencia képén. A sárga, fehér és piros koordináták a vízszintes, függőleges és a képsíkra merőleges irányokat jelölik. Az ábrát érdemes színes nyomtatásban vagy elektronikus formátumban nézni.*

Vizuális "Debugger"

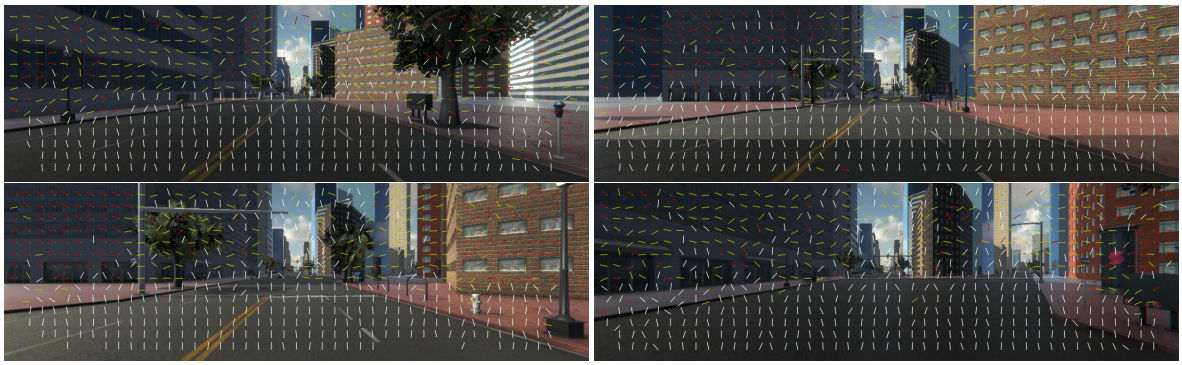
Kifejlesztettünk egy eszközt a rekonstrukciós csővezeték futtatására és a kiszámított felületi normálisok megjelenítésére. Ezt az eszközt angolul "Visual Debugger"-nek neveztük el. A pontmegfeleléseket, valamint az irányokat manuálisan választottuk ki a detektálási hibák elkerülése érdekében. Az alapmátrixot automatikusan becsültük a nyolcpontos módszerrel [60]. A stabilizálás a szabványos RANSAC [46] módszerrel történt.

A vizuális debugger néhány eredménye látható az 26 ábrán. Módszerünket a KITTI [52] és a Malaga [25] adathalmazokon teszteltük.

Felületi normálvektorok

Teljesen automatikus tesztelési eljárást is végeztünk a teljes rekonstrukciós csővezeték felhasználásával. Egy példa a normálvektorokra a 27. ábrán látható. A normálvektorokat különböző színnel rajzoltuk: a normálisok koordinátáinak abszolút értékeit vettük figyelembe; ha egy vektornak a legnagyobb abszolút koordinátája az x, y, vagy a z, akkor rendre sárga, fehér, illetve piros színnel rajzoltuk ki.

Normálvektor becslése optikai áramlásból. Bár e részfejezet középpontjában az áll, hogy megmutassuk, hogyan lehet az affin transzformációkat két megfelelő irányból és az alapmátrixból megbecsülni, itt azt is bemutatjuk, hogy más módon is hatékonyan lehet becsülni az affin transzformációkat. A 4.8.1. fejezetben áttekintettük, hogyan lehet egy transzformációt egy adott helyen meghatározni, ha adott a képek közötti optikai áramlás ismert. A felületi normálisok a 4.1. fejezetben javasolt opti-



28. ábra. *Az LG-SVL [3] szimulátor által generált képeken megjelenített becsült felületi normálisok. A sárga, fehér és piros koordináták a vízszintes, függőleges és a képsíkra merőleges irányokat jelölik. Az affin transzformációkat az optikai áramlásból számítjuk. Az ábrát érdemes színes nyomtatásban vagy elektronikus formátumban nézni.*

mális módszerrel becsülhetők.

A képsorozatokat az LG-SVL szimulátor [3] segítségével készítettük. Az optikai áramlásokat a HD3 előre betanított mély hálózat [182] számította ki. Az eredményül kapott képek a 28. ábrán láthatóak, a becsült felületi normálisok kirajzolódnak. Bár az affin transzformációk minden egyes pixelhelyzetre becsülhetők, a mintavételezéshez a könnyebb értelmezhetőség miatt szabályos ritkás rácsot alkalmazunk.

4.9. Összefoglalás

A disszertációnak ebben a fejezetében – amely egyébként a leghosszabb az összes között –, megmutattam, hogy az elmúlt tíz évben milyen geometriai törvényszerűségeinkre épülő becslő algoritmusokat sikerült bemutatnunk a tudományos közösségnek. A munkából az alábbi téziseket sikerült kimondani:

II.1. tézis. Normálvektor optimális becslése affin transzformációból, kalibrált kamerapár esetén [15, 9, 10, 110, 56].

Az I.1 tézis szerinti első törvény alapján Baráth Dániel doktoranduszommal közösen az alábbiak szerint definiáltuk az optimális normálvektorbecslés problémáját legkisebb négyzetes értelemben:

$$\mathbf{n} = \arg_{\mathbf{n}} \min \sum_{i=1}^{2} \sum_{j=1}^{2} \left(\frac{\mathbf{n}^T \mathbf{w}_{ij}}{\mathbf{n}^T \mathbf{w}_c} - a_{ij} \right)^2 , \tag{50}$$

ahol

$$\begin{aligned} \mathbf{w}_{11} &= (\nabla \Pi_v^1 \times \nabla \Pi_u^2) , \quad \mathbf{w}_{12} = (\nabla \Pi_u^2 \times \nabla \Pi_u^1) , \\ \mathbf{w}_{21} &= (\nabla \Pi_v^1 \times \nabla \Pi_v^2) , \quad \mathbf{w}_{22} = (\nabla \Pi_v^2 \times \nabla \Pi_u^1) , \\ \mathbf{w}_c &= (\nabla \Pi_u^1 \times \nabla \Pi_v^1) . \end{aligned} \tag{51}$$

Megmutattuk, hogy ez a feladat legkisebb négyzetes értelemben optimálisan megoldható. Három eljárást is kidolgoztunk, a második és a harmadik eljárás esetén kutatócsoportunk kollégáinak segítségét is igénybe vettünk:

– Negyedfokú polinom segítségével.

– Harmadfokú polinom segítségével.[26]

– Lineáris összefüggések hányadosaként.[27]

Az újabb módszerek egyre gyorsabbak, a futási idő csökkentése motiválta az újabb eljárások kifejlesztését.

II.2. tézis. Lyukkamera vetítő mátrixának becslése affin transzformációból.

Eichhardt Iván kollégámmal, kutatócsoportunk oszlopos tagjával közösen megmutattuk, hogy az I.1 tézis szerinti első törvénye alapján – amelyet a 22. összefüggés ír le –, ha ismerjük az affin transzformációt, a felületi pontokat és normálvektorokat, továbbá a vetületeket a képeken, akkor lyukkamerára a projektív mátrix paramétereit meg tudjuk határozni az egyik kamerára, ha a másik kamerának belső paramétereit ismerjük. Erre alapozva kidolgoztunk egy algoritmust projektív mátrix becslésére.

A kalibrációs algoritmus működőképességét kamera-projektor páros kalibrációján keresztül mutattuk meg. A projektor maga inverz kamerának fogható fel, amely nem képet (leképzést) készít a háromdimenziós világról, hanem a síkbeli mintát vetíti ki a térbe.

II.3. tézis. Homográfia becslése affin transzformációkból [17, 16].

Az I.1. altézis második törvényszerűsége négy egyenletet ír fel, amely összeköti a homográfiát és a lokális affin transzformációkat lyukkamera modell alkalmazása esetén. Az összefüggésekből Baráth Dániel doktorandusszal két becslő algoritmust készítettünk:

1. **HA algoritmus.** Általános esetben, ha a kamerák paraméterei teljesen ismeretlenek, két affin megfelelésből ki tudjuk számolni a kapcsolódó homográfiát; szemben a hagyományos, kizárólag pontmegfeleléseket alkalmazó módszerrel [62], amelynek legalább négy megfelelésre van szüksége.

2. **HAF algoritmus.** Ha a két kép között az alapvető (fundamentális) mátrix ismert, egyetlen affin transzformáció is elég a becsléshez.

A javasolt módszerek univerzálisak abban az értelemben, hogy mind a minimális, mind a túlhatározott esetekben alkalmazhatóak.

II.4. tézis. Síkok klaszterezése lokális affin transzformációk segítségével, kalibrált kamerák esetén [35].

A II.3. altézisben kimondott HAF algoritmus segítségével kidolgoztunk Baráth Dániellel és Jiri Matas cseh kollégával közösen egy szegmentáló eljárást, amely az azonos síkhoz tartozó affin megfeleltetéseket csoportosítja, hiszen egyetlen megfelelésből a homográfiát meg tudja becsülni a HAF algoritmus, ha a két kép között az epipoláris geometriát leíró fundamentális mátrix ismert. A kiszámított homográfia alapján a hozzá tartozó térbeli síkot meg lehet határozni, és az azonos síkokat eredményező lokális affin transzformációk kiválasztása alapján a síkfelületet a képen szegmentálni lehet. Ez a módszer a HAF algoritmus kiterjesztésének tekinthető több (multi) modell illesztésére.

[26] A levezetés megalkotásában Nghia Le Minh BSc-s hallgató segítségét kell kiemelni, aki a formális bizonyítást végső alakját megalkotta.

[27] Itt Lóczi Lajos kollégám segítségét kell kiemelnem, aki a Mathematica alkalmazását vetette be, és a leggyorsabb megoldót így találta meg.

II.5. tézis. Lényegi és alapvető mátrix becslése lokális affin transzformáció alapján [18, 19].

Az I.2-es tézis egy vektoros összefüggést ad, amelyik egy lokális affin transzformációt és a fundamentális mátrixot összeköti. Ennek az egyenletnek a felhasználásával doktorandusz halgatómmal közösen készítettünk egy homogén lineáris egyenletrendszert, amelyből legkisebb négyzetes értelemben optimális becslő eljárást alkottunk meg. Az affin összefüggésből két homogén egyenletet kapunk, ezt egészítettük ki a pontmegfelelésekből származó összefüggéssel [62], ezért minden lokális affin transzformáció három egyenletet ad. Algoritmusunk ezért három megfelelésből, homogén lineáris egyenletrendszer megoldásán keresztül becsli meg a fundamentális mátrixot, szemben a hagyományos pontalapú algoritmussal [62], amely legalább hét megfelelés meglétét feltételezi.

A javasolt algoritmusunk minimális és túlhatározott esetben egyaránt működőképes.

II.6. tézis. Félig kalibrált kamera kalibrációja két lokális affin transzformációból [19].

A II.5-ös altézis speciális esete, amikor a kamera félig kalibrált, azaz ismerjük a döféspontot, és csak a fókusztáv ismeretlen. Ekkor a kamera belső paraméterei közül csupán egy ismeretlen, így az alapvető mátrix összesen hat szabadságfokkal bír. Ezért két lokális affin transzformáció elégséges a megoldáshoz, hiszen ezek összesen hat egyenletet adnak. Ehhez javasoltunk egy algoritmust, amelyik a homogén lineáris egyenletrendszerből származó nulltérben (magtérben) geometriai és optikai törvényszerűségeket figyelembe véve megkeresi a feltételeknek legjobban megfelelő megoldást.

II.7. tézis. Síkban mozgás optimális becslése kalibrált kamerák esetén egyetlen lokális affin transzformációból [159].

Javasoltunk Baráth Dániellel doktorandusszal egy becslő algoritmust, amelyik síkban mozgás és teljesen kalibrált kamerák esetén a síkban mozgás két szabadságfokát képes megbecsülni. A két szabadságfok a függőleges elforgatás és a síkban elmozdulás iránya. Az elmozdulásnak a nagyságát a térbeli látás törvényszerűségei miatt nem lehet meghatározni, csak az irányára lehet becslést adni.

Az algoritmusunk egy algebrai hibát minimalizál, a pontmegfelelésekből egy, a lokális affin transzformációból további kettő egyenletet lehet felírni, így összesen három egyenlet áll rendelkezésre a két szabadságfokra, ezért a feladat már egy affin megfelelés teljes megléte esetén is túlhatározott. A lineáris egyenletrendszer megoldására legkisebb négyzetes értelemben optimális algoritmust javasolunk, amely tetszőleges számú affin megfelelés esetén működni képes. Az eljárás minimális [28] és a túlhatározott eseteket egyaránt kezelni képes.

II.8. tézis. Síkban mozgás becslése félig kalibrált kamerák esetén egyetlen lokális affin transzformációból [160].

A II. 7. altézishez nagyon hasonló algoritmust dolgoztunk ki Baráth Dániellel közösen arra az esetre, ha a síkban mozgás esetén ugyanazzal a félig kalibrált kamerával veszünk fel két képet. Ekkor a két képen az egyetlen ismeretlen belső paraméter a

[28] Minimális esetet akkor kaphatunk, ha a három egyenletből az egyiket elhagyjuk, például a pontmegfeleléseket nem vesszük figyelembe, csak a két affin törvényszerűséget.

fókusztávolság.

Megmutattuk, hogy az alapvető mátrixnak így három szabad paramétere van: a közös fókusztávolság, a két kép közötti függőleges elforgatás és az eltolás iránya. Javasoltunk egy lineáris megoldót, amely egy lokális affin transzformációból képes ezt a három paramétert megbecsülni. Az eljárás túlhatározott esetre is működik, amikor egynél több transzformációból nyerünk információt, de az optimalitása a módszernek ebben az esetben nem garantált.

II.9. tézis. Lokális affin transzformációk becslése egyenesekből [110].

Nghia Le Minh ELTE-s BSc hallgatómmal közös munkánkban azt mutattuk meg, hogy a képeken egymásnak megfeleltetett egyenesek irányait felhasználva is ki lehet nyerni az affin transzformációkat. Munkánkban megkülönböztettünk skálázott és skálázatlan egyeneseket. Utóbbi esetében az egyenesek iránya adott csupán, skálázott esetben a szakaszok hosszváltozatának lokális approximációja is ismert.

Ha az alapvető mátrix adott, akkor az eipoláris egyenesek normálvektoraira fel tudunk írni egy skálázott összefüggést, ahogyan azt például a 24. összefüggésben már meg is tettük, ezáltal egy inhomogén lineáris összefüggésünk van az affin transzformációra.

Ezeknek az affin transzformációknak a kinyeréséhez többféle algoritmust készítettünk. A becslési módszereink a lineáris részét képesek meghatározni az affin transzformációnak, az eltolást a pozíciók adhatják meg. Ha a skála nem ismert, a négy affin paraméterhez minden egyes ismeretlen skála egy becsülnivaló paramétert ad hozzá. A lineáris egyenletek ebben az esetben homogének az affin transzformációra nézve. Fontos feltétel, hogy legalább egy skálát ismerni kell, tehát egy inhomogén egyenletre szükség van a teljes becsléshez.

5. III. téziscsoport: Kötegelt behangolás újszerű alkalmazása nem hagyományos problémaosztályokra

Ez a fejezet egy újabb területtel ismerteti meg az Olvasót, amelynek a célja a háromdimenziós rekonstrukcióban és kamera autókalibrációban alapvetőnek számító kötegelt behangolás (angolul: Budle Adjustment, BA) módszer felhasználása két olyan célra, amellyel eddig ismereteim szerint nem foglalkozott senki sem.

5.1. Bevezetés

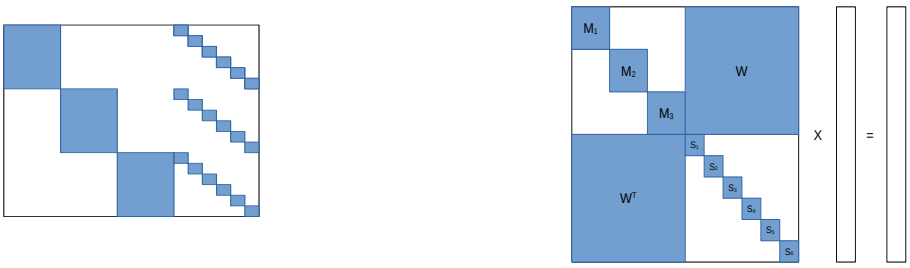
A kötegelt behangolás olyan problémaosztályok esetén alkalmazható, amikor nagy mennyiségű adat viszonylag kevés paramétertől függ, és az adatok kétdimenziós mátrixba szervezhetőek. A 29. ábra mutat egy ilyen példát.

Eredeti verzióját sokképes rekonstrukcióra fejlesztették ki [167]. Amennyiben van N darab kameránk és K darab pontunk, minden egyes kamerához minimum hat, maximum tizenegy paramétert kell meghatároznunk. Vegyük az első esetet, ami kalibrált kamerák esetén lehetséges, azaz amikor a belső paraméterek ismertek, és csak a külső paramétereket kell becsülni: három forgatási és három eltolási paraméter számítandó ebben az esetben. Az M darab térbeli pont értelemszerűen három szabad paraméterrel, a három térbeli koordinátával (X, Y és Z) rendelkezik. Összesen tehát $6N + 3M$ paraméter becslését jelenti a rekonstrukciós folyamat.

Ha minden pont minden képen látszik – ami csak elméletileg lehetséges, a gyakorlati esetekben takarások is előfordulnak –, összesen $2N \cdot M$ koordinátaértéket tudunk leolvasni. A kettes szorzó onnan jön, hogy a képeken vízszintes és függőleges koordinátákat egyaránt le tudunk olvasni.

Ha megszámoljuk az ismert koordinátákat és a becsült paramétereket, a feladat megoldásához az szükséges, hogy ismert adatokból legalább annyi legyen, mint számítandó értékekből. Nagy kép- és pontszám esetén $2N \cdot M >> (6N + 3M)$, azaz ekkor a feladat jelentősen túlhatározott.

Ha felírunk valamilyen hibafüggvényt a rekonstrukcióra, az i-edig kép j-edik pontja csak az i-edik kamera paramétereitől és a j-edik pont térbeli koordinátáitól függ. Az összes többi kamera és pont nem befolyásolja a vetítési pont koordinátáit. Ezért, ha a hibafüggvényt az ismeretlen paraméterek szerint deriváltjuk, a laza függőség miatt sok zérus értéket kapunk. Más szóval, a Jacobi-mátrix elég ritka lesz, ahogyan azt a 29. ábra bal oldali képén látjuk három kamera és hat pont rekonstrukciója esetén. Numerikus algoritmusok alkalmazása esetén a linearizált normálegyenlet is ritka lesz, ugyanannak az ábrának a jobb oldali képe ezt a tényt mutatja meg. A normálegyenletet invertálni kell a megoldáshoz. Általános esetben az invertálandó mátrix mérete a paraméterek számával egyenlő. A normálegyenlet speciális formáján jól látható, hogy a főátló mentén kisebb blokkok találhatóak. A teljes mátrix inverziója a kisebb blokkok inverziójának segítségével megadható, ahogyan azt Laurakis és Argyros [102] munkája nagyon szemléletesem elmagyarázza, a cikk elolvasát jó szívvel ajánlom az érdeklődőknek.



(a) *A Jacobi-mátrix felépítése három kamerás és hat pontos rekonstrukció esetében.* **(b)** *A bal oldali feladathoz tartozó normálegyenlet.*

29. ábra. *Numerikus optimalizálás kötegelt behangolás segítségével. A nem nulla elemek világoskékek az ábrán. Észrevehető, hogy a normálegyenletben a főátló mentén kisebb blokkok találhatóak, ezek invertálása gyorsan és numerikus szempontból pontosan elvégezhető.*

Ahogyan az ebből a dolgozatból is kiderül, kameraképek alapján lehetséges egy tárgy háromdimenziós modelljét megalkotni.

A képalapú rekonstrukciós módszerek közös jellemzője az, hogy gyakran nem képesek rekonstruálni a felszín nagyon finom részleteit, melyeket nevezhetünk érdességnek. A textúra alkalmazása a modellre elfedheti az érdesség hiányát, hogy bizonyos mértékig elfedheti a durvaságot, de változó megvilágítás esetén a részletek hiánya nyilvánvalóvá válik. A számítógépes grafika [130] ezt a problémát úgy oldja meg, hogy normálvektorokat rendelnek a felülethez, így megváltoztatva a felület eredeti irányultságát. Mivel a renderelés során a normálvektorokat az árnyalási összefüggésekben is használják, a normálvektorok alkalmazása lehetővé teszi, hogy a modell részletesebbnek tűnjön, mint amilyen valójában.

Léteznek azonban más rekonstrukciós technikák is:

– Az úgynevezett vizuális burok algoritmus [92] egy durva modellt hoz létre egy képsorozatból. A képsorozatokon meghatározzák a tárgy kontúrjait, és ebből készítenek egy térfogati (volumetrikus) modellt az úgynevezett térfaragás (angolul: Space Carving) eljárással [89]. Ezek a rekonstruált modellek sajnos elég durvák. Pontosabb módszerek, mint a lézeres vagy strukturált fényszkennelés [144] is alkalmazhatóak, feltéve, hogy a megvilágító eszközök rendelkezésre állnak és nagyon pontosan be vannak kalibrálva. Megjegyzendő, hogy a textúra rekonstrukció csak az árnyékokból is lehetséges, ahogyan azt Yu és munkatársai [185] bemutatják.

– Fotometrikus sztereó esetén laborkörülményeket alkalmazunk. Ideális esetben a teljesen sötét szobában semmi sem látszik, ha nem helyezünk el mi magunk fényforrásokat. Ha bekapcsolunk egy megvilágítást, a Lambert-törvény alapján a fényforrás és a felületi (merőleges) irány meghatározza a tárgy színét a megvizsgált helyeket. Fotometrikus sztereó esetén a feladat az intenzitásokból kiszámolni a térbeli modellhez tartozó merőleges irányokat (normálvektorokat), továbbá a fényforrások pozícióját és a megvilágítás erejét.

Ha egy objektumról rögzített nézőpontból, de különböző fényviszonyok között készítünk képeket, akkor a megfigyelt intenzitásértékek felhasználhatók a normális irányok kiszámítására. Ezt a technikát fotometrikus sztereónak nevezik, és három évtizeddel ezelőtt Woodham [179, 180] publikálta az eljárás alapelveit. Azóta számos tanulmány jelent meg ezen a területen, amelyek megoldásokat kínálnak a problémára. A domináns szakirodalmi megoldások [179, 73, 75, 40, 120, 67] feltételezik, hogy párhuzamos fényforrást alkalmaznak a megvilágításhoz.

A fotometriai sztereó-technika legújabb tanulmányai általában az árnyékok és a fénypontok észlelésére összpontosítanak [68, 7]. Egy másik nagyon érdekes megközelítés az úgynevezett színes fotometrikus sztereó [32, 20, 71, 68]: ha a jelenetet legalább három különböző színű fényforrás világítja meg, akkor a rekonstrukciót egyetlen kép segítségével lehet elvégezni. Ennek eredményeképpen nem-merev (dinamikus) objektum [66, 76] rekonstrukciója is lehetséges.

A fotometriai sztereó elsődleges alkalmazási területe a 3D rekonstrukció [180], olyan látványos területekkel, mint például a ruhamodellezés [66]. Érdekesség, hogy még orvosi képalkotásra [93] is lehet alkalmazni.

Ebben a tézisben feltételezzük, hogy a térbeli felület kezdeti becslése adott, és a módszerünk célja a kezdeti geometria finomítása a normálértékek kiszámításával. Ez a megközelítés hasonló a Nehab és mtsai. [120] illetve a Jankó Zsolt-féle módszerekhez [75], de mi nem feltételezünk párhuzamos fényforrásokat, míg a klasszikus fotometrikus sztereó algoritmusok ezt teszik. A párhuzamos fényforrásokkal történő megvilágítás nagyon komoly hátránya, hogy a metrikus rekonstrukció nem érhető el többértelműség [22] miatt. Ez a többértelműség akkor kerülhető el, ha különböző megkötéseket és feltételezéseket alkalmazunk, mint pl. integrálható normálisok [186, 68], vagy kalibrált fényforrások [189].

A munkánkban a Bundle Adjustment (BA) algoritmust [167] alkalmazzuk fotometrikus sztereóra. A BA algoritmus a jól ismert Levenberg-féle numerikus optimalizálás alkalmazása azokra a problémákra, ahol a Jakobi-mátrix speciális, meglehetősen ritka. A módszert általában az ún. 'Structure from Motion' problémák megoldására [63] használják, erre fejlesztették ki az ezredforduló környékén. *Ebben a disszertációban igazolni fogjuk, hogy a kötegelt behangolás képes a fotometrikus sztereó problémáját megoldani.* Megjegyezzük, hogy ismert egy másik technika is, nevezetesen a legkisebb négyzetek váltakozása, amely mind a fotometrikus sztereó [77] és a Structure from Motion [27, 91] esetében alkalmazható, mivel mindkét probléma felírható a következő módon faktorizálással írható át, ha párhuzamos fényforrásokat alkalmazunk. A mi esetünkben a fő újdonság, hogy az algoritmusunk segítségével a pontszerű fényforrásokat is kezelni tudjuk.

Miután a BA numerikus algoritmus, kezdeti értékek szükségesek a numerikus eljárás elindításához. Ezeket a kezdeti paraméterértékek mi Jacobs módszerével [73] kapjuk meg. Ezután egy tetszőleges fényforrásmodell esetén a finomítja a felületi normálisokat a fényforrás paramétereivel együtt.

Olyan tesztmintákkal foglalkozunk, amelyekben a felvett képek minden képpontja érvényes, azaz a rekonstruálandó objektum egyik része sincsen (ön)árnyékolás alatt. Ez a megállapítás nem teljesen igaz a valós tesztekre. A javasolt BA algoritmus azonban képes kezelni az árnyékokat és a fénypontokat, ha a megfelelő pixelek fel vannak címkézve, így ezeket a paraméterek inicializálása során kell felismerni. A Hernandez és mtsai. által kidolgozott algoritmust [67] használtuk az érvénytelen pixeleket felismerésére. Az általunk használt BA implementáció [102] képes az ér-

vénytelen értékeket adó pixelek kizárására is.

Nem a mi munkánk az első, amelyben a nem párhuzamos fénysugarakat feltételezzük [178], mi azonban nem kvadratikus optimalizálást használunk a a távoli fényforrások által megvilágított objektum rekonstruálására. Ismereteim szerint a BA ilyetén használata újszerű ötlet. A pontszerű fényforrás alkalmazása igen komoly előnnyel jár az irányítottal szemben: az utóbbinál létezik egy többértelműség [22]. Ha nincs a-priori információ a felszínről vagy a fényforrásokról, a normálvektorok csak egy ismeretlen affin transzformáció erejéik számíthatóak ki. Pont fényforrás esetén azonban lehetséges az egyértelmű metrikus rekonstrukció is.

5.2. A kötegelt behangolás alkalmazása fotometrikus sztereóra

Ebben a szakaszban bemutatom a fotometrikus sztereó alapelveit, majd a saját hozzájárulásunkat ismertetem, melynek az eredménye egy újszerű algoritmus.

5.2.1. Fotometrikus sztereó.

Megközelítésünkhöz a következőket feltételezzük:

1. Egy kezdeti modellt adunk meg bemenetként. A cél az, hogy ezt a modellt finomítsuk a normálisok kiszámításával.

2. A bemeneti képeket kalibrált kamerával készítjük ugyanarról a helyről (nézőpontból).

3. Minden képen a tárgyat egyetlen fényforrás világítja meg. A fényforrás pozíciója képenként változik.

A fotometrikus sztereó alapproblémájában két ismeretlen paraméterkészlet van. Minden egyes kép esetében a fényforrás paramétereit a háromdimenziós $\mathbf{a}_i$ vektor írja le. A vektor hossza adja meg az intenzitás nagyságát, a vektor iránya a beérkező fény irányát. A szintén háromdimenziós $\mathbf{b}_p$ vektor a visszatükröződést meghatározó paraméterek tartalmazza, ami alapesetben a normálvektor. Mivel a kameraparaméterek ismertek és minden kép esetében azonosak, a tárgy vetülete által elfoglalt minden egyes képpont megfelel egy látható 3D pontnak a tárgy felszínén, és ez a pixel-tárgy megfeleltetés egy az egyhez reláció. Ezért a "pont" kifejezést használjuk mind ezekre a felületi pontokra, mind pedig azok vetületeire. Az algoritmus elsődleges kimenetele a felszíni normálvektorok lesznek.

A megvilágítási egyenlet az, amely meghatározza, hogy ezek a paraméterek hogyan hozzák létre a p pont intenzitását, amelyet az i képen megfigyelhetünk:

$$I(i,p) = f(\mathbf{a}_i, \mathbf{b}_p). \tag{52}$$

Az intenzitások $I(i,p)$ értékei ismertek – a képek pixeleinek intenzitásai –, az f függvényt az itt leírtak szerint határozzuk meg a fényforrásmodellel, a felületmodellel és a fényforrásmodellel összhangban.

Ennek az egyenletnek a szerkezete lehetővé teszi, hogy a BA algoritmus [167] segítségével egyidejűleg finomítsuk a paramétereket oly módon, hogy az egyenlet két oldala közötti hiba minimális legyen.

A kötegelt behangolásnak szüksége van a paraméterek kezdeti értékeire; ezeket úgy kaptuk meg, hogy feltételeztük, hogy az objektum felülete lamberti[29], a fényforrások pedig irányítottak, mivel ebben az esetben a paraméterek analitikusan kiszámíthatók Jacobs-módszerének segítségével[73].

A megvilágítási egyenletnek megfelelően egy képpont intenzitása csak az adott i-edik képhez tartozó $\mathbf{a}_i$ irányvektortól, és a p-edik felület $\mathbf{b}_p$ felületi paramétereitől függ az intenzitás. Ez teljesen hasonló a 'Structure from Motion' feladathoz, amikor az f-edik képkockán az i-edik jellegzetes pont csak az f-edik kamera külső és belső paramétereitől, és a rekonstruált pontfelhő p-edik pontjának térbeli koordinátáitól függ. Így a BA algoritmus kicsi változtatásokkal felhasználható a fotometrikus sztereó paramétereinek együttes optimalizálására.

A BA algoritmus a Levenberg-minimalizáláson alapul, amelyet mi az alábbiakban röviden ismertetjük.

A fotometrikus BA célja a teljes intenzitáshiba minimalizálása. Az aktuális intenzitást az optimalizálandó paraméterek tekintetében az 52. egyenlet segítségével számítjuk ki minden egyes pixelre, a hibaértékek a számított és mért intenzitásai közötti különbségből származik. Maga a teljes intenzitáshiba minden kép minden egyes pixelének legkisebb négyzetes hibájával egyezik meg.

Megjegyzendő, hogy a hibafüggvénybe további hibatételek is beilleszthetők: a fény és a felület paraméterei felhasználhatók ezekben a hibatételekben. Fontos, hogy ezeket a paramétereket egymástól függetlenül kell használni. Például a kezdeti és a kapott normálértékek közötti különbség büntethető.

5.2.2. Levenberg-Marquardt-féle hibacsökkentés

Adott egy $\mathbf{f}(\mathbf{x})$ vektor-vektor függvény és egy $\mathbf{y}$ célvektor, a feladat, hogy megtaláljuk azt a $\mathbf{x}$ vektort, amely minimalizálja az $||\mathbf{y} - \mathbf{f}(\mathbf{x})||$ hibát.

A Levenberg-féle optimalizálás [96, 108] a Gauss-Newton és a gradiens módszerek keveréke. Egy iteratív algoritmus, amely minden egyes lépésben frissíti a paramétereket a következő képlet szerint:

$$(\mathbf{J}^T\mathbf{J} + \mu\mathbf{I})\Delta\mathbf{x} = \mathbf{J}^T(\mathbf{y} - \mathbf{f}(\mathbf{x})), \tag{53}$$

ahol $\mathbf{J}$ a Jakobi-mátrix, μ egy skalár, amely a súlyozást végzi a két komponens között.

5.2.3. A Levenberg-Marquardt algoritmus alkalmazása fotometrikus sztereóra

Fotometrikus sztereó esetén az 52. megvilágítási egyenletből származó $\mathbf{f}(\mathbf{a}_i, \mathbf{b}_p)$ formula adja a hiba alapját. A nemlineáris optimalizációs algoritmusok alkalmazásának nehézsége abban rejlik, hogy a $\mathbf{J}^T\mathbf{J} + \mu\mathbf{I}$ egy olyan invertálandó mátrix, amelynek dimenziói lineárisan nőnek a pixelek számával. A mi esetünkben, feltételezve $100\,000$ pontot (3 paraméter minden egyes ponthoz) és 10 képet (4 paraméterek minden egyes pont fényforráshoz), a paraméterek száma $3 \cdot 100\,000 + 10 \cdot 4 = 300\,040$. Ez adja a mátrix méretét, amelyet minden egyes iterációban invertálni kell.

Ezt a problémát a kötegelt behangolás használatával lehet enyhíteni. Azáltal, hogy szétválasztjuk a pont és a fényforrás paramétereit, az 53. normálegyenlet a következőképpen írható le:

[29] A lamberti felülteket a köznyelv mattnak hívja. Azt a felületet nevezzük lambertinek, amelyik a beérkező fény energiáját minden irányban ugyanolyan valószínűséggel veri vissza.

$$\begin{bmatrix} \mathbf{U} & \mathbf{X} \\ \mathbf{X}^T & \mathbf{V} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{s} \\ \Delta\mathbf{m} \end{bmatrix} = \mathbf{J}^T \begin{bmatrix} \epsilon_\mathbf{s} \\ \epsilon_\mathbf{m} \end{bmatrix}, \tag{54}$$

ahol $\Delta\mathbf{s}$ és $\Delta\mathbf{m}$ vektorok tartalmazzák a pontok és a fényforrások paramétereit, és $\epsilon_\mathbf{s}$ és $\epsilon_\mathbf{m}$ pedig az $\mathbf{y} - \mathbf{f}(\mathbf{x})$ hiba megfelelő komponense. Az $\mathbf{U}$ mátrix a fényparaméterek parciális deriváltakból áll, míg $\mathbf{V}$ tartalmazza a normálvektorokhoz tartozó deriváltakat. A $\mathbf{X}$ mátrixban ezek a parciális deriváltak egymással szorozva vannak.

Ha a fenti egyenletet balról megszorozzuk a

$$\begin{bmatrix} \mathbf{I} & -\mathbf{X}\mathbf{V}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

mátrixszal, akkor az első blokkból a második együtthatómátrixot ki tudjuk nullázni. Ekkor az egyenletrendszerből a paraméterek pontosítását a következőképpen fejezhetjük ki:

$$\Delta\mathbf{s} = (\mathbf{U} - \mathbf{X}\mathbf{V}^{-1}\mathbf{X}^T)^{-1}(\mathbf{J}^T\epsilon_\mathbf{s} - \mathbf{X}\mathbf{V}^{-1}\mathbf{J}^T\epsilon_\mathbf{m}), \tag{55}$$

$$\Delta\mathbf{m} = \mathbf{V}^{-1}(\mathbf{J}^T\epsilon_\mathbf{m} - \mathbf{X}^T\Delta\mathbf{s}). \tag{56}$$

Az utóbbi formulák előnye, hogy a $\mathbf{V}$ és az $(\mathbf{U} - \mathbf{X}\mathbf{V}^{-1}\mathbf{X}^T)$ mátrixokat kell invertálni, és ez gyorsabban elvégezhető az átalakítások után. A $\mathbf{U}$ mátrix mérete megegyezik a fényforrás paramétereinek teljes számával, és a paraméterek közötti függetlenség miatt $\mathbf{V}$ egy blokk-diagonális mátrix. Így példánkban a minimalizálás minden egyes iterációban $N = 100\,000$ darabszámú, egyenként 3×3 méretű kis blokkok invertálását végzi el a $\mathbf{V}$ mátrixot tekintve. Ezen kívül szükséges még egyetlen $10 \cdot 4 = 40$ méretű mátrix invertálása, ami a korszerű numerikus algoritmusok segítségével pontosan és meglehetősen gyorsan elvégezhető.

5.2.4. Alkalmazás pontszerű fényforrásra

A kötegelt behangolásnak megvan az a hatalmas előnye, hogy tetszőleges világítási egyenletekhez, azaz különböző felület- és fényforrásmodellekhez könnyen adaptálható a módszer.

Példaként itt van a megvilágítási egyenlet Lambert-féle felületre és pontra vonatkozóan fényforrás esetén:

$$I(i,p) = k\frac{\lambda}{\lambda + |\mathbf{l}_i - \mathbf{x}|^2}\frac{\mathbf{l}_i - \mathbf{x}}{|\mathbf{l}_i - \mathbf{x}|} \cdot \mathbf{b}_p, \tag{57}$$

ahol a fényforrás paraméterei a k fénysűrűség, a fényforrás $\mathbf{l}_i$ háromdimenziós pozíciója (együttesen adják a fényparaméter vektorát $\mathbf{a}_i = [l_{i_1}\ l_{i_2}\ l_{i_2}\ l_{i_3}\ k]$), és egy λ skalár , amely a fény csillapításával függ össze[30].

Egy pont leírható a háromdimenziós pozíciójával (jele: $\mathbf{x}$ vektor) és a $\mathbf{b}_p$ normálvektorával. A normálvektor nem egységvektor, mert a diffúz fénytöréssel előskáláztunk, azaz a skála azt határozza meg, hogy a fény hányadrészét sugározza vissza az adott felületi pont. Megjegyzendő azonban, hogy a pozíciók nem hangolható paraméterek, hanem állandóak, előre adottak, azaz a kezdeti modell által meghatározottak.

[30] Ez a modell összhangban van a pontszerű fény grafikus rendszerek fényforrásaihoz, mint például Blender vagy POV-Ray (www.blender.org, www.povray.org).

A λ paraméter értéket rögzítettnek tekintettük, így marad fényforrásonként négy, pixelenként pedig három beállítandó szabad paraméter.

A $\mathbf{U}$, $\mathbf{V}$ és $\mathbf{X}$ mátrixok kiszámítása az 57. összefüggés alapján összetett képleteket eredményez. Az explicit jelölés helyett a mátrixok szimbolikus számításához szükséges *MATLAB* kódot a L függelékben mutatjuk be.

5.2.5. A paraméterek inicializálása

Mivel a BA algoritmus numerikusan finomít, az inicializálásához meg kell adnunk egy durva becslést a fényforrások ($\mathbf{a}_i$) és a felületi pontok ($\mathbf{b}_p$) kezdeti értékeire. Lamberti visszaverődést és irányított fényforrásokat feltételezve egy egyszerű RANSAC [46] alapú robusztus módszert alkalmazunk a paraméterek kezdeti értékének meghatározására, hasonlóan Hernandezék [67] és Jankó Zsolt módszereihez [75]. Az i-edik fényforrás irányát $\mathbf{d}_i$-vel jelölöm, ez a $\mathbf{a}_i$ paraméterezés speciális esete irányított fényforrásra.

Lambert koszinusztörvénye miatt felírhatjuk a világítási egyenletet: $\mathbf{g}(\mathbf{d}_i, \mathbf{b}_p) = \\ = \mathbf{d}_i^T \mathbf{b}_p$. Három megfelelő pontot kiválasztva és az 52. egyenletet felhasználva a fény irányát a következőképpen lehet kiszámítani:

$$\mathbf{d}_i = [\mathbf{b}_{p_1} \mathbf{b}_{p_2} \mathbf{b}_{p_3}]^{-T} \begin{bmatrix} I(i,p_1) \\ I(i,p_2) \\ I(i,p_3) \end{bmatrix}. \tag{58}$$

Ezzel tehát, irány-fényforrást feltételezve, kezdeti értéket tudunk a fény irányára adni.

Itt a három megfelelő pont olyan pozíciókat jelent, ahol a kezdeti $\mathbf{b}_p$ vektorok közel vannak a valós pozíciókhoz . A fentiek szerint feltételezzük, hogy van egy durva kezdeti háromszögháló [31], és azt is feltételezzük, hogy ennek a hálónak van egy olyan részhalmaza, ahol a pont térbeli koordinátái közel vannak a valós paraméterekhez. Vagyis, ha ebből a részhalmazból három tetszőleges $\mathbf{b}_p$-t választunk, az 58. egyenlet jó közelítést fog adni a fényforrás irányára. Miután ez az érték a numerikus finomításnak csupán a kezdeti értéke lesz, kisebb hibák nagy valószínűséggel nem befolyásolják az optimalizáció végeredményét.

Adott $\mathbf{d}_i$ és $\mathbf{b}_p$ vektorok esetén az $\mathbf{a}_i = [l_{i_1}\ l_{i_2}\ l_{i_3}\ k]$ vektor becslését úgy kaphatjuk meg, hogy a $\mathbf{l}_i$ fénypozíciós vektort a $\mathbf{d}_i$ irányába mozgatjuk, és a fénysugárzás k paraméterét úgy állítjuk be, hogy az a lehető legjobban illeszkedjen az $I(i,p)$ megfigyelt pixelintenzitásokhoz, azaz az 57. összefüggés minél kisebb hibát adjon.

Egy RANSAC-alapú [46] módszert használunk arra, hogy ilyen ponthármasokat kapjunk. A módszer azon az elváráson alapul, hogy rossz választás esetén sok kiugró érték lesz.

A módszer a következőképpen működik:

1. Véletlenszerűen kiválasztunk a háromszöghálóból három pontot.
2. Az 58. egyenlet alapján kiszámítjuk az irányított fényforrás modelljét.

[31] Angolul a háromszöghálót mesh-nek hívják, ami háromszögelt pontfelhőt jelent, ahogyan azt a számítógépes grafikában [90] alkalmazzák

3. Megszámoljuk, hány pont támogatja az előző lépésben kapott fényirányt. Azt mondjuk, hogy egy p pont akkor támogatja a modellt, ha

$$\forall i: \quad \left|I(i,p) - \mathbf{d}_i^T \mathbf{b}_p\right| < \epsilon, \tag{59}$$

   ahol ϵ egy kicsi, empirikusan beállított küszöbérték.

4. Az 1–3 lépéseket ismételjük.

5. Azt a fényirányt választjuk, amelyet a legtöbb pont támogat.

Ez a módszer alkalmas arra, hogy durva becslést adjon a fényforrás irányára. A fényforrások kezdeti pozícióit is meg kell határozni. Erre a következők módszert javasoljuk: A tárgy középpontjából (súlypontjából) elindulunk, és egy viszonylag távoli pontot választunk kezdeti középpontnak, hiszen minél messszebb van egy pontfényforrás, annál párhuzamosabbak a fénysugarai. Mi úgy választottuk meg ezt a pozíciót, hogy a tárgypontok szórását kiszámítottuk a háromszöghálóból a középponthoz viszonyítva, és a fényforrás távolsága ennek a szórásnak a kétszerese.

Ha a fényforrások helyei és a felszíni pontok háromdimenziós koordinátái ismertek, a lámpák negyedik paramétere, a sugárzása, triviálisan becsülhető az 57. egyenletből, hiszen a feladat lineáris a $\mathbf{b}_p$ normálvektorra nézve.

5.2.6. Teszteredmények

A javasolt módszert szintetikus (renderelt) és valós képen is teszteltük. A fényforrásokat mind a szintetikus, mind a valós tesztekben véletlenszerűen helyeztük el a tárgy körül. A fények minden egyes pozícióból az egész tárgyat megvilágítják.

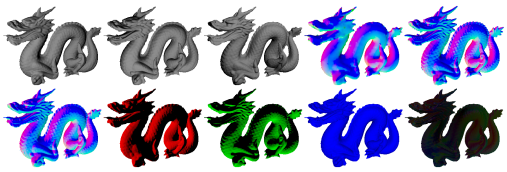
Szintetikus tesztek.

Szintetikus tesztek alatt itt számítógépes grafika által elkészített, azaz mesterségesen renderelt képek feldolgozását értjük.

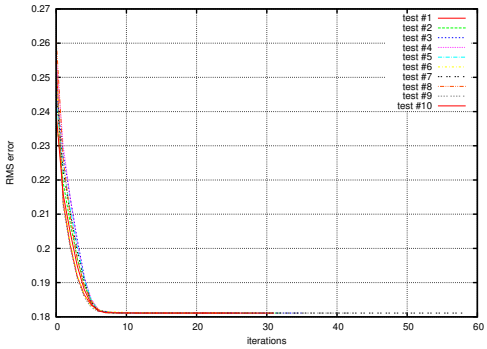
'Dragon' képsorozat. Az első (és egyben a harmadik) tesztsorozat a Stanford 3D Scanning Repositoryból[32] származik. A letöltött háromszögelt 3D hálót importáltuk Blender-be, és több képet rendereltünk mozgó pontfényforrással. A kiszámított szintetikus képeket a 30. ábra felső sorában láthatjuk.

Részletesen megvizsgáltuk a "Dragon" sorozatot. A letöltött háló nagyon részletes, ezért a kiszámított normálvektorok pontosak lesznek, ezeket használjuk alapigazságnak. A normáltérképet a javasolt módszerrel rekonstruáltuk. A rekonstruált normáltérképet és annak három komponensét a 30. ábrán lehet megtekinteni. Összehasonlítottuk az eredeti (bemeneti) képeket a szimulált képekkel. A pixelintenzitások RMS (Root Mean Square) hibája $0{,}18$ volt. Ezt a tesztet többször megismételtük, és mindig ugyanazt az értéket kaptuk. Meg kell jegyeznünk, hogy az inicializáláshoz használt RANSAC [46] robusztus becslés nem determinisztikus, ezért előfordulhatna, hogy nem ugyanoda konvergál a módszer, de ilyet nem tapasztaltunk. A hiba konvergenciáját tíz különböző futtatásra a 31. ábra mutatja be.

A rekonstrukció és az alapigazság normálvektorait is összehasonlítottuk, ahogyan az a jobb oldalon látható a 30. ábra alsó sorában. A hiba a vektorok közötti euklideszi különbség a térben. A normálvektorok hibájának RMS értéke 0.05 volt. Ez a hiba a paraméterek inicializálása után még 0.13 volt. Mindezek azt mutatják, hogy *a javasolt fotometrikus BA algoritmus jelentősen javítja a rekonstrukció minőségét*.



30. ábra. *'Dragon' képsorozat. Felül: Három példakép a kilenc renderelt képből, a kezdeti normálkép, utolsóként pedig az alapigazság normálkép. Alul: A rekonstruált normáltérkép, annak vörös, zöld és kék csatornája, végül pedig a normálhiba.*



31. ábra. *Az intenzitáshiba csökkenése az iterációk számának függvényében tíz különböző tesztesetre.*

'Buddha' képsorozat. A harmadik szintetizált példa a "Buddha" szekvenciával dolgozik. A bemeneti képek és a rekonstruált normáltérkép a 32. ábrán látható. A BA a normál térkép hibáját 0.09-ről 0.06-ra javítja az iterációk során.

Kvantitatív összehasonlítás.

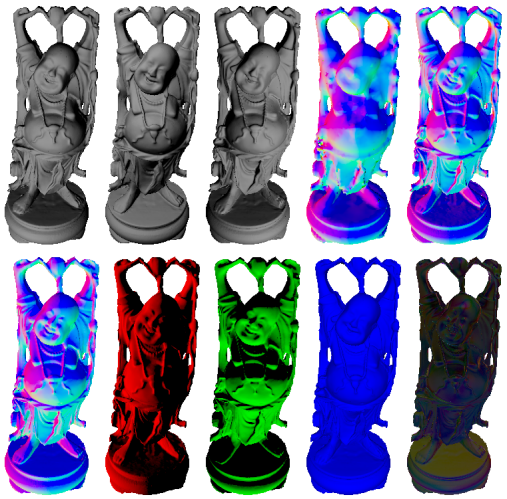
Módszerünket Hernandez és mtsai. [67] módszerével hasonlítottuk össze szintetikus adatokon. Az összes szintetikus tesztsorozat Blenderrel generáltuk. Ezután a képeket mind a javasolt, mind a rivális módszerrel feldolgoztuk. (A rivális módszert mi is implementáltuk, mivel annak nem találtuk meg a közzétett változatát.)

Az alapigazság normálvektorokat a Blenderből kinyertük, és összehasonlítottuk a rekonstruált vektorokkal. A hibavektor az alapigazság és a rekonstruált vektorok közötti különbség. A rekonstrukció minőségét a hibavektorok normájának négyzetes középértéke (RMS) határozza meg.

A kapott RMS hibák az 5. táblázatban szerepelnek. Egyértelműen kijelenthető, hogy a javasolt módszer lényegesen pontosabb, mint a a rivális [67] mindhárom szintetikus tesztesetnél.

Ezen túlmenően megvizsgáltuk az irányított és a pontszerű fényforrásokkal végzett rekonstrukciók közötti különbséget is a "Dragon" szekvencia esetében. A tesztképeket irányított fényforrással generáltuk Blenderben. Több képet rendereltünk, a képek között a fényforrások irányát változtattuk meg. Ezután a normáltérképet a paraméterek inicializálására javasolt RANSAC-alapú módszerrel rekonstruáltuk, amelyet az 5.2.5. fejezetben ismertettünk. A kapott normál térképvektorok RMS hi-

[32] http://graphics.stanford.edu/data/3Dscanrep/



32. ábra. *'Buddha' képsorozat. Felül: Három példakép a kilenc renderelt képből, a kezdeti normálkép, utolsóként pedig az alapigazság normálkép. Alul: A rekonstruált normáltérkép, annak vörös, zöld és kék csatornája, végül pedig a normálhiba.*

bája 0.16 volt, ami jelentősen nagyobb, mint a pontfényforrással kapott 0.12 hiba. *Ez az eredmény megerősít bennünket abban a hitben, hogy pontosabb normáltérkép-helyreállítás lehetséges, ha a rekonstruálandó objektumot irányított fényforrás helyett pontszerű fényforrással világítjuk meg.*

5. táblázat. *A rekonstrukció hibájának RMS értéke szintetikus teszt esetén.*

Tesztsorozat neve	Hernandez és mtsai. [67]	Javasolt eljárás
Logo	0.169	0.043
Dragon	0.129	0.049
Buddha	0.087	0.059

Valós tesztek.

'BME-logo'. Az első teszt során Blender által generált képeket használtuk. Az objektum egy kis dombornyomott fémlap, amely jó közelítésel síknak tekinthető. A normálkép felületét szerettük volna az algoritmusunkkal megkapni.

A rekonstrukció végrehajtási ideje 1 perc 43 másodperc volt [33]. A bemeneti képeket és a rekonstruált normáltérképet a 33. ábrán láthatjuk. Itt a normáltérképet – ahogyan az a grafikában általánosan szokásos – RGB képként ábrázoljuk: a normálvektor $(255 \cdot v_1, 255 \cdot v_2, 255 \cdot v_3)$ színkóddal lett rajzolva. Például egy tisztán piros pixelt a $[1 \quad 0 \quad 0]^T$ vektorral kódoljuk.

Helyhiány miatt itt további valós teszteket nem tudok közölni, de az N függelékben két további tárgy rekonstrukcióját meg lehet tekinteni.

[33] Ezek a vizsgálatok 2010-ben készültek, akkori hardvert használva.

33. ábra. *Felső sor: 8-ból 3 renderelt kép a "BME logo" szekvenciából, a kezdeti normáltérkép és az alapnormálok. Alsó sor: Rekonstruált normáltérkép, annak R, G, B komponensei és a normáltérkép hibaképe.*

5.3. Irányított pontfelhő térbeli rekonstrukciója

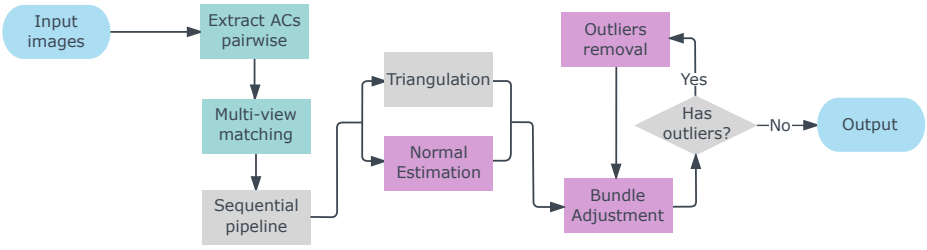
Ebben a szakaszban megmutatom egy másik alkalmazását a kötegelt behangolásnak: a módszert az affin transzformációk feldolgozására is fel lehet használni.

Ahogy azt már fentebb taglaltam, az eredeti kötegelt behangolást [167] térbeli pontfelhők rekonstrukciójára fejlesztették ki. A 4. fejezetben megmutattam, hogyan lehet az affin transzformációk segítségével felületi normálvektorokat rekonstruálni, és így – a háromszögeléssel [61] kiegészítve –, *irányított pontfelhőket* lehet kapni.

Egy affin transzformációt két kép megadott kis részlete között lehet értelmezni. A kötegelt behangolás során egy mért érték egy térbeli ponttól és egy kamera paraméterkészletétől függ. Esetünkben a mért affin transzformáció két kamera paramétereitől is függ, ahogyan azt a 22. összefüggésben korábban megismerhettük. Ha $\mathbf{A}_{ij}$ jelöli egy adott felületdarabka i-edik és j-edik képre vetített mintái között a transzformációt, akkor ezt fel lehet bontani az alábbi módon:

$$\mathbf{A}_{ij} = \mathbf{A}_j \mathbf{A}_i^{-1},$$

ahol $\mathbf{A}_k$ a térbeli felületdarabka és a k-adik kép közötti transzformáció. Ekkor az így felbontott $\mathbf{A}_k$ affin transzformáció már csak a térbeli transzformációtól függ. Így a kötegelt behangolás alapelve alkalmazható.



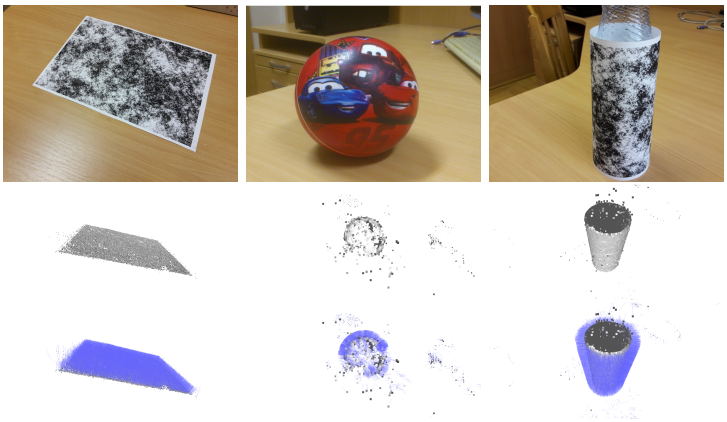
34. ábra. *Rekonstrukciós csővezeték kötegelt behangolás alkalmazásával.*

Az általunk javasolt rekonstrukciós csővezeték a 34. ábrán látható. A csővezeték bal oldalán látható a sokképes alaprekonstrukció, amely sok sztereó rekonstrukciót tartalmaz. Ezek a korábban ismertetett módon készítenek irányított pontfelhőt, háromszögelésen (triangulation) és felületi normálvektorbecslésen keresztül.

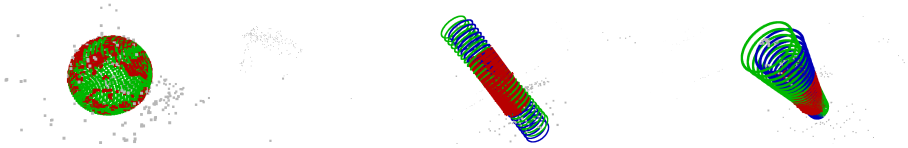
A már felbontott affin transzformációkkal és a vetületi pontokkal a javasolt kötegelt behangolás (bundle adjustment) algoritmust futtattuk. A kapott eredményekre

még egy kiugró érték szűrést (outlier removal) is alkalmazunk, amelyek a durva becsléseket – például ha hátulról látszódik egy felület –, kiszűrik.

Helyhiány miatt a részleteket a disszertációban nem tudom közölni, az eredeti munkában [161] további fontos részletekkel ismerkedhet meg az Olvasó.



35. ábra. *Rekonstrukciós eredmények a felületi pontok és normálisok mennyiségi összehasonlításához. Fent: Egy példa a bemeneti képből. Középen: A javasolt csővezeték által visszaadott rekonstruált pontfelhő. Alul: Ugyanezek a modellek kék vonalmetszetekkel megjelenített felületi normálvektorokkal.*



36. ábra. *Rekonstruált gömb (balra) és a henger két nézete (középen és jobbra). A helyesnek címkézett rekonstruált pontokat, a kiugró értékeket és a beillesztett modelleket piros, szürke, illetve zöld szín jelöli. A henger illesztése esetén a kék szín a RANSAC [46] által kiszámított kezdeti modellt jelöli. A kiugró értékek a RANSAC minimális modelljének felelnek meg.*

5.3.1. Tesztelési eredmények

Noha az eredeti munkánkban [161] kvantitatív kiértékelések is szerepelnek, terjedelmi korlátok miatt a disszertációban csak a jóval látványosabb valós eredmények közül ismeretetek néhányat. A 35. ábrán azt láthatjuk, hogy egyszerű geometriai primitívek (sík, gömb, henger) felvételeiből hogyan lehet a felületi pontokat és a normálvektorokat rekonstruálni. A 36. ábrán az eredményül kapott irányított pontfelhőkre felületeket is illesztettünk, hogy a kapott térbeli felületi pontok minőségét is ábrázolni tudjuk.

Egy másik alkalmazási terület síkfelületek rekonstrukciója. Az ember alkotta világban sok ilyen található, itt épületfalak segítségével mutatom be a rekonstrukciós csővezeték eredményét. A 37. ábrán tekinthetőek meg épületek térbeli rekonstrukciói. A felületeket a jobb kinézet kedvéért textúrával is elláttuk.

hajder_274_24

37. ábra. *Épületrekonstrukciós példa. Balra a kiválasztott felületdarabok láthatóak, mellettük a rekonstruált pontok és normálvektorok a képre vetítve. A jobb oldali képeken két különböző nézetből láthatóak a rekonstruált térbeli felületek.*

5.4. Összefoglalás

A harmadik téziscsoport a háromdimenziós látásban jól ismert kötegelt behangolás algoritmus alkalmazásával foglalkozik. Kollégáimmal közösen megmutattuk, hogy ezt a klasszikusnak nevezhető módszert más problémaosztályokra is sikeresen lehet alkalmazni.

Első újításként a fotometrikus sztereó elnevezésű problémára ismertettem egy új módszert, ahol a cél – laborkörülmények között készült képekből – a felületi normálvektorok és a megvilágítás paramétereinek meghatározása. Az eredeti eljárás faktorizációs algoritmus segítségével működik, azonban csak irányfényforrások alkalmazása esetén alkalmazhatóak, hiszen a fényforrás és a kameraparaméterek akkor lineáris összefüggésben – két vektor skalárszorzata segítségével – határozzák meg a képeken az intenzitások nagyságát. A saját eljárásunk a kötegelt behangolással összhangban numerikus eljárással működnek, ezért nemlineáris összefüggéseket is képes kezelni a fény- és a felületi paraméterek között.

Tudomásom szerint szintén szakirodalmi újításnak számít a fejezet második részében taglalt eljárás, amely az eredeti kötegelt behangoláshoz hasonlóan háromdimenziós rekonstrukciókkal foglalkozik, azonban eredményképpen felületi normálvektorokkal kiterjesztett pontfelhőt, azaz irányított pontfelhőt eredményez. Az ismert (mért) értékekhez a jellegzetes pontok pozícióin túl a képek közötti lokális affin transzformációkat is felhasználja a kifejlesztett új eljárás.

A két ismertetett újszerű módszer alapján az alábbi két tézist fogalmazom meg:

III.1. tézis. Kötegelt behangolás alkalmazása fotometrikus sztereóra [48].

Fodor Bálint doktorandusz hallgatómmal és Jankó Zsolt SZTAKI-s kollégámmal közösen sikerült egy újszerű eljárást készíteni kötegelt behangolás alkalmazásával, amely a fotometrikus sztereó problémát oldja meg pontszerű fényforrás esetén. Tetszőleges számú fényforrást alkalmazni tudunk, a bemenet egy képsorozat, ahol a kamera helyét a rekonstruálandó tárgyhoz képest nem változtatjuk meg. Minden megvilágítás egy képet eredményez a sorozatban. A módszer laboratóriumi körülményeket igényel abból a szempontból, hogy külső megvilágításból nem szűrődhet be fény. Minden felvételnél egyetlen megvilágítás engedélyezett.

A numerikus optimalizációnál a kezdeti paraméterek meghatározásához párhuzamos fényforrás alkalmazását javasoljuk. A rekonstrukció felírható paraméter-minimalizálásként,

ahol az optimalizálandó paraméterek a fények és a felületi normálvektorok. A kötegelt behangolás alkalmazásával tetszőleges fényforrás alkalmazható, mi pontszerű fényforrás feltételezésével mutattuk meg a módszer létjogosultságát.

III.2. tézis.Kötegelt behangolás alkalmazása irányított pontok rekonstrukciójára, affin transzformációk alkalmazásával [161].

Eichhardt Iván kollégámmal, kutatócsoportunk tagjával közösen megmutattuk 2017-ben, hogy a hagyományos kötegelt behangolás algoritmus kiegészíthető az affin transzformációkkal, ezért Structure from Motion (SfM) eljárások esetén pontos rekonstrukciós eredmény kapható numerikus optimalizálással annak ellenére, hogy a finomítandó paraméterek száma igen nagy. A hagyományos pontalapú rekonstrukcióhoz képest minden térbeli pont kiegészíthető a felületi normálvektorral is. A normálvektornak a nagysága nem tartalmaz érdemi információt, csak az iránya, ezért a térbeli normálvektor két további paramétert jelent, azaz az irányított pont összesen öt paraméterrel írható le. Így a becslés kiegészül pontonként további ismeretlenekkel, ugyanakkor az egyes képmintákhoz tartozó affin transzformációkat a képekből meg lehet határozni, azaz a képeken a 2D koordinátákon túl az affin transzformációk is megjelennek ismert adatként.

6. IV. téziscsoport: Új módszerek LiDAR mérőeszközök és perspektív kamerák kalibrációjára

Az utolsó kutatási terület, melyet jelenleg is nagy intenzitással művelek/művelünk, különböző modalitású szenzorok közös kalibrációja. Az elmúlt években LiDAR letapogató eszközök és digitális kamerák összekalibrálására koncentráltunk, tudományos eredményeket is elértünk, sőt, a hivatkozásaim komoly része is erről a területről származik.

6.1. Szakirodalmi áttekintés

Napjainkban egyre fontosabbá válik a környezetünkből minél több és pontosabb információ kinyerése. Ez különösen az autonóm autók és robotok esetében jelentős feladat, amelyeknek emberi beavatkozás nélkül kell irányítaniuk magukat. A 3D gépi érzékelés többféleképpen is megvalósítható, például kamerák, mikrofonok, radarok, szkennerek segítségével, hogy csak néhány lehetőséget említsek.

Mostanában az egyik igen kedvelt technológia a LiDAR, amely a fénysugarak által mért távolságok alapján egy ritkás pontfelhő kinyerésével képes felmérni a környezetünket. A 3D LiDAR technológia fő előnye az aktív megvilágítás, amely a környezeti féntől függetlenül működik. Tetszőleges fényviszonyok között használható, és az érzékelők nagy távolságból is képesek pontosan letapogatni a háromdimenziós világot. Korlátja a technológiának a felbontás, még a dolgozat megírásakor legkorszerűbbnek számító eszköz is maximum 256 sugárral rendelkezik, és ez a felbontás már 100 méteres távolság esetén is nagyon kevés pontot eredményez egy ott lévő objektum esetében. A LiDAR mérőeszközök meglehetősen drágák, egyetemünkön mi 16 sugaras Velodyne VLP-16 eszközöket használunk, már ezeknek az ára is millió forint felett van.

Ugyanakkor sok előnye is van ennek a mérési technológiának. Az RGB-kamerák nagy felbontású, színes képeket készítenek, és a fényviszonyokkal is meg kell birkózniuk. Különösen éjszaka jelentkeznek problémák, de nappali fényben a takarás és az árnyék is gondot okozhat a képfeldolgozásban. LiDAR alkalmazása esetén a fényviszonyok nem okoznak akkora gondot, bár az esős időben a fény tükröződése a nedves felületek mérési hatékonyságát jelentősen befolyásolja.

Szerencsére a LiDAR-érzékelők legtöbb hátránya a kamerákkal kompenzálható, és fordítva, így a 3D LiDAR-okat és a kamerákat gyakran együtt használják az objektumok észlelésére [30, 132, 133], a jelenetek háromdimenziós rekonstrukciójára [111, 155, 187] vagy éppen navigációs feladatok megoldására [112].

A 3D-s látás segítségével a pontfelhőket sűrűbbé lehet tenni, azonban két különböző modalitással nyert különálló pontfelhő nagyon pontos regisztrációja komoly kihívást jelent. A szenzorok együttműködéséhez szükség van a külső kalibrációra, ami azt jelenti, hogy a relatív helyzetüket (eltolás és orientáció/forgatás) ismerni kell. Létezik néhány illesztő algoritmus, de ezek pontossága nem kielégítő.

Robotokra és autókra rögzítés esetében az eszközök helye és tájolása nem változik az idő múlásával. Ezért elég egyszer meghatározni az elhelyezkedésüket, azaz elég egyszer bekalibrálni az eszközöket.

A kamera-LiDAR érzékelőpár kalibrálására számos módszert javasoltak. A korai munkák a 2D LiDAR eszközökre koncentrálnak, ahogyan az például Zhang és Pless

dolgozatában [188] elolvasható. Azóta a 3D LiDAR-ok kalibrálására is sok energiát fordítottak a kutatók. A módszerek listája négy csoportra osztható:

1. Jónéhány módszer [53, 126] síkbeli sakktáblákat használ.
2. Más algoritmusok [129, 172] egyéb síkbeli felületek különböző formáit alkalmazzák.
3. Az ipar gyakran használ gömb alakú céltárgyakat a különböző modalitású érzékelők kalibrálásához [143, 83]. Ezeket akár optikai kutatásokhoz [181, 100] is hasznosítani lehet.
4. Létrehoztak olyan eljárásokat [50, 127] is, amelyek egyáltalán nem használnak kalibrációs objektumokat. Azonban ezeknek a pontossága messze elmarad a kívánt szinttől, ezért itt nem foglalkozunk velük.

A síklemez alapú módszerek fő problémája, hogy a LiDAR pontfelhőben nehéz az élek pontos detektálása. Különösen, ha alacsony felbontású eszközöket használnak, például a nagyon népszerű Velodyne VLP-16-ot [34], melyet mi magunk is használunk, és amely csak 16-os függőleges lézersugarakkal rendelkezik. További problémák merülnek fel, ha hagyományos sakktáblamintákat használnak síkbeli táblákon. Ez a minta a fekete és fehér színei miatt erősen rontja a LiDAR pontfelhők minőségét [138, 129]. A legtöbb esetben a digitális kamerákat előzetesen kalibrálni kell, vagyis a digitális kamerák belső paraméterei ismertek.

Rodruguez és mtsai. [138] tanulmányukban egy fekete kör alapú síkot (táblát) használnak a sakktáblaminták okozta nagy zaj elkerülése érdekében. Algoritmusuk a kör középpontjának háromdimenziós koordinátáit és a sík normálvektorát keresi. A kalibrációs objektumnak legalább hat pozíciójára van szükség, a LiDAR-kamera merev transzformáció kezdeti becslését a jól ismert Levenberg-Marquardt [96, 108] (LM) algoritmussal finomítják.

Alismail munkatársaival közzétett [6] egy automatikus kalibrációs módszert, amely síkbeli kalibrációs objektumot használ. Egy fekete kör alakú terület található rajta, és ennek a középpontja meg van jelölve. A kör középpontját és normálisát egyetlen nézeti képből számítják ki, és ők is a Random Sample Consensus [46] (RANSAC) módszert alkalmazzák a síkok robusztus kinyerésére. Végül az ICP [35]-szerű algoritmust futtatják, de pont-sík távolság alkalmazásával[29]. A külső paraméterekre a megoldást nemlineáris optimalizálással, a Levenberg-Marquardt [96, 108] numerikus minimalizálás segítségével kapják meg.

A Park és munkatársai által bevezetett [129] módszer egy fehér, homogén, háromszög vagy rombusz alakú sík táblát használ a kalibráláshoz. Több képet kell készíteni a tábláról több pozícióból, legalább három táblát kell egyszerre használni. Algoritmusuk hátránya, hogy a síkbeli tábla térbeli koordinátáit becsléssel, nem pedig méréssel határozzák meg. Ez a tény befolyásolja a kalibráció pontosságát.

Gongék [55] 2013-ban közzétettek egy módszert, amelynél a kalibráláshoz legalább két szkennelésre van szüksége ugyanarról a háromszög tárgyról, amelyet mindkét műszerrel mértek. Ez lényegesen több feldolgozandó adatot eredményez. Az ő módszerüknek 20 másodpercbe telik a kalibrálás kilenc megfigyelés felhasználásával.

[34] Manapság Velodyne Puch néven hivatkoznak rá.
[35] ICP: Iterative Closest Point

Módszerük komoly hátránya, hogy ilyen objektum előállítása nem egyszerű. Ráadásul a kalibrálás sok emberi beavatkozást igényel, a háromszögpontok szétválasztását és a kapcsolódó síkok kiválasztását a képeken kézzel kell elvégezni.

A Velas és mtsai. által [172] javasolt módszer egy szokatlan kalibrációs objektumot használ. Fehér hátteret és egy sík tárgyat feltételeznek, amely négy kör alakú lyukat tartalmaz. Módszerük Levison és Thrun [97] munkáján alapul. A lyukakat mind a 3D LiDAR pontfelhőben, mind a felvett képen automatikusan detektálják. Ezt a módszert azonban nem sikerült reprodukálnunk a Velodyne HDL-16 LiDAR-unkkal, mivel ehhez a helyes működéshez jóval sűrűbb pontfelhőre van szükség.

Geiger és alkotói csapata [53], akik a nagy sikerű KITTI adatbázist [52] készítették, bemutattak egy módszert a LiDAR-kamera páros kalibrálására, amely csak egy LiDAR mérést és egyetlen kamera képet veszi igénybe. A módszer teljesen automatikus, azonban a kalibráláshoz több sakktáblára és legalább két kamerára van szükség.

A disszertációban megmutatom, hogy az elmúlt években munkatársaimmal milyen eredményeket értünk el. Három eljárást ismertetek:

– Egy általános kalibrációs eljárást, amely dobozokat alkalmaz kalibrációs tárgyként.

– Egy másik, szintén általános eljárást, ahol egy szabályos gömb alakú objektum segíti a kalibrációt.

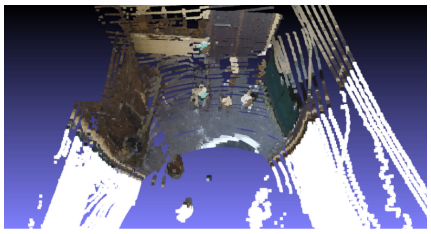
– Sakktábla segítségével egy speciális módszert is bemutatok, ahol az egyedi rögzítés miatt a szabadságfokok száma alacsonyabb.

6.2. LiDAR és kamera kalibrációja dobozok segítségével

Mint a bevezetőben említettem, módszereink célja a kamera-LiDAR rendszer külső paramétereinek pontos meghatározása, szaknyelven kalibrálása. Ennek a célnak az eléréséhez közönséges dobozokat választottunk. Ezeknek a dobozoknak a fő előnye, hogy nincsen szükség rendkívüli nyomtatásra vagy gyártásra. Dobozokat másrészről azért használunk, mert oldalaik egymásra merőlegesek, így e síkok metszéspontjai pontosan kiszámíthatók a kis felbontású LiDAR-ral is megszerzett pontfelhőből.

A síkbeli kalibrációs objektumokat használó más módszereket erősen befolyásolja a síkszélek pontatlan mérése. A mi módszerünk csak a pontok helyét használja, és nem vesz figyelembe más információt, például az intenzitást, így bármilyen típusú háromdimenziós LiDAR letapogató eszközzel használható.

(a) *Kép a három dobozról, amelyek a szenzor előtt helyezkednek el.*



(b) *A jelenet háromdimenziós, színezett pontfelhője.*

38. ábra. *Mérés három doboz segítségével. A jelenetet egy autóra rögzített Velodyne HDL-64-es Lidar vette fel. A pontfelhőt a kamera segítségével színeztük ki: miután a két eszközt egymáshoz kalibráltuk, a LiDAR pontjait a kamerákra tudtuk vetíteni, és a színértéket a pixelekből ki tudtuk nyerni.*

A kalibrációs folyamatot egyetlen kalibrációs objektumra mutatjuk be, azonban a módszer könnyen kiterjeszthető tetszőleges számú doboz használatára. Ez az autonóm vezetésnél gyakori eset, amikor a kamerák és LiDAR-ok körgyűrűs szerkezetben vannak elhelyezve, így néhányuk látótartományának nincs metszete . Mivel a módszer a belső paramétereket használja, a kalibrációs doboz elhelyezése szinte tetszőleges, a kép torzulása nincs hatással a kalibrációra. Az egyetlen követelmény, hogy a kalibrációs doboz három oldalának jól láthatónak kell lennie a pontfelhőben és a képen is.
A módszer bemenetei a következők:

1. pontfelhő(k), amelye(ke)t a LiDAR eszköz tapogatott le;
2. kameránként egy kép;
3. a kamerák belső paraméterei és
4. a doboz(ok) mérete(i).

6.2.1. Összefüggő pontok klaszterezése

A kalibrációs algoritmusunk alapötlete az, hogy ha a kalibrációs doboz hét látható sarka – három látható sík metszéspontja mentén – ismert a térben, és e sarkok vetületei is ismertek a képen, akkor a probléma PnP [95] problémára[36] egyszerűsödik. Az első cél tehát ezeknek a pontoknak a megtalálása, pontosabban becslése a pontfelhőben.

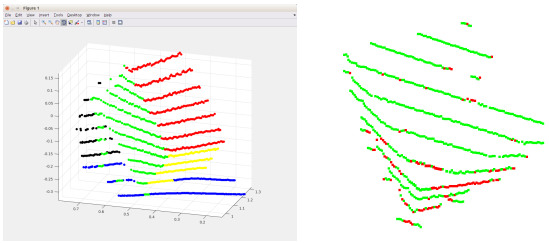
A módszerünk első lépéseként ki kell vágni a kalibrációs doboz környéki területet. A vágás kézzel történik, nem kell nagyon pontosnak lennie. Amint azt később bemutatjuk, a kézi kijelölés után az algoritmus automatikusan el tudja különíteni a doboz oldalait az erre a területre eső egyéb objektumoktól. Ezután az algoritmus síkokat keres a pontfelhőben. A területen belül található pontok alacsony száma miatt egy egyszerű szekvenciális RANSAC [49] alkalmazható. A RANSAC minden egyes iterációban – az euklideszi távolság alapján – kiválasztja azt a síkot, amelyikhez a legtöbb pont tartozik. Ezután a kalibrációs doboz látható oldalait detektálja. Azokat a síkokat

[36] PnP: Perspective n-point Problem.

kell kiválasztani, amelyek a leginkább merőlegesek egymásra. Ehhez bevezetjük az alábbi hibát, ennek lehetséges minimumát kell meghatározni:

$$E(\mathbf{n_1}, \mathbf{n_2}, \mathbf{n_3}) = |\mathbf{n_1}^T \mathbf{n_2}| + |\mathbf{n_1}^T \mathbf{n_3}| + |\mathbf{n_2}^T \mathbf{n_3}|, \tag{60}$$

ahol $\mathbf{n_k}$ a k-adik sík normálvektora ($k \in \{1,2,3\}$). Mivel a pontfelhőben a síkok száma alacsony, így a kimerítő keresés a síkjelöltek között nem igényel sok számítási időt. Ennek a lépésnek az eredménye látható a 39. ábra bal oldali képén, ahol a dobozt egy székre helyeztük, és a sugáron kívül eső pontokat kizártuk. A szekvenciális RANSAC algoritmus [49] öt különböző síkot talál, majd a doboz három oldalaként a piros, a zöld és a sárga síkot választotta ki.



39. ábra. *Bal oldal: Szekvenciális RANSAC által megtalált síkok. Jobb oldal: a zöld pontok a dobozmodellhez tartoznak.*

6.2.2. Robusztikfikáció: kiugró értékek eltávolítása

A javasolt algoritmus következő lépése a kiugró értékek eltávolítása. Fontos ez a részfeladat, mivel a következő lépésben az illesztett síkokat az euklideszi távolságok alapján finomítjuk, és a kiugró értékek rossz illeszkedést eredményezhetnek. A LiDAR pontfelhőt erősen befolyásolhatja a zaj, például a textúra okozta távolság-tükröződés torzítása, ahogyan azt Park és munkatársai is megemlítik [129]. A kiugró pontokat – amelyek a kalibrációs dobozba tartoznak, de erős zajhatás alatt állnak –, kizárjuk a további számításokból. Az előző lépésben kapott síkok pontjait $L1$, $L2$ és $L3$ néven jegyezzük meg, a sorrend nem számít.

Az algoritmus ismét a RANSAC módszert [46] használja a kiugró értékek meghatározására, de ezúttal a modell három merőleges síkot tartalmaz. Az előző lépésben azokat a síkokat választottuk ki, amelyek a 60. formula alapján a legkisebb hibát adják, ekkor a merőlegesség még nem volt szempont. A modellillesztés a következőképpen történik: először három pontot választunk ki $L1$-ből, ezek a pontok meghatároznak egy síkot, majd további két pontot választunk ki $L2$-ből, ez a két pont meghatároz egy újabb síkot, amely merőleges az elsőre, és végül még egy pontot választunk ki $L3$-ból, ez határozza meg egyértelműen a harmadik síkot, amely merőleges az első kettőre.

Az előző szakasz és ez a rész redundánsnak tűnhet első látásra, de ez mégsem igaz. Az első célja a kalibrációs doboz elkülönítése a többi objektumtól, míg ez a második lépés a zajos dobozpontok meghatározására szolgál.

6.2.3. A doboz paramétereinek iteratív finomítása

Miután a kiugró értékek kikerültek a ponthalmazokból, két lépésből álló iteratív algoritmus finomítja a doboz síkjait a pontokhoz. A finomítás feladata, hogy három, egymásra merőleges síkot pontosan illesszen a ponthalmazokhoz.

Forgatás meghatározása. A forgatási lépésben két síkot választunk ki, és elforgatjuk őket a metszésvonaluk mentén – azaz a doboz éleinek mentén. Az elforgatás minimalizálja a pontok és a síkok közötti négyzetes hibák összegét.

Legyen $\mathbf{p_j^1}$, $\mathbf{p_j^2}$, és $\mathbf{p_j^3}$ az $L1$, $L2$ és $L3$ tartományban a pontok jele, ahol j a pontok indexe. Jelölje $\mathbf{q^i}$ az i-edik síkban fekvő pontot, amelynek nem kell a $\mathbf{p_j^i}$ halmazban lennie. Legyen $\mathbf{n}^i$ az i-dik sík normálvektorai. Az i-edik síkot egyértelműen meghatározza a $\mathbf{q^i}$ pont és az $\mathbf{n}^i$ normálvektor.

Az illesztési problémát úgy tekintjük, hogy a pontok négyzetes távolságainak összegét szeretnénk minimalizálni a megfelelő síkhoz képest. A legkisebb négyzetek költségfüggvényhez a pont-sík távolságokat használtuk. Ezt a következőképpen írhatjuk le:

$$C = \sum_{i=1}^{3} \sum_{j=1}^{m_i} \left| \left(\mathbf{p_j^i} - \mathbf{q^i}\right)^T \mathbf{n^i} \right|^2 . \tag{61}$$

Az általánosság elvének megsértése nélkül a világ koordináta-rendszerét a kockasíkok metszéspontjához rögzítjük, a koordináta-rendszer fő irányait pedig a kocka élei adják. Tekintsük a harmadik tengely körüli γ szögű elforgatást! Ekkor a forgatási mátrix a következő:

$$\mathbf{R}_Z^T = \begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{62}$$

ahol $c = \cos\gamma$ és $s = \sin\gamma$. A három sík normálvektorai a következőek: $[1 \quad 0 \quad 0]^T$, $[0 \quad 1 \quad 0]^T$, és $[0 \quad 0 \quad 1]^T$. A forgatás az illesztési hibát nem befolyásolja, ezért a minimalizálási feladatot úgy írhatjuk, hogy

$$C'' = \sum_{i=1,2} \sum_{j=1}^{m_i} \left| \left(\begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p_j^i} - \mathbf{q^i} \right)^T \mathbf{n^i} \right|^2 . \tag{63}$$

Elemi átalakítások után a költségfüggvény $|\mathbf{Ax}|$ alakba írható $\mathbf{x}^T\mathbf{x} = 1$ feltétellel, ahol

$$\mathbf{A} = \begin{bmatrix} x_1^1 & -y_1^1 \\ \vdots & \vdots \\ x_{m_1}^1 & -y_{m_1}^1 \\ y_1^2 & x_1^2 \\ \vdots & \vdots \\ y_{m_2}^2 & x_{m_2}^2 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} c \\ s \end{bmatrix}. \tag{64}$$

Az m_1 és m_2 konstansok az első és a második síkhoz tartozó pontszámokat jelölik. A $\mathbf{x}$ értékére legkisebb négyzetes értelemben optimális megoldást az $\mathbf{A}^T\mathbf{A}$ mátrix legkisebb sajátértéknek megfelelő sajátvektorából nyerjük ki[37]. A γ szöget a következőképpen számítjuk ki: $\gamma = atan2(s, c)$.

[37] Az $\mathbf{A}^T\mathbf{A}$ mátrixnak mindig két nemnegatív valós sajátértéke van.

A X és Z tengelyek körüli elforgások hasonló módon számíthatóak ki.

Eltolás meghatározása

A dobozhoz tartozó eltolás a síkok három normálisa mentén, egyesével történik. Az eltolás meghatározásához nagy segítség, hogy eltoláskor csak a kiválasztott sík pontillesztési hibája változik, ha a doboz eltolása párhuzamos a sík normálisával.

A 61. összefüggésben leírt költségfüggvény így módosul:

$$C''' = \sum_{j=1}^{m_i} \left(\left(\mathbf{p_j^i} - \mathbf{q^i} - \alpha \mathbf{n} \right)^T \mathbf{n^i} \right)^2 \tag{65}$$

ahol α az eltolás hosszát jelzi. Az i-edig síkhoz m_i darab pont tartozik. Erre a matematikai problémára könnyen lehet legkisebb négyzetes értelemben zárt alakú megoldást adni, hiszen

$$\alpha = \frac{\sum_{j=1}^{m_i} \left(\mathbf{p}_j^i - \mathbf{q}^i \right)^T \mathbf{n}^i}{m_i \mathbf{n}^T \mathbf{n}^i}.$$

A $\mathbf{t}$ eltolásvektort a három merőleges irányban egyenként kell kiszámítani. A feladat azonban egyetlen lépésben is megoldható. Ebben az esetben tekintsük az eltolási problémát a következő költségfüggvény minimalizálásának:

$$C'''' = \sum_{i=1}^{3} \sum_{j=1}^{m_i} \left| \left(\mathbf{p_j^i} - \mathbf{q^i} - \mathbf{t} \right)^T \mathbf{n^i} \right|^2. \tag{66}$$

Mindez a $\mathbf{Bt} = \mathbf{c}$ homogén lineáris egyenletrendszer alakjába írható, amennyiben

$$\mathbf{B} = \begin{bmatrix} \mathbf{n^{1T}} \\ \vdots \\ \mathbf{n^{1T}} \\ \mathbf{n^{2T}} \\ \vdots \\ \mathbf{n^{2T}} \\ \mathbf{n^{3T}} \\ \vdots \\ \mathbf{n^{3T}} \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \mathbf{n^{1T}} \left(\mathbf{p_1^1} - \mathbf{q^1} \right) \\ \vdots \\ \mathbf{n^{1T}} \left(\mathbf{p_{m_1}^1} - \mathbf{q^1} \right) \\ \mathbf{n^{2T}} \left(\mathbf{p_1^2} - \mathbf{q^2} \right) \\ \vdots \\ \mathbf{n^{2T}} \left(\mathbf{p_{m_2}^2} - \mathbf{q^2} \right) \\ \mathbf{n^{3T}} \left(\mathbf{p_1^3} - \mathbf{q^3} \right) \\ \vdots \\ \mathbf{n^{3T}} \left(\mathbf{p_{m_3}^3} - \mathbf{q^3} \right) \end{bmatrix}, \tag{67}$$

ahol az $\mathbf{n^{jT}}$ sorvektor az $\mathbf{n^j}$ oszlopvektor transzponáltját jelöli. A költségfüggvény minden egyes tagja egy egyenletet ad a rendszerhez. A megoldás a jól ismert $\mathbf{t} = \left(\mathbf{B}^T \mathbf{B} \right)^{-1} \mathbf{B}^T \mathbf{c}$ formula segítségével számítható ki. Mivel az $\mathbf{n^1}$, $\mathbf{n^2}$ és $\mathbf{n^3}$ vektorok merőlegesek egymásra, a $\mathbf{B}$ mátrix mindig nem-szinguláris, így a mátrix inverze mindig kiszámítható.

6.2.4. Konvergencia

A fenti forgatási és eltolási lépéseket konvergenciáig ismételjük. Azt tapasztaltuk, hogy kevesebb mint 30 iteráció elegendő volt a szigorú leállási feltétel teljesítéséhez. A javasolt módszert úgy futtattuk, hogy a koordináta-rendszer origója, ahol a síkok a tengelyhez igazodtak, adta a kezdeti eltolási/forgatási paramétereket. Azt tapasztaltuk, hogy az iteráció kivétel nélkül minden esetben konvergált.

6.2.5. Külső paraméterek kalibrálása

Külső paraméterek alatt a kamera és a lidar koordináta-rendszerei közötti transzformációt értjük, amely egy egybevágósági transzformációt jelent háromdimenziós elforgatással és eltolással.

A doboz oldalainak iteratív finomítása után a kalibrációs doboz sarokpontjai kiszámíthatók, hiszen a doboz méretei ismertek. A 44. ábra színes pontjai a doboz sarkait jelölik, a dobozok teljes környezete volt az algoritmus bemenete, miután manuálisan kivágtuk az eredeti pontfelhőkből.

Ha egy LiDAR-LiDAR páros külső paramétereit ismerni kell, az algoritmus különkülön is meg tudja találni a LiDAR pontfelhőkben a sarokpontokat. Ezután a két pontfelhő közötti transzformáció kiszámítható a dobozok sarkainak helyeiből pontregisztrációval [8, 69, 70][38].

A kalibrációs módszer LiDAR-kamerás rendszerek kalibrálására is használható. Ebben az esetben a kalibrációs doboz sarkainak vetületeit kell kiválasztani a feldolgozott képen. Ezután a kiválasztott pontokat a Harris [58] sarokdetektorral finomítjuk. A külső paraméterek megtalálásának problémája ekvivalens egy PnP problémával, mivel a 3D-2D pont-pont megfeleltetések ismertek. Mi az Effective-PnP (EPnP) algoritmust [94] alkalmaztuk. Mivel a kamerák már kalibrálva vannak, a kamera belső paramétereit a PnP futtatása során figyelembe vettük. A radiális és tangenciális torzításnak nincs hatása a kalibráció pontosságára, hiszen a képeket a kalibráció után ezen paraméterek ismeretében ki lehet egyenesíteni[39].

6.2.6. Vizsgálati eredmények

A módszert szintetikus és valós tesztadatokon egyaránt ellenőriztük. A szintetikus tesztekhez a Blender Sensor Simulator (Blensor) [1] programot használtuk, amely egy nyílt forráskódú szimulációs csomag a széles körben használt 3D modellező és renderelő szoftverhez[40]. A valós eredmények nagy és kis felbontású LiDAR-eszközökből származó színes pontfelhőkből és pontfelhő-fúziókból állnak.

Szintetikus tesztek

A Blender egy ingyenes és nyílt forráskódú szoftver 3D szimulációhoz, rendereléshez, animációhoz és modellezéshez. Támogatja több kamera használatát, számos paramétert állíthatunk be, akár fotórealisztikus képek is menthetők. A Blender szintetikus környezetén belül sakktáblákat és egy kalibrációs dobozt modellezünk a valósághű tesztforgatókönyvek szimulálásához. A Blensor [1][41] egy szenzorszimulációs csomag, amely a Blender képességeit különböző LiDAR és fényidő-alapú eszközökkel bővíti. A szintetikus tesztekhez a Velodyne-64 LiDAR-t használtuk az alapértelmezett beállításokkal, azonban lehetőség van a pásztázási tartomány, a zajszint, a forgási sebesség beállítására. A mi tesztünkben csak a zajszintet változtattuk. Az érzékelőtől mért alapigazság (GT) távolsághoz Gauss-zajt adtunk hozzá, amelynek várható értéke zérus, szórása pedig 0 és $0{,}14$ között változott. Ez azt jelenti, hogy a zajos pontok a távolságérzékelő által vetett sugarakon helyezkedtek el.

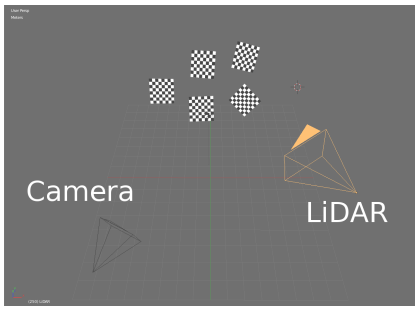
[38] A feladat megoldása a D függelékben olvasható.

[39] Például az OpenCV függvénykönyvtárban az *undistort* függvény alkalmas a képek kiegyenesítésére.

[40] www.blender.org

[41] www.blensor.org

A virtuális jelenet egy példája látható a 40 ábrán, amelyet a Blensor [1] szimulációs csomag hoz létre. A jelenet létrehozása során szem előtt tartottuk, hogy három riválisalgoritmusnak ugyanazt a kamera-LiDAR rendszerbeállítást kell használnia, és mindegyik más-más objektum(ok)at használt a kalibráláshoz.



40. ábra. *Példa a Blensor által létrehozott tájképre [1]. A bal és jobb oldali piramisok a kamera, illetve a LiDAR helyét mutatják.*

Az alábbi módszereket hasonlítottuk össze:

– Az első algoritmus, amelyet Geiger kutatócsapata mutatott be [53], sakktáblákat használ.

– Park és munkatársainak módszere [129] poligonális táblákat alkalmaz,

– míg a javasolt algoritmus egy (virtuális) dobozt alkalmaz a kalibrálandó berendezések relatív pózának meghatározásához.

Geigerék algoritmusának, amelyet a grafikonokon *'kitti'* néven jelöltem, legalább két kameraképre van szüksége a kalibráláshoz, minél több sakktábláról. A teljesen automatikus módszer a kameraképeket feldolgozva rekonstruálja a sakktáblákat, majd megpróbálja egyesíteni a pontfelhőt a LiDAR érzékelő által felvett pontfelhővel. Az utolsó lépés problémás lehet, mert sokszor téved, hiszen nagymértékben függ a kalibrálás környezetétől. A kalibrációs adatok megszerzéséhez a szerzők weboldalán található online demo[42]-t használtuk.

A *'polygonal'* néven megjelölt algoritmust Park és munkatársai fejlesztették [129]. Tesztjeinkben a módszert a szerzők javaslatának megfelelően öt gyémánt alakú síkbeli táblával használtuk. E táblák pontjait egyenként, kézzel kell kiválasztani a LiDAR pontfelhőben. Ezután a táblák síkjait a RANSAC segítségével becsültük meg, és a LiDAR pásztázási vonalak segítségével kiszámítottuk a virtuális pontokat. Ezek a pontok az egyes táblák éleinek becslésére szolgálnak, és ezen élek metszéspontjai a tábla sarkait eredményezik. A metszéspontok vetületeit a FAST [139, 140] jellemzőpontokból választottuk ki. Végül a külső paraméterek kinyerésére szinguláris érték dekompozíciót (SVD) és a Levenberg-Marquardt algoritmust [96, 108] használtunk.

Valós kísérleteink során azt tapasztaltuk, hogy a LiDAR térbeli pontfelhő zajszintje nemcsak a tárgy textúrájától és anyagától, hanem a LiDAR érzékelő típusától is függ. Még az azonos gyártótól származó LiDAR-ok is különbözhetnek – a Velodyne-16 ritkás pontfelhője pontosabb, mint a nagy modelljéből, a Velodyne-64-ből származó sűrűbb pontfelhő.

[42] www.cvlibs.net/datasets/kitti

Az algoritmusokat a LiDAR pontfelhőre ható Gauss-zajjal teszteltük. A szórás (σ) 0 és 0.14 között változott. A szintetikus tesztekben a LiDAR érzékelőt a $(4.72945, -$ $-5.24017, 8.76321)$, míg a főkamerát a $(-5.5727, -6.98041, 3.55163)$ ponton helyeztük el, a távolságokat méterben mértük. Két kamerát és öt sakktáblát használtunk a Geiger-módszerhez, öt négyzet alakú táblát a Parkék által [129] javasolt módszerhez, és csak egy kalibrációs dobozt helyeztünk el a saját algoritmusunkhoz. A sakktáblák és a síkbeli táblák mérete 2×2 méter, a kalibrációs dobozé $1 \times 2 \times 3$ méter.

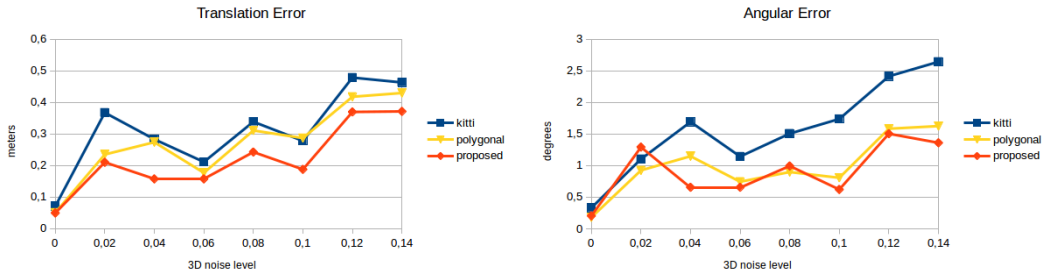
A módszerek értékeléséhez összehasonlítottuk a külső paramétereket (forgatás és eltolás) és az alapigazságot (GT). A 47. ábra felső ábráján látható fordítási hiba az alapigazság forgatási vektor és az algoritmusok által becsült vektor közötti euklideszi távolság.

A forgatási hiba kiszámítása nem triviális, mivel egy forgatás többféleképpen is ábrázolható: három szög, vagy ortonormális mátrix, vagy éppen egy tengely és egy forgatási szög segítségével. Több hibametrikát is összehasonlítottunk, ezek jellemzői a teszteseteink esetében megegyeztek. Egyszerűsége miatt a hibát reprezentáló $\mathbf{R}_{GT}^T\mathbf{R}$ mátrix forgási szögét a jól ismert képlettel [36] számoltuk ki az alábbiak szerint:

$$\alpha = cos^{-1}\left(\text{tr}\left(\mathbf{R}_{GT}^T\mathbf{R}\right) - 1\right)/2), \tag{68}$$

ahol $\mathbf{R}_{GT}$ a Blensor adatokból kinyert alapigazság (Ground Truth – GT) forgatási mátrix, $\mathbf{R}$ pedig a tesztelt algoritmusok által kapott forgatási mátrix.

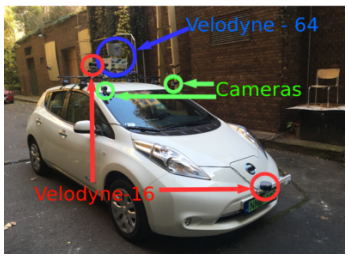
A 47. ábra az algoritmusok eltolási és forgatási hibáit mutatja. Egyértelműnek tűnik, hogy Parkék módszere, amelyet az ábrákon *'polygonal'* jelöli, és a mi módszerünk, amelyet *'proposed'* jelöl, nagyon pontosan ki tudja számítani a forgatási mátrixokat, hibájuk még nagy szórású Gauss-zaj jelenlétében sem haladja meg az $1{,}5$ fokot. Ez jó eredmény, mert még a forgatási mátrixok kis hibája is erősen befolyásolhatja a színes pontfelhő eredményét, különösen akkor, ha az objektumok nagy távolságban helyezkednek el. Az algoritmusok eltolási hibája 5 centiméter és 0,5 méter között változik, lásd a 47. ábra felső diagramját. Látható, hogy a javasolt módszer lényegesen pontosabb, mint a többi. A különbségek 5 és 10 centiméter között vannak. Mind a forgatási, mind az eltolási hibák megközelítőleg lineárisan növekednek a Gauss-zajjal.



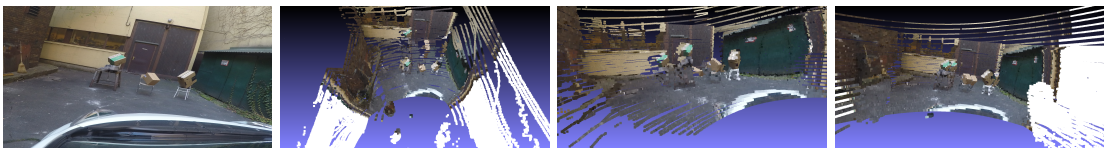
41. ábra. *Eltolási (balra) és forgatási hiba (jobbra) növekvő, Gauss eloszlású zajjal szennyezett háromdimenziós koordináták esetén.*

Valós Tesztek

A tesztelési helyszínen 3 kalibrációs dobozt helyeztünk el. Az összehasonlítás kedvéért a LiDAR pontfelhő egyes pontjainak a kameraképre történő visszavetítését el-



42. ábra. *Egy Velodyne-64, kettő Velodyne-16 és szintén kettő kamera van felszerelve a tesztautóra.*



43. ábra. *Három LiDAR-felvétel pontfelhőfúziója, amelyet az autonom autó bal oldali kamerája alapján színezünk ki. Az első két képen a kamerakép és a pontfelhő látható távolabbról. A két jobb oldali kép a kamera nézőpontjából, illetve a Velodyne-64 LiDAR-ról készült. A pontfelhő színezése során nem vettük figyelembe a LiDAR különböző nézetei által okozott takarást, a fehér pontok a kamera nézetén kívül helyezkednek el.*

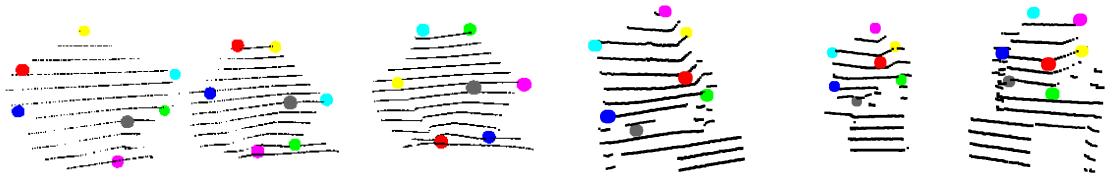
végeztük. A tesztelés a következőképpen zajlik: először három kartondobozt helyeztünk a székek tetejére, hogy a Velodyne-16 LiDAR számára látható legyen. Öt sakktáblát helyeztünk el a Geiger-csoport módszerének [53] futtatásához. Egy képet a kamera eredeti helyzetében, egy másikat pedig egy másik helyen készítettünk, mivel a kalibráláshoz legalább két képre van szüksége a módszernek.

A 45. ábrán a részleges pontfelhők újravetített pontjai láthatóak, a feldolgozott jelenet pedig a 44. ábrán látható. A bal oldali képen a Geiger-féle módszer segítségével kapott külső paraméterek, a jobb oldalon a mi algoritmusunkkal kapott külső paramétereket használtuk a visszavetítéshez. Megjegyezném, hogy a sakktábla sarkainak, a kalibrációs dobozok tetejének és a zöld szék lábának pontjainak vetített helyei helytelenül eltolódtak. Az elsodródás a módszer pontatlanságát jelenti. A jobb oldalon ezzel szemben a visszavetített pontok egybevágnak a megfelelő képi régiókkal, ami arra utal, hogy a javasolt módszer minősége jelentősen nagyobb.

A módszert az autonóm autónk adatfelvevő-rendszere által nyert valós adatokon is teszteltük. Az autóra egy nagy felbontású és két kisebb felbontású LiDAR érzékelőt, valamint két kamerát szereltünk fel, lásd a 42. ábrát. A kalibrációt a SZTAKI hátsó udvarában végeztük el. A 43. ábra a kalibráció eredményét mutatja, amely a nagy és a két alacsony felbontású LiDAR eszköz pontfelhőfúziója, és az RGB kamerák intenzitásértékeinek felhasználásával színezett: a képtérben lévő helyeket a térbeli pontok kamerákra való vetítésével kaptuk meg a javasolt algoritmus által kapott külső kameraparaméterek felhasználásával.

6.3. Kalibráció gömbökkel

A gömb felülete nagyon szerencsés abból a szempontból, hogy mind a LiDAR pontfelhőjén [171], mind a kameraképen a gömb vetületének kontúrja [166, 162] nagy



44. ábra. *Az algoritmus még a Velodyne VLP-16 által generált ritka pontfelhőben is megtalálja a sarkokat.*

45. ábra. *A visszavetített doboz és sakktáblapontok a 44. ábrán látható tesztesetről. A Geiger és munkatársai módszerével [53] becsült kamera paramétereket a bal oldali eredményén használtuk, a javasolt módszer eredménye a jobb oldalon látható.*

pontossággal meghatározható. A geometriában régóta ismert tény, ahogyan azt később itt is belátjuk, hogy a gömb képének kontúrja középpontos síkra vetítés (lyukkamera) alkalmazása esetén ellipszis, hiszen a gömb összekötve a perspektív kamera fókuszpontjával kúpfelületet alkot, a kontúr pedig ennek a kúpfelületnek a metszete a képsíkkal.

6.3.1. Gömb helyének meghatározása LiDAR pontfelhőből

A gömb felületének implicit alakja az alábbi – már a középiskolában is megtanult – módon írható le:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2, \tag{69}$$

ahol $\mathbf{x}_0 = \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^T$ a gömb középpontja és r jelöli a sugarat. Egy $\mathbf{x} = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \in \mathbb{R}^3$ térbeli pont akkor esik a gömb felületére, ha az implicit egyenletet kielégíti. A 69. egyenletből, az eredeti alakot megtartva, elemi átalakításokkal írhatjuk fel az alábbi implicit formulát, ha azt akarjuk elönteni, hogy az $\mathbf{x}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T$ pont rajta van-e a síkon:

$$x_i^2 - 2x_0x_i + x_0^2 + y_i^2 - 2y_0y_i + y_0^2 + z_i^2 - 2z_0z_i + z_0^2 - r^2 = 0. \tag{70}$$

Amennyiben több pontunk is van: $\{\mathbf{x}_i\}$, $i \in \{1,2,\ldots,N\}$, mátrixos alakban a következő rendszert alkothatjuk meg:

$$\begin{bmatrix} -2x_1 & -2y_1 & -2z_1 & 1 \\ -2x_2 & -2y_2 & -2z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -2x_N & -2y_N & -2z_N & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ x_0^2 + y_0^2 + z_0^2 - r^2 \end{bmatrix} = - \begin{bmatrix} x_1^2 + y_1^2 + z_1^2 \\ x_2^2 + y_2^2 + z_2^2 \\ \vdots \\ x_N^2 + y_N^2 + z_N^2 \end{bmatrix}. \tag{71}$$

Ez az inhomogén rendszer $N > 4$ esetében túlhatározott. Szerencsére ekkor is meg lehet oldani a bal oldali együtthatómátrix Moore-Penrose pszeudo-inverzének felhasználásával. Az inverzet balról meg kell szorozni az inhomogén rendszer jobb oldali vektorával. Eredményképpen egy négy dimenziós vektort kapunk: $\boldsymbol{v} = [v_1, v_2, v_3, v_4]^T$, melynek első három koordinátája adja a gömb középpontját, a negyedikből pedig a gömb sugarát lehet kiszámolni: $r = \sqrt{x_0^2 + y_0^2 + z_0^2 - v_4}$.

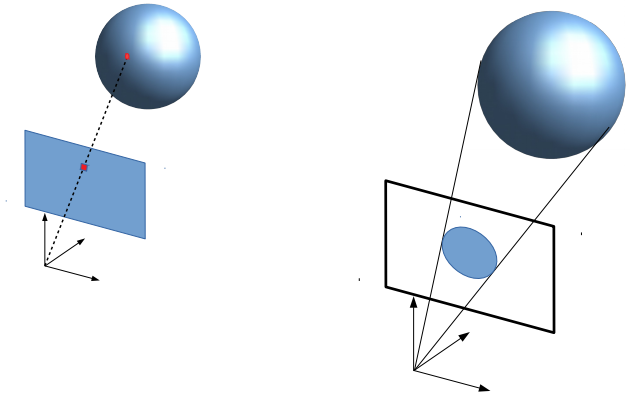
Ennek a megoldásnak nagy előnye, hogy rendkívül gyors, és mind az egyértelmű, mint a túlhatározott eseteket képes kezelni. A 69. egyenlet geometriailag is értelmezhető, egy pont távolságának a négyzetét adja meg a körív legközelebbi pontjához képest. A módszer hátránya, hogy pszeudo-inverz alkalmazásával legkisebb négyzetes értelemben lehet optimumot meghatározni, de itt az eredeti hiba már a geometria távolság négyzete. Ezért legkisebb négyzetes hibaminimalizálás a geometriai hibák összegének már a negyedik hatványát fogja figyelembe venni. Így a módszer kevésbé robusztus a sima kettes norma alkalmazásához képest, pedig már a kettes norma sem tekinthető annak. Ezért a kiugró pontokat mindenképpen ki kell szűrni a pontos becsléshez. A függelék E.2.1. fejezetében ismertetett RANSAC [46] algoritmus ehhez a szűréshez kiválóan alkalmazható.

6.3.2. Gömb helyének meghatározása kameraképekből

Ebben a szakaszban megmutatom, hogyan lehet kamera képein a kontúrpontok segítségével a gömb középpontját meghatározni. Először térbeli metszéseken keresztül levezetem, hogyan lehet a gömb kontúrpontjait meghatározni a képen. Ehhez kivetítem a képpontokat a térbe, és azokat a pontokat választom kontúrnak, amely éppen érinti a gömböt. Az így kapott kontúrpontok ellipszist írnak le. Utána a kapott paraméterekből azt is leírom, hogy ellipszisillesztés segítségével hogyan lehet az ellipszis paramétereiből a gömb koordinátáit meghatározni, ha a gömb sugara ismert.

Sugármetszés gömbbel.

A vetítősík és egy általános gömb metszéspontjának meghatározása kicsit bonyolultabb. A gömb látható képének kontúrja a képen egy ellipszis, ennek meghatározását ismertetem ebben a szakaszban.



46. ábra. *Gömb perspektív vetítése képsíkra. A bal oldalon a piros pixel meghatározza a szaggatott vonallal rajzolt vetítőegyenest, amely a gömbfelületet a piros pontban metszi el. A jobb oldali ábrán látszik, hogy a gömb perspektív vetítése a képpsíkra egy ellipszissel határolt területet eredményez.*

Vektoros formában a 69. egyenlet lényegesen tömörebb formában adható meg:

$$(\mathbf{x}-\mathbf{s})^T(\mathbf{x}-\mathbf{s})-r^2=0, \tag{72}$$

amennyiben $\mathbf{s}=\begin{bmatrix}x_0 & y_0 & z_0\end{bmatrix}^T$ vektor jelöli a gömb középpontját, $\mathbf{x}=\begin{bmatrix}x & y & z\end{bmatrix}^T$ vektor pedig a térbeli pontot.

A gömb maga egy speciális kúpszelet. A teljesség kedvéért a metszési feladatot általános kúpszeletekre mutatom meg. Az általános kúpszelet implicit egyenlete így írható fel:

$$\begin{bmatrix}\mathbf{x}^T & 1\end{bmatrix}\mathbf{M}\begin{bmatrix}\mathbf{x}\\1\end{bmatrix}=0, \tag{73}$$

ahol $\mathbf{M}=\mathbf{M}^T$ egy 4×4 egy szimmetrikus mátrix, amelyet kisebb részekre is lehet bontani:

$$\begin{bmatrix}\mathbf{x}^T & 1\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{M}} & \mathbf{b}\\ \mathbf{b}^T & \nu\end{bmatrix}\begin{bmatrix}\mathbf{x}\\1\end{bmatrix}=0. \tag{74}$$

Fontos tény, hogy a kisebb, 3×3-as $\tilde{\mathbf{M}}$ részmátrix szintén szimmetrikus.

Gömb esetén az $\mathbf{M}$ mátrix egyszerűbb alakra hozható:

$$\mathbf{M}=\begin{bmatrix}\mathbf{I} & -\mathbf{s}\\ -\mathbf{s}^T & \mathbf{s}^T\mathbf{s}-r^2\end{bmatrix}. \tag{75}$$

Így ebben a speciális esetben $\tilde{\mathbf{M}}=\mathbf{I}, \mathbf{b}=-\mathbf{s}$ és $\nu=\mathbf{s}^T\mathbf{s}-r^2$.

A kúpszeletek kontúrjai a képen.

Lyukkamera esetén, ahogyan azt korábban láttuk a 4. egyenletben, $\mathbf{x}=\alpha\mathbf{K}^{-1}\mathbf{u}$.

Ha ezt rávetítjük a képre, a következő formulát kapjuk:

$$\begin{bmatrix}\alpha\mathbf{u}^T\mathbf{K}^{-T} & 1\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{M}} & \mathbf{b}\\ \mathbf{b}^T & \nu\end{bmatrix}\begin{bmatrix}\alpha\mathbf{K}^{-1}\mathbf{u}\\1\end{bmatrix}=0.$$

Ez egy $\alpha^2\mathbf{u}^T\mathbf{K}^{-T}\tilde{\mathbf{M}}\mathbf{K}^{-1}\mathbf{u} + 2\alpha\nu\mathbf{b}^T\mathbf{K}^{-1}\mathbf{u} + \nu = 0$ alakban felírható másodfokú egyenletet ad. Geometriailag mindez azt jelenti, hogy egy sugárnak a felülettel két metszéspontja van. A kontúrpontok esetén a vetítősugár éppen érinti a gömböt, ezért egy megoldást kapunk. Ez algebrailag akkor lehetséges, ha a másodfokú egyenlet diszkriminánsa éppen nulla, azaz $\left(2\nu\mathbf{b}^T\mathbf{K}^{-1}\mathbf{u}\right)^2 - 4\left(\mathbf{u}^T\mathbf{K}^{-T}\tilde{\mathbf{M}}\mathbf{K}^{-1}\mathbf{u}\right)\nu = 0$.

Átírva:

$$\left(\mathbf{b}^T\mathbf{K}^{-1}\mathbf{u}\right)^2 = v\mathbf{u}^T\mathbf{K}^{-T}\tilde{\mathbf{M}}\mathbf{K}^{-1}\mathbf{u}. \tag{76}$$

Ezzel pedig megkaptuk a kúpszelet egyenletét a képen.

Speciális eset: gömb vetülete.

Ahogy azt fentebb megmutattam, gömb esetén felírható, hogy $\mathbf{b} = -\mathbf{s}$, $\nu = \mathbf{s}^T\mathbf{s} - r^2$ és $\tilde{\mathbf{M}} = \mathbf{I}$. Ha a kamerák kalibráltak, a kép pixeleinek koordinátáit normalizálni lehet, ami után $\mathbf{K} = \mathbf{I}$. Ezért a 76. egyenlet a következő alakra hozható:

$$\left(\mathbf{s}^T\mathbf{u}\right)^2 = \left(\mathbf{s}^T\mathbf{s} - r^2\right)\mathbf{u}^T\mathbf{u}. \tag{77}$$

Ha az $\mathbf{x}$ vektornak a koordinátáit behelyettesítjük, az

$$\left(ux_0 + vy_0 + z_0\right)^2 = \left(x_0^2 + y_0^2 + z_0^2 - r^2\right)\left(u^2 + v^2 + 1\right) \tag{78}$$

összefüggést kapjuk. Ez pedig elemi átalakítások után átírható a

$$\begin{aligned}&\left(r^2 - y_0^2 - z_0^2\right)u^2 + \left(r^2 - x_0^2 - z_0^2\right)v^2 + \\ &2x_0y_0uv + 2x_0z_0u + 2y_0z_0v + r^2 - x_0^2 - y_0^2 = 0\end{aligned} \tag{79}$$

implicit formába.

Ez egy másodrendű görbe, azaz egy kúpszelet a képsíkon az

$$A\hat{u}^2 + B\hat{u}\hat{v} + C\hat{v}^2 + D\hat{u} + E\hat{v} + F = 0 \tag{80}$$

alakban, ahol a paraméterek a következőek:

$$\begin{aligned} A &= r^2 - y_0^2 - z_0^2, & B &= 2x_0y_0, \\ C &= r^2 - x_0^2 - z_0^2, & D &= 2x_0z_0, \\ E &= 2y_0z_0, & F &= r^2 - x_0^2 - y_0^2. \end{aligned} \tag{81}$$

2023-as folyóiratcikkünkben [166] megmutattuk, hogy a kúpszelet ellipszis lesz, amennyiben a gömb teljes terjedelmében a kamera előtt van.

Amennyiben legalább öt kontúrpont adott, a Fitzgibbon algoritmus [47] segítségével az ellipszis paramétereit meg tudjuk határozni.

A paraméterekre kapott értékek alapján írhatjuk, hogy

$$\begin{aligned} \frac{B}{D} = \frac{y_0}{z_0} &\rightarrow Bz_0 - y_0D = 0, \\ \frac{B}{E} = \frac{x_0}{z_0} &\rightarrow Bz_0 - x_0E = 0, \\ \frac{D}{E} = \frac{x_0}{y_0} &\rightarrow Dy_0 - Ex_0 = 0. \end{aligned}$$

Mátrixos alakban ugyanez egyszerűbben is kifejezhető:

$$\begin{bmatrix} 0 & -D & B \\ -E & 0 & B \\ -E & D & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = 0. \tag{82}$$

Fontos megjegyezni, hogy ennek a mátrixnak az elemei lineárisan összefüggők, ezért a mátrix szinguláris, tehát a determinánsa nulla. Ebből pedig egyenesen az következik, hogy a megoldásnak van egy magtere (nullvektora).

A gömb középpontja a nullvektor alapján határozható meg. Azonban a nullvektor, ahogyan a sajátvektorok is, skálázásra érzéketlenek. Tehát, ha $[x_0 \quad y_0 \quad z_0]^T$ nullvektor, akkor $[\alpha x_0 \quad \alpha y_0 \quad \alpha z_0]^T$ is az, ahol α valós szám az (ismeretlen) skála.

Ezt a skálát szerencsére meg tudjuk határozni, ha a gömb r sugarát ismerjük. A 81. egyenlet első paraméteréből kifejezhető, hogy

$$A = r^2 - \alpha^2 y_0^2 - \alpha^2 z_0^2.$$

Ekkor α az alábbi kifejezés segítségével számítható:

$$\alpha = \pm \sqrt{\frac{r^2 - A}{y_0^2 + z_0^2}}.$$

A képletben a $\pm$ előjelet is meg lehet határozni, azt a megoldást kell kiválasztani, ahol a kapott gömb a kamera előtt és nem mögött van.

6.3.3. Kamera-LiDAR kalibráció

Ahhoz, hogy a kamerákat és a LiDAR mérőeszközöket egymáshoz tudjuk kalibrálni, utolsó lépésként a gömbök helyét a LiDAR pontfelhőkön [171] és a kameraképeken [166, 162] meg kell határozni. Azaz, ha egy konkrét gömböt a kamerával és LiDAR eszközzel (időben szinkronizáltan) rögzítünk, a kontúrpontokból, illetve a kapott pontfelhőből meghatározható a gömb helye mindkét rendszeren. Ezt a két pontot, melyet p_{LiDAR}-ral és p_{camera}-val jelölünk, egy egybevágósági transzformáció képes egymásba transzformálni:

$$\mathbf{p}_{LiDAR} = \mathbf{R}\mathbf{p}_{camera} + \mathbf{t}, \tag{83}$$

ahol $\mathbf{R}$ a forgatási mátrix (ortonormalitás miatt három szabadságfok), $\mathbf{t}$ az eltolásvektor (másik három szabadságfok). A skálázást abben az esetben nem kell kiszámítani, ha a gömbök sugarát pontosan ismerjük.

Miután minden egyes pontra háromdimenziós egyenletünk van, pontonként három egyenletet kapunk. Ezért első hallásra azt hihetjük, hogy kettő pont elegendő a transzformáció meghatározásához. Azonban geometriai megfontolások alapján legalább négy pontra van szükség. Erre a problémára legkisebb négyzetes értelemben optimális megoldást [8] lehet adni, melyet pontregisztációs eljárás néven is szoktak hivatkozni. A D függelékben ismertetem az algoritmus működését, bizonnyítással együtt.

Ha tehát veszünk egy kalibrációs gömböt, és legalább három különböző pozícióban elhelyezzük, ahonnan LiDAR mérőeszközökkel és digitális kamerákkal felvételeket készítünk, akkor az eszközök közötti transzformációt meg lehet határozni. Ezzel a módszerrel tetszőleges számú digitális kamerát és LiDAR eszközt lehet egymáshoz kalibrálni.

Módszer	Kalibrációs objektum	Feltételek	Futási idő
KITTI	Legaláb négy (sík) sakktábla	**Automatikus**, de nem minden képre működik	Percek
Polygonal	Leglább három sík	Félautomatikus, pontfelhőben manuális kijelölés szükséges	–
Box Calibration (Doboz kalibráció)	Doboz három merőleges oldallal	Félautomatikus: doboz sarkait a képeken kézzel kell kijelölni	–
Proposed (Javasolt)	Legalább négy gömb	**Teljesen automatikus**	Percek

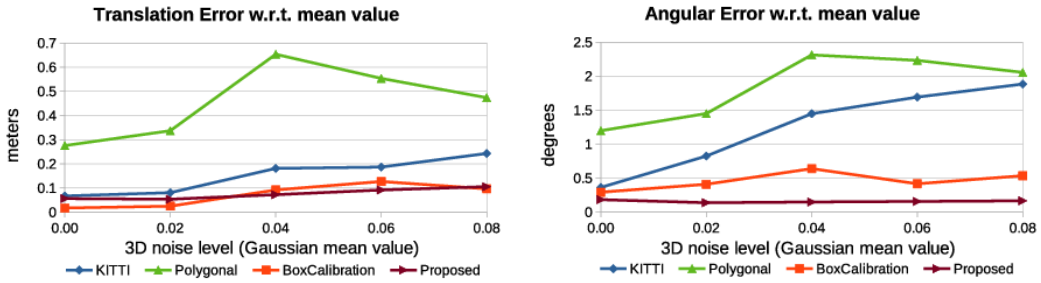
6. táblázat. *Az összehasonlított módszerek jellemzői.*

6.3.4. Tesztelési eredmények

Ez a szakasz ismerteti az összehasonlításhoz kiválasztott módszereket, valamint a szintetikus és a valós világbeli tesztsorozatok eredményeit. A Blensor szenzorszimulációs csomagban létrehozott virtuális jelenet tartalmaz egy kamerát, egy LiDAR-t és a tesztelt módszerekhez szükséges kalibrációs objektumokat. A szintetikus tesztek előnye, hogy a riválisok technikák kvantitatív módon összehasonlíthatók. A valós tesztek során a kapott külső paraméterek minőségét a LiDAR pontfelhő és a kamerakép újravetítésének vizuális összehasonlításával értékeljük.

Összehasonlított módszerek.

A gömbös kalibráción túl három másik módszert választottunk ki a javasolt módszerrel való versengésre, ezek összehasonlítását a 6. táblázat foglalja össze.



47. ábra. *A szintetikus tesztek eredményei: Eltolási (balra) és forgatási (jobbra) átlaghiba növekvő Gauss eloszlású zaj hozzáadásával.*

A KITTI kalibrációs eszköztárat (`www.cvlibs.net/software/calibration/`) Andreas Geiger és kollégái [53] tették közzé. Több sakktáblát használnak a kamerák és a LiDAR érzékelők külső kalibrálásához. A sakktáblákat egyenként detektálják a kameraképeken, és meghatározzák a kamerák külső paramétereit (is). Ezekkel a paraméterekkel a sakktábla sarkai térben is rekonstruálhatók a megfelelő sakktábla sarkok képeken való illesztésével. A sakktáblák síkjait a LiDAR pontfelhőben a síkhoz tartozó

adatpontok normálvektorának becslésével határozzák meg. Végül a kameraképekből rekonstruált sakktáblasíkok és a LiDAR-pontfelhőből detektált síkok közötti megfelelés adja meg a kamera és a LiDAR eszközök végső paramétereit. A síkok közötti síkillesztési eljárás azonban kétértelmű lehet, és néhány hamis egyezést eredményezhet.

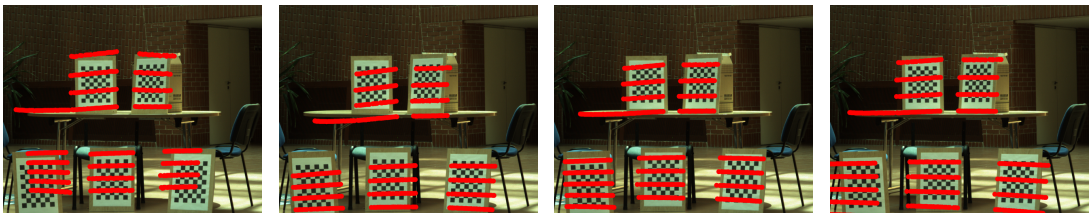
A Park és mtsai. [129] által bevezetett **Polygonal** módszer a fehér háromszög vagy rombusz alakú síkbeli célpontok észlelésén alapul. A legtöbb kalibrációs módszer a LiDAR pontfelhőben lévő objektumok széleinek megtalálásával küzd. A szerzők ezt a problémát virtuális pontok bevezetésével oldják meg, amelyek két egymást követő LiDAR pont között helyezkednek el. A virtuális pontokhoz vonalakat illesztenek, és kiszámítják a kalibrációs objektumok sarkait. Végül a sarkok vetületeit a FAST [139, 140] jellegzetes leírók segítségével választják ki, és PnP problémát oldanak meg 3D-2D megfeleltetések alapján. A módszer fő hátránya, hogy mind a síkok LiDAR-pontjainak, mind a megfelelő FAST jellegzetes pontoknak a kiválasztását kézzel kell elvégezni.

A harmadik összehasonlított módszer a 6.2. fejezetben ismertetett dobozt használó kalibrációs eljárás.

Szintetikus teszteredmény.

Valósághű kalibrációs tesztkörnyezetet hoztunk létre Blenderben, a Blensor [1] LiDAR szimulációs csomag segítségével. A tesztképek 1920×1080 felbontásúak. Összehasonlítottuk a javasolt módszert és három párhuzamos módszert az alapigazság (GT) adatokkal, miközben a LiDAR pontfelhőkhöz zajt adtunk. A jelenet egy digitális kamerát és egy Velodyne HDL-64E2 LiDAR készüléket szimulált.

A módszereket additív Gauss eloszlású zajjal teszteltük. Először a μ átlagértéket 0.00 és 0.08 méter között változtattuk, $\sigma = 0.02$ szórású Gauss-zajt használva. Ez a forgatókönyv úgy értelmezhető, hogy a tesztobjektumok eltávolodnak a LiDAR-tól. Ezután egy nulla átlagú Gauss-zajt használtunk változó σ értékkel 0.00 és 0.10 méter között. A tesztjelenetek a KITTI esetében 5 sakktáblát, a Polygonal esetében 5 poligonális táblát, a Box Calibration esetében egy dobozt, a javasolt módszerek esetében pedig 4 gömböt tartalmaztak.



(a) *Polygonal* **(b)** *KITTI* **(c)** *Box Calibration* **(d)** *Proposed*

48. ábra. *Valós körülmények között végzett tesztek eredményei. A képek a sakktáblák néhány LiDAR-pontjának vetítését mutatják a képekre a kapott külső paramétereinek felhasználásával. A vetített pontok elcsúszása mutatja a kalibrációs hibát. A vetítésre használt pontok a sakktáblákhoz és az asztal széléhez tartoznak.*

A kalibrációs módszerek hibáját az általuk becsült külső paraméterek és az alapigazság közötti különbségként mértük. Az eltolás hibáját a vektorok közötti különbség euklideszi normája adta, a szöghibát pedig a következőképpen számítottuk ki:

$$\epsilon = cos^{-1}\left(0.5\left(trace\left(\mathbf{R}_{\mathcal{GT}}^{T}\mathbf{R}\right) - 1\right)\right), \tag{84}$$

ahol $\mathbf{R}_{\mathcal{GT}}$ és $\mathbf{R}$ jelöli az alapigazság és a becsült forgatási mátrixot.

A 47. ábra mutatja a szintetikus tesztek eredményeit. A bal és jobb oldali oszlopok az eltolási és a forgatási átlaghibát mutatják. Egyértelműen megállapítható, hogy a javasolt eljárás felülmúlja a konkurens módszereket a forgatás megbecslése során, és eltolás esetében is csak a saját, dobozt használó módszer képes hasonló teljesítményre.

További szintetikus összehasonlításokat a 2020-as publikációnkban [170] olvasható, helyhiány miatt itt nem tudok további eredményeket közölni.

Valós tesztek

A valós körülmények között végzett teszteket egy Velodyne VLP-16 Lidar és egy iCube NS4133CU kamera segítségével végeztük el, 1280×1024-os felbontással, kis látószögű objektívekkel felszerelve. Négy különböző jelenetet rögzítettek az eszközök, amelyek pozíciója fix volt. Így a külső paraméterek a vizsgálat során állandóak maradtak. Minden jelenet tartalmazta a rivális módszerekhez szükséges kalibrációs objektumokat, hasonlóan a szintetikus teszthez.

A kalibrációs objektum esetünkben egy 30 cm sugarú gömb. A valós tesztek eredményei megerősítik a LiDAR adatokból történő gömbsugár becslés pontosságát, mivel az eredmények $30{,}53$ centiméteres sugarat eredményeztek, $1{,}45$ centiméteres szórással.

A rivális módszerek kvantitatív összehasonlítása nem lehetséges a valós teszt esetében, mivel az alapigazság (ground truth) nem áll rendelkezésre. Ezért az összehasonlítás vizuálisan történik, a LiDAR pontfelhő egy részének a kameraképre való visszavetítésével, amihez a módszerek becsült paramétereit használtuk fel. Ha a külső paraméterek pontosak, akkor az újravetített LiDAR pontok lefedik a megfelelő pixelt, ellenkező esetben az újravetítések eltolódnak. Az esetleges nagyobb mértékű eltolódása azt jelzi, hogy a becsült pozíciók/forgatások kevésbé pontosak.

A 48. ábra a valós világbeli tesztelés eredményeit mutatja be. A piros körök a sakktáblákat használó KITTI kalibrációs jelenet újravetítései. A Poligonális és a KITTI kalibrációs módszer eltolódása jelentősebb, mint a Doboz és a javasolt kalibrációs módszeré. Ez a pózok rosszabb minőségére utal. A dobozos kalibráció újravetítései pontosabbak, azonban a javasolt kalibráció még mindig jobban teljesít a többinél. Lényegében a javasolt módszer a valós tesztekben hasonló eredményeket ad, mint a legjobb módszerek, a teljesen automatizált eljárás hátrányai ellenére.

Időigény.

A javasolt kalibrációs csővezeték szűk keresztmetszete időigény szempontjából a RANSAC-alapú gömb- és ellipszisillesztés. A szükséges futási idő elsősorban a RANSAC algoritmus [46] iterációinak számától függ. Még a legrosszabb esetben sem több néhány percnél, ami összehasonlítható a másik teljesen automatizált módszerrel a 6. táblázat mérési eredményei alapján. A javasolt kalibrációs csővezeték többi része csak gyors algoritmusokat tartalmaz, ezek időigénye egy másodperc alatt van.

6.4. Egytengelyes kalibráció

Eddig a fejezetig általános LiDAR-kamera kalibrációs eljárásokat ismertettem, először doboz, majd gömb alkalmazásával. Amennyiben az eszközök rögzítését is ránk bízzák, lehetőségünk van olyan konfiguráció megalkotására, amely segítségével a kalibrációs probléma lényegesen egyszerűsödik. A 49. ábrán láthatjuk a javaslatunkat: egy speciális, 3D nyomtatott rögzítést készítettünk a kamerának, mellyel a kamerát

a LiDAR mérőeszköz tetejére lehet rögzíteni. A LiDAR keret teteje vízszintes, és a rendszer koordináta-rendszere úgy lett beállítva, hogy a Z tengely felfelé mutat [43]

Ezért, ha a két eszközt egymás fölé helyezzük, egyedül a Z tengely körüli forgatás az ismeretlen. A Velodyne VLP-16 eszköz esetén a ház henger alakú, és az Y és X tengelyek irányát csak sejteni lehet, pontosan meghatározni nem.

Ha a kamera és a LiDAR koordináta-rendszerében levő két pontot $\mathbf{p}_C$-vel és $\mathbf{p}_L$-el jelöljük, akkor a két tengely közötti (egybevágósági) transzformációt írhatjuk az alábbi egyenlettel: $\mathbf{p}_C = \mathbf{R}\left(\mathbf{P}_L - \mathbf{t}\right)$.

Miután a forgatás a Z tengely körül történik, az eltolás pedig ugyanazon tengely mentén, írhatjuk, hogy

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}. \tag{85}$$

Az elmúlt években kidolgoztunk egy speciális kalibrációs algoritmust, amelyet itt röviden ismertetek. A kalibráció során visszanyúlunk a szokásokhoz, és egy közönséges sakktábla segítségét vesszük igénybe. Ezt a táblát tarthatjuk kézben, de akár a falra is feltehetjük, ahogyan az az 50. ábra alsó képén látható.

Egy–két bekezdés erejéig megemlíteném, hogy az érzékelők bekalibrálása akkor lesz nagyon pontos, ha kalibrációs objektum használatát és jól kontrollált körülményeket követelünk meg. Ekkor lehet a legpontosabban meghatározni az eszközök közti eltolást és elforgatást. Sajnos a használat közben az eszközöket mozgatjuk, rázódnak, ezért a rögzítések el tudnak mozdulni. Az egyetemünk környékén is az utak állapota bőven hagy kívánnivalót, ezért a mi esetünkben az elállítódás valós veszély. Ilyenkor újrakalibrálás szüksége, ami elég költséges. Széria-érett technológia esetén szakszervizben lennének képesek ezt precízen elvégezni. Amennyiben a sakktáblát egy falra feltesszük, ahogyan azt az 50. ábra is mutatja, minden beálláskor a mi módszerünk automatikusan el tudja végezni ezt az újrakalibrálást, szakember segítsége nem szükséges.

6.4.1. Forgatás meghatározása

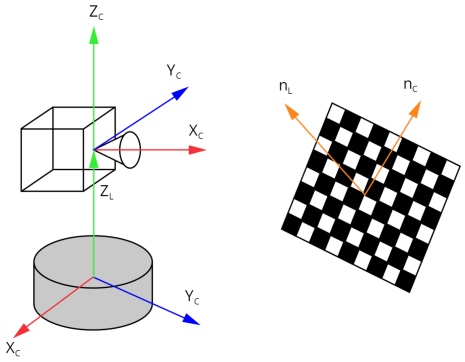
A forgatás meghatározásához a síkok normálvektorait fogjuk felhasználni. Ha a LiDAR pontfelhőre RANSAC alapú robusztus síkillesztő eljárást futtatunk, akkor a síkfelületet meg tudjuk becsülni. Ha kikötjük, hogy a jármű előtt legyen a sakktábla, az a keresési teret szűkíteni tudja.

A sakktábla esetében a Zhang-féle klasszikus kalibrálási módszert [190] alkalmazhatjuk, ekkor a sakktábla és a kamera síkja közötti homográfiából az elforgatás meghatározható. A $[0 \quad 0 \quad 1]^T$ normálvektort elforgatva megkapjuk a kamera szemszögéből a normálvektor koordinátáit.

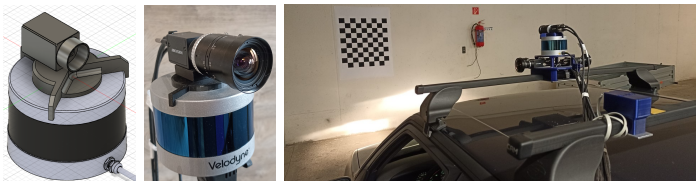
Ha $n_C = [x_c \quad y_c \quad z_c]^T$ és $n_L = [x_L \quad y_L \quad z_L]^T$ jelöli a kamera és a LiDAR rendszerében a normálvektorokat, akkor írhatjuk, hogy

$$\begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} \tag{86}$$

[43] A digitális kamerák esetében hagyományosan az Y mutat felfelé, és a Z előre, de az egyszerűség kedvéért átneveztük őket.



49. ábra. *A speciális kamera-LiDAR elrendezés sematikus ábrája. A koordinátarendszereket piros, zöld és kék színnel jelöljük. A zölddel jelölt tengelyek megegyeznek a kamerán és a LiDAR-on.*



50. ábra. *Kamera-LiDAR konfiguráció. Egy speciális, 3D nyomtatott rögzítőelem tarja a mérőeszközöket.* ***Balra:*** *A berendezés 3D CAD modellje.* ***Középső ábra:*** *A megvalósított kamera-LiDAR páros.* ***Jobb ábra:*** *Egy autó tetejére szerelve a berendezés. A mérőautó további kamerákat is tartalmaz. A falon jól látható a kalibrációhoz felhasznált sakktábla, amelyből egyetlen egy is elég a helyes működéshez.*

Az utolsó koordináta nem függ a θ szögtől, ezért a normálvektorok ellenőrzésre ugyan lehet használni, de a szögbecslésre nem.

Az első két koordinátára picit átalakítva írhatjuk, hogy

$$\begin{bmatrix} x_C \\ y_C \end{bmatrix} = \begin{bmatrix} x_L & -y_L \\ y_L & x_L \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}. \tag{87}$$

Ez pedig egy érdekes, $\mathbf{A}x = \mathbf{b}$ alakú egyenletrendszer egy megkötéssel: $\mathbf{x} = 1$, azaz $\mathbf{x}$ egységvektor. A megoldás a C függelékben olvasható.

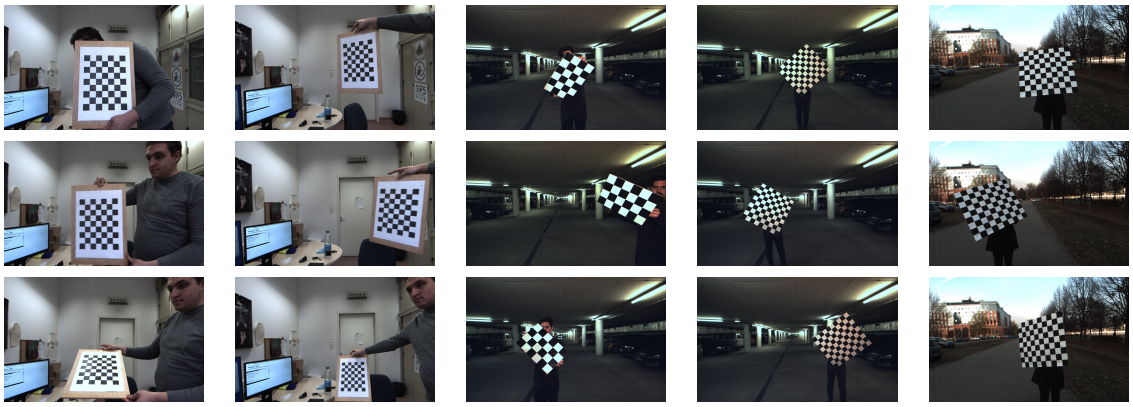
6.4.2. Eltolás

Az eltolás becslése szintén lehetséges. Mivel azonban mind a LiDAR, mind a kamera és az optika, továbbá a saját tervezésű rögzítés adatai pontosan ismert, az eltolás vektora megadható. Egy érdekesség, hogy esetünkben a 3D nyomtatott rögzítést úgy készítettük el, hogy $t_x = t_y = 0$ legyen.

6.4.3. Kísérleti eredmények

A javasolt módszer minőségi és mennyiségi tesztelését valós és virtuálisan generált képeken és pontfelhőkön végeztük el, összehasonlítva a MATLAB-ban végzett párhuzamos kalibrációs folyamattal. Ebben a részben értékeljük ki az eredményeket és megvizsgáljuk a tesztelt megközelítések pontosságát.

Valós tesztek.



Dataset #1 Dataset #2 Dataset #3 Dataset #4 Dataset #5

51. ábra. *Néhány bemeneti kép az öt különböző valós adatkészletből. Az első és második adatkészletben szereplő képek és pontfelhők egy irodában készültek, egy kis* 7×9*-es, 31 mm-es mezővel rendelkező sakktáblával. A harmadik adatkészletet az ELTE lágymányosi mélygarázsában gyűjtöttük* 60 *mm-es négyzet méretű* 4×6*-os sakktáblával. A negyedik és ötödik adatkészlethez nagy, 81 mm-es méretű,* 9×10*-es felbontású sakktáblát használtunk a garázsban és az ELTE külső parkolójában.*

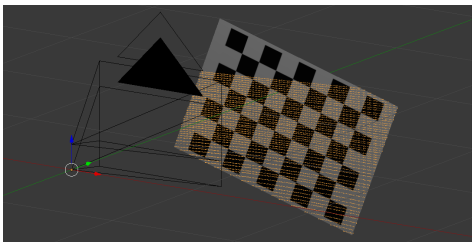
A tesztekhez egyrészről Veldoyne VLP-16 LiDAR-t, másrészről egy Hikvision MV-CA020-20GC digitális kamerát Fujinon SV-0614H lencsékkel alkalmaztunk, így gyűjtöttünk össze a pontfelhő-képpárokat. A lencsék kiváló minőségűek, nem perspektivikus torzításuk elhanyagolható. A kamerát a LiDAR tetejére szereltük a speciális, saját gyártású 3D nyomtatott tartók segítségével. Az adatokat különböző beállítások mellett vettük fel, összesen öt adatsort gyűjtöttünk különböző helyszíneken: irodában, garázsban és egy parkolóban három különböző sakktáblával. A sakktábla méretek 4×5-tól 9×10-ig terjednek. Néhány bemeneti kép az 51. ábrán tekinthető meg.
Forgatási szög pontossága.

Az első kísérlet a szögbecslés pontosságát vizsgálja. Minden adatsoron a kalibrációt egyenként végeztük el minden egymásnak megfelelő kép – pontfelhő páron.

Kvantitatív értékeléseinket a becsült szögek szórása alapján végeztük. Az eredményeket a 7. táblázat tartalmazza, ahol az eredmények pontosságának meghatározását mutatjuk meg. Az alacsonyabb szórás pontosabb becslést jelent. A legjobb eredményeket minden forgatókönyvben félkövér számokkal emeltük ki. Az adatokból kiszűrtük a kiugró értékeket.

	Egyenként	***Egyenként – szűrve***	***10 kép***
Dataset #1	12.1753	2.3022	**0.9759**
Dataset #2	27.2212	2.3786	**1.0583**
Dataset #3	38.5715	**0.6895**	1.0392
Dataset #4	0.9453	-	**0.4899**
Dataset #5	28.9916	**1.2363**	4.0653

7. táblázat. *A táblázat tartalmazza a különböző adatsorokra számított forgatási szögeknek a szórását.* ***Egyenként****: Csak egyetlen képet használunk a kalibráláshoz.* ***Egyenként – szűrve****: Egy képet használunk, de a kiugró értékeket kiszűrjük;* ***10 kép****: Tíz véletlenszerűen kiválasztott képre végzett túlhatározott becslés.*



52. ábra. *Virtuális LiDAR és kamera sakktáblával a Blensorban.*

Forgatási szög		
Alapigazság	**Becsült érték**	**Becslési hiba**
$0°$	0.0000°	0.0000°
45°	44.9032°	0.0976°
90°	89.6333°	0.3660°
135°	134.9589°	0.0410°
180°	179.9999°	2.05°

8. táblázat. *A javasolt módszer kalibrációs hibája virtuálisan generált adatokon.*

Tesztek szimulátorral.

A Blensor [44] eszközt használtuk a virtuális tesztek elvégzése. A virtuális kamera és a LiDAR beállítása az 52. ábrán látható. Minden ismert elforgatási szögnél ki tudjuk értékelni a becsült értékeket, mivel a szimulátorból ismert az alapigazság (GT).

Az eredmények a 8. táblázatban láthatóak. Öt különböző beállítás eredményeit hasonlítjuk össze, az alapigazság szögei $0°$ és $180°$ között vannak. A közelített forgás átlagos hibája a javasolt kalibrációs módszerrel mindössze $0{,}1009°$. Ez arra utal, hogy a javasolt kalibrációs módszerrel kiváló minőségű szögbecslés végezhető.

6.5. Összefoglalás

Az utolsó, IV. téziscsoportban megmutattam, hogyan lehet digitális kamerákat és LiDAR letapogatókat egymáshoz kalibrálni. Kalibráció alatt a külső paramétereknek a meghatározását értem, azaz annak az egybevágósági transzformációnak a becslését, amely a LiDAR koordináta-rendszeréből a méréseket átviszi a kamera rendszerébe. Ez a gyakorlatban három elforgatási és három eltolási paraméter becslését jelenti.

A dolgozatban egy félautomatikus és két teljesen automatikus módszert mutattam be, melyeket az elmúlt években publikáltunk kollégáimmal közösen:

1. Az első eljárás dobozokat alkalmazott. A módszer előnye, hogy ilyen dobozokat könnyen találunk a környezetünkben. A módszer félautomatikus, a pontfelhőben meg kell jelölni azt a térfogatrészt, amire a dobozillesztés –immár automatikusan – az algoritmus elvégezni. A képen a dobozok sarkait szintén manuálisan kell kijelölni. A kalibrációs paramétereket PnP algoritmus segítségével számíthatjuk.

[44] A Blensor egy nyílt forráskódú szimulációs csomag LIDAR és Kinect szenzorokhoz, amely együttműködik a Blender számítógépes látóeszközzel. További részleteket a www.blensor.org oldalon találhatóak.

2. A második eljárás gömbfelületeket használ a kalibrációhoz. A gömb kontúrpontjai a képen ellipszis formálnak, ezért automatikusan detektálni lehetséges, majd az ellipszisből – kalibrált kamerák esetében – ismert sugár esetén a gömb középpontját meg lehet határozni. A LiDAR pontfelhőn hagyományosnak tekinthető felületillesztéssel lehet kiszámítani a gömb középpontját. Ha legalább négy gömböt ismerünk a kamera és a LiDAR rendszerében is, akkor pontregisztrációs eljárással kiszámíthatjuk az egybevágósági transzformáció hat szabad paraméterét.

3. A javasolt harmadik módszer esetén egy speciális, saját tervezésű rögzítés miatt a hat szabadságfok kettőre csökkenthető, hiszen a függőleges tengelyek egybeesnek a kamerán és a LiDAR letapogatón. Sakktábla alkalmazásával lehetséges a kalibráció elvégzése, a sakktábla felületének normálvektora adja a fő segítséget, hiszen azt mind a kameraképből, mind a LiDAR pontfelhőből automatikusan ki lehet nyerni. Az egyszerűsítés miatt a kalibrációs probléma matematikailag könnyen kezelhető, lineáris algoritmus segítségével lehet legkisebb négyzetes értelemben optimális megoldást adni.

A gyakorlati vizsgáltok megmutatták, hogy a módszerek valós esetben is jól működnek, és alkalmasak arra, hogy LiDAR-kamera felvételeket egyesítsünk.

A fejezetek alapján az alábbi téziseket mondom ki:

IV.1. tézis. Kamera-LiDAR kalibráció téglatest (doboz) segítségével [135, 134].

Pusztai Zoltán doktoranduszommal közösen kidolgoztunk egy újszerű eljárást LiDAR letapogató eszköz és digitális kamera kalibrációjára, amely téglatestet (dobozt) alkalmaz kalibrációs objektumként, szemben a meglévő módszerekkel [53, 172], melyek a legtöbb esetben sakktábla segítségét veszik igénybe a kalibrációs feladat megoldásra.

A LiDAR pontfelhő feldolgozására javasoltunk egy félautomatikus eljárást, amely a merőlegesség figyelembe vételével meghatározza a doboz csúcspontját és az éleit. Az eljárás először robusztus illesztéssel merőleges oldalakat keres, majd iteratív eljárással beforgatja és betolja a látható oldalakat a megfelelő pozícióba. Ismert dobozméret esetében így a doboz csúcsainak pozíciója egyértelműen megbecsülhető.

A kamera képén a doboz sarkait manuálisan kell meghatározni. A pontfelhőn becsült és a képen meghatározott pontok között egy Perspektív n-pont [95] (PnP) algoritmus adja meg a két eszköz közötti relatív eltolást és elfogatást.

Az eljárás könnyen kiterjeszthető tetszőleges számú kamera és LiDAR letapogató eszköz alkalmazására.

IV.2. tézis. Kamera-LiDAR kalibráció gömb segítségével [170, 166].

Tóth Tekla doktoranduszom közreműködésével kidolgoztunk egy új módszert, amely gömb segítségével határozza meg digitális kamerák és LiDAR letapogató eszközök elhelyezkedését. Az eljárásunk teljesen automatikus.

Egy kamera képén egy gömb kontúrpontjai speciális ellipszist alkotnak, az ellipszis paramétereit meghatároztuk a gömb pozíciójának és sugarának függvényében. Megmutattuk, hogy a detektált ellipszisből hogyan lehet meghatározni a gömb középpontját a képen. Ismert sugár és kamera belső paraméterek esetében a gömb térbeli pozíciója is meghatározható.

A LiDAR pontfelhőn robusztus illesztés segítségével a középpont és a sugár szintén meghatározható, erre a feladatra a területen szabványosnak mondható eljárások robusztifikálását javasoljuk.

A kamerák és LiDAR-ok közötti elforgatás és eltolás megoldására pontregisztrációs eljárást alkalmazunk. Módszerünk tetszőleges számú kamerára és LiDAR eszközre működik, ha legalább négy gömbre el tudjuk a szükséges illesztési feladatokat végezni. Túlhatározott esetben, azaz négynél több felvétel esetén is működik az eljárásunk, a gömbök számának növelésével a kalibráció pontossága növekszik.

IV.3. tézis. Egy forgatási paraméterre redukált kamera-LiDAR kalibráció [163].

Tófalvi Tamás és Tóth Tekla doktorandusz hallgatókkal közösen egy harmadik eljárást is kidolgoztunk LiDAR-kamera kalibrációra, amely sakktáblát használ a külső paraméterek kiszámítására. A fő újdonság, hogy a kamerát és a LiDAR-t egy speciális, saját tervezésű és gyártású 3D-nyomtatott alkatrész segítségével egymáshoz rögzítjük [45], így a két eszköz relatív orientációja egyszerűsödik, mindössze egy forgatási szöget kell meghatározni.

Az általunk javasolt módszer sakktáblák síkjait vizsgálja, és ezek normálvektorait használja a helyes forgatási szög megbecslésére. A javasolt megoldó lineáris problémaként írható fel, a forgatásra így legkisebb négyzetes értelemben optimális megoldást tudunk adni, amelyik minimális és túlhatározott esetekben egyaránt működik.

A módszer egy képet igényli csak, ezért online kalibrálásra is használható, ha a jármű elhalad egy falra (síkra) rögzített sakktáblaminta mellett.

[45] A rögzítés tervezése és 3D nyomtatása Kovács Bandó mérnökünk munkája.

7. Összegzés

Ez a disszertáció az elmúlt tizenöt évben elért tudományos eredményeimet próbálta nagyon röviden összefoglalni, amellyel az MTA doktora címre pályázom. A másfél évtizednyi munkámat négy csoportba rendeztem.

Az első két téziscsoportban a lokális affin transzformációk alkalmazásával foglalkoztam. Az első csoport az elméleti eredményekkel foglalkozik, három törvényt mondtam ki a transzformációk és a sztereó látás kapcsolatáról. A második téziscsoport a leggazdagabb: kutatócsoportom doktori hallgatói és munkatársai segítségével az elméleti eredményekből származtatott becslő eljárásokat fejlesztettünk ki, ezeket ismertettem itt. Talán egy értekezésben szabad annyi személyes benyomást közölni, hogy bár nem ez a legnépszerűbb kutatási területem, ahol a legtöbb hivatkozást kapták a munkáink, de hozzám mégis ezek a problémák állnak legközelebb. Egyrészt, mert ez egy igazi "magyaros" terület, a legtöbb eredmény hazánkhoz köthető kutatók érték el, másrészt nagyon sok helyen optimális megoldást sikerült a megcélzott problémákra találni, ezért a megoldások pontosságát elméletileg is sikerült megalapozni.

A harmadik téziscsoport egy speciális numerikus algoritmus, a kötegelt behangolás alkalmazását mutatja be két újabb rekonstrukciós problémaosztályra. A két osztályban közös, hogy felületre merőleges irányokat állítanak elő. Az egyik esetben a megvilágítás változtatásával lehetséges új felületi információt kinyerni a képekből, a második ismertetett módszer esetében az affin transzformációkat használtuk ugyanerre a célra.

Az utolsó, negyedik téziscsoport LiDAR érzékelők és digitális kamerák külső paramétereinek kalibrációjával foglalkozik. A szakirodalomban szinte egyeduralkodó, sakktáblás kalibrációval szemben két másik geometriai primitívnek, a téglatestnek és a gömbnek felhasználhatóságára mutattam be két kalibrációs eljárást a disszertációban, melyekkel LiDARok és kamerák kalibrálhatóak egymáshoz. Érdekesség, hogy erre a területre érkezett a legtöbb hivatkozás, ami nemcsak azt mutatja, hogy egy igen népszerű és magas gyakorlati értékű területről van szó, hanem azt is, hogy jó időben sikerült a terület kutatásaiba bekapcsolódni.

A tézisekhez kapcsolódó publikációk jegyzéke a tézisfüzet végén olvasható. Kilenc folyóiratcikk és további huszonegy nemzetközi fórumokon jegyzett konferenciapublikáció kapcsolódik a megfogalmazott eredményekhez. Számomra mindez azt jelenti, hogy az elért eredményeket a számítógépes látás szakmai közössége értékesnek tartja, és ezért befogadja.

Amiről nem szól, nem szólhat ez a tézisfüzet, az a jövő. Munkám során megtapasztalhattam, hogy egy terület művelése szinte minden esetben újabb irányokat nyit meg, ezért biztos vagyok benne, hogy a következő időszakban is érdekes problémákkal találom/találjuk szembe magunkat, és remélem, hogy sikerül még értékes elméleteket és algoritmusokat hozzáadni a háromdimenziós számítógépes látás világához.

8. Irodalomjegyzék

Hivatkozások

[1] Blender sensor simulation. http://www.blensor.org.

[2] *Számítógépes látás*. Typotex Kiadó, Budapest, 2011 2011.

[3] LGSVL Simulator, 2019. https://www.lgsvlsimulator.com/.

[4] Anubhav Agarwal, C.V. Jawahar, and P.J. Narayanan. A Survey of Planar Homography Estimation Techniques. Technical report, IIT-Hyderabad, 2005.

[5] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, 2011.

[6] Hatem Alismail, L Douglas Baker, and Brett Browning. Automatic calibration of a range sensor and camera system. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 286–292. IEEE, 2012.

[7] Vasileios Argyriou, Maria Petrou, and Svetlana Barsky. Photometric stereo with an arbitrary number of illuminants. *Computer Vision and Image Understanding*, 114(8):887–900, 2010.

[8] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *PAMI*, 9(5):698–700, 1987.

[9] D. Barath, J. Molnar, and **L. Hajder**. Novel methods for estimating surface normals from affine transformations. In *Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 316–337. Springer International Publishing, 2016.

[10] Daniel Barath, Ivan Eichhardt, and **Levente Hajder**. Optimal multi-view surface normal estimation using affine correspondences. *IEEE Trans. Image Process.*, 28(7):3301–3311, 2019.

[11] Daniel Barath and Jiri Matas. Multi-class model fitting by energy minimization and mode-seeking. In *Proceedings of the European Conference on Computer Vision (European Conference on Computer Vision)*, pages 221–236, 2018.

[12] Daniel Barath and Jiri Matas. Progressive-x: Efficient, anytime, multi-model fitting algorithm. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3780–3788, 2019.

[13] Daniel Barath and Jiri Matas. Graph-cut RANSAC: local optimization on spatially coherent structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):4961–4974, 2022.

[14] Daniel Barath, Dmytro Mishkin, Ivan Eichhardt, Ilia Shipachev, and Jiri Matas. Efficient initial pose-graph generation for global sfm. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[15] Daniel Barath, Jozsef Molnar, and **Levente Hajder**. Optimal Surface Normal from Affine Transformation. In *VISAPP 2015*, pages 305–316, 2015.

[16] Daniel Barath and **Levente Hajder**. Novel ways to estimate homography from local affine transformations. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: VISAPP, (VISIGRAPP 2016)*, pages 432–443. INSTICC, SciTePress, 2016.

[17] Daniel Barath and **Levente Hajder**. A theory of point-wise homography estimation. *Pattern Recognition Letters*, 94:7–14, 2017.

[18] Daniel Barath and **Levente Hajder**. Efficient recovery of essential matrix from two affine correspondences. *IEEE Transactions on Image Processing*, 27(11):5328–5337, 2018.

[19] Daniel Barath, Tekla Toth, and **Levente Hajder**. A minimal solution for two-view focal-length estimation using two affine correspondences. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2557–2565, 2017.

[20] Svetlana Barsky and Maria Petrou. Design issues for a colour photometric stereo system. *Journal of Mathematical Imaging and Vision*, 24:143–162, 2006.

[21] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[22] Peter N. Belhumeur, David J. Kriegman, and Alan L. Yuille. The bas-relief ambiguity. *International Journal of Computer Vision*, 35:1040–1046, 1999.

[23] Jacob Bentolila and Joseph M. Francos. Conic epipolar constraints from affine correspondences. *Comput. Vis. Image Underst.*, 122:105–114, 2014.

[24] Åke Björck. *Numerical Methods for Least Squares Problems*. Siam, 1996.

[25] J-L. Blanco-Claraco, F-Á Moreno-Dueñas, and J. G. Jiménez. The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. *I. J. Robotics Res.*, 33(2):207–214, 2014.

[26] András Bódis-Szomorú, Hayko Riemenschneider, and Luc Van Gool. Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[27] A. M. Buchanan and A. W. Fitzgibbon. Damped Newton Algorithms for Matrix Factorization with Missing Data. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 316–322, 2005.

[28] M. Bujnak, Z. Kukelova, and T. Pajdla. Robust focal length estimation by voting in multi-view scene reconstruction. In *Asian Conference on Computer Vision*, pages 13–24. Springer, 2009.

[29] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, April 1992.

[30] Hyunggi Cho, Young-Woo Seo, BVK Vijaya Kumar, and Ragunathan Raj Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1836–1843. IEEE, 2014.

[31] Sunglok Choi and Jong-Hwan Kim. Fast and reliable minimal relative pose estimation under planar motion. *Image Vis. Comput.*, 69:103–112, 2018.

[32] Per H. Christensen and Linda G. Shapiro. Three-dimensional shape from color photometric stereo. *International Journal of Computer Vision*, 13:213–227, 1994.

[33] V. Codreanu, F. Dong, B. Liu, J. Roerdink, D. Williams, P. Yang, and B. Yasar. GPU-ASIFT: A fast fully affine-invariant feature extraction algorithm. In *2013 International Conference on High Performance Computing & Simulation*, pages 474–481. IEEE, 2013.

[34] Jiri Matas Daniel Barath and **Levente Hajder**. Accurate closed-form estimation of local affine transformations consistent with the epipolar geometry. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 11.1–11.12. BMVA Press, 2016.

[35] Jiri Matas Daniel Barath and **Levente Hajder**. Multi-h: Efficient recovery of tangent planes in stereo images. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 13.1–13.13. BMVA Press, 2016.

[36] David Eberly. Rotation Representations and Performance Issues. January 2002, Magic Software, Inc. (Online; accessed 21 June 2017).

[37] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 369–378, 1997.

[38] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Toward geometric deep SLAM. *CoRR*, 2017.

[39] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 224–236, 2018.

[40] Ondrej Drbohlav and Mike Chantler. Can two specular pixels calibrate photometric stereo? In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1850–1857, 2005.

[41] David J. Eck. Introduction to computer graphics. online book, 2024. Version 1.4, accessed at February, 2024.

[42] Ivan Eichhardt and **Levente Hajder**. Improvement of camera calibration using surface normals. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3745–3750, 2016.

[43] M. Elgendy. *Deep Learning for Vision Systems*. Manning Publications, 2020.

[44] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: large-scale direct monocular SLAM. In *European Conference on Computer Vision*, 2014.

[45] Olivier Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. Technical Report RR-0856, INRIA, 1988.

[46] M. Fischler and R. Bolles. RANdom SAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 24:358–367, 1981.

[47] A.W. Fitzgibbon, M. Pilu, and R.B. Fisher. Direct Least Square Fitting of Ellipses. *IEEE Trans. on PAMI*, 21(5):476–480, 1999.

[48] Bálint Fodor, Csaba Kazó, Jankó Zsolt, and **Levente Hajder**. Normal map recovery using bundle adjustment. *IET Computer Vision*, 8:66 – 75, 2014.

[49] David F. Fouhey. Multi-model Estimation in the Presence of Outliers. Bachelor's thesis, Middlebury College, 2011.

[50] Robert Frohlich, Zoltan Kato, Alain Trémeau, Levente Tamas, Shadi Shabo, and Yona Waksman. Region based fusion of 3d and 2d visual data for cultural heritage objects. In *23rd International Conference on Pattern Recognition, ICPR 2016, Cancún, Mexico, December 4-8, 2016*, pages 2404–2409, 2016.

[51] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[52] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[53] Andreas Geiger, Frank Moosmann, Omer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. In *IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA*, pages 3936–3943. IEEE, 2012.

[54] Christopher Geyer and Konstantinos Daniilidis. A unifying theory for central panoramic systems and practical applications. In David Vernon, editor, *Computer Vision - ECCV 2000, 6th European Conference on Computer Vision, Dublin, Ireland, June 26 - July 1, 2000, Proceedings, Part II*, volume 1843 of *Lecture Notes in Computer Science*, pages 445–461. Springer, 2000.

[55] Xiaojin Gong, Ying Lin, and Jilin Liu. 3d lidar-camera extrinsic calibration using an arbitrary trihedron. *Sensors*, 13(2), 2013.

[56] Levente Hajder, Lajos Lóczi, and Daniel Barath. Fast globally optimal surface normal from an affine correspondence. In *International Conference on Computer Vision*, pages 3367–3378, 2023.

[57] Ernest L. Hall. *Computer image processing and recognition*. Computer science and applied mathematics. New York: Academic Press, 1979.

[58] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[59] R. Hartley and H. Li. An efficient hidden variable approach to minimal-case camera motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 34(12):2303–2314, 2012.

[60] R. I. Hartley. In defence of the 8-point algorithm. *International Conference on Computer Vision*, pages 1064–1070, 1995.

[61] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding: CVIU*, 68(2):146–157, 1997.

[62] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[63] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision (2nd edition)*. Cambridge University Press, 2004.

[64] M. Hassaballah and A.I. Awad. *Deep Learning in Computer Vision: Principles and Applications*. Digital Imaging and Computer Vision. CRC Press, 2020.

[65] Jared Heinly, Johannes L. Schönberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world* in six days. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[66] C. Hernández, G. Vogiatzis, G. J. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *Proc. of the 11th IEEE Intl. Conf. on Comp. Vision (ICCV)*, 2007.

[67] Carlos Hernández, George Vogiatzis, and Roberto Cipolla. Multi-view photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:548–554, 2007.

[68] Carlos Hernandez, George Vogiatzis, and Roberto Cipolla. Overcoming shadows in 3-source photometric stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33:419–426, 2011.

[69] B.K.P. Horn. Closed-form Solution of Absolute Orientation using Unit Quaternions. *Journal of the Optical Society of America*, 4:629–642, 1987.

[70] B.K.P. Horn, H.M. Hilden, and S. Negahdaripourt. Closed-form Solution of Absolute Orientation Using Orthonormal Matrices. *Journal of the Optical Society of America*, 5(7):1127–1135, 1988.

[71] Osamu Ikeda and Ye Duan. Color photometric stereo for albedo and shape reconstruction. In *Proceedings of the 2008 IEEE Workshop on Applications of Computer Vision*, 2008.

[72] Hossam Isack and Yuri Boykov. Energy-based geometric multi-model fitting. *International journal of computer vision*, 97(2):123–147, 2012.

[73] David W. Jacobs. Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–212, 1997.

[74] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[75] Zsolt Janko. *Photorealistic 3D Models of Real-World Objects*. PhD thesis, Eotvos Lorand University, 2006.

[76] Zsolt Jankó, Amaël Delaunoy, and Emmanuel Prados. Colour dynamic photometric stereo for textured surfaces. In *Proceedings of the 10th Asian conference on Computer vision - Volume Part II*, ACCV'10, pages 55–66, 2011.

[77] C. Julia, F. Lumbreras, and A.D. Sappa. A factorization-based approach to photometric stereo. *International Journal of Image and Graphics*, 21(1):115–119, 2011.

[78] Kenichi Kanatani. Optimal homography computation with a reliability measure. In *Proceedings of IAPR Workshop on Machine Vision Applications, MVA*, pages 426–429, 1998.

[79] Juho Kannala, Mikko Salo, and Janne Heikkilä. Algorithms for computing a planar homography from conics in correspondence. In *Proceedings of the British Machine Vision Conference*, 2006.

[80] Firas Kastantin and Levente Hajder. Visual tilt correction for vehicle-mounted cameras. In Giovanni Maria Farinella, Petia Radeva, and Kadi Bouatouch, editors, *VISIGRAPP*, pages 863–870. SCITEPRESS, 2022.

[81] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Eurographics*, pages 61–70, 2006.

[82] Michael Kazhdan and Hugues Hoppe. Screened Poisson surface reconstruction. *ACM Transactions on Graphics*, page 29, 2013.

[83] Alain Klüng and Felix Meli. Self calibration method for 3d roundness of spheres using an ultraprecision coordinate measuring machine. In *Proceedings of the 5th EUSPEN International Conference*, pages 193–196, 2005.

[84] Kevin Köser. *Geometric estimation with local affine frames and free-form surfaces*. PhD thesis, University of Kiel, 2009.

[85] Erwin Kreyszig. *Differential geometry*. Dover Publications, 1991.

[86] Stefan Kruger and Andrew Calway. Image registration using multiresolution frequency domain correlation. In *Proceedings of the British Machine Vision Conference*, 1998.

[87] Z. Kukelova, T. Pajdla, and M. Bujnak. *Algebraic methods in computer vision*. PhD thesis, Center for Machine Perception, Czech Technical University, Prague, Czech republic, 2012.

[88] M. Pawan Kumar, Saurabh Goyal, Sujit Kuthirummal, C. V. Jawahar, and P. J. Narayanan. Discrete contours in multiple views: approximation and recognition. *Image and Vision Computing*, 22(14):1229–1239, 2004.

[89] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 307–314 vol.1, 1999.

[90] Szirmay-Kalos L., Antal Gy., and Csonka Ferenc. *Háromdimenziós grafika, animáció és játékfejlesztés*. ComputerBooks, Budapest, 2003.

[91] L. Hajder and Á. Pernek and Cs. Kazó. Weak-Perspective Structure from Motion by Fast Alternation. *The Visual Computer*, 27(5):387–399, 2011.

[92] A. Laurentini. The Visual Hull Concept for Silhouette-Based Image Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(2):150–162, 1994.

[93] Simon Lee and Michael Brady. Integrating stereo and photometric stereo to monitor the development of glaucoma. *Image and Vision Computing*, 9:39–44, March 1991.

[94] V. Lepetit, F.Moreno-Noguer, and P.Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal Computer Vision*, 81(2), 2009.

[95] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal Of Computer Vision*, 81:155–166, 2009.

[96] K. Levenberg. A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[97] Jesse Levinson and Sebastian Thrun. Automatic online calibration of cameras and lasers. In *Robotics: Science and Systems*, 2013.

[98] H. Li. A simple solution to the six-point two-view focal-length problem. In *IEEE Proceedings of the European Conference on Computer Vision*, 2006.

[99] Hongdong Li and Richard Hartley. A non-iterative method for correcting lens distortion from nine-point correspondences. In *In Proc. OmniVision'05, ICCV-workshop*, 2005.

[100] Zhen Liu, Qun Wu, Suining Wu, and Xiao Pan. Flexible and accurate camera calibration using grid spherical images. *Optics Express*, 25(13):15269–15285, 2017.

[101] H.C. Longuet-Higgins. A Computer Algorithm for Reconstructing a Scene from Two Projections. *Nature*, 293:133–135, September 1981.

[102] M. Lourakis and A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the Levenberg–Marquardt algorithm. *Institute of Computer Science - FORTH, Tech. Rep. FORTH-ICS TR-340-2004*, 2004.

[103] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, ICCV '99, pages 1150–1157, 1999.

[104] Quang-Tuan Luong and Olivier D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *Int. Journal of Computer Vision*, 17(1):43–75, 1996.

[105] Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel A. Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual-inertial localization revisited. *IJRR*, 39(9), 2020.

[106] Szirmay-Kalos László. *Számítógépes grafika*. Computerbooks, 1999.

[107] Ezio Malis and Manuel Vargas. Deeper understanding of the homography decomposition for vision-based control. Technical Report RR-6303, INRIA, 2007.

[108] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11:431–441, 1963.

[109] Z. Megyesi, G. Kos, and D. Chetverikov. Dense 3D Reconstruction from Images by Normal Aided Matching. *Machine Graphics & Vision*, 15:3–28, 2006.

[110] Nghia Minh and **Levente Hajder**. Affine transformation from fundamental matrix and two directions. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP,*, pages 819–826. INSTICC, SciTePress, 2020.

[111] Faraz M Mirzaei, Dimitrios G Kottas, and Stergios I Roumeliotis. 3d lidar–camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization. *The International Journal of Robotics Research*, 31(4):452–467, 2012.

[112] Peyman Moghadam, Wijerupage Sardha Wijesoma, and Dong Jun Feng. Improving path planning and mapping based on stereo vision and lidar. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 384–389. IEEE, 2008.

[113] József Molnár and Dmitry Chetverikov. Quadratic transformation for planar mapping of implicit surfaces. *J. Math. Imaging Vis.*, 48(1):176–184, 2014.

[114] József Molnár, Rui Huang, and Zoltan Kato. 3d reconstruction of planar surface patches: A direct solution. *ACCV Big Data in 3D Vision Workshop*, 2014.

[115] Jean-Michel Morel and Guoshen Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[116] Pawan Kumar Mudigonda, Pawan Kumar, Mudigonda C. V. Jawahar, and P. J. Narayanan. Geometric structure computation from conics. In *In ICVGIP*, pages 9–14, 2004.

[117] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5), 2015.

[118] V. Murino, U. Castellani, A. Etrari, and A. Fusiello. Registration of very time-distant aerial images. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume III, pages 989–992. IEEE Signal Processing Society, 2002.

[119] Przemyslaw Musialski, Peter Wonka, Daniel G. Aliaga, Michael Wimmer, Luc van Gool, and Werner Purgathofer. A survey of urban reconstruction. In *EUROGRAPHICS 2012 State of the Art Reports*, pages 1–28, 2012.

[120] Diego Nehab, Szymon Rusinkiewicz, James Davis, and Ravi Ramamoorthi. Efficiently combining positions and normals for precise 3D geometry. *ACM Trans. Graph.*, 24(3):536–543, 2005.

[121] D. Nister. An efficient solution to the five-point relative pose problem. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:195, 2003.

[122] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004.

[123] David Nistér, Oleg Naroditsky, and James R. Bergen. Visual odometry. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2004.

[124] David Nistér, Oleg Naroditsky, and James R. Bergen. Visual odometry for ground vehicle applications. *J. Field Robotics*, 23(1), 2006.

[125] Rózsa P. *Lineáis algebra és alkalmazásai*. Nemzeti Tankönyvkiadó, Budapest, 1991.

[126] Gaurav Pandey, James McBride, Silvio Savarese, and Ryan Eustice. Extrinsic calibration of a 3d laser scanner and an omnidirectional camera. In *7th IFAC Symposium on Intelligent Autonomous Vehicles*, volume 7(1), Leece, Italy, September 2010.

[127] Gaurav Pandey, James R. McBride, Silvio Savarese, and Ryan M. Eustice. Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 2053–2059, Toronto, Canada, July 2012.

[128] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. Meshloc: Mesh-based visual localization. In *European Conference on Computer Vision*, 2022.

[129] Yoonsu Park, Seokmin Yun, Chee Sun Won, Kyungeun Cho, Kyhyun Um, and Sungdae Sim. Calibration between color camera and 3d lidar instruments with a polygonal planar board. *Sensors*, 14(3):5333–5353, 2014.

[130] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2016.

[131] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. *Int. Journal Comput. Vision*, 78(2-3):143–167, 2008.

[132] Cristiano Premebida, Oswaldo Ludwig, and Urbano Nunes. Lidar and vision-based pedestrian detection system. *Journal of Field Robotics*, 26(9):696–711, 2009.

[133] Cristiano Premebida, Gonçalo Monteiro, Urbano Nunes, and Paulo Peixoto. A lidar and vision-based approach for pedestrian and vehicle detection and tracking. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 1044–1049. IEEE, 2007.

[134] Zoltan Pusztai and **Levente Hajder**. Accurate calibration of lidar-camera systems using ordinary boxes. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 394–402, 2017.

[135] Zoltán Pusztai, Iván Eichhardt, and **Levente Hajder**. Accurate calibration of multi-lidar-multi-camera systems. *Sensors*, 18(7), 2018.

[136] C. Raposo and J. P. Barreto. Theory and practice of structure-from-motion using affine correspondences. In *IEEE Proceedings on Computer Vision and Pattern Recognition*, 2016.

[137] Carolina Raposo and João Pedro Barreto. Accurate reconstruction of oriented 3d points using affine correspondences. In *ECCV*, volume 12373, pages 545–560. Springer, 2020.

[138] S.A. Rodriguez F, V. Fremont, and P. Bonnifait. Extrinsic calibration between a multi-layer lidar and a camera.

[139] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 430–443. Springer-Verlag, 2006.

[140] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):105–119, January 2010.

[141] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, NY, 1987.

[142] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. ORB: an efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2564–2571, 2011.

[143] T. Ruijl, J. Franse, and J. van Eijk. Ultra precision cmm aiming for the ultimate concept. In *Proceedings of the 2nd EUSPEN International Conference*, pages 234–237, 2001.

[144] Joaquim Salvi, Jordi Pages, and Joan Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37:827–849, 2004.

[145] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 4937–4946, 2020.

[146] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *European Conference on Computer Vision*, 2012.

[147] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomás Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[148] D. Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International Journal of Computer Vision*, 95(1):74–85, 2011.

[149] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *2006 IEEE International Conference on Computer Vision Systems, January 5-7, 2006, St. Johns University, Manhattan, New York City, New York, NY, USA, Proceedings, CDROM*, page 45. IEEE Computer Society, 2006.

[150] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[151] Agnes Seiler, David Großmann, and Bert Jüttler. Spline surface fitting using normal data and norm-like functions. *Comput. Aided Geom. Des.*, 64:37–49, 2018.

[152] J.G. Semple and G.T. Kneebone. *Algebraic Projective Geometry*. Oxford University Press, 1952.

[153] L. Shapira, S. Avidan, and A. Shamir. Mode-detection via median-shift. In *IEEE Proceedings of the International Conference on Computer Vision*, 2009.

[154] C.C. Slama. *Manual of Photogrammetry*. Book, 1980.

[155] Laurent Smadja, Jérôme Ninot, and Thomas Gavrilovic. Road extraction and environment interpretation from lidar sensors. *IAPRS*, 38:281–286, 2010.

[156] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80:189–210, November 2008.

[157] H. Stewenius, C. Engels, and D. Nister. Recent developments on direct relative orientation. In *International Society for Photogrammetry and Remote Sensing*, pages 60:284–294, 2006.

[158] Attila Tanács, Andras Majdik, **Levente Hajder**, József Molnár, Zsolt Sánta, and Zoltan Kato. Collaborative mobile 3d reconstruction of urban scenes. In *Computer Vision - ACCV 2014 Workshops - Singapore, Singapore, November 1-2, 2014, Revised Selected Papers, Part III*, pages 486–501, 2014.

[159] **Levente Hajder** and Daniel Barath. Least-squares optimal relative planar motion for vehicle-mounted cameras. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8644–8650, 2020.

[160] **Levente Hajder** and Daniel Barath. Relative planar motion for vehicle-mounted cameras from a single affine correspondence. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8651–8657, 2020.

[161] **Levente Hajder** and Ivan Eichhardt. Computer vision meets geometric modeling: Multi-view reconstruction of surface points and normals using affine correspondences. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2427–2435, 2017.

[162] **Levente Hajder**, Tekla Tóth, and Zoltán Pusztai. Automatic estimation of sphere centers from images of calibrated cameras. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP,*, pages 490–497. INSTICC, SciTePress, 2020.

[163] Tamás Tófalvi, Bandó Kovács, **Levente Hajder**, and Tekla Tóth. Lidar-camera calibration in an uniaxial 1-dof sensor system. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2022, Volume 4: VISAPP, Online Streaming, February 6-8, 2022*, pages 730–738. SCITEPRESS, 2022.

[164] Tomasi, C. and Kanade, T. Shape and Motion from Image Streams under orthography: A factorization approach. *IJCV*, 9:137–154, 1992.

[165] P. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor, 1997.

[166] Tekla Tóth and **Levente Hajder**. A minimal solution for image-based sphere estimation. *International Journal of Computer Vision*, page 1428–1447, 2023.

[167] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment – a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.

[168] J. W. Tukey. Mathematics and the picturing of data. *Proceedings of the International Congress of Mathematicians*, 2:523–531, 1975.

[169] K. Turkowski. Transformations of surface normal vectors. In *Tech. Rep. 22, Apple Computer*, 1990.

[170] Tekla Tóth, Zoltán Pusztai, and **Levente Hajder**. Automatic lidar-camera calibration of extrinsic parameters using a spherical target. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8580–8586, 2020.

[171] Tekla Tóth. and **Levente Hajder**. Robust fitting of geometric primitives on lidar data. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, pages 622–629. INSTICC, SciTePress, 2019.

[172] Martin Veĺas, Michal Španěl, Zdeněk Materna, and Adam Herout. Calibration of rgb camera with velodyne lidar. In *WSCG 2014 Communication Papers Proceedings*, pages 135–144, 2014.

[173] Jonathan Ventura. P1ac: Revisiting absolute pose from a single affine correspondence. *arXiv preprint arXiv:2011.08790*, 2020.

[174] Hoang-Hiep Vu, Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. High accuracy and visibility-consistent dense multiview stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(5):889–901, 2012.

[175] D. Weinshall and C. Tomasi. Linear and Incremental Acquisition of Invariant Shape Models From Image Sequences. *PAMI*, 17(5):512–517, 1995.

[176] E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series*, 1937.

[177] L. Wilson, J. S. Hampton, and H. C. Balen. Photoclinometry of terrestrial and planetary surfaces. *Lunar and Planetary Science*, 14(1):912–913, 1985.

[178] K. Wojciechowski, B. Smolka, H. Palus, R.S. Kozera, W. Skarbek, and L. Noakes. Noise Reduction in Photometric Stereo with non-Distant Light Sources. In *International Conference on Computer Vision and Graphics*, pages 103–110, 2004.

[179] Robert J. Woodham. Photometric stereo: A reflectance map technique for determining surface orientation from image intensity. In *Image Understanding Systems and Industrial Applications, Proc. SPIE*, volume 155, pages 136–143, 1978.

[180] Robert J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139–144, 1980.

[181] Fang Xiong et al. Lens-Free Optical Detection of Thermal Motion of a Submillimeter Sphere Diamagnetically Levitated in High Vacuum. *Phys. Rev. Applied*, 16(1):L011003, 2021.

[182] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[183] G. Yu and J-M. Morel. A fully affine invariant image comparison method. In *International Conference on Acoustics, Speech and Signal Processing*, pages 1597–1600. IEEE, 2009.

[184] Guoshen Yu and Jean-Michel Morel. ASIFT: An Algorithm for Fully Affine Invariant Comparison. *Image Processing On Line*, 1:11–38, 2011. `https://doi.org/10.5201/ipol.2011.my-asift`.

[185] Yizhou Yu and Johnny T. Chang. Shadow graphs and 3d texture reconstruction. *International Journal of Comput. Vision*, 62:35–60, 2005.

[186] A. Yuille and D. Snow. Shape and albedo from multiple images using integrability. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 158–164, 1997.

[187] Ji Zhang and Sanjiv Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2174–2181. IEEE, 2015.

[188] Qilong Zhang and Robert Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, September 28 - October 2, 2004*, pages 2301–2306, 2004.

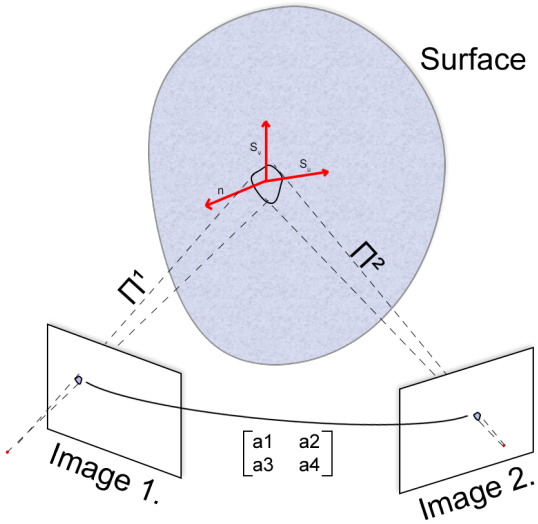
[189] Ruo Zhang, Ping-sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:690–706, 1999.

[190] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[191] Siyu Zhu, Runze Zhang, Lei Zhou, Tianwei Shen, Tian Fang, Ping Tan, and Long Quan. Very large-scale global sfm by distributed motion averaging. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

A. Az általános sztereó összefüggés levezetése az affin transzformáció, a kamera vetítőfüggvényei és a felületi normálvektor között

Adott egy térbeli lapka, amely két képen látszik. Lokális közelítésben a lapka tekinthető síknak. A feladat ábrázolása az 53. képen látható. A lapkát nem ismerjük térben, azonban a két képen a vetületekhez tartozó pixeleket kétdimenziós képfeldolgozási módszerekkel meg tudjuk becsülni. A cél a felületi lapka $\mathbf{n}$ normálvektorának meghatározása.



53. ábra. *Térbeli lapka perspektíven vetítve egy képpárra. A képeken a kiválasztott mintadarabkák között a 2×2-es affin transzformáció írja le a kapcsolatot.*

Az $[X, Y, Z]^T$ háromdimenziós vektorral megadott felületi pontok kétdimenziós koordinátáit a Π vetítőfüggvénnyel számoljuk a térbeli pontból, a felület háromdimenziós pontját pedig parametrikus alakban írjuk le:

$$x = \Pi_x(X, Y, Z) \qquad y = \Pi_y(X, Y, Z)$$
$$X = X(u, v), \qquad Y = Y(u, v), \qquad Z = Z(u, v).$$

Differenciális geometriából [85] jól ismert tény, hogy az érintővektorok felírhatóak a paraméteres alakban megadott felület parciális deriváltjaiból, a normálvektor pedig

a két érintővektor vektoriális szorzatából kapható meg:

$$\mathbf{S}_u = \begin{bmatrix} \frac{\partial X(u,v)}{\partial u} \\ \frac{\partial Y(u,v)}{\partial u} \\ \frac{\partial Z(u,v)}{\partial u} \end{bmatrix},$$

$$\mathbf{S}_v = \begin{bmatrix} \frac{\partial X(u,v)}{\partial v} \\ \frac{\partial Y(u,v)}{\partial v} \\ \frac{\partial Z(u,v)}{\partial v} \end{bmatrix},$$

$$\mathbf{n} = \mathbf{S}_u \times \mathbf{S}_v.$$

Az $[X, Y, Z]^T$ térbeli pont és az $\mathbf{S}_u$ és $\mathbf{S}_v$ érintővektorok a felület adott pontjában lévő érintősíkját is meghatározzák. Lokálisan a felület közelíthető ezzel az érintősíkkal. Feltételezzük ugyebár, hogy a felületről két kép készült. A felületdarabka vetülete a lokális környezetében az elsőrendű Taylor sor segítségével jól közelíthető:

$$\begin{bmatrix} x + \Delta x \\ y + \Delta y \end{bmatrix} \approx \begin{bmatrix} \Pi_x(X,Y,Z) \\ \Pi_y(X,Y,Z) \end{bmatrix} + \begin{bmatrix} \frac{\partial \Pi_x(X,Y,Z)}{\partial u} & \frac{\partial \Pi_x(X,Y,Z)}{\partial v} \\ \frac{\partial \Pi_y(X,Y,Z)}{\partial u} & \frac{\partial \Pi_y(X,Y,Z)}{\partial v} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}.$$

Nézzük meg ezek után, hogy a parciális deriváltak segítségével a térbeli érintősíkon és az egyik képen levő minta hogyan feleltethető meg affin transzformáció segítségével:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \approx \mathbf{A} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} \frac{\partial \Pi_x(X,Y,Z)}{\partial u} & \frac{\partial \Pi_x(X,Y,Z)}{\partial v}, \\ \frac{\partial \Pi_y(X,Y,Z)}{\partial u} & \frac{\partial \Pi_y(X,Y,Z)}{\partial v} \end{bmatrix}.$$

A parciális deriváltak a láncszabály segítségével módosíthatóak. Például:

$$\frac{\partial \Pi_x(X,Y,Z)}{\partial u} = \frac{\partial \Pi_x(X,Y,Z)}{\partial X}\frac{X}{\partial u} + \frac{\partial \Pi_x(X,Y,Z)}{\partial Y}\frac{Y}{\partial u} + \frac{\partial \Pi_x(X,Y,Z)}{\partial Z}\frac{Z}{\partial u} = \nabla \Pi_x^T \mathbf{S}_u,$$

ahol $\nabla \Pi_x$ a vetítőfüggvény gradiense X, Y és Z felületi koordináták szerint. Hasonlóan:

$$\frac{\partial \Pi_x}{\partial v} = \nabla \Pi_x^T \mathbf{S}_v \qquad \frac{\partial \Pi_y}{\partial u} = \nabla \Pi_y^T \mathbf{S}_u \qquad \frac{\partial \Pi_y}{\partial v} = \nabla \Pi_y^T \mathbf{S}_v.$$

Ebből következik, hogy magát az affin transzformációt így is fel tudjuk írni:

$$\mathbf{A} = \begin{bmatrix} \nabla\Pi_x^T \\ \nabla\Pi_y^T \end{bmatrix} \begin{bmatrix} \mathbf{S}_u & \mathbf{S}_v \end{bmatrix}.$$

Mivel sztereó képpárunk van, hiszen a térbeli alakzatunkról két képet készítettünk, az affin transzformáció a két képen látható ugyanazon minta között felírható az $\mathbf{A}_1$ transzformáció inverzének és az $\mathbf{A}_2$-es transzformációnak a szorzatával. (Az előbbi az első képen levő minta és a térbeli minta közötti kapcsolatot írja le, az utóbbi a térbeli és a második képen levő minta kapcsolatát.) Formálisan felírhatjuk az alábbi összefüggést:

$$\begin{bmatrix} \Delta x_2 & \Delta y_2 \end{bmatrix}^T = \mathbf{A}_2\mathbf{A}_1^{-1} \begin{bmatrix} \Delta x_1 & \Delta y_1 \end{bmatrix}^T.$$

A két kép közötti affin transzformációt felírhatjuk tehát $\mathbf{A}_2\mathbf{A}_1^{-1}$ alakban. A további átalakításhoz nézzük meg, hogy az $\mathbf{A}$ mátrix inverze hogyan alakul:

$$\mathbf{A}^{-1} = \frac{1}{det\left(\mathbf{A}\right)} \begin{bmatrix} \nabla^T\Pi_x\mathbf{S}_u & -\nabla^T\Pi_y\mathbf{S}_u \\ -\nabla^T\Pi_x\mathbf{S}_v & \nabla^T\Pi_y\mathbf{S}_v \end{bmatrix},$$

ahol $det(\mathbf{A}) = \nabla^T\Pi_x\mathbf{S}_u\nabla^T\Pi_y\mathbf{S}_v - \nabla^T\Pi_x\mathbf{S}_v\nabla^T\Pi_y\mathbf{S}_u$. Ha figyelembe vesszük, hogy $\mathbf{S}_v\mathbf{S}_u^T - \mathbf{S}_u\mathbf{S}_v^T = [\mathbf{n}]_\times$, akkor egyszerű átalakításokkal a következő alakra hozható az affin transzformáció:

$$\mathbf{A}_1^{-1}\mathbf{A}_2 = \frac{1}{\nabla^T\Pi_x^1\left[\mathbf{n}\right]_\times\nabla\Pi_y^1} \begin{bmatrix} \nabla^T\Pi_x^2\left[\mathbf{n}\right]_\times\nabla\Pi_y^1 & \nabla^T\Pi_x^1\left[\mathbf{n}\right]_\times\nabla\Pi_x^2 \\ \nabla^T\Pi_y^2\left[\mathbf{n}\right]_\times\nabla\Pi_y^1 & \nabla^T\Pi_x^1\left[\mathbf{n}\right]_\times\nabla\Pi_y^2 \end{bmatrix}.$$

Fontos megjegyezni, hogy skálázásra érzéketlen a formula, hiszen mind a determináns, mind a mátrix elemei $[\mathbf{n}]_\times$-el meg lettek szorozva. Az $\mathbf{a}^T\left[\mathbf{n}\right]_\times\mathbf{b}$ kifejezést szokás skaláris hármas szorzatnak is hívni. Ha figyelembe vesszük, hogy $\mathbf{a}^T\left[\mathbf{n}\right]_\times\mathbf{b}$ egyenlő $\mathbf{n}^T(\mathbf{b}\times\mathbf{a})$-vel, az affin transzformáció végleges formáját így kapjuk meg:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \mathbf{A}_1^{-1}\mathbf{A}_2 = \frac{1}{\mathbf{n}^T\mathbf{w}_5} \begin{bmatrix} \mathbf{n}^T\mathbf{w}_1 & \mathbf{n}^T\mathbf{w}_2 \\ \mathbf{n}^T\mathbf{w}_3 & \mathbf{n}^T\mathbf{w}_4 \end{bmatrix}, \tag{88}$$

ahol $\mathbf{w}_1 = \nabla\Pi_y^1 \times \nabla\Pi_x^2$, $\mathbf{w}_2 = \nabla\Pi_x^2 \times \nabla\Pi_x^1$, $\mathbf{w}_3 = \nabla\Pi_y^1 \times \nabla\Pi_y^2$, $\mathbf{w}_4 = \nabla\Pi_y^2 \times \nabla\Pi_x^1$ és $\mathbf{w}_5 = \nabla\Pi_y^1 \times \nabla\Pi_x^1$.

Ez a levezetés egy nagyon fontos összefüggéshez vezetett, hiszen a 88. egyenlet minden kameramodell esetén igaz. Mindössze a vetítő Π függvényeket kell megadni és a gradiensüket kiszámolni.

B. Paraméterekben lineáris problémák legkisebb négyzetes értelemben vett becslése

Ebben a fejezetben röviden áttekintem az alapvető lineáris becslési módszereket. Téziseimben mind a homogén, mind az inhomogén problémaosztály megoldásait használom.

B.1. Inhomogén lineáris egyenletrendszerek

Elsőként azt mutatom meg, hogy az

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{89}$$

algebrai feladatot hogyan tudjuk legkisebb négyzetes értelemben megoldani. A feladatban $\mathbf{A}$ egy $M \times N$ -es mátrix, ahol $M \geq N$. Ez a mátrix, akárcsak az M dimenziós $\mathbf{b}$ vektor, ismert elemeket tartalmaz. A cél az N dimenziós, ismeretleneket tartalmazó $\mathbf{x}$ vektor meghatározása.

Ha az $\mathbf{A}$ mátrix négyzetes lenne, akkor egyszerűen az inverzével kellene megszorozni a $\mathbf{b}$ vektort, és máris rendelkezésre állna az eredmény. Amennyiben téglalap alakú a mátrix, a bal és a jobb oldal különbségéből képzett hibavektor normáját kell minimalizálni. Legkisebb négyzetes értelemben a minimumkeresés az L_2 (kettes) norma négyzetét veszi figyelembe, annak a szélsőértékét próbálja megtalálni [46]. Formálisan ezt a feladatot így írhatjuk fel:

$$\mathbf{x} = \arg_{\mathbf{x}} \min ||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 . \tag{90}$$

Tétel

Az $\mathbf{A}\mathbf{x} = \mathbf{b}$ túlhatározott inhomogén egyenletrendszer legkisebb négyzetes optimális megoldása az $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ vektor.

Bizonyítás

Adott egy $\mathbf{A}\mathbf{x} = \mathbf{b}$ egyenletrendszer, ahol az $\mathbf{x}$ vektor N darab ismeretlent tartalmaz, és összesen M darab egyenletünk van. A feladat négyzetes értelemben minimalizálni az $\epsilon = \mathbf{A}\mathbf{x} - \mathbf{b}$ hibavektor kettes normáját. A költségfüggvény ezért így írható fel:

$$J = \epsilon^T\epsilon = (\mathbf{A}\mathbf{x} - \mathbf{b})^T(\mathbf{A}\mathbf{x} - \mathbf{b}) = (\mathbf{x}^T\mathbf{A}^T - \mathbf{b}^T)(\mathbf{A}\mathbf{x} - \mathbf{b}),$$

azaz

$$J = \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{b}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{b}.$$

Ennek pedig az $\mathbf{x}$ szerinti gradiense adja a megoldást, amikor a gradiens vektor egyenlő zérussal:

$$\nabla_{\mathbf{x}} J = 2\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{A}^T\mathbf{b} = 0.$$

Ezért a megoldás a következő alakban kapható meg:

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}.$$

[46] Miután a hiba kvadratikus, biztos, hogy a szélsőérték minimumot jelent.

Megjegyzés. Az $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ kifejezést az $\mathbf{A}$ mátrix Moore-Penrose-féle pszeudo-inverzének is szokás nevezni. A pszeudo-inverzet akkor is ki lehet számolni szinguláris értékek szerinti felbontás (SVD) segítségével, ha $(\mathbf{A}^T\mathbf{A})$ mátrix szinguláris, azaz nem létezik inverze.

B.2. Homogén lineáris egyenletrendszerek

Amennyiben a problémánk inhomogén, azaz a 89. összefüggésben a $\mathbf{b}$ vektor összes eleme nulla, az $\mathbf{x} = 0$ megoldás triviálisan helyes. Általában azonban a nem triviális megoldást keressük, ebben az esetben egy megkötést kell adnunk az $\mathbf{x}$ vektor elemeire. Leggyakrabban azt kötik ki, hogy az eredmény legyen egységvektor. Mi is ezt a megkötést alkalmazzuk.

Tétel

Az $\mathbf{Ax} = 0$ túlhatározott egyenletrendszer megoldása $\| \mathbf{x} \|_2 = 1$ megkötéssel az $\mathbf{A}^T\mathbf{A}$ mátrix legkisebb sajátértékéhez tartozó sajátvektor.

Első bizonyítás

Vezessünk be egy hibafüggvényt: $J = \| \mathbf{Ax} \|_2^2$!

Lagrange optimalizálással felhasználhatjuk az $\| \mathbf{x} \|_2^2 = 1$ megkötést, és ekkor egy módosított költségfüggvényt kapunk:

$$J' = \| \mathbf{Ax} \|_2^2 + \lambda(1 - \| \mathbf{x} \|_2^2) = \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} + \lambda(1 - \mathbf{x}^T\mathbf{x}).$$

A gradiens $\mathbf{x}$ vektor szerint:

$$\nabla_{\mathbf{x}} J' = 2\mathbf{A}^T\mathbf{Ax} - 2\lambda\mathbf{x} = 0.$$

A végső megoldás a következő alakban kapható meg:

$$\mathbf{A}^T\mathbf{Ax} = \lambda\mathbf{x}.$$

A hibafüggvény szélsőértékeit akkor kapjuk meg, amikor $\mathbf{x}$ az $\mathbf{A}^T\mathbf{A}$ sajátvektora.

A hibafüggvény minimuma ezekben az esetekben:

$$J = \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} = \lambda\mathbf{x}^T\mathbf{x} = \lambda.$$

Ez nyilvánvalóan akkor lesz a legkisebb, ha a legkisebb λ-t választjuk[47], azaz $\mathbf{x}$ az $\mathbf{A}^T\mathbf{A}$ mátrix legkisebb sajátértékéhez tartozó sajátvektor.

Alternatív bizonyítás

A feladat, hogy a $J = \| \mathbf{Ax} \|_2^2$ hibafüggvényt minimalizáljuk. Használjuk az $\mathbf{A}$ mátrix szinguláris érték szerinti felbontását: $\mathbf{A} = \mathbf{USV}^T$, ahol tudjuk, hogy $\mathbf{U}$ és $\mathbf{V}$ egyaránt ortonormált mátrix. A hibafüggvény így néz ki:

$$J = \| \mathbf{USV}^T\mathbf{x} \|_2^2.$$

Mivel $\mathbf{U}$ ortonormált, azaz csak egy elforgatást reprezentál, ami a kettes normát nem befolyásolja, teljesül, hogy $\| \mathbf{USV}^T\mathbf{x} \|_2 = \| \mathbf{SV}^T\mathbf{x} \|_2$. Ezen túl vezessük be az $\mathbf{y} = \mathbf{V}^T\mathbf{x}$ vektort mint új jelölést. Az is igaz, hogy $\| \mathbf{V}^T\mathbf{x} \|_2 = \| \mathbf{x} \|_2 = 1$, hiszen $\mathbf{V}$ is ortonormált.

Tehát a hibafüggvény így írható fel, $\| \mathbf{y} \|_2 = 1$ megkötéssel:

[47] $\mathbf{A}^T\mathbf{A}$ szimmetrikus, ezért pozitív szemidefinit mátrix, tehát az összes sajátértéke nemnegatív valós szám [24].

$$J = \| \mathbf{Sy} \| .$$

Mivel $\mathbf{S}$ diagonál mátrix, azaz csak a főátlóban vannak nem zérus elemei, és az átló csökkenő sorrendben pozitív szinguláris értékeket tartalmaz, a hibafüggvény akkor lesz minimális, ha $\mathbf{y} = [0,0,...,0,1]^T$. Ebben az esetben $\mathbf{x} = \mathbf{V}^T[0,0,...,0,1]^T$, azaz $\mathbf{x}$ vektor a $\mathbf{V}$ mátrix utolsó sora, ami az SVD működéséből adódóan az $\mathbf{A}^T\mathbf{A}$ legkisebb sajátértékéhez tartozó sajátvektor.

Következmények

A bizonyítás során azt mutattuk meg, hogy az $||\mathbf{x}||_2 = 1$ feltétellel $\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x}$ kifejezés minimális, ha $\mathbf{x}$ értékének az $\mathbf{A}^T\mathbf{A}$ mátrix legkisebb sajátértékéhez tartozó sajátvektort választjuk. Mindez akkor is igaz, ha $\mathbf{x}^T\mathbf{B}\mathbf{x}$ értékét szeretnénk minimalizálni. Természetesen ekkor $\mathbf{B}$ legkisebb sajátértékéhez tartozó sajátvektort keressük. Ha nem minimális értéket, hanem maximálisat keresünk, a két bizonyításból következik, hogy $\mathbf{B}$ legnagyobb sajátértékéhez tartozó sajátvektor a megoldás.

C. Az $\arg\min_{\mathbf{y}} \|\mathbf{Fg}-\mathbf{h}\|_2$ probléma megoldása $\|\mathbf{g}\|_2 = 1$ feltétellel

Ennek a fejezetnek a függelékben az a célja, hogy megmutassam, hogyan lehet az $\mathbf{Fg} = \mathbf{h}$ egyenletet legkisebb négyzetek értelmében optimálisan megoldani a $\mathbf{g}^T\mathbf{g} = 1$ feltétel figyelembe vételével. A J költségfüggvény a következőképpen írható fel, ha a feltételt a λ Lagrange-multiplikátorral bevezetjük a feladatba:

$$J = (\mathbf{Fg}-\mathbf{h})^T(\mathbf{Fg}-\mathbf{h}) + \lambda \mathbf{g}^T\mathbf{g}.$$

Az optimális megoldás a J gradiense a $\mathbf{g}$ vektor figyelembe vételével:

$$\nabla_{\mathbf{g}} J = 2\mathbf{F}^T(\mathbf{Fg}-\mathbf{h}) + 2\lambda\mathbf{g} = 0.$$

Ezért az optimális megoldásra írhatjuk, hogy $\mathbf{g} = \left(\mathbf{F}^T\mathbf{F} + \lambda\mathbf{I}\right)^{-1}\mathbf{F}^T\mathbf{h}$. Ebben az alakban a λ multiplikátor még ismeretlen, azt a megkötés segítségével tudjuk kiszámítani.

Az egyszerűség kedvéért jelöljük az $\mathbf{F}^{\mathrm{T}}\mathbf{h}$ vektort $\mathbf{r}$-rel, az $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ szimmetrikus mátrixot pedig $\mathbf{L}$-lel! Ekkor $\mathbf{g} = (\mathbf{L}+\lambda\mathbf{I})^{-1}\mathbf{r}$. Végezetül a $\mathbf{g}^{\mathrm{T}}\mathbf{g} = 1$ megkötésből a következő alakot lehet kapni a behelyettesítés után:

$$\mathbf{r}^{\mathrm{T}}(\mathbf{L}+\lambda\mathbf{I})^{-\mathrm{T}}(\mathbf{L}+\lambda\mathbf{I})^{-1}\mathbf{r} = 1. \tag{91}$$

A mátrix inverze így írható le:

$$(\mathbf{L}+\lambda\mathbf{I})^{-1} = \frac{\mathrm{adj}\,(\mathbf{L}+\lambda\mathbf{I})}{\det(\mathbf{L}+\lambda\mathbf{I})},$$

ahol $\mathrm{adj}\,(\mathbf{L}+\lambda\mathbf{I})$ és $\det(\mathbf{L}+\lambda\mathbf{I})$ jelölik az $\mathbf{L}+\lambda\mathbf{I}$ mátrix adjungáltját és determinánsát.

Ez a 91. egyenletbe helyettesítve adja, hogy

$$\mathbf{r}^{\mathrm{T}}\mathrm{adj}^{\mathrm{T}}(\mathbf{L}+\lambda\mathbf{I})\,\mathrm{adj}\,(\mathbf{L}+\lambda\mathbf{I})\,\mathbf{r} = \det{}^2(\mathbf{L}+\lambda\mathbf{I}).$$

Az egyenlet mindkét oldala polinomokat tartalmaz λ változóra nézve. A bal és a jobb oldal polinom foka $2n-2$, illetve $2n$. Ha az oldalakon lévő kifejezéseket kivonjuk egymásból, akkor egy $2n$-edfokú polinomot kapunk. Megjegyezzük, hogy a disszertációban tárgyalt esetben, amikor egy forgás egyetlen szögét becsüljük, $n = 2$.

Az optimális megoldást e polinom valós gyökerei alapján kapjuk meg. A becsült λ_i értékekből (polinom valós gyökei) a vektorokat a következőképpen számítjuk ki: $\mathbf{g}_i = (\mathbf{L}+\lambda_i\mathbf{I})^{-1}\mathbf{r}$. Ekkor a minimális normával rendelkező $\mathbf{Fg}_i - \mathbf{h}$ (hiba)vektor adja a feladat optimális megoldását.

D. Pontregisztrációs probléma

Ebben a függelékben megmutatom hogyan lehet – legkisebb négyzetes értelemben – optimális megoldást adni a háromdimenziós látásban és pontfelhő feldolgozásban igen gyakran felbukkanó pontregiszrációs problémára.

D.1. Pontregisztrációs feladat definíciója

Adott két ponthalmaz $\mathbf{p}_i$ és $\mathbf{o}_i$, $i = 1,2,\ldots,N$, ahol tudjuk, hogy a pontok egy térbeli euklideszi transzformációval (eltolással és elforgatással) egymásba vihetőek:

$$\mathbf{p}_i = q\mathbf{R}\mathbf{o}_i + \mathbf{t}. \tag{92}$$

Az összefüggésben $\mathbf{R}$ egy ortonormált mátrix, $\mathbf{t}$ egy eltolásvektor, q pedig a skálázást reprezentáló pozitív valós szám.

Amennyiben a pontjaink zajosak, $\mathbf{R}$, q és $\mathbf{t}$ becslését szeretnénk meghatározni úgy, hogy az alábbi – legkisebb négyzetes – hiba minimális legyen:

$$J = \sum_{i=1}^{N} ||\mathbf{p}_i - q\mathbf{R}\mathbf{o}_i - \mathbf{t}||^2 . \tag{93}$$

A konkrét feladat a költségfüggvényt minimalizálni a paraméterek szerint: $[\mathbf{R}, q, \mathbf{t}] = = \arg_{\mathbf{R},q,\mathbf{t}}\min J$.

A hibafüggvény kifejtve:

$$J = \sum_{i=1}^{N} \left\{ \mathbf{p}_i^T\mathbf{p}_i + q^2\mathbf{o}_i^T\mathbf{o}_i + \mathbf{t}^T\mathbf{t} + 2\left(q\mathbf{t}^T\mathbf{R}\mathbf{o}_i - \mathbf{p}_i^T\mathbf{t} - q\mathbf{p}_i^T\mathbf{R}\mathbf{o}_i\right)\right\}. \tag{94}$$

D.2. Pontregisztrációs probléma optimális megoldása

D.2.1. A t eltolásvektor számítása

A költségfüggvény $\mathbf{t}$ szerinti gradiense az alábbiak szerint alakul:

$$\nabla_{\mathbf{t}} J = \sum_{i=1}^{N} (2\mathbf{t} + 2q\mathbf{R}\mathbf{o}_i - 2\mathbf{p}_i) = 0. \tag{95}$$

Ebből $\mathbf{t}$-re az alábbi összefüggést kaphatjuk:

$$\mathbf{t} = \left(q\mathbf{R}\frac{1}{N}\sum_{i=1}^{N}\mathbf{o}_i - \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i \right) = 0. \tag{96}$$

Tudjuk, hogy $1/N\sum_{i=1}^{N}\mathbf{o}_i$ és $1/N\sum_{i=1}^{N}\mathbf{p}_i$ az első, illetve a második objektum súlypontja. Amennyiben origónak választjuk a súlypontot, a skálázás és az eltolás nem változtatja meg a súlypontot: továbbra is az origóban marad. Éppen ezért a megoldás az optimális eltolást adó 96. összefüggés szerint, hogy a súlypontokat az origóba visszük.

D.2.2. Optimális forgatás számítása

Tegyük fel, hogy az eltolást már kiküszöböltük, és mindkét objektum súlypontja az origóban van. Az egyszerűbb jelölés kedvéért továbbra is $\mathbf{p}_i$ és $\mathbf{o}_i$ jelölje a pontok koordinátáit. A költségfüggvény így változik:

$$J' = \sum_{i=1}^{N} \left\{ \mathbf{p}_i^T \mathbf{p}_i + q^2 \mathbf{o}_i^T \mathbf{o}_i - 2q \mathbf{p}_i^T \mathbf{R} \mathbf{o}_i \right\} \tag{97}$$

Első látásra úgy tűnik, hogy a forgatási mátrix ($\mathbf{R}$) és a skálázás (q) összefüggenek egymással, szerencsére azonban az optimális forgatást nem befolyásolja a skálázás, hiszen csak a szumma harmadik tagjában szerepel $\mathbf{R}$ és q együtt, de q-t ki lehet emelni a szumma elé.

A feladat megoldásához a következő lemmát kell belátnunk:

Lemma: Minden $\mathbf{A}$ általános és $\mathbf{R}$ ortonormált mátrixra igaz, hogy $\text{tr}(\mathbf{A}\mathbf{A}^T) \geq \geq \text{tr}(\mathbf{R}\mathbf{A}\mathbf{A}^T)$.
Lemma bizonyítása: Jelöljük $\mathbf{a}_i$-vel a $\mathbf{A}$ mátrix i-edik sorvektorát. Ekkor igaz, hogy

$$\text{tr}(\mathbf{R}\mathbf{A}\mathbf{A}^T) = \text{tr}(\mathbf{A}^T\mathbf{R}\mathbf{A}) = \sum_{i=1}^{N} \mathbf{a}_i^T \mathbf{R} \mathbf{a}_i \tag{98}$$

A Cauchy-Bunyakovszkij-Schwarz egyenlőtlenség[48] alapján írhatjuk:

$$\sum_{i=1}^{N} \mathbf{a}_i^T \mathbf{R} \mathbf{a}_i \leq \sum_{i=1}^{N} \mathbf{a}_i^T \mathbf{a}_i \tag{99}$$

Ezzel beláttuk a segédtételt, hiszen $\sum_{i=1}^{N} \mathbf{a}_i^T \mathbf{a}_i = \text{tr}(\mathbf{A}\mathbf{A}^T)$.

Rátérhetünk az optimális forgatás kiszámítását kimondó tételre:

Tétel:A forgatási mátrixra optimális megoldást ad az $\mathbf{R} = \mathbf{V}\mathbf{U}^T$ összefüggés, ahol $\mathbf{V}$ és $\mathbf{U}$ mátrixok a $\mathbf{H}$ mátrix szinguláris-érték szerinti felbontásából jönnek: $\mathbf{H} = = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, ahol $\mathbf{H} = \sum_{i=1}^{N} \mathbf{p}_i \mathbf{o}_i^T$.
Bizonyítás: A feladat a költségfüggvény minimumának meghatározása $\mathbf{R}$ szerint.

Mivel csak a 97. összefüggés harmadik tagjában szerepel a forgatásmátrix, a feladat ekvivalens $\sum_{i=1}^{N} q \mathbf{p}_i^T \mathbf{R} \mathbf{o}_i$ maximalizálásával, ami a skálázás kiemelésével és elhagyásával[49] $\sum_{i=1}^{N} \mathbf{p}_i^T \mathbf{R} \mathbf{o}_i$ maximalizálásává alakul át.

Könnyű belátni, hogy $\sum_{i=1}^{N} \mathbf{p}_i^T \mathbf{R} \mathbf{o}_i = \text{tr}(\mathbf{R}\mathbf{H})$.

Írjuk fel $\mathbf{H}$ szinguláris érték szerinti felbontását: $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, ahol $\mathbf{U}$ és $\mathbf{V}$ 3×3-as ortonormált mátrixok, $\mathbf{\Sigma}$ pedig diagonál-mátrix (nemnegatív elemekkel). Legyen $\mathbf{X} = \mathbf{V}\mathbf{U}^T$. Mivel $\mathbf{V}$ két ortonormált mátrix szorzata, maga is ortonormált. Írjuk fel $\mathbf{X}$ és $\mathbf{H}$ szorzatát:

$$\mathbf{X}\mathbf{H} = \mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T. \tag{100}$$

Eredményül egy szimmetrikus mátrixot kaptunk, melyet $\mathbf{A}\mathbf{A}^T$ alakban írhatunk fel, ha $\mathbf{A} = \mathbf{V}\sqrt{\mathbf{\Sigma}}$. Emiatt a lemmánkat alkalmazhatjuk:

$$\text{tr}(\mathbf{X}\mathbf{H}) \geq \text{tr}(\mathbf{R}_{opt}\mathbf{X}\mathbf{H}). \tag{101}$$

[48] Két tetszőleges $\mathbf{a}$ és $\mathbf{b}$ vektorra igaz, hogy $(\mathbf{a}^T\mathbf{b})^2 \leq (\mathbf{a}^T\mathbf{a})(\mathbf{b}^T\mathbf{b})$ (A tétel tetszőleges számú vektorra is felírható, de itt most csak a két vektoros esetet használjuk.)

[49] q skálázási tényező egy nemnegatív valós szám. Negatív esetben tükrözést valósít meg, ami forgatással is helyettesíthető, ezért zárhatjuk ki a negatív értékeket.

Azaz a kifejezés $\mathbf{R}_{opt} = \mathbf{I}$ esetén maximális, vagyis $\sum_{i=1}^{N} \mathbf{p}_i^T \mathbf{R}\mathbf{o}_i$ maximális megoldása $\mathbf{R} = \mathbf{X} = \mathbf{V}\mathbf{U}^T$.

D.2.3. Optimális skálázás számítása

Az optimális eltolás és elforgatás ismeretében a skálázás meghatározása egyszerű. A helyes eredményhez a költségfüggvény q szerinti deriváltját kell nullával egyenlővé tenni:

$$\frac{\partial J'}{\partial q} = \sum_{i=1}^{N} \left\{ 2q\mathbf{o}_i^T\mathbf{o}_i - 2\mathbf{p}_i^T\mathbf{R}\mathbf{o}_i \right\} = 0. \tag{102}$$

Ennek az egyenletnek a megoldása adja a végleges alakot a skálázásra:

$$q = \frac{\sum_{i=1}^{N} \mathbf{p}_i^T\mathbf{R}\mathbf{o}_i}{\sum_{i=1}^{N} \mathbf{o}_i^T\mathbf{o}_i}. \tag{103}$$

D.2.4. Az algoritmus összefoglalása

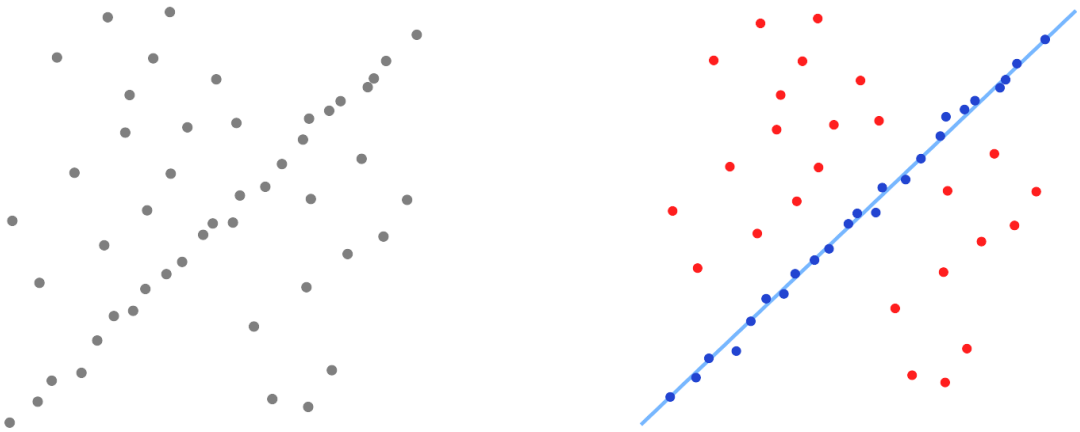
Végezetül összefoglalom az algoritmus lépéseit:

1. A két ponthalmaz súlypontját ki kell számítani, közöttük levő eltolás adja meg a ponthalmazok közötti optimális eltolást.
2. A súlypontokat az origóba kell vinni, majd az eltolt koordinátákból $\mathbf{H} = \sum \mathbf{p}_i\mathbf{o}_i^T$ mátrixot SVD-vel fel kell bontani ($\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^T$), és $\mathbf{R} = \mathbf{V}\mathbf{U}^T$ adja az optimális elforgatást.
3. A skálázás a $q = \frac{\sum_{i=1}^{N} \mathbf{p}_i^T\mathbf{R}\mathbf{o}_i}{\sum_{i=1}^{N} \mathbf{o}_i^T\mathbf{o}_i}$ hányados segítségével számítható.

Megjegyzés: a módszert természetesen robusztussá lehet tenni RANSAC/LMedS/LTS algoritmusokkal, ezzel kapcsolatos részleteket a kedves Olvasó a függelék következő, E fejezetében talál.

A pontregisztrációs algoritmus legalább három pontot igényel, és segítségével a transzformációt már ki lehet számolni, tehát három ponttal már lehet minimális modellt építeni pl. a RANSAC algoritmus részére.

hajder_274_24



(a) *Kiugró értékekkel terhelt pontfelhő* **(b)** *Robusztus illesztés eredménye*

54. ábra. *Adatpontok vonalillesztésre. A pontok egy része egyenest alkot, de sok kiugró érték (outlier) is található a ponthalmazban. A jobb oldalon a sikeresen illesztett egyenest láthatjuk, a kiugó pontokat pirossal jelöltem meg.*

E. Robusztus becslési módszerek

A függelék ezen fejezetében áttekintem aszámítógépes látásban használt legfontosabb módszereket, amelyekkel egy becslést robusztussá tudunk tenni, azaz a kiugró pontokat meg tudjuk határozni, és ki tudjuk az adathalmazból szűrni.

E.1. Bevezetés

Becslésekkel már korábban is foglalkoztunk, tudjuk, hogy számtalan matematikailag megalapozott becslési módszer létezik. Ezek közül is kiemelkedik a legkisebb négyzetes módszer, mely a legegyszerűbb becslési eljárás, ráadásul lineáris esetben optimális becslést tudunk produkálni.

Nehézséget okot, ha hibás adatok kerülnek a ponthalmazainkba, hiszen hibás értékek belekeverésével a hagyományos eljárások csak nagyon rossz becsléseket képesek produkálni.

A probléma súlyosságát egy konkrét példával szemléltetjük: adott n darab kétdimenziós pont, amelyre szeretnénk egyeneseket illeszteni. Erre a feladatra egy konkrét példát az 54a. ábrán láthatunk.

A sok véletlen pont között felsejlik középen egy egyenes képe, amely dominál, de rengeteg külső pont is található. A külső pontokat outlier-eknek, a modellhez tartozó pontokat inliereknek szokás hívni. A cél az 54b. ábrán láthatóhoz hasonló egyenes meghatározása, illesztése.

E.2. Monte-Carlo elvű robusztus módszerek

Ezeket a robusztus módszereket azért nevezték el a szerencsejátékok fővárosáról, Monte-Carlóról, mert véletlenszerű kiválasztáson alapulnak: néhány véletlenszerűen kiválasztott mintapontból készítenek ún. minimális modelleket. Ezt sokszor végzik el, és végül az így kapott minimális modellek közül azt választják ki, amelyik a legjobban

megfelel bizonyos kritériumoknak. A kaszinók rulettjét kell elképzelni, ahol számokat sorsolunk ki, és nem fontos, hogy minden egyes pörgetéskor nyerjünk, elég, ha jókor megütjük a főnyereményt.

A robusztus algoritmusokat az alábbi két lépésre lehet bontani:

1. Modellek alkotása Monte-Carlo elvű módszerek esetén véletlen pontok ismétlődő kiválasztásával.
2. Legjobb modell meghatározása.
3. Kiválasztott (legjobb) modellnek megfelelő pontok meghatározása.
4. A végső modell meghatározása az összes kiválasztott pontok figyelembe vételével.

Az egyes robusztus módszerek különféleképpen valósítják meg a két lépést, de általában minden egyes módszer ezt a két fő lépést tartalmazza. Itt két módszert vizsgálunk meg: az ún. RANSAC és az LMedS/LTS eljárásokat.

A vizsgálat előtt azonban még meg kell állapítanunk, hogy hányszor kell a véletlen pontkiválasztást megismételni. Tegyük fel, hogy p darab pontból szeretnénk a modellt kiszámolni, az outlierek aránya pedig legyen κ. Az inlierek aránya ekkor értelemszerűen $1-\kappa$. Annak a valószínűsége, hogy mind a p darab pont inlier legyen $(1-\kappa)^p$. A véletlen pontkiválasztást ismételjük meg m-szer! Annak a valószínűsége, hogy egyik modellalkotás sem jár sikerrel, azaz a p darab pontba minden esetben legalább egy outlier keveredik, felírható az alábbi összefüggés segítségével:

$$1-\Gamma = (1-(1-\kappa)^p)^m$$

Értelemszerűen Γ jelöli annak a valószínűségét, hogy legalább egy darab jó mintánk születik, azaz $\Gamma = 1-(1-(1-\kappa)^p)^m$.

A véletlen kiválasztás számát pedig m kifejtésével kaphatjuk meg:

$$m = \frac{ln(1-\Gamma)}{ln[1-(1-\kappa)^p]}$$

Ennél a képletnél beszédesebb, ha kiszámoljuk néhány konkrét értékre a szükséges mintaszámot. Legyen 95% annak a valószínűsége, hogy jó modellt kapunk legalább egyszer. Az outlierek arányát és a szükséges kiválasztási számat az alábbi táblázattal szemléltethetjük:

– Ha egyenest szeretnénk illeszteni, két pont szükséges. Ekkor $p=2$.

$\kappa[\%]$	5	10	20	30	40	50	60	70	80
m	2	2	3	5	7	11	18	32	74

– $p=3$ (pl. pontregisztrációs probléma [8] vagy síkillesztés)

$\kappa[\%]$	5	10	20	30	40	50	60	70	80
m	2	3	5	8	13	23	46	110	373

– Ha $p=4$ (pl. gyengén perspektív rekonstrukció [164])

$\kappa[\%]$	5	10	20	30	40	50	60	70	80
m	2	3	6	11	22	47	116	369	1871

– $p = 7$ (pl. 7 pontos fundamentális mátrix becslés [104])

$\kappa[\%]$	5	10	20	30	40	50	60	70	80
m	3	5	13	35	106	382	1827	13692	233963

– $p = 8$ (pl. 8 pontos fundamentális mátrix becslés [60])

$\kappa[\%]$	5	10	20	30	40	50	60	70	80
m	3	6	17	51	177	766	4570	$4{,}56 \cdot 10^4$	$1{,}17 \cdot 10^6$

Jól látható, hogy az outlierek növekedésével nagyon romlik a szükséges műveletszám, és ezáltal drasztikusan nő a futási idő. Ezért sebesség szempontjából két fontos megállapítást tehetünk:

– A futási idő szempontjából kedvező, ha minél kevesebb outlier keveredik a mintáinkba.

– A minimális modell megalkotásához szükséges pontok számától szintén radikálisan függ az eredmény. Ha a modellünket kevesebb pontból tudjuk megalkotni, sokkal alacsonyabb futási időre számíthatunk.

E.2.1. RANSAC (RANdom SAmpling Consensus)

A RANSAC algoritmus talán a legismertebb robusztus módszer a számítógépes látásban, 1981-es megalkotása [46] óta az alapcikk több tízezer hivatkozása és számtalan továbbfejlesztése mutatja a módszer népszerűségét.

Először is szükségünk van egy modellre, amely a rendelkezésre álló pontokból előállítható. Például egyenes illesztése esetén az egyenest leíró modell két valós paraméter, a és b, hiszen az egyenes $y = ax + b$ alakban adható meg. Két pont meghatároz egy egyenest.

A RANSAC módszernek is az az alapelve, hogy a lehető legkevesebb pontból meghatároz egy minimális modellt, és utána megnézi, hogy mely pontok illeszkednek erre a modellre. Az egyenes illesztés esetén tehát meghatározzuk a és b paramétereket két pont segítségével, majd az i-edik pont koordinátáit behelyettesítve megkapjuk a hibát a pontra:

$$\epsilon_i = y_i - b - ax_i$$

Ezek után számoljuk össze azokat a pontokat, amelyek megadott küszöbön belül vannak, akkor megkapjuk, hogy hány pont támogatja a modellt (latin szóval konszenzust alkot a modellel). Ezt a küszöböt jelöljük ϵ_{thr}-rel, A 'thr' rövidítés utal a küszöb angol nevére, a threshold-ra. Ezek után újabb két pontot választunk, és újra egy egyenest húzunk, ahol a távolságokat és a küszöbön belül levő pontokat újfent meghatározhatjuk.

A sok lehetséges modell közül azt választjuk ki, amelyikhez a legtöbb konszenzusos pont tartozik. Végül a a modellt a konszenzusos modellből meghatározott inlier pontok alapján újra kell számolni.

A RANSAC algoritmus nagy előnye, hogy egyszerű, gyorsan implementálható. Hátránya, hogy az ϵ_{thr} küszöböt előzetesen meg kell határozni, és ez nem is olyan egyszerű feladat: ha a küszöb szigorú, sok jó pontot kidobunk, ha nagyon laza a küszöb, akkor outlierek is bekerülhetnek az adatainkba, és ezek rontják a végeredmény minőségét.

E.2.2. LMedS (Least MEDian of Squares)

A Least MEDian of Squares segítségével a küszöb eltüntethető. Statisztikusok arra a megállapításra jutottak [141], hogy ha az egyes ϵ_i hibaértékek Gauss-eloszlást követnek, akkor az inlierek pozícióinak szórására robusztus becslést adhatunk. Ehhez be kell vezetni egy súlyozó számot:

$$s^0 = 1{,}4826\frac{5}{n-p}median\{\epsilon_i\},$$

ahol n az összes pont száma, amiből p darabot kell választani a modellépítéshez. Sokszor ismételjük a modellalkotást, véletlenszerűen mindig kiválasztva p darab pontot. Minden esetben kiszámoljuk s^0-t a teljes adathalmazra. Végül azt a minimális modellt választjuk ki, amelyikhez kapott s^0 a legkisebb. Az lesz a helyes modell, amelyiknél a medián – azaz a sorba rakott ϵ_i-k közül a középső – a legkisebb.

Ha megvan a minimális s^0, akkor meg lehet becsülni a valódi szórást az alábbi összefüggés segítségével:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(w_i\epsilon_i^2)}{\sum_{i=1}^{N} w_i - 4}},$$

ahol N jelöli a minták számát, w_i pedig egy bináris változó: ha $\epsilon_i < 2{,}5s^0$, akkor $w_i = 1$, egyébként nulla.

Végezetül meg kell határozni a becsült helyes modellhez tartozó pontokat. Azok a pontokat jelöljük inliereknek (modellhez tartozónak), amelyekhez tartozó ϵ_i hibaértékek 2.5σ-nál kisebbek.

E.2.3. LTS (Least Trimmed Squares)

Az LMedS módszernek az a hátránya, hogy nem működik helyesen, ha az kiugró értékek aránya ötven százalék fölé megy, hiszen a medián akkor outlierből származó hibára fog mutatni helyes modell esetén is. Ezért egy apró változtatást végeztek a kutatók: a s^0-t meghatározó összefüggésben lecserélték a mediánt az első d darab legjobb hiba összegére. Az a szerencsés választás, ha d a várható outlier aránynál nem sokkal kisebb. (De mindenképpen kisebb!)

F. Affin transzformációk felbontása

Ebben a függelékben megmutatom, hogyan lehet az affin transzformációkat komponensekre bontani. Az affin transzformációk az alábbi összefüggéssel mozgatják az első képen a $\mathbf{p}_1$ pontot a második képen a $\mathbf{p}_2$ pontba:

$$\mathbf{p}_2 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \mathbf{p}_1 + \begin{bmatrix} a_u \\ a_v \end{bmatrix}, \quad \mathbf{p}_1 = \begin{bmatrix} u_1 \\ v_1 \end{bmatrix}, \quad \mathbf{p}_2 = \begin{bmatrix} u_2 \\ v_2 \end{bmatrix}.$$

F.1. Kategóriák szerinti felbontás

Az affin (lineáris) transzformációk alapvetően négy kategóriába oszthatóak:

1. Eltolás: a fenti összefüggésben az $[a_u \quad a_v]^T$ vektor határozza meg az *eltolást*.
2. Kétdimenziós (síkban) *forgatás* origó körül, α szöggel:

$$\mathbf{A}_{\text{rot}} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}.$$

3. *Skálázás* mindkét (vízszintes és függőleges) fő irány mentén:

$$\mathbf{A}_{\text{scales}} = \begin{bmatrix} s_u & 0 \\ 0 & s_v \end{bmatrix}.$$

4. *Nyírás*:

$$\mathbf{A}_{\text{skew}} = \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}.$$

Az 55. ábrán láthatóak az egyes kategóriák. Az affin transzformáció lineáris részét, amelyikkel a $\mathbf{p}_1$ vektort közvetlenül megszorozzuk, fel lehet írni a kategóriák kompizíciójaként:

$$\mathbf{A} = \mathbf{A}_{\text{rot}}\mathbf{A}_{\text{scales}}\mathbf{A}_{\text{skew}}.$$

Egy általános affin transzformáció hatása ezért az alábbi formulával írható le:

$$\mathbf{p}_1 = \mathbf{A}_{\text{rot}}\mathbf{A}_{\text{scales}}\mathbf{A}_{\text{skew}}\mathbf{p}_2 + [a_u \quad a_v]^T.$$

A transzformációs mátrixok elemeit behelyettesítve kaphatjuk, hogy

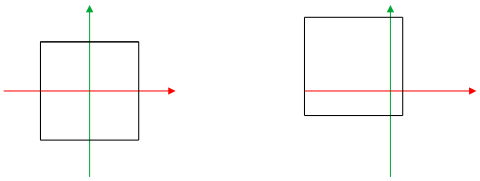
$$\mathbf{p}_1 = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} s_u & 0 \\ 0 & s_v \end{bmatrix} \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} \mathbf{p}_2 + \begin{bmatrix} a_u \\ a_v \end{bmatrix}.$$

Az 56. ábrán azt láthatjuk, hogy az általános affin transzformáció egy négyzetből paralelogrammát, körből pedig nyírt ellipszist képes készíteni.

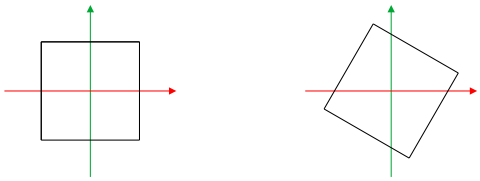
A transzformáció komponenseinek meghatározása Mivel az eltolást egyértelműen meghatározza a hozzáadott vektor, az alábbi lineáris részt kell részekre bontani:

$$\mathbf{A} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} s_u & 0 \\ 0 & s_v \end{bmatrix} \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}.$$

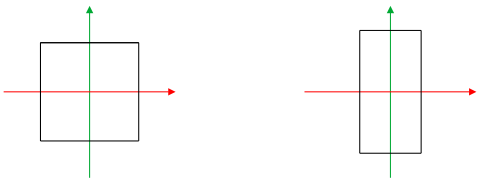
Kifejtve írhatjuk, hogy



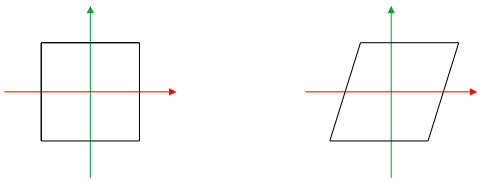
(a) *Eltolás*



(b) *Forgatás*

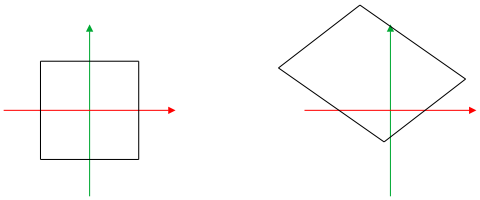


(c) *Vízszintes és függőleges skálázás*

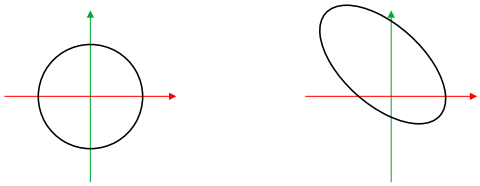


(d) *Nyírás*

55. ábra. *Affin transzformációk kategóriái. Az ábrák a négyzet mint példaobjektum alakját mutatják meg az egyes transzformációk után.*



(a) *Négyzet → parallelogramma*



(b) *Kör → Nyírt ellipszis*

56. ábra. *Általános affin transzformáció hatása. A négyzet transzformáltja paralelogramma lesz, a körből pedig elnyírt ellipszist lehet készíteni.*

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} s_u \cos\alpha & s_u b \cos\alpha - s_v \sin\alpha \\ s_u \sin\alpha & s_u b \sin\alpha + s_v \cos\alpha \end{bmatrix}.$$

Miután $a_{11} = s_u \cos\alpha$ és $a_{21} = s_u \sin\alpha$, az α forgatási szög és a vízszintes s_u skálázás polárkoordinátaként képzelhető el, ahol a skála a sugár. Ezért igaz, hogy

$$\begin{aligned} \alpha = &\ \operatorname{atan2}(a_{21}, a_{11}), \\ s_u = &\ \sqrt{a_{11}^2 + a_{21}^2}. \end{aligned}$$

A másik két paraméterre az $a_{12} = s_u b \cos\alpha - s_v \sin\alpha$ és az $a_{22} = s_u b \sin\alpha + s_v \cos\alpha$ összefüggések adottak. α szög és s_u skála ismertek, ezért a másik két paraméterre, az s_v függőleges skálára és a b nyírási paraméterre fel lehet írni, hogy

$$\begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix} = \begin{bmatrix} -\sin\alpha & s_u \cos\alpha \\ \cos\alpha & s_u \sin\alpha \end{bmatrix} \begin{bmatrix} s_v \\ b \end{bmatrix}.$$

Ez pedig egy inhomogén, lineáris egyenletrendszer, aminek egyértelmű megoldása van, tehát s_v és b a mátrix inverzének felhasználásával könnyen kiszámítható.

Többértelműség. A dekompozíciónak van egy többértelműsége, amelyet az okoz, hogy a 180 fokos forgatás megfelel egy középpontos tükrözésnek, a középpontos tükrözés pedig felfogható úgy is, hogy mind a vízszintes, mind a függőleges tengelyre tükrözünk.

Ha az eredeti α szöget $\pi/2$-vel elforgatjuk, majd skálázzuk, akkor az alábbi levezetést tehetjük:

$$\begin{aligned} &\begin{bmatrix} \cos(\alpha \pm \pi/2) & -\sin(\alpha \pm \pi/2) \\ \sin(\alpha \pm \pi/2) & \cos(\alpha \pm \pi/2) \end{bmatrix} \begin{bmatrix} s_u & 0 \\ 0 & s_v \end{bmatrix} = \\ &= \begin{bmatrix} -\cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & -\cos(\alpha) \end{bmatrix} \begin{bmatrix} s_u & 0 \\ 0 & s_v \end{bmatrix} = \\ &= \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} -s_u & 0 \\ 0 & -s_v \end{bmatrix}. \end{aligned}$$

Az eredmény alapján kijelenthetjük, az s_u és s_v skálatényezők egyidejű invertálása ekvivalens azzal, ha az eredeti α szöget $180°$-kal megnöveljük/lecsökkentjük.

F.2. Szinguláris érték szerinti felbontás

Másik népszerű felbontás SVD [50] segítségével bontja fel az affin transzformáció lineáris részét:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \mathbf{U} \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \mathbf{V}^T, \tag{104}$$

[50] Az SVD (Singular Value Decomposition) A szinguláris értékek szerinti felbontás. Minden valós elemű mátrix egyértelműen felbontható két ortonormált és egy diagonál mátrix szorzatává.

ahol $\mathbf{U}$ és $\mathbf{V}$ azonos méretű (2×2-es) ortonormált mátrix, a középső mátrix diagonál, a főátlóban nemnegatív valós számok szerepelnek. Ez a felbontás tehát két forgatásra és egy skálázásra bontja a transzformációt. A skálázáshoz a vízszintes és függőleges méretezés külön megadható.

Az ortonormált mátrixokra felírhatjuk, hogy

$$\mathbf{U} = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix}, \qquad \mathbf{V} = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix}.$$

Az SVD alapú felbontás tehát két forgatási szögre (β és γ) és két valós skálázási paraméterre (s_1 és s_2) bontja fel a 2×2-es általános affin transzformációt.

G. A lehetséges homográfiák terének csökkentése ismert fundamentális mátrix esetén

Amennyiben egy háromdimenziós térben két kamera adott, mellyel egy síkfelületet fényképezünk le, akkor mind a sík-sík homográfia, mind a fundamentális mátrix segítségével összefüggéseket fogalmazhatunk meg a sík pontjainak vetületei között. A fundamentális mátrixot és a homográfiát össze is köthetjük [62]:

$$\left[\mathbf{e}^{(2)}\right]_\times \mathbf{H} = \lambda \mathbf{F}, \tag{105}$$

ahol $\mathbf{e}^{(2)} = [e_x^{(2)}, e_y^{(2)}, 1]^T$ a második képen az epipólust jelöli, λ pedig az $\mathbf{F}$ fundamentális mátrixhoz tartozó ismeretlen skála. A $[\mathbf{v}]_\times$ operátor a $\mathbf{v}$ vektorral (balról) elvégzendő vektoriális szorzást reprezentálja. Meg kell jegyezni, hogy a $[\mathbf{v}]_\times$ keresztszorzat mátrix rangja mindig legfeljebb kettő, ezért bármelyik sora a mátrixnak meghatározható a másik kettő lineáris kombinációjaként.

A dolgozatban a 105. összefüggés megmutatja, hogy a fundamentális mátrix ismerete hogyan csökkenti a homográfia becslésnek a szabadságfokát. A mátrix utolsó sorát a lineáris összefüggés miatt nem tudjuk felhasználni.

A redundáns ismereteket kihagyva írhatjuk, hogy

$$\begin{bmatrix} 0 & -1 & e_y \\ 1 & 0 & -e_x \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \lambda \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \end{bmatrix}. \tag{106}$$

Az elemeket egyesével véve felírhatjuk a végső összefüggéseket, melyek a fundamentális mátrixot [51] és a homográfiát összekötik:

$$\begin{aligned} h_{11} &= e_x h_{31} + \lambda f_{21}, \quad & h_{12} &= e_x h_{32} + \lambda f_{22}, \\ h_{13} &= e_x h_{33} + \lambda f_{23}, \quad & h_{21} &= e_y h_{31} - \lambda f_{11}, \\ h_{22} &= e_y h_{32} - \lambda f_{12}, \quad & h_{23} &= e_y h_{33} - \lambda f_{13}. \end{aligned} \tag{107}$$

A kapott alakokból triviálisan látszik, hogy a h_{31}, h_{32} és h_{33} elemekből a másik hat eleme a homográfiának egyértelműen meghatározható, ha $\mathbf{F}$ ismert. Tehát a homográfia becslésének szabadságfoka hárommal csökken.

Mind a $\mathbf{H}$ homográfia, mind az $\mathbf{F}$ fundamentális mátrix skálázásra invariáns. Ezért a skálára előírhatunk egy újabb megkötést. Esetünkben válasszuk a $\lambda = 1$ feltételt.

Miután a DLT algoritmust [62] használó homográfia-becslés esetében minden pontmegfelelés két egyenletet ad, logikus lenne, hogy a három szabadságfokot két pontpár segítségével megkaphatjuk. Sajnos a két egyenlet összefüggő, ezért csak ez egyik ad új információt. *Így három pontra van szükség, hogy ismert fundamentális mátrix esetén a homográfiát meg tudjuk becsülni.* Ezért 3PT-nek hívjuk ebben a dokumentumban ezt a kizárólag pontpozíciókat alkalmazó módszert.

[51] Az epipólus a fundamentális mátrix nullteréből egyértelműen számítható.

H. Az affin transzformáció normalizálása

Ha két egymásnak megfelelő pontot ismerünk a két képen, melyeket $\mathbf{x}^{(1)}$-el és $\mathbf{x}^{(2)}$ jelöljük, a célunk ebben a részben, hogy az $\mathbf{x}'^{(1)} = \mathbf{T}_1\mathbf{x}^{(1)}$ és $\mathbf{x}'^{(2)} = \mathbf{T}_2\mathbf{x}^{(2)}$ formulával leírt normalizálás hatását megvizsgáljuk a 2×2-es $\mathbf{A}$ affin transzformációra. A normalizálás esetünkben a hagyományos megoldást [62], azaz a skálázást és eltolást jelenti. A transzformációs mátrix ekkor így néz ki:

$$\mathbf{T}_1 = \begin{bmatrix} s_x^{(1)} & 0 & t_x^{(1)} \\ 0 & s_y^{(1)} & t_y^{(1)} \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{T}_2 = \begin{bmatrix} s_x^{(2)} & 0 & t_x^{(2)} \\ 0 & s_y^{(2)} & t_y^{(2)} \\ 0 & 0 & 1 \end{bmatrix}. \tag{108}$$

Egy tetszőleges $\mathbf{x}^{(i)} = [u^{(i)}, v^{(i)}]$ kétdimenziós pont koordinátái így alakulnak át a transzformáció során:

$$\mathbf{x}'^{(i)} = \begin{bmatrix} s_x^{(i)} & 0 & t_x^{(i)} \\ 0 & s_y^{(i)} & t_y^{(i)} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u^{(i)} \\ v^{(i)} \\ 1 \end{bmatrix} = \begin{bmatrix} s_x^{(i)}u^{(i)} + t_x^{(i)} \\ s_y^{(i)}v^{(i)} + t_y^{(i)} \\ 1 \end{bmatrix}.$$

Ha a $\mathbf{H}$ jelöli az eredeti homográfiát a normalizálatlan koordonátákra, akkor $\mathbf{x}_2 \sim$ $\sim \mathbf{H}\mathbf{x}_1$.

Amikor a normalizált koordinátákat használjuk, az összefüggés átalakul: $\mathbf{T}_2^{-1}\mathbf{x}'^{(2)} \sim$ $\sim \mathbf{H}\mathbf{T}_1^{-1}\mathbf{x}'^{(1)}$. Ezért $\mathbf{H}' = \mathbf{T}_2\mathbf{H}\mathbf{T}_1^{-1}$. Ehhez a $\mathbf{T}_1$ inverzét meg kell határoznunk:

$$\mathbf{T}_1^{-1} = \begin{bmatrix} 1/s_x^{(1)} & 0 & -t_x^{(1)}/s_x^{(1)} \\ 0 & 1/s_y^{(1)} & -t_y^{(1)}/s_y^{(1)} \\ 0 & 0 & 1 \end{bmatrix}. \tag{109}$$

A normalizált homográfiára írhatjuk, hogy

$$\mathbf{H}' = \mathbf{T}_2\mathbf{H}\mathbf{T}_1^{-1} = \begin{bmatrix} s_x^{(2)} & 0 & t_x^{(2)} \\ 0 & s_y^{(2)} & t_y^{(2)} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} 1/s_x^{(1)} & 0 & -t_x^{(1)}/s_x^{(1)} \\ 0 & 1/s_y^{(1)} & -t_y^{(1)}/s_y^{(1)} \\ 0 & 0 & 1 \end{bmatrix} \tag{110}$$

$$\mathbf{H}' = \begin{bmatrix} s_x^{(2)}h_{11} + t_x^{(2)}h_{31} & s_x^{(2)}h_{12} + t_x^{(2)}h_{32} & s_x^{(2)}h_{13} + t_x^{(2)}h_{33} \\ s_y^{(2)}h_{21} + t_y^{(2)}h_{31} & s_y^{(2)}h_{22} + t_y^{(2)}h_{32} & s_y^{(2)}h_{23} + t_y^{(2)}h_{33} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} 1/s_x^{(1)} & 0 & -t_x^{(1)}/s_x^{(1)} \\ 0 & 1/s_y^{(1)} & -t_y^{(1)}/s_y^{(1)} \\ 0 & 0 & 1 \end{bmatrix} \tag{111}$$

$$\mathbf{H}' = \begin{bmatrix} \frac{s_x^{(2)}}{s_x^{(1)}}h_{11} + \frac{t_x^{(2)}}{s_x^{(1)}}h_{31} & \frac{s_x^{(2)}}{s_y^{(1)}}h_{12} + \frac{t_x^{(2)}}{s_y^{(1)}}h_{32} & * \\ \frac{s_y^{(2)}}{s_x^{(1)}}h_{21} + \frac{t_y^{(2)}}{s_x^{(1)}}h_{31} & \frac{s_y^{(2)}}{s_y^{(1)}}h_{22} + \frac{t_y^{(2)}}{s_y^{(1)}}h_{32} & * \\ \frac{1}{s_x^{(1)}}h_{31} & \frac{1}{s_y^{(1)}}h_{32} & -h_{31}t_x^{(1)}/s_x^{(1)} - h_{32}t_y^{(1)}/s_y^{(1)} + h_{33} \end{bmatrix} \tag{112}$$

A 16. összefüggés (II. törvény) alapján az affin transzformációt a két csillaggal jelölt elem nem befolyásolja, ezért ezeket nem vesszük figyelembe a továbbiakban, az értékét az egyszerűség kedvéért jelöltem csillaggal.

A II. törvény alapján írhatjuk, hogy

$$a'_{11} = \frac{h'_{11} - h'_{31}u'^{(2)}}{s'}, \quad a'_{12} = \frac{h'_{12} - h'_{32}u'^{(2)}}{s'},$$
$$a'_{21} = \frac{h'_{21} - h'_{31}v'^{(2)}}{s'}, \quad a'_{22} = \frac{h'_{22} - h'_{32}v'^{(2)}}{s'}.$$

Ezután a normalizált s' projektív mélységet fejezem ki:

$$\begin{aligned} s' = \frac{1}{s_x^{(1)}} h_{31} u'^{(1)} + \frac{1}{s_y^{(1)}} h_{32} v'^{(1)} - \\ \frac{1}{s_x^{(1)}} h_{31} t_x^{(1)} - \frac{1}{s_y^{(1)}} h_{32} t_y^{(1)} + h_{33} = \\ \frac{1}{s_x^{(1)}} h_{31} \left(u'^{(1)} - t_x^{(1)} \right) + \frac{1}{s_y^{(1)}} h_{32} \left(v'^{(1)} - t_y^{(1)} \right) + h_{33} = \\ u^{(1)} h_{31} + v^{(1)} h_{32} + h_{33} = s. \end{aligned}$$

Azaz pontosan a normalizálatlan s értéket kaptuk vissza. Tehát megállapíthatjuk, hogy ezt a skálát a normalizálás nem befolyásolja.

A normalizált affin transzformáció bal felső elemét nézzük először meg, az s' skálával beszorozva:

$$\begin{aligned} s' a'_{11} = h'_{11} - h'_{31} u'^{(2)} = \\ \frac{s_x^{(2)}}{s_x^{(1)}} h_{11} + \frac{t_x^{(2)}}{s_x^{(1)}} h_{31} - \frac{1}{s_x^{(1)}} h_{31} \left(s_x^{(2)} u^{(2)} + t_x^{(2)} \right) = \\ \frac{s_x^{(2)}}{s_x^{(1)}} h_{11} + \frac{t_x^{(2)}}{s_x^{(1)}} h_{31} - \frac{s_x^{(2)}}{s_x^{(1)}} u^{(2)} h_{31} - \frac{t_x^{(2)}}{s_x^{(1)}} h_{31} = \\ \frac{s_x^{(2)}}{s_x^{(1)}} h_{11} - \frac{s_x^{(2)}}{s_x^{(1)}} u^{(2)} h_{31}. \end{aligned}$$

A másik három elemet teljesen hasonlóan kapjuk meg:

$$\begin{aligned} s' a'_{12} = h'_{12} - h'_{32} u'^{(2)} = \frac{s_x^{(2)}}{s_y^{(1)}} h_{12} - \frac{s_x^{(2)}}{s_x^{(1)}} u^{(2)} h_{32}, \\ s' a'_{21} = h'_{21} - h'_{31} v'^{(2)} = \frac{s_y^{(2)}}{s_x^{(1)}} h_{21} - \frac{s_y^{(2)}}{s_x^{(1)}} v^{(2)} h_{31}, \\ s' a'_{22} = h'_{22} - h'_{32} v'^{(2)} = \frac{s_y^{(2)}}{s_y^{(1)}} h_{22} - \frac{s_y^{(2)}}{s_y^{(1)}} v^{(2)} h_{32}. \end{aligned}$$

Ezzel megkaptuk a végleges összefüggéseket a normalizálás hatásának vizsgálatára.

I. Kamera előtti gömb kontúrjának meghatározása

Ebben a részben azt mutatjuk meg, hogy a IV. téziscsoportban használt kalibrációs gömb kontúrja mikor ad ellipszist a képen.

A másodfokú görbe akkor lesz ellipszis, ha a 81. összefüggésben megadott másodfokú görbe (kúpszelet) paramétereire igaz, hogy $B^2 - 4AC < 0$. Esetünkben:

$$\begin{aligned}
B^2 - 4AC = (2x_0y_0)^2 - 4\left(r^2 - y_0^2 - z_0^2\right)\left(r^2 - x_0^2 - z_0^2\right) = \\
4x_0^2y_0^2 - 4\left[r^4 - r^2\left(x_0^2 + z_0^2\right) - r^2\left(y_0^2 + z_0^2\right) + y_0^2x_0^2 + y_0^2z_0^2 + x_0^2z_0^2 + z_0^4\right] = \\
4\left[-r^4 + r^2\left(x_0^2 + y_0^2 + 2z_0^2\right) - y_0^2z_0^2 - x_0^2z_0^2 - z_0^4\right] = \\
4\left[-r^4 + r^2\left(x_0^2 + y_0^2 + 2z_0^2\right) - z_0^2\left(x_0^2 + y_0^2 + z_0^2\right)\right] = \\
4\left[-r^4 + r^2z_0^2 + r^2\left(x_0^2 + y_0^2 + z_0^2\right) - z_0^2\left(x_0^2 + y_0^2 + z_0^2\right)\right] = \\
4\left[-r^4 + r^2z_0^2 + \left(r^2 - z_0^2\right)\left(x_0^2 + y_0^2 + z_0^2\right)\right].
\end{aligned}$$

Ez a kifejezés akkor negatív, ha

$$-r^4 + r^2z_0^2 + \left(r^2 - z_0^2\right)\left(x_0^2 + y_0^2 + z_0^2\right) < 0.$$

r^2-et az első két tagból kiemelve tovább lehet a bal oldalt egyszerűsíteni:

$$\left(r^2 - z_0^2\right)\left(x_0^2 + y_0^2 + z_0^2 - r^2\right) < 0. \tag{113}$$

A kifejezés előjelét a két szorzótényező előjele adja meg. Ha a gömb teljes egészében a kamera előtt van, akkor biztos, hogy $z_0^2 > r^2$, és ekkor az is triviálisan igaz, hogy $(x_0^2 + y_0^2 + z_0^2) > r^2$. Így azt kapjuk, hogy $(r^2 - z_0^2) < 0$ és $(x_0^2 + y_0^2 + z_0^2 - r^2) > > 0$, tehát a 113. összefüggésben a szorzatnak negatív az előjele, ha a gömb a kamera előtt helyezkedik el.

Azaz kijelenthetjük, hogy a kamera előtt elhelyezkedő gömb kontúrpontjai a képen egy ellipszist alkotnak.

J. Síkban mozgás

Síkban mozgás esetén feltételezzük, hogy a kamera egy járműhöz van rögzítve, és ez a jármű egy sík úttesten közlekedik. Két különböző pillanatban rögzítjük a kamerának a képét. Feltételezzük továbbá, hogy a kamera úgy lett rögzítve, hogy az optikai tengelye párhuzamos az úttal.

Ekkor a két kép között az eltolásvektor a második koordinátájában változatlan ezért írhatjuk, hogy

$$\mathbf{t} = \begin{bmatrix} t_x \\ 0 \\ t_z \end{bmatrix} = \rho \begin{bmatrix} \cos\alpha \\ 0 \\ \sin\alpha \end{bmatrix}, \qquad \rightarrow [\mathbf{t}]_\times = \rho \begin{bmatrix} 0 & -\sin\alpha & 0 \\ \sin\alpha & 0 & -\cos\alpha \\ 0 & \cos\alpha & 0 \end{bmatrix}.$$

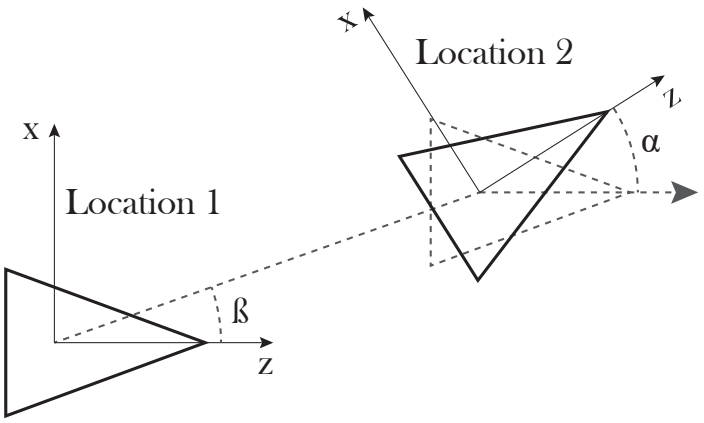
Így a kétdimenziós eltolás a síkban a mozgás irányával és nagyságával adható meg.

A forgás – miután a második tengely a függőleges (gravitációs) iránynak felel meg –, egy szög segítségével adható meg:

$$\mathbf{R} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}.$$

Lényegi mátrix síkban mozgás esetén. A lényegi mátrix a 7. összefüggés segítségével adható meg. Trigonometrikus alapösszefüggések segítségével végezzük az alábbi levezetést:

$$\begin{aligned} \mathbf{E} = [\mathbf{t}]_\times \mathbf{R} = \rho \begin{bmatrix} 0 & -\sin\alpha & 0 \\ \sin\alpha & 0 & -\cos\alpha \\ 0 & \cos\alpha & 0 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} = \\ \begin{bmatrix} 0 & -\sin\alpha & 0 \\ \sin\alpha\cos\beta + \cos\alpha\sin\beta & 0 & \sin\alpha\sin\beta - \cos\alpha\cos\beta \\ 0 & \cos\alpha & 0 \end{bmatrix} = \\ \begin{bmatrix} 0 & -\sin\alpha & 0 \\ \sin(\alpha+\beta) & 0 & -\cos(\alpha+\beta) \\ 0 & \cos\alpha & 0 \end{bmatrix}. \end{aligned} \tag{114}$$



57. ábra. *Síkban mozgás esetén két szöggel tudjuk leírni a mozgást.*

Alapvető mátrix síkban mozgás esetén. Ha a kameráink kalibráltak, akkor a $\mathbf{K}_1$ és $\mathbf{K}_2$ belső paramétereket tartalmazó kalibrációs mátrixokat ismerjük. A jól ismert $\mathbf{X}^T\mathbf{E}\mathbf{X}_1 = 0$ összefüggés a lényegi mátrixot és a térbeli $\mathbf{X}_1$ és $\mathbf{X}_2$ pontokat köti össze. A kalibrációs mátrixok ismerete esetén írhatjuk, hogy $\mathbf{X}_1 \sim \mathbf{K}^{-1}\mathbf{p}_1$ és $\mathbf{X}_2 \sim \mathbf{K}_2^{-1}\mathbf{p}_2$. A síkban mozgás esetén egy kamerát használunk, amely járműre van rögzítve. Ezért a belső paramétereket tartalmazó kameramátrixok megegyeznek a két képre, azaz $\mathbf{K}_1 = \mathbf{K}_2$. Az egyszerűség kedvéért a továbbiakban $\mathbf{K}$-val jelöljük ezt a mátrixot. Ekkor:

$$\mathbf{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \mathbf{K}^{-1} = \begin{bmatrix} 1/f & 0 & -u_0/f \\ 0 & 1/f & -v_0/f \\ 0 & 0 & 1 \end{bmatrix}.$$

Félig kalibrált esetben csak a fókusztávot tekintjük ismeretlennek, a döféspontot az origóba helyeztük. Ebben az esetben pedig azt írhatjuk, hogy

$$\mathbf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \mathbf{K}^{-1} = \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Az alapvető mátrixra igaz, hogy $\mathbf{F} = \mathbf{K}^{-T}\mathbf{E}\mathbf{K}^{-1}$. Ekkor a kameraparamétereket a 7. formulában megadott lényegi mátrix elemeibe behelyettesítve azt kapjuk, hogy

$$\mathbf{F} = \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ -u_0/f & -v_0/f & 1 \end{bmatrix} \begin{bmatrix} 0 & -\sin\alpha & 0 \\ \sin(\alpha+\beta) & 0 & -\cos(\alpha+\beta) \\ 0 & \cos\alpha & 0 \end{bmatrix} \begin{bmatrix} 1/f & 0 & -u_0/f \\ 0 & 1/f & -v_0/f \\ 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ -u_0/f & -v_0/f & 1 \end{bmatrix} \begin{bmatrix} 0 & -\frac{\sin\alpha}{f} & \frac{\sin\alpha}{f}v_0 \\ \frac{\sin(\alpha+\beta)}{f} & 0 & -\frac{\sin(\alpha+\beta)}{f}u_0 - \cos(\alpha+\beta) \\ 0 & \frac{\cos\alpha}{f} & -\frac{\cos\alpha}{f}v_0 \end{bmatrix} =$$

$$\begin{bmatrix} 0 & -\frac{\sin\alpha}{f^2} & \frac{\sin\alpha}{f^2}v_0 \\ \frac{\sin(\alpha+\beta)}{f^2} & 0 & -\frac{\sin(\alpha+\beta)}{f^2}u_0 - \frac{\cos(\alpha+\beta)}{f} \\ -\frac{\sin(\alpha+\beta)}{f^2}v_0 & \frac{\cos\alpha}{f} + \frac{\sin\alpha}{f^2}u_0 & -\frac{\sin\alpha}{f^2}u_0v_0 + \frac{\sin(\alpha+\beta)}{f}u_0v_0 + \frac{\cos(\alpha+\beta)}{f}v_0 - -\frac{\cos\alpha}{f}v_0 \end{bmatrix}$$

Félig kalibrált kamerák esetén $u_0 = v_0 = 0$. Ekkor a tagok lényegesen egyszerűsödnek, kialakul a lényegi mátrixnál a "sakktáblás" mintázat:

$$\mathbf{F} = \begin{bmatrix} 0 & -\frac{\sin\alpha}{f^2} & 0 \\ \frac{\sin(\alpha+\beta)}{f^2} & 0 & -\frac{\cos(\alpha+\beta)}{f} \\ 0 & \frac{\cos\alpha}{f} & 0 \end{bmatrix}.$$

K. Legkisebb négyzetes optimális megoldó síkban mozgáshoz

Ebben a szakaszban egy – a legkisebb négyzetek szempontjából optimális – algoritmust javasolok a lényegi mátrix becslésére síkban mozgás esetére, ha legalább három egyenlet áll rendelkezésre. Túlhatározott esetben, azaz négy vagy még több egyenlet esetén is optimális a módszer.

A probléma $\mathbf{Ax} = \mathbf{0}$ homogén lineáris alakban adott, ahol

$$\mathbf{x} = [\cos\beta \quad \sin\beta \quad \cos(\alpha+\beta) \quad \sin(\alpha+\beta)]^{\mathrm{T}}.$$

Az $\mathbf{x}$ vektor koordinátái összefüggnek. Az első kettő és az utolsó kettő érték ugyanannak a szögnek a szinusz- és koszinuszfüggvénye. A homogenitásból következik, hogy a megoldás csak egy skáláig határozható meg. Ezért, ha az $\mathbf{x}$ paramétervektort két 2D alvektorra osztjuk, akkor e vektorok hosszának egyenlőnek kell lennie.

A megkötések Lagrange-szorzók alkalmazásával adhatóak hozzá. A skálázási (méretarányos) többértelműség és a két részvektor egyenlősége két külön multiplikátort igényel.

A méretarányos kétértelműség azonban az egyik koordináta $\mathbf{x}$ vektorban való rögzítésével kezelhető. Például az utolsó koordinátát a következőképpen rögzíthetjük: $\mathbf{x} = [\gamma \quad \delta \quad \epsilon \quad 1]^T$. Ez a paraméterezés azonban instabil, ha az utolsó koordináta – ami $\sin(\alpha+\beta)$-nak felel meg –, közel van a nullához. Ebben az esetben egy másik, nagyon hasonló paraméterezést kell alkalmazni, a $\mathbf{x}$ harmadik koordinátáját kell eggyé tenni: $\mathbf{x} = [\gamma \quad \delta \quad 1 \quad \epsilon]^T$. Itt az előbbi esetet tárgyaljuk, a másik paraméterezés egyszerűen kezelhető, hiszen a két eset egymásnak megfeleltethető az $\mathbf{A}$ együtthatómátrix harmadik és negyedik oszlopának felcserélésével.

Jelöljük az $\mathbf{A}$ mátrix i-edik sorát $\mathbf{a}_i$ vektorral, és fogalmazzuk meg a problémát az új paraméterekkel egy J költségfüggvény minimalizálásaként, amely a következőképpen írható fel:

$$J = \|\gamma\mathbf{a_1} + \delta\mathbf{a_2} + \epsilon\mathbf{a_3} + \mathbf{a_4}\|_2^2 . \tag{115}$$

A 2D alvektorok hosszára vonatkozó kényszer (azaz $\gamma^2 + \delta^2 = \epsilon^2 + 1$) érvényesítése érdekében csak egyetlen Lagrange-szorzót vezetünk be, és a költségfüggvényt a következőképpen módosítjuk:

$$\widehat{J} = \|\gamma\mathbf{a_1} + \delta\mathbf{a_2} + \epsilon\mathbf{a_3} + \mathbf{a_4}\|_2^2 + \lambda\left(\gamma^2 + \delta^2 - \epsilon^2 - 1\right).$$

A minimumot az ismeretlen paraméterek (γ, δ és ϵ) deriváltjaival kapjuk a következőképpen:

$$\begin{aligned}
\frac{\partial\widehat{J}}{\partial\gamma} &= 2(\gamma\mathbf{a}_1^{\mathrm{T}} + \delta\mathbf{a}_2^{\mathrm{T}} + \epsilon\mathbf{a}_3^{\mathrm{T}} + \mathbf{a}_4^{\mathrm{T}})\mathbf{a}_1 + 2\lambda\gamma &&= 0,\\
\frac{\partial\widehat{J}}{\partial\delta} &= 2(\gamma\mathbf{a}_1^{\mathrm{T}} + \delta\mathbf{a}_2^{\mathrm{T}} + \epsilon\mathbf{a}_3^{\mathrm{T}} + \mathbf{a}_4^{\mathrm{T}})\mathbf{a}_2 + 2\lambda\delta &&= 0,\\
\frac{\partial\widehat{J}}{\partial\epsilon} &= 2(\gamma\mathbf{a}_1^{\mathrm{T}} + \delta\mathbf{a}_2^{\mathrm{T}} + \epsilon\mathbf{a}_3^{\mathrm{T}} + \mathbf{a}_4^{\mathrm{T}})\mathbf{a}_3 - 2\lambda\epsilon &&= 0.
\end{aligned}$$

Mátrixos alakban mindezt így írhatjuk:

$$\underbrace{\begin{bmatrix} \lambda + \mathbf{a}_1^{\mathrm{T}}\mathbf{a}_1 & \mathbf{a}_1^{\mathrm{T}}\mathbf{a}_2 & \mathbf{a}_1^{\mathrm{T}}\mathbf{a}_3 \\ \mathbf{a}_2^{\mathrm{T}}\mathbf{a}_1 & \lambda + \mathbf{a}_2^{\mathrm{T}}\mathbf{a}_2 & \mathbf{a}_2^{\mathrm{T}}\mathbf{a}_3 \\ \mathbf{a}_3^{\mathrm{T}}\mathbf{a}_1 & \mathbf{a}_3^{\mathrm{T}}\mathbf{a}_2 & \mathbf{a}_3^{\mathrm{T}}\mathbf{a}_3 - \lambda \end{bmatrix}}_{\mathbf{M}(\lambda)} \underbrace{\begin{bmatrix} \gamma \\ \delta \\ \epsilon \end{bmatrix}}_{\mathbf{x}} = \underbrace{-\begin{bmatrix} \mathbf{a}_1^{\mathrm{T}} \\ \mathbf{a}_2^{\mathrm{T}} \\ \mathbf{a}_3^{\mathrm{T}} \end{bmatrix} \mathbf{a}_4}_{\mathbf{b}}.$$

Ez az alak még kompaktabban a következő homogén lineáris egyenletrendszer segítségével adható meg: $\mathbf{M}(\lambda)\,\mathbf{x} = \mathbf{b}$. Az $\mathbf{x}$ vektor elemeit úgy kaphatjuk meg, hogy ezt a rendszert megszorozzuk az $\mathbf{M}^{-1}(\lambda)$ inverz mátrixszal[52]. Az inverzet magát pedig felírhatjuk az adjungált mátrixnak és a determinánsnak a hányadosaként:

$$\begin{bmatrix} \gamma \\ \delta \\ \epsilon \end{bmatrix} = -\frac{\mathrm{adj}(\mathbf{M}(\lambda))}{\det(\mathbf{M}(\lambda))} \begin{bmatrix} \mathbf{a}_1^{\mathrm{T}} \\ \mathbf{a}_2^{\mathrm{T}} \\ \mathbf{a}_3^{\mathrm{T}} \end{bmatrix} \mathbf{a}_4. \tag{116}$$

Az adjungált mátrixot a következőképpen fejthetjük ki:

$$\mathrm{adj}(\mathbf{M}) = \begin{bmatrix} \left(\lambda+\mathbf{a}_2^{\mathrm{T}}\mathbf{a}_2\right)\left(\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_3-\lambda\right)-\left(\mathbf{a}_2^{\mathrm{T}}\mathbf{a}_3\right)^2 & \mathbf{a}_2^{\mathrm{T}}\mathbf{a}_3\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_1-\mathbf{a}_2^{\mathrm{T}}\mathbf{a}_1\left(\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_3-\lambda\right) & \mathbf{a}_2^{\mathrm{T}}\mathbf{a}_1\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_2-\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_1\left(\lambda+\mathbf{a}_2^{\mathrm{T}}\mathbf{a}_2\right) \\ \mathbf{a}_1^{\mathrm{T}}\mathbf{a}_3\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_2-\mathbf{a}_2^{\mathrm{T}}\mathbf{a}_1\left(\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_3-\lambda\right) & \left(\lambda+\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_1\right)\left(\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_3-\lambda\right)-\left(\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_3\right)^2 & \mathbf{a}_1^{\mathrm{T}}\mathbf{a}_2\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_1-\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_1\left(\lambda+\mathbf{a}_1^{\mathrm{T}}a_1\right) \\ \mathbf{a}_2^{\mathrm{T}}\mathbf{a}_3\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_1-\mathbf{a}_2^{\mathrm{T}}\mathbf{a}_1\left(\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_3-\lambda\right) & \mathbf{a}_1^{\mathrm{T}}\mathbf{a}_2\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_1-\mathbf{a}_3^{\mathrm{T}}\mathbf{a}_1\left(\lambda+\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_1\right) & \left(\lambda+\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_1\right)\left(\lambda+\mathbf{a}_2^{\mathrm{T}}\mathbf{a}_2\right)-\left(\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_2\right)^2 \end{bmatrix}. \tag{117}$$

$\mathrm{adj}(\mathbf{M})$ elemeit a λ polinomjai alkotják, a polinomok fokszáma egy vagy kettő. Az $\mathbf{M}$ mátrix determinánsa pedig egy harmadfokú polinom. Így minden eleme kifejezhető egy kvadratikus és egy köbös polinom hányadosaként a következő formában:

$$\gamma = \mathbf{P}_1^2(\lambda)/\mathbf{P}_4^3(\lambda), \quad \delta = \mathbf{P}_2^2(\lambda)/\mathbf{P}_4^3(\lambda), \quad \epsilon = \mathbf{P}_3^2(\lambda)/\mathbf{P}_4^3(\lambda).$$

A $\mathbf{P}_1^2$, $\mathbf{P}_2^2$ és $\mathbf{P}_3^2$ polinomokat az $\mathrm{adj}(\mathbf{M}(\lambda))$ adjungált mátrix első, második és harmadik sorának az $\begin{bmatrix} \mathbf{a}_4^{\mathrm{T}}\mathbf{a}_1 & \mathbf{a}_4^{\mathrm{T}}\mathbf{a}_2 & \mathbf{a}_4^{\mathrm{T}}\mathbf{a}_3 \end{bmatrix}^{\mathrm{T}}$ vektorral való szorzásával kapjuk. $\mathbf{P}_4^3$ pedig az $\mathbf{M}(\lambda)$ mátrix determinánsa.

A Lagrange-szorzó meghatározásához a $\gamma^2+\delta^2-\epsilon^2-1=0$ feltételt kell figyelembe venni. Ez egy hatodfokú λ polinomot eredményez a következőképpen:

$$\left(\mathbf{P}_1^2(\lambda)\right)^2 + \left(\mathbf{P}_2^2(\lambda)\right)^2 - \left(\mathbf{P}_3^2(\lambda)\right)^2 - \left(\mathbf{P}_4^3(\lambda)\right)^2 = 0.$$

Ennek a polinomnak legfeljebb hat különböző gyöke van. Csak a valós paramétereket kell megtartani, a komplex gyököket elvethetjük. A γ, δ és ϵ paraméterekre vonatkozó megoldásokat úgy kapjuk meg, hogy a λ becsült valós gyökeit behelyettesítjük a 116. egyenletbe. A polinomok megoldásához az Eigen könyvtárat[53] alkalmazzuk, ez a könyvtár kiforrott, emiatt másnak is jó szívvel ajánlom a használatát algebrai alkalmazásokhoz.

Degenerált esetek. A javasolt algoritmus a két szöget egy lineáris egyenletrendszer segítségével becsüli meg, amely legalább három független egyenletből áll. Az együtthatómátrix két részre osztható, a bal oldali két oszlop a α szöghöz, a harmadik és negyedik az $(\alpha+\beta)$ szöghöz tartozik. Algebrailag a konfiguráció csak akkor degenerált, ha az egyik rész csak nulla elemekből áll. Ha az egyenleteink pontmegfelelésből származnak, akkor ez csak abban az esetben lehetséges, ha az összes pontmegfelelés második koordinátája nulla, azaz a pontok rajta vannak a horizonton.

[52] A mátrix invertálható, kivéve nagyon speciális degenerált eseteket.

[53] Eigen Library: `https://eigen.tuxfamily.org/index.php?title=Main_Page`

L. Levenberg-Marquardt algoritmus fotomoterikus sztereóhoz

A következő *Matlab* kód kiszámítása *Symbolic Toolbox* használatával generálja a a mátrixokat a fotometrikus sztereó alapú Levenberg-Marquardt algoritmushoz. Fontos megjegyzés, hogy a *ccode* parancs segítségével közvetlenül is generálhatunk *C/C++* forráskódot a $\mathbf{U}$, $\mathbf{V}$ vagy $\mathbf{X}$ szimbólumokból.

```
function [J, JtJ, U, V, X] = get_mats()

        bp1 = sym('bp1');
        bp2 = sym('bp2');
        bp3 = sym('bp3');
        x1 = sym('x1');
        x2 = sym('x2');
        x3 = sym('x3');
        li1 = sym('li1');
        li2 = sym('li2');
        li3 = sym('li3');
        k = sym('k');
        lambda = sym('lambda');

        L = [li1; li2; li3];
        S = [x1; x2; x3];
        N = [bp1; bp2; bp3];
        R = L - S;
        mag_r2 = R(1)^2 + R(2)^2 + R(3)^2;
        mag_r = sqrt(mag_r2);
        prodrn = R(1) * N(1) + R(2) * N(2) + R(3) * N(3);

        f = k * lambda / (lambda + mag_r2) * prodrn / mag_r;

        J = jacobian(f, [li1, li2, li3, k, bp1, bp2, bp3]);
        JtJ = J' * J;

        U = JtJ(1:3, 1:3);
        V = JtJ(4:7, 4:7);
        X = JtJ(1:3, 4:7)
end
```

M. Kamera válaszfüggvényének becslése

Kísérleteink során megállapítottuk, hogy a pontszerű fényforrások által megvilágított tárgyak normálvektorainak becsléséhez alkalmazott számítás meglehetősen érzékeny a kamera válaszfüggvényére. A válaszfüggvényt az határozza meg, hogy a megvilágítás sugárzása hogyan képeződik le a pixelekre. Számos hatékony becslés létezik a számítógépes grafikai szakirodalomban: mi a Debevec és mtsai. módszerét [37] választottuk egyszerűsége miatt. A módszert itt röviden ismertetem.

Egy referenciajelenet lefényképezése több $\Delta t_j, j = 1 \dots F$ expozíciós idővel történik. A pixelek intenzitásértékeit P_i^j jelöli, ahol az i index a pixel sorszáma a képkockán belül.

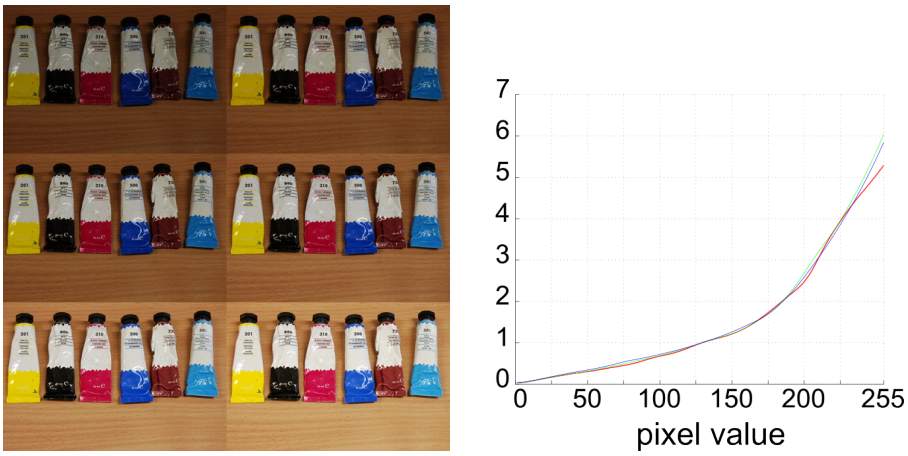
A konkrét intenzitásérték az expozíciós idő és a háromdimenziós jelenetből származó sugárzásnak a függvénye:

$$P_i^j = \mathrm{crf}(R_i \Delta t_j), \tag{118}$$

ahol R_i a sugárzás, és $\mathrm{crf}()$ a kamera válaszfüggvénye.

A cél az alábbi kvadratikus célfüggvény minimalizálása:

$$\sum_{i=1}^{P} \sum_{j=1}^{F} \left[g(P_i^j) - \ln R_i - \ln \Delta t_j \right]^2 + \alpha \sum_{z=0}^{255} g''(z), \tag{119}$$



58. ábra. *Balra: Referenciaképek Nikon™D50 kamerával, különböző expozíciós idővel. Jobbra: A kamera becsült inverz válaszfüggvényei a három színcsatornára.*

ahol $g = \ln\left(\mathrm{crf}^{-1}\right)$ az inverz kamera válaszfüggvénynek a logaritmusa[54]. A g függvény értelmezési tartománya a $[0 \dots 255]$ intervallum, tehát 256 (kvantált) értékkel ábrázolható. Az α egy simasági állandó, így a mimimalizálandó függvény második tagja elősegíti, hogy a függvény szomszédos értékei között ne legyen nagy ugrás. A sugárzások $\ln R_i$ logaritmusai szintén ismeretlenek. Ha az inverz kamera második válaszfüggvény deriváltját a diszkrét értékekből a következőképpen közelítjük: $g''(x) = = g(x-1) - 2g(x) + g(x+1)$, akkor a probléma lineáris lesz az ismeretlen változókra

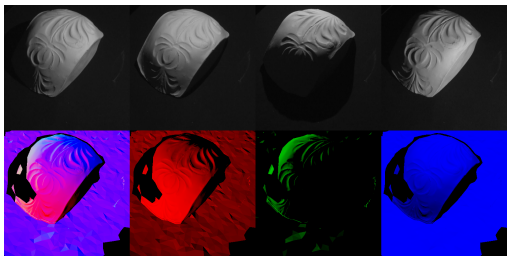
[54] CRF: Camera Response Function.

vonatkoztatva, és a megoldás az együtthatómátrix szinguláris értékű felbontásával adható meg.

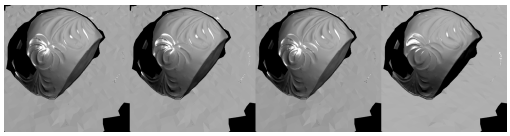
Kiszámítottuk a crf értékét a mi Nikon™D50 digitális fényképezőgépünk vörös, zöld és kék csatornáira. A kiszámított válaszfüggvények és a referenciajelenetünk képeit az 58. ábra szemlélteti. Vizsgálatainkban az α paraméter értékét – tapasztalataink alapján – 100-ra állítottuk be.

N. További rekonstrukciós eredmények fotometrikus sztereó segítségével

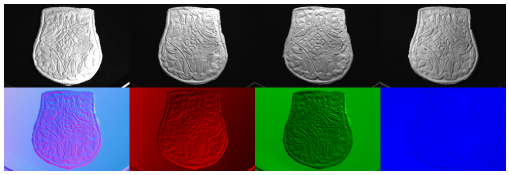
A függeléknek ebben a fejezetében a fotometrikus sztereó algoritmussal kapott eredményeket tettem be, mindkét esetben valós objektumok normáltérképeit rekonstruáltuk. A fő szövegben helyhiány miatt csak néhány eredmény beillesztésére volt lehetőség, reményeim szerint az Olvasó is szívesen böngészi az eredményeket.



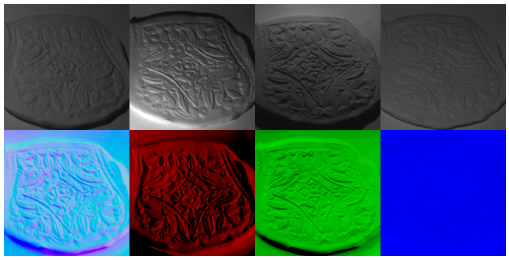
59. ábra. *Felül: négy példa a húsz bemeneti képből. Alul: a normáltérkép RGB színekkel kódolva.*



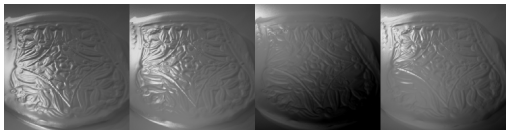
60. ábra. *Mesterségesen készített (renderelt) képek az 59. ábrán látható tárgy rekonstruált normáltérképének segítségével.*



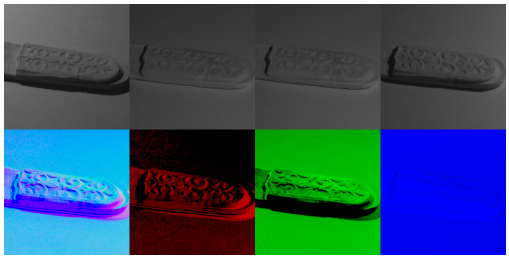
61. ábra. *Felül: négy példa a nyolc bemeneti képből. Alul: a normáltérkép RGB színekkel kódolva.*



62. ábra. *Felül: négy példa a nyolc bemeneti képből. Alul: a normáltérkép RGB színekkel kódolva.*



63. ábra. *Mesterségesen készített (renderelt) képek a 62. ábrán látható tárgy rekonstruált normáltérképének segítségével.*



64. ábra. *Felül: négy példa a nyolc bemeneti képből. Alul: a normáltérkép RGB színekkel kódolva.*

65. ábra. *Mesterségesen készített (renderelt) képek a 64. ábrán látható tárgy rekonstruált normáltérképének segítségével.*