

Skálázottan merőleges kamera optimális kalibrációja

Hajder Levente

MTA SZTAKI

Geometria Modellezés és Számítógépes Látás Laboratórium

hajder@sztaki.hu

Absztrakt. A kamera kalibráció a háromdimenziós számítógépes látás egyik fontos alapproblémája, hiszen pontosan kalibrált kamerák nélkül pontos háromdimenziós becsléseket sem lehet végezni. Kamerakalibrációra számtalan jól működő algoritmus létezik, ezek többsége perspektív kamerákkal foglalkozik, és kezdeti becslés után numerikus optimalizálással javítják a kalibráció minőségét. Sajnos nem bizonyítható, hogy ezek az algoritmusok megtalálják az optimumot. Ebben a tanulmányban azt mutatjuk meg, hogy korábban kifejlesztett, gyengén perspektív kamerakalibrációs algoritmusunk segítségével az optimális megoldás elérhető.

1. Bevezetés

Optimális algoritmusok kifejlesztése soknézőpontos rekonstrukciókhoz [5] kihívásokkal teli feladat. Ez a cikk a számítógépes látás egyik alapproblémájával, a kamerakalibrálással foglalkozik. A téma nem új, a 80-as évektől számos kiváló kutató publikált kalibráló módszereket. Perspektív kamerára számtalan jól működő algoritmust [4, 16] javasoltak. Ezek a módszerek először egy közelítő megoldást becsülnek, majd numerikus optimalizálással [9, 10] javítják a kapott közelítést. Sajnos az algoritmusok nem garantálják, hogy megtalálják a globális optimumot. Abban az esetben, ha a kamera belső paraméterei már ismertek, konvex optimalizálás segítségével a külső paramétereket meg lehet optimálisan találni, ahogyan azt Olsson és Mtsai [12] megmutatták.

Sokkal egyszerűbb a helyzet, ha affin kamerákat alkalmazunk, hiszen ebben az esetben a probléma lineáris [14]. Gyengén perspektív [2] és paraperspektív [6] kamerák kalibrációjával is foglalkoztak, azonban ezek a módszerek sem optimálisak, mert arra fókuszálnak, hogyan lehet a valódi perspektív és az egyszerűsített kameramodellek közötti átjárást megvalósítani. Ezek a tanulmányok az optimalitással nem foglalkoznak.

Jelen cikk szerzője szintén foglalkozott kamerakalibrációval. 2010-ben sikerült megmutatni, hogy a gyengén perspektív kamera optimális kalibrációja [3] egy tízedfokú polinom segítségével megoldható. Korábban javasoltunk rekonstrukciós módszert skálázottan ortonormált kamerára [13] is. Ebben a tanulmányban azt mutatjuk be, hogy a rekonstrukció során alkalmazott ötlet segítségével a kamera kalibrációját el lehet végezni, és az így kapott iteratív kalibráló algoritmus legkisebb négyzetes értelemben optimális megoldást ad.

2. A feladat megfogalmazása

Amennyiben adott egy térbeli pont, és a 2D-s vetülete a képen, a kalibrációs algoritmusok célja a $3D \rightarrow 2D$ vetítés paramétereinek megtalálása. A kalibráló tárgyunk i -edik pontjának koordinátáit X_i , Y_i és Z_i jelöli, a vetítés utáni koordinátákat pedig u_i és v_i .

Perspektív kamera esetén a vetítést a jól ismert összefüggés segítségével kapjuk meg:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \sim C[R|T] \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}, \quad (1)$$

ahol R az ortonormált forgatási mátrix, T az eltolási vektor a kamera fókuszpontja és a világ koordinátarendszerének középpontja között. (R -et és T -t a kamera külső paramétereinek szokás hívni.) A " \sim " jel azt jelenti, hogy az egyenlőség teljesüléséhez az egyik oldalt egy skalárral meg kell szorozni. A kamera belső paramétereit a C -vel jelölt felső háromszög mátrixban találhatók. [4].

A gyengén perspektív kamera akkor alkalmazható, ha a tárgy mélysége jóval kisebb, mint a fókuszpont és a tárgy pontjainak a távolsága. Ebben az esetben a perspektív vetítést az alábbi összefüggéssel közelíthetjük:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = [M|t] \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}, \quad (2)$$

ahol M -et mozgási mátrixnak hívják, t a kétdimenziós eltolási vektor. Az eltolási vektor a világ koordinátarendszerének vetületét adja meg a képen. M mozgási mátrix két sora háromdimenziós sorvektorokat tartalmaz, melyeket m_1^T -tal és m_2^T -tal jelölünk. Gyengén perspektív kamera esetén ez a két sor merőleges egymásra: $m_1^T m_2 = 0$. Ebben a cikkben skálázottan ortonormált kamerával foglalkozunk. Ez a gyengén perspektív kamerának egy speciális esete: nemcsak merőleges a két sorvektor, hanem a hosszuk is megegyezik: $m_1^T m_1 = m_2^T m_2$.

A kamerakalibráció során a célunk az összes pontra a következőképpen felírható visszavetítési hibát minimalizálni:

$$\|W - MS\|^2, \quad (3)$$

ahol $\|\cdot\|^2$ az ún. Frobenius-norma négyzetét (mátrix elemeinek négyzetösszegét) jelöli, S mátrix tartalmazza a háromdimenziós pontokat, W mátrix pedig a megfelelő vetületek kétdimenziós koordinátáit:

$$S = \begin{bmatrix} X_1 & X_2 & \dots & X_P \\ Y_1 & Y_2 & \dots & Y_P \\ Z_1 & Z_2 & \dots & Z_P \end{bmatrix}, \quad (4)$$

$$W = \begin{bmatrix} u_1 & u_2 & \dots & u_P \\ v_1 & v_2 & \dots & v_P \end{bmatrix}. \quad (5)$$

(P -vel jelöljük a pontok számát.)

3. Skálázottan ortonormált kamera kalibrációja

Ahogy azt megmutattuk [3], a gyengén perspektív kalibráció Lagrange-multiplikátoros optimalizálással egy tízedfokú polinom zérushelyeinek keresésére vezethető vissza. Skálázottan ortonormált kalibráció esetén ez a megoldás sajnos nem vezet eredményre, hiszen két megkötésünk is van, szemben a gyenge perspektíva egy megkötésével. A mátrixinverziók miatt a behelyettesítés így lehetetlenné válik.

Ezért alkalmazzuk a következő trükköt, ahogyan azt már korábban is javasoltuk [13]: egészítsük ki a kétdimenziós koordinátákat és az M mozgási mátrixot egy harmadik sorral:

$$M = \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix}. \quad (6)$$

Ez a harmadik sorvektor legyen az első két vektorra merőleges, hossza pedig azonos az első kettőével. Ebben az esetben az M mozgásmátrixot fel tudjuk írni egy ortonormált R mátrix és egy q valós szám szorzataként: $M = qR$.

Ha az M mozgásmátrix már ki lett egészítve a harmadik sorral, a W mérési mátrixot is ki lehet egészíteni:

$$W = \begin{bmatrix} w_1^T \\ w_2^T \\ w_3^T \end{bmatrix}, \quad (7)$$

ahol $w_1^T = [u_1 u_2 \dots u_F]^T$ és $w_2^T = [u_1 u_2 \dots u_F]^T$. A harmadik sort, ha már kiegészítettük M -et, könnyen ki lehet számolni: $w_3^T = m_3^T S$.

A kalibráló algoritmusunk során a cél kiszámolni R ortonormált mátrixot és q valós számmal ábrázolt skálázást, ha ismerjük S és W mátrixot, más szóval van egy referenciaobjektumunk pontokkal, és ismerjük annak vetületeit is.

Az optimális kalibráló algoritmus egy iteráció, aminek a vázát az 1. algoritmuson láthatjuk. Az iteráció mindössze két lépést tartalmaz, az első kiegészíti az M és W mátrixokat, a másik lépés pedig egy 3D→3D regisztrációt alkalmaz. Azért nevezzük regisztrációnak, mert kiegészített W mérési mátrix esetén az S struktúra mátrixban és W -ben ugyanannak a pontnak a háromdimenziós koordinátái vannak, egymáshoz képest skálázva és elforgatva. Ha a skálázást nem vesszük figyelembe, a probléma a 3D-s elforgatás megtalálása. Ezt hívják a geometriában pontthalmazok regisztrációjának. A regisztrációs probléma optimálisan megoldható, a tökéletes megoldást több eltérő módszerrel meg lehet találni [1, 7, 8]. Ahogy azt megmutattuk korábban [13], az elforgatás független a skálázástól. Ha pedig az elforgatás már ismert, a skálázás maga lineáris becsléssel meghatározható [13].

Ha pontosítottuk a regisztrációs lépéssel a q skálázást és az R ortonormált mátrixot, a W mérési mátrix kiegészítését újra és újra el kell végezni. A visszavetítési hibát mind a regisztrációs, mind a kiegészítő lépés csökkenti, vagy legalábbis nem növeli. Mivel a visszavetítési hiba nemnegatív, és csak csökkenni tud, ezért biztosak lehetünk abban, hogy az eljárás konvergál.

Algorithm 1 A skálázottan ortonormált kamera kalibrációjának váza.

```

repeat
   $w_3 \leftarrow \text{Kiegészítés}(R, t, S, q)$ 
   $R, t, q \leftarrow \text{Regisztráció}(S, w_1, w_2, w_3)$ 
until konvergencia.
  
```

4. A lokális optimum problémája

Ahogy azt már korábban említettük, az előző fejezetben ismertetett kalibrációs algoritmus a globális optimumot megtalálja. Ebben a fejezetben azt mutatjuk meg, hogy a módszer törvényszerűen a globális optimumhoz konvergál, hiszen lokális minimum nem létezik.

A bizonyítást először egyszerűsítéssel végezzük el: ha a 3D→2D vetítés helyett 2D→1D vetítéssel képzeljük el a problémát, minden igaz marad, azt leszámítva, hogy a mozgásmátrix csak egy sort tartalmaz, kiegészítve kettőt. A bizonyítás ebben az esetben könnyebben megérthető, mivel kétdimenzióban a forgatási mátrix leírható egy paraméter (például a forgatási szög) segítségével.

4.1. 2D → 1D vetítés

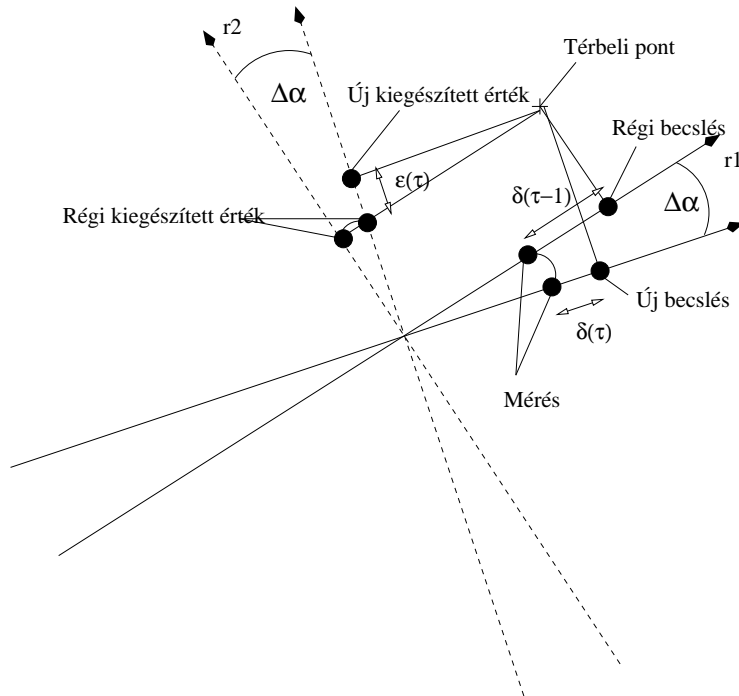
Ez egyszerűsített esetben a visszavetítési hiba így néz ki:

$$\sum_{i=1}^N \left\| u_i - m_{2D} \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \right\|^2, \quad (8)$$

ahol $m_{2D} = qr_1^T$ egy sorvektor, amelyiket ki lehet egészíteni az r_2 vektorral. r_2 merőleges r_1 -re, mindkettő egységvektor. A mért u_i koordinátát szintén ki lehet egészíteni a második koordinátával: $v_i = qr_2[X_i, Y_i]^T$. Az optimális regisztráció például Arun és mtsai. módszerével [1] kétdimenzióban is meghatározható optimálisan.

Mint ahogyan azt már láthattuk, két önmagában optimális lépésből áll a kalibráló algoritmus. A "csalás" ott lép be, hogy a W mérési mátrixot kiegészítjük új (második) koordinátákkal, amit a regisztráció során pontosan ugyanúgy veszünk figyelembe, mint a mért u_i koordinátákat. Attól tartunk, hogy ez a kiegészítés eltéríti a jó megoldástól az algoritmust. Ez abban az esetben lehetséges, ha a forgatás/skálázás esetén az első sor négyzetes hibájának a deriváltja abszolútértékben kisebb, mint a második soré, azaz azt a javulást, amit az első sor okoz, a második sor képes visszahúzni. Könnyű belátni, hogy a skálázás ezt nem befolyásolja, hiszen mindkét sor ugyanúgy meg van szorozva q -val. Az elforgatás kérdése már bonyolultabb. Szerencsére tudjuk, hogy a módszerünk konvergál valahova. Amikor már nagyon-nagyon közel jár a minimumhoz, tudjuk, hogy nagyon kicsi szöggel is felírhatjuk a forgatást. A forgatási mátrix így írható fel alapesetben:

$$R = \begin{bmatrix} \cos \Delta\alpha & \sin \Delta\alpha \\ -\sin \Delta\alpha & \cos \Delta\alpha \end{bmatrix}. \quad (9)$$



1. ábra: A regisztráció és a kiegészítés hibájának változása szomszédos iterációk között

Ez nagyon kicsi $\Delta\alpha$ esetén a jól ismert szögfüggvényes közelítésekkel átírható:

$$R \approx \begin{bmatrix} 1 & \Delta\alpha \\ -\Delta\alpha & 1 \end{bmatrix}. \quad (10)$$

A kiegészítés akkor állítja meg a konvergenciát lokális minimumban, ha a regisztráció javulásának a deriváltja abszolútértékben kisebb, mint a kiegészítés romlása. (A kiegészítés mindenképpen romlik, hiszen minden egyes kiegészítő lépés nullára csökkenti a hibát a kiegészített (harmadik) koordinátákban.)

Az 1. ábrán láthatjuk az iteráció két lépése közötti változást, ha $\Delta\alpha$ -val változik meg a szög. Az R mátrix két sora adja meg a koordinátatenglyeket, ezek forognak a regisztrációs lépés során. Az aktuális iteráció számát τ -val jelöltük az ábrán.

Jelöljük ϵ_i -vel a regisztráció javulását, δ_i -vel a kiegészítés hibájának növekedését a regisztráció után.

A regisztráló lépésben a kapott forgatásból jövő koordinátaváltozás könnyen kifejezhető:

$$\begin{bmatrix} 1 & \Delta\alpha \\ -\Delta\alpha & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \end{bmatrix} - \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \quad (11)$$

$$= \begin{bmatrix} 0 & \Delta\alpha \\ -\Delta\alpha & 0 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \quad (12)$$

$$= \Delta\alpha \begin{bmatrix} Y_i \\ -X_i \end{bmatrix}. \quad (13)$$

A regisztráció hibája az elforgatott koordináta és a mért érték különbsége:

$$\epsilon_i = u_i - [1 \ \Delta\alpha] \begin{bmatrix} X_i \\ Y_i \end{bmatrix} = u_i - X_i - \Delta\alpha Y_i, \quad (14)$$

a kiegészítésből származó hiba pedig a 13. összefüggés második koordinátája: $\delta_i = -\Delta\alpha X_i$.

Az algoritmus akkor ragad benne a rossz pozícióban, ha a szögváltozás szerinti derivált abszolútértéke nagyobb a kiegészítés hibájára, mint a regisztráció javulására nézve:

$$\left| \frac{\partial \sum_{i=1}^N e_i^2}{\partial \Delta\alpha} \right| < \left| \frac{\partial \sum_{i=1}^N \delta_i^2}{\partial \Delta\alpha} \right|. \quad (15)$$

Tudjuk, hogy $\frac{\partial \sum_{i=1}^N \delta_i^2}{\partial \Delta\alpha} \rightarrow 0$ ha $\Delta\alpha \rightarrow 0$, hiszen $\frac{\partial \delta_i^2}{\partial \Delta\alpha} = \Delta\alpha X_i^2$. Másrészt az is igaz, hogy $\frac{\partial \sum_{i=1}^N e_i^2}{\partial \Delta\alpha} > 0$ minden esetben, kivéve, amikor $u_i - X_i = 0 \ \forall i$, mivel $\frac{\partial e_i^2}{\partial \Delta\alpha} = 2(u_i - X_i - \Delta\alpha Y_i)Y_i$. Ez azt jelenti, hogy a regisztráció javító hatása minden esetben erősebb, mint a kiegészítés koordinátájában okozott hiba, kivéve, ha $u_i - X_i = 0 \ \forall i$. Ez utóbbi eset azonban azt jelenti, hogy a regisztráció tökéletes, már korábban elértük a minimumot. (Ráadásul ez csak abban a kivételes esetben lehetséges, ha az $[u_i, v_i]^T$ koordinátákban semmi zaj nincsen, nulla hibával regisztrálni kalibrálni lehet a kamerát, Ilyen gyakorlatilag sosem fordul elő.)

Tehát nem tud az algoritmus lokális minimumba futni, akárhonnán indítjuk, ugyanazt a megoldást kapjuk.

4.2. 3D \rightarrow 2D vetítés

Ebben az esetben is feltételezhetjük, hogy már közel vagyunk a megoldáshoz. Kis változás esetén a forgatási mátrix háromdimenzióban is leírható kis szög és egy tengely segítségével. (A forgatási mátrixnak három szabadsági foka van, a tengely kettőt, a szög egyet ad.) Tegyük fel, hogy a forgatás a Z tengely körül történik. Ebben az esetben a kis $\Delta\alpha$ szöggel forgatás így írható le:

$$R' \approx \begin{bmatrix} 1 & \Delta\alpha & 0 \\ -\Delta\alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (16)$$

Általános esetben nem igaz, hogy a Z tengely körül forgatunk, ezért az eredeti koordinátarendszerből el kell úgy forgatnunk a pontokat, hogy ez igaz legyen. Ehhez egy $Q = [q_1, q_2, q_3]^T$ háromdimenziós transzformációt vezetünk be úgy, hogy igaz legyen: $R = QR'$. A regisztrációs lépésre a 2D→1D esethez hasonlóan igaznak kell lennie, hogy a regisztráció javulását nem képes elrontani a kiegészített koordinátákból származó hiba.

Az i . pontra jutó kiegészítési hibát jelöljük továbbra is δ_i -vel. Így tudjuk meghatározni:

$$\delta_i = q_3^T \left(\begin{bmatrix} 1 & \Delta\alpha & 0 \\ -\Delta\alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - I_{3 \times 3} \right) \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \quad (17)$$

$$= \Delta\alpha q_3^T \begin{bmatrix} Y_i \\ -X_i \\ Z_i \end{bmatrix}. \quad (18)$$

A regisztrációs hiba két koordináta javulásából adódik. Az első koordinátáé így néz ki:

$$\epsilon_i = u_i - q_1^T \begin{bmatrix} 1 & \Delta\alpha & 0 \\ -\Delta\alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \quad (19)$$

$$= u_i - q_1^T \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} - \Delta\alpha q_1^T \begin{bmatrix} Y_i \\ -X_i \\ Z_i \end{bmatrix}. \quad (20)$$

A második koordináta értelemszerűen hasonló:

$$\gamma_i = v_i - q_2^T \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} - \Delta\alpha q_2^T \begin{bmatrix} Y_i \\ -X_i \\ Z_i \end{bmatrix}. \quad (21)$$

A lokális minimum abba az esetben kerülhető el, ha mindig igaz, hogy

$$\left| \frac{\partial \sum_{i=1}^N \epsilon_i^2}{\partial \Delta\alpha} + \frac{\partial \sum_{i=1}^N \gamma_i^2}{\partial \Delta\alpha} \right| < \left| \frac{\partial \sum_{i=1}^N \delta_i^2}{\partial \Delta\alpha} \right|. \quad (22)$$

Ez pedig igaz, kivéve ha $u_i = q_1^T \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$, és $v_i = q_2^T \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$ minden i -re.

Ez pedig csak akkor igaz, ha már tökéletesen egymáshoz vannak regisztrálva a ponthalmazok, azaz már a globális minimumban vagyunk.

Tehát kijelenthetjük, hogy **a kalibrációs algoritmus mindig ugyanoda, a globális minimumhoz konvergál.**

5. Eredmények

Gyengén perspektív kamerakalibrációt önmagában ritkán szoktak alkalmazni, mivel rögzített kamerák esetén célszerűbb a valóságot jobban közelítő perspektív kalibrációt használni. Mozgó tárgyak rekonstrukciójánál azonban nagyon hasznos a gyengén perspektív kamera. Mi is azért fejlesztettük ki a jelen tanulmányban ismertetett optimális kalibrációs algoritmust, hogy mozgó tárgyak követett pontjainak háromdimenziós rekonstrukcióját végezzük. Ez akkor lehetséges, ha közben magának a kamerának a paramétereit is kiszámítjuk. (Ezt a folyamatot nevezik a kamera autokalibrációnak).

Ezért ebben a cikkben tesztelési eredmények rekonstrukciós eredményeket mutatunk, noha a kalibráló módszerünk optimalizását szintetikus adatokon kipróbáltuk: ugyanarra a mesterségesen generált 3D-s és 2D-s megfeleltetett pontthalmazra különböző induló paraméterekkel lefuttattuk az algoritmust. Kivétel nélkül ugyanahhoz az eredményhez konvergált a módszer. Ha zajmentes adatokkal futtattuk, szintén kivétel nélkül a helyes megoldáshoz konvergált a módszer, akárhonnan is indítottuk.

Mivel a javasolt rekonstrukciós módszer megegyezik a korábban publikált algoritmusunkkal [13], itt csak valós képsorozatokon futtatott rekonstrukciós eredményeket mutatunk meg. Szintetikus eredményeket és konkurens módszerekkel való összehasonlításokat a korábbi publikációnkban [13] található meg a kedves Olvasó.

”Arc” tesztorozat. Első valós tesztünk egy mereven tartott arc rekonstrukciója. A kiindulási képeket a 2. ábrán láthatjuk. A képeken automatikusan követtük a pontokat az AAM [11] (Active Appearance Model) segítségével. Számszerűen 44 pontot követtünk 331 képkockán keresztül. A rekonstrukció eredménye a 3. ábrán látható. A rekonstrukciós eredményt 1 másodperc alatt számította ki a Core4Quad processzoros (2.33GHz, 4GByte RAM) gépünk.



2. ábra: 4 tesztkép az ”Arc” sorozatból

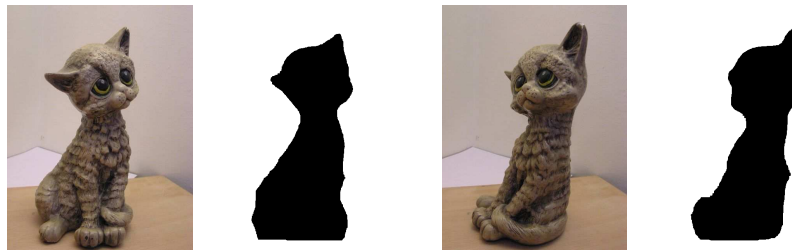
”Dínó” tesztorozat

A forgó műanyag dinoszaurusz követet pontjai az Oxford Egyetem honlapjáról¹ töltöttük le. A rekonstruálandó tárgyról képek az 5. ábrán láthatóak. Összesen 319 pontot és 36 képet tartalmaz a követett pontok listája. Öntakarás miatt nem mindegyik pont látszik, de ez nem okoz problémát a rekonstrukciónak.

¹ <http://www.robots.ox.ac.uk/~amb/>



3. ábra: Az arc rekonstrált modellje



4. ábra: A "Macska" tesztorozat néhány képe, és a hozzájuk tartozó maszkok.

algoritmusnak, amely 36 másodperces futás után adta a 6. ábrán látható eredményt.



5. ábra: 4 tesztkép a "Dínó" sorozatból

"Macska" tesztorozat Az utolsó tesztorozatot magunk készítettük. Egy macskát ábrázoló szobrot megforgattuk az asztalon, összesen 92 fényképet készítve. A háttérrel és a macskát saját módszerünk segítségével elválasztottuk, ahogyan az a 4. ábrán látszik. Ezek után KLT algoritmus [15] segítségével pontokat választottunk ki, és követtük őket. Amennyiben egy pont közel került a háttérhez, eldobtuk. Összesen 2290 pontot követtünk. A rekonstrukció, amely 199 másodperces futás eredményeképpen keletkezett, a 7. ábrán tekinthető meg.



6. ábra: A "Dínó" sorozat rekonstruált háromdimenziós pontjai



7. ábra: A "Macska" sorozat rekonstruált háromdimenziós pontjai

6. Összefoglalás

Ebben a cikkben megmutattuk, hogy a skálázottan ortonormált kamerát hogyan kell iteratív algoritmus segítségével kalibrálni. Bebizonyítottuk, hogy a kidolgozott algoritmus a kalibráció visszavetítési hibáját legkisebb négyzetes értelemben minimalizálja, és a minimum minden esetben globális. Rekonstrukciós példán keresztül igazoltuk, hogy az itt ismertetett algoritmus gyakorlati alkalmazásokba is beépíthető.

Köszönetnyilvánítás

A szerző ezúton szeretne köszönetet mondani Kazó Csabának, Csetverikov Dmitrijnek és Takács Dánielnek, hogy segítettek az eredmények elkészítéséhez nélkülözhetetlen pontkövetések és előtér-háttér szétválasztó módszerek elkészítésében. A munka az NKTH-OTKA CK 78409 pályázat keretében készült.

Irodalom

1. K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Trans. on PAMI*, 9(5):698–700, 1987.
2. Daniel F. DeMenthon and Larry S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995.
3. L. Hajder. L_2 -Optimal Weak-Perspective Camera Calibration. In *Fifth Hungarian Conference on Computer Graphics and Geometrics*, pages 89–96, 2010.
4. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
5. Richard Hartley and Fredrik Kahl. Optimal algorithms in multiview geometry. In *Proceedings of the Asian Conf. Computer Vision*, pages 13–34, 2007.
6. Radu Horaud, Fadi Dornaika, Bart Lamiroy, and Stéphane Christy. Object pose: The link between weak perspective, paraperspective and full perspective. *International Journal of Computer Vision*, 22(2):173–189, 1997.
7. B.K.P. Horn. Closed-form Solution of Absolute Orientation using Unit Quaternions. *Journal of the Optical Society of America*, 4:629–642, 1987.
8. B.K.P. Horn, H.M. Hilden, and S. Negahdaripour. Closed-form Solution of Absolute Orientation Using Orthonormal Matrices. *Journal of the Optical Society of America*, 5(7):1127–1135, 1988.
9. K. Levenberg. A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.
10. D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11:431–441, 1963.
11. Iain Matthews and Simon Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60:135–164, 2003.
12. Carl Olsson, Fredrik Kahl, and Magnus Oskarsson. Optimal estimation of perspective camera pose. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 5–8, 2006.

13. Á. Pernek, L. Hajder, and Cs. Kazó. Metric Reconstruction with Missing Data under Weak-Perspective. In *British Machine Vision Conference*, pages 109–116, 2008.
14. Heung-Yeung Shum, Katsushi Ikeuchi, and Raj Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(9):854–867, 1995.
15. Tomasi, C. and Shi, J. Good Features to Track. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 593–600, 1994.
16. Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. on PAMI*, 22(11):1330–1334, 2000.