



Írta:

**KATÓ ZOLTÁN
CZÚNI LÁSZLÓ**

SZÁMÍTÓGÉPES LÁTÁS

Egyetemi tananyag



TYPOTEX

2011

COPYRIGHT: © 2011–2016, Dr. Kató Zoltán, Szegedi Tudományegyetem
Természettudományi és Informatikai Kar Képfeldolgozás és Számítógépes Grafika Tanszék;
Dr. Czúni László, Pannon Egyetem Műszaki Informatikai Kar Villamosmérnöki és
Információs Rendszerek Tanszék

LEKTORÁLTA: Dr. Hajder Levente, Magyar Tudományos Akadémia Számítástechnikai és
Automatizálási Kutatóintézet

Creative Commons NonCommercial-NoDerivs 3.0 (CC BY-NC-ND 3.0)

A szerző nevének feltüntetése mellett nem kereskedelmi céllal szabadon másolható,
terjeszthető, megjelentethető és előadható, de nem módosítható.

TÁMOGATÁS:

Készült a TÁMOP-4.1.2-08/1/A-2009-0008 számú, „Tananyagfejlesztés mérnök
informatikus, programtervező informatikus és gazdaságinformatikus képzésekhez” című
projekt keretében.



ISBN 978-963-279-512-6

KÉSZÜLT: a **Typotex Kiadó** gondozásában

FELELŐS VEZETŐ: **Votisky Zsuzsa**

AZ ELEKTRONIKUS KIADÁST ELŐKÉSZÍTETTE: **Juhász Lehel**

KULCSSZAVAK:

3D rekonstrukció, kamerakalibráció, sztereó képpár, fundamentális mátrix,
mozgásdetektálás, optikai áramlás, mozgáskövetés.

ÖSSZEFOGLALÁS:

A számítógépes látás az emberi látás azon funkcióit valósítja meg, amelyek a retinai kép elemzését végzik. Ezek elsősorban a képi tartalom értelmezésére irányulnak: a látott képből következtet az objektumok 3D alakjára (felület rekonstrukció), az objektumok térbeli elhelyezkedésére, egymáshoz való viszonyára (mélységi információ kinyerése), illetve több, időben egymást követő képből a mozgás érzékelése és a mozgó objektumok követése. A könyv két fő témakört ölel fel: az első, Statikus látás című részében a ma már klasszikusnak mondható sztereó rekonstrukció problémáját tárgyaljuk. Ez lényegében az emberi látás térbeli érzékelésének számítógépes megvalósítása. A második, Dinamikus látás című részében egyetlen kamerával készült mozgókép sorozat elemzésével foglalkozunk. Tárgyaljuk a 3D mozgás, és annak projektív leképezése közötti kapcsolatot, a mozgás becslést optikai áramlás segítségével, illetve a mozgó objektumok követésére szolgáló algoritmusokat.

Tartalomjegyzék

Előszó	7
I. Statikus látás	9
1. A projektív kamera	12
1.1. A lyukkamera	12
1.1.1. A kamera koordináta rendszer és a kamera mátrix	13
1.1.2. A kép koordináta rendszer és a kalibrációs mátrix	13
1.1.3. A világ koordináta rendszer és a kamera külső paraméterei	15
1.2. A kamera mátrix geometriai elemzése	16
1.2.1. A kamera középpont	16
1.2.2. A világ koordináta rendszer képe	17
1.2.3. A fősík és a tengelysíkok	17
1.2.4. Az optikai tengely és a fópont	18
1.3. Kamera kalibráció	18
1.3.1. Algebrai alapmegoldás	19
1.3.2. A kalibrációs minta tervezése	21
1.3.3. Normalizálás	21
1.3.4. Geometriai hiba minimalizálása és a <i>Gold Standard</i> algoritmus	22
1.3.5. További feltételek a kalibrációs mátrixban	22
2. A projektív kamera képe	24
2.1. Az euklideszi tér elemeinek képe	24
2.1.1. Sík képe	24
2.1.2. Egyenes képe	25
2.1.3. Kúpszelet képe és visszavetítése	25
2.1.4. Másodrendű felület képe	26
2.2. A projektív tér képzetes elemeinek képe	26
2.2.1. A végtelen távoli sík képe	26
2.2.2. Az ideális kúpszelet képe	27
2.2.3. Eltűnési pont	28
2.2.4. Eltűnési egyenes	29
2.3. A projektív tér elemei és a kalibráció kapcsolata	30

2.3.1.	Kalibráció az ideális kúpszelet képéből	30
2.3.2.	Eltűnési pontok és a kamera főpontja	31
2.3.3.	Kalibráció 3 sík képéből	32
3.	Sztereó képpár geometriája	34
3.1.	Epipoláris geometria	34
3.2.	A fundamentális mátrix	36
3.2.1.	A fundamentális mátrix kiszámítása	38
3.2.2.	A fundamentális mátrix további tulajdonságai	40
3.3.	Az esszenciális mátrix	40
3.4.	Standard sztereó konfiguráció	41
3.4.1.	Képek rektifikálása	42
4.	3D rekonstrukció	45
4.1.	Sztereó megfeleltetések	45
4.2.	Kanonikus kamerák	49
4.3.	Sztereó rekonstrukció	50
4.3.1.	Projektív rekonstrukció	50
4.3.2.	3D pontok meghatározása	51
4.3.3.	Rétegelt rekonstrukció	53
II.	Dinamikus látás	56
5.	Mozgás leképezése, mozgásmodellek	58
5.1.	A mozgásmező	58
5.2.	Mozgás leképezése álló kamera esetén	59
5.3.	Mozgás leképezése mozgó kamera esetén	61
5.4.	Haladó mozgást végző kamera	62
5.5.	Forgó mozgást végző kamera	63
5.6.	Sík felület elmozdulása	64
5.7.	Mozgás-parallaxis	65
6.	Optikai áramlás	67
6.1.	Az optikai áramlás meghatározásának módszerei	67
6.2.	Az intenzitás-megmaradás elve	68
6.3.	Az apertúra probléma	69
6.4.	Blokkegyezés alapú algoritmus	71
6.5.	Horn és Schunck algoritmus	71
6.6.	Lucas és Kanade algoritmus	72
6.7.	A mozgásbecslés korlátozásai	73

7. Mozgáskövetés	76
7.1. Kanade-Lucas-Tomasi követő	76
7.2. Követés mint valószínűségi következtetés	77
7.3. A Kálmán szűrő és alkalmazása	78
7.3.1. A Kálmán szűrő alkalmazása vizuális követésre	80
7.4. Részecske szűrő – részecske követő	81
III. Függelék	83
A Projektív geometria és lineáris algebra alapok	84
A.1. Projektív sík és projektív tér	84
A.1.1. Homogén koordináta és ideális pont	84
A.1.2. Egyenes és sík	85
A.1.3. Kúpszelet	85
A.2. Mátrixok felbontása	86
A.2.1. Cholesky felbontás	86
A.2.2. RQ vagy QR felbontás	86
A.2.3. SVD felbontás	86
Irodalomjegyzék	87

Előszó

A számítógépes látás az emberi látás azon funkcióit valósítja meg, amelyek a retinai kép elemzését végzik. Ezek elsősorban a képi tartalom értelmezésére irányulnak: a látott képből következtet az objektumok 3D alakjára (felület rekonstrukció), az objektumok térbeli elhelyezkedésére, egymáshoz való viszonyára (mélységi információ kinyerése), illetve több, időben egymást követő képből a mozgás érzékelése és a mozgó objektumok követése.

Könyvünk első, *Statikus látás* című részében a ma már klasszikusnak mondható sztereó rekonstrukció problémáját tárgyaljuk. Ez lényegében az emberi látás térbeli érzékelésének számítógépes megvalósítása, mely csak annyiban statikus, hogy –hasonlóan az emberi látáshoz– az egyébként dinamikus 3D valóság egyetlen időpillanataról rögzít egy képpárt. A számítógépes látás esetén azonban a kamerák szinkronizálása nem feltétlenül szükséges, ha a 3D valóság statikus. Ekkor a két képet nagyobb időbeli eltéréssel is készíthetjük, hiszen a leképezett látvány közben nem változik. A témakörben több angol nyelvű összefoglaló mű készült [4, 18], jelen tankönyv elsősorban a *Hartley* és *Zisserman* könyvet [4] veszi alapul és annak jelölésrendszerét követi.

Könyvünk második, *Dinamikus látás* című részében egyetlen kamerával készült mozgó-kép sorozat elemzésével foglalkozunk. Vizsgáljuk majd a 3D mozgás, és annak projektív leképezése közötti kapcsolatot. A mozgás érzékelése, hasonlóan az emberi mozgás érzékeléshez, a vizuális elmozdulás elemzésén alapszik. Ebből az ún. optikai áramlásból következtethetünk azután valós 2D illetve 3D elmozdulásokra. További fontos probléma a mozgó objektumok követése, mely elengedhetetlenül szükséges a magasabb szintű mozgás értelmezéshez (mint például trajektória elemzés).

Tudomásunk szerint könyvünk az első magyar nyelvű munka ebben a témakörben, ezért igyekeztünk az angol szakkifejezések minél kifejezőbb magyar megfelelőit is megtalálni. Reméljük, hogy tankönyvünket nem csak egyetemi vagy főiskolai hallgatók olvassák majd, hanem a műszaki fejlesztésekben részt vevő mérnökök, programozók is hasznosnak találják a könyv által bemutatott technológiákat.

Szeged és Veszprém, 2011. július 3.

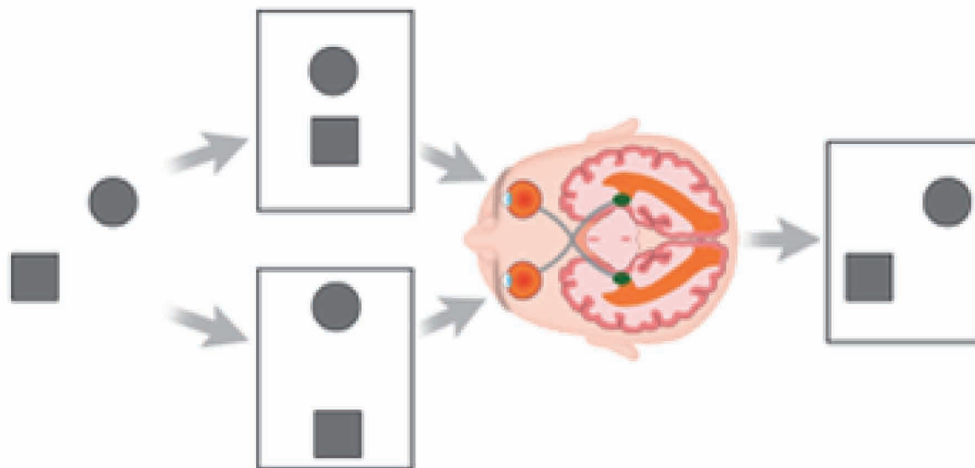
Kató Zoltán *SZTE Képfeldolgozás és Számítógépes Grafika Tanszék*
Czúni László *PE Műszaki Informatikai Kar*

Első rész

Statikus látás

Bevezetés

Az emberi látás egyik alapvető feladata a 3D világ érzékelése a valóság szemünk által felfogott 2D vetületeinek alapján. Tehát matematikai értelemben egy inverz problémát kell megoldanunk. Azonban ez az inverz probléma önmagában nem oldható meg, hiszen a $3D \rightarrow 2D$ vetületképzés során a mélységi információt teljesen elveszítjük. Ezért általában a 3D rekonstrukcióhoz valamilyen regularizálásra is szükségünk van, amely a 3D világról szerzet *a priori* ismereteinket alapszik. Néhány ilyen jól ismert, és tudattalanul is alkalmazott vizuális ismeretet láthatunk a 2 ábrán.



1. ábra. Az emberi 3D érzékelés a két szem által látott projektív kép alapján működik. Hogyan lesz a 2D képpárból egyetlen 3D látvány?

Magához a 3D rekonstrukcióhoz a legfontosabb – és egyetlen fizikai – eszközünk a két, egymástól viszonylag kis távolságra lévő szemünk, melyek a 3D világról két projektív képet szolgáltatnak két különböző, de egymáshoz közeli nézőpontból. Ebből a képpárból azután az emberi agy rendkívüli hatékonysággal állítja össze a valóság 3D látványát (ld. 1 ábra).

Könyvünk jelen részében ezen két képpár és a 3D látvány közötti geometriai kapcsolatot fogjuk vizsgálni, és olyan algoritmusokat konstruálunk, amelyekkel egy számítógép is hasonló rekonstrukcióra képes.

Az itt tárgyalt anyag megértéséhez csupán minimális képfeldolgozási előismeret [19] és néhány projektív geometriai alapfogalom (melyhez magyar és angol nyelvű összefoglaló is elérhető az interneten) ismerete szükséges.



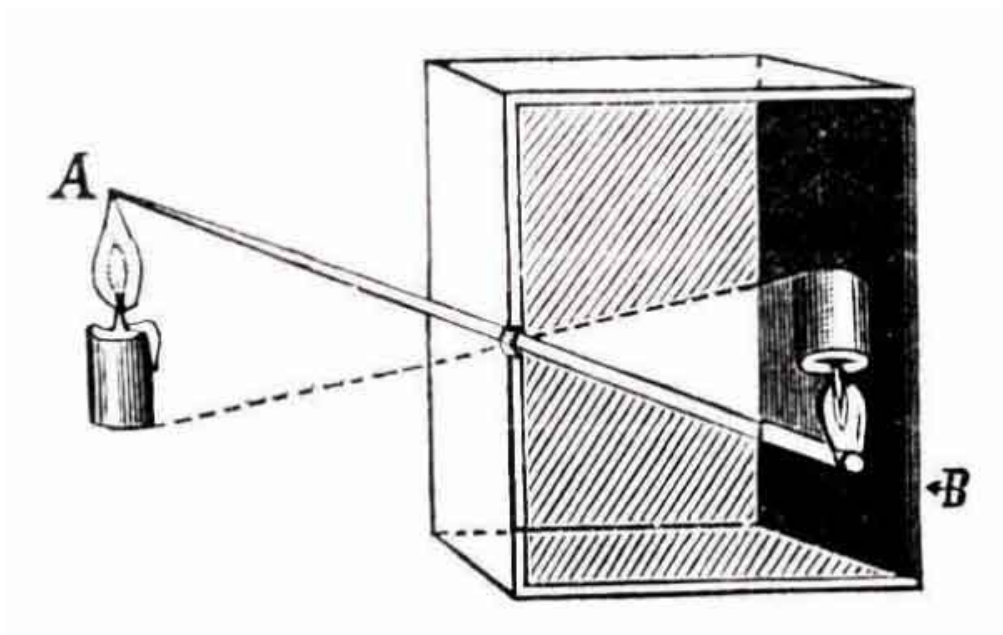
2. ábra. **Felső sor:** Vizuális tapasztalatunk alapján a megvilágítás felülről érkezik, ezért ha fejjel lefelé fordítjuk ugyanazt a képet, akkor ami eddig kifelé domborodott, azt befelé domborodónak érzékeljük. **Alul** a jól ismert lineáris perspektívára látunk példát, amelyet a festészetben is kihasználnak: a párhuzamos egyenesek a kamerától távolodva összetartanak – ettől lesz a képnek „mélysége”.

1. fejezet

A projektív kamera

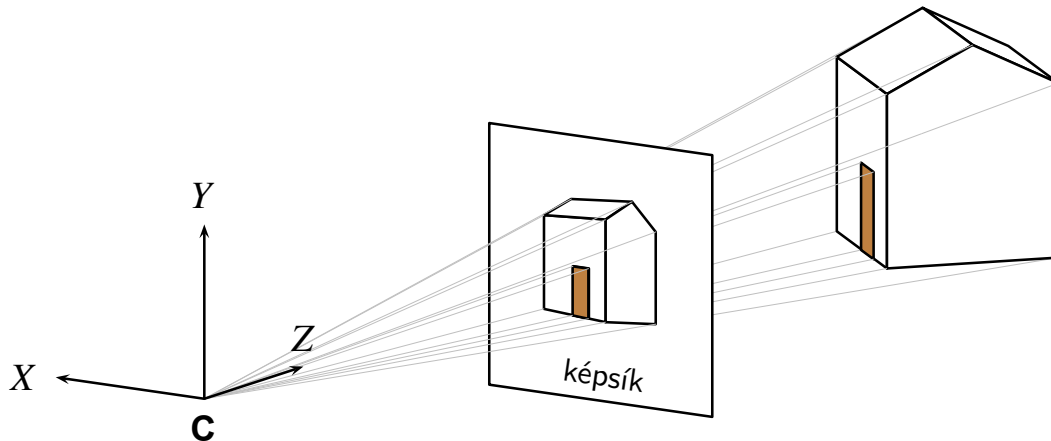
Egy közönséges digitális kamera felfogható egy olyan eszközként, amely a látóterében lévő 3D valós *látványról* egy 2D *képet* készít, vagyis matematikai értelemben nem más, mint egy $3D \rightarrow 2D$ leképezés. Könyvünkben olyan kamerákkal foglalkozunk, ahol ez a leképezés *középpontos vetítés* formájában valósul meg.

1.1. A lyukkamera



1.1. ábra. Camera obscura (Fizyka, 1910)

A legegyszerűbb projektív kamera a lyukkamera (latinul *camera obscura*), amely egy zárt dobozból (az un. sötétkamrából) áll, melynek egyik oldalán egy kis méretű lyuk található. A lyukon keresztül a fénysugarak bejutnak a dobozba és a lyukkal átellenes oldalra vetítik a kamera által látott *látvány* fordított képét (ld. 1.1 ábra).



1.2. ábra. A lyukkamera geometriai modellje: A képsíkot a \mathbf{C} fókuszpont elé helyezve oldalhelyes képet kapunk.

1.1.1. A kamera koordináta rendszer és a kamera mátrix

A lyukkamera (továbbiakban kamera) geometriai modelljéhez először is rögzítsünk egy 3D euklideszi koordináta rendszert, amelyet *kamera koordináta rendszernek* nevezünk (ld. 1.2 ábra). A \mathbf{C} -vel jelölt origó a kamera vetítési középpontjában, a *fókuszpontban* található (ld. 1.3 ábra), és a Z tengely a fő vetítősugár, ami a kamera *optikai tengelye*. A Z tengelyre merőleges $Z = f$ sík a vetítési sík, amit *képsíknak* nevezünk, f pedig a kamera *fókusz távolsága*. Az optikai tengely képsíkon vett \mathbf{o} dőféspontját nevezzük *főpontnak*, a kamera koordináta rendszer XY síkját pedig *fősíknak*, amely párhuzamos a képsíkkal.

Egy 3D $\mathbf{X} = (X_1, X_2, X_3)^T$ pont $\mathbf{x} = (x_1, x_2)$ képét az \mathbf{XC} vetítősugár képsíkkal vett metszéspontja adja. A képpont koordinátáit egy olyan 2D euklideszi koordináta rendszerben kapjuk, melynek origója az \mathbf{o} főpont, x és y tengelyei pedig a kamera koordináta rendszer megfelelő X és Y tengelyeivel párhuzamosak. Ezek után a hasonló háromszögekből (ld. 1.3 ábra) egyszerűen adódik a kamera leképezése: $(X_1, X_2, X_3)^T \mapsto (f X_1 / X_3, f X_2 / X_3)^T$. Homogén koordinátákra áttérve a leképezés egy mátrix szorzás formájában adható meg:

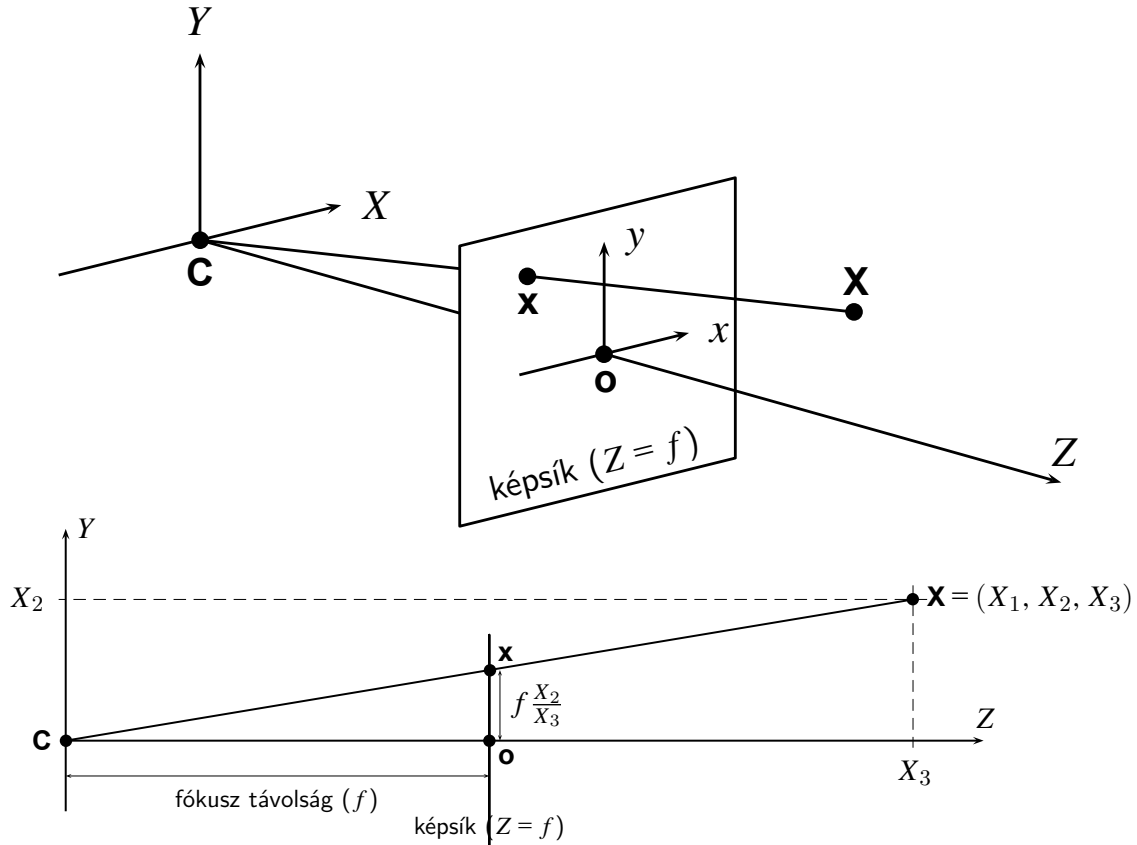
$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f X_1 \\ f X_2 \\ X_3 \end{pmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix} \quad (1.1)$$

Vegyük észre, hogy a homogén képpont koordináta $(f X_1, f X_2, X_3)^T = (f X_1 / X_3, f X_2 / X_3, 1)^T$ alakban is írható, ahonnan közvetlenül adódik a képpont $(f X_1 / X_3, f X_2 / X_3)^T$ inhomogén koordinátája. A fenti 3×4 mátrixot *homogén kamera mátrixnak* nevezzük és \mathbf{P} -vel jelöljük, amivel a fenti összefüggést tömörebb formában is megadhatjuk:

$$\mathbf{x} = \mathbf{PX} \quad (1.2)$$

1.1.2. A kép koordináta rendszer és a kalibrációs mátrix

A gyakorlatban a képpontokat azonban nem a fenti speciális 2D koordináta rendszerben adjuk meg, hanem egy általános helyzetű *kép koordináta rendszerben*, melynek origója tipikusan a



1.3. ábra. Egy valós 3D pont (\mathbf{X}) és képe (\mathbf{x}) közötti kapcsolat.

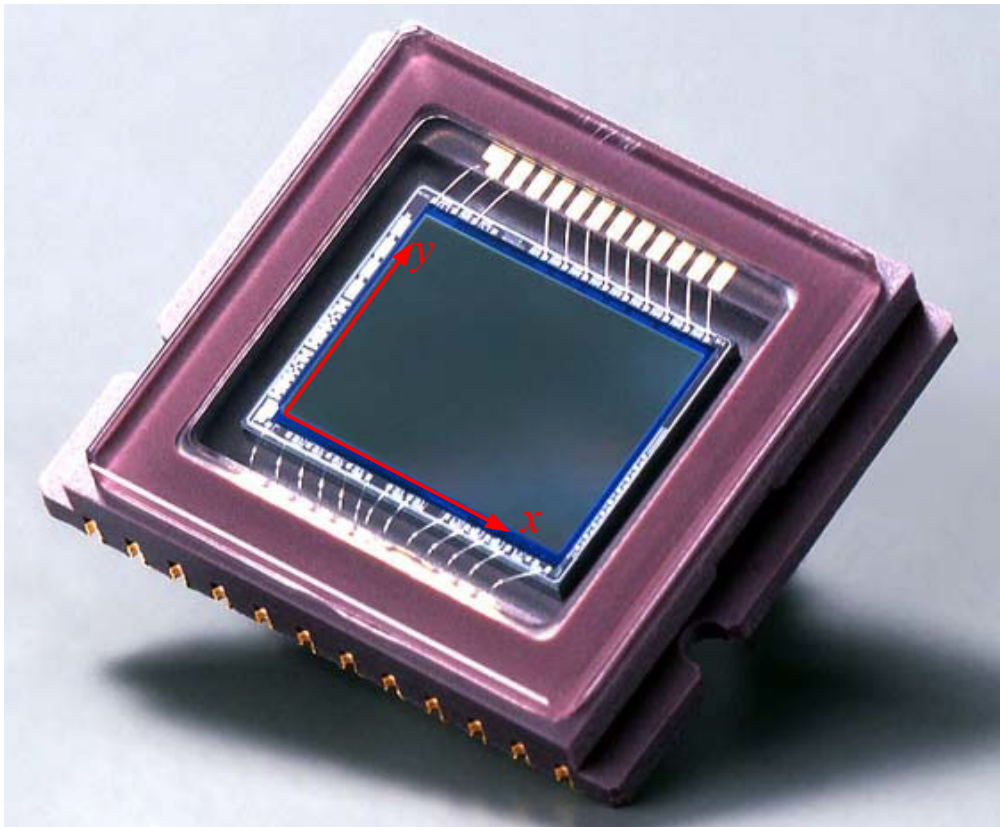
kép valamelyik sarkában található. Ez a kamera leképezésében egy eltolás formájában jelenik meg, az eltolás pedig a főpont \mathbf{o} kép koordináta rendszerben vett (o_x, o_y) koordinátája lesz:

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX_1 + X_3o_x \\ fX_2 + X_3o_y \\ X_3 \end{pmatrix} = \begin{bmatrix} f & o_x & 0 \\ f & o_y & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix} \quad (1.3)$$

A digitális kamerákban található szenzorok (pl. CCD, ld. 1.4 ábra) azonban egy diszkrét, pixelekből álló képet állítanak elő. Ezért a kép koordináta rendszerünk skálabeosztása pixeleken adott, amely az eredeti folytonos koordináta rendszerhez képest egy tengelyenként különböző skálafaktor formájában jelenik meg: s_x illetve s_y . A skálafaktorok értéke azt fejezi ki, hogy az eredeti folytonos koordináta rendszer egységnyi beosztására hány pixel jut. Ha például a folytonos koordináta rendszeren miliméteres beosztást tételezünk fel, akkor s_x a kamera x tengely mentén vett felbontását adja meg pixel/miliméterben. Az így előálló kamera mátrix az alábbi alakban írható:

$$\begin{bmatrix} s_x f & s_x o_x & 0 \\ & s_y f & s_y o_y & 0 \\ & & 1 & 0 \end{bmatrix} \quad (1.4)$$

Ha $s_x = s_y$, akkor a fenti kamera mátrix négyzet alakú pixeleket tételez fel, ami a gyakorlatban használt digitális kamerák döntő többségére igaz. Elképzelhető azonban olyan kamera mátrix



1.4. ábra. CCD szenzor és kép koordináta rendszer.

is, amelyben a pixelek még csak nem is téglalapok, hanem általános paralelogrammák, vagyis az x és y tengelyek nem merőlegesek. Egy tipikus példa erre, amikor egy képről készítünk egy újabb képet, és a kamera képsíkja nem párhuzamos a lefényképezett kép síkjával. Ezt a fajta torzulást egy a nyírás formájában vehetjük figyelembe a kamera mátrixban:

$$\begin{bmatrix} s_x f & a & s_x o_x & 0 \\ & s_y f & s_y o_y & 0 \\ & & 1 & 0 \end{bmatrix} \quad (1.5)$$

A kamera mátrixot írhatjuk az alábbi formában is:

$$\mathbf{P} = \mathbf{K} [\mathbf{I} | \mathbf{0}], \quad (1.6)$$

ahol \mathbf{I} egy 3×3 egységmátrix és a 3×3 felső trianguláris \mathbf{K} mátrixot *kamera kalibrációs mátrix*nak nevezzük. A kalibrációs mátrix tartalmazza a kamera *belső paramétereit*, melyek száma a fenti általános esetben 5. Ezek a paraméterek kizárólag a kamera belső tulajdonságaitól függenek, azokat nem befolyásolja sem a kamera helyzete sem pedig iránya.

1.1.3. A világ koordináta rendszer és a kamera külső paramétereit

Az eddig tárgyalt kamera mátrix a 3D kamera koordináta rendszerben adott pontok képét állítja elő a kép koordináta rendszerben. A gyakorlatban azonban a 3D pontok egy általános

helyzetű 3D euklideszi koordináta rendszerben, a *világ koordináta rendszerben* vannak megadva. A két koordináta rendszer között egy 3D merevtest transzformáció hat, mely az alábbi elemi transzformációk kompozíciója:

1. Egy \mathbf{t} eltolás, amely a világ koordináta rendszer origóját átviszi a kamera középpontba. Vegyük észre, hogy az eltolás nagysága pontosan a kamera középpont világ koordináta rendszerbeli inhomogén koordinátája lesz, melyet $\tilde{\mathbf{C}}$ jelöl, vagyis $\mathbf{t} = -\tilde{\mathbf{C}}$
2. Ezután következik egy \mathbf{R} forgatás, amely a világ koordináta tengelyeket illeszti a kamera koordináta tengelyekre.

A leképezés homogén koordinátákkal az alábbi alakban írható fel:

$$\mathbf{X}' = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix} \quad (1.7)$$

ahol \mathbf{X}' az \mathbf{X} homogén pont kamera koordináta rendszerbeli homogén koordinátája. Összefoglalva tehát az általános kamera mátrixot, amely a világ koordináta rendszerből képez le a kép koordináta rendszerbe, az alábbi alakokban írhatjuk:

$$\mathbf{P} = \mathbf{K} [\mathbf{R} | \mathbf{Rt}] = \mathbf{KR} [\mathbf{I} | \mathbf{t}] \quad (1.8)$$

$$= \mathbf{K} [\mathbf{R} | -\mathbf{R}\tilde{\mathbf{C}}] = \mathbf{KR} [\mathbf{I} | -\tilde{\mathbf{C}}] \quad (1.9)$$

Az (\mathbf{R}, \mathbf{t}) transzformáció 3 forgatási és 3 eltolási paraméterét a kamera *külső paramétereinek* nevezzük, mert ezek kizárólag a kamera világ koordináta rendszerben vett helyzetétől és irányától függenek. Vegyük észre, hogy ha a világ koordináta rendszer megegyezik a kamera koordináta rendszerrel, akkor $(\mathbf{R}, \mathbf{t}) = (\mathbf{I}, \mathbf{0})$, vagyis a kamera mátrix $\mathbf{P} = \mathbf{K} [\mathbf{I} | \mathbf{0}]$ alakban írható fel.

1.2. A kamera mátrix geometriai elemzése

A továbbiakban megvizsgáljuk, hogy a leképezés különböző geometriai elemei hogyan fejthetők ki a kamera mátrixból.

1.2.1. A kamera középpont

Mivel \mathbf{P} rangja 3, de 4 oszlopa van, ezért nulltere 1 dimenziós, más szóval a $\mathbf{P}\mathbf{C} = \mathbf{0}$ egyenletrendszer \mathbf{C} által generált megoldásai egy egyenesre esnek. Könnyen beláthatjuk, hogy a 4 elemű homogén \mathbf{C} vektor nem más, mint a kamera középpont. Ehhez vegyünk egy tetszőleges \mathbf{A} pontot, és a $\overline{\mathbf{CA}}$ egyenest, melynek pontjai az alábbi egyenlettel adhatók meg:

$$\mathbf{X}(\lambda) = \lambda\mathbf{A} + (1 - \lambda)\mathbf{C} \quad (1.10)$$

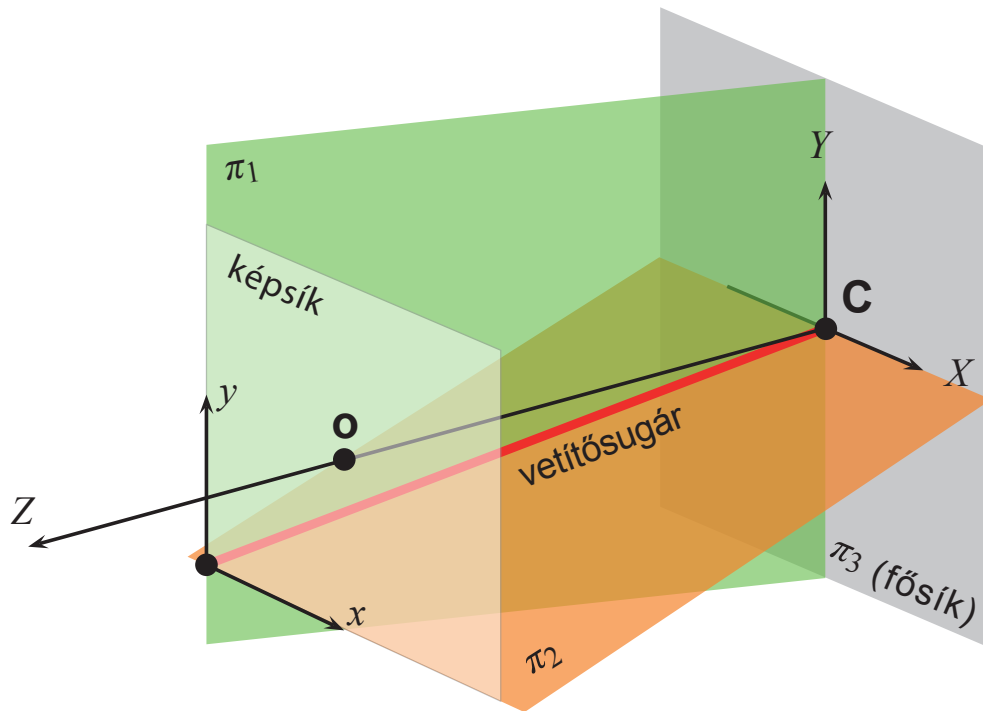
Az egyenes pontjait a \mathbf{P} kamera az alábbiak szerint vetíti:

$$\mathbf{x} = \mathbf{P}\mathbf{X}(\lambda) = \lambda\mathbf{P}\mathbf{A} + (1 - \lambda)\mathbf{P}\mathbf{C} = \lambda\mathbf{P}\mathbf{A} \quad (1.11)$$

Mivel $\mathbf{P}\mathbf{C} = \mathbf{0}$, az egyenes valamennyi pontja egyetlen képpontba, a $\mathbf{P}\mathbf{A}$ pontba képeződik le, amiből következik, hogy maga az egyenes egy vetítősugár, amely átmegy a kamera középponton. Mivel \mathbf{A} választása tetszőleges volt, ezért bármely \mathbf{A} pontra igaz az, hogy az $\mathbf{X}(\lambda)$ egyenes egy vetítősugár, amiből következik, hogy \mathbf{C} a kamera középpont homogén koordinátája.

1.2.2. A világ koordináta rendszer képe

Jelöljük a kamera mátrix oszlopvektorait rendre $\mathbf{p}_1, \dots, \mathbf{p}_4$ -el. Ezek a 3 elemű vektorok speciális képpontoknak felelnek meg, amelyekből az első három a világ koordináta rendszer tengelyeinek eltűnési pontjai (az eltűnési pont fogalmát ld. a 2.2.3 fejezetben), míg \mathbf{p}_4 az origó képe. Tekintsük például az X tengelyt, melynek iránya $\mathbf{D} = (1, 0, 0, 0)^T$, vagyis a tengely végtelen távoli síkon vett dőféspontja, melynek képe a kamerában (vagyis a tengely eltűnési pontja) $\mathbf{P}\mathbf{D} = \mathbf{p}_1$ lesz. A többi tengelyre hasonlóan vezethető le az összefüggés. Az origó $\mathbf{O} = (0, 0, 0, 1)^T$ képe a kamerában pedig $\mathbf{P}\mathbf{O} = \mathbf{p}_4$ lesz.



1.5. ábra. A kamera mátrix $\boldsymbol{\pi}_1^T, \dots, \boldsymbol{\pi}_3^T$ sorvektorai által meghatározott síkok.

1.2.3. A fősík és a tengelysíkok

Jelöljük a kamera mátrix sorvektorait rendre $\boldsymbol{\pi}_1^T, \dots, \boldsymbol{\pi}_3^T$ -al. Ezek a 4 elemű vektorok speciális síkoknak felelnek meg a kamera koordináta rendszerben (ld. 1.5 ábra). A kamera fősíkja

az optikai tengelyre merőleges, a kamera középponton átmenő sík, vagyis a kamera koordináta rendszer XY síkja. A sík pontjait a kamera a képsík végtelen távoli egyenesére képezi le, hiszen valamennyi vetítősugár a képsíkkal párhuzamosan fut, így azok a képsíkkal a végtelen távoli egyenesen metszik egymást. Formálisan a fősík pontjainak képét tehát $\mathbf{PX} = (x_1, x_2, 0)^T$ alakban írhatjuk. A végtelen távoli egyenesre tehát pontosan azokat a pontokat képezi le a kamera, amelyekre igaz az, hogy $\boldsymbol{\pi}_3^T \mathbf{X} = 0$, vagyis a fősíkot $\boldsymbol{\pi}_3^T$ adja. Vegyük észre, hogy a kamera középpont mindhárom síkon rajta van, hiszen $\mathbf{PC} = \mathbf{0}$ és ezért $\forall i : \boldsymbol{\pi}_i^T \mathbf{C} = 0$.

Most tekintsük a $\boldsymbol{\pi}_1^T$ sík pontjait, melyek kielégítik a $\boldsymbol{\pi}_1^T \mathbf{X} = 0$ egyenletet. Ezeket a pontokat a kamera a $\mathbf{PX} = (0, x_2, x_3)^T$ pontokra képezi le, ami pontosan a kép koordináta-rendszer y tengelyének felel meg. Természetesen a \mathbf{C} kamera középpont is a síkon van, ezért a $\boldsymbol{\pi}_1^T$ sík megegyezik a \mathbf{C} kamera középpont és a kép koordináta rendszer y tengelye által meghatározott síkkal. Hasonlóan adódik, hogy $\boldsymbol{\pi}_2^T$ a \mathbf{C} és az x tengely által meghatározott sík lesz. Vegyük észre, hogy a $\boldsymbol{\pi}_1^T$ és $\boldsymbol{\pi}_2^T$ síkok helyzete függ a kép koordináta-rendszer megválasztásától, továbbá ezen síkok metszete pontosan a kép koordináta-rendszer origójának vetítősugarával egyezik meg.

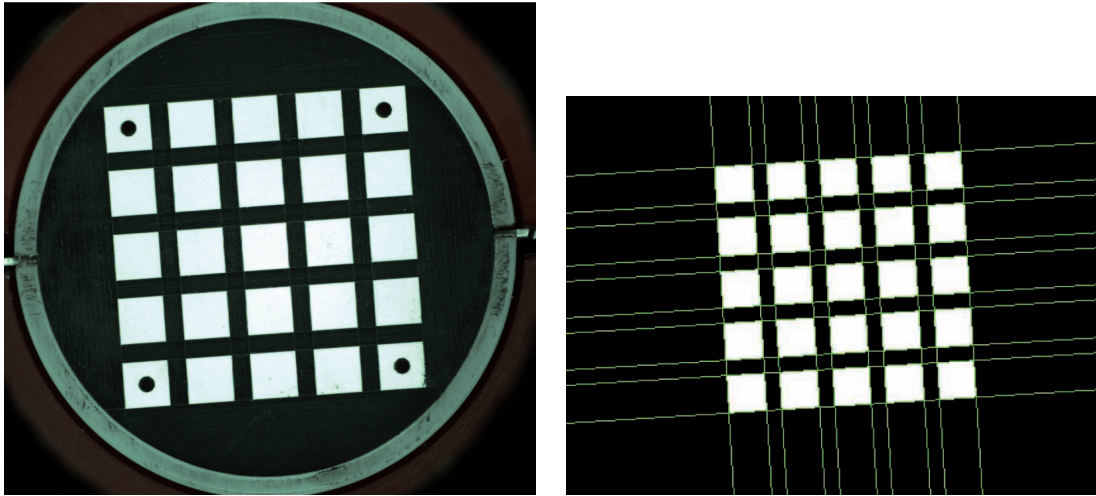
1.2.4. Az optikai tengely és a főpont

A kamera optikai tengelye átmegy a kamera középponton és merőleges a $\boldsymbol{\pi}_3^T$ fősíkra, míg az optikai tengely képsíkkal vett dőféspontja az \mathbf{o} főpont. Mivel $\boldsymbol{\pi}_3^T$ a \mathbf{P} kamera mátrix harmadik sora, így $\boldsymbol{\pi}_3^T = (P_{31}, P_{32}, P_{33}, P_{34})^T$ és a sík normálvektora $(P_{31}, P_{32}, P_{33})^T$, melyet irányként is felírhatunk a végtelen távoli síkon vett $(P_{31}, P_{32}, P_{33}, 0)^T$ pont formájában. Mivel a pont utolsó koordinátája 0, ezért a fenti pont képének előállításában a kamera mátrix utolsó oszlopára nincs szükség. Bontsuk fel tehát a kamera mátrixot $\mathbf{P} = [\mathbf{M} | \mathbf{p}_4]$ alakban, ahol \mathbf{M} egy 3×3 mátrix. Ezt felhasználva $(P_{31}, P_{32}, P_{33}, 0)^T = (\mathbf{m}_3 | 0)^T$, ahol \mathbf{m}_3 az \mathbf{M} mátrix harmadik sora, és a főpont koordinátáját $\mathbf{o} = \mathbf{M}\mathbf{m}_3$ alakban kapjuk.

Az előzőekben már láttuk, hogy a fősík normálvektora \mathbf{m}_3^T , ami tehát az optikai tengelyt is kijelöli. Azonban a kamera mátrix csak egy előjel erejéig meghatározott, ezért nem tudhatjuk, hogy az \mathbf{m}_3^T előre, a kamera által látott látvány irányába mutat, vagy hátrafelé. Egyszerűen belátható viszont, hogy $\det(\mathbf{M})\mathbf{m}_3$ egyértelműen megadja az optikai tengelyt kijelölő helyes irányítottágú vektort (részletesen ld. [4]).

1.3. Kamera kalibráció

Az előzőekben láttuk, hogy egy kamera lényegében a mátrixával egyértelműen reprezentálható. Ha ismert a kamera mátrix, akkor egy tetszőleges pont képét elő tudjuk állítani, illetve bármely képpontot vissza is tudunk vetíteni. A gyakorlatban tehát fontos, hogy meg tudjuk határozni a kamera mátrixot. Ezt a folyamatot *kamera kalibrációnak* nevezzük. Ebben a fejezetben azokat a kalibrációs eljárásokat fogjuk bemutatni, ahol elegendő számú *pontmegfeleltetés* áll rendelkezésünkre a 3D valós pontok és azok 2D képei között. Ilyen pontmegfeleltetéseket a gyakorlatban úgy szoktunk előállítani, hogy egy ún. *kalibrációs mintát* helyezünk a kamera elé, melynek ismerjük a pontos 3D méreteit, és az erről készült képen azután könnyen megtalálhatjuk a mintának megfelelő képpontokat. Ezek a könnyen azonosítható pontok a mintá-



1.6. ábra. Tipikus kalibrációs minta. A pontmegfeleltetéseket a négyzetek sarokpontjai szolgáltatják, melyeket megkaphatjuk például a négyzetek oldalaira illesztett egyenesek metszéspontjaiként.

zat sarokpontjai, ezért általában sakktabla-szerű kalibrációs mintákat szoktak használni. Egy ilyen kalibrációs mintát mutat az 1.6 ábra.

1.3.1. Algebrai alapmegoldás

Jelölje $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ a pontmegfeleltetéseinket, valamint legyen N a pontpárok száma. Valamennyi pontpárra teljesül a jól ismert $\mathbf{x}_i = \mathbf{P}\mathbf{X}_i$ összefüggés, amelyet vektoriális szorzat segítségével $\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \mathbf{0}$ alakban fejezhetünk ki. Most fejezzük ki a $\mathbf{P}\mathbf{X}_i$ szorzatot:

$$\mathbf{P}\mathbf{X}_i = \begin{pmatrix} \boldsymbol{\pi}_1^T \mathbf{X}_i \\ \boldsymbol{\pi}_2^T \mathbf{X}_i \\ \boldsymbol{\pi}_3^T \mathbf{X}_i \end{pmatrix} \equiv \begin{pmatrix} \mathbf{X}_i^T \boldsymbol{\pi}_1 \\ \mathbf{X}_i^T \boldsymbol{\pi}_2 \\ \mathbf{X}_i^T \boldsymbol{\pi}_3 \end{pmatrix} \quad (1.12)$$

Az $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3})^T$ jelölést bevezetve, a fenti képlet felhasználásával kifejezhetjük magát a vektoriális szorzatot:

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \begin{pmatrix} x_{i2} \mathbf{X}_i^T \boldsymbol{\pi}_3 - x_{i3} \mathbf{X}_i^T \boldsymbol{\pi}_2 \\ x_{i3} \mathbf{X}_i^T \boldsymbol{\pi}_1 - x_{i1} \mathbf{X}_i^T \boldsymbol{\pi}_3 \\ x_{i1} \mathbf{X}_i^T \boldsymbol{\pi}_2 - x_{i2} \mathbf{X}_i^T \boldsymbol{\pi}_1 \end{pmatrix} \quad (1.13)$$

Most írjuk fel a fenti vektort egy mátrix-vektor szorzat segítségével úgy, hogy a vektor tartalmazza az ismeretlen kamera mátrix elemeit:

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \begin{bmatrix} \mathbf{0}^T & -x_{i3} \mathbf{X}_i^T & x_{i2} \mathbf{X}_i^T \\ x_{i3} \mathbf{X}_i^T & \mathbf{0}^T & -x_{i1} \mathbf{X}_i^T \\ -x_{i2} \mathbf{X}_i^T & x_{i1} \mathbf{X}_i^T & \mathbf{0}^T \end{bmatrix} \begin{pmatrix} \boldsymbol{\pi}_1 \\ \boldsymbol{\pi}_2 \\ \boldsymbol{\pi}_3 \end{pmatrix} \quad (1.14)$$

Mivel a pontjaink homogén koordinátákkal vannak megadva, ezért a fenti 3 egyenlet közül csak kettő lesz lineárisan független. Egyszerűen beláthatjuk például, hogy a harmadik sor kifejezhető az első kettő lineáris kombinációjaként, ha feltesszük, hogy $x_{i3} = 1$. Ez nem jelent

megszorítást, hiszen a homogén koordinátákat végigosztva x_{i3} -al pontosan ezt a standard alakot kapjuk. Ekkor az első sort $-x_{i1}$ -el a másodikat pedig $-x_{i2}$ -vel szorozva a kettő összege pontosan a harmadik sort adja. Ezért egy $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ pontpár az alábbi két lineárisan független egyenletet adja:

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \begin{bmatrix} \mathbf{0}^T & -x_{i3}\mathbf{X}_i^T & x_{i2}\mathbf{X}_i^T \\ x_{i3}\mathbf{X}_i^T & \mathbf{0}^T & -x_{i1}\mathbf{X}_i^T \end{bmatrix} \begin{pmatrix} \boldsymbol{\pi}_1 \\ \boldsymbol{\pi}_2 \\ \boldsymbol{\pi}_3 \end{pmatrix} = \mathbf{0} \quad (1.15)$$

Az N pontmegfeleltetés tehát $2N$ egyenletet ad, amit mátrix alakban is írhatunk:

$$\mathbf{A}\mathbf{p} = \mathbf{0} \quad (1.16)$$

ahol $\mathbf{p} = (P_{11}, \dots, P_{34})^T$ a kamera mátrix ismeretlen elemeit tartalmazza, míg \mathbf{A} egy $2N \times 12$ mátrix. A fenti egyenletrendszer megoldása adja a kamera mátrix elemeit.



1.7. ábra. A leggyakrabban használt kalibrációs minta a Tsai rács

Mivel \mathbf{P} -nek 12 eleme van, de szabadsági foka 11 (ugyanis a homogén mátrix csak egy skálafaktor erejéig meghatározott), 11 egyenletre van szükségünk az egyértelmű megoldáshoz, ami lényegében az \mathbf{A} mátrix 1 dimenziós nulltere lesz, feltéve hogy a 11×12 méretű \mathbf{A} mátrix rangja 11. Ehhez pontosan $5\frac{1}{2}$ pontmegfeleltetésre van szükség, ahol a $\frac{1}{2}$ megfeleltetés azt jelenti, hogy az egyik pont esetén elegendő annak x vagy y koordinátáját ismernünk.

A gyakorlatban azonban a pont koordináták zajjal terheltek, ezért az 1.15 képlet csak közelítően teljesül, így az egyenletrendszernek nem lesz egzakt megoldása. A klasszikus megoldás ilyen esetekben egy túlhatározott egyenletrendszer felállítása, ami a mi esetünkben $N \geq 6$ pontpár felhasználását jelenti. Ilyenkor az egyenletrendszer $\|\mathbf{A}\mathbf{p}\|$ algebrai hibáját minimalizáljuk a további feltétellel, hogy $\|\mathbf{p}\| = 1$. Ezt a \mathbf{p} megoldást klasszikus módon az \mathbf{A} mátrix SVD felbontásával állíthatjuk elő.

1.3.2. A kalibrációs minta tervezése

A kalibrációs minta tervezésénél ügyelni kell arra is, hogy a minta által szolgáltatott \mathbf{X}_i pontok ne vezessenek elfajuló esetre (ilyen például, amikor a pontjaink egy sík és a kamera középponton átmenő egyenes unióján helyezkednek el), ilyenkor ugyanis a kamera mátrix nem határozható meg egyértelműen a képpontokból. Ezért a gyakorlatban széleskörűen elterjedt kalibrációs minta az 1.7 ábrán látható *Tsai rács*, amely két egymásra merőleges síkból áll, melyeken sakktábla-minta található. A négyzetek sarokpontjainak megfelelő \mathbf{x}_i képpontokat megkaphatjuk úgy, hogy az egy vonalba eső négyzetek oldalaira egyenest illesztünk, majd meghatározzuk ezen egyenesek metszéspontjait (ld. 1.6 ábra). Ez a módszer feltételezi, hogy a kameránk nemlineáris torzítása elhanyagolható. Ellenkező esetben a négyzetek sarokpontjait sarokdetektálóval (pl. Harris vagy SUSAN) kell meghatározni, majd a nemlineáris deformáció meghatározása után elvégezhetjük a kamera mátrix számítását.

1.3.3. Normalizálás

Vegyük észre, hogy az 1.16 képlet lineáris egyenletrendszerének együttthatói könnyen nagyságrendekkel eltérhetnek egymástól, hiszen ezeket a pontok koordinátaiból kapjuk az 1.15 képlet szerint. Ez azonban kedvezőtlenül befolyásolhatja az egyenletrendszer megoldását, hiszen a nagyobb együttthatóval szereplő ismeretlenekben már egy kis hiba is nagy mértékben megnöveli az algebrai hibát, míg egy kisebb együttthatójú ismeretlen hibája eredendően kevésbé befolyásolja az algebrai hiba nagyságát. Ezáltal tehát kiszámíthatatlan módon befolyásolhatjuk az egyes paraméterek meghatározásának pontosságát. Ezért elengedhetetlenül szükséges a pont koordinátákat az egyenletrendszer konstrukciója előtt *normalizálni* úgy, hogy azok hasonló nagyságrendűek legyenek. Ezt az alábbi összetett transzformációval érhetjük el:

1. Számoljuk ki a pontjaink középpontját, majd az így kapott vektorral toljunk el valamennyi pontot. Ezzel az origó a centroidba került.
2. Ezután alkalmazzunk egy olyan skálázást, mellyel a pontok origótól vett átlagos távolsága a képpontok esetén $\sqrt{2}$, míg a 3D kalibrációs minta pontjai esetén $\sqrt{3}$ lesz.

A fenti normalizálást végrehajtva az átlagpont koordinátája az egység vektor lesz, és így mind a képpontjaink mind pedig a kalibrációs mintánk egy-egy kanonikus koordinátarendszerbe kerül, amely független a kiindulási kép illetve a kalibrációs minta valós méreteitől. Természetesen a normalizált pontok alapján kiszámolt \mathbf{P}' kamera mátrix a kanonikus koordináta rendszerek pontjai között hat:

$$\mathbf{x}'_i = \mathbf{P}'\mathbf{X}'_i \quad (1.17)$$

ahol $\mathbf{x}'_i = \mathbf{N}\mathbf{x}_i$ illetve $\mathbf{X}'_i = \mathbf{M}\mathbf{X}_i$ a normalizálás utáni pont koordinátákat jelöli. Könnyű belátni, hogy az eredeti \mathbf{P} kamera mátrixot megfelelő *denormalizálással* kapjuk, hiszen

$$\mathbf{N}\mathbf{x}_i = \mathbf{x}'_i = \mathbf{P}'\mathbf{X}'_i = \mathbf{P}'\mathbf{M}\mathbf{X}_i \quad (1.18)$$

Balról szorozva \mathbf{N}^{-1} -el:

$$\mathbf{x}_i = \mathbf{N}^{-1}\mathbf{P}'\mathbf{M}\mathbf{X}_i \quad (1.19)$$

és mivel $\mathbf{x}_i = \mathbf{P}\mathbf{X}_i$, ezért

$$\mathbf{P} = \mathbf{N}^{-1}\mathbf{P}'\mathbf{M} \quad (1.20)$$

1.3.4. Geometriai hiba minimalizálása és a *Gold Standard* algoritmus

Az előzőekben láttuk, hogy hogyan kaphatunk becslést a kamera mátrix elemeire egy lineáris egyenletrendszer megoldásával. Az így kapott megoldás azonban nem lesz szükségszerűen a legjobb becslés geometriai értelemben, hiszen az 1.16 egyenletrendszer megoldása az algebrai hibát minimalizálja. Mivel a kalibrációs mintát nagy pontossággal szokták előállítani, hiszen az szolgál etalonként a kamera számára, ezért a kalibráció során a \mathbf{X}_i pontok koordinátáit hibamentesnek tekintjük. Ekkor a képen mérhető geometriai hibát az alábbi képlettel fejezhetjük ki:

$$\sum_i \|\mathbf{x}_i - \mathbf{P}\mathbf{X}_i\|^2 \quad (1.21)$$

vagyis a képből kinyert \mathbf{x}_i pontok és a becsült kamera mátrix által előállított tényleges $\mathbf{P}\mathbf{X}_i$ pontok négyzetes távolságainak összege. Amennyiben feltételezzük, hogy a pont koordinátáit terhelő hiba Gauss eloszlást követ, akkor a fenti hibát minimalizáló \mathbf{P} mátrix lesz a kamera mátrix *legnagyobb valószínűség* (angolul Maximum Likelihood) értelemben vett becslése. Ez a minimalizálási probléma lényegében a legkisebb négyzetek módszerére vezet, amit numerikus optimalizálással oldhatunk meg. Egy ilyen klasszikus módszer például a *Levenberg-Marquard algoritmus* [4].

Az eddigieket felhasználva megfogalmazhatjuk a kamera mátrix meghatározására szolgáló 1 (*Gold Standard*) algoritmust.

1. Algoritmus: Kamera kalibráció *Gold Standard* algoritmus

Input : $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}_{i=1}^N$, $N \geq 6$ pontmegfeleltetés

Output: A geometriai hibát ((1.21) képlet) minimalizáló \mathbf{P} kamera mátrix

- 1 Normalizáljuk az \mathbf{X}_i és \mathbf{x}_i pontokat rendre egy \mathbf{M} és egy \mathbf{N} hasonlósági transzformációval.
 - 2 A normalizált pont koordinátákból az 1.3.1 fejezetben tárgyalt módon konstruált $\mathbf{A}\mathbf{p} = \mathbf{0}$, $\|\mathbf{p}\| = 1$ egyenletrendszer megoldását az \mathbf{A} mátrix legkisebb szinguláris értékéhez tartozó egységnyi normájú szinguláris vektora adja, melyet egyszerűen megkaphatunk az \mathbf{A} SVD felbontásából.
 - 3 Az előző lépésben kapott megoldást kezdeti értéknek használva minimalizáljuk az 1.21 képlettel megadott geometriai hibát a *Levenberg-Marquardt* algoritmussal.
 - 4 Az előző lépésben kapott \mathbf{P}' megoldásból a keresett kamera mátrixot az 1.20 képlet szerinti denormalizálással kapjuk.
-

1.3.5. További feltételek a kalibrációs mátrixban

Ahogy azt már az 1.1.2 fejezetben említettük, a gyakorlatban használt digitális kamerák döntő többsége négyzetes pixelelkel rendelkezik, vagyis a kalibrációs mátrix nyírási paramétere $a = 0$, és a skálázási paraméterek mindkét tengelyen azonosak, vagyis $s_x = s_y$. Az

eddigiekben tárgyalt algoritmusok azonban egy általános kamera mátrixot határoznak meg, így nem garantálható az, hogy az eredményül kapott mátrix kielégíti ezeket a feltételeket.

Egyik legegyszerűbb, de nem a legjobb lehetőség a fenti feltételek biztosítására, hogy az eredményül kapott \mathbf{P} általános kamera mátrixot felbontjuk az 1.1.3 fejezetben ismertett $\mathbf{KR} \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{C}} \end{bmatrix}$ alakban. Mivel $\mathbf{P} = [\mathbf{M} | \mathbf{p}_4]$ alakban is írható, ezért:

$$\mathbf{KR} = RQ(\mathbf{M}) \quad (1.22)$$

vagyis a felső trianguláris \mathbf{K} kalibrációs mátrix és az \mathbf{R} forgatási mátrixot az \mathbf{M} mátrix RQ -felbontásából kapjuk, feltéve, hogy \mathbf{M} rangja 3. Ez utóbbi feltétel azonban minden véges kamera középpontú és 3 rangú kamera mátrix esetén teljesül. Ezután a $\tilde{\mathbf{C}}$ kamera középpontot pedig megkaphatjuk $-\mathbf{M}^{-1}\mathbf{p}_4$ alakban. Az így kapott kamera összetevőkben azután a kívánt értékeket beállítva (a fenti példában $a = 0$ és $s_x = s_y$) megkapjuk a kívánt feltételeknek eleget tevő kamera mátrixot. Ez a módszer jól működik, ha a kiindulási mátrixunkban már jó közelítéssel teljesülnek a kívánt feltételek (vagyis $a \approx 0$ és $s_x \approx s_y$).

Sajnos a gyakorlatban ez ritkán teljesül, ezért inkább az 1 (*Gold Standard*) algoritmus 3. lépésében szereplő, az 1.21 képlettel megadott geometriai hibához adott újabb költségtagok segítségével szokták a feltételeket kielégítő optimális kamera mátrixot előállítani. A fenti példa esetén tehát a minimalizálandó költségfüggvény az 1.21 képlet helyett az alábbi lesz:

$$\sum_i \|\mathbf{x}_i - \mathbf{P}\mathbf{x}_i\|^2 + wa^2 + w(s_x - s_y)^2 \quad (1.23)$$

ahol a w súly értéke kezdetben kicsi, majd minden iteráció után növeljük az értékét. Ezzel biztosítjuk, hogy a feltételeket fokozatosan érvényesítjük és az így kapott kamera mátrix elemei már numerikusan nagyon jól kielégítik azt. Ezután a megfelelő elemeket az előző bekezdésben bemutatott faktorizációs módszerrel már biztonságosan beállíthatjuk a kívánt értékekre. Az 1 (*Gold Standard*) algoritmus többi lépése változatlan marad.

2. fejezet

A projektív kamera képe

Ebben a fejezetben tovább vizsgáljuk a projektív kamera leképezésének tulajdonságait és azt fogjuk sorra venni, hogy a 3D látvány elemi geometriai alkotóelemei hogyan jelennek meg a kamera képében. Vizsgálódásunk kiterjed majd a hagyományos euklideszi geometriából jól ismert egyenes, sík, kúpszelet (vagy másodrendű görbe) leképezésére, de olyan *képzetes* projektív geometriai elemek leképezését és tulajdonságait is tárgyaljuk majd, mint az eltűnési pont, eltűnési egyenes, illetve az ideális kúpszelet képe és ezek kapcsolata a kamera kalibrációs mátrixához.

2.1. Az euklideszi tér elemeinek képe

Az előző fejezetben elsősorban egyetlen 3D pont és annak képe közötti kapcsolatot vizsgáltuk. Itt további, magasabb dimenziójú geometriai struktúrák, konkrétan egyenes, sík, kúpszelet, illetve másodrendű felületek (pl. ellipszoid, *illetve további példák*) képét fogjuk vizsgálni.

2.1.1. Sík képe

Tekintsük a π síkot, amely nem tartalmazza a kamera középpontot, és vegyük fel a világ koordináta rendszert úgy, hogy annak XY síkja, vagyis a $Z = 0$ sík a π síkkal essen egybe. Ekkor a sík pontjainak képét az alábbiak szerint kapjuk:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = [\mathbf{p}_1 \ \mathbf{p}_2 \ \mathbf{p}_3 \ \mathbf{p}_4] \begin{pmatrix} X_1 \\ X_2 \\ 0 \\ 1 \end{pmatrix} = [\mathbf{p}_1 \ \mathbf{p}_2 \ \mathbf{p}_4] \begin{pmatrix} X_1 \\ X_2 \\ 1 \end{pmatrix} \equiv \mathbf{H}\tilde{\mathbf{x}} \quad (2.1)$$

Vagyis a \mathbf{P} kamera mátrixból kapott 3×3 \mathbf{H} *síkhomográfia* segítségével kaphatjuk meg a π sík $\tilde{\mathbf{x}}$ pontjainak képét, ahol \mathbf{H} rangja 3, vagyis ebben az esetben a leképezés invertálható lesz.

2.1.2. Egyenes képe

Könnyen beláthatjuk, hogy egy egyenes képe szintén egyenes lesz. Formálisan tekintsük az \mathbf{A} és \mathbf{B} 3D pontok által meghatározott egyenes pontjait:

$$\mathbf{X}(\lambda) = \mathbf{A} + \lambda \mathbf{B} \quad (2.2)$$

Az egyenes pontjainak képét ismét a jól ismert módon kaphatjuk meg:

$$\mathbf{x}(\lambda) = \mathbf{P}\mathbf{X}(\lambda) = \mathbf{P}(\mathbf{A} + \lambda \mathbf{B}) = \mathbf{P}\mathbf{A} + \lambda \mathbf{P}\mathbf{B} = \mathbf{a} + \lambda \mathbf{b} \quad (2.3)$$

ahol \mathbf{a} az \mathbf{A} , míg \mathbf{b} a \mathbf{B} pont képe, tehát az eredeti $\overline{\mathbf{AB}}$ egyenes képe az $\overline{\mathbf{ab}}$ egyenes lesz.

Az $\overline{\mathbf{ab}}$ egyenes visszavetítése azonban a kamera középpont és maga az $\overline{\mathbf{ab}}$ egyenes által meghatározott sík lesz, amely természetesen tartalmazza az eredeti $\overline{\mathbf{AB}}$ egyenest. Általánosan is belátható, hogy egy \mathbf{l} képi egyenes visszavetítése a $\mathbf{P}^T \mathbf{l}$ sík lesz. Az \mathbf{l} egyenes pontjai kielégítik az $\mathbf{x}^T \mathbf{l} = 0$ egyenletet, tehát egy \mathbf{X} 3D pont $\mathbf{P}\mathbf{X}$ képe pontosan akkor lesz rajta az \mathbf{l} egyenesen, ha

$$\mathbf{x}^T \mathbf{P}^T \mathbf{l} \equiv \mathbf{X}^T (\mathbf{P}^T \mathbf{l}) = 0 \quad (2.4)$$

Ebből következik, hogy $\mathbf{P}^T \mathbf{l}$ pontosan egy olyan síkot határoz meg a 3D térben, amelynek \mathbf{X} pontjaira teljesül, hogy a képük rajta van az \mathbf{l} egyenesen. A sík normálisát a kamera koordináta rendszerben megkaphatjuk $\mathbf{n} = \mathbf{K}^T \mathbf{l}$ alakban, az így kapott \mathbf{n} azonban általában nem lesz egységvektor. Mivel a kamera koordináta rendszerben a kamera mátrix $\mathbf{P} = \mathbf{K} [\mathbf{I} | \mathbf{0}]$ alakú lesz (ld. 1.1.3 fejezet), \mathbf{l} pontjait visszavetítve a megfelelő vetítő egyenesek irányvektorait kapjuk (részletesebben ld. 2.2.1 fejezet): $\mathbf{d} = \mathbf{K}^{-1} \mathbf{x}$. Ezek nyilvánvalóan merőlegesek lesznek a sík normálisára, tehát kielégítik a $\mathbf{d}^T \mathbf{n} = (\mathbf{x}^T \mathbf{K}^{-T}) \mathbf{n} = 0$ összefüggést. Másrészt az egyenes pontjai kielégítik az $\mathbf{x}^T \mathbf{l} = 0$ egyenletet is, tehát $\mathbf{l} = \mathbf{K}^{-T} \mathbf{n}$, amiből megkapjuk az $\mathbf{n} = \mathbf{K}^T \mathbf{l}$ eredményt.

2.1.3. Kúpszelet képe és visszavetítése

Egy \mathbf{C} kúpszelet képét hasonlóan állíthatjuk elő, mint a sík képét, hiszen \mathbf{C} egy 3D térbeli $\boldsymbol{\pi}$ sík része, így a pontjait ugyanaz a homográfia képezi le, mint magát a $\boldsymbol{\pi}$ síkot. Tekintsük tehát a 2.1 képlet alapján a kúpszelet egyenletét:

$$\tilde{\mathbf{x}}^T \mathbf{C} \tilde{\mathbf{x}} = (\mathbf{x}^T \mathbf{H}^{-T}) \mathbf{C} (\mathbf{H}^{-1} \mathbf{x}) = \mathbf{x}^T (\mathbf{H}^{-T} \mathbf{C} \mathbf{H}^{-1}) \mathbf{x} \quad (2.5)$$

vagyis a kúpszelet képe a $\mathbf{H}^{-T} \mathbf{C} \mathbf{H}^{-1}$ kúpszelet lesz.

Egy képsíkbeli \mathbf{C} kúpszelet visszavetítése egy \mathbf{Q} kúp lesz, melynek csúcsa a kamera középpontba esik, a kúp palástját pedig a kúpszeletet érintő vetítősugarak alkotják. A kúpszelet pontjaira teljesül az $\mathbf{x}^T \mathbf{C} \mathbf{x} = 0$ összefüggés, amely azon térbeli \mathbf{X} pontokra lesz igaz, amelyek kielégítik a

$$0 = \mathbf{x}^T \mathbf{C} \mathbf{x} = (\mathbf{P}\mathbf{X})^T \mathbf{C} \mathbf{P}\mathbf{X} \equiv \mathbf{X}^T (\mathbf{P}^T \mathbf{C} \mathbf{P}) \mathbf{X} \quad (2.6)$$

egyenletet. Ebből következik, hogy $\mathbf{Q} = \mathbf{P}^T \mathbf{C} \mathbf{P}$ pontosan egy olyan kúpot határoz meg a 3D térben, amelynek \mathbf{X} pontjaira teljesül, hogy a képük rajta van az \mathbf{C} kúpszeleten. A kúp csúcsa a \mathbf{C} kamera középpont lesz, hiszen $\mathbf{P}\mathbf{C} = 0$, amiből következik, hogy

$$\mathbf{Q}\mathbf{C} \equiv \mathbf{P}^T \mathbf{C} (\mathbf{P}\mathbf{C}) = 0 \quad (2.7)$$

2.1.4. Másodrendű felület képe

Tekintsünk egy \mathbf{Q} kvadratikus felszínt a 3D térben. A felszín képét úgy kapjuk, hogy a kamera középpontból vetítősugarakat bocsájtok a felszínre. Az így keletkezett kép körvonalát azok a vetítősugarak rajzolják meg, melyek érintik a \mathbf{Q} felszínt. Könnyen beláthatjuk, hogy ez a körvonal az alábbi kúpszelet lesz:

$$\mathbf{C}^* = \mathbf{P}\mathbf{Q}^*\mathbf{P}^T \quad (2.8)$$

ugyanis \mathbf{C}^* bármely \mathbf{l} érintőjére teljesül az $\mathbf{l}^T \mathbf{C}^* \mathbf{l} = 0$ összefüggés, továbbá ezen érintők visszavetítései olyan $\boldsymbol{\pi} = \mathbf{P}^T \mathbf{l}$ síkok lesznek, melyek érintik a \mathbf{Q} felszínt, ezért kielégítik a $\boldsymbol{\pi}^T \mathbf{Q}^* \boldsymbol{\pi}$ egyenletet. Minden érintőre igaz tehát:

$$\boldsymbol{\pi}^T \mathbf{Q}^* \boldsymbol{\pi} = \mathbf{l}^T (\mathbf{P}\mathbf{Q}^*\mathbf{P}^T) \mathbf{l} \equiv \mathbf{l}^T \mathbf{C}^* \mathbf{l} = 0 \quad (2.9)$$

Vagyis teljesül a 2.8 képlet.

A \mathbf{Q} felszínt érintő vetítősugarak érintőpontjai szintén egy kúpszeletet határoznak meg, mely a \mathbf{Q} felszín $\boldsymbol{\pi} = \mathbf{Q}\mathbf{C}$ síkkal vett metszete lesz. A sík egyenlete egyszerűen adódik a \mathbf{C} kamera középpont és \mathbf{Q} pólus-polár relációjából.

2.2. A projektív tér képzetes elemeinek képe

A továbbiakban a projektív tér azon elemeinek képét fogjuk vizsgálni, amelyek a végtelen távoli síkon találhatóak. Fontos megjegyezni, hogy az előzőekben tárgyalt euklideszi tér elemei természetes részei a projektív térnek is, de míg azok valós képet generálnak, addig az itt tárgyalt geometriai komponensek képei *képzetesek*, vagyis azok ugyan megszerkeszthetők a képsíkon, de nem generálnak látható képet.

2.2.1. A végtelen távoli sík képe

A $\boldsymbol{\pi}_\infty = (0,0,0,1)^T$ végtelen távoli sík pontjait $\mathbf{X}_\infty = (\mathbf{d}^T, 0)^T$ alakban írhatjuk. A \mathbf{d}^T 3 elemű vektor értelmezhető a 3D euklideszi tér egy irányvektoraként, ezért a végtelen távoli sík pontjait *irányoknak* is nevezzük. Természetesen egy \mathbf{X}_∞ pont képét ugyanúgy kapjuk, mint bármely más valós pont képét:

$$\mathbf{x} = \mathbf{P}\mathbf{X}_\infty = \mathbf{K}\mathbf{R}[\mathbf{I} | -\tilde{\mathbf{C}}] \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} = \mathbf{K}\mathbf{R}\mathbf{d} \quad (2.10)$$

Hasonlóan a 2.1.1 fejezetben tárgyalt valós sík képéhez, a leképezés itt is egy síkhomográfia lesz, mely

$$\mathbf{H} = \mathbf{K}\mathbf{R} \quad (2.11)$$

alakban írható. Vegyük észre, hogy ez a leképezés független a kamera pozíciójától, kizárólag annak irányától és belső paramétereitől függ.

Amennyiben a világ koordináta rendszer megegyezik a kamera koordináta rendszerrel (vagyis $\mathbf{P} = \mathbf{K}[\mathbf{I}|\mathbf{0}]$), akkor a \mathbf{H} síkhomográfia megegyezik a \mathbf{K} kalibrációs mátrixszal

$$\mathbf{x} = \mathbf{K}\mathbf{d} \quad (2.12)$$

Továbbá a $\lambda \mathbf{d}$ egyenesek megfelelnek a kamera vetítő egyeneseknek, és ezen vetítősugarak pontjainak képét egy skálafaktor erejéig szintén \mathbf{Kd} alakban kapjuk:

$$\mathbf{x} = \mathbf{K} [\mathbf{I} | \mathbf{0}] \begin{pmatrix} \lambda \mathbf{d} \\ 1 \end{pmatrix} = \mathbf{Kd} \quad (2.13)$$

Vagyis \mathbf{K} nem más, mint egy *affin* leképezés a kép pontjai és azok vetítőegyenesei között. A vetítőegyenesek irányvektorait

$$\mathbf{d} = \mathbf{K}^{-1} \mathbf{x} \quad (2.14)$$

alakban kapjuk, ahol \mathbf{d} általában nem egységnyi hosszúságú. Két vetítősugar által bezárt szöveget a jól ismert koszinusz képlettel kaphatjuk meg, melyet a fenti összefüggés felhasználásával kifejezhetünk a képpont koordináták és a kalibrációs mátrix segítségével is:

$$\begin{aligned} \cos(\theta) &= \frac{\mathbf{d}_1^T \mathbf{d}_2}{\|\mathbf{d}_1\| \|\mathbf{d}_2\|} = \frac{(\mathbf{K}^{-1} \mathbf{x}_1)^T (\mathbf{K}^{-1} \mathbf{x}_2)}{\sqrt{(\mathbf{K}^{-1} \mathbf{x}_1)^T (\mathbf{K}^{-1} \mathbf{x}_1)} \sqrt{(\mathbf{K}^{-1} \mathbf{x}_2)^T (\mathbf{K}^{-1} \mathbf{x}_2)}} \\ &= \frac{\mathbf{x}_1^T (\mathbf{K}^{-T} \mathbf{K}^{-1}) \mathbf{x}_2}{\sqrt{\mathbf{x}_1^T (\mathbf{K}^{-T} \mathbf{K}^{-1}) \mathbf{x}_1} \sqrt{\mathbf{x}_2^T (\mathbf{K}^{-T} \mathbf{K}^{-1}) \mathbf{x}_2}} \end{aligned} \quad (2.15)$$

Egy kalibrált kamerát tehát felfoghatunk egyfajta *irányszenzorként*, amely képes mérni a vetítősugarak irányát.

2.2.2. Az ideális kúpszelet képe

Az Ω_∞ ideális kúpszelet egy olyan speciális másodrendű görbe a $\pi_\infty = (0,0,0,1)^T$ végtelen távoli síkon, melynek pontjai kielégítik az alábbi két egyenletet:

$$x_1^2 + x_2^2 + x_3^2 = 0 \quad (2.16)$$

$$x_4 = 0 \quad (2.17)$$

A második egyenlet lényegében azt jelenti, hogy Ω_∞ pontjai a π_∞ végtelen távoli síkon találhatóak, tehát $(x_1, x_2, x_3, 0)^T = (\mathbf{d}^T, 0)^T$ alakú irányok. Ezen pontokra az első egyenletet felírhatjuk mátrix alakban is:

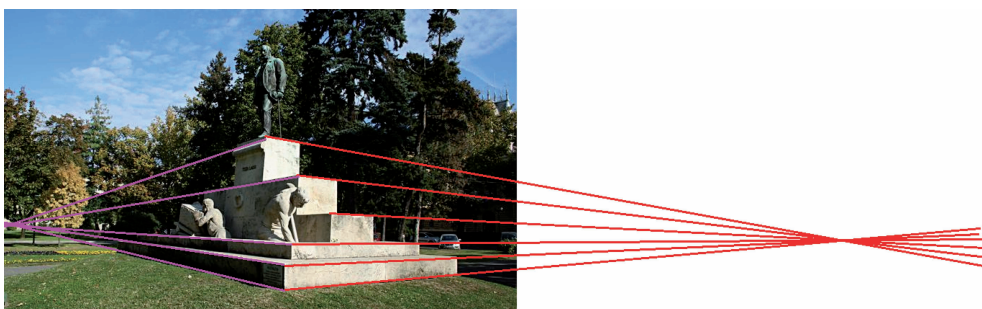
$$(x_1, x_2, x_3) \mathbf{I} (x_1, x_2, x_3)^T = \mathbf{d}^T \mathbf{I} \mathbf{d} = 0 \quad (2.18)$$

Ω_∞ tehát egy olyan kúpszelet, melynek mátrixa az \mathbf{I} egységmátrix és valamennyi pontja a π_∞ képzetes pontja. Ebből következik, hogy Ω_∞ képe a 2.11 képlet szerinti síkhomográfiával áll elő, vagyis ha $\mathbf{x} = \mathbf{KRd}$, akkor

$$\mathbf{d}^T \mathbf{I} \mathbf{d} = \mathbf{x}^T (\mathbf{KR})^{-T} \mathbf{I} (\mathbf{KR})^{-1} \mathbf{x} = \mathbf{x}^T (\mathbf{K}^{-T} \mathbf{R} \mathbf{R}^{-1} \mathbf{K}^{-1}) \mathbf{x} = \mathbf{x}^T (\mathbf{K}^{-T} \mathbf{K}^{-1}) \mathbf{x} \quad (2.19)$$

Az Ω_∞ ideális kúpszelet képe tehát egy olyan ω kúpszelet, melynek mátrixa

$$\omega = \mathbf{K}^{-T} \mathbf{K}^{-1} = (\mathbf{K} \mathbf{K}^T)^{-1} \quad (2.20)$$



2.1. ábra. Párhuzamos egyenesek képe a képsíkon egy pontban metszik egymást, amit az egyenesek eltűnési pontjának nevezünk.

és ω pontjai is képzeteselek. A fenti összefüggés fontos következménye, hogy ω kizárólag a kamera belső paramétereitől függ. Ha ω ismert, akkor abból Cholesky felbontással egyértelműen meghatározhatjuk a kamera kalibrációs mátrixát. Végezetül vegyük észre, hogy a 2.15 képlet felírható ω segítségével is:

$$\cos(\theta) = \frac{\mathbf{x}_1^T \omega \mathbf{x}_2}{\sqrt{\mathbf{x}_1^T \omega \mathbf{x}_1} \sqrt{\mathbf{x}_2^T \omega \mathbf{x}_2}} \quad (2.21)$$

A fenti összefüggés egy fontos speciális esete, amikor a két képpont ortogonális irányoknak felel meg, ekkor ugyanis

$$\mathbf{x}_1^T \omega \mathbf{x}_2 = 0 \quad (2.22)$$

Ha ismertek ilyen képpontok, akkor azokból ω -ra nézve lineáris egyenleteket konstruálhatunk.

2.2.3. Eltűnési pont

A projektív térben a párhuzamos egyenesek végtelen távoli síkon vett dőféspontja egyetlen közös pont lesz, mégpedig az egyenesek iránya. Jól ismert jelenség, hogy a párhuzamos egyenes képe a projektív kamerában összetartó egyeneseket produkál, melyek egy pontban metszik egymást – ez a pont nem más, mint a végtelen távoli síkon lévő irány képe, melyet az egyenesek *eltűnési pontjának* nevezünk (ld. 2.1 ábra). Mindezt formálisan is levezethetjük: Tekintsük az \mathbf{A} ponton átmenő $\mathbf{D} = (\mathbf{d}^T, 0)^T$ irányú $\mathbf{X}(\lambda) = \mathbf{A} + \lambda \mathbf{D}$ 3D egyenest. Az egyenes képét a $\mathbf{P} = \mathbf{K}[\mathbf{I}|\mathbf{0}]$ kamrában az alábbiak szerint kapjuk:

$$\mathbf{x}(\lambda) = \mathbf{P}\mathbf{A} + \lambda \mathbf{P}\mathbf{D} = \mathbf{a} + \lambda \mathbf{K}\mathbf{d} \quad (2.23)$$

ahol \mathbf{a} az \mathbf{A} képe. Az egyenes \mathbf{v} eltűnési pontját úgy kapjuk, hogy vesszük a $\lambda \rightarrow \infty$ határértéket, ami az egyenesen a \mathbf{D} pontnak felel meg:

$$\mathbf{v} = \lim_{\lambda \rightarrow \infty} \mathbf{x}(\lambda) = \lim_{\lambda \rightarrow \infty} (\mathbf{a} + \lambda \mathbf{K}\mathbf{d}) = \mathbf{K}\mathbf{d} \quad (2.24)$$

Megjegyezzük, hogy a fenti egyenlőség a homogén osztás elvégzése után teljesül pontosan, egyébként egy skálafaktor erejéig igaz. Az eltűnési pont kizárólag az egyenes irányától függ, tehát bármely ponton átmenő, de \mathbf{D} irányú egyenes eltűnési pontja ugyanaz a $\mathbf{K}\mathbf{d}$ pont lesz.

Ebből következik, hogy a \mathbf{v} eltűnési ponthoz tartozó vetítő sugár szükségszerűen párhuzamos lesz a 3D térben az összes \mathbf{D} irányú egyenessel, hiszen az pontosan a \mathbf{C} ponton átmenő \mathbf{D} irányú egyenes lesz.

Eltűnési pontok segítségével könnyen meghatározhatjuk a kamera \mathbf{R} forgatási mátrixát. Mivel a végtelen távoli sík mindig megjelenik a kamera képében és ez független a kamera pozíciójától, ezért ugyanazon kamera két különböző pozícióban és irányban készített képén megtaláljuk ugyanazon eltűnési pontot. Jelölje ezt rendre \mathbf{v} illetve \mathbf{v}' a két képen. Az eltűnési pontokhoz tartozó vetítősugarak normalizált irányvektorait megkaphatjuk a jól ismert $\mathbf{d} = \mathbf{K}^{-1}\mathbf{v}/\|\mathbf{K}^{-1}\mathbf{v}\|$ illetve $\mathbf{d}' = \mathbf{K}^{-1}\mathbf{v}'/\|\mathbf{K}^{-1}\mathbf{v}'\|$ alakban. A két irányvektor között pontosan a két kamera közötti \mathbf{R} forgatás hat, vagyis $\mathbf{d}' = \mathbf{R}\mathbf{d}$, amiből \mathbf{R} elemeire 2 független egyenletet kapunk. Két ilyen pontpárból már elegendő lineáris egyenletet kapunk \mathbf{R} meghatározására.

Két egyenes által bezárt szöget is könnyen meghatározhatjuk az egyenes \mathbf{v}_1 illetve \mathbf{v}_2 eltűnési pontjának ismeretében, hiszen a pontokhoz tartozó vetítő egyenesek a 3D térben párhuzamosak az egyenesekkel, és a két sugár által bezárt szöget az ideális kúpszelet képének ismeretében a 2.21 képlet segítségével könnyen megkaphatjuk:

$$\cos(\theta) = \frac{\mathbf{v}_1^T \omega \mathbf{v}_2}{\sqrt{\mathbf{v}_1^T \omega \mathbf{v}_1} \sqrt{\mathbf{v}_2^T \omega \mathbf{v}_2}} \quad (2.25)$$

Merőleges egyenesek esetén teljesül a 2.22 képlet szerinti összefüggés:

$$\mathbf{v}_1^T \omega \mathbf{v}_2 = 0 \quad (2.26)$$

Természetesen az eltűnési pontok képzetesek, ezért látható képet nem generálnak, de kiszámíthatóak. Ezt legegyszerűbben úgy tehetjük meg, ha ismert legalább két párhuzamos egyenes képe, melyek metszéspontjaként előáll azok eltűnési pontja. Mivel a képi mérések mindig tartalmaznak legalább diszkretizálási hibát, ezért több egyenes esetén nem kapunk közös metszéspontot. Egy megoldási lehetőség az összes egyenespár metszéspontját meghatározni, majd ezen pontok centroidját tekinteni az eltűnési pontnak.

2.2.4. Eltűnési egyenes

A projektív térben a párhuzamos síkok egyetlen egyenesben metszik a végtelen távoli síkot. Ennek az egyenesnek a képe a párhuzamos síkok *eltűnési egyenese*. A *horizont* például egy olyan, mindenki által jól ismert, eltűnési egyenes, amely a föld és az ég találkozásának képe. Nyilvánvalóan egy adott síkon belüli párhuzamos egyenesek eltűnési pontja a sík eltűnési egyenesén található, akár csak valamennyi, a síkkal párhuzamos egyenes eltűnési pontja.

Ha ismert a kamera kalibrációs mátrixa, akkor könnyen kiszámolhatjuk egy sík normálvektorát a sík eltűnési egyenesének ismeretében. A 2.1.2 fejezetben ismertetett összefüggések alapján könnyen belátható, hogy a kamera középponton átmenő, \mathbf{n} normálisú sík a képsíkot az $\mathbf{l} = \mathbf{K}^{-T} \mathbf{n}$ egyenesben metszi. Következésképpen \mathbf{l} valamennyi, \mathbf{n} normálisú sík eltűnési egyenese lesz, vagyis ezen síkok normálisa $\mathbf{n} = \mathbf{K}^T \mathbf{l}$ lesz.

Az eltűnési pontokhoz hasonlóan, két sík által bezárt szöget is könnyedén meghatározhatjuk a síkok eltűnési egyeneseinek ismeretében, hiszen:

$$\cos(\theta) = \frac{\mathbf{l}_1^T \boldsymbol{\omega}^* \mathbf{l}_2}{\sqrt{\mathbf{l}_1^T \boldsymbol{\omega}^* \mathbf{l}_1} \sqrt{\mathbf{l}_2^T \boldsymbol{\omega}^* \mathbf{l}_2}} \quad (2.27)$$

ahol $\boldsymbol{\omega}^*$ az $\boldsymbol{\omega}$ érintőivel definiált duálisa. Merőleges síkok esetén az alábbi összefüggés teljesül:

$$\mathbf{l}_1^T \boldsymbol{\omega}^* \mathbf{l}_2 = 0 \quad (2.28)$$

Továbbá egy merőleges egyenes és sík között az alábbi összefüggések teljesülnek:

$$\mathbf{l} = \boldsymbol{\omega} \mathbf{v} \text{ illetve } \mathbf{v} = \boldsymbol{\omega}^* \mathbf{l} \quad (2.29)$$

Az eltűnési egyenesek az eltűnési ponthoz hasonlóan képzetesek. Eltűnési egyenest könnyen meghatározhatunk az egyenesre eső eltűnési pontok ismeretében, hiszen így a feladatot megoldhatjuk egy kalsszikus egyenes illesztéssel adott pontokra. Ehhez természetesen szükség van legalább két, a síkkal párhuzamos egyenes halmazra. Numerikusan azok az egyenes halmazok kedvezőek, amelyek nagyobb szöget zárnak be egymással, hiszen a kis szöget bezáró egyenes-halmazok eltűnési pontja túl közel eshet egymáshoz, ami azután az eltűnési egyenes illesztését bizonytalanná teheti.

2.3. A projektív tér elemei és a kalibráció kapcsolata

A továbbiakban megvizsgáljuk a kalibrációs mátrix és a projektív tér képzetes elemei közötti kapcsolatot. Látni fogjuk, hogy ezekből az összefüggésekből meghatározhatjuk a kamera kalibrációs mátrixát a kamera által készített egyetlen kép segítségével, speciális kalibrációs minta használata nélkül.

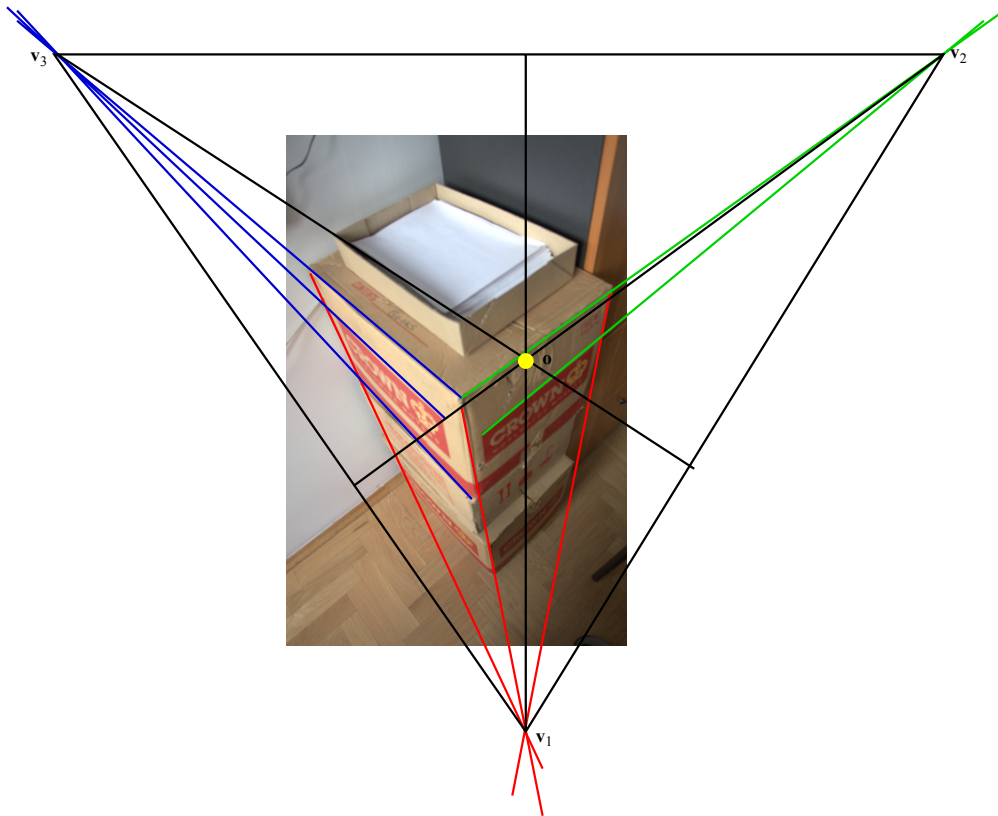
2.3.1. Kalibráció az ideális kúpszelet képéből

A legegyszerűbben az ideális kúpszelet képéből tudjuk előállítani a kalibrációs mátrixot, hiszen közöttük a 2.2.2 fejezetben tárgyalt $\boldsymbol{\omega} = (\mathbf{K}\mathbf{K}^T)^{-1}$ közvetlen összefüggés van. $\boldsymbol{\omega}$ ismeretében tehát \mathbf{K} -t megkaphatjuk Cholesky felbontással.

Az ideális kúpszelet képét egyszerűen meghatározhatjuk páronként merőleges egyenes-halmazok eltűnési pontjainak segítségével a 2.22 képlet felhasználásával, továbbá figyelembe vehetünk a kamera belső paramétereire vonatkozó megszorításokat. Ezek tipikusan négyzetes pixeleket írnak elő, vagyis a kalibrációs mátrixban (ld. 1.1.2 fejezet)

1. $a = 0 \Rightarrow \omega_{12} = \omega_{21} = 0$ és
2. $s_x = s_y \Rightarrow \omega_{11} = \omega_{22}$.

Vegyük észre, hogy valamennyi feltétel lineáris az $\boldsymbol{\omega}$ elemeiben, ezért ezekből egyszerűen konstruálhatunk egy lineáris egyenletrendszert, melynek megoldása közvetlenül megadja



2.2. ábra. Három merőleges irány \mathbf{v}_1 , \mathbf{v}_2 , \mathbf{v}_3 eltűnési pontja által meghatározott háromszög \mathbf{o} magasságpontja lesz a kamera főpontja.

az ω ismeretlen elemeit. Figyelembe véve ω szimmetriáját, összesen 6 ismeretlenünk van. Ha négyzetes pixeleket tételezünk fel, akkor a fentiek alapján már van 3 egyenletünk, és további 3 egyenletet kaphatunk 3, páronként merőleges irányoknak megfelelő eltűnési pontból a 2.22 képlet segítségével.

2.3.2. Eltűnési pontok és a kamera főpontja

Ha adott 3 páronként merőleges irányoknak megfelelő eltűnési pont: \mathbf{v}_1 , \mathbf{v}_2 , \mathbf{v}_3 , akkor az általuk meghatározott háromszög magasságpontja lesz a a kamera főpontja (ld. 2.2 ábra). Egy háromszög magasságpontja nem más, mint a 3 magasság közös metszéspontja (hasznos interaktív ábra [itt](#)). Tekintsük most a 2.2 ábra \mathbf{v}_1 , \mathbf{v}_2 , \mathbf{v}_3 háromszögét. Mivel a háromszög oldalait 2–2 eltűnési pont határozza meg, ezért ezek eltűnési egyenesek lesznek. Egy ilyen egyenest visszavetítve egy \mathbf{n} normálisú $\mathbf{\pi}$ síkot kapunk, amely tartalmazza a \mathbf{C} kamera középpontot is. A szemben lévő \mathbf{v} eltűnési pontot visszavetítve pedig egy egyenest kapunk, amely merőleges lesz a $\mathbf{\pi}$ síkra és amely szintén tartalmazza a \mathbf{C} kamera középpontot. A kamera optikai tengelye a képsíkot az \mathbf{o} főpontban dőfi át. Tekintsük most a \mathbf{C} , \mathbf{o} , és \mathbf{v} által meghatározott síkot, amely a képsíkot egy \mathbf{l} egyenesben metszi, míg az eltűnési egyenest egy \mathbf{x} pontban. Nyilvánvalóan \mathbf{o} is \mathbf{l} -en van, és \mathbf{l} merőleges az eltűnési egyenesre. Vegyük észre, hogy ezek



2.3. ábra. A projektív kép tulajdonságait kihasználva kalibrálhatjuk a kameránkat 3 nem egy síkba eső, de egyébként nem feltétlenül derékszögben elhelyezett négyzetről készített kép alapján.

az egyenesek alkotják a \mathbf{v}_1 , \mathbf{v}_2 , \mathbf{v}_3 háromszög magasságait, és mivel \mathbf{o} mindhármukon rajta van, ezért szükségképpen \mathbf{o} a metszéspontjuk, vagyis a háromszög magasságpontja.

2.3.3. Kalibráció 3 sík képéből

A projektív kép tulajdonságait kihasználva kalibrálhatjuk a kameránkat egy egyszerű, háziilag elkészíthető kalibrációs eszköz segítségével. A kalibrációhoz 3 nem egy síkba eső, de egyébként nem feltétlenül derékszögben elhelyezett négyzetről készítünk képet (a négyzete-t egyszerűen nyomtatóval állíthatjuk elő, ld. 2.3 ábra), amely elegendő feltételt szolgáltat ω meghatározásához. Tekintsünk először egy négyzetet, melynek térbeli síkját jelölje $\boldsymbol{\pi}$. A négyzet 4 sarokpontja és azok képe által definiált pontpárok elegendő feltételt szolgáltatnak a képsík és $\boldsymbol{\pi}$ közötti \mathbf{H} síkhomográfia meghatározásához. Vegyük fel úgy a koordináta rendszerünket, hogy abban a négyzet egységnyi legyen, vagyis a sarkainak koordinátái $(0,0)^T$, $(0,1)^T$, $(1,0)^T$, $(1,1)^T$, lesznek. Természetesen a $\boldsymbol{\pi}$ sík a projektív térben metszeni fogja a végtelen távoli síkot, és az azon található Ω_∞ ideális kúpszeletet két cirkuláris pontban: $(1, \iota, 0)^T$ és $(1, -\iota, 0)^T$. A cirkuláris pontok koordinátáit nem befolyásolja a koordináta rendszer illesztése a négyzetre, hiszen az csak egy hasonlósági transzformáció lesz, amely ezeket a pontokat helyben hagyja. A korábban kiszámolt \mathbf{H} síkhomográfíát alkalmazva a két cirkuláris pontra, kapunk két pontot az ω -n:

$$\mathbf{H} \begin{pmatrix} 1 \\ \pm \iota \\ 0 \end{pmatrix} = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] \begin{pmatrix} 1 \\ \pm \iota \\ 0 \end{pmatrix} = \mathbf{h}_1 \pm \mathbf{h}_2 \iota \quad (2.30)$$

A 3 négyzetből tehát a fenti módon kapunk 6 pontot, amiből ω -t meghatározhatjuk, hiszen – mivel a cirkuláris pontok képe az ω -n van – teljesül az alábbi összefüggés:

$$(\mathbf{h}_1 \pm \mathbf{h}_{2l})^T \omega (\mathbf{h}_1 \pm \mathbf{h}_{2l}) = 0 \quad (2.31)$$

amiből a valós és képzetes részekre az alábbi egyenletek adódnak:

$$\mathbf{h}_1^T \omega \mathbf{h}_2 = 0 \text{ és } \mathbf{h}_1^T \omega \mathbf{h}_1 = \mathbf{h}_2^T \omega \mathbf{h}_2 \quad (2.32)$$

amelyek egy cirkuláris pontpárra két, ω -ban lineáris egyenleteket szolgáltatnak, így a 3 négyzet képéből ω meghatározható.

3. fejezet

Sztereoó képpár geometriája

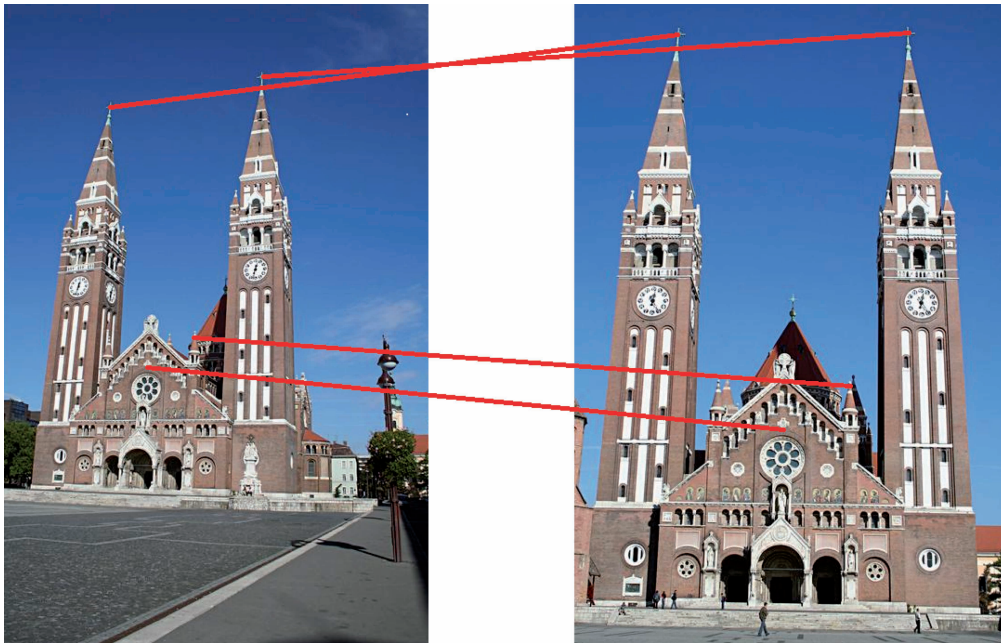
Az eddigiekben megismertük a projektív kamera és képének tulajdonságait. Ebben a fejezetben azt fogjuk megvizsgálni, hogy milyen összefüggések vannak két, ugyanazon 3D látványt leképező projektív kamera képe között. Az ilyen képpárokat *sztereoó képpárnak* nevezzük. Ezek az összefüggések szolgálnak majd a képek alapján történő 3D rekonstrukció alapjául, melyet a következő fejezetben fogunk tárgyalni.

3.1. Epipoláris geometria

Tekintsünk egy 3D \mathbf{X} pontot és annak \mathbf{x} és \mathbf{x}' képeit két projektív kamerában (ld. 3.2 ábra). A projektív kamera tulajdonságaiból adódik, hogy az \mathbf{x} és \mathbf{x}' képpontok az \mathbf{X} és a \mathbf{C} , \mathbf{C}' kamera középpontok által meghatározott π síkra esnek, amit *epipoláris síknak* nevezzük. Az is nyilvánvaló, hogy az \mathbf{x} és \mathbf{x}' képpontokat visszavetítve a vetítősugarak pontosan az \mathbf{X} pontban metszik egymást, és mindkét vetítősugár a π síkon van. A két kamera középpontot összelötő $\overline{\mathbf{CC}'}$ egyenest *bázis egyenesnek* nevezzük. A bázis egyenes szintén illeszkedik a π síkra, és mivel a kamera középpontok rögzítettek, ezért a bázis egyenes sem változik. Vegyük észre, hogy az epipoláris sík helyzete függ \mathbf{X} -től, de minden esetben illeszkedik a bázis egyenesre. Tehát két kamera epipoláris síkjai azon síkok halmaza lesz, amelyek illeszkednek a bázis egyenesre, lényegében a körül elforgatva π -t megkaphatjuk a többi epipoláris síkot (ld. 3.3 ábra).

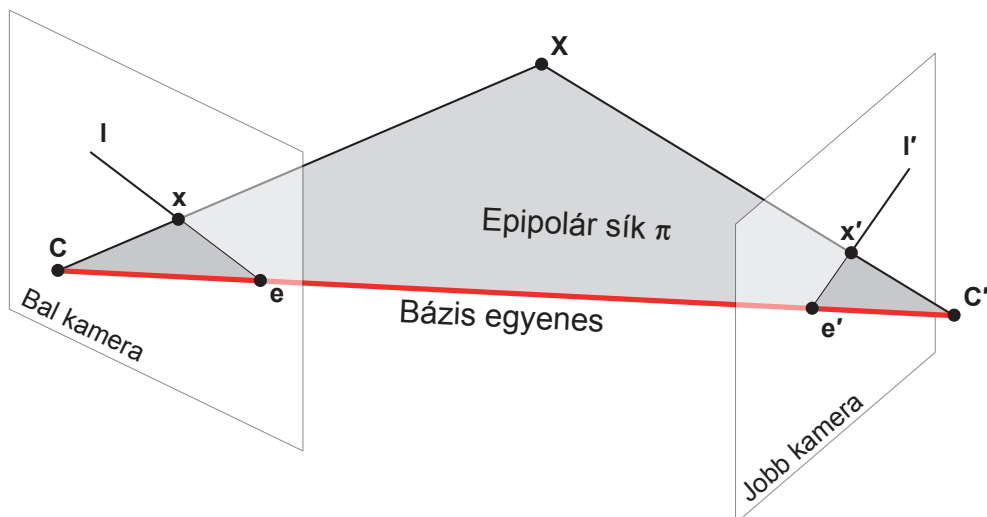
Mivel a bázis egyenes a két kamera középpontot köti össze, ezért azt tekinthetjük egy speciális vetítősugárnak, amely az egyik kamera középpont képét állítja elő a másik kamerában. Ezen \mathbf{e} , \mathbf{e}' pontokat *epipólusnak* nevezzük. Az epipólusok lényegében a bázisegyenes dőfspontjai a bal és jobb oldali képsíkkal, de tekinthetjük úgy is, mint a bázis egyenes eltűnési pontja a bal és jobb oldali kamerában.

Az epipoláris sík metszete a képsíkokkal egy-egy egyenes lesz. Ezeket az \mathbf{l} , \mathbf{l}' egyeneseket *epipoláris egyeneseknek* nevezzük. Az epipoláris egyenesek illeszkednek az epipólusokra, hiszen az epipoláris sík illeszkedik a bázisegyenesre. Ezért valamennyi epipoláris egyenes illeszkedik az epipólusokra, tehát az epipólust tartalmazó egyenesek halmaza adja a kép epipoláris egyeneseit, melyeket lényegében \mathbf{l} -nek \mathbf{e} körüli, illetve \mathbf{l}' -nek \mathbf{e}' körüli elforgatásával kapunk. Geometriai értelemben az \mathbf{l}' epipoláris egyenes nem más, mint a bal oldali kamera \mathbf{x} -hez tartozó vetítősugarának képe a jobb oldali kamerában, és természetesen hasonló össze-

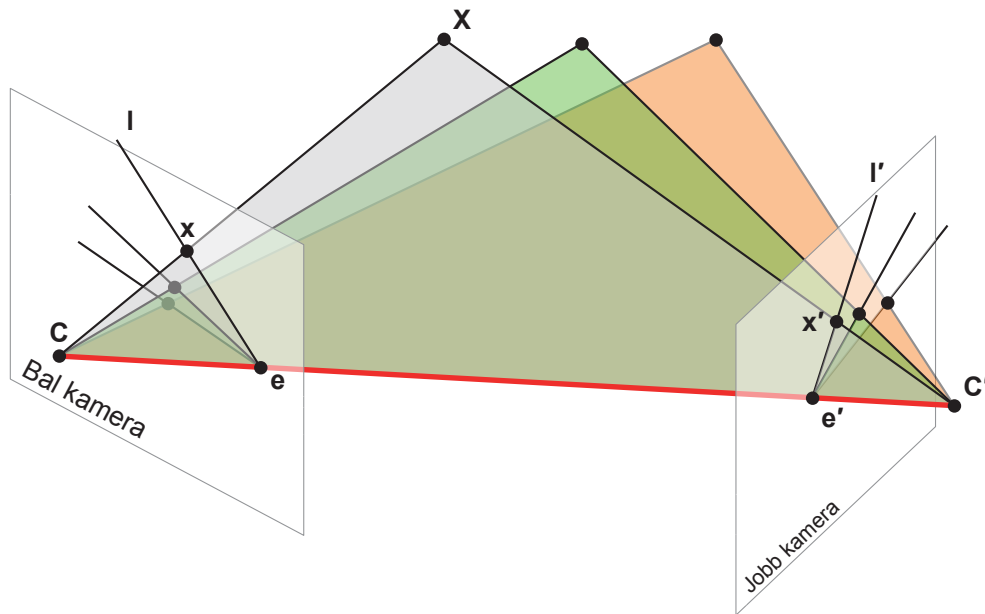


3.1. ábra. Két kamera által készített sztereó képpár esetén mindkét képen megtaláljuk ugyanazon 3D valós pont képét. Mi a geometriai kapcsolat ezen képpontok között?

függés áll fenn az l -re is. Ezekből adódik az a fontos összefüggés, amely szerint egy \mathbf{x} képpont \mathbf{x}' párja a másik kamerában az \mathbf{x} -hez tartozó l' epipoláris egyenesen található.



3.2. ábra. Két kamera képe között az epipoláris geometria teremt kapcsolatot. A bal és jobb oldali kamera középpontok képei az \mathbf{e} és \mathbf{e}' epipólusok, míg az \mathbf{x} és \mathbf{x}' pontokhoz tartozó epipoláris egyenesek rendre l' és l .



3.3. ábra. Az epipoláris síkok és epipoláris egyenesek helyzete \mathbf{X} -től függ.

3.2. A fundamentális mátrix

A fundamentális mátrix a legalapvetőbb epipoláris geometriai összefüggés algebrai megfogalmazása két kalibrálatlan kamera képe között. A fundamentális mátrixot viszonylag egyszerűen, néhány pontmegfeleltetés segítségével meghatározhatjuk, és abból azután megkaphatjuk a két kamera kanonikus alakját, sőt a két képből projektív rekonstrukciót is elő tudunk állítani.

Az előző fejezetben láttuk, hogy minden egyes \mathbf{x} képponthoz tartozik egy l' epipoláris egyenes a másik kamera képén, vagyis ez az epipoláris geometriai összefüggés egy $\mathbf{x} \mapsto l'$ leképezést definiál, ami lényegében a képpontok egyeneseknek történő projektív megfeleltetése lesz. Ezt a leképezést egy \mathbf{F} mátrix segítségével adhatjuk meg:

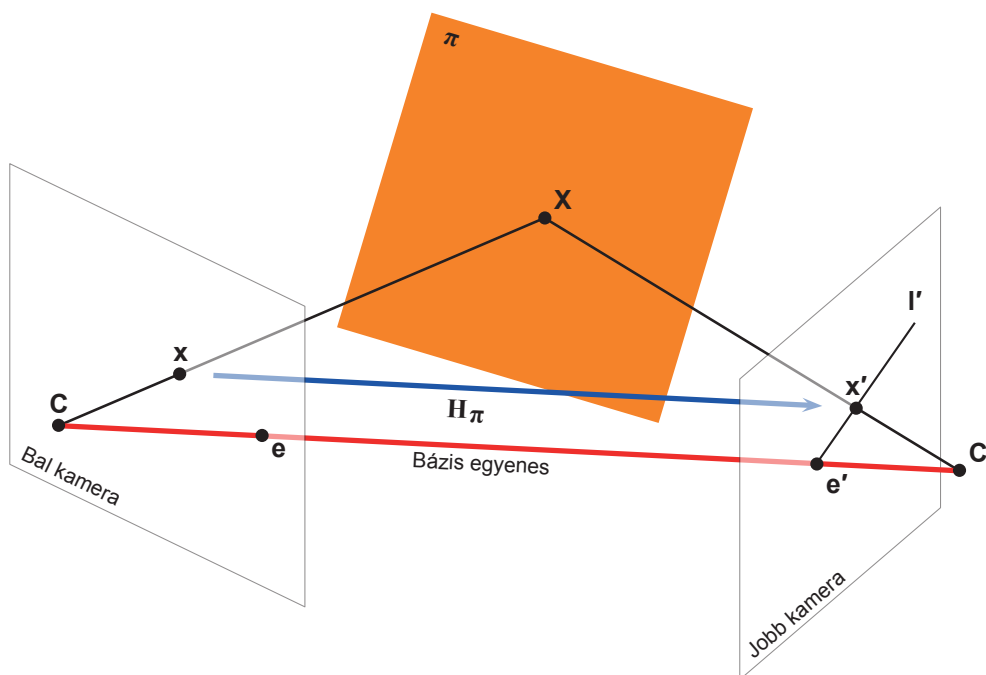
$$l' = \mathbf{F}\mathbf{x} \quad (3.1)$$

A 3×3 \mathbf{F} mátrixot *fundamentális mátrix*nak nevezünk. Egy síkhomográfiával ellentétben azonban \mathbf{F} rangja 2, így 7 a szabadsági foka és nem invertálható.

Tekintsünk egy $\mathbf{x} \leftrightarrow \mathbf{x}'$ pontpárt. Tudjuk, hogy \mathbf{x}' az l' epipoláris egyenesen van, vagyis kielégíti az $\mathbf{x}'^T l' = 0$ egyenletet. Másrészt l' -t kifejezhetjük a fundamentális mátrix segítségével a 3.1 képlet alapján, így megkapjuk egy képpár valamennyi $\mathbf{x} \leftrightarrow \mathbf{x}'$ pontmegfeleltetéseire vonatkozó legalapvetőbb összefüggést:

$$\mathbf{x}'^T \mathbf{F}\mathbf{x} = 0 \quad (3.2)$$

A fundamentális mátrix meghatározásához vegyünk egy tetszőleges π síkot, amely egyik kamera középpontra sem illeszkedik (ld. 3.4 ábra). Ekkor az egyik kamerából visszavetítve a \mathbf{x} pontot, a vetítésugar \mathbf{X} -ben metszi a π síkot. Ezt a dőféspontot azután leképezzük a másik



3.4. ábra. Egy π sík által generált \mathbf{H}_π síkhomográfia: $\mathbf{x}' = \mathbf{H}_\pi \mathbf{x}$.

kamerában, ahol \mathbf{x}' lesz a képe, tehát \mathbf{x} és \mathbf{x}' ugyanannak a pontnak a képei a két kamerában, vagyis projektíven ekvivalensek. Mivel \mathbf{X} rajta van a \mathbf{x} ponthoz tartozó vetítő egyenesen, ezért \mathbf{x}' rajta lesz az l' epipoláris egyenesen, hiszen az pontosan a vetítősugár képe a másik kamerában. Valamennyi ilyen \mathbf{x} , \mathbf{x}' pontpár projektíven ekvivalens lesz tehát, az így kapott \mathbf{H}_π síkhomográfiait pedig a π sík által generált homográfianak nevezzük. Ezek alapján egy \mathbf{x} ponthoz tartozó l' epipoláris egyenest megszerkeszthetünk úgy, mint az \mathbf{x}' ponton és az \mathbf{e}' epipóluson átmenő egyenes, vagyis:

$$l' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_{\times} \mathbf{x}' \quad (3.3)$$

ahol $[\mathbf{e}']_{\times}$ a keresztszorzatot kiváltó, \mathbf{e}' -ből képzett antiszimmetrikus mátrix:

$$[\mathbf{e}']_{\times} = \begin{bmatrix} 0 & -e'_3 & e'_2 \\ e'_3 & 0 & -e'_1 \\ -e'_2 & e'_1 & 0 \end{bmatrix} \quad (3.4)$$

Mivel $\mathbf{x}' = \mathbf{H}_\pi \mathbf{x}$, ezért a fenti képletet átírhatjuk az alábbi alakban is, amiből megkapjuk az \mathbf{F} mátrixot:

$$l' = [\mathbf{e}']_{\times} \mathbf{H}_\pi \mathbf{x} = \mathbf{F} \mathbf{x} \implies \mathbf{F} = [\mathbf{e}']_{\times} \mathbf{H}_\pi \quad (3.5)$$

Természetesen a fundamentális mátrix létezésének nem feltétlenül egy ilyen π sík megléte, hiszen annak szerepe csupán a két kép közötti megfeleltetés előállítására.

Ahogy a továbbiakban látni fogjuk, \mathbf{F} -et felírhatjuk a két kamera mátrix segítségével is. Ehhez a jól ismert $\mathbf{x} = \mathbf{P}\mathbf{X}$ egyenletet megoldva az alábbi, egy szabad paraméterrel rendelkező megoldást kapjuk:

$$\mathbf{X}(\lambda) = \mathbf{P}^+ \mathbf{x} + \lambda \mathbf{C} \quad (3.6)$$

ahol $\mathbf{P}^+ = \mathbf{P}^T (\mathbf{P}\mathbf{P}^T)^{-1}$ a \mathbf{P} kamera mátrix pszeudo inverze. A fenti képlet geometriai értelemben a \mathbf{C} kamera középponton átmenő, a $\mathbf{P}^+\mathbf{x}$ pontot tartalmazó vetítőegyenes egyenlete. Ezen két pont képét a másik kamerában $\mathbf{P}'\mathbf{P}^+\mathbf{x}$ illetve $\mathbf{P}'\mathbf{C}$ alakban kapjuk. A fenti vetítésugár képét pedig megszerkeszthetjük úgy, mint a két ponton átmenő egyenest:

$$\mathbf{l}' = (\mathbf{P}'\mathbf{C}) \times (\mathbf{P}'\mathbf{P}^+\mathbf{x}) \quad (3.7)$$

Vegyük észre, hogy a kamera középpont képe az \mathbf{e}' epipólust adja, amit behelyettesítve a fenti képletbe megkapjuk az \mathbf{F} mátrixot:

$$\mathbf{l}' = [\mathbf{e}']_{\times} (\mathbf{P}'\mathbf{P}^+)\mathbf{x} \implies \mathbf{F} = [\mathbf{e}']_{\times} \mathbf{P}'\mathbf{P}^+ \quad (3.8)$$

A fundamentális mátrixra kapott két képlet lényegében megegyezik, hiszen a fenti képletben $\mathbf{P}'\mathbf{P}^+$ nem más, mint a 3.5 képlet \mathbf{H}_{Π} síkhomográfiája. A fenti képletből az is kiderül, hogy egy $(\mathbf{P}, \mathbf{P}')$ kamera pár egyértelműen meghatározza a közöttük ható fundamentális mátrixot. A fordított állítás azonban nem teljesül. Sőt, a fundamentális mátrix immunis a projektív homográfiára, vagyis ha a $(\mathbf{P}, \mathbf{P}')$ kamerák között az \mathbf{F} mátrix hat, akkor ugyanez lesz a fundamentális mátrixa a $(\mathbf{P}\mathbf{H}, \mathbf{P}'\mathbf{H})$ kameráknak, ahol \mathbf{H} egy maximális rangú projektív transzformáció (vagyis egy homográfia). Ha ugyanis $\mathbf{x} \leftrightarrow \mathbf{x}'$ pontmegfeleltetések az \mathbf{X} pont képei a $(\mathbf{P}, \mathbf{P}')$ kamerákban, akkor ugyanezen pontok lesznek a képei a $\mathbf{H}^{-1}\mathbf{X}$ pontnak a $(\mathbf{P}\mathbf{H}, \mathbf{P}'\mathbf{H})$ kamerákban, hiszen $\mathbf{x} = \mathbf{P}\mathbf{X} = (\mathbf{P}\mathbf{H})(\mathbf{H}^{-1}\mathbf{X})$, másrészt $\mathbf{x}'^T \mathbf{F}\mathbf{x} = 0$, vagyis mindkét kamerapár esetén ugyanaz az \mathbf{F} fundamentális mátrix teremt kapcsolatot a két kép között.

3.2.1. A fundamentális mátrix kiszámítása

Az előző fejezetben láttuk, hogy minden $\mathbf{x} \leftrightarrow \mathbf{x}'$ pontpárra teljesül a 3.2 képlet, ami egy lineáris egyenletet ad \mathbf{F} elemeire:

$$x'_1 x_1 f_{11} + x'_1 x_2 f_{12} + x'_1 f_{13} + x'_2 x_1 f_{21} + x'_2 x_2 f_{22} + x'_2 f_{23} + x_1 f_{31} + x_2 f_{32} + f_{33} = 0 \quad (3.9)$$

ahol $\mathbf{x} = (x_1, x_2, 1)^T$, $\mathbf{x}' = (x'_1, x'_2, 1)^T$ és \mathbf{F} elemeit f_{ij} jelöli. Az ismeretleneket sorfolytonosan egy 9 elemű \mathbf{f} vektorként írva, a fenti egyenletet az alábbi két vektor skaláris szorzataként írhatjuk:

$$(x'_1 x_1, x'_1 x_2, x'_1, x'_2 x_1, x'_2 x_2, x'_2, x_1, x_2, 1) \mathbf{f} = 0 \quad (3.10)$$

n pontmegfeleltetésből tehát egy n egyenletből álló lineáris egyenletrendszer kapunk \mathbf{f} -re:

$$\mathbf{A}\mathbf{f} = 0 \quad (3.11)$$

ahol \mathbf{A} sorai az egyes pontpárokból a 3.10 képlet szerint állnak elő. Az egyenletrendszernek akkor van (egy skálafaktor erejéig) egyértelmű megoldása, ha \mathbf{A} rangja 8. Mivel \mathbf{A} -nak 9 oszlopa van, ezért a rangja 9 is lehet. Ebben az esetben csak a *legkisebb négyzetes hiba* szerint lesz megoldás, ami minimalizálja az egyenletrendszer $\|\mathbf{A}\mathbf{f}\|$ algebrai hibáját és kielégíti a $\|\mathbf{f}\| = 1$ feltételt. A megoldást az \mathbf{A} SVD felbontásából kaphatjuk meg (a legkisebb szinguláris értékhez tartozó szinguláris vektor lesz a megoldás).

Van azonban még egy fontos feltétel \mathbf{F} -re: A fundamentális mátrix rangja 2, vagyis $\det(\mathbf{F})=0$. Ha ez nem teljesül, akkor az epipoláris egyenesek nem fogják egy pontban metszeni

egymást. A fenti egyenletrendszer megoldása általában nem elégíti ki ezt a feltételt, ezért ezt külön kell kezelni. A legegyszerűbb megoldás az egyenletrendszer megoldásaként kapott $\hat{\mathbf{F}}$ mátrixhoz legközelebbi, kettes rangú mátrixot venni, vagyis azt az \mathbf{F} mátrixot, amelyre a $\|\hat{\mathbf{F}} - \mathbf{F}\|$ Frobenius norma minimális, és $\det(\mathbf{F}) = 0$. Ezt szintén SVD felbontással állíthatjuk elő. Legyen $\hat{\mathbf{F}} = U \text{diag}(d_1, d_2, d_3) V^T$, ahol az SVD felbontás tulajdonságából következően $d_1 \geq d_2 \geq d_3$. A legkisebb szinguláris értéket kinullázva megkapjuk azt az $\mathbf{F} = U \text{diag}(d_1, d_2, 0) V^T$ mátrixot, amely kielégíti mindkét feltételt.

Mivel a fundamentális mátrix szabadsági foka 7, ezért minimálisan 7 lineárisan független egyenletre van szükségünk, vagyis \mathbf{A} egy 7×9 mátrix lesz, melynek rangja 7. Ebben az esetben azonban az egyenletrendszer megoldása 2 dimeziós lesz, amit $\mathbf{F}_1 + \alpha \mathbf{F}_2$ alakban írhatunk, ahol \mathbf{F}_1 és \mathbf{F}_2 az \mathbf{A} nullterét kifeszítő két megoldás. A megoldásnak azonban szingulárisnak kell lennie, amit $\det(\mathbf{F}_1 + \alpha \mathbf{F}_2) = 0$ alakban fejezhetünk ki. Ebből α -ra egy harmadfokú polinom egyenletet kapunk, amelynek 1 vagy 3 valós megoldása lesz, amit behelyettesítve az eredeti egyenletbe megkapjuk a keresett fundamentális mátrixot (ha esetleg több megoldás van, akkor is csak egy lesz geometriailag helyes). α meghatározható az $\mathbf{F}_2^{-1} \mathbf{F}_1$ valós sajátértékeiként is, hiszen $\det(\mathbf{F}_1 + \alpha \mathbf{F}_2) = \det(\mathbf{F}_2) \det(\mathbf{F}_2^{-1} \mathbf{F}_1 + \alpha \mathbf{I}) = 0$.

2. Algoritmus: Normalizált 8 pontos algoritmus

Input : $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}_{i=1}^N$, $N \geq 8$ pontmegfeleltetés

Output: \mathbf{F} fundamentális mátrix

- 1 Normalizáljuk az \mathbf{x}_i és \mathbf{x}'_i pontokat rendre egy \mathbf{M} és egy \mathbf{N} hasonlósági transzformációval az 1.3.3 fejezetben tárgyalt módon.
 - 2 A normalizált pont koordinátákból a 3.10 képlet alapján konstruált $\mathbf{A}\mathbf{f} = \mathbf{0}$, $\|\mathbf{f}\| = 1$ egyenletrendszer megoldását az \mathbf{A} mátrix legkisebb szinguláris értékéhez tartozó egységnyi normájú szinguláris vektora adja, melyet egyszerűen megkaphatunk az \mathbf{A} SVD felbontásából.
 - 3 Határozzuk meg SVD felbontás segítségével az előző lépésben kapott $\hat{\mathbf{F}} = U \text{diag}(d_1, d_2, d_3) V^T$ megoldáshoz legközelebbi szinguláris $\tilde{\mathbf{F}} = U \text{diag}(d_1, d_2, 0) V^T$ mátrixot. Ez lesz a *normalizált* képek között ható fundamentális mátrix.
 - 4 Az előző lépésben kapott $\tilde{\mathbf{F}}$ megoldásból a keresett fundamentális mátrixot a 3.12 képlet szerinti denormalizálással kapjuk.
-

A legegyszerűbb megoldást azonban a *normalizált 8 pontos algoritmus* (ld. 2 algoritmus) szolgáltatja. Ehhez legalább 8 pontmegfeleltetésre van szükségünk, amelyekből a 3.10 képlet alapján lineáris egyenletrendszert konstruálunk, amit a *legkisebb négyzetes hiba* szerint oldunk meg. A megoldás numerikus stabilitása szempontjából kritikus a képpont koordináták megfelelő normalizálása, amely az 1.3.3 fejezetben tárgyalt módon végezhető el. Az eredményül kapott mátrixot itt is denormalizálnunk kell, hogy megkapjuk az eredeti képpárok között ható \mathbf{F} mátrixot. Ha az \mathbf{x}_i és \mathbf{x}'_i pontokat rendre egy \mathbf{M} és egy \mathbf{N} hasonlósági transzformációval normalizáljuk, és a normalizált pontok között ható fundamentális mátrixot $\tilde{\mathbf{F}}$ jelöli, akkor \mathbf{F} és $\tilde{\mathbf{F}}$ között az alábbi összefüggés áll fent:

$$0 = (\mathbf{N}\mathbf{x}'_i)^T \tilde{\mathbf{F}}(\mathbf{M}\mathbf{x}_i) = \mathbf{x}'_i{}^T (\mathbf{N}^T \tilde{\mathbf{F}} \mathbf{M}) \mathbf{x}_i \implies \mathbf{F} = \mathbf{N}^T \tilde{\mathbf{F}} \mathbf{M} \quad (3.12)$$

3.2.2. A fundamentális mátrix további tulajdonságai

Ha ismert a $(\mathbf{P}, \mathbf{P}')$ kamerák közötti \mathbf{F} fundamentális mátrix, akkor a $(\mathbf{P}', \mathbf{P})$ kamerák fundamentális mátrixa \mathbf{F}^T lesz. Ugyanis egy $\mathbf{x} \leftrightarrow \mathbf{x}'$ pontpárra teljesül az $\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$ összefüggés. Mindkét oldal transzponáltját véve kapjuk:

$$0 = (\mathbf{x}'^T \mathbf{F} \mathbf{x})^T = \mathbf{x}^T \mathbf{F}^T \mathbf{x}' \quad (3.13)$$

ami pontosan a fordított irányú összefüggést adja, ahol a fundamentális mátrix \mathbf{F}^T lesz. Továbbá az epipoláris egyenesekre is felírhatjuk a jól ismert összefüggést:

$$\mathbf{l}' = \mathbf{F} \mathbf{x} \implies \mathbf{l} = \mathbf{F}^T \mathbf{x}' \quad (3.14)$$

Az epipólusokat szintén meghatározhatjuk \mathbf{F} ismeretében. Mivel az epipólus valamennyi epipoláris egyenesen rajta van, ezért tetszőleges $\mathbf{x} \neq \mathbf{e}$ pontra teljesül:

$$0 = \mathbf{e}'^T \mathbf{l}' = \mathbf{e}'^T (\mathbf{F} \mathbf{x}) = (\mathbf{e}'^T \mathbf{F}) \mathbf{x} \quad (3.15)$$

Amiből következik, hogy

$$\mathbf{e}'^T \mathbf{F} = 0 \text{ illetve } \mathbf{F} \mathbf{e} = 0 \quad (3.16)$$

A fenti lineáris egyenleteket megoldva kiszámíthatjuk az \mathbf{e}, \mathbf{e}' epipólusokat.

Végezetül egy fontos speciális esetet vizsgálunk meg, amikor a két kamera között csak egy eltolás van. Ilyen helyzet a gyakorlatban is gyakran előfordul, amikor például ugyanazzal a kamerával készítünk két képet, az egyik pozícióból eltolva a kamerát a másikba – vagy rögzített kamera esetén magát a 3D objektumot eltolva ellenkező irányba. Később látni fogjuk, hogy egy ilyen kamera párnak kitüntetett szerepe lesz a 3D rekonstrukcióban is. Tekintsünk tehát egy $\mathbf{P} = \mathbf{K}[\mathbf{I}|\mathbf{0}]$ kamerát és annak \mathbf{t} -vel eltolt párját $\mathbf{P}' = \mathbf{K}[\mathbf{I}|\mathbf{t}]$. A 3.8 összefüggésből kapjuk:

$$\mathbf{F} = [\mathbf{e}']_{\times} \mathbf{P}' \mathbf{P}^+ = [\mathbf{e}']_{\times} \mathbf{K} \mathbf{K}^{-1} = [\mathbf{e}']_{\times} \quad (3.17)$$

Ha még az is teljesül, hogy az eltolás az x tengellyel párhuzamosan történt, akkor a kamerák képei egy közös síkra esnek, amellyel a bázisegyenes párhuzamos lesz. Ebből viszont az következik, hogy az epipólusok a végtelen távoli síkon vannak, vagyis $\mathbf{e}' = (1, 0, 0)^T$, és az epipoláris egyenesek megegyeznek a kép raszter-soraival.

3.3. Az esszenciális mátrix

Ha ismertek a kamerák kalibrációs mátrixai, akkor az általuk meghatározott fundamentális mátrix tovább egyszerűsíthető. Ehhez tekintsünk egy általános $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ kamera mátrixot. Mivel \mathbf{K} ismert, ezért annak inverzével szorozva a képpontokat megkapjuk azok normalizált koordinátáit $\tilde{\mathbf{x}} = \mathbf{K}^{-1} \mathbf{x}$. Mivel $\mathbf{x} = \mathbf{P} \mathbf{X}$, ezért

$$\tilde{\mathbf{x}} = \mathbf{K}^{-1} \mathbf{P} \mathbf{X} = \mathbf{K}^{-1} \mathbf{K} [\mathbf{R}|\mathbf{t}] \mathbf{X} = [\mathbf{R}|\mathbf{t}] \mathbf{X} \quad (3.18)$$

vagyis ezen $\tilde{\mathbf{x}}$ pontok a megfelelő \mathbf{X} képei lesznek egy olyan normalizált kamerában, melynek kalibrációs mátrixa az egységmátrix. Tekintsünk most két ilyen normalizált kamerát: $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$

illetve $\mathbf{P}' = [\mathbf{R}|\mathbf{t}]$. Az általuk meghatározott \mathbf{E} fundamentális mátrixot *esszenciális mátrix*nak nevezzük, és így egy $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ pontpárra teljesül a 3.2 szerinti összefüggés

$$\tilde{\mathbf{x}}'^T \mathbf{E} \tilde{\mathbf{x}} = 0 \quad (3.19)$$

A fenti összefüggés alapján könnyen kifejezhetjük \mathbf{E} -t \mathbf{F} segítségével:

$$0 = \tilde{\mathbf{x}}'^T \mathbf{E} \tilde{\mathbf{x}} = (\mathbf{K}'^{-1} \mathbf{x}')^T \mathbf{E} (\mathbf{K}^{-1} \mathbf{x}) = \mathbf{x}'^T (\mathbf{K}'^{-T} \mathbf{E} \mathbf{K}^{-1}) \mathbf{x} = \mathbf{x}'^T \mathbf{F} \mathbf{x} \quad (3.20)$$

vagyis

$$\mathbf{F} = \mathbf{K}'^{-T} \mathbf{E} \mathbf{K}^{-1} \implies \mathbf{E} = \mathbf{K}'^T \mathbf{F} \mathbf{K} \quad (3.21)$$

Tekintsük most a 3.8 képlet alapján az \mathbf{E} esszenciális mátrixot, ahol a kamera mátrixok a fentebb megadott normalizált kamerákkal egyenlőek, illetve az első kamera középpontja $\mathbf{C} = (0, 0, 0, 1)^T$:

$$\mathbf{E} = [\mathbf{e}']_{\times} \mathbf{P}' \mathbf{P}^+ = [\mathbf{P}' \mathbf{C}]_{\times} [\mathbf{R}|\mathbf{t}] \begin{bmatrix} \mathbf{I} \\ \mathbf{0}^T \end{bmatrix} = [\mathbf{t}]_{\times} \mathbf{R} \quad (3.22)$$

Vagyis az esszenciális mátrix a kamerák közötti transzformációtól függ. A mátrix szabadsági foka 5, hiszen, ahogy a fundamentális mátrix, \mathbf{E} is csak egy skálafaktor erejéig meghatározott. Az esszenciális mátrixot kiszámíthatjuk a 2 algoritmus segítségével a 3.19 képlet alapján, vagy a fundamentális mátrix ismeretében a 3.21 képlet felhasználásával is.

3.4. Standard sztereó konfiguráció

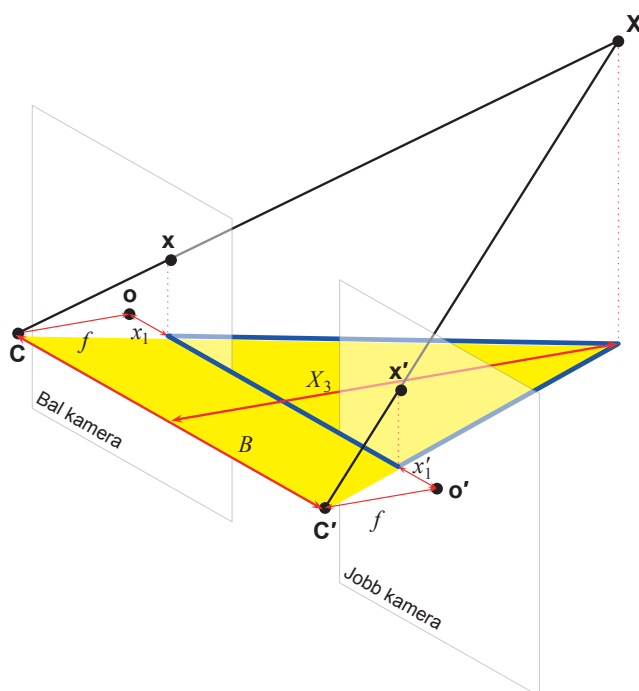
Standard sztereó konfiguráció alatt egy olyan speciális kamerapárt értünk, ahol a kamerák kalibrációs mátrixai megegyeznek, képsíkjaik közös képsíkra esnek és a kép koordináta rendszer tengelyei azonos irányúak (ld. 3.5 ábra). Ebből következően a két kamera középpontja között csak egy eltolás van, hiszen fősíkjaik is egy síkba esnek. A 3.2.2 fejezetben már láttuk, hogy ilyen esetben $\mathbf{F} = [\mathbf{e}']_{\times}$ és az epipólusok a végtelen távoli síkon vannak (vagyis $\mathbf{e}' = (1, 0, 0,)^T$), az epipoláris egyenesek pedig a képek rasztersorai lesznek. Ekkor tehát a 3.2 képlet az alábbiak szerint alakul:

$$\begin{aligned} 0 &= \mathbf{x}'^T \mathbf{F} \mathbf{x} = (x'_1, x'_2, 1) [\mathbf{e}']_{\times} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = (x'_1, x'_2, 1) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} \\ &= x_2 - x'_2 \implies x_2 = x'_2 \end{aligned} \quad (3.23)$$

vagyis a bal oldali kép egy \mathbf{x} pontjához tartozó epipoláris egyenes pontjai pontosan az \mathbf{x} pont rasztersorával megegyező rasztersoron lesznek a jobb oldali kamerában. Így tehát két egymásnak megfelelő pont esetén eltérés csak azok első koordinátaiban lehet.

Egyszerű összefüggést vezethetünk le egy $\mathbf{X} = (X_1, X_2, X_3)^T$ pont mélysége és \mathbf{x}, \mathbf{x}' bal és jobb oldali képeinek koordinátái között a 3.5 ábra alapján. Ehhez tekintsük az ábrán látható sárga és kék hasonló háromszögeket, melyekre az alábbi aránypárt írhatjuk fel:

$$\frac{B}{X_3} = \frac{B + x'_1 - x_1}{X_3 - f} \quad (3.24)$$



3.5. ábra. Standard sztereó konfiguráció: azonos kamerák, melyek képsíkjai közös síkra esnek, és középpontjaik között egy eltolás van.

amiből X_3 -ra az alábbi összefüggést kapjuk:

$$X_3 = f \frac{B}{x_1 - x'_1} \quad (3.25)$$

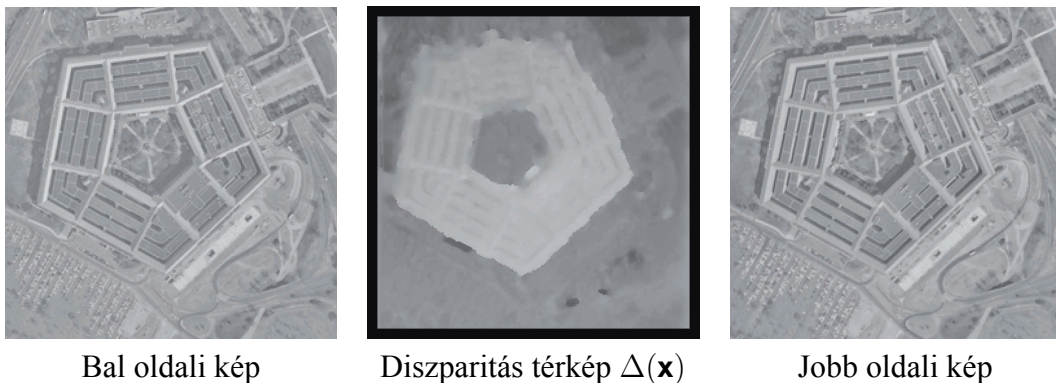
A nevezőben lévő $\delta = x_1 - x'_1$ különbség a két képpont x koordinátáinak különbsége, amit *diszparitás*nak nevezünk. A fenti képletet tehát az alábbi tömör formában is írhatjuk:

$$X_3 = f \frac{B}{\delta} \quad (3.26)$$

Amiből jól látható, hogy a diszparitás és a mélység egymással fordítottan arányos. A standard sztereó konfiguráció nagy előnye tehát, hogy egyrészt az epipoláris egyenesek azonosok a rasztorsorokkal és így a pontmegfeleltetések keresése nagyban leegyszerűsödik, másrészt ekkor csak x irányú diszparitás van, ami a mélységi rekonstrukciót könnyíti meg. Egy adott sztereó képpár közötti diszparitás értékek összességét *diszparitás térkép*nek nevezzük, amit szürkeárnyalatos képként is megjeleníthetünk (ld. 3.6 ábra).

3.4.1. Képek rektifikálása

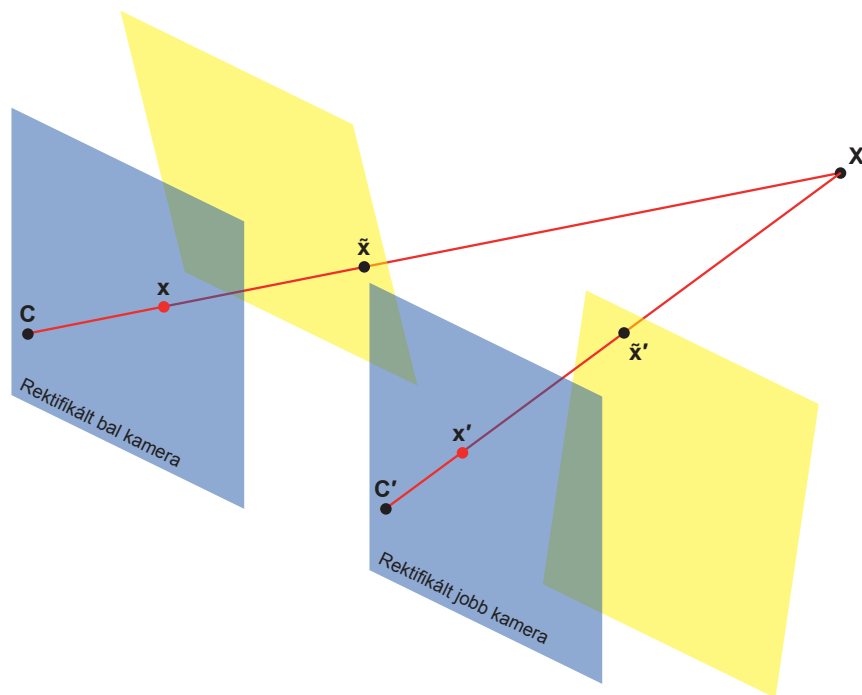
Nem standard konfigurációban készített sztereó képpárok esetén szintetikus generálhatjuk a standard konfigurációnak megfelelő képpárt (ld. 3.7 ábra), és így ekkor is kihasználhatjuk a standard konfiguráció előnyeit. Ezt az eljárást nevezzük *rektifikálás*nak. Az eljárás lényege egy síkhomográfia meghatározása az eredeti és a standard konfiguráció képsíkja között. Az



3.6. ábra. Diszparitás térkép: a világosabb értékek közelebbi pontokat jeleznek. A bal oldali kép egy \mathbf{x} pontjának megfelelő \mathbf{x}' pontot a jobb képen az $\mathbf{x}' = (x_1 - \Delta(\mathbf{x}), x_2, 1)^T$ összefüggéssel kapjuk meg a diszparitás térképből. (A pentagon rektifikált sztereó képpár a *Carnegie Mellon Vision and Autonomous Systems Center* sztereó képi adatbázisában érhető el.)

így kapott transzformációt alkalmazva megkapjuk azt a képet, amit akkor keletkezett volna, ha a kameráink már eredetileg standard konfigurációban lettek volna.

Keressük tehát azt a \mathbf{H} síkhomográfiát, amely az epipólust –egy skálafaktor erejéig– az $(1, 0, 0)^T$ pontba viszi (ekkor lesznek az epipoláris egyenesek párhuzamosak az x tengellyel). Erre több megoldás is van, azonban ezek közül azt a transzformációt kell választanunk, amelyik minimális képi torzítást okoz. Ezt úgy formalizálhatjuk, hogy a kép közepén (vagyis az \mathbf{o}



3.7. ábra. Standard sztereó konfiguráció előállítása általános helyzetű sztereó képpárból rektifikálással.

illetve \mathbf{o}' főpontok körül) a transzformáció jó közelítéssel csak merevtest deformációt enged meg. Feltéve tehát, hogy \mathbf{o} az origó és $\mathbf{e} = (f, 0, 1)^T$ az x tengelyen található, az alábbi \mathbf{G} transzformáció az epipólust a kívánt $(f, 0, 0)^T$ pontba képezi le:

$$\mathbf{G}\mathbf{e} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix} \begin{pmatrix} f \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} f \\ 0 \\ 0 \end{pmatrix} \quad (3.27)$$

Egy tetszőleges $\mathbf{x} = (x_1, x_2, 1)^T$ pontra pedig

$$\mathbf{G}\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ 1 - x_1/f \end{pmatrix} = \begin{pmatrix} x_1(1 + x_1/f + \dots) \\ x_2(1 + x_1/f + \dots) \\ 1 \end{pmatrix}, \quad (3.28)$$

ahol a homogén osztást Taylor sorral közelítjük. A végtelen összegből az első két tagot meghagyva a transzformáció Jacobi mátrixának első rendű közelítéseként kapjuk

$$\frac{\partial \mathbf{G}}{\partial (x_1, x_2)}(x_1, x_2) = \begin{bmatrix} 1 + 2x_1/f & 0 \\ x_2/f & 1 + x_1/f \end{bmatrix} \quad (3.29)$$

ami az \mathbf{o} origóban pontosan az egységmátrixot adja, tehát itt a \mathbf{G} az identikus transzformációt közelíti. Ha most egy általános helyzetű kép koordináta rendszert tételezünk fel, akkor $\mathbf{H} = \mathbf{GRT}$, ahol \mathbf{T} az az eltolás, ami \mathbf{o} -t az origóba viszi, \mathbf{R} pedig az az origó körüli forgatás, ami \mathbf{e} -t az $(f, 0, 1)^T$ pontba viszi az x tengelyen, ahonnan az előbb tárgyalt módon \mathbf{G} a megfelelő végtelen távoli pontba viszi \mathbf{e} -t. A másik képre hasonlóan kaphatunk egy \mathbf{H}' homográfiát. Ezek után természetesen még az is szükséges, hogy a bal és jobb oldali képeken az epipoláris egyenesek egymásnak megfelelő rasztersorok legyenek, vagyis tetszőleges, egymásnak megfelelő \mathbf{l}, \mathbf{l}' epipoláris egyenesekre teljesülnie kell a $\mathbf{H}^{-T}\mathbf{l} = \mathbf{H}'^{-T}\mathbf{l}'$ összefüggésnek is. Tehát ha a fenti módon meghatároztunk egy \mathbf{H} transzformációt, akkor ahhoz a $\sum_i \|\mathbf{H}\mathbf{x}_i, \mathbf{H}'\mathbf{x}'_i\|^2$ négyzetes hiba minimalizálásával kereshetjük meg a másik képhez tartozó megfelelő \mathbf{H}' transzformációt. Vegyük észre, hogy ez a minimalizálás egyúttal a képeket x irányban is úgy illeszti, hogy a pontok között minimális legyen a diszparitás.

4. fejezet

3D rekonstrukció

Az eddig megismert geometriai összefüggések alapján most megvizsgáljuk, hogy egy sztereó képpárból hogyan és milyen mértékben tudjuk rekonstruálni az eredeti 3D látványt. Ehhez lényegében három fontos kérdést kell megválaszolnunk:

1. Melyek a bal és jobb kép egymásnak megfelelő $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ pontpárjai?
2. A pontmegfeleltetések ismeretében hogyan határozzuk meg a \mathbf{P} , \mathbf{P}' kamera mátrixokat?
3. A kamera mátrixok ismeretében mely \mathbf{X}_i 3D pontok lesznek az $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ pontpárok rekonstrukciói?

A továbbiakban ezekre a kérdésekre adjuk meg a választ.

4.1. Sztereó megfeleltetések

Pontmegfeleltetések meghatározása a képfeldolgozás egyik legrégebbi alapproblémája. A megoldások sokfélék, és jelen könyvünk terjedelmét meghaladja a kérdéskör átfogó tárgyalása. Itt kifejezetten a sztereó képpárok közötti megfeleltetések meghatározására fogunk koncentrálni, és azon belül is az ún. *sűrű illesztés* problémájára. Sűrű illesztésen olyan pontpárosítást értünk, amely a kép valamennyi pixelére megadja a neki megfelelő pixelt a másik képen. Ezzel szemben a *ritka illesztés* csak a pixelek egy adott képjellemző alapján előre kiválasztott részhalmazára határozza meg a párosításokat. Ennek tipikus példája, amikor mindkét képből kinyerjük a sarokpontokat, majd az így kapott ponthalmazok pontjait feleltetjük meg egymásnak. A sztereó rekonstrukcióhoz régebben – főleg a korlátozott számítási kapacitásból fakadóan – inkább ritka illesztést használtak, de manapság a sűrű illesztési módszerek váltak meghatározóvá. A számítógépes látás más területein, mint pl. mozgáskövetéshez, azonban manapság is inkább ritka illesztést alkalmaznak.

A 3.4 fejezetben ismertetett standard sztereó konfigurációt feltételezve az egymásnak megfelelő pixelek a két kép azonos sorában találhatóak. Ez az epipoláris geometriai összefüggés tehát egy \mathbf{x} pont \mathbf{x}' párjának meghatározását egy általános 2D keresésről 1D keresésre szűkíti, ezzel jelentősen csökkentve a párosítás számításigényét. A továbbiakban tehát azt vizsgáljuk, hogy az egymásnak megfelelő pixelsorokon belül hogyan tudjuk párosítani a pixeleket. Ehhez először is egy hasonlósági mértéket kell definiálnunk, amellyel az egyes pixelek

összevethetőek. Ennek legegyszerűbb módszere a fotometriai alapú összehasonlítás, vagyis a potenciális \mathbf{x} , \mathbf{x}' pontpárok intenzitását vagy színét kell összehasonlítani. Az egyszerűség kedvéért tekintsünk szürkeárnyalatos képeket, ahol a pixeleket az intenzitásukkal jellemezhetjük. A bal és jobb oldali képfüggvényt rendre f -el és g -vel jelölve, azt tételezzük fel, hogy ugyanazon \mathbf{X} 3D pont képe mindkét kamerában ugyanolyan intenzitásértéket generál, vagyis

$$\mathbf{x} \leftrightarrow \mathbf{x}' \implies f(\mathbf{x}) = g(\mathbf{x}') \quad (4.1)$$

Ezt az összefüggést azonban a valóságban sok egyéb körülmény befolyásolhatja, ezért csak közelítően teljesül (pl. képi zaj, felületek csillogása, stb...). Erre a problémára a klasszikus megoldás, hogy az összehasonlítást nem pixel szinten végezzük, hanem a pixel kis környezetében, egy megfelelő nagyságú *ablak*ban vesszük az átlagintenzitásokat, és arra vizsgáljuk a fenti egyenlőséget. Az intenzitásértékek közvetlen összehasonlítását azonban instabillá teheti a két kamera eltérő expozíciója, vagy a leképezett 3D felületek által visszavert fény intenzitásának változása a kamera nézőpontjának függvényében. Ezeket kiküszöbölhetjük az ablakon belüli pixelértékek normalizálásával. Tekintsünk tehát egy $n \times n$ méretű W illetve W' ablakot a vizsgálandó \mathbf{x} és \mathbf{x}' pixelek körül. A két ablakban a pixelértékek normalizálást elvégezhetjük az alábbi képlet segítségével:

$$\forall \mathbf{x} \in W : \tilde{f}(\mathbf{x}) = \frac{f(\mathbf{x}) - \mu_W}{\sigma_W} \quad (4.2)$$

ahol μ_W és σ_W az ablak átlagintenzitása illetve az ablak magnitúdója:

$$\mu_W = \frac{1}{n^2} \sum_{\mathbf{x} \in W} f(\mathbf{x}) \quad (4.3)$$

$$\sigma_W = \sqrt{\sum_{\mathbf{x} \in W} (f(\mathbf{x}) - \mu_W)^2} \quad (4.4)$$

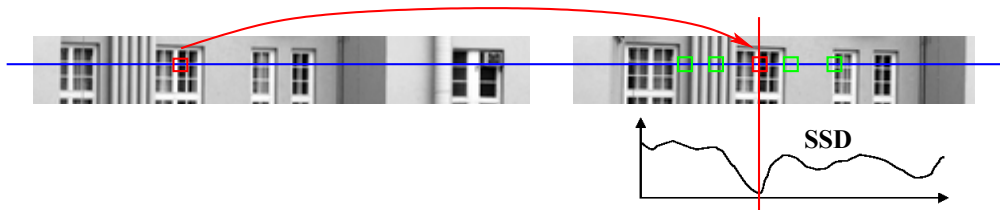
Hasonló összefüggést vezethetünk le $\tilde{g}(\mathbf{x}')$ -re is. Az így kapott normalizált ablakok tartalmából azután n^2 dimenziós vektorokat képezhetünk úgy, hogy az ablak sorait egymás alá másoljuk. Jelöljük ezeket a vektorokat rendre \mathbf{W} illetve \mathbf{W}' -vel. A normalizálás miatt a vektorok egységnyi hosszúak lesznek. Az \mathbf{x} , \mathbf{x}' pontok hasonlóságát ezek után mérhetjük

SSD: a két vektor különbségével, ami megfelel a pixelértékek négyzetes eltéréseinek összegével (*Sum of Squared Differences – SSD*):

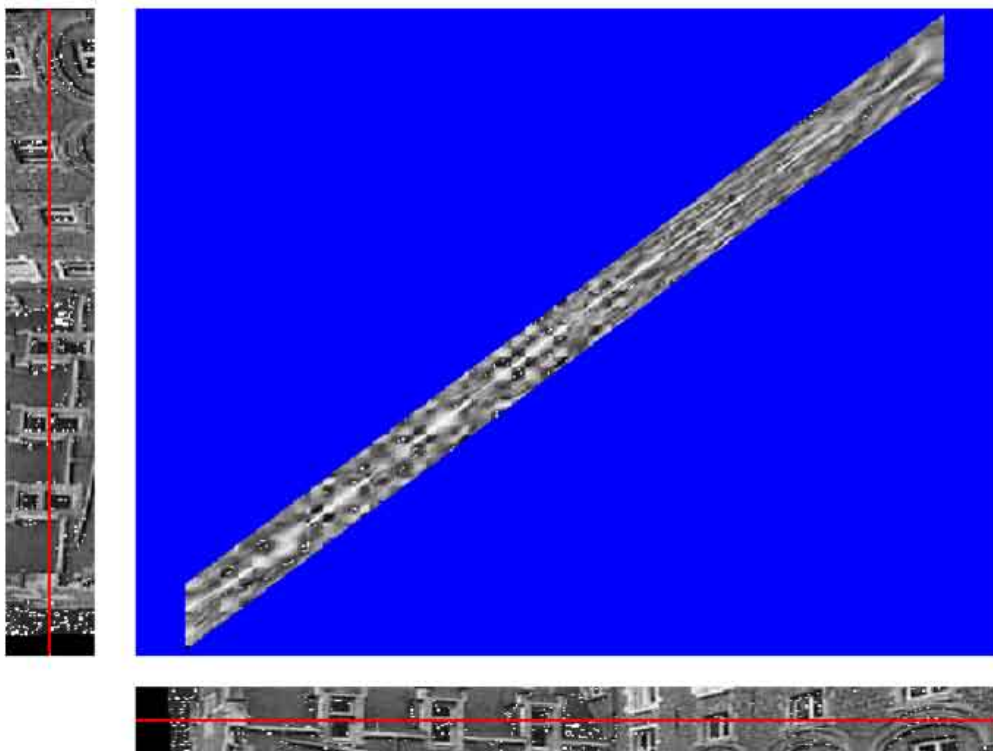
$$SSD = \|\mathbf{W} - \mathbf{W}'\|^2 = \sum_{i=1}^{n^2} (\tilde{f}(\mathbf{x}_i) - \tilde{g}(\mathbf{x}'_i))^2 \quad (4.5)$$

NCC: a két vektor által bezárt θ szög nagyságával (*Normalized Cross-Correlation – NCC*):

$$NCC = \cos(\theta) = \mathbf{W} \cdot \mathbf{W}' = \sum_{i=1}^{n^2} \tilde{f}(\mathbf{x}_i) \tilde{g}(\mathbf{x}'_i) \quad (4.6)$$



4.1. ábra. Sztereó pontmegfeleltetés: a rasztersoron csúsztatott ablakban kiszámolt SSD értékek minimumhelyén lesz a keresett pont.



4.2. ábra. A bal és jobb oldali kép azonos rasztersorához tartozó megfeleltetési költségek egy mátrixot alkotnak. A késsel jelölt területek kívül esnek a $0 \dots \delta_{max}$ diszparitási korláton.

A eddigiek alapján tehát a bal oldali kép egy rögzített $\mathbf{x} = (x_1, x_2, 1)^T$ pontjához a jobb oldali kép azonos rasztersorában lévő $\mathbf{x}'(\delta) = (x_1 - \delta, x_2, 1)^T$ pixelek közül kell kiválasztanunk azt, amely a fenti fotometriai mértékek valamelyike szerint a leghasonlóbb. A fenti két hasonlósági mérték szerint kapunk tehát két függvényt:

$$D^{SSD}(\mathbf{x}, \delta) = \|\mathbf{W} - \mathbf{W}'(\delta)\|^2 \quad (4.7)$$

$$D^{NCC}(\mathbf{x}, \delta) = \mathbf{W} \cdot \mathbf{W}'(\delta) \quad (4.8)$$

ahol $\mathbf{W}'(\delta)$ a jobb oldali kép $\mathbf{x}'(\delta) = (x_1 - \delta, x_2, 1)^T$ pixeléhez tartozó ablakból képzett vektor. Lényegében $\mathbf{W}'(\delta)$ egy csúszóablak az \mathbf{x} -hez tartozó epipoláris egyenesen (rasztersoron), amellyel minden pixelre előállítjuk a fotometriai hasonlóság mértékét. Mivel ezek diszkrét (pixeleken értelmezett) függvények, ezért *diszparitás képek* is nevezzük őket. Vegyük észre,

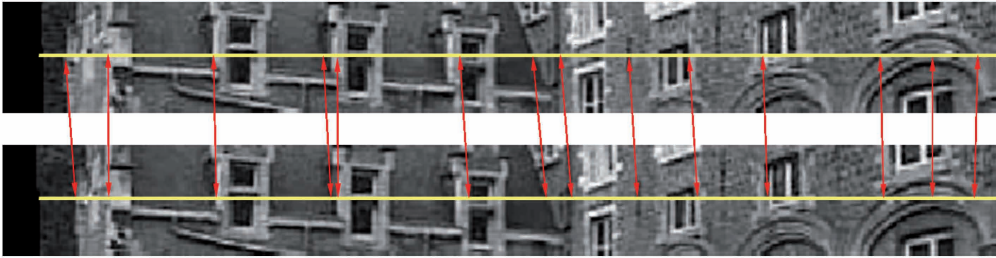
hogy egy $D(\mathbf{x}, \delta)$ diszparitás kép tulajdonképpen egy 3D képtömb lesz, amelynek xy kiterjedése megegyezik a sztereó képek méretével, z irányban pedig minden egyes \mathbf{x} pixelnek megfelelő pozícióban megtaláljuk az $\mathbf{x}'(\delta)$ pontokkal vett fotometriai hasonlóság értékeket. Vagyis a diszparitás képtömb egy vízszintes szelete (ami egy 2D mátrix/kép) megadja a bal és jobb oldali képek egy megegyező rasztersorában lévő pixelek összes lehetséges párosításának hasonlósági értékét (ld. 4.2 ábra). Ha rögzítjük \mathbf{x} -et, akkor $D(\mathbf{x}, \delta)$ egy egyváltozós függvénynek tekinthető (lényegében az előbbi 2D mátrix egyik sora lesz), melyet a δ diszparitás szerint optimalizálva a $D^{SSD}(\mathbf{x}, \delta)$ minimum helye, illetve a $D^{NCC}(\mathbf{x}, \delta)$ maximum helye adja az illeszkedő pixel $\hat{\delta}$ diszparitását:

$$\begin{aligned}\hat{\delta} &= \arg \min_{\delta} \|\mathbf{W} - \mathbf{W}'(\delta)\|^2 \\ &= \arg \max_{\delta} \mathbf{W} \cdot \mathbf{W}'(\delta)\end{aligned}\quad (4.9)$$

Tehát $\mathbf{x} \leftrightarrow \hat{\mathbf{x}}'$, ahol $\hat{\mathbf{x}}' = (x_1 - \hat{\delta}, x_2, 1)^T$ (ld. 4.1 ábra).

Ha tudjuk, hogy a rekonstruálandó objektum mélységi kiterjedése korlátozott, akkor ebből a 3.26 képlet alapján a diszparitás δ_{max} maximális megengedett értékére következtethetünk, aminek eredményeképpen a fenti optimalizálást megszoríthatjuk az epipoláris egyenesen a $\delta = 0 \dots \delta_{max}$ -nak megfelelő szakaszra. δ_{max} meghatározható a rekonstruálandó 3D objektumról meglevő *a priori* ismeretek alapján, vagy ennek hiányában megfelelő jellemző pontok rekonstrukciójából is.

Egy sztereó képpár esetén minden pixelnek maximum egy párja lehet. Abban az esetben nem lesz párja egy pixelnek, ha a neki megfelelő \mathbf{X} 3D pont csak az egyik kamerában látszik. Konkrétan ha csak a bal oldali képen látszik, akkor a képpont *takarásban* van, ha pedig csak a jobb oldali képen látható, akkor a képpont *előtűnik*.



4.3. ábra. A bal és jobb oldali kép azonos rasztersorán az egymásnak megfelelő pixelek relatív sorrendje azonos.

További megszorítást jelent a párosítás során, ha kikötjük, hogy a bal és jobb oldali képen a pixelek relatív sorrendje megegyezik (ld. 4.3 ábra).

Egy rasztersor mentén mindezen feltételeknek eleget tevő pontmegfeleltetéseket polinom időben előállíthatjuk *dinamikus programozás* segítségével. Itt most a bal és jobb oldali kép azonos r rasztersorának párosítását fogjuk részletesebben megvizsgálni. Ahogyan már az előbb láttuk, ezen rasztersorok pixeleinek lehetséges párosításaihoz tartozó hasonlósági értékeket a $D(\mathbf{x}, \delta)$ diszparitás képtömb $y = r$ szelete tartalmazza, amely egy $\mathbf{D}_r(i, j)$ mátrix lesz. A dinamikus programozás megvalósításához egy $\mathbf{C}(i, j)$ költség mátrixot konstruálunk, ahol a mátrix minden eleme (avagy állapota) a $D_r(i, j)$ mátrix elemének és az előző állapot

költségének a kombinációja lesz. Az aggregált illeszkedési költséget rekurtzivan számolhatjuk az alábbi képletekkel:

$$C(i, j, M) = \min(C(i-1, j-1, M), C(i-1, j, L), C(i-1, j-1, R)) + \mathbf{D}_r(i, j) \quad (4.10)$$

$$C(i, j, L) = \min(C(i-1, j-1, M), C(i-1, j, L)) + \tau \quad (4.11)$$

$$C(i, j, R) = \min(C(i, j-1, M), C(i, j-1, R)) + \tau \quad (4.12)$$

ahol τ a takarás/előtűnés konstans költsége. A megfeleltetéseket a két rasztorsor pontjai között a $\mathbf{C}(i, j)$ mátrixban kialakult minimális költségű útvonal adja meg, amely megfelelő pontpárok esetén M állapotnak felel meg, míg takarás esetén vízszintesen (L állapot) illetve előtűnés esetén függőlegesen (R állapot) halad. A $\delta = 0 \dots \delta_{max}$ diszparitási korlátot a költségmátrix konstrukciójánál vesszük figyelembe: csak az adott intervallumba eső diszparitásokat engedjük meg a költségek számításánál. Természetesen τ helyes megválasztása is kritikus, hiszen hibás értéke kedvezőtlenül befolyásolhatja a megfeleltetések pontosságát. A módszer további hátránya, hogy nem biztosítja az egyes rasztorsorok közötti konzisztenciát (ami lényegében a diszparitás térkép simaságát jelenti). Ezt azonban utólagos feldolgozással meg lehet oldani. A módszer nagy előnye azonban, hogy nagyon gyors és jó eredményeket szolgáltat, ha sikerült a paramétereket megfelelően beállítani.

4.2. Kanonikus kamerák

Kamera kalibrációval már többször is foglalkoztunk az eddigiekben: az 1.3 fejezetben tárgyaltuk a kamerák kalibrációs minta alapján történő kalibrációját. Azután a 2.3 fejezetben megvizsgáltuk a projektív kamera képe és kalibrációs mátrixa közötti összefüggést, melyet az ideális kúpszelet képe határoz meg.

Egy sztereó képpár esetében azonban rendelkezésünkre áll (vagy könnyen kiszámítható) az \mathbf{F} fundamentális mátrix, melynek segítségével előállíthatunk egy *kanonikus kamera* párt. Ahogyan a 3.2 fejezetben már láttuk, a fundamentális mátrix immunis a projektív homográfia, vagyis ha a $(\mathbf{P}, \mathbf{P}')$ kamerák között az \mathbf{F} mátrix hat, akkor ugyanez lesz a fundamentális mátrixa a $(\mathbf{PH}, \mathbf{P}'\mathbf{H})$ kameráknak, ahol \mathbf{H} egy maximális rangú projektív transzformáció. Vagyis több kamera párnak is lehet ugyanaz a fundamentális mátrixa. Formálisan is belátható (ld. [4]), hogy ha két kamerapárnak ugyanaz a fundamentális mátrixa, akkor közöttük egy projektív transzformáció hat, vagyis a fundamentális mátrix egy *projektív transzformáció erejéig* határozza meg a $(\mathbf{P}, \mathbf{P}')$ kamera mátrixokat.

Ezen ekvivalens kamera párokra tehát célszerű egy rögzített, un. *kanonikus kamera* párt meghatározni. Ehhez válasszuk a $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ bal oldali kamerát. A kérdés az, hogy a fundamentális mátrix ismeretében ehhez hogyan határozhatjuk meg a \mathbf{P}' jobb oldali kamerát. Ehhez tekintsük a $\mathbf{P}' = [\mathbf{SF}|\mathbf{e}']$ kamerát, ahol \mathbf{e}' az epipólus, \mathbf{S} pedig egy nyírás-szimmetrikus mátrix. Az így előállt $(\mathbf{P}, \mathbf{P}')$ kamerák fundamentális mátrixa \mathbf{F} lesz. Ehhez elegendő belátni, hogy \mathbf{F} akkor és csakis akkor lesz a $(\mathbf{P}, \mathbf{P}')$ fundamentális mátrixa, ha $\mathbf{P}'^T \mathbf{F} \mathbf{P}$ nyírás-szimmetrikus. Ez utóbbi viszont ekvivalens azzal, hogy $\forall \mathbf{x} : \mathbf{x}^T \mathbf{P}'^T \mathbf{F} \mathbf{P} \mathbf{x} = 0$ és mivel $\mathbf{x} = \mathbf{P} \mathbf{X}$ illetve $\mathbf{x}' = \mathbf{P}' \mathbf{X}$, ezért ez tovább ekvivalens az $\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$, amiből következik, hogy \mathbf{F} a fundamentális mátrix. Ez alapján a fenti $(\mathbf{P}, \mathbf{P}')$ kamera párra kis számolással belátható, hogy fundamentális mátrixa \mathbf{F} . Ahhoz, hogy \mathbf{P}' valóban érvényes kamera mátrix legyen, az is szükséges, hogy rangja 3

legyen. Belátható, hogy ez az $\mathbf{S}=[\mathbf{e}']_{\times}$ választás esetén teljesül. Ekkor ugyanis $\mathbf{P}'=[[\mathbf{e}']_{\times}\mathbf{F}|\mathbf{e}']$, és mivel $\mathbf{e}'^T\mathbf{e}'\neq 0$, ezért \mathbf{P}' rangja 3 lesz (ld. [4]).

Az \mathbf{F} fundamentális mátrix ismeretében tehát meghatározhatunk egy $(\mathbf{P}, \mathbf{P}')$ kanonikus kamera párt, amely egy projektív transzformáció erejéig ekvivalens az eredeti kamera párossal:

$$\mathbf{P} = [\mathbf{I}|\mathbf{0}] \quad \mathbf{P}' = [[\mathbf{e}']_{\times}\mathbf{F}|\mathbf{e}'] \quad (4.13)$$

4.3. Sztereó rekonstrukció

Sztereó rekonstrukció alatt értjük egy sztereó képpár $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ pontmegfeleltetése alapján a pontpárokhoz tartozó \mathbf{X}_i 3D pontok valamint a képeket előállító $(\mathbf{P}, \mathbf{P}')$ kamera pár meghatározását, amelyek kielégítik az alábbi feltételeket:

$$\mathbf{x}_i = \mathbf{P}\mathbf{X}_i \quad \text{és} \quad \mathbf{x}'_i = \mathbf{P}'\mathbf{X}_i \quad (4.14)$$

Természetesen a rekonstrukció eredménye nem feltétlenül lesz az eredeti látvánnyal megegyező, ún. *metrikus rekonstrukció*, hanem azzal csak valamilyen tarnszformáció erejéig ekvivalens változata. Így beszélhetünk *affin rekonstrukcióról*, ha a rekonstruált és az eredeti látvány között egy affin deformáció van. A legáltalánosabb esetben, amikor az $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ pontpárokon kívül semmilyen más információ nem áll rendelkezésünkre, *projektív rekonstrukciót* kapunk. Hogy ezek közül melyik rekonstrukciót tudjuk előállítani, az attól függ, hogy a kamera mátrixok mennyire ismertek. Kalibrált kamerákból metrikus rekonstrukciót tudunk előállítani, részlegesen kalibrált kamerákból affint, míg kalibrálatlan kamerákból csak projektív rekonstrukciót.

Fontos megjegyezni azonban, hogy a metrikus rekonstrukció is csak egy *hasonlósági transzformáció* erejéig képes helyreállítani a 3D látványt. Nyilvánvalóan a rekonstruált látvány *eredeti* világ koordináta rendszerbeli 3D pozícióját és orientációját nem tudjuk csupán a kamerák és a képek ismeretében rekonstruálni: például a 3.1 ábra sztereó képpárját rekonstruálva még nem tudjuk megmondani, hogy a szegedi Dóm pontosan hol található Szegeden (vagyis az eredeti pozícióját), sem pedig annak tájolását (vagyis az eredeti orientációt), hiszen ehhez tudnunk kellene, hogy a rekonstruált látványon merre van észak illetve legalább egy pont geográfiai koordinátáit – ez azonban a kamerák képéből és a kamera mátrixokból nem határozható meg (de például a képhez társított GPS koordinátákból már igen). Ugyanígy azt sem tudjuk leolvasni egy ilyen rekonstruált látványról, hogy milyen magas a szegedi Dóm tornya – tehát az eredeti koordináta rendszer léptéke is ismeretlen. A rekonstrukció alapján ugyanis nem lehet megállapítani, hogy a képek a Dóm élethű makettjéről, vagy ténylegesen az épületről készültek. Ezt az effektust használták ki például számos filmtrükk megvalósításánál is.

4.3.1. Projektív rekonstrukció

Ha elegendő $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ pontmegfeleltetés áll rendelkezésünkre, amelyből a fundamentális mátrix egyértelműen kiszámítható, akkor minden egyéb információ nélkül a \mathbf{X}_i 3D pontokat egy projektív defomráció erejéig rekonstruálni tudjuk. Természetesen a bázis egyenesre eső \mathbf{X}_i pontokat nem lehet rekonstruálni, hiszen ekkor a képpontok az epipólusok lesznek és a 4.14 képlet

teljesül a bázis egyenes valamennyi \mathbf{X}_i pontjára, vagyis az $(\mathbf{e}, \mathbf{e}', \mathbf{X}_i)$ hármas nem egyértelműen meghatározott. A számítógépes látás egyik legfontosabb alaptétele az ún. *projektív rekonstrukciós tétel*, amely kimondja, hogy ha egy sztereó képpáron adott pontmegfeleltetés halmaz egyértelműen meghatározza a fundamentális mátrixot, akkor a 3D látványt és a kamerákat rekonstruálhatjuk kizárólag ezen megfeleltetésekből, és bármely két ilyen rekonstrukció projektíven ekvivalens lesz.

4.3.1. Tétel (Projektív rekonstrukciós tétel). *Adottak $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ pontpárok a sztereó képek között, melyek egyértelműen meghatározzák az \mathbf{F} fundamentális mátrixot a $\forall i : \mathbf{x}'_i{}^T \mathbf{F} \mathbf{x}_i = 0$ összefüggések által. Továbbá legyen $(\mathbf{P}^1, \mathbf{P}'^1, \mathbf{X}_i^1)$ és $(\mathbf{P}^2, \mathbf{P}'^2, \mathbf{X}_i^2)$ két rekonstrukció. Ekkor létezik egy \mathbf{H} nem szinguláris transzformáció, amelyre:*

$$\mathbf{P}^2 = \mathbf{P}^1 \mathbf{H}^{-1}, \quad \mathbf{P}'^2 = \mathbf{P}'^1 \mathbf{H}^{-1}, \quad \text{és } \forall i : \mathbf{X}_i^2 = \mathbf{H} \mathbf{X}_i^1 \quad (4.15)$$

kivéve azon i -kre, amelyekre $\mathbf{F} \mathbf{x}_i = \mathbf{x}'_i{}^T \mathbf{F} = \mathbf{0}$.

Mivel a fundamentális mátrix egyértelműen meghatározott, ezért mindkét kamera párnak ugyanaz lesz a fundamentális mátrixa. Ebből viszont következik, hogy a két kamera páros között egy \mathbf{H} projektív transzformáció hat (ld. 4.2 fejezet és [4]), vagyis $\mathbf{P}^2 = \mathbf{P}^1 \mathbf{H}^{-1}$ illetve $\mathbf{P}'^2 = \mathbf{P}'^1 \mathbf{H}^{-1}$. Ami a pontokat illeti, vegyük észre, hogy

$$\mathbf{P}^2(\mathbf{H} \mathbf{X}_i^1) = \mathbf{P}^1 \mathbf{H}^{-1} \mathbf{H} \mathbf{X}_i^1 = \mathbf{P}^1 \mathbf{X}_i^1 = \mathbf{x}_i \quad (4.16)$$

másrészt $\mathbf{P}^2 \mathbf{X}_i^2 = \mathbf{x}_i$, vagyis $\mathbf{P}^2(\mathbf{H} \mathbf{X}_i^1) = \mathbf{P}^2 \mathbf{X}_i^2$. Mivel mindkét pont ugyanarra a képpontra képezi le a \mathbf{P}^2 kamera, ezért $\mathbf{H} \mathbf{X}_i^1$ és \mathbf{X}_i^2 ugyanazon a vetítősugáron található. A \mathbf{P}^2 kamerára hasonló összefüggés vezethető le, így csak két eset lehetséges: vagy $\mathbf{X}_i^2 = \mathbf{H} \mathbf{X}_i^1$, vagy pedig mindkét pont a bázisegyenesen található, amikor is a képpontok az epipólusok lesznek és $\mathbf{F} \mathbf{x}_i = \mathbf{x}'_i{}^T \mathbf{F} = \mathbf{0}$.

A tétel jelentősége tehát abban áll, hogy csupán pontmegfeleltetésekből előállíthatunk egy projektív rekonstrukciót az alábbi lépésekkel:

1. Az $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ pontpárok alapján meghatározzuk az \mathbf{F} fundamentális mátrixot (ld. 3.2.1 fejezet)
2. A fundamentális mátrix segítségével felírhatjuk a kanonikus kamerákat $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$, $\mathbf{P}' = [[\mathbf{e}']_{\times} \mathbf{F} | \mathbf{e}']$ alakban (ld. 4.2 fejezet)
3. A kamerák ismeretében kiszámíthatjuk az $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ pontokhoz tartozó \mathbf{X}_i 3D pontokat (ld. 4.3.2 fejezet). Az így kapott rekonstrukció az eredeti látvány projektív deformáltja lesz.

4.3.2. 3D pontok meghatározása

Ha adottak a \mathbf{P} , \mathbf{P}' sztereó kamera mátrixok, akkor az $\mathbf{x} \leftrightarrow \mathbf{x}'$ pontpárból egyszerű *háromszögeléssel* előállítható a nekik megfelelő \mathbf{X} 3D pont: Mivel a kamera mátrixok ismertek, ezért az \mathbf{x} és \mathbf{x}' képpontokat visszavetítve a vetítősugarak metszéspontja adja a \mathbf{X} pontot (ld. 3.2 ábra). Ha a kamera mátrixokat hibátlannak tételezzük is fel, sajnos a gyakorlatban nem lehetséges a

képpont koordinátákat teljes pontossággal megállapítani, így a visszavetített sugarak általában nem fogják egymást metszeni. Ez azt jelenti, hogy nem létezik olyan \mathbf{X} pont, amelyre teljesülnek az $\mathbf{x} = \mathbf{P}\mathbf{X}$ és $\mathbf{x}' = \mathbf{P}'\mathbf{X}$ összefüggések. Ekkor természetesen az $\mathbf{x}'^T \mathbf{F}\mathbf{x} = 0$ epipoláris feltétel sem teljesül. Ezért közvetlenül nem oldhatjuk meg a háromszögelést, hanem becsléssel kell előállítanunk a legjobb megoldást, amihez egy megfelelő költségfüggvényt kell konstruálni, majd azt minimalizálni.

Először a lineáris alapmegoldást tárgyaljuk, amely hasonlóan az 1.3.1 fejezetben tárgyalt kamera kalibrációs módszerhez, egy lineáris egyenletrendszer megoldásából áll. Adottak tehát a \mathbf{P} , \mathbf{P}' kamera mátrixok, és az $\mathbf{x} \leftrightarrow \mathbf{x}'$ képpontoknak megfelelő \mathbf{X} térbeli pontot szeretnénk rekonstruálni. Ezekre felírhatjuk a jól ismert $\mathbf{x} = \mathbf{P}\mathbf{X}$ és $\mathbf{x}' = \mathbf{P}'\mathbf{X}$ összefüggésekből képzett keresztszorzatokat (ezzel kiküszöbölve a homogén koordinátából adódó skálafaktort):

$$\mathbf{x} \times \mathbf{P}\mathbf{X} = \mathbf{0} \quad \text{és} \quad \mathbf{x}' \times \mathbf{P}'\mathbf{X} = \mathbf{0} \quad (4.17)$$

Mindkét képpontra kapunk tehát 3 egyenletet:

$$\begin{pmatrix} x_1(\boldsymbol{\pi}_3^T \mathbf{X}) - (\boldsymbol{\pi}_1^T \mathbf{X}) \\ x_2(\boldsymbol{\pi}_3^T \mathbf{X}) - (\boldsymbol{\pi}_2^T \mathbf{X}) \\ x_1(\boldsymbol{\pi}_2^T \mathbf{X}) - x_2(\boldsymbol{\pi}_1^T \mathbf{X}) \end{pmatrix} = \mathbf{0} \quad (4.18)$$

ahol $\mathbf{x} = (x_1, x_2, 1)^T$. A másik képre hasonló egyenleteket írhatunk fel. A fenti egyenletekből kettő lesz lineárisan független. Megtartva az első kettőt, a két képből összesen 4 egyenletet kapunk, amelyet felírhatunk $\mathbf{A}\mathbf{X} = \mathbf{0}$ alakban:

$$\begin{bmatrix} x_1 \boldsymbol{\pi}_3^T - \boldsymbol{\pi}_1^T \\ x_2 \boldsymbol{\pi}_3^T - \boldsymbol{\pi}_2^T \\ x_1' \boldsymbol{\pi}_3'^T - \boldsymbol{\pi}_1'^T \\ x_2' \boldsymbol{\pi}_3'^T - \boldsymbol{\pi}_2'^T \end{bmatrix} \mathbf{X} = \mathbf{0} \quad (4.19)$$

Az egyenletrendszert a szokásos módon megoldhatjuk az \mathbf{A} mátrix SVD felbontásával, amiből a legkisebb szinguláris értékhez tartozó egységnyi szinguláris vektor adja a megoldást, ami kielégíti a $\|\mathbf{X}\| = 1$ feltételt. A másik megoldási lehetőség, hogy \mathbf{X} utolsó koordinátáját rögzítjük: $X_4 = 1$, amivel az eddigi homogén egyenletrendszerből egy 4 egyenletből álló, 3 ismeretlenes inhomogén rendszert kapunk, amit legkisebb négyzetek módszerével oldhatunk meg. Látszólag nincs lényegi különbség a két módszer között. Azonban ha projektív rekonstrukciót végzünk, akkor az inhomogén rendszerben eleve kizárjuk a végtelen távoli pontokat a rekonstrukcióból, ami nem kívánatos.

Ennél nagyobb probléma azonban, hogy egyik megoldás sem lesz projektív invariáns. Ha a kameráinkat lecseréljük a $\mathbf{P}\mathbf{H}^{-1}$, $\mathbf{P}'\mathbf{H}^{-1}$ párra, akkor az \mathbf{A} mátrixból $\mathbf{A}\mathbf{H}^{-1}$ lesz. Ha az eredeti rendszer egy $\hat{\mathbf{X}}$ megoldása $\mathbf{A}\hat{\mathbf{X}} = \boldsymbol{\epsilon}$ algebrai hibát eredményez, akkor az új rendszerben a $\mathbf{H}\hat{\mathbf{X}}$ szintén $\mathbf{A}\mathbf{H}^{-1}(\mathbf{H}\hat{\mathbf{X}}) = \boldsymbol{\epsilon}$ hibát ad. Viszont a másik feltételt már $\mathbf{H}\hat{\mathbf{X}}$ egyik esetben sem fogja kielégíteni, hiszen sem a $\|\hat{\mathbf{X}}\| = 1$ sem pedig a $\hat{X}_4 = 1$ nem lesz invariáns a \mathbf{H} transzformációra, vagyis végeredményben az új rendszernek egy $\hat{\mathbf{X}}$ -től független $\hat{\mathbf{X}}'$ megoldása lesz.

Projektív invariáns megoldást kaphatunk, ha nem az algebrai, hanem a geometriai hibát minimalizáljuk. A geometriai hibát azonban nem mérhetjük a projektív térben, mivel a rekonstrukció során a rekonstruált pont projektív deformált változatát kapjuk, és mivel a projektív transzformáció nem távolság és szög tartó, ezért nem lehet konzisztens hibamértéket

konstruálni. Ezért a geometriai hibát a képeken mérjük. Ha tökéletes pontmegfeleltetésünk lenne, akkor $\mathbf{x} \leftrightarrow \mathbf{x}'$ -re teljesülne az $\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$ epipoláris geometriai összefüggés. Ez azonban a mi esetünkben csak közelítően teljesül. Ehelyett létezik tehát $\mathbf{x} \leftrightarrow \mathbf{x}'$ közelében egy $\hat{\mathbf{x}} \leftrightarrow \hat{\mathbf{x}}'$ pontpár amely pontosan kielégíti az $\hat{\mathbf{x}}'^T \mathbf{F} \hat{\mathbf{x}} = 0$ epipoláris geometriai feltételt. Ezek a pontok egy $\hat{\mathbf{X}}$ 3D pont $\hat{\mathbf{x}} = \mathbf{P} \hat{\mathbf{X}}$ és $\hat{\mathbf{x}}' = \mathbf{P}' \hat{\mathbf{X}}$ képeinek felelnek meg, és az alábbi hibafüggvény feltételes minimalizálásával találhatjuk meg őket:

$$\|\mathbf{x} - \mathbf{P} \hat{\mathbf{X}}\| + \|\mathbf{x}' - \mathbf{P}' \hat{\mathbf{X}}\| \text{ feltéve } \hat{\mathbf{x}}'^T \mathbf{F} \hat{\mathbf{x}} = 0 \quad (4.20)$$

Megjegyezzük, hogy a fenti normák kiszámolása előtt homogén osztással célszerű inhomogén koordinátákra áttérni. Magát a minimalizálást elvégezhetjük a klasszikus *Levenberg-Marquardt algoritmus*sal, de létezik direkt megoldás is [4]. A nemlineáris optimalizálás inicializálásához felhasználhatjuk az algebrai megoldást.

Mivel a háromszögelés adott kamerákból dolgozik, ezért a rekonstrukció meghatározottsága a kamera mátrixoktól függ. Ha csak a fundamentális mátrixból kiszámolt kanonikus kamerák állnak rendelkezésre, akkor a háromszögeléssel projektív rekonstrukciót kapunk. Ha azonban a kamerák kalibráltak voltak, akkor metrikus rekonstrukció lesz az eredmény.

4.3.3. Rétegelt rekonstrukció

A rétegelt rekonstrukció során először egy projektív rekonstrukciót állítunk elő, majd ezt lépésenként kiigazítjuk először affin rekonstrukcióra, majd végül metrikusra. Ez természetesen további információt igényel vagy a kameráról vagy magáról a rekonstruálandó látványról, hiszen magukból a pontmegfeleltetésekből csak projektív rekonstrukciót tudunk meghatározni.

Projektívből affin rekonstrukció

Az affin rekonstrukció lényege, hogy a $\boldsymbol{\pi}_\infty = (0, 0, 0, 1)^T$ végtelen távoli síkot a helyére teszi. A projektív rekonstrukciónkban azonban ez a sík egy általános $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \pi_4)^T$ síkként jelenik meg. A $\boldsymbol{\pi}$ sík ismeretében tehát csak azt a projektív transzformációt kell meghatározni, amely $\boldsymbol{\pi}$ -t a helyére teszi. Könnyen belátható, hogy ez a transzformáció az alábbi inverz transzponáltja lesz, vagyis $\boldsymbol{\pi}_\infty = \mathbf{H}^{-T} \boldsymbol{\pi}$, ahol:

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \boldsymbol{\pi}^T & \end{bmatrix} \quad (4.21)$$

Alkalmazva tehát \mathbf{H}^{-T} -t a projektív rekonstrukcióra és a kamerákra, egy affin rekonstrukciót kapunk, ahol pl. a párhuzamos egyenesek már párhuzamosak lesznek.

Magát a $\boldsymbol{\pi}$ síkot meghatározhatjuk például 3 párhuzamos egyeneshalmaz segítségével, melyek metszéspontja a végtelen távoli síkon 3 pontot határoz meg. Ez a három pont lényegében az egyenesek eltűnési pontjainak rekonstrukciói (ld. 2.2 ábra). Azonban nem szükséges mindkét képen meghatározni az eltűnési pontokat, hiszen az eltűnési pontok is kielégítik az epipoláris geometriai feltételt, vagyis egy \mathbf{v} eltűnési pont párja az $\mathbf{F} \mathbf{v}$ epipoláris egyenesen lesz. Másrészt ez a pont a megfelelő egyenesek képeinek mindegyikén rajta lesz, vagyis egy tetszőleges ilyen l' egyenes segítségével \mathbf{X} -et kifejezhetjük:

$$([\mathbf{v}]_{\times} \mathbf{P}) \mathbf{X} = \mathbf{0} \quad \text{és} \quad (l'^T \mathbf{P}) \mathbf{X} = 0 \quad (4.22)$$

Egy másik lehetőség a π_∞ végtelen távoli sík által generált \mathbf{H}_∞ homográfia meghatározása. Ez lényegében a már korábban tárgyalt π sík által generált \mathbf{H}_π homográfia (ld. 3.2 fejezet és 3.4 ábra) speciális esete. Legyenek a kameráink $\mathbf{P} = [\mathbf{M}|\mathbf{m}]$ illetve $\mathbf{P}' = [\mathbf{M}'|\mathbf{m}']$. Mivel a végtelen távoli sík pontjait $\mathbf{X} = (X_1, X_2, X_3, 0)^T = (\tilde{\mathbf{X}}^T, 0)^T$ alakban írhatjuk, ezért a pont képei $\mathbf{x} = \mathbf{M}\tilde{\mathbf{X}}$ és $\mathbf{x}' = \mathbf{M}'\tilde{\mathbf{X}}$ lesznek. Tehát $\mathbf{x}' = (\mathbf{M}'\mathbf{M}^{-1})\mathbf{x}$, vagyis $\mathbf{H}_\infty = \mathbf{M}'\mathbf{M}^{-1}$. Ez természetesen affin rekonstrukció esetén érvényes, amit ha a $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ és $\mathbf{P}' = [\mathbf{M}'|\mathbf{e}']$ kanonikus kamerákkal kaptunk, akkor szükségképpen $\mathbf{H}_\infty = \mathbf{M}'$ lesz. Megfordítva, ha ismert \mathbf{H}_∞ , akkor egy affin rekonstrukciót kaphatunk a $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ és $\mathbf{P}' = [\mathbf{H}_\infty|\mathbf{e}']$ kamerákkal. \mathbf{H}_∞ közvetlenül meghatározható a sztereó képpárról 3 eltűnési pontpár segítségével, ami lényegében ekvivalens a π_∞ meghatározásával.

Affinból metrikus rekonstrukció

A metrikus rekonstrukció lényege az Ω_∞ ideális kúpszelet azonosítása. Miután az affin rekonstrukcióhoz már a helyére tettük a végtelen távoli síkot, a következő lépés Ω_∞ azonosítása, majd ez alapján annak a transzformációnak a meghatározása, amely Ω_∞ -t a helyére teszi (ld. 2.2.2 fejezet). A gyakorlatban ezt legkönnyebben az ideális kúpszelet képének (ω) meghatározásával érhetjük el. Ez nem meglepő, hiszen ha ω ismert, akkor abból már a \mathbf{K} kalibrációs mátrixok közvetlenül megkaphatjuk, és kalibrált kamerákkal tudunk metrikus rekonstrukciót végrehajtani.

Ha adott az egyik képen ω , és a hozzá tartozó kamera mátrix $\mathbf{P} = [\mathbf{M}|\mathbf{m}]$, amellyel egy affin rekonstrukciót hajtottunk végre, akkor az affin rekonstrukciót metrikussá alakíthatjuk a

$$\mathbf{H} = \begin{bmatrix} \mathbf{A}^{-1} & \\ & 1 \end{bmatrix} \quad (4.23)$$

3D transzformációval, ahol az \mathbf{A} mátrixot Cholesky felbontással kaphatjuk meg az alábbiak szerint:

$$\mathbf{A}\mathbf{A}^T = (\mathbf{M}^T \omega \mathbf{M})^{-1} \quad (4.24)$$

Ugyanis a transzformációt alkalmazva a kamerára, az új kamera mátrix $\tilde{\mathbf{P}} = \mathbf{P}\mathbf{H}^{-1} = [\tilde{\mathbf{M}}|\tilde{\mathbf{m}}]$, ahol ha

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{A} & \\ & 1 \end{bmatrix} \quad (4.25)$$

akkor $\tilde{\mathbf{M}} = \mathbf{M}\mathbf{A}$. Másrészt $\tilde{\mathbf{M}} = \mathbf{K}\mathbf{R}$ alakban bontható fel és $\omega^{-1} = \mathbf{K}\mathbf{K}^T$ (ld. 2.2.2 fejezet), ezért $\omega^{-1} = \tilde{\mathbf{M}}\tilde{\mathbf{M}}^T$. Ezekből kapjuk az alábbi összefüggést:

$$\omega^{-1} = \tilde{\mathbf{M}}\tilde{\mathbf{M}}^T = \mathbf{M}\mathbf{A}\mathbf{A}^T\mathbf{M}^T \implies \mathbf{A}\mathbf{A}^T = (\mathbf{M}^T \omega \mathbf{M})^{-1} \quad (4.26)$$

Az ω -t meghatározhatjuk a 2.3 fejezetben ismertett módon.

Itt jegyezzük meg, hogy lehetséges a projektív rekonstrukciót közvetlenül metrikussá igazítani, ha rendelkezésünkre áll legalább 5 olyan $\tilde{\mathbf{X}}$ pont, melyeknek pontosan ismerjük a 3D koordinátáit és amelyek közül semelyik négyes nem esik egy síkra. Ezen pontok \mathbf{X}_i projektív rekonstrukcióját a hozzájuk tartozó megfelelő $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ pontpárokból kapjuk. A projektív rekonstrukcióból metrikusat kaphatunk a megfelelő \mathbf{H} transzformációval, melyet a pontjainkból konstruált lineáris egyenletrendszerből kapunk:

$$\forall i : \tilde{\mathbf{X}} = \mathbf{H}\mathbf{X}_i \quad (4.27)$$

Minden pont 3 független egyenletet szolgáltat, és mivel \mathbf{H} szabadsági foka 15, ezért legalább 5 pontra van szükségünk. A \mathbf{X}_i projektív rekonstrukció nem feltétlenül szükséges, felírhatjuk az egyenleteket a képpontokra is, hiszen

$$\forall i : \mathbf{x}_i = \mathbf{P}\mathbf{H}^{-1}\bar{\mathbf{X}} \quad \text{és} \quad \mathbf{x}'_i = \mathbf{P}'\mathbf{H}^{-1}\bar{\mathbf{X}} \quad (4.28)$$

amely 2-2 független egyenletet ad képpontonként, de $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ pontpáronként csak hármat, mivel a képpontok egy síkba esnek.

Második rész

Dinamikus látás

Bevezetés

Mindennapjainkat dinamikusan változó világban éljük. A tárgyak helyzete, alakja, a fényviszonyok, helyzetünk a minket körülvevő térben folyamatosan változik, ennek megfelelően a tárgyi világ optikai leképezése is. Tankönyvünk második része a változó világ leképezésének modellezésével, a mozgás érzékelésével, követésével és az ebből levont alapvető következtetésekkel foglalkozik. A mozgásinformáció optikai–számítógépes feldolgozásának számos gyakorlati alkalmazása van, mint pl:

- videó-felügyelet: egy nagyobb területen, akár több kamera használata esetén kell mozgó személyeket vagy járműveket detektálni, követni, mozgásukat elemezni;
- videókódolás és tömörítés: egy videó képkockái közti (ún. inter-frame) redundancia kiaknázása abból a célból, hogy a videók digitális tárolása kevesebb tárhelyet, csatorna kapacitást igényeljen;
- 3D-s struktúrák feltérképezése, mélységi információ mérése: pl. több, hagyományos 2D-s felvétel segítségével az objektum-pontok térbeli pozícióját meg tudjuk határozni;
- videóeffektusok, filmfelújítás: a képi hibák digitális javításához szükséges a képkockák közti megfeleltetés ismerete, ezáltal csökkenteni lehet a kép remegését, a véletlenszerűen felbukkanó foltokat el tudjuk tüntetni, vagy akár lehetséges a mozgó objektumok automatikus átszínezése;
- robot-navigáció, akadály-detekció, ütközés-előrejelzés: gépjárművek vagy robotok mozgása esetén az úton lévő akadályok térbeli pozícióját kell megbecsülni, egyes alkalmazásoknál jelezni kell, hogy előforulhat-e ütközés, ill. mikor várható annak a bekövetkezése [5], [11].

Ezen feladatok megoldását a következő eljárások segítik elő:

- a videót alkotó képkockák közti megfeleléség meghatározása: a kép egyes részleteinek egy másik (időben következő vagy időben megelőző) kép mely részletei felelnek meg;
- rekonstrukció: a kamera saját mozgását jellemző paramétereknek, a tér struktúrájának, a térben fellelhető objektumok pozíciójának kiszámítása a kamera felvételei alapján;
- mozgás szegmentálás: egy képkockán a hasonló mozgást végző, összetartozó területek foltszerű kijelölése;
- mozgás alapú videó elemzés: különböző események, speciális mozgásminták – mint pl. ember járása, kézjel gesztusok – felismerése.

5. fejezet

Mozgás leképezése, mozgásmodellek

A fizika definíciója szerint a mozgás egy test pozíciójának vagy irányának változása az időben. Míg kiterjedt szilárd testek mozgását haladó és forgó mozgással lehet jellemezni, pontszerű objektumok esetén elegendő haladó mozgásról beszélni. A haladó mozgást sebessége (ang. velocity) jellemezi, amely vektor jellegű mennyiség, nagysága (ang. speed) és iránya (ang. direction) határozza meg. Vizsgáljuk most meg, hogy ha a körülöttünk lévő világot egy kamerán keresztül szemléljük, mit tapasztalhatunk annak változásáról.

Egy videokamera felvételét szemlélve azt vehetjük észre, hogy annak képe nem állandó. Még ha távolról szemlélve a kép tartalma nem is változik, közelebről, esetleg a kontraszt növelése után észrevehetjük, hogy nagyon kicsi változások szinte mindig jellemzik a digitális képsorozatot. A pixelek értékeinek változását több minden is okozhatja, mint pl. a látómezőben történő mozgás, a kamera elmozdulása vagy remegése, a fényviszonyok változása, a képrögzítő eszközben fellépő véletlenszerű elektronikus zaj, tömörítési hiba. Sajnos az utóbbi néhány jelenség jelentősen megnehezíti a dolgunkat akkor, ha a látómezőben lévő objektumok mozgását szeretnénk meghatározni.

Alapvetően tehát megkülönböztethetünk valamilyen mozgásból vagy valamilyen egyéb változásból adódó pixel-érték változást (lásd 5.1. ábra). Ennek megfelelően beszélhetünk *változásdetekciós* és *mozgásdetekciós* eljárásokról. Mivel előbbi általában kisebb számítási komplexitással jár, ezért sok alkalmazásnál előfeldolgozási lépésként megelőzi az utóbbit.

Tankönyvünkben nem foglalkozunk változásdetekciós mechanizmusokkal – mint pl. az egyszerű képkockák közti különbségképzés vagy a méltán népszerű kevert Gauss háttérmodellet alkalmazó Stauffer és Grimson módszer [15] –, csupán a mozgásdetekció, modellezés és követés alapelveit tárgyaljuk olyan esetekben, amikor vagy valamilyen objektum mozdul el a kamera látómezőjében vagy maga a kamera végez mozgást (esetleg mindkettő fennáll). A kamera mozgását *sajátmozgásnak* (ang. egomotion) nevezzük.

5.1. A mozgásmező

A kamera látómezőjében lévő pontok tényleges mozgását jellemző, a kép 2D-s síkjára vetített mozgásvektorok által alkotott vektormezőt *mozgásmezőnek* nevezzük. A levetítés által meghatározott 2D-s mozgásvektorok az adott pontok elmozdulását írják le t_1 és t_2 időpontok között. A mozgásmező egy ideális vektormezőt jelöl, valójában általában ennek csak egy

5.1. ábra. Biztonsági videófelvétel és egymást követő képkockáinak különbségi képe hisztogramszéthúzás után. Jól látható, hogy nem csak a lényeges mozgásoknál jelentős a különbség, hanem a fa leveleinél is, és a tömörítési hiba is számottevő, erős blokkosodás figyelhető meg.

közelítő becslését tudjuk mérni (ill. szemünkkel érzékelni), mivel a 3D tér nem minden pontjának mozgását tudjuk meghatározni csupán a vetületi képek alapján. A gyakorlatban tehát az ún. *optikai áramlással* (lásd 6 fejezet) tudunk dolgozni, aminek a meghatározását különböző zavaró tényezők (pl. zajok, torzulások) nehezítik. A 5.2. ábrán egy olyan példát mutatunk be, ahol az objektum (Angliában és az USA-ban a borbélyok jelzésére funkcionáló harántcsíkos forgó oszlop) elmozdulása és annak detekciója közti különbség jól érzékelhető. A 5.2. ábrán pedig egy animáción láthatjuk ennek a vizuális hatását.

Ha minden pixelhez van rendelve mozgásvektor, akkor *sűrű mozgásmezőről*, ellenkező esetben – például amikor csak négyzet alakú blokkokhoz van rendelve mozgásvektor – *ritka mozgásmezőről* beszélünk.

5.2. Mozgás leképezése álló kamera esetén

Mozgást akkor érzékelhetünk a képen, ha a 3D tér – illetve annak bizonyos részei, objektumai – és a kamera egymáshoz képest elmozdul. Ha koordináta rendszerünket a kamera középpontjába helyezzük, ill. a „világi” pontokat $\mathbf{X} = (X, Y, Z)^T$ módon jelöljük, akkor egy pont sebességét a

$$\mathbf{U} = \frac{d\mathbf{X}}{dt} = \left(\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt} \right)^T \quad (5.1)$$

kifejezés írja le. A kamera \mathbf{X} világi pontot $\mathbf{x} = (x, y)^T$ képpontba képezi le perspektívikus projekcióval (lásd 5.3. ábra):

5.2. ábra. Mozgásmező és optikai áramlás különbsége. A forgó oszlop úgy tűnhet a szemlélő számára, mintha nem forogna, hanem felfele haladna (lásd apertúra probléma).

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z} \quad (5.2)$$

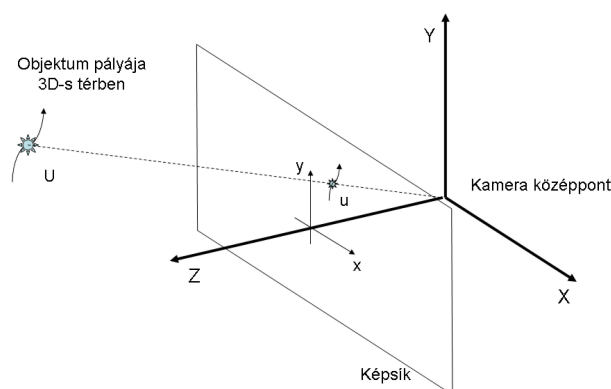
ahol f a fókusz távolságot jelenti. Mivel a pontunk a 3D térben mozgásban van, ennél fogva X függvénye t -nek, csakúgy, mint annak vetülete \mathbf{x} is. \mathbf{x} sebessége a képernyőn:

$$\mathbf{u} = \frac{d\mathbf{x}}{dt} = \left(\frac{dx}{dt}, \frac{dy}{dt} \right)^T. \quad (5.3)$$

Behelyettesítve 5.2-t 5.3-ba kapjuk a következő kifejezést:

$$\mathbf{u} = \frac{f}{Z} \left(\frac{dX}{dt}, \frac{dY}{dt} \right)^T - \frac{f}{Z^2} \frac{dZ}{dt} (X, Y)^T. \quad (5.4)$$

Ez az egyenlet adja meg a kapcsolatát a valós objektumok térbeli mozgásának és a kamera képére transzformált mozgásvektoroknak, azaz ez határozza meg az optikai mezőt, aminek közelítő meghatározása az optikai áramlás. Amennyiben azonban meg is tudjuk viszonylag jól határozni \mathbf{u} -t, még mindig igen nehéz következtetni a 3D-s térben történő valódi mozgásokra, hiszen túl sok ismeretlen van az egyenlet jobb oldalán. Az objektumra ill. az objektum mozgására vonatkozó bizonyos megszorítások, feltételek esetén azonban lehetséges információ kinyerése a valós térben való mozgásról is, így lehetséges 3D struktúrák meghatározása, vagy pl. ütközésselőrejelző eljárások kidolgozása is [5], [11]. Ezeknek a bemutatása azonban túlmutat dolgozatunk terjedelmén.



5.3. ábra. Mozgó pont a kameracentrikus koordináta rendszerben és projektív vetítéssel keletkezett képe

5.3. Mozgás leképezése mozgó kamera esetén

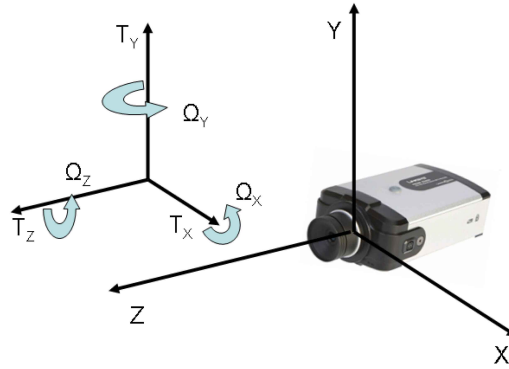
Ahhoz, hogy jól használható modelleket kapjunk a mozgásmezővel kapcsolatosan, érdemes bizonyos megszorításokat tennünk a mozgás jellegét, illetve magukat az objektumokat illetően. Először is feltételeznünk kell, hogy ún. *mervevtest mozgásról* van szó, azaz az objektumok térbeli alakja nem változik a felvétel során. Természetesen a gyakorlatban ez csak több-kevésbé teljesül, hiszen pl. a mozgást végző emberi test vagy a rajta lévő ruha esetében is szinte folyamatos az alakváltozás. Azonban a mozgás sebességéhez képest minél nagyobb az időbeli mintavételi frekvencia, ez a hatás annál kevésbé számottevő – két egymást követő képkocka közti alakváltozás egyre inkább elhanyagolható lesz (miközben az optikai áramlás meghatározása egyre pontosabbá válik).

Vizsgálódásunk szempontjából egy másik ésszerű feltételezés az, hogy a kamerát körülvevő teret *statikusnak* feltételezzük, míg a kamera mozgását igyekszünk minél kevesebb paraméterrel jellemezni. A legtöbb gyakorlati esetben ez a feltétel sem fog maradéktalanul teljesülni, hiszen általában a 3D teret nagyfokú dinamizmus jellemzi. Mégis előnyös ez a megközelítés, hiszen pont az ezt a szabályt megszegő lokális elmozdulások segítenek a mozgásinformáció elemzésében. Pl. egy robotba beépített ütközés–előrejelző optikai rendszerrel azokat a pontokat kell detektálni, amelyek nem felelnek meg valamilyen globális vagy lokális szabályszerűségnek, a képen látott képpontok elmozdulása nem felel meg valamilyen globális mozgásmodellnek (pl. lásd a 5.15. egyenletet, vagy [1], [8] cikkeket).

A kamera origójába helyezett koordináta rendszert alapul véve a kamera mozgását a három tengely mentén egyenesen haladó $\mathbf{T} = (T_x, T_y, T_z)^T$ (transzlációs) mozgással, és a három tengely körüli $\mathbf{\Omega} = (\Omega_x, \Omega_y, \Omega_z)^T$ forgással tudjuk jellemezni (lásd 5.4 ábra).

Ebben a rendszerben a tér egy tetszőleges pontjának sebessége:

$$\mathbf{U} = -(\mathbf{\Omega} \times \mathbf{X} + \mathbf{T}), \quad (5.5)$$



5.4. ábra. A kamera mozgásának jellemzése $\mathbf{T} = (T_x, T_y, T_z)^T$ és $\mathbf{\Omega} = (\Omega_x, \Omega_y, \Omega_z)^T$ vektorokkal

azaz minden merev testre igaz az, hogy mozgása egy haladó és egy forgó komponens összegéből áll. Ezek után fejezzük ki a képpontok sebességét úgy, hogy 5.4-ben kifejtjük a $\frac{dX}{dt}$, $\frac{dY}{dt}$ és $\frac{dZ}{dt}$ változókat 5.5 segítségével:

$$\mathbf{u} = \frac{1}{Z} \mathbf{A} \mathbf{T} + \mathbf{B} \mathbf{\Omega}, \quad (5.6)$$

ahol

$$\mathbf{A} = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix} \quad (5.7)$$

$$\mathbf{B} = \begin{bmatrix} \frac{xy}{f} & -\left(f + \frac{x^2}{f}\right) & y \\ f + \frac{y^2}{f} & -\frac{xy}{f} & -x \end{bmatrix}. \quad (5.8)$$

Vegyük észre, hogy az 5.6. egyenletben – amit a *mozgásmező alapegyenletének* is neveznek – arra keressük a választ, hogyan függ a mozgásmező vektorainak $\mathbf{u}(x, y)$ értéke a kamera mozgásának globális $\mathbf{T}, \mathbf{\Omega}$ értékeitől, valamint az $\mathbf{A}(x, y)$, $\mathbf{B}(x, y)$ és $Z(x, y)$ pozíció-függő változóktól. Azaz ha tudjuk a kamera nagyítását, a pontok távolságát és a kamera mozgását, akkor a mozgásmező egyértelműen meghatározott a kép minden pixelére. Az egyenlet első fele a kamera haladó mozgásnak hozzájárulását, a második fele pedig a forgó mozgás hozzájárulását adja meg $\mathbf{u}(x, y)$ vektorhoz. Érdekes a két tag hatását külön-külön is megvizsgálni.

5.4. Haladó mozgást végző kamera

Amennyiben a felvétel készítő kamera csak tisztán haladó mozgást végez ($\mathbf{\Omega} = \mathbf{0}$), az 5.6. egyenlet második tagja elhagyható:

$$\mathbf{u} = \frac{1}{Z} \mathbf{A} \mathbf{T}. \quad (5.9)$$

Az egyenlet bal oldalán lévő vektor vízszintes (horizontális) és függőleges (vertikális) komponensei kifejezhetők a következő formula szerint:

$$u_h = \frac{1}{Z}(x - x_0)T_z, \quad u_v = \frac{1}{Z}(y - y_0)T_z, \quad (5.10)$$

ahol

$$x_0 = f \frac{T_x}{T_z}, \quad y_0 = f \frac{T_y}{T_z}. \quad (5.11)$$

Mindez azt jelenti, hogy $\mathbf{u}(x, y)$ az $\mathbf{x} = (x_0, y_0)$ pontban $\mathbf{0}$ -t vesz fel. Ezt a speciális pontot FOE-nak (Focus of Expansion – Expanzió fókusz) nevezzük, mivel a szemlélő számára úgy tűnik, hogy a végtelen távolban ebből a pontból bontakoznak ki a felénk közeledni látszó objektumok. Az 5.5. ábra első példája egy olyan esetet mutat, ahol a FOE a kép közepén található.

A FOC-ról (Focus of Concentration – Koncentrációs fókusz) akkor beszélünk, ha nem közeledés, hanem távolodás figyelhető meg.

Példának okáért tekintsünk két esetet:

1. $T_z \neq 0 \Rightarrow$ Ebben az esetben a sebességkomponensek fordítottan arányosak a Z tengelyen mért távolsággal és egyenesen arányosak a FOE-től vett, képernyőn mért x ill. y tengely menti távolsággal.
2. $T_z = 0 \Rightarrow$ Ilyen esetekben a mozgásvektorok párhuzamosak egymással és fordítottan arányosak a Z tengelyen mért távolsággal. Pl. oldalra nézünk ki egy egyenesen haladó autó ablakából. (Ilyenkor azt látjuk, hogy a távoli hegyek alig mozdulnak el, miközben az út menti fák gyorsan rohannak el mellettünk, függetlenül attól, hogy a kép melyik részéről beszélünk hiszen \mathbf{u} sebesség vektor független az (x, y) pozíciótól.)

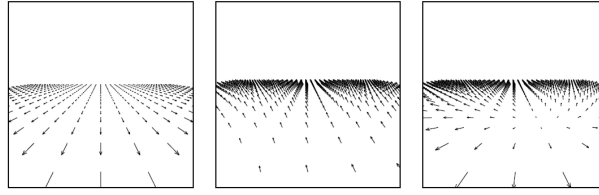
(Megjegyezzük, hogy a FOE pozíciója nem feltétlenül van a képkereten belül.)

5.5. Forgó mozgást végző kamera

Ha a felvétel olyan kamerával készül, amelyik tisztán forgó mozgást végez ($\mathbf{T} = \mathbf{0}$, $\boldsymbol{\Omega} \neq \mathbf{0}$), az 5.6. egyenletnek az 1. tagja hagyható el:

$$\mathbf{u} = \mathbf{B}\boldsymbol{\Omega}. \quad (5.12)$$

Az egyenletből látható, hogy a mozgásmező nem függ a pontok kamerától való térbeli távolságától, aminek az a közvetlen következménye, hogy kizárólagosan forgó mozgás esetén nincsen mód a tér szerkezetének 3D-s rekonstrukciójára. A képen mért elmozdulás Ω_x -nek és Ω_y -nak négyzetes függvénye, ha tehát kicsi a kamera látószöge, akkor ezek a komponensek kis hibával akár el is hanyagolhatóak. A képernyőn úgy látjuk, hogy a $\left(\frac{\Omega_x}{\Omega_z} f, \frac{\Omega_y}{\Omega_z} f\right)$ középpont körül forognak el a képpontok, azaz ebben a pontban az elmozdulás 0 lesz. (Ennek a kitüntetett pontnak a neve az angol irodalomban AOR (Axis of Rotation.))



5.5. ábra. Mozgó kamera speciális esetei sík felület felett. Balról jobbra: haladó mozgás, forgó mozgás, forgó és haladó mozgás.

5.6. Sík felület elmozdulása

Nagyon sok alkalmazásnál előfordul, hogy egy sík felület elmozdulását vizsgáljuk, vagy a kamera elmozdulását akarjuk megbecsülni, miközben egy többé-kevésbé sík felület képét látjuk. Túlnyomórészt ilyenek például a közlekedésben használt rendszerek, ahol az út vagy általánosabban a földfelszín síknak tekinthető [1]. Vegyük a sík egy \mathbf{P} pontját, amely a kameracentrikus koordináta rendszerben van megadva, \mathbf{n} pedig legyen ennek a síknak a normálvektora. Ekkor felírható a következő összefüggés:

$$\mathbf{n}^T \mathbf{P} = d, \quad (5.13)$$

ahol d a kamera középpont és a sík távolsága. Ha a kamera mozgását Ω elfordulás és \mathbf{T} haladó mozgás jellemzi, akkor 5.2 és 5.13 felhasználásával kapjuk a következő összefüggést:

$$\frac{n_x + n_y + n_z}{f} Z = d, \quad (5.14)$$

majd pedig ebből Z -t kifejezve és az 5.6. egyenletbe behelyettesítve megkapjuk, hogy sík felület esetén hogyan látszik a képernyőn a pontok mozgása:

$$\begin{aligned} u_h &= \frac{1}{fd} (a_1 x^2 + a_2 xy + a_3 fx + a_4 fy + a_5 f^2) \\ u_v &= \frac{1}{fd} (a_1 xy + a_2 y^2 + a_6 fy + a_7 fx + a_8 f^2). \end{aligned} \quad (5.15)$$

Az egyes együtthatók a következő értékeket veszik fel:

$$\begin{aligned} a_1 &= -d\Omega_y + T_z n_x, & a_2 &= d\Omega_x + T_z n_y, \\ a_3 &= T_z n_z - T_x n_x, & a_4 &= d\Omega_z - T_x n_y, \\ a_5 &= -d\Omega_y - T_x n_z, & a_6 &= T_z n_z - T_y n_y, \\ a_7 &= -d\Omega_z - T_y n_x, & a_8 &= d\Omega_x - T_y n_z. \end{aligned} \quad (5.16)$$

Mindezek alapján látható, hogy ilyen esetekben a képernyőn látható mozgás a koordináták másodfokú polinomiális függvénye, ahol az együtthatókat egyértelműen meghatározza a sík elhelyezkedése (d, \mathbf{n}) illetve a mozgás két paramétere (\mathbf{T}, Ω) . Ha \mathbf{n} helyére $\frac{\mathbf{T}}{\|\mathbf{T}\|}$ -t, \mathbf{T} helyére $\|\mathbf{T}\|\mathbf{n}$ -t, Ω helyére pedig $\Omega + \mathbf{n} \times \mathbf{T}/d$ írunk, nem változnak az a_i együtthatók. Tehát csupán a mozgásmező megfigyeléséből nem lehet egyértelműen megállapítani a sík paramétereit ill. a mozgás paramétereit, hiszen több olyan különböző eset is elképzelhető, amikor a paraméterek azonosak.

5.7. Mozgás-parallaxis

Általánosan tekintve a *parallaxis* a testek egymáshoz viszonyított helyzetének változása eltérő irányokból nézve. A mi tárgyalásunk fényében a mozgás-parallaxis annyit tesz, hogy a különböző pillanatokban egymásba eső pontok *relatív mozgásmezője* nem függ a kamera forgását meghatározó mozgás-komponenstől. Ezt úgy a legegyszerűbb szemléletesen elképzelni, ha a fejünket vízszintesen, függőleges tengely mentén elforgatva a távolban lévő tárgyak egymáshoz képesti pozíciója nem fog változni (a jelenséget az 5.6. ábra szemlélteti. Mivel a fejünk nem pont a szem középpontja, hanem nagyjából a nyak tengelye körül fordul el, ezért hogy ezt a hatást pontosan érzékeljük – az eltérést kompenzálva – a fejünket elfordulás közben kicsit mozgatni is kell.)

A jelenséget a következőképpen írhatjuk le formálisan. Legyen a vizsgált két képpont \mathbf{p}_1 és \mathbf{p}_2 , melyek egyazon (x, y) pozícióban láthatók t_1 és t_2 pillanatokban. Az 5.6. egyenletet felelevenítve könnyen beláthatjuk, hogy mivel a forgásból adódó komponens nem függ Z -től, így a két sebességvektor (u_{p_1}, u_{p_2}) csak az esetleges elmozdulásból adódóan lehet más. Írjuk fel tehát a *relatív mozgásmezőt* a két pontra, azaz a két időpillanatban vett vektor különbségét:

$$\begin{aligned}\Delta u_h &= u_{h,p_1} - u_{h,p_2} = (T_z x - T_x f) \left(\frac{1}{Z_{p_1}} - \frac{1}{Z_{p_2}} \right) \\ \Delta u_v &= u_{v,p_1} - u_{v,p_2} = (T_z y - T_y f) \left(\frac{1}{Z_{p_1}} - \frac{1}{Z_{p_2}} \right).\end{aligned}\quad (5.17)$$

Látható, hogy a relatív mozgásmező csak az elmozdulástól függ és minél nagyobb a különbség Z -ben, annál nagyobb lesz az eltérés mértéke.

5.6. ábra. A mozgás-parallaxis szemléltetése. Vajon a gömb, a hátsó kocka, vagy a felirat van közelebb? Ha a kamera jobbra-balra forog (fent) akkor ez nehezen eldönthető kérdés. Ha a kamera oldalirányban (középen) vagy előre halad (lent), akkor könnyű a válasz. Az animáció interaktív változata is elérhető [23].

6. fejezet

Optikai áramlás

Az eddigiekben \mathbf{u} -val jelöltük a képernyőn látható elmozdulás vektorokat, ezeknek az összességét pedig mozgásmezőnek neveztük. A gyakorlatban azonban ennek a meghatározása igen nehézkes és a legtöbb esetben csak a közelítő becslését határozhatjuk meg. Ezt a közelítő mozgásmezőt *optikai áramlás*-nak nevezzük.

A továbbiakban a mozgásbecslés alapvető feltételezéseit és módszereit tekintjük át. A számítások útján meghatározott optikai áramlás lehet

- pixel alapú, ekkor a kép minden pixeléhez rendelünk vektort (sűrű mozgásmező),
- blokk alapú, amikor bizonyos méretű blokkokhoz tartozik egy-egy vektor,
- tulajdonság alapú, amikor csak kitüntetett pontokhoz rendelünk mozgásvektorokat
- objektum alapú, amikor a képen látható objektumok területe felett határozzuk meg a vektorokat.

6.1. Az optikai áramlás meghatározásának módszerei

Mivel a videóelemzés egyik alapvető lépése a mozgásanalízis, ezért igen sokféle módszert dolgoztak ki az optikai áramlás becslésére. Ezek az alábbiak felsorolásszerűen:

- blokkegyezés alapú módszerek,
- Horn és Schunck algoritmus,
- Lucas és Kanade módszere,
- tér-időbeli szűrők,
- fázis korrelációs módszerek.

Dolgozatunkban ezek közül három alapvető megközelítést fogunk bemutatni a következő fejezetekben.

6.2. Az intenzitás-megmaradás elve

Ahhoz, hogy a képkockák között megfeleltetést tudjunk tenni, alapvetően azt kell hogy feltételezzük, hogy a 3D-s térben lévő objektumok kamerába vetülő színe két egymást követő felvétel során nem változik, a pontok csupán elmozdulnak kockáról kockára. Ez természetesen csak egy durva közelítése a valóságnak, hiszen akár a környezeti megvilágítást, akár az objektum saját fényét, akár a felületek irányát, akár a takarásokat vagy az objektumok alakjának változását tekintjük, mind sérti szigorú feltételezésünket. Azonban mégis ebből az egyszerű feltételből kiindulva olyan számítási modellek alkothatók, amelyek alkalmasak lesznek a mozgásmező közelítő becslésére. Az *intenzitás megmaradásának elve* tehát így írható fel:

$$f(x, y, t) = f(x + u_v, y + u_h, t + 1), \quad (6.1)$$

ahol u_v illetve u_h a pontok elmozdulását adja meg t és $t + 1$ időpillanatok között.

Az intenzitás-megmaradás elvét felírhatjuk a következő egyenlettel is:

$$f(x(t), y(t), t + 1) = c, \quad (6.2)$$

azaz egy c intenzitású pont pozíciója $x(t)$ és $y(t)$ függvények szerint változik az időben, miközben az intenzitás konstans marad. Vegyük az előbbi egyenletnek időbeli teljes deriváltját:

$$0 = \frac{d}{dt} f(x(t), y(t), t + 1) = \frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial y} \frac{dy}{dt} + \frac{\partial f}{\partial t} \frac{dt}{dt}. \quad (6.3)$$

Természetesen az x és y szerint deriváltak megfelelnek u_h -nak és u_v -nak. A fenti 6.3. egyenletet *mozgás-gradiens megszorításnak* nevezik, és le lehet vezetni másfajta okfejtéssel is, amit röviden bemutatunk a következőekben.

Az egyszerűség kedvéért most egydimenziós esetben nézzük meg ezt a levezetést. Két egymást követő pillanatban vegye fel f először f_1 -et, majd pedig f_2 -t. Feltesszük, hogy a pont intenzitása, színe nem változik, ekkor igaz az intenzitás-megmaradás elve: $f_2(x) = f_1(x - u)$. Ezek után írjuk fel $f_1(x - u)$ Taylor sorát az x pontban:

$$f_1(x - u) = f_1(x) - u \frac{df_1}{dx} + O\left(\frac{d^2 f_1}{dx^2}\right). \quad (6.4)$$

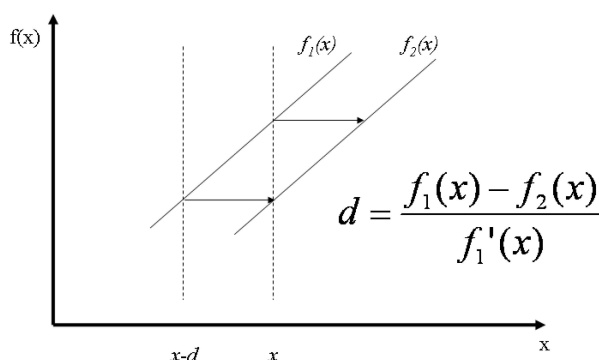
Az egyenletet átrendezve kapjuk a következő kifejezést:

$$f_1(x) - f_2(x) = u \frac{df_1}{dx} - O\left(\frac{df_1}{dx^2}\right), \quad (6.5)$$

majd pedig a magasabb rendű deriváltat elhanyagolva és az egyenletet átrendezve felírhatjuk a következőt:

$$0 = u \frac{df_1}{dx} + \frac{df_1}{dt}. \quad (6.6)$$

Ez utóbbi egyenlet pedig valójában megfelel a 6.3. kifejezésnek, igaz, most csak egydimenziós esetre megfogalmazva. Szemléltetésnek tekintjük a 6.1. ábrát.



6.1. ábra. Elmozdulás becslése deriváltak segítségével. Ha a képfüggvény lineárisan közelíthető, a 6.5. egyenlet magasabb rendű deriváltja elhagyható.

Megjegyzés Az f képfüggvény idő és térbeli differenciáljának számítása a képfeldolgozás eszközeivel igen egyszerű és gyorsan kivitelezhető. Az x és y irányú (azaz függőleges és vízszintes) differencia számítását a konvolúció egyenletével oldjuk meg, ahol a konvolúciós kernel mérete és alakja gondoskodik arról, hogy a diszkrét reprezentáció kellően sima ill. zajtűrő legyen:

$$\frac{df}{dx}(i, j) = (f * k)(i, j) = \sum_m \sum_n f(m, n) \cdot k(i - m, j - n), \quad (6.7)$$

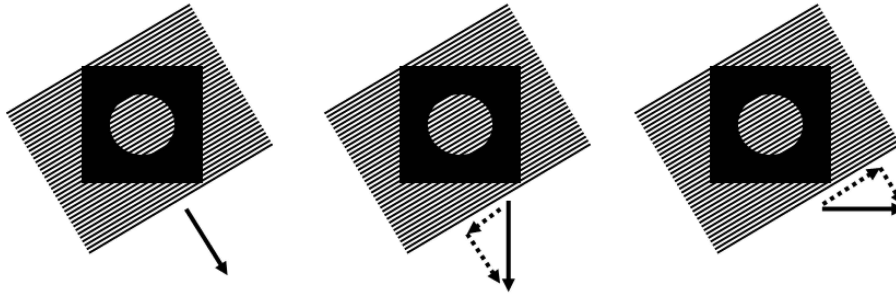
ahol m és n csupán a konvolúciós kernel méretének megfelelő területen végzi az összegzést, ugyanis azon kívüli területeken a kernel értéke mindenhol 0. A k konvolúciós kernel – függőleges és vízszintes differencia számítására – a Prewitt operátor esetén:

$$\mathbf{k}_v = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \mathbf{k}_h = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}. \quad (6.8)$$

Míg e két operátor vízszintes és függőleges irányban veszi a vizsgált ponttól jobbra és balra lévő intenzitásértékek különbségét, addig az időbeli differencia számítását rendszerint egyszerűen a két egymást követő képkocka különbségével végezzük el.

6.3. Az apertúra probléma

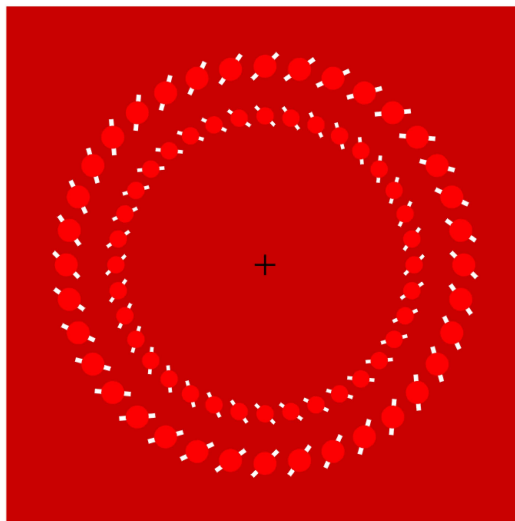
A mozgásbecslés különböző módjaival a következő fejezetekben foglalkozunk, azonban mindegyik megoldásra igaz lesz az, hogy lokális tulajdonságok alapján, az adott pixel környezetében való mérésekkel határozza meg a mozgásvektorokat. Ez azonban nem mindig jár sikerrel, ugyanis ha ebben a lokális környezetben a pixelintenzitások eloszlása egyenletes vagy az adott mozgással párhuzamos irányban nincs intenzitás-változás a jelben, akkor az elmozdulás nem jár vizuális hatással – legalábbis, ha megfigyelésünket a pont szűk környezetére koncentrálnak (lásd 6.2. ábra).



6.2. ábra. A lyukon keresztül szemlélve a vonalakkal párhuzamos mozgáskomponens nem detektálható. Ennélfogva hiába mozgatjuk a sraffozott mintát a fekete nyíl irányába – mint ahogy azt a középső és jobb oldali ábra illusztrálja –, azt mindig csak jobbra-le látjuk elmozdulni.

A probléma elkerülése érdekében (és a zaj hatásának eliminálására) mindegyik módszer valamilyen módon nem egy pixelre, hanem az adott pozíció egy-egy kisebb környezetére foglamazza meg az intenzitás megmaradás elvét.

Az apertúra probléma az emberi látórendszer esetében is könnyen előfordul, ennek érdekes illusztrációja a 6.3. ábra.



6.3. ábra. „Candies” – az apertúra probléma következménye, hogy a kép közepére koncentrálnva, a képhez közeledve ill. attól távolodva a két körmintát ellenkező irányban forgónak érzékeljük. (c) Akiyoshi Kitaoka 2003 – a szerző engedélyével

6.4. Blokkegyezés alapú algoritmus

A 6.1. egyenletben definiált intenzitás megmaradás elvéből egyenesen következik, hogy két képkocka pontjai közti egyezőséget egyszerűen meghatározhatjuk keresztkorreláció vagy négyzetes különbség alapján:

$$\mathbf{u}(x, y, t) = \arg \max_{u_v, u_h} \sum_{i=-M/2}^{M/2} \sum_{j=-N/2}^{N/2} f(x+i, y+j, t) f(x+i+u_v, y+j+u_h, t+1), \quad (6.9)$$

illetve

$$\mathbf{u}(x, y, t) = \arg \min_{u_v, u_h} \sum_{i=-M/2}^{M/2} \sum_{j=-N/2}^{N/2} (f(x+i, y+j, t) - f(x+i+u_v, y+j+u_h, t+1))^2, \quad (6.10)$$

ahol M és N a keresési ablak méretét jelöli.

Ezeknek a módszereknek előnyük az egyszerű implementáció, hátrányuk pedig az ablakméret, azaz a keresési tér növelésével négyzetesen növekvő számítási komplexitásuk, ill. hogy nehezen viselik a kép különböző torzulásait. Ezt elkerülendő, előszeretettel különböző módosításait alkalmazzák, lehetővé téve az eljárás invariáns viselkedését az átlagos intenzitás változására, vagy a kép affin transzformációjára. A keresés gyorsítására számos megoldás született a többfelbontású reprezentációtól kezdve különböző iteratív keresési megoldásokig [22].

6.5. Horn és Schunck algoritmus

Horn és Schunck algoritmus [6] az intenzitás-megmaradás elvét alapul véve az ún. mozgásgradiens elvét (6.3. egyenlet) követve törekszik arra, hogy megbecsülje u_v -t és u_h -t. Azokon a helyeken, ahol $\frac{\partial f}{\partial x}$ és $\frac{\partial f}{\partial y}$ értéke kicsi, ott jelentősen megnő a megoldás bizonytalansága, illetve a különböző zajhatások miatt is egy-egy pont önmagában sok esetben nem használható a mozgásvektor meghatározására. A probléma megoldására az algoritmus a vektormező simaságát feltételezi, ezért nem a 6.3 függvény minimalizálására törekszik, hanem a vektormező Laplace függvényét is beépíti a minimalizálandó E függvénybe:

$$E = \left(\frac{df}{dx} u_h + \frac{df}{dy} u_v + \frac{df}{dt} \right)^2 + \alpha^2 (\Delta u_h^2 + \Delta u_v^2), \quad (6.11)$$

ahol Δ az ún. Laplace operátor, amely a második parciális deriváltak összegét jelenti:

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}. \quad (6.12)$$

A szerzők eredeti cikkükben a Laplace értékének közelítésére a $3(\bar{u} - u)$ kifejezést használták, ahol \bar{u} az adott pixel környezetének súlyozott átlagát jelenti (1/6 és 1/12 súlyokkal). Az $(\bar{u} - u)$ kifejezés számítását a következő konvolúciós kernellel végezték [6]:

$$\mathbf{k}_v = \begin{bmatrix} 1/12 & 1/6 & 1/12 \\ 1/6 & -1 & 1/6 \\ 1/12 & 1/6 & 1/12 \end{bmatrix}. \quad (6.13)$$

A 6.11. egyenletben az α konstans szerepe a simaság biztosítása; minél nagyobb α értéke, annál simább eredményt várunk el \mathbf{u} -ra nézve. E minimumát úgy tudjuk megtalálni, ha vesszük a fenti 6.11. egyenlet u_h és u_v szerinti deriváltját és annak 0 helyét keressük. Az algebrai levezetéstől eltekintve megadjuk azt az egyenletet, amiből u_v és u_h már explicit kifejezhető:

$$\begin{aligned} \left(\alpha^2 + \left(\frac{df}{dx} \right)^2 + \left(\frac{df}{dy} \right)^2 \right) (u_h - \bar{u}_h) &= -\frac{df}{dx} \left(\frac{df}{dx} \bar{u}_h + \frac{df}{dy} \bar{u}_v + \frac{df}{dt} \right) \\ \left(\alpha^2 + \left(\frac{df}{dx} \right)^2 + \left(\frac{df}{dy} \right)^2 \right) (u_v - \bar{u}_v) &= -\frac{df}{dy} \left(\frac{df}{dx} \bar{u}_h + \frac{df}{dy} \bar{u}_v + \frac{df}{dt} \right). \end{aligned} \quad (6.14)$$

Ebből az egyenletből u_h és u_v már kifejezhető, de mivel \mathbf{u} függ $\bar{\mathbf{u}}$ -tól és vice-versa, ezért iteratív módon lehet kifejezni u_h -t és u_v -t:

$$\begin{aligned} u_h^{k+1} &= \bar{u}_h^k - \frac{\frac{df}{dx} \left(\frac{df}{dx} \bar{u}_h^k + \frac{df}{dy} \bar{u}_v^k + \frac{df}{dt} \right)}{\alpha^2 + \left(\frac{df}{dx} \right)^2 + \left(\frac{df}{dy} \right)^2} \\ u_v^{k+1} &= \bar{u}_v^k - \frac{\frac{df}{dy} \left(\frac{df}{dx} \bar{u}_h^k + \frac{df}{dy} \bar{u}_v^k + \frac{df}{dt} \right)}{\alpha^2 + \left(\frac{df}{dx} \right)^2 + \left(\frac{df}{dy} \right)^2}. \end{aligned} \quad (6.15)$$

ahol k jelenti az iterációs lépésszámot. A Horn és Schunck-féle módszer előnye, hogy képes valamelyest az apertúra problémára is megoldást adni, viszonylag gyorsan számolható, azonban többé-kevésbé zajérzékeny.

6.6. Lucas és Kanade algoritmus

Lucas és Kanade népszerű módszere (LK) [10] szintén a gradiens megszorításból indul ki és az előző módszerhez hasonlóan feltételezi, hogy lokálisan sima a keresett vektorfüggvény. A simaság azt jelenti, hogy egy tetszőlegesen választott pont környékén is nagyon hasonló u_v és u_h értékekre számítunk.

Ahhoz, hogy egy pontban meg tudjuk határozni a komponens értéket, minimum két egyenletre van szükségünk:

$$\begin{bmatrix} \frac{df}{dx}(x_1, y_1, t) & \frac{df}{dy}(x_1, y_1, t) \\ \frac{df}{dx}(x_2, y_2, t) & \frac{df}{dy}(x_2, y_2, t) \end{bmatrix} \begin{pmatrix} u_h \\ u_v \end{pmatrix} + \begin{pmatrix} \frac{df}{dt}(x_1, y_1, t) \\ \frac{df}{dt}(x_2, y_2, t) \end{pmatrix} = 0. \quad (6.16)$$

A felső egyenlet az (x_1, y_1) , míg a második sor az (x_2, y_2) pontra van felírva. Természetesen az egyenletet több pontra is felírva hasonló elvárásaink vannak, bár nem valószínű, hogy 0 lesz az egyenlet értéke, viszont a keresett \mathbf{u} minimalizálni fogja a kifejezést:

$$E(\mathbf{u}) = \sum_{x,y} G(x, y) \left(\frac{df}{dx} u_h(x, y, t) + \frac{df}{dy} u_v(x, y, t) + \frac{df}{dt}(x, y, t) \right)^2, \quad (6.17)$$

ahol $G(x, y)$ egy átlagoló függvény, ami tipikusan egy Gauss függvény alakját veszi fel, azaz a középponttól távolodva a távolság függvényében egyre kisebb súlyt alkalmazunk, miközben az összeadást (x, y) pont környezetében elvégezzük.

Mivel nem tudjuk garantálni, hogy 6.17. egyenlet pontosan 0 értéket vegyen fel, ezért annak minimumhelyét keressük úgy, hogy a deriváltjától várjuk el, hogy 0 legyen:

$$\begin{aligned} \frac{E(\mathbf{u})}{du_h} &= \sum_{x,y} G(x, y) \left(\left(\frac{df}{dx} \right)^2 u_h(x, y, t) + \frac{df}{dx} \frac{df}{dy} u_v(x, y, t) + \frac{df}{dx} \frac{df}{dt} (x, y, t) \right)^2 \\ \frac{E(\mathbf{u})}{du_v} &= \sum_{x,y} G(x, y) \left(\left(\frac{df}{dy} \right)^2 u_v(x, y, t) + \frac{df}{dx} \frac{df}{dy} u_h(x, y, t) + \frac{df}{dy} \frac{df}{dt} (x, y, t) \right)^2. \end{aligned} \quad (6.18)$$

Mátrix formulával felírva kapjuk:

$$\mathbf{M}\mathbf{u} + \mathbf{b} = \mathbf{0}, \quad (6.19)$$

ahol

$$\mathbf{M} = \begin{bmatrix} \sum G \left(\frac{df}{dx} \right)^2 & \sum G \frac{df}{dx} \frac{df}{dy} \\ \sum G \frac{df}{dx} \frac{df}{dy} & \sum G \left(\frac{df}{dy} \right)^2 \end{bmatrix} \quad (6.20)$$

és $\mathbf{b} = \begin{pmatrix} \sum G \frac{df}{dx} \frac{df}{dt} \\ \sum G \frac{df}{dy} \frac{df}{dt} \end{pmatrix}$.

\mathbf{u} -t megkaphatjuk:

$$\mathbf{u} = -\mathbf{M}^{-1}\mathbf{b}, \quad (6.21)$$

feltéve, hogy M invertálható. Belátható továbbá, hogy:

$$\det(M) = \lambda_1 \lambda_2 = 0, \quad (6.22)$$

azaz M nem invertálható, ha egyik vagy mindkét (λ_1 és λ_2) sajátértéke 0. Ennek elkerülése miatt van szükség erős simításra, azaz minél nagyobb területen való átlagolásra.

A Lucas és Kanade algoritmus fő korlátozását az jelenti, hogy a deriváltak segítségével csak kisebb elmozdulások becsülhetők. Ennek a legegyszerűbb feloldását az algoritmus többfelbontásos alkalmazása jelenti [21], amikor egy ún. kép-piramist¹ generálunk a képekből, és a piramis különböző szintjeit mind felhasználjuk a becslésre. A piramis magasabb szintjein – ahol a kép kicsinyítve áll elő – egy-egy pixelyi elmozdulás több pixelyi távolságnak felel meg az eredeti képen.

Az algoritmus kimenetének illusztrációját a 6.4. ábra szemlélteti.

6.7. A mozgásbecslés korlátozásai

Az a tapasztalat, miszerint a különböző módszerek nem tökéletesen működnek a legtöbb videófelvétel esetén, számos okra vezethető vissza. Két alapvető ok az intenzitás-megmaradás

¹A kép-piramisok alkalmazása igen hamar elterjedt a képfeldolgozásban, már a 80-as években megjelentek algoritmusok a képek többszintű reprezentációjára [20].

6.4. ábra. Mozgásbecslés a Lucas és Kanade algoritmussal. A felület mintázata, a mozgás jellege és a geometriai torzulások miatt a becslés nem mindig pontos. Míg a fenti kontrasztos ábrán a mozgás nem túl jól követhető, a lenti ábrán kisebb a követés hibája. Az animáció interaktív változata is elérhető [23].

elvének sérülése és az, hogy a becselő eljárások jelentős része a képek deriváltjának becslését végzi, és diszkrét képeken az idő vagy térbeli differenciálás olykor csak igen pontatlanul számítható ki.

Konkrétan a következő jelenségek okoznak nehézséget:

- fényviszonyok változása: lámpa ki/be kapcsolása, napsütés, felhősödés;
- objektumok takarásba kerülése, kitakarása;

- sima felületek jelenléte: lásd 6.3. fejezet;
- nem merevtest mozgás: pl. ruha gyűrődik;
- túl nagy vagy túl kicsi elmozdulások: gyors mozgás esetén az időbeli derivált becslése nem lehetséges;
- zaj: gyenge fényviszonyok esetén jelentős termikus zaj rontja a mozgásbecslő egyenletek minimalizálását.

7. fejezet

Mozgáskövetés

Az előző fejezetben csupán néhány képkockát használtuk fel a mozgás megbecslésére, az optikai áramlást meghatározó módszerek tipikusan csak 2 képkocka elemzését végzik. Ugyanakkor az eddig ismertetett módszerek alapvetően nyers adatok (blokkok, pixelek) hasonlóságát vizsgálták a képkockák között, nem törekedtek objektum szintű elemzésre.

A *mozgáskövetés* feladata objektumok közti kapcsolat megteremtése több képkockán keresztül, azaz segítségükkel lehetséges objektumok útvonalának végigkövetése egy videószekvencián. Azonban a gyakorlatban az objektumok elhatárolása a környezetüktől, körvonaluknak pontos meghatározása sokszor igen nehéz feladat, a különböző követő eljárások nem csak a képpontok színét, hanem sebességét az ívek alakját és a képrészletek egyéb jellemzőinek paramétereit is beépítik modelljükbe.

Alapvetően léteznek *top-down* módszerek, amelyek valamilyen – az objektumra jellemző – template-et, modellt fognak követni (pl. részecske követő) és vannak *bottom-up* megközelítések, amelyek a képet szegmentálják, és a különböző foltok követését végzik, így ezekből építkezve követik a tényleges (kiterjedt) objektumokat.

Más megközelítés szerint: ha két képkocka között vizsgáljuk a tartalmi egyezést, akkor tulajdonság egyezésről, ha több kocka között, akkor tulajdonság követésről beszélünk. Az előbbi a sztereó algoritmusokban használatos, míg utóbbi a mozgáselemzésben.

Alapvető kérdésként merül fel, hogy egy adott kép esetén mely pontok azok, amelyeket elméletileg könnyen lehet követni és megfelelő információt adnak a mozgásról.

7.1. Kanade-Lucas-Tomasi követő

A Kanade-Lucas-Tomasi (KLT) követő lényege az LK algoritmus (lásd 6.6. fejezet) továbbgondolása. Bár két képkocka közti kapcsolat megállapítására alapvetően hatékony a korábban ismertetett algoritmus, azonban könnyen hibázhat az eljárás abban az esetben, ha a vizsgált képterületek nem eléggé változatosak (aperúra probléma), vagy ha időben túl sokat változnak a pixelértékek néhány képkocka alatt.

Az első eset elkerülése érdekében csak olyan pontokat érdemes követni, ahol a mozgásvektorok eleve jól meghatározhatóak [16] (tehát nem sima területekről van szó). Ezt a feltételt azok a pontok teljesítették, amelyek megfeleltek a 6.22. egyenletnek, hiszen 6.17 optimális megoldásához az M mátrix sajátértékei nem lehetnek nullák.



7.1. ábra. A KLT kimenete 3 különböző képre. Jól látható, hogy az élek találkozásánál lettek kijelölve pontok.

A [14] cikkben az eredeti elgondolást továbbfejlesztve azonban azt is felteszik a szerzők, hogy nem elengedő egymást követő képkockákon erős éltartalmú és egymáshoz hasonló részeket találni, hanem akkor lehet sikeres a követés, ha sikerül nagyobb időbeli differencia esetén is hasonlóságot találjunk. Ekkor azonban már megengedjük a minták ún. affin transzformációját, mivel a kamera elfordulása, közeledése–távolodása (kicsinyítés–nagyítás) a látott objektumoktól már jelentősebb lehet. (Az eljárás illusztrációját a 7.1. ábra mutatja.)

7.2. Követés mint valószínűségi következtetés

A követési feladatok viszonylag nagy bizonytalansággal bírnak, mivel az objektumok színe, alakja és mozgásának jellemzői nem feltétlenül állandóak, valamint méréseink, kamerás megfigyeléseink általában torzultak, zajjal terheltek. Ebből a megfontolásból célszerű valószínűségi modelleket alkotni és segítségükkel optimális becslést adni az objektumok helyzetére. Kézenfekvő a Bayes tétel használata, miszerint:

$$P(\mathbf{x}_t | \mathbf{z}_t) = \frac{P(\mathbf{z}_t | \mathbf{x}_t) P(\mathbf{x}_t)}{\mathbf{z}_t} = \propto P(\mathbf{z}_t | \mathbf{x}_t) P(\mathbf{x}_t), \quad (7.1)$$

ahol \mathbf{x}_t jellemzi a megfigyelt terület állapotát – pl. pozícióját, sebességét, színét, vagy akár egy vonal görbületét –, \mathbf{z}_t pedig a konkrét megfigyelésünket, mérésünket, azaz a ténylegesen ismert adatokat a t időpontban. (A \propto skálázás erejéig egyenlő jelölés arra utal, hogy a két oldalon lévő mennyiség egymás konstansszorososa.) A két (vektor) változó nem feltétlenül ugyanazokat a típusú adatokat tartalmazza, például lehetséges olyan modellt használni, amikor csak az egyes pontok pozícióját ($\mathbf{z} = (x, y)^T$) mérjük, míg a pont állapotához a sebessége ($\mathbf{x} = (x, y, \dot{x}, \dot{y})^T$) is hozzátartozik.

A feladatot tehát így tudjuk megfogalmazni: Keressük azon $\hat{\mathbf{x}}$ állapotváltozót, ami adott megfigyelés esetén a maximális valószínűséggel bír. Ehhez elegendő az előző egyenlet jobb oldalát kiértékelni különböző lehetséges \mathbf{x}_t értékek esetén, miközben \mathbf{z}_t változatlan marad:

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} P(\mathbf{z}_t | \mathbf{x}_t) P(\mathbf{x}_t). \quad (7.2)$$

A kérdés tehát az, miként tudunk a két tag valószínűségére becslést tenni. Az első taghoz olyan H becslő függvényt kell konstruálni, ami esetén $\|\mathbf{z}_t - \mathbf{H}(\mathbf{x}_t)\|$ nagysága fordítottan arányos lesz $P(\mathbf{z}_t | \mathbf{x}_t)$ valószínűséggel. A fenti példa esetén H megadható a következő mátrix operátorral:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (7.3)$$

$P(\mathbf{x}_t)$ prior értékére a korábbi megfigyelésekből tudunk becsléseket adni. Amennyiben a mozgás folyamatát Markov láncként modellezük, feltételezésünk azt jelenti, hogy a követendő pont előző állapotaitól függ az aktuális állapota. Első rendű Markov lánc esetén:

$$P(\mathbf{x}_t | \mathbf{x}_{t-1}) = P(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \mathbf{x}_{t-3}, \dots), \quad (7.4)$$

azaz elég egy korábbi állapot ismerete, míg magasabb rendű Markov lánc esetén több korábbi állapotot is figyelembe kell venni $P(\mathbf{x}_t)$ becsléséhez. Első rendű Markov modellnek megfelelően 7.2 így írható fel:

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} P(\mathbf{z}_t | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{x}_{t-1}). \quad (7.5)$$

Tekintsünk egy egyszerű példát F állapot-becslő operátorra, ami \mathbf{x}_t becslését $\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1}$ módon végzi:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7.6)$$

Utolsó lépésként tehát olyan valószínűség függvényt kell konstruálni, ami esetén $\|\mathbf{x}_t - \mathbf{F}\mathbf{x}_{t-1}\|$ nagysága fordítottan arányos lesz a valószínűséggel.

7.3. A Kálmán szűrő és alkalmazása

A Kálmán szűrő¹ egy rekurzív szűrő, ami egy dinamikus rendszer állapotát becsüli meg zajos, vagy nem teljes mérések sorozatából [9].

A legegyszerűbb lineáris dinamikus rendszerben (ang. Linear Dynamical System) a diszkrét idejű modell rejtett állapotára és a megfigyelésekre a következő egyenleteket fogalmazzuk meg:

$$\begin{aligned}\mathbf{x}_t &= \Phi \mathbf{x}_{t-1} + \epsilon_t \\ \mathbf{z}_t &= \mathbf{H} \mathbf{x}_t + \mu_t,\end{aligned}\quad (7.7)$$

ahol Φ az állapot átmenetet határozza meg, H az állapot és a megfigyelés kapcsolatát definiálja, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ és $\mu \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ a különböző zajok jellemzéséért felelős normál eloszlású ($\mathbf{0}$ várható értékű és \mathbf{Q} ill. \mathbf{R} kovarianciájú) változók.

A Kálmán szűrő predikciós-korrektíós fázisokból álló iteratív számítási folyamat, aminek a célja, hogy a becslés során négyzetes értelemben legjobban közelítse a rendszer állapotváltozóját.

A becslési folyamatban a következő változókat vezetjük be:

- $\hat{\mathbf{x}}_t = \Phi \mathbf{x}_{t-1}$: állapot becslése a megelőző állapot alapján
- $\hat{\mathbf{z}}_t = \mathbf{H} \mathbf{x}_t$: a mérési érték becslése az aktuális állapot alapján
- $\hat{\mathbf{P}}_t = E(\Delta \mathbf{x} \Delta \mathbf{x}^T) = \Phi \mathbf{P}_{t-1} \Phi^T + \mathbf{Q}$: állapot becslés hibájának kovarianciája, ahol a becslési hiba $\Delta \mathbf{x} = \mathbf{x} - E(\hat{\mathbf{x}})$
- $\mathbf{v}_t = \mathbf{z}_t - \hat{\mathbf{z}}_t$: a megfigyelés korrekciós értéke
- $\mathbf{S}_t = \mathbf{H} \hat{\mathbf{P}}_t \mathbf{H}^T + \mathbf{R}$: \mathbf{v}_t pontosítási érték kovarianciája
- $\mathbf{K}_t = \hat{\mathbf{P}}_t \mathbf{H}^T \mathbf{S}_t^{-1}$: erősítési tényező, a korrekció mértékét tudjuk vele beállítani
- $\mathbf{x}_t = \hat{\mathbf{x}}_t + \mathbf{K}_t \mathbf{v}_t$: állapotváltozó korrekciója
- $\mathbf{P}_t = (\mathbf{I} - \mathbf{K} \mathbf{H}) \hat{\mathbf{P}}_t$: \mathbf{P}_t frissítése, ahol \mathbf{I} az identikus mátrix.

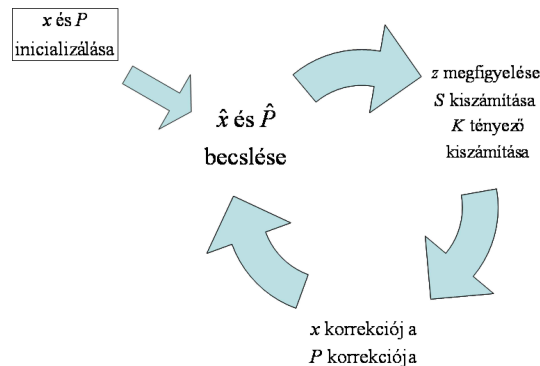
Elttekintve az algoritmus levezetésétől, csak annak lépéseit ismertetjük a most bevezetett változók felhasználásával:

1. válasszunk kiindulási értéket $\hat{\mathbf{x}}$ -nek és $\hat{\mathbf{P}}$ -nek,
2. végezzük el az állapot becslését: $\hat{\mathbf{x}}_t = \Phi \mathbf{x}_{t-1}$,

¹Kálmán Rudolf Emil (Budapest, 1930. május 19.) amerikai-magyar villamosmérnök, matematikus. Munkásságának fő eredményeit a folyamatirányításban, szabályozáselméletben és operációkutatásban érte el az USA-ban. Szűrőssel kapcsolatos elképzeléseit eleinte erős kritika fogadta, ezért azokat először a mechanika területén publikálta, később azonban igen jelentős sikert ért el a villamosmérnöki és informatikai területeken.

3. tegyünk becslést \mathbf{P} -re: $\hat{\mathbf{P}}_t = \mathbf{\Phi}\mathbf{P}_{t-1}\mathbf{\Phi}^T + \mathbf{Q}$,
4. végezzük el \mathbf{z}_t mérését,
5. számoljuk ki a becslés hibáját: $\mathbf{v}_t = \mathbf{z}_t - \hat{\mathbf{z}}_t$,
6. ennek kovarianciáját: $\mathbf{S}_t = \mathbf{H}\hat{\mathbf{P}}_t\mathbf{H}^T + \mathbf{R}$,
7. számoljuk ki az erősítési tényezőt: \mathbf{K}_t ,
8. majd következik a \mathbf{x}_t korrekciója
9. és \mathbf{P}_t frissítése,
10. végül pedig ugorjunk a 2. lépéshez és ismételjük meg a lépéseket a különböző időpillanatokra.

A Kálmán szűrő alkalmazásának több előnye is van: míg a számítások nagy része viszonylag egyszerűen elvégezhető mátrix műveletekkel, a memória igénye sem túl nagy a módszernek, mivel a becslés implicit tartalmazza a korábbi állapotokra vonatkozó információkat, elegendő csak egy megelőző állapotot változóit tárolni. A paraméterbecslés valós időben meghatározható, s közben a becslési hibákat folyamatosan lehet monitorozni, ebből pedig következtetni lehet, ha a rendszerben hirtelen jelentősebb változások állnak elő.



7.2. ábra. A Kálmán szűrő egymás követő predikciós–korrekciós fázisai

7.3.1. A Kálmán szűrő alkalmazása vizuális követésre

A Kálmán szűrő igen egyszerű és közvetlen alkalmazása vizuális követésre, ha egy bináris előtér-maszk pozíciójának követésére alkalmazzuk. Első lépésként egy előtér-háttér szegmálással a mozgó objektumoknak megfelelő foltokat detektáljuk, majd meghatározzuk ezek tömegközéppontjának koordinátáit. Ezen koordinátákat tekintjük az objektum állapotának, a szűrő feladata pedig ezek követése képkockáról képkockára.

Az eljárás a következő lényegi lépésekből áll (lásd 7.2. ábra):

1. Inicializálás: A képen megkeressük az objektumokat, illetve az azoknak megfelelő előtér maszkot.
2. Jóslás: A Kálmán szűrő segítségével megjósoljuk az objektum helyzetét a következő képkockán az eddigi információk alapján.
3. Korrekció: Megkeressük a legközelebbi objektumot a jósolt pozíció környezetében, majd a tényleges pozícióját felhasználjuk az állapot korrigálására.

A szűrő előnye, hogy alapvetően megtudjuk, hogy kb. hol kell keresni az objektumot, így pl. a blokkegyezés algoritmus futása felgyorsulhat. Nagy előnye a Kálmán szűrőnek, hogy kisebb kitakarásokat kezelni tud. Ez azt jelenti, hogy ha két objektum egymás pályáját keresztezi, akkor az egyik kitakarhatja a másikat, és a kitakarás után nem triviális, hogy melyik objektum melyik volt a kitakarás előtti pillanatban. Ebben segít a Kálmán szűrő. Ha azt vesszük észre, hogy az objektumot eltakarta egy másik objektum, akkor a Kálmán szűrővel csak a következő állapot jóslását végezzük, és nem korrigáljuk az állapotot. Így ha a kitakarás megszűnt, akkor a jósolt állapot az objektum valódi pozíciója közelében kell, hogy legyen. Bővebben a Kálmán szűrő alkalmazásáról a vizuális követésben többek között Cuevas, Zaldivar és Rojas munkájában olvashatunk [2].

7.4. Részecske szűrő – részecske követő

Bár a Kálmán szűrő sok pozitív tulajdonsággal rendelkezik, alapvetően gyenge pontja, hogy Gauss sűrűségfüggvényen alapul, ami lényegileg csak egy munkahipotézis modellezését teszi lehetővé. Alternatív hipotéziseket nem tud kezelni, így nehezen oldja meg azokat az eseteket, amikor egy kiterjedt objektum alakja viszonylag gyorsan változik (pl. nem szilárdtest mozgás) és több ilyen objektum van összezsúfolva a képen. Bár léteznek a Kálmán szűrőnek továbbfejlesztett verziói a bonyolultabb esetek kezelésére, a részecske szűrő megalkotásakor ezeket a szempontokat fokozott hangsúllyal vették figyelembe.

A részecske szűrő módszereket a 90-es évek közepén dolgozták ki [3], [7], azóta igen nagy számban jelentek meg különböző változatai, speciális alkalmazásai (pl. [12]).

A megközelítés szerint a rendszer állapotát egy N db mintából álló halmaz $S = \{(s^i, p^i) | i = 1, \dots, N\}$ jellemzi, maguk a minták az ún. súlyozott részecskék. s^i jelenti a lehetséges állapotokat, p^i a hozzájuk rendelt súlyokat, ami gyakorlatilag annak a valószínűségnek felel meg, miszerint a követendő objektum s^i állapotban van ($\sum_{i=1}^N p^i = 1$). Ebből a konstrukcióból következik, hogy igen sokféle minta-halmazt tudunk létrehozni, nem kell valamilyen könnyen definiálható függvény alakot (pl. Gauss) követni az állapoteloszlásnak, modellünket folyamatosan a mérésekhez tudjuk igazítani. (Tárgyalásunkban az egyszerűség kedvéért skálár változókat használunk, de természetesen az itt bemutatott modell alkalmazható vektorral jellemezhető adatok esetén is.)

Az algoritmus egy adott pillanatában az objektum átlagos állapota:

$$E(S) = \sum_{i=1}^N p^i s^i. \quad (7.8)$$

A modellezés során tehát a halmaz méretétől függően a kevésbé valószínű állapotokat is megtartjuk p^i valószínűséggel:

$$p^i = P(z_t | x_t = s_t^i), \quad (7.9)$$

ami azt fejezi ki, hogy mennyire felel meg az adott állapot az aktuális z_t megfigyelésnek. Így ha pl. az objektum takarásba kerül rövidebb ideig, nem veszítjük el róla az információkat, csupán az adott súlytényezők fognak csökkenni, majd pedig ha kikerül a takarásból, a megfelelő súlyok újra növekedhetnek. Vegyük észre, hogy 7.9 nem felel meg a szokásos posteriori valószínűségnek (lásd 7.1), aminek az az oka, hogy az állapotok determinisztikus módon fognak egymásból következni, tehát nincs értelme $P(x_t)$ ill. $P(x_t | x_{t-1})$ kiértékelésének.

Az algoritmusnak ezt a determinisztikus voltát az fogja ellensúlyozni, hogy bizonyos véletlenszerűséggel új részecskéket fogunk kiválasztani, használni, így ha s_{t+1}^i becslése önmagában nem is lenne elég jó, és emiatt a pontot elveszítenénk, az állapothalmaz folyamatos frissítésével jó esélyünk van, hogy mindig lesz olyan részecskénk, ami a követendő objektumhoz tartozik.

A részecske szűrők működése három fő lépésből áll:

1. Mintavétel: válasszunk ki a képből a részecskéket. Adott pont kiválasztásának valószínűségét p_i határozza meg. Nagy valószínűséggel lesznek olyan részecskéink, amelyek hasonlítanak a követendő tárgyra, de lesz néhány olyan is, ami éppen nem.
2. Becslés: becsüljük meg, hogy merre mozognak a részecskéink. Ez egy nagyrészt determinisztikus lépés (bár véletlen zaj is része a modellnek):

$$s_{t+1}^i = A s_t^i + n, \quad (7.10)$$

ahol A az elmozdulás függvénye, n pedig egyedi zaj. A is figyelembe veheti s^i egyedi tulajdonságait (mint pl. pozíció, sebesség, gyorsulás).

3. Mérés: értékeljük ki az egyes részecskék vizuális tulajdonságait, mennyire hasonlít az adott helyen látott kép a várt modellhez (pl. egy követendő arc képéhez).

A követendő objektum modell-paramétereit folyamatosan lehet frissíteni, de csak akkor, ha nincs nagy változás a tulajdonságokban. Utóbbi esetben ugyanis valószínűleg takarásba került a követendő objektum és nem szeretnénk azt, hogy hamar elveszítsük, majd a takaró objektumot kövessük a következő lépésekben:

$$modell_{t+1} = (1 - \alpha)modell_t + \alpha n, \quad (7.11)$$

ahol n az új modell paramétereit jelenti, α pedig az aktuális egyezés függvénye.

Harmadik rész

Függelék

A függelék

Projektív geometria és lineáris algebra alapok

Itt a teljesség igénye nélkül röviden összefoglaljuk a könyv megértéséhez szükséges legfontosabb matematikai alapfogalmakat.

A.1. Projektív sík és projektív tér

A mindenki által ismert euklideszi geometria jól reprezentálja a bennünket körülvevő valós világot: az tárgyakkal jól meghatározott mérete van; az egymást metsző egyenesek meghatározott szöveget zárnak be; illetve párhuzamosak, ha ugyanarra a síkra esnek és nem metszik egymást. Mindezen tulajdonságok nem változnak egy euklideszi (vagy merev test) transzformáció során, mely eltolásból és forgatásból áll.

A helyzet azonban egészen más egy projektív kamera képén: sem a méretek sem a szögek nem marad változatlanok, sőt a párhuzamos egyenesek egymást metszhetnek. Ezen jelenségek hatékony reprezentálására és leírására szolgál a *projektív geometria*, mely -hasonlóan az euklideszi geometriához- több dimenziós lehet. Ezek közül számunkra a 2D *projektív sík* illetve a 3D *projektív tér* megismerése lesz hasznos. Egy projektív kamera lényegében a projektív térből a projektív síkra képez le. Az egyszerűség kedvéért az alapfogalmakat a projektív síkon vezetjük be, és a projektív tér esetén a különbségeket tárgyaljuk.

A.1.1. Homogén koordináta és ideális pont

Tekintsünk egy $\mathbf{x} = (x_1, x_2)^T$ pontot az euklideszi síkon. Ugyanezen pont koordinátáját a projektív

síkon megkapjuk egy harmadik koordináta hozzávételével, melynek értéke 1: $(x_1, x_2, 1)^T$. Ezt a koordinátát nevezzük *homogén koordinátának*, míg az euklideszit *inhomogén koordinátának*. Általánosan is igaz, hogy egy n dimenziós euklideszi pont homogén koordinátáját egy $(n + 1)$ -edik koordináta hozzávételével kapjuk, melynek értéke 1. A homogén koordináták egyik legfontosabb tulajdonsága, hogy globális skálázásra érzéketlenek, vagyis $\forall \alpha \neq 0: (x_1, x_2, 1)^T = (\alpha x_1, \alpha x_2, \alpha)^T$. Ennek megfelelően a $(0, 0, 0)^T$ homogén koordináta nem

megengedett. Homogén koordinátából könnyen visszatérhetünk inhomogénbe (vagyis a projektív térből euklideszi térbe), ha az utolsó koordinátával rendre elosztjuk a koordinátákat. Ezt nevezzük *homogén osztásnak*.

Vegyük észre, hogy a projektív térnek több pontja van mint az euklideszinek: ezen pontok utolsó koordinátája 0, és így ezen pontok nem vihetők át az euklideszi térbe. Ezeket a pontokat *ideális pontoknak* vagy *végtelen távoli pontoknak* nevezzük. Ezek a pontok lényegében *irányoknak* felelnek meg, pl. a $(0,1,0)^T$ pont a sík vertikális irányának felel meg. A projektív síkon az ideális pontok a *végtelen távoli egyenesen* helyezkednek el, míg a projektív tér ideális pontjai alkotják a *végtelen távoli síkot*. Az előbbi az $\mathbf{l}_\infty = (0,0,1)^T$ egyenes, míg az utóbbi a $\boldsymbol{\pi}_\infty = (0,0,0,1)^T$ sík.

A.1.2. Egyenes és sík

Az euklideszi síkon egy egyenes egyenletét felírhatjuk a jól ismert $ax_1 + bx_2 + c = 0$ alakban. Mivel az egyenlőség változatlanul érvényes marad egy tetszőleges, nullától különböző skálázás után: $ax_1X_3 + bx_2X_3 + cX_3 = 0$, amit írhatunk $\mathbf{l}^T \mathbf{X} = \mathbf{X}^T \mathbf{l} = 0$ alakban is, ahol $\mathbf{l} = (a, b, c)^T$ az egyenes, $\mathbf{X} = (X_1, X_2, X_3)^T$ pedig az egyenes egy pontja. Ebből látható, hogy a projektív síkon az egyenes és pont reprezentációja megegyezik és egymás duálisai. Ezen reprezentáció nagy előnye, hogy két $\mathbf{X}_1, \mathbf{X}_2$ pont által meghatározott egyenest egyszerűen az $\mathbf{l} = \mathbf{X}_1 \times \mathbf{X}_2$ alakban kapjuk, akárcsak két $\mathbf{l}_1, \mathbf{l}_2$ egyenes metszéspontját: $\mathbf{X} = \mathbf{l}_1 \times \mathbf{l}_2$.

A projektív térben az egyenesek reprezentációja lényegesen bonyolultabb, de erre általában nincs is szükségünk. A pont és sík azonban egymás duálisai, és egy $\boldsymbol{\pi}$ sík \mathbf{X} pontjai kielégítik a $\mathbf{X}^T \boldsymbol{\pi} = 0$ egyenletet. Fontos még megemlíteni a *síkhomográfia* fogalmát, amely két sík között definiált projektív transzformációt jelent. Mivel itt nem történik dimenzió csökkenés, ezért a síkhomográfia mátrixának rangja maximális és így invertálható.

A.1.3. Kúpszelet

Egy kúpszelet \mathbf{X} pontjait egy másodrendű egyenlettel szoktuk megadni: $aX_1^2 + bX_2^2 + cX_3^2 + dX_1X_2 + eX_1X_3 + fX_2X_3 = 0$, melyet írhatunk mátrix alakban is: $\mathbf{X}^T \mathbf{C} \mathbf{X} = 0$, ahol

$$\mathbf{C} = \begin{bmatrix} a & d/2 & e/2 \\ d/2 & b & f/2 \\ e/2 & f/2 & c \end{bmatrix} \quad (\text{A.1})$$

szimmetrikus mátrix. Mivel pont és egyenes egymás duálisai, ezért nem meglepő, hogy a kúpszelet önmaga duálisai lesz. A kúpszeletet ugyanis megadhatjuk pontjaival, ahogyan az előbb láttuk, de megadhatjuk az \mathbf{l} érintő egyenesei által is, melyet a \mathbf{C}^* duális kúpszeletnek hívunk: $\mathbf{l}^T \mathbf{C}^* \mathbf{l} = 0$. A duális egyszerűen $\mathbf{C}^* = \mathbf{C}^{-1}$ alakban írható fel egy skálafaktor erejéig.

A kúpszeletek fontos tulajdonsága, hogy minden kör pontosan két rögzített pontban metszi a végtelen távoli egyenest. A kör mátrixa diagonális lesz, és egyenlete $X_1^2 + X_2^2 + X_3^2 = 0$ alakú, amiből következik, hogy a végtelen távoli egyenessel vett metszete $X_1^2 + X_2^2 = 0$, hiszen ekkor $X_3 = 0$. Ennek az egyenletnek két komplex gyöke van: $(1, \iota, 0)^T$ és $(1, -\iota, 0)^T$, melyeket *circuláris* vagy *abszolút pontoknak* nevezünk.

A.2. Mátrixok felbontása

A továbbiakban a könyvben használt mátrix faktorizációkat tekintjük át röviden.

A.2.1. Cholesky felbontás

A Cholesky felbontást szimmetrikus pozitív definit mátrixok esetén használhatjuk, amely a kiindulási \mathbf{A} mátrixot egy alsó trianguláris mátrix és annak konjugált transzponáltjának szorzatára bontja: $\mathbf{A} = \mathbf{L}\mathbf{L}^*$. Az \mathbf{L} és \mathbf{L}^* mátrixok egyértelműen meghatározottak.

A.2.2. RQ vagy QR felbontás

Egy négyzetes \mathbf{A} mátrix felbontható $\mathbf{A} = \mathbf{QR}$ alakban, ahol \mathbf{Q} egy ortogonális mátrix, vagyis teljesül rá a $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ összefüggés, míg \mathbf{R} felső trianguláris. Megjegyezzük, hogy ez a felbontás nem egyértelmű.

A.2.3. SVD felbontás

Könyvünkben a leggyakrabban használt felbontás az SVD, vagy szinguláris érték szerinti felbontás, mely egy tetszőleges $m \times n$ mátrixot három mátrix szorzatára bont: $\mathbf{A} = \mathbf{UDV}^T$, ahol a \mathbf{D} diagonális mátrix az \mathbf{A} szinguláris értékeit tartalmazza nagyság szerint rendezve, és mint ilyen, egyértelműen meghatározott. A másik két mátrix oszlopai ortogonálisak, de maguk a mátrixok nem egyértelműek. Az SVD felbontás segítségével oldhatunk meg például lineáris túlhatározott egyenletrendszereket.

Irodalomjegyzék

- [1] C. Braillon, C. Pradalier, J. L. Crowley, C. Laugier, Real-time moving obstacle detection using optical flow models, Intelligent Vehicles Symposium 2006, June 13–15, 2006, Tokyo, Japan pp. 466–471
- [2] Cuevas, E., Zaldivar, D., Rojas, R. (2005), Kalman filter for vision tracking. Measurement, (August), 1–18.
- [3] N. Gordon, D. Salmond, Bayesian State Estimation for Tracking and Guidance Using the Bootstrap Filter, Journal of Guidance, Control and Dynamics 18 (6) (1995), pp. 1434–1443.
- [4] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [5] Horst 1994 Horst, R. V.; Hogema, J. Times to collision and collision avoidance systems. Proceedings of the 6th ICTCT Workshop. Austrian Road Safety Board, Salzburg, 1993.
- [6] B.K.P. Horn and B.G. Schunck, Determining optical flow. Artificial Intelligence, vol 17, pp. 185–203, 1981.
- [7] M. Isard, A. Blake, Contour Tracking by Stochastic Propagation of Conditional Density, European Conference on Computer Vision (1996) pp. 343–356.
- [8] Boyoon Jung and Gaurav S. Sukhatme, Detecting Moving Objects using a Single Camera on a Mobile Robot in an Outdoor Environment. Proceedings of the 8th Conference on Intelligent Autonomous Systems, pp. 980–987, Amsterdam, The Netherlands, March 10–13, 2004.
- [9] Kalman, Rudolph Emil, A New Approach to Linear Filtering and Prediction Problems, Transactions of the ASME–Journal of Basic Engineering, vol 82, No Series D, pages 35–45, 1960.
- [10] B. D. Lucas and T. Kanade (1981), An iterative image registration technique with an application to stereo vision. Proceedings of Imaging Understanding Workshop, pp. 121–130
- [11] Meyer, F.G. Time-to-collision from first order models of the motion field. IEEE Trans. on Robotics and Automation, 10:792–798. 1994.

- [12] Nummiaro, K., Koller-Meier, E., Gool, L.V.: An adaptive color-based particle filter. *Image and Vision Computing* 21 (2003) 99–110
- [13] Linda G. Shapiro, George C. Stockman, *Computer Vision*. Prentice Hall, February 2, 2001
- [14] Jianbo Shi and Carlo Tomasi. Good Features to Track. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.
- [15] C. Stauffer and W. E. L. Grimson, Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on.*, Vol 2, 1999, 246–252
- [16] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. *Carnegie Mellon University Technical Report CMU-CS-91-132*, April 1991
- [17] Emanuele Trucco, Alessandro Verri, *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998
- [18] Olivier Faugeras. *Three Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1996.
- [19] Palágyi Kálmán. *Képfeldolgozás haladóknak*. Online egyetemi jegyzet. 2011.
- [20] Burt, P. J. and Adelson, E. H., The Laplacian Pyramid as a Compact Code, *IEEE Transactions on Communications*, Vol 31, 1983, pp. 532–540
- [21] Bouguet, J. Y., *Pyramidal Implementation of the Lucas Kanade Feature Tracker*. Part of OpenCV Documentation.
- [22] K.W. Cheng, S.C. Chan, Fast block matching algorithms for motion estimation, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol 4, pp. 2311–2314, 1996
- [23] www.virt.vein.hu/czuni/szamitogepeslatas.html