

Sztereo kamerapár-alapú objektumfelismerő  
rendszer fejlesztése

Józsa Csongor

2008. december 11.

## **Kivonat**

Jelen dolgozat célja egy olyan binokuláris látórendszer megtervezésének és megvalósítási lehetőségeinek bemutatása, mely a megbízható objektumfelismeréshez kellően jó minőségű mélységi adatokat generál, szem előtt tartva a hordozhatósági szempontokat és a beágyazott rendszerek által állított sebesség- és tárkorlátokat. A kétkamerás sztereó látórendszerek működési elvének megértéséhez szükséges elméleti alapok - perspektív transzformáció, epipoláris geometria, megfeleltetési probléma - ismertetése után, egy Linux-alapú rendszer fejlesztése kerül bemutatásra: a teljes rendszerterv elkészítésétől kezdve, a kameravezérlő szoftver és a mélységi adatok generálásához szükséges algoritmusok implementálásáig és teszteléséig.

## **Abstract**

This paper presents the design and implementation of binocular computer vision systems capable of generating high quality depth-maps suitable for robust object recognition, regarding the question of portability and the processing capacity and space constraints of embedded systems. After giving an overview of the theory of stereo vision systems - including perspective transformation, epipolar geometry and correspondence problem - the development process of a Linux-based machine vision system is presented from the design of the whole system to the implementation and testing of the camera driver software and the algorithms needed to generate range values.

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>6</b>
<b>2. A sztereo látás alapú 3D rekonstrukció és objektumfelismerés elméletének áttekintése</b>	<b>8</b>
2.1. 3D rekonstrukció jelentősége, alkalmazási területei . . . . .	8
2.2. 3D rekonstrukciós mérési technológiák áttekintése . . . . .	8
2.2.1. Osztályozási szempontok . . . . .	8
2.2.2. Aktív rendszerek . . . . .	9
2.2.3. Passzív rendszerek . . . . .	9
2.3. Binokuláris látórendszerek . . . . .	11
2.3.1. A sztereo látás . . . . .	11
2.3.2. Pinhole kameramodell, kameramátrix . . . . .	12
2.3.3. Epipoláris geometriai alapfogalmak . . . . .	14
2.3.4. Trianguláció . . . . .	15
2.3.5. Megfeleltetési probléma . . . . .	16
2.3.6. Epipoláris kényszerek . . . . .	16
<b>3. Sztereo látórendszer tervezése</b>	<b>19</b>
3.1. 3D rekonstrukció és objektum felismerés folyamata . . . . .	19
3.1.1. A folyamat áttekintése, részfolyamatokra bontása . . . . .	19
3.1.2. Kalibráció . . . . .	20
3.1.3. Képek betöltése . . . . .	20
3.1.4. Torzításmentesítés . . . . .	20
3.1.5. Jellemző pontok kiválasztása . . . . .	20
3.1.6. Pontpárok keresése . . . . .	21
3.1.7. Mélységtérkép generálása . . . . .	21
3.1.8. Jellemzők keresése a mélységképben . . . . .	21

3.1.9.	Objektumok detektálása és osztályozása . . . . .	21
3.2.	Megvalósítandó szoftverkomponensek . . . . .	22
<b>4.</b>	<b>A sztereo látórendszer implementációja</b>	<b>24</b>
4.1.	Felhasznált hardverkomponensek . . . . .	24
4.1.1.	Bumblebee sztereo kamera . . . . .	24
4.1.2.	A kamera vezérlése . . . . .	24
4.2.	Felhasznált szoftverkomponensek . . . . .	25
4.2.1.	Általános megfontolások . . . . .	25
4.2.2.	A kamera vezérléséhez használt függvénykönyvtár . . . . .	26
4.2.3.	A kamera vezérléséhez szükséges kernelmodulok . . . . .	26
4.2.4.	Magasszintű képfeldolgozáshoz használt könyvtár . . . . .	27
4.3.	Elkészített szoftverkomponensek . . . . .	27
4.3.1.	Kameravezérlő . . . . .	27
4.3.2.	Kalibráció, torzításmentesítés . . . . .	34
4.3.3.	Jellemző képpontok keresése . . . . .	35
4.3.4.	Megfeleltetési probléma megoldása . . . . .	35
4.3.5.	Mélységi térkép generálása . . . . .	36
<b>5.</b>	<b>Tesztelési eredmények</b>	<b>37</b>
5.1.	I. teszt elrendezés . . . . .	38
5.2.	II. teszt elrendezés . . . . .	43
<b>6.</b>	<b>Értékelés, további fejlesztések</b>	<b>48</b>
<b>7.</b>	<b>Köszönetnyilvánítás</b>	<b>49</b>

# 1. Bevezetés

Az objektumfelismerő rendszerek célja annak megállapítása, hogy egy bemeneti adathalmazban szerepel-e az adott rendszer rendelkezésére álló adatbázisban található - valamilyen jellemző tulajdonságoknak megfelelő - objektum, és ha igen, osztályozni azt aszerint, hogy melyik objektumleírásnak felel meg leginkább. A használt objektumjellemzőket és a bemeneti adathalmazt tekintve többféle megközelítés létezik. Dolgozhatunk például közvetlenül az intenzitásképeken, vagy vizsgálhatjuk a jelenet 3D-s reprezentációját.

A 3D rekonstrukciónak és a kapott távolságadatokból történő objektumfelismerésnek számos szerteágazó alkalmazási területe van és ennek megfelelően a különféle rendszerek esetében igen változatos alapelvekkel és implementációs megoldásokkal találkozhatunk. Példaként említhetőek a közvetlen távolságmérésen alapuló SONAR, illetve LIDAR rendszerek, a motion stereo elvű, illetve optical flow rendszerek, valamint a kétkamerás látórendszerek.

A trianguláción alapuló binokuláris látórendszerek elmélete jól kidolgozott, azonban sűrű és pontos mélységi térképeket megbízhatóan generáló látórendszer megtervezése és implementálása korántsem triviális feladat, melynek legfőbb oka a megfeleltetési probléma - azaz a több nézetből készült képeken az összetartozó pixelek megtalálásának - bonyolultsága. A probléma megoldására többféle módszer létezik, maguk előnyeivel és hátrányaival, egy adott alkalmazáshoz legjobban idomuló eljárás kiválasztásakor figyelembe kell venni ezek komplexitását, az általuk generált mélységi adatok pontosságát és megbízhatóságát, a takart pixelek detektálhatóságát és a mélységtérkép sűrűségét.

A megfeleltetési problémát megoldó algoritmus keresési terét különféle kényszerekkel és feltevésekkel jelentős mértékben csökkenthetjük, azonban bizonyos esetekben ez természetesen csökkenti a rendszer általános célú hasz-

nálhatóságát, mégis szükség van ezekre, kiváltképp, ha real-time adatfeldolgozásról van szó. Ilyen korlátozások lehetnek például az epipoláris geometriából adódó kényszerek, vagy azon feltételezések miszerint a vizsgálandó felületek lokálisan folytonosak, illetve lokálisan párhuzamosak a képsíkkal.

Általános esetben a megfeleltetési probléma nem oldható meg teljesen, hiszen előfordulhatnak takart pixelek. Ezek detektálását is meg kell oldani és valamilyen interpolációt alkalmazni a hozzájuk tartozó távolságadatok meghatározására.

A pixelpárok ismeretében az adott pixelhez tartozó mélységértékek triangulációval határozhatóak meg, azaz a kamerák által végzett vetítési transzformációt ismerve előállíthatjuk annak inverzét, melyet a pontpárra alkalmazva két egyenest kapunk, az eredeti pont pedig ezen egyenesek metszéspontjában található. A vetítési transzformációk meghatározása a kamerakalibrációs folyamat során történik. Ez általában ismert elrendezésű markerekkel ellátott jelenetekről készült képek segítségével történik.

A következőkben, a szükséges elméleti alapok áttekintése után egy, a fent felvázolt folyamatot megvalósító rendszer megtervezése, implementálása, tesztelése és az elkészült rendszer értékelése kerül ismertetésre.

## **2. A sztereo látás alapú 3D rekonstrukció és objektumfelismerés elméletének áttekintése**

### **2.1. 3D rekonstrukció jelentősége, alkalmazási területei**

A gépi látás egyik alapvető, mai napig is dinamikusan fejlődő területe a 3D rekonstrukció, azaz egy valós világbeli objektumról egy lehetőség szerinti minél pontosabb modell elkészítése. Erre igen sok területen mutatkozik igény, példaként említhetők - a teljesség igénye nélkül - orvosi, katonai, minőség-biztosítási, járműipari, robotikai, szórakoztatóipari, régészeti alkalmazások.

### **2.2. 3D rekonstrukciós mérési technológiák áttekintése**

#### **2.2.1. Osztályozási szempontok**

Egy 3D látórendszer fejlesztése során alapvető kérdés a térbeli modellalkotás alapjául szolgáló távolságadatok gyűjtéséhez használt mérési technika. A különböző alkalmazási területek különböző követelményeket állítanak például a hatótávolság, vagy a mérési pontosság tekintetében a 3D rekonstrukciós rendszerekkel szemben, így ebben a kérdésben igen sok eltérő megközelítéssel találkozhatunk. A következőkben ezekről a mérési technológiákról láthatunk egy rövid áttekintést.

A 3D látórendszerek tekintetében az egyik legalapvetőbb osztályozási szempont az, hogy az ún. aktív, avagy a passzív rendszerek közé sorolhatjuk a vizsgált rendszert. Aktív rendszerről akkor beszélhetünk, ha a mérések során kibocsájtunk valamilyen jelet és ennek viselkedését vizsgáljuk. Ha nincs ilyen aktív jeladó, akkor passzív rendszerről van szó.

### **2.2.2. Aktív rendszerek**

Az aktív rendszereken belül különbséget tehetünk direkt és indirekt rendszerek között. A direkt rendszerek közé tartoznak például az ún. time-of-flight (TOF) elvű ultrahangos, vagy lézeres rendszerek. Működési alapelvük, hogy a jeladó és a szenzor elrendezésének, illetve a jel adott közegbeli haladási sebességének ismeretében az objektum egy pontjáról visszaverődő jelnek a kibocsátási és a szenzorba való beérkezési idejének különbségéből a vizsgált pont távolsága közvetlenül meghatározható. Lézeres TOF rendszerekkel igen nagy távolságokat is viszonylag pontosan lehet mérni, illetve alkalmasak kisebb távolságok egészen precíz meghatározására.

Indirekt rendszerek esetében nem közvetlen távolságmérés történik, az alapelv inkább a későbbiekben ismertetésre kerülő kétkamerás rendszerekéhez hasonlít, azzal a különbséggel, hogy az egyik szenzort egy jeladó - általában lézer - helyettesíti. A távolságmeghatározás triangulációval történik, azaz a lézer ismert kibocsátási pontjából és irányából, illetve a visszaverődő lézernyaláb beesési helyének viszonyából meghatározható a felület azon pontjának térbeli helye, ahonnan a lézer visszaverődött. Sokszor nem is csupán egyetlen pontot vizsgálunk egyszerre, tehát pontszerű fényforrás helyett bonyolultabb strukturált fény kerül alkalmazásra, például egy, vagy több fénykést használunk, esetleg más mintát, melyekkel egyetlen pont helyett egy egész térrész letapogatható egy méréssel. Természetesen az adatok kinyerése ekkor több számítást igényel, illetve felléphetnek megfeleltetési problémák is.[1]

### **2.2.3. Passzív rendszerek**

Passzív rendszerek esetén aktív jeladó nélkül dolgozunk, általában intenzitásképekből kísérünk meg térbeli információt kinyerni. Ezen rendszerek lehetnek monokuláris, illetve binokuláris, vagy esetleg még több nézőpontot

használó rendszerek.

A monokuláris rendszerek közé tartoznak a shape-from-shading (SFS) rendszerek, melyek az árnyalást használják 3D információ előállítására. Általában egyetlen, ismert megvilágítású, fekete-fehér intenzitásképen vizsgálják az egyes pontok szürkeségi szintjét és a Lambert-féle visszaverődési törvényt alkalmazva következtetnek a képen szereplő objektumok formájára. [2] Érdemes megjegyezni, hogy a shape-from-shading az emberi térérzékelésben is jelentős szerepet játszik, azonban működési módja jelentősen eltér a gépi megvalósításokétól.[3]

Egy másik módszer a shape-from-shadow, mely az objektumon keletkező árnyékokat használja. Ezen módszert követő eljárások során több kép készül az adott objektumról más-más szögből érkező megvilágítással. Az objektum felületén megjelenő árnyékokból és a megvilágító fénysugár beesési szögéből különféle következtetéseket lehet levonni az objektum formájára vonatkozóan.[4]

Egy érdekes átmenetet képeznek a monokuláris és a binokuláris rendszerek között a mozgás alapú sztereó módszerek, melyeknél egy kamerát használunk, azonban több képet készítünk vele úgy, hogy vagy a kamerát mozdtítjuk el két kép készítése között, vagy pedig a vizsgált tárgy végez valamilyen mozgást.

A binokuláris rendszerek esetén két kamerával készítünk felvételeket, különböző nézőpontból. Talán ez a technika a legtermészetesebb megközelítése a 3D rekonstrukciónak, hiszen az emberi látórendszer is nagymértékben kihasználja. A binokuláris rendszerek működési elvét a következő fejezet részletesen ismerteti.

## 2.3. Binokuláris látórendszerek

### 2.3.1. A sztereo látás

A sztereopszisnak, vagy sztereo látásnak hívjuk azt a folyamatot, melynek segítségével egy háromdimenziós jelenetről két eltérő nézőpontból készített vetület alapján valamilyen mélységinformációt nyerhetünk ki. A jelenség tanulmányozása a kezdetekben természetesen az emberi látórendszerhez kapcsolódott. Már Leonardo da Vinci is említi *Trattato della Pittura* című munkájában, hogy

"...egy festmény, készüljön bár legtökéletesebb műgonddal és pontossággal, tekintettel a kontúrjaira, árnyaira és fényeire, nem keltethet azonos hatást a valós tárgyakkal, kivéve, ha azokat nagy távolságról és egy szemmel nézzük."

A sztereo látás első leírását Charles Wheatstone adta, az ő 1838-as munkája [5] előtt az általánosan elfogadott elmélet az volt, hogy egy tárgy csak akkor érzékelhető helyesen, ha annak minden pontjából érkező fénysugár a két retina azonos pontjára esik. Wheatstone megmutatta, hogy közeli objektumok esetén jelentős eltérés van a két szem által látott kép között és részben ez kelti a térérzetet az emberben. Ő találta fel és alkotta meg az első sztereoszkópot, mellyel két darab kétdimenziós képpel (melyek egyazon objektumról készültek, de más nézőpontból) mélységérzetet tudott keltetni a szemlélőben, ezzel nagyon szemléletes és látványos módon alátámasztva elméletét.

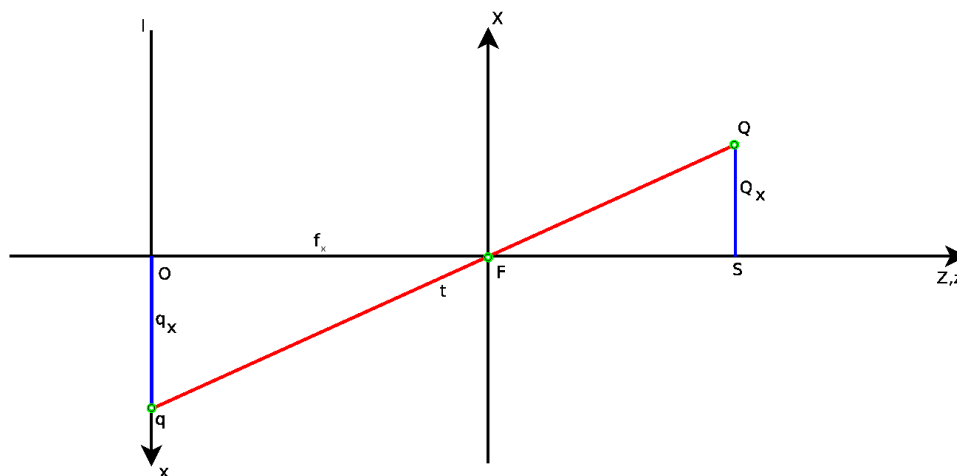
A következő lépés az emberi sztereo látás működésének megértése felé a Julesz-féle random dot stereogram[6], mely tulajdonképpen két véletlenszerűen előállított fekete-fehér ponthalmaz, egy apró változtatással, mely kis változás sztereoszkópon keresztül vizsgálva az emberben mélységbeli különségérzetet kelt. Ez azt bizonyítja, hogy az emberi sztereo-alapú térérzékelés

a tárgyak felismerése előtt történik, lehetséges tehát mélységkép generálása előzetes objektumfelismerés nélkül is.

### 2.3.2. Pinhole kameramodell, kameramátrix

A sztereo látás működésének teljes megértéséhez szükséges annak ismerete, hogy egy szenzoron - legyen az az emberi szem, vagy egy CCD érzékelő - pontosan milyen kép keletkezik, azaz hogyan lehet matematikai módszerekkel leírni a térbeli objektumok síkba vetítését. A következőkben a pinhole kameramodell működésével és matematikai modellezésével foglalkozunk.

A kamerák által végzett transzformáció egy széles körben használható modellje az ideális pinhole kameramodell. Ennek használata során a kameraapertúrát pontszerűnek tételezzük fel. Az ezen modellnek megfelelő kamera működését az 1. ábra szemlélteti.



1. ábra. A pinhole kameramodell

Egy tetszőleges  $Q$  pontnak a képsíkra vett vetülete azon  $q$  pont, mely a  $Q$  ponton és az  $F$  fókuszponton áthaladó  $t$  vetítési egyenes és az  $I$  képsík metszeténél található. Ha tekintjük az  $OqF$  és a  $FSQ$  hasonló háromszögeket,

akkor adódik, hogy:

$$\frac{q_x}{f_x} = \frac{Q_x}{Q_z}$$

azaz

$$q_x = f_x \cdot \frac{Q_x}{Q_z}$$

Hasonló összefüggés adódik az  $y$ -irányú koordinátákra:

$$q_y = f_y \cdot \frac{Q_y}{Q_z}$$

E két összefüggés tömörebben, a vizsgált pontok homogén koordinátás reprezentációjával fölírva:[14]

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} Q_x \\ Q_y \\ Q_z \\ 1 \end{bmatrix}$$

Ahol a homogén osztást elvégezve megkapjuk a képsíkbeli koordinátákat.

$$\begin{bmatrix} q_x \\ q_y \\ 1 \end{bmatrix} = \begin{bmatrix} U/W \\ V/W \\ W/W \end{bmatrix}$$

A fent ismertetett transzformációt leíró mátrixot hívjuk kameramátrixnak, mely kizárólag a kamerára jellemző - intrinsic - paramétereket tartalmaz. Ez kiegészülhet még egy-egy konstans eltolásértékkel  $x$  és  $y$  tengelyek mentén, amennyiben a képsík és az optikai tengely metszéspontja nem esik egybe a képközéppel. Ebben az esetben a kameramátrix alakja:

$$C = \begin{bmatrix} f_x & 0 & 0 & c_x \\ 0 & f_y & 0 & c_y \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Az intrinsic paramétereken kívül foglalkoznunk kell még az ún. extrinsic paraméterekkel, melyek nem kizárólag a kamera jellemzői, hanem függenek a kamera pillanatnyi elhelyezkedésétől, illetve a koordinátarendszerünk megválasztásától. A teljes transzformáció tehát leírható a kameramátrix és egy, a kamera helyzetét leíró pozíció-orientáció mátrix szorzatával:

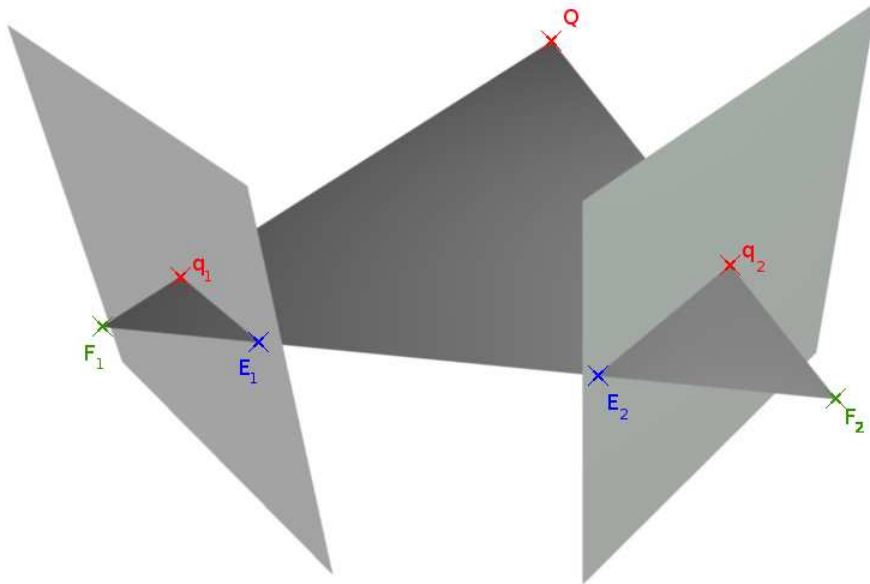
$$P = C \cdot D = \begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A következőkben feltesszük, hogy két olyan kamerával dolgozunk, melyek a fentiekben leírt kameramodellnek megfelelően működnek.

### 2.3.3. Epipoláris geometriai alapfogalmak

Egy objektum két különböző nézőpontból készült vetülete és az eredeti objektum kapcsolata az epipoláris geometria segítségével írható le. A 2. ábra szemlélteti egy  $Q$  pont vetítését két különböző képsíkra a fent ismertett pinhole kameramodell használva. A vetületek a  $q_1$ -el és  $q_2$ -vel jelzett pontok. A kamerák fókuszpontjait  $F_1$  és  $F_2$  jelöli. A két kamera fókuszpontját összekötő egyenes az alapvonal. Az alapvonal és a képsíkok metszéspontjait epipólusoknak nevezzük. Ezek az ábrán  $E_1$ -el és  $E_2$ -vel vannak jelölve. A bal oldali kamera epipólusa úgy is definiálható, mint a jobb oldali kamera fókuszpontjának a bal oldali kamera képsíkjára vett vetülete, és ugyanígy fordítva: a jobb oldali kamera epipólusa nem más, mint a bal oldali kamera

fókuszpontjának vetülete a jobb oldali kamera képsíkján. Az  $F_1$ ,  $F_2$  és  $Q$  pont által meghatározott síkot epipoláris síknak nevezzük.[7][9]



2. ábra. Epipoláris geometria

#### 2.3.4. Trianguláció

A 2. ábrán látható elrendezésből jól látható, hogy ha ismerjük egy tetszőleges  $Q$  pont két vetületét  $q_1$ -et és  $q_2$ -t, akkor az  $F_1$  és  $F_2$  fókuszpontok ismeretében egyértelműen meghatározható  $Q$  térbeli helyzete, hiszen rajta kell lennie mind a  $q_1F_1$ , mind a  $q_2F_2$  egyeneseken.  $Q$  ilyen módon történő meghatározását hívják triangulációnak.

### 2.3.5. Megfeleltetési probléma

Az előzőekben láthattuk, hogy egy  $Q$  pont  $q_1$  és  $q_2$  vetületének ismeretében  $Q$  viszonylag egyszerűen meghatározható, a binokuláris gépi látórendszerek megvalósításának legnehezebb problémája azonban a megfelelő  $q_1$  és  $q_2$  párok meghatározása. Ezen megfeleltetési probléma megoldására számos módszer létezik[10], melyek eltérő komplexitással rendelkeznek, illetve a kapott diszparitás-térkép sűrűsége is különböző.

A legegyszerűbb módszerek a valamilyen mintaillesztéses eljárásokon alapulnak. Lehet ez például élpixelek, élszegmensek, vagy görbék illesztése. Ezen módszerek viszonylag kevés adattal dolgoznak ezért futási idejük és tárigényük nem túl nagy. A segítségükkel kapott eredmények általában elég pontosak, de a kapott diszparitás térkép ritka.

Léteznek még terület-alapú módszerek, ezek lényege, hogy a képek szeleteinek intenzitásértékei közötti korrelációt vizsgálják. Ezek a módszerek gazdagon textúrázott képeken jól működnek, azonban jelentős korlátozó tényezőjük lehet, hogy feltételezik, hogy a vizsgált felületek az adott képsíkkal lokálisan közel párhuzamosak.

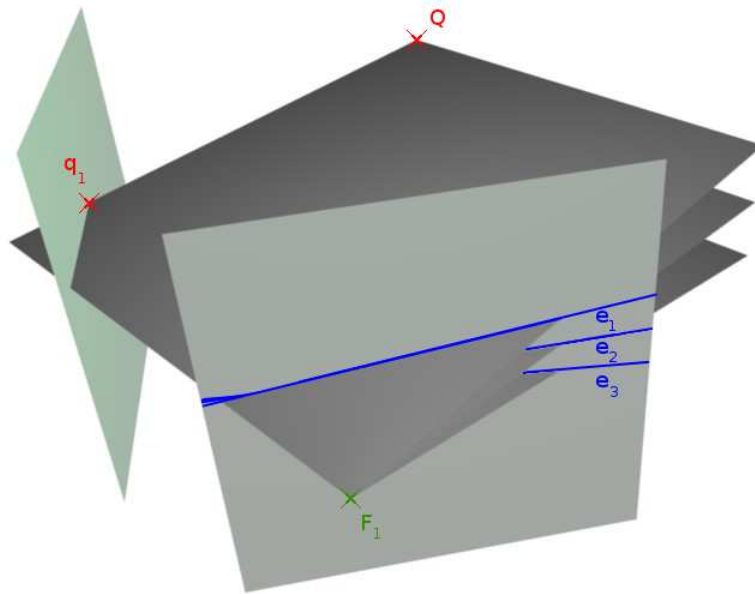
A fentiek egy újabb alternatíváját jelentik a fázis-alapú módszerek, melyek a két kamera képének Fourier-transzformáltjait vizsgálják.

Megemlítendőek még az energia-alapú módszerek, melyek a megfeleltetési problémát egy minimalizációs és regularizációs feladatként oldják meg, majd valamilyen iterációs módszerrel adnak becslést a mélységértékekre.

### 2.3.6. Epipoláris kényszerek

A 3. ábra azt mutatja, hogyan használhatóak az epipoláris geometriából adódó kényszerek a 3D rekonstrukció során felmerülő párosítási probléma (azaz egy valamilyen képsíkon adott pont 3D térbeli megfelelőjének másik

képsíkra vett vetületének megtalálásának problémája) keresési terének szűkítésére. Ha adott a térben egy tetszőleges  $Q$  pont, melynek a bal oldali képsíkra vett  $q_1$  vetülete ismert, akkor a perspektív transzformáció működéséből adódóan bizonyos, hogy az eredeti  $Q$  pont a bal oldali kamera  $F_1$  fókuszpontja és  $q_1$  által meghatározott egyenesen fekszik. Ez az egyenes meghatározható az eredeti  $Q$  ismerete nélkül, mivel csak a kameratranszformáció inverzét kell  $q_1$ -re alkalmaznunk. A  $q_1Q$  egyenest a bal oldali képsíkra vetítve pedig megkapjuk  $q_1$  lehetséges megfelelőinek halmazát. Ezt az egyenest nevezzük epipoláris egyenesnek. Az elrendezésből adódóan az epipoláris egyenesek mindig átmennek az azonos képsíkon levő epipóluson.[8]



3. ábra. Epipoláris kényszerek

A fent leírt, az egyik képsíkon adott pont és a hozzá tartozó másik képsíkbeli epipoláris egyenes közti összefüggést algebrai alakban a

$$p_2^T \cdot F \cdot p_1 = 0$$

egyenlet írja le, ahol  $F$  az ún. fundamentális mátrix.

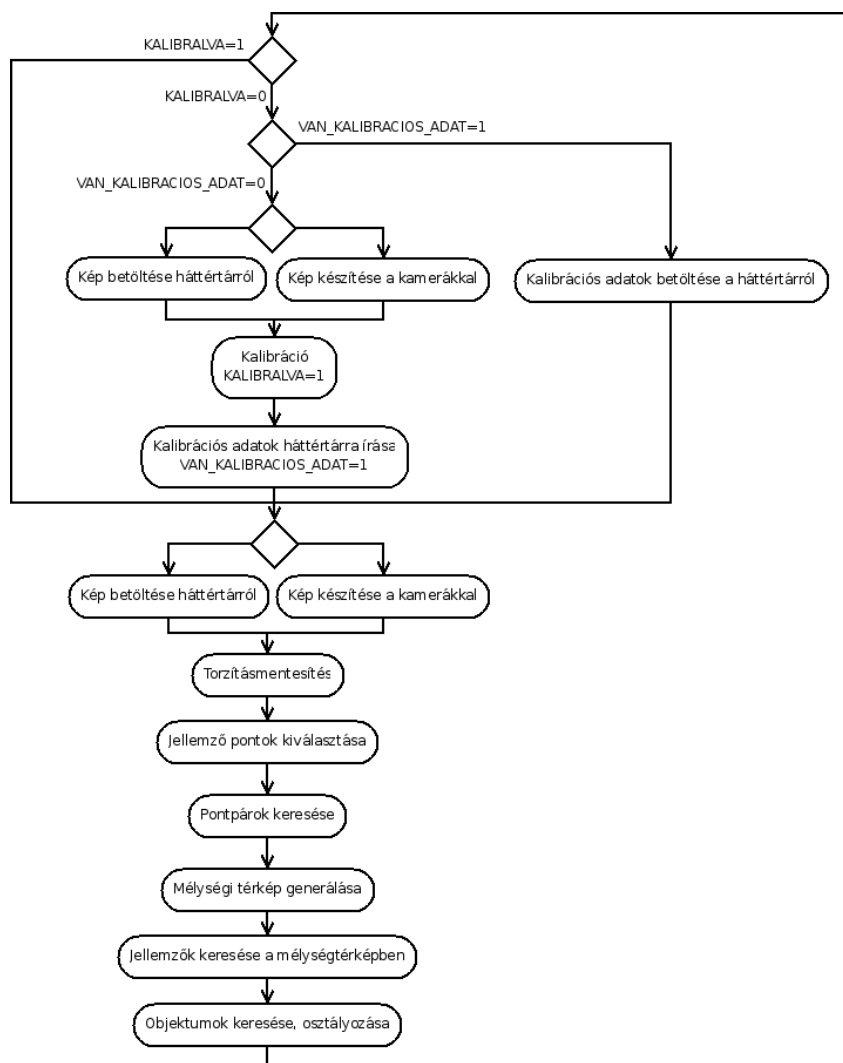
Megjegyzendő, hogy ha a két kamera képsíkja közös, akkor az epipoláris egyenesek párhuzamosak, ezért a keresés azok mentén egyszerűbbé válik. Amennyiben ez nem kivitelezhető, egy lineáris transzformáció segítségével akkor is elérhető, hogy párhuzamos epipoláris egyeneseket kapjunk. Ezt a műveletet rektifikációnak hívják.

### 3. Sztereo látórendszer tervezése

#### 3.1. 3D rekonstrukció és objektum felismerés folyamata

##### 3.1.1. A folyamat áttekintése, részfolyamatokra bontása

A rendszer által elvégzendő komplex műveletet kisebb részfolyamatokra bontva, a rendszer a 4. ábrán látható séma szerint működik.



4. ábra. A rendszer folyamatábrája

### **3.1.2. Kalibráció**

Az általunk használt 3D rekonstrukciós módszer használatához mindenképp szükséges a képek torzításmentessége, illetve a fundamentális mátrix ismerete. A torzításmentesítéshez meg kell határozni a lencsetorzítási együtthatókat. Ezen információkhoz a kalibrációs folyamat során tudunk hozzájutni, vagyis a képek tényleges vizsgálata előtt a kamerákat kalibrálni kell, vagy pedig egy korábbi kalibráció eredményeit kell betölteni (ezt legtöbb esetben megtehetjük, hiszen a kamerák közös házban vannak, így egymáshoz viszonyított geometriájuk nem változik és a lencsetorzítási paraméterek megváltozásához is általában igen extrém változások szükségesek a környezeti körülményekben.)

### **3.1.3. Képek betöltése**

A feldolgozandó képek betöltése jelentheti pillanatkép készítését a kamerával, de a fejlesztés és tesztelés során hasznos az is, ha lehetőségünk van ezeket a képeket háttértárra írni, majd onnan betölteni.

### **3.1.4. Torzításmentesítés**

A betöltött képeken egy torzításmentesítési lépést kell végrehajtani a 3D információ kinyerése előtt. Ennek során a kalibráció folyamán meghatározott lencsetorzítási paramétereknek megfelelően a képről valamilyen interpolációs eljárás segítségével megpróbáljuk eltávolítani a lencsetorzítás hatásait.

### **3.1.5. Jellemző pontok kiválasztása**

Ezen lépés feladata kiválasztani a megfeleltetési problémát megoldó algoritmus bemenetére adott pontokat. A megfeleltetési probléma bizonyos pontokra nagyon nehezen, míg másokra könnyebben megoldható. Ez természetesen

tesen az adott esetben használt megfeleltetési algoritmus típusától is függ, de tipikusan az előbbi csoportba tartoznak a homogén felületeken elhelyezkedő, míg utóbbiba a sarok-, él-, vagy gazdag textúrázottságú felületeken elhelyezkedő pontok.

### **3.1.6. Pontpárok keresése**

Ebben a lépésben a kiválasztott pontokhoz a rendszer megkísérli meghatározni a másik képsíkon lévő megfelelőjüket. Ehhez kihasználja a fentebb ismertetett epipoláris kényszereket, így kétdimenziós keresési tér helyett egy egyenes mentén dolgozhatunk. Megjegyzendő, hogy ez elé még beiktatható egy rektifikációs lépés a képek párhuzamossá tételére és ezáltal a keresési algoritmus egyszerűsítésére.

### **3.1.7. Mélységtérkép generálása**

Ezen részfolyamat bemenete a megfeleltetési probléma megoldásaként kapott pontpárok halmaza, kimenete a pontpárokhoz rendelt mélységértékek halmaza. A mélységértékek meghatározása a fentebb ismertetett triangulációs eljárással történik.

### **3.1.8. Jellemzők keresése a mélységképben**

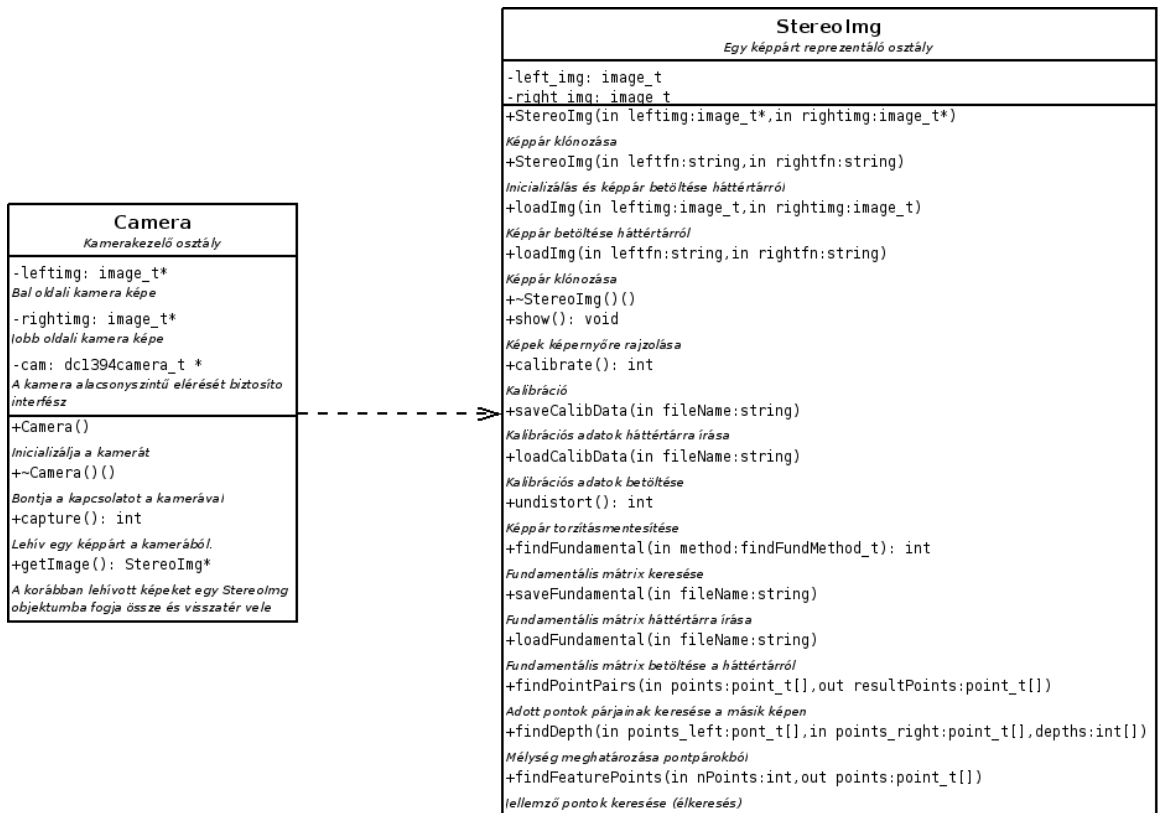
A kapott mélységképben a rendszer olyan skálázás, illetve nézet invariáns jellemzőket keres, melyek jól leírják a keresendő objektumot, azaz a mélységi térképen detektálható és osztályozható általuk.

### **3.1.9. Objektumok detektálása és osztályozása**

Az előző lépésben meghatározott jellemzőket a rendszer összeveti egy előre ismert adatbázissal és ez alapján megkísérli osztályozni az objektumokat.

### 3.2. Megvalósítandó szoftverkomponensek

Az elkészítendő rendszer két fő komponensre bontható: a kamera kezelését, illetve a sztereo képpárokon végzett műveleteket megvalósító osztály. Ezek alapján a rendszer osztálydiagramja:



5. ábra. A rendszer osztálydiagramja

A *Camera* osztály implementálja azon eljárásokat, melyek segítségével a kamera inicializálható, konfigurálható és kinyerhető belőle egy sztereo képpár.

A *StereoImg* osztály példányait a *Camera* osztály egy példánya hozza létre, ez a komponens a sztereo képpár manipulációiért felelős, mint például a kalibrációs adatok kinyerése, torzításmentesítés, jellemző pontok keresése, megfeleltetési probléma megoldása, mélységtérkép generálása.

## 4. A sztereo látórendszer implementációja

### 4.1. Felhasznált hardverkomponensek

#### 4.1.1. Bumblebee sztereo kamera

Sztereo képpárok készítésére a Point Grey Research Inc. Bumblebee modelljét használtam (6. ábra).[11] Ez egy kétlencsés kamerarendszer, mellyel másodpercenként 30 színes, 8 bites, 640x480 pixel felbontású sztereo kép-pár készíthető. A készülék IEEE 1394 csatlakozóval rendelkezik, a két szenzor egyetlen virtuális eszközként működik, a szinkronizálást a kameravezérlő elektronikája automatikusan elvégzi, azaz a két kameráról garantáltan egy időben készült képet kapunk.



6. ábra. Bumblebee sztereo kamera

#### 4.1.2. A kamera vezérlése

A kamerával történő kommunikáció az IIDC 1394 szabványban foglaltak szerint történhet. Ez egy ipari szabvány, melyet a 1394 Trade Association adott ki és az IEEE 1394-et használó digitális kamerák és PC-k közti adatátvitel és vezérlés szabványos módját hivatott meghatározni.[13] Többek között definiálja a használható képformátumokat, a kamerák működési módjait, az

adatátviteli paramétereiket, illetve az ezekhez tartozó kontrollregiszterek egy halmazát.

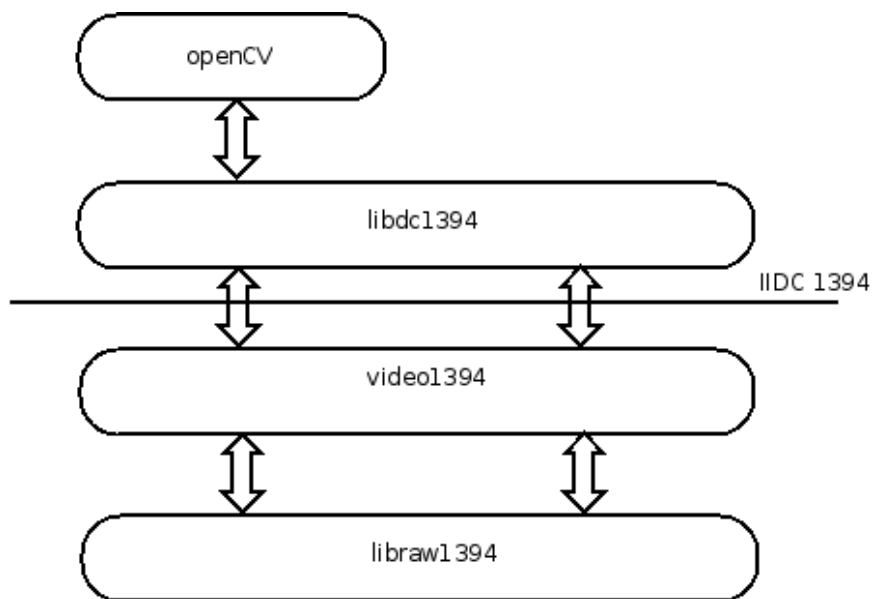
## 4.2. Felhasznált szoftverkomponensek

### 4.2.1. Általános megfontolások

Az elkészítendő látórendszer fejlesztése PC-n történik, azonban az elkészült rendszernek maximális mértékben hordozhatónak kell lennie, hogy a későbbiekben beágyazott platformon, mikrokontrolleres környezetben is használni lehessen, ezért a lehetőség szerinti minél szélesebb körű hardvertámogatottság az egyik elsődleges szempont volt a felhasználható szoftverelemek kiválasztásánál. Ezt szem előtt tartva a fejlesztést Linux operációs rendszeren végeztem.

A gyártó egy elsődlegesen Windows platformra írt keretrendszert mellékel a termékhez, melynek Linuxos verziója csak korlátozott mértékben használható, így a feladatot nehezíti, hogy egy a kamera kezelését megvalósító komponenst is el kellett készíteni, melynek során problémát jelentett, hogy a kameráról csupán minimális jellegű dokumentáció áll rendelkezésre. (Ennek oka egyrészt a gyártó saját keretrendszerének előtérbe helyezése lehet, másrészt, hogy a kamera már nincs a támogatott termékek között, a következő generációs Bumblebee2 megjelenése miatt.)

A 7. ábrán a rendszer fejlesztése során felhasznált könyvtárak áttekintő ábrája látható.



7. ábra. A felhasznált szoftver architektúra

#### 4.2.2. A kamera vezérléséhez használt függvénykönyvtár

A kamera kezeléséhez a libdc1394 függvénykönyvtárat használtam.[12] Ez a nyílt forráskódú könyvtár használható a legnépszerűbb operációs rendszereken, segítségével vezérelhetőek az IIDC szabványnak megfelelő digitális kamerák. Beállíthatóak a kívánt átviteli paraméterek, a használt képformátum, valamint lehetőséget nyújt a kamera regiszterszintű konfigurálására is.

#### 4.2.3. A kamera vezérléséhez szükséges kernelmodulok

A libdc1394 az alacsonyszintű IEEE 1394-en történő átvitelt megvalósító libraw1394 Linux kernelmodult használja, azonban amennyiben ki akarjuk használni a DMA átvitel előnyeit, a video1394 kernelmodulra is szükség van, mely magasabb szintű hozzáférést biztosít a firewire interfészekhez.

#### **4.2.4. Magasszintű képfeldolgozáshoz használt könyvtár**

A magasabb szintű képmanipulációs műveletek megkönnyítésére a képek kezeléséhez az Intel Open Source Computer Vision Library-t (openCV, libcv) használtam.[14] Ahogy a neve is mutatja ez egy nyílt forráskódú, szabadon felhasználható függvénykönyvtár. Támogatja a legtöbb Windows verziót és UNIX-jellegű operációs rendszert. Elsősorban gépi látással kapcsolatos és képfeldolgozási feladatokat ellátó rendszer fejlesztéséhez nyújt segítséget. Jelen projekt tekintetében külön hasznos, hogy az alapvető szűrőkön és algoritmusokon túl a háromdimenziós rekonstrukcióhoz és a kamerakalibrációhoz is implementál néhány széles körben használt eljárást.

### **4.3. Elkészített szoftverkomponensek**

#### **4.3.1. Kameravezérlő**

Mint korábban említésre került, a két kamerát a PC-n futó program egyetlen eszközként érzékeli. Ennek megfelelően a kamera egy speciális képformátumot használ az átvitel során. A 8. ábrán látható a kameráról érkező nyers adat RGB képformátumú képként értelmezve.



8. ábra. A kamerából kinyerhető nyers adat illusztrálása

A kép valójában 3 képet tartalmaz egymással átlapolva (ezt RGB formátumban értelmezve az első kép a vörös, a második a zöld, a harmadik a kék csatornán fog megjelenni). Ha a kamerák képeit külön szeretnénk feldolgozni, először is a három csatornát külön kell választanunk. Ennek eredményeként kapott képek láthatóak a 9., 10. és 11. ábrán.



9. ábra. A kamerából kinyerhető nyers adat első csatornája



10. ábra. A kamerából kinyerhető nyers adat második csatornája



11. ábra. A kamerából kinyerhető nyers adat harmadik csatornája

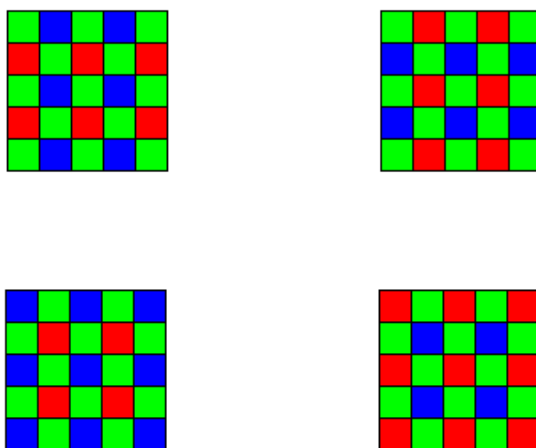
Jól látható, hogy az első kép a jobb, míg a második a bal oldali kamera képét tartalmazza. A harmadik csatorna elsőre véletlen zajnak tűnhet, azonban a kép bizonyos területein korrelál a bal oldali kamera képével. Lehetséges, hogy valamilyen térbeli információt hordoz, azonban ennek formátumáról nem áll rendelkezésre információ. Ha ez valóban egy mélységi térkép, akkor

sem lehet kielégítően finom felosztású, mivel a kép hisztogramját megvizsgálva látható, hogy az csupán 16 diszkrét értéket tartalmaz (12. ábra).



12. ábra. A kamerából kinyert adatfolyam harmadik csatornájának hisztogramja

Megfigyelve a két kamera képét látható, hogy a pixelek intenzitása egy jellegzetes ismétlődő mintázat szerint változik a képen, ugyanis a képek nem szürkeskálás, hanem színes képek, azonban nem RGB formátumban, hanem Bayer elrendezésben, vagyis közvetlenül a CCD szenzorok elrendezésének megfelelő formátumban kapjuk az intenzitásértékeket.



13. ábra. Bayer mintázatok

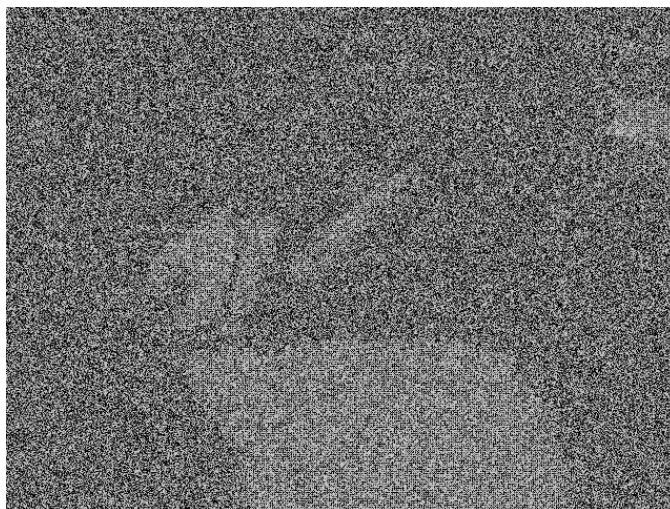
A 13. ábrán a Bayer elrendezés 4 variációja látható, ezek közül a Bumblebee a jobb felsőt, azaz a BG elrendezést használja. Az openCV számos képkonverziós eljárást megvalósít, ezek között szerepel egy Bayer-RGB interpolációs szűrő, melynek segítségével könnyebben kezelhető, hagyományos RGB formára hozhatjuk a képeket. A konverzió eredményét a 14., 15. és 16. ábrán láthatjuk.



14. ábra. A bal oldali kamera képe



15. ábra. A jobb oldali kamera képe



16. ábra. A kamerából kinyert adatfolyam harmadik csatornája

Láthatjuk tehát, hogy az első csatorna tartalmazza a jobb oldali, a második csatorna pedig a bal oldali szenzor által készített képet, a BG bayer mintázatnak megfelelő formátumban.

A fent ismertetett folyamat megtalálása, melynek segítségével a két szenzor képe kinyerhető a kamerából, dokumentáció hiányában nem volt magától értetődő feladat, a helyzetet nehezítette az is, hogy a fejlesztés során használt PC FireWire interfész kártyáján lévő két port közül az egyikben hibás adat érkezik, ez azonban csak az algoritmus elkészülte után vált egyértelművé. Egy ilyen hiba elrontja a Bayer mintázatot, így a képben csíkok jelennek meg (ez a nyers képen általában nem túl feltűnő, azonban a végső RGB formába konvertált kép nyilvánvalóan teljességgel használhatatlan lesz). Ez többször tévútra vezetett, mivel a hibát nem a hardverben, hanem a használt algoritmusban kerestem. A 17. ábrán látható egy hibás, átalakítás nélküli kép, melynek alsó harmadán jól láthatóak a fent említett zavaró csíkok.



17. ábra. Hibás kép, átalakítás nélkül

#### 4.3.2. Kalibráció, torzításmentesítés

Az intrinsic paraméterek, a fundamentális mátrix és a lencsetorzítási együtthatók meghatározása az openCV beépített kalibrációs függvényeinek segítségével történt. Ezek a függvények a fentebb ismertetett pinhole kameramoddellel dolgoznak, a nemlineáris lencsetorzítás esetében pedig az alábbi polinomos közelítést alkalmazzák:

$$x' = x \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4) + 2 \cdot p_1 \cdot x \cdot y + p_2 \cdot (r^2 + 2 \cdot x^2)$$

$$y' = y \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4) + p_1 \cdot (r^2 + 2 \cdot y^2) + 2 \cdot p_2 \cdot x \cdot y$$

ahol  $r^2 = x^2 + y^2$ ,  $k_1$  és  $k_2$  a radiális torzítási együtthatók,  $p_1$  és  $p_2$  pedig a tangenciális torzítási együtthatók.

A kalibrációhoz a rendszer sakktáblamintázatot tartalmazó képeket használ, melyeken az openCV automatikusan képes a sarokpontokat detektálni.

### 4.3.3. Jellemző képpontok keresése

A rendszer a megfeleltetési problémát megoldó függvény bemenetére adott ponthalmaz kiválasztására Sobel-élkeresést használ az alábbi kernellel:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

A konvolúció elvégzése után a ponthalmaz tényleges kiválasztása egy viszonylag alacsony értékű küszöbözéssel történik.

### 4.3.4. Megfeleltetési probléma megoldása

Az egyik képen adott pixelekhez tartozó másik képbeli pixelek meghatározása az epipoláris kényszerek alkalmazásával történik. Az epipoláris egyenes mentén való keresés implementálásakor kihasználtam, hogy a két kamera képsíkja koplanáris, az epipoláris egyenesek rektifikációs lépés alkalmazása nélkül is párhuzamosak egymással és a kép  $x$ -tengelyével.

Azt viszont figyelembe kellett venni, hogy a kalibrációs pontatlanságból adódóan a keresendő pont a gyakorlatban nem mindig a számolt epipoláris egyenesre esik, ezért a keresés nem csak az egyenes mentén hanem annak kis környezetében, egy sávban történik.

A megfeleltetéshez kezdetben csak a pontok intenzitásvektorait használtam, de ez meglehetősen rossz eredményt adott. Az intenzitásértékeket és

a kép megfelelő helyen vett differenciáltjának értékét együttesen használva sem kaptam sokkal meggyőzőbb eredményt. A megoldást a pontok környezetéből vett több pixelt tartalmazó minták összehasonlítása hozta. A két vizsgálandó pont körüli 30x30-as négyzetben az intenzitások összes négyzetes eltérését vizsgálva az előbbieknél sokkal jobb eredményre jutottam.

#### 4.3.5. Mélységi térkép generálása

A kapott pontpárok ismeretében a kalibráció során meghatározott kameratranszformációk segítségével elvégezhető a trianguláció, azaz meghatározható a ponthoz tartozó vetítési egyenes.

Az implementáció során a nehézséget az okozza, hogy nem elég az egyenesek analitikus metszéspontját meghatározni, mivel a kamerák véges felbontóképességéből, és a számítási pontatlanságokból eredően a legtöbb esetben a két egyenes kitérő. Éppen ezért a mélységi térkép generálásakor nem mindkét egyenes egyenletét kielégítő pontot keresi a rendszer, hanem a két egyenest összekötő legrövidebb szakasz felezőpontját.

A mélységi adatok egy szürkeskálás képben kerülnek reprezentálásra, ahol a pixelek értékei a bal oldali kamera által az adott pixelre vetített pont és a kamera síkjának távolságával arányosak.

## 5. Tesztelési eredmények

Az alábbiakban két szemléletes elrendezésről készített képpár és annak a fentiekben ismertetett folyamat különböző lépcsőfokain rögzített megfelelője található.

## 5.1. I. teszt elrendezés

A 18. és 19. ábrákon az első jelenetről készített bemeneti képpár látható.



18. ábra. I. teszt elrendezés, feldolgozatlan kép, jobb oldali kamera képe



19. ábra. I. teszt elrendezés, feldolgozatlan kép, bal oldali kamera képe

A 20. és 21. ábrákon az első jelenetről készített képpár látható a torzításmentesítés elvégzése után, a torzításmentesítés hatása ezeken a képen

elsősorban a sarkokban figyelhető meg, az interpoláció során a kép szélein egy vékony sáv elveszik.

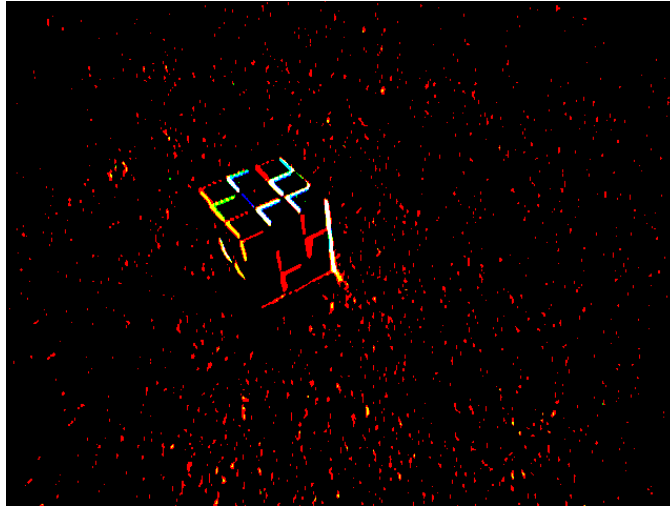


20. ábra. I. teszt elrendezés, torzításmentesített kép, jobb oldali kamera képe



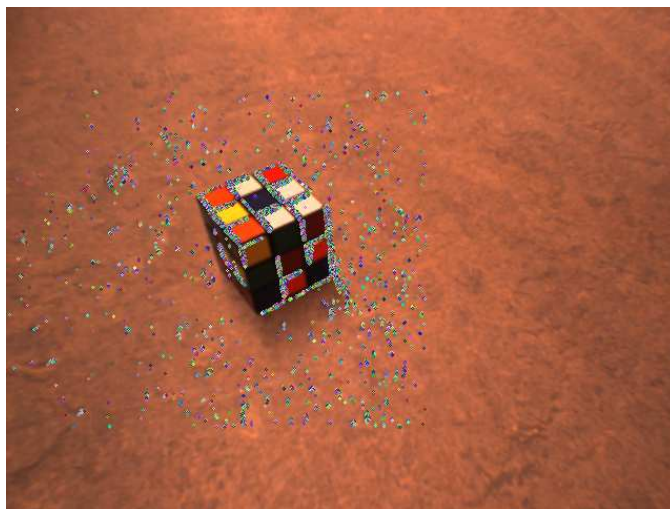
21. ábra. I. teszt elrendezés, torzításmentesített kép, bal oldali kamera képe

A 22. ábrán az első jelenetről a jobb oldali kamera által készített kép látható az élkiemelés elvégzése után.



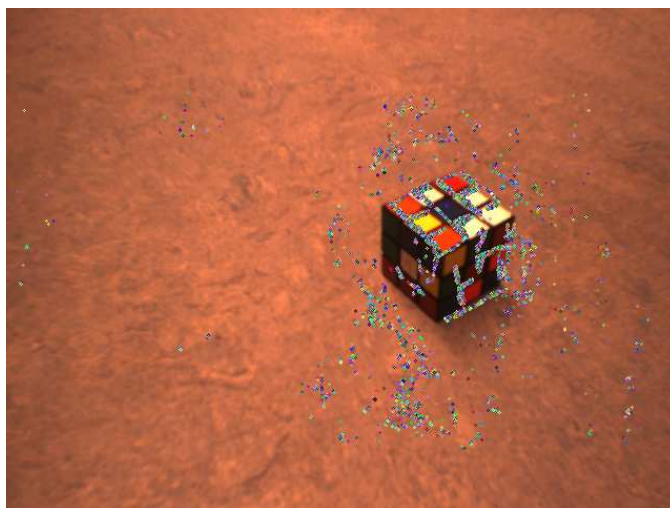
22. ábra. I. teszt elrendezés, élkeresés eredménye

A 23. ábrán az első jelenetről a jobb oldali kamera által készített kép látható, az élkiemelés és küszöbözés során a megfeleltetési problémát megoldó algoritmus bemeneti értékeinek kiválasztott pontok megjelölésével.



23. ábra. I. teszt elrendezés, megfeleltetési probléma bemenete, jobb oldali kamera képe

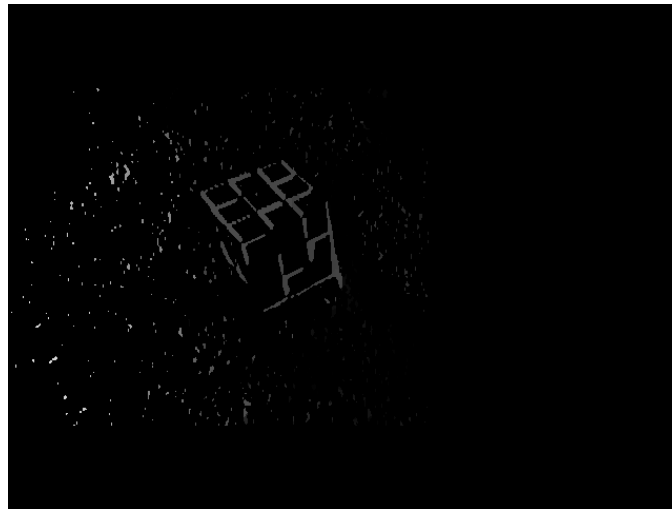
A 24. ábrán az első jelenetről a bal oldali kamera által készített kép látható a megfeleltetési probléma megoldásaként adódó pontok megjelölésével.



24. ábra. I. teszt elrendezés, megfeleltetési probléma megoldása, bal oldali kamera képe

A 25. ábrán a megfeleltetési probléma megoldásaként kapott pontpárok-

ból generált mélységi térkép látható. Látható, hogy a kocka pontjai nagyjából konzekvensek, de a mélységi felbontás nem elég jó ahhoz, hogy a felületet vissza lehessen állítani. Az is megfigyelhető, hogy a padló textúrájából adódóan sok nehezen azonosítható pont is a jellemző pontok közé lett osztályozva, ezekre a mélységi érték nem minden esetben helyes.



25. ábra. I. teszt elrendezés, mélységi térkép

## 5.2. II. teszt elrendezés

A 26. és 27. ábrákon a második jelenetről készített bemeneti képpár látható.



26. ábra. II. teszt elrendezés, feldolgozatlan kép, jobb oldali kamera képe



27. ábra. II. teszt elrendezés, feldolgozatlan kép, bal oldali kamera képe

A 28. és 29. ábrákon a második jelenetről készített képpár látható a torzításmentesítés elvégzése után, a torzításmentesítés hatása jól megfigyelhető

az egyenes vonalak görbületeinek eltűnésében.

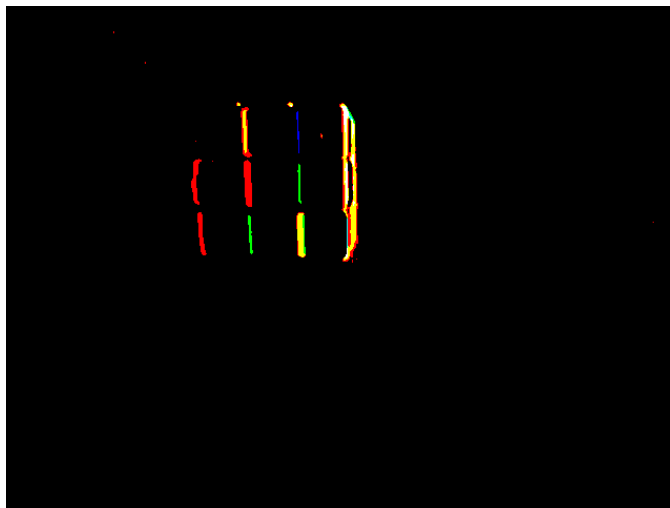


28. ábra. II. teszt elrendezés, torzításmentesített kép, jobb oldali kamera képe



29. ábra. II. teszt elrendezés, torzításmentesített kép, bal oldali kamera képe

A 30. ábrán a második jelenetről a jobb oldali kamera által készített kép látható az élkiemelés elvégzése után.



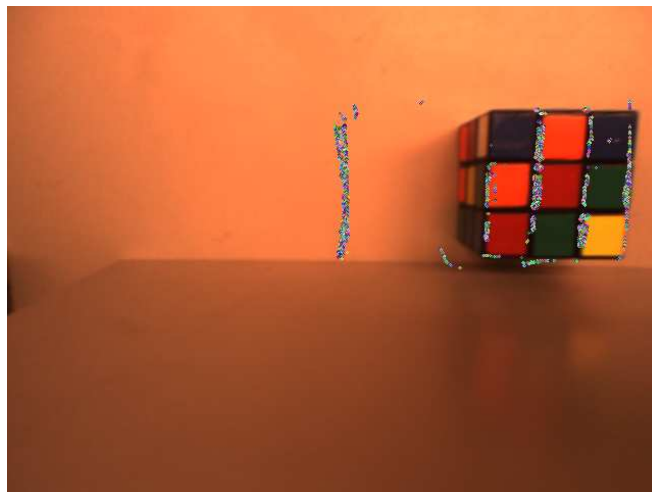
30. ábra. II. teszt elrendezés, élkeresés eredménye

A 31. ábrán a második jelenetről a jobb oldali kamera által készített kép látható, az élkiemelés és küszöbözés során a megfeleltetési problémát megoldó algoritmus bemeneti értékeinek kiválasztott pontok megjelölésével.



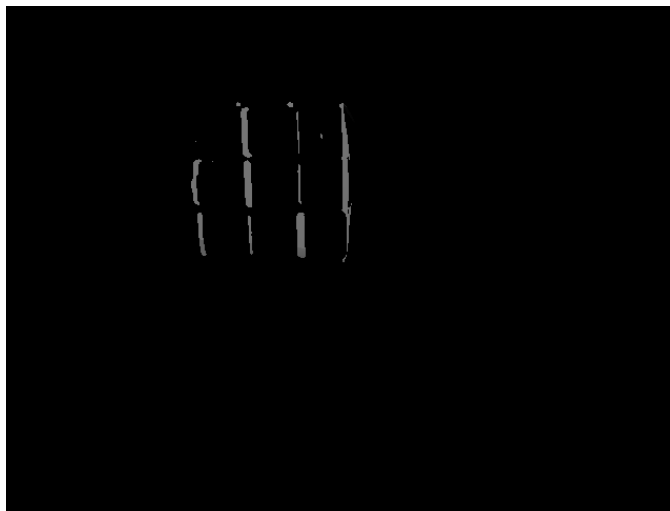
31. ábra. II. teszt elrendezés, megfeleltetési probléma bemenete, jobb oldali kamera képe

A 32. ábrán az első jelenetről a bal oldali kamera által készített kép látható a megfeleltetési probléma megoldásaiként adódó pontok megjelölésével. Jól látszik hogy azon pontokra, melyek nem látszanak mindkét kamera képén hibás megoldást kapunk.



32. ábra. II. teszt elrendezés, megfeleltetési probléma megoldása, bal oldali kamera képe

A 33. ábrán a megfeleltetési probléma megoldásaként kapott pontpárok-ból generált mélységi térkép látható. Megfigyelhető, hogy a kocka azon pontjaira, melyek mindkét kamera szemszögéből láthatóak nagyjából konzekvens értékeket kaptunk, azonban a takarásban lévő pontokat a rendszer jóval távolabbinak mérte, mint közvetlen szomszédjaikat.



33. ábra. II. teszt elrendezés, mélységi térkép

## 6. Értékelés, további fejlesztések

A teszteredményekből az a következtetés vonható le, hogy az elkészült rendszer a jó markerpontokkal rendelkező képeken megbízhatóan generál mélységi térképet, a legfőbb hiányossága a takarások kezelése (ld. II. teszteset). Ez viszonylag könnyen kiküszöbölhető a későbbiekben, például a megfeleltetési probléma kétirányú megoldásával (figyelembe kell azonban venni, hogy ez jelentős többletszámítással jár). Az is megállapítható, hogy a kapott mélységi térkép nem elég sűrű megbízható objektumfelismeréshez végrehajtásához, ezen valamilyen interpolációval, vagy esetleg a megfeleltetési probléma más megközelítésű megoldásával lehetne segíteni.

## 7. Köszönetnyilvánítás

Köszönet illeti Dr. Vajda Ferencet, aki témavezetőként segítette munkámat, Furmann Márton villamosmérnök hallgatót, a rendszer megtervezésében való részvételért, illetve a kalibráció implementálásáért, Helybély Ádámot a kamera-kezelő elkészítéséhez nyújtott tanácsokért, valamint a BME Irányítástechnika és Informatika Tanszékének 3D Érzékelés és Mobilrobotika Kutatócsoportjának munkatársait a laborban végzett munkában nyújtott segítségért.

## Hivatkozások

- [1] Lassó András: *Lézeres távolságmérés*
- [2] Emmanuel Prados, Olivier Faugeras: *Shape From Shading*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999
- [3] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, Mubarak Shah: *Shape from Shading: A Survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999
- [4] Yizhou Yu: *Shape from Shadow*
- [5] Charles Wheatstone: *Contributions to the Physiology of Vision.—Part the First. On some remarkable, and hitherto unobserved, Phenomena of Binocular Vision.*, 1838
- [6] Ralph M Siegel: *Choices: The Science of Bela Julesz*, PLoS Biology, 2004
- [7] Richard Hartley, Andrew Zisserman: *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004
- [8] Gang Xu, Zhengyou Zhang: *Epipolar Geometry in Stereo, Motion, and Object Recognition*, Kluwer Academic Publishers, 1996
- [9] Subhashis Banerjee: *Multiple Views Geometry*, Dept. Computer Science and Engineering, IIT Delhi, 2008
- [10] Luis Alvarez, Rachid Deriche, Javier Sánchez, Joachim Weickert: *Dense Disparity Map Estimation Respecting Image Discontinuities: A PDE and Scale-Space Based Approach*, Institut National de Recherche en Informatique et en Automatique, 2000

- [11] *Point Grey Research Inc. - Legacy Products*,  
<http://www.ptgrey.com/products/legacy.asp>
- [12] *libdc1394 Homepage*, <http://damien.douxchamps.net/ieee1394/libdc1394/>
- [13] *1394 Trade Association: Specifications*,  
<http://www.1394ta.org/developers/Specifications.html>
- [14] *OpenCV Reference Manuals*, <http://www.seas.upenn.edu/~bensapp/opencvdocs/>