

TDK-dolgozat

Klimek Gabriel

Óbudai Egyetem
Neumann János Informatikai Kar
Szoftvertchnológia Intézet



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

QR-KÓD DETEKTÁLÁS PÁRHUZAMOSOKKAL

Szerző(k): **Klimek Gabriel**
mérnök informatikus szak, V. évfolyam

Konzulens(ek): **Dr. Vámosy Zoltán**
egyetemi docens

Budapest, 2012.

TARTALOMJEGYZÉK

1	Bevezetés.....	2
2	A QR-kód	3
2.1	A QR-kód felépítése.....	3
2.2	QR-kód detektálási módszerek	4
3	Éldetektálás	6
3.1	A deriváltak alkalmazása éldetektálásra	6
3.2	A Canny-éldetektáló.....	8
4	Vonaldetektálás	9
4.1	A Hough-transzformáció.....	9
4.2	A PClines algoritmus	11
5	Párhuzamos egyenesek detektálása	15
5.1	Az élpontok detektálása és szűrése	15
5.2	A Hough-tér akkumulációja	16
5.3	A párhuzamos egyenesek modelljeinek felállítása	17
6	A QR-kód mintavételezése a detektált párhuzamosok segítségével	22
7	Implementáció	24
7.1	Implementációs megfontolások	24
7.2	Az algoritmus lépéseinek összefoglalása	25
8	Tesztelés.....	27
8.1	Mintacsomag és tesztelési terv	27
8.2	Az algoritmus eddigi eredményeinek ismertetése.....	29
9	Összefoglalás	31
10	Irodalomjegyzék	32

1 BEVEZETÉS

A vonalkód széles körben elterjedt adatrepresentációs technika, amely optikai leolvasásra lett kifejlesztve és általában információt hordoz arról az objektumról, amelyre valamilyen módon rögzítették. A már régóta használatban lévő egydimenziós vonalkódokhoz a későbbiekben, olyan kétdimenziós kódok csatlakoztak, amelyek célja egyrészt az adatkapacitás növelése, másrészt a gyorsabb és hibátűrőbb leolvasás volt. Az adatkapacitás növekedésével a tárolható adatok típusválasztéka is szélesedett.

A dolgozat célja egy párhuzamos egyenesek detektálásán alapuló algoritmus, a PClines [4] felhasználása QR-kódok detektálására, amelyek egyre nagyobb mértékben jelennek meg a mindennapokban. Ez olyan detektálási technikákat követel meg, amelyek változatos, kontrollálatlan körülmények között, nem ipari célra készült eszközökkel is eredményesen használhatóak. Manapság a digitális képfeldolgozás eszközeivel már szinte bármely, kamerával rendelkező eszköz vonalkód detektorrá alakítható. A választás az elterjedtsége miatt esett a QR-kódra, azonban a későbbiekben láthatóvá válik, hogy a módszer más mátrixkódok esetében is alkalmazható.

A QR-kód eredete, felépítése és a jelenleg detektálásra használt módszerek áttekintése a második fejezetben olvasható. A módszerek az általuk detektálásra használt alapvető technikák alapján kerültek csoportosításra. A harmadik fejezet az algoritmus első jelentős műveletét, az éldetektálást tárgyalja. Bemutatásra kerülnek a legelterjedtebb éldetektáló operátorok és az élpontok lényeges tulajdonságainak meghatározására szolgáló eljárások. A negyedik fejezet a vonaldetektálás egyes módszereit tárgyalja, bemutatva a módszer alapkonceptiójának számító Hough-transzformációt és a PClines algoritmus által alkalmazott párhuzamos koordináta-rendszerrel összefüggő elméletet. A párhuzamos egyenesek detektálásával az ötödik fejezet foglalkozik, amely ismerteti párhuzamos egyenesek modelljének generálását és az ezzel kapcsolatban fellépő problémákat. A hatodik fejezet tárgyalja a QR-kód kinyerését a meghatározott modell segítségével és normalizálásának legfontosabb aspektusait. Az implementációs megfontolásokat és az algoritmus lépéseinek összefoglalását a hetedik fejezet tartalmazza. A nyolcadik fejezet a teszteléssel foglalkozik. Ismerteti a tesztelésre vonatkozó terveket és összefoglalja az algoritmus más implementációi által elért eredményeket, összevetve azokat más rendszerekkel. A kilencedik fejezet néhány záró gondolatot tartalmaz az algoritmusról és a kutatással kapcsolatos jövőbeli tervekről.

2 A QR-KÓD

A QR-kódot¹ (QR Code²) [3] a Denso Wave Inc. fejlesztette ki 1994-ben a Toyota gyár járműveinek azonosítására és követésére a gyártási folyamat során. Fajtáját tekintve kétdimenziós, ún. mátrix-kódról van szó, amely megalkotásánál a vállalat célja az volt, hogy egy gyorsan leolvasható, nagy adatkapacitással bíró, a legtöbb transzformációra invariáns és hibatűrő vonalkód típust készítsenek. A megalkotást követő években a QR-kód fokozatosan szabvánnyá vált Japánban, majd más országokban is. 2000-től az ISO/EIC 18004:2000 szabványként volt elérhető, majd ezt 2006-ban felváltotta az ISO/EIC 18004:2006 szabvány, amely egy 2009-es módosítással jelenleg is érvényben van [12]. Eközben 2004-ben a QR-kód egy kisebb változata, a mikro QR-kód (Micro QR Code) is szabvánnyá vált [3]. Az utóbbi években a QR-kód elterjedt az iparon kívül is, és napjainkban sok esetben szolgál adatrepresentációs célokra, hirdetések kiegészítőjeként, a tömegközlekedésben, illetve más az ipartól független felhasználási területeken [19]. Ehhez vélhetően nagyban hozzájárult az is, hogy a Denso Wave Inc. nem számít fel licenrdíjat a vonalkód használatáért [3].



2.1 ábra: 1-es verziójú QR-kód

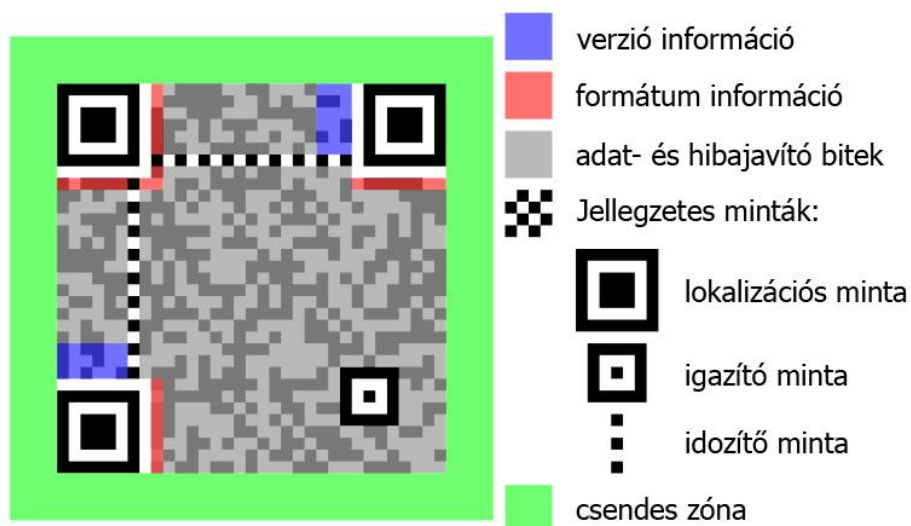
2.1 A QR-kód felépítése

A QR-kód legjellegzetesebb részei a sarkaiban található jellegzetes minták, amelyek a vonalkód lokalizálására, illetve pozíciójának megállapítására szolgálnak [12]. A mikro QR-kód a kompaktság érdekében csak egyetlen ilyen mintát tartalmaz a bal felső sarkában. A kódban további minták találhatóak, amelyek a leolvasás könnyítését szolgálják. Egyes minták megléte és mennyisége a QR-kód verziójától függ. A verzió határozza meg, hogy a QR-kód hány modulból épül fel, tehát mennyi adatot tartalmaz.

¹ Quick Response Code – jelentése: gyors válaszú kód

² A QR Code a Denso Wave Inc. regisztrált védjegye Japánban és más országokban.

A modul a QR-kód legkisebb egysége, amelyet egy fekete vagy fehér négyzet képvisel. A QR-kódok mérete 21×21 modultól (1-es verzió) egészen 177×177 modulig (40-es verzió) terjedhet. Adott azonban még egy paraméter, a hibajavító képesség, amelyet L, M, Q, H kategóriákba sorolnak, amelyek rendre 7, 15, 25, 30%-ban határozzák meg visszaállítható kódszavak mennyiségét, ahol egy kódszó 8 bites a használt hibajavító eljárás pedig a Reed-Solomon kódolás [12]. A verzióra és a formátumra vonatkozó információkat a 2.2 ábrán látható kék és piros részek tartalmazzák.



2.2 ábra: A QR-kód felépítése [19]

Az igazításhoz használatos minták az 1-es verzióban nem találhatók meg, a magasabb verziószámú QR-kódokban pedig egyre nagyobb mennyiségben találhatók és a nem-lineáris torzulások detektálására és javítására szolgálnak. Az időzítő minta célja, hogy a sor- és oszlopszámot könnyebb legyen meghatározni. A szabvány egy négy modul széles, ún. csendes zónát³ ír elő a vonalkód körül a könnyebb detektálás érdekében, ezt azonban a készítők gyakran elhanyagolják.

2.2 QR-kód detektálási módszerek

A QR-kódok detektálására jelenleg számos eltérő megközelítés van használatban. Az ISO szabvány tartalmaz egy referencia algoritmust erre a célra [12], azonban az idők során már többfajta hasonlóan jó, vagy jobb megoldás született.

A megközelítések egy része a QR-kód sarkaiban elhelyezett lokalizációs minták jellegzetes voltára alapoz, amelyek a képen végigiterálva lokalizálhatóak [13]. Olyan részeket keresnek, ahol sötét-világos-sötét-világos-sötét részek váltakoznak $1:1:3:1:1$ arányban [15]. Nagyon gyors megvalósítás készíthető így, mivel a kód megtalálásához

³ Quiet Zone, de Free Zone-ként is hivatkoznak rá, amely szabad zónát jelent.

mindössze egyszer kell végigiterálni a képen, ugyanakkor csak viszonylag ideális körülmények között alkalmazható. Ideális körülménynek tekinthető, amikor a vonalkód a kép jelentős részét kitölti és mentes olyan torzulásoktól, amelyek a keresett jellegzetességeket nagy mértékben megváltoztatják. Természetesen fennáll a veszélye annak is, hogy a képen található, nem a kódhoz tartozó részekben előfordul ez a jellegzetes tulajdonság és ebből kifolyólag megghiúsítja a detektálást. Ezért fontos, hogy a vonalkód a kép szignifikáns részét kitöltse. A szabványban ismertetett referencia algoritmus is e módszerek közé tartozik. Az eljárást más módszerekkel kombinálva, amelyek kiszűrik és normalizálják a QR-kódot szintén jó eredmények érhetőek el [5] [13]. Egyes megközelítések azzal a feltételezéssel is élnek, hogy a kép középpontja a QR-kódhoz tartozik, és innen indítanak több irányba keresést, hogy lokalizálják a kód széleit [14]. Ez a QR-kód körül lévő csendes-zóna miatt lehetséges.

Szintén a pixel alapú technikákhoz sorolható a textúra analízis, amely a mátrixkódokra jellemző mintázatokot keres a képen [10]. A mintázatok keresése folyamán figyelembe vehetőek a pixelek olyan tulajdonságai, mint az intenzitás vagy a szín. A QR-kódokra jellemző, hogy a pixelek jellegzetes eloszlást mutatnak és ez alapján felmérhető, hogy a kép egy bizonyos tartományában milyen valószínűséggel található kód. A keresési teret leszűkítve ezután a keresést finomítva, vagy más technikát alkalmazva lehet pontosan lokalizálni a kódot. Ez egyre több olyan módszerben jelenik meg, amelyek az egy képen található több QR-kód detektálására fókuszálnak [16], hiszen az algoritmusok egy jelentős része azzal a feltételezéssel él, hogy a képen csak egyetlen QR-kód található. Egyes módszerek egyszerű, másfajta objektumok detektálására is alkalmas szűrők többlépcsős alkalmazásával, és e szűrők által szolgáltatott eredmények kombinálásával lokalizálják a QR-kódot [2].

A QR-kód különböző konvolúciós szűrőkkel és egyéb morfológiai operátorok bevetésével is kinyerhető [1], de ezek a módszerek nagy számításigényűek. Az olyan műveletek, mint a dilatáció, erózió és egyéb morfológiai operátorok hosszadalmas feldolgozást eredményeznek, amihez az egyes pixelek többszöri feldolgozása nagyban hozzájárul. Ez leginkább a valósidejű alkalmazások esetében hátrányos.

A lokalizáció során felhasználható a QR-kódok, illetve a mátrixkódok többségének azon tulajdonsága is, hogy egy rácsot töltenek ki, ahol a rácselemek a modulok. Az élek detektálásával felépíthető a QR-kódra feszülő rács egy modellje, és a rácselemekből való mintavételezés útján kinyerhetővé válnak a modulok értékei, miután sikerült meghatározni a kód irányultságát és az esetleges torzításokat [5] [13] [16]. Az ebbe a kategóriába tartozó módszerek nagyon ígéretesek, mivel jól megbirkóznak a QR-kód akár perspektív torzításával is, ugyanakkor szenvednek minden olyan problémától, amelyek az élszűrésre jellemzőek. A dolgozatban tárgyalt PClines algoritmus [4] is az éldetektáláson alapuló megközelítések közé sorolható.

3 ÉLDETEKTÁLÁS

Egy kép élpontjai lokális jellemzők, amelyek az egyes régiókat választják el egymástól [7]. Azzal a feltételezéssel élve keressük őket, hogy a pixel intenzitásokban jelentkező szakadások egyben a régiókban is szakadások [13]. Ideális esetben ezek a szakadások rövidek és jelentősek, aminek következtében az ideális él csupán pixelnyi vastagságú. A valóságban azonban az élek valamilyen átmenettel rendelkeznek, amely függ a képrögzítő eszköz minőségétől, a megvilágítástól és egyéb tényezőktől. Ezekből az átmenetekből kell az élfeldolgozás során a lehető legtöbb információt kinyernünk.

A különböző éldetektáló algoritmusok általában egy szürkeárnyaltos csatornával dolgoznak, amely keletkezhet több csatorna összevonásából, de lehet akár több rögzített csatorna közül az egyik [5]. Lehetőség van arra is, hogy speciális algoritmusokkal több csatornát egyenként feldolgozva az eredményeket végül egyesítsük.

Az élatmenetekből való információ kinyerésének a legalapvetőbb módszerei közé tartozik az első és második deriváltak használata, amelyekkel lokalizálható maga az átmenet, illetve a második derivált esetében az átmenet határai, az ún. nullátmenetek [7].

3.1 A deriváltak alkalmazása éldetektálásra

Az első derivált nem más, mint a szomszédos pixelek intenzitáskülönbségei. Közelítésére konvolúciót használunk, amely a kép feldolgozását jelenti egy súlyozott maszk (kernel) segítségével [7]. Ezen diszkrét differenciál operátorok a kernel méretében és értékeiben térnek el. A kernelekkel a képen haladva a pixelintenzitások kernel értékek által súlyozott összegzése történik. A képen a megfelelő irányultságú kernel-párral végighaladva a gradiens közelítését kapjuk, amelyeket G_x -el és G_y -al jelölhetünk. Ezekből a gradiens képekből a $G(x, y)$ függvény segítségével kiszámítható az élek válasza (erőssége), amely a gradiens értékek összege. Szintén kinyerhető az élek θ iránya. A válaszra gyakran alkalmazzák az Euklideszi-normát (3.1), az él θ irányultsága pedig a gradiens értékek hányadosának arkusz tangensével számítható (3.2). Természetesen más, akár az egyszerűbben számolható Manhattan-norma is alkalmazható az élválasz számítására, ami ugyan nem tartalmaz gyökvonást, sem hatványozás, de alkalmazása a pontosság kárára mehet.

$$G(x, y) = \sqrt{G_x^2 + G_y^2} \quad (3.1)$$

$$\theta = \text{atan}\left(\frac{G_x}{G_y}\right) \quad (3.2)$$

A konvolúciós kernelek általában négyzetesek és páratlan számú cellát tartalmaznak, ezzel biztosítva, hogy a kernelnek legyen egyértelmű központi eleme. Ekkor az eredeti

kép és a gradiens kép értékei között egyértelmű megfeleltetés lehetséges. A nagyobb kernel méret ellenállóbbá teszi az operátort a zajokkal szemben az értékek pedig további hatással vannak a viselkedésére. A 3×3 méretű kernelt használó operátorok nagyon elterjedtek és az alábbi változatokban gyakran fordulnak elő:

- **Roberts operátor** – Ez az operátor főként a kép átlós intenzitáskülönbségeit emeli ki. Bár kisebb a manapság elterjed 3×3 operátoroknál, mert így gyorsabban számolható, ez főként manapság nem kompenzálja a nagy zajérzékenységet és azt, hogy a páros cellaméret miatt nem szolgáltat egyértelmű megfeleltetést az eredeti és a gradiens kép között.
- **Prewitt operátor** – A legegyszerűbb 3×3 mérettel rendelkező operátor, amelyre G_x esetben egy $[1, 1, 1]^T$ átlagoló és egy $[-1, 0, 1]$ differenciáló kernel szorzataként tekinthetünk. Ennek köszönhetően a gradiens közelítését simítással számolja, azonban a kép nagyfrekvenciás részein még így sem túl hatékony.
- **Sobel operátor** – A Prewitt operátorral együtt az egyik leggyakrabban használt operátor. Módosított súlyai hozzájárulnak ahhoz, hogy sokkal jobban kezelje a zajt, mint az előző operátorok. A minél kisebb zajérzékenység fontos a deriváltak számításánál.

a)	<table> <tr><td>-1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> </table> <table> <tr><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	-1	0	0	1	0	-1	1	0	c)	<table> <tr><td>-1</td><td>-2</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>2</td><td>1</td></tr> </table> <table> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>-2</td><td>0</td><td>2</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> </table>	-1	-2	-1	0	0	0	1	2	1	-1	0	1	-2	0	2	-1	0	1										
-1	0																																						
0	1																																						
0	-1																																						
1	0																																						
-1	-2	-1																																					
0	0	0																																					
1	2	1																																					
-1	0	1																																					
-2	0	2																																					
-1	0	1																																					
b)	<table> <tr><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> <table> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> </table>	-1	-1	-1	0	0	0	1	1	1	-1	0	1	-1	0	1	-1	0	1	d)	<table> <tr><td>-3</td><td>-10</td><td>-3</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>3</td><td>10</td><td>3</td></tr> </table> <table> <tr><td>-3</td><td>0</td><td>3</td></tr> <tr><td>-10</td><td>0</td><td>10</td></tr> <tr><td>-3</td><td>0</td><td>3</td></tr> </table>	-3	-10	-3	0	0	0	3	10	3	-3	0	3	-10	0	10	-3	0	3
-1	-1	-1																																					
0	0	0																																					
1	1	1																																					
-1	0	1																																					
-1	0	1																																					
-1	0	1																																					
-3	-10	-3																																					
0	0	0																																					
3	10	3																																					
-3	0	3																																					
-10	0	10																																					
-3	0	3																																					

3.1 ábra: **a)** Roberts operátor, **b)** Prewitt operátor, **c)** Sobel operátor, **d)** Sharr operátor

- **Sharr operátor** – Mivel diszkrét differenciális operátorokról van szó, amelyek csak közelítik a gradienst, ezért lényeges szempont a zajsűrés mellett az is, hogy a közelítés minél pontosabb legyen. A Sharr operátor egy pontosabb közelítést ad a gradiensre, mint a Sobel operátor.

A tárgyalt 3×3 méretű operátorok átlós alakjai is használatosak, amelyek a Roberts operátorhoz hasonlóan a diagonális élekre adnak szignifikáns választ [7]. A konvolúciós kernel az átlagoló komponens nélkül is alkalmazható 3×1 formában, amely az élpontok szögeinek valamivel pontosabb értékeihez vezethet, mint a 3×3 esetben [5].

Ahhoz, hogy egy kernel válaszához tartozó pixelt élpontnak tekinthessünk, szükség van egy küszöbértékre, amely meghatározza, hogy mekkora az a minimális válasz, amelytől már annak tekinthető. A küszöbérték meghatározása általában a körülményektől függően, experimentálisan történik. Az eljárás során nem kell feltétlenül egyetlen, globális küszöböt dolgozni, hanem lokális küszöbölés is alkalmazható. A módszer hátránya, hogy akárcsak az él, a válasz is több pixelből tevődik össze, ami problémát jelent az él pontos helyzetének meghatározásában [13].

A második derivált szintén konvolúciós módszerrel közelíthető és az él két határát képes detektálni. Értelmezés szempontjából problémát jelenthet az élekre adott két válasz olyan tekintetben, hogy melyiket használjuk. Amennyiben a két válaszártéket, amelyek előjelükben különböznek, grafikusán értelmezzük, és egy képzeletbeli vonallal összekötjük őket, a vonal nullátmenete jó becslést ad a vastag él középvezetési pozíciójára. Jelentős hátránya a második deriválnak, hogy a rossz a zajtűrése [7], és bizonyos esetekben elveszíti az éles éleket [13]. Tekintettel arra, hogy a QR-kódok, és általában a mátrixkódok esetében nem vastag élekről van szó, illetve, hogy a bemeneti kép nagy valószínűséggel zajos, a második derivált sokkal kevésbé alkalmas a vonalkódok éleinek detektálására, mint a már ismertetett és a következőkben tárgyalt módszerek.

3.2 A Canny-éldetektáló

Az első deriváltat használó módszerek esetében fellépő több pixeles válasz problémájára igyekszik megoldást adni a Canny-éldetektáló (John F. Canny, 1986) [7], amely az él minél pontosabb lokalizálására törekszik. Minden élre igyekszik egy pixeles választ szolgáltatni. Ennek érdekében a képet a detektálás első lépéseként Gauss-szűrővel történő zajszűréssel készíti elő. Ezt követően közelíti a gradienseket a 3.1 differenciál operátorok valamelyikével vízszintes, függőleges és mindkét átlós irányban. A potenciális élpontok θ szögeit a $\{0, 45, 90, 135\}$ értékek valamelyikére kerekíti. A kapott gradiensek irányában kiválasztásra kerülnek azok az értékek, amelyek lokális maximumok, ami egy bináris képet szolgáltat a kiválasztott maximumokkal. Utolsó lépésként egy hiszterézises küszöbölést hajt végre, azzal a feltételezéssel élve, hogy a fontos élék folytonosan helyezkednek el. A magasabb küszöböt alkalmazva kijelölhetők az erős élpontok, majd ezekből kiindulva az alacsonyabb küszöböt alkalmazva kijelölhető a többi élpont, amennyiben azokhoz vezet út az erős élpontokból. Az algoritmus lényeges eleme a Gauss-szűrő és a hiszterézis küszöbeinek megválasztása, amelyekre, mint általában nem léteznek univerzális értékek. A Canny-éldetektálóval pontosabb eredmények érhetőek el, mint egyszerűen a gradiens közelítéssel, de a nagyobb számításigény miatt, főként valós idejű alkalmazásokban, nem mindig alkalmas a feladatra.

4 VONALDETEKTÁLÁS

Vonaldetektálás során adott n darab pontnak keressük azon részhalmazát, amelyek egy vonalon fekszenek. A problémára számos megoldás létezik, amelyek eltérő pontossággal és számítási komplexitással rendelkeznek. A legegyszerűbb megközelítés az, hogy minden lehetséges pontpárra meghatározzuk az általuk definiált egyenest, majd megkeressük azon pontok részhalmazát, amelyek közel helyezkednek el az egyeneshez [7]. Ekkor $(n(n-1))/2$ egyenest kéne megvizsgálnunk, amelyekre $n((n-1))/2 \sim n^3$ összehasonlítást kellene elvégeznünk. Ez a módszer tehát hiába triviális, ha számításigénye alkalmatlanná teszi a feladatra. További lehetőség a lineáris regresszió használata, amely úgy próbál egyenest illeszteni egy pontthalmazra, hogy a pontoknak az egyenestől számított négyzetes távolsága minimális legyen. Ez a megoldás már szofisztikáltabb az előzőnél, azonban problémát okoz az egyenestől távol eső pontok által okozott torzítás. Erre a megoldást a RANSAC⁴ algoritmus jelenthet megoldást, amely egy iteratív, nem determinisztikus módszer. Az iterációk során kizár bizonyos, a feltételezett egyenestől távol eső pontokat és így módon pontosítja az eredményt. Bizonyos számú iteráció után már elfogadható eredményeket szolgáltat.

4.1 A Hough-transzformáció

A Hough-transzformáció [9], amellyel részletesen is foglalkozunk egy olyan megközelítés, amely igyekszik hatékony megoldást nyújtani a problémára és ezt úgy teszi, hogy egy másik koordináta-rendszerbe konvertálja a pontokat. Ennek az a feltétele, hogy a detektálni kívánt alakzat, amely a legegyszerűbb esetben egy egyenes, rendelkezzen paraméteres egyenlettel. Ezzel a megközelítéssel pl. körök és ellipszisek is detektálhatóak, jelenleg azonban csak a feladat szempontjából számunkra hasznos egyenesekre szorítkozunk.

Egy egyenes két paraméterrel egyértelműen leírható a 4.1 egyenlet segítségével. Az egyenlet paraméterei rendre az egyenes meredeksége és metszete az ordinátatengellyel⁵.

$$y = ax + b \quad (4.1)$$

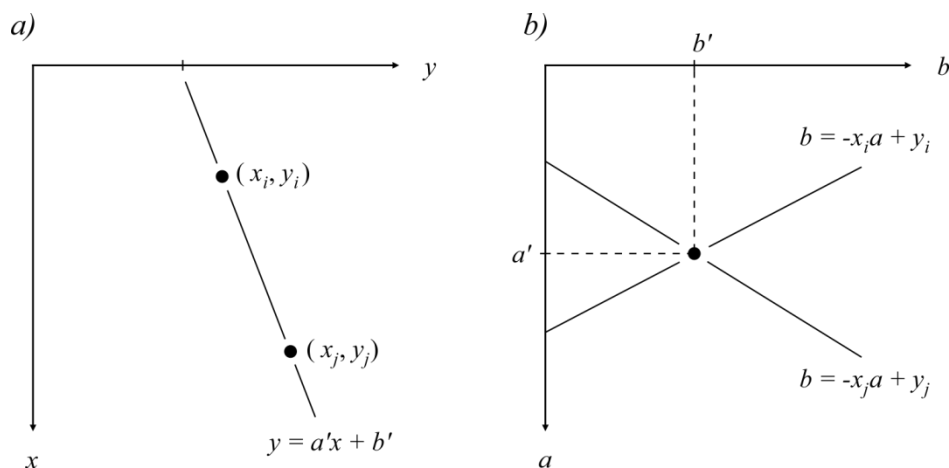
Az (x_i, y_i) ponton áthaladó összes egyenes kielégíti ezt az egyenletet. Az egyenlet átrendezésével kapjuk a 4.2 formát, amelyet az ab -síkból (paramétersíkban) ábrázolva egy egyenest kapunk az (x_i, y_i) pontra [7].

$$b = -x_i a + y_i \quad (4.2)$$

⁴ RANdom SAmple Consensus

⁵ Az angol szakirodalom „slope–intercept” kifejezéssel hivatkozik erre a paraméterezésre.

Ha tekintünk egy másik, (x_j, y_j) pontot, amelynek az előző ponthoz hasonlóan szintén van egy egyenese az ab -síkbán, és ez a két egyenes ott metszi egymást az (a', b') pontban, akkor az eredeti, Descartes koordináta-rendszerben mindkét pont az a', b' paraméterek által meghatározott egyenesen fekszik (4.1 ábra).



4.1 ábra: **a)** kolleáris⁶ pontok az xy -síkbán, **b)** kolleáris pontok az ab -síkbán [7]

A megközelítéssel az a probléma, hogy a függőleges egyeneseket nem tudja kezelni, ugyanis, amikor az egyenes közelít a függőlegeshez, a meredeksége a végtelenbe tart. Megoldásként le kell cserélni a paramétereket és az egyenlet egy más formáját kell használni. A két paraméter az egyenes x -tengellyel bezárt szöge ($\theta \in [-\pi, \pi]$) és az origótól mért távolsága ($\rho \in \mathbb{R}^+$)⁷ [6]. A pontok ebben az esetben a 4.3 egyenlettel írhatóak le és a θ és ρ által meghatározott ún. polár koordináta-rendszerben egy szinuszoiddal reprezentálhatóak. A szinuszoid, ugyanúgy, mint az ab -síkbán az egyenes, az (x_i, y_i) ponton áthaladó egyenesek összességével adódik.

$$x * \cos \theta + y * \sin \theta - \rho = 0 \quad (4.3)$$

A két koordináta-rendszer és az (x_i, y_i) pont kapcsolata a következő összefüggésekkel jellemezhető:

- az (x_i, y_i) pont a Descartes koordináta-rendszerben egy szinuszoid a Hough-térben,
- a Hough-tér egy pontja megfelel egy egyenesnek a Descartes koordináta-rendszerben,
- az egy egyenesen lévő pontok a Descartes koordináta-rendszerben, közös ponton áthaladó szinuszoidoknak felelnek meg a Hough-térben,
- egy szinuszoid a Hough-térben az összes (x_i, y_i) ponton áthaladó egyenesnek felel meg a Descartes koordináta-rendszerben.

⁶ egyazon egyenesen fekvő

⁷ Az angol szakirodalom az „angle–radius” kifejezéssel hivatkozik erre a paraméterezésre.

A Hough-tér ezen tulajdonságát dualitásnak nevezzük, amely ezt a paraméterezést a PTLM⁸-ek közé sorolja [4]. A módszer segítségével tehát meghatározhatóak az egy egyenesen fekvő pontok, mégpedig úgy, hogy a Hough-térben egy adott küszöbértéknél nagyobb mennyiségben egymást metsző szinuszoidok metszéspontjainak paramétereit a 4.3 egyenletbe behelyettesítve az y minden egyes értékére kapunk egy x értéket.

A Hough-transzformáció számításbeli attraktivitása abban rejlik, hogy a $\theta\rho$ -teret akkumulátor cellákra bontja fel. Az (i, j) koordináták és az $A(i, j)$ akkumulátor érték az (θ_i, ρ_j) paraméterekhez tartozó egyenesnek felel meg a Descartes koordináta-rendszerben. A pontokra minden lehetséges a értékre a b értékét kiszámítva a kapott értéket a legközelebbi megengedett (i, j) értékre kerekítjük. Ekkor az $A(i, j)$ cella értékét eggyel növeljük. Az algoritmus végén az $A(i, j)$ cellában lévő érték az (θ_i, ρ_j) paraméterű egyenesen fekvő pontok számát jelöli az xy -síkban. A paraméterter felbontása természetesen befolyásolja a kapott eredmény pontosságát. A θ -tengelyt K inkrementumra bontva minden (x_i, y_i) pontra K darab ρ értéket kapunk. Egy n pontból álló ponthalmazra ezért nK számítást kell elvégezni, tehát az algoritmus komplexitása n -ben lineáris, amennyiben a K értéke nem haladja meg az n -ét [7].

Amikor egy képen egyeneseket keresünk, akkor hasznunkra válik, hogy a Hough-transzformáció olyanokat is képes detektálni, amelyekben szakadások vannak. Természetesen az egyenes meghatározása után a pixelek szomszédságát vizsgálva mi döntjük el, hogy mekkora szakadás esetén tekintjük még a vonalat összefüggőnek. Elmosódott vagy torzított képek esetén ez a tulajdonság felettébb hasznosnak bizonyul. Fontos megjegyezni azonban, hogy véletlenszerűen elhelyezett tárgyak is generálhatnak egyenest tehát célszerű a keresési teret minél jobban leszűkíteni, illetve valamilyen módon szűrni az így keletkezett egyeneseket.

4.2 A PClines algoritmus

A PClines⁹ algoritmus a Hough-tér egy variánsát, a $\mathcal{TS}$ -teret használja a transzformáció során, amely a vonaldetektálás tekintetében rendelkezik mindazon tulajdonságokkal, amelyekkel a Hough-tér [4]. A $\mathcal{TS}$ -tér ugyanakkor a paraméterezése révén a számolás szempontjából előnyösebb tulajdonságokkal rendelkezik, mint a klasszikus Hough-tér.

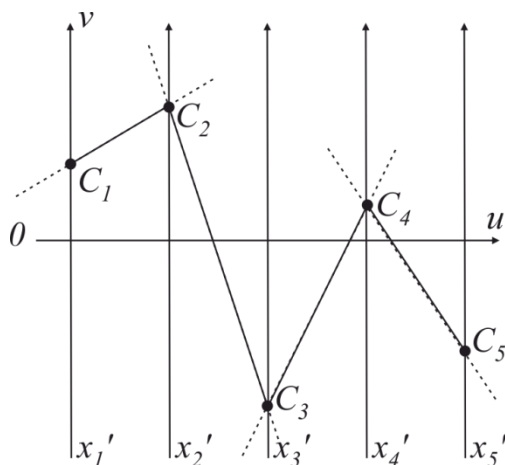
A párhuzamos koordináta-rendszer

A párhuzamos koordináta-rendszer (Maurice d'Ocagne, 1885) azon matematikai megoldások egyike, amelyek a saját korukban nem találtak jelentős felhasználást és feledésbe merültek. Később Alfred Inselberg munkássága révén újra előkerült, mint hatékony eszköze a többdimenziós adatok reprezentációjának és megjelenítésének [11].

⁸ Point To Line Mapping – azon paraméterezések halmaza, amelyekre teljesül a dualitás.

⁹ Parallel Coordinate lines – a párhuzamos koordináta rendszerre utal, amelyet az algoritmus alkalmaz.

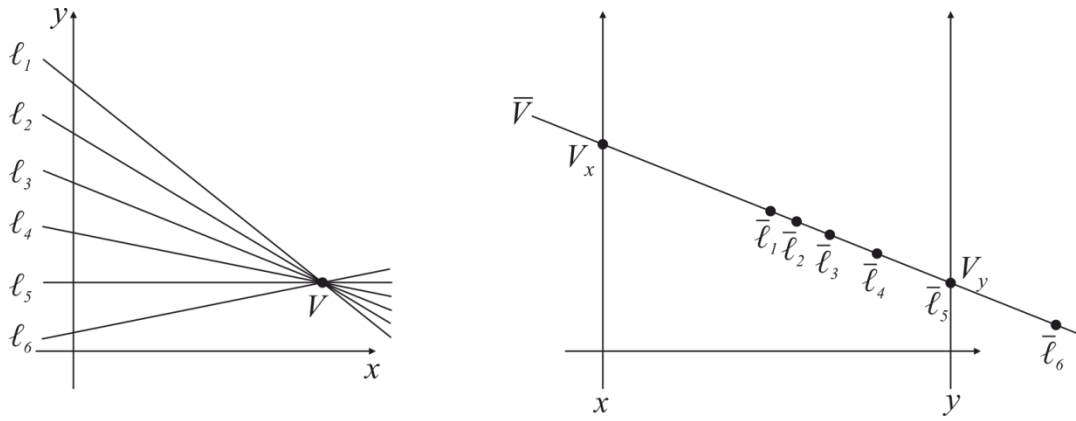
Ez főként az olyan adatokra igaz, amelyek három és több dimenziósak, mivel ezek ábrázolása jelentősen egyszerűbb azáltal, hogy nem kell a projekcióhoz hasonló technikákhoz folyamodni. Ráadásul időben nem korlátozza a megjeleníthető információ mennyiségét. Az N dimenziós Euklideszi-tér egy vektorát a Descartes koordináta-rendszerben pontként ábrázolhatjuk. A párhuzamos koordináta-rendszer azonban egy olyan teret alkot, amelyben a tengelyek kölcsönösen párhuzamosak és egyenlő távolságokra helyezkednek el egymástól. Ebben a térben az említett N dimenziós vektor $N-1$ egyenessel ábrázolható, amelyek összekötik a tengelyeket és együttesen egy törött vonalat alkotnak.



4.2 ábra: $C = (C_1, C_2, C_3, C_4, C_5)$ vektor ábrázolása ötdimenziós párhuzamos koordináta-rendszerben [4]

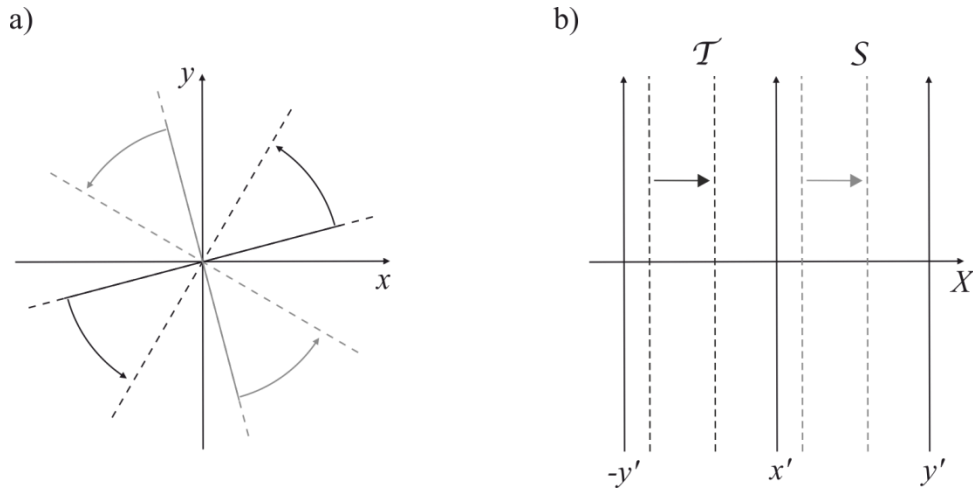
A képfeldolgozásra jellemző kétdimenziós esetben egy (x_i, y_i) pont a párhuzamos koordináta rendszerben egy egyenes lesz, amely az x' az x_i , az y' tengelyen pedig az y_i értéket veszi fel. A kollineáris pontok képei a párhuzamos koordináta-rendszerben egymást egy pontban metsző egyenesek. A metszet az xy -síkbán található egyenes képe, amelyen az említett pontok fekszenek. A vázolt reláció alapján, akár csak a Hough-tér esetében, definiálható egy PTLM a két tér között [4].

Vannak olyan speciális esetek, mint az $y = x$ egyenes, amelynek a párhuzamos koordináta-rendszerbeli képe a végtelenben fekszik, mivel pontjainak képei párhuzamosak egymással. A kétdimenziós Euklideszi-térnek ezt a korlátozását a projektív-tér úgy oldja fel, hogy az $ax + by + c = 0$ egyeneshez a $(db, -c, a + b)$ koordinátákat rendeli, ahol d a párhuzamos tengelyek közti távolság. A projektív térben két egyenes mindig metsző és az Euklideszi-tér párhuzamosai un. ideális pontban metszik egymást [8].



4.3 ábra: **a)** egymást egy V pontban metsző egyenesek **b)** az egyenesek és metszéspontjuk párhuzamos koordináta-rendszerbeli reprezentációi [13]

Az Euklideszi- és projektív-tér között további összefüggés is fennáll, amely a forgatás és eltolás műveleteinek dualitásaként jelenik meg. Az Euklideszi-térben való forgatás a párhuzamos koordinátákat alkalmazó projektív-térben, mint horizontális irányú eltolás jelenik meg (4.4 ábra) [13]. Ez a későbbiekben lesz fontos, amikor csak bizonyos irányultságú egyenesekkel kívánunk dolgozni.

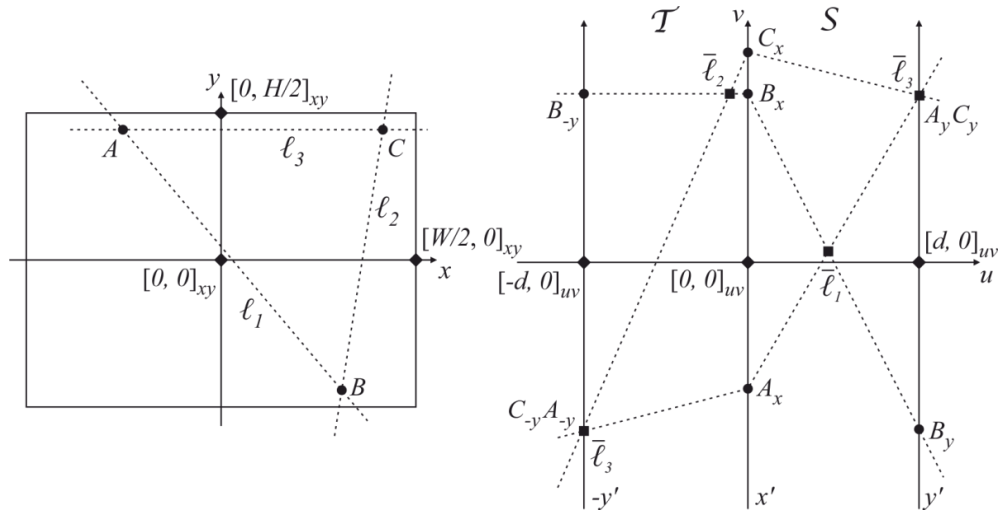


4.4 ábra: **a)** rotáció az Euklideszi-térben **b)** megfelelő transláció a projektív-térben

A $\mathcal{TS}$ -tér

Az $y = mx + b$ egyenes párhuzamos koordináta-rendszerbeli képe tehát az x' , y' tengelyek között van, de csak abban az esetben, ha m , vagyis az egyenes meredeksége $-\infty < m < 0$. Ez az x -tengelyhez képest az óramutató járásával ellentétes irányban a 90° – 180° közötti egyeneseket érinti. Az $m = 0$ esetben az egyenes képe az y' , $m = \pm\infty$ esetben pedig az x' tengelyen fekszik. Előbbi egy vízszintes, utóbbi egy függőleges egyenesnek felel meg a Descartes koordináta-rendszerben. Olyan esetekben, mint az $m = 1$, az egyenes képe kívül esik az x' , y' tengelyek által határolt térrészen. Jelöljük

ezt a térrészt S -el (Straight) és vezessünk be még egy, $-y'$ tengelyt, amely az y' tengely inverze. Az x' , $-y'$ tengelyek által határolt térrészt jelöljük $\mathcal{T}$ -vel (Twisted). A $\mathcal{T}$ altérbe fognak leképződni az $0 < m < \infty$ meredekségű, vagyis az x -tengellyel $0-90^\circ$ -ot bezáró egyenesek. Az $m = 0$ és $m = \pm\infty$ esetek kivételével az egyenesnek az egyesített $\mathcal{TS}$ -térben egyetlen képe lesz [4]. Utóbbi speciális esetet tekintve jellemző a térre, hogy ha azt az y' és $-y'$ tengelyek mentén összekapcsoljuk, akkor a Möbius-szalagnak megfelelő jelenséget kapunk, tehát a tér átfordításával a $-y$ tengelyről az y' tengelyre jutunk [5]. A párhuzamos koordináta-rendszer által tehát olyan térhez jutunk, amelybe a megfelelő paraméterezéssel minden egyenes áttanszformálható a Descartes koordináta-rendszerből.



4.5 ábra: Egymást metsző egyenesek és képeik a $\mathcal{TS}$ -térben [4]

A $\mathcal{TS}$ -tér a projektív-tér u - v koordinátáit használja és a 4.4, 4.5 képletekkel számítható korlátokkal jellemezhető:

$$-d \leq u \leq d \quad (4.4)$$

$$-\max\left(\frac{W}{2}, \frac{H}{2}\right) \leq v \leq \max\left(\frac{W}{2}, \frac{H}{2}\right) \quad (4.5)$$

, ahol:

- W az eredeti kép szélessége,
- H az eredeti kép magassága,
- d a párhuzamos tengelyek távolsága.

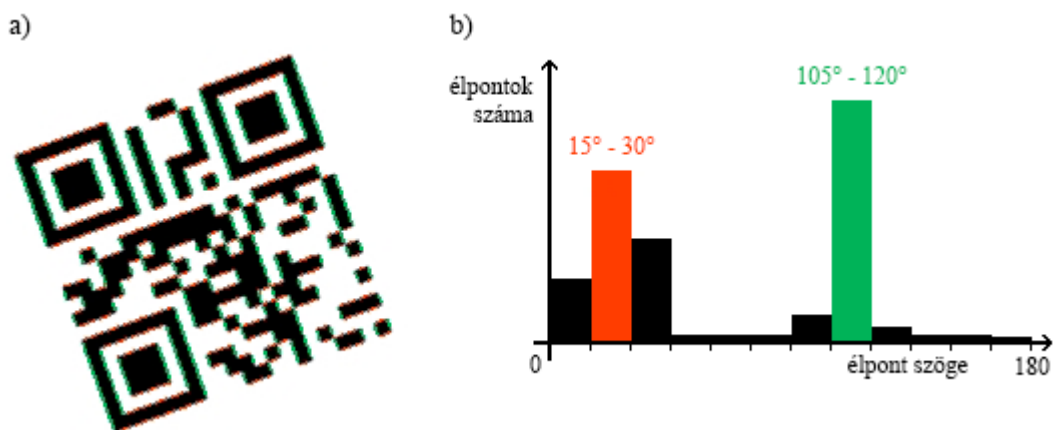
A $\mathcal{TS}$ -teret egyenletesen diszkretizálva a Hough-transzformációnál ismertetett módon kialakíthatjuk az akkumulátor-mátrixot. Az (x_i, y_i) pont két egyenesre fog leképződni, amelyek közül az egyik az S -el, a másik a $\mathcal{T}$ -vel jelölt térrészben lesz. A vonalak végpontjainak vízszintes koordinátái rögzítve vannak a $\{0, d, -d\}$ értékeken. A függőleges koordináták az x , y , illetve $-y$ értékeket veszik fel.

5 PÁRHUZAMOS EGYENESEK DETEKTÁLÁSA

Az előző fejezetekben vázolt elméleti háttér segítségével a QR-kódot úgy próbáljuk detektálni, hogy a felépítésére jellemző négyzetrácsos struktúrát próbáljuk minél jobban közelíteni, majd az egyes rácselemeket mintavételezve kinyerni a QR-kód bináris képét. A fejezetben további módszerek kerülnek ismertetésre, amelyek szorosan összefüggenek egyes részfeladatokkal vagy ihlették azok megoldási módját.

5.1 Az élpontok detektálása és szűrése

A detektálás első lépése a bemeneti kép éleinek detektálása valamely, a 3. fejezetben ismertetett módszerrel. Mivel az algoritmus abból a feltételezésből indul ki, hogy a QR-kódot két, egymással közelítőleg 90° -ot bezáró egyenesek halmaza alkotja, az éldetektálás során kinyert θ szögek alapján hisztogramot készítünk [16], ami a HOG¹⁰ [18] technikához hasonlós. Ehhez az élpontok szögeit 180° -os tartományba konvertáljuk, hiszen egy egyenes pontjai kétfajta irányultsággal rendelkezhetnek a teljes szögtartományban. A szögtartományt egyenletes intervallumokra felosztva, minden élpontra a szögének megfelelő intervallumhoz tartozó edényt inkrementálva kialakítjuk a hisztogramot. Amennyiben a kezdeti feltételezésünk helyes, úgy a hisztogramnak tartalmaznia kell két, egymástól nagyjából 90° -nyi távolságra lévő csúcsot, amelyek a domináns irányokat képviselik a képen.



5.1 ábra: **a)** QR-kód a detektált szignifikáns irányokkal **b)** a szignifikáns irányok az élpontok szögei alapján készült hisztogramon

Az élpontok szűrése első körben a domináns irányokhoz (5.1 ábra), és esetlegesen a velük közvetlenül szomszédos edényekhez tartozó élpontokat tartja meg. Ez hivatott megszabadulni a zaj egy részétől, valamint a következő lépéseket is gyorsítja azáltal, hogy a detektált élpontoknak csak egy részével kell tovább dolgozni. A megmaradt

¹⁰ Histogram of Oriented Gradients

élpontok egy további szűrésen is átesnek, amikor a két irányhoz tartozóak közül irányonként az N darab legnagyobb válasszal rendelkező kerül kiválasztásra, amelyek egy adott küszöbnél is nagyobbak, és csupán ezekre végezzük el az akkumulációt a Hough-térben.

5.2 A Hough-tér akkumulációja

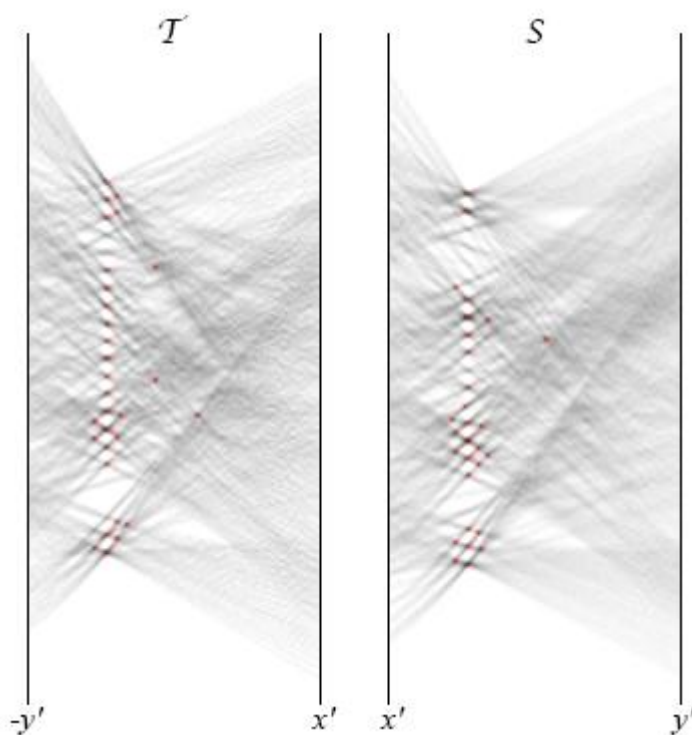
Az algoritmusnak ezt a részét nem csak az előző lépésben végzett szűrés, hanem az Euklideszi- és projektív-tér 4.2 fejezetben tárgyalt dualitása is gyorsítja (4.4 ábra). Egy $[\theta - \alpha, \theta + \alpha]$ szögtartomány a párhuzamos koordinátákat használó projektív-térben az $[u - \delta, u + \delta]$ intervallumra képződik le. Ez az intervallum egy oszlopot képez a projektív-térben. Az Euklideszi-tér egy φ szöge és a projektív-tér δ -ja között pedig az 5.1 összefüggés áll fenn

$$\delta = \frac{\varphi d}{90} \quad (5.1)$$

, ahol:

d a párhuzamos tengelyek közti távolság.

A vázolt összefüggésnek köszönhetően tehát nem szükséges a pontot a teljes Hough-térben akkumulálnunk, hanem annak csak egy oszlopos tartományában, ami szintén gyorsítja a feldolgozást [4].



5.2 ábra: A teljes $\mathcal{TS}$ -tér akkumulációja, a bejelölt lokális maximumokkal

Az akkumulált pontok a kolineáris pontokra jellemző metszéspontokat, azaz lokális maximumokat alkotnak, amelyek a képen található szignifikáns egyenesek képei. A domináns irányú egyenesek csoportjainak egyike az S a másik a $\mathcal{T}$ térrészben fog elhelyezkedni, amennyiben a fejezet elején tett feltételezésünk helyes. Ez azt jelenti, hogy mindkét térrész esetében végre kell hajtani az akkumulációt az adott szögtartományhoz tartozó, választott pontokkal. Ezek a pontok fognak a következő lépésben a párhuzamos egyenesek modelljeinek felállítására szolgálni.

5.3 A párhuzamos egyenesek modelljeinek felállítása

Képek rögzítése során a valóságban háromdimenziós párhuzamosok kétdimenziós térbe való perspektív projekciója miatt azok egy adott, ún. távlatpontban metszővé válnak [8]. A QR-kód párhuzamosainak mindkét szögtartományhoz tartozó csoportja rendelkezik ilyen távlatponttal. A párhuzamosok modelljének megalkotásához kezelniük kell ezt a jelenséget, amihez a kettősvizony lesz segítségünkre.

A kettősvizony

A kettősvizony ¹¹ [17] egy viszonyszám, amely négy a már említett tulajdonságú egyenes és egy mindegyiküket metsző egyenes négy pontja (metszéspontok) között áll fenn.

A projektív geometriában ugyanis ez a viszony a centrális vetítésre invariáns.

$$(A, B; C, D) = \frac{AC \cdot BD}{BC \cdot AD} \quad (5.2)$$

, ahol:

A, B, C, D a kolineáris metszéspontok (5.3 ábra).

Az említett centrális vetítésre vonatkozó invariancia miatt tehát az 5.3 ábrán látható esetre teljesül az $(A, B; C, D) = (A', B'; C', D')$ egyenlőség.

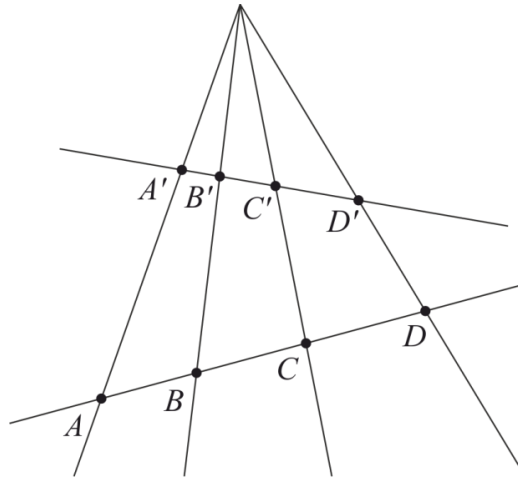
Az összefüggéseket az egyenesek párhuzamos koordináta-rendszerbeli képeire alkalmazva tehát definiálható az

$$\frac{|\bar{\ell}_c \bar{\ell}_a| : |\bar{\ell}_c \bar{\ell}_b|}{|\bar{\ell}_d \bar{\ell}_a| : |\bar{\ell}_d \bar{\ell}_b|} = \frac{(c - a) : (c - b)}{(d - a) : (d - b)} \quad (5.3)$$

összefüggés, ahol:

a, b, c, d az egyenesek indexei, amelyekre $a < b < c < d$
 $|\bar{\ell}_i \bar{\ell}_j|$ az ℓ_i, ℓ_j egyenesek képeinek távolsága a párhuzamos koordináta-rendszerben.

¹¹ Cross-ratio



5.3 ábra: Párhuzamos egyenesek centrális projekciója a kettősvizonyt szemléltető metsző egyenesekkel [5]

A modellt alkotó jelöltek kiválasztása

A Hough-térben történt akkumulációval nyert lokális maximumok nagy valószínűséggel az adott szögtartományba tartozó párhuzamos egyenesek képei. A modell felállításához szükséges ezen egyenesek távlatpontjának meghatározására, hiszen ha párhuzamosak, akkor ebben az egy pontban metszik egymást. Ehhez mindkét térrészben meg kell határoznunk azt az egyenest, amely az Euklideszi-térben lévő távlatpont lehető legpontosabb képe. A kettősvizony felhasználásával egy lehetséges modell meghatározásához három inicializáló pontra van szükség, amelyekkel egyértelműen meg tudunk határozni egy negyedeket. Ezzel a módszerrel aztán további pontokat tudunk meghatározni, lépésenként felépítve a párhuzamosok egész modelljét.

A három pont kiválasztása nem egyszerű feladat, mivel a lehető legpontosabb modellt szeretnénk felállítani. Felmerül az ötlet, hogy több ponthármast is kipróbáljunk és valamilyen metrika segítségével értékeljük azokat a végén kiválasztva a legkedvezőbb értékelést elérőt. Nevezzük ezentúl a ponthármast hipotéziseknek [16]. Az algoritmus sebességét befolyásolja, hogy mennyi hipotézist értékelünk ki, ugyanakkor célszerű lenne megtartani az algoritmus determinisztikus voltát. Ez úgy érhető el, hogy annyi lokális maximumot használunk fel, amennyiből minden hipotézis kiértékelhető. A módszer sokban hasonlít a RANSAC algoritmusra azonban nélkülözi annak nemdeterminisztikus voltát [13]. Természetesen a determinisztikus működés nem feltétele a jó eredménynek, csupán ajánlás a reprodukálhatóság miatt és annak elkerülése érdekében, hogy az algoritmus pont a legmegfelelőbb hipotézist nem értékeli ki.

A hipotézisek kiértékelése és a modell meghatározása

A távlatpont meghatározásához felhasználhatjuk a választott hipotézist, azonban már ezt megelőzően is megbecsülhetjük az elhelyezkedését. A választott N darab lokális maximumot egy újabb TS -térben akkumulálva kereshetjük az általuk meghatározott maximumot [5]. Ennek kedvező esetben elő kell állnia hiszen azt feltételezzük, hogy a lokális maximumok párhuzamosok képei, tehát az előző TS -térben kolineárisaknak kell lenniük. Amennyiben ez a maximum az új akkumulátortérben előáll, ez lesz majd a távlatpont modellje az egész algoritmus folyamán. Megtehetjük azonban azt is, hogy minden kiértékelésre váró hipotézishez hozzárendeljük a hozzá leginkább illeszkedő távlatponti képet, ami a három pontot leginkább közelítő egyenes. Felhasználhatjuk a 4. fejezetben említett lineáris regressziót megkeresve azt az egyenest, amelytől a hipotézis pontjainak vett négyzetes távolsága¹² a lehető legkisebb [13]. Ekkor a távlatpont modellje mindig az aktuális hipotézishez kapcsolódik és annak kiértékelése folyamán áll fenn.

A következő lépésnél az algoritmus több irányt vehet. Egy lehetséges irány az, hogy elkészítjük az akkumulátor-tér egy metszetét a távlatpont képe mentén. Ezzel egy hisztogramszerű eredményt kapunk, amely a távlatpont képének diszkrét pontjain vett értékeket tartalmazza az akkumulátor-térből. Az egyszerűség kedvéért nevezzük ezt a hivatkozott szakirodalom alapján profilnak [5]. Feltételezésünk szerint a profilon található csúcsok a keresett párhuzamosokat reprezentálják, hiszen az előző akkumulátor-térben ebből a feltevésből indultunk ki. A profil azonban továbbra is tartalmazhat zajokat, amik az éldetektálás tökéletlensége miatt keletkeztek. Egy küszöbölést kell tehát végrehajtanunk valamilyen módon. Egy lehetséges megoldás, hogy a csúcsokat először a legnagyobb érték egy kis részével inicializált küszöbvel szűrjük. A megmaradt csúcsok értékeiből, a legnagyobbat és a legkisebbet elhagyva, átlagot számítva, ennek az átlagnak egy részéből új küszöböt állítunk, amellyel egy újabb szűrést hajtunk végre. További lehetséges megközelítés az ún. „Peakiness test” alkalmazása, amely a csúcsokat csúcsosságuk alapján rangsorolja. A csúcsosság egy olyan metrika, amely meghatározza, hogy egy csúcs az általa és öt körülvevő völgyek által definiált négyszögletű területet milyen mértékben tölti ki, és a magas, keskeny csúcsokat részesíti előnyben. A megfelelő küszöb megválasztásával, ezzel a módszerrel is kiszűrhetők a jelentéktelen, zaj által előidézett csúcsok. Az ezt a megközelítést használó módszer csak ekkor, a szűréssel kapott csúcsokból kezd hipotéziseket alkotni, majd a profilon határozza meg a modellt. A modell meghatározását azzal kezdjük, hogy keressük a csúcsok leghosszabb olyan sorozatát, amelyet belülálló¹³ csúcsok alkotnak, a többi csúcs pedig kívülálló¹⁴. Ehhez a kettősviszony képletét használjuk fel, amelybe

¹² SSD – Squared Sum of Differences

¹³ Inlier – lásd: <http://www.fomi.hu/honlap/magyar/szaklap/2002/12/4.pdf>

¹⁴ Outlier – lásd: ¹²Inlier.

először a csúcsok között elhelyezkedő csúcsok számát helyettesítjük be, majd egy következő lépésben a profil kezdetétől vett távolságait. A távolságok meghatározhatóak, mivel a profil bekorlátozható, ugyanis az akkumulátor-tér is az volt. Egy negyedik csúcsot meghatározva ellenőrizhetjük, hogy az mennyire közelíti a tényleges csúcs pozícióját és a kapott távolság küszöbölésével eldöntjük, hogy elfogadjuk, vagy elvetjük az eredményt. Az egész modell meghatározásához szükséges hipotézist minden esetben a belülálló csúcsokkal inicializáljuk és profilban lévő csúcsok számán felül, mindkét irányban meghatározunk még néhány lehetséges párhuzamost, hogy biztosan lefedjük az egész QR-kódot.

A távlatpont képének lineáris regresszióval történő közelítésekor a hipotézist alkotó pontoknak a kapott egyenesre való projekciója szükséges [13]. Ezt a már tárgyalt módszerhez hasonlóan a TS -tér metszetének elkészítése követi a távlatpontot jelképező egyenes mentén. Az egyenesre projektált pontok egy adott fixponttól vett távolságait és a köztük található lokális maximumok számát a kettősviszony képletbe behelyettesítve következhet a modell többi pontjának meghatározása. A mindkét irányba vett további pontok számítása itt nem becslés alapján, hanem egy mohó algoritmus segítségével zajlik, amelyet több megállási feltétel korlátoz. Ezek kisszámú, meghatározott küszöb alatti, vagy egy előző hipotetikus minimum alatti maximumokat engedélyeznek, majd a küszöböt elérve leállítják a további pontok számolását.

A modellalkotás során felvetődő problémák és azok kezelése

Mindkét tárgyalt esetben problémát jelentenek az esetleges zaj által fennmaradt csúcsok, illetve a tévesen kiszűrt párhuzamosok csúcsai. Ez három jellegzetes esetet eredményezhet [5]:

- a hipotézist alkotó három pont közül az egyik nem párhuzamost reprezentál,
- a hipotézis pontjai között található csúcsok valamelyike nem párhuzamost reprezentál, vagy hiányzik egy párhuzamoshoz tartozó csúcs,
- a hipotézis mindhárom pontja párhuzamost reprezentál és a köztük lévő csúcsok száma is hűen tükrözi a tényleges párhuzamosok számát.

Az első két esetben a hipotézis által generált modell pontatlan és így használhatatlan lesz. A harmadik eset az egyetlen, amely jó modellt szolgáltat.

Esetleges további probléma a QR-kód képének jelentős perspektív torzulása, amely nem felel meg annak a feltevésünknek, hogy az egyes szögtartományokhoz tartozó párhuzamosok csoportjainak képei a szögek által határolt térész akkumulált oszlopában lesznek. Ekkor ugyanis bár meghatározhatóak a modell által a párhuzamosoknak megfelelő pontok, értékeik azonban nem ellenőrizhetők, mivel nem végeztük el a szükséges akkumulációt [13]. Az algoritmus esetén tehát azt feltételezzük, hogy a perspektív torzulás nem jelentős.

A leírt módszer jellegzetessége az is, hogy a modell a hipotézis három pontjában remekül fog illeszkedni a tényleges párhuzamosokra, azonban a pontoktól távolodva egyre pontatlanabb lesz. Erről csak az egyik hivatkozott irodalom tesz említést [5], amely egy lehetséges megoldást is kínál. Jelöljük a hipotézis pontjait h_1, h_2, h_3 -al. A h_1 pontot egy e_0 távolsággal balra, majd jobbra eltolva számoljuk ki újra a modellt és metrikájával annak jóságát. Amennyiben a metrika a h_1 pont eredeti helyén szolgáltatja a legkedvezőbb eredményt, folytassuk az algoritmust a hipotézis következő pontjával, egyébként pedig folytassuk a mozgást a legkedvezőbb eredményt szolgáltató irányba mindaddig, amíg egyre jobb eredményeket kapunk. Mindhárom pont feldolgozása után végezzük el az említett lépéseket újra az $e_1 = e_0/2$ távolságot használva, és így tovább. Az eljárás nem biztosítja, hogy az optimalizáció nem ragad be a metrika egy lokális minimumába, ezért több hipotézissel is ajánlott elvégezni a műveletet és a legkedvezőbbet választani.

6 A QR-KÓD MINTAVÉTELEZÉSE A DETEKTÁLT PÁRHUZAMOSOK SEGÍTSÉGÉVEL

A párhuzamosok két csoportjához már a modell építése során kiszámítjuk a minimumhelyeket, mivel a metrika ezeket is figyelembe kell, hogy vegye. Az egymást követő maximumhelyek közötti minimumhelyek olyan képzeletbeli párhuzamosokat reprezentálnak, amely a QR-kód moduljainak közepén haladnak át. A mintavételi pontokat e két halmazba tartozó képzeletbeli párhuzamosok Descartes-szorzata szolgáltatja. A $\mathcal{TS}$ -tér pontjaiból az egyeneseket az 6.1 és 6.2 képletekkel kapjuk meg a már ismertetett θ - ρ paraméterezésben. Az 6.1 képletben a $\mathcal{T}$ térrészből a $d - u$ az S térrészből pedig a $d + u$ összefüggéssel végezhetjük el az inverz transzformációt.

$$\theta = \operatorname{atan}\left(\frac{u}{d \pm u}\right) \quad (6.1)$$

$$\rho = \frac{vd}{\sqrt{(d-u)^2 + u^2}} \quad (6.2)$$

, ahol:

d a párhuzamos koordináta-tengelyek egymástól vett távolsága,
 u, v a pont koordinátája a projektív-térben.

Az így kapott párhuzamosok metszéspontjainak u - v koordinátáit a $\theta_1 < \pi$ esetben az 6.3, 6.4 képletekkel számíthatjuk ki:

$$u = \frac{\rho_1 - v \sin \theta_1}{\cos \theta_1} \quad (6.3)$$

$$v = \frac{\rho_1 \cos \theta_2 - \cos \theta_1 \rho_2}{\sin \theta_1 \cos \theta_2 - \sin \theta_2 \cos \theta_1} \quad (6.4)$$

, ahol:

θ_1, θ_2 az első, illetve második egyenes x -tengellyel bezárt szöge,
 ρ_1, ρ_2 az első, illetve második egyenes távolsága az origótól.

Minden további θ_1 -re az 6.5, 6.6 képletek ajánlottak. Ez a megközelítés pontosabb eredményeket szolgáltat a $\theta_1 \sim 0$ és $\theta_1 \sim \pi/2$ esetekben.

$$u = \frac{\rho_1 \sin \theta_2 - \sin \theta_1 \rho_2}{\cos \theta_1 \sin \theta_2 - \cos \theta_2 \sin \theta_1} \quad (6.5)$$

$$v = \frac{\rho_1 - u \cos \theta_1}{\sin \theta_1} \quad (6.6)$$

Az így kapott metszéspontok u - v koordinátáit az eredeti kép középpontjához kell viszonyítani [5].

A mintavételezéssel nyert alacsony felbontású kép tartalmazza a QR-kódot, amely még szürkeárnyaltos. Ahhoz, hogy hatékonyan tudjunk vele dolgozni binarizálni kell a képet, erre pedig a valamilyen küszöbölési eljárásra van szükség.



6.1 ábra: Mintavételi pontok és a kinyert alacsony felbontású kép [13]

Mínthogy a fényviszonyok sokfélék lehetnek, a globális küszöbölés nem tűnik hatékony megoldásnak. Az egyenetlen megvilágítással a lokális, adaptív küszöbölési technikák sokkal jobban megbirkóznak [7]. Ilyen lokális küszöbölés például a Niblack algoritmus is (W. Niblack, 1986), de a képet kisebb, egyforma cellákra bontva, és azokban lokális küszöbölést alkalmazva [13] szintén elfogadható eredményeket érhetünk el.

Az előállított bináris képen egy egyszerű csúszóablakos mintaillesztéssel megtalálhatjuk a lokalizációs mintákat, és a QR-kódot, ha szükséges, egy forgatással a megfelelő irányultságba állíthatjuk a dekódoláshoz. Az így kapott kép vélhetően mentes a torzításoktól a modelltől fakadó hibákat kivéve. A csúszóablakos mintavételezés során kijelölésre kerülnek a lokalizációs minták legvalószínűbb helyei, majd az egymáshoz viszonyított pozíciójuk alapján kerül meghatározásra az orientáció. Ezt nagyban nehezítheti, ha valamelyik lokalizációs minta nem szerepel a képen, vagy hiányos. Ilyenkor még bizonyos esetekben segítségül szolgálhatnak az igazító minták, amennyiben azok szerepelnek a képen. A 6.1 ábrán látható, hogy a mintavételezés a bejelölt pontokkal nem a standard orientációban állította elő.

A lokalizációs minták megtalálása fontos továbbá a QR-kód kivágása miatt is, hiszen a dekódoló eljárásnak csupán a QR-kódot kell átadnunk, így azt ki kell vágnunk a környezetből, amely azért kerül a képre, mert a modellkészítés során néhány pozícióval több került meghatározásra az extrém esetek jobb kezelhetőségének érdekében.

7 IMPLEMENTÁCIÓ

7.1 Implementációs megfontolások

Az implementációval kapcsolatos egyik legfontosabb szempont, hogy milyen környezetben kerül felhasználásra az algoritmus. A detektálásra rendelkezésre álló idő, illetve az elvárt minőség alapján kell döntenünk arról, hogy milyen programozási nyelven, milyen paraméterekkel rendelkező hardver felhasználásával implementáljuk az algoritmust. A részfeladatok nagy része jól párhuzamosítható feladat így felmerülhet akár a manapság feltörekvőben lévő GPGPU¹⁵-kra való implementáció. Ugyanakkor a bevezetésben említett mobil platformok jelentősége is nagy, ami egy teljesen más megközelítést követel. A módszer ezen a téren is tartogat lehetőségeket abban, hogy több helyen is redukálhatjuk a számítások mennyiségét, ez azonban a detektálás valószínűségének csökkenéséhez vezethet. Az egyes részfeladatokban elvégzendő számítások mennyiségének finomhangolásával és körültekintő teszteléssel ideális esetben elérhető egy optimum a minőség és végrehajtási idő között.

A végeredmény minősége a kiértékelt modellek számán és metrikák mennyiségén felül leginkább a paraméterek optimális meghatározásától függ. Az algoritmus lépéseinek zöménél valamilyen küszöb, megállási feltétel vagy egyéb paraméter befolyásolja a következő lépésbe kerülő adatok minőségét és mennyiségét. A minőség biztosítása azt a célt szolgálja, hogy az esetleges hibák nagy része, ideális esetben az összes, minél korábban szűrésre kerüljenek. Az adatok mennyiségének korlátozása pedig a nélkülözhető adatokkal történő felesleges számításokat hivatott megelőzni. A paraméterek megválasztásának bonyolultsága jellegzetessége a képfeldolgozásnak és általában nem adhatóak meg univerzális értékek a paraméterek többségére vonatkozóan. Az értékeket általában empirikus módon, valamilyen, a felhasználás körülményeit a lehető legjobban tükröző tesztsomagot kiértékelve lehet becsülni és finomítani.

A dolgozat által tárgyalt QR-kód detektálási eljárásának több implementációja is létezik már [5][13][16], amelyeket feldolgozási sebességük képessé tesz valósidejű felhasználásra a manapság átlagosnak tekinthető x86-os processzoron. Minthogy e dolgozat célja elsődlegesen nem a valósidejű felhasználhatóság bizonyítása, hanem az eljárás képességeinek minél széleskörűbb feltárása az elkészítésre kerülő implementáció az áttekinthetőségre és bővíthetőségre helyezi a hangsúlyt.

Az implementáció x86-os architektúrára készül C# nyelven a Microsoft .NET keretrendszerének, illetve az OpenCV¹⁶ és annak a .NET keretrendszerre íródott

¹⁵ General-Purpose computing on Graphics Processing Units – általánosan programozható grafikus chippek.

¹⁶ <http://opencv.org/>

menedzselt absztrakciós rétegének, az Emgu CV¹⁷-nek felhasználásával. Tekintettel arra, hogy a dolgozat a QR-kód detektálás képfeldolgozási problémakörét tárgyalja a QR-kód normalizált, alacsony felbontású, binarizált képét dekódoló eljárás egy már megvalósított implementáció lesz, amely a jövőben kerül kiválasztásra. A felsorolt technológiák választását leginkább a velük felgyülemlett tapasztalataim indokolják, amelyek a fentebb említett célok figyelembevételével és a dolgozat tudományos volta miatt megfelelnek a célra.

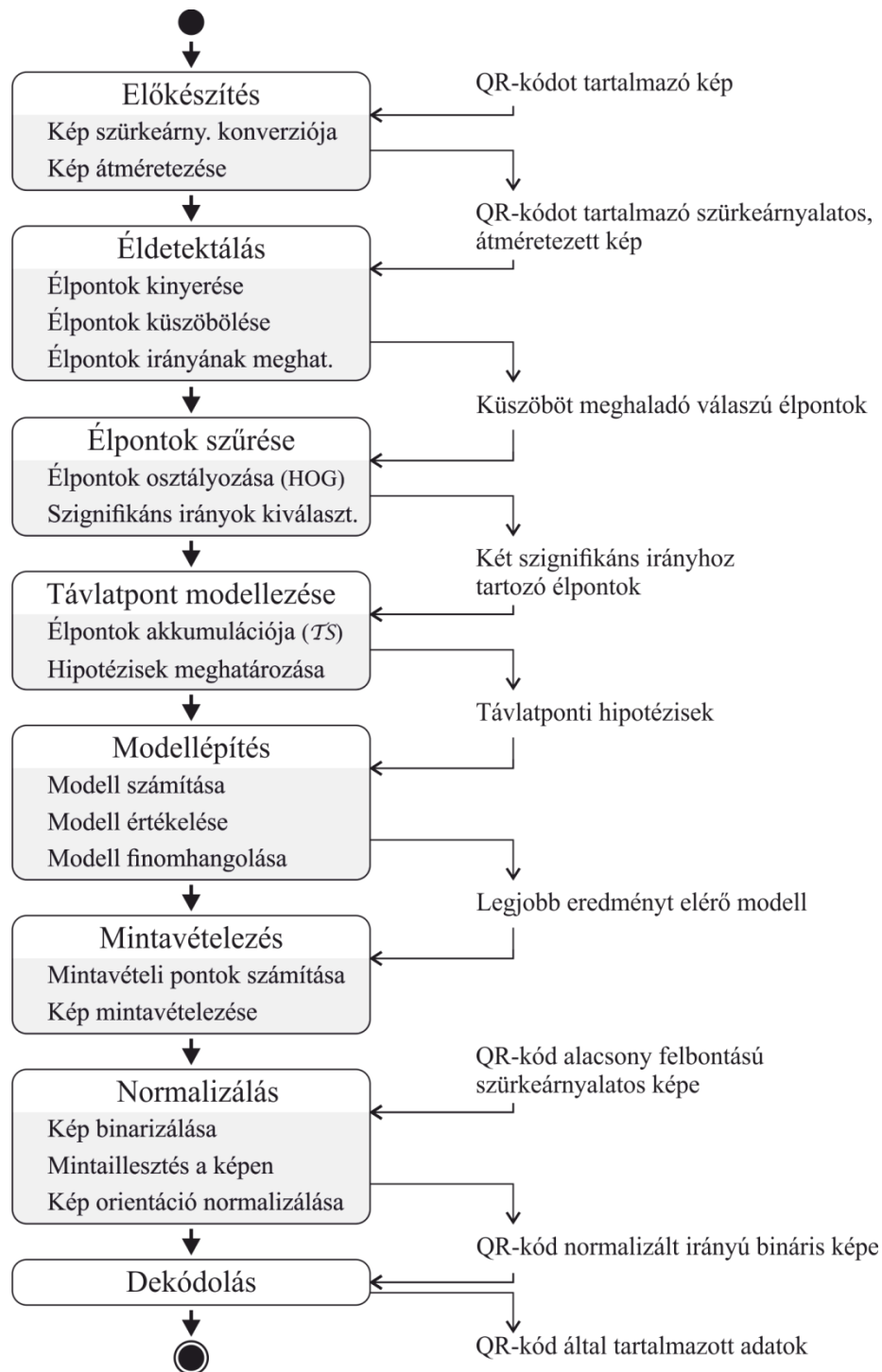
7.2 Az algoritmus lépéseinek összefoglalása

Az algoritmus végrehajtását a főbb lépések mentén tagolva, majd ezeket további részfeladatokra bontva a 7.1 ábrán megfigyelhető felépítéshez juthatunk. Az ábra a lépéseken kívül a szignifikáns bemeneteiket és kimeneteiket is tartalmazza.

Az egész algoritmus bemenetét egy a detektálni kívánt QR-kódról készült képzi, amelynél azzal a feltételezéssel élünk, hogy a képen pontosan egy QR-kód található. A bemenetként szolgáló képet az előkészítési fázisban előbb szürkeárnyalatossá konvertáljuk, majd amennyiben szükséges csökkentjük a méretét, hogy megelőzzük a túlságosan terjedelmes, kevésbé szignifikáns élválaszokat. Az éldetektálás a 3. fejezetben tárgyalt módszerek valamelyikével történik és csak a küszöböt meghaladó válaszü élpontok kerülnek megtartásra. A dolgozathoz fejlesztett implementáció az 5.1 ábrán látható QR-kód élpontjait a 3×1 krenelméretű Sobel operátorral határozta meg. Az élpontok irányultságának meghatározása a 3.1 fejezetben ismertetett módon történik. A következő lépésben az élpontok szűrése pont ezen irányultságok alapján történik és kiválasztása kerül az a két szignifikáns irány, amelyek nagy valószínűséggel a keresett párhuzamosok két csoportját alkotják. Az 5.1 ábrán megfigyelhető, hogy a szignifikáns irányok jól elkülönülnek a többi iránytól. A hisztogram felbontásának finomításával azonban a lokális maximumok környezetének vizsgálata egyre fontosabbá válik, mivel az irányultság számításának pontatlansága több edénybe sorolja a számunkra lényeges élpontokat. A TS -térben már csak ezek az élpontok kerülnek akkumulációra, majd mindkét térrészben megtörténik a lokális maximumok meghatározása, amelyek egyedi ponthármasokként hipotéziseket alkotnak majd. Jelen implementáció a pontok akkumulálására Bresenham vonalrajzoló algoritmusát (J. E. Bresenham, 1962) alkalmazza (5.2 ábra). A távlatpont képének meghatározása lineáris regresszióval közelíti az adott hipotézis pontjait. Ezen hipotézisek a modellépítés bemenetei, amely során az összes hipotézis alapján kiértékelésre kerül a hozzá tartozó modell pontossága egy a számított és tényleges minimumok és maximumok távolságán alapuló metrikával. A legjobb eredményt elérő modell kerül felhasználásra a mintavételi pontok meghatározásához, amely a QR-kód alacsony felbontású, szürkeárnyaltos képét generálja. A kapott képet a dekódoláshoz binarizálni kell, mintaillesztéssel meg kell

¹⁷ http://www.emgu.com/wiki/index.php/Main_Page

találni a lokalizációs mintákat és ezek alapján be kell állítani az orientációját. Ezt követően a QR-kód bináris képe rendelkezésre áll a dekódoló algoritmus számára, amely, mint egy bináris értékeket tartalmazó mátrixot feldolgozhatja azt.



7.1 ábra: Az algoritmus főbb lépései az általuk felhasznált bemenetek és generált kimenetek ismertetésével

8 TESZTELÉS

8.1 Mintacsomag és tesztelési terv

A QR-kódok tesztelésére jelenleg nincs általánosan elfogadott és publikált tesztsomag, így azt mindenkinek saját magának kell létrehoznia vagy beszerezni valamilyen forrásból. A tesztsomag reprezentatív volta ugyanakkor nagyon fontos faktor nem csak az algoritmus értékelésére, hanem az egyes paraméterek finomhangolása tekintetében is, hiszen ezek azok a minták, amelyek a valós felhasználást hivatottak reprezentálni. Szükséges, hogy a mintacsomag lefedje a legalapvetőbb orientációkat és perspektívákat, méreteket, fényviszonyokat, zajokat és egyéb torzító tényezőket. Az algoritmus eddigi implementációit felvonultató szakirodalmak alapján [5][13][16] a 8.1 ábrán látható kategóriák kerültek meghatározásra az algoritmus tesztelésére.

Tesztelési metrikák

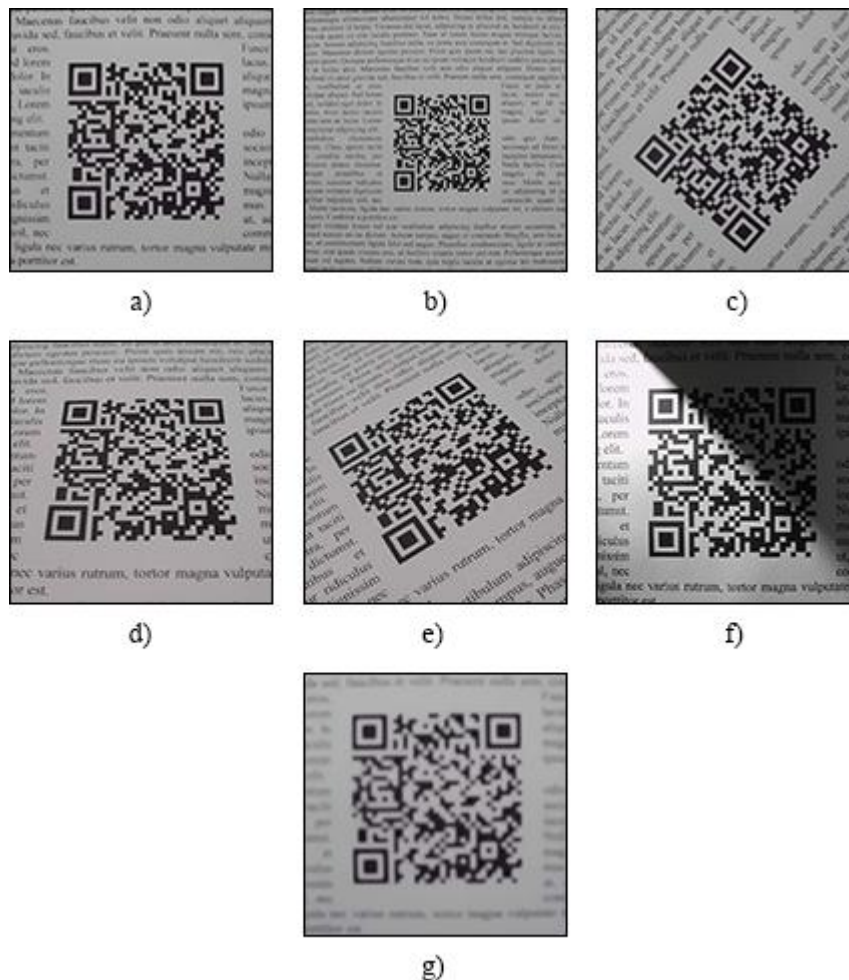
Az algoritmus eredményességének értékelése több módon is lehetséges attól függően, hogy alkalmazunk-e dekódoló modult vagy sem. A feldolgozott szakirodalmak egy metrikát használt [5][16], amelyet *MX* metrikának nevezett. Ez a metrika megadja azoknak a QR-kód mintáknak a mennyiségét százalékban, amelyeket legalább *X* százalékban sikerült pontosan detektálni. A metrika kihasználja azt, hogy a tesztelési mintákhoz felhasznált QR-kód ismert és ezáltal az algoritmus által szolgáltatott alacsony felbontású, normalizált QR-kód kép modulonként összevethető vele. Az *MX* metrikák közül hármát alkalmaztak, mégpedig az *M95*, *M99* és *M100* metrikákat és eredményeiket is ezek alapján közölték. A problémakör képfeldolgozási részében érdekelteknek ez a megközelítés elégséges. a valós felhasználáshoz azonban közelebb áll a dekódolást is alkalmazó tesztelés, amely a QR-kód hibajavító szintjétől függően képes vagy nem képes dekódolni az ugyanakkora *MX* metrikájú QR-kódot. Továbbá lényeges, hogy az *MX* metrika nem szolgáltat információt arról, hogy a hibásan detektált modulok a QR-kódban hol találhatóak és így arról sem, hogy jellegzetes mintát alkotó vagy adatbitről van-e szó. Az utóbbi helyzet pedig lényeges a QR-kód dekódolhatóságának szempontjából.

Tesztelési terv

A felvonultatott érvek és ellenérvek alapján a készülő mintacsomag tartalmazni fog különböző verziójú, illetve hibajavítási képességű QR-kódokat minden fentebb említett kategóriában, hogy a későbbiekben lehetőség legyen a dekódolhatóság széleskörűbb vizsgálatára. A kezdeti teszteléshez és a paraméterek elfogadható detektálási szinteket produkáló értékeinek becsléséhez elegendő az *MX* metrika, így első körben a tesztelés ennek segítségével fog zajlani. A tesztelés automatizálásához minden minta

metaadatokkal lesz ellátva, hogy a valódi QR-kóddal való összevetés és az eredmények különböző kategóriákhoz kapcsolása lehetséges legyen emberi beavatkozás nélkül is.

A díszes és egyedi tervezésű QR-kódok egy része bár nagyon céltudatosan van megtervezve, hogy a detektálást lehetővé tegye, egyes változatok olyan tulajdonságokban térnek el a klasszikus QR-kódoktól, hogy azok az algoritmus alapvető feltételezéseit nem elégítik ki. Ilyenek például a kör alakú modulokat tartalmazó díszes QR-kódok.



8.1 ábra: QR-kód tesztesetek **a)** kép szignifikáns részét elfoglaló **b)** kép kis részét elfoglaló **c)** elforgatott **d)** különböző perspektívákból rögzített **e)** különböző perspektívákból rögzített és elforgatott **f)** egyenlőtlen megvilágítású **g)** elmosódott

A tesztelés további fontos aspektusa az azonos és más módszereket használó implementációkkal való összevetés. Ez a már említett általánosan elfogadott mintacsomag hiányában némiképp bonyolult, azonban a tendenciák megállapítására így is lehetőség van. Továbbá a szabadon hozzáférhető implementációk letesztelhetők a saját mintacsomaggal is, ami sokkal reprezentatívabb összevetést tesz lehetővé. A

jelenleg elterjedtebb és sokak által hivatkozott szoftverek a ZBar¹⁸ és az ZXing¹⁹, amelyek magát a visszafejtést is tartalmazzák így az ezekkel való összevetés erősen ajánlott.

A paraméterek finomhangolása és az egyes részfeladatokra alkalmazott módszerek összevetése szintén része lesz a dolgozatnak, ahol ezek a paraméterek és módszerek leginkább a sikeres detektálások arányán keresztül kerülnek összevetésre, azonban néhány esetben érdemes lehet megvizsgálni a feldolgozási időre gyakorolt hatásukat is. A vizsgálni kívánt paraméterek és módszerek a következők:

- éldetektálási módszer,
- éldetektálási küszöb,
- a HOG-ban lévő edények száma (a szögtartomány felosztásának felbontása),
- a $\mathcal{TS}$ -térben akkumulálásra kerülő élpontok mennyisége,
- a $\mathcal{TS}$ -tér szélessége,
- a modell építés különböző küszöbszámai
- az adaptív küszöböléshez használt módszerek és küszöbök.

8.2 Az algoritmus eddigi eredményeinek ismertetése

Az alábbiakban összegyűjtésre kerültek a módszert implementáló szakirodalmak által elért eredmények és összevetésük a már említett ZBar és Xzing szoftverekkel. A táblázatok adatai a sikeres detektálások százalékos arányát adják meg a mintahalmaz egészéhez képest.

Kategória / Módszer	PClines [13]	XZing	ZBar
Egyszerű	94,4	88,9	93,3
Elforgatott	95,1	80,7	46,9
Perspektíva	51,7	35,2	41,1
Pers. + elforg.	39,0	19,5	22,9
Megvilágítás	83,7	20,9	76,7
Elmosódott	0,0	2,2	15,5
Összesített	60,7	41,2	49,4

8.1 táblázat: A [13] szakirodalom által elért eredmények és összehasonlításuk a Zbar és XZing szoftverekkel.

¹⁸ <http://zbar.sourceforge.net/>

¹⁹ <http://code.google.com/p/zxing/>

Az eredményekből látható, hogy a nem egységes mintacsomag és metrika, tehát, hogy mit tekintünk sikeres detektálásnak, megnehezíti az összevetést a más módszerek által elért eredményekkel. Ez a jelenség leginkább a 8.1 és 8.3 táblázatok ZBar eredményein figyelhető meg.

Kategória / Módszer	PClines [5]	PClines ²⁰	XZing
Egyszerű	93,8	94,7	94,4
Elforgatott	84,8	88,2	58,2
Perspektíva	81,2	93,4	72,7
Pers. + elforg.	82,5	90,8	9,8
Összesített	85,5	91,8	58,8

8.2 táblázat: Az [5] szakirodalom által elért eredmények és összevetésük egy másik PClines implementációval és az XZing szoftverrel

A különböző PClines megvalósítások eredményei arra engednek következtetni, hogy ez a megközelítés felveszi a versenyt a manapság elterjedt szoftverekkel, illetve, hogy az átlagosnál jobban kezeli a különböző perspektívákból rögzített QR-kódokat.

Kategória / Módszer	PClines [16]	ZBar
Egyszerű	90,6	90,6
Elforgatott	88,9	98,2
Perspektíva	94,7	57,9
Pers. + elforg.	88,0	60,0
Megvilágítás	69,2	46,1
Elmosódott	52,9	67,9
Összesített	80,5	74,8

8.3 táblázat: A [16] szakirodalom által elért eredmények és összevetésük a ZBar szoftverrel

A leggyengébb eredményeket a felvonultatott módszerek az egyenetlen megvilágítású és az elmosódott QR-kódok esetében produkálják. Mindkét esetben olyan problémákról van szó, amelyek részben az éldetektálásból erednek.

Az implementáció további fejlesztése során hasonló eredményekre számíthatunk, mint amilyeneket a felvonultatott PClines implementációk produkáltak.

²⁰ M. Dubská és mások. A [16] szakirodalom algoritmusának egy másik változata.

9 ÖSSZEFOGLALÁS

A PClines algoritmus egy sok különböző technikát, a párhuzamos koordináta-rendszerrel eredeti módon vegyítő módja a QR-kód detektálásnak. Az módszer leírásából kitűnik, hogy általánosítható más mátrixkódok detektálására is, ami az egyforma méretű, négyzet alakú modulokat használó mátrixkódok esetén a legegyszerűbb. Az algoritmus magán viseli az éldetektálási módszerek problémáit, mivel ez az első és egyben kritikus része a detektálás minőségének.

Az algoritmusra hivatkozó irodalmakból kitűnik, hogy a 8.2 fejezetben, a tesztelésénél vázolt szituációk egy részével jól megbirkózik, vannak azonban tulajdonságok, amelyek nagyban megnehezítik, olykor pedig gyakorlatilag lehetetlenné teszik a sikeres detektálást. Az eredményességet negatív irányba leginkább befolyásoló jelenségek az elmosódottság, amely már az éldetektálásnál komoly problémákat okoz, a QR-kód aránya a kép egészéhez képest, amely megnöveli a hibásan detektált párhuzamosok valószínűségét valamint a QR-kód hullámos felszínen történő rögzítése, mivel az algoritmus által használt egyenletek erre nincsenek felkészítve. A legtöbb mátrixkód detektáló módszerhez hasonlóan a PClines sem képes hatékonyan kezelni a képen található több kódot, mert nem képes a párhuzamosokat egyértelműen egy kódhoz rendelni [5]. Ez bizonyos előszűrésekkel kezelhető, azonban az ilyen előszűréseket más módszerek is alkalmazhatják, hogy meghatározzák az egyes kódok helyét a képen és külön hajtják végre rajtuk a detektálást, így ez a tény nem csak a PClines algoritmust erősíti [16].

A felvonultatott lépések sokasága arra enged következtetni, hogy a módszer esetleg túl lassú a valós idejű felhasználáshoz, azonban a megvalósítások bizonyítják, hogy az algoritmus megfelelően teljesít ilyen feltételek mellett is [5] [13] [16].

A dolgozathoz kapcsolódó kutatás jövőbeli céljaihoz tartozik az algoritmus implementációjának további fejlesztése a 7.1 fejezetben ismertetett módon az algoritmus 7.1 ábrán megfigyelhető logikai felépítése alapján, egy mintacsomag már folyamatban lévő létrehozása a végleges implementáció tesztelésére, és az egyes paramétereknek és részfeladatokra alkalmazható módszereknek az algoritmus hatékonyságára gyakorolt minél széleskörűbb elemzése. Az algoritmus összevetésre kerül néhány korábban ismertetett, szabadon hozzáférhető megoldással, illetve a 8.2 fejezetben látható táblázatos eredmények tendenciáival is.

10 IRODALOMJEGYZÉK

- [1] S. Arnould, G. Awcock, R. Thomas: *Remote bar-code localisation using mathematical morphology*. Seventh International Conference on Image Processing and Its Applications, 2. kötet, o. 642-646, 1999.
- [2] L. Belussi, N. Hirata: *Fast QR code detection in arbitrarily acquired images*. Conference on Graphics, Patterns and Images, SIBGRAPI 2011, o. 281 – 288, 2011.
- [3] Denso Wave Inc.: *QR Code information homepage*, [online]. <http://www.qrcode.com/en/index.html>, 2010. [2012.11.04]
- [4] M. Dubská, A. Herout, J. Havel: *PClines - line detection using parallel coordinates*. Computer Vision and Pattern Recognition (CVPR), IEEE Conference, o. 1489 – 1494, 2011.
- [5] M. Dobrovolný: *Detekce a rozpoznání maticového kódu v reálném čase*. Brno Institute of Technology, 2012, Semestral project.
- [6] R. O. Duda, P. E. Hart: *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Comm. ACM. 15. kötet, o. 11 - 15., 1971.
- [7] R. C. Gonzales, R. E. Woods: *Digital Image Processing*. 2nd, Prentice Hall, 2001, ISBN 0-201-18075-8.
- [8] R. Hartley, A. Zisserman: *Multiple View Geometry in Computer Vision*. 2nd, Cambridge University Press, 2003, ISBN 0521 54051 8.
- [9] P. V. C. Hough: *A Method and Means for Recognizing Complex Patterns*. US Patent 3,069,654, 1962.
- [10] H. Hu, W. Xu, Q. Huang: *A 2D barcode extraction method based on texture direction analysis*. Fifth International Conference on Image and Graphics, ICIG '09, o. 759 – 762, 2009.
- [11] A. Inselberg: *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, 2009, ISBN 978-0-387-21507-5.
- [12] International Organization for Standardization: *Information Technology - Automatic Identification and Data Capture Techniques - QR Code 2005 Bar Code Symbol Specification*. ISO/IEC 18004:2006, 2006.
- [13] M. Kaluža: *Detection, Localization and Decoding QR Code in an Image*. Brno Institute of Technology, 2012, Master's thesis.
- [14] E. Ohbuchi, H. Hanaizumi, L. A. Hock: *Barcode readers using the camera device in mobile phones*. Cyberworlds, International Conference, o. 260 - 265, 2004.

- [15] T. J. Soon: *QR Code*, Synthesis Journal 2008, Information Technology Standards Committee, 2008, o. 59 - 78, ISSN 0219-4767.
- [16] I. Szentandrás, M. Dubská, A. Herout: *Fast Detection and Recognition of QR codes in High-Resolution Images*. Graph@FIT, Brno Institute of Technology, 2012.
- [17] Wikipedia: *Cross-ratio*, [online].
http://en.wikipedia.org/wiki/Cross_ratio, 2012. [2012.11.07]
- [18] Wikipedia: *Histogram of oriented gradients*, [online].
http://en.wikipedia.org/wiki/Histogram_of_oriented_gradients, 2012.
[2012.11.04]
- [19] Wikipedia: *QR Code*, [online].
http://en.wikipedia.org/wiki/QR_code, 2012. [2012.11.04]