

Gépjárművezetés biztonságnövelése képfeldolgozás alkalmazásával



Készítette: Bencsik Blanka

Felkészítő tanár: Kőrösi Gábor

Iskola: Bolyai Tehetséggondozó Gimnázium és Kollégium

Évfolyam: 4. évfolyam

2016/17, Zenta

1. TARTALOMJEGYZÉK

1. Tartalomjegyzék
2. Bevezető
3. Képfeldolgozás
 - 3.1. Általánosan a képfeldolgozásról
 - 3.2. Digitális képek a számítógépben
 - 3.3. Hisztogram transzformációk
 - 3.3.1. Hisztogram
 - 3.3.2. Küszöbölés
 - 3.4. Konvolúciónalapú műveletek
 - 3.4.1. Konvolúció
 - 3.4.2. Zajszűrés
 - 3.4.3. Élkiemelés
4. Program leírása
 - 4.1. Előfeldolgozás
 - 4.2. Szegmentálás
 - 4.3. Sávtartás
 - 4.3.1. Hough-transzformáció
 - 4.3.2. Vonal helyzetének matematikai számítása
 - 4.4. Kanyar felismerése
 - 4.5. Szaggatott vonal felismerése
5. Összefoglaló
6. Irodalomjegyzék

2. BEVEZETŐ

A közlekedés mindennapjaink szerves részét képezi. A nagyvárosi tömegközlekedés mellett szinte minden család rendelkezik saját gépjárművel, ami bármikor rendelkezésére áll, ha gyorsan és egyszerűen el szeretne jutni valahova. Az autó megjelenése több, mint 100 évvel ezelőttre tehető, s vitathatatlan, hogy gyökeresen megváltoztatta az életet, megkönnyítette az emberek mindennapjait. A könnyítés mellett azonban a veszélyt is magával hozta. A technológia rohamosan fejlődött, s ez mára egy rendkívülien rohanó világot eredményezett. A veszélynek a Föld minden pontján ki vagyunk téve, de ez jelentősen hatványozódik az utakon, ahol a sebesség és a száguldás pillanatok alatt vezethet tragédiához. A világon évente átlagosan 1,25 millió ember veszti életét közúti balesetben. [10] Ennek a hatalmas számnak viszont csupán néhány százaléka adódik az út rossz minőségéből, a gépkocsi meghibásodásából vagy a rossz időjárási viszonyokból. A balesetek több mint 90 %-a az emberi hibáknak, marasztalásoknak tudható be. [14] Több informatikai cég fejlesztőinek fejéből is kipattant az ötlet, hogy ha nem emberek vezetnék az autót, akkor nem is történne ennyi baleset. Így ma már több világvezető cég is dolgozik az önvezető autó kifejlesztésén, mint amilyen a Google, Tesla vagy a Honda. Az ötlet alacsonyabb szinten is elterjedt. Némely egyetemeken a hallgatók számára szerveznek önvezetőautó-versenyt, ahol követelmény a jármű vonal menti haladása, sebességszabályozás, közlekedési táblák felismerése. Az önvezető autónak több fokozata van, kezdve attól, hogy milyen fokozaton segíti a vezetőt a vezetésben, mennyire helyettesíti azt. Igazi önvezető autóról akkor beszélhetünk, ha a gépjármű bármiféle emberi beavatkozás nélkül képes haladni a közúton. Míg az utóbbi ugyan már létezik, ám csupán tesztelés és további fejlesztés alatt áll, addig a különböző közlekedést segítő felszereléseket már ma is megvásárolhatják az érdeklődők (pl. tempomat, parkolást segítő rendszer, más közúti haladást könnyítő rendszerek, stb.).

Részben az utóbbi, másrészt a megdöbbentő elhalálozási szám, az informatika kedvelése és a képfeldolgozás iránti érdeklődésem együttesen motivált arra, hogy létrehozzak egy programot, mely könnyíti az autóvezető számára a vezetést, s ezáltal csökkentheti a közúti baleset bekövetkezésének valószínűségét. Az elkészült program képes megállapítani a gépkocsira rögzített kamera által készített felvételtől, hogy az úton hol vannak az sávjelző vonalak, hol halad az autó, helyesen tartja-e a sávot, merre kanyarodik az út, valamint, hogy engedélyezett-e az előzés, azaz szágatott-e a vonal. Mindezekről a vezető grafikus formában kap értesítést. Az elképzelés kivitelezéséhez képfeldolgozást alkalmaztam. A számítógépes képfeldolgozás az alkalmazott matematika, elektronika és a számítástechnika egyik legrohamosabban fejlődő ága. Képfeldolgozás során egy képből számítógép segítségével kinyerjük a környezetünkben lévő információkat, feldolgozzuk, majd kiértékeljük azt. A program elkészítéséhez a Microsoft Visual Studio 2010 szoftvercsomagot, valamint az OpenCV képfeldolgozási könyvtárat használtam. Az utóbbi alapszintű megismerése megkövetelte, hogy munkám elkezdése előtt alapos tudást szerezzek a képfeldolgozás terén.

3. KÉPFELDOLGOZÁS

3.1. Általánosan a képfeldolgozásról

A számítógépes képfeldolgozás az alkalmazott matematika, elektronika és a számítástechnika egyik legrohamosabban fejlődő ága. Képfeldolgozás során egy képből számítógép segítségével kinyerjük a környezetünkben lévő információkat, feldolgozzuk, majd kiértékeljük azt. Megjelenése a számítástechnika megjelenésének idejére tehető (1950-es évek). Eleinte a csillagok fényességének megállapítására, vírus felépítésének megállapítására egy mikroszkópos kép alapján, valamint térképek vizsgálatára használták. [2] Alkalmazása ma már szinte mindenhol megtalálható, kezdve a mezőgazdaságnál, az orvostudományon, iparon, gyártási folyamatokon, meteorológián keresztül egészen a bűnügyig, ahol vagy a fejlesztést vagy a mindennapi munkát segíti. A képfeldolgozási algoritmusokkal valóban tucatnyi különböző problémát lehet megoldani, különböző területeken, ez által millióféleképpen lehetnek, alapvető felépítésük azonban mégis megegyezik.

A számítógépes képfeldolgozás alaplépései a következők:

1. Képkészítés.

Mindazon műveletek összessége, amíg a kép eljut a számítógép memóriájába vagy valamely háttértárolójába. Ezek a képek számos forrásból származhatnak. Digitális képet készíthetünk digitális fényképezővel, kamerával, de bármilyen más kép is digitalizálható szkennel segítségével.

2. Előfeldolgozás.

Mielőtt a képet bármilyen jellegű feldolgozásra bocsátanánk, érdemes azt a lehető legjobb állapotba hozni, hogy a későbbiekben a képen lévő zaj, árnyékok, kontraszthibák ne eredményezzenek tévedést az azonosítás során. Ezért használatosak a különböző képjavítási műveletek, melyek célja a kép minőségének feljavítása (pl. szűrés, élesség javítása, stb.), esetleg vágás, nagyítás.

3. Szegmentálás.

A kép felbontása alkotó részekre. Legalapvetőbb művelet az objektumok elválasztása a háttértől.

4. Azonosítás.

A képen található objektumok felismerése. A szegmentálás során könnyen felismerhetővé vált objektumokból azonosítás. Válogatás, hogy ezek közül melyik adat a fontos számunkra.

5. Utófeldolgozás.

A felismert objektumokból további következtetések levonása, szemléltetése. [4]

A továbbiakban azokról a képfeldolgozási műveletekről lesz szó részletesen, melyeket a munkám elkészítése során alkalmaztam.

3.2. Digitális képek a számítógépben

"A kép egy minimum kétdimenziós információhordozó. Egy digitális kép képpontokból áll, minden egyes képpont a többbitől független, saját információt tartalmaz." [4] A képeket a valóságban is egy kétdimenziós mátrixként képzeljük el, s a számítógép is pontosan így kezeli, melyben a képpontok (pixelek) sorokba és oszlopokba vannak rendezve. Egy kép szélességét az határozza meg, hogy egy sorában hány képpont van, magasságát pedig az, hogy egy oszlopát hány képpont alkotja. Egy képpont információja lehet bináris (fekete vagy fehér, 1-es vagy 0-s, igaz vagy hamis) és intenzitás érték vagy szürke árnyalat (16, 256, vagy egyéb számú fokozat a fekete és fehér között). Egy kép állhat egy vagy több csatornából. Ez azt jelenti, hogy hány alapszín alkotja. Az emberi szemmel érzékelhető színeket három csatornával lehet leírni, ehhez hasonlóan a színes képernyő is RGB kódolást használ (Red-Green-Blue, azaz piros-zöld-kék). [4] Egy 8 bites digitális kép egy képpontjának színét úgy kell elképzelnünk, hogy minden alapszín adott erősségekben van jelen (256 különböző értékben), s ezeknek a színeknek keveréke adja a képpont valódi színét.

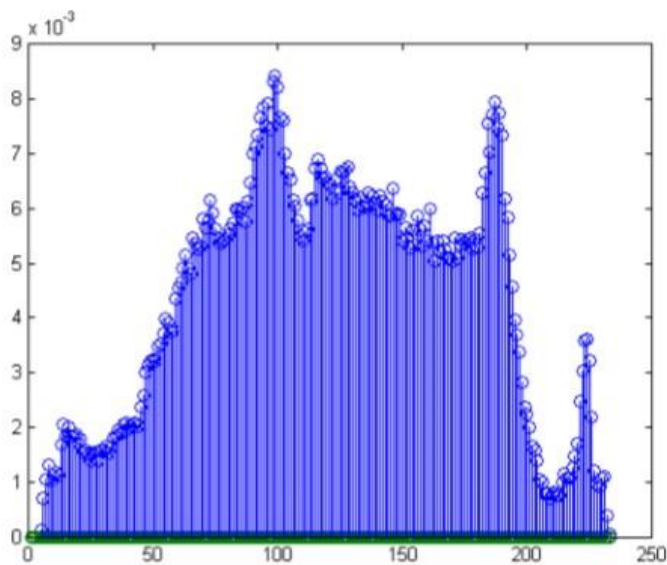
Általában a képfeldolgozásban, amikor nem a színeken van a hangsúly, az egyszerűségkedvéért szürke képpel dolgoznak. Egy többcsatornás színes képet úgy tudunk fekete-fehér képpé alakítani, hogy átlagoljuk egy pixel csatornáinak értékét. Tehát, ha egy RGB képről van szó, összeadjuk az adott pixelre vonatkozó mindhárom csatorna intenzitását, elosztjuk a csatornák számával, és ezt adjuk értékül a pixel minden csatornájának.

3.3. Hisztogram transzformációk.

3.3.1. Hisztogram

A korábbiakban már szó volt a képpont intenzitás értékéről. Ezt a képpont világosságának is nevezhetjük. "A hisztogram a kép világosságának sűrűségfüggvénye. Értelmezési tartományát a kép világosság skálája képezi." [1] Egy kép hisztogramjának készítésekor összeszámoljuk, hogy adott világossághoz hány képpont tartozik. "A normálás érdekében elosztjuk azt a kép pontjainak számával. Így lesz a hisztogram integrálja a teljes tartományra egységnyi, ami a sűrűségfüggvény fontos tulajdonsága." [1]

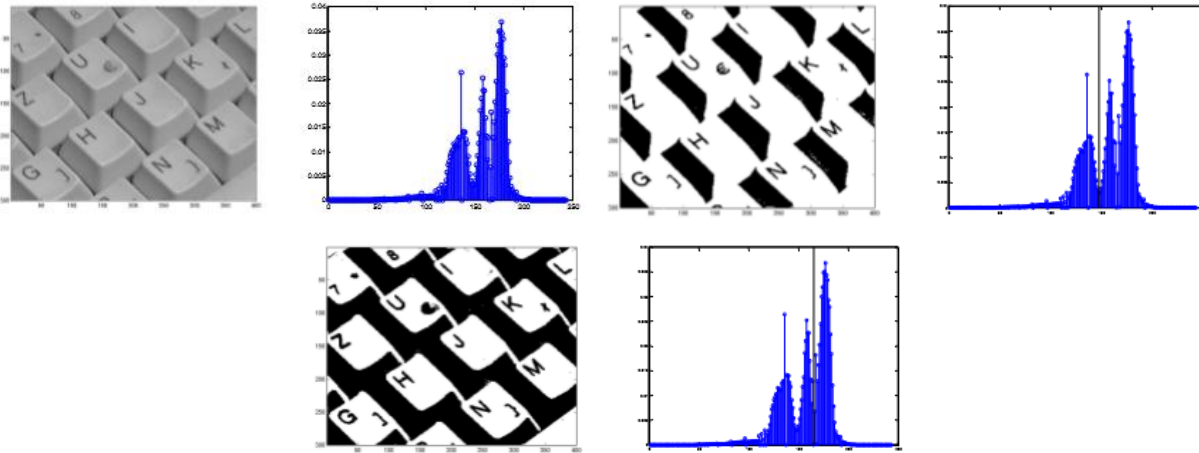
Értelemszerűen egy kép akkor jó, helyesen exponált, ha a hisztogramján megjelenik az egész, adott képre vonatkozó világosságtartomány, és az minden pontjában hasonló értéket vesz fel (3.1. ábra). Ha egy kép hisztogramjában túlnyomóak a világos tartományba eső értékek, akkor a kép túlexponált (túl fényes), ellenkező esetben alulexponált (túl sötét). Ez egyik esetben sem jó, hiszen ilyenkor a kép egyes részei egybeolvadnak, és a fontos információk nem kivehetőek. Ezeket azonban különböző hisztogram-transzformációkkal javítani tudjuk.



3.1. ábra. Példa a fekete-fehér kép hisztogramjára. [1]

3.3.2. Küszöbölés

Ez a művelet egyben a képszegmentáláshoz tartozik, ami nem más mint a kép felosztása jelentőséges részeire. Legegyszerűbb esetben ez 2 régiót jelent: a tárgy részt és a háttérét. [2] E művelet végrehajtásához is a hisztogramot hívjuk segítségül. "Egyik lehetséges megoldás a világosság alapján elvégzett osztályozás. Egy alkalmas világosság küszöb alapján bináris képpé alakíthatjuk a szürke árnyalatos képet." [1] Egyszerűsítve tehát, a hisztogram alapján meghatározunk egy küszöböt. Amely képpontok ez a küszöb alá esnek azokat feketére "színezzük", amik pedig fölé, azokat fehérre. De vajon hogyan határozzuk meg azt az ideális küszöböt? A legelterjedtebb és legmegbízhatóbb módja a küszöbkeresésnek a manuális választás. [2] A hisztogramról ránézésre határozzuk meg az optimális értéket. A küszöbölés elvégzése után tehát különböző képeket kaphatunk attól függően, hogy hova tesszük a küszöbértéket (3.2. ábra). " Emellett félig automata eljárások is léteznek a küszöbérték meghatározására." [5] . Ennél pontosabb kijelölést tesz még lehetővé a hisztogram matematikai elemzése. [1]

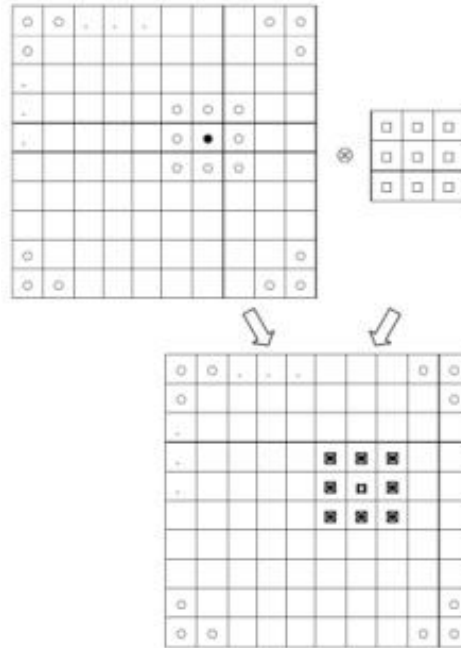


3.2. ábra. Az eredeti kép, a eltérés a szegmentált képek között, a küszöbérték megválasztásától függően, és a hozzájuk tartozó histogramok. [1]

3.4. Konvolúciónalapú műveletek

3.4.1. Konvolúció

A konvolúció egy lokális képfeldolgozási művelet. " A lokális műveletek eredménye nem csak a képpont eredeti világosságától, hanem a szomszédos képpontoktól is függ. A képpont új világosságát a transzformációnál figyelembe vett környezet mérete és a kijelölt művelet határozza meg. Konvolúción a képpont és környezete világosságának súlyozott összegét értjük. A súlyozó együtthatókat a konvolúció operátora tartalmazza. " [1] A konvolúció operátorát egy mátrixként kell elképzelni. Műveletvégzéskor ennek a maszknak a közepét helyezzük az adott képpontra (az operátor rendszerint páratlan számú sort és oszlopot tartalmaz, így van középső eleme), és összeszorozzuk annak környezetével (3.3. ábra). Ám, ez a művelet nem mátrixszorzást, hanem elemenkénti műveleteket jelent.[1] Az eredményt pl. 3x3-as maszk esetében 9 képpont információjából kapjuk, ami most egy képpont értéke lesz. Ezért mondjuk, hogy egy képpont új világosságát a környezete befolyásolja. A konvolúciós műveletet a kép összes pontjára alkalmazzuk. A kép szélén található képpontok, melyekről "lelógna" a maszk, elhanyagolhatóak, néha azonban úgy oldják meg ezt a problémát, hogy kiszélesítik a képet a képben található színárnyalatok átlagával.



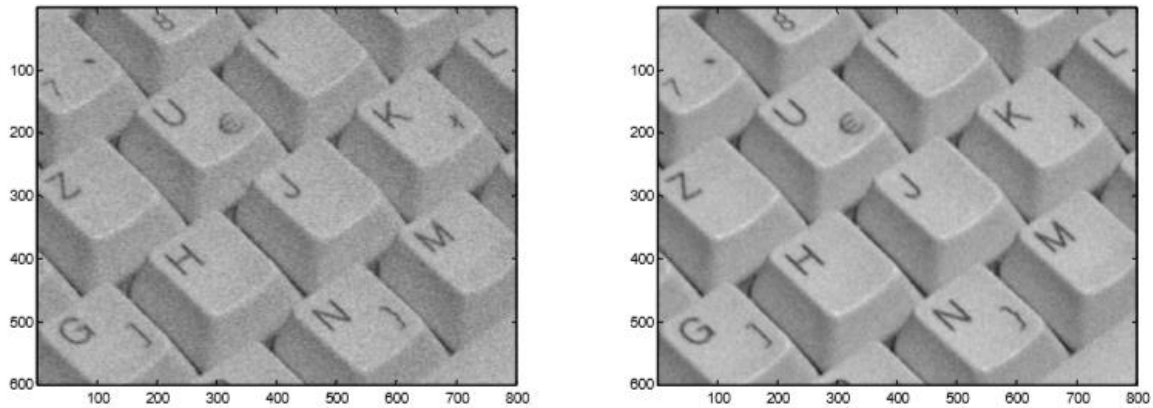
3.3.ábra. Konvolúció szemléltetése. [1]

3.4.2. Zajsűrés

A képjavítás egyik fontos feladata a képben a zaj csökkentése, mely nagyban megkönnyíti a későbbi képpel való dolgozást. A jelenlegi modern fotográfiai felszerelések ellenére is minden képben található felesleges információ, azaz zaj. Ez keletkezhet véletlenszerűen az exponálás során, a fény digitális jellé alakításakor, esetleg a kép másolásakor, stb. Ez a zaj hozzáadódik a képpont (pixel) eredeti világosságához, s ebből kifolyólag az el fog térni a környezetétől. "Ezt a különbséget csökkenthetjük, ha a képpont világosságát környezete átlagával helyettesítjük. A zajcsökkentő operátorok fontos tulajdonsága, hogy elemeinek összege egy. Ha nem így lenne, a kép átlagos világossága eltolódna." [1] A zajcsökkentő operátor egy konvolúciós maszk, mely elemeinek összege egy, tetszőleges nagyságú (általában 3x3 vagy 5x5), mellyel egyenként végigszorozzuk a kép minden egyes képpontját, tompítva ezzel a zajos képpontot, amely átlagosan szétterül a környező területen, az operátor nagyságával arányosan. Szorzáskor a maszk középpontját helyezzük az adott képpontra. Az átlagolás azonban nem csak a kívánt képponton megy végbe, hanem a kép összes pixelén, ezért az élek is eltompulnak. Minél erősebb a zajsűrés, annál lágyabb lesz a kép.

$$z_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}; \quad z_2 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

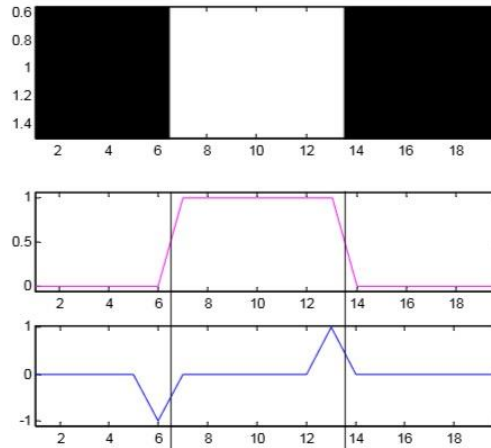
3.4. ábra. Példák 3x3-as zajszűrő operátorokra. [1]



3.5. ábra. Kép zajszűrés előtt és után. [1]

3.4.3. Élkiemelés

Sok esetben egy képen lévő tárgy nem egészére, csupán annak alakjára, körvonalára van szükségünk. Ezért végzünk a képen élkiemelést. "Digitális képen élek ott találhatóak, ahol a szomszédos pixelek között nagy az intenzitáskülönbség." [3] "A különbségképzés alkalmas az élkiemelésre, hiszen homogén foltoknál nulla, és annál nagyobb, minél nagyobb a világosságok közötti különbség. A vízszintes és függőleges irányú változásokat parciális differencia képzésével emelhetjük ki. (3.6. ábra) Függőleges éleknél például vízszintes irányban tapasztalunk világosság változást." [1]



3.6. ábra. Függőleges élek és a vízszintes irányú parciális differencia. Vízszintesen 20 pixelből álló képrészlet, világosság , első differencia. [1]

Gradiens számításon alapuló élkimielés

A gradiens egy folytonos függvény adott pontjában az a vektor, melynek iránya a legnagyobb változás iránya, intenzitása pedig a legnagyobb változás intenzitása az adott pontban. [2]

"Gradiens:

$$\nabla f = \frac{\partial f}{\partial x} \vec{i} + \frac{\partial f}{\partial y} \vec{j}$$

, ahol $\vec{i}$ és $\vec{j}$ az egységvektorok, x és y pedig az irányok." [2]

"A gradiens számításon alapuló élkimielési módszerek közös jellemzője, hogy két, egymásra merőleges irányban számol gradienst, mindkettőnek megvan a saját operátora." [1] Az élkimielőoperátor hasonlóan a zajszűrő operátorhoz egy mátrix, viszont itt az elemek összege 0. Többféle ilyen operátoris létezik: a Roberts operátor egy 2x2-es maszk, ami az átlók irányában képez különbséget, míg a Prewitt és a Sobel operátorok 3x3-as nagyságúak, és vízszintes és függőleges irányban számolják a differenciát.

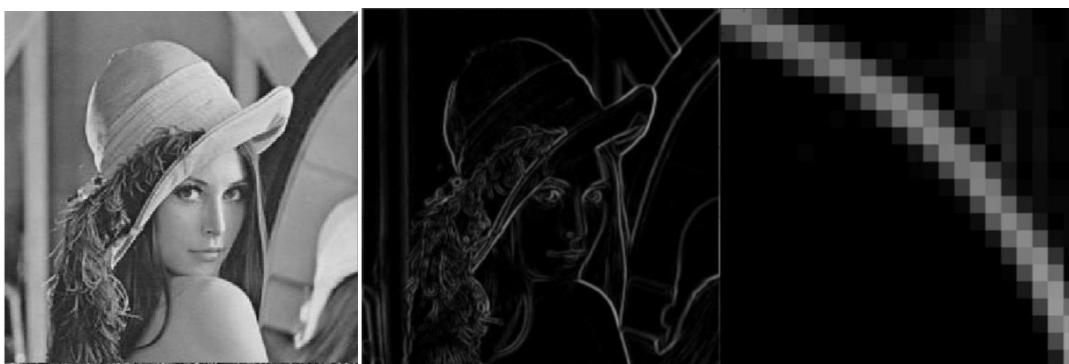
$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

3.7. ábra. Sobel operátorok. [3]

A két egymásra merőleges irányú differenciából a következőképp határozható meg a gradiens nagysága és iránya:

$$G = \sqrt{G_x^2 + G_y^2} \quad , \quad \varphi = \arctg \frac{G_x}{G_y} . [3]$$

A Sobel és Prewitt operátorok esetében, 3x3-as méretéből kifolyólag egy 3 pixeles vastagságú élet kapunk (3.8. ábra).

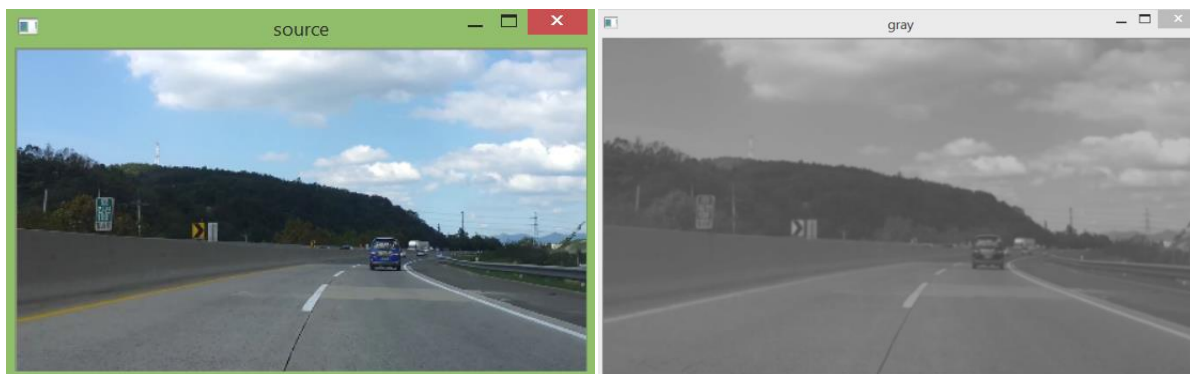


3.8. ábra. Eredeti kép, Prewitt operátorral végzett éldetektálás eredménye és az él vastagsága.[1]

4. PROGRAM LEÍRÁSA

4.1. Előfeldolgozás

A program dolgozhat élesben, webkamerás felvétellel, de akár előzőleg rögzített videóval is. Mindkét esetben a videót képkockákra bontja, s minden képen elvégzi a programkódba foglalt műveleteket. Mint már korábban említésre került, a számítógép RGB színrendszerben kódolja a képeket, ahol nem kerül kifejezésre a sárga szín. Márpedig erre szükség van, hiszen az útburkolati jelek nemcsak fehérek, sárgák is lehetnek. A program ezért a képet HSV színtérbe konvertálja. A HSVszínrendszer a következő komponensek alapján határozza meg a színt: színárnyalat (hue), szaturáció (saturation), ami a fehértől való eltérését mutatja, valamint a világosság (value, lightness), ami a szín feketétől való eltérését mutatja. [9]A konvertálás elvégzése után a program szürke képpé alakítja a meglévőt. Ekkor a sárga vonal is fehérebb árnyalatú lesz, mintha RGB képpel dolgoznánk.

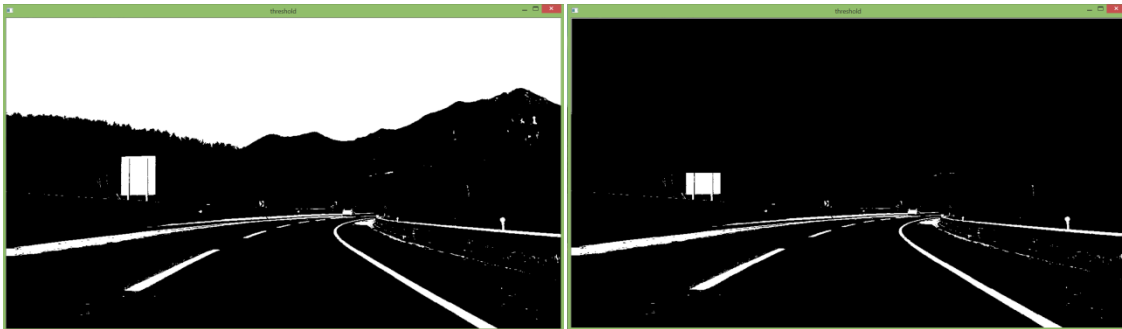


4.1. ábra. Az eredeti kép és a HSV színrendszerből szürkévé alakított változata.

4.2. Szegmentálás

A program feladata szempontjából a képnek csupán arra a részére van szükségünk, ahol az úttest található, s azon belül is azok a fontos részek, ahol a sávjelző fehér vonalak találhatóak. A képen tehát szegmentálást kell végezni, amivel az értékes részeket elválasztjuk a háttértől. Az előbbiekben már említett globális küszöbölési módszertől eltérően, itt lokális küszöbölési módszert, azaz adaptív küszöbölést (adaptive thresholding) alkalmazunk. Ez az előzőtől eltérően nem az egész képre határoz meg egy küszöbértéket, hanem külön-külön a kép kisebb tartományaira, csökkentve így a hibalehetőséget. Adaptív küszöböléskor az adott tartományba eső képpontok közül megkeressük a legkisebbet és a legnagyobbat. A két érték különbségének meg kell haladnia egy meghatározott differenciaértéket, különben az eltérés csupán szennyeződésnek tudható be. Ekkor az adott tartományba eső képpontok mindegyike 0 értéket kap (fekete). Ha lokális tartományon belüli maximum és minimum különbsége eleget tesz a feltételnek, egy-egy képpont csak akkor nyilvánítható fehérnek, ha értéke közelebb van a maximumhoz, mint a minimumhoz.

A küszöbölést csak a kép alsó felén végezzük el, feltételezve, hogy a kép felső felében égbolt, hegyek, táj, azaz számomra jelentéktelen információ található, azt egyszerűen lefeketítjük (4.2. ábra).



4.2. ábra. A szegmentált kép és annak lefeketített felső része.

Sávfelismeréskor a sávjelölő vonalak belső széléhez illesztett vonalat keressük. Vonalfelismerést pedig csak éleket tartalmazó képen lehet eredményesen véghezvinni. Tehát nem a képen lévő objektumok egészére, csupán azoknak körvonalára van szükség, ezért a szegmentált képen élkiemelést kell végezni (4.3. ábra).



4.3. ábra. A szegmentált kép élkiemelés után.

Ezen műveletek elvégzése után a képből kinyerhetők a szükséges információk, amelyek feldolgozásra kerülnek.

4.3. Sávfelismerés

Ahhoz, hogy vizsgálni tudjuk, hogy a gépjármű helyesen halad-e az útsávban, először azt kell meghatározni, hol van az útsáv, és hol halad a gépjármű. A program ezen része ennek a problémának a feldolgozásával foglalkozik, majd az eredményről grafikus ábrázolást ad a felhasználó számára.

A képen két vonalat szeretnénk meghatározni: az útsáv két szélén elhelyezkedő sávjelző vonalakat. Ez a két vonal határolja be az útsávot, melyen a gépjármű halad. Ehhez a képfeldolgozásban a vonalfelismeréshezközismerten használt eljárást, a Hough-transzformációt alkalmazzuk.

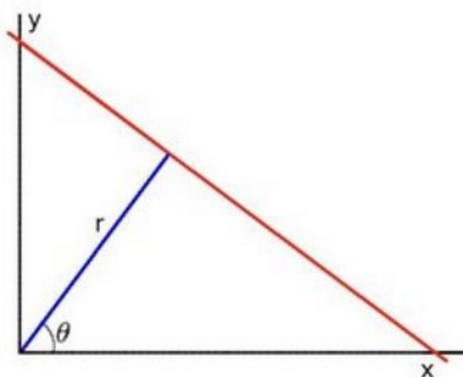
4.3.1 Hough-transzformáció

A Hough-transzformáció egy olyan módszer, mely az adott kép objektumainak olyan részhalmazát határozza meg, amelyekre közös egyenes illeszthető. [6]

A Hough-transzformáció során polárkoordináta-rendszerben dolgozunk, mely a sík minden pontját egy szög és egy távolság adattal látja el. Egy egyenes egyenlete így írható le polárkoordináta-rendszerben:

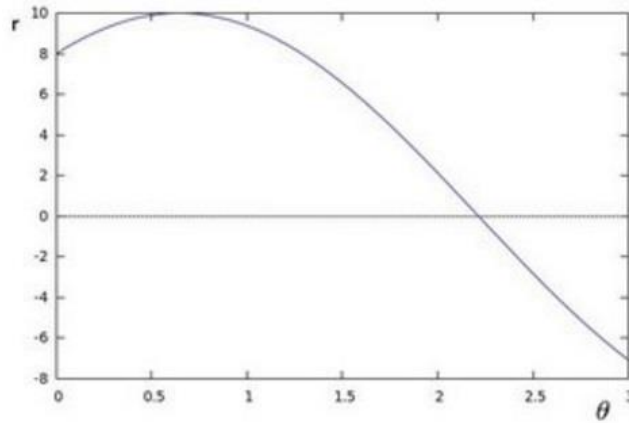
$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right) \Rightarrow r = x\cos\theta + y\sin\theta,$$

ahol r a Descartes-féle koordináta-rendszerben az egyenesre az origóból húzott merőleges, θ pedig r és az x -tengely között bezárt szög (4.4. ábra).



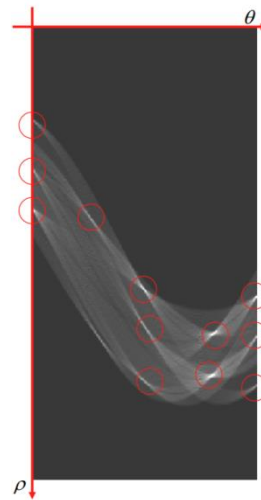
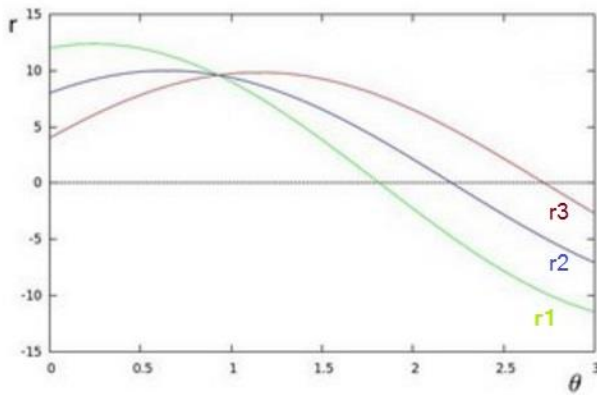
4.4. ábra. Egyenes a Descartes-féle koordináta-rendszerben. [7]

Minden (x_0, y_0) pontra a Descartes-féle koordináta-rendszerben, definiálni tudjuk a polárkoordináta-rendszerben azoknak az egyeneseknek az összességét, melyek áthaladnak ezeken a pontokon: $r = x_0\cos\theta + y_0\sin\theta$. Ez a polárkoordináta-rendszerben ábrázolva egy sinusgörbét alkot (4.5.ábra).



4.5. ábra. Descartes-féle koordináta-rendszerben lévő (x_0, y_0) ponton áthaladó egyenesek összességének ábrázolása polárkoordináta-rendszerben. [7]

A transzformációt elvégezzük a kép minden pontjára. Ha két különböző ponthoz tartozó görbék metszik egymást a polárkoordináta-rendszerben, az azt jelenti, hogy a két pont ugyanazon az egyenesen fekszik a Descartes-féle koordináta-rendszerben. Például a 4.6. ábrán jól látható, hogy x_1 , x_2 és x_3 pontok egy egyeneshez tartoznak. A görbéik (r, θ) pontban metszik egymást. Ezek a paraméterek határozzák meg azt az egyenest, amelyen a három pont a Descartes-féle koordináta-rendszerben elhelyezkedik. A 4.7. ábrán látható polárkoordináta-tér egy kép minden pontjára elvégzett Hough-transzformáció eredményét mutatja.

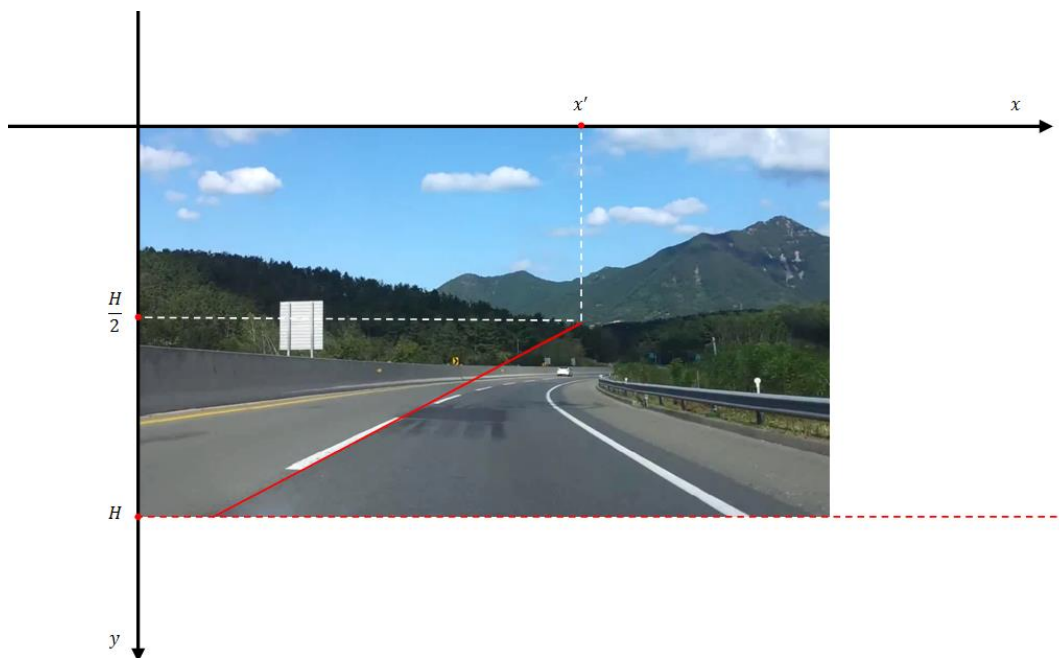


4.6. ábra. Görbék metszéspontja a polárkoordináta-rendszerben. [7] 4.7. ábra[8]

4.3.2. Vonal helyzetének matematikai számítása

A transzformáció elvégzése után számos vonalat kapunk a kép minden területéről. Számunkra viszont csak azok fontosak, melyek az útsávot jelölő jelzővonalakhoz tartoznak. Következő lépés tehát a szükséges adatok kinyerése az összes közül. A kamerabeállításoknak megfelelően a sávjelző vonalak dőlésszögei bizonyos keretek közé szoríthatóak.

A 4.8. ábrán az videofelvétel bizonyos képkockája látható a koordináta-rendszerbe helyezve, ahol H a kép magassága, azaz a kép függőleges oldalát alkotó képpontok száma. Ezen már ábrázolva van az azonosított sávjelző vonal, ám annak ábrázolásához és az azzal való további műveletek elvégzéséhez ki kell számítani annak pontos elhelyezkedését a koordináta-rendszerben.



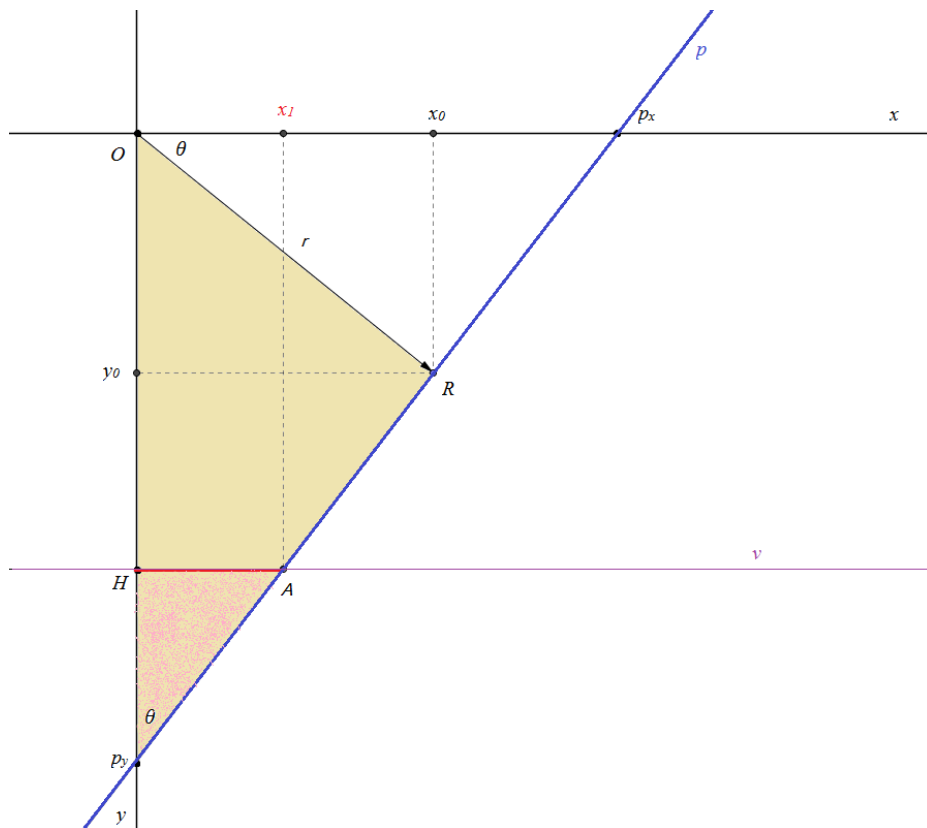
4.8. ábra. Adott képkocka koordináta-rendszerbe helyezve.

A beazonosított vonalaknak ismerjük a θ szögét és r vektorát. Ebből kiindulva számítjuk ki a derékszögű-koordinátarendszerbeli helyzetét. Az analitikus geometriai számítások szempontjából három eset különböztethető meg:

- egyenes metszéspontja a kép aljával
- bal oldali egyenes ($\theta < 90^\circ$) metszéspontja a kép tetszőleges magasságában húzott vízszintes egyenessel
- jobb oldali egyenes ($\theta > 90^\circ$) metszéspontja a kép tetszőleges magasságában húzott vízszintes egyenessel

a. egyenesek metszéspontja a aljával

Ezen metszéspont számítása mind a bal- mind a jobboldali vonal esetén ugyanúgy történik. A számítás szemléltetése az 4.9. ábrán látható. Az ábrán v egyenes jelöli a kép alját, p egyenes pedig a sávjelző vonalat. A két egyenes metszéspontja A . Mivel koordináta-rendszerben dolgozunk, az A pontot x és y koordinátájának megadásával tudjuk meghatározni: $A(x_1, y_1)$. A pont y koordinátája a kép magassága (H). Ezek után az A pont meghatározásához már csak az x koordinátájára, x_1 -re van szükség.



4.9. ábra. Bal oldali sávjelző vonal ábrázolása derékszögű koordináta-rendszerben.

Ez a távolság a $(0, H)$, $(0, p_y)$ és A pontok által meghatározott háromszögből határozható meg, ahol x_1 az AH szakasznak felel meg, p_y pedig a p egyenes metszéspontja az x -tengellyel. A háromszögben a θ szögre felírt tangens szögfüggvény a következő:

$$\tan \theta = \frac{x_1}{p_y - H} \Rightarrow x_1 = \tan \theta * (p_y - H)$$

Ebben az egyenletben a p_y az egyetlen ismeretlen. Ez az $O(0,0)$, $(0, p_y)$ és $R(x_0, y_0)$ csúcsokkal rendelkező háromszögből számítható ki. A θ szögre felírt szinusz szögfüggvény:

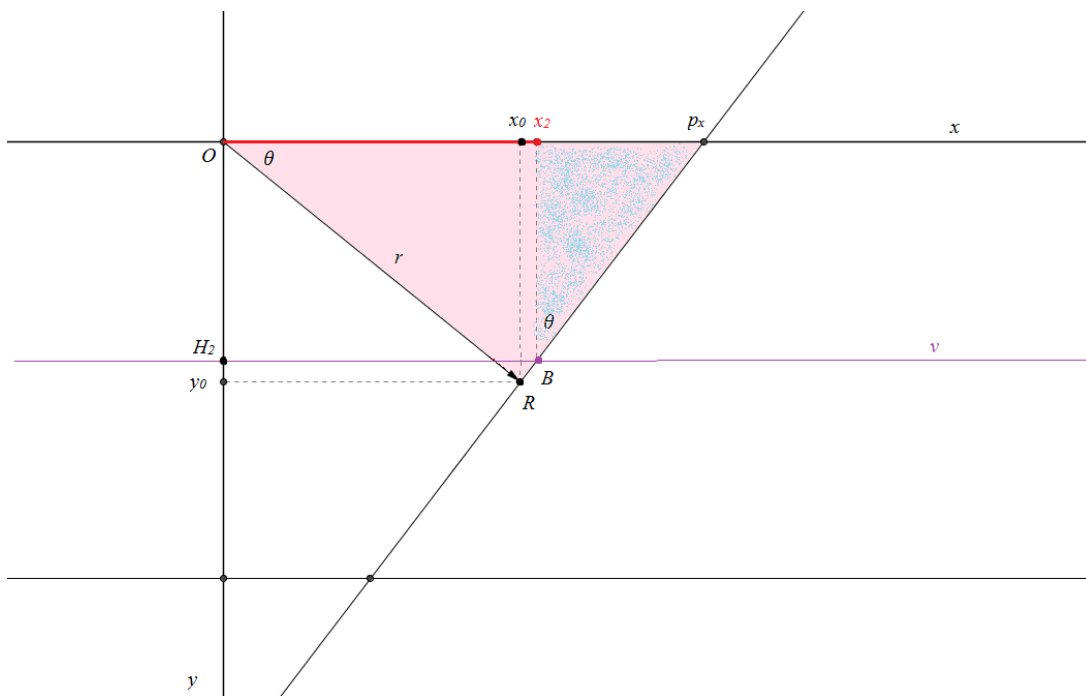
$$\sin \theta = \frac{r}{p_y} \Rightarrow p_y = \frac{r}{\sin \theta}$$

Visszahelyettesítve ez x_1 egyenletébe:

$$x_1 = \tan \theta * \frac{r}{\sin \theta - H}$$

A pont koordinátái tehát: $A\left(\tan \theta * \frac{r}{\sin \theta - H}, H\right)$

b. bal oldali egyenes ($\theta < 90^\circ$) metszéspontja a kép tetszőleges magasságában húzott vízszintes egyenessel



4.10. ábra. Bal oldali sávjelző vonal ábrázolása derékszögű koordináta-rendszerben.

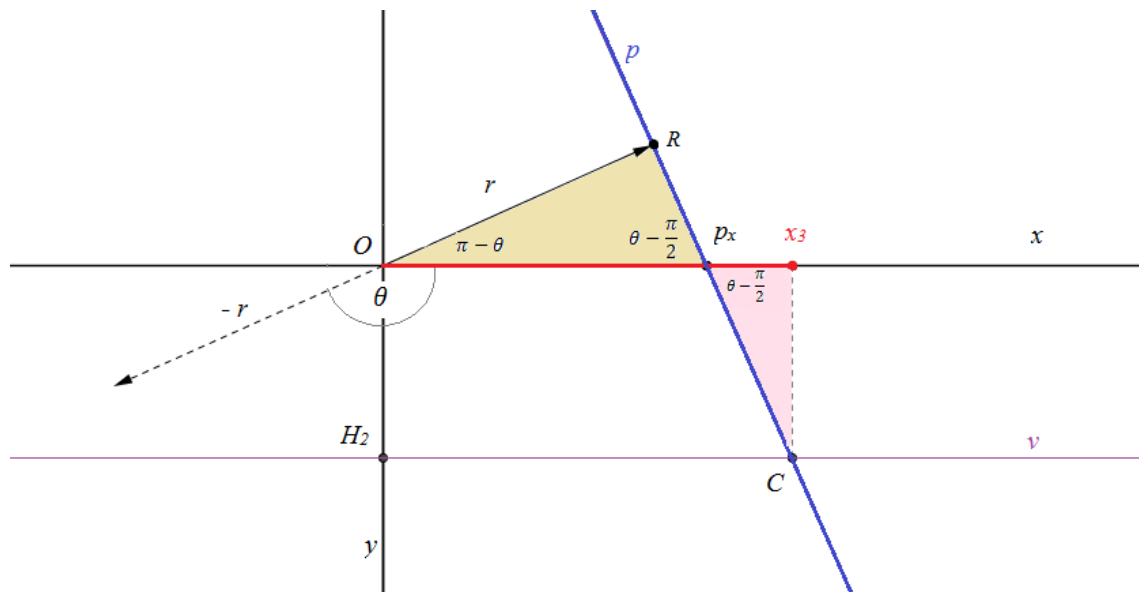
A második ábrán a v egyenes a kép magasságának felezővonala, p egyenes pedig ismét a sávjelző vonalat jelöli. A két egyenes metszéspontja: $B(x_2, y_2)$. A B pont y koordinátája ebben az esetben is ismert, ez a kép magasságának fele (H_2). A B pont x koordinátájának kiszámításához az $(x_2, 0)$, $B(x_2, H_2)$ és $(p_x, 0)$ csúcsokkal meghatározott háromszögből indulunk ki.

A θ szögre felírt tangens szögfüggvény:

$$\tan \theta = \frac{p_x - x_2}{H_2} \quad \Rightarrow \quad x_2 = p_x - (\tan \theta * H_2)$$

$$\cos \theta = \frac{r}{p_x} \Rightarrow p_x = \frac{r}{\cos \theta}$$
$$x_2 = \frac{r}{\cos \theta} - (\tan \theta * H_2)$$

c. jobb oldali egyenes ($\theta > 90^\circ$) metszéspontja a kép tetszőleges magasságában húzott vízszintes egyenessel



A 4.11. ábrán látható módon a jobb oldali egyeneshez tartozó θ szög 90° -nál nagyobb, emiatt kicsit módosítani kell a számításokon. Mivel a jobb oldali sávjelző vonal koordináta-rendszerben ábrázolva egy csökkenő egyenest ad, az egyeneshez tartozó r vektor a koordináta-rendszer egy másik negyedében lesz található az előző esethez képest. Ebből kifolyólag az egyenes H_2 magassághoz tartozó x -koordinátája (x_3) is más arányok szerint számítható. A 4.11. ábrán

rózsaszínnel kijelölt $(p_x, 0)$, $C(x_3, H_2)$ és $(x_3, 0)$ csúcsok által alkotott háromszögből indulunk ki. A számításhoz szükséges szögek π -vel és θ -val való kifejezése után, felírom a tangens szögfüggvényt az említett háromszögben található $\theta - \frac{\pi}{2}$ szögre:

$$\tan\left(\theta - \frac{\pi}{2}\right) = \frac{H_2}{x_3 - x_p} \quad \Rightarrow \quad x_3 = \frac{H_2}{\tan\left(\theta - \frac{\pi}{2}\right)} - x_p$$

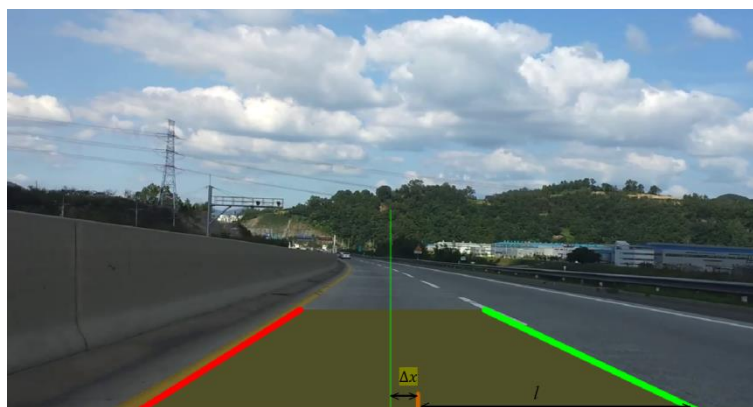
x_p -t az $O(0, 0)$, $(p_x, 0)$ és $R(x_0, y_0)$ pontok által meghatározott háromszögből tudjuk kiszámolni. Felírjuk a következő szögfüggvényt:

$$\sin\left(\theta - \frac{\pi}{2}\right) = \frac{r}{x_p} \quad \Rightarrow \quad x_p = \frac{r}{\sin\left(\theta - \frac{\pi}{2}\right)}$$

Ezt visszahelyettesítve x_3 egyenletébe, ezt kapjuk:

$$x_3 = \frac{H_2}{\tan\left(\theta - \frac{\pi}{2}\right)} - \frac{r}{\sin\left(\theta - \frac{\pi}{2}\right)}$$

A kamera gépjárműn való beállításainak megfelelően a jármű akkor halad tökéletesen, az útsáv közepén, ha a felvételen a kép középvonala egybeesik a képen azonosított útsáv közepével. A 4.12. ábrán látható zöld függőleges vonal jelöli a kép közepét, a rövid narancssárga pedig az útsáv közepét. A köztük lévő távolság Δx , ami az jelöli, a gépjármű mennyire tér el az útsáv közepétől. Ez százalékban kifejezve: $\frac{\Delta x}{l} * 100$. Ha a kitérés meghaladja a 20 %-ot a program figyelmeztető hangjelzést ad a vezetőnek.

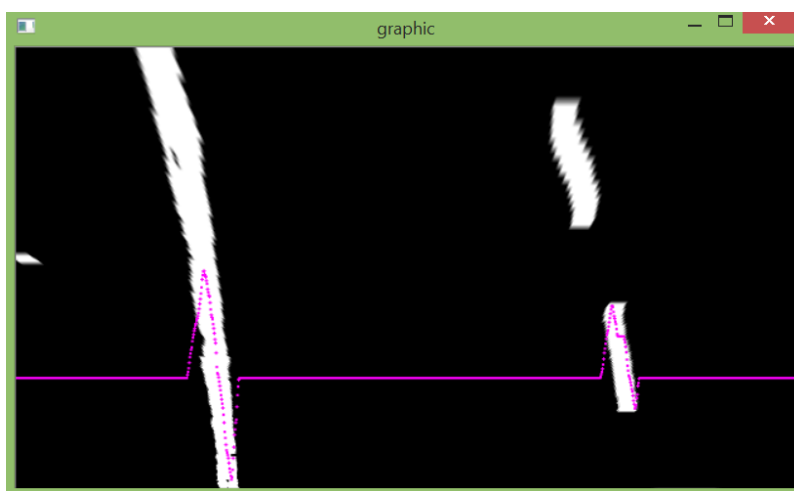


4.12. ábra. Az gépjármű helyzetének eltérése az útsáv közepétől.

4.4. Kanyar felismerése

A képen lévő vonalak görbületének megállapítása egy perspektívában ábrázolt képen nem a legideálisabb. Sokkal egyszerűbb viszont felülnézetből, ahol a párhuzamos vonalak valóban párhuzamosnak látszanak. Ezért az út kanyarodási irányának felismeréséhez a program először a perspektívában látható képet felülnézetivé transzformálja.

Megfigyelések alapján megállapítható, hogy az átranszformált képen az út jobbra kanyarodásakor a sávjelző vonalak egyértelműen jobb irányba, balra kanyarodásakor bal irányba dőlnek, egyenes haladása esetén pedig a két oldalsó sávjelző vonal vagy párhuzamos egymással, vagy kissé egymás felé dől. A program egy vágást végez a kép magasságának háromnegyedén, majd oszloponként összeszámolja a fehér pixeleket, külön a kép magasságának felétől a vágásvonalig, és a vágásvonaltól a kép aljáig terjedő tartományokban. Ezután az alsó tartományban meghatározott értékekből kivonja a felsőtartományértékeit. Ekkor a vágásvonal mentén ahol a sávjelző vonal áthalad rajta, egy hullámot kapunk (4.13. ábra). A hullám negatívba nyúló része a vágásvonal alatti, pozitívba nyúló része pedig a vágásvonal feletti fehér pixeleket mutatja. A sávjelző vonal haladási iránya abból határozható meg, hogy a hullám melyik tartományba nyúló része következik előbb, vagyis a hullám szélsőértékei között a függvény növekszik vagy csökken. Képzeljük csak el: ha a vágásvonalról szemlélünk egy balra dőlő vonalat, akkor az alulról jobbról közelíti meg a vágásvonalat, elhagyva azt tovább jobbra halad. Tehát a hullám pozitív része következik először, majd utána a negatív, azaz a szélsőértékek közti rész csökkenő lesz. Ez látható az 4.13. ábrán is. Ellenkező esetben a fordítottja történik.



4.13. ábra. Sávjelző vonalak felülnézetből a vágásvonallal.

Az út kanyarodási irányát a program 10 képkockánként határozza meg az addig összeszámlolt csökkenő vagy növekvő hullámokból. Ha a hullám 10 képkocka alatt az esetek legalább 80 %-ában csökkenést mutatott, a vonal jobbra kanyarodik, legalább 80 %-os növekvés esetében pedig balra. Az út két oldalán található sávjelző vonal a következőképp határozza meg az út kanyarodási irányát:

$$\text{út iránya} = \begin{cases} \text{jobbra kanyarodik,} & \text{ha } j \text{ és } b \text{ is jobbra kanyarodik} \\ \text{balra kanyarodik,} & \text{ha } j \text{ és } b \text{ is balra kanyarodik} \\ \text{egyenes,} & \text{ha } j \text{ balra és } b \text{ jobbra kanyarodik} \end{cases},$$

ahol j az út jobb, b pedig a bal oldalán lévő sávjelző vonalat jelöli.

A felhasználó számára a képernyőn megjelenő nyilak jelzik a kanyarodási irányt.

4.5. A sávjelző vonal szaggatottságának megállapítása.

Vezetéskor sokan esnek abba a hibába, hogy ha valami vagy valaki az útjában áll, sietségében előzésbe/kerülésbe kezd, megfélekedve arról, hogy leellenőrizze, az adott útszakaszon egyáltalán engedélyezett-e az előzés. Ez szerencsésebb helyzetben következmény nélkül is véghezvihető, jobb esetben pénzbírságot von maga után, ám sajnos, ez a hiba balesethez is vezethet. Fontosnak tartottam tehát, hogy az elkészült programomnak szerves részét képezze a sávjelző vonal szaggatottságának megállapítása, és a vezető folyamatos tájékoztatása arról, hogy az útburkolati jelzések alapján engedélyezett-e az előzés az adott útszakaszon.

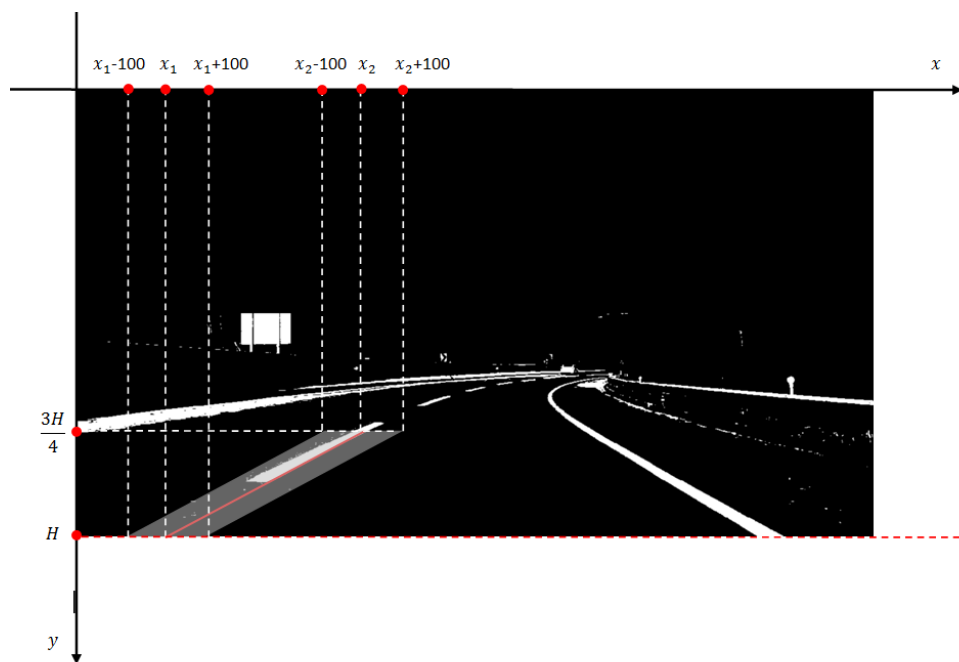
A sávjelző vonal folytonosságának vizsgálatát a szegmentált képen véggezzük. Ez fekete-fehér kép, fekete a háttér, a hasznos részek pedig, ebben az esetben a sávjelző vonalak, fehérek. A vizsgálatot az előzőleg Hough-transzformációval azonosított vonal mentén kell végezni. Ez a vonal azonban csak 1 pixel vastagságú, ami nem elegendő ahhoz, hogy kellő pontosságú eredményt kapjunk. Ezért a vonal 100 pixeles környezetével dolgozunk. Az 4.14. ábrán látható piros vonal, a Hough-transzformációval azonosított vonal, a fehérrel kijelölt terület pedig annak 100 pixeles környezete.

A program összeszámolja, hogy adott magasság szinteken a vonal 100 pixeles környezetében hány fehér képpont található, és azt egy tömbben tárolja. A vizsgálat a kép azon magasság szintjén kezdődik, ahol a jobb és a bal sávjelző vonalak metszik egymást, és a kép aljáig tart. Speciális esetben, ha a sávjelző vonal nem érné el a kép alját, hanem a kép oldalán kilóg, akkor a vizsgálat megszakad, elkerülve így az értelmetlen értékek beszámítását. Ezután meg kell határozni a küszöbértéket, ami azt mutatja, legkevesebb mennyi fehér képpont szükséges ahhoz, hogy azt mondhassuk, adott magasság szinten megjelenik a sávjelző vonal. Ez elsősorban furcsán hangzik, de mivel a vonal ferdén áll a képen, a képpontszámítás pedig vízszintesen történik, megeshet, hogy a szaggatott vonal sarka csak néhány fehér képpontot hordoz egy

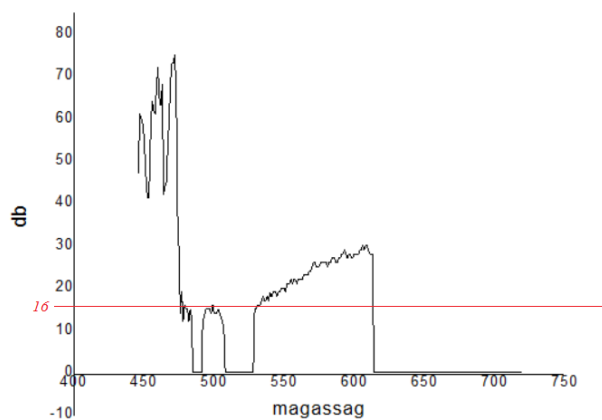
magasságszinten a vizsgált százból. Ez nyilvánvalóan nem számítható a sávjelző vonal részének, ami normál esetben legalább 50-60 fehér képpontot tartalmaz egy szinten. A határértéket a következőképp számítjuk:

$$h = \frac{1}{2 \cdot db} \sum_k^n f_k ,$$

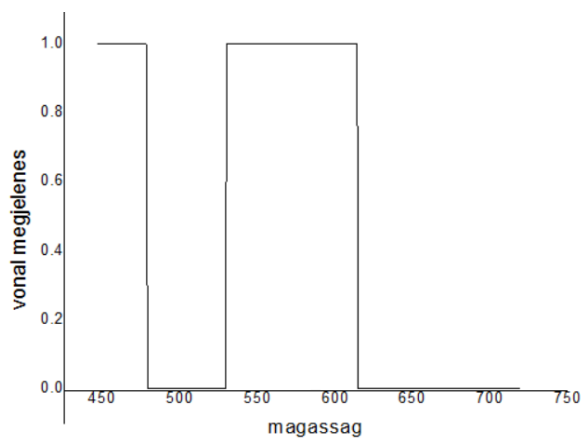
ahol db a fehér képpontokat tartalmazó magasságszintek száma, k a magasságszint, ahol a vizsgálat kezdetét veszi, n a magasságszint, ameddig a vizsgálat tart, f_k pedig a fehér képpontok száma adott magasságszinten.



4.14. ábra. Sávjelző vonal folytonosságának vizsgálata a szegmentált képen.



4.15. ábra. Fehér képpontok darabszáma adott magasságszinteken.



4.16. ábra. Vonal szaggatottsága.

A 4.15. ábrán látható grafikon vízszintes tengelyén a kép magassága látható, függőleges tengelyén pedig az adott magasságszinten található fehér képpontok darabszáma. A grafikonon 16 darabnál meghúzott piros, vízszintes egyenes a küszöbértéket jelöli.

A küszöbértéken felül eső értékek az mutatják, hogy adott magasságszinten megjelenik-e a sávjelző vonal. Ezek új, 1-es értéket kapnak. Azon értékek, melyek nem érik el a küszöbértéket, 0-ás értéket kapnak. A 4.16. ábrán látható a 4.15. ábrán látható grafikon küszöbölés utáni verziója. A vízszintes tengelyen a kép magasságszintjei láthatóak, a függőleges tengelyen pedig a sávjelző vonal megjelenése. Jól látható, hogy a példa egy szaggatott vonalat mutat be.

Ezután a program eldönti, hogy a vizsgált vonal folytonos-e. Megszámolja, hogy az egyenes mentén hány olyan magasságszint van, ahol nem jelenik meg a sávjelző vonal. Ha ez szakasz nagyobb, mint a teljes vizsgált szakasz 35 %-a, akkor az sávjelző vonal szaggatottnak, ellenkező esetben teltnék minősül. A felhasználó számára az eredmény a kijelzőn válik láthatóvá: szaggatott vonal esetén egy zöld színű, telt vonal esetében pedig egy piros vonal jelenik meg.

5. ÖSSZEFOGLALÓ

Az eredmény egy hatékony, látványos program, mely használata jelentősen megkönnyítheti a gépjárművezető dolgát, s akár a baleset bekövetkezését is megelőzheti. Megalapozott képfeldolgozási eljárásokkal dolgozik, és az eredményről grafikus kijelzést ad a vezető számára. Kijelöli az útsávot, amelyben a gépjármű halad, villogó jelzést ad, ha túlságosan kitért a sávból, meghatározza a sávjelző vonalak folytonosságát, szaggatott vonal esetén zöld vonalla, telt vonal esetében pedig pirossal jelöli azt. Ezen felül meghatározza az út kanyarodási irányát, és nyilakkal jelzi azt a felhasználó számára.

Munkám során rengeteg tapasztalatot szereztem. Megismerkedtem számos új és érdekes dologgal az informatika és programozás terén, megéreztem a képfeldolgozás nehézségét, de egyben szépségét is, és mindenekfelett megbizonyosodást nyertem arról, hogy milyen irányban szeretném a további tanulmányaimat folytatni.

Munkám, mint minden más informatikai találmány, hatékonysága ellenére természetesen bővíthető, továbbfejleszthető, s ennek végrehajtása jelen is van a jövőbeli terveim közt.

6. IRODALOMJEGYZÉK

- [1]- Czap László: Képfeldolgozás., Miskolci Egyetem.
- [2]- Wayne Niblack: An Introduction to Digital Image Processing, Prentice-Hall International Ltd, UK, 1986.
- [3]- http://www.inf.u-szeged.hu/~gnemeth/kurzusok/kepfel1/7_gyakorlat.pdf
- [4]- Szilágyi László: Képfeldolgozás jegyzet., 2008.
- [5]- Ralf Kohler: A Segmentation System Based on Thresholding, Computer Graphics and Image Processing, Vol. 15, 1981, pp. 319-388.
- [6] - Fazekas Attila, Kormos János: Digitális képfeldolgozás matematikai alapjai., Debreceni Egyetem Informatikai Intézet.
- [7] - OpenCV 2.4.13.2 documentation. <http://docs.opencv.org/2.4/index.html>
- [8] - http://users.nik.uni-obuda.hu/vamosy/SZPE2011/Eloadasok/06_KepfeldolgozasVZ20080505_kepernyore.pdf
- [9] - <http://infohost.nmt.edu/tcc/help/pubs/colortheory/web/hsv.html>
- [10] - http://www.who.int/gho/road_safety/mortality/traffic_deaths_number/en
- [11] - Henning Bassmann, Philipp W. Besslich: Ad Oculos Digital Image Processing., International Thomson Publishing Company, Sheffield, 1995.
- [12] - Theo Pavlidis: Algorithms for Graphics and Image Processing., Computer Science Press, Rockville, 1982.
- [13] - Anil K. Jain: Fundamentals of Digital Image Processing., Prentice-Hall, Inc. New Jersey, 1989.
- [14] - <http://www.fajltube.com/gazdasag/auto/A-kozuti-kozlekedesbiztonsag-e55261.php>
- [15] - <http://www.ijser.org/researchpaper%5CA-Simple-Birds-Eye-View-Transformation-Technique.pdf>
- [16] - <http://orfe.princeton.edu/~alaink/PAVE2014SummerWorkshop/PAVE-Workshop-FundamentalsComputerVisionCCY.pdf>
- [17] - http://www.inf.u-szeged.hu/~gnemeth/kurzusok/kepfel1/7_gyakorlat.pdf
- [18] - <http://research.ijcaonline.org/volume113/number9/pxc3901789.pdf>
- [19] - Amartansh Dubey, K. M. Bhurchandi: Robust and Real Time Detection of Curvy Lanes (Curves) With Desired Slopes for Driving Assistance and Autonomous Vehicles, Department of Electronics and Communication Engineering, Visvesvaraya National Institute of Technology, Nagpur, India, 2015.
- [20] - <http://homepages.inf.ed.ac.uk/rbf/HIPR2/classify.htm>
- [21] - <http://homepages.inf.ed.ac.uk/rbf/HIPR2/threshld.htm>
- [22] - http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT6/node2.html
- [23] - <http://user.engineering.uiowa.edu/~dip/LECTURE/PreProcessing3.html#edge>
- [24] – Palágyi Kálmán: Képfeldolgozás haladóknak., Typotex Kiadó, 2011.