

Gépjárműtípus felismerés beépített szoftver segítségével¹

Max Gyula

Budapest Műszaki és Gazdaságtudományi Egyetem,
Automatizálási és Alkalmazott Informatikai Tanszék
1521 Budapest, Pf. 91, e-mail: max@aut.bme.hu

Abstract: Közlekedési események megfigyelése során az a cél, hogy az aktuálisan vizsgált folyamatot minél gyorsabban és pontosabban tudjuk elemezni. Ezért az elemzés folyamata csak real-time módon történhet, ahol nemcsak a vizsgált objektumok mozgása lehet fontos tényező, hanem maga a mozgó objektum is. A gépjármű típus és modellazonosító (Vehicle Make and Modell Recognition - MMR) technikákat ma már számos probléma megoldása során használhatjuk, mint pl. útvonal ellenőrzés vagy sebesség megosztás. A cikkben bemutatunk egy könnyen megvalósítható eszközt, amely alkalmassá a járművek típusának azonosítására. Az OpenCV programcsomag segítségével bemutatjuk, hogy hogyan oldhatjuk meg a gépjárművek típusfelismerését, mert sok esetben nem a jármű pontos azonosítására van szükségünk, hanem csak néhány, a megfigyelő számára fontos adat tesztelésére.

1. BEVEZETÉS

Intelligens forgalom irányító rendszerek esetén akár személyre vagy csoportra szabott üzenetek továbbítására is lehetőségünk van. Egy alacsony közúti híd alatti áthaladás során figyelmeztethetjük a vezetőket a várható veszélyre, sőt alternatív útvonalat is javasolhatunk nekik. Az általános szöveg megjelenítése során, ha rendszerünk felismer egy nagyméretű járművet, akkor a szöveg villogtatásával vagy hangjelzéssel hívhatja fel a vezetők figyelmét a veszélyre. Hasonló feladat vár a rendszerre akkor is, ha például a forgalom gyorsítása miatt – többsávos út esetén – a gyors járműveket a belső sávokba, míg a lassabbakat a külső sávokba szeretnénk irányítani.

A rendőri munka segítése során előfordulhat, hogy egy meghatározott típusú járművet kell azonosítani. Ebben az esetben csak azokkal a járművekkel kell foglalkoznunk, amelyek típusa megegyezik a keresett járműjével. A típus gyors azonosításával jelentős időt nyerhetünk a körözött jármű azonosításához.

Ha végiggondoljuk az előbbi feladatokat, akkor láthatjuk, hogy ezekben az esetekben elsődlegesen nem a jármű pontos azonosítása a feladatunk, elegendő néhány speciális paraméter ellenőrzése. A mindennapi közlekedési események megfigyelése nem egyszerű feladat, mert a beavatkozást kívánó események csak ritkán történnek, és ezek időtartama is meglehetősen rövid. Ha ezek a rendszerek humán megfigyelők segítségével ellenőrzik a közlekedést, akkor a monotonia jelentős hibaforrásként lép be ebbe a rendszerbe.

Egyértelmű, hogy a közlekedési események szemmel történő ellenőrzése (Max, 2012) nagy figyelmet, és speciális megfigyelő eszközöket igényel. Mégis mindig van esély arra, hogy sürgős beavatkozást igénylő események színhelye.

2. AZONOSÍTÁSI TECHNIKÁK

Az MMR rendszerek alapvetően abból a feltételezésből indulnak ki, hogy ha ismert az egyes típusok és modellek teljes felépítése, amihez képi tartalom is kapcsolható, akkor a felismerés problémája megoldható. Egy robusztus jármű MMR technikától viszont elvárjuk, hogy képes legyen felismerni a jármű típusa (Max, 2013) és modellje (Zafar et al., 2007) mellett a változó környezetet is. A legtöbb meglévő gépjármű felismerő rendszer ezek alapján először szegmentálja a vizsgálandó objektumot, majd ellenőrzött feltételek felhasználása mellett megpróbálják azonosítani a különböző gyártmányokat és modelleket.

Magát az azonosítás műveletét két csoportra bonthatjuk:

- statikus azonosítás
- dinamikus azonosítás

Az első csoportba tartoznak azok a technikák, amelyek fénykép vagy videó felvételek alapján kísérlék meg az azonosítást. Ez a módszer alapvetően eltér a dinamikus módszertől.

A statikus módszerek alkalmazása során ugyanis lehetőség van a képek időben nem limitált, többszöri ellenőrzésére is. Amíg egy elmosódott Trafipax képet többszöri nagyítással, szűrővel vagy egyéb technikák alkalmazásával, idővel talán felismerhetővé lehet varázsolni, addig egy térfelnyelő kamera felvételén rögzített baleset során roncsolódott gépkocsi azonosítására az előző módszerhez képest sokkal kevesebb idő áll rendelkezésünkre. A statikus módszerek előnye

¹ This work is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. This project is supported by the New Hungary Development Plan (Project ID: TAMOP-4.2.1/B-09/1/KMR-2010-0002).

alapvetően a megismételhetőségben jelenik meg. Ez egyben azt is jelenti, hogy olyan információkat is fel tudnak használni az azonosításra, amelyek a kép készítésének pillanatában még nem álltak rendelkezésre.

Ezzel szemben a dinamikus – real-time – felismerési módszerek csak azokkal az információkkal rendelkeznek, amelyek a felvétel készítésének pillanatában a rendszer számára hozzáférhetőek voltak. Mivel a statikus azonosítási technikák időkorlát nélkül képi adatfeldolgozási műveletekre vezethetők vissza, ezért ezzel a kérdéssel ebben a cikkben nem foglalkozunk.

A real-time azonosítási technikák alkalmazása során az időkorlát mellett számos más problémát is meg kell oldani. Nehézséget jelent a felismerésben például:

- a másodpercenként készített felvételek száma
- a képfelbontás minősége
- a változó megjelenés

A másodpercenként készített felvételek száma még az azonosítás első fázisában, a szegmentálásban jelent problémát. Egy 72 km/h sebességgel haladó jármű másodpercenként 20 métert tesz meg. Ha kameránk másodpercenként csak 4-5 képet készít, mert például a képfeldolgozási idő nem tesz lehetővé ennél gyorsabb felvételt készítést, akkor ez azt jelenti, hogy a vizsgált járművet minden egyes felvétel után újra meg kell keresni, és azonosítani a képen, mert az előző felvételhez képest járművünk helyén már egy másik gépjármű állhat. Ez egyértelműen megnöveli az adatfeldolgozás idejét, míg ha normál, másodpercenként 24-25 képkockás felvételeket tudunk készíteni a vizsgált objektumról, akkor az ilyen rövid idő elteltével, még biztosan fedésbe hozható az előző felvételen található gépjárművel, vagyis az objektum környezetétől való szegmentálása egyszerűbben elvégezhető.

A képfelbontás minősége a későbbi azonosítási folyamatban játszik fontos szerepet. Egy nagyfelbontású képen, nagyobb esélye van egy MMR rendszernek tipikus azonosítási pontokat találni egy alacsonyabb felbontású képhez képest. A viszonylag finom (kis méretű) azonosítási pontok, például a gyártó emblémájának a gépjárművön elfoglalt helye, nagyon sokat segítenek a különböző járműtípusok megkülönböztetésében.

A változó megjelenés is erősen gátolja a felismerést. A tetőcsomagtartók, vagy a csomagtartókban elhelyezett nagyméretű tárgyak jelentősen rontják az azonosíthatóságot. Egy tetőcsomagtartón elhelyezett mosógép például alapvetően megváltoztatja a gépjármű fizikai paramétereit. Egy kombi csomagtartójában elhelyezett doboz felismerhetetlenné teheti például a gépjármű hátsó ablakát, aminek hiányában a jármű azonosíthatatlanná válik. Ugyanilyen problémát jelentenek az időjárási tényezők is. A jármű színének meghatározása a megvilágítás erősségétől függően meglehetősen széles tartományban változhat, ami szélsőséges esetben pontatlan színmeghatározást is jelenthet.

3. A MODELL

Modellünkben, a közlekedési események szemléltetéséhez és a mozgások követéséhez kétdimenziós leírást használunk, amelyen tanulmányozhatjuk a közlekedés különböző mozzanatait, mint pl. az előzést, a kanyarodást, a dugókat vagy akár a közlekedésben résztvevők viselkedési mintáit. A mozgások leírásánál, a gyakorlati tapasztalatokat figyelembe véve, négy vagy hat irányban követjük a résztvevők mozgását: előre, hátra, jobbra, balra valamint jobbra előre és balra előre. Induláskor, a felületen lévő mozgó objektumok véletlenszerűen és számban helyezkednek el, akárcsak egy pillanattfelvételen.

Az 1. ábrán látható modell feladata, hogy ezen kezdeti értéket mellett, egyéb feltételek ismerete nélkül is, automatikusan fel tudja ismerni az éppen folyó közlekedési eseményeket és a közlekedési szabályok ismerete mellett tudja értelmezni a közlekedésben résztvevők szokásait.

Modellünk két – képfelismerő és szabályfelismerő – alrendszerből áll. A képfelismerő rendszer három – képközel, kommunikációs és beavatkozó – réteget, míg a szabályfelismerő két réteget tartalmaz.

3.1 Képközel réteg

A felismerés elengedhetetlen feltétele a kamera. Az események feltérképezéséhez használhatunk egy, de akár több kamerát is (Horáud, 2006). A kamerákat többnyire vízszintesen állítjuk be úgy, hogy vizsgált felület minél nagyobb legyen. Számos esetben azonban, ahol ez megoldható, a függőleges kamera felfüggesztés javasolt. A terület geometria felépítésének ismerete általában nem szükséges, ha azonban rendelkezésre áll, segíthet a szokatlan közlekedési események (pl. szabálytalan sávváltás, dugó vagy baleset) feltárásánál.

A képközel réteg feladata a kamerák által felvett képeken a mozgó objektumok detektálása. Egyszerű kocka modellt használunk hat csúcsponttal a mozgó objektumok háromdimenziós adatainak meghatározásánál. Ugyanezt a modellt használjuk a járművekhez kapcsolódó árnyékok felismerésénél is, amely hat lehetséges kapcsolatot ír le az objektum, a fényforrás és a kamera között (Mikic et al., 2003; Yoneyama et al., 2003).

A látási viszonyok miatt alapvetően különböző eljárást használunk a nappali és az éjszakai detektáláshoz.

3.2 Kommunikációs réteg

A kommunikációs réteg megpróbálja analizálni bejövő kép és az útfelület külvilág kapcsolatát. Igazi kapcsolat nincs a külvilág és a kép között, mivel kezdetben nincsenek ismereteink a külvilágról.

Ha nem ismerjük a felfestett közlekedési sávok helyét, csak a forgalom alapján tudunk következtetni vagy becslést adni rájuk.

Ha egy a kétdimenziós képen látható $pk=(xp,yp)$ pontot meg



CAETS

„IFFK 2014” Budapest

Online: ISBN 978-963-88875-3-5

CD: ISBN 978-963-88875-2-8

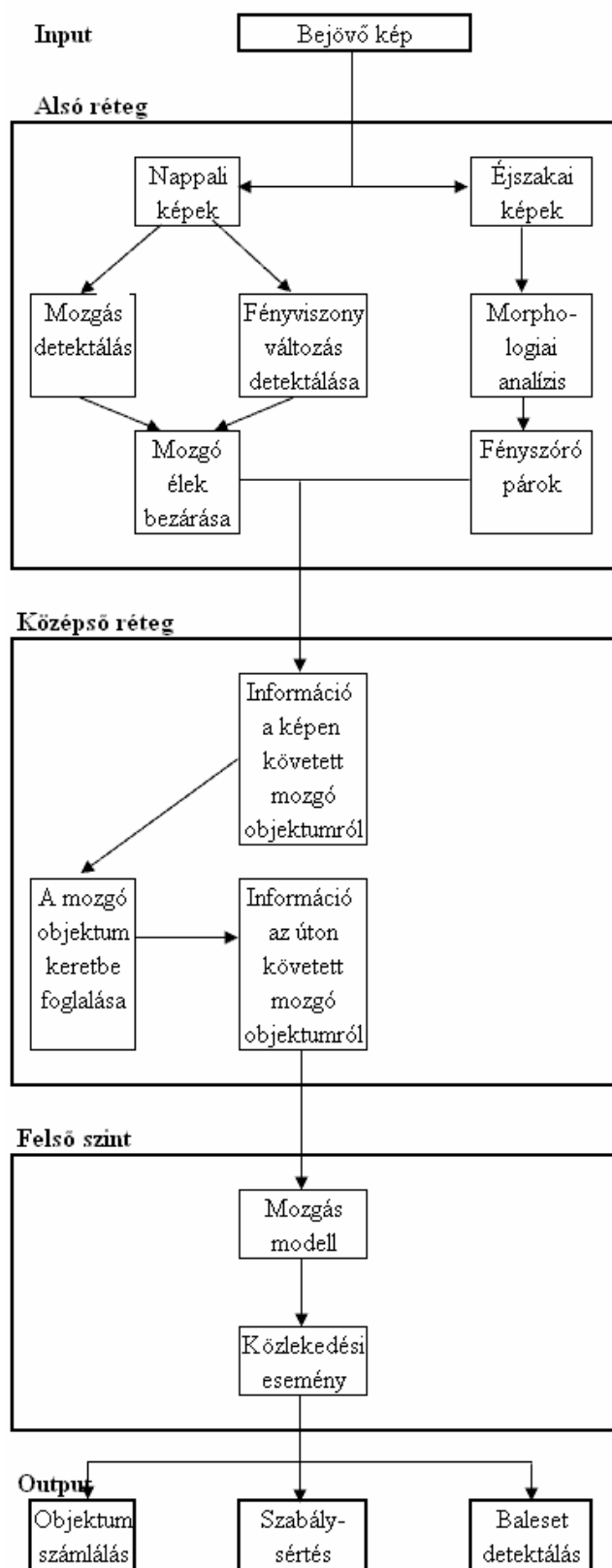
További logók helye

Paper xx

Copyright 2014 Budapest, MMA

Editor: Dr. Péter Tamás

Képfeldolgozás



1. ábra: A modell felépítése

akarjuk feleltetni az útfelszín $pr=(x_r,y_r)$ pontjának, akkor azt a következő reláció segítségével oldhatjuk meg. A feladat megoldásához legalább négy képpont pár ismerete szükséges a kép és az útfelület között, hogy meg lehessen határozni az (1) összefüggés „a” paramétereit.

Az egyik lehetséges megoldás az, hogy választunk egy útjellel jelzett téglalapot az út felületén. Ez a megoldás akkor használható, ha a kamera tisztán látja ezeket a pontokat.

A réteg kimenete az útfelületen követett objektum adatai, amelyet a bejövő képről kaptunk. A kapott információkat a kiegészítő modulon keresztül visszacsatoljuk.

3.3 Beavatkozó réteg

Ez a réteg felelős a kommunikációs réteg és a kiegészítő modul által összegyűjtött információk alapján a közlekedési esemény azonosításáért. A kiegészítő modul analizálja a helyszíni információkat, mint pl.: az egyes sávok helyét vagy a sávok számát. Bár számos technika ismert a sávokra vonatkozó információk feldolgozására, az általunk választott módszer statisztikai feldolgozásnak számít. A mozgó objektumok középvezetési átlagolását használjuk erre a célra, mivel a legtöbb detektált mozgó objektum útjának legnagyobb részét ugyanabban a sávban teszi meg. Miután azonosítottuk a helyszínt a kiegészítő modul a közlekedési szabályokat az adott helyszínhez tartozó hely és mozgás információkra fordítja le.

Két egyszerű példán - sávvaltoztatás és dugó esetében - bemutatjuk az elvégzendő feladatokat.

A kiválasztott mozgó objektum a fentiek szerint lesz detektálva a következő feltételek szerint:

1. Minden mozgó járművek besorolunk valamelyik, a modell által detektált sávba. Ha az aktuális képen a besorolt sáv különbözik az előzőtől, akkor az sávvaltoztatást jelent.
2. Dugót akkor detektálunk, ha a különböző sávokban álló mozgó objektumok átlagsebessége az aznapi átlagsebességhez tartozó, előredefiniált érték alatt marad.

Bár ezt az értéket előre kell definiálni, az egyes közlekedési eseményeket már könnyű megfigyelni miután beállítottuk ezt az értéket. Bár a határértékeket ez a réteg generálja, a kapott értékeket visszacsatoljuk a kiegészítő modulon keresztül az összes azonosított mozgó objektumra. Ha ez az érték minden mozgó objektumra igazává válik, akkor bekövetkezett a megfigyelni kívánt közlekedési esemény.

4. MEGVALÓSÍTÁS

Az OpenCV (Open source Computer Vision) egy nyílt forráskódú (lásd <http://opensource.org>) valós idejű számítógépes látás könyvtár, több mint 500 beépített függvénnyel. OpenCV célja a valós idejű képfeldolgozó



alkalmazások számítási hatékonyságának növelése. Az OpenCV függvényeket - többmagos processzorokra - C és C++ nyelven írták, és Linux, Windows és Mac rendszerekhez egyéni interfészek segítségével Python, Ruby, Matlab, és más nyelvekhez illesztették. A könyvtár ingyenesen letölthető a <http://SourceForge.net/projects/opencvlibrary> honlapról. Az OpenCV egyik célja az volt, hogy a felhasználók kezébe adjon egy egyszerűen használható számítógépes látás feldolgozó infrastruktúrát, amely gyors segítséget nyújt a meglehetősen bonyolult látásalkalmazások kialakításához. Az OpenCV könyvtár több mint 500 beépített függvénye, amely 2500-nál is több optimalizált algoritmust tartalmaz, számos területen segíti a képfeldolgozó alkalmazásokat, úgymint pl. a gyártás-ellenőrzés, az orvosi képalkotás, a biztonságtechnika, a kamerakalibrálás, a sztereólátás, az interaktív művészet, a robotika, stb. Mivel a számítógépes látást és a gépi tanulást gyakran használják ugyanabban az alkalmazásban, az OpenCV tartalmaz egy teljesen általános célú Machine Learning Library (MLL) is. Ez az alkönyvtár statisztikai alakfelismerésre és a klaszterezésre összpontosít.

Az alábbi egyszerű példa bemutatja, hogy mit csinálunk, ha végig akarunk menni egy 8 bites unsigned karakterekből felépülő, 3 csatornás kép (pl.: RGB mátrix) minden pixelen:

```
cv::Mat imgmat(cv::Size(640,480),CV_8UC3);
```

A kép pixellein szeretnénk végigmenni:

```
for(int row=0; row<imgmat.rows; row++) {
    for(int col=0; col<imgmat.cols; col++) {
        ---
    }
}
```

A következő példa a pixelek írását, olvasását mutatja be:

Olvasás:

```
unsigned char kek = imgmat.at<cv::Vec3b>(sor,oszl)[0];
unsigned char zold = imgmat.at<cv::Vec3b>(sor,oszl)[1];
unsigned char piros = imgmat.at<cv::Vec3b>( sor,oszl)[2];
```

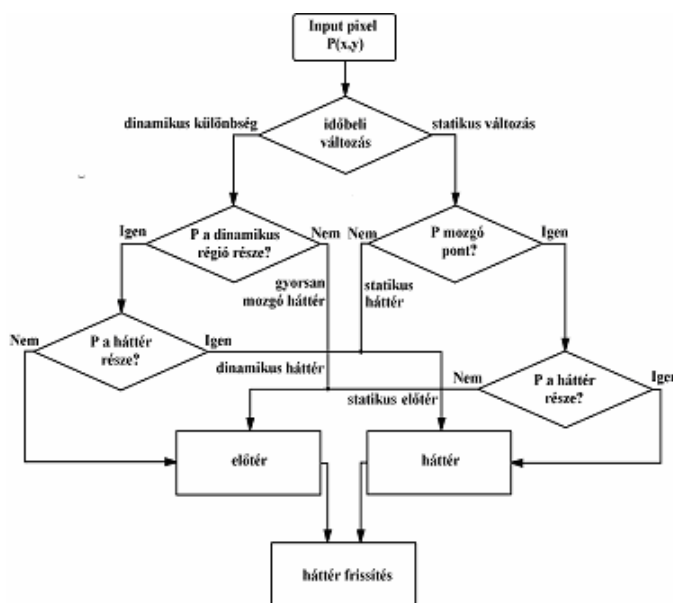
Írás:

```
imgmat.at<cv::Vec3b>(sor,oszl)[0] = kek;
imgmat.at<cv::Vec3b>(sor,oszl)[1] = zold;
imgmat.at<cv::Vec3b>(sor,oszl)[2] = piros;
```

Az eljárás előnye, hogy egyszerű és könnyen érthető. Hátrányai közt azonban meg kell említeni, hogy ha ezt a fenti 6 sort berakjuk a két for ciklus közé, akkor iszonyatosan megnő a CPU terhelés. CPU terhelés növekedése akkor látszik igazán, amikor a 640x480-as webkameránk 30 FPS-sel küldi a képeket, és minden frame-re végre akarjuk hajtani a fenti elemzés. Ekkor a kimeneti képen látszik, hogy kicsit akad.

Az elemzés folytatásához a képen meghatározott pontok, ún. markerek beazonosítása a következő lépés. Mozdó objektumok azonosítása során ezeket a markereket maga a mozgás fogja előállítani számunkra. Ehhez azonban

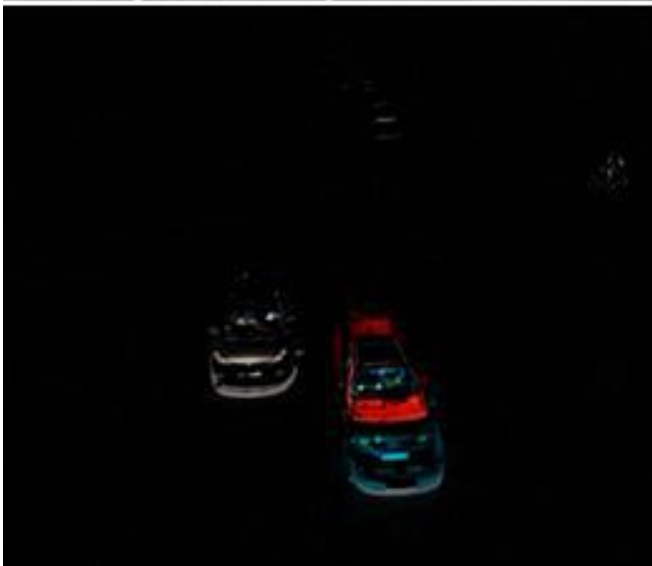
elsődlegesen meg kell határoznunk azt a képterületet, ahol a mozgás történik, hogy el tudjuk választani attól a területtől, ami mozdulatlan. A kép ez utóbbi része a mozdulatlan háttér. A 2. ábrán a háttér kiszűrésének fázisait figyelhetjük meg.



2. ábra: A háttér kiszűrésének fázisai

A háttér meghatározása után újra a mozgó markerekre kell koncentrálnunk. A markerek felismeréséhez első lépésként éldetektálásra van szükség. Az élek meghatározására több módszert is használhatunk. Az első lényege, hogy az éleket a jelenlegi kép és a háttér különbségeként keletkező eredő képen határozzunk meg, míg a második lehetőség során két egymást követő kép különbsége mutatja az élek helyét. Zajmentes környezetet feltételezve az első módszer ad jobb közelítést, ahogy ez a 3. ábrán is látszik, mert ebben az esetben már a mozgó objektum mérete is közelíthető, míg a második módszer eredménye a mozgó objektum elmozdulást tartalmazó képe. A marker pozíciója és helye ezután már meghatározható a kamerához képest, vagyis egy tetszőleges virtuális objektumot helyezhetünk el modellünkben.

Ezt követően meg kell keresni a befoglaló éleket, melyek sokszögekké alakíthatóak [Max, 2007], majd a kapott sokszögeket szűrni kell a négyszögekre. Miután megvannak a négyszögek, mintát kell venni az élek melletti külső, de főleg a belső pontokból, hiszen ebből szűrhetők ki a további lehetséges markerek. A befoglaló éleken kívül, a kontúr vonalaktól adott távolságra a háttérre jellemző markereket, míg az éleken belül, az adott mozgó objektumra jellemző markereket kell azonosítani. Ezek előállítását után nagy valószínűséggel már csak a tényleges markerek maradnak meg a lehetséges elemek halmazában [3?]. Ezek a feladatok az OpenCV [5?] képfeldolgozó könyvtár függvényei segítségével már megoldhatóak. Az OpenCV-nek, mint az új típusú kiterjesztett valóság fejlesztőeszközökének, már nincs szükségük fekete keretes markerekre, sem a megjelenítendő virtuális elem pozíciójának kiszámításához, hanem



2. ábra: Objektum azonosítás keretek kivonásából

tetszőleges kép is alkalmazható erre a feladatra. Néhány irodalmi hivatkozás (KaewTraKulPong et al., 2001). azonban arra is felhívja a figyelmet, hogy a pontosabb adatok meghatározása érdekében a mozgó objektumhoz tartozó vetett árnyékot is el kell távolítani a objektum mellől. Természetesen minél kontrasztosabb, részletesebb a kép, annál jobb eredmény érhető el. Sikeres objektum detektálás után, a modellben a markerek helyére illeszthető az öt leíró virtuális elem, melyre a markerek mozgására jellemző paraméterek ismerete esetén pl. a mozgáspályákat meg lehet határozni, vagy ötvözve a képernyőn a virtuális valóságot az tényleges valósággal, össze lehet vetni a megfelelő mozgáspálya transzformációk kiszámítása után hogy a modellfolyamatok mennyire közelítik meg a valóságot. A továbbiakban ezek a markerek lesznek hivatva arra, hogy segítségükkel be tudjuk azonosítani az adott jármű típusát. Az azonosításhoz adatbázisunknak többféle markert is tartalmaznia kell, mert a kamera nézőpontjához igazítva kell



3. ábra: A kontúr és a marker kialakítás lépései

meghatároznunk ezeket a lehetséges markereket. Nem mindegy, hogy a járművet elő- vagy oldalnézetben látjuk-e a megfigyelés során. Az árnyék miatt a 3. ábrán látható jármű azonosítására a bal oldali kontúrvonal melletti lámpák az irányadók, mert a jobboldaliakat csak az árnyék eliminálása után lehet a pontos helyén azonosítani. Ugyancsak ez vonatkozik a visszapillantó tükörre is. Marker pozíció transzformációk segítségével ez a feladat is megoldható, mert ha mondjuk a pozíciókat a baloldali visszapillantó tükörhöz viszonyítva adjuk meg, akkor az árnyékból adódó méretkülönbség nem játszik szerepet az azonosításnál. Ezekben az esetekben természetesen a számításokra fordítható idő szerepét kell meghatároznunk. A méret és a markerek pozícióinak ellenőrzése után kísérrelhetjük meg a jármű azonosítását.

5. EREDMÉNYEK

Az elemzését két országúti **és egy városi** videó felvétel képei alapján készítjük el. Az első felvétel erős, míg a második gyenge nappali fényben készült, így a kontúrok az első videón élesek, azaz elemzéshez csak kismértékű zajt adnak hozzá, míg a másodikon gyengék, azaz a felvétel zajosabb. Ennek hatásait a későbbiekben látni is fogjuk.

5.1 A videó lejátszása

Ezért első lépésként meg kell jeleníteni a videót. Az alábbi programrészlet lehetőséget ad számunkra a felvett videó képkockáinkénti (keretenkénti) lejátszására. A program addig működik, amíg a videó képkockái el nem fogynak.

```
char* name = "Video";
const char* filename = "Video1.avi";
cvNamedWindow(name,CV_WINDOW_AUTOSIZE);
CvCapture* capture = cvCreateFileCapture(filename);
IplImage* frame;
```

```
while (1) {
    frame = cvQueryFrame( capture );
    if( !frame ) break;
    cvShowImage( name, frame );
}
```

```
cvReleaseCapture( &capture );
cvDestroyWindow( name );
...
```

5.2 A háttér kiszűrése

A feldolgozás első lépéseként a videó egymás utáni képkockáiból elő kell állítani a megfigyelt terület háttérét. A háttér a megfigyelt terület azon része, ahol vizsgálandó tevékenység nem történik. A háttér kiszűrése után a képkockáinként vizsgálandó terület mérete általában jelentősen lecsökken.

```
#include<opencv2/opencv.hpp>
#include<iostream>
#include<vector>
```

```
int main(int argc, char *argv[]){
    cv::Mat keret;
    cv::Mat hatter;
    cv::Mat eloter;
    cv::VideoCapture video(0);
    cv::BackgroundSubtractorMOG2 hter;

    hter.nmixtures = 3;
    hter.bShadowDetection = false;

    std::vector<std::vector<cv::Point>> kontur;

    cv::namedWindow("Keret");
    cv::namedWindow("Hatter");
```

```
for(;;) {
    video >> keret;
    hter.operator()(keret,eloter);
    hter.getBackgroundImage(hatter);
    cv::erode(eloter,eloter,cv::Mat());
    cv::dilate(eloter,eloter,cv::Mat());
    cv::findContours(eloter,kontur,
        CV_RETR_EXTERNAL,
        CV_CHAIN_APPROX_NONE);
    cv::drawContours(keret,kontur,-1,
        cv::Scalar(0,0,255),2);
    cv::imshow("Keret",keret);
    cv::imshow("Hatter",hatter);
    if(cv::waitKey(30) >= 0) break;
}
return 0;
}
```

A háttér kiszűrése azonban nem egy egyszeri művelet, állandó frissítést kíván. Hosszabb megfigyelés során a háttér újraszámítása az időjárási vagy a napszaki viszonyok változása miatt fontos feladat. Jelen változatban csak nappali fényben történő vizsgálatokat hajtottunk létre.



4. ábra: A háttér kiszűrése



„IFFK 2014” Budapest

Online: ISBN 978-963-88875-3-5
CD: ISBN 978-963-88875-2-8

CAETS

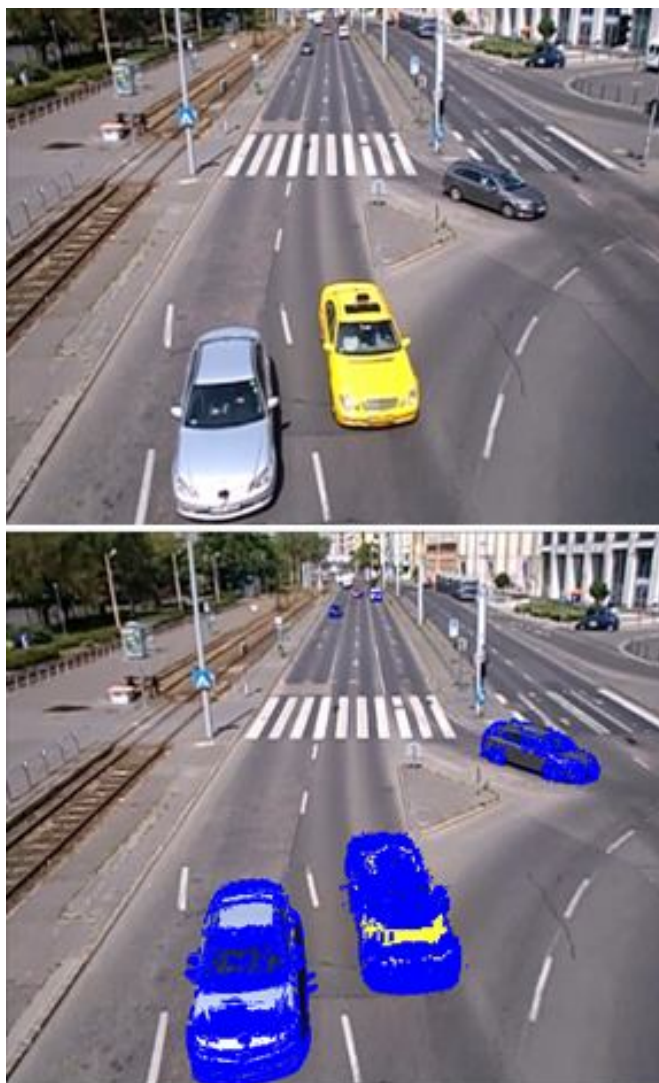
További logók helye

Paper xx

Copyright 2014 Budapest, MMA
Editor: Dr. Péter Tamás

5.3 A mozgó objektumok azonosítása

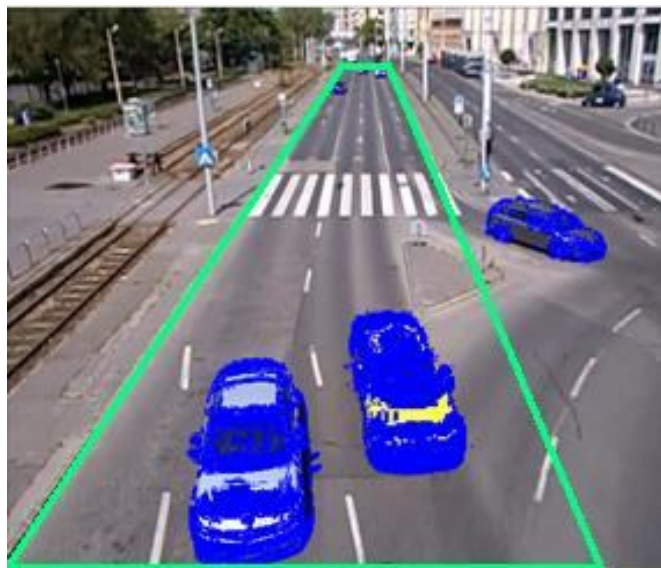
A háttér kiszűrése után még két lépés maradt a típusazonosításig. Ezek közül az első a mozgó objektumok azonosítása és követése addig, amíg a számunkra legjobb nézeti képet nem adja a típusazonosításhoz. Ez általában azt a kép legnagyobb nagyítással rendelkező részét jelenti, ami a legtöbb esetben a kép alsó részén található meg. Az 5.2-ben megadott programrészletet kis átalakítással használhatjuk a mozgó objektumok azonosítására is, ha a háttér kiszűrésével már végeztünk. Az aktuális képkocka és a statikus háttérkép különbsége az aktuális képkockán kitakarja a mozgó objektumot. Az OpenCV lehetőséget ad arra, hogy a felismert mozgó objektumot, ahogy az az 5. ábrán is látszik, kiemeljük.



5. ábra: A mozgó objektumok azonosítása

A kiemelés jól mutatja, hogy az azonosítás a képkocka alsó régiójában működik megfelelően, míg a felsőbb részben az azonosítás nem kielégítő. Néhány mozgó objektum azonosítása után még pontosabban meghatározhatjuk az előtér méreteit. Ha az 5. ábrán csak a szembejövő forgalmat vizsgáljuk, akkor a megfigyelés alá bevonható területet a 6.

ábra mutatja. A 6. ábrán újonnan kijelölt terület nem sokban tér el az eredetileg kijelölt előtértől, de a munkaterület csökkenése minden esetben értékes CPU idő megtakarítást is jelent.



6. ábra: A vizsgálandó munkaterület csökkentése

5.4 Típus azonosítás

Feladatunk befejezésének utolsó lépése maga a típusazonosítás. Amennyiben megfelelő markereket választottunk, úgy az azonosítást könnyen elvégezhetjük. A 7. ábrán ennek a folyamatnak a menetét követhetjük végig. Az azonosítás során az SURF (Speed Up Robust Features) algoritmust használjuk. Az algoritmus lényege, hogy maga az eljárás a kép transzformációktól független. A jellegzetességek, a markerek megfeleltetésével jól alkalmazható objektumok azonosítására. Az eljárás során markereket és integrálképeket is megadhatunk, attól függően, hogy milyen adatbázis áll a rendelkezésünkre. A 7. ábra felső részén láthatjuk, hogy az azonosításra szánt markerek egy részét programunk nem a vizsgált objektumon találja meg, de a mozgó objektum minél inkább a képkocka alsó régiójába ér, annál pontosabbá válik a felismerés. Külön érdekessége a vizsgálatnak, hogy a fekete-fehér és a színes azonosítási folyamatokat összevetve, a fekete-fehér azonosítások pontosabb eredményt szolgáltattak, mint a színesek.

6. FEJLESZTÉSI LEHETŐSÉGEK

Jelenlegi állapotában a megfigyelő rendszerünk, egy időben csak egyetlen mozgó objektum azonosítására alkalmas. Erre is csak abban az esetben, ha a vizsgált objektum markerei a rendelkezésére állnak. A szín és a mozgó objektumok lévő feliratok meg gátolják az azonosítást viszont, ha jelentős eltérés van jármű méretében, pl. nagyméretű tárgy a felső csomagtartóban, akkor a felismerés nagyon bizonytalanra válik.

A fejlesztések egyik iránya tehát az egy képkockán található járművek számának növelése, ami egyrészt az azonosítandó



„IFFK 2014” Budapest

Online: ISBN 978-963-88875-3-5

CD: ISBN 978-963-88875-2-8

CAETS

További lapok helye

Paper xx

Copyright 2014 Budapest, MMA

Editor: Dr. Péter Tamás

objektumok számának növelését másrészt, pedig az egyidejű azonosítások számának növelését is jelenti.

A fejlesztések másik irányát a bonyolultabb vizsgálati feladatok kialakítása jelentené. Ez nem csak a sebesség mérését vagy a sávváltások azonosítását értjük, hanem az elkövetett közlekedési szabálytalanságok felismerését is. Jelenleg már folynak kísérletek arra, hogy néhány egyszerű szabálytalanságot, pl. a gyorsajtást azonosítsuk. Ugyancsak elkezdjük a gyakori sávváltások miatt kialakuló veszélyhelyzetek képi vizsgálatát is.



7. ábra: A jármű azonosítás lépései

FELHASZNÁLT IRODALOM

Max, G., (2007). *Képfeldolgozó eljárások a közlekedésben*, Alkalmazott Informatikai Kongresszus, Kaposvár

Max, G., (2012). *Közlekedési szabálytalanságok*, IFFK 2012, Budapest, ISBN 978-963-88875-2-8

Max, G., (2013). *Gépjárműtípus felismerés*, IFFK 2013, Budapest, ISBN 978-963-88875-2-8

Zafar, I., Acar, B.S., Edirisinghe, E.A. (2007). *Vehicle make & model identification using scale invariant transforms*, 7th International Conference on Visualization, Imaging and Image Processing, pp.271-276, ISBN: 978-0-88986-692-8

KaewTraKulPong P., Bowden R. (2001). *An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection*, 2nd European Workshop on Advanced Video Based Surveillance Systems, Kluwer Academic Publishers

Horaud R., Knossow D., Michaelis M., (2006), *Camera cooperation for achieving visual attention*, Machine Vision and Applications (2006) 16(6) pp. 331–342

Mikic I., Cosman P. C., Kogut G. T., Trivedi M. M., (2003), *Moving Shadows and Object Detection in Traffic Scenes*, Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, Vol.1, pp.321-324

Yoneyama A., Yeh C. H., Jay Kuo C.-C., (2003) *Moving cast shadow elimination for robust vehicle extraction based on 2D joint vehicle/shadow models*, IEEE Proc. International Conference on Advanced Video and Signal Based Surveillance, pp. 229-236



CAETS

„IFFK 2014” Budapest

Online: ISBN 978-963-88875-3-5

CD: ISBN 978-963-88875-2-8

További lapok helye

Paper xx

Copyright 2014 Budapest, MMA

Editor: Dr. Péter Tamás