

Laboratorium 14

Na koniec zajęć wysyłamy pliki z wykonanymi ćwiczeniami na adres **dydaktyka.sk@gmail.com**. Pliki NALEŻY nazywać w następujący sposób: **imię_nazwisko_nrindexu_zadanieX.cpp** a w tytule maila podać: **Imię i Nazwisko**. Proszę nie opuszczać sali laboratoryjnej zanim prowadzący nie potwierdzi otrzymania maila z zadaniami. W przypadku nie otrzymania maila przez prowadzącego z jakichkolwiek przyczyn obecność na zajęciach nie zostanie uznana.

Zaawansowane wykrywanie i zliczanie ruchomych obiektów

Na poprzednich zajęciach zajmowaliśmy się prostym wykrywaniem i śledzeniem ruchomych obiektów na przykładzie samochodów pędzących po autostradzie (film z <http://goo.gl/NokSmd>). Dzisiaj spróbujemy podejścia nieco bardziej zaawansowanego, a żeby sobie jeszcze bardziej skomplikować życie dorzucimy zliczanie przejeżdżających samochodów.

W OpenCV poza wykrywaniem obiektów przy użyciu **Haar Cascades** (którego należy odpowiednio "przyuczyć" do wykrywania), algorytmu odejmowania tła (**Background Subtraction** z poprzedniej listy) powiązanego z prostym wykrywaniem niekształtnych obiektów, przyszedł czas na bardziej poprawne podejście. Dzisiaj do algorytmu odejmowania tła dorzucimy wykrywanie obiektów wielokonturowych. Ale zanim do tego dojdziemy zacznijmy standardowo naszą funkcję *main()*:

```
int main()
{
    VideoCapture film ("video1.avi");
    if (!film.isOpened())
    {
        cout<<"Nie moge odtworzyc pliku."<<endl;
        return (-1);
    }

    while(waitKey(30)!=27)
    {
        bool wczytano = film.read(klatka);
        if(!wczytano) break;
        ~~~
        ...
    }
}
```

Tym razem pojawiła się nowa linijka *if(!wczytano) break;* zabezpieczająca program przed "wysypaniem się" w przypadku doczytania pliku do końca. Aby było całkiem poprawnie *main()* powinna zakończyć się również czym więcej niż samym *return 0;*

```

...
}
cvDestroyAllWindows();
film.release();
return (0);
}

```

Zmienne globalne oraz *include'y* pozostaną takie jak poprzednim razem. Spróbujmy jednak wycisnąć więcej ze sposobu odcinania tła. Mi udało się uzyskać całkiem dobre parametry przy następującym podejściu:

```

MOG(klatka, foregroundMask);
imshow("Stage1", foregroundMask);
threshold(foregroundMask, binary, 200, 255, CV_THRESH_BINARY);
imshow("Stage2", binary);
blur(binary, binary, Size(20, 20) ); //Rozmycie
imshow("Stage3", binary);
Mat element(4, 4, CV_8U, Scalar(1));
erode(binary, binary, element); //Erozja
imshow("Stage4", binary);
threshold(binary, binary, 60, 255, CV_THRESH_BINARY);
imshow("Detection base", binary);

```

Jak możecie zaobserwować w poszczególnych oknach, podwójne filtrowanie do obrazu binarnego jest niezbędne. Przy załadowaniu innego filmu może być oczywiście konieczne dopasowanie odpowiednich parametrów nad którymi już (mam nadzieję) panujecie.

Spróbujmy obrobić teraz klatkę filmu przy użyciu poznanej już wcześniej funkcji *findContours()*:

```

vector<vector<Point> > contours;
vector<Point> contours_poly;
Rect boundRect;
findContours( binary.clone(), contours, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE, Point(0, 0) );
int max = 0, i_cont = 0;
for( int i = 0; i < contours.size(); i++ )
{
    approxPolyDP( Mat(contours[i]), contours_poly, 3, true );
    boundRect = boundingRect( Mat(contours_poly) );
    rectangle( klatka, boundRect.tl(), boundRect.br(), Scalar(125, 250, 125), 2, 8, 0 );
    line( klatka, boundRect.tl(), boundRect.br(), Scalar(250, 125, 125), 2, 8, 0);
    i_cont++; //zliczanie ilości obiektów
}

```

Do pętli *for* wstawiamy licznik obiektów *i_cont*, który wykorzystamy za chwilę.

Moment, moment... która zmienna jest tutaj bezużyteczna niczemu nie służąc i można ją usunąć?

A teraz ostateczna manipulacja kodem:

```

//wyświetlanie ilości obiektów
stringstream out;
out << i_cont << "cars";
putText( klatka, out.str(), Point(220, 220), CV_FONT_HERSHEY_COMPLEX, 1, Scalar(0, 255, 250), 2, 8 );
imshow("Detection2", klatka);

```

i powinniście uzyskać następujący efekt:



Jak widać efekt nie jest doskonały...

HINT: Aby spowolnić odtwarzanie filmu można wykorzystać funkcję `_sleep(100)`; gdzie wartość czasu podajemy w milisekundach.

ŻYCZYMY WIELU SUKCESÓW W DALSZYM PROGRAMOWANIU