

Laborator 3 SRF: Detectia obiectelor aflate în mișcare.

Librăria OpenCV

Scopul lucrării

Scopul acestei lucrări este familiarizarea studentului cu librăria OpenCV. La finalul acestui laborator studenții vor fi capabili să implementeze un algoritm de detașare față de fundal a obiectelor aflate în mișcare, implementând, pe baza metodelor OpenCV, un algoritm de eliminare a fundalului.

1. Breviar teoretic:

OpenCV este o librărie de funcții cu aplicații în sistemele computerizate de urmărire a imaginilor în timp real (**realtime computer vision**). A fost dezvoltată inițial de Intel pe baza C++, iar în prezent funcționează pe toate platformele.

1.1. Aplicații ale OpenCV

Inițial, librăria OpenCV era gândită pentru a realiza ray-tracing în timp real și pereti de afișaj 3D. În prezent poate fi folosită în aplicații de:

- odometrie vizuală (egomotion: determinarea poziției și orientării spațiale a robotului prin analiza imaginilor provenite de la camera video),
- recunoaștere facială (face recognition: tehnologie capabilă să detecteze o imagine facială și să facă corespondența cu o imagine asemănătoare din cadrul unei baze de date)
- recunoașterea gesturilor (gesture recognition: interpretarea cu ajutorul calculatorului a gesturilor umane folosind algoritmi matematici)
- interacțiunea om-mășină (human-computer interaction: studiază interfața de comunicare dintre om și calculator și căi pentru a o îmbunătăți)
- robotica și detecția obstacolelor
- detecția obiectelor și segmentarea imaginilor
- stereoscopie (stereopsis : percepția adâncimii și a distanțelor dintre obiectele unei imagini folosind vederea stereoscopică, adică 2 camere de filmat în loc de una, aflate la o mică distanță una de cealaltă)
- urmărirea mișcărilor
- realitate augmentată (augmented reality)

1.2. Algoritmi

- Arbori de decizie/învățare (decision tree learning)
- Padure de arbori de decizie (random forest classifier)

- Algoritmul speranta-maximizare (expectation-maximization)
- Metoda k-NN (k-Nearest Neighbours)
- Clasificatorul Bayesian naiv (Naive Bayes Classifier)
- Retele neuronale artificiale (artificial neural networks)
- Retele neuronale profunde (deep neural networks)
- Masina cu vectori-suport (Support-Vector Machine)

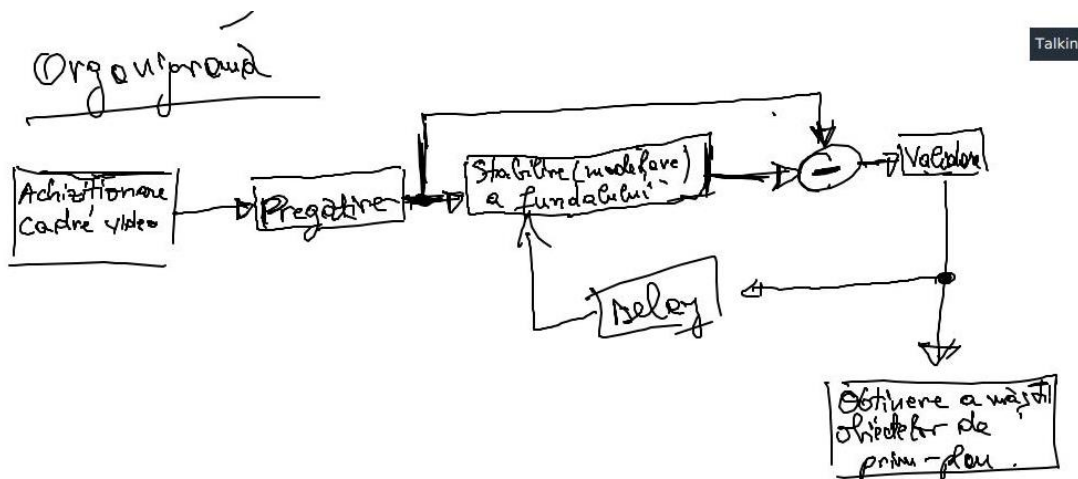
1.3. Background subtraction

Background Subtraction (Substracția fundalului): este un caz particular de substracție a imaginilor, pentru care diferența se realizează între pixelii cadrului curent și pixelii unei imagini de referință (numită fundal).

Aplicații:

- Securitate/Supraveghere video
- Monitorizare/Analiza trafic
- Sisteme de urmărire (tracking)

Organigrama generală:



Mască obiectelor de prim-plan: este o imagine binară (pe două niveluri) în care cu unul dintre niveluri (alb) sunt figurate obiectele de prim-plan, iar cu celălalt nivel (negru) sunt reprezentați pixelii fundalului.

Cei mai buni algoritmi pentru stabilirea fundalului:

-> Metoda a diferenței cadrelor: este o metoda ne-recursiva care memoreaza intr-un buffer un anumit numar de cadre pe care le foloseste, prin mediere, la estimarea fundalului.

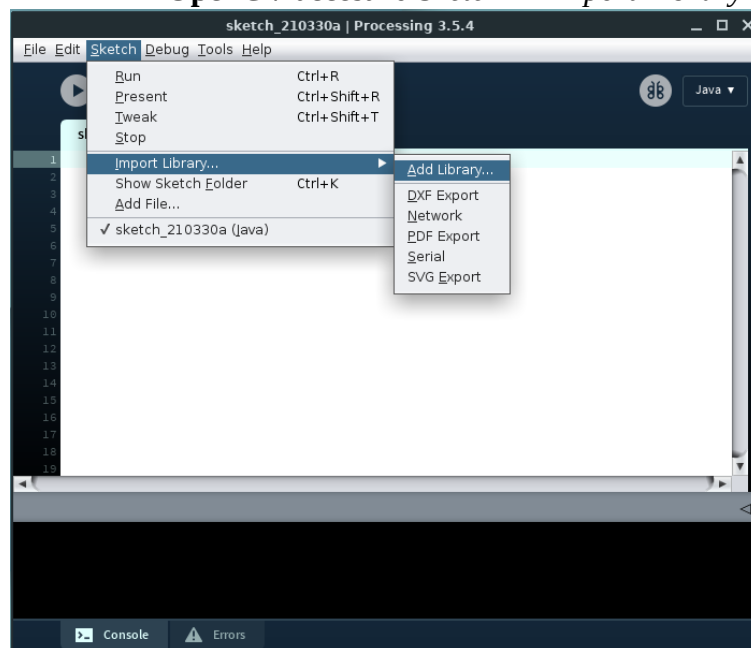
-> Metoda sumei de Gaussiene (Mixture of Gaussians): fundalul nu este modelat ca o matrice de pixeli de culoare fixa, ci ca o suma ponderata de funcții Gaussiene.

2. Desfășurarea lucrării:

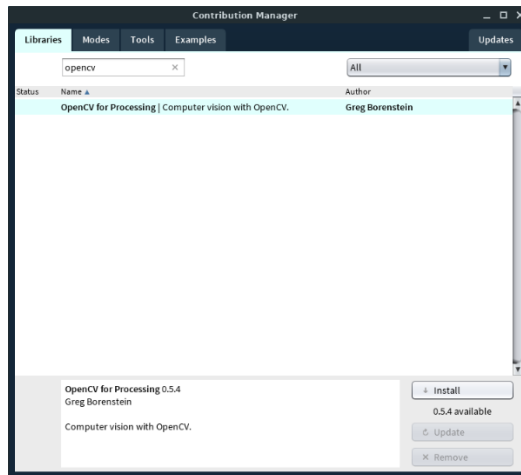
Lucrarea de față își propune, pe baza exemplificării unui program preexistent odată cu instalarea librăriei OpenCV, să urmărească tematica de detecție a obiectelor aflate în mișcare într-un videoclip oferit ca resursă. Videoul de test trebuie filmat folosind o cameră fixă, deoarece fundalul trebuie să rămână fix pe toată durata înregistrării video.

Ca etapă preliminară, va trebui să importăm librăria **OpenCV** și deschiderea fișierului-exemplu, urmând pașii următori:

1) Instalarea librăriei **OpenCV** accesând *Sketch -> Import Library -> AddLibrary*

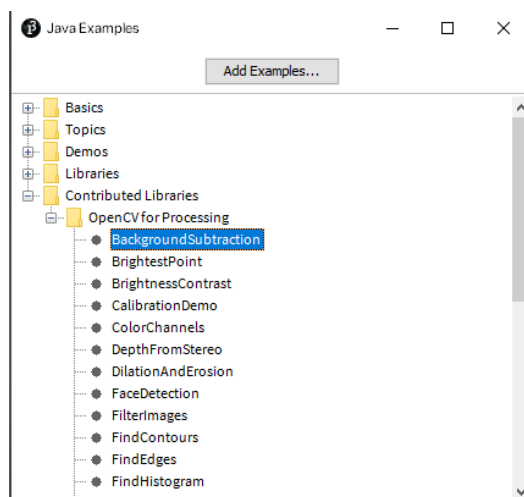


2) Apăsăm butonul *Install*



3) Deschiderea exemplului prin

*File -> Examples -> Contributed Libraries -> OpenCV for Processing -> **BackgroundSubtraction***



Ni se va deschide un program pe baza căruia vom face rezolvarea exercițiilor propuse mai jos.

ATENȚIE:

1) *Observăm că la prima rulare a aplicației, programul poate afișa un mesaj de eroare.*

*Pentru îndreptarea sa, în interiorul funcției **setup()**, trebuie adăugată linia de cod selectată cu galben din figura următoare.*

```

void setup() {
  size(720, 480);
  video = new Movie(this, "street.mov");
  opencv = new OpenCV(this, 720, 480);

  opencv.startBackgroundSubtraction(5, 3, 0.5);

  video.loop();
  while(video.height==0) delay(2);
  video.play();
}

```

- 2) Pentru studenții care folosesc mediul de lucru Processing pe Windows, se va afișa o atenționare de tipul care se referă la faptul că nu există o sursă reală de captare a informației video de tip webcam. Aceasta nu va influența algoritmul în vreun fel.

WARNING: no real random source present!

Programul inițial, după repararea erorii discutate anterior, arată așadar în felul următor:

```

void setup() {
  size(720, 480);
  video = new Movie(this, "street.mov");
  opencv = new OpenCV(this, 720, 480);

  opencv.startBackgroundSubtraction(5, 3, 0.5);

  video.loop();
  while(video.height==0) delay(2);
  video.play();
}

void draw() {
  image(video, 0, 0);
  opencv.loadImage(video);

  opencv.updateBackground();

  opencv.dilate();
  opencv.erode();

  noFill();
  stroke(255, 0, 0);
  strokeWeight(3);
  for (Contour contour : opencv.findContours()) {
    contour.draw();
  }
}

void movieEvent(Movie m) {
  m.read();
}

```

Cerinte:

- 1) Explicați și comentați codul programului.
- 2) Modificați programul încât, la apăsarea tastei Space, videoclipul se va opri (pauză) iar, la reapăsarea tastei, videoclipul va reporni de unde a rămas;
- 3) Afișați numărul cadrului în partea stânga sus.
- 4) Modificați programul încât chenarele de selecție al obiectelor de mișcare să se afișeze doar dacă obiectele depășesc un anumit prag de dimensiune (threshold). Vom folosi tasta a pentru a mări pragul și tasta z pentru a micșora pragul;
- 5) Afișați, în locul videoclipului inițial, cadrele care rezultă în urma aplicării algoritmului de BackgroundSubtraction pe acest videoclip;
- 6) Aplicați funcția blur() folosind diferiți parametri de intensitate, înainte de aplicarea algoritmului de BackgroundSubtraction (observăm îmbunătățiri la nivelul de eliminare al zgomotului);
- 7) Afișați, în paralel, videoclipul inițial și imaginea binară folosită de obiectul OpenCV pentru detecția conturilor obiectelor ce se mișcă.

Rezolvare:

- 2) Pentru a putea rezolva acest exercițiu, este necesar să parcurgem următorii pași:

- la începutul programului vom adăuga o variabilă de tip **boolean** numită *rulează* și îi vom da valoarea **true**: `boolean ruleaza=true;`
- ne vom folosi de metoda **keyPressed()** pentru a putea desemna acțiunea de a opri și respectiv de a reporni videoclipul respectiv. Totodată, când tasta Space (fie caracterul ' ' sau codul ASCII 32) este apăsată, ne vom folosi de funcțiile **noLoop()** și de **video.pause()** pentru a putea opri videoclipul și a păstra activitatea aceasta într-o buclă.

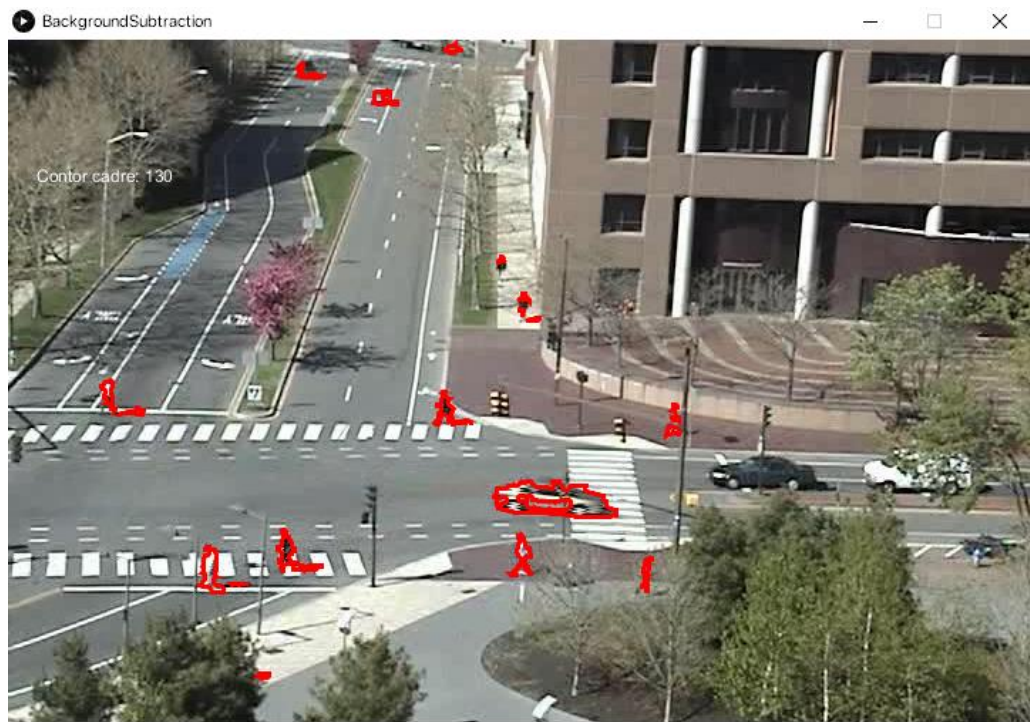
```
public void keyPressed(){  
  
    if(key == 32){  
        if(ruleaza) {noLoop(); video.pause();}  
        if(!ruleaza){ loop(); video.loop(); }  
        ruleaza=!ruleaza;  
    }  
}
```

După ce am introdus aceste linii de cod, vom observa că, la apăsarea tastei **Space**, videoclipul se va opri și va rămâne așa iar la reapăsarea acestei taste, videoclipul va continua să curgă.

- 3) Pentru a putea afișa numărul de cadre afișate în momentul de timp, ne vom folosi de variabila **frameCount**. În cadrul funcției **draw()** vom introduce linia de cod aflată mai jos. Ultimii parametri reprezintă coordonatele x și y pentru a putea fi plasat textul (în stânga sus, unde noi dorim să fie).

```
text("Contor cadre: " + frameCount, 20,100);
```

Vom observa că în acea zonă ne va apărea framerate-ul:

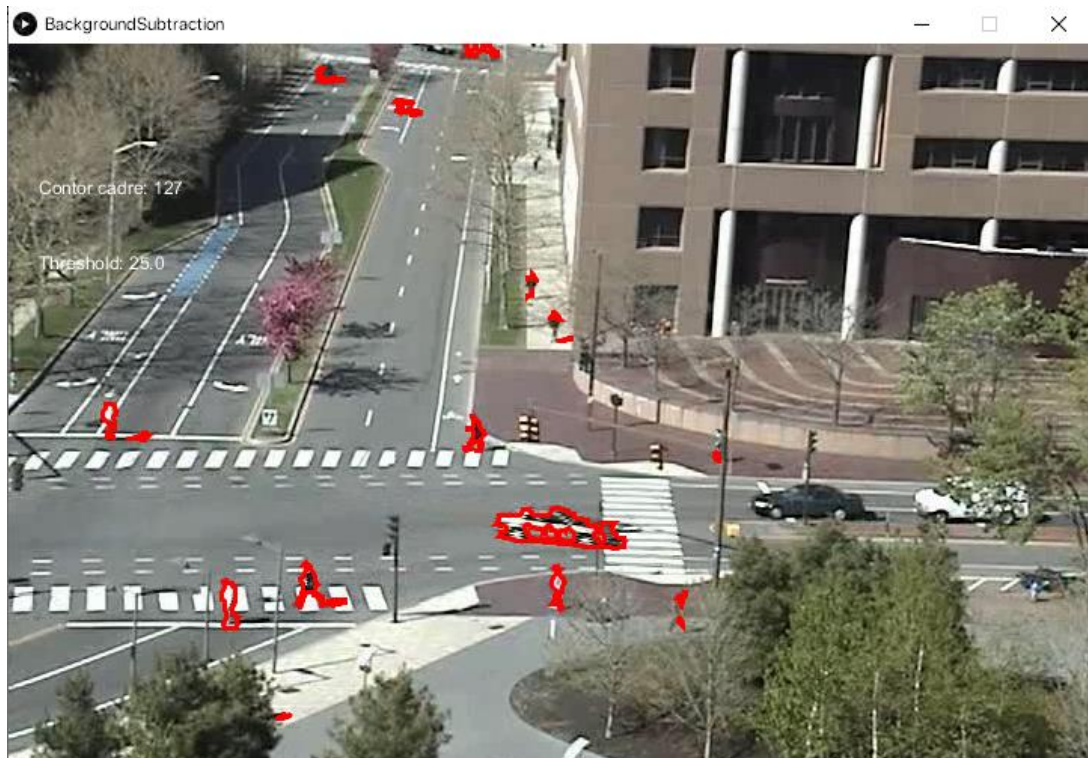


4)

- Vom avea nevoie de o variabilă numită `threshold` care va avea valoarea 25: `float threshold = 25;`
- În interiorul funcției `keyPressed()`, vom avea nevoie să interogăm folosirea tastei `a` și a tastei `z` în felul următor:

```
public void keyPressed(){  
    if(key == 32){  
        if(ruleaza) {noLoop(); video.pause();}  
        if(!ruleaza){ loop(); video.loop(); }  
        ruleaza=!ruleaza;  
    }  
  
    // Pentru threshold  
    if (key == 'a') {  
        threshold+=25;  
    } else if (key == 'z') {  
        threshold-=25;  
    }  
}
```

Rezultatul este următorul:



5) Acest algoritm separă informația neesențială de cea esențială (obiectele aflate în mișcare), lăsând vizibile doar elementele care sunt dinamice. Așadar, în interiorul metodei **draw()**, vom aplica liniile de cod de mai jos:

```
PImage videoProcesat = opencv.getSnapshot();
image(videoProcesat,0,0);
```

Rezultatul este următorul:

