UNIVERSITATEA ”ALEXANDRU-IOAN CUZA” DIN IAS, I

**FACULTATEA DE INFORMATICA˘**



LUCRARE DE LICENT, A˘

**Object tracking**

propusa˘ de

**Mihai-Cătălin Bujor**

**Sesiunea:** iulie, 2019

### Coordonator s, tiint, ific

UNIVERSITATEA ”ALEXANDRU-IOAN CUZA” DIN IAS, I

**FACULTATEA DE INFORMATICA˘**

**Object tracking**

**Mihai-Cătălin Bujor**

**Sesiunea:** iulie, 2019

### Coordonator s, tiint, ific

Avizat, Iˆndruma˘tor lucrare de licent, a˘, Conf. Dr. Ionescu Ionel.

Data: ............................ Semna˘tura: ............................

#### Declarat, ie privind originalitatea cont, inutului lucra˘rii de licent, a˘

Subsemnatul **Mihai-Cătălin Bujor** domiciliat ˆın **Romaˆnia, jud. Vaslui, mun. Vaslui, strada Ștefan cel Mare bl. 173, et. 3, ap. 5**, na˘scut la data de **14 noiembrie 1997**, identificat prin CNP **1234567891234**, absolvent al Faculta˘t, ii de informatica˘, **Facultatea de informatica˘** specializarea **informatica˘**, promoția 2019, declar pe propria ra˘spundere cunoscaˆnd consecint, ele falsului ˆın declarat, ii ˆın sensul art. 326 din Noul Cod Penal s, i dispozit, iile Legii Educat, iei Nat, ionale nr. 1/2011 art. 143 al. 4 s, i 5 referitoare la pla- giat, ca˘ lucrarea de licent, a˘ cu titlul **Object tracking** elaborata˘ sub ˆındrumarea doamnei **Lect. Dr. Ignat Anca**, pe care urmeaza˘ sa˘ o sust, in ˆın fat, a comisiei este originala˘, ˆımi apart, ine s, i ˆımi asum cont, inutul sa˘u ˆın ˆıntregime.

De asemenea, declar ca˘ sunt de acord ca lucrarea mea de licent, a˘ sa˘ fie verificata˘

prin orice modalitate legala˘ pentru confirmarea originalita˘t, ii, consimt, ind inclusiv la

introducerea cont, inutului ei ˆıntr-o baza˘ de date ˆın acest scop.

Am luat la cunos, tint, a˘

despre faptul ca˘

este interzisa˘

comercializarea de lucra˘ri

s, tiint, ifice ˆın vederea facilita˘rii falsifica˘rii de ca˘tre cumpa˘ra˘tor a calita˘t, ii de autor al unei lucra˘ri de licent, a˘, de diploma˘ sau de disertat, ie s, i ˆın acest sens, declar pe proprie ra˘spundere ca˘ lucrarea de fat, a˘ nu a fost copiata˘ ci reprezinta˘ rodul cerceta˘rii pe care am ˆıntreprins-o.

Data: ............................ Semna˘tura: ............................

#### Declarat, ie de consimt, a˘maˆnt

Prin prezenta declar ca˘

sunt de acord ca lucrarea de licent, a˘

cu titlul **Object**

**tracking**, codul sursa˘ al programelor s, i celelalte cont, inuturi (gra- fice, multimedia, date de test, etc.) care ˆınsot, esc aceasta˘ lucrare sa˘ fie utilizate ˆın cadrul Faculta˘t, ii de informatica˘.

De asemenea, sunt de acord ca Facultatea de informatica˘ de la Universitatea

”Alexandru-Ioan Cuza” din Ias, i, sa˘

utilizeze, modifice, reproduca˘

s, i sa˘

distribuie ˆın

scopuri necomerciale programele-calculator, format executabil s, i sursa˘, realizate de mine ˆın cadrul prezentei lucra˘ri de licent, a˘.

Absolvent **Mihai-Cătălin Bujor**

Data: ............................ Semna˘tura: ............................

# Cuprins

[Motivat, ie](#_bookmark0) 2

[Introducere](#_bookmark1) 3

1. Colectarea Datelor de test 4

[1.1](#_bookmark3) Criteriile ce au stat la baza selecției . . . . . . . . . . . . . . . . . . . . . 4

[1.2 Titlul sect, iunii 2](#_bookmark4) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

1. [Titlul celui de-al doilea capitol](#_bookmark5) 6

[2.1 Titlul sect, iunii 1](#_bookmark6) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

[2.2 Titlul sect, iunii 2](#_bookmark7) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

[2.3 Titlul sect, iunii 3](#_bookmark8) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

1. [Titlul celui de-al treilea capitol](#_bookmark9) 8

[3.1 Titlul sect, iunii 1](#_bookmark10) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

[3.2 Titlul sect, iunii 2](#_bookmark11) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

[Concluzii](#_bookmark12) 10

[Bibliografie](#_bookmark13) 11

# Motivat,ie

# Introducere

# Capitolul 1

**Colectarea datelor de test**

Primul pas in procesul dezvoltării a fost reprezentat de colectarea datelor de test reprezentate de fisiere in format video . Selecția acestora s-a făcut în mare parte folosind platforma YouTube 8M precum și proiecte open-source cu tematici similare. Ulterior selecției datelor si măsurătorilor realizate am conceput un standard privind dimensiunea datelor de intrare pentru a oferi rezultate cât mai exacte astfel fiecare video primit ca input este redimensionat la următoarele dimensiuni: (lățime 1280p, lungime 680p).

## Criteriile ce au stat la baza selecției

În procesul de colectare a datelor de intrare am urmărit ca acestea să îndeplinească următoarele criterii:

* să prezinte un subiect aflat în mișcare sau ce urmează a se pune în mișcare
* să prezinte zgomote reprezentate de obstacole , schimbări de lumină sau fluctuații de viteză pentru a putea acoperi o gamă cât mai largă de cazuri.
* să prezinte un subiect diversificat din punct de vedere al dimensiunii, culorii și poziției sale inițiale.
* video-ul să fie realizat folosind o cameră stabilă (în care diferența dintre frame-urile consecutive să nu fie mare)

## Restricție privind datele de intrare

În vederea obținerii rezultatelor corecte metodele folosite prezintă o restricție privind datele de intrare ce constă în faptul că video-ul trebuie realizat cu o cameră stabilă. Această condiție este necesară deoarece detecția subiectului respectiv a backgroundului se realizează în fiecare frame din input alături de predicția privind următoarele posibile poziții ale subiectului (bounding-boxuri).

Bounding boxurile reprezintă coordonatele dreptunghiului ce cuprinde o imagine în care se poziționează obiectul ce se dorește a fi urmărit. Dacă pe parcursul video-ului subiectul nu mai este prezent în frame bounding boxul dispare de asemenea. Cazul în care video-ul este realizat cu o cameră aflată în mișcare (în care diferența dintre frame-urile consecutive este mare) conduce la calcularea greșită a viitoarelor posibile poziții deoarece această predictiție se realizează folosind informațiile referitoare la poziția inițială a subiectului din frame-ul precedent . Cu cât diferența dintre frame-uri este mai mare cu atât riscul de a “pierde” urma subiectului este mai mare.

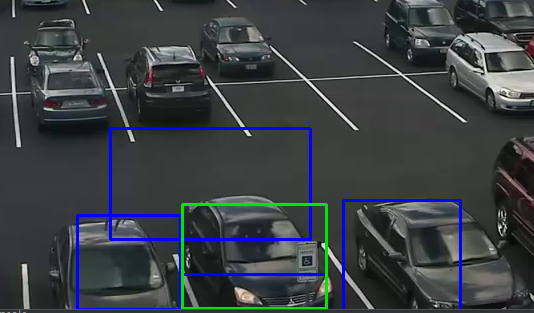


Figura 1

În imaginea de mai sus este prezentat modul în care se realizează predicția viitoarelor poziții (culoarea albastră) pentru un bounding-box oferit ca input de către user (culoare verde). După cum se poate observa metoda folosită estimează toate cele 4 modalități de deplasare posibile pentru subiect (înainte , înapoi, stănga ,dreapta) datorită faptului că nu se cunoaște poziția sa initială și direcția de deplasare . Folosind această modalitate sunt acoperite si cazurile în care subiectul (pentru cazul de mai sus) s-ar afla orientat în altă direcție. (de exemplu orizontal)

# Capitolul 2

**Metodele de tracking folosite.**

Procesul inițial de dezvoltare a constat în utilizarea și efectuarea unor măsuratori asupra a 5 metode din biblioteca OpenCV :

* Boosting Tracker (OpenCV 3.0.0)
* Mil Tracker (OpenCV 3.0.0)
* CSRT Tracker (OpenCV 3.4.2)
* KCF Tracker (OpenCV 3.1.0)
* MedianFlow Tracker (OpenCV 3.0.0)

## Boosting Tracker

Reprezintă unul din cele mai vechi metode din cele menționate. Este bazat pe o variantă “on-line” a algoritmul AdaBoost. Diferența dintre versiunea “on-line” si cea “off-line” a algoritmului este dată de modul în care se obțin datele și de modul în care este construit clasificatorul puternic. Versiunea de antrenament “off-line” beneficiază de toate datele din avans. Pe de altă parte versiunea “on-line” folosește câte un eșantion pentru fiecare iterație. Construcția clasificatorului în versiunea de antrenament “off-line” se face adăugând un clasificator slab în fiecare rundă pe când în versiunea on-line clasificatorul este inițializat la început și actualizat după fiecare eșantion. Pentru a înțelege modul de funcționare a acestui algoritm este necesar să definim următorii termeni :

**Clasificator slab:** Acest clasificator obține rezultate puțin mai bune decât selecția aleatoare, de exemplu pentru o problemă binară de decizie rata erorii trebuie să fie mai mică decât 50%). Ipoteza slabă obținută de un clasificator slab corespunde unei trăsături și se obține prin aplicarea unui algoritm de învățare.

**Selector :** Dată o mulțime de M clasificatori slabi ce au următoarele ipoteze :

Selectorul alege exact una din acestea , unde m este ales în funcție de criteriul de optimizare.

( , unde reprezintă eroarea estimată pentru fiecare clasificator slab).

**Clasificator puternic :** Dat un set de N clasificatori slabi , un clasificator puternic este calculat printr-o combinație liniară de selectori. Mai mult valoarea de încredere (o notăm cu conf) poate fi considerată drept o măsurătoare pentru acest tip de clasificator.

Ideea principala a acestui algoritm este introducerea selectorilor. Aceștia sunt inițializați aleator, iar fiecare din ei deține un grup separat de clase de clasificatori slabi. La întâmpinarea unui nou eșantion de formare , clasificatorii slabi ai fiecarui selector sunt actualizați. Această procedură de actualizare este cea care necesită cel mare timp de procesare. Considerând toate acestea problema de urmărire poate fi formulată ca o problemă binară de clasificare pentru a obține robustețe prin actualizarea continuă a clasificatorului actual al subiectului. Așa cum am menționat mai sus procesul de urmărire a subiectului face presupunerea că detecția sa a fost deja realizată. Regiunea inițială (bounding-boxul ) este considerat imagine pozitivă pentru tracker. Totodata sunt stabilite și regiunile negative (de aceeasi dimensiune) ca fiind regiunile învecinate cu acesta ce vor fi considerate drept fundal. Cu ajutorul acestora putem itera pentru a obține un prim model care este deja stabil. Procedura de antrenare selectează clasificatorul cu eroarea estimată minimală pentru a face parte din clasificatorul puternic. Pentru a putea estima eroarea clasificatorilor slabi se consideră 𝞴 denumit importanța unui eșantion ce este propagată prin setul de N selectori. În faza inițială se consideră 𝞴=1. Valoarea acetuia crește daca eșantionul este greșit clasificat și invers altfel. În final clasificatorul puternic se obține sub forma unei combinații liniare formată din selectori.

În contextul “tracking-ului” clasificatorul executa 4 pași :

1. Dată o poziție în runda i a subiectului stabilește acesta stabilește zona de căutare conform imaginii de mai sus .
2. Pentru fiecare subzona ce aparține zonei de căutare atribuie o pondere conform procedurii de mai sus . (Inițial ponderile sunt egale pentru toate vecinătățile)
3. Determină vecinătatea cu ponderea cea mai bună
4. Actualizează clasificatorul cu noua pondere.

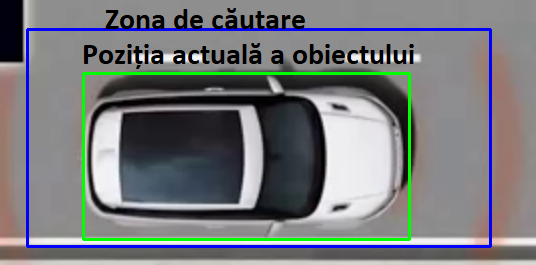


Figura2

## Mil Tracker

Mil (Multiple instance learning) reprezintă o metodă de învățare supervizată din învățarea automată. Aceasta presupune găsirea unei funcții ce realizează o asociere între datele de intrare și cele de iesire în baza unor alte perechi de tipul (date de intrare, date de ieșire) ce sunt considerate date de antrenament (de aici și numele de învățare). Scopul acestei proceduri este de a aproxima această funcție astfel încât aceasta să fie capabilă să clasifice noi date de intrare (date de test ) în mod corect.

Dacă în mod tradițional se utilizează un set de instanțe ce sunt clasificate individual în timp ce MIL folosește o mulțime de astfel de instanțe clasificate împreună ce poartă numele de “ bags”. Cu alte cuvinte pentru a estima în mod tradițional este necesar un set de date de forma unde reprezintă o instanță (în cazul nostru un vector format din imaginile corespunzătoare vecinătăților) și eticheta instanței. În timp ce pentru MIL datele de antrenament au forma unde un „bag” } , iar reprezintă eticheta “bag-ului” definită astfel :

Cum sunt clasificate aceste mulțimi ? În cazul cel mai simplu al clasificării binare o astfel de mulțime se clasifică negativ dacă toate instanțele sale sunt negative, respectiv pozitiv dacă aceasta conține cel putin o instanță clasificată pozitiv. Pentru a putea clasifica “bag-urile” ne propunem să antrenăm un clasificator de tip boosting pentru a maximiza funcția de verosimilitate a mulțimilor.

Având în vedere faptul că această funcție este definită pentru “bag-uri”, iar scopul nostru este de a obține un clasificator ce estimează , trebuie să exprimăm (probabilitatea ca un “bag” să fie clasificat pozitiv) în funcție de fiecare instanță recurgem la modelul următor :

Această ecuație are proprietatea că dacă una din instanțe are probabilitate mare atunci probabilitatea întregului “bag” este de asemenea mare.

În contextul “tracking-ului” dificultatea constă în selectarea datelor ce reprezintă un exemplu pozitiv respectiv negativ. Cele mai multe metode printre care și boosting-ul prezentat mai sus folosesc drept exemplu pozitiv locația curentă a subiectului în timp ce vecinătățile acestuia sunt clasificate drept instanțe negative. Dacă în schimb locația inițială nu este precisă modelul este actualizat în baza unor date parțial corecte. Ca urmare după un număr de iterații de acest tip există posibilitatea ca “tracker-ul” să piardă urma obiectului .

Pe de altă parte o altă modalitate constă în utilizarea mai multor exemple pozitive formate din cele mai apropiate vecinătăți ale subiectului . În cazul nostru aceste exemple positive sunt reprezentate de “bounding-boxurile” din jurul fiecărui obiect targetat urmând ca aceste date să fie trimise către algoritmul de învățare pentru a se decide care dintre ele este cel mai corect (această procedură are loc pentru fiecare din mulțimile de instanțe după următorul pseudocod) :

1. Algoritmul determină vecinătățile (bounding-boxurile) reprezentate sub forma de vectori folosind o raza de căutare notată cu reprezintă locația subiectului în frame-ul .
2. Clasifică vecinătatea reprezintă o variabilă binară ce indică prezența sau absența obiectului în imaginile corespunzătoare vecinătăților acestuia.
3. Actualizează locația “tracker-ului” folosind o strategie “greedy” = astfel nu menținem o distribuție asupra locațiilor din raza a subiectului și oferim șansă egală   “tracker-ului “ la timpul
4. Formează două mulțimi } și } ce reprezintă un update al modelului ce reprezintă fundalul.
5. Actualizează modelul cu un “bag” pozitiv și negative fiecare conținând câte o singură imagine din mulțimea

Utilizarea unei astfel de metode implică de asemenea și dezavantaje deoarece modelul poate deveni confuz după un număr de runde ceea ce reduce puterea de decizie a clasificatorului și totodata și timpul de execuție. Datorită faptului că această metodă încearcă să optimizeze mereu cel mai slab clasificator face ca identificarea posibilelor erori să se facă mult mai greu.

## Kcf Tracker

Kcf (Kernelized Correlation Filters) este o metodă ce utilizează tehnici similare cu cei

doi algoritmi prezentati mai sus plecând de la observația că eșantioanele pozitive folosite de MIL prezintă zone de suprapunere ce pot fi modelate sub forma unor matrici circulare .

Matricile circulare reprezintă un model de matrice în care fiecare diagonală descendentă de la stânga la dreapta este constantă și în care fiecare linie este rotită un element în dreapta față de linia anterioară.

Am notat cu diagonala matricii ce conține coeficienții transformării discrete Fourier al vectorului a, iar U este matricea unitară. Acestă modalitate de reprezentare ne permitem să utilizăm operații Fourier în locul celor clasice pentru a putea optimiza timpul de execuție. De asemenea cu ajutorul acestor matrici putem recurge la un model de regresie  “ridge regression” având drept scop găsirea unei funcții prin care dorim să minimizăm erorile pentru eșantioanele și regresiile lor

Am notat cu parametrul de regularizare ce controlează fenomenul de suprapunere. Aceasta apare ca o eroare de modelare atunci când o funcție este prea apropiată de un set limitat de puncte din setul de date. este numit minimizator și are următoarea formă unde reprezintă o matrice a căror linii sunt reprezentate de eșantioane, iar fiecare element este o țintă de regresie , iar este matricea identitate.

Modul de calcul pentru în forma de mai sus este mult prea costisitor având o complexitate exponențială ) , așadar pentru a optimiza soluția putem rescrie folosind domeniul Fourier drept , iar este conjugata lui . Pentru numere reale cele 2 ecuații sunt similare.

În contextul tracking-ului funcția de regresie este antrenată cu toate shiftările eșantionului de input . Acest lucru se face deoarece dorim să obținem robustețe chiar si în cazul în care obiectul urmărit este supus traducerii. Pentru un vector a de dimensiune m, matricea circulară este o matrice formată din toate shiftările lui a. Aceasta are proprietatea că poate reprezenta convoluția . Convoluția reprezintă o operație matematică peste două funcții prin care se produce o a treia funcție ce indică modul cum forma unei funcții este modificată de cealaltă.

Având în vedere proprietățile menționate putem rescrie formula factorului d minimizare sub forma :

Pentru a obține o mai bună discriminare este necesar să evaluăm cele de mai sus folosind caracteristici dimensionale superioare. În ecuația de regresie sunt necesare numai rezultatele multiplicării acestor vectori . Folosind formula de calcul de mai sus reducem timpul de execuție doar la a calcula transformarea discretă Fourier ce are o complexitatea . Așa cum este prezentat în Figura 3 modul de funcționare a KCF poate fi împărțit în cinci blocuri cu rolul de detecție, antrenare creare și actualizare a modelului.

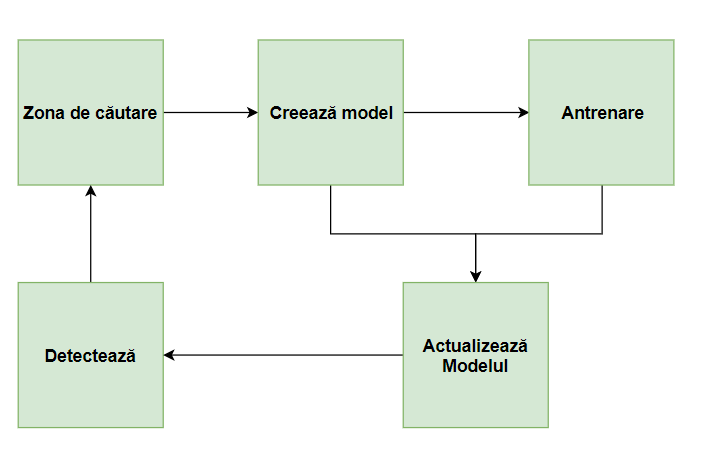


Figura 3

## MedianFlow Tracker

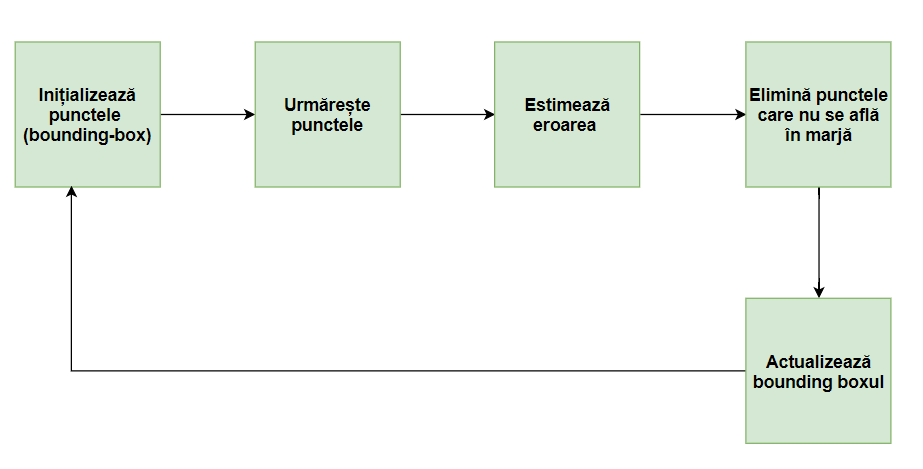
Această metodă de are la bază așa numita “forward-backward consistency” ce face presupunerea că urmărirea corectă trebuie să fie independenta de fluxul de timp. Putem sintetiza această modalitate sub forma unui algoritm ce presupune următorii pași:

1. Tracker-ul produce o traiectorie pentru subiect : considerăm o secvență de imagini și locația subiectului în frame-ul t. Acest subiect este urmărit  înainte pe parcursul celor k frame-uri ceea ce produce traiectoria definită prin am notat cu f   modul de urmărire “forward”.
2. În ultimul frame din această traiectorie se produce o validare ce constă în urmărirea subiectului in sens invers (de la ultimul frame către primul). Scopul nostru este de a estima eroarea acestei traiectorii în secvența de imagini . Pentru a putea deci estima această eroare construim traiectoria de validare, punctul este urmărit înapoi până la primul frame : unde .
3. Următorul pas presupune compararea traiectoriilor, dacă acestea diferă mult prea mult se consideră ca fiind incorectă. Eroarea deci se poate defini prin distanța dintre cele două traiectorii . Putem defini mai multe moduri de calcul pentru această distanță de exemplu cea Euclidiană

Așa cum am menționat mai sus ipoteza FB (“forward-backward”) se bazează pe faptul că aceste puncte sunt independente. În realitate acest caz se întâmplă foarte rar, totuși punctele alcătuiesc blocuri ce se mișcă împreună. La rândul lor blocurile pot fi considerate obiecte (mașini , oameni) . Urmăritorul primește ca date de intrare o pereche de imagini și un bounding-box (ce reprezintă de fapt un bloc) și produce bounding-boxul .

Pentru a putea urmări blocul se folosește metoda Lucas-Kenade ce va genera fluxul mișcării între cele două perechi de imagini . Metoda Lucas-Kenade este o metodă de estimare a debitului optic. Aceasta presupune că debitul este în esență constant într-o vecinătate locală a pixelului examinat și rezolvă ecuațiile de bază ale fluxului optic pentru toți pixeli vecini folosind metoda celor mai mici pătrate. Un aspect ce merită menționat este legat de faptul că această metodă pleacă de la presupunerea că diferențele de luminozitate între cadrele consecutive sunt foarte mici.

Calitatea predicției punctelor este ulterior estimată, urmând ca fiecare din puncte să primească o eroare. Următorul pas presupune eliminarea a jumătate din cele mai slabe puncte, iar cealaltă jumătate este folosită pentru prezicerea întregului “box”. Estimarea deplasării cutiei delimitate de punctele rămase se efectuează folosind valoarea mediană peste fiecare dimensiune spațială. Această dimensiune se calculează pentru fiecare pereche de puncte calculând distanța dintre punctul curent si cel anterior . Părțile obiectului care nu care nu se găsesc în această scală nu sunt luate în considerare în procesul de votare.



## CSRT Tracker

# Concluzii

# Bibliografie

* + - Author1, *Book1*, 2018
    - Author2, *Boook2*, 2017