

# Extragerea Trasaturilor SIFT

---

Studenti: Alexandru PANA, 30232

Adrian SOUCUP, 30232

Indrumator: Asist. Dr. Anca CIURTE

5 Iunie 2013

## 1. Introducere

Problema pe care ne dorim să o rezolvăm este cautarea eficientă și eficientă a unei imagini într-o altă imagine. Un algoritm naiv de cautare pixel cu pixel nu este nici eficient (aria de cautare este foarte mare) și nici eficient (cea mai mică schimbare în rotație sau dimensiune va afecta negativ cautarea). Ne dorim să folosim o altă reprezentare a datelor de intrare, astfel încât eficiența și eficacitatea algoritmului să crească. Această reprezentare redusă va fi un set de trasaturi descriptive ale imaginii cautate. Procesul de transformare a unui set de date într-un set de trasaturi se numește "extragere de trasaturi". Pentru a obține trasaturi cât mai descriptive, trebuie luate în considerare mai multe aspecte.

Implementarea prezentată în acest proiect își propune să construiască un set de trasaturi SIFT pentru o imagine oarecare. Trasaturile SIFT au proprietatea de a rămâne consistente în urma modificărilor de rotație și redimensionare. Trasaturile rezultate vor fi definite de poziția acestora în imaginea originală, împreună cu un vector de orientare asociat.

## 2. Background-ul teoretic

Algoritmul de extragere a trasaturilor SIFT a fost descris de către David G. Lowe în lucrarea sa "*Distinctive Image Features from Scale-Invariant Keypoints*" publicată în 5 Ianuarie 2004 în "*International Journal of Computer Vision*". Lucrarea descrie extragerea unor trasaturi invariante la rotație și redimensionarea imaginii, și parțial invariante la schimbare în iluminare și perspectivă.

Algoritmul urmărește următorii pași:

1. **Detectia de puncte extreme in scale-space:** Acest pas caută puncte de minim sau maxim local în toate dimensiunile spațiului scale-space, generat eficient folosind diferența-de-Gaussian. Aceste puncte extreme sunt considerate puncte de interes invariante la redimensionare sau rotație.
2. **Localizarea punctelor cheie:** Punctele cheie sunt alese pe baza stabilității punctelor extreme. Se caută, de exemplu, filtrarea punctelor care se află pe muchii.
3. **Asignarea Orientațiilor:** Una sau mai multe orientări sunt alocate fiecărui punct, în funcție de orientarea gradientilor locali ai imaginii. Toate operațiile viitoare vor fi aplicate pe data de imagine transformată relativ la orientare, locația și dimensiunea fiecărei trasaturi, oferind invarianța la aceste transformări.
4. **Asignarea Descriptorilor:** Gradientii locali ai imaginii sunt măsurați la dimensiunea specificată de fiecare punct cheie. Aceștia sunt transformați într-o reprezentare parțial invariantă la modificări în iluminare.

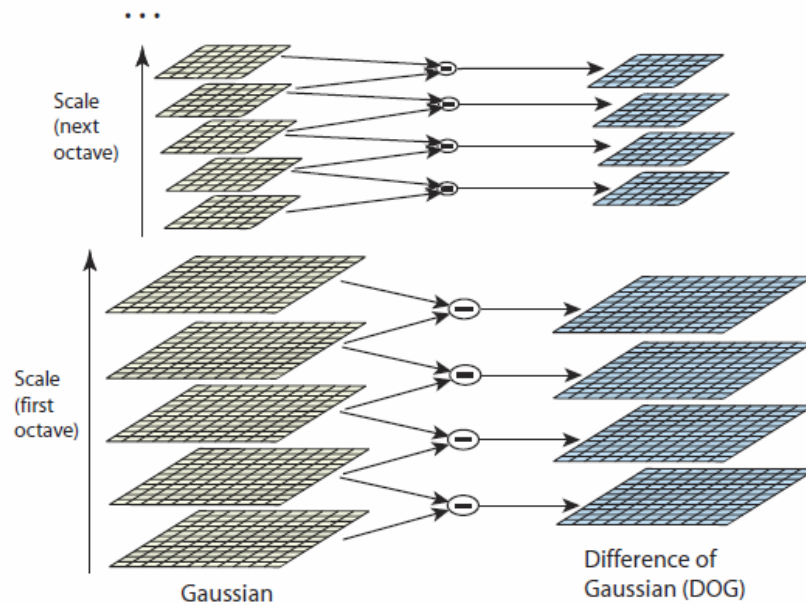
### 3. Implementare si design

#### 1. Descrierea functionalitatii

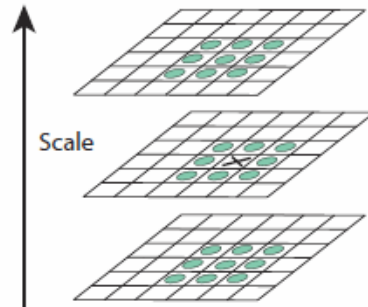
Algoritmul este implementat in C++ folosind libraria OpenCV ( Open Computer Vision ). Fiecare pas descris in lucrarea lui David Lowe este implementat folosind o clasa templatizata pe dimensiunea spatiului invariant la redimensionare. Templatizarea claselor se face pe dimensiunea spatiului scale-space, pentru a aloca matricea de imagini static, pe stiva. Pentru a permite o cuplare mai stransa a componentelor ( fiecare componenta depinde strans de rezultatul celei anterioare ), doar prima componenta este templatizata pe dimensiunea spatiului. Urmatoarele sunt templatizate pe tipul componentei anterioare, fapt care face posibila calcularea dimensiunii fiecarei matrici in functie de dimensiunea originala.

**ScaleSpace** Construieste un o matrice de imagini pornind de la imaginea originala. Fiecare coloana din aceasta matrice este formata redimensionand imaginea originala cu un factor de 0.5 ( octaves ). Liniile ( scales ) sunt obtinute aplicand un filtru gaussian cu o deviatie standard de *sigma* fata de imaginea de pe linia anterioara. Pe imaginile de pe prima linie nu se aplica filtrul gaussian. Prima coloana este construita folosind un factor de redimensionare de 2.

**DifferenceOfGaussians** Aceasta clasa porneste de la matricea generata de clasa *ScaleSpace* si construieste o matrice de imagini scazand din fiecare imagine imaginea de pe linia urmatoare. In acest fel se obtin imagini apropiate de rezultatul aplicarii unui filtru Laplacian.



**MinimaAndMaxima** Aceasta clasa porneste de la matricea generata de clasa *DifferenceOfGaussians* si cauta punctele de minim si maxim intre vecinii 8 ai fiecarui punct atat pe scala curenta, cat si pe scalele precedenta si urmatoare.



**Orientations** Aceasta clasa genereaza pentru fiecare punct de maxim gasit anterior o lista de dimensiuni ( magnitudes ) si directii. Rezultatul este o lista de Keypoints care contin impreuna cu dimensiunile si directiile, pozitia punctului cheie in imaginea originala.

**DescriptorAssignment** Aceasta clasa genereaza pentru fiecare keypoint un descriptor format dintr-un vector ce contine valorile tuturor orientatilor calculate la pasul precedent.

O clasa aditionala pentru cronometrarea algoritmului este folosita la fiecare pas pentru a calcula durata de executie. Tipul total de executie este afisat la sfarsitul programului.

## 2. Mod de utilizare

Programul este numit 'sift.exe' si se porneste din linia de comanda:

```
sift.exe <imagine_intrare> <folder_output>
```

Primul parametru este imaginea din care vrem sa extragem trasaturile SIFT. Al doilea parametru este folderul in care trasaturile si imaginile pariale vor fi salvate.

## 4. Rezultate experimentale

Programul functioneaza cu orice imagine de intrare. Aceasta poate fi de orice dimensiune, si (aproape) orice tip. Procesul de extragere a trasaturilor SIFT va folosi In folderul specificat la rularea programului se vor salva toate imaginile intermediare. In acest fel se pot verifica imaginile din scale-space, difference-of-Gaussian sau extremele. Rezultatele sunt interpretabile, validarea corectitudinii acestor se face manual. De asemenea, rezultatele sunt foarte sensibile la coeficientii sigma alesi in pasul initial de construire a spatiului scale-space.

## 5. Concluzii

Extragerea trasaturilor este doar primul pas in procesul de cautare al unei imagini in alta imagine. Cautarea este implementata separat, independent de modul in care sunt extrase trasaturile. Cel mai bun mod de a demonstra corectitudinea algoritmului de extragere a trasaturilor dar si proprietatile trasaturilor este de a folosi trasaturile realizate intr-un algoritm de cautare. Din pacate algoritmul de cautare nu face parte din scopul acestui proiect, iar trasaturile nu pot fi testate practic.

## Bibliografie

1. Lowe, G. David. Distinctive Image Features from Scale-Invariant Keypoints
2. <http://www.aishack.in/2010/05/sift-scale-invariant-feature-transform/>
3. [http://en.wikipedia.org/wiki/Feature\\_extraction](http://en.wikipedia.org/wiki/Feature_extraction)