

SIFT (scale-invariant feature transform)

SIFT je algoritmus z oblasti počítačového videnia, určený pre detekciu a popísanie základných rysov obrázku. Bol patentovaný Univerzitou Britskej Kolumbie v Kanade a v roku 1999 ho publikoval David Lowe. Tento algoritmus pozostáva z troch hlavných častí a to:

- 1) Detekcia kľúčových bodov (*ang. keypoints detection*).
- 2) Vytvorenie deskriptorov pre tieto kľúčové body (*ang. keypoint descriptor*)
- 3) Spájanie základných rysov obrázku (*ang. feature matching*)

Keypoint Detection

Prvým krokom tohto algoritmu je získanie kľúčových bodov. Obrázok najskôr rozmazáme, čo nám odstráni zbytočný šum z obrázka a tým vyniknú dôležité časti. Pôvodný obrázok zmenšíme a toto rozmazávanie opakujeme. Rozmazávaním a zmenou veľkosti obrázku získame tzv. scale space. Tieto obrázky budeme medzi sebou následne odčítavať, čím získame Difference of Gaussian z ktorých určíme kľúčové body. Tieto body nájdeme s pomocou lokálneho maxima a minima, ktoré získame porovnávaním každého pixelu s okolitými pixelmi a s pixelmi s predchádzajúcej a nasledujúcej októvy – dokopy ho porovnáme s 26 pixelmi. Následne vyberieme stabilné kľúčové body.

Keypoint Descriptor

Druhá časť algoritmu sa venuje vytvoreniu deskriptorov. Pri vytváraní deskriptorov musíme najskôr vypočítať gradienty pre osi x a y. Tieto vypočítame tak, že pre os x odčítame hodnoty vpravo a vľavo od požadovaného

pixelu a absolútna hodnota tohto rozdielu je náš gradient pre os x. Podobný postup opakujeme aj pre os y. S týmito hodnotami dokážeme vypočítať magnitúdu a orientáciu každého pixelu v obrázku. Z týchto hodnôt vytvoríme histogram orientovaných gradientov (*ang. Histogram of Oriented Gradients*). Podľa jeho vrcholu vieme určiť orientáciu pixelu. Ak je vrcholov viac, vygenerujeme ďalší kľúčový bod. Deskriptor vytvoríme tak, že pre každý kľúčový bod vezmeme blok okolitých 16x16 pixelov. Tento blok rozdelíme na 4x4 podčasti a pre každý vytvoríme HoG. S týmito hodnotami dokážeme deskriptor vizualizovať.

Feature matching

Spájanie kľúčových bodov je posledná časť algoritmu SIFT. Pre dva rôzne obrázky sú vytvorené kľúčové body a ich unikátne deskriptory. Podľa týchto údajov vieme spojiť kľúčové body a toto spojenie zobrazíť. Býva konvenciou obmedziť počet zobrazených spojení - kvôli prehľadnosti.

Aplikácie SIFT algoritmu

- rozpoznávanie objektov,
- rozpoznávanie osôb,
- spájanie obrázkov (napr. pri vytváraní panoramatickej fotografie),
- rozpoznávanie gest a podobne.

Súvisiace algoritmy

- SURF je jednoduchšia a rýchlejšia verzia algoritmu SIFT.
- Harris Corner Detector slúži na extrakciu hrán a základných rysov obrázku.
- Fast je algoritmus na detekciu hrán a základných rysov obrázku a bol vyvinutý hlavne pre použitie v reálnom čase.

Pseudokód:

Algorithm 1: Main of function

input: original = Image from computer (64x128 px)

Result: Visualized descriptors

Main ():

- Bitmap* = MakeImageGrayscale(*original*);
- List* < *Bitmap* > *cells* = GetCells(*Bitmap*);
- foreach** *cell* in *cells* **do**
- List* < *double*[] > *hists* = GetHistForEveryCell(*cell*);
 - VisualizeDescriptor(*hists*);

Algorithm 2: Make image gray-scale

Result: Gray-scale image

MakeImageGrayscale (*Bitmap*)

- create gray-scale color matrix;
- convert image to grayscale using this matrix;

Algorithm 3: Divide image into separate cells (8x16)

Result: ListOfBitmaps

GetCells (*Bitmap*)

- Divide bitmap into cells (each cell 8x8 px);
- Add this cells to list of bitmaps;

Algorithm 4: Create histogram for every cell

Result: double[36]

GetHistForEveryCells (*Bitmap*)

- Create histogram based on magnitude and orientation of every pixel in given Bitmap;
- $Magnitude = \sqrt{Gx^2 + Gy^2}$;
- $Orientation = \arctan(Gy/Gx)$;

Algorithm 5: Visualize descriptors

input: double[36]

Result: Bitmap

VisualizeDescriptor (*double*[36])

- Create sets of degrees, adds them to list of 8 sets. For every set there is created direction arrow.
