

VEŽBA 11

Segmentacija slike upotrebom mašinskog učenja

Potrebno predznanje

- Poznavanje programskog jezika C
- 2D signali
- RGB i YUV prostori boja
- Filtriranje 2D signala u vremenskom domenu
- Računanje histograma slike

ZADATAK

1.1 Zadatak 1

Završiti implementaciju funkcije

- `static vector<vector<int>> kMeans(vector<KMeansPoint> points, int nFeatures, int K);`

Funkcija vrši grupisanje tačaka u N-dimenzionom prostoru koristeći *K-means* algoritam.

Algoritam se sastoji iz sledećih koraka:

- Za zadati broj grupa K inicijalizovati K centralnih tačaka u N-dimenzionom prostoru.
- Svaku tačku pridružiti grupi (klasteru) čija centralna tačka je najbliža zadatoj tački.
- Pronaći novu centralnu tačku za svaki klaster tako što će se izračunati aritmetička sredina svih tačaka u klasteru
- Ponavljati process dokle god barem jedna centralna tačka menja svoju poziciju.

Parametri funkcije su:

- `points` – vektor ulaznih tačaka
- `nFeatures` – broj obeležja kojima je opisana tačka
- `K` – broj grupa (klastera)

1.2 Zadatak 2

Realizovati funkciju

- `void IntensityPlusPositionBasedKMeans(uchar input[], int xSize, int ySize, int K);`

Koja vrši grupisanje piksela u slici na osnovu nivoa osvetljaja i položaja.

Parametri funkcije su:

- `input` – Ulazna slika u RGB formatu
- `xSize` – širina slike
- `ySize` – visina slike
- `K` –Zadati broj grupa (klastera)

Unutar funkcije potrebno je izvršiti konverziju slike iz RGB u YUV prostor boja. Potom je potrebno napraviti vektor tačaka (tipa `KMeansPoint`) koji za svaku tačku sadrži opis u formi [osvetljaj, horizontalni položaj, verzikalni položaj]. Sva tri obeležja je potrebno skalirati tako da se vrednosti nalaze u istom opsegu. Potom pozvati funkciju `kMeans` kako bi se izvršila segmentacija. Svaki dobijeni segment obojiti različitom bojom.

1.3 Zadatak 3

Realizovati funkciju

- `void ColorPlusPositionBasedKMeans(uchar input[], int xSize, int ySize, int K);`

Koja vrši grupisanje piksela u slici na osnovu boje i položaja.

Parametri funkcije su:

- `input` – Ulazna slika u RGB formatu
- `xSize` – širina slike
- `ySize` – visina slike
- `K` –Zadati broj grupa (klastera)

Unutar funkcije je potrebno napraviti vector tačaka (tipa `KMeansPoint`) koji za svaku tačku sadrži opis u formi [nivo crvene, nivo zelene, nivo plave, horizontalni položaj, vertikalni položaj]. Svih pet obeležja je potrebno skalirati tako da se vrednosti nalaze u istom opsegu. Potom pozvati funkciju `kMeans` kako bi se izvršila segmentacija. Svaki dobijeni segment obojiti različitom bojom.

1.4 Zadatak 4

Realizovati funkciju

- `void SIFTBasedKMeans(uchar input[], int xSize, int ySize, int K);`

Koja vrši računanje SIFT obeležja, i potom grupisanje obeležja koristeći `kMeans` algoritam.

Parametri funkcije su:

- `input` – Ulazna slika u RGB formatu
- `xSize` – širina slike
- `ySize` – visina slike
- `K` –Zadati broj grupa (klastera)

Izračunati SIFT obeležja u slici na isti način kao u prethodnoj vežbi. Pre same obrade neophodno je izvršiti konverziju iz RGB u YUV prostor boja. Pretraga obeležja se vrši na osnovu Y komponente slike. Za računanje SIFT transformacije iskoristiti funkciju:

- `SiftKeypointList calculateSIFT(const uchar Y_buff[], int width, int height);`

Povratna vrednost funkcije je lista ključnih tačaka predstavljenih u formi strukture *SiftKeypoint*.

Nakon izračunavanja potrebno je na osnovu detektovanih tačaka napraviti vektor tačaka u formi `KMeansPoint`. Za deskriptor obeležja koristiti SIFT deskriptor. Pozvati funkciju `kMeans`.

Na osnovu dobijenih klastera napraviti K novih lista tipa `SiftKeypointList`, gde svaka lista sadrži tačke koje pripadaju jednom klasteru. Svaki klaster obojiti različitom bojom koristeći funkciju:

- `void markSIFTKeypointsRGB(uchar imgRGB[], int width, int height, SiftKeypointList kpList, uchar R, uchar G, uchar B);`