

CAMshift algoritem za detekcijo in slednje objektov

Gaja Žumer

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenija
E-pošta: gzl671@student.uni-lj.si

Povzetek. abstract

Ključne besede: algoritmi sledenja

CAMshift tracking algorithm

abstract

1 UVOD

Meanshift je algoritem za lokalno iskanje maksimuma funkcije gostote oziroma iskanje modusa v vzorcu. Naj bo x začetni približek za maksimum in naj bo K jedro, ki uteži sosednje točke glede na njihovo razdaljo do x . Zaradi kasnejše poenostavitve je priporočljivo uporabljati Epanechnikovo jedro [5], razširjena pa je tudi uporaba Gaussovega jedra. Utežena srednja vrednost gostote na nekem območju N je enaka:

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) \cdot x_i}{\sum_{x_i \in N(x)} K(x_i - x)},$$

kjer je $N(x)$ okolica x . Razlik $m(x) - x$ pravimo *mean shift vektor* in kaže v smeri povečanja gostote. Mean shift je iterativna metoda iskanja maksimuma in jo ponavljamo toliko časa, dokler ni mean-shift vektor dovolj majhen. CAMshift (ang. *Continuously Adaptive Meansift*) je izboljšava Meanshift algoritma za potrebe sledenja. Z vsakim uspešno najdenim modusom namreč prilagodi območje sledenja, tako da se ta najbolj prilagodi novi situaciji. To nam pride prav predvsem v primerih, ko opazovani predmet med premikanjem spreminja obliko (se bodisi povečuje oziroma pomanjšuje bodisi rotira okoli svoje osi).

Algoritem uporablja Meanshift algoritem za iskanje lokalnih optimumov v funkcijah gostote

verjetnosti z fiksnim oknom (glej opis v seminarju 9.). Gostote verjetnosti določimo napodlagi frekvenčne analize pojavnosti značilnic, tipično so to barve ali teksture objektov zanimanja. Izhodiščno literaturo najdete v knjigi [3], poglavje 5.2. Pri-lagoditev Meanshift algoritma za namen sledenja objektov v videu je v adaptivnivelikosti okna

2 MEANSHIFT ALGORITEM

MeanShift je iterativna metoda za iskanje gosteje poseljenih območij. Kot je bilo namignjeno že v prejšnjem poglavju, je MeanShift algoritem identičnem gradientni metodi na oceni gostote z jedrom (*kernel density estimation (KDE)*). V resničnem svetu imamo namreč po navadi končno število vzorcev $\{x_i\}_{i \in [0, \dots, N]}$, ki so že sami po sebi ocena funkcije gostote. Ker pa so nam matematično bolj privlačne zvezne funkcije, si pri oceni funkcije gostote pomagamo z jedri na način:

$$p(x) = \frac{1}{N} \sum_{i=1}^N K(x - x_i),$$

pri čemer je N število vzorcev in K jedro. Jedro mora biti:

- normalizirano: $\int_{R^d} K(x) dx = 1$
- simetrično: $\int_{R^d} x K(x) dx = 0$
- homoskedastično: $\int_{R^d} x x^T K(x) dx = cI$

Poznamo različne vrste jeder, omenila pa bi tri predstavnike, ki so pomembni na področju sledenja.

- Epanechnikovo jedro: $K_E(x) = \{c(1 - \|x\|^2), \|x\| \leq 1$
0, sicer

- uniformno jedro: $K_U(x) = \{c, \|x\| \leq 1, 0, \text{ sicer}\}$
- Gaussovo jedro: $K_G(x) = c \cdot \exp -\frac{1}{2}\|x\|^2$

V nadaljevanju bomo pisali $K(x - x_i) = c \cdot k(\|\frac{1}{h}(x - x_i)\|^2)$. Sledi enačba funkcije gostote:

$$p(x) = c \sum_{i=1}^N w_i k(\|\frac{1}{h}(x - x_i)\|^2),$$

$$\sum_{i=1}^N w_i = 1$$

Če sedaj izračunamo odvod gostote funkcije $\delta p(x) = \frac{\partial}{\partial x} p(x)$ in pri tem upoštevamo $g(x) = -k'(x)$ dobimo

$$\delta p(x) = \frac{2c}{h^2} (\sum_{i=1}^N w_i x_i g(\|\frac{x - x_i}{h}\|) - x \sum_{i=1}^N w_i g(\|x\|))$$

. Ker iščemo maksimum funkcije gostote, velja $\frac{\partial}{\partial x} p(x) = 0$. To nam da enačbo ...

Koraki MeanShifta: prva slička videa:

- inicializiraj željeno območje sledenja (ROI)
- vsakemu pikslu v ROI določi določi mesto v histogramu q glede na njegovo vrednost.
- iz jedra določi utež za izbrani piksel in jo dodaj k vrednosti histograma
- normaliziraj histogram
- trenutna ocena pozicije objekta je vezana še na prejšnjo sličico. Izrežemo obočje in tudi na njem (po enakem postopku kot prej) izračunamo histogram p
- izračunaj uteži po enačbi $v = \sqrt{(\frac{q_u}{p_u + \epsilon_{ps}})}$, pri čemer je ϵ_{ps} zelo majhna vrednost
- izračunajmo sliko iz histograma v: za vsak piksel pogledamo na katero mesto v histogramu bi spadal in mu pripišemo vrednost histograma v tisti točki. S te mdobimo sliko razdeljeno na segmente
- pomnožimo dobljeno sliko z jedrom $g(x)$. Ta korak lahko preskočimo, če smo si na začetku pametno izbrali jedro $k(x)$
- po že dobro znani formuli izračunajte $x^{(k+1)}$
- postopek ponavljajte toliko časa dokler ne bo mean-shift vektor manjši od ena ali pa dokler ne boste presegli vnaprej določeno število iteracij
- dokler sličic ne zmanjka ponavljaj:
- izračunaj histogram p z uporabo Epanechnikovskega jedra
- za vsak piksel v omejenem območju izračunaj utež

- izračunaj novo pozicijo $x^{(k+1)}$
- postopek ponavljaj do ustavitvenih pogojev

3 CAMSHIFT ALGORITEM

4 REZULTATI

Implementacija Meanshift algoritma je spisana v programskem jeziku Python brez uporabe naprednejših funkcij knjižnic *numpy* in *opencv* kot so na primer *calcHist*, *calcBackProject*, *meanShift* ipd.

5 ZAKLJUČEK

CAMshift je primeren kot osnova za sledenje objektom, saj bi ga za boljše delovanje morali združiti s kakšnim prilagodlivejšim algoritmom (npr. Kalmanov filter ([6]) ali particle filter ([7])). Poleg tega ima tudi pomanjkljivosti pri sledenju votlih objektov, kar bi za silo lahko popravili s primernim jedrom. CAMshift odpove tudi v primerum, da objekt zapusti vidno polje in se vanj ne vrne več oziroma se vrne, vendar ne po isti poti kot je odšel.

LITERATURA

- [1] R. Klette, *Concise computer vision - An introduction into theory and algorithms*, Springer-Verlag London, 2014.
- [2] A. Varfolomeiev, O. Antonyuk, O. Lysenko, *Camshift object tracking algorithm implementation on DM6437 EVM*, Preceedings of the 4th European DSP in Education and Research Conference.
- [3] H. Kaeslin, *Digital integrated circuit design: from VLSI architectures to CMOS fabrication*, Cambridge University Press, 2008.
- [4] <https://fr.wikipedia.org/wiki/Camshift>, maj 2019.
- [5] M. Kristan, *Advanced CV methods, Mean Shift tracking*, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani
- [6] G. W. Kim, D. S. Kang *Improved CAMshift algorithm based on kalman filter*, Advanced Science and Technology Letters, 98, pp.135-137, 2015.
- [7] Z. Wang, X. Yang, Y. Xu, S. Yu, *CamShift guided particle filter for visual tracking*,