

IZVEDBA ALGORITMA RAČUNALNIŠKEGA VIDA NA OMREŽNI KAMERI

Vildana Sulić, Janez Perš, Matej Kristan, Andreja Jarc, Matej Perše,
Stanislav Kovačič

Laboratorij za strojni vid
Fakulteta za elektrotehniko
Univerza v Ljubljani
E-pošta: vildana.sulic@fe.uni-lj.si
URL: <http://vision.fe.uni-lj.si>

POVZETEK: Članek opisuje motivacijo, izzive in nekatere rešitve, ki pridejo v poštev pri izvedbi algoritma računalniškega vida na naprednejših omrežnih kamerah. Kot primer je opisana realizacija algoritma za sledenje objektov na cenovno ugodni kameri Axis 207, z uporabo prosto dostopnih in odprtokodnih rešitev.

1. UVOD

V zadnjih letih so se na tržišču pojavile videonadzorne kamere, ki delujejo kot omrežne kamere, njihova cena pa je že dosegla ceno, ki omogoča množično uporabo. Na njih vedno pogosteje srečamo programsko opremo, ki je bila prirejena in prenešena iz osebnih računalnikov. Tako lahko na kamerah srečamo operacijski sistem Linux, specializirana programska oprema, s katero so realizirane različne funkcije kamere pa pogosto temelji na odprtokodnih rešitvah.

Ker so medtem tudi vgrajeni mikroprocesorji in mikrokrmilniki napredovali, tako cenovno kot glede zmogljivosti, ponuja ta kombinacija odlične možnosti, da funkcionalnost teh dokaj poceni kamer razširimo z metodami obdelave slik in metodami računalniškega vida. Na ta način upamo, da bi lahko nekatere algoritme računalniškega vida vpeljali v široko uporabo brez stroškov, ovir, in nenazadnje odpora uporabnikov, ki so povezani z vpeljavo osebnih računalnikov v takšno okolje.

2. MOTIVACIJA

Nedvomno je najbolj očitna prednost izvedbe algoritmov neposredno na kameri to, da odpade strošek samostojnega osebnega računalnika, ki bi moral sicer igrati vlogo procesne enote. Omrežne kamere so cenovno vsaj v rangu osebnih računalnikov, možno pa je dobiti tudi precej cenejše modele. S tem je skupni strošek izvedbe bistveno manjši kot bi bil z uporabo samostojnega osebnega računalnika.

Druga pomembna prednost je gotovo dojemanje tehnologije s strani uporabnikov. V okolju kjer je potreba po videonadzoru, je samoumevno, da so prisotne nadzorne kamere, malo težje pa naročniki sistema sprejmejo predlog, da je treba mrežo kamer dopolniti še z mrežo zmogljivih osebnih računalnikov, namenjenih izključno obdelavi slike, še težje pa sprejmejo s tem povezane stroške, saj niso zajeti le v ceni kamer. Na nek način bi bil inteligentni videonadzorni sistem pravzaprav sistem strojnega vida, zato se splača držati nekaterih napotkov, ki so pri sistemih strojnega vida v industriji že skoraj samoumevni [1]. Nedvomno je koristno, če naročniki in uporabniki sistema računalniškega vida niti ne dojemajo kot računalnik, ampak kot del običajne naprave, ki so jo v tistem okolju vajeni.

Tretja prednost je v tem, da so dovolj zmogljive omrežne kamere že nameščene v veliko okoljih, kjer obstaja potreba po inteligentnejšem nadzoru, torej tudi po uporabi algoritmov računalniškega vida. S pristopom, ki ga opisujemo v tem članku, bi bilo možno kompletno omrežje nadzornih kamer nadgraditi v porazdeljen inteligenten videonadzorni sistem in vanj vpeljati metode računalniškega vida brez kakršnegakoli poseganja v strojno opremo, napeljavo ali sam koncept videonadzora.

Nenazadnje je takšna kamera preizkušena platforma za razvoj aplikacij. Proizvajalci poskrbijo, da kamere delujejo zanesljivo, imajo vse potrebne vmesnike in programsko podporo, zato je vključevanje dodatne funkcionalnosti lahko razmeroma enostavno, če proizvajalec tega nalašč ne onemogoča.

3. PRIMER: OMREŽNA KAMERA AXIS 207

Kamera je izdelek švedskega podjetja Axis Communications. Njena cena je med 250 in 350 €. Ima fiksno nameščen objektiv, z možnostjo ročne nastavitve ostrine. Ostali tehnični podatki, zanimivi za razvijalca programske opreme, so:

- RGB slikovni senzor VGA ločljivosti (640 x 480), obstajajo še različice z višjo ločljivostjo, maksimalna frekvenca zajemanja slik je 30 slik/s.
- Mrežna povezava 100 Mbit, obstajajo tudi različice z brezžičnim, WiFi vmesnikom.
- Arhitektura procesorja ARM9 (ARM9TDMI) z dodatnimi razširitvami za kompresijo videa v realnem času (ARTPEC - A) , 32 MB RAM in 8 MB flash pomnilnika – 2.1 MB flash pomnilnika je prostega za uporabniške aplikacije.
- Operacijski sistem Linux 2.6 z uClibc standardno C knjižnico primerno za vgradne arhitekture.
- RAM disk za shranjevanje začasnih datotek (8 MB nezasedenega prostora v tovarniški konfiguraciji).
- Delujoč HTTP strežnik (Boa 0.94).

- Možnost dostopa do omrežne kamere preko telnet protokola in vgrajenega FTP strežnika. Pri tovarniško nastavljeni kameri to možnost omogočimo preko spletnega vmesnika.

Proizvajalec kamere ponuja razvojno okolje (*gcc toolchain*) na osnovi prevajalnika gcc za ARM arhitekturo procesorja [2]. Prevajalnik in celotno razvojno okolje je na voljo v obliki izvorne kode in ga je možno prevesti na namiznem računalniku z operacijskim sistemom Linux. Proizvajalec ne ponuja dokumentacije, ki bi omogočala neposreden dostop do zajetih slik, vendar pa objavljen protokol za dostop do slike in videa v JPEG in MJPEG preko spletnega vmesnika razkriva tudi način kako do teh podatkov dostopati lokalno.



Slika 1: Kamera Axis 207 (dimenzije: 85 x 55 x 34 mm)

4. PREVAJANJE PROGRAMOV

Vse kamere tega proizvajalca podpirajo programski vmesnik VAPIX [3], ki skrbi za funkcionalno prikazovanje slik, kontrolo funkcij na kameri ter nastavljanje parametrov. Dostop do VAPIX vmesnika je možen preko vgrajenega HTTP strežnika .

Vsa orodja, ki jih rabimo za razvoj programov, je možno dobiti na proizvajalčevi spletni strani [4], konfiguriranje orodij pa je opisano v [2]. Tako na primer gcc prevajalnik uporabimo na naslednji način:

```
arm-linux-gcc -march=armv4 -mtune=arm9tdmi -Wall -Wshadow -O2 -static
```

Gre za navzkrižno prevajanje – program prevedemo na osebni računalnik in ga preko FTP protokola prenesemo na kamero.

Programi so s knjižnicami povezani statično, saj je večina operacijskega sistema in na sistemu dostopnih knjižnic shranjenih v flash pomnilniku. Shranjene so torej v tistem delu, ki ga uporabnik ne more enostavno zapisati in zaradi tega knjižnic na kameri ni možno enostavno nadgrajevati. (V nasprotju s tem so sistemske nastavitve - /etc direktorij - shranjene tako, da jih lahko enostavno popravljamo.)

4.1 Združljivost z obstoječimi programi

Na kameri smo poskušali prevesti nekaj programov in knjižnic, ki jih bomo potrebovali pri nadaljnjem razvoju. Uspešno smo prevedli:

- Knjižnico IJG JPEG
- Knjižnico zLib (kompresija podatkov)
- Orodje za risanje grafov GNUPlot
- Tekstovni urejevalnik Joe
- Orodje za prenos datotek Wget

Na podlagi tega pričakujemo, da je na enak način možno prevesti večino programske opreme, vsaj odprtokodne. Odprtokodne programe, ki imajo pravilno sestavljeno datoteko *configure* prevedemo tako, da prevajanje najprej konfiguriramo z naslednjim ukazom:

```
./configure --host=arm-linux CC=arm-linux-gcc CFLAGS='-march=armv4 -mtune=arm9tdmi -Wall -Wshadow -O2' LDFLAGS='-static'
```

Na ta način se večina programov prevede brez sprememb, včasih so potrebne minimalne spremembe (recimo način merjenja časa pri programu Wget).

5. PRIMER: SLEDENJE OBJEKTOV

Pri uporabi kamere pri videonadzoru je ena bolj zanimivih aplikacij sledenje objektov, oziroma ljudi. Na kameri smo realizirali enostaven algoritem na podlagi odštevanja slik razlike. Program je napisan v programskem jeziku C in preveden zgoraj opisan način, deluje pa takole:

1. Pokliče v kamero tovarniško vgrajen program *bitmap*. Ta program zajame svežo sliko iz slikovnega senzorja ter jo v BMP formatu izpiše na standardni izhod – ki ga preusmerimo v datoteko na RAM disku.

2. Prebere zajeto datoteko in datoteko s sliko ozadja, ki je ravno tako shranjena v RAM disku, ju odšteje, izvede filtriranje in upragsavljanje ter izračuna parametre področij na binarni (upragsavljeni) sliki.
3. Parametre področij izpiše na standardni izhod v obliki tabele.

Če programu podamo ustrezen parameter, bo namesto sledenja osvežil sliko ozadja z zadnjo zajeto sliko.

Tak program lahko zaženemo, če se prijavimo na kamero kot uporabnik sistema preko protokola telnet, kar pa za praktično uporabo ni primerno. V kamero vgrajen HTTP vmesnik podpira tudi izvajanje programov, ki se podrejajo specifikacijam CGI [5] (Common Gateway Interface). Tako lahko program z nekaj prilagoditvami zažene vsak uporabnik, ki ima dostop do spletnega vmesnika kamere. Izvajanje programa je možno popolnoma avtomatizirati, saj lahko do URL naslova, ki zažene program za sledenje objektov, dostopajo tudi zunanji programi (na primer centralni strežnik).

Primer komunikacije med zunanjim uporabnikom oziroma zunanjim programom in kamero:

URL, ki ga vnese uporabnik v spletni brskalnik:

```
http://192.168.90.127/mvg-cgi/tracker2.cgi?command=0&iter=5&thresh=100
```

Vidimo, da lahko programu podamo tudi nekaj parametrov. Poleg ukaza (sledenje ali osvežitev slike ozadja) še število iteracij glajenja slike razlike in prag za detekcijo objektov.

Uporabnik dobi vrnjen naslednji rezultat:

```
Ime slike: in.bmp
Ime ozadja: back.bmp
Ime rezultata: out.bmp
Število iteracij je: 5
Prag: 100
Podatki o sliki in.bmp: x_size = 640, y_size = 480, p_size = 0
Podatki o ozadju: x_size = 640, y_size = 480, p_size = 0

n_blobs 2

Najdeni blobi:

n      x_min    y_min    x_max    y_max    x_center  y_center
-----
0      177      472     191     478     182       475
1      197      403     382     478     281       452
```

Kamera potrebuje približno 2 sekundi, da odgovori na zahtevo uporabnika. Domnevamo, da je neoptimalen način dekodiranja in zajema slike največji faktor pri tako počasnem delovanju. Naši testi kažejo, da je mogoče JPEG sliko iz senzorja dekodirati s pomočjo IJG JPEG knjižnice v 0.1 sekunde, če nas zanima samo sivinska slika. Kamera sicer nima enote za računanje s plavajočo vejico, zato je velika prednost uporabe IJG JPEG knjižnice tudi možnost dekodiranja slike samo z uporabo celoštevilskih operacij.

6. ZAKLJUČEK

Ilustrirali smo, da je na sodobnih omrežnih kamerah možno realizirati tudi algoritme računalniškega vida, s čimer odpade potreba po dodatnem osebem računalniku, cena strojne opreme pa je v takem primeru lahko bistveno nižja. Seveda smo omejeni z relativno nizko zmogljivostjo procesorjev, vgrajenih v takšne kamere. Domnevamo, da bomo lahko s pazljivo izbiro algoritmov v prihodnje na kameri dodali funkcionalnost, ki bo uporabna v praktičnih aplikacijah, obenem pa bo delovala dovolj hitro tudi z uporabo kompleksnejših algoritmov. Možne aplikacije segajo vse od sledenja ljudi, inteligentnega videonadzora, pa vse do razpoznavanja značilnosti oseb na slikah (na primer razpoznavanje obrazov).

LITERATURA

1. B. G. Batchelor, P. F. Whelan (1994), Machine vision systems: Proverbs, principles, prejudices and priorities, *Proceedings of the SPIE – The international Society for Optical Engineering*, Vol. 2347 - Machine Vision Applications, Architectures and Systems Integration III, Boston (USA), str. 374 – 383.
2. B. Jaglin, B. Mattsson (2007), Implementing the PalCom Protocol in an Axis Network Camera, Master of Science Thesis, Lund Institute of Technology
<http://www.lucas.lth.se/axis-palcom/report.pdf>
3. http://www.axis.com/techsup/cam_servers/dev/cam_http_api_index.php
Axis Communications Network Video Developer Pages: VAPIX
4. http://www.axis.com/ftp/pub_soft/cam_srv/arm_tools/arm_tools.htm
ARM Compiler Toolchain
5. <http://www.cs.tut.fi/~jkorpela/forms/cgic.html>
Getting Started with CGI Programming in C