FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

PROJEKT IZ PREDMETA RASPOZNAVANJE UZORAKA

AK. GOD. 2015/16

Sustav za video nadzor u zatvorenom prostoru

Autori:

Silvestar BADAK

Arijana BRLEK

Tomislav GUDELJ

Petra MARČE

Nino UZELAC

Ante ŽUŽUL

20. siječnja 2016.

Sadržaj

1. Projektni zadatak

1.1. Opis projektnog zadatka

Uklanjanje pozadine (engl. background subtraction) je uobičajen problem računalnog vida. Koristi se onda kada je potrebno pohraniti ogromnu količinu podataka (u obliku slika, engl. frame) uz što veću uštedu memorije. Pozadinom smatramo sve statične objekte, odnosno objekte koje smatramo nebitnim podacima za određen problem koji razmatramo. One objekte koje smatramo pozadinom bojimo bijelom bojom, a one koji su dinamični bojimo crnom bojom. Sliku koja se sastoji od piksela, od kojih je svaki predstavljen RGB (engl. red, green and blue color system) komponentama, pojednostavljujemo tako da joj pridjeljujemo samo dvije moguće boje te ćemo kodiranjem dobiti znatno veću uštedu memorijskog prostora. Analiziranje slike obavlja se na razini piksela. Svakom se pikselu dodjeljuje model čiji parametri pripadaju nekoj statističkoj distribuciji. U ovom projektu je razvijen efikasno prilagodljivi algoritam koristeći GMM model (engl. Gaussian Mixture Model)[7]. Cilj algoritma je detekcija uljeza, odnosno objekata koji nemaju uobičajeno ponašanje u sceni. U ovom slučaju to su dinamični objekti. Detekcija se obavlja tako da se uočavaju dijelovi scene koji ne odgovaraju modelu. Takav proces se naziva uklanjanje pozadine (engl. background subtraction). Statistički model sadrži gustoću razdiobe za svaki piksel. Idući korak je odrediti parametre tih razdioba[10]. U ovom projektnom zadatku model GMM koristi Gauss-ovu normalnu razdiobu s parametrima srednje vrijednosti (μ) i standardne devijacije (σ). Model koji za svaki piksel uzima po jednu razdiobu jednostavan je model koji ne daje dobre rezultate. GMM model u ovom projektu za svaki piksel računa parametre za više komponenti čiji se broj dinamički mijenja. Svaka komponenta prikazana je jednom Gaussovom normalnom distribucijom. Ovim možemo zaključiti da se broj parametara prilagođava za svaki piksel.

1.2. Pregled i opis srodnih rješenja

Svaki piksel prikazan je vektorom $\vec{x}$ koji sadrži komponente boja u nekom od odabranih sustava boja (npr. RGB sustav). Uklanjanje pozadine obavlja se tako da za svaki piksel provjeravamo pripada li pozadini (BG - background objects) ili je on dio nekog prednjeg objekta (FG - foreground object).
Bayseova odluka dana je formulom:

$$R = \frac{p(BG|\vec{x}^{(t)})}{p(FG|\vec{x}^{(t)})} = \frac{p(\vec{x}^{(t)}|BG)p(BG)}{p(\vec{x}^{(t)}|FG)p(FG)}$$

Uklanjanje pozadine koristi se pri praćenju objekata. Kod praćenja ne posjedujemo nikakve informacije o objektima niti koliko će se oni puta pojavljivati u sceni pa zbog toga pretpostavljamo da je

$$p(FG) = p(BG).$$

Praćenje objekata može poslužiti za unaprijeđenje uklanjanja pozadine[3, 9]. Pretpostavka je i uniformna distribucija za pojavljivanje prednjih objekata (FG - foreground object) $p(\vec{x}^{(t)}|FG) = c_{FG}$.
Uzimajući u obzir sve pretpostavke, je li objekt pripada pozadini ili ne modeliramo formulom:

$$p(\vec{x}^{(t)}|BG) > c_{trh}(= Rc_{FG}).$$

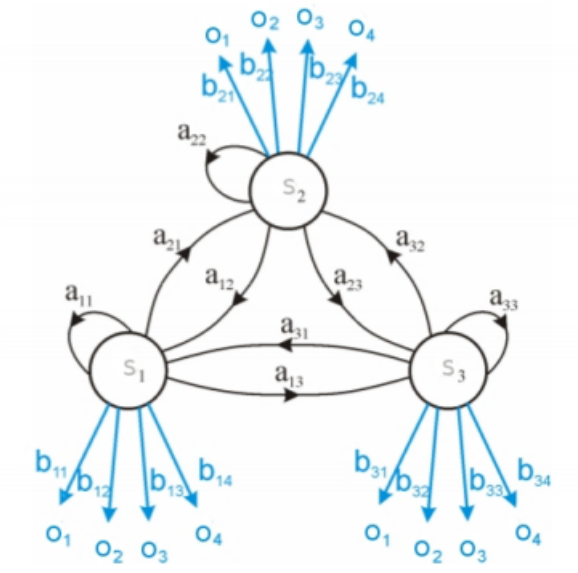
Pozadina (BG - background objects) se uči na skupu za treniranje koji se dinamički mijenja sa dolaskom novih slika (engl. frames). Pretpostavka je da su uzorci neovisni i glavni je problem kako efikasno procijeniti gustoću razdiobe te je prilagoditi na moguće izmjene. Jezgreno bazirane gustoće razdiobe korištene su u [2], a mi koristimo GMM iz [7]. Model koji koristi jezgreno bazirane gustoće razdiobe dan je sa formulom:

$$Pr(x_t) = \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{d}\frac{1}{\sqrt{2\pi\sigma_j^2}}e^{-\frac{1}{2}\frac{(x_{t_j}-x_{i_j})^2}{\sigma_j^2}}$$

gdje se koristi multivarijantna Gaussova razdioba $N(0, \sum)$ s parametrima μ_j i σ_j i gdje se pretpostavlja dijeljena dijagonalna kovarijacijska matrica. Postoje modeli koji uzimaju vremenski slijed slika kod kojih odluka ovisi o vrijednosti piksela iz prošle slike. Za primjer uzimamo [8] i [5] kod kojih je distribucija modelirana kao autoregresivni proces. Centralna ideja takvih modela je generirati mehanizam za predviđanje koji može odrediti aktualnu sliku (engl. frame) koristeći k zadnjih promatranih slika.
Matematička formulacija problema je:

$$I_{pred}(t) = f(I(t-1), I(t-2), ..., I(t-k))$$

gdje je f funkcija k-tog reda koja se treba izvesti. U [4] korišteni su skriveni Markovljevi modeli (engl. HMM - hidden Markov models) kod kojih se podrazumijeva da je modelirani sustav Markovljev proces sa skrivenim stanjima u kojima sa određenom vjerojatnošću emitira bilo koji element iz skupa simbola $T = \{b_1, b_2, ..., b_M\}$. Značajku koju je proces emitirao u trenutku t i stanju q_t označavamo sa O_t.
HMM se može smatrati najjednostavnijom Bayesovom mrežom.

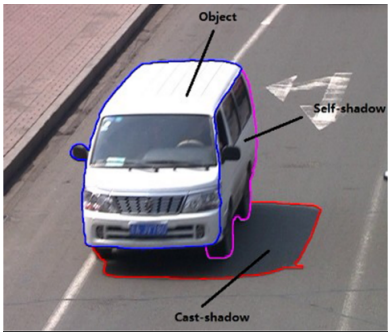


Slika 1.1: Skriveni Markovljev Model(HMM)

Spomenuti modeli su jako spori i teža je prilagodba na promjene u sceni. Mogu se koristiti kao poboljšanje algoritma korištenog u ovom radu.
Druga srodna tema je detekcija sjene objekata. Uz detekciju objekata obično detektiramo i sjenu što je pojašnjeno u [6]. Algoritmi za detektiranje sjene mogu biti formulirani tako da se baziraju na fizikalnim osnovama formulacije sjene. Model za detektiranje sjene dan je formulom:

$$s_k(x, y) = E_k(x, y)\rho_k(x, y)$$

gdje je s_k osvijetljenost točke u koordinatama (x,y) u vremenu k, $\rho_k(x, y)$ refleksija površine objekta, a $E_k(x, y)$ količina svjetlosne energije koju je upila površina S.



Slika 1.2: Primjer pokretne sjene sa prikazom dijela objekta koji nije osvijetljen (engl. slef-shadow) i onog dijela koji je projiciran na tlo (engl. cast-shadow)

1.3. Konceptualno rješenje zadatka

U ovom se radu baziramo na GMM modelu (engl. Gaussian Mixture Model)[7] koji smo koritili za uklanjanje pozadine (BG - background objects). Za svaki piksel smo izračunali GMM model s određenim brojem komponenata koji je parametar modela. Pripada li piksel pozadini ili ne donosimo odlukom $p(\vec{x}^{(t)}|BG) > c_{trh}(= Rc_{FG})$. Za svaki se model računaju parametri razdiobe za svaku komponentu na temelju skupa za učenje koji je za svaki novi period T različit. Razdioba koja se koristi u GMM modelu je Gaussova normalna razdioba.

U praksi se osjetljivost scene može mijenjati postepeno ili naglo, ovisno o uvjetima okoline. Da bi se prilagodili promjenama moramo osvježavati skup za učenje. Skup za učenje osvježavamo periodom T tako da izbacujemo "najstariji" primjer iz skupa te dodajemo "najnoviji". Tako u početku sve objekte smatramo prednjim objektima FG do nekog trenutka kad odaberemo dovoljan broj primjera za učenje da algoritam može razlučiti ono što je učestalo u sceni, a što nije. Prednji objekti koji se pojavljuju mogu biti statični neko vrijeme pa ih algoritam u nekom trenutku može svrstati u stražnje objekte, odnosno pozadinu. Svaki se piksel obrađuje te boji bijelom bojom ako ne pripada pozadini (sastavni je dio objekta koji se kreće) ili crnom bojom ako pripada pozadini (nije nam bitan jer je uobičajena pojava scene).

Slika 1.3: Primjer pokretnog objekta u sceni

2. Postupak rješavanja zadatka

2.1. Primjena GMM modela za odvajanje objekata od pozadine u slici

Intenzitet svjetla na slici općenito zavisi od geometrije scene, položaju kamere i same osvijetljenosti scene. Ako pretpostavimo da se u videonadzornom sustavu ne mjenja položaj kamera te se stalno snima ista scena, jedino što može utjecati na promjene inteziteta svjetla u slikama jest osvijetljenost scene. Osvijetljenost scene se u praksi može mijenajti postupno kroz vrijeme ili iznenadno. Postupne se promjene osvijetljenja češće događaju u vanjskim scenama zbog promjene doba dana ili vremenskih prilika. Nagle promjene osvjetljenja u untrašnjim scenama mogu biti rezultat paljenja i gašenja rasvjete u prostoriji. Da bi se model mogao prilagođavati promjenama osvijetljenja u sceni, potrebno je neprestano ažurirati skup za učenje novim informacijama o osvijetljenju, a odbacivati stare. S tim ciljem odabire se određeni vremenski period T, pa u trenutku t na raspolaganju imamo skup uzoraka X_T.

$$X_T = \{x^{(t)}, x^{(t-1)}, ..., x^{(t-T)}\}$$

Odvajanje objekata od pozadine na slici može se promatrati kao klasifikacijski problem u dvije klase od kojih jedna pripada pozadini, a druga objektima. Za svaki piksel slike dakle treba donjeti odluku pripada li pozadini ili nekom od objekata. U tom kontekstu X_T je naš skup za učenje na temelju kojeg se donose klasifikacijske odluke. Model koji se u okviru ovog rada koristi za klasifikaciju je model Gaussove mješavine.

2.1.1. Model Gaussove mješavine

Model mješane gustoće vjerojatnosti općenito modelira se kao linearna kombinacija M gustoća vjerojatnosti:

$$p(\vec{x}) = \sum_{m=1}^{M} \pi_m p(\vec{x} \mid \vec{\theta_m})$$

gdje je π_m težina komponente m, a θ_m predstavlja parametre konkretne razdiobe. Za težine komponenata vrijedi ograničenje:

$$\sum_{m=1}^{M} \pi_m = 1$$

Gaussova mješavima je parametarska funkcija gustoće vjerojatnosti sačinjena od težinskih suma Gaussovih komponenata:

$$p(\vec{x}) = \sum_{m=1}^{M} \pi_m \mathcal{N}(\vec{x} \mid \vec{\mu}_m, \Sigma_m)$$

gdje su $\vec{\mu}_m$ i $\mathbf{\Sigma}_m$ vektor očekivanja i kovarijancijska matrica m-te Gaussove razdiobe. Postoji više varijanti GMM-a u kojima kovarijancijske matrice mogu biti punog ranga ili dijagonalne, a parametri se mogu dijeliti ili vezati između Gaussovih komponenata. Izbor odgovarajuće varijante ovisi o količini dostupnih podataka za treniranje. U okviru ovog projekta koristi se diagonalna kovarijacijska matrica. Vrijedi:

$$\Sigma_m = \sigma_m^2 I$$

Parametri Gaussovih razdioba dobiveni su kao statističke procjene nad skupom uzoraka pa je ispravnije pisati:

$$\hat{p}(\vec{x} \mid X_t) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(\vec{x} \mid \hat{\vec{\mu}}_m, \hat{\sigma}_m^2 I)$$

Kad god na raspolaganju imamo novi primjer za učenje, on se ubacuje u skup za učenje na mjesto najstarijeg primjera. S obzirom da se skup za učenje promjenio potrebno je ponovo procijeniti sve parametre razdioba i vrijednost funkcije $p(\vec{x} \mid X_t)$. Osim toga, provode se i sljedeće rekurzivne jednadžbe:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m)$$

$$\hat{\vec{\mu}}_m \leftarrow \hat{\vec{\mu}}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\vec{\delta}_m$$

$$\hat{\sigma}_m^2 \leftarrow \hat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)(\vec{\delta}_m^T\vec{\delta}_m - \hat{\sigma}_m^2)$$

gdje je $\vec{\delta}_m = \vec{x}^{(t)} - \hat{\vec{\mu}}_m$ razlika između primjera i srednje vrijednosti razdiobe. Umjesto već spomenutog vremenskog intervala T ovdje je uvedena konstanta $\alpha = 1/T$ koja ograničava utjecaj starijih informacija. Za novi primjer pripadnost $o_m^{(t)}$ je jednaka jedinici za komponentu m koja je u blizini tog primjera te ima najveću važnost odnosno težinu $\hat{\pi}_m$. Za sve ostale komponente vrijedi $o_m^{(t)} = 0$. Definiramo da je primjer blizu

komponenti m ukoliko je njegova Mahalanobisova udaljenost od nje primjerice manja od $3\hat{\sigma}_m$. Kvadratna udaljenost primjera od komponente m računa se kao

$$D_m^2(\vec{x}^{(t)}) = \vec{\delta}_m^T \vec{\delta}_m / \hat{\sigma}_m^2$$

Ukoliko primjer nije blizu nijedne komponente od njih M generira se nova komponenta sa parametrima:

$$\hat{\pi}_{M+1} = \alpha$$

$$\hat{\vec{\mu}}_{M+1} = \vec{x}^{(t)}$$

$$\hat{\sigma}_{M+1} = \sigma_0$$

Ako je dostignut maksimalan broj komponenti odbacuje se komponenta s najmanjom težinom $\hat{\pi}_m$.

Ovaj algoritam predstavlja on-line algoritam grupiranja. Tipično će objeki koji ne pripadaju pozadini biti reprezentirani dodatnim grupama sa malim težinama $\hat{\pi}_m$ Uzimajući to u obzir možemo aproksimirati model pozadine sa B najvećih grupa:

$$\hat{p}(\vec{x} \mid X_t, BG) \approx \sum_{m=1}^{B} \hat{\pi}_m \mathcal{N}(\vec{x} \mid \hat{\vec{\mu}}_m, \hat{\sigma}_m^2 I)$$

Ako su komponente sortitane po padajućim vrijednostima težina $\hat{\pi}_m$ imamo:

$$B = \underset{b}{argmin}(\sum_{m=1}^{b} \hat{\pi}_m > (1 - c_f))$$

gdje je c_f mjera maksimalnog udjela piksela koji mogu pripadati prednjem dijelu slike bez utjecaja na model pozadine.

Dođe li novi objekt u scenu i ostane statičan neko vrijeme, vjerojatno će generirati novi stabilni klaster. Kako je stara pozadina sada zaklonjena tim objektom, težina njegovog klastera π_{B+1} će se konstantno povećavati. Ukoliko taj objekt ostane statičan dovoljno dugo, njegova će težina postati veća od c_f i smatrat će se dijelom pozadine.

2.1.2. Biranje broja komponenti

Težina π_m opisuje koliko podataka pripada m-toj komponenti GMM-a. To možemo promatrati kao vjerojatnost da uzorak dolazi iz m-te komponente, a u ovom slučaju π_m predstavlja osnovnu multinomijalnu distribuciju. Pretpostavimo da imamo skup koji sadrži t uzoraka od kojih svaki pripada jednoj komponenti GMM-a. Također pretpostavimo da je broj uzoraka koji pripadaju m-toj komponenti jednak:

$$n_m = \sum_{i=1}^{t} o_m^{(i)}$$

Za n_m-ove pretpostavljamo da se ponašaju prema kategoričkoj ("multinomijalnoj") distribuciji opisanu sljedećom funkcijom izglednosti:

$$\mathcal{L} = \prod_{m=1}^{M} \pi_m^{n_m}$$

Za izračun maksimalne izglednosti (engl. *Maximum Likelihood (ML)*) koristimo sljedeću formulu :

$$\frac{\partial}{\partial \hat{\pi}_m} \left(log\mathcal{L} + \lambda(\sum_{m=1}^{M} \hat{\pi}_m - 1) \right) = 0$$

Nakon rješavanja po λ dobijemo izraz:

$$\hat{\pi}_m^{(t)} = \frac{n_m}{t} = \frac{1}{t} \sum_{i=1}^{t} o_m^{(i)}$$

$\hat{\pi}_m^{(t)}$ označava procjenu iz t uzoraka. Izraz se može napisati rekurzivno kao funkcija procjene $\hat{\pi}_m^{(t-1)}$ za $t-1$ uzorak i pripadnosti $o_m^{(t)}$ posljednjeg uzorka:

$$\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + \frac{1}{t}(o_m^{(t)} - \hat{\pi}_m^{(t-1)})$$

Sada možemo vidjeti ako fiksiramo $\frac{1}{t}$ na $\alpha = \frac{1}{T}$ dobiti ćemo jednadžbu za popravljanje parametra $\hat{\pi}_m^{(t)}$ iz prethodne točke.

$$\begin{aligned} \hat{\pi}_m^{(t)} &= \hat{\pi}_m^{(t-1)} + \tfrac{1}{t}(o_m^{(t)} - \hat{\pi}_m^{(t-1)}) \\ &= \hat{\pi}_m^{(t-1)} + \tfrac{1}{T}(o_m^{(t)} - \hat{\pi}_m^{(t-1)}) \\ &= \hat{\pi}_m^{(t-1)} + \alpha(o_m^{(t)} - \hat{\pi}_m^{(t-1)}) \end{aligned} \tag{2.1}$$

Fiksiranje utjecaja novog uzorka znači da se više pouzdamo u novi uzorak, a doprinos starih uzoraka se eksponencijalno smanjuje.

Apriorno znanje za kategoričku distribuciju možemo prikazati koristeći konjugatni par, tj. Dirichlet-ov aprior:

$$\mathcal{P} = \prod_{m=1}^{M} \pi_m^{c_m}$$

Koeficijenti c_m predstavljaju apriorno znanje (u MAP(engl. *Maximum a posteriori*) smislu) za razred m, tj. broj broj uzoraka koji pripadaju tom razredu apriorno. Koristimo negativne koeficijente $c_m = -c$ po uzoru na [11]. Negativno apriorno znanje nam osigurava da razred prihvaćamo samo ako imamo dovoljeno dokaza iz podataka da taj razred zaista postoji. MAP izraz koji smo spomenuli uključuje apriorno znanje i definiran je sljedećim izrazom:

$$\frac{\partial}{\partial \hat{\pi}_m} \left(log\mathcal{L} + log\mathcal{P} + \lambda(\sum_{m=1}^{M} \hat{\pi}_m - 1) \right) = 0$$

$$\frac{\partial}{\partial \hat{\pi}_m}\left(log\mathcal{L} + log\prod_{m=1}^{M}\pi_m^{-c} + \lambda(\sum_{m=1}^{M}\hat{\pi}_m - 1)\right) = 0$$

Nakon izračuna dobijemo sljedeće:

$$\hat{\pi}_m^{(t)} = \frac{1}{K}\sum_{i=1}^{t}\left(o_m^{(i)} - c\right)$$

gdje nam je K definiran formulom:

$$K = \sum_{m=1}^{M}\left(\sum_{i=1}^{t} o_m^{(i)} - c\right) = t - Mc$$

Uvrstiviši ovaj izraz u prethodnu formulu dobijemo :

$$\hat{\pi}_m^{(t)} = \frac{\hat{\prod}_m - \frac{c}{t}}{1 - \frac{Mc}{t}}$$

gdje $\hat{\prod}_m = \frac{1}{t}\sum_{i=1}^{t} o_m^{(t)}$ predstavlja ML procjenu, a pristranost (engl. *bias*) iz apriora predstavljen je kao c/t. Pristranost se smanjue se povećenjem veličine skupa za učenje, tj. za velike vrijednosti t. U slučaju prihvatljivosti male vrijednosti pristranosti možemo je fiksirati tako da izraz c/t napišemo kao $c_T = c/T$ s nekim velikim T. Ovim smo pretpostavili da je pristranost uvijek jednaka i da iznosi kao kada imamo skup s T uzoraka. Sada možemo napisati još pojednostavljeniju formulu za korekciju težina $\hat{\pi}_m$ s postavljenim $c/t = c_T$:

$$\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + \frac{1}{t}\left(\frac{o_m^{(t)}}{1 - Mc_T} - \hat{\pi}_m^{(t-1)}\right) - \frac{1}{t}\frac{c_T}{1 - Mc_T}$$

Sada možemo ići još korak dalje. Pošto znamo da nam M predstavlja broj komponenti koji je najčešće mali pozitivan broj, te također c_T je najčešće mali decimalni broj. Što znači da je njihov umnožak također vrlo mali decimalni broj pa onda možemo uvesti još slijedeće pojednostavljenje: $1 - Mc_T \approx 1$. Čime možemo pojednostaviti dalje prethodnu formulu u :

$$\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + \frac{1}{t}\left(o_m^{(t)} - \hat{\pi}_m^{(t-1)}\right) - \frac{1}{t}c_T$$

Ako sada još uvedemo α-u koja je definirana kako je prije opsiano, dobivamo sljedeći izraz za korekciju težina:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) - \alpha c_T$$

Koristiti ćemo ovu jednadžbu za korekcije. Nakon svake korekcije moramo raditi normalizaciju tako da suma π_m-ova bude jednaka jedan. Zapičinjemo postupak tako da

jednu komponentu GMM-a centriramo na prvi uzorak i nakon toga nove komponente se dodaju kako je opisano u prethodnom podnaslovu. Dirichlet-ovo apriori s negativnim težinama prigušiti će komponente koje nisu sadržane u podatcima i komponentu m odbacujemo kada njena težina π_m poprimi negativnu vrijednost. Ovime osiguravamo da težine mješavina uvijek ostaju pozivine vrijednosti.

Treba još primjetiti da korištenje

$$\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + \frac{1}{t - Mc_T}(o_m^{(t)} - \hat{\pi}_m^{(t-1)})$$

nije previše korisno. Ako počnemo s velikom vrijednosti t tada izbjegavamo negativne popravke koji se događaju za malu vrijednost parametra t, ali također gubimo utjecaj i apriornog znanja. Ovo nam predstavlja opravdanje zbog kojeg smo fiksirali utjecaj apriornog znanja.

3. Ispitivanje rješenja

3.1. Ispitne Baze

U ovom projektu su za ispitne podatke korišteni videi. Izabrana su tri videa:

1. video sa puno pokreta u sceni (ubrzani snimak kretanja ljudi)
2. video sa umjerenim brojem pokreta u sceni (snimak prolaska usporenog auta)
3. video sa malim brojem pokreta u sceni(čovjek na benzinskoj stanici)

Zbog ograničenih mogućnosti računalnog sklopovlja, svi videi (odnosno ulazi podaci iz kamere) su smanjene rezolucije (npr. 380x300). Korišteni algoritam za svaki piksel izgrađuje model s više komponenti, stoga za slike s manjom rezolucijom potrebno je izgraditi manje modela, što u konačnici rezultira bržim izvođenjem algoritma (složenost je $O(N^2)$).

Osim videa, za testiranje u realnom vremenu isprobavali smo i direktno snimanje sa kamere. Aplikacija kao i za video, stvara skup za učenje prvih par slika i zatim obrađuje nadolazeće slike i uspješno određuje koji su pikseli pozadinski, a koji pripadaju objektima u sceni.

Skup za učenje ovog algoritma je ovisan o trenutku u kojem se nalazimo, tj. neki određeni broj slika koji su bili prije slike koja se trenutno prikazuje. Odabiremo period T i za neko vrijeme t imamo sljedeći skup za učenje: $\chi^T = \{x^{(t)}, x^{(t-1)}, ..., x^{(t-T)}\}$

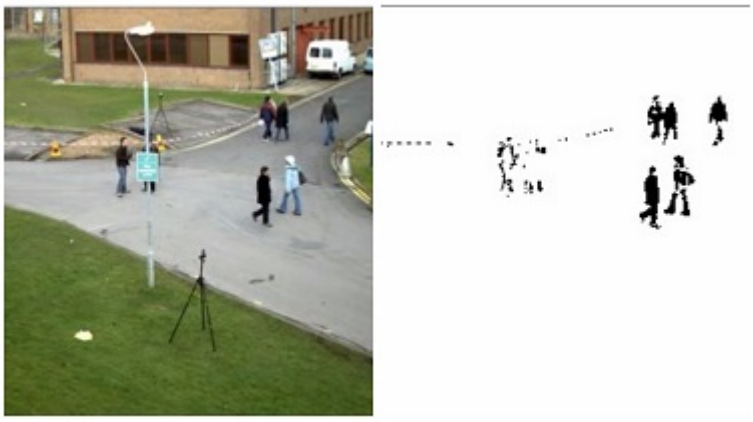
3.2. Rezultati učenja i ispitavanja

U navedenim videima smo imali različita potrebna vremena za procesiranje neke slike u nizu, ovisno o broju pokreta koji su bili prisutni u videu.

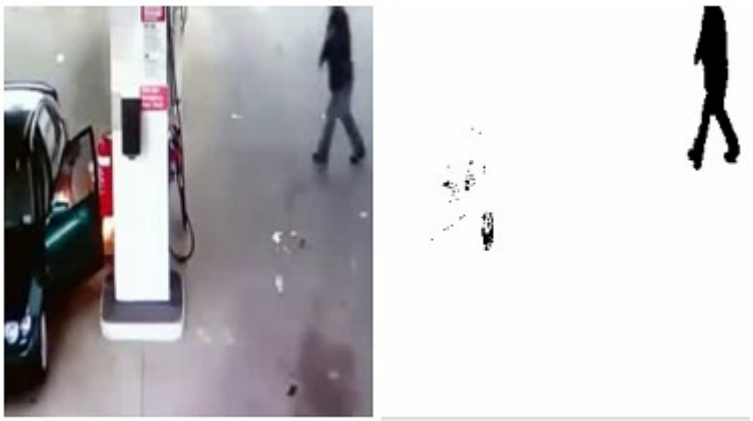
U prvom videu s velikim brojem pokreta (slika 3.1) je potrebno najmanji dio vremena za procesiranje statičnih dijelova scene. Stoga, vrijeme obrade slike je također kratko, u prosjeku 0.017 ms po slici. Za razliku od prvog videa, u drugom je bio umjeren broj pokreta (slika 3.2). Na benzinskoj stanici prilazi čovjek svojem autu, te je

jedine pokrete u videu ostvarivao čovjek. Zbog toga je i malo veće vrijeme obrađivanja svake slike u nizu, u prosjeku 0.021 ms po slici. Posljednji video, prolazak jednog auta u ulici sadržavao je vrlo malo pokreta, a pokreti koji su se dogodili, bili su kratkog trajanja (Slika 3.3). U vrlo statičnoj sceni vrijeme procesiranja slike je nešto veće, približno 0.026 ms prosječno.

Slika 3.1: Scena s velikim brojem pokreta



Slika 3.2: Scena s umjerenim brojem pokreta



Također aplikacija je testirana i u realnom vremenu, odnosno slike su se čitale direktno s kamere. Naravno kao i za video, vrijeme obrade slike ovisi o broju statičnih dijelova scene. U scenama u kojima se snimaju 1-2 osobe prosječno vrijeme obrade slike je 0.022 ms. Povećanjem broja pokretnih objekata u sceni smanjuje se vrijeme obrade. Kamera korištena pri obradi slike u realnom vremenu je Manta g033 gigaE standard, koja omogućava do 88 FPS(Frame Per Second, odnosno broj slika u sekundi koje kamera može slikati). Veličina ćelijica je 9.9 nm, a rezolucija je 656x492. Kamera

Slika 3.3: Scena s malim brojem pokreta

s ovim specifikacijama je omogućila veću otpornost na šum pri snimanju, odnosno kvalitetnije slike. To znači da će "na oko" statične scene doživljavati male promjene boje po pikselima.

3.3. Analiza Rezultata

Napravljena aplikacija vrlo dobro određuje statične dijelove scene kao što je i demonstrirano na slikama. Općenito, broj pokreta u sceni određuje koliko je vremenski zahtjevna obrada te scene. U vrlo dinamičnim scenama, pogotovo koje sadrže vrlo malo statičnih dijelova, brzina obrade slike je velika. To je posljedica načina rada algoritma, jer algoritam pri usporedbi s prošlim slikama nailazi na velike razlike u vrijednostima piksela. Za video gdje su scene prilično statične, potrebno vrijeme procesiranja je veće, zbog toga što algoritam treba obaviti složenije računske operacije, kako bi ustvrdio da piksel ne prelazi graničnu vrijednost, da bi se piksel smatrao da pripada nekom od prednjih objekata u sceni.

U scenama gdje objekti ulaze scenu i potom ostaju nepomični, ti se objekti smatraju pozadinski, to jest stapaju se sa pozadinom. To može biti željeni i neželjeni efekt, no na takve situacije je nemoguće programski utjecati s ovakvim pristupom rješavanja ovog problema. Također problem pri radu ovog algoritma su objekti kojima su neki dijelovi pokretni, a neki su nepokretni. Primjerice čovjek koji pri razgovoru gestikulira rukama, dok ostali dijelovi tijela su nepomični. U tom slučaju program ne razaznaje da je ruka dio većeg objekta, pa se na ekranu za iscrtavanje slika bez pozadine iscrtava samo ruka, i naravno ako je kamera kojom je snimljen video veće kvalitete, iscrtaju se još poneki dijelovi tijela.

Veliku ulogu igra i kamera kojom se snima video za obradu. Veća kvaliteta kamere daje manja odstupanja po pikselima i na kraju bolje rezultate rada aplikacije. Na primjer, obična web kamera daje mutne slike, zatim se i objekt brže stopi s okolinom, ako objekt postaje statičan u sceni. Zato korištenje kamere s boljom kvalitetom mogu pomoći napravljenoj aplikaciji da detektira objekte koji se vrlo malo gibaju, kao na primjer lišće na laganom povjetarcu.

4. Opis programske implementacije rješenja

Programsko rješenje napisano je u programskom jeziku C++ uz korištenje biblioteke OpenCV i programskog okvira Qt.

4.1. OpenCV

Razlog za korištenje biblioteke OpenCV je velika količina već implementiranih funkcija koje obuhvaćaju mnoga područja računalnog vida kao što su dohvaćanje slike s kamere ili videa i obrada slike općenito.

Open Computer Vision library je biblioteka otvorenog koda pod BSD licencom koja sadrži preko 500 algoritama iz podrucja računalnog vida. Algoritmi rade na obradi slika u stvarnom vremenu. Dostupna je na Linux, Windows i MacOS X operacijskim sustavima, a pisana u programskim jezicima C i C++. Sadrži programska sučcelja prema Pythonu, Rubyu, Matlabu, i drugim jezicima. Sadrži module za rad sa matricnim strukturama podataka, rad sa video sadržajem iz datoteka ili kamera, procesiranje i transformaciju slika, izgradnju histograma i kontura, alate za pracenje pokretnih objekata, kalibriranje kamera, izgradnju projekcije i 3D pogleda, algoritme za strojno ucenje, te modul za izgradnju grafičkog sučelja.

U ovom radu OpenCV korišten je za spajanje na web-kameru. Da bi dohvatili video, potrebno je funkciji *VideoCapture* predati redni broj web-kamere ili naziv video datoteke. Nakon toga, čita se slika po slika i šalje u daljnju obradu.

4.2. Qt

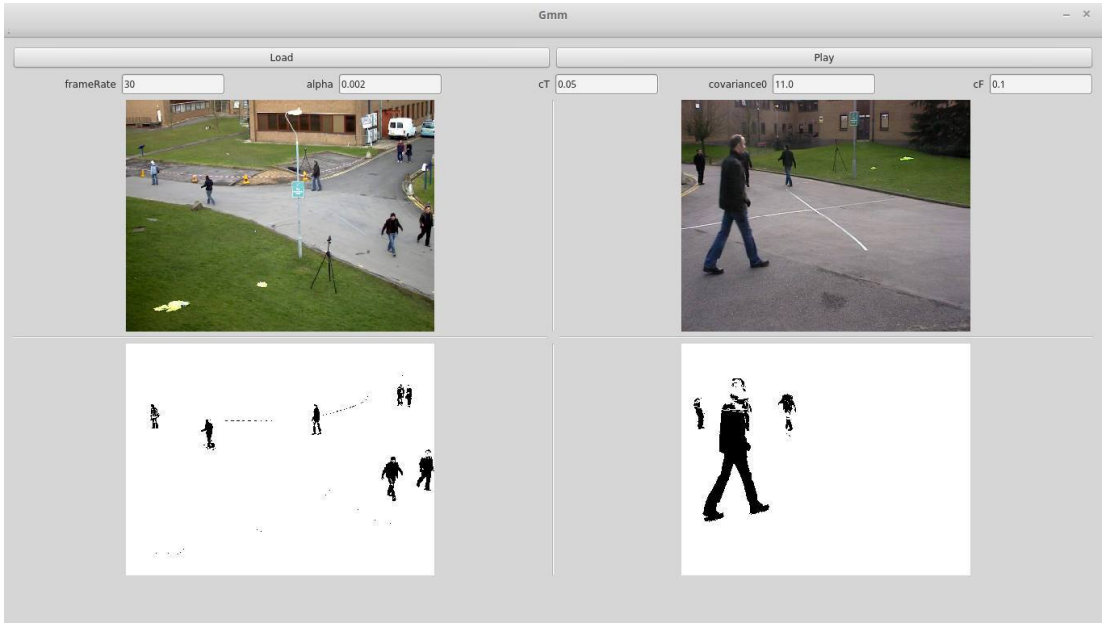
Za izradu grafičkog sučelja korišten je programski okvir *Qt* [1]. Qt je multiplatformski (Linux, Windows, OS X, Symbian) radni okvir koji se široko koristi za razvoj aplikacija s grafičkim sučeljem u jeziku C++. Distribuiran je pod GNU Lesser General Public licencom, a dostupne su komercijalna verzija i *open source* verzija (otvoreni kod). Osim za izradu grafičkih sučelja, Qt se može koristiti i za razvoj aplikacija bez grafičkog sučelja budući da nudi razna proširenja za C++ jezik poput vlastitih stan-

dardnih tipova podataka, kao i podršku za rad s bazama podataka, XML parsiranje, paralelizam te mrežno programiranje. Za početak rada dovoljno je skinuti i instalirati SDK za odgovarajući operacijski sustav. SDK uz sve biblioteke sadrži i *Qt Creator IDE*, jednostavno razvojno okruženje s raznim mogućnostima koje uključuju i sučelja za dizajniranje prozora aplikacija.

U ovom radu korišten je Qt otvorenog koda verzije 5.5. i razvojno okruženje *Qt Creator IDE*.

4.3. Način korištenja

Pri pokretanju programa javlja se prozor sa četiri dijela u kojima se prikazuju izvorne i obrađene slike s uklonjenom pozadinom ispod njih (Slika 4.1). Za učitavanje videa potrebno je pritiskom na gumb *Load* odabrati izvor učitavanja. Obrada počinje pritiskom na gumb *Play*.



Slika 4.1: Grafičko sučelje aplikacije

Ovisno o izvoru učitavanja slika, aplikacija može raditi na dva načina. Prvi način je onaj u kojemu se slike prikazuju i obrađuju u stvarnom vremenu, odnosno izravno s kamere. U tom slučaju korisnik odabire dvije kamere s koje će dobivati sliku. Ako se korisnik odluči za ovaj način rada, preporuča se koristiti dvije web-kamere. Ako je korisnik unaprijed snimio video koji želi obraditi, odabire opciju *Load* i odabire dva željena videa. Kako bi algoritam dao zadovoljavajuće rezultate, preporuča se učitati video minimalne rezolucije 720x540. Također, aplikacija podržava mješoviti način

rada u kojem su izvori slike jedan video i jedna kamera. Taj se način pokreće ako korisnik učita samo jedan video. Tada će se za drugi izvor slike automatski koristiti web kamera ako ona postoji na računalu.

Bez obzira na način na koji aplikacija radi, ispod originalnih slika u stvarnom se vremenu prikazuje njihova verzija obrađena algoritmom GMM opisanom u poglavlju 2. Budući da uspješnost algoritma ovisi o parametrima koje algoritam koristi, korisniku je omogućeno podešavanje nekih od parametara. To su:

- Frame rate - Parametar koji određuje broj frameova (slika) u sekundi. Pretpostavljena vrijednost je 30.
- Alpha - konstanta koja ograničava utjecacaj starijih informacija, postavlja se na otprilike 1/T (2.1)
- Cf - Mjera maksimalnog udjela piksela koji mogu pripadati prednjem dijelu slike bez utjecaja na model pozadine (2.1.1)
- Ct - Fiksirana prihvatljiva vrijednost pristranosti (2.1.2)

5. Zaključak

U okviru ovog rada ostvarena je aplikacija za odvajanje pozadine uporabom algoritma GMM koja radi u stvarnom vremenu (engl. *real time*). GMM algoritam prilagođava se promjenama osvijetljenja u sceni neprestanim ažuriranjem informacija o osvijetljenju i odbacivanjem starih. Koristi Gaussovu normalnu razdiobu i za svaki piksel određuje pripada li pozadini ili ne. Pokazalo se kako uklanjanje pozadine nije jednostavan problem, a osim GMM-om moguće ga je riješiti i na druge načine. Neki od njih su korištenje jezgreno baziranih gustoća razdioba, skriveni Markovljevi modeli (HMM) i slično.

Opisani postupak za uklanjanje pozadine u većini slučajeva na ispitnim uzorcima daje dobre rezultate u uklanjanju pozadine s videa kao i s web-kamere u stvarnom vremenu. Budući da algoritam cijelo vrijeme ažurira informacije, potencijalni problemi se javljaju kada je vremenski interval u kojem objekti na sceni miruju prevelik. Također, problem su video zapisi loše kvalitete. Iako se model pokazao dosta uspješnim, poboljšanja su uvijek moguća. Uz određene prilagodbe parametara rezultati modela mogu biti bolji. Drugo poboljšanja moglo bi bilo korištenje nekog drugog algoritma koji pokazuje bolje rezultate od GMM-a ili korištenje više algoritama istovremeno, odnosno hibrida. Također, aplikacija bi se mogla i proširiti i umjesto dvije kamere, moglo bi se koristiti više kamera. U tom slučaju neka bi se scena mogla promatrati iz više od dva kuta, a aplikacija iskoristiti za videonadzor i detekciju objekata koji se kreću na sceni.

Razvijeno rješenje pokazalo je dobre rezultate uz korištenje algoritma GMM. Moglo bi poslužiti kao temelj za daljnji rad na području uklanjanja pozadine, a uz opisana poboljšanja učinkovitost bi bila još bolja.

Literatura

[1] Qt Company. Qt documentation. 2016.

[2] Ahmed M. Elgammal, David Harwood, i Larry S. Davis. Non-parametric model for background subtraction. U *Proceedings of the 6th European Conference on Computer Vision-Part II*, stranice 751–767, 2000.

[3] Michael Harville. A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models. U *In Proceedings of ECCV*, stranice 543–560, 2002.

[4] Jien Kato, Toyohide Watanabe, Sebastien Joga, Jens Rittscher, i Andrew Blake. An hmm-based segmentation method for traffic monitoring movies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1291–1296, 2002. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2002.1033221.

[5] Antoine Monnet, Anurag Mittal, Nikos Paragios, i Visvanathan Ramesh. Background modeling and subtraction of dynamic scenes. U *ICCV*, stranice 1305–1312, 2003.

[6] Andrea Prati, Ivana Mikic, Mohan M. Trivedi, i Rita Cucchiara. Detecting moving shadows: Algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:918–923, 2003.

[7] Chris Stauffer i W. Eric L. Grimson. Adaptive background mixture models for real-time tracking. *In Proceedings CVPR*, stranice 246–252, 1999.

[8] Kentaro Toyama, John Krumm, Barry Brumitt, i Brian Meyers. Wallflower: Principles and practice of background maintenance. U *ICCV*, stranice 255–261, 1999.

[9] Paul J. Withagen, Klamer Schutte, i Frans C. A. Groen. Likelihood-based object detection and object tracking using color histograms and em. U *ICIP (1)*, stranice 589–592, 2002.

[10] Christopher Wren, Ali Azarbayejani, Trevor Darrell, i Alex Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:780–785, 1997.

[11] Zoran Zivkovic i Ferdinand van der Heijden. Recursive unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, stranice 651–656, 2004.