Elektrotehniški vestnik XX(Y): 1–6, YEAR
Electrotechnical Review, Ljubljana, Slovenija

Robustno vizualno učenje na podlagi podprostorov

Danijel Skočaj

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Tržaška 25, 1001 Ljubljana, Slovenija
E-pošta: danijel.skocaj@fri.uni-lj.si

Povzetek. Vizualno učenje, tj. učenje iz vizualnih podatkov, mora biti robusten in kontinuiran proces. Vsi razpoložljivi vizualni podatki namreč niso enako pomembni; v primeru prekrivanj in drugih nezaželenih motenj v vidnem polju so lahko nekateri celo zavajajoči. Človeški vizualni sistem obravnava vizualne podatke selektivno in zgradi učinkovite predstavitve opazovanih predmetov in prizorov tudi v slabih pogojih. Te predstavitve lahko nato še posodablja z na novo pridobljenimi informacijami in jih tako prilagaja spremembam. V tem članku predstavljamo več metod, ki uvedejo podobne principe tudi na področje strojnega vizualnega učenja in razpoznavanja. Vizualno učenje je realizirano z modeliranjem osnovanim na direktnem videzu predmetov in prizorov. Gradnja modelov temelji na metodi glavnih komponent (PCA), ki pa ima v svoji standardni izvedbi pomanjkljivosti, ki onemogočajo uveljavitev prej omenjenih načel. Za premostitev teh pomanjkljivosti smo predlagali več razširitev standardne metode glavnih komponent, tj. metode za inkrementalno, uteženo in robustno učenje. Predlagane metode smo tudi ovrednotili na različnih slikovnih domenah. Iz rezultatov je razvidna uporabnost metod za vizualno učenje in razpoznavanje v različnih primerih.

Ključne besede: računalniški vid, na videzu zasnovano modeliranje, analiza glavnih komponent, inkrementalno učenje, robustno učenje

Robust subspace visual learning

Extended abstract. In the real world, visual learning is supposed to be a robust and continuous process. All available visual data is not equally important; in the case of occlusions or other undesirable intrusions in the field of view some visual data can even be misleading. Human visual system treats visual data selectively and builds efficient representations of observed objects and scenes even under non-ideal conditions. Furthermore, these representations can afterwards be updated with newly acquired information. In this paper we present several methods, which introduce similar principles in the machine visual learning and recognition as well.

We approach visual learning by the appearance-based modeling of objects and scenes. Models are built using principal component analysis (PCA), which has several shortcomings with respect to the premises mentioned above. In order to overcome them, we propose several extensions of the standard PCA.

PCA-based learning is traditionally performed in a batch mode, thus requiring all training images to be given in advance. Since this is not admissible in the framework of continuous learning, we propose an incremental method that processes images sequentially one by one and updates the representation at each step accordingly. Each image can be discarded immediately after the model is updated, which makes the method perfectly well suited for real on-line scenarios.

In addition, in the standard PCA approach all pixels of an image receive equal treatment. Also, all training images have equal influence on the estimation of principal subspace. In this paper, we present a generalized PCA approach, which estimates principal axes and principal components considering weighted pixels and images. We further extend this weighted approach into a method for learning from incomplete data, which builds the model of an object even when the part of input data is missing.

Images of objects and scenes are not always ideal; they may contain various deceptive additions like reflections or occlusions. PCA in its standard form is intrinsically non-robust to such non-gaussian noise. Several methods for robust recognition have already been proposed, however robust learning has been tackled very rarely. In the paper we present a novel approach to the robust subspace learning. The proposed batch and incremental methods detect inconsistencies in the training images and build the representations from consistent data only. As a result, the obtained models are more robust and efficient enabling more reliable visual learning and recognition even when the learning conditions are not ideal.

In the paper we briefly describe all the methods mentioned above. We also present evaluation results of the proposed algorithms on different image domains (Fig. 1) and determine the applicability of the methods in different scenarios. A detailed derivation and analysis of all the methods can be found in [7].

Key words: computer vision, appearance-based modeling, principal component analysis, incremental learning, robust learning

Prejet 4. december, 2003
Odobren 13. december, 2004

1 Uvod

Čeprav je vid oz. interpretacija vizualne informacije zelo preprosta naloga za človeka, je to zelo zahtevno opravilo za računalnik. Raziskovalci na področju računalniškega vida se že zelo dolgo trudijo, da bi zmogljivosti računalniških sistemov za vizualno zaznavanje vsaj deloma približali sposobnostim človeka. Potem ko so različne psihofizične, nevrofiziološke in behavioristične študije postregle z dokazi, da je človeško razpoznavanje objektov zelo odvisno od smeri pogleda na predmet [10], se je tudi v računalniškem vidu bolj uveljavil *glediščno-osnovani* (ang. view-based) pristop do učenja in razpoznavanja objektov in prizorov. Pri tem pristopu se objekti ne modelirajo kot glediščno invariantni 3-D modeli, temveč je predstavitev sestavljena iz množice 2-D pogledov. Da pa tak pristop ne bi bil časovno in prostorsko preveč potraten, je treba vse vhodne slike zajete z različnih pogledov učinkovito predstaviti v zgoščeni predstavitvi, ki omogoča hitro iskanje oz. neposredno ujemanje testnih slik z učnimi.

V ta namen se pogosto uporabljajo metode na podlagi podprostorov, kot je *metoda glavnih komponent PCA* (ang. principal component analysis) [4]. Osnovna ideja metode PCA je, da se visokodimenzijski vhodni podatki na učinkovit način preslikajo v nizkodimenzijski podprostor in pri tem ohranijo čim več informacije. Napaka, ki jo naredimo pri preslikavi vhodnih podatkov v podprostor, ki ima glavne smeri za bazne vektorje, je minimalna (v smislu napake najmanjših kvadratov) med vsemi linearnimi preslikavami v podprostor enake dimenzije.

Kljub preprostosti (ali pa ravno zaradi nje) ima na videzu zasnovani pristop, temelječ na analizi glavnih komponent, kar precej pomanjkljivosti. Tako se glavni podprostor ponavadi računa s paketno metodo, kjer se vse učne slike procesirajo hkrati, kar pomeni, da morajo biti vse podane vnaprej. Ravno tako standardna metoda za vizualno učenje ne upošteva dejstva, da imajo lahko tako različni deli slik kot tudi celotne posamezne slike različen vpliv na proces učenja ter da nekateri deli slik lahko celo manjkajo, čeprav je to v praksi zelo verjetno. Analiza glavnih komponent je tudi zelo občutljiva na šum na slikah, kar je v resničnih primerih lahko resen problem.

Razni avtorji so se že v preteklosti posvečali tem problemom in tako predstavili algoritme za inkrementalno [1, 3], uteženo [11, 2], pa tudi robustno [2] gradnjo podprostorov. Metode, ki jih predstavljamo v tem članku, pomenijo korak naprej od že predlaganih. Posebej so prilagojene gradnji podprostorov za vizualno učenje in razpoznavanje in dosledno upoštevajo zahteve, ki naj bi jih izpolnjeval adaptivni in robustni sistem. Tako so za vse omenjene pristope (uteženo učenje, učenje iz delnih podatkov, robustno učenje) predstavljene paketne in inkrementalne metode ter analize za kakšne vrste slik in pod kakšnimi pogoji so te metode uporabne.

Zaradi omejenega prostora, ki je na voljo, so vsi predlagani pristopi zelo zgoščeno opisani in ovrednoteni. Natančna izpeljava in analiza vseh metod je na voljo v [7].

2 Osnovna PCA

Recimo, da imamo N M-dimenzionalnih vektorjev (učnih slik) $\mathbf{x}_j$ zloženih v matriko podatkov $\mathbf{X} \in \mathbb{R}^{M\times N}$. Naš cilj je najti takšne smeri $\mathbf{u}_i$ (vektorje dolžine 1) v vhodnem prostoru $\mathbb{R}^M$, ki maksimirajo varianco preslikav vseh vhodnih vektorjev na vektorje $\mathbf{u}_i$. Izkaže se, da moramo za to rešiti klasičen problem razcepa z lastnimi vrednostmi (EVD) kovariančne matrike $\mathbf{C}$:

$$\mathbf{C}\mathbf{u} = \lambda\mathbf{u} \ , \tag{1}$$

ki ga lahko rešimo tudi z razcepom s singularnimi vrednostmi (SVD):

$$\mathbf{C} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top \ , \tag{2}$$

tako da ortonormalna matrika $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N] \in \mathbb{R}^{M\times N}$ vsebuje po stolpcih lastne vektorje $\mathbf{u}_1, \ldots, \mathbf{u}_N$, diagonalna matrika $\mathbf{\Lambda} \in \mathbb{R}^{N\times N}$ pa na diagonali lastne vrednosti $\lambda_1, \ldots, \lambda_N$. Predpostavili bomo, da so lastne vrednosti in pripadajoči lastni vektorji urejeni po padajočem vrstnem redu lastnih vrednosti, torej $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$. Tako je večina variabilnosti vhodnih vektorjev vsebovana že v prvih lastnih vektorjih, ki jih imenujemo tudi *glavni vektorji* ali *glavne osi*. Ohranimo torej samo prvih $k, k \ll N$ lastnih vektorjev, torej $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \mathbb{R}^{M\times k}$. Z njimi in z vektorji koeficientov (nizkodimenzijskimi predstavitvami učnih slik), ki sestavljajo matriko $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N] \in \mathbb{R}^{k\times N}$, lahko opišemo učne slike (z odšteto povprečno sliko $\boldsymbol{\mu}$) $\hat{\mathbf{x}}_i$ takole:

$$\hat{\mathbf{X}} = \mathbf{U}\mathbf{A} \ , \tag{3}$$

s čimer se zagotavlja minimalna rekonstrukcijska napaka:

$$e = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(\hat{x}_{ij} - \sum_{l=1}^{k} u_{il}a_{lj}\right)^2 \ . \tag{4}$$

Glavni podprostor pa lahko dobimo tudi z EM algoritmom [6]. Algoritem izračuna glavne smeri in glavne komponente na iterativen način z izmeničnim izvajanjem naslednjih korakov E in M:

- **E-korak**: $\mathbf{A} = (\mathbf{U}^\top\mathbf{U})^{-1}\mathbf{U}^\top\hat{\mathbf{X}}$
- **M-korak**: $\mathbf{U} = \hat{\mathbf{X}}\mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}$.

3 Inkrementalna PCA

V tem razdelku bomo predstavili algoritem, ki zgradi glavni podprostor postopoma po korakih [9]. Recimo, da imamo že zgrajen glavni podprostor iz prvih n slik. V koraku $n+1$ bi lahko izračunali novi glavni podprostor iz *rekonstrukcij* prvih n slik in nove slike z navadno paketno metodo. Seveda pa bi bila računska zahtevnost takšnega inkrementalnega algoritma previsoka, saj bi morali na vsakem koraku izvesti paketno metodo PCA na množici visokodimenzijskih podatkov. Identične rezultate pa lahko dobimo z uporabo nizkodimenzijskih *vektorjev koeficientov* (predstavitev) prvih n slik, saj so vektorji koeficientov in njihove rekonstrukcije v bistvu iste točke predstavljene v različnih koordinatnih sistemih. Ker je dimenzija glavnega podprostora majhna, je takšen algoritem (Algoritem 1) zelo učinkovit.

Algorithm 1 : Inkrementalna PCA

Input: trenutni povprečni vektor $\boldsymbol{\mu}^{(n)}$, trenutni lastni vektorji $\mathbf{U}^{(n)}$, trenutni koeficienti $\mathbf{A}^{(n)}$, nova vhodna slika $\mathbf{x}$.

Output: novi povprečni vektor $\boldsymbol{\mu}^{(n+1)}$, novi lastni vektorji $\mathbf{U}^{(n+1)}$, novi koeficienti $\mathbf{A}^{(n+1)}$, nove lastne vrednosti $\boldsymbol{\lambda}^{(n+1)}$.

1: Projiciraj novo sliko $\mathbf{x}$ v trenutni lastni prostor: $\mathbf{a} = \mathbf{U}^{(n)\top}(\mathbf{x} - \boldsymbol{\mu}^{(n)})$.
2: Rekonstruiraj novo sliko: $\mathbf{y} = \mathbf{U}^{(n)}\mathbf{a} + \boldsymbol{\mu}^{(n)}$.
3: Izračunaj vektor razlike: $\mathbf{r} = \mathbf{x} - \mathbf{y}$. $\mathbf{r}$ je pravokoten na $\mathbf{U}^{(n)}$.
4: Dodaj $\mathbf{r}$ kot nov bazni vektor: $\mathbf{U}' = \begin{bmatrix} \mathbf{U}^{(n)} & \frac{\mathbf{r}}{\|\mathbf{r}\|} \end{bmatrix}$.
5: Določi vrednosti koeficientov v novi bazi: $\mathbf{A}' = \begin{bmatrix} \mathbf{A}^{(n)} & \mathbf{a} \\ \mathbf{0} & \|\mathbf{r}\| \end{bmatrix}$.
6: Izvedi PCA na $\mathbf{A}'$. Dobi povprečno vrednost $\boldsymbol{\mu}''$, lastne vektorje $\mathbf{U}''$ in lastne vrednosti $\boldsymbol{\lambda}''$.
7: Projiciraj vektorje koeficientov na novo bazo: $\mathbf{A}^{(n+1)} = \mathbf{U}''^{\top}(\mathbf{A}' - \boldsymbol{\mu}''\mathbf{1}_{1\times n+1})$.
8: Zavrti podprostor $\mathbf{U}'$ za $\mathbf{U}''$: $\mathbf{U}^{(n+1)} = \mathbf{U}'\mathbf{U}''$.
9: Popravi povprečni vektor: $\boldsymbol{\mu}^{(n+1)} = \boldsymbol{\mu}^{(n)} + \mathbf{U}'\boldsymbol{\mu}''$.
10: Nove lastne vrednosti: $\boldsymbol{\lambda}^{(n+1)} = \boldsymbol{\lambda}''$.

Algoritem poveča dimenzijo glavnega podprostora za eno. Če hočemo ohraniti dimenzijo podprostora nespremenjeno, lahko zavržemo najmanj pomemben bazni vektor. Za začetni podprostor lahko postavimo kar prvo učno sliko, torej $\boldsymbol{\mu}^{(1)} = \mathbf{x}_1$, $\mathbf{U}^{(1)} = \mathbf{0}_{M\times 1}$, in $\mathbf{A}^{(1)} = 0$. Tako je algoritem popolnoma inkrementalen, saj na vsakem koraku zahteva samo eno sliko. Ker se predstavitve slik hranijo in obnavljajo skozi celoten proces učenja, lahko namreč vsako učno sliko takoj po uporabi zavržemo.

4 Utežena PCA

4.1 Paketna metoda za uteženo PCA

Standardna metoda za analizo glavnih komponent se lahko prevede na uteženo PCA, če v minimizacijsko funkcijo (4) dodamo uteži. Naj matrika $\mathbf{W} \in \mathbb{R}^{M\times N}$ vsebuje uteži, tako da je w_{ij} utež i-tega slikovnega elementa na j-ti sliki. Cilj je minimizirati *uteženo* kvadratno rekonstrukcijsko napako

$$\mathcal{E} = \sum_{i=1}^{M}\sum_{j=1}^{N} w_{ij}\left(\hat{x}_{ij} - \sum_{l=1}^{k} u_{il}a_{lj}\right)^2 \quad . \qquad (5)$$

Vrednosti matrike $\hat{\mathbf{X}}$ so v tem primeru dobljene z odštevanjem *uteženega* povprečnega vektorja $\boldsymbol{\mu}$ od učnih slik $\mathbf{x}_i$.

V praksi se ponavadi srečamo z dvema tipoma uteži: s *časovnimi* utežmi ${}^{\mathbf{t}}\mathbf{w} \in \mathbb{R}^{1\times N}$, ki utežijo posamezne slike, in s *prostorskimi* utežmi ${}^{\mathbf{s}}\mathbf{w} \in \mathbb{R}^{M}$, ki utežijo posamezne slikovne elemente neke slike.

Časovne uteži določajo vpliv posameznih slik na proces učenja. Če želimo neki sliki dodeliti večjo vlogo pri določanju glavnih osi, ji priredimo večjo utež. V tem primeru torej maksimiramo uteženo varianco projekcij. Lastni prostor lahko tako izračunamo z razcepom s singularnimi vrednostmi *utežene* kovariančne matrike.

Algoritem za izračun glavnega podprostora z upoštevanjem *splošnih uteži* bomo izpeljali iz EM algoritma. Spomnimo se, da v E-koraku tega algoritma izračunamo nove vrednosti $\mathbf{a}_j$ takole: $\mathbf{a}_j = (\mathbf{U}^\top\mathbf{U})^{-1}\mathbf{U}^\top\hat{\mathbf{x}}_j$. Ta izraz pa lahko izrazimo tudi s pseudoinverzom $\mathbf{a}_j = \mathbf{U}^\dagger\hat{\mathbf{x}}_j$, ki pa je po drugi strani ekvivalenten reševanju ustreznega sistema linearnih enačb. Podobne opazke veljajo tudi za M-korak EM algoritma. Če posamezne enačbe ustrezno utežimo, lahko EM algoritem priredimo za iskanje takšnega podprostora, ki minimizira uteženo rekonstrukcijsko napako. Koraka E in M sta sedaj videti takole:

- **E-korak**: Izračunaj $\mathbf{A}$ na naslednji način: Za vsako sliko j, $j = 1\ldots N$ reši naslednji sistem linearnih enačb v smislu napake najmanjših kvadratov:

$$\sqrt{w_{ij}}\hat{x}_{ij} = \sqrt{w_{ij}}\sum_{p=1}^{k} u_{ip}a_{pj} \ , \ i = 1\ldots M \ . \quad (6)$$

- **M-korak**: Izračunaj $\mathbf{U}$ takole: Za vsak slikovni element i, $i = 1\ldots M$ reši naslednji sistem linearnih enačb v smislu napake najmanjših kvadratov:

$$\sqrt{w_{ij}}\hat{x}_{ij} = \sqrt{w_{ij}}\sum_{p=1}^{k} u_{ip}a_{pj} \ , \ j = 1\ldots N \ . \quad (7)$$

4.2 Paketna metoda za PCA na delnih podatkih

V praktičnih aplikacijah se pogosto zgodi, da vrednosti nekaterih slikovnih elementov niso definirane. Takšne situacije lahko obravnavamo kot poseben primer uteženega učenja, kjer postavimo uteži manjkajočih elementov na nič. Pri algoritmu EM to pomeni, da postavimo v sisteme linearnih enačb (6) in (7) samo tiste enačbe, ki izhajajo iz poznanih slikovnih elementov. Takšno reševanje tega problema pa je včasih slabo pogojeno. Zato predlagamo, da se v primerih, ko so učne slike posnete v zaporedju, tako da so sosednje slike med seboj precej podobne, uporabi še dodatna omejitev. V M-koraku poleg minimizacije rekonstrukcijske napake v znanih slikovnih elementih hkrati minimiziramo tudi drugi odvod rekonstruiranih vrednosti manjkajočih slikovnih elementov [8]. To zagotavlja, da bodo vrednosti rekonstruiranih manjkajočih slikovnih elementov gladke skozi čas.

Alternativno lahko vrednosti $\mathbf{U}$ v M-koraku izračunamo tudi z uporabo navadne paketne metode PCA na celotni matriki vhodnih podatkov pri pogoju, da so manjkajoči elementi ustrezno zapolnjeni. Vprašanje je, katera je optimalna metoda za polnjenje manjkajočih elementov. Ker ne poznamo vrednosti vseh slikovnih elementov slike, so nekatere koordinate ustrezne točke v prostoru slik nedefinirane. Pravilni položaj točke je torej nekje v podprostoru, ki ga določajo manjkajoče koordinate. Ob poznavanju glavnega podprostora $\mathbf{U}$, ki modelira vhodne podatke, je optimalna lokacija točka v podprostoru manjkajočih elementov, ki je najbliže glavnemu podprostoru. Dobimo jo tako, da manjkajoče slikovne elemente zapolnimo z rekonstruiranimi vrednostmi, ki jih izračunamo iz koeficientov, dobljenih v E-koraku trenutne iteracije z uporabo glavnih osi, dobljenih v M-koraku predhodne iteracije.

4.3 Inkrementalna utežena PCA

Inkrementalni algoritem lahko dopolnimo s principi uteženega učenja in tako dobimo algoritem za inkrementalno uteženo učenje [9].

Časovne uteži je zelo preprosto vključiti v inkrementalni algoritem. Jedro tega algoritma je še vedno standardna paketna metoda PCA na vektorjih koeficientov. Vse, kar moramo storiti, je, da to metodo zamenjamo z uteženo paketno metodo.

Prostorske uteži pa uvedemo v inkrementalno gradnjo z dodatnim korakom. Naj bodo prostorske uteži vrednosti z intervala med 0 in 1. Če je vrednost uteži 1, to pomeni, da se na vrednost slikovnega elementa popolnoma zanesemo in jo uporabimo takšno, kot je. Če je vrednost uteži 0, to pomeni, da je vrednost slikovnega elementa popolnoma nerelevantna in se nanjo popolnoma nič ne zanesemo. Takšno vrednost lahko rekonstruiramo z upoštevanjem trenutnega modela, ki smo ga zgradili v predhodnih korakih. S postavljanjem uteži med 0 in 1 lahko uravnavamo vpliv vrednosti slikovnega elementa in vpliv trenutnega modela (rekonstruirane vrednosti). Na začetku inkrementalnega algoritma torej uvedemo nov korak, ki najprej izračuna koeficiente nove slike $\mathbf{x}$ na utežen način (6) in te koeficiente rekonstruira v sliko $\mathbf{y}$. Nato dobimo novo sliko kot uteženo vsoto originalne in rekonstruirane slike

$$x_i^{new} = {}^s\!w_i x_i + (1 - {}^s\!w_i) y_i \ , \ i = 1 \dots M \ , \qquad (8)$$

ki jo potem uporabimo za posodobitev trenutnega glavnega podprostora z inkrementalnim algoritmom.

4.4 Inkrementalna PCA na delnih podatkih

V primeru algoritma za računanje glavnega podprostora v prisotnosti manjkajočih slikovnih elementov se inkrementalni uteženi algoritem poenostavi tako, da imajo uteži lahko samo vrednosti 0 (manjkajoči slikovni elementi) ali 1 (znani slikovni elementi). Korak z uteženo vsoto se torej poenostavi na polnjenje vrednosti manjkajočih slikovnih elementov z rekonstruiranimi vrednostmi na podoben način kot pri paketni metodi, ki temelji na iterativni rekonstrukciji manjkajočih slikovnih elementov.

5 Robustna PCA

Pri uteženi metodi za izračunavanje glavnega podprostora predpostavljamo, da so uteži (ali manjkajoči slikovni elementi) poznane. V praktičnih primerih pa to velikokrat ne drži, tako da mora algoritem sam odkriti odstopajoče slikovne elemente (ang. outliers). To je naloga robustnih algoritmov.

5.1 Robustna paketna metoda

Paketni robustni algoritem [8] najprej odkrije odstopajoče slikovne elemente na vhodnih slikah. Kot odstopajoči se obravnavajo vsi slikovni elementi, ki niso konsistentni z istoležnimi elementi na drugih slikah. Konsistenca se preverja s primerjanjem rekonstrukcijskih napak. Algoritem najprej iz vseh (neidealnih) vhodnih slik zgradi glavni podprostor na nerobusten način. Vse slike nato projicira v dobljeni podprostor in jih rekonstruira. Ker odstopajoči slikovni elementi niso konsistentni z drugimi, je rekonstrukcijska napaka v njih ponavadi večja. Zato algoritem obravnava vse slikovne elemente z veliko rekonstrukcijsko napako kot manjkajoče slikovne elemente in nato izračuna robustni glavni podprostor z eno izmed prej omenjenih metod za učenje iz delnih podatkov. Nato se vrednosti teh slikovnih elementov v matriki vhodnih podatkov zamenjajo z rekonstruiranimi vrednostmi in celoten postopek se ponovi. Algoritem ponavadi zelo hitro skonvergira in že po nekaj iteracijah dobimo v vseh slikovnih elementih vrednosti, ki so konsistentne s preostalimi.

Če množica učnih slik vsebuje manjše število slik z zelo veliko odstopajočimi slikovnimi elementi, lahko ta postopek še izboljšamo tako, da najprej odkrijemo *odstopajoče slike*, torej slike, v katerih je povprečna rekonstrukcijska napaka slikovnih elementov velika, nato pa za odkrivanje *odstopajočih slikovnih elementov* na vseh slikah uporabimo samo informacijo, vsebovano v dobrih slikah.

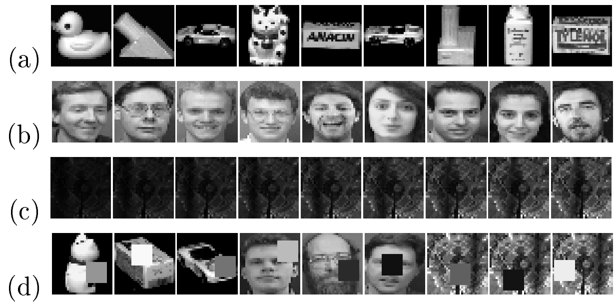
5.2 Robustna inkrementalna metoda

Robustni inkrementalni algoritem [9] je zelo podoben inkrementalnemu algoritmu za učenje iz delnih podatkov. Slednjemu je dodan samo še korak, ki odkrije odstopajoče slikovne elemente, ki se potem obravnavajo kot manjkajoči slikovni elementi. Nova vhodna slika se torej najprej robustno projicira [5] v trenutni glavni podprostor in nato rekonstruira. Na podlagi rekonstrukcijskih napak se določijo odstopajoči slikovni elementi in njihove vrednosti se zamenjajo z rekonstruiranimi vrednostmi. Tako popravljena slika se nato uporabi za posodobitev glavnega podprostora z inkrementalnim algoritmom.

6 Ovrednotenje algoritmov

Med raziskovalnim delom smo izvedli veliko poskusov, s katerimi smo analizirali učinkovitost predlaganih algoritmov. Veliko rezultatov je predstavljenih v [7], v tem članku pa bomo na zgoščen način predstavili le najpomembnejše.

Da bi dobili realistično sliko o uporabnosti predlaganih algoritmov, smo le-te testirali na treh različnih slikovnih domenah: na slikah predmetov, obrazov in ozadja. Za slike predmetov (slika 1a) je namreč značilno, da niso časovno najbolj korelirane, saj se med seboj precej razlikujejo. Obrazi (slika 1b) so časovno veliko bolj koherentni, še najbolj pa so časovno korelirane slike ozadja (slika 1c). Ozadje je namreč statično, spreminja se samo globalna osvetlitev, ki se enakomerno povečuje, in vertikalna senca, ki potuje čez sliko. Za slike ozadja je tudi značilno, da se zelo počasi in enakomerno spreminjajo, kar do neke mere velja tudi za predmete, ker so bile sosednje slike zajete iz sosednjih zornih kotov. To še najmanj velja za slike obrazov, ker le-ti niso bili posneti po kakšnem določenem vrstnem redu poz. Za oceno uteženih in robustnih algoritmov smo nato na vse slike, razen na prvih 40 v obeh vrstnih redih (vseh je bilo 720) naključno dodali kvadratek kot nezaželen in moteč element (slika 1d).



Slika 1. Nekaj učnih slik: (a) predmeti, (b) obrazi, (c) ozadje. (d) Nekaj deloma prekritih slik.
Figure 1. Some training images: (a) objects, (b) faces, (c) background. (d) Some occluded training images.

Glavne lastnosti vseh treh tipov slik in učinkovitosti različnih algoritmov so povzete v tabeli 1. Navedli bomo samo nekaj najbolj značilnih rezultatov.

Učne slike smo na vhode algoritmov podajali v dveh različnih vrstnih redih. Najprej smo jih uredili po naravnem zaporednem vrstnem redu – najprej po vrsti vse slike enega predmeta, nato drugega in tako naprej. Nato smo jih uredili v izmeničnem vrstnem redu – najprej po ena slika vsakega predmeta, nato druga slika vsakega predmeta in tako naprej. Ugotovili smo, da vrstni red pomembno vpliva na rezultate inkrementalnih algoritmov, saj smo pri izmeničnem vrstnem redu vhodnih slik dobili precej boljše rezultate. To velja tako za navadno inkrementalno metodo kot za uteženo, še najbolj pa za robustno inkrementalno metodo. Pri navadni inkrementalni metodi je izmenični vrstni red ugodnejši, ker nam že na začetku učenja zgradi dovolj dober približek končnega glavnega podprostora, ki se nato samo še izboljšuje. Pri uteženi metodi je ta vpliv še večji, ker se trenutni model uporablja tudi za določanje novih vrednosti slikovnih elementov z majhnimi utežmi na novi sliki. Največji vpliv pa ima vrstni red pri robustni metodi, saj se v tem primeru trenutni model uporablja tudi za odkrivanje odstopajočih slikovnih elementov na novi sliki. Če trenutni model vsaj zelo približno ne zajema tudi videza nove slike, se vsi slikovni elementi, ki se na njej razlikujejo od prejšnjih istoležnih elementov, označijo za odstopajoče, torej tudi tisti, ki dejansko to niso, ampak prinašajo samo neko novo informacijo, ki bo sčasoma postala konsistentna. Zato moramo težiti k temu, da algoritmom za inkrementalno učenje (predvsem robustnemu) že na začetku podamo vsaj nekaj slik, ki v grobem predstavljajo različne videze predmeta ali prizora.

Osnovna inkrementalna metoda je zelo

učinkovita, saj bistveno ne poslabša rezultatov paketne metode (povečanje kvadratne rekonstrukcijske napake je bilo vedno manj kot 10-odstotno). Pri deloma prekritih slikah se dobro obnesejo tudi utežene metode in metode za učenje iz deloma manjkajočih podatkov. Pri testiranju teh metod smo namreč predpostavili, da so lokacije odstopajočih slikovnih elementov poznane, zato smo tem slikovnim elementom priredili manjše uteži oz. jih označili za manjkajoče. Predlagane metode so uspešno rekonstruirale prave vrednosti odstopajočih slikovnih elementov in zgradile predstavitve, ki nezaželenih kvadratkov niso vsebovale.

Rezultati robustne metode pa so bili zelo odvisni od vhodnih slik. Zelo slabo se je obnesla pri modeliranju predmetov. Problem je namreč v tem, da slike predmetov niso najbolj korelirane, tako da določanje odstopajočih slikovnih elementov samo na podlagi konsistentnosti posameznih slikovnih elementov ni bilo dovolj uspešno. Posledično so bili v predstavitve vgrajeni tudi nekateri kvadratki, izvzeti pa nekateri deli predmetov, ki so bili odkriti kot odstopajoči slikovni elementi, čeravno to niso bili (recimo robovi predmetov). Ta problem je bil veliko manj opazen pri slikah obrazov, ki so časovno bolj koherentne. Najboljše rezultate pa je robustni algoritem dosegel pri modeliranju ozadja, kjer je uspešno odkril večino odstopajočih slikovnih elementov in zgradil predstavitev ozadja brez motečih kvadratkov.

	PR	OB	OZ
časovna korelacija	-	+	++
enakomernost spreminjanja slik	+	-	++
pomen vrstnega reda slik	+	-	++
učinkovitost inkr. metode	++	++	++
učinkovitost utežene metode	+	+	++
učinkovitost robustne metode	- -	+	++

Tabela 1. Glavne lastnosti treh tipov slik (PR predmeti, OB obrazi, OZ ozadje) in učinkovitost algoritmov (- - zelo majhna, - majhna, + velika, ++ zelo velika.

7 Sklep

V članku smo se v glavnem ukvarjali s povečevanjem robustnosti vizualnega učenja zasnovanega na uporabi podprostorov. Analizirali smo različne probleme, ki nastajajo pri tem, in predstavili različne rešitve, odvisne od količine informacij, ki je na voljo.

V situacijah, v katerih so zanesljivosti posameznih slik in slikovnih elementov že vnaprej poznane, lahko uporabimo predstavljene metode za *uteženo učenje*. Za primere, ko so na slikah nekateri slikovni elementi nedefinirani, smo predlagali algoritme za *učenje ob manjkajočih podatkih*. Za primere, ko nimamo nobenih dodatnih informacij o vsebini slik, pa smo predlagali algoritme za *robustno učenje*, ki poskušajo sami odkriti odstopajoče slikovne elemente in zgraditi robustni glavni podprostor. Za vse te pristope k vizualnemu učenju na podlagi podprostorov smo predlagali tako *paketne* kot tudi *inkrementalne* algoritme.

Predstavljene algoritme lahko uporabimo ne le na sivinskih učnih slikah, ampak tudi na slikah drugih modalnosti. Lahko pa na slikah poiščemo tudi razne druge značilke, lokalne ali globalne, ki jih nato uporabimo za modeliranje predmetov s PCA. Tako bi lahko čisto holistični glediščno-osrediščeni pristop obogatili z lokalnimi geometričnimi značilkami in strukturnimi informacijami. To pa bi lahko še dodatno bistveno pripomoglo k izboljšanju učinkovitosti na videzu zasnovanega vizualnega učenja in razpoznavanja.

8 Literatura

[1] S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler, and H. Zhang. An eigenspace update algorithm for image analysis. *Graphical Models and Image Processing*, 59(5):321–332, September 1997.

[2] F. De la Torre and M. J. Black. A framework for robust subspace learning. *IJCV*, 54(1):117–142, 2003.

[3] P. Hall, D. Marshall, and R. Martin. Merging and spliting eigenspace models. *PAMI*, 22(9):1042–1048, 2000.

[4] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[5] A. Leonardis and H. Bischof. Robust recognition using eigenimages. *CVIU*, 78:99–118, 2000.

[6] S. Roweis. EM algorithms for PCA and SPCA. In *NIPS'97*, pages 626–632, 1997.

[7] D. Skočaj. *Robust subspace visual learning and recognition.* PhD thesis, University of Ljubljana, Faculty of compuer and information science, 2003.

[8] D. Skočaj, H. Bischof, and A. Leonardis. A robust PCA algorithm for building representations from panoramic images. In *Computer Vision – ECCV 2002*, volume IV, pages 761–775, May 2002.

[9] D. Skočaj and A. Leonardis. Weighted and robust incremental method for subspace learning. *ICCV 2003*, II:1494–1501, October 2003.

[10] M. J. Tarr and H. H. Bülthoff. *Image-based object recognition in man, monkey and machine.* MIT/Elsevier, 1998.

[11] T. Wiberg. Computation of principal components when data are missing. In *Proc. Second Symp. Computational Statistics*, pages 229–236, 1976.

Danijel Skočaj je samostojni raziskovalec na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer je leta 2003 tudi doktoriral. Glavno področje njegovega dela je avtomatsko modeliranje objektov iz vizualnih informacij s poudarkom na vizualnem učenju in razpoznavanju na podlagi videza.