

**УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ  
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ**

**Игор Марић  
Драган Грачанин**

**ИЗДВАЈАЊЕ РЕГИОНА У КОЈИМА СЕ НАЛАЗИ  
ТАБЛИЦА СА СЛИКЕ, НОРМАЛИЗАЦИЈА  
ВЕЛИЧИНЕ ТАБЛИЦЕ И УКЛАЊАЊЕ ДИСТОРЗИЈЕ**

семестрални рад

**Предмет: Дигитална обрада слике  
Предметни наставник: Проф. др Зденка Бабић  
Асистент: мр Владимир Рисојевић**

**Бања Лука, Јуни 2007.**

## 1. Увод

У данашње вријеме које представља еру модерних информационих технологија, савремених технолошких и научних достигнућа, постоји потреба да се изврши интеграција свих информационих система у једну јединствену мрежу која ће служити као поуздан и брз извор информација. У ту сврху потребно је извршити дигитализацију већ постојећих база података као и пројектовати дигиталне системе који ће у будуће бити дио мреже и обављати функцију сензора за пријем визуелних или било којих других типова података. Један такав систем јесте систем за аутоматско препознавање регистарских таблица.

Систем за аутоматско препознавање регистарских таблица представља интелигентну опрему која је у могућности да препозна возило по његовој регистарској табlici као и његовог власника у реалном окружењу. Овакви системи би били корисни за регистрацију возила и њихових власника, као и пружање информација и помоћи другим системима у мрежи. Захваљујући овоме, развијене су многе технике и аутоматско препознавање таблица се данас користи у многим саобраћајним и сигурносним апликацијама, као што су паркинзи, приступ возила, контрола на граници, праћење украдених возила, итд.

На паркингу, регистарске табlice се користе за израчунавање трајања паркирања. Када возило уђе на приступној рампи, препознају се регистарске табlice и сачувају у базу. Када возило касније изађе са паркинга, табlice се поново препознавају и пореде са већ снимљеним у бази. Временска разлика се користи за израчунавање трајања паркинга.

Систем за аутоматско препознавање таблица се користи и за одобравање приступа возилима. На примјер, препознавањем таблица може се забранити или одобрити приступ возилу у одређено подручје.

Такође, у многим земљама се користи за препознавање отуђених аутомобила на границама или наплатним рампама, поредећи табlice возила са таблицама са тзв. «црне листе».

Систем се може искористити и за евидентирање саобраћајних прекршаја као и за усмјеравање возила заобилазним путевима у тренутцима великих саобраћајних гужви.

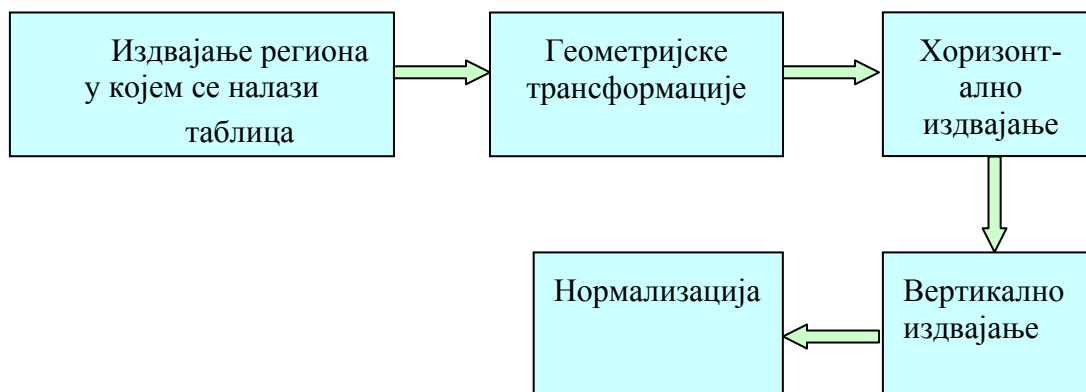
У овом раду смо се базирали на једном од сегмената у поступку аутоматског препознавања регистарских таблица, а то је: издвајање региона у којима се налази таблица са слике, нормализација величине табlice и уклањање дисторзије.

У самом старту располагали смо “avi“ fajlom, gdje smo napravili “capture“ за фрејмове који су нам се учинили најинтересантнији, водећи рачуна да имамо слике снимљене из разних углова (са перспективом) и накошене слике јер су нам ове најрепрезентативније.

Са слика аутомобила регионе на којима се налазе табlice смо издвајали полуаутоматски, тј. користили смо наредбу «crop» у MATLAB-у, где смо визуелном перцепцијом одредили регион а затим наредбом «crop» «исјекли» таблицу.

Затим су се вршиле геометријске трансформације: ротирање таблица, исправљање косих таблица у лијево и у десно, тзв. «deskewing» и нормализација величине таблице. Таква излазна слика је даље ишла на сегментацију.

Дакле, ток нашег рада могао би се представити сљедећим блок дијаграмом.



слика 1: Ток рада

## 2. Геометријске трансформације

Угао за који је таблица заротирана смо одредили кориштењем «Радон» трансформације. Прије тога требало ја одредити хоризонталне ивице слике ради лакшег проналажења угла ротације. Хоризонталне ивице слике смо одредили кориштењем собеловог градијентног филтра. Слика са издвојеним хоризонталним ивицама изгледа као на сљедећој слици:



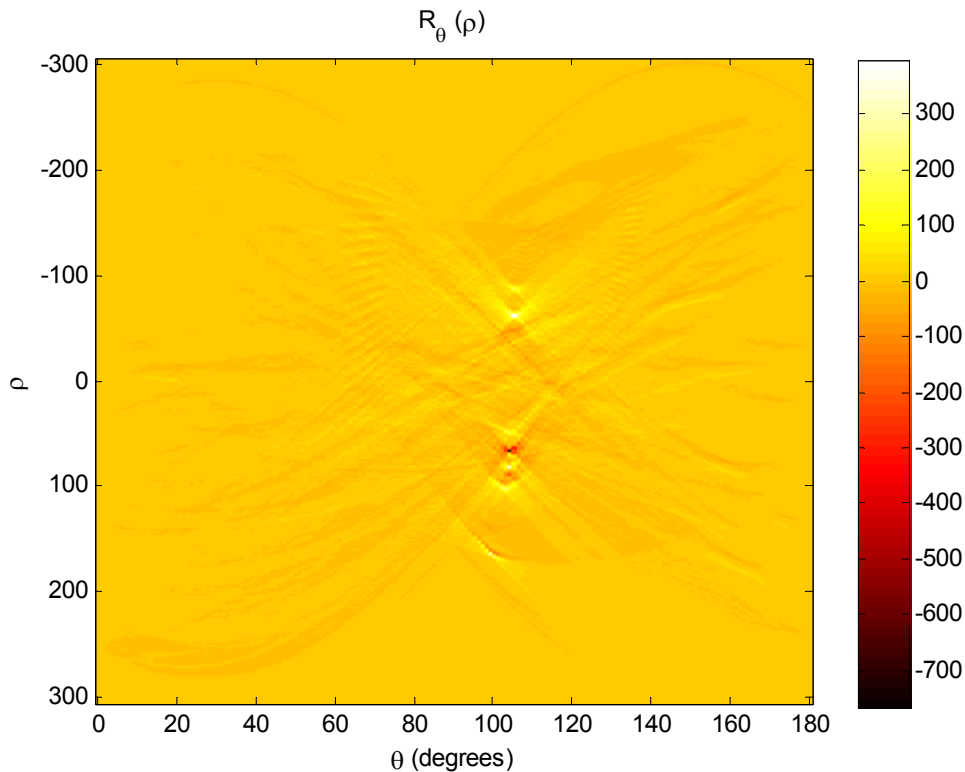
а.)



б.)

Слика 2: а.) Полазна сива слика таблице, б.) Хоризонталне ивице

Када се примјени радонова трансформација на слику хоризонталних ивица добије се дијаграм параметарског простора са координатама  $\rho$  и  $\theta$ , као на сљедећој слици:



Слика 3 : Параметарски простор Радон трансформације

Очитавањем максималне амплитуде (интензитета) са дијаграма параметарског простора, као и координата те тачке, добија се угао ротације као  $\varphi = 90^\circ - \theta$ .<sup>1</sup>

Сада је потребно на таблицу примјенити геометријску просторну трансформацију како би смо таблицу која је искривљена (схеар) за угао  $\varphi$  искривили за угао  $-\varphi$  и добили исправљену таблицу. За то смо искористили трансформациону матрицу  $A$  која изгледа овако:

$$A = \begin{bmatrix} 1 & -\tan(\varphi) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

---

<sup>1</sup> Треба напоменути да је овај угао у степенима, док функција  $\tan$  у Матлаб-у тражи вриједност у радијанима.

Ако у општу једначину  $b(x, y) = a(x', y') = a[f(x, y), g(x, y)]$ , гдје је  $a(\cdot)$  оригинална слика, а  $b(\cdot)$  резултујућа слика, ставимо  $f(x, y) = x - \tan(\varphi) \cdot y$  и  $g(x, y) = y$  добићемо трансформисану слику.

Трансформација у матричној форми изгледа овако:

$$\begin{bmatrix} f(x, y) \\ g(x, y) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -\tan(\varphi) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Све ово је имплементирано у MATLAB-у функцијама “maketform” и “imtransform” што смо и ми искористили. [5]

Слика након исправљања изгледа као на сљедећој слици:



а.)



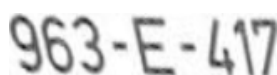
б.)

Слика 4: а.) Оригинална слика , б.) Слика после трансформације

Потребно је напоменути да угао израчунат помоћу Радон-ове трансформације не мора представљати тзв. *skew angle*, већ се може радити и о углу за који је слика накошена. У пракси је ситуација да слика има перспективу вјероватнија од ситуације да имамо накошену слику. Иначе, ми смо проблему приступили тако да смо креирали *deskewing* механизам на начин који је изнад објашњен. Ово за слике са већим накошењем може резултовати накошеним словима на издвојеној табlici, као на идућим сликама.



а.)



б.)

Слика 5: а.) Прије , б.) Послије издвајања табlice и урађеног *deskewinga*

Оваква ситуација, јасно, отежава сегментацију карактера методом вертикалне пројекције или је потпуно онемогућује.

За уклањање накошености слике може се на једноставан начин искористити функција *imrotate*, при чему је угао одређен на исти начин помоћу Радон-ове трансформације. Ми смо овај корак заобишли, јер се накошене слике цијелог аутомобила ипак рјеђе појављују, а с техничке стране тешко је уклопити корекцију накошености и перспективе да би то произвело задовољавајуће резултате.

Као још једна напомена, могуће је умјесто Радон-ове користити Хаф-ову трансформацију за одређивање одговарајућег угла.

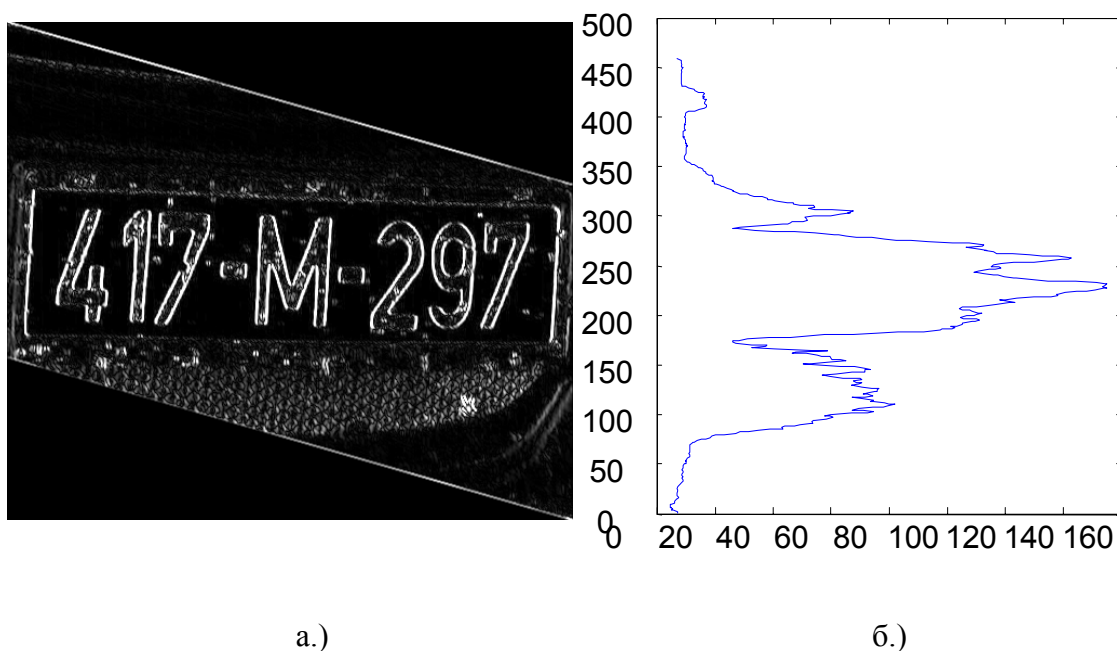
### 3. Хоризонтално издвајање таблице

Након геометријских просторних трансформација добијамо слику спремну за сегментацију региона у којем се налази таблица (број таблице).

Наравно, на поменутој таблици има још тога да се поправи што се тиче евентуалне појаве перспективе, или искривљења у другу страну, али због ограничења нашег курса као и чињенице да ће се у реалним условима настојати избјећи такав случај, у овом раду нећемо се бавити тим проблемом.

Дакле, издвајање таблице смо имплементирали помоћу алгорита хоризонталних и вертикалних пројекција хоризонталних и вертикалних ивица слике. Циљ је био препознати регију између броја таблице и оквира таблице. Тај дио таблице је бијеле боје и у идеалном случају на њему се непојављују никакве ивице.тј. натписи (објекти). Када се пронађу вертикалне ивице и изврши вертикална пројекција, тако што се сумирају све колоне слике вертикалних ивица

по формули  $p(y) = \sum_{j=0}^{w-1} a(j, y)$ , добије се ситуација као на сљедећој слици:



Слика 6: а.) Ветикалне ивице, б.) Вертикална пројекција

Примјећује се да на свим пројекцијама доминира дио таблице гдје има највисте ивица, што у ствари представља број таблице. Сада је било потребно одредити индексе гдје почиње и завршава тај доминантни (широки) пик. То смо



постигли постављањем прага којег смо хеуристички утврдили да важи за све слике. Тај праг износи 0.3 .Проналажењем диференција (првог извод) индекса вертикалне пројекције и налажењем максимума диференције добијамо ширину регије на којој сеналази број таблице.

Мапирањем почетка и краја те регије долазимо до тражених координата за хоризонтално исјецање таблице.Када се изврши хоризонтално исјецање добија се таблица као на следећој слици:



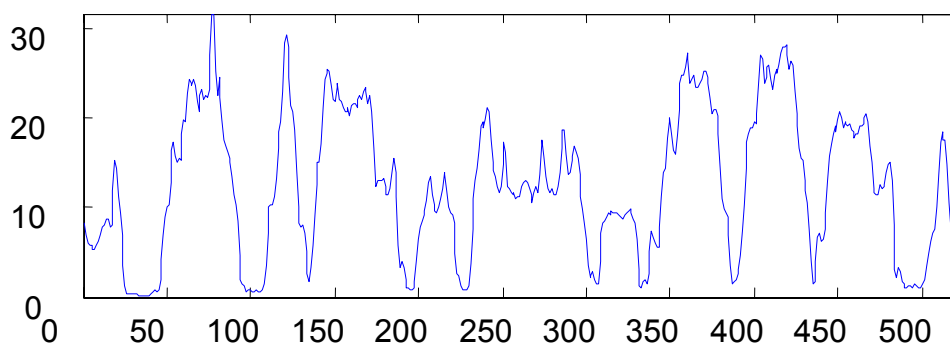
Слика 7 : Хоризонтално исјечена таблица

## 4. Вертикално издвајање таблице

Код вертикалног издвајања таблице имамо сличну ситуацију као и код хоризонталног. Прво се пронађу хоризонталне ивице, а затим се изврши хоризонтална пројекција по формули  $p(x) = \sum_{i=0}^{h-1} a(x, i)$ . Пројекција је приказана на следећој слици:



а.)



б.)

Слика 8 : а.) Хоризонталне ивице б.) хоризонтална пројекција

Сада је циљ пронаћи прве локалне минимуме са лијеве и десне стране пројекције, јер они представљају регију између броја таблице и преосталог оквира. У овом случају смо из практичних разлога ради лакшег проналажења прага инвертовали пројекцију по формули  $p'(x) = \max(p(x)) - p(x)$  и одредили праг чија је вриједност 0.8. Сада смо издвојили све индексе пројекције  $p'(x)$  за вриједност пројекције већу од задатог прага. Праг је и у овом случају одређен хеуристички да задовољи што већи број таблица. Диференцирањем пронађених индекса долазимо до ширина свих пикова у пројекцији.

Нас интересују само први и задњи пик пројекције  $p'(x)$ , односно пресеци тих пикова са задатим прагом, па налазећи индексе тих пресека долазимо до тражене ширине броја таблице.


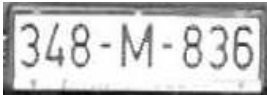
Коначан облик издвојене таблице дат је на следећој слици:





A large, bold, black and white image of the license plate number "417-M-297". The characters are thick and slightly irregular, suggesting they were extracted from a scan of a physical plate.

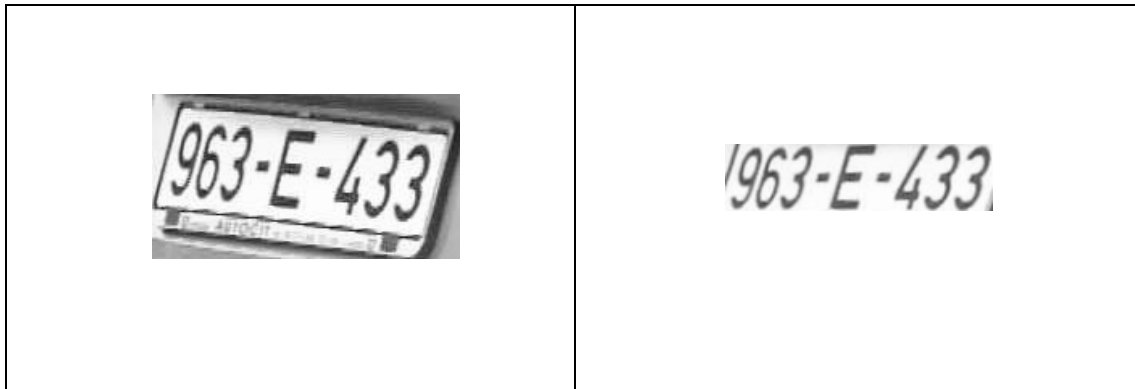
Слика 9 : Издвојена таблица

## 5. Експерименти

На следећим сликама дато је неколико примјера издвојених таблица овим алгоритмом који смо реализовали у MATLAB-у функцијом “`isjecanje.m`”:

	963-E-417
	348-M-836

	547-J-427
	025-K-724
	722-K-166
	980-E-359



Наравно, треба истаћи да алгоритам ради боље ако је аквизиција слике боља. У нашем случају су обрађене неке слике таблица које ће се у реалним околностима ријетко десити, јер на сваком мјесту гдје је постављена оваква опрема води се рачуна о брзини кретања возила, мјесту са ког се врши фотографисање, положај возила приликом фотографисања и слично. Свим овим предрадњама се доприноси лакшем и прецизнијем издвајању таблица.

## 6. Закључак

Прво што би се могло урадити на побољшању овога алгоритма је прављење потпуне аутоматизације исјецања, а то подразумијева аутоматизацију издвајања региона на којем се налази таблица са слике аутомобила, или неке веће регије око таблице нпр. браника. Ово је тежак посао, али уз придавање пажње правилној аквизицији слике, не и немогућ.

Самим тим поступком и наш алгоритам ће радити боље тако што ће се прагови лакше налазити и биће флексибилнији за све слике.

Као што је већ наведено у уводу, сврха и значај ових система је вишеструк, па и њихова комерцијална вриједност није занемарљива. Оваквим системима у многоме би се олакшао рад и контрола у многим службама, као и безбједност и сигурност свих аутомобила у промету.

Дубљим истраживањем, већом теоријском подлогом као и практичном хеуристичком, алгоритам би се могао усавршити доприносећи ка још једном кораку у процесу аутоматизације и интеграције система и учења машина процесима које човјек обавља крајње једноставно!

## 7. Литература:

- [1] Ondrej Martinsky, Algorithmic and Mathematical Principles of Automatic Number Plate Recognition Systems, B.Sc. Thesis, Brno University of Technology, Brno 2007.
- [2] Haim Cohen, Gadi Bergman, Johanan Erez, Car License Plate Recognition, Project in computer vision, The Vision and Image Science Laboratory, Faculty of Electrical Engineering Technion, Israel
- [3] Bar-Hen Ron, Johanan Erez, A Real-time vehicle License Plate Recognition (LPR) System, The Vision and Image Science Laboratory, Faculty of Electrical Engineering Technion, Israel, Winter 2002
- [4] Gonzalez R., Woods R.: **Digital Image Processing**, Prentice Hall, Upper Saddle River, New Jersey, 2002
- [5] MATLAB Help

## Прилог:

Функција “ isjecanje.m ” која реализује описани алгоритам у MATLAB-у:

```
% Semestralni rad iz Digitalne Obrade Slike
% Izdvajanje regiona u kojima se nalazi tablica sa slike, normalizacija
% velicine tablice i uklanjanje distorzije.
% Student:
% Igor Maric
% Dragan Gracanin
% Supervizor:
% mr Vladimir Risojevic

% Help: Ulaz u funkciju je slika u boji, a izlaz siva slika izdvojene
% tablice!

function slika=isjecanje(a);
b=rgb2gray(a);
```

```

b=im2double(b);
% figure,imshow(b,[],title('siva tablica'))

%sobel(izdvajanje ivica)
hsx=fspecial('sobel');
bsx=imfilter(b,hsx,'replicate');
% figure,imshow(bsx),title('Horizontalne ivice');
hsy=hsx';
bsy=abs(imfilter(b,hsy,'replicate'));
% figure,imshow(bsy),title('Vertikalne ivice');
bs=sqrt(bsx.^2+bsy.^2);
% figure,imshow(bs),title('Ivice')

%rotacija i transformacija
theta = 0:180;
[R,xp] = radon(bsx,theta);
% figure, imagesc(theta, xp, R); colormap(hot);
% xlabel('\theta (degrees)'); ylabel('\rho');
% title('R_{\theta} (\rho)');
% colorbar
[Y,I]=max(max(R));
teta=90-I; % Ugao ya koji je tablica iskrivljena
T=[1 -tan(teta*pi/180) 0;0 1 0;0 0 1];
tform=maketform('affine',T);
b1=imtransform(b,tform); %Ispravljanje tablice
% figure,imshow(b1,[],title('Ispravljena tablica'));

%Vertikalna projekcija
bsy=abs(imfilter(b1,hsy,'replicate'));
N=size(bsy,2);
M=size(bsy,1);
% figure, imshow(bsy),truesize;
% title('Vertikalne ivice');
py=sum(bsy, 2);
ii = find(py <= 0.3*max(py));
dii = diff(ii);
[m, kk] = max(dii);
ii1 = ii(kk);
ii2 = ii(kk+1);
b1 = b1(ii1:ii2, :); %Tablica isjecena po horizontali
% figure, imshow(b1,[])

%Horizontalna projekcija
bsx = abs(imfilter(b1, hsx, 'replicate'));
% figure,imshow(bsx),title('Horizontalne ivice');
px = sum(bsx,1);
% figure,plot(px),title('Horizontalna projekcija');
N=length(px);
px1 = max(px) - px;
% figure, plot(px1), title('Horizontalna projekcija(inverzna)');
jj = find(px1 >= 0.8*max(px1));
djj=diff(jj);
kk=find(djj~=1);
jj1=jj(min(kk));
jj2=jj(max(kk)+1);
slika=b1(:, jj1:jj2);
slika=imresize(slika,[40,160],'bilinear');
% figure, imshow(slika),title('Tablica');

```