УДК: 519.711; 621.31.15

Алгоритмы сглаживания и утоньшения для грузинских печатных символов

Теа Тодуа
Грузинский Технический Университет, ул. М. Костава 77, 0175, Тбилиси, Грузия
Факультет Информатики и Систем Управления, Ассоциированный Профессор

Аннотация

Обсуждены проблемы распознавания грузинских печатных символов. Предложены алгоритмы сглаживания и утоньшения для решения этих проблем: процедура сглаживания дает возможность сокращения градаций изображений на краях растра, утоньшения идеально сокращает данные, сохраняет значительные характеристики символа, устраняет существующий шум без внесения новых, собственных искажений.

Ключевые слова: *распознавания, сглаживания, утоньшения*

1. Введение

Задачей автоматического распознавания текстов является перевод напечатанных на бумаге символов в цифровые данные, которые потом могут быть легко обработаны текстовым редактором. Эта задача является довольно трудной, поскольку каждый печатный символ можеть быть представлен несколькими способами за счет того, что существует множество различных шрифтов и стилей.

Исследовательской деятельностью в области распознавания печатных и рукописных текстов занимаются во всем мире уже несколько десятилетий подряд. За это время удалось создать довольно эффективные системы распознавания для многих языков мира. Однако, открытой остается проблема распознавания грузинских печатных символов.

В результате сканирования символов получаются реализации различных размерностей. Для выбора оптимальной размерности были предусмотрены следующие требования: размерность растра должно быть настолько большим, чтобы избежать нежелательного взаимного сближения линий и дуг в изображении, а также случаев потери специфического нароста – зубца. Изображение должно быть достаточно малым, чтобы не имело место представление символа в искаженном и непропорциональном виде.

С учетом вышеупомянутого, в результате экспериментального анализа учебной выборки, полученной с помощью сканирования, было установлено, что растр с размерностью 32X27 является достаточно адекватним для представления символов грузинского алфавита.

Следует отметить, что выбор размерности не в полной мере устраняет ложные соединения или искажения другого типа, становится необходимым выполнение процесса препарирования.

2. Сглаживание и искоренение разрывов

Процедура сглаживания дает возможность сокращения градаций изображений на краях растра. Кроме того, возможно избежать разрыви в структуре изображения вызванного технологическими процессами печати и сканирования.

Процедура сглаживания разделяется на 3 этапа: 1) сглаживание на краях; 2) глубокое сглаживание с краев; 3) внутреннее сглаживание.

Первоначальное изображение дано в виде бинарной матрицы, в следствие чего имеем

$$X = \left\| x_{ij} \right\|; \quad \forall x_{ij} \in \{0,1\}; \quad i = \overline{0;I}; \quad j = \overline{0;J}; \quad I \in N^{+}, \quad J \in N^{+}$$

Процедура сглаживания на краях, для выбранного измерения растра 32X27 осуществляется для групп пикселей, находящихся на краях растра:

1) когда $i = 0$, тогда $x_{ij} = 1$ и $x_{i+1,j} = 1$,

если а) $x_{ij} = 0 \cap x_{i+1,j} = 0 \cap x_{i+2,j} = 1 \cap x_{i,j-1} = 1 \cap x_{i+1,j-1} = 1$;

б) $x_{ij} = 0 \cap x_{i+1,j} = 1 \cap x_{i+2,j} = 1$; в) $x_{ij} = 0 \cap x_{i+1,j} = 1 \cap x_{i+2,j} = 0$;

2) когда $i = I - 1$, тогда $x_{ij} = 1$ и $x_{i-1,j} = 1$,

если а) $x_{ij} = 0 \cap x_{i-1,j} = 0 \cap x_{i-2,j} = 1$;

б) $x_{ij} = 0 \cap x_{i-1,j} = 1 \cap x_{i-2,j} = 1$; в) $x_{ij} = 0 \cap x_{i-1,j} = 1 \cap x_{i-2,j} = 0$;

3) когда $j = 0$, тогда $x_{ij} = 1$ и $x_{i,j+1} = 1$,

если а) $x_{ij} = 0 \cap x_{i,j+1} = 0 \cap x_{i,j+2} = 1$;

б) $x_{ij} = 0 \cap x_{i,j+1} = 1 \cap x_{i,j+2} = 1$; в) $x_{ij} = 0 \cap x_{i,j+1} = 1 \cap x_{i,j+2} = 0$;

4) когда $j = J - 1$, тогда $x_{ij} = 1$ и $x_{i,j-1} = 1$,

если а) $x_{ij} = 0 \cap x_{i,j-1} = 0 \cap x_{i,j-2} = 1$;

б) $x_{ij} = 0 \cap x_{i,j-1} = 1 \cap x_{i,j-2} = 1$; в) $x_{ij} = 0 \cap x_{i,j-1} = 1 \cap x_{i,j-2} = 0$;

Процедура глубокого сглаживания с краев подразумевает замену фоновых структур на краях растра на структуры изображения, в том случае, когда фоновые структуры ограничены структурами изображения поблизости от краев растра. Во время внутреннего сглаживания для определенной части растра в переходе 101 или 010 фоновая структура в первом случае или структура изображения во втором случае меняется на противоположную структуру.

000111110000011000001111000 111111111111011111111111111 111111111111111111111111111
011111111111011111111111100 111111111111011111111111111 111111111111111111111111111
011111111111011111110111100 111111111111011111110111111 111111111111111111111111111
011110011111001111110111110 111110011111001111110111111 111110011111001111111111111
111110011111001111110111110 111110011111001111110111111 111110011111001111111111111
111110011110000111100011110 111110011110000111100011111 111110011110000111100011111
111110011110000111100011111 111110011110000111100011111 111110011110000111100011111
111110011110000111100011111 111110011110000111100011111 111110011110000111100011111
111110011110000111100011111 111110011110000111100011111 111110011110000111100011111
111110011110000011100011111 111110011110000011100011111 111110011110000011100011111
111110011110000011100011111 111110011110000011100011111 111110011110000011100011111
111110000000000000000011110 111110000000000000000011111 111110000000000000000011111
111110000000000000000011100 111110000000000000000011111 111110000000000000000011111
111111100000000000000011100 111111100000000000000011111 111111100000000000000011111
011111100000000000000010000 111111100000000000000010000 111111100000000000000000000
011111111100000000000010000 111111111100000000000010000 111111111100000000000000000
001111111111100000000000000 111111111111100000000000000 111111111111100000000000000
001111111111100000000000000 111111111111100000000000000 111111111111100000000000000
011111111111111100000000000 111111111111111100000000000 111111111111111100000000000
011111111111111100000000000 111111111111111100000000000 111111111111111100000000000
001111111111111111110000000 111111111111111111110000000 111111111111111111110000000
001111111111111111111100000 111111111111111111111100000 111111111111111111111100000
000000000001111111111100000 000000000001111111111100000 000000000001111111111100000
000000000000000011111111000 000000000000000011111111000 000000000000000011111111000
000000000000000011111111100 000000000000000011111111111 000000000000000011111111111
000000000000000000111111100 000000000000000000111111111 000000000000000000111111111
000000000000000000011111110 000000000000000000011111111 000000000000000000011111111
000000000000000000011111110 000000000000000000011111111 000000000000000000011111111
000000000000000000000111110 000000000000000000000111111 000000000000000000000111111
000000000000000000000011110 000000000000000000000011111 000000000000000000000011111
000000000000000000000011110 000000000000000000000011111 000000000000000000000011111
000000000000000000000001110 000000000000000000000011111 000000000000000000000011111

Рис. 1. Первоначальное изображение символа «ლ» и результаты, полученные путем искоренение разрывов и сглаживания

Непосредственно с процессом сглаживания связана процедура искоренения разрывов. Разрыв кардинально меняет реальное описание символа. В таком случае один символ анализируется как несколько отдельных символов и этим значительно понижает возможность его правильного распознавания.

Процедура искоренения разрывов в первоначальном изображении осуществляется слудующим образом:

а) если $x_{ij} = 0 \bigcap x_{i,j-1} = 1 \bigcap x_{i,j+1} = 1$, тогда $x_{ij} = 1$;

б) если $x_{ij} = 0 \bigcap x_{i,j-1} = 1 \bigcap x_{i,j+1} = 0 \bigcap x_{i,j+2} = 1$, тогда $x_{ij} = 1$, $x_{i,j+1} = 1$

На рис. 1. представлены первоначальное изображение символа «ლ» и результаты, полученные с помощью процедур сглаживания и искоренения разрывов. На рисунке видно, как после осуществления процедуры глубокого сглаживания, устраняются т. н. наросты в изображении.

3. Утоньшение

Для осуществления правильного утоньшения изображения стало необходимым выявление характерного элемента грузинского алфавита – зубца (например, «ვ», «ვ», «ვ», «ვ») и его сохранение при утоньшении.

В статье предложени два алгоритма различного типа для детекции зубцов и утоньшения изображении: 1. алгоритм утоньшения и детекции зубцов с применением т. н. матрицы переходов; 2. алгоритм утоньшения и детекции зубцов с применением метода контактов между строками.

Использованная в первом алгоритме матрица переходов является компактным описанием символа. Переходом называется замена структур изображения на фоновые структуры в каждой строке изображения. Процедура получения матриц переходов из первоначальной $X = \left\| x_{ij} \right\|$ матрицы описывается таким образом: $i-$я строка первоначальной матрицы $X_i = x_{i1}, x_{i2}, x_{i3} ..., x_{ij}, ..., x_{iJ}$ содержит в себе непрерывную последовательность нулей и единиц $X_i \supset \left(XR_i^1, XR_i^2 ..., XR_i^{k_i} ..., XR_I^{K_i} \right)$, где $\forall XR_i^{k_i} = x_i^{k_i}[h_{k_i}]; h_{k_i} = \overline{j_{k_i}, J_{k_i}}$; $K \in N^+$. h_{k_i} представляет последовательность элементов в $XR_i^{k_i}$ подмножестве.

Количество переходов для i-й строки $K_i = \text{Card}\left\{ XR_i^{k_i} \right\}$, если $\forall x_i^{k_i}[h_{k_i}] = 0 \Rightarrow GAD_i^{k_i} = 0$, если $\forall x_i^{k_i}[h_{k_i}] = 1 \Rightarrow GAD_i^{k_i} = 1$, где $GAD_i^{k_i}$ представляет k_i переход в i-й строке. $GD = \max\left\{ K_i \right\}$, $GD \le GANR$, где GD является максимальное количество переходов, полученное в результате вычислений, GANR - размерность столбцов матрицы переходов. Для получения матрицы переходов необходим осуществление процедур заполнения:

если $GAD_i^{K_i} = 0 \Rightarrow GAD_i^{k_i+f_i} = 0$; если $GAD_i^{K_i} = 1 \Rightarrow GAD_i^{k_i+f_i} = 1$; $K_i + f_i = GANR$; $f \in N^+$

Окончательно, получается матрица $GAD = \left\| GAD_i^{k_i+f_i} \right\|$, размерность которого $Ix(K_i + f_i)$.

С учетом характерных своиств грузинских печатных символов, поиск зубцов в символе осуществляется в экспериментально предопределенном зоне растра и в этом зоне определяются условия существования (несуществования) зубцов. В первую очередь определяется начало зубца. Зафиксирование начало зубца еще не означает существование реального зубца. Становится необходимым определение координат конца зубца. Условия существования конца зубца: $x_{ij} = 1 \bigcap x_{i,j-1} = 0 \bigcap x_{i+1,j-1} = 0 \bigcap x_{i+1,j} = 0$

Проблемным вопросом является исключение т.н. ложных зубцов. Для выявления ложных зубцов определяется количество непреривных последовательностей единиц и нулей в начальном и в последнем строке зубца.

В символе найденный зубец является ложным, когда выполняются условия:

а) $CT2 > 0,\ T_{21} = 2,\ T_{20} \le 2;$

б) $CT1 > 0,\ CB1 > 0,\ T_{11} = 2,\ T_{10} = 2,\ B_{11} = 2,\ B_{10} = 2;$

в) $CT1 > 0, CB1 > 0, T_{11} = 1, T_{10} = 1, B_{11} = 2, B_{10} = 1, B_{1z} = 2, B_{0z} = 2$

где $CT1$обозначает строку, в котором начинается первый зубец, $CB1$- отмечает строку, в котором заканчивается первый зубец. $CT2$ обозначает ту же величину для второго зубца, в случае его существования в символе. T_{11} и T_{10} являются количеством непреривных последовательностей нулей и единиц в начальном строке первого зубца. B_{11} и B_{10} являются количеством непреривных последовательностей нулей и единиц в последнем строке первого зубца. B_{1z} и B_{0z} являются количеством непреривных последовательностей нулей и единиц в предпоследнем строке первого зубца. T_{21} и T_{20} являются количеством непреривных последовательностей нулей и единиц в начальном строке второго зубца.

Выполнение вышеупомянутых процедур исключает нахождение ложных зубцов в символе. После выявления зубцов осуществляется процедура утоньшения. Во время утоньшении, как и во время детекции зубцов, особенное внимание уделяется матрице переходов. Основной принцип состоит в том, что при изменении переходов в двух последующих строках в матрице переходов, вышеотмеченные две строки не утоньшаются. Это обеспечивает устранение разрывов, которые могуть возникнуть во время утоньшения в точках соединения.

00000000011111000000000000
00000011111111000000000000
00000111111111000000000000
00000000011111111111111100
00000000011111111111111111
00000000000111111111111111
00000000000000000111111111
00000000000000011111111111
00000000000111111111111111
00000000000111111111111100
00000000000011111111111111
00000000000011111111111111
00000000000000000111111111
00000000000000000111111111
00000000000000011111111111
00000000000000011111111111
00000000000111111111111100
00000000000111111111111100
00000000000011111111111100
00000000000000000111111111
00111110000000000111111111
00111110000000000111111111
00111111100000000111111111
11111111100000000111111111
11111111100000000111111111
11111111100000000111111111
11111111100000000111111100
11111111111100000111111100
00111111111100000111111100
00111111111111111111111100
00111111111111111111111100
00000000011111111000000000

00000000001111000000000000
00000000001111000000000000
00000000001111000000000000
00000000001111111111111111
00000000011111111111111111
00000000000000000000001111
00000000000000000000001111
00000000000000011111111111
00000000000111111111111111
00000000000111111111111111
00000000000011111111111111
00000000000011111111111111
00000000000000000000001111
00000000000000000000001111
00000000000000011111111111
00000000000000011111111111
00000000000111111111111111
00000000000111111111111111
00000000000011111111111111
00000000000000000000001111
11110000000000000000001111
11110000000000000000001111
11110000000000000000001111
11110000000000000000001111
11110000000000000000001111
11110000000000000000001111
11110000000000000000001111
11111111111111111111111111
11111111111111111111111111
11111111111111111111111111

Рис. 2. Символ «3» и результат его утоньшения

Процедура утоньшения не осуществляется для $i-$й и $(i-1)$-й строк, при испольнении последующих условии:

а) $GAD_i^1=1 \bigcap GAD_i^2=0 \bigcap GAD_i^3=1 \bigcap GAD_i^4=1 \bigcap GAD_{i-1}^1=0 \bigcap GAD_{i-1}^2=1 \bigcap GAD_{i-1}^3=1$

б) $GAD_i{}^1=0 \bigcap GAD_i{}^2=1 \bigcap GAD_i{}^3=0 \bigcap GAD_i{}^4=1 \bigcap GAD_{i-1}{}^1=0 \bigcap GAD_{i-1}{}^2=1 \bigcap GAD_{i-1}{}^3=1;$

в) $GAD_i{}^1=1 \bigcap GAD_{i-1}{}^1=0;$ г) $GAD_i{}^K=0 \bigcap GAD_{i-1}^{K_{i-1}}=1;$

На рис.2 показано Символ «3» и результат его утоньшения. При утоньшении изображения осуществляется сокращение количество пикселов первоначальной изображении до определенного количества. В обсуждаемом алгоритме его значение принимается равним 4. Утоньшаются пиксели изображения в строках, где не существует зубец и где не происходит такое резкое изменение переходов, при котором утоньшение может вызвать разрыв в изображении.

Второй алгоритм утоньшения и детекция зубцов обосновывается на принципе определения связности между $i, i-1$ и $i+1$ строкамы, что обеспечивает получение утоньшенной изображении из первоначальной изображении без разрывов. В символе не существует зубец, если:

а) $GAD_I^1=1 \bigcap GAD_I^2=0 \bigcap GAD_I^3=1;$

б) $j_{K_i} < 0.15J \bigcap J_{K_i} > 0.8J$, где j_{k_i} представляет координату первую с лева единицы последнего перехода i-й строки, J_{K_i} - представляет координату первую с права единицы последнего перехода i-й строки;

в) $j_{K_i} - j_{K_{i-1}} \geq 0$

Необходимая условия для существования начало зубца: $j_{K_i} - j_{K_{i-1}} < 0$.

При нахождении одного зубца, зубец ложный, если

а) $CT1 >= 0{,}6I$ или $CB1 >= 0.6I$; б) $CT1 >= 0.25I$; $CB1 < 0.3I$

При нахождении два зубца, оба являются ложным, если $CT2 < 0.25I$.

Утоньшение осуществляется для i-й строки при условии, что существует контакт предидущей строки с последующим строком.

Описание процесса определения контакта между строками и утоньшения формализуется следующим образом:

$KONT_j = KONZ_j \bigcap KONQ_j$, где $KONZ_j$ обозначает контакт со предидушей строкой.

$KONQ_j -$ обозначает контакт со следующей строкой.

$KONZ_j = 1$, если $XR_i^{k_i} \bigcap XR_{i-1,j}^{k_{i-1}} = 1 \Rightarrow$ наидется хотя бы один $x_i^{k_i}[h_{k_i}] \bigcap x_{i-1}^{k_{i-1}}[h_{k_{i-1}}] = 1$, где $x_i^{k_i}[h_{k_i}] \in XR_i^{k_i}$

$KONQ_j = 1$, если $XR_i^{k_i} \bigcap XR_{i+1}^{k_{i+1}} = 1 \Rightarrow$ наидется хотя бы один $x_i^{k_i}[h_{k_i}] \bigcap x_{i+1}^{k_{i+1}}[h_{k_{i+1}}] = 1$, где $x_i^{k_i}[h_{k_i}] \in XR_i^{k_i}$.

При выполнении условии $KONZ_j \bigcap KONQ_j = 1$ остается один пиксель, а другие стираются, иными словами, происходит утоньшение до одного пиксела.

Для правильного осуществления процесса утоньшения большое значение имеет определение направления утоньшения. Если изображения начинается с края, при утоньшении выбирается направления от центра до краев растра.

Применены следующие ограничения:

а) длинные линий не утоньшаются. Длинным называется линия, длина которого является больше 60% горизонтальной размерности растра.

в) Если изображение не начинается с края, тогда при выборе направлении утоньшения принимается во внимание количество последовательность непреривных единиц в данной строке.

На рис. 3. показан грузинский символ «ვ» и результат его утоньшения до одного пикселя. На рисунке виден зубец, сохраненный в утоньшенном символе.

Вышеописанные алгоритмы утоньшения реализованы на языке программирования С++. Первый алгоритм утоньшения, в котором утоньшения происходит с помощью матриц переходов, разработан для метода мини и макси портретов, который используется для последующего распознавания изображении.

Мини и Макси портреты получаются из сканированных символов путьем процедуры суперпозиций:

$MAX_i = \bigcup_m \{x_{ni}^{mi}\}$; $MIN_i = \bigcap_m \{x_{ni}^{mi}\}$; $\forall x_{ni}^{mi} = \{0,1\}$; $n = \overline{1,N}$, $i = \overline{1,I}$, где $N = H * V$ - размерность пространства признаков, множество $X_i = \{x_{ni}\}$ является реализацией учебной выборки образа A_i; $x_n - n-$ый признак; $m_i = Card\{X_i\}$, $H -$размерность по горизонтали, $V -$по вертикали, $MAX_i -$макси портрет, а $MIN_i -$мини портрет того же образа, $I = Card\{A\}$.

```
00000000000001111111000000000   11111111111111111111111111111
00000011111111111111111110000   10000000000000000000000000001
00000011111111111111111110000   10000000000000000000000000001
00001111111100011111111111100   10000000000000000000000000001
00111111111000111111111111100   10000000000000000000000000001
00111111111000001111111111100   10000000000000000000000000001
00111111111000000111111111111   10000000000000000000000000001
00111111111000000111111111111   10000000000000000000000000001
00111111111000000111111111111   10000000000000000000000000001
00111111111000000111111111111   10000000000000000000000000001
00111111111000001111111111100   10000000000000001111111111111
00111111111000111111111111100   10000000000000011111111111111
00000000000000111111111111100   00000000000000011111111111111
00000000000000111111111110000   00000000000000011111110000000
00000000000000111111111111100   00000000000000000000011111111
00000000000000000111111111100   00000000000000000000000000001
00000000000000000111111111111   00000000000000000000000000001
00000000000000000111111111111   00000000000000000000000000001
00000000000000000111111111111   00000000000000000000000000001
00111111100000000111111111111   10000000000000000000000000001
00111111100000000111111111111   10000000000000000000000000001
00111111100000000111111111111   10000000000000000000000000001
11111111100000000111111111111   10000000000000000000000000001
11111111100000000111111111111   10000000000000000000000000001
11111111100000000111111111111   10000000000000000000000000001
11111111100000000111111111100   10000000000000000000000000001
11111111100000000111111111100   10000000000000000000000000001
11111111111000000111111111100   10000000000000000000000000001
00111111111000000111111111100   10000000000000000000000000001
00111111111111011111111110000   10000000000000000000000000001
00111111111111111111111110000   10000000000000000000000000001
00001111111111111111111000000   11111111111111111111111111111
```

Рис. 3. символ «ვ» и результат его утоньшения до одного пикселя

Если градации реализации некоторого образа большие, тогда эталонные описания макси портрета можеть охватывать целый растр или его значительный часть. Для избежания таких ситуации стало необходимым разработка алгоритма утоньшения. Поскольку в первом алгоритме разрыви в структуре изображений устранены с учетом факта изменения переходов в строках матрице переходов, полученная изображения в результате осуществления первого алгоритма является сравнительно толстым, чем результат полученный из второго алгоритма.

Следует отметить, что это не является недостатком первого алгоритма, так как при использовании метода мини и макси портретов предпочтительно толстые изображения. Второй алгоритм решает сложную и проблемную задачу распознавания – утоньшения до толщины одного пикселя, также решено проблема утоньшения диагональных элементов до одного пикселя без разрывов. Изображение с толщиной одного пикселя не годна для распознавания изображении с помощью метода мини и макси портретов – получается искаженный мини портрет или возможно его обнуление для всего растра. Второй алгоритм утоньшения разработан для распознавания изображении синтаксическим методом, так как после реализации этого алгоритма получается максимально утоньшенный символ, без всяких лишных структур.

4. Заключение

Работа обсужденных алгоритмов экспериментально проверено для несколько десятков грузинских шрифтов. Оны осуществляют сглаживание и утоньшение изображений без разрывов и без внесения новых искажении. Известные алгоритмы утоньшения в большей части представляют себя алгоритмы многократного применения, иными словами в процессе утоньшении происходит проход растра несколько раз. Выше обсужденный первый алгоритм утоньшения, который использует матрицу переходов, проходит растр дважды (первый проход – составление матрицу переходов, второй – процедура утоньшения), алгоритм утоньшения с помощью контактов между строками проходит растр только один раз. По проведенным экспериментам установлена, что после осуществления процессов препарирования улучшилась надежность распознавания грузинских печатных символов.

Литература:

1. Todua T. Data preprocessing for recognition of printed texts. Georgian Electronic Scientific Journal: "Computer Science and Telecommunications". http://gesj.internet-academy.org.ge/gesj_articles/1456.pdf; #3 (17), 2008;
2. Verulava O., Iremadze I., Todua T. Preparation of pattern realizations by neighbour pixel method. Georgian Technical University. Proceedings of the International Scientific Conference "Information Technologies 2008". Tbilisi, 2008;
3. Verulava O., Todua T., Gvichiani T., Zhvania T. About one Algorithm of stylization of Georgian Printed Symbols. Georgian Elecronic Scientific Journal: "Computer Science and Telecommunications". http://gesj.internet-academy.org.ge/gesj_articles/1292.pdf; #4(11), 2006;
4. Todua T. Preparation of Stilized Image. Georgian Academy of Sciences. A. Eliashvili Institute of Control Systems. Proceedings of The International Scientific Conference "Problems of Control And Power Engineering". Tbilisi, 2004;
5. Todua T. Computer Preprocessing of Georgian Printed Symbols. Georgian Electronic Scientific Journal „Computer Sciences and Technologies". http://gesj.internet-academy.org.ge. #1, 2004;
6. Lam L., Lee S.-W., Suen C. Thinning methodologies - a comprehensive survey. IEEE Tansactions on Pattern Analysis and Machine Intelligence, Vol.14, No 9, 1992, pp.869-885.
7. Бутаков Е.А., В. И. Островский, И. П. Фадеев. Обработка Изображении на ЭВМ. Москва, "радио и связь", 1987

Статья получена: 2009-04-02