Предмет: Разпознаване на образи

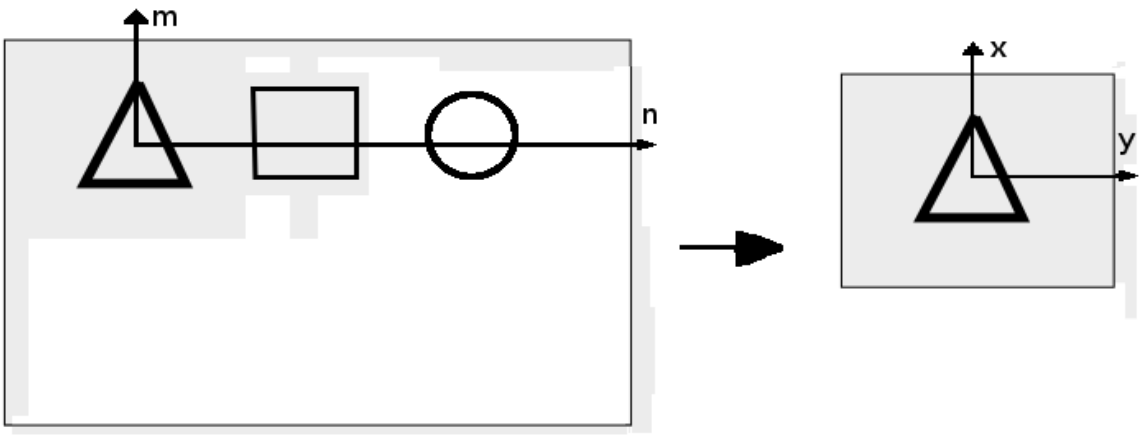
Преподавател: доц. Димо Т. Димов

Изготвил: Георги Тотев, фак. № М-23672, КГ

Дата: юли 2012

12. Детерминирани методи за разпознаване - методи за точно и еластично сравняване. Взаимна корелация между образ и еталон. Бързо изчисление по Фурие

Ако разполагаме с изображение, което искаме да сравним с множество от еталонни изображения, трябва да се определи критерий за подобие между изображенията. Разликата между изображенията се дефинира като $\sum_{x,y} |I(x,y) - E(x,y)|$, където I(x,y) е интензитетът на пиксела с координати x и y от изображението, а E(x,y) е интензитетът на пиксел със същите координати от еталона. Еталонът, с който има минимална разлика, ще има максимално съвпадение. Горната формула може да бъде подобрена, като се отчете, че еталоните са в собствена координатна система. Така формулата добива вида: $\sum_{x,y} |E(x,y) - I(x+n, y+m)|$, където m и n е отместването на еталона спрямо изображението по Ox и Oy.



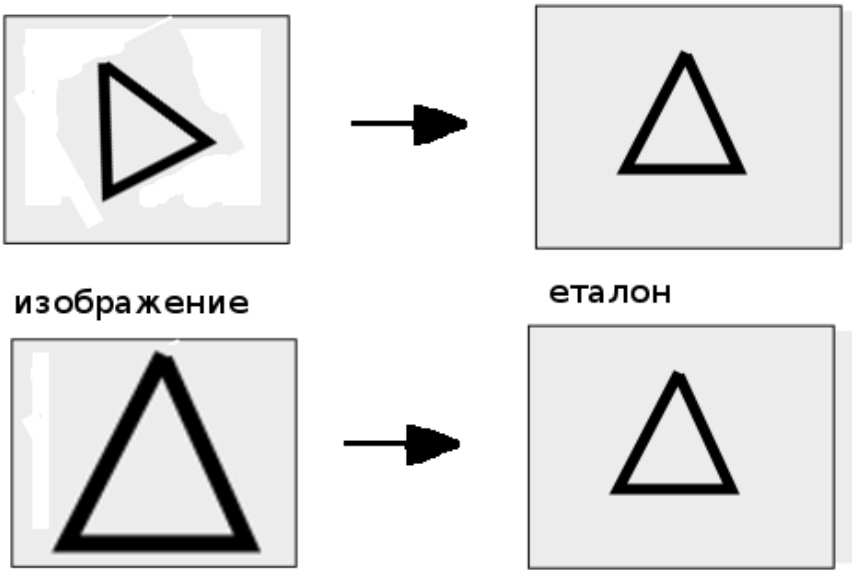
Следователно за намиране на най-доброто съвпадение с еталон трябва да търсим минимума на горната формула.

$$\sum_{x,y}(E(x,y)-I(x+n,y+m))^2=\sum_{x,y}E^2(x,y)+\sum_{x,y}I^2(x+n,y+m)-2\sum_{x,y}E(x,y)I(x+n,y+m)\rightarrow min$$

Първите два члена са постоянни за дадени изображение и еталон и съществено влияние оказва само третия, който е корелация между изображението и еталона. Следователно за намиране на съвпадение трябва да търсим максимума на корелацията: $\sum_{x,y}E(x,y)I(x+n,y+m)\rightarrow max$ Дотук този подход има съществен недостатък: така изразена корелацията има изчислителна сложност $O(N^4)$, което е неприложимо за практически цели. Решението е да се използва бързо преобразование на Фурие.

Аналогично на Теоремата за конволюцията се доказва теоремата за корелацията, която гласи: „$\mathcal{F}\{f \star g\} = (\mathcal{F}\{f\})^* \cdot \mathcal{F}\{g\}$,където f и g са функции, със * е отбелязано комплексно спрегнатото, а$\mathcal{F}$е преобразование на Фурие. Следователно трябва да се минимизира $\mathcal{F}^{-1}\{(\mathcal{F}\{I\})^* \cdot \mathcal{F}\{E\}\}$., което смъква сложността, тъй като сложността на бързата Фурие трансформация е само O(N*log N), като трябва да изпълним изчислението за всички еталони.

Дотук разгледания подход беше директно сравнение. За него приемаме идеализирани условия, при които изображението има същия мащаб като еталона и не е под наклон спрямо него. В реалните случай тези условия не винаги ще са изпълнени. За да преодолеем това изпълняваме различни геометрични трансформации върху еталона: мащабиране и ротация. Когато използваме геометрични трансформации преди пресмятането на корелацията вече говорим за еластично сравнение.



Бързи трансформации на Фурие.

Най-често използваният алгоритъм за Бърза трансформация на Фурие е описан в статия от 1965г. от Cooley и Tukey. Алгоритъмът има единственото изискване размерът на входните данни да е степен на 2, като някои модификации на алгоритъма преодоляват и това изискване.

Дискретният ред на Фурие има следния вид:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk},$$

Ако разделим редът на 2 и пресметнем отделно четните и нечетните входни компоненти, и после ги комбинираме ще можем да пресметнем целия ред. Формулата добива вида:

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N}(2m)k} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k}.$$

където 2m са четните индекси и 2m + 1 са нечетните. Вадим общ множител пред сумата и получаваме:

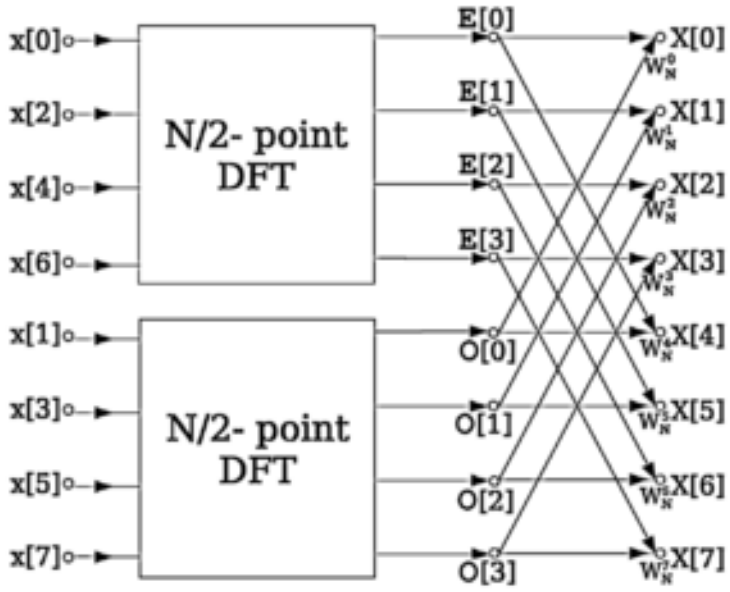
$$X_k = \underbrace{\sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N/2} mk}}_{\text{DFT of even-indexed part of } x_m} + e^{-\frac{2\pi i}{N}k} \underbrace{\sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2} mk}}_{\text{DFT of odd-indexed part of } x_m} = E_k + e^{-\frac{2\pi i}{N}k} O_k.$$

От тук личи, че двете половини са от своя страна дискретните редове на Фурие за четните и нечетните индекси. С Ek е отбелязано дискретното Фурие преобразование на четните индекси, а с Ok – на нечетните. Това разделение може да се приложи рекурсивно, като всяка половина се разделя от своя страна също на четни и нечетни индекси. Всяка от половините има два пъти по-малък брой входни елементи. Поради свойството на дискретно преобразование на Фурие, че изходът е периодичен, изходът за дискретните преобразувания за индекси 0 <= k < N/2 и N/2 <= k < N за ред с дължина N/2 са равни. За множителят пред нечетния ред е изпълнено, че $e^{-2\pi i(k+N/2)/N} = -e^{-2\pi ik/N}$, от което следва крайната формула:

$$X_k = \begin{cases} E_k + e^{-\frac{2\pi i}{N}k} O_k & \text{if } k < N/2 \\ E_{k-N/2} - e^{-\frac{2\pi i}{N}(k-N/2)} O_{k-N/2} & \text{if } k \geq N/2. \end{cases}$$

Алгоритъмът печели от разделянето по време (децимация по време) и преизползването на изчислените междинни резултати за получаване на повече от една нови стойности. Постигнатата накрая алгоритмична сложност е O(N*log N). Реализации на алгоритъма обикновено не използват рекурсия, а се изпълняват итеративно. Съществуват обобщения на алгоритъма за

разделяне на входните данни не само на две, а и на 3, 5, 7 и т.н. части.



Насоченият сигнален граф на изчислението на бърза трансформация на Фурие е прието да се нарича „пеперуда“ (butterfly/бабочка) поради специфичната му форма. На най-ниското му ниво има дискретно преобразование със две входни стойности. В обобщението за други разделяния, това преобразование се заменя със такова над повече входни стойности.

Бързото преобразование на Фурие в двумерния случай няма особености. Както и при обикновената дискретна трансформация на Фурие, се изпълнява едномерна трансформация на всеки ред, след това с получените стойности се се изпълнява едномерна трансформация за колоните.

17. Повишаване на достоверността на разпознаването чрез речник с образци. Разстояние между низове

Внимание! Тоя въпрос не му хареса на Димов. Очакваше да се напише още за информационния излишък, как ако има пр. 100 000 думи в езика, в речника да има пр 300 000 или 400 000, за да може да се възстановяват грешките, или поне аз това запомних. Един колега със същия въпрос, който мина по-леко, ми каза по време на изпита, че е вадил от Advances In Structural And Syntactic Pattern Recognition: Recent Advances in String Matching *(H Bunke)*

Има много начини за измерване на разстоянието между низове. Едни от тях е да се използва разстоянието на Хеминг. Разстоянието на Хеминг в контекста на символни низове е равно на броят позиции на които сравняваните низове са различни или иначе казано броят на заместванията, които трябва да се направят, за да станат двата низа еднакви. Пр. Разстоянието по Хеминг между низовете „Тошо“ и „Пешо“ е 2. Разстоянието на Хеминг обаче засяга само низове с равна дължина.

По общ начин за сравняване на низове е разстоянието по Левенщайн. При него се сравняват низове с произволна дължина, като пак се броят операциите, нужни за да станат низовете еднакви. Позволените операции са вече освен замяна на символ и добавяне и изтриване на символ. Разширение на тази метрика е разстоянието на Дамерау-Левенщайн, при което броят на операциите се разширява с разместването на два съседни символа в низа. Според статията на Дамерау тези 4 операции наброяват над 80% от всички правописни грешки при хората.

Примерен алгоритъм за намиране на разстоянието по Левенщайн

r(a, b) = (a == b)? 0: 1;

Имаме два низа s и t с дължини съответно n и m. Запълва се масив d[n+1][m+1], като в последният елемент ще се получи търсеното разстояние L(s, t)

```
for (int i = 0; i < n; ++i)
        d[i][0] = i;
for (int j = 0; j < m; ++j)
        d[0][j] = j
```

за останалите

```
for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j)
```

$$d(i, j) = \min(d(i-1, j)+1, d(i, j-1)+1, d(i-1, j-1) + r(s(i), t(j)))$$

За целите на разпознаването на текст може да се отчита, че някои грешки са по-често срещани от други и да им се присвои съответна цена. Две популярни ценови функции са където на всички операции се присвоява равна цена и където на заместването се присвоява два пъти по висока цена отколкото на останалите 2 операции.

Типична схема за разпознаване на ръкописен текст съдържа следните компоненти

1. Предварителна обработка (preprocessing) – бинаризация на изображението, установяване на наклона на текст, например чрез трансформация на Хаф и ротация под необходимия ъгъл;

2. Разпознаване на образите:

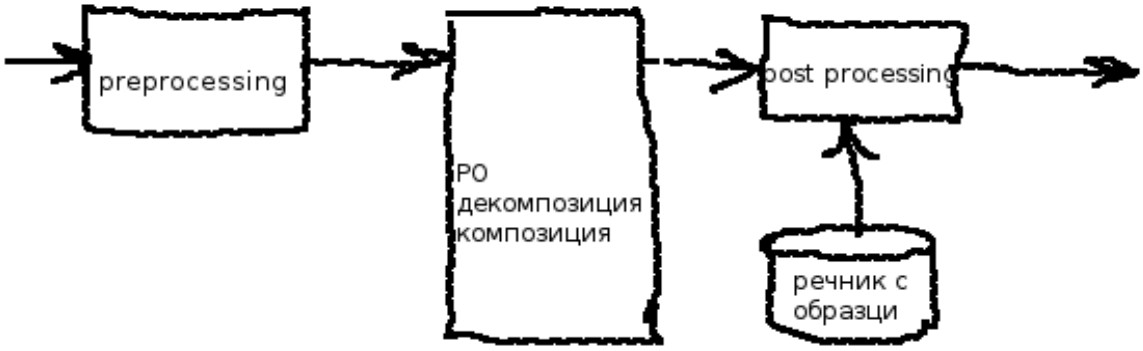
– разпознаване на графеми (елементарните съставящи на буквата) и сегментиране;

– „истинско“ разпознаване

– съставяне на правила за композиция на букви (морфеми) от графемите, на думите (лексеми) от буквите, на фразите от думи

(на този етап има високо ниво на грешките)

3. последваща обработка (post processing) – филтрират се допустимите думи с помощта на речник с цел отстраняване на информационния излишък.



Само разпознаването на образи не е достатъчно за разпознаване на ръкописен текст, необходимо е познаване на контекста за добри резултати. Комбинирането на различни източници на знания (морфологични, филологически) и на различни нива на контекст (графеми, букви, срички, думи, изречения) правят разпознаването много по-ефективно.