

# БЪРЗ ПОГЛЪЩЪЩ АЛГОРИТЪМ ЗА КОМПРЕСИЯ, ЧРЕЗ ЕНЕРГИЙНО МИНИМИЗИРАНЕ НА АКТИВНИ КОНТУРИ

гл. ас. д-р инж. Слава Миланова Йорданова

Технически университет – гр. Варна  
Факултет по Изчислителна техника и автоматика  
катедра: Компютърни системи и технологии  
sl@windmail.net

*Yordanova S. M. Fast greedy algorithm for active contours. The greedy algorithm is a fast iterative method for energy minimisation of snakes for image contour detection. In the Letter, an even faster algorithm is proposed. The new algorithm has the same performance as the conventional greedy algorithm, but reduces the computing time by 30% on DCT.*

## 1. ВЪВЕДЕНИЕ:

Разпознаването на даден образ е процес, в който се установява степента на прилика спрямо известно множество от възможни за него класове. Приликата или още съответствието се оценява с помощта на различни формални подходи - разнообразни математически критерии. Силата на тези критерии обаче, е пряко зависима от информативните способности на описващите образа формални признаци. Ето защо винаги са се търсели преходи в такива признакови пространства, които са оптимални в смисъла на информативността или с други думи, това са пространства, в които структурните характеристики на даден клас се изясняват по-добре.

## 2. ФОРМУЛИРАНЕ НА РАЗПОЗНАВАЩО ПРАВИЛО.

Разглеждаме дадено изображение като изградено от различни по енергийна стойност пиксели. Свързвайки пикселите, които са с еднакви или близки по стойност енергии, се получават модели на криви (контури), определени със своята енергийна стойност. Контурите могат да бъдат вътрешни и външни. Определя се един базисен контур (крива), като енергетичен-намаляващ spline, който може да е управляем под влиянието на вътрешен контур, на самото изображение и на външно въздействие. Контурът се описва, като параметрична крива  $v(s) = (x(s), y(s))$ , където дължината на дъга  $s$  е позиционен параметър. Енергията съответстваща на контура се определя като:

$$E_{snake}^* = \int_0^1 E_{snake}(v(s)) ds = \int_0^1 E_{internal}(v(s)) + E_{image}(v(s)) + E_{constraint}(v(s)) ds$$

където: *Einternal* представлява вътрешната енергия на контура, получена поради извиване или разкъсаности, *Eimage* има отношение към енергията на цялото изображение, и *Econstraint* е енергията на външното въздействие. Местоположението на кривите отговаря на локалния минимум на функционалната енергия.

Поглъщащият алгоритъм позволява контур с контролирана първа и втора непрекъснатост да попадне в област с определена енергия (по-голяма или по-малка) в рамките на изображението. Функционално енергията може да бъде намалена от предлаганият алгоритъм и енергетическата функция т.е. :

$$E = \int (\alpha(s)E_{continuity} + \beta(s)E_{curvature} + \gamma(s)E_{image}) ds$$

В този алгоритъм, енергетичната функция е изчислена както във всяка зададена точка така и във нейните съседни.

$$E_{continuity} = |d - |v_i - v_{i-1}||$$

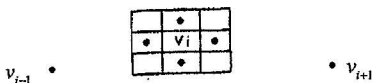
$$E_{curvature} = |v_{i-1} - 2v_i + v_{i+1}|^2$$

Точките преди и след нея на контура са използвани в изчислението на непрекъснатостта. Непрекъснатостта на контурите, средното разстояние между точките с еднакъв потенциал, дават разумна и бърза оценка на съответната кривина. Началната позиция на итерацията в изображението се означава с (*Mez*), s се определен като (*min - Mez*)/(*Макс - min*), където (*Макс*) и (*min*) са максималния и минимален градиент на съседните точки. Местоположението имащо минимална стойност е избрано като новата позиция на следващата итерация. Накрая на всяка итерация, кривината във всеки пункт на новия контур е точно определена, тогава и само тогава, когато стойността му е по-голяма от определен праг и се установява нула за следващата итерация.

### 3. ПОГЛЪЩАЩ АЛГОРИТЪМ

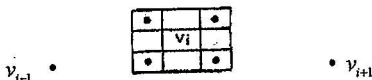
Поглъщащият алгоритъм е бързодействащ метод на итерациите, за процедура на минимална енергетична итерация на активен модел на зададена структура. Изчисленията необходими в този алгоритъм могат да се зададат със следните стъпки:

- Изчисляване на енергетичната функция на дадена точка, и на нейните гранични (съседни) точки. С това се определя и началото на областта.
- Накрая на всяка итерация, се определя кривината във всяка област на новия контур.



Фиг.1 Бързодействащ поглъщащ алгоритъм при  $n = 1$

Съседните пиксели на първият пиксел имат минимална стойност на енергетичната функция, също така имат малки стойности. Следователно изчислението е необходимо за търсенето на нова позиция може да е по-бързо, като се определят стойностите на осемте съседни пиксела по схемата показана на фигурите 1 и 2. По този начин поглъщащият алгоритъм покрива всички точки от областта.



Фиг.2 Бързодействащ поглъщащ алгоритъм при  $n = -1$

Енергията на кривата (контура)  $E_{curvature}$ , осигурява добра оценка на кривина в околността на определена точка. Тогава и само тогава когато  $E_{curvature}$  е достатъчно голяма, определена от зададена предварително стойност за праг, в околността на точката, величината на кривина  $c_i = \left| \frac{u_i}{|u_i|} - \frac{u_{i+1}}{|u_{i+1}|} \right|^2$  е също така голяма. Следователно, кривината в околност на точка с  $E_{curvature}$  е изчислена, само когато нейното стойност е по-голяма от зададен праг. Този нов поглъщащ алгоритъм се описва както следва:

```

j_min = j
if j_min not current location, pstmoved+ = 1
pattern = - pattern
/* process to determine where to allow corners in the next iteration */
for i = 0 to n - 1
if E_curvature < threshold1 then
c_i = \left| \frac{u_i}{|u_i|} - \frac{u_{i+1}}{|u_{i+1}|} \right|^2
for i = 0 to n - 1
if (c_i > c_{i-1} and c_i > c_{i+1} and c_i > threshold2 and mag(v_i) > threshold3)
then beta_i = 0
until pstmoved < threshold3

```

```

Pseudo-code
do
/* loop to move points to new locations */
for  $i = 0$  to  $n$ 
 $E_{\min} = BIG$ 
if pattern = 1 then
for  $j = 0$  to 4
 $k = j$ th point in Fig. 1
 $E_j = \alpha_i E_{continuity,k} + \beta_i E_{curvature,k} + \gamma_i E_{image,k}$ 
if  $E_j < E_{\min}$  then
 $E_{\min} = E_j$ 
 $j_{\min} = j$ 
else
for  $j = 0$  to 4
 $k = j$ th point in Fig. 2
 $E_j = \alpha_i E_{continuity,k} + \beta_i E_{curvature,k} + \gamma_i E_{image,k}$ 
if  $E_j < E_{\min}$  then
 $E_{\min} = E_j$ 

```



Фиг.3 Погълещ алгоритъм

#### 4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Експериментите са направени с използване на изображения зададени от международният комитет по стандартизация ISO, които са определени като

базисни за доказване качествата на предсавените алгоритми за компресия на изображения. Броят на итерациите и времето за обработка на изображенията с използване на двата алгоритъма са сравнени и табулирани в Таблица 1. Позиционни параметри са: (1.0,1.0,1.0), (1.0,1.0,1.2), (1.0,1.2,1.0) и (1.2,1.0,1.0). Направено е сравнение на бързодействието на предлаганият алгоритъм с компресия на същите изображения, чрез използване на Бързо дискретно косинусово преобразуване на Фурие.

Таблица 1: Брой на итерации и време за изпълнение на бързодействащ алгоритъм за компресия

$\alpha, \beta, \gamma$ (позиция)		DCT	Fast greedy
(1.0, 1.0, 1.0)	Iterations	17	20
	Time [s]	1.0730	0.7118
(1.0, 1.0, 1.2)	Iterations	12	18
	Time [s]	0.7618	0.6422
(1.0, 1.2, 1.0)	Iterations	16	18
	Time [s]	1.0105	0.6335
(1.2, 1.0, 1.0)	Iterations	17	18

Времето необходимо за бързодействащия поглъщащ алгоритъм е с 30% по-малко, от това при компресия използваща дискретно косинусово преобразуване.

По същество беше изложено кратко едно обобщение на изследвания, свързани с реалното приложение на различни методи от теорията на разпознаването на образи, както и главните изводи до които сме стигнали, гледайки на тях критично. Естествено е всяко обобщение да породи нови идеи, какъвто е случаят с предлаганият тук подход към компресиране със загуба на информация, използвайки енергийният потенциал на контурите на изображението.

## 5. ЛИТЕРАТУРА

1. Kass M., Witkin A. "Active contour model", Proc. first Int. Conf. on Computer Vision, London 2001
2. Williams D. "A fast algorithm for active contours and curvature estimation", Image Understanding, 2000
3. Йорданова Сл. М. Рачев Б., Наумов В.: "Мултимедия и компресия на информация", България, МТМТ, 2000г.

Работата е рецензирана от доц. д-р А. Антонов – кат. "Компютърни системи и технологии", ТУ- гр. Варна.