ЛАБОРАТОРНА РОБОТА №7

ОБЧИСЛЕННЯ ФРАКТАЛЬНОЇ РОЗМІРНОСТІ

Цель: изучение основных алгоритмов вычисления фрактальной размерности.

При описании свойств фрактала важную роль играет такая его характеристика как фрактальная размерность. Дадим общее определение этой величины. Пусть d – обычная Евклидова размерность пространства, в котором находится наш фрактальный объект ($d=1$ – линия, $d=2$ – плоскость, $d=3$ – обычное трехмерное пространство). Покроем теперь этот объект целиком d-мерными "шарами" радиуса r. Предположим, что нам потребовалось для этого не менее, чем $N(r)$ шаров. Тогда, если при достаточно малых r величина $N(r)$ меняется с по степенному закону

$$N(r) : \frac{1}{r^D}$$

то – D называется размерностью *Хаусдорфа-Безиковича* или *фрактальной размерностью* этого объекта.

Используя понятие фрактальной размерности можно дать более строгое, чем приведенное во введении, определение фрактала. Согласно этому определению фрактал представляет собой объект, размерность *Хаусдорфа-Безиковича* которого больше его топологической размерности (0 – для россыпи точек, 1 – для кривой, 2 – для поверхности и т.д.).

Формулу (1) можно переписать также в виде

$$D = -\lim_{r\to 0}\frac{\ln N(r)}{\ln r}$$

Это и служит общим определением фрактальной размерности. В соответствии с ним величина является локальной характеристикой данного объекта. Совершенно ясно, что мы получили бы при оценке фрактальной размерности тот же самый результат, если бы использовали процедуру покрытия фрактала кубами (квадратами, если фрактальный объект располагается на плоскости).

Наиболее просто реализовать программно можно алгоритм вычисления фрактальной размерности называемый box-counting. Его идея заключается в том, что евклидово пространство, содержащее изображение объекта, разделяют сеткой с ячейкой размера r и подсчитываются непустые, занятые исследуемым объектом, квадраты $N(r)$. Затем размер r уменьшают, и снова подсчитывают число непустых полей $N(r)$. Наклон графика в логарифмическом масштабе $N(r)$ от $1/r$ соответствует величине размерности.

Программная реализация алгоритма для бинарных изображений приведена в листинге boxcount.m

Листинг boxcount.m

```
function D = boxcount(c)
warning off
c = logical(squeeze(c));
warning on
dim = ndims(c); % dim is 2 for a vector or a matrix, 3 for
a cube
if dim ~= 2
error('dimension is not 2');
end
% transpose the vector to a 1-by-n vector
if length(c)==numel(c)
dim=1;
if size(c,1)~=1
c = c';
end
end
width = max(size(c)); % largest size of the box
p = log(width)/log(2); % number of generations
% remap the array if the sizes are not all equal,
% or if they are not power of two
% (this slows down the computation!)
if p~=round(p) || any(size(c)~=width)
p = ceil(p);
width = 2^p;
mz = zeros(width, width);
mz(1:size(c,1), 1:size(c,2)) = c;
c = mz;
end
n=zeros(1,p+1); % pre-allocate the number of box of size r
n(p+1) = sum(c(:));
for g=(p-1):-1:0
siz = 2^(p-g);
siz2 = round(siz/2);
for i=1:siz:(width-siz+1)
for j=1:siz:(width-siz+1)
c(i,j) = ( c(i,j) || c(i+siz2,j) || c(i,j+siz2)
|| c(i+siz2,j+siz2) );
end
```

```
end
n(g+1) = sum(sum(c(1:siz:(width-siz+1),1:siz:(widthsiz+
1))));
end
n = n(end:-1:1);
r = 2.^(0:p); % box size (1, 2, 4, 8...)
loglog(r,n,'s-');
xlabel('r, box size'); ylabel('n(r), number of boxes');
title([num2str(dim) 'D box-count']);
koef = polyfit(-log(r), log(n), 1);
D = koef(1);
```

ЗАДАНИЕ

1. Модифицировать функцию boxcount для анализа яркостных изображений (функция должна обрабатывать трёхмерную матрицу) в системах MAXIMA и MathCAD алгоритм CCT.