

1 Primo Esercizio

1.1 Primo obiettivo

Il problema affrontato consiste nel tentare di risalire ad un'immagine sconosciuta conoscendo i risultati di k e $k+1$ iterazioni dell'algoritmo del gatto di Arnold, la cui matrice non è nota.

Per far questo individuiamo i parametri della matrice incognita mediante il seguente procedimento:

- 1) Si fanno variare le 4 componenti della matrice 2×2 tra gli interi;
- 2) Si applica l'algoritmo di Arnold in forma vettoriale all'iterazione k -esima;
- 3) Si controlla se il risultato coincide o meno con l'iterazione $k+1$ -esima;
- 4) Si determinano i parametri della trasformazione;
- 5) Si calcola l'inversa (modulo 1024, cioè la dimensione dell'immagine);
- 6) Si applica k volte la trasformazione inversa alla prima immagine risalendo così all'immagine di partenza;

Nota: le immagini utilizzate sono le immagini n°6.

Di seguito è riportata l'immagine ottenuta:



Figura 1: Immagine originale

1.2 Script e function

```
%function 1
%function arnold2k

function B = arnold2k(A, k, M)
%%ARNOLD2 Applica k passi della mappa di Arnold.
% Questa funziona applica K passi della mappa del gatto di Arnold
% all'immagine rappresentata dalla matrice A. Se viene specificato
% anche il parametro opzionale M (una matrice 2x2), questo viene
% utilizzato al posto di [2 1 ; 1 1] per definire la mappa.

m = size(A, 1);
```

```

if ~exist('M', 'var')
    M = [ 2 1 ; 1 1 ];
end

N = eye(2);

for j = 1 : k
    N = mod(M * N, m);
end

% Costruiamo una matrice 2 x m^2 che contiene tutti gli indici (i,j)
II = zeros(2, m^2);
II(2, :) = kron(1:m, ones(1,m));
II(1, :) = kron(ones(1,m), 1:m);

IIh = 1 + mod(N * (II - 1), m);

% Costruiamo un vettore 1 x m^2 che corrisponde agli indici in IIh
Ih = [ 1 , m ] * (IIh - [ 0 ; 1 ] * ones(1, m^2));

B = A;
for j = 1 : size(B, 3)
    B(Ih + (j-1)*m^2) = reshape(A(:, :, j), m*m, 1);
end
end

```

La function 2 trova esponente determina il numero k di volte in cui la mappa di Arnold va applicata per poter riavere l'immagine originale, ammesso che k esista.

```

%function 2

%function trova_esponente
function l = trova_esponente(M, l)
%PERIODO Determina l'ordine di M modulo n.

% M matrice simmetrica 2x2, ne possiamo avere al massimo n^3
N = eye(2, 2);

for i = 1 : l^3
    N = mod(M * N, l);

    if all( N - eye(2) == 0, 'all' )
        l = i;
        return;
    end
end

error('Non esiste m tale che M^m = I');

end

```

Si riporta di seguito lo script composto da :

- 1) una parte di codice che cerca pixel unici;
- 2) troviamo la loro immagine tramite la trasformazione nella seconda immagine;
- 3) troviamo dunque dei vettori linearmente indipendenti e svolgiamo calcoli di algebra lineare trovando così la matrice M e dunque l'immagine originale.

```

%script

A = imread("img_6_32.png");
B = zeros(1,4);
%ricerca pixel unici
for i = 1:20

```

```

    for j = 1:20
        [row,col] = find(A(:,:,1)==i & A(:,:,2)==j);
        sz = size(col);
        if(sz(1)==1)
            B(end+1,:) = [row,col,i,j];
        end
    end
end
sz = size(B);
a = B(2,:);
b = zeros(1,4);
i=2;
e=0;
while(i<=sz & e==0)
    if(a(1)*B(i,2) ~= B(i,1)*a(2))
        b = B(i,:);
        e=1;
    end
    i = i+1;
end

B = imread("img_6_33.png");
a1 = a(1:2);
b1 = b(1:2);
[row,col] = find(B(:,:,1)==a(3) & B(:,:,2)==a(4));
a2 = [row,col];
[row,col] = find(B(:,:,1)==b(3) & B(:,:,2)==b(4));
b2 = [row,col];

%soluzioni sistema lineare 2x2
a1 = a1-1;
b1 = b1-1;
a2 = a2-1;
b2 = b2-1;
a = [0 ;0];
b = [0 ;0];
for i = 1:1024
    for j = 1:1024
        if(mod(i*a1(1)+j*a1(2),1024) == a2(1))
            a(:,end+1) = [i; j];
        end
        if(mod(i*b1(1)+j*b1(2),1024) == b2(1))
            b(:,end+1) = [i ;j];
        end
    end
end
n = size(a);
m = size(b);
i = 2; e=0;
while(i<=n(2) & e==0)
    j=2;
    while(j<=m(2) & e==0)
        if(a(:,i) == b(:,j))
            ab= a(:,i);
            e=1;
        end
        j = j+1;
    end
    i = i+1;
end
end

```

```

c = [0 ;0];
d = [0 ;0];
for i = 1:1024
    for j = 1:1024
        if(mod(i*a1(1)+j*a1(2),1024) == a2(2))
            c(:,end+1) = [i; j];
        end
        if(mod(i*b1(1)+j*b1(2),1024) == b2(2))
            d(:,end+1) = [i ;j];
        end
    end
end
n = size(c);
m = size(d);
i = 2; e=0;
while(i<=n(2) & e==0)
    j=2;
    while(j<=m(2) & e==0)
        if(c(:,i) == d(:,j))
            cd= c(:,i);
            e=1;
        end
        j = j+1;
    end
    i = i+1;
end

%finale
M = [ab' ; cd'];
E=arnold2k(A,trova_esponente(M,1024)-32, M);
imagesc(E);

```

1.3 Note

Ho lavorato con Chiara Di Sano.