Interpolazione di immagini

Data un'immagine in formato png (o jpeg o altro) di $m \times n$ pixel, la si vuole rappresentare con un numero maggiore di pixel (ad esempio $2m \times 2n$ pixel).

Da una foto di Alexa da Pixabay

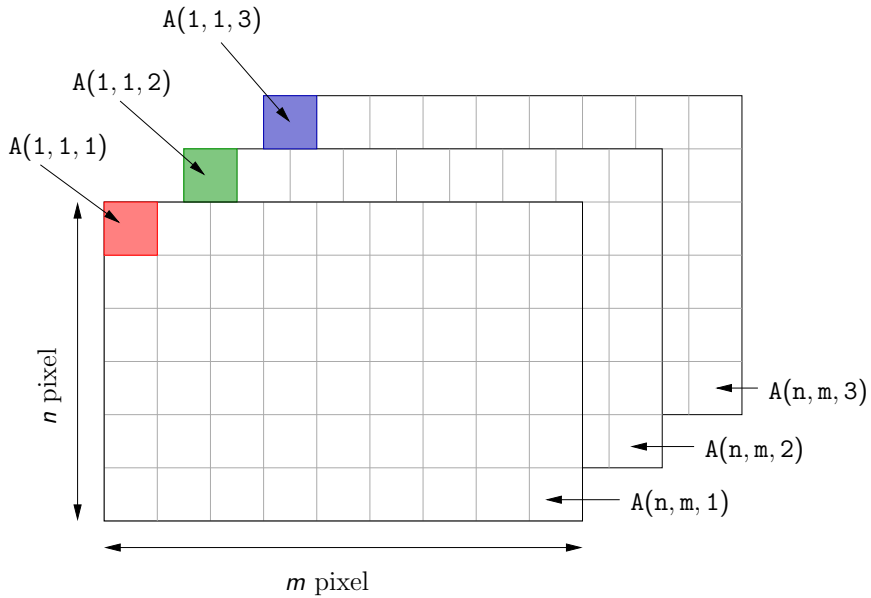
Immagini in matlab

Scaricare *seahorse.png* dalla pagina del corso.

```
% per leggere l'immagine dal file trecolori.png
A=imread('seahorse','png');
% per visualizzare il contenuto di A
figure(1); clf; imshow(A);
% per salvare il contenuto di A  nel file new.png
imwrite(A,'new','png');

% se si ha un file .jpg:
% per leggere l'immagine dal file trecolori.jpg
A=imread('seahorse','jpg');
% per salvare il contenuto di A  nel file new.jpg
imwrite(A,'new','jpg');
```

Come Matlab gestisce le immagini

Un'immagine di $m \times n$ pixel è memorizzata in un array a 3 indici `A=(n,m,3)`

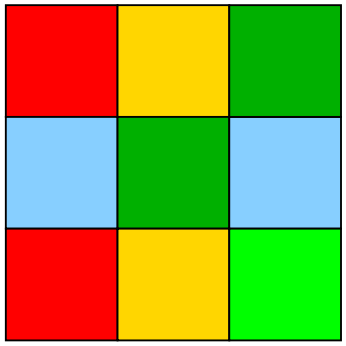


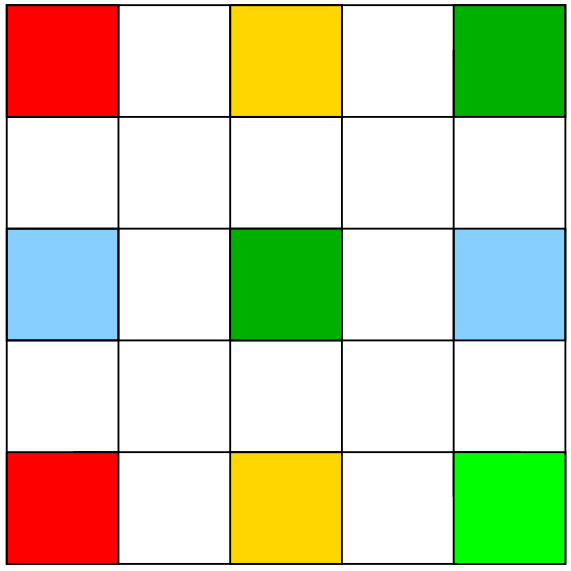
n= numero pixel lungo la y = numero di righe di `A`
m= numero pixel lungo la x = numero di colonne di `A`
La combinazione dell'intensità dei tre colori base (RGB) nella posizione `(i,j)` è il colore finale del pixel `(i,j)`.
`A` è di tipo `uint8`: ogni elemento occupa 8bit e può assumere solo valori interi tra 0 e 255.

```
>> whos
  Name          Size                    Bytes  Class
  A           230x280x3                193200  uint8
```

Come aumentare il numero di pixel

1. stabiliamo un fattore di ingrandimento (es: `fattore=2`)
2. ingrandiamo l'immagine introducendo dei pixel intermedi





e andiamo a valutare lì le intensità dei colori:

1. con semplice copiatura dei pixel (*nearest interpolation*),
2. con una interpolazione bilineare composita (*bilinear interpolation*),
3. con spline cubiche (*cubicspline interpolation*).

Conversione da uint8 a double

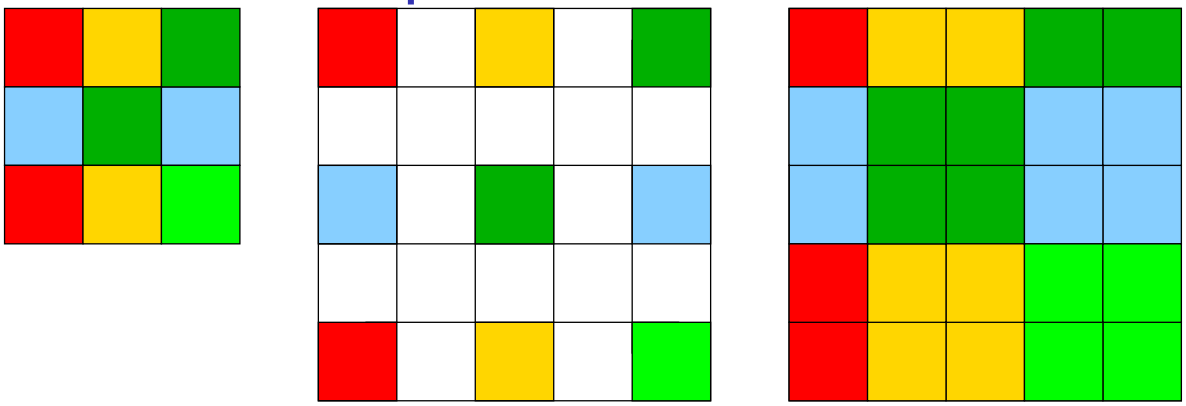
```
% per copiare le tre pagine di A in 3 matrici
% di tipo double (su cui fare i conti)

[n,m,dim]=size(A);
R=double(A(:,:,1));
G=double(A(:,:,2));
B=double(A(:,:,3));
```

Ognuna di queste tre matrici contiene i valori di una funzione di due variabili. Provare a visualizzarle con il comando `mesh`:

```
figure; mesh(R)
figure; mesh(G)
figure; mesh(B)
```

Il nostro obiettivo è interpolare queste funzioni in punti più fitti rispetto a quelli corrispondenti ai pixel.

1. Nearest interpolation



Si replica il contenuto del pixel più vicino.
È realizzabile con la function `interp2` di MATLAB:

```
z1=interp2(x,y,z,x1,y1,'nearest')
```

- `x,y`= nodi di interpolazione (dove conosco l'intensità di colore) sono le coordinate dei pixel (cioè valori interi da 1 a m (in x) e da 1 a n (in y)):

  ```
  [x,y]=meshgrid(1:m,1:n);
  ```

- `z`= intensità di colore nei pixel, cioè una delle matrici `R, G, B`
- `x1,y1`= coordinate dei pixel e valori intermedi

  ```
  [x1,y1]=meshgrid(1:0.5:m,1:0.5:n);
  ```

- `z1`= interpolatore, cioè l'intensità di colore nelle vecchie e nuove coordinate.

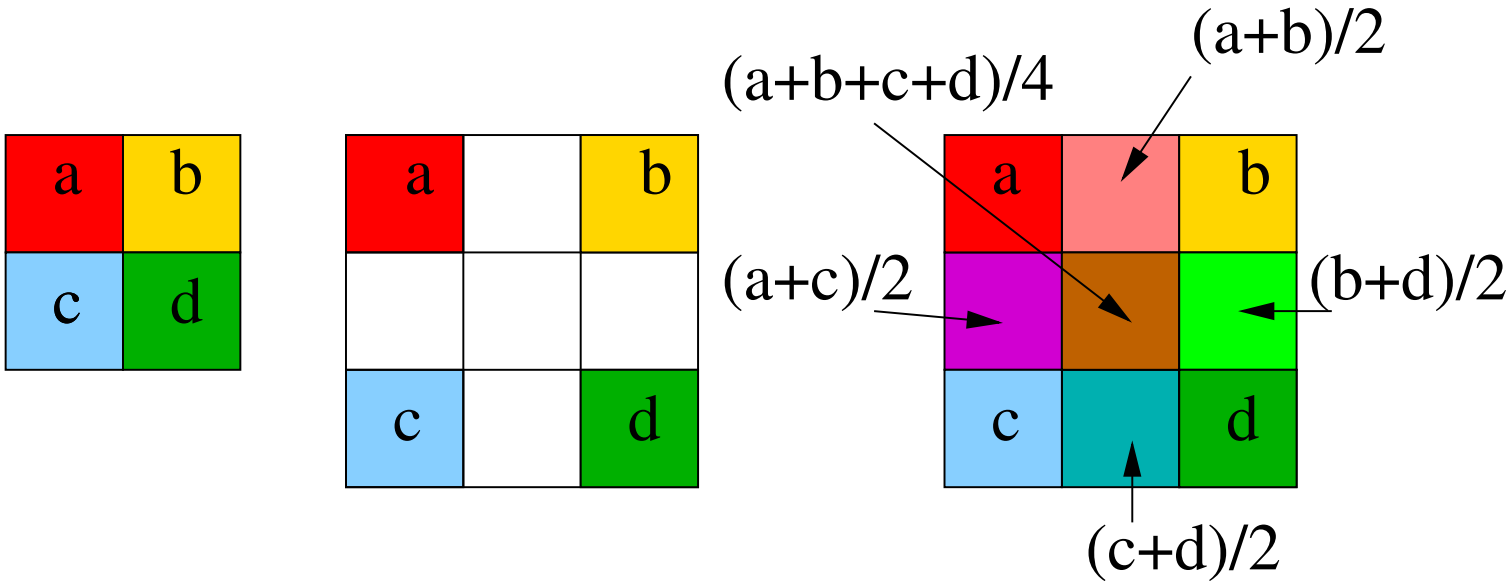
- Effettuare l'interpolazione su tutte e tre le matrici `R,G,B` e salvare il risultato dell'interpolazione in nuove matrici `R1,G1,B1`,
- costruire una nuova variabile `A1` di tipo `uint8` contenente la nuova immagine:

```
[n1,m1]=size(R1);
A1=uint8(zeros(n1,m1,3));
A1(:,:,1)=uint8(R1);
A1(:,:,2)=uint8(G1);
A1(:,:,3)=uint8(B1);
```

- visualizzarla con `imshow`.

Nearest interpolation

2. Interpolazione bi-lineare



Questa è interpolazione bilineare composita in 2D, realizzabile in MATLAB con la function `interp2` e metodo `'linear'`:

```
z1=interp2(x,y,z,x1,y1,'linear');
```

Generare tre nuove matrici `R2`, `G2`, `B2` e una nuova immagine `A2`, quindi visualizzarla.

Bilinear interpolation

3. Interpolazione con spline cubiche in 2D

I valori intermedi tra un pixel e l'altro sono costruiti con interpolazione spline in 2D.
È realizzabile in MATLAB con la function `interp2` e metodo `'spline'`:

```
z1=interp2(x,y,z,x1,y1,'spline');
```

Generare tre nuove matrici `R3, G3, B3` e una nuova immagine `A3`, quindi visualizzarla.

Spline interpolation

Conclusioni

- nearest interpolation è grezza
- bilinear interpolation è più regolare, l'immagine ha i contorni più lisci
- spline interpolation è ancora più regolare, l'immagine sembra avere una profondità maggiore, tuttavia attenzione che spline può introdurre delle oscillazioni spurie attorno ai contorni (provare a scaricare l'immagine insect.png).