Ingegneria Biomedica A.A. 2014-2015

Bio@NECST
28 settembre 2015

DIPARTIMENTO DI ELETTRONICA E INFORMAZIONE

POLITECNICO DI MILANO

# Elaborazione di immagini biomediche

di Laura Barilli, Sara Bridio, Giulia Core, Marco Gucciardi

**Obiettivo:**

Creazione di un programma in linguaggio C che permetta l'elaborazione di immagini biomediche

# Strumenti

## Identificazione del paziente e storicizzazione

- Caricamento immagini di interesse
- Ordinamento immagini

## Scelta immagine e operazione

- Riconoscimento immagine scelta
- Selezione operazione

## Elaborazione immagine e visualizzazione dei risultati

- Numero di livelli di grigio
- Distribuzione livelli di grigio
- Riduzione dei livelli di grigio a 10
- Definizione del contorno
- Area inclusa nel contorno dell'immagine ($mm^2$ e pixels)
- Area totale zone molto scure e molto chiare ($mm^2$ e pixels)
- Variazione delle aree totali (zone chiare e scure) fra l'immagine scelta e la successiva ($mm^2$ e pixels)
- Variazione delle aree totali (zone chiare e scure) fra l'immagine scelta e una seconda immagine appartenente allo stesso paziente ($mm^2$ e pixels)

# Immagini Bitmap

```
typedef struct {
    unsigned int size;
    unsigned short int reserved1;
    unsigned short int reserved2;
    unsigned int offset;
} Header;

typedef struct {
    unsigned int size;
    int width,height;
    unsigned short int planes;
    unsigned short int bits;
    unsigned int compression;
    unsigned int imagesize;
    int xresolution,yresolution;
    unsigned int ncolours;
    unsigned int importantcolours;
} Info;

typedef struct {
    unsigned char grey;
} Pixel;
```

```
typedef struct{
    unsigned char magic[2];
    Header header;
    Info info;
    unsigned char color_table[DATA_DIM*4];
    Pixel dato[DATA_DIM][DATA_DIM];
}BMP_Image;
```

# Immagini Bitmap

```
typedef struct {
    unsigned int size;
    unsigned short int reserved1;
    unsigned short int reserved2;
    unsigned int offset;
} Header;

typedef struct {
    unsigned int size;
    int width,height;
    unsigned short int planes;
    unsigned short int bits;
    unsigned int compression;
    unsigned int imagesize;
    int xresolution,yresolution;
    unsigned int ncolours;
    unsigned int importantcolours;
} Info;

typedef struct {
    unsigned char grey;
} Pixel;
```

```
typedef struct{
    unsigned char magic[2];
    Header header;
    Info info;
    unsigned char color_table[DATA_DIM*4];
    Pixel dato[DATA_DIM][DATA_DIM];
}BMP_Image;
```

**Requisiti delle nostre immagini**

Dimensioni: 256 x 256 pixel

Numero bits/pixel: 8

Colori: scala di grigi

## Caricamento e ordinamento:

- Controllo dei requisiti durante l'apertura del file in lettura

```
int loadBMP(char * filename, BMP_Image *img);
```

```
typedef struct{
    char IDfin[21];
    char ID[11];
    data dataimg;
    char filename[20];
    BMP_Image newimage;
} imgcorr;
```

- Aggiunta informazioni a immagine BMP

```
imgcorr caricaimm();
```

- Ordinamento secondo IDfin

```
struct miaStruttura* ordinamento(struct miaStruttura* testa, int* numeri);
```

```
struct miaStruttura{
    imgcorr immagine;
    struct miaStruttura* next;
};
```

- Salvataggio delle immagini

```
int saveBMP(BMP_Image image, char * filename);
```

# Scelta immagine e operazione

## Riconoscimento:

```
struct miaStruttura* riconoscimento (char* str_ID, struct miaStruttura* lista, int num);
```

## Selezione operazione:

```
printf("\nScegli il numero dell'opzione\n");
scanf(" %c",&scelta);

switch(scelta){
    case '1':
        num_grigi(trovata);
        break;
    case '2':
        distribuzione_grigi(trovata,vet);
        break;
    case '3':
        riduzione(trovata);
        break;
    case '4':
        contorno(trovata);
        break;
    case '5':
        area_contorno(trovata);
        break;
    case '6':
        area_chiara_scura(trovata);
        break;
    case '7':
        variazione_imm_consec(trovata);
        break;
    case '8':
        variazione_imm_scelta(trovata,testina,tot_img);
        break;
    case '9':
        printf("Uscita dal programma.\n");
        return 0;
        break;
    default:
        printf("\n\n ERRORE!!!!\n\n");
}
```

# Elaborazione immagine e risultati (1)

## Numero di grigi:

Calcolo del numero di livelli di grigio presenti nell'immagine

```
void num_grigi(struct miaStruttura* struct_img);
```

```
(struct_img->immagine.newimage.dato[i][j].grey)
```

## Distribuzione livelli di grigio:

Calcolo della distribuzione dei livelli di grigio presenti nell'immagine

```
void distribuzione_grigi(struct miaStruttura* struct_img, int* v);
```

## Riduzione dei livelli di grigio (da 256 a 10):

```
void riduzione(struct miaStruttura* struct_img);
```

# Elaborazione immagine e risultati (2)



Immagine originale

Immagine con 10 livelli di grigio

http://www.cosy.sbg.ac.at/~pmeerw/Watermarking/lena.html
*8bit grayscale*

# Elaborazione immagine e risultati (3)

## Definizione del contorno:

```
void contorno(struct miaStruttura* img_in);
```

- Selezione della zona di interesse dell'immagine → ridurre problemi relativi agli artefatti

Immagine originale | Immagine tagliata | Definizione contorno

http://www.med.harvard.edu/aanlib/cases/caseNA/pb9.htm

## Area inclusa nel contorno dell'immagine ($mm^2$ e pixels):

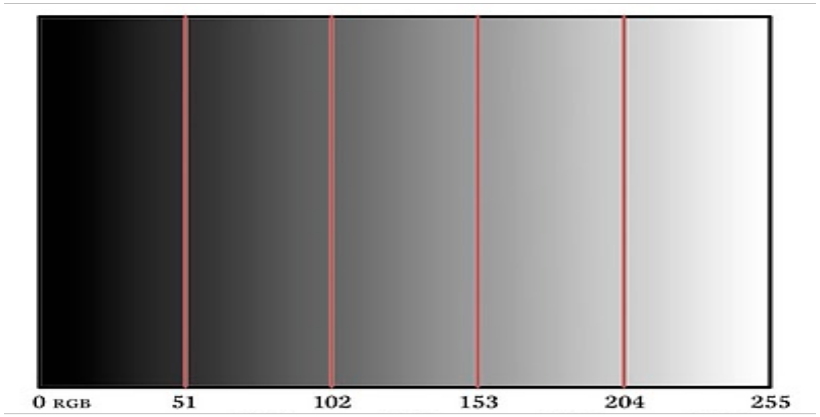
```
void area_contorno(struct miaStruttura* img_in);
```

## Area totale zone molto scure e molto chiare ($mm^2$ e pixels):

Calcolo del numero di pixel scuri e del numero di pixel chiari presenti all'interno del contorno dell'immagine

```
void area_chiara_scura(struct miaStruttura* img_in);
```

Soglia molto scuro → minore di 70
Soglia molto chiaro → maggiore di 185



**Variazione delle aree totali (zone chiare e scure) fra l'immagine scelta e la successiva ($mm^2$ e pixels):**

```
void variazione_imm_consec(struct miaStruttura* img_in);
```

**Variazione delle aree totali (zone chiare e scure) fra l'immagine scelta e una seconda immagine appartenente allo stesso paziente ($mm^2$ e pixels):**

```
void variazione_imm_scelta(struct miaStruttura* img_in, struct miaStruttura* capolista, int num);
```

# Conclusioni

Ulteriori requisiti delle immagini elaborate dal programma:

- Immagine priva di artefatti
- Sfondo dell'immagine nero

Problemi di conversioni delle immagini biomediche