C++, Sobel e rilevamento dei margini

evilsocket
http://evilsocket.altervista.org/

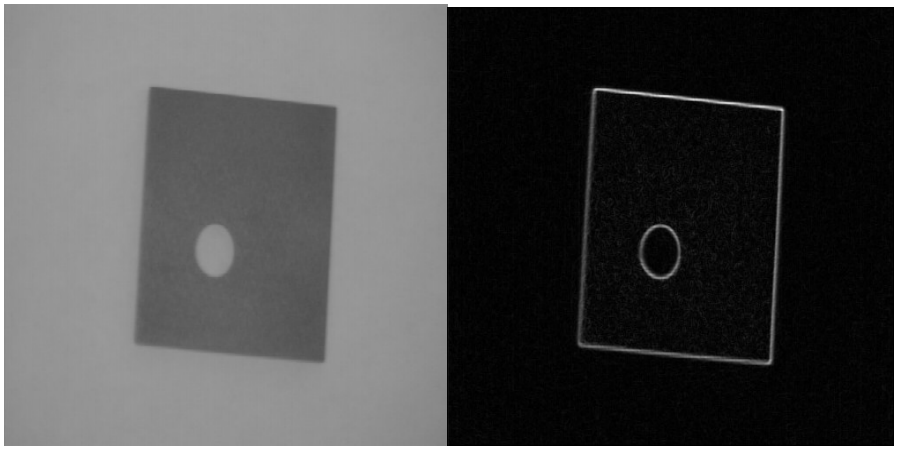
Ciao ragazzi, in questo articlo parleremo di uno degli algoritmi + famosi x il rilevamento dei bordi sulle immagini .

Le mappe di Sobel !

Premetto che è un articolo abbastanza avanzato, diretto ai + esperti in c/c++, magari con qualche esperienza sull elaborazione digitale delle immagini .

Nel rilevamento dei margini di Sobel, si usano delle matrici 3x3 (conosciute anche come "maschere di convoluzione"), una per il gradiente sull asse X e una, com'è ovvio su quello Y ^^ .
Praticamente, quello che andremo a fare, è moltiplicare ogni tripletta RGB dell immagine per la mappa di Sobel, finkè non otteniamo i due gradienti ... poi non resta che calcolare il valore assoluto della somma degli stessi .
In seguito, si confronta il risultato ottenuto con un valore di soglia preimpostato ... se è maggiore o uguale della soglia ... beh ... il pixel alla posizione X,Y è parte di un margine ^^ .

Ecco un esempio di una trasformazione di sobel :



Ora farò parlare il codice :D

```
#include <math.h>
```

```
#include <stdio.h>

// piccola macro x convertire i byte dell RGB in un int
#define RGB(r,g,b) ((((word)r)<<16)|(((word)g)<<8)|((word)b))

// mappa di soble per l'asse X
const double _SOBEL_Gx[3][3] = { {-1.0,+0.0,+1.0},
                                 {-2.0,+0.0,+2.0},
                                 {-1.0,+0.0,+1.0}
                               };
// mappa di soble per l'asse Y
const double _SOBEL_Gy[3][3] = { {+1.0,+2.0,+1.0},
                                 {+0.0,+0.0,+0.0},
                                 {-1.0,-2.0,-1.0}
                               };

// calcola i gradienti x-y di un pixel
double get_sobel_gradient ( unsigned char* pimg, int width, int height, int x,
int y )
{
    double sobel_gradient_x = 0,
           sobel_gradient_y = 0;

    int mx = 0,
        my = 0,
        sx = 0,
        sy = 0;

    for( mx = x; mx < x + 3 ; mx++ )
    {
        sy = 0;

        for( my = y; my < y + 3 ; my++ )
        {
            if( mx < width && my < height )
            {
                int r,g,b,idx;

                idx = (mx + width * my) * 3;

                r = pimg[ idx + 2 ];
                g = pimg[ idx + 1 ];
```

```
                b = pimg[ idx + 0 ];

                sobel_gradient_x += RGB(r,g,b) * _SOBEL_Gx[sx][sy];
                sobel_gradient_y += RGB(r,g,b) * _SOBEL_Gy[sx][sy];
            }
            sy++ ;
        }

        sx++ ;
    }

    return abs(sobel_gradient_x) + abs(sobel_gradient_y);
}

int main(void)
{
// ... qui creiamo/apriamo l'immagine come un unsigned char * img
// in base al formato dal quale stiamo leggendo (ex: gif, png, bmp, jpg, etc
..)

    double threshold = 5000.0;

    for( int y = 0; y < height; y++ )
    {
        for( int x = 0; x < width; x++ )
        {
            if( get_sobel_gradient(img,width,height,x,y) >= threshold )
                printf( "Il pixel a [%d,%d] è parte di un margine !\n", x, y );
        }
    }

    return 0;
}
```

Spero che possa interessare a qualcuno questo articolo e/o questo argomento .

evilsocket