

OpenCV

una completa libreria open source per la computer vision

Da sempre uno dei campi più affascinanti della ricerca scientifica è volto alla riproduzione artificiale delle capacità umane. Tra queste capacità la visione, intesa come pura acquisizione di immagini, è attualmente considerabile un problema già risolto, o almeno un punto già segnato, visto che le capacità visive di sistemi ottici e relativi sensori hanno ampiamente superato le possibilità dell'occhio umano in quanto a sensibilità, velocità e risoluzione. Il passo successivo, cioè la capacità di interpretare ed utilizzare correttamente le informazioni acquisite, presenta invece ancora molti problemi insoluti. Convertire un'immagine in informazioni "oggettive" astruendone il contenuto dalla pura rappresentazione luminosa, sebbene sia un'operazione banale per un cervello umano adulto è, a tutt'oggi, un problema di elevata complessità per un sistema automatico. Oltretutto il campo di ricerca è evidentemente molto giovane, con meno di trent'anni di esperienza.

In quest'ottica si inserisce la necessità di una base comune di strumenti analitici, primo dei quali una libreria che raccolga le funzionalità degli algoritmi più utilizzati, oltre che una serie di formati di rappresentazione dei dati secondo standard aperti e condivisi. Le librerie OpenCV (Open Computer Vision) nascono appunto a questo scopo.

Lo sviluppo prende le mosse da un gruppo di ricerca sponsorizzato da Intel. E' infatti parzialmente basata sulla Intel Image Processing Library (IPL). Tale prodotto è oggi integrato nella libreria commerciale IIPP (Intel Integrated Performance Primitives), con cui conserva piena compatibilità, e che può eventualmente rendere disponibili un completo ventaglio di funzioni di trattamento segnali (audio, video, sintesi vocale, crittografia, ecc) oltre che una migliore ottimizzazione delle prestazioni. Si avvale inoltre di numerosi contributi dalle più svariate provenienze. A tal proposito l'elenco di credits citati dalla pagina ufficiale è quantomeno impressionante, si va da ricercatori del MIT, fino a docenti della Berkley University. Inutile sottolineare come questo offra già una certa garanzia relativamente alla buona qualità del codice e degli algoritmi applicati.

Tra i punti di forza sottolineiamo inoltre la politica di licen-

za utilizzata, in stile BSD. A grandi linee questo permette una libera redistribuzione sia in forma sorgente che binaria, anche all'interno di prodotti commerciali, a condizione di mantenere le note di copyright e di non utilizzare il nome Intel a scopo promozionale di prodotti derivati.

Panoramica

Affrontiamo subito un argomento che può essere causa di equivoci. Con il termine "libreria grafica" infatti si identificano genericamente almeno tre famiglie di librerie i cui scopi sono sostanzialmente differenti:

- I Toolkit, ovvero librerie di primitive per la creazione di oggetti grafici di interfaccia (finestre, icone, bottoni, ecc).
- Librerie di rendering e multimedia, come DirectX e OpenGL, orientate alla massima performance nella creazione di effetti poligonali o vettoriali. L'utilizzo più comune è teso all'ottenimento di elevate prestazioni grafiche sfruttate ad esempio nei videogiochi o nelle applicazioni multimediali.
- Librerie di gestione hardware grafico, come digitalizzatori e frame-grabber. Pur includendo tipicamente una base di funzioni di trattamento sono generalmente da considerarsi come API dei relativi "driver" hardware.

Le OpenCV, pur includendo alcune funzionalità tipiche di ciascuna delle famiglie citate, non fanno parte di nessuno di questi gruppi. L'utilizzo primario è infatti quello collegato alla visione artificiale, il cui problema principale, come già visto, è quello di estrarre dalle immagini dati significativi e trattabili in modo automatico. Tale campo di studio trova le sue applicazioni più comuni nella robotica, nei sistemi di videosorveglianza evoluti e nei sistemi di monitoraggio e sicurezza, oltre che in ogni sistema di archiviazione automatica di informazioni visive.

La libreria include attualmente più di 300 funzioni, che coprono le più svariate esigenze di trattamento di immagini, comprese funzioni matematiche ottimizzate (elevamento a potenza, logaritmi, conversioni cartesiane-polari, ecc.) ed un completo pacchetto di algebra matriciale, sviluppato funzionalmente al resto del sistema.

Informazioni pratiche

I siti principali del progetto sono due, il primo facente capo ad Intel (intel.com/technology/computing/opencv/) ed il secondo il classico hosting presso sourceforge (sourceforge.net/projects/opencvlibrary/) dal quale è possibile procedere al download sia dei sorgenti che degli installativi per le piattaforme Microsoft. A loro volta la maggior parte delle distribuzioni linux includono OpenCV tra i pacchetti disponibili.

La portabilità è quindi decisamente estesa, essendo disponibili versioni per tutte le varianti di Ms-Windows e per tutti i sistemi POSIX (Linux/BSD/UNIX/MacOsX), anche se il supporto ufficiale è, per ora, limitato ai primi. A questo proposito non ci si faccia spaventare dai numeri di versione (non ancora giunta alla 1.0) e dalla etichetta "beta", in quanto il livello di stabilità attualmente offerto assicura la possibilità di utilizzo anche all'interno di ambienti di produzione.

Il classico pacchetto installativo include, oltre ai binari ed agli header necessari alla compilazione, anche una discreta varietà di esempi, molto utili ad apprendere la sintassi e le tecniche di base. Sono inoltre disponibili diversi documenti introduttivi, oltre ad una reference in formato HTML.

Per esigenze di spazio non ci dilungheremo sulle impostazioni di base necessarie ai differenti ambienti di sviluppo. Sottolineiamo comunque come il problema della compatibilità sia stato affrontato e risolto in modo particolarmente capillare, e quindi, tramite poche e semplici operazioni sarà possibile essere subito produttivi all'interno dei più comuni ambienti di sviluppo disponibili nel mondo Windows (VisualC++, Borland C++ Builder, l'ottimo DevC++, ecc.), Linux (gcc, Kdevelop, Eclipse, ecc.) e Unix in generale.

La libreria è divisa in alcuni binari distinti ("dll" o "so" a seconda dei sistemi), motivo per cui il cui linking completo non è sempre necessario:

- **CxCore.** È l'oggetto principale nonché l'unico strettamente indispensabile. Include infatti tutte le funzioni di inizializzazione delle strutture utilizzate, l'algebra lineare e le altre funzioni di base.
- **Cv e CvAux.** Includono praticamente tutte le funzioni di trattamento ed analisi, quindi il cuore delle OpenCv.
- **HighGui.** Include alcune comode funzioni GUI, come ad esempio la tipica finestra di display, oltre alle funzioni di salvataggio e caricamento immagini da file.
- **CvCam.** Include funzioni di acquisizione video e di gestione delle telecamere (per il momento ancora piuttosto limitate).

Le funzioni più comuni

Cominciamo la nostra panoramica con un esempio di codice, allo scopo di dimostrare la buona leggibilità della sintassi utilizzata.

```
#include <stdio.h>
#include "cv.h"
#include "highgui.h"
int main(int argc, char *argv[])
```

```
{
    // carica una prima immagine da disco
    IplImage* img0 = cvLoadImage( "prova.jpg", -1 );
    // crea una seconda immagine con dimensioni trasposte
    IplImage* img1 = cvCreateImage(cvSize(img0->height,img0->width)
        ,img0->depth
        ,img0->nChannels);
    // traspone la prima immagine nella seconda
    cvTranspose(img0,img1);
    // crea una finestra display
    cvNamedWindow( "image1", CV_WINDOW_AUTOSIZE );
    // visualizza l'immagine trasposta
    cvShowImage( "image1", img1 );
    // attende la pressione di un tasto
    cvWaitKey(0);
    // libera le risorse
    cvReleaseImage( &img0 );
    cvReleaseImage( &img1 );
    return(0);
}
```

Si noti in particolare l'istanza automatica al caricamento dell'immagine da disco (`cvLoadImage`) e ancora il dimensionamento automatico della finestra di display.

Come si vede tutte le funzioni della libreria sono accomunate dal prefisso "cv", mentre i nomi delle classi utilizzate hanno prefisso "Cv" (con la C maiuscola), ad eccezione della `IplImage`, ereditata evidentemente dalla libreria IPL. Per restare in tema di convenzioni facciamo notare una piccola ambiguità: in alcuni casi le dimensioni seguono la convenzione XY (larghezza-altezza) più comune nella grafica, mentre in altri invece l'ordine RC (righe-colonne), questo perchè la maggior parte delle operazioni sono applicabili sia ad immagini che ad arbitrarie matrici numeriche. In questi casi la documentazione si riferisce sempre ad un virtuale oggetto array (`CvArr`), applicabile ad entrambe le categorie (immagini o array multidimensionali).

A proposito di oggetti disponibili, oltre ai già visti `CvMat` (matrici numeriche) e `IplImage` (buffer immagine), ed alle strutture di base indispensabili (`size`, `point`, `rect`, ecc) sono disponibili alcune comode strutture dinamiche quali buffer dinamici a contenuto arbitrario (`CvMemStorage`), liste dinamiche (`CvSeq`), grafi e alberi, con le necessarie funzioni di gestione.

Nel nostro esempio abbiamo visto l'applicazione di una trasposizione, inutile dire che nella libreria sono presenti tutte le più comuni funzioni di trasformazione geometrica (rotazione, ridimensionamento, LUT, ecc.).

È inoltre disponibile un completo ventaglio di funzioni di disegno diretto: linee, ellissi, poligonali, testo, ecc.

Più interessanti le funzioni di accesso diretto ai buffer di memoria, ottimizzate dal punto di vista delle prestazioni. Anche in virtù della chiarezza dei formati di memorizzazione interna tali funzioni permettono un facile scambio dati con oggetti provenienti da altre librerie. Non è infatti infrequente la necessità di ricorrere a librerie dedicate, specifiche per la gestione l'hardware adottato, ad esempio per l'acquisizione da digitalizzatori, dovendo poi passare i buffer immagine a funzioni OpenCV. In tali casi la chiarezza dei formati permette, ricorrendo a qualche wrapper scritto ad-hoc, un

facile interscambio di dati tra le differenti librerie.

Chiudiamo la panoramica sulle funzionalità elementari citando la possibilità di sfruttare praticamente in ogni trasformazione ROI (Region Of Interest) e COI (Channel Of Interest). Tramite apposite funzioni è possibile cioè limitare l'azione delle successive elaborazioni ad un canale colore (COI) o ad una zona dell'immagine (ROI), ottenendo quindi l'effetto di una mascheratura, senza necessità di effettuare costose operazioni di copia.

L'uso dell'aritmetica

Come già accennato le librerie includono tra le funzionalità di base (cioè nella libreria `cxcore`) un completo set di funzioni di algebra matriciale, che risulta estremamente prezioso nel trattamento immagini (che sono ovviamente trattabili come ogni altra matrice). La **Figura 1** illustra ad esempio come una semplice differenza in valore assoluto, applicata punto per punto, possa quantificare la perdita di informazione causata da una compressione jpeg. In particolare il valor medio dell'immagine ottenuta offre un semplice parametro oggettivo complessivo, utilizzabile nel caso in cui sia necessaria una valutazione automatica.

Le operazioni algebriche sono inoltre ampiamente utilizzate come base nelle analisi di movimento (motion detection). In particolare la media (o l'accumulo progressivo delle parti equivalenti) tra fotogrammi successivi può essere proficuamente utilizzata per acquisire lo sfondo all'interno di una scena contenente molti oggetti con movimenti irregolari, come ad esempio nell'inquadratura di una porzione di strada trafficata. Lo sfondo così identificato è poi facilmente eliminato dalla scena (tramite differenza), permettendo così di restringere l'informazione analizzata ai soli oggetti in movimento, ad uso, per esempio, di un algoritmo di inseguimento o di tracciamento del movimento (object tracking).

Analogamente è facile intuire come una semplice differenza sia spesso sufficiente all'identificazione di oggetti "nuovi" all'interno di una inquadratura fissa (tipico problema della videosorveglianza).

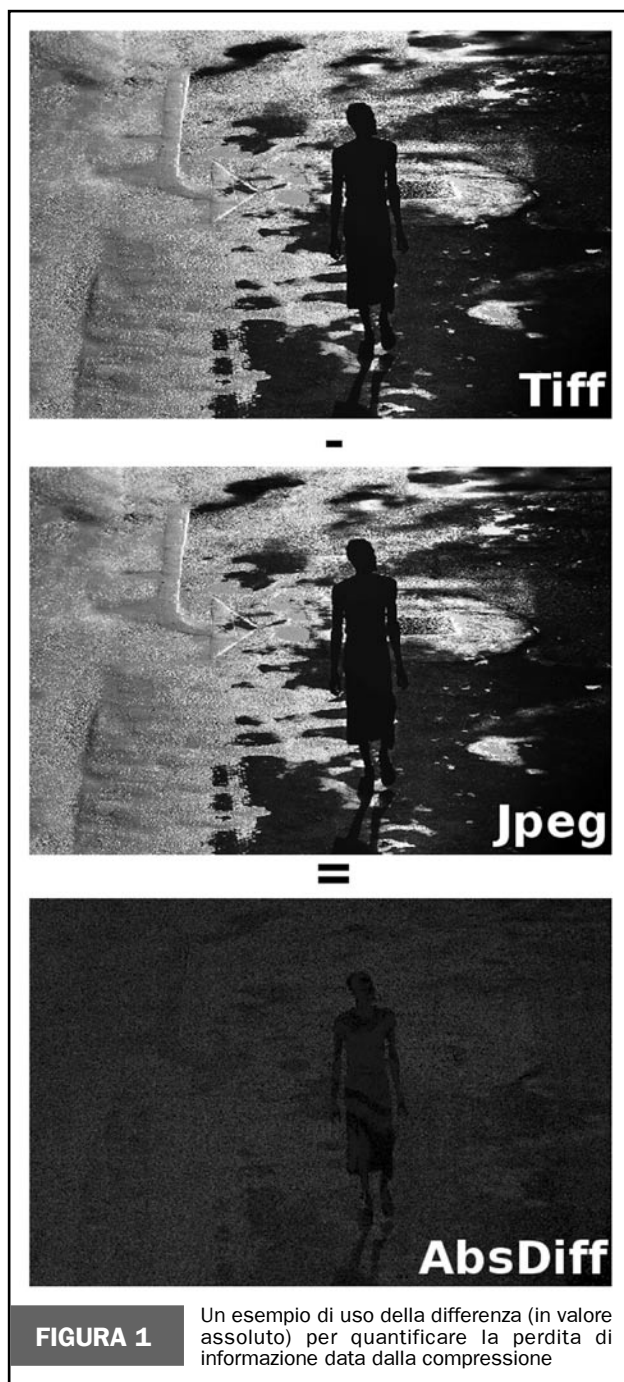
La differenza viene inoltre utilizzata come base per gli algoritmi di compressione video (h26x e simili), nei quali il processo prevede di codificare solo le porzioni di immagine differenti rispetto al fotogramma precedente, ottenendo immediatamente un grande guadagno di banda nel caso di inquadrature statiche, o solo parzialmente varianti.

Vedremo più avanti come la libreria includa, per tutti gli scopi accennati, anche funzioni molto più raffinate.

Operazioni morfologiche e denoising

Spesso il trattamento viene facilitato da una leggera perdita di definizione dell'immagine, anche se questo (è il caso di dire "a prima vista") può apparire al nostro occhio come una semplice sfocatura. Ciò non deve sorprendere, considerato che praticamente ogni trattamento delle immagini ha in effetti lo scopo di filtrare l'informazione utile, eliminando tutta l'informazione non funzionale al risultato cercato.

A questo scopo OpenCV mette a disposizione un ven-



taglio completo di funzioni di smoothing, che vanno dal semplice blur (sfocatura) a dimensione variabile, fino al filtro gaussiano. Similmente è disponibile la più completa gamma di filtri di soglia (binaria, passa-basso, passa-alto, ecc) anche adattativi.

Molto utilizzate nella semplificazione degli elementi di un'immagine sono anche le classiche operazioni morfologiche di apertura, chiusura, erosione, dilatazione. L'applicazione di tali filtri tende ad eliminare i dettagli più piccoli di una immagine senza modificare le caratteristiche essenziali di forma, ma semplificando l'immagine ad uso di successive analisi geometriche. In **Figura 2** è possibile notare come un'operazione di chiusura elimini da un'immagine binarizzata tutti i dettagli minori, aumentando la

leggibilità delle forme principali e diminuendo così lo sforzo computazionale di eventuali successive elaborazioni.

Naturalmente queste operazioni vengono utilizzate proficuamente anche per ridurre disturbi, piccole imprecisioni o rumore (denoising). Per utilizzi simili (downsampling e riduzione del rumore) è presente anche un'implementazione del metodo della piramide gaussiana.

Molto interessanti infine, sempre a proposito di correzione, varie funzioni di calibrazione e successivo recupero della distorsione ottica provocata dagli inevitabili difetti delle lenti di ripresa.

Cenni alle funzioni più avanzate

L'esplorazione delle funzioni messe a disposizione dalla libreria cv, che, come già visto, è il core del pacchetto OpenCV, riserva ancora molte sorprese. In particolare sono immediatamente disponibili, ed ottimizzati nelle prestazioni, molti degli algoritmi più raffinati oggi disponibili. Naturalmente la comprensione di molte delle funzionalità, richiede uno studio approfondito, impossibile da affrontare in poche righe. Ci limiteremo quindi ad accennare qualcuno degli algoritmi più utilizzati.

Oltre a quanto già visto, uno dei problemi più comuni affrontati nel trattamento immagini riguarda la delimitazione di oggetti, o in generale la loro identificazione, a scopo di

misura, di conteggio o di identificazione, come ad esempio nel riconoscimento di caratteri (OCR) oppure nella trasformazione di immagini da raster a vettoriali. La **Figura 3** presenta un esempio di applicazione dell'algoritmo di Canny per l'identificazione degli spigoli nell'immagine, ovvero delle zone di confine tra aree differenti. Tale algoritmo è attualmente considerato uno dei migliori disponibili allo scopo, anche considerando la facilità di messa a punto dei parametri, fattore spesso primario nell'utilizzo di trattamenti raffinati che, se da un lato permettono di arrivare ad isolare con chiarezza l'informazione cercata, dall'altro richiedono spesso un notevole lavoro di affinamento dei parametri coinvolti. Oltre al metodo di Canny sono disponibili, per scopi analoghi, i filtri di Sobel e Laplace. Complementare a questi e ad altri metodi è presente una completa serie di funzioni di classificazione geometrica dei contorni, utili a trasformare in informazione numerica le immagini trattate.

Le funzioni di classificazione vengono utilizzate anche per un altro dei comuni problemi di analisi, ovvero la blob analisi, cioè l'identificazione (misura, conteggio, ecc.) di aree omogenee, procedimento molto utilizzato nell'analisi di immagini ottenute al di fuori del campo visibile. Questa funzionalità, pur non esplicitamente disponibile attraverso una funzione ad essa dedicata, è facilmente ottenibile tramite l'ottimo algoritmo di flood fill, che lavora in modo simile al "pennello magico" disponibile in tutti i programmi di



FIGURA 2

Soglia e successiva chiusura per isolare delle scritte (i caratteri sono birmani)

fotoritocco.

Sempre a proposito dell'identificazione di pattern geometrici è disponibile la trasformata di Hough che, molto utilizzata in astronomia, permette, tra le altre cose, l'identificazione di linee rette e più in generale di pattern regolari all'interno di un'immagine.

In materia di analisi di immagini in movimento (motion detection, object tracking, ecc.) sono presenti applicazioni degli algoritmi di Lucas & Kanade, e di Horn & Schunck, utili per il calcolo dei flussi ottici per la rilevazione del movimento, oltre che per il block matching, cioè per la rilevazione di elementi uguali in posizioni differenti tra immagini successive.

Ancora per quanto riguarda il pattern matching è presente un classificatore basato sul

metodo di Haar ed ottimizzato per il riconoscimento di volti umani, ma utilizzabile per il riconoscimento di oggetti arbitrari, dopo una adeguata istruzione.

Citiamo infine le implementazioni del filtro di Kalman e dell'algoritmo di condensation, utilizzati nella riduzione del rumore e nei problemi previsione.

Conclusioni

In questo articolo abbiamo offerto una breve panoramica sulle problematiche della visione artificiale ed in particolare sull'utilizzo delle librerie OpenCV come utile strumento, condiviso ed in crescita, orientato alla soluzione di molti dei problemi connessi a tale campo di studio. Naturalmente le funzioni disponibili sono molte di più di quante lo spazio a disposizione ci abbia concesso di illustrare.

Ad uso di quanti fossero interessati ad approfondire l'argomento rimandiamo, oltre che alla documentazione ufficiale, ad alcuni link di approfondimento:

- Esempi, documentazione e molti post sul gruppo di discussione Yahoo!, dedicato a OpenCV: <http://groups.yahoo.com/group/OpenCV/>
- Per una rapida comprensione dei vari processi di analisi tramite esercitazioni interattive: <http://www-dsp.elet.polimi.it/ispg/eti/laboratorio/sessioni/index.htm>
- The computer vision homepage (link ragionati, hardware, software, pubblicazioni, ecc.): <http://www.cs.cmu.edu/~cil/vision.html>
- Completo compendio didattico sulla computer vision dall'università di Edimburgo: <http://homepages.inf.ed.ac.uk/rbf/CVonline/>

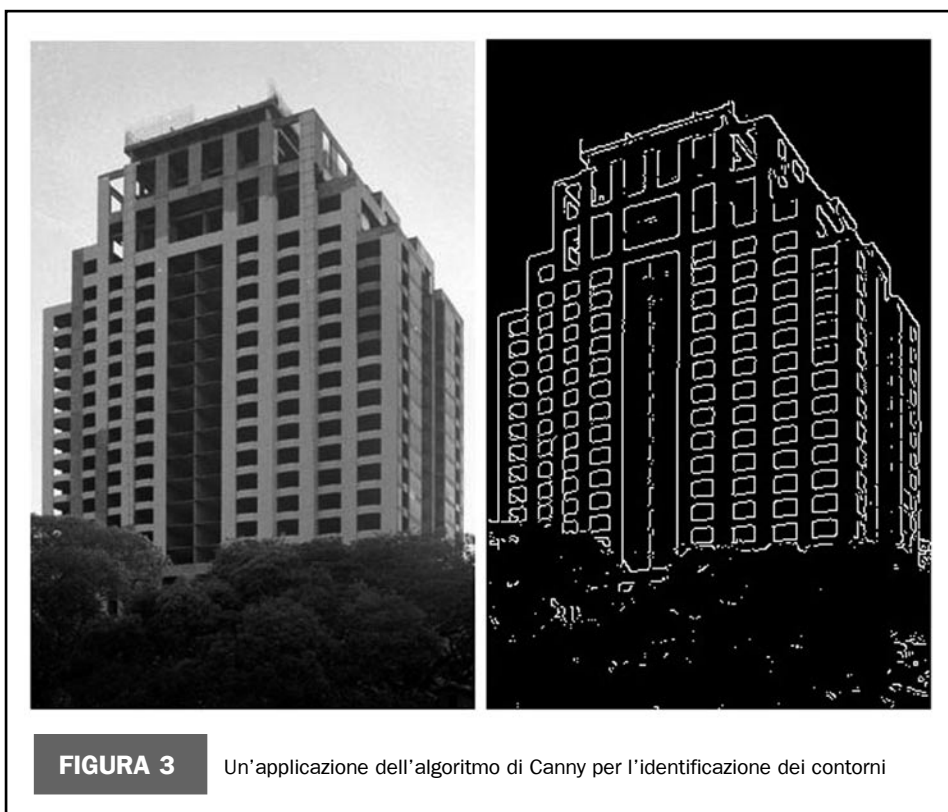


FIGURA 3

Un'applicazione dell'algoritmo di Canny per l'identificazione dei contorni

Un ultimo cenno relativamente alla "concorrenza": le librerie che integrano funzioni di analisi d'immagine sono molte, meno invece quelle dedicate specificatamente a questo scopo. La rosa si restringe poi di molto limitando le scelte a quanto liberamente utilizzabile e disponibile in forma di codice sorgente. Considerato inoltre la provenienza Intel e la completezza delle OpenCV non restano molti concorrenti con cui confrontarsi. Ciononostante va detto che un confronto serio non può prescindere dallo specifico campo applicativo, ed è possibile che, per una particolare applicazione, si ottengano migliori risultati da qualche altro prodotto, magari più limitato ma più focalizzato su una specifica funzione. Inutile sottolineare come ricorrere a diverse librerie per diverse funzioni non faciliti certo la vita del programmatore, e quindi come la completezza sia un requisito importante per un prodotto di questo tipo. La lungimirante scelta di Intel di rendere liberamente disponibile il codice sorgente dovrebbe inoltre assicurare nel futuro delle OpenCv una continua e costante evoluzione.

Matteo Lucarelli

Ingegnere, lavora come consulente nella progettazione e prototipazione di sistemi innovativi di monitoraggio e sicurezza. Quando possibile preferisce lavorare con strumenti open source (e rilasciare il codice sotto GPL). Le immagini, base delle elaborazioni che illustrano questo articolo, sono state gentilmente fornite dall'amico fotogiornalista Roberto Giussani.