Applicazioni pratiche della visione artificiale

Parte 1.2 – OpenCV

Curato da: Ing. Francesco La Rosa
Università di Messina – Facoltà di Ingegneria

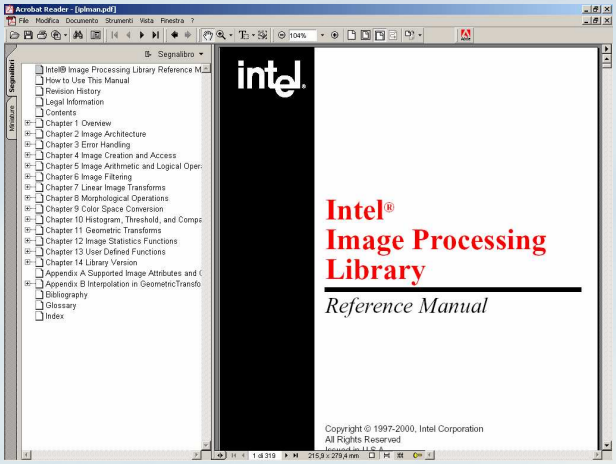
Corso di Calcolatori II – A.A. 2003/2004
Ing. Giancarlo Iannizzotto

Manuali

All'interno del materiale fornito per l'installazione
si possono trovare i seguenti file di documentazione:

- iplman.pdf
- Reference Manual (formato html, ...\opencv\docs\index.html)
- Cvcam.rtf (documento sulla libreria cvcam, ...\opencv\docs\cvcam.rtf)
- Opencv-0_9_5.pdf (manuale di OpenCV3.1 in formato pdf)
- Opencvman_old.pdf (manuale di OpenCV aggiornato alla versione 2.1)

Iplman



Iplman

Il manuale fornisce conoscenze di base sull'architettura della libreria IPL (Image Processing Library) ed una descrizione dettagliata delle funzioni in essa contenute.

Le funzioni sono raccolte in gruppi in base alla loro funzionalità.

Ogni gruppo è descritto in un capitolo differente (dal capitolo 3 al capitolo 14).

Iplman

Il manuale delle IPL è costituito da 14 capitoli:

ø Capitolo 1: "Overview"

Fornisce informazioni riguardo all'organizzazione del manuale e alla notazione adottata.

ø Capitolo 2 "Image Architecture"

Descrive l'architettura di supporto alle immagini (color models, data types, etc.) ed image tiling ("Tiling is a method of image representation in which the image is broken up into smaller images, or tiles, to allow for complicated memory management schemes").

Iplman

øCapitolo 3 "Error Handling"

Fornisce informazioni sulle funzioni di gestione degli errori incluse nella libreria. Sono anche descritti gestori degli errori definibili dall'utente.

øCapitolo 4: "Image Creation and Access"

Descrive le funzioni usate per: creare, modificare ed accedere ad attributi dell'immagine; modificare il bordo dell'immagine ed effettuare un tiling della stessa; allocare memoria per tipi differenti di dati. Il capitolo descrive anche funzioni che facilitano l'esecuzione di operazioni in ambienti a finestre.

Iplman

- Capitolo 5: "Image Arithmetic and Logical Operations"

  Descrive operazioni di image processing che modificano il valore dei pixel usando semplici operazioni logiche o aritmetiche.

- Capitolo 6: "Image Filtering"

  Descrive operazioni di filtraggio lineari o non-lineari che possono essere applicate alle immagini.

Iplman

- Capitolo 7: "Linear Image Transforms"

  Descrive la Fast Fourier Transform (FFT) e la Discrete Cosine Transform (DCT) implementate nella libreria.

- Capitolo 8: "Morphological Operations"

  Descrive le funzioni che eseguono le operazioni di erosion, dilation, e ulteriori combinazioni delle prime.

Iplman

- Capitolo 9 : "Color Space Conversion"

  Descrive le possibili conversioni di colore supportate dalla libreria .

- Chapter 10 : "Histogram, Threshold, and Compare Functions"

  Descrive funzioni che implementano operazioni utili al trattamento delle immagini quali: contrast stretching, histogram computation, histogram equalization, thresholding.

Iplman

- Capitolo 11 "Image Geometric Transforms"

  Descrive le trasformazioni geometriche supportate: resizing, flipping, rotation e vari tipi di warping (Image warping = rearranging the pixels of a picture).

- Capitolo 12 "Image Statistics Functions"

  Descrive funzioni che permettono di calcolare la norma, i momenti, il valore massimo e quello minimo di un'immagine.

Iplman

- Chapter 13 : "User-Defined Functions"

  Descrive le funzioni di libreria che abilitano l'utente alla creazione di proprie funzioni di image processing.

- Chapter 14 "Library Version"

  Descrive la funzione iplGetLibVersion() che ritorna la versione (ed ulteriori informazioni) della libreria in uso.

OpenCV - Reference Manual

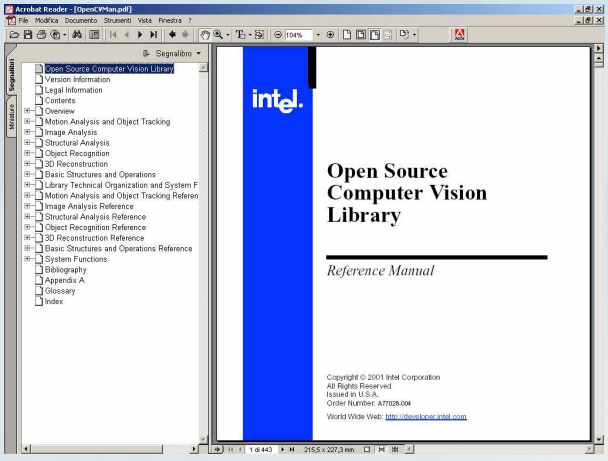
Il Reference Manual delle OpenCV è diviso nei seguenti capitoli:

- Basic Structures and Operations Reference
- Image Processing and Analysis Reference
- Structural Analysis Reference
- Motion Analysis and Object Tracking Reference
- Object Recognition Reference

OpenCV - Reference Manual

- Camera Calibration and 3D Reconstruction Reference
- Experimental Functionality
- GUI and Video Acquisition Reference
- Bibliography
- CVcam manual (RTF)

OpenCVman_old



OpenCVman_old

Il manuale delle OpenCV_old può essere diviso in due parti principali:

ø Programmer Guide;

ø Reference.

Nel manuale sono inclusi alcuni programmi d'esempio utili per familiarizzare con le funzioni di libreria.

OpenCVman

Programmer Guide

I concetti teorici che stanno alla base dei componenti implementati all'interno della libreria sono, in linea di massima, spiegati in Programmer Guide.

OpenCVman_old

Reference

La Reference non è del tutto attendibile visto che non è aggiornata alla versione 3.1 (si ferma alla 2.1).

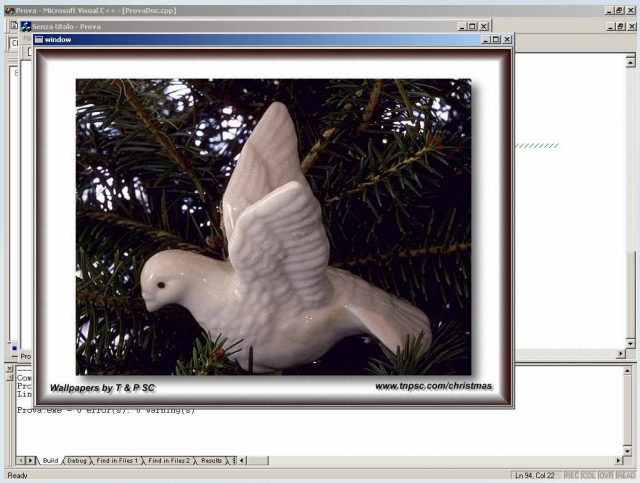
Loading di una Bitmap

```
void CProvaDoc::OnEseguiAcquisisci()
{       IplImage *src;// dichiaro un'IplImage (struttura dati
                      // usata per immagini 2D)
        char path[]="C\\colombina.bmp";
        //creo una finestra di nome "window"
        cvNamedWindow("window",CV_WINDOW_AUTOSIZE);
        src=cvLoadImage(path); //carico l'immagine
                               //contenuta nel file colombina.bmp
                               //in  "src" (la memoria  per l'immagine è
                               //allocata in modo implicito)
        cvShowImage("window",src); //mostro nella finestra
                             // "window" l'immagine contenuta in "src"
        …
```

Loading di una Bitmap

```
…
        cvWaitKey(0); //blocco l'esecuzione del programma
                      //fino a quando non viene premuto un
                      //tasto qualsiasi
        cvDestroyWindow("window"); //elimino la finestra
                                   //dallo schermo
        cvReleaseImage(&src);//rilascio la memoria allocata
                             //per "src"
}
```


Loading di una Bitmap



Loading di una Bitmap 2

```
void CProvaDoc::OnAcquisisciOpen()
{       IplImage *src;// dichiaro un'IplImage (struttura dati
                      // usata per immagini 2D)
        char path[256];
        //Inizializzazione di path
        memset(path,0,256);
        //scelgo il file da aprire attraverso la finestra open-file
        OPENFILENAME fn;
        fn.lStructSize = sizeof (OPENFILENAME);
        fn.hwndOwner = NULL;
        fn.lpstrFilter = NULL;
        fn.lpstrFile = path;
        fn.nMaxFile = 256;
        …
```


Loading di una Bitmap 2

```
…
        fn.lpstrFileTitle = NULL;
        fn.lpstrInitialDir = NULL;
        fn.lpstrTitle = NULL;
        fn.Flags = NULL;
        fn.lpstrDefExt = "bmp";
        fn.hInstance = DLLhinst;
        fn.lpfnHook = NULL;
        fn.lpstrCustomFilter = NULL;
        fn.lCustData = NULL;
        if(!GetOpenFileName(&fn))
                return ;
…
```

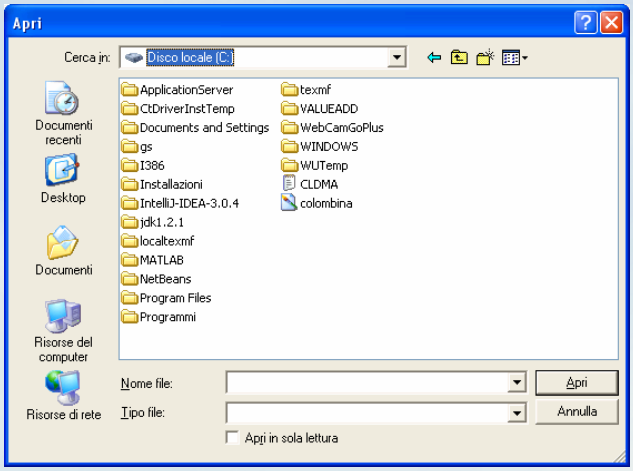
Loading di una Bitmap 2

```
…
        cvNamedWindow("window",CV_WINDOW_AUTOSIZE);
        src=cvLoadImage(path); //carico l'immagine contenuta
                               //nel file selezionato tramite la dialog box
                               //open-file)
        cvShowImage("window",src); //mostro nella finestra
                                   // "window" l'immagine contenuta in "src"
…
```


Loading di una Bitmap 2

```
…
        cvWaitKey(0); //blocco l'esecuzione del programma
                      //fino a quando non viene premuto un
                      //tasto qualsiasi
        cvDestroyWindow("window"); //elimino la finestra
                                   //dallo schermo
        cvReleaseImage(&src);//rilascio la memoria allocata
                             //per "src"
}
```

Loading di una Bitmap 2



Color2Gray

```
void CProvaDoc::OnEseguiColor2gray()
{       IplImage *src, *src_gray;
        char path[]="C\\colombina.bmp";
        src=cvLoadImage(path); //carico l'immagine
                               //contenuta nel file colombina.bmp
                               //in  "src"
        CvSize size_src=cvSize(src->width,src->height);
                               //creo la variabile "size_src" di
                               //tipo "CvSize" per contenere
                               //larghezza ed altezza dell'immagine
        …
```

Color2Gray

```
…
src_gray=cvCreateImage(size_src,src->depth,1);
                    //creo l'immagine src_gray con
                    //le dimensioni fissate da size_src
                    //e con un solo canale
iplColorToGray(src,src_gray);
                   //passo "src" convertita a toni di grigio
                   // nella variabile "src_gray"
cvNamedWindow("window color",CV_WINDOW_AUTOSIZE);
// la costante "CV_WINDOW_AUTOSIZE" richiede che
// la window si adatti alle dimensioni dell'immagine
cvShowImage("window color",src);
cvNamedWindow("window gray",1); …
```


Color2Gray

```
        …
        cvShowImage("window gray",src_gray);
        cvWaitKey(0);
        //chiusura delle finestre
        cvDestroyWindow("window color");
        cvDestroyWindow("window gray");
        //rilascio della memoria allocata
        cvReleaseImage (&src);
        cvReleaseImage (&src_gray);
}
```

Color2Gray

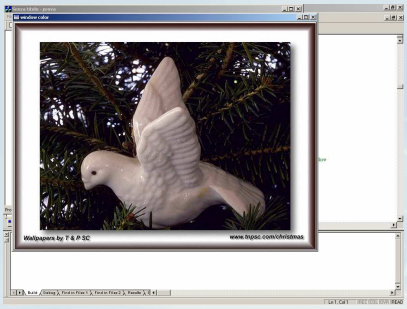


Immagine a colori

Immagine a toni di grigio

cvCvtColor

- Una soluzione alternativa alla funzione iplColorToGray per la conversione di un'immagine a colori in una a toni di grigio è data dalla funzione di OpenCV *cvCvtColor*.
- La funzione *cvCvtColor* consente, in generale, la conversione di un'immagine da uno spazio di colore ad un altro. Diversi sono gli spazi di colore attualmente gestiti dalla funzione (RGB, HSV, …).

cvCvtColor

void cvCvtColor(const CvArr* src, CvArr* dst, int code);

Parametri della funzione:

Src

The source 8-bit image.

dst

The destination 8-bit image.

code

Color conversion operation that can be specifed using CV_<src_color_space>2<dst_color_space> constants.

Note:

Per un elenco degli spazi di colore supportati dalla funzione cvCvtColor consulta il Reference Manual.

Thresholding di un'immagine

```
void CProvaDoc::OnEseguiColor2gray()
{       IplImage *src, *src_gray, *dst;
        char path[]="C\\colombina.bmp";
        //load dell'immagine
        src=cvLoadImage(path);
        CvSize size_src=cvSize(src->width,src->height);
        src_gray=cvCreateImage(size_src,src->depth,1);
        //conversione Color2Gray
        iplColorToGray(src,src_gray);
        dst=cvCreateImage(size_src,src->depth,1);
```

Thresholding di un'immagine

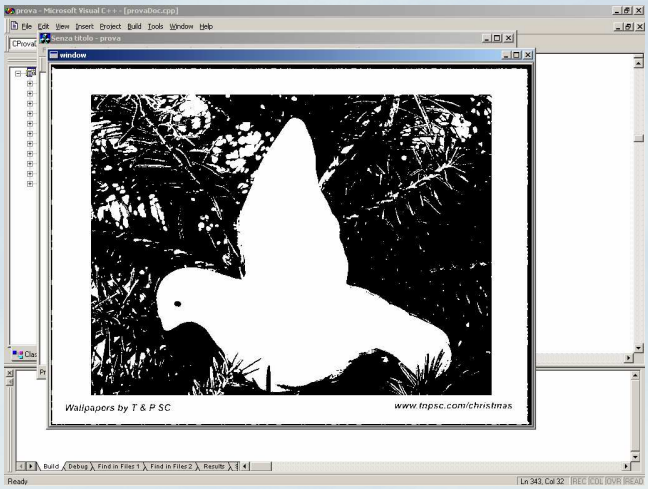
```
…
int threshold=127; //fisso il valore di threshold
//operazione di thresholding
iplThreshold(src_gray, dst,threshold);
cvNamedWindow("window",1);
//mostro a video l'immagine sogliata
cvShowImage("window",dst);
cvWaitKey(0);
cvDestroyWindow("window");
…
```


Thresholding di un'immagine

```
…
//rilascio della memoria allocata
cvReleaseImage (&src);
cvReleaseImage (&src_gray);
cvReleaseImage (&dst);
}
```

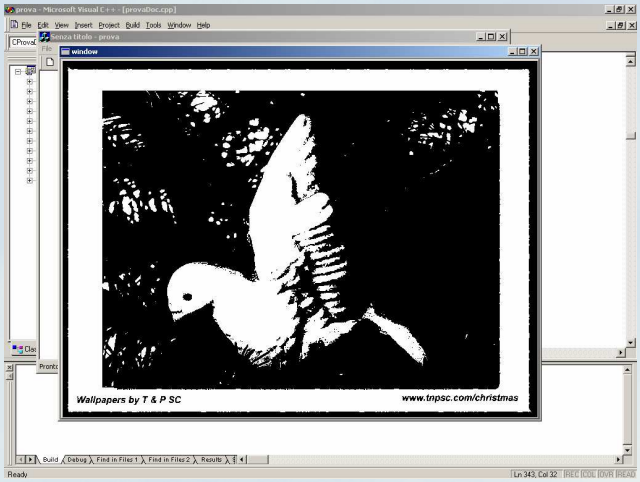
Thresholding di un'immagine

Thresholding di un'immagine



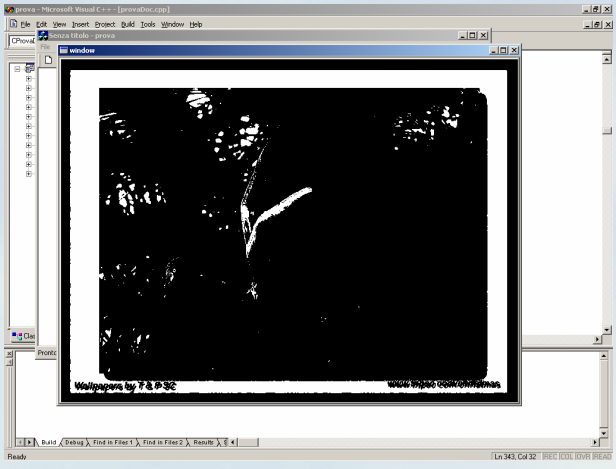
Threshold=50

Thresholding di un'immagine



Threshold=127

Thresholding di un'immagine



Threshold=200

SetRoi

```
void CProvaDoc::OnSetRoi()
{   IplImage *src; // ROI= Region of Interest
    char path[]="C\\colombina.bmp";
    src=cvLoadImage(path);
    cvNamedWindow("window color",1);
    cvShowImage("window color ",src);
   //dichiarazione ed inizializzazione dei parametri necessari
   //per il setting della ROI.
    int X=100, Y=100, width=400, height=400;
    …
```

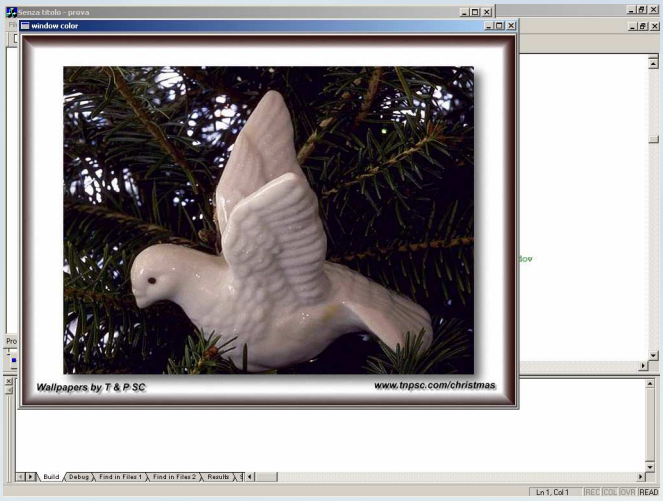

SetRoi

```
…
CvRect  roi_rect=cvRect(X, Y, width, height);
cvSetImageROI (src,roi_rect); //setting della ROI
cvNamedWindow("window set_roi",1);
cvShowImage("window set_roi",src);
cvWaitKey(0);
…
```

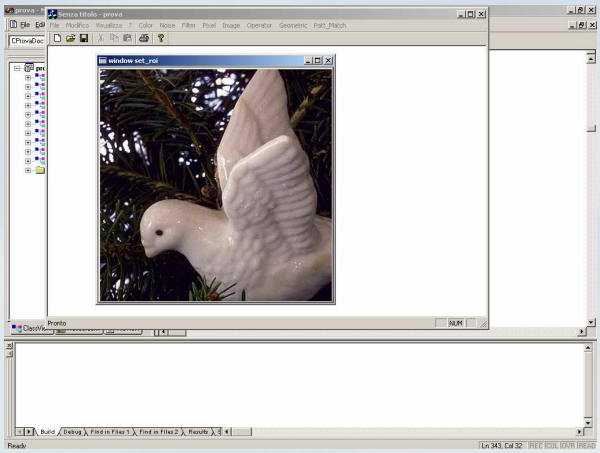
SetRoi

```
…
//chiusura delle finestre
cvDestroyWindow("window color");
cvDestroyWindow("window set_roi");
//rilascio la memoria allocata per le immagini
cvReleaseImage (&src);
}
```


SetRoi



SetRoi



ROI Image

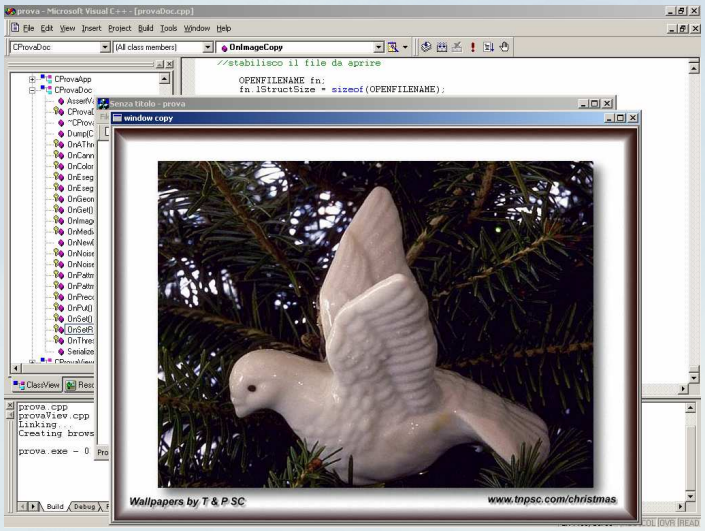
Copy

```
void CProvaDoc::OnImageCopy()
{   IplImage *src, *dst;
    char path[]="C\\colombina.bmp";
    src=cvLoadImage(path);
    CvSize size_src=cvSize( src->width,src->height);
    dst=cvCreateImage(size_src,src->depth,src->nChannels);
    cvCopyImage (src,dst); //copia di "src" in "dst"
    …
```

Copy

```
…
cvNamedWindow("window copy",1);
cvShowImage("window copy",dst);
cvWaitKey(0);
//chiusura della finestra
cvDestroyWindow("window copy");
//rilascio la memoria allocata per le immagini
cvReleaseImage (&src);
cvReleaseImage (&dst);
}
```


Copy



Pix2Plane

```
void CProvaDoc::OnColorPix2plane()
{   IplImage *src;
    IplImage *PlaneR, *PlaneG, *PlaneB;
    char path[]="C\\pix2plane.bmp";
    src=cvLoadImage(path);
    CvSize size_src=cvSize( src->width,src->height);
    //creo un'immagine per ogni piano di colore
    PlaneR=cvCreateImage(size_src,IPL_DEPTH_8U,1);
    PlaneG=cvCreateImage(size_src,IPL_DEPTH_8U,1);
    PlaneB=cvCreateImage(size_src,IPL_DEPTH_8U,1);
    // splitto un' immagine a colori nei suoi piani componenti
    cvCvtPixToPlane(src, PlaneB, PlaneG, PlaneR,0); //0=NULL
    …
```

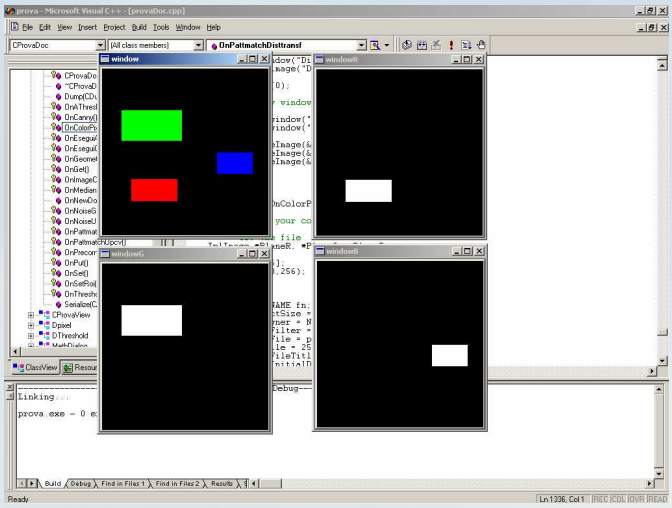

Pix2Plane

```
…
cvNamedWindow("window",1);
cvNamedWindow("windowR",1);
cvNamedWindow("windowG",1);
cvNamedWindow("windowB",1);
//mostro l'immagine a colori ed i
//canali R, G e B.
cvShowImage("window",src);
cvShowImage("windowR",PlaneR);
cvShowImage("windowG",PlaneG);
cvShowImage("windowB",PlaneB);
…
```

Pix2Plane

```
…
//chiusura delle finestre
cvDestroyWindow("window");
cvDestroyWindow(" windowR ");
cvDestroyWindow(" windowG");
cvDestroyWindow(" windowB");
//rilascio la memoria allocata per le immagini
cvReleaseImage (&src);
cvReleaseImage (&PlaneR);
cvReleaseImage (&PlaneG);
cvReleaseImage (&PlaneB);
}
```

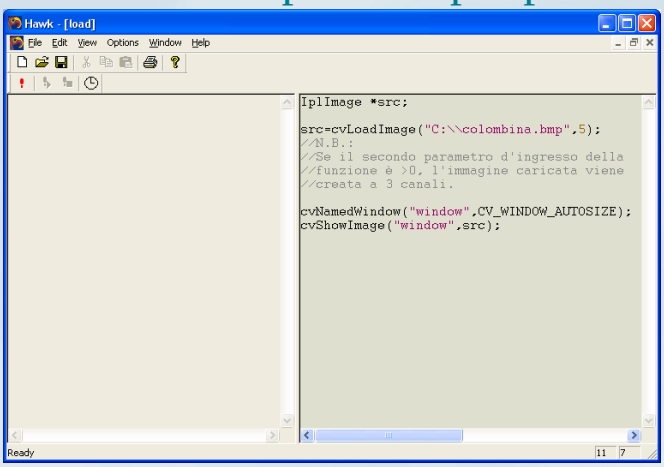

Pix2Plane



Hawk – un interprete … per provare

- The Hawk application has been created for fast demo creation and for research purpose. It represents to be a C interpreter (compatible with ISO C), with built in functions of OpenCV, IPL and interface functions to Win32.
- Interprete:
  - …\OpenCV\bin\Hawk\Release\Hawk.exe
- Tutorial:
  - …\OpenCV\apps\Hawk\Hawk.pdf

Hawk – un interprete … per provare



Un esempio in cui viene caricata e visualizzata una bmp.

"!" per eseguire lo *script*.