

Algoritmi per l'analisi delle immagini

Algoritmi per l'Analisi delle Immagini

VisAlgo

AB&T Tecnologie Informatiche™
Via dell'About, 2/A • 10015 Ivrea
Fax 0125 234397
www.bausano.net
info@bausano.net

Informazioni legali

Le informazioni con questo documento, incluse URL e gli altri riferimenti sul sito Internet, possono cambiare senza alcun avvertimento.

A meno di specifica annotazione, i riferimenti a compagnie, organizzazioni, prodotti, persone ed eventi sono fittizi e non associate con reali compagnie, organizzazioni, prodotti, persone ed eventi.

L'AB&T Tecnologie Informatiche™ può registrare, licenziare, richiedere il copyright o marchi e rivendicare la proprietà intellettuale a tutti gli argomenti trattati in questo documento.

Senza limitare i diritti sotto copyright, nessuna parte di questo documento può essere riprodotta, o modificata o trasmessa sotto ogni forma o mezzo (elettronico, meccanico, per fotocopiare, per registrare, od altro), senza l'espressa autorizzazione dell' AB&T Tecnologie Informatiche™.

Eccetto che per accordi scritti con la AB&T Tecnologie Informatiche™ la fornitura di questo documento non autorizza nessuno alla registrazione, a dare licenze, a richiedere il copyright o marchi e a rivendicare la proprietà intellettuale agli argomenti trattati in questo documento.

AxesBrain™, VisAlgo™, ScadaMERCURIO™ sono marchi registrati

©AxesBrain, ©VisAlgo sono tutelati da copyright

ActiveX, DirectX, JScript, Microsoft, Microsoft Press, MS-DOS, Visual Basic, Visual C++, Win32, Win32s, Windows, WDM, Windows NT, Windows 2000 e Windows Me sono o prodotti o marchi registrati dalla Microsoft Corporation negli Stati Uniti e/o negli altri Paesi. I nomi di compagnie e prodotti citati in questo documento possono essere marchi registrati dai loro proprietari.

Sommario

Introduzione 1

CAPITOLO 1 **- Open Close -**

OpenCGVSBASE 4

CloseGVSBASE 10

CAPITOLO 2 **- Window parameter -**

SetActiveCamera 14

SetWindowParameter 15

ChangeWindowParameter 17

ToXRegion 18

ToYRegion 19

GetImageWindowBorder 20

CAPITOLO 3 **- Picture on Window -**

LoadBmpImage 22

SaveBmpImage 23

PutImageOnWindow 24

CAPITOLO 4 **- Pixel Management -**

SetFilterOnRegion 28

SetBrightnessOnRegion 29

SetContrastOnRegion 30

IntegrationOfRegion 31

CAPITOLO 5

- Pixel Value -

GetPixelValue	34
GetAveragePixelsOfRegion	35
GetAveragePixelsCircleRegion	36
GetAveragePixelsRectRegion	37
GetPixelMaxLuxRegion	38
LoadIstogramRegion	39
DrawIstogram	41
GetBackGroundLimits	43

CAPITOLO 6

- Blobs -

GetAllBlobs	46
RemoveAllBlobs	49
GetValueBlob	50
GetDimensionBlob	51
GetCircleBlob	54
GetEllipseBlob	55
GetXAxesRectBlob	56
GetYAxesRectBlob	59
GetAllPixelsBlob	62
GetEdgeBlob	63
DrawBlob	64

CAPITOLO 7

- Intersected Shapes -

GetAllIntersectedShapes	68
RemoveAllIntersectedShapes	71
GetDimensionIntersectedShape	72
GetEdgeIntersectedShape	74
DrawIntersectedShape	75

CAPITOLO 8

- Edge -

GetBestEdgeLineOnRegion	78
GetBestEdgeCircleOnRegion	80
PointsFromBestEdgeLineAcross EdgeOnRegion	82
MaximusFromBestEdgeLine AcrossEdgeOnRegion	84
PointsFromLine2PointAcross EdgeOnRegion	86
MaximusFromLine2PointAcross EdgeOnRegion	88

CAPITOLO 9

- Elements management -

PointsAcross2Elements	91
PointFromLineAcrossLine	92
Points2FromLineAcrossRegion	94
DrawElementLine	96
DrawLine	97
DrawElementCircle	98
DrawTextOnVideo	99
Draw2ElementsLine	100
SubPixelPosition	101

CAPITOLO 10

- Edge locator -

LineEdgeLocator	102
GetEdgeByPosition	104
GetEdgeBySize	106
GetAllEdgeFind	108
GetAllSortEdgeFind	110
CircleByEdge	112
EllipseByEdge	114
FindAllConicsByEdge	116
GetValueEllipseConicByEdge	118

CAPITOLO 11 **- Fast Fourier -**

FFT2	102
iFFT2	104
SetZeroPartialFFT2	106
LoadMemoryFFT2	108
closeFFT2	110
Filter_iFFT2	111
HighLowPassFilter_iFFT2	112
Multy_iFFT2	113
GetFirsts3FrequencyFFT2	114
ClearBelowModulFFT2	115

CAPITOLO 12 **- Estrazione di soglie e varie -**

GetThreshold	116
SetThreshold	117
SaveFileTiff	118
EdgeDetection	119
ConvertMultiThreshold	120
SetOffsetCenterImage	121
GetOffsetCenterImage	122
EdgeOnMaskRegion	123
GetXYAxesRectBlob	124

CAPITOLO 13 **- PatternMatching -**

PMFind	160
PMCreateTemplate	162
PMLoadTemplate	163
PMSaveTemplate	164
PMGetMatchingData	165
PMGetParameter	169
PMSetParameter	170
PMGetDefaultParameters	171
PMGetTemplatImage	172
PMDraw	174

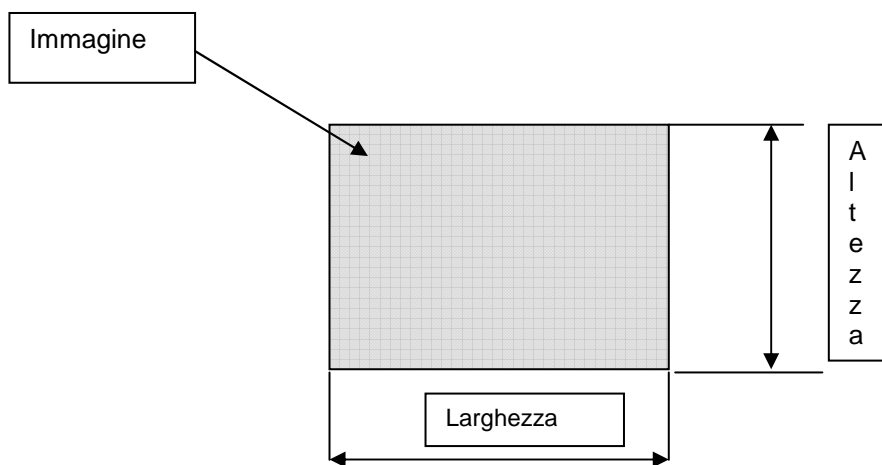
APPENDICE A
- Visual Basic -

Declare	176
Esempio	181

Algoritmi per l'analisi delle immagini

Gli algoritmi per l'analisi delle immagini lavorano su un'area di memoria precedentemente caricata.

Per effettuare l'analisi è necessario caricare l'immagine nell'area di memoria il cui puntatore viene allocato dalla funzione di apertura della libreria. L'area di memoria viene allocata dalla libreria di dimensioni richieste sia in larghezza che in altezza e deve essere caricata dall'utente prima di richiamare le funzioni di analisi



L'area è costituita da un numero di bytes corrispondente alla larghezza per l'altezza, ogni byte rappresenta il valore del livello di grigio di ogni singolo pixel. Il valore di grigio del pixel rappresenta la luce incidente, 0 nessuna luce (nero) 255 massimo valore di luce (bianco).

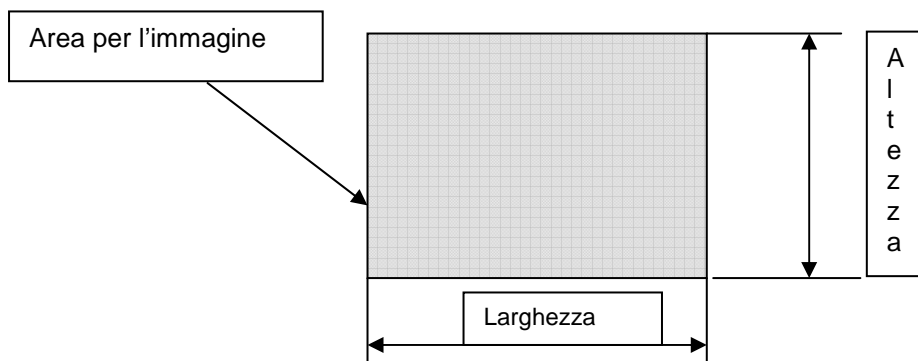
Open e close

Per poter utilizzare la libreria VisAlgo è necessario effettuare l'operazione di apertura del sistema, con le dimensioni dell'immagine, il sistema predispone un'area di memoria della dimensione opportuna e restituisce un puntatore attraverso il quale sarà possibile caricare l'immagine digitalizzata.

L'operazione di chiusura rilascia l'area preallocata.

- 1- **OpenCGVSBASE** Apertura della libreria
- 2- **CloseCGVSBASE** Chiusura della libreria

L'area ha una dimensione in byte equivalente della larghezza per altezza



OpenCGVSBASE

Aprire le funzioni della libreria VisAlgo

```
GVSBASEAPI int WINAPI OpenCGVSBASE (  
    int                iImageMaxX,  
    int                iImageMaxY,  
    unsigned char * *  pbMemoryArea,  
    int                iNumberCam,  
    int *              piHandle  
);
```

Valore di ritorno

1 OK
0 libreria già aperta

Parametri

int iImageMaxX	Dimensione in pixel della larghezza (X)
int iImageMaxY	Dimensione in pixel della altezza (Y)
unsigned char * * pbMemoryArea	Puntatore dell'area che la libreria ha predisposto.
int iNumberCam	NumeroCamera 0-15 (istanza di lavoro)
int * piHandle	Ritorna handle con cui richiamare le funzioni che lavorano su questa camera o istanza

Esempio: PX610

```
void GetImage( FRAMEHANDLE frh , char * pArea)
{
    int i;
    BYTE *p;

    p = pArea;
    for( i=ImageMaxY-1; i>=0; --i )
    {
        GetRow(frh, p, i);
        p += ImageMaxX;
    }
}

int          iImageMaxX;
int          iImageMaxY;
unsigned char * pbMemoryArea;
FRAMEHANDLE  frh[2][3];
FGHANDLE     fgh[2];
int          tagQ[2];

tagQ[0]=0;
tagQ[1]=0;
if (!InitLibrary())
{
    MessageBox(0, "InitLibrary Failed", "vision.exe", MB_OK);
    return FALSE;
}
fgh[0] = AllocateFG(-1);
if(!fgh)
{
    return FALSE;
}
ResetFG(fgh[0]);
iImageMaxX = 768;
iImageMaxY = 575;
SetImageSize(fgh[0], ImageMaxX, 256, 16 , 0, ImageMaxX , ImageMaxY , 8);
iBoardRev = ReadRevision(fgh[0]);
frh[0][0] = AllocateBuffer(ImageMaxX, ImageMaxY, 8);
if(!frh[0])
{
    MessageBox(0, "Allocate Buffer Failed on frh[0]", "vision.exe", MB_OK);
    return FALSE;
}
frh[1][0] = AllocateBuffer(ImageMaxX, ImageMaxY, 8);
if(!frh[1])
{
    MessageBox(0, "Allocate Buffer Failed on frh[1]", "vision.exe", MB_OK);
    return FALSE;
}
}
```

```
if(!AllocBuffer())
{
    MessageBox(0, "Failed to allocate scratch buffer", "vision.exe", MB_OK);
    return FALSE;
}
if ( !OpenCGVSBASE ( iImageMaxX, iImageMaxY, &pbMemoryArea ) )
{
    MessageBox(0, "Failed to buffer", "vision.exe", MB_OK);
    return FALSE;
}

.....
Grab( fgh[0], frh[0][0], 0 );
if(CheckError(fgh[0]))
{
    ++ErrorCount;
}
else
{
    GetImage(frh[0][0], pbMemoryArea );
    ++FrameCount;
}
.....
```

Esempio: PXC2000

```
void GetImage(FRAME __PX_FAR *frh , char * pArea )
{
    int i,len;
    BYTE __PX_HUGE *p;

    len=ImageMaxX*(((PIXEL_TYPE&0xFF)+7)>>3);
    p = pArea;
    for( i=ImageMaxY-1; i>=0 ; --i )
    {
        frame.GetRow(frh, p, i);
        p += len;
    }
}

#define PIXEL_TYPE          PBITS_Y8
PXC200                      pxc200;
FRAMELIB                    frame;
long                        len;
FRAME __PX_FAR *           frh[2];
long                        fgh[2];
int                          tagQ[2];

tagQ[0]=0;
tagQ[1]=0;
fgh[0] = 0;
if (!imagenation_OpenLibrary("pxc2_NT.dll",&pxc200,sizeof(pxc200)))
{
    return FALSE;
}
if (!imagenation_OpenLibrary("frame_32.dll",&frame,sizeof(frame)))
{
    return FALSE;
}
fgh[0] = pxc200.AllocateFG(-1);
if(!fgh[0])
{
    return FALSE;
}
```

```

switch(pxc200.VideoType(fgh[0]))
{
    case 1:
        ImageMaxX = 640;
        ImageMaxY = 485;
        pxc200.SetWidth(fgh[0],ImageMaxX);
        pxc200.SetHeight(fgh[0],ImageMaxY);
        pxc200.SetLeft(fgh[0],0);
        pxc200.SetTop(fgh[0],0);
        pxc200.SetXResolution(fgh[0],ImageMaxX);
        pxc200.SetYResolution(fgh[0],ImageMaxY);
        break;
    case 0:
    case 2:
        ImageMaxX = 768;
        ImageMaxY = 575;
        pxc200.SetWidth(fgh[0],ImageMaxX);
        pxc200.SetHeight(fgh[0],ImageMaxY);
        pxc200.SetLeft(fgh[0],0);
        pxc200.SetTop(fgh[0],0);
        pxc200.SetXResolution(fgh[0],ImageMaxX);
        pxc200.SetYResolution(fgh[0],ImageMaxY);
        break;
}
iBoardRev = pxc200.ReadRevision(fgh[0]);
frh[0] = pxc200.AllocateBuffer(ImageMaxX, ImageMaxY, PIXEL_TYPE);
if(!frh[0])
{
    MessageBox(0, "Allocate Buffer Failed on frh[0]", "PX200", MB_OK);
    return FALSE;
}
frh[1] = pxc200.AllocateBuffer(ImageMaxX, ImageMaxY, PIXEL_TYPE);
if(!frh[1])
{
    MessageBox(0, "Allocate Buffer Failed on frh[1]", "PX200", MB_OK);
    return FALSE;
}
}

```

```
if ( !OpenCGVSBASE ( iImageMaxX, iImageMaxY, &pbMemoryArea ) )
{
    MessageBox(0, "Failed to buffer", "vision.exe", MB_OK);
    return FALSE;
}

.....
pxc200.Grab(fgh[0], frh[0], 0 );
if(pxc200.CheckError(fgh[0])&ERR_CORRUPT)
{
    ++ErrorCount;
}
else
{
    GetImage(frh[0], pbMemoryArea);
    ++FrameCount;
}

.....
```


Esempio: apertura 4 telecamere

```
GetClientRect( &rect );
pDC = GetDC();
m_cdc.CreateCompatibleDC(pDC);
m_Bitmap.CreateCompatibleBitmap( pDC, rect.right - rect.left ,rect.bottom - rect.top);

m_cdc.PatBlt(
    rect.left,
    rect.top,
    rect.Width(),
    rect.Height(),
    WHITENESS
);
m_cdc.SelectObject( &m_Bitmap );
m_cdc.SelectObject(&m_Pen);
m_cdc.SetBkMode( OPAQUE );
m_cdc.SetBkColor( RGB( 0,0,0 ) );
m_cdc.SetTextColor( RGB( 0,255,0 ) );
m_dScala = 1.;
m_dDeltaX = 22300;
m_dDeltaY = 11500;
m_iXcentro = (rect.right - rect.left)/2;
m_iYcentro = (rect.bottom - rect.top)/2;
ReleaseDC( pDC );
if ( !OpenFrameGrabber( iNumeroTelecamere, iDoppia, iProgressiva, &iImageMaxX,
                        &iImageMaxY ) )
{
    // Errore nella apertura della piastra di visione
    iNumeroTelecamere=0;
}
GetGlobals( m_hWnd,
            m_cdc.m_hDC,
            &WindowX,
            &WindowY,
            &iXOffset,
            &iYOffset,
            &iXOffv,
            &iYOffv
);
OpenCGVSBASE(
    iImageMaxX ,
    iImageMaxY ,
    &pbMemoryArea0,
    0
);
OpenCGVSBASE(
    iImageMaxX ,
    iImageMaxY ,
    &pbMemoryArea1,
    1
);
```

```

OpenCGVSBASE(
    iImageMaxX ,
    iImageMaxY ,
    &pbMemoryArea2,
    2
);
OpenCGVSBASE(
    iImageMaxX ,
    iImageMaxY ,
    &pbMemoryArea3,
    3
);
SetWindowParameters(
    this->m_hWnd,
    0, // iXoffsetWindowCenter,
    0, // iYoffsetWindowCenter,
    RGB(128,0,0), // iLinePenColor,
    RGB(255,0,128), // iCirclePenColor,
    RGB(0,128,0), // iArcPenColor
    iWindowsID,
    RGB(255,0,0), // iCoordinatePenColor,
    RGB(255,0,0) // iBorderPenColor
);
if ( iNumeroTelecamere < 1 )
{
    LoadBmpImage( sPathAndNameFileBmp1, iCamera );
}

```

CloseCGVSBASE

Chiude le operazioni della libreria VisAlgo e rilascia la memoria predisposta per l'immagine

```
GVSBASEAPI int WINAPI CloseCGVSBASE( int iHandle );
```

Valore di ritorno

- 1 OK
- 0 libreria mai aperta

Parametri

int iHandle Handle di camera o istanza (Default 0)

Esempio: PX610

```
int i;

for(i=0; i<2; ++i)
{
    if(tagQ[i])
    {
        while(!IsFinished(fgh[0], tagQ[i]))
        {
            WaitVB(fgh[0]);
        }
    }
}
for(i=0; i<2; ++i)
{
    if(frh[i])
    {
        FreeFrame(frh[i][0]);
    }
}
else
{
    if(fgh)
    {
        FreeFG(fgh[0]);
    }
}
ExitLibrary();
if ( !CloseCGVSBASE ( ) )
{
    MessageBox(0, "Failed to close VisAlgo", "vision.exe", MB_OK);
    return FALSE;
}
```

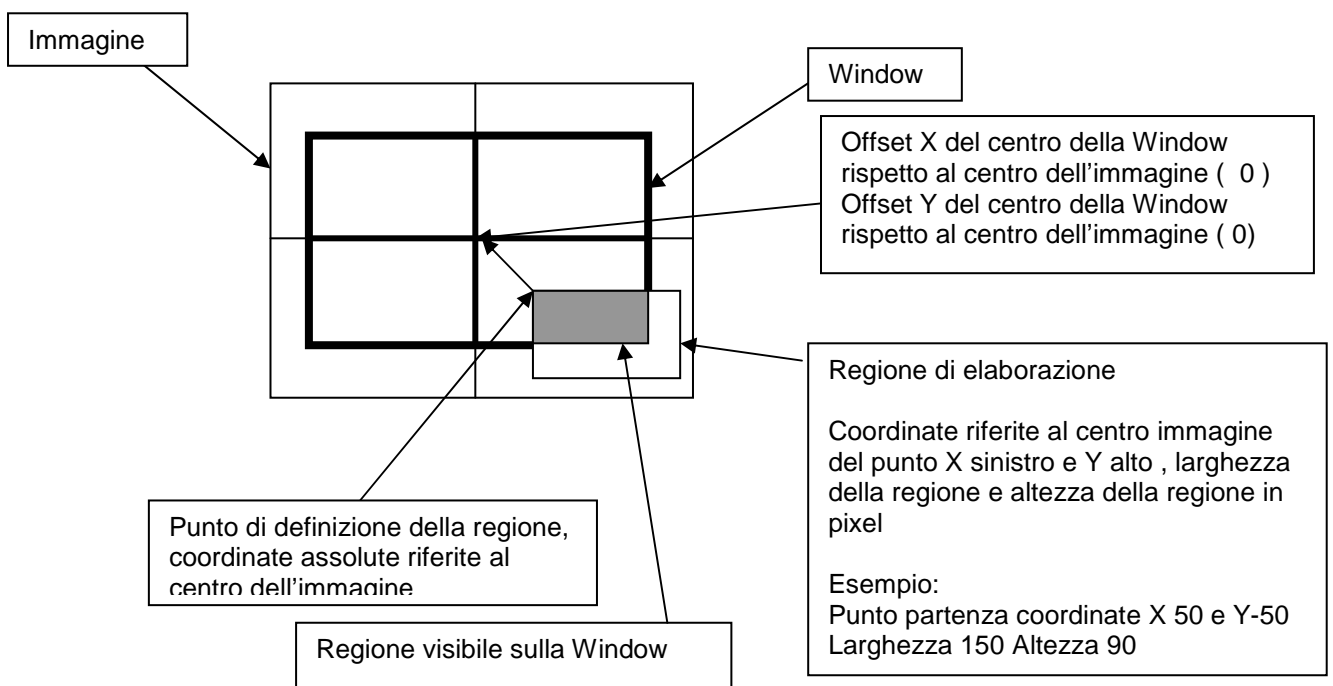
Esempio: PXC2000

```
int i;
for(i=0; i<2; ++i)
{
    if(tagQ[i])
        pxc200.WaitFinished(fgh[0], tagQ[i]);
}
for(i=0; i<2; ++i)
{
    if(frh[i])
        frame.FreeFrame(frh[i]);
}
if(fgh[0])
    pxc200.FreeFG(fgh[0]);
imagination_CloseLibrary(&pxc200);
imagination_CloseLibrary(&frame);
if ( !CloseCGVSBASE ( ) )
{
    MessageBox(0, "Failed to close VisAlgo", "vision.exe", MB_OK);
    return FALSE;
}
```

Window parameters

Per la rappresentazione dell'immagine è necessario utilizzare una Window con cui interagire, in questo capitolo sono descritte le funzioni per la definizione e l'utilizzo della finestre di rappresentazione dell'immagine.

3- SetActiveCamera	Definisce il numero della camera attiva
4- SetWindowParameters	Definisce i parametri della finestra con la telecamera attiva
5- ChangeWindowParameters	Cambia i valori X e Y del centro della finestra
6- ToXRegion	Converte la coordinata a X immagine in coordinata X finestra
7- ToYRegion	Converte la coordinata a Y immagine in coordinata Y finestra
8- GetImageWindowBorder	Fornisce l'offset delle coordinate del bordo superiore della finestra



SetActiveCamera

Attiva la camera

```
GVSBASEAPI int WINAPI SetActiveCamera (  
    int iNumberCameraActive    // 1-4  
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iNumberCameraActive Numero della camera, tutte le operazioni che seguiranno saranno abbinata al numero della camera attiva

Esempio: PX600

```
// Attiva la quarta camera  
SetCamera( fgh[0], 3, 0);  
SetActiveCamera(4);
```

Esempio: PXC2000

```
// Attiva la quarta camera  
pxc200.SetCamera( fgh[0], 3, 0);  
SetActiveCamera(4);
```

SetWindowParameters

Definisce i parametri per posizionare la finestra sull'immagine della camera attiva, definire inoltre i colori per i disegni, da effettuare sulla finestra.

```
GVSBASEAPI int WINAPI SetWindowParameters (  
    HWND hwnd,  
    int  iXoffsetWindowCenter,  
    int  iYoffsetWindowCenter,  
    int  iLinePenColor,  
    int  iCirclePenColor,  
    int  iArcPenColor,  
    int  iWindowsID,  
    int  iCoordMaskPenColor ,  
    int  iBorderPenColor  
    );
```

Valore di ritorno

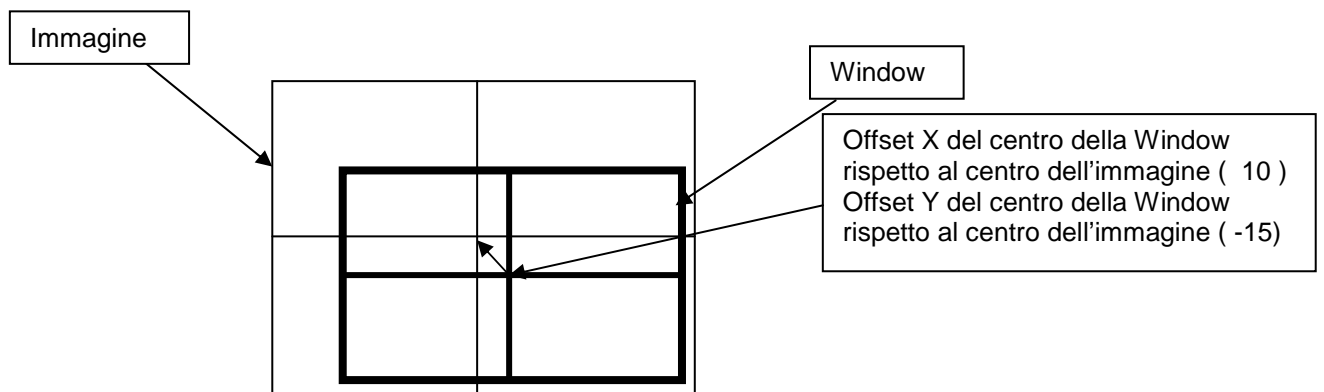
1 OK
0 errore nella chiamata

Parametri

HWND hwnd	Handle della finestra
int iXoffsetWindowCenter	Valore X dell'offset del centro della finestra rispetto al centro dell'immagine
int iYoffsetWindowCenter	Valore Y dell'offset del centro della finestra rispetto al centro dell'immagine
int iLinePenColor	Colore RGB della linea
int iCirclePenColor	Colore RGB del cerchio
int iArcPenColor	Colore RGB dell'arco
int iWindowsID	Identificativo della finestra 0-15(Default 0)
int iCoordMaskPenColor	Colore RGB delle coordinate
int iBorderPenColor	Colore RGB del bordo

Esempio:

```
iWindowsID = 0;  
SetWindowParameters(  
    this->m_hWnd,  
    0, // iXoffsetWindowCenter,  
    0, // iYoffsetWindowCenter,  
    RGB(128,0,0), // iLinePenColor,  
    RGB(255,0,128), // iCirclePenColor,  
    RGB(0,128,0), // iArcPenColor  
    iWindowsID,  
    RGB(255,0,0), // iCoordinatePenColor,  
    RGB(255,0,0) // iBorderPenColor  
);
```



ChangeWindowParameters

Somma algebricamente i valori dell'offset del centro della telecamera rispetto al centro dell'immagine con quelli specificati nei parametri.

```
GVSBASEAPI int WINAPI ChangeWindowParameters (  
    int    iXoffsetWindowCenter,  
    int    iYoffsetWindowCenter,  
    int    iWindowsID  
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iXoffsetWindowCenter Valore X dell'offset del centro della finestra
 rispetto al centro dell'immagine che viene sommato
 algebricamente all'attuale
int iYoffsetWindowCenter Valore Y dell'offset del centro della finestra
 rispetto al centro dell'immagine che viene sommato
 algebricamente all'attuale
int iWindowsID Identificativo della finestra 0-15(Default 0)

Esempio:

```
ChangeWindowParameters ( 10, 15, 0 );
```

ToXRegion

Ritorna il valore della X di un pixel rispetto alla finestra.

```
GVSBASEAPI double WINAPI ToXRegion( double dX , int iWindowsID);
```

Valore di ritorno

double Valore rispetto alla finestra

Parametri

double dX	Valore X del pixel nell'immagine
int iWindowsID	Identificativo della finestra 0-15(Default 0)

Esempio:

```
SetPixel( hdc , ToXRegion( X ) , ToYRegion( Y ), RGB( 255,255,0 ) );
```

ToYRegion

Ritorna il valore della Y di un pixel rispetto alla finestra.

```
GVSBASEAPI double WINAPI ToYRegion( double dY , int iWindowsID);
```

Valore di ritorno

double Valore rispetto alla finestra

Parametri

double dY	Valore Y del pixel nell'immagine
int iWindowsID	Identificativo della finestra 0-15(Default 0)

Esempio:

```
SetPixel( hdc , ToXRegion( X, 0 ) , ToYRegion( Y, 0 ), RGB( 255,255,0 ) ) ;
```

GetImageWindowBorder

Ritorna i valori dell'offset della finestra

```
GVSBASEAPI int WINAPI GetImageWindowBorder(  
    int * piXoffsetWindowBorderLeft,  
    int * piYoffsetWindowBorderTop,  
    int iWindowsID  
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int * piXoffsetWindowBorderLeft Valore X dell'offset del centro della finestra
int * piYoffsetWindowBorderTop Valore Y dell'offset del centro della finestra
int iWindowsID Identificativo della finestra 0-15(Default 0)

Esempio:

```
int    iXoffset,iYoffset;  
  
GetImageWindowBorder( &iXoffset, &iYoffset,0 );  
SetPixel( hdc , X- iXoffset , Y - iYoffset, RGB( 255,255,0 ) );
```

Picture on Window

Per la rappresentazione dell'immagine è necessario utilizzare una Window con cui interagire, in questo capitolo sono descritte le funzioni per la definizione e l'utilizzo della finestre di rappresentazione dell'immagine.

9- **LoadBmpImage**

Carica un immagine da un file

10- **SaveBmpImage**

Salva l'immagine su un file

11- **PutImageOnWindow**

Mette l'immagine sulla finestra

LoadBmpImage

Carica un'immagine BMP da un file

```
GVSBASEAPI int WINAPI LoadBmpImage(  
    char * sPathAndNameFileBmp,  
    int    iHandle  
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

char * sPathAndNameFileBmp Nome con Path del file da caricare nella
 area dell'immagine.
int iHandle Handle di camera o istanza (Default 0)

Esempio:

```
LoadBmpImage(\\bmp\\picture.bmp,0);
```

SaveBmpImage

Salva l'immagine presente nell'area su un file con formato BMP

```
GVSBASEAPI int WINAPI SaveBmpImage(
    HDC     hdc,
    char *  sPathAndNameFileBmp,
    int     iHandle
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

HDC hdc	Device Context della Window
char * sPathAndNameFileBmp	Nome con Path del file su cui salvare l'immagine.
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
SaveBmpImage("\\bmp\\picture.bmp",0);
```


PutImageOnWindow

Trasferisce l'immagine presente nell'area sulla Window

```
GVSBASEAPI int WINAPI PutImageOnWindow(  
    HDC     hdc,  
    int     iMode,  
    int     iHandle  
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

HDC	hdc	Device Context della Window
int	iMode	Modalità della rappresentazione dell'immagine // 0 = Gray // 1 = Isolux // 2 = Green // 3 = Yellow // 4 = Red // 5 = Blue // 18 = All Green // 19 = All Yellow // 20 = All Red // 21 = All Blue // 30 = All Black
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

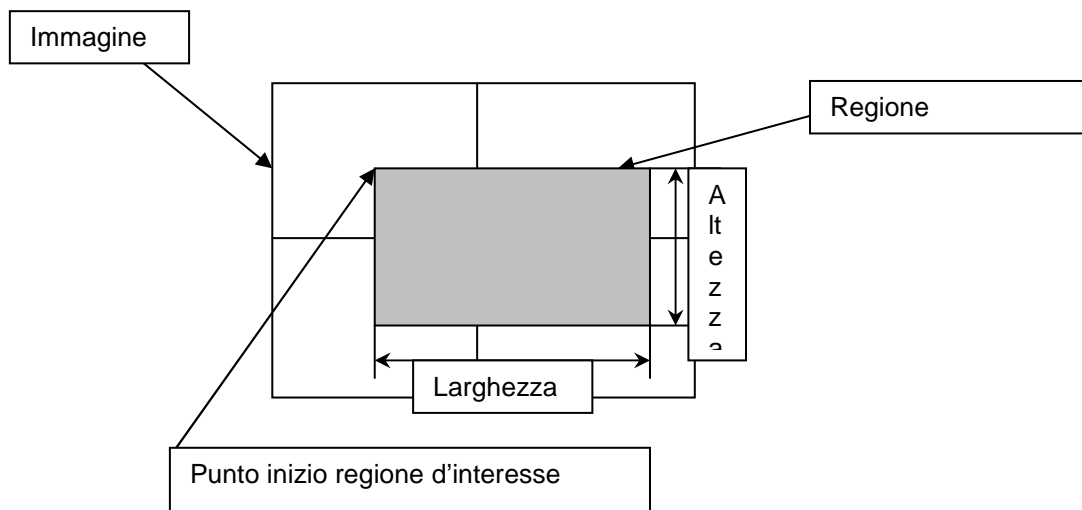
```
m_cdc.PatBlt(  
    rect.left,  
    rect.top,  
    rect.Width(),  
    rect.Height(),  
    WHITENESS  
    );  
  
CaptureImmagine();  
PutImageOnWindow ( m_cdc.m_hDC , 0, 0 ); // Gray mode  
...  
...  
  
pDC = GetDC();  
pDC->BitBlt(  
    0,  
    0,  
    rect.Width(),  
    rect.Height(),  
    &m_cdc,  
    0,  
    0,  
    SRCCOPY  
    );  
ReleaseDC( pDC );
```


Pixel management

L'immagine contenuta nell'area di lavoro può subire delle manipolazioni filtrandola , lavorando sulla luminosità o sul contrasto oppure integrando più immagini consecutivamente.

L'elaborazione dell'immagine è fatta nella regione d'interesse, la definizione della regione è realizzata tramite il punto superiore a sinistra indicandone le coordinate rispetto al centro dell'immagine, la larghezza e l'altezza.

- | | |
|----------------------------------|--|
| 12- SetFilterOnRegion | Filtra una regione dell'immagine con un tipo di filtro |
| 13- SetBrightnessOnRegion | Definisce la luminosità |
| 14- SetContrastOnRegion | Definisce il contrasto |
| 15- IntegrationOfRegion | Somma più immagini |



SetFilterOnRegion

Filtra una regione dell'immagine con un tipo di filtro

```
GVSBASEAPI int WINAPI SetFilterOnRegion(  
    int iLeftRegion,  
    int iTopRegion,  
    int iWidthRegion,  
    int iHeightRegion,  
    int iTypeFilter,  
    int iFilterParam,  
    int iHandle  
);
```

Valore di ritorno

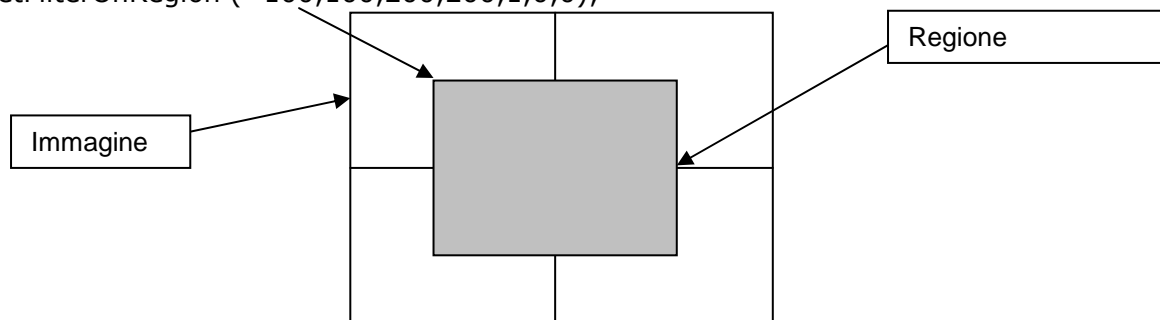
1 OK
0 errore nella chiamata

Parametri

int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse
int iTypeFilter	Tipo di filtro // 0 nofiltro // 1 Media 8 // 2 Media 9 // 3 Media soglia iFilterParam
int iFilterParam	Valore di soglia se si utilizza il 3 nel iTypeFilter
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
SetFilterOnRegion ( -100,100,200,200,1,0,0);
```



SetBrightnessOnRegion

Definisce la luminosità di una regione dell'immagine nell'area

```
GVSBASEAPI int WINAPI SetBrightnessOnRegion(  
    int iLeftRegion,  
    int iTopRegion,  
    int iWidthRegion,  
    int iHeightRegion,  
    int iValue,  
    int iHandle  
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse
int iValue	Valore della luminosità da 0 a 255
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
SetBrightnessOnRegion ( -100,100,200,200,30,0);
```

SetContrastOnRegion

Definisce il contrasto di una regione dell'immagine nell'area

```
GVSBASEAPI int WINAPI SetContrastOnRegion (  
    int iLeftRegion,  
    int iTopRegion,  
    int iWidthRegion,  
    int iHeightRegion,  
    int iValue,  
    int iHandle  
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse
int iValue	Valore del contrasto
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
SetContranstOnRegion ( -100,100,200,200,30,0);
```

IntegrationOfRegion

Somma più immagini

```
GVSBASEAPI int WINAPI IntegrationOfRegion(
    int    iLeftRegion,
    int    iTopRegion,
    int    iWidthRegion,
    int    iHeightRegion,
    int    iFase, // 1    = Start with First Shape ,
                //      = AddShape
                // 99   = Add Shape and Close
                Integration
    int    iHandle
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse
int iFase	Fase dell'integrazione // 1 = Start with First Shape , // = AddShape // 99 = Add Shape and Close Integration
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
GetImage( pArea );
IntegrationOfRegion ( -100,100,200,200, 1,0);
for ( i=0; i<32; i++)
{
    GetImage( pArea );
    IntegrationOfRegion ( -100,100,200,200, 2,0);
}
GetImage( pArea );
IntegrationOfRegion ( -100,100,200,200, 99,0);
```


Pixel value

Il pixel rappresenta l'unità base dell'immagine, le seguenti funzioni operano per estrarre il valore del pixel .

16- GetPixelValue	Fornisce il valore di un pixel
17- GetAveragePixelsOfRegion	Fornisce la media dei valori dei pixel di una regione
18- GetAveragePixelsCircleRegion	Fornisce la media dei valori dei pixel di una regione circolare
19- GetAveragePixelsRectRegion	Fornisce la media dei valori dei pixel di una regione rettangolare
20- GetPixelMaxLuxRegion	Fornisce il pixel più luminoso di una regione
21- LoadIstogramRegion	Carica l'istogramma di una regione
22- DrawIstogram	Disegna l'istogramma
23- GetBackGroundLimits	Fornisce i limiti o soglie del BackGround

GetPixelValue

Fornisce il valore di un pixel

```
GVSBASEAPI unsigned char WINAPI GetPixelValue (  
    int ix,  
    int iy,  
    int iHandle  
    );
```

Valore di ritorno

unsigned char Byte dove viene caricato il valore del pixel

Parametri

int iX Coordinata X rispetto al centro dell'immagine
int iY Coordinata Y rispetto al centro dell'immagine
int iHandle Handle di camera o istanza (Default 0)

Esempio:

```
unsigned char bVal;  
bVal = GetPixelValue ( -100,100,0);
```

GetAveragePixelsOfRegion

Fornisce la media dei valori dei pixel di una regione

```
GVSBASEAPI unsigned char WINAPI GetAveragePixelsOfRegion(  
    int iLeftRegion,  
    int iTopRegion,  
    int iWidthRegion,  
    int iHeightRegion,  
    int iHandle  
    );
```

Valore di ritorno

unsigned char Byte dove viene caricato il valore del pixel

Parametri

int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
unsigned char bVal;  
bVal = GetAveragePixelsOfRegion ( -100,100,200,200,0);
```

GetAveragePixelsCircleRegion

Fornisce la media dei valori dei pixel di una regione circolare

```
GVSBASEAPI unsigned char WINAPI GetAveragePixelsCircleRegion(  
    int iCenterXRegion,  
    int iCenterYRegion,  
    int iRadius,  
    int iHandle  
    );
```

Valore di ritorno

unsigned char Byte dove viene caricato il valore del pixel

Parametri

int iCenterXRegion	Coordinata X rispetto al centro dell'immagine del centro della regione d'interesse
int iCenterYRegion	Coordinata Y rispetto al centro dell'immagine del centro della regione d'interesse
int iRadius	Raggio
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
unsigned char bVal;
```

```
bVal = GetAveragePixelsCircleRegion ( 0,0,100,0);
```

GetAveragePixelsRectRegion

Fornisce la media dei valori dei pixel di una regione rettangolare

```
GVSBASEAPI unsigned char WINAPI GetAveragePixelsRectRegion(
    int iCenterXRegion,
    int iCenterYRegion,
    int iHalfSide1,
    int iHalfSide2
    int iHandle
    );
```

Valore di ritorno

unsigned char Byte dove viene caricato il valore del pixel

Parametri

int iCenterXRegion	Coordinata X rispetto al centro dell'immagine del centro della regione d'interesse
int iCenterYRegion	Coordinata Y rispetto al centro dell'immagine del centro della regione d'interesse
int iHalfSide1	Dimensione del mezzo lato lungo la X
int iHalfSide2	Dimensione del mezzo lato lungo la Y
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
unsigned char bVal;
```

```
bVal = GetAveragePixelsRectRegion ( 0,0,100,150,0);
```

GetPixelMaxLuxRegion

Fornisce il pixel più luminoso di una regione

```
GVSBASEAPI unsigned char WINAPI GetPixelMaxLuxRegion(
    int    iLeftRegion,
    int    iTopRegion,
    int    iWidthRegion,
    int    iHeightRegion,
    int    iSizeX,
    int    iSizeY,
    int *  piX,
    int *  piY,
    int    iHandle
    );
```

Valore di ritorno

unsigned char Byte dove viene caricato il valore del pixel

Parametri

int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse
int iSizeX	Dimensione X della zona minima
int iSizeY	Dimensione Y della zona minima
int * piX	Puntatore dove viene caricato la coordinata X del punto centrale della zona a massima illuminazione
int * piY	Puntatore dove viene caricato la coordinata Y del punto centrale della zona a massima illuminazione
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
unsigned char bVal;
int          iPozX,iPozY
```

```
bVal = GetPixelMaxLuxRegion ( -100,100,200,200,5,5,&iPozX,&iPozY,0);
```

LoadIstogramRegion

Carica l'istogramma di una regione

```
GVSBASEAPI int WINAPI LoadIstogramRegion(  
    int    iLeftRegion,  
    int    iTopRegion,  
    int    iWidthRegion,  
    int    iHeightRegion,  
    int * pBaricenter ,  
    int * pPeakWhite,  
    int * pBaricenterWhite,  
    int * pHeightest,  
    int * pPeakBlack,  
    int * pBaricenterBlack,  
    int * pLowest,  
    int * iTable,  
    int    iHandle  
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse
int * pBaricenter	Posizione del baricentro (0-255)
int * pPeakWhite	Posizione del picco bianco (0-255)
int * pBaricenterWhite	Posizione del baricentro bianco (0-255)
int * pHeightest	Posizione del piu alto (0-255)
int * pPeakBlack	Posizione del picco nero (0-255)
int * pBaricenterBlack	Posizione del baricentro nero (0-255)
int * pLowest	Posizione del piu basso (0-255)
int * iTable	Tabella di 256 valori dove per ogni posizione contiene il numero di pixel con il valore di pixel equivalente alla valore della posizione
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
unsigned long  iTable[256];

if (!LoadIstogramRegion ( x1, y1,
                          x2, y2,
                          &iBaricentro,
                          &iPiccoBianco, &iBaricentroBianco, &iPiuAlto,
                          &iPiccoNero, &iBaricentroNero, &iPiuBasso, &iTable,0 ) )
{
    return;
}
```

DrawHistogram

Disegna l'istogramma di una regione

```
GVSBASEAPI void WINAPI DrawHistogram(
    HDC     hdc,
    int     iBaricenter,
    int     iPeakWhite,
    int     iBaricenterWhite,
    int     iHeightest,
    int     iPeakBlack,
    int     iBaricenterBlack,
    int     iLowest ,
    int *   iTable
);
```

Parametri

HDC	hdc	Device Context della Window
int	iBaricenter	Posizione del baricentro (0-255)
int	iPeakWhite	Posizione del picco bianco (0-255)
int	iBaricenterWhite	Posizione del baricentro bianco (0-255)
int	iHeightest	Posizione del piu alto (0-255)
int	iPeakBlack	Posizione del picco nero (0-255)
int	iBaricenterBlack	Posizione del baricentro nero (0-255)
int	iLowest	Posizione del piu basso (0-255)
int	* iTable	Tabella di 256 valori dove per ogni posizione contiene il numero di pixel con il valore di pixel equivalente alla valore della posizione

Esempio:

```
unsigned long  iTable[256];

if (!LoadIstogramRegion ( x1, y1,
                        x2, y2,
                        &iBaricentro,
                        &iPiccoBianco, &iBaricentroBianco, &iPiuAlto,
                        &iPiccoNero, &iBaricentroNero, &iPiuBasso, &iTable ) )
{
    return;
}
DrawIstogram ( m_cdc.m_hDC,
              x1, y1,
              x2, y2,
              &iBaricentro,
              &iPiccoBianco, &iBaricentroBianco, &iPiuAlto,
              &iPiccoNero, &iBaricentroNero, &iPiuBasso, &iTable ) ;
```

GetBackGroundLimits

Fornisce i limiti o soglie del BackGround

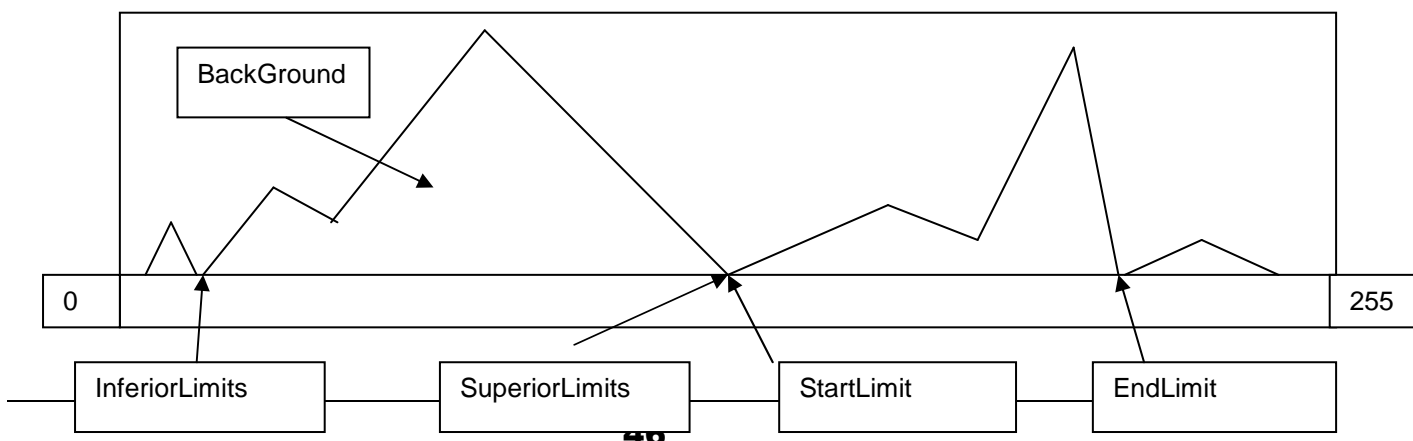
```
GVSBASEAPI int WINAPI GetBackGroundLimits(
    int    iLeftRegion,
    int    iTopRegion,
    int    iWidthRegion,
    int    iHeightRegion,
    int *  pStartLimit,
    int *  pEndLimit,
    int *  pInferiorLimit,
    int *  pSuperiorLimit,
    int    iHandle
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse
int * pStartLimit	Inizio limite del BackGround
int * pEndLimit	Fine limite del BackGround
int * pInferiorLimit	Limite inferiore
int * pSuperiorLimit	Limite superiore
int iHandle	Handle di camera o istanza (Default 0)



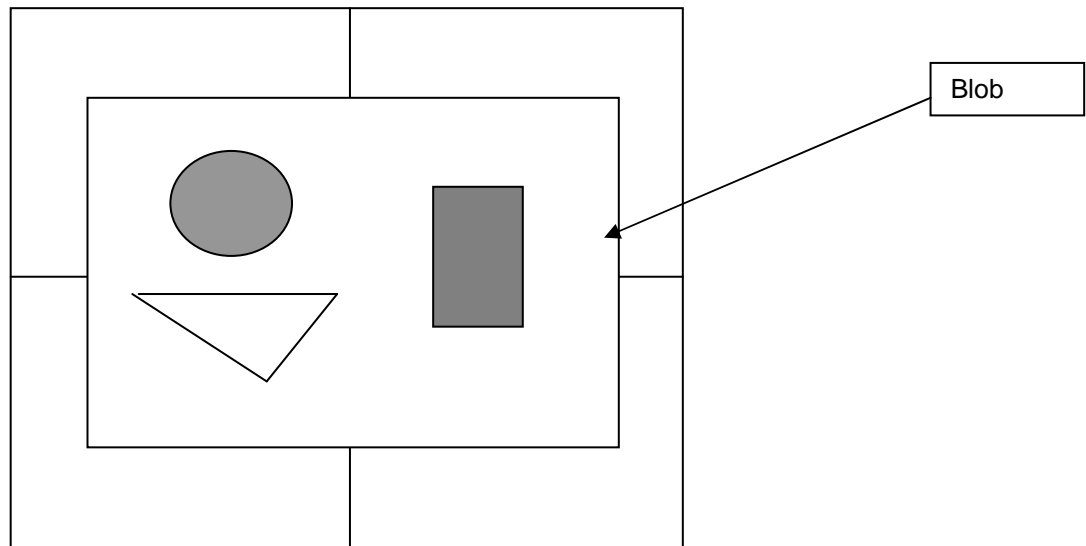
Esempio:

```
if (!GetBackGroundLimits ( x1, y1,  
                           x2, y2,  
                           &iIntDa,  
                           &iIntA,  
                           &iExtDa,  
                           &iExtA,  
                           0  
                           ))  
{  
    return;  
}
```

Blobs

I Blobs sono delle agglomerati di pixels che rappresentano delle figure, i blobs hanno come parti distintive il baricentro, il perimetro, l'area le e le dimensioni.

Le funzionalità lavorano su un archivio di blobs che sono estratti da una regione d'interesse .



24- **GetAllBlobs**

25- **RemoveAllBlobs**

26- **GetValueBlob**

27- **GetDimensionBlob**

28- **GetCircleBlob**

29- **GetEllipseBlob**

30- **GetXAxesRectBlob**

31- **GetYAxesRectBlob**

32- **GetAllPixelsBlob**

33- **GetEdgeBlob**

34- **DrawBlob**

Acquisisce tutti i blob di una regione e li archivia

Elimina l'archivio blobs

Fornisce i valori caratteristici di un blob

Fornisce le dimensioni di un blob

Fornisce il cerchio più performante di un blob

Fornisce l'ellisse più performante di un blob

Fornisce il rettangolo più performante lungo l'asse X

Fornisce il rettangolo più performante lungo l'asse Y

Fornisce tutti i pixel del blob

Fornisce tutti i bordi del blob

Disegna le caratteristiche di un blob

GetAllBlobs

Acquisisce tutti i blob di una regione e li archivia
(è stato introdotto la possibilità dell'assottigliamento del blob)

```
GVSBASEAPI int WINAPI GetAllBlobs(  
    int iLeft,  
    int iTop,  
    int iWidth,  
    int iHeight,  
    int iMode,  
    int iModeLowParam,  
    int iModeHighParam,  
    int iTypeBlob,  
    int iLowArea,  
    int iTypeFilter,  
    int iFilterParam,  
    int iFlagDistanzaCentro,  
    int iPerc,  
    int iHandle  
);
```

Valore di ritorno

int Numero di blob acquisiti
0 errore nella chiamata

Parametri

int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse

int iMode

Modalità per l'acquisizione dei blobs

0 = Automatico con estrazione background

iModeLowParam : // 1 internal 0 external

if (iModeLowParam) // 1 internal 0 external

{

 iModalita = 2;

 iEMin = iStartLimit;

 iEMax = iEndLimit;

}

else

{

 iModalita = 3;

 iEMin = iInferiorLimit;

 iEMax = iSuperiorLimit;

}

1 = soglia bianco

iModeLowParam = valore soglia

2 = soglia nero

iModeLowParam = valore soglia

3= estrai interno

iModeLowParam = soglia inferiore

iModeHighParam = soglia superiore

4= estrai esterno

iModeLowParam = soglia inferiore

iModeHighParam = soglia superiore

77 = Estrazione soglie con

iModeLowParam = 0 Nero

 1 Bianco

iModeHightParam

0: // Fuzzy ENTROPY Shannon

1: // Fuzzy Yager

2: // Iterative selection

3: // Grey level histograms

4: // Johannessen method for using entropy

5: // Kapur method for using entropy

6: // The two histogram peaks

7: // Minimum error thresholding

8: // Mean

9: // Pun method for using entropy

10: // The black percentage

11: // The two histogram peaks

21: // Thr_relax

22: // Multisoglia

iPerc = percentuale di variazione della soglia

trovata

99 = Automatico con Baricentri

iModeLowParam

- 0: // MaxNeri Sotto PiccoNero
- 1: // UltraNeri Sotto BaricentroNero
- 2: // AltiNeri Sotto BaricentroNero
- 3: // MedioNeri Sotto
(Baricentro + BaricentroNero) / 2
- 4: // Neri Sotto Baricentro
- 5: // Bianchi Da Baricentro in poi
- 6: // MedioBianchi Da
(Baricentro + BaricentroBianco) / 2 in poi
- 7: // Alti Bianchi Da BaricentroBianco in poi
- 8: // Ultra Bianchi Da BaricentroBianco in poi
- 9: // MaxBianchi Da PiccoBianco in poi

int iModeLowParam
int iModeHighParam

int iTypeBlob Tipo di blob
 0 = Blob vuoto
 1 = Blob pieno

int iLowArea Area minima del blob

N.B. Se iLowArea è negativa si attiva l'algoritmo di assottigliamento del Blob

int iTypeFilter Tipo di filtro
 // 0 nofiltro
 // 1 Media 8
 ia.ia.ia
 ia.ic.ia media dei soli ia
 ia.ia.ia

 // 2 Media 9
 ia.ia.ia
 ia.ic.ia media degli ia piu ic
 ia.ia.ia

 // 3 Media soglia iFilterParam
 // 5 Media 25 5x5 media degli ia piu ic
 // 7 Media 49 7x7 media degli ia piu ic
 // 9 Media 81 9x9 media degli ia piu ic

int iFilterParam Valore di soglia se si utilizza il 3 nel iTypeFilter
 per tipo di filtro con soglia si intende che
 vengono presi in considerazione nella media 3x3
 degli adiacenti al centro solo i pixel che hanno
 come differenza tra il valore del pixel centrale ed
 il valore del pixel adiacente un valore inferiore al
 valore indicato nel iFilterParam
 ia.ia.ia
 ia.ic.ia
 ia.ia.ia

 se l'assoluto(ic-ia) < iFilterParam usato nella
 media 3x3
 se l'assoluto(ic-ia) >= iFilterParam **non** usato
 nella media 3x3

int iFlagDistanzaCentro Flag ordinamento per distanza dal centro
int iPerc Percentuale di variazione della soglia

trovata, la soglia trovata viene variata sommando o sottraendo una percentuale del valore della soglia trovata

int iHandle Handle di camera o istanza (Default 0)

Esempio:

```
if (!GetAllBlobs (     -100, 100,200,200,
                  0,     // Automatico con estrazione background
                  1,     //     iEMin    = iStartLimit;
                  //     iEMax    = iEndLimit;
                  0,     //
                  0,     //     Blob vuoti
                  200,    //     200 pixel di area minima
                  0,     //     Nessun Filtro
                  0     //
                  ) )
{
    return;
}
```

RemoveAllBlobs

Elimina l'archivio blobs

```
GVSBASEAPI int WINAPI RemoveAllBlobs( );
```

Valore di ritorno

1	OK
0	errore nella chiamata

Esempio:

```
if (!RemoveAllBlobs ( ) )  
{  
    return;  
}
```

GetValueBlob

Fornisce i valori caratteristici di un blob

```
GVSBASEAPI int WINAPI GetValueBlob(  
    int          iNumberBlobBySize,  
    double * pdXGeometric,  
    double * pdYGeometric,  
    double * pdArea,  
    double * pdLengthBorder  
    double * pdXBaricenter,  
    double * pdYBaricenter,  
    double * pdAreaGravity,  
    double * pdMediaGrigiBlob,  
    double * pdMediaGrigiROI  
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iNumberBlobBySize	Numero del blob in dimensioni dal più grande al più piccolo
double *	pdXGeometric	Ritorna il valore della coordinata X del baricentro geometrico (valore pixel 1 o 0)
double *	pdYGeometric	Ritorna il valore della coordinata Y del baricentro geometrico (valore pixel 1 o 0)
double *	pdArea	Ritorna il numero di pixel del blob (Area)
double *	pdLengthBorder	Ritorna il numero di pixel del contorno del blob (Perimetro)
double *	pdXBaricenter	Ritorna il valore della coordinata X del baricentro (valore pixel 0-255)
double *	pdYBaricenter	Ritorna il valore della coordinata Y del baricentro (valore pixel 0-255)
double *	pdAreaGravity	Ritorna la somma del valore dei pixel del blob (Area valore pixel 0-255)
double *	pdMediaGrigiBlob	Ritorna il valore medio dei pixels del BLOB (Valore pixel 0-255)
double *	pdMediaGrigiROI	Ritorna il valore medio dei pixels della ROI (Valore pixel 0-255)senza il Blob

Esempio:

```
double dXGeometric ;
double dYGeometric ;
double dXBaricenter ;
double dYBaricenter ;
double dArea;
double dAreaGravity;
double dMediaGrigiBlob;
double dMediaGrigiROI;

if (!GetValueBlob ( 1,
                  &dXGeometric,
                  &dYGeometric,
                  &dArea,
                  &dLengthBorder,
                  &dXBaricenter,
                  &dYBaricenter,
                  &dAreaGravity,
                  &dMediaGrigiBlob,
                  &dMediaGrigiROI
                ))
{
    return;
}
```

GetDimensionBlob

Fornisce le dimensioni di un blob

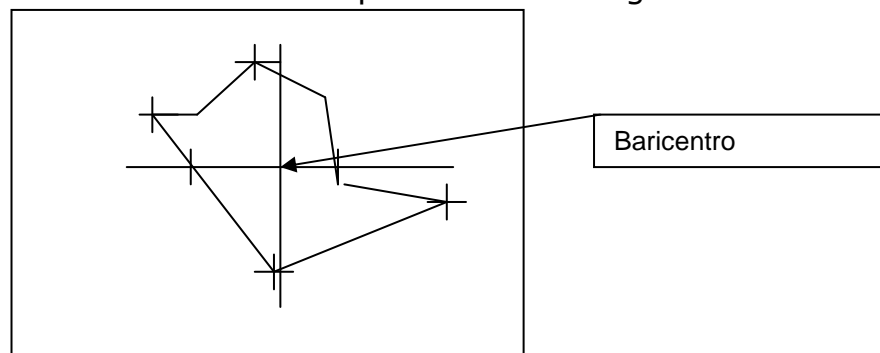
```
GVSBASEAPI int WINAPI GetDimensionBlob(
    int          iNumberBlobBySize,
    double *    pdXHigh,
    double *    pdYHigh,
    double *    pdXLow,
    double *    pdYLow,
    double *    pdXRight,
    double *    pdYRight,
    double *    pdXLeft,
    double *    pdYLeft,
    double *    pdXHighAlongBaricenter,
    double *    pdYHighAlongBaricenter,
    double *    pdXLowAlongBaricenter,
    double *    pdYLowAlongBaricenter,
    double *    pdXRightAlongBaricenter,
    double *    pdYRightAlongBaricenter,
    double *    pdXLeftAlongBaricenter,
    double *    pdYLeftAlongBaricenter
);
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

int	iNumberBlobBySize	Numero del blob in dimensioni dal più grande al più piccolo
double *	pdXHigh	Ritorna la coordinata X del punto del bordo del blob più alto
double *	pdYHigh	Ritorna la coordinata Y del punto del bordo del blob più alto
double *	pdXLow	Ritorna la coordinata X del punto del bordo del blob più bassa
double *	pdYLow	Ritorna la coordinata Y del punto del bordo del blob più bassa
double *	pdXRight	Ritorna la coordinata X del punto del bordo del blob più a destra
double *	pdYRight	Ritorna la coordinata Y del punto del bordo del blob più a destra
double *	pdXLeft	Ritorna la coordinata X del punto del bordo del blob più a sinistra
double *	pdYLeft	Ritorna la coordinata Y del punto del bordo del blob più a sinistra
double *	pdXHighAlongBaricenter	Ritorna la coordinata X del punto del bordo del blob più alto lungo il baricentro
double *	pdYHighAlongBaricenter	Ritorna la coordinata Y del punto del bordo del blob più alto lungo il baricentro
double *	pdXLowAlongBaricenter	Ritorna la coordinata X del punto del bordo del blob più basso lungo il baricentro
double *	pdYLowAlongBaricenter	Ritorna la coordinata Y del punto del bordo del blob più basso lungo il baricentro
double *	pdXRightAlongBaricenter	Ritorna la coordinata X del punto del bordo del blob più a destra lungo il baricentro
double *	pdYRightAlongBaricenter	Ritorna la coordinata Y del punto del bordo del blob più a destra lungo il baricentro
double *	pdXLeftAlongBaricenter	Ritorna la coordinata X del punto del bordo del blob più a sinistra lungo il baricentro
double *	pdYLeftAlongBaricenter	Ritorna la coordinata Y del punto del bordo del blob più a sinistra lungo il baricentro



Esempio:

```
double * dXHigh
double * dYHigh
double * dXLow
double * dYLow
double * dXRight
double * dYRight
double * dXLeft
double * dYLeft
double * dXHighAlongBaricenter
double * dYHighAlongBaricenter
double * dXLowAlongBaricenter
double * dYLowAlongBaricenter
double * dXRightAlongBaricenter
double * dYRightAlongBaricenter
double * dXLeftAlongBaricenter
double * dYLeftAlongBaricenter
```

```
if (!GetDimensionBlob (    1,
                        &dXHigh,
                        &dYHigh,
                        &dXLow,
                        &dYLow,
                        &dXRight,
                        &dYRight,
                        &dXLeft,
                        &dYLeft,
                        &dXHighAlongBaricenter,
                        &dYHighAlongBaricenter,
                        &dXLowAlongBaricenter,
                        &dYLowAlongBaricenter,
                        &dXRightAlongBaricenter,
                        &dYRightAlongBaricenter,
                        &dXLeftAlongBaricenter,
                        &dYLeftAlongBaricenter
                        ) )
{
    return;
}
```

GetCircleBlob

Fornisce il cerchio più performante di un blob

```
GVSBASEAPI int WINAPI GetCircleBlob(
    int          iNumberBlobBySize,
    double       dPixelGap,
    double *     pdRadius,
    double *     pdXCenter,
    double *     pdYCenter,
    double *     pdRoutness,
    int          iCorona,
    double *     pdRadiusExt,
    double *     pdXCenterExt,
    double *     pdYCenterExt,
    double *     pdRadiusInt,
    double *     pdXCenterInt,
    double *     pdYCenterInt,
    double *     pdRadiusMiddle,
    double *     pdXCenterMiddle,
    double *     pdYCenterMiddle
);
```

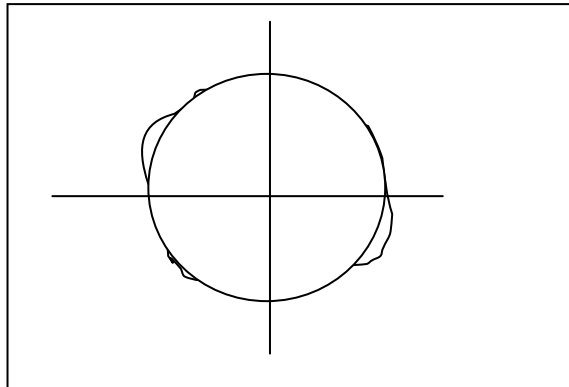
Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iNumberBlobBySize	Numero del blob in dimensioni dal più grande al più piccolo
double	dPixelGap	Valore di gap massimo ammesso
double *	pdRadius	Ritorna il raggio del cerchio
double *	pdXCenter	Ritorna la coordinata X del centro del cerchio
double *	pdYCenter	Ritorna la coordinata Y del centro del cerchio
double *	pdRoutness	Ritorna il valore della rotondità NumPuntiBordo entro il gap / NumPuntiBordo totali * 100
int	iCorona	Flag per ottenere la corona circolare
double *	pdRadiusExt	Ritorna il raggio del cerchio esterno
double *	pdXCenterExt	Ritorna la coordinata X del centro del cerchio
double *	pdYCenterExt	Ritorna la coordinata Y del centro del cerchio
double *	pdRadiusInt	Ritorna il raggio del cerchio interno
double *	pdXCenterInt	Ritorna la coordinata X del centro del cerchio

double *	pdYCenterInt	Ritorna la coordinata Y del centro del cerchio
double *	pdRadiusMiddle	Ritorna il raggio del cerchio intermedio
double *	pdXCenterMiddle	Ritorna la coordinata X del centro del cerchio
double *	pdYCenterMiddle	Ritorna la coordinata Y del centro del cerchio



Esempio:

```
double * dRadius;  
double * dXCenter;  
double * dYCenter;  
double * dRoutness;
```

```
if (!GetCircleBlob ( 1,  
                    2,  
                    & dRadius,  
                    & dXCenter,  
                    & dYCenter,  
                    & dRoutness  
                    iCorona  
                    pdRadiusExt  
                    pdXCenterExt  
                    pdYCenterExt  
                    pdRadiusInt  
                    pdXCenterInt  
                    pdYCenterInt  
                    pdRadiusMiddle  
                    pdXCenterMiddle  
                    pdYCenterMiddle  
                    ))  
{  
    return;  
}
```

GetEllipseBlob

Fornisce l'ellisse più performante di un blob

```
GVSBASEAPI int WINAPI GetEllipseBlob (  
    int          iNumberBlobBySize,  
    double *    pdXc,  
    double *    pdYc,  
    double *    pdAngA,  
    double *    pdRgA,  
    double *    pdAngB,  
    double *    pdRgB,  
    double *    pdX1,  
    double *    pdY1,  
    double *    pdX2,  
    double *    pdY2,  
    double *    pdX3,  
    double *    pdY3,  
    double *    pdX4,  
    double *    pdY4,  
    int          iCorona,  
    double      dGap  
    );
```

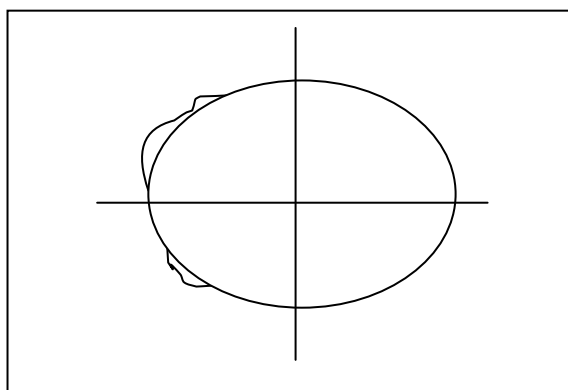
Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iNumberBlobBySize	Numero del blob in dimensioni dal più grande al più piccolo
double *	pdXc	Ritorna la coordinata X del centro del cerchio
double *	pdYc	Ritorna la coordinata Y del centro del cerchio
double *	pdAngA	Angolo A dell'asse maggiore
double *	pdRgA	Raggio maggiore
double *	pdAngB	Angolo B dell'asse minore
double *	pdRgB	Raggio minore

double * pdX1	X del primo punto d' intersezione dell'asse maggiore
double * pdY1	Y del primo punto d' intersezione dell'asse maggiore
double * pdX2	X del secondo punto d' intersezione dell'asse maggiore
double * pdY2	Y del secondo punto d' intersezione dell'asse maggiore
double * pdX3	X del primo punto d' intersezione dell'asse minore
double * pdY3	Y del primo punto d' intersezione dell'asse minore
double * pdX4	X del secondo punto d' intersezione dell'asse minore
double * pdY4	Y del secondo punto d' intersezione dell'asse minore
int iCorona	1 = Ellisse normale 2 = Ellisse Esterna 4 = Ellisse Interna 8 = Ellisse Gap 0x100 orientamento con distanza dei punti 0x000 orientamento distanza baricentro
double dGap	Valore di gap massimo ammesso



Esempio:

```
if (!GetEllipseBlob (
    iNumberBlobBySize,
    pdXc,
    pdYc,
    pdAngA,
    pdRgA,
    pdAngB,
    pdRgB,
    pdX1,
    pdY1,
    pdX2,
    pdX2,
    pdX3,
    pdY3,
    pdX4,
    pdX4,
    iCorona,
    dPixelGap
))
{
    return;
}
```


GetXAxesRectBlob

Fornisce il rettangolo più performante lungo l'asse X

```
GVSBASEAPI int WINAPI GetXAxesRectBlob(
    int            iNumberBlobBySize,
    double         dPixelGap,
    double * pdXRectHighRight,
    double * pdYRectHighRight,
    double * pdXRectLowRight,
    double * pdYRectLowRight,
    double * pdXRectHighLeft,
    double * pdYRectHighLeft,
    double * pdXRectLowLeft,
    double * pdYRectLowLeft,
    double * pdXRightAlongBaricenterY,
    double * pdYRightAlongBaricenterY,
    double * pdXLeftAlongBaricenterY,
    double * pdYLeftAlongBaricenterY,
    double * pdDiameter
    );
```

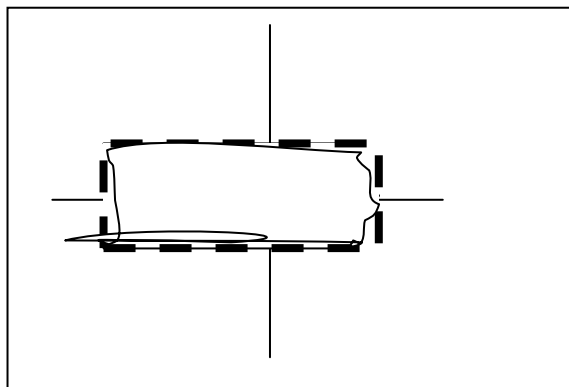
Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iNumberBlobBySize Numero del blob in dimensioni dal più grande al più piccolo
double dPixelGap Valore di gap massimo ammesso
double * pdXRectHighRight Coordinata X del punto alto a destra
double * pdYRectHighRight Coordinata Y del punto alto a destra
double * pdXRectLowRight Coordinata X del punto basso a destra
double * pdYRectLowRight Coordinata Y del punto basso a destra
double * pdXRectHighLeft Coordinata X del punto alto a sinistra
double * pdYRectHighLeft Coordinata Y del punto alto a sinistra
double * pdXRectLowLeft Coordinata X del punto basso a sinistra
double * pdYRectLowLeft Coordinata Y del punto basso a sinistra
double * pdXRightAlongBaricenterY Coordinata X del punto a destra lungo il baricentro coordinata Y
double * pdYRightAlongBaricenterY Coordinata Y del punto a destra lungo il baricentro coordinata Y

double * pdXLeftAlongBaricenterY Coordinata X del punto a sinistra lungo
il baricentro coordinata Y
double * pdYLeftAlongBaricenterY Coordinata Y del punto a sinistra lungo
il baricentro coordinata Y
double * pdDiameter Diametro misurato attraverso la media sul
baricentro coordinata Y



Esempio:

```
double dXRectHighRight;
double dYRectHighRight;
double dXRectLowRight;
double dYRectLowRight;
double dXRectHighLeft;
double dYRectHighLeft;
double dXRectLowLeft;
double dYRectLowLeft;
double dXRightAlongBaricenterY;
double dYRightAlongBaricenterY;
double dXLeftAlongBaricenterY;
double dYLeftAlongBaricenterY;
double dDiameter;

if (!GetXAxesRectBlob (    1, 2,
                          &dRectHighRight,
                          &dRectHighRight,
                          &dXRectLowRight,
                          &dYRectLowRight,
                          &dXRectHighLeft,
                          &dYRectHighLeft,
                          &dXRectLowLeft,
                          &dYRectLowLeft,
                          &dXRightAlongBaricenterY,
                          &dYRightAlongBaricenterY,
                          &dXLeftAlongBaricenterY,
                          &dYLeftAlongBaricenterY,
                          &dDiameter
                          ))
{
    return;
}
```

GetYAxesRectBlob

Fornisce il rettangolo più performante lungo l'asse Y

```
GVSBASEAPI int WINAPI GetYAxesRectBlob(
    int          iNumberBlobBySize,
    double       dPixelGap,
    double *     pdXRectHighRight,
    double *     pdYRectHighRight,
    double *     pdXRectLowRight,
    double *     pdYRectLowRight,
    double *     pdXRectHighLeft,
    double *     pdYRectHighLeft,
    double *     pdXRectLowLeft,
    double *     pdYRectLowLeft,
    double *     pdXHighAlongBaricenterX,
    double *     pdYHighAlongBaricenterX,
    double *     pdXLowAlongBaricenterX,
    double *     pdYLowAlongBaricenterX,
    double *     pdDiameter
);
```

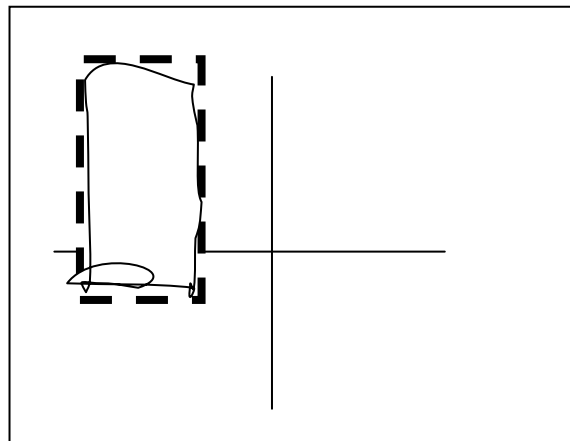
Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iNumberBlobBySize Numero del blob in dimensioni dal più grande al più piccolo
double dPixelGap Valore di gap massimo ammesso
double * pdXRectHighRight Coordinata X del punto alto a destra
double * pdYRectHighRight Coordinata Y del punto alto a destra
double * pdXRectLowRight Coordinata X del punto basso a destra
double * pdYRectLowRight Coordinata Y del punto basso a destra
double * pdXRectHighLeft Coordinata X del punto alto a sinistra
double * pdYRectHighLeft Coordinata Y del punto alto a sinistra
double * pdXRectLowLeft Coordinata X del punto basso a sinistra
double * pdYRectLowLeft Coordinata Y del punto basso a sinistra
double * pdXRightAlongBaricenterX Coordinata X del punto a destra lungo il baricentro coordinata X
double * pdYRightAlongBaricenterX Coordinata Y del punto a destra lungo il baricentro coordinata X

double * pdXLeftAlongBaricenterX Coordinata X del punto a sinistra lungo
il baricentro coordinata X
double * pdYLeftAlongBaricenterX Coordinata Y del punto a sinistra lungo
il baricentro coordinata X
double * pdDiameter Diametro misurato attraverso la media sul
baricentro coordinata X



Esempio:

```
double dXRectHighRight;
double dYRectHighRight;
double dXRectLowRight;
double dYRectLowRight;
double dXRectHighLeft;
double dYRectHighLeft;
double dXRectLowLeft;
double dYRectLowLeft;
double dXRightAlongBaricenterX;
double dYRightAlongBaricenterX;
double dXLeftAlongBaricenterX;
double dYLeftAlongBaricenterX;
double dDiameter;

if (!GetYAxesRectBlob (    1, 2,
                          &dRectHighRight,
                          &dRectHighRight,
                          &dXRectLowRight,
                          &dYRectLowRight,
                          &dXRectHighLeft,
                          &dYRectHighLeft,
                          &dXRectLowLeft,
                          &dYRectLowLeft,
                          &dXRightAlongBaricenterX,
                          &dYRightAlongBaricenterX,
                          &dXLeftAlongBaricenterX,
                          &dYLeftAlongBaricenterX,
                          &dDiameter
                          ))
{
    return;
}
```

GetAllPixelsBlob

Fornisce tutti i pixel del blob

```
GVSBASEAPI int WINAPI GetAllPixelsBlob(  
    int        iNumberBlobBySize,  
    int        iNumberPixelMaxToLoad,  
    int *      piNumberTotalPixel,  
    int *      piPixelX,  
    int *      piPixelY  
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iNumberBlobBySize	Numero del blob in dimensioni dal più grande al più piccolo
int	iNumberPixelMaxToLoad	Numero massimo di punti da caricare
int *	piNumberTotalPixel	Ritorna il numero di punti effettivamente caricati del area del blob
int *	piPixelX	Vettore di coordinate X dei pixels
int *	piPixelY	Vettore di coordinate Y dei pixels

Esempio:

```
#define NUMBER_MAX     1024  
  
int    iNumberPixelLoad;  
int    piXPixel[NUMBER_MAX];  
int    piYPixel[NUMBER_MAX];  
  
if (!GetAllPixelsBlob (1,  
                      NUMBER_MAX,  
                      &iNumberPixelLoad,  
                      piXPixel,  
                      piYPixel  
                      ))  
{  
    return;  
}
```

GetEdgeBlob

Fornisce tutti i bordi del blob

```
GVSBASEAPI int WINAPI GetEdgeBlob(
    int    iNumberBlobBySize,
    int    iNumberPixelMaxToLoad,
    int *  piNumberEdgeTotalPixel,
    int *  piPixelX,
    int *  piPixelY
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iNumberBlobBySize	Numero del blob in dimensioni dal più grande al più piccolo
int	iNumberPixelMaxToLoad	Numero massimo di punti da caricare
int *	piNumberTotalPixel	Ritorna il numero di punti effettivamente caricati del bordo del blob
int *	piPixelX	Vettore di coordinate X dei pixels
int *	piPixelY	Vettore di coordinate Y dei pixels

Esempio:

```
#define NUMBER_MAX     1024

int    iNumberPixelLoad;
int    piXPixel[NUMBER_MAX];
int    piYPixel[NUMBER_MAX];

if (!GetEdgeBlob (    1,
                   NUMBER_MAX,
                   &iNumberPixelLoad,
                   piXPixel,
                   piYPixel
                   ) )
{
    return;
}
```


DrawBlob

Fornisce tutti i bordi del blob

```
GVSBASEAPI int WINAPI DrawBlob( int iNumberBlobBySize ,
                                   int iFlagBlobPicture,
                                   unsigned int uRGBPictureColor,
                                   int iFlagBlobEdge,
                                   unsigned int uRGBEdgeColor,
                                   int iFlagValue,
                                   int iFlagDimension,
                                   int iFlagCircle,
                                   int iFlagXAxesRect,
                                   int iFlagYAxesRect,
                                   HDC hdc,
                                   int iWindowsID,
                                   int iFlagBorderROI,
                                   int iFlagInternalBorderROI,
                                   int iFlagCoordinateROI,
                                   int iFlagEllipse
                                   );
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

int iNumberBlobBySize	Numero del blob in dimensioni dal più grande al più piccolo
int iFlagBlobPicture	Richiesta disegno del blob
unsigned int uRGBPictureColor	Colori RGB del disegno dei pixel del blob
int iFlagBlobEdge	Richiesta disegno del bordo del blob
unsigned int uRGBEdgeColor	Colori RGB del disegno del bordo del blob
int iFlagValue	Richiesta disegno valori caratteristici
int iFlagDimension	Richiesta disegno delle dimensioni
int iFlagCircle	Richiesta disegno del cerchio
int iFlagXAxesRect	Richiesta disegno rettangolo assi lungo X
int iFlagYAxesRect	Richiesta disegno rettangolo assi lungo Y
HDC hdc	Device Context della Window
int iWindowsID	Identificativo della finestra 0-15(Default 0)
int iFlagBorderROI	Richiesta disegno del bordo del ROI
int iFlagInternalBorderROI	Richiesta disegno del bordo interno del ROI
int iFlagCoordinateROI	Richiesta disegno delle coordinate ROI
int iFlagEllipse	Richiesta disegno dell'ellisse

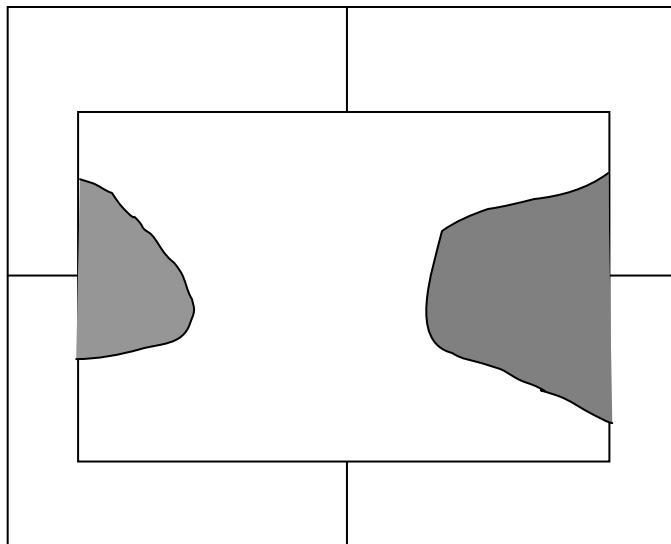
Esempio:

```
if (!DrawBlob (      1,
                   1,      Richiesta disegno del blob
                   RGB(128,0,0), Colori RGB del disegno dei pixel del blob
                   1,      Richiesta disegno del bordo del blob
                   RGB(128,128,0), Colori RGB del disegno dei bordo del blob
                   1,      Richiesta disegno valori caratteristici
                   1,      Richiesta disegno delle dimensioni
                   1,      Richiesta disegno del cerchio
                   1,      Richiesta disegno rettangolo assi lungo X
                   1,      Richiesta disegno rettangolo assi lungo Y
                   m_cdc.m_hDC
                   ))
{
    return;
}
```

Intersected Shapes

Le figure intersecate sono delle agglomerati di pixels che rappresentano delle figure che vengono intersecate dai bordi di una regione d'interesse, le figure intersecate hanno come parti distintive il baricentro, il perimetro, l'area le e le dimensioni.

Le funzionalità lavorano su un archivio di figure intersecate che sono estratte da una regione d'interesse .



35- **GetAllIntersectedShapes**

Acquisisce tutte le figure intersecate di una regione e le archivia

36- **RemoveAllIntersectedShapes**

Elimina l'archivio delle figure intersecate

37- **GetDimensionIntersectedShape**

Fornisce le dimensioni di una figura intersecata

38- **GetEdgeIntersectedShape**

Fornisce i bordi di una figura intersecata

39- **DrawIntersectedShape**

Disegna gli elementi caratteristici di una figura intersecata

GetAllIntersectedShapes

Acquisisce tutte le figure intersecate di una regione e le archivia

```
GVSBASEAPI int WINAPI GetAllIntersectedShapes(  
    int iLeft,  
    int iTop,  
    int iWidth,  
    int iHeight,  
    int iMode,  
    int iModeLowParam,  
    int iModeHighParam,  
    int iTypeFilter,  
    int iFilterParam  
    int iPerc,  
    int iHandle ,  
    );
```

Valore di ritorno

int Numero di figure intersecate acquisite

0 errore nella chiamata

Parametri

int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse

int iMode

Modalità per l'acquisizione dei blobs

0 = Automatico con estrazione background

iModeLowParam : // 1 internal 0 external

if (iModeLowParam) // 1 internal 0 external

{

 iModalita = 2;

 iEMin = iStartLimit;

 iEMax = iEndLimit;

}

else

{

 iModalita = 3;

 iEMin = iInferiorLimit;

 iEMax = iSuperiorLimit;

}

1 = soglia bianco

iModeLowParam = valore soglia

2 = soglia nero

iModeLowParam = valore soglia

3= estrai interno

iModeLowParam = soglia inferiore

iModeHighParam = soglia superiore

4= estrai esterno

iModeLowParam = soglia inferiore

iModeHighParam = soglia superiore

77 = Estrazione soglie con

iModeLowParam = 0 Nero

 1 Bianco

iModeHighParam

0: // Fuzzy ENTROPY Shannon

1: // Fuzzy Yager

2: // Iterative selection

3: // Grey level histograms

4: // Johanness method for using entropy

5: // Kapur method for using entropy

6: // The two histogram peaks

7: // Minimum error thresholding

8: // Mean

9: // Pun method for using entropy

10: // The black percentage

11: // The two histogram peaks

21: // Thr_relax

22: // Multisoglia

iPerc = percentuale di variazione della soglia
trovata

99 = Automatico con Baricentri

iModeLowParam

0: // MaxNeri Sotto PiccoNero

1: // UltraNeri Sotto BaricentroNero

2: // AltiNeri Sotto BaricentroNero

3: // MedioNeri Sotto

(Baricentro + BaricentroNero) / 2

4: // Neri Sotto Baricentro

5: // Bianchi Da Baricentro in poi

6: // MedioBianchi Da

(Baricentro + BaricentroBianco) / 2 in poi

7: // Alti Bianchi Da BaricentroBianco in poi

8: // Ultra Bianchi Da BaricentroBianco in poi

9: // MaxBianchi Da PiccoBianco in poi

int iModeLowParam

int iModeHighParam

int iTypeFilter	Tipo di filtro // 0 nofiltro // 1 Media 8 // 2 Media 9 // 3 Media soglia iFilterParam
int iFilterParam	Valore di soglia se si utilizza il 3 nel iTypeFilter
int iPerc	Percentuale di variazione della soglia trovata
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```

if (!GetAllIntersectedShapes (    -100, 100,200,200,
    0, // Automatico con estrazione background
    1, //    iEMin = iStartLimit;
        //    iEMax = iEndLimit;
    0, //
    0, //    Nessun Filtro
    0, //
    0 //
    ))
{
    return;
}

```

RemoveAllIntersectedShapes

Elimina l'archivio delle figure intersecate

GVSBASEAPI int WINAPI **RemoveAllIntersectedShapes**(void);

Valore di ritorno

1 OK
0 errore nella chiamata

Esempio:

```
if (!RemoveAllIntersectedShapes ( ) )  
{  
    return;  
}
```


GetDimensionIntersectedShape

Fornisce le dimensioni di una figura intersecata

```
GVSBASEAPI int WINAPI GetDimensionIntersectedShape(  
    int      iNumberIntersectedShapeBySize ,  
    double * pdLength,  
    double * pdXHigh,  
    double * pdYHigh,  
    double * pdXLow,  
    double * pdYLow,  
    double * pdXRight,  
    double * pdYRight,  
    double * pdXLeft,  
    double * pdYLeft  
    );
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

int	iNumberIntersectedShapeBySize	Numero della figura in dimensioni dalla più grande alla più piccola
double *	pdXHigh	Ritorna la coordinata X del punto del bordo della figura più alto
double *	pdYHigh	Ritorna la coordinata Y del punto del bordo della figura più alto
double *	pdXLow	Ritorna la coordinata X del punto del bordo della figura più bassa
double *	pdYLow	Ritorna la coordinata Y del punto del bordo della figura più bassa
double *	pdXRight	Ritorna la coordinata X del punto del bordo della figura più a destra
double *	pdYRight	Ritorna la coordinata Y del punto del bordo della figura più a destra
double *	pdXLeft	Ritorna la coordinata X del punto del bordo della figura più a sinistra
double *	pdYLeft	Ritorna la coordinata Y del punto del bordo della figura più a sinistra

Esempio:

```
double * dXHigh  
double * dYHigh  
double * dXLow  
double * dYLow  
double * dXRight  
double * dYRight  
double * dXLeft  
double * dYLeft
```

```
if (!GetDimensionIntersectedShape ( 1,  
                                     &dXHigh,  
                                     &dYHigh,  
                                     &dXLow,  
                                     &dYLow,  
                                     &dXRight,  
                                     &dYRight,  
                                     &dXLeft,  
                                     &dYLeft  
                                     ))  
{  
    return;  
}
```

GetEdgeIntersectedShape

Fornisce i bordi di una figura intersecata

```
GVSBASEAPI int WINAPI GetEdgeIntersectedShape(
    int iNumberIntersectedShapeBySize ,
    int iNumberPixelMaxToLoad,
    int * piNumberEdgeTotalPixel,
    int * piPixelX,
    int * piPixelY
    );
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

int	iNumberIntersectedShapeBySize	Numero della figura in dimensioni dalla più grande alla più piccola
int	iNumberPixelMaxToLoad	Numero massimo di punti da caricare
int *	piNumberTotalPixel	Ritorna il numero di punti effettivamente caricati del bordo della figura
int *	piPixelX	Vettore di coordinate X dei pixels
int *	piPixelY	Vettore di coordinate Y dei pixels

Esempio:

```
#define NUMBER_MAX    1024

int iNumberPixelLoad;
int piXPixel[NUMBER_MAX];
int piYPixel[NUMBER_MAX];

if (!GetEdgeIntersectedShape ( 1,
                              NUMBER_MAX,
                              &iNumberPixelLoad,
                              piXPixel,
                              piYPixel
                              ))
{
    return;
}
```

DrawIntersectedShape

Fornisce i bordi di una figura intersecata

```
GVSBASEAPI int WINAPI DrawIntersectedShape(
    int iNumberIntersectedShapeBySize ,
    int iFlagIntersectPictureEdge,
    unsigned int uRGBEdgeColor,
    int iFlagDimension,
    HDC hdc,
    int iWindowsID,
    int iFlagBorderROI,
    int iFlagInternalBorderROI,
    int iFlagCoordinateROI
);
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

int	iNumberIntersectedShapeBySize	Numero della figura in dimensioni dalla più grande alla più piccola
int	iFlagIntersectPictureEdge	Richiesta disegno del bordo della figura
unsigned int	uRGBEdgeColor	Colori RGB del disegno del bordo
int	iFlagDimension	Richiesta disegno delle dimensioni
HDC	hdc	Device Context della Window
int	iWindowsID	Identificativo della finestra 0-15(Default 0)
int	iFlagBorderROI	Richiesta disegno del bordo del ROI
int	iFlagInternalBorderROI	Richiesta disegno del bordo interno del ROI
int	iFlagCoordinateROI	Richiesta disegno delle coordinate ROI

Esempio:

```
if (!DrawIntersectedShape (1,
                            1,           Richiesta disegno della figura
                            RGB(128,128,0), Colori RGB del disegno dei bordo
                            1,           Richiesta disegno delle dimensioni
                            m_cdc.m_hDC
                            ))
{
    return;
}
```

EDGE

I bordi possono essere utilizzati per ottenere degli elementi geometrici con cui effettuare operazioni di elaborazione, gli elementi geometrici sono rette, cerchi e punti. Le funzioni operano o sui blob o sulle figure intersecate

Le rette sono definite dal valore della :

Distanza

Seno

Coseno

I cerchi sono definiti

Raggio

Coordinata X del centro

Coordinata Y del centro

40- **GetBestEdgeLineOnRegion**

Fornisce la miglior retta sui bordi di un elemento

41- **GetBestEdgeCircleOnRegion**

Fornisce il miglior cerchio sui bordi di un elemento

42- **PointsFromBestEdgeLineAcrossEdgeOnRegion**

Fornisce i punti della miglior retta estratta sui bordi di un elemento

43- **MaximusFromBestEdgeLineAcrossEdgeOnRegion**

Fornisce i massimi della miglior retta estratta sui bordi di un elemento

44- **PointsFromLine2PointAcrossEdgeOnRegion**

Fornisce i punti della miglior retta estratta tra due punti sui bordi di un elemento

45- **MaximusFromLine2PointAcrossEdgeOnRegion**

Fornisce i massimi della miglior retta estratta tra due punti sui bordi di un elemento

GetBestEdgeLineOnRegion

Fornisce la miglior retta sui bordi di un elemento

```
GVSBASEAPI int WINAPI GetBestEdgeLineOnRegion(
    int iType, // 1 = Blob 2=
    IntersectedShape
    int iNumberBlobOrIntersectedShapeBySize ,
    int iLeftRegion,
    int iTopRegion,
    int iWidthRegion,
    int iHeightRegion,
    double dPixelGap,
    double * dLineDistance,
    double * dLineSin,
    double * dLineCos,
    double * dGap,
    double * dMinGap,
    double * dMaxGap
);
```

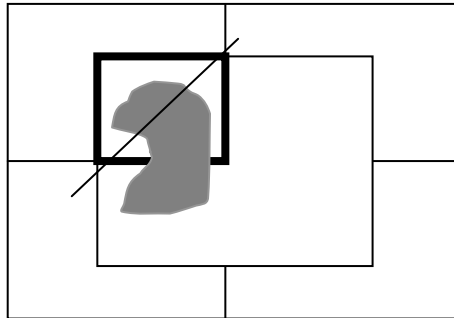
Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iType Tipo di immagine: 1 = Blob 2= IntersectedShape
int iNumberBlobOrIntersectedShapeBySize Numero della figura in dimensioni dalla più grande alla più piccola
int iLeftRegion Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion Larghezza della regione d'interesse
int iHeightRegion Altezza della regione d'interesse

double	dPixelGap	Valore di gap massimo ammesso
double *	dLineDistance	Ritorna il valore della distanza della retta
double *	dLineSin	Ritorna il valore del seno della retta
double *	dLineCos	Ritorna il valore del coseno della retta
double *	dGap	Ritorna l'errore di forma medio
double *	dMinGap	Ritorna l'errore di forma minimo
double *	dMaxGap	Ritorna l'errore di forma massimo



Esempio:

```
double dLineDistance;
double dLineSin;
double dLineCos;
double dGap;
double dMinGap;
double dMaxGap;
```

```
if (!GetBestEdgeLineOnRegion (
    1,
    1,
    -100,100,100,100,
    &dLineDistance,
    &dLineSin,
    &dLineCos,
    &dGap,
    &dMinGap,
    &dMaxGap
))
{
    return;
}
```


GetBestEdgeCircleOnRegion

Fornisce il miglior cerchio sui bordi di un elemento

```
GVSBASEAPI int WINAPI GetBestEdgeCircleOnRegion(
    int          iType, // 1 = Blob 2=
    IntersectedShape
    int          iNumberBlobOrIntersectedShapeBySize ,
    int          iLeftRegion,
    int          iTopRegion,
    int          iWidthRegion,
    int          iHeightRegion,
    double       dPixelGap,
    double *     dCircleRadius,
    double *     dCircleXCenter,
    double *     dCircleYCenter,
    double *     dGap,
    double *     dMinGap,
    double *     dMaxGap
    );
```

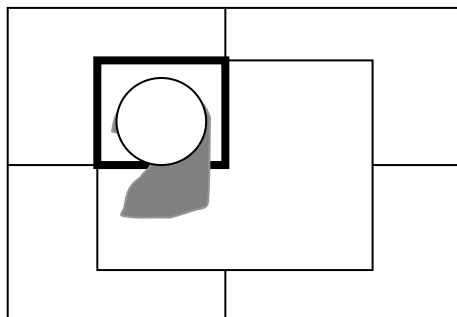
Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iType	Tipo di immagine: 1 = Blob 2= IntersectedShape
int iNumberBlobOrIntersectedShapeBySize	Numero della figura in dimensioni dalla più grande alla più piccola
int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse

double dPixelGap Valore di gap massimo ammesso
 double * dCircleRadius Ritorna il valore del raggio del cerchio
 double * dCircleXCenter Ritorna il valore della coordinata X del centro
 double * dCircleYCenter Ritorna il valore della coordinata X del centro
 double * dGap Ritorna l'errore di forma medio
 double * dMinGap Ritorna l'errore di forma minimo
 double * dMaxGap Ritorna l'errore di forma massimo



Esempio:

```

double dCircleRadius;
double dCircleXCenter;
double dCircleYCenter;
double dGap;
double dMinGap;
double dMaxGap;
  
```

```

if (!GetBestEdgeCircleOnRegion (
    1,
    1,
    -100,100,100,100,
    &dCircleRadius,
    &dCircleXCenter,
    &dCircleYCenter,
    &dGap,
    &dMinGap,
    &dMaxGap
))
{
    return;
}
  
```

PointsFromBestEdgeLineAcrossEdgeOnRegion

Fornisce i punti della miglior retta estratta sui bordi di un elemento

```
GVSBASEAPI int WINAPI PointsFromBestEdgeLineAcrossEdgeOnRegion(
    int         iTyPe, // 1 = Blob 2= IntersectedShape
    int         iNumberBlobOrIntersectedShapeBySize,
    int         iLeftRegion,
    int         iTopRegion,
    int         iWidthRegion,
    int         iHeightRegion,
    int         iNumberPointsMaxToLoad,
    double      dPixelGap,
    int         * piNumbersPointLoad,
    double      * pXCrossPoint,
    double      * pYCrossPoint
    );
```

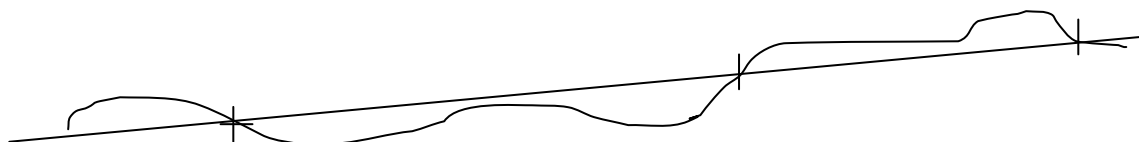
Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iTyPe	Tipo di immagine: 1 = Blob 2= IntersectedShape
int	iNumberBlobOrIntersectedShapeBySize	Numero della figura in dimensioni dalla più grande alla più piccola
int	iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iWidthRegion	Larghezza della regione d'interesse
int	iHeightRegion	Altezza della regione d'interesse

int	iNumberPointsMaxToLoad	Numero massimo di punti da caricare
double	dPixelGap	Valore di gap massimo ammesso
int *	piNumbersPointLoad	Ritorna il numero massimo di punti caricati
double *	pXCrossPoint	Vettore di coordinate X dei pixels di intersezione del bordo con la retta trovata
double *	pYCrossPoint	Vettore di coordinate Y dei pixels di intersezione del bordo con la retta trovata



Esempio:

```
#define NUMBER_MAX    1024

int    iNumberPixelLoad;
double pdXPixel[NUMBER_MAX];
double pdYPixel[NUMBER_MAX];

if (!PointsFromBestEdgeLineAcrossEdgeOnRegion (
    1,
    1,
    -100,100,100,100,
    NUMBER_MAX,
    2,
    &iNumberPixelLoad,
    pdXPixel,
    pdYPixel
))
{
    return;
}
```

MaximusFromBestEdgeLineAcrossEdgeOnRegion

Fornisce i massimi della miglior retta estratta sui bordi di un elemento

```
GVSBASEAPI int WINAPI MaximusFromBestEdgeLineAcrossEdgeOnRegion(
    int         iType, // 1 = Blob 2= IntersectedShape
    int         iNumberBlobOrIntersectedShapeBySize,
    int         iLeftRegion,
    int         iTopRegion,
    int         iWidthRegion,
    int         iHeightRegion,
    int         iNumberPointsMaxToLoad,
    int         * piNumbersPointLoad,
    double      * pXMaxPoint,
    double      * pYMaxPoint,
    double      * pValueDistance,
    double      * pXCrossPoint,
    double      * pYCrossPoint
    );
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

int iType	Tipo di immagine: 1 = Blob 2= IntersectedShape
int iNumberBlobOrIntersectedShapeBySize	Numero della figura in dimensioni dalla più grande alla più piccola
int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse

int	iNumberPointsMaxToLoad	Numero massimo di punti da caricare
double	dPixelGap	Valore di gap massimo ammesso
int *	piNumbersPointLoad	Ritorna il numero massimo di punti caricati
double *	pXMaxPoint	Vettore di coordinate X dei massimi di intersezione del bordo con la retta trovata
double *	pYMaxPoint	Vettore di coordinate Y dei massimi di intersezione del bordo con la retta trovata
double *	pValueDistance	Vettore delle distanze dei massimi trovati
double *	pXCrossPoint	Vettore di coordinate X dei pixels di intersezione del bordo con la retta trovata
double *	pYCrossPoint	Vettore di coordinate Y dei pixels di intersezione del bordo con la retta trovata



Esempio:

```
#define NUMBER_MAX    1024

int    iNumberPixelLoad;
double pdMaxXPixel[NUMBER_MAX];
double pdMaxYPixel[NUMBER_MAX];
double pdDistance[NUMBER_MAX];
double pdXPixel[NUMBER_MAX];
double pdYPixel[NUMBER_MAX];

if (!MaximusFromBestEdgeLineAcrossEdgeOnRegion (
    1,
    1,
    -100,100,100,100,
    NUMBER_MAX,
    &iNumberPixelLoad,
    pdMaxXPixel,
    pdMaxYPixel,
    pdDistance,
    pdXPixel,
    pdYPixel
))
{
    return;
}
```

PointsFromLine2PointAcrossEdgeOnRegion

Fornisce i punti della miglior retta estratta tra due punti sui bordi di un elemento

```
GVSBASEAPI int WINAPI PointsFromLine2PointAcrossEdgeOnRegion(
    int          iType, // 1 = Blob 2= IntersectedShape
    int          iNumberBlobOrIntersectedShapeBySize,
    int          iLeftRegion,
    int          iTopRegion,
    int          iWidthRegion,
    int          iHeightRegion,
    int          iXPoint1Region,
    int          iYPoint1Region,
    int          iXPoint2Region,
    int          iYPoint2Region,
    int          iNumberPointsMaxToLoad,
    double       dPixelGap,
    int          * piNumbersPointLoad,
    double       * pXCrossPoint,
    double       * pYCrossPoint
);
```

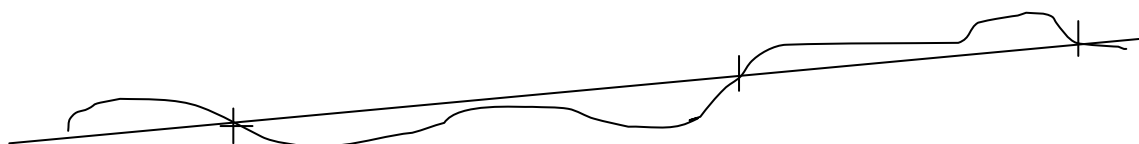
Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iType	Tipo di immagine: 1 = Blob 2= IntersectedShape
int	iNumberBlobOrIntersectedShapeBySize	Numero della figura in dimensioni dalla più grande alla più piccola
int	iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iWidthRegion	Larghezza della regione d'interesse
int	iHeightRegion	Altezza della regione d'interesse

int	iXPoint1Region	Coordinata X del primo punto
int	iYPoint1Region	Coordinata Y del primo punto
int	iXPoint2Region	Coordinata X del secondo punto
int	iYPoint2Region	Coordinata X del secondo punto
int	iNumberPointsMaxToLoad	Numero massimo di punti da caricare
double	dPixelGap	Valore di gap minimo
int *	piNumbersPointLoad	Ritorna il numero massimo di punti caricati
double *	pXCrossPoint	Vettore di coordinate X dei pixels di intersezione del bordo con la retta trovata
double *	pYCrossPoint	Vettore di coordinate Y dei pixels di intersezione del bordo con la retta trovata



Esempio:

```
#define NUMBER_MAX    1024

int    iNumberPixelLoad;
double pdXPixel[NUMBER_MAX];
double pdYPixel[NUMBER_MAX];

if (!PointsFromLine2PointAcrossEdgeOnRegion (
    1,
    1,
    -100,100,100,100,
    -100,0,50,5,
    NUMBER_MAX,
    2,
    &iNumberPixelLoad,
    pdXPixel,
    pdYPixel
))
{
    return;
}
```


MaximusFromLine2PointAcrossEdgeOnRegion

Fornisce i massimi della miglior retta estratta tra due punti sui bordi di un elemento

```
GVSBASEAPI int WINAPI MaximusFromLine2PointAcrossEdgeOnRegion(  
    int             iType, // 1 = Blob 2= IntersectedShape  
    int             iNumberBlobOrIntersectedShapeBySize,  
    int             iLeftRegion,  
    int             iTopRegion,  
    int             iWidthRegion,  
    int             iHeightRegion,  
    int             iXPoint1Region,  
    int             iYPoint1Region,  
    int             iXPoint2Region,  
    int             iYPoint2Region,  
    int             iNumberPointsMaxToLoad,  
    int             * piNumbersPointLoad,  
    double          * pXMaxPoint,  
    double          * pYMaxPoint,  
    double          * pValueDistance,  
    double          * pXCrossPoint,  
    double          * pYCrossPoint  
);
```

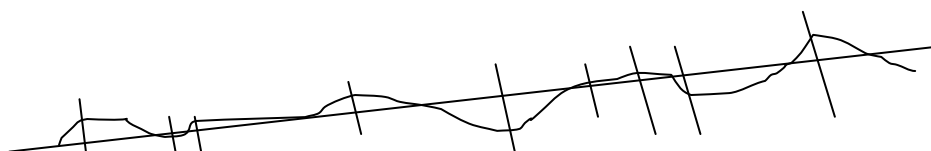
Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iType	Tipo di immagine: 1 = Blob 2= IntersectedShape
int	iNumberBlobOrIntersectedShapeBySize	Numero della figura in dimensioni dalla più grande alla più piccola
int	iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iWidthRegion	Larghezza della regione d'interesse
int	iHeightRegion	Altezza della regione d'interesse

int	iXPoint1Region	Coordinata X del primo punto
int	iYPoint1Region	Coordinata Y del primo punto
int	iXPoint2Region	Coordinata X del secondo punto
int	iYPoint2Region	Coordinata X del secondo punto
int	iNumberPointsMaxToLoad	Numero massimo di punti da caricare
double	dPixelGap	Valore di gap massimo ammesso
int	* piNumbersPointLoad	Ritorna il numero massimo di punti caricati
double	* pXMaxPoint	Vettore di coordinate X dei massimi di intersezione del bordo con la retta trovata
double	* pYMaxPoint	Vettore di coordinate Y dei massimi di intersezione del bordo con la retta trovata
double	* pValueDistance	Vettore delle distanze dei massimi trovati
double	* pXCrossPoint	Vettore di coordinate X dei pixels di intersezione del bordo con la retta trovata
double	* pYCrossPoint	Vettore di coordinate Y dei pixels di intersezione del bordo con la retta trovata



Esempio:

```

#define NUMBER_MAX      1024
int    iNumberPixelLoad;
double pdMaxXPixel[NUMBER_MAX];
double pdMaxYPixel[NUMBER_MAX];
double pdDistance[NUMBER_MAX];
double pdXPixel[NUMBER_MAX];
double pdYPixel[NUMBER_MAX];

if (!MaximusFromLine2PointAcrossEdgeOnRegion (
    1,
    1,
    -100,100,100,100,
    -100,0,50,5,
    NUMBER_MAX,
    &iNumberPixelLoad,
    pdMaxXPixel,
    pdMaxYPixel,
    pdDistance,
    pdXPixel,
    pdYPixel
))
{
    return;
}

```

Elements management

Le funzioni di questo capitolo effettuano delle operazioni sugli elementi

46- PointsAcross2Elements*	Fornisce il punto intersezione tra due elementi
47- PointFromLineAcrossLine	Fornisce il punto intersezione tra due rette
48- Points2FromLineAcrossRegion	Fornisce I due punti d'intersezione di una retta con il rettangolo di una regione
49- DrawElementLine	Disegna una retta
50- DrawLine	Disegna una retta
51- DrawElementCircle	Disegna un cerchio
52- DrawTextOnVideo *	Disegna un testo sul video
53- Draw2ElementsLine	Disegna due rette
54- SubPixelPosition	Fornisce il valore in subpixel di un pixel

PointsAcross2Elements

Fornisce il punto intersezione tra due rette

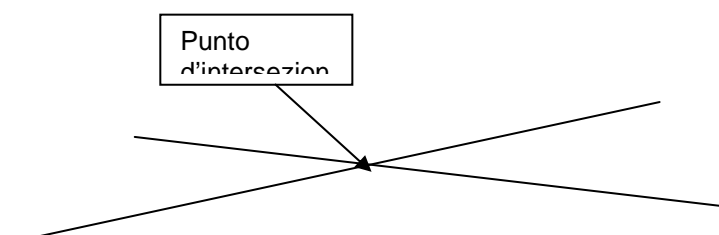
```
GVSBASEAPI int WINAPI PointsAcross2Elements (  
    int        iTypeFirstElement, // 1=line 2=radius  
    double     dA1, // dLineDistance dCircleRadius ,  
    double     dA2, // dLineSin dCircleXCenter  
    double     dA3, // dLineCos dCircleYCenter  
    int        iTypeSecondElement, // 1=line 2=radius  
    double     dB1, // dLineDistance dCircleRadius ,  
    double     dB2, // dLineSin dCircleXCenter  
    double     dB3, // dLineCos dCircleYCenter  
    int        * piNumberPoint,  
    double     * pPAX,  
    double     * pPAY,  
    double     * pPBX,  
    double     * pPBY,  
    double     * pDist );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

double dLineDistance1	Valore della distanza della 1 retta
double dLineSin1	Valore del seno della 1 retta
double dLineCos1	Valore del coseno della 1 retta
double dLineDistance2	Valore della distanza della 2 retta
double dLineSin2	Valore del seno della 2 retta
double dLineCos2	Valore del coseno della 2 retta
double * pPX	Ritorna il valore della coordinata X del punto d'intersezione
double * pPY	Ritorna il valore della coordinata Y del punto d'intersezione



Esempio:

```
double dXP;  
double dYP;  
  
if (!PointsAcross2Elements (      0,1,0,  
                                0,0,1,  
                                &dXP,  
                                &dYP  
                                ))  
{  
    return;  
}
```

Points2FromLineAcrossRegion

Fornisce i due punti d'intersezione di una retta con il rettangolo di una regione

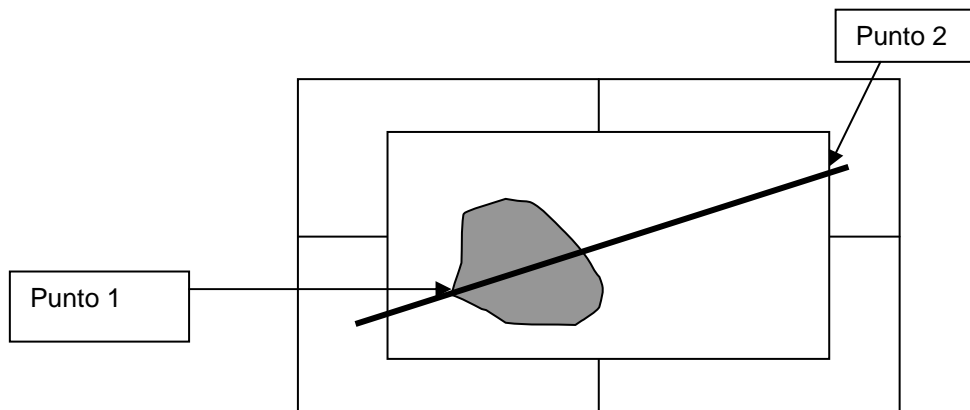
```
GVSBASEAPI int WINAPI Points2FromLineAcrossRegion(
    double    dLineDistance,
    double    dLineSin,
    double    dLineCos,
    int       iLeftRegion,
    int       iTopRegion,
    int       iWidthRegion,
    int       iHeightRegion,
    double *  pPX1 ,
    double *  pPY1 ,
    double *  pPX2 ,
    double *  pPY2
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

double dLineDistance	Valore della distanza della retta
double dLineSin	Valore del seno della retta
double dLineCos	Valore del coseno della retta
int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse
double * pPX1	Ritorna il valore della coordinata X del 1 punto
double * pPY1	Ritorna il valore della coordinata X del 1 punto
double * pPX2	Ritorna il valore della coordinata X del 2 punto
double * pPY2	Ritorna il valore della coordinata Y del 2 punto



Esempio:

```
double dPX1;
double dPY1;
double dPX2;
double dPY2;
```

```
if (!Points2FromLineAcrossRegion (0,1,0,
                                  -100,100,200,200,
                                  &dPX1,
                                  &dPY1,
                                  &dPX2,
                                  &dPY2
                                  ))
{
    return;
}
```

DrawElementLine

Disegna una retta

```
GVSBASEAPI int WINAPI DrawElementLine(
    double    dLineDistance,
    double    dLineSin,
    double    dLineCos,
    HDC       hdc
    int       iWindowsID,
    int       iHandle
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

double	dLineDistance	Valore della distanza della retta
double	dLineSin	Valore del seno della retta
double	dLineCos	Valore del coseno della retta
HDC	hdc	Device Context della Window
int	iWindowsID	Identificativo della finestra 0-15(Default 0)
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
if (!DrawElementLine ( 0,1,0,
                      m_cdc.m_hDC,
                      0,
                      0
                      ))
{
    return;
}
```


DrawLine

Disegna una retta

```
GVSBASEAPI int WINAPI DrawLine(  
    double dXFrom,  
    double dYFrom,  
    double dXTo,  
    double dYTo,  
    HDC     hdc,  
    int     iWindowsID,  
    int     iHandle  
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

double	dXFrom	Valore della X iniziale
double	dYFrom	Valore della Y iniziale
double	dXTo	Valore della X finale
double	dYTo	Valore della Y finale
HDC	hdc	Device Context della Window
int	iWindowsID	Identificativo della finestra 0-15(Default 0)
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
if (!DrawLine ( 0,1,0,45,  
                m_cdc.m_hDC,  
                0,  
                0  
                ))  
{  
    return;  
}
```

DrawElementCircle

Disegna un cerchio

```
GVSBASEAPI int WINAPI DrawElementCircle(
    double dCircleRadius,
    double dCircleXCenter,
    double dCircleYCenter,
    HDC     hdc
    int     iWindowsID,
    int     iHandle
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

double	dCircleRadius	Valore del raggio del cerchio
double	dCircleXCenter	Valore della coordinata X centro del cerchio
double	dCircleYCenter	Valore della coordinata Y centro del cerchio
HDC	hdc	Device Context della Window
int	iWindowsID	Identificativo della finestra 0-15(Default 0)
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
if (!DrawElementCircle ( 0,1,0,
                        m_cdc.m_hDC,
                        0,0
                        ))
{
    return;
}
```

DrawTextOnVideo

Disegna un cerchio

```
GVSBASEAPI int WINAPI DrawTextOnVideo (  
    double    dPosX,  
    double    dPosY,  
    int       iFlagStringa,  
    char *    pStringa,  
    int       iFlagAng,  
    int       iFlagAngMilliToSessa,  
    double    dAng,  
    int       iFlagX,  
    double    dX,  
    int       iFlagY,  
    double    dY,  
    int       FontNumber,  
    COLORREF  colore,  
    HDC       hdc,  
    int       iNumberTotalDigit,  
    int       iNumberDecimalDigit,  
    int       iWindowsID,  
    int       iHandle  
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

double	dCircleRadius	Valore del raggio del cerchio
double	dCircleXCenter	Valore della coordinata X centro del cerchio
double	dCircleYCenter	Valore della coordinata Y centro del cerchio
HDC	hdc	Device Context della Window
int	iWindowsID	Identificativo della finestra 0-15(Default 0)
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
if (!DrawTextOnVideo ( 0,1,0,  
                      m_cdc.m_hDC,0,0  
                      ))  
{  
    return;  
}
```

Draw2ElementsLine

Disegna due rette

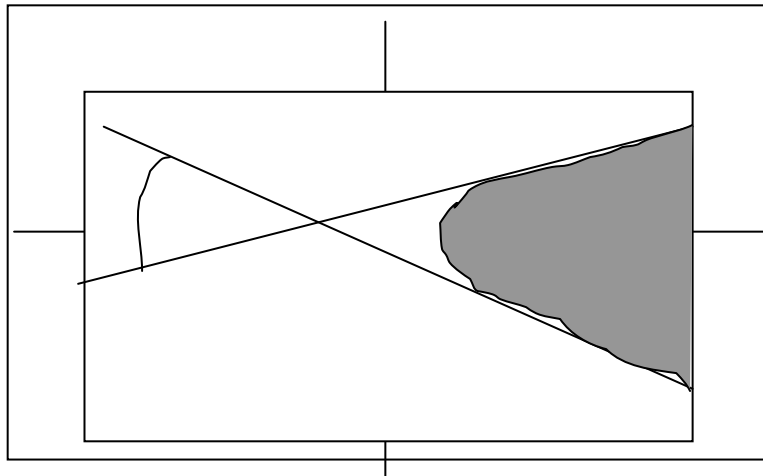
```
GVSBASEAPI int WINAPI Draw2ElementsLine(  
    double    dLineDistance1,  
    double    dLineSin1,  
    double    dLineCos1,  
    double    dLineDistance2,  
    double    dLineSin2,  
    double    dLineCos2,  
    int       iFlagWithArc,  
    double    dRadiusArc,  
    HDC       hdc,  
    int       iWindowsID,  
    int       iHandle  
);
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

double	dLineDistance1	Valore della distanza della 1 retta
double	dLineSin1	Valore del seno della 1 retta
double	dLineCos1	Valore del coseno della 1 retta
double	dLineDistance2	Valore della distanza della 2 retta
double	dLineSin2	Valore del seno della 2 retta
double	dLineCos2	Valore del coseno della 2 retta
int	iFlagWithArc	Flag se disegnare l'arco tra le rette
double	dRadiusArc	Valore del raggio dell'arco
HDC	hdc	Device Context della Window
int	iWindowsID	Identificativo della finestra 0-15(Default 0)
int	iHandle	Handle di camera o istanza (Default 0)



Esempio:

```
if (!Draw2ElementsLine ( 0,1,0,  
                        0,0,1,  
                        1,  
                        12,  
                        m_cdc.m_hDC  
                        ))  
{  
    return;  
}
```

SubPixelPosition

Fornisce il valore in subpixel di un pixel

```
GVSBASEAPI void SubPixelPosition(  
    int      ix,  
    int      iy,  
    double * pdX,  
    double * pdY,  
    int      iHandle  
);
```

Parametri

int ix	Valore della coordinata X del pixel
int iy	Valore della coordinata Y del pixel
double * pdX	Ritorna il valore della coordinata X del subpixel
double * pdY	Ritorna il valore della coordinata Y del subpixel
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

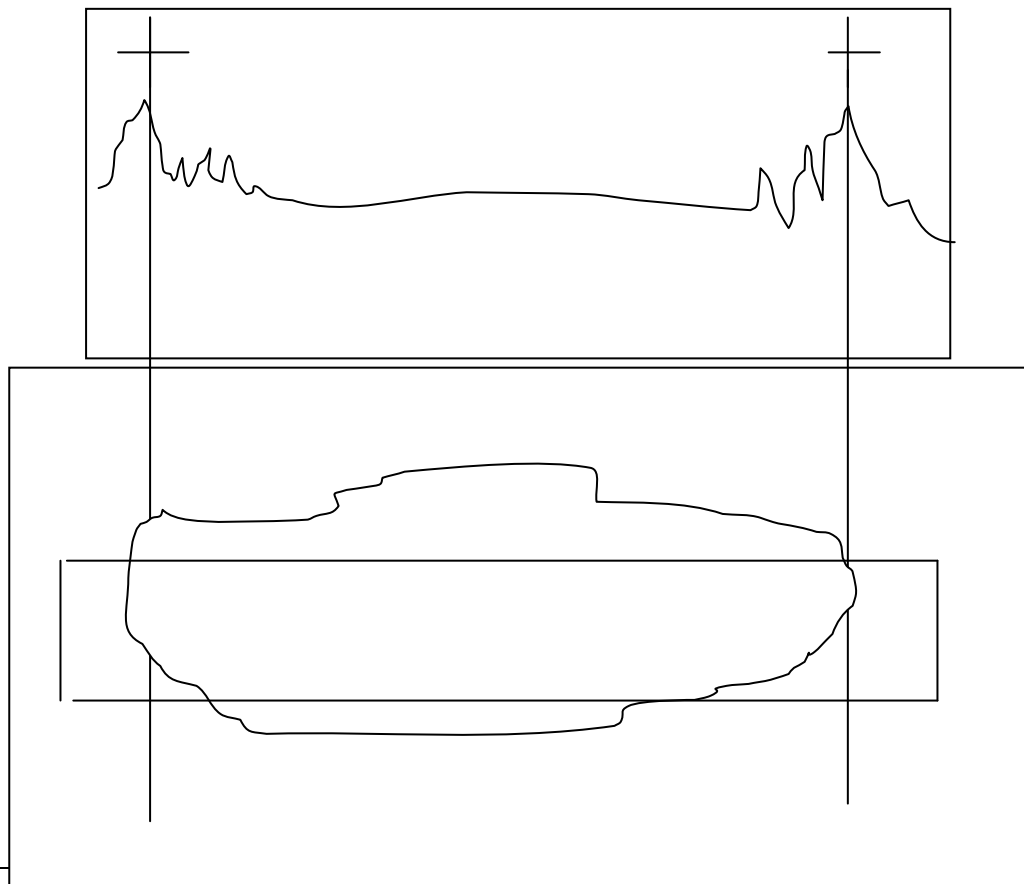
```
double dX;  
double dY;  
  
if (!SubPixelPosition ( 12,  
                        10,  
                        &dX,  
                        &dY,  
                        0  
                        ))  
{  
    return;  
}
```

Edge locator

Le funzioni del Edge Locator individuano i bordi su una banda lineare che attraversa l'immagine.

55- LineEdgeLocator	Estrae i bordi lungo una banda lineare
56- GetEdgeByPosition	Fornisce il bordo in base alla posizione
57- GetEdgeBySize	Fornisce il bordo in base alla dimensione
58- GetAllEdgeFind	Fornisce tutti i bordi estratti
59- GetAllSortEdgeFind	Fornisce tutti i bordi estratti ordinati secondo la dimensione
60- CircleByEdge	Genera un cerchio effettuando il best fit tra i punti definiti dai bordi estratti
61- EllipseByEdge	Genera un ellisse effettuando il best fit tra i punti definiti dai bordi estratti
62- FindAllConicsByEdge	Estrae le coniche in una zona circolare definita
63- GetValueEllipseConicByEdge	Fornisce i valori dell' ellisse individuata da un numero progressivo delle coniche estratte con la " FindAllConicsByEdge "

L'Edge Locator esegue la media del valore dei pixel in linea infine estrae il differenziale tra gli adiacenti (filtro 1) oppure tra i due adiacenti (filtro 2), dai differenziali maggiori possiamo individuare i bordi.



LineEdgeLocator

Estrae i bordi lungo una banda lineare

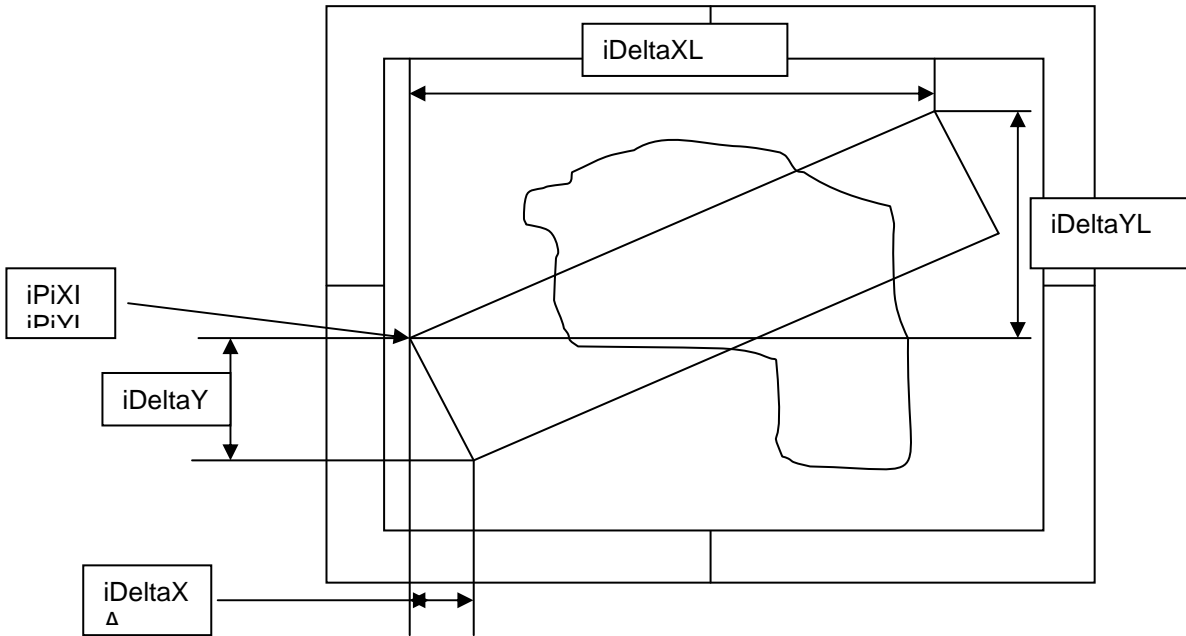
```
GVSBASEAPI int LineEdgeLocator(
    int iPiXI, int iPiYI, int
iDeltaXL, int iDeltaYL ,
    int iDeltaXA, int iDeltaYA ,
    int iFilter,
    int * piEdgeNumber ,
    int * piEdgeAverage,
    int iHandle
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iPiXI	Valore coordinata X punto iniziale
int iPiYI	Valore coordinata Y punto iniziale
int iDeltaXL	Valore delta X larghezza
int iDeltaYL	Valore delta Y larghezza
int iDeltaXA	Valore delta X altezza
int iDeltaYA	Valore delta Y altezza
int iFilter	Tipo di filtro o differenziale 1 o 2
int * piEdgeNumber	Ritorna il numero di bordi trovati nella fascia
int * piEdgeAverage	Ritorna il valore medio del bordo
int iHandle	Handle di camera o istanza (Default 0)



Esempio:

```
int iEdgeNumber;
int iEdgeAverage;
```

```
if (!LineEdgeLocator ( 12,
                      10,
                      100,
                      0,
                      0,
                      10,
                      1,
                      &iEdgeNumber,
                      &iEdgeAverage,
                      0
                    ))
{
    return;
}
```

GetEdgeByPosition

Fornisce il bordo in base alla posizione

```
GVSBASEAPI int GetEdgeByPosition(
    int          iNumberEdgebyPosition,
    int          iEdgeThreshold,
    int *        iPixelValue,
    int *        iX,
    int *        iY,
    int *        iDifferentialThreshold,
    double *     iDistance,
    int          iHandle
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iNumberEdgebyPosition	Numero della soglia per posizione
int	iEdgeThreshold	Valore di soglia minimo del differenziale
int *	iPixelValue	Ritorna il valore del pixel medio del bordo
int *	iX	Ritorna la coordinata X del mediano del bordo
int *	iY	Ritorna la coordinata Y del mediano del bordo
int *	iDifferentialThreshold	Ritorna il valore del differenziale medio del bordo
double *	iDistance	Ritorna la distanza del mediano del bordo dal inizio del mediano della banda di rilevamento
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
int iPixelValue;
int iX;
int iY;
int iDifferentialThreshold;
double iDistance;

if (!GetEdgeByPosition ( 1,
                        4,
                        &iPixelValue,
                        &iX,
                        &iY,
                        &iDifferentialThreshold,
                        &iDistance,
                        0
                      ))
{
    return;
}
```

GetEdgeBySize

Fornisce il bordo in base alla dimensione

```
GVSBASEAPI int GetEdgeBySize(
    int          iNumberEdgebySize,
    int *        iPixelValue,
    int *        iX,
    int *        iY,
    int *        iDifferentialThreshold,
    double *     iDistance,
    int          iHandle
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iNumberEdgebyPosition	Numero della soglia per dimensione
int *	iPixelValue	Ritorna il valore del pixel medio del bordo
int *	iX	Ritorna la coordinata X del mediano del bordo
int *	iY	Ritorna la coordinata Y del mediano del bordo
int *	iDifferentialThreshold	Ritorna il valore del differenziale medio del bordo
double *	iDistance	Ritorna la distanza del mediano del bordo dal inizio del mediano della banda di rilevamento
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
int  iPixelValue;
int  iX;
int  iY;
int  iDifferentialThreshold;
double iDistance;

if (!GetEdgeBySize ( 1,
                    &iPixelValue,
                    &iX,
                    &iY,
                    &iDifferentialThreshold,
                    &iDistance,
                    0
                  ))
{
    return;
}
```

GetAllEdgeFind

Fornisce tutti i bordi estratti

```
GVSBASEAPI int GetAllEdgeFind(  
    int          iMaximumNumber,  
    int *        iDifferentialThreshold ,  
    double *     iDistance,  
    int          iHandle  
);
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

int	iMaximumNumber	Numero massimo di bordi da caricare
int *	iDifferentialThreshold	Ritorna il valore del differenziale medio del bordo
double *	iDistance	Ritorna la distanza del mediano del bordo dal inizio del mediano della banda di rilevamento
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
int iDifferentialThreshold;  
double iDistance;  
  
if (!GetAllEdgeFind ( 100,  
                    &iDifferentialThreshold,  
                    &iDistance,  
                    0  
                    ))  
{  
    return;  
}
```

GetAllSortEdgeFind

Fornisce tutti i bordi estratti ordinati secondo la dimensione

```
GVSBASEAPI int GetAllSortEdgeFind(  
    int          iMaximumNumber,  
    int *        iPixelValue,  
    int *        iX,  
    int *        iY,  
    int *        iDifferentialThreshold,  
    double *     iDistance,  
    int          iHandle  
);
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

int	iMaximumNumber	Numero massimo di bordi da caricare
int *	iPixelValue	Ritorna il valore del pixel medio del bordo
int *	iX	Ritorna la coordinata X del mediano del bordo
int *	iY	Ritorna la coordinata Y del mediano del bordo
int *	iDifferentialThreshold	Ritorna il valore del differenziale medio del bordo
double *	iDistance	Ritorna la distanza del mediano del bordo dal inizio del mediano della banda di rilevamento
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
int  iPixelValue;
int  iX;
int  iY;
int  iDifferentialThreshold;
double iDistance;

if (!GetAllSortEdgeFind ( 100,
                        &iPixelValue,
                        &iX,
                        &iY,
                        &iDifferentialThreshold,
                        &iDistance,
                        0
                        ))
{
    return;
}
```

CircleByEdge

Fornisce il cerchio più performante di una serie di bordi

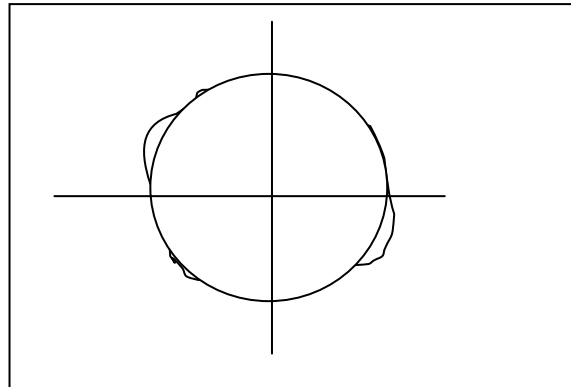
```
GVSBASEAPI int WINAPI CircleByEdge(
    double    dXCenterPoint,
    double    dYCenterPoint,
    double    dRadiusMax,
    double    dRadiusMin,
    double    dDegreeStartFirstArc,
    double    dDegreeEndFirstArc,
    double    dDegreeStartSecondArc,
    double    dDegreeEndSecondArc,
    double    dDifferenziale, // 7
    double    dPixelGap,     // 2
    double *  pdCircleRadius,
    double *  pdCircleXCenter,
    double *  pdCircleYCenter,
    double *  dGap,
    double *  dMinGap,
    double *  dMaxGap,
    int      iHandle
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

double	dXCenterPoint	Centro X da cui si dirama la ricerca dei bordi
double	dYCenterPoint	Centro X da cui si dirama la ricerca dei bordi
double	dRadiusMax	Raggio Massimo per la ricerca dei bordi
double	dRadiusMin	Raggio Minimo per la ricerca dei bordi
double	dDegreeStartFirstArc	Angolo di partenza primo settore
double	dDegreeEndFirstArc	Angolo finale primo settore
double	dDegreeStartSecondArc	Angolo di partenza secondo settore
double	dDegreeEndSecondArc	Angolo finale secondo settore
double	dDifferenziale	Valore di differenziale per definire il bordo
double	dPixelGap	Valore del Gap minimo
double *	pdCircleRadius	Ritorna il raggio del cerchio
double *	pdCircleXCenter	Ritorna la coordinata X del centro del cerchio
double *	pdCircleYCenter	Ritorna la coordinata Y del centro del cerchio
double *	dGap	Ritorna l'errore di forma medio
double *	dMinGap	Ritorna l'errore di forma minimo
double *	dMaxGap	Ritorna l'errore di forma massimo
int	iHandle	Handle di camera o istanza (Default 0)



Esempio:

```
double dCircleRadius;  
double dCircleXCenter;  
double dCircleYCenter;  
double dGap;  
double dMinGap;  
double dMaxGap;
```

```
if (!CircleByEdge (  
    0,          // dXCenterPoint,  
    0,          // dYCenterPoint,  
    150,        // dRadiusMax,  
    70,         // dRadiusMin,  
    150,        // dDegreeStartFirstArc,  
    210,        // dDegreeEndFirstArc,  
    330,        // dDegreeStartSecondArc,  
    390,        // dDegreeEndSecondArc,  
    150,        // dDifferenziale, // 7  
    2,          // dPixelGap,      // 2  
    &dCircleRadius,  
    &dCircleXCenter,  
    &dCircleYCenter,  
    &dGap,  
    &dMinGap,  
    &dMaxGap,  
    0 // iHandle  
)  
{  
    return;  
}
```

EllipseByEdge

Fornisce l'ellisse più performante di una serie di bordi

```
GVSBASEAPI int WINAPI EllipseByEdge (  
    double    dXCenterPoint,  
    double    dYCenterPoint,  
    double    dRadiusMax,  
    double    dRadiusMin,  
    double    dDegreeStartFirstArc,  
    double    dDegreeEndFirstArc,  
    double    dDegreeStartSecondArc,  
    double    dDegreeEndSecondArc,  
    double    dDifferenziale,  
    double    dPixelGap,  
    double *  pdXc,  
    double *  pdYc,  
    double *  pdAngA,  
    double *  pdRgA,  
    double *  pdAngB,  
    double *  pdRgB,  
    double *  pdX1,  
    double *  pdY1,  
    double *  pdX2,  
    double *  pdY2,  
    double *  pdX3,  
    double *  pdY3,  
    double *  pdX4,  
    double *  pdY4,  
    int       iDrawEllipse,  
    HDC       hdc,  
    int       iHandle  
);
```

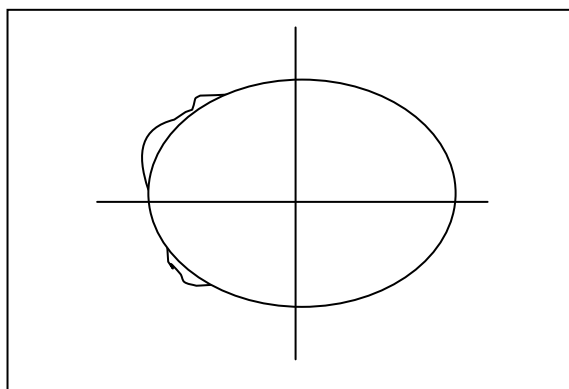
Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

double	dXCenterPoint	Centro X da cui si dirama la ricerca dei bordi
double	dYCenterPoint	Centro X da cui si dirama la ricerca dei bordi
double	dRadiusMax	Raggio Massimo per la ricerca dei bordi
double	dRadiusMin	Raggio Minimo per la ricerca dei bordi
double	dDegreeStartFirstArc	Angolo di partenza primo settore
double	dDegreeEndFirstArc	Angolo finale primo settore

double	dDegreeStartSecondArc	Angolo di partenza secondo settore
double	dDegreeEndSecondArc	Angolo finale secondo settore
double	dDifferenziale	Valore di differenziale per definire il bordo
double	dPixelGap	Valore del Gap minimo
double *	pdXc	Ritorna la coordinata X del centro dell'ellisse
double *	pdYc	Ritorna la coordinata Y del centro dell'ellisse
double *	pdAngA	Angolo A dell'asse maggiore
double *	pdRgA	Raggio maggiore
double *	pdAngB	Angolo B dell'asse minore
double *	pdRgB	Raggio minore
double *	pdX1	X del primo punto d' intersezione dell'asse maggiore
double *	pdY1	Y del primo punto d' intersezione dell'asse maggiore
double *	pdX2	X del secondo punto d' intersezione dell'asse maggiore
double *	pdY2	Y del secondo punto d' intersezione dell'asse maggiore
double *	pdX3	X del primo punto d' intersezione dell'asse minore
double *	pdY3	Y del primo punto d' intersezione dell'asse minore
double *	pdX4	X del secondo punto d' intersezione dell'asse minore
double *	pdY4	Y del secondo punto d' intersezione dell'asse minore
int	iDrawElisse	Flag disegna ellisse
HDC	hdc	Device Context della Window
int	iHandle	Handle di camera o istanza (Default 0)



Esempio:

```
double dXc;
double dYc;
double dAngA;
double dRgA;
double dAngB;
double dRgB;
double dX1;
double dY1;
double dX2;
double dY2;
double dX3;
double dY3;
double dX4;
double dY4;
if (!EllipseByEdge (
    0, // dXCenterPoint,
    0, // dYCenterPoint,
    150, // dRadiusMax,
    120, // dRadiusMin,
    0, // dDegreeStartFirstArc,
    180, // dDegreeEndFirstArc,
    181, // dDegreeStartSecondArc,
    360, // dDegreeEndSecondArc,
    80, // dDifferenziale,
    50, // dPixelGap,
    &dXc,
    &dYc,
    &dAngA,
    &dRgA,
    &dAngB,
    &dRgB,
    &dX1,
    &dY1,
    &dX2,
    &dY2,
    &dX3,
    &dY3,
    &dX4,
    &dY4,
    1, // iDrawEllipse,
    m_cdc.m_hDC // hdc,
    0 // iHandle
    )
{
    return;
}
```

FindAllConicsByEdge

Estrae le coniche in una zona circolare definita

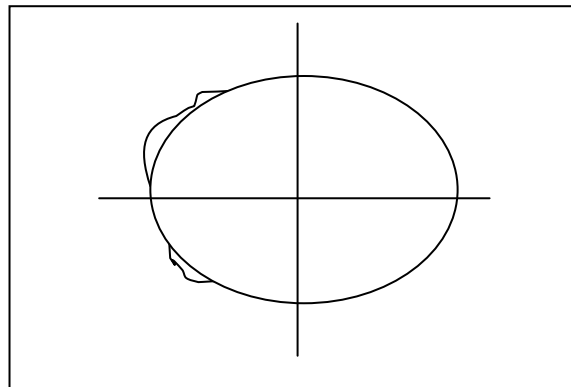
```
GVSBASEAPI int WINAPI FindAllConicsByEdge (  
    double    dXCenterPoint,  
    double    dYCenterPoint,  
    double    dRadiusMax,  
    double    dRadiusMin,  
    double    dPixelSizeLine,  
    double    dDifferenziale,  
    double    dOffsetDegree,  
    int       iHandle  
);
```

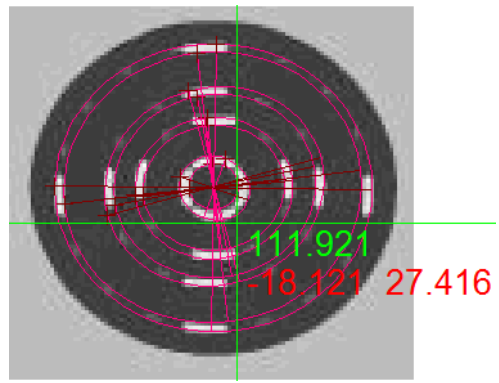
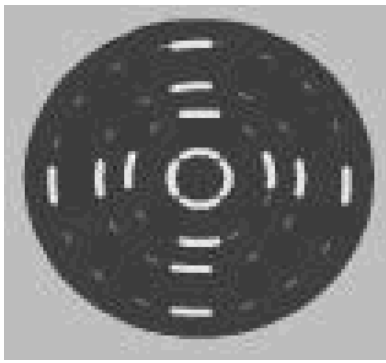
Valore di ritorno

Numero Coniche trovate
0 errore nella chiamata

Parametri

double	dXCenterPoint	Centro X da cui si dirama la ricerca dei bordi
double	dYCenterPoint	Centro X da cui si dirama la ricerca dei bordi
double	dRadiusMax	Raggio Massimo per la ricerca dei bordi
double	dRadiusMin	Raggio Minimo per la ricerca dei bordi
double	dPixelSizeLine	Angolo in gradi di ricerca per ogni ramo della croce in più ed in meno
double	dOffsetDegree	Offset in gradi della croce di ricerca
int	iHandle	Handle di camera o istanza (Default 0)





```
-DIM/120
-LET/G3,1
-DIS/1:
-DIS/2:
-DIS/3:
-DIS/4:
-LET/L1,3 // iNumberCameraActive
-ARI/1,12,VisAlgo,SetActiveCamera,L1,L1,L21,L21
```

pippo

```
-SWA/1
-LET/L1,0 // iTipoPalette,
-LET/L2,1 // iNumerolmagini,
-ARI/1,12,VisionBase,ReadImage,L1,L1,L21,L21,30000,pippo
```

```
-LET/L1,-345 // iLeft,
-LET/L2,200 // iTop,
-LET/L3,685 // iWidth,
-LET/L4,400 // iHeight,
-LET/L5,77 // iMode,
-LET/L6,2 // iModeLowParam,
-LET/L7,0 // iModeHighParam,
-LET/L8,0 // iTypeBlob
-LET/L9,200 // iLowArea
-LET/L10,0 // iTypeFilter
-LET/L11,0 // iFilterParam
-LET/L12,1 // iFlagDistanzaCentro
```

```
-ARI/1,12,VisAlgo,GetAllBlobs,L1,L12,L20,L28,3000,pippo
```

```
-LET/L1,1 // iNumberBlobBySize
-ARI/1,12,VisAlgo,GetValueBlob,L1,L1,L20,L28,3000,pippo
-DIS/1:XBaricenter,L21
-DIS/2:YBaricenter,L22
-DIS/3:Area,L23
```

```
-LET/L1,1 // iNumberBlobBySize ,
-LET/L2,0 // dPixelGap,
-ARI/1,12,VisAlgo,GetCircleBlob,L1,L2,L30,L48,3000,pippo
-DIS/4:Radius,L31
-DIS/5:XCenter,L32
-DIS/6:YCenter,L33
-DIS/7:Routness,L34
```

```
-LET/L11,1 // iNumberBlobBySize ,
-ARI/1,12,VisAlgo,GetEllipseBlob,L11,L11,L30,L44
```



```

-LET/L21,L31 // dXCenterPoint
-LET/L22,L32 // dYCenterPoint
-LET/L23,L36 // dRadiusMax
-LET/L24,5 // dRadiusMin
-LET/L25,10 // dSizeDegree +- Cross large
-LET/L26,150 // dDifferenziale
-LET/L27,0 // offsetDegree Start Cross
-ARI/1,12,VisAlgo,FindAllConicsByEdge,L21,L27,L41,L41,3000,pippo

```

```

-LET/L10,1
-LET/L11,L41

```

LoopD-

```

-JGT/L10,L11,Fii
-LET/L21,L10 // NumberConicFromCenter
-LET/L22,1 // DrawEllipse
-ARI/1,12,VisAlgo,GetValueEllipseConicByEdge,L21,L22,L41,L47,3000,pippo
-LET/L10,L10+1
-JMP/LoopD

```

Fii-

```

-LET/L29,L46 // Raggio minore ultima conica,
-LET/L31,0
-LET/L21,0
-LET/L22,0
-LET/L23,0 // iFlagStringa
-LET/L24,0 // pStringa,
-LET/L25,0 // iFlagAng,
-LET/L26,0 // iFlagAngMilliToSessa,
-LET/L27,0 // dAng,
-LET/L28,1 // iFlagX,
-LET/L30,0 // iFlagY,
-LET/L32,3 // Numero font,
-LET/L33,80 // colore,
-ARI/1,12,VisAlgo,DrawTextOnVideo,L21,L33,L34,L34,3000,pippo

```

```

-LET/L29,L41 // dX centro
-LET/L31,L42 // dY centro
-LET/L21,0 // iX,
-LET/L22,-30 // iY,
-LET/L23,0 // iFlagStringa
-LET/L24,0 // pStringa,
-LET/L25,0 // iFlagAng,
-LET/L26,0 // iFlagAngMilliToSessa,
-LET/L27,0 // dAng,
-LET/L28,1 // iFlagX,
-LET/L30,1 // iFlagY,
-LET/L32,3 // Numero font
-LET/L33,800 // colore,
-ARI/1,12,VisAlgo,DrawTextOnVideo,L21,L33,L34,L34,3000,pippo

```

```

-ARI/1,12,VisionBase,ShowImage,L1,L1,L20,L21,300,pipp

```

Esempio:

```
if (!FindAllConicsByEdge (
    0,      // dXCenterPoint,
    0,      // dYCenterPoint,
    150,    // dRadiusMax,
    120,    // dRadiusMin,
    4,      // dPixelSizeLine,
    150,    // dDifferenziale,
    0,      // dOffsetDegree,
    0       // iHandle
        )
{
    return;
}
```

GetValueEllipseConicByEdge

Fornisce i valori dell' ellisse individuata da un numero progressivo delle coniche estratte con la "FindAllConicsByEdge"

```
GVSBASEAPI int WINAPI GetValueEllipseConicByEdge (
    int          iNumberConicFromCenter,
    double * pdXc,
    double * pdYc,
    double * pdAngA,
    double * pdRgA,
    double * pdAngB,
    double * pdRgB,
    double * pdX1,
    double * pdY1,
    double * pdX2,
    double * pdY2,
    double * pdX3,
    double * pdY3,
    double * pdX4,
    double * pdY4,
    int          iDrawEllipse,
    unsigned int uRGBEdgeColor,
    HDC          hdc,
    int          iHandle
);
```

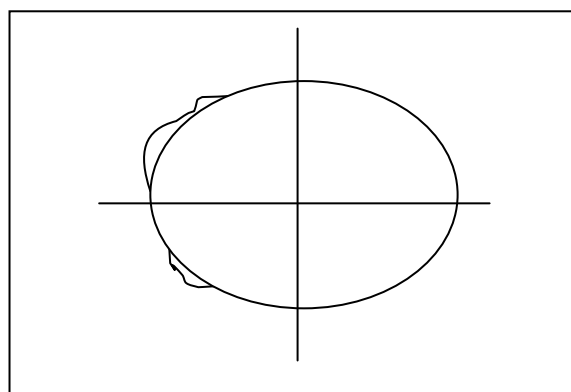
Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iNumberConicFromCenter Numero progressivo delle coniche estratte
double * pdXc Ritorna la coordinata X del centro dell'ellisse
double * pdYc Ritorna la coordinata Y del centro dell'ellisse
double * pdAngA Angolo A dell'asse maggiore
double * pdRgA Raggio maggiore
double * pdAngB Angolo B dell'asse minore
double * pdRgB Raggio minore
double * pdX1 X del primo punto d' intersezione dell'asse maggiore
double * pdY1 Y del primo punto d' intersezione dell'asse maggiore
double * pdX2 X del secondo punto d' intersezione dell'asse maggiore
double * pdY2 Y del secondo punto d' intersezione dell'asse maggiore
double * pdX3 X del primo punto d' intersezione dell'asse minore
double * pdY3 Y del primo punto d' intersezione dell'asse minore

double * pdX4 X del secondo punto d' intersezione dell'asse minore
double * pdY4 Y del secondo punto d' intersezione dell'asse minore
int iDrawElisse Flag disegna ellisse
unsigned int uRGBEdgeColor RGB della linea dell'ellisse disegnata
HDC hdc Device Context della Window
int iHandle Handle di camera o istanza (Default 0)



Esempio:

```
double dXc;  
double dYc;  
double dAngA;  
double dRgA;  
double dAngB;  
double dRgB;  
double dX1;  
double dY1;  
double dX2;  
double dY2;  
double dX3;  
double dY3;  
double dX4;  
double dY4;  
  
if (!GetValueEllipseConicByEdge (  
    1,  
    &dXc ,  
    &dYc ,  
    &dAngA ,  
    &dRgA ,  
    &dAngB ,  
    &dRgB ,  
    &dX1 ,  
    &dY1 ,  
    &dX2 ,  
    &dY2 ,  
    &dX3 ,  
    &dY3 ,  
    &dX4 ,  
    &dY4 ,  
    1, // iDrawEllipse,  
    0xF00 ,  
    m_cdc.m_hDC // hdc ,  
    0 // iHandle  
    )  
{  
    return;  
}
```

Fast Fourier tranform

Le funzioni di questo capitolo effettuano delle operazioni con la trasformata di Fourier

62- FFT2	Fast Fourier Trasformer 2 dimensioni
63- iFFT2	Inversa Fast Fourier Trasformer 2 dimensioni
64- LoadMemoryFFT2	Visualizza la FFT2
65- SetZeroPartialFFT2	Azzera parzialmente la Fast Fourier Trasformer
66- closeFFT2	Chiude la Fast Fourier Trasformer 2 dimensioni
67- Filter_iFFT2	Applica dei filtri alla Fast Fourier Trasformer
68- HighLowPassFilter_iFFT2	Applica un filtro passa basso o passo alto
69- Multy_iFTT2	Moltiplica due Fast Fourier Trasformer
70- GetFirsts3FrequencyFFT2	Prende le prime tre frequenze della Fast Fourier Trasformer
71- ClearBelowModulFFT2	Azzera la parte di FFT2 inferiore ad dato modulo

FFT2

Fast Fourier Trasform 2

```
GVSBASEAPI int FFT2 (  
    int      iLeft,  
    int      iTop,  
    int      iWidth,  
    int      iHeight,  
    int *    pHandleFFT2,  
    int      iHandle  
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iLeft	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iTop	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iWidth	Larghezza della regione d'interesse
int	iHeight	Altezza della regione d'interesse
int *	pHandleFFT2	Ritorna l'handler della FFT2
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
if (!FFT2 (  
    ix1,  
    iy1,  
    ilx,  
    ily,  
    &iHandleFFT2,  
    iCamera  
    ))  
{  
    return;  
}
```

iFFT2

Fast Fourier Trasform 2 inversa

```
GVSBASEAPI int iFFT2 (  
    int          iHandleFFT2  
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iHandleFFT2 Handle della trasformata 2

Esempio:

```
if (!iFFT2 (  
    iHandleFFT2  
))  
{  
    return;  
}
```


Multy_iFFT2

Moltiplica due trasformate

```
GVSBASEAPI int Multy_iFFT2 (  
    int iHandleFFTA,  
    int iHandleFFTB  
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iHandleFFTA Handle della trasformata A
int iHandleFFTB Handle della trasformata B

Esempio:

```
if (!Multy_iFFT2 (  
    iHandleFFTA  
    iHandleFFTB ))  
{  
    return;  
}
```

Filter_iFFT2

Applica dei filtri alla Fast Fourier Trasformer

```
GVSBASEAPI int Filter_iFFT2 (  
    int          iHandleFFT2,  
    double       K1Re,  
    double       K1Im,  
    double       K2Re,  
    double       K2Im,  
    double       K3Re,  
    double       K3Im,  
    double       K4Re,  
    double       K4Im    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iHandleFFT2 Handle della trasformata 2

Esempio:

```
Filter_iFFT2(  
    iHandleFFT2,  
    m_dValueRA,  
    m_dValueIA,  
    m_dValueRB,  
    m_dValueIB,  
    m_dValueRC,  
    m_dValueIC,  
    m_dValueRD,  
    m_dValueID  
);
```

HighLowPassFilter_iFFT2

Applica un filtro passa basso o passo alto

```
GVSBASEAPI int HighLowPassFilter_iFFT2 (  
    int          iHandleFFT2,  
    double       iRadius1,  
    double       iRadius2,  
    int          iZeroInternalRadius1,  
    int          iZeroBetwenRadius1Radius2,  
    int          iZeroExternalRadius2 );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iHandleFFT2 Handle della trasformata 2

Esempio:

```
HighLowPassFilter_iFFT2(  
    iHandleFFT2,  
    m_dValueR1,  
    m_dValueR2,  
    m_iCheckZeroR1,  
    m_iCheckR1r2,  
    m_iCheckR2  
);
```

SetZeroPartialFFT2

Azzera parzialmente la Fast Fourier Trasformer

```
GVSBASEAPI int SetZeroPartialFFT2 (
    int  iHandleFFT2,
    int  iBeginFirstRow,
    int  iEndFirstRow,
    int  iBeginLastRow,
    int  iEndLastRow,
    int  iBeginFirstCol,
    int  iEndFirstCol,
    int  iBeginLastCol,
    int  iEndLastCol );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iHandleFFT2 Handle della trasformata 2

Esempio:

```
SetZeroPartialFFT2(
    iHandleFFT2,
    0,            // iBeginFirstRow,
    2,            // iEndFirstRow,
    ily-1-2,      // iBeginLastRow,
    ily-1,        // iEndLastRow,
    0,            // iBeginFirstCol,
    2,            // iEndFirstCol,
    ilx/2+1-1-2, // iBeginLastCol,
    ilx/2+1-1     // iEndLastCol
);
```

LoadMemoryFFT2

Visualizza la FFT2

```
GVSBASEAPI int LoadMemoryFFT2 (  
    int iHandleFFT2,  
    double dLogaritmo  
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iHandleFFT2 Handle della trasformata

Esempio:

```
LoadMemoryFFT2(  
    iHandleFFT2,  
    2        // 1= Module  
           // 2= Phase  
);
```

GetFirsts3FrequencyFFT2

Prende le prime tre frequenze della Fast Fourier Trasformer

```
GVSBASEAPI int GetFirsts3FrequencyFFT2 (  
    int      iHandleFFT2,  
    double * pdReal1,  
    double * pdImm1,  
    double * pdAng1,  
    double * pdMod1,  
    int     * piXpos1,  
    int     * piYpos1,  
    double * pdReal2,  
    double * pdImm2,  
    double * pdAng2,  
    double * pdMod2,  
    int     * piXpos2,  
    int     * piYpos2,  
    double * pdReal3,  
    double * pdImm3,  
    double * pdAng3,  
    double * pdMod3,  
    int     * piXpos3,  
    int     * piYpos3  
);
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iHandleFFT2 Handle della trasformata 2

Esempio:

```
GetFirsts3FrequencyFFT2(  
    iHandleFFT2,  
    &dReal1,  
    &dImm1,  
    &dAng1,  
    &dMod1,  
    &iXpos1,  
    &iYpos1,  
    &dReal2,  
    &dImm2,  
    &dAng2,  
    &dMod2,  
    &iXpos2,  
    &iYpos2,  
    &dReal3,  
    &dImm3,  
    &dAng3,  
    &dMod3,  
    &iXpos3,  
    &iYpos3  
);
```

ClearBelowModulFFT2

Azzera la parte di FFT2 inferiore ad dato modulo

```
GVSBASEAPI int ClearBelowModulFFT2 (  
    int iHandleFFT2,  
    double dValPerc );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iHandleFFT2 Handle della trasformata 2

Esempio:

```
ClearBelowModulFFT2(  
    iHandleFFT2,  
    m_dValueVTP  
);
```


closeFFT2

Chiude la Fast Fourier Trasformer 2 dimensioni

```
GVSBASEAPI int closeFFT2 (  
    int iHandleFFT2,  
    );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iHandleFFT2 Handle della trasformata 2

Esempio:

```
closeFFT2(  
    iHandleFFT2  
    );
```

Estrazione di soglie e varie

Le funzioni di questo capitolo effettuano delle operazioni come l'estrazione di soglie ed altre funzioni varie

72- GetThreshold	Acquisisce una soglia
73- SetThreshold	Definisce la soglia
74- SaveFileTiff	Salva un'immagine come file Tiff
75- EdgeDetection	Definisce un bordo
76- ConvertMultiThreshold	Converte una ROI con una MultiSoglia
77- SetOffsetCenterImage	Defisce un offset al centro di una immagine
78- GetOffsetCenterImage	Acquisisce l' offset al centro di una immagine
79- EdgeOnMaskRegion	Best Fit and MultiPoints on Edge
80- GetXYAxesRectBlob	Definisce un rettangolo XY

GetThreshold

Acquisisce una soglia

```
GVSBASEAPI int GetThreshold (  
    int iLeftRegion,  
    int iTopRegion,  
    int iWidthRegion,  
    int iHeightRegion  
    int method,  
    int iPerc,  
    int iHandle );
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

int	iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iWidthRegion	Larghezza della regione d'interesse
int	iHeightRegion	Altezza della regione d'interesse
int	iMode	Modalità per l'acquisizione dei blobs 0: // Fuzzy ENTROPY Shannon 1: // Fuzzy Yager 2: // Iterative selection 3: // Grey level histograms 4: // Johanssen method for using entropy 5: // Kapur method for using entropy 6: // The two histogram peaks 7: // Minimum error thresholding 8: // Mean 9: // Pun method for using entropy 10: // The black percentage 11: // The two histogram peaks 21: // Thr_relax 22: // Multisoglia
int	iPerc	Percentuale di variazione della soglia trovata

int iHandle Handle di camera o istanza (Default 0)

Esempio:

```
GetThreshold(  
    iLeftRegion,  
    iTopRegion,  
    iWidthRegion,  
    iHeightRegion,  
    method,  
    iPerc  
    iCamera  
    );
```

SetThreshold

Definisce la soglia

```
GVSBASEAPI int SetThreshold (  
    int iLeftRegion,  
    int iTopRegion,  
    int iWidthRegion,  
    int iHeightRegion  
    int method,  
    int iPerc,  
    int iHandle );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iWidthRegion	Larghezza della regione d'interesse
int	iHeightRegion	Altezza della regione d'interesse
int	iMode	Modalità per l'acquisizione dei blobs 0: // Fuzzy ENTROPY Shannon 1: // Fuzzy Yager 2: // Iterative selection 3: // Grey level histograms 4: // Johanssen method for using entropy 5: // Kapur method for using entropy 6: // The two histogram peaks 7: // Minimum error thresholding 8: // Mean 9: // Pun method for using entropy 10: // The black percentage 11: // The two histogram peaks 21: // Thr_relax 22: // Multisoglia
int	iPerc	Percentuale di variazione della soglia trovata
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
SetThreshold(  
    iLeftRegion,  
    iTopRegion,  
    iWidthRegion,  
    iHeightRegion,  
    method,  
    iPerc  
    iCamera  
);
```

SaveFileTiff

Salva un'immagine come file Tiff

```
GVSBASEAPI int SaveFileTiff (
                                int      iLeftRegion,
                                int      iTopRegion,
                                int      iWidthRegion,
                                int      iHeightRegion,
                                char *   sPathAndNameFileTiff,
                                int      iHandle   );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iLeftRegion Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion Larghezza della regione d'interesse
int iHeightRegion Altezza della regione d'interesse
char * sPathAndNameFileTiff Nome del file su cui salvare l'area
int iHandle Handle di camera o istanza (Default 0)

Esempio:

```
SaveFileTiff(
    ix1,
    iy1,
    ilx,
    ily,
    "\\PROVA.TIF",
    iCamera
);
```

EdgeDetection

Definisce un bordo

```
GVSBASEAPI int EdgeDetection (  
    int iLeftRegion,  
    int iTopRegion,  
    int iWidthRegion,  
    int iHeightRegion,  
    int method,  
    double s,  
    int low,  
    int high,  
    int iHandle );
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

int	iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iWidthRegion	Larghezza della regione d'interesse
int	iHeightRegion	Altezza della regione d'interesse
int	imethod	Metodo
double	s	Soglia
int	low	basso
int	high	alto
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
EdgeDetection(  
    ix1,  
    iy1,  
    ilx,  
    ily,  
    m_iTipoEdge,  
    m_dValoreSigma,  
    0,  
    -1,  
    iCamera  
);
```

ConvertMultiThreshold

Converte una ROI con una MultiSoglia

```
GVSBASEAPI int ConvertMultiThreshold (
                                int iLeftRegion,
                                int iTopRegion,
                                int iWidthRegion,
                                int iHeightRegion,
                                int method,
                                int iHandle );
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

int	iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iWidthRegion	Larghezza della regione d'interesse
int	iHeightRegion	Altezza della regione d'interesse
int	imethod	Metodo
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
if (!ConvertMultiThreshold (
                                ix1,
                                iy1,
                                ilx,
                                ily,
                                m_iTipoSoglia,
                                iCamera
                                ))
{
    return;
}
```


SetOffsetCenterImage

Defisce un offset al centro di una immagine

```
GVSBASEAPI int SetOffsetCenterImage (
                                double dX,
                                double dY,
                                int    iHandle );
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

double dX	Coordinata X dell'offset del centro dell'immagine
double dY	Coordinata Y dell'offset del centro dell'immagine
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
if (!SetOffsetCenterImage ( 100,
                            27,
                            iCamera
                            ))
{
    return;
}
```

GetOffsetCenterImage

Acquisisce l' offset al centro di una immagine

```
GVSBASEAPI int GetOffsetCenterImage (  
    double * dX,  
    double * dY,  
    int     iHandle    );
```

Valore di ritorno

1	OK
0	errore nella chiamata

Parametri

double * pdX	Ritorna la coordinata X dell'offset del centro dell'immagine
double * pdY	Ritorna la coordinata Y dell'offset del centro dell'immagine
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
double dX;  
double dY;  
  
if (!GetOffsetCenterImage (  
    &dX,  
    &dY,  
    iCamera  
    ))  
{  
    return;  
}
```

EdgeOnMaskRegion

Best Fit and MultiPoints on Edge

```

GVSBASEAPI int EdgeOnMaskRegion (
    int             iType, // 1 = Blob 2= IntersectedShape
    int             iNumberBlobOrIntersectedShapeBySize,
    int             iLeftRegionA,
    int             iTopRegionA,
    int             iWidthRegionA,
    int             iHeightRegionA,
    int             iLeftRegionB,
    int             iTopRegionB,
    int             iWidthRegionB,
    int             iHeightRegionB,
    int             iFlagDrawRegionA,
    int             iFlagDrawRegionB,
    int             iFlagCoordinateROIA,
    int             iFlagCoordinateROI B,
    HDC             hdc,
    int *           piNumberPointsBorder,
    int *           piNumberPointsLoadInRegionA,
    int *           piNumberPointsLoadInRegionB,
    int *           piNumberPointsLoadOutRegionAB,
    int *           piNumberPointsLoadInRegionAB,
    double *        pdPerCentPointsLoadInRegionA,
    double *        pdPerCentPointsLoadInRegionB,
    double *        pdPerCentPointsLoadOutRegionAB,
    double *        pdPerCentPointsLoadInRegionAB,
    int             iWindowsID );
  
```

Valore di ritorno

1 OK

0 errore nella chiamata

Parametri

int iType Tipo di immagine: 1 = Blob 2= IntersectedShape

int iNumberBlobOrIntersectedShapeBySize Numero della figura in
dimensioni dalla più grande alla
più piccola

int iLeftRegionA Coordinata X rispetto al centro dell'immagine del punto
superiore a sinistra della regione d'interesse

int iTopRegionA	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegionA	Larghezza della regione d'interesse
int iHeightRegion A	Altezza della regione d'interesse
int iLeftRegionB	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegionB	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegionB	Larghezza della regione d'interesse
int iHeightRegion B	Altezza della regione d'interesse
int iFlagDrawRegionA	Richiesta disegno della Regione A
int iFlagDrawRegionB	Richiesta disegno della Regione B
int iFlagCoordinateROIA	Richiesta disegno delle coordinate ROI A
int iFlagCoordinateROIB	Richiesta disegno delle coordinate ROI B
HDC hdc	Device Context della Window
int * piNumberPointsBorder	
int * piNumberPointsLoadInRegionA	
int * piNumberPointsLoadInRegionB	
int * piNumberPointsLoadOutRegionAB	
int * piNumberPointsLoadInRegionAB	
double * pdPerCentPointsLoadInRegionA	
double * pdPerCentPointsLoadInRegionB	
double * pdPerCentPointsLoadOutRegionAB	
double * pdPerCentPointsLoadInRegionAB	
int iWindowsID	Identificativo della finestra 0-15(Default 0)

Esempio:

```
iRitorno = EdgeOnMaskRegion(  
    iType, // 1 = Blob 2= Intersecte  
    iNumberBlobOrIntersectedShapeBySi  
    iLeftRegionA,  
    iTopRegionA,  
    iWidthRegionA,  
    iHeightRegionA,  
    iLeftRegionB,  
    iTopRegionB,  
    iWidthRegionB,  
    iHeightRegionB,  
    iFlagDrawRegionA,  
    iFlagDrawRegionB,  
    iFlagCoordinateROIA,  
    iFlagCoordinateROIB,  
    m_Visio.m_cdc.m_hDC,  
    piNumberPointsBorder,  
    piNumberPointsLoadInRegionA,  
    piNumberPointsLoadInRegionB,  
    piNumberPointsLoadOutRegionAB,  
    piNumberPointsLoadInRegionAB,  
    pdPerCentPointsLoadInRegionA,  
    pdPerCentPointsLoadInRegionB,  
    pdPerCentPointsLoadOutRegionAB,  
    pdPerCentPointsLoadInRegionAB,  
    iWindowsID  
);
```

GetXYAxesRectBlob

Definisce un rettangolo XY

```
GVSBASEAPI int WINAPI GetXYAxesRectBlob(
    int          iNumberBlobBySize,
    double       dPixelGap,
    double *     pdXRectHighRight,
    double *     pdYRectHighRight,
    double *     pdXRectLowRight,
    double *     pdYRectLowRight,
    double *     pdXRectHighLeft,
    double *     pdYRectHighLeft,
    double *     pdXRectLowLeft,
    double *     pdYRectLowLeft,
    double *     pdXRightAlongBaricenterY,
    double *     pdYRightAlongBaricenterY,
    double *     pdXLeftAlongBaricenterY,
    double *     pdYLeftAlongBaricenterY,
    double *     pdDiameter
    );
```

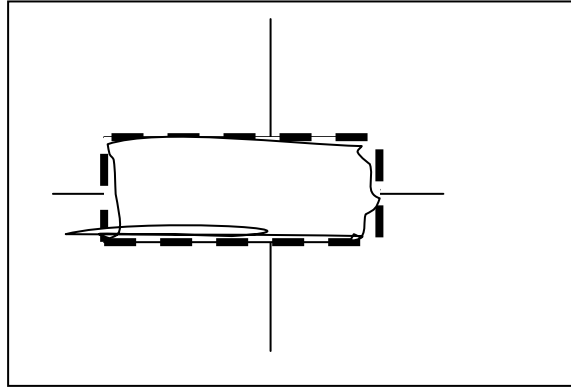
Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iNumberBlobBySize Numero del blob in dimensioni dal più grande al più piccolo
double dPixelGap Valore di gap massimo ammesso
double * pdXRectHighRight Coordinata X del punto alto a destra
double * pdYRectHighRight Coordinata Y del punto alto a destra
double * pdXRectLowRight Coordinata X del punto basso a destra
double * pdYRectLowRight Coordinata Y del punto basso a destra
double * pdXRectHighLeft Coordinata X del punto alto a sinistra
double * pdYRectHighLeft Coordinata Y del punto alto a sinistra
double * pdXRectLowLeft Coordinata X del punto basso a sinistra
double * pdYRectLowLeft Coordinata Y del punto basso a sinistra
double * pdXRightAlongBaricenterY Coordinata X del punto a destra lungo il baricentro coordinata Y
double * pdYRightAlongBaricenterY Coordinata Y del punto a destra lungo il baricentro coordinata Y

double * pdXLeftAlongBaricenterY Coordinata X del punto a sinistra lungo
il baricentro coordinata Y
double * pdYLeftAlongBaricenterY Coordinata Y del punto a sinistra lungo
il baricentro coordinata Y
double * pdDiameter Diametro misurato attraverso la media sul
baricentro coordinata Y



Esempio:

```
double dXRectHighRight;
double dYRectHighRight;
double dXRectLowRight;
double dYRectLowRight;
double dXRectHighLeft;
double dYRectHighLeft;
double dXRectLowLeft;
double dYRectLowLeft;
double dXRightAlongBaricenterY;
double dYRightAlongBaricenterY;
double dXLeftAlongBaricenterY;
double dYLeftAlongBaricenterY;
double dDiameter;

if (!GetXYAxesRectBlob ( 1, 2,
                        &dRectHighRight,
                        &dRectHighRight,
                        &dXRectLowRight,
                        &dYRectLowRight,
                        &dXRectHighLeft,
                        &dYRectHighLeft,
                        &dXRectLowLeft,
                        &dYRectLowLeft,
                        &dXRightAlongBaricenterY,
                        &dYRightAlongBaricenterY,
                        &dXLeftAlongBaricenterY,
                        &dYLeftAlongBaricenterY,
                        &dDiameter
                        ))
{
    return;
}
```

Pattern Matching

Le funzioni di questo capitolo effettuano delle operazioni di comparazione e ricerca di figure apprese

81- PMFind	Esegue la ricerca dei pattern
82- PMCreateTemplate	Crea un template per il Pattern Matching
83- PMSaveTemplate	Salva un template per il Pattern Matching
84- PMLoadTemplate	Carica un template per il Pattern Matching
85- PMGetMatchingData	Fornisce tutti dati del pattern riconosciuto tramite l'elemento
86- PMGetParameter	Fornisce il valore di un parametro del pattern matching
87- PMSetParameter	Imposta il valore di un parametro del pattern matching
88- PMGetDefaultParameters	Ripristina i valori di Default
89- PMGetTemplateImage	Restituisce l'immagine del Template
90- PMDraw	Disegna il pattern riconosciuto secondo l'elemento richiesto

PMFind

Questa funzione trova tutte le figure uguali al template caricato prima e ne restituisce il numero unico parametro di uscita i parametri di ingresso sono: Left Top larghezza e altezza della ROI in cui cercare, l'angolo di partenza e l'angolo finale di ricerca minore e' l'angolo, piu' veloce e' l'algoritmo.

L'ultimo parametro indica quanto precisa deve essere la ricerca i valori sono compresi tra 0 e 1

1 = match perfetto (consiglio 0.8)

```
GVSBASEAPI int PMFind(
                                int iLeft,
                                int iTop,
                                int iWidth,
                                int iHeight,
                                double dStartAngle = 0,
                                double dEndAngle = 0,
                                double dMinScore = 0.8,
                                HDC hdc = NULL,
                                int iHandle = -1
                                )
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int iWidthRegion	Larghezza della regione d'interesse
int iHeightRegion	Altezza della regione d'interesse
double dStartAngle	Angolo di partenza
double dEndAngle	Angolo finale di ricerca
double dMinScore	Precisione di ricerca 1 match perfetto
HDC hdc	Device Context della Window
int iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
if (!PMFind(
    -380, // iLeft,
    280,  // iTop,
    760,  // iWidth,
    560,  // iHeight,
    0,    // dStartAngle = 0,
    0,    // dEndAngle = 0,
    0.8,  // dMinScore = 0.8,
    NULL, // hdc
    -1    // iHandle = -1
))
{
    return;
}
```

PMCreateTemplate

Questa funzione definisce un Template a cui è abbinata la ROI di lavoro

```
GVSBASEAPI int PMCreateTemplate(
                                int iLeft,
                                int iTop,
                                int iWidth,
                                int iHeight,
                                int iHandle = -1
                                )
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int	iLeftRegion	Coordinata X rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iTopRegion	Coordinata Y rispetto al centro dell'immagine del punto superiore a sinistra della regione d'interesse
int	iWidthRegion	Larghezza della regione d'interesse
int	iHeightRegion	Altezza della regione d'interesse
int	iHandle	Handle di camera o istanza (Default 0)

Esempio:

```
if (!PMCreateTemplate(
    -380, // iLeft,
    280,  // iTop,
    760,  // iWidth,
    560,  // iHeight,
    -1    // iHandle = -1
))
{
    return;
}
```


PMSaveTemplate

Questa funzione salva il Template di riferimento

```
GVSBASEAPI int PMSaveTemplate(  
                                char * sPathAndNameFileBmp  
                                )
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

char * sPathAndNameFileBmp Path e nome del file.bmp del template

Esempio:

```
if (!PMSaveTemplate(  
    "C:\\\\TemplateImg1.bmp"  
    ))  
{  
    return;  
}
```

PMLoadTemplate

Questa funzione acquisisce il Template di riferimento

```
GVSBASEAPI int PMLoadTemplate(  
                                char * sPathAndNameFileBmp  
                                )
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

char * sPathAndNameFileBmp Path e nome del file.bmp del template

Esempio:

```
if (!PMLoadTemplate(  
    "C:\\\\TemplateImg1.bmp"  
    ))  
{  
    return;  
}
```

PMGetMatchingData

Questa funzione ritorna i parametri delle figure trovate come unico parametro di ingresso ha il numero della figura in uscita i parametri sono i seguenti:

0=parametri assenti 1=parametri presenti

iXCenter = centro in X della figura trovata

iYCenter = centro in Y della figura trovata

dAngle = angolo della figura trovata in gradi

dScore = maggiore o uguale alla precisione di ricerca impostata indica il valore di match trovato (1=match perfetto)

iLeft = coordinata sinistra del rettangolo contenete la figura trovata (non ruotata)

iTop = coordinata superiore del rettangolo contenete la figura trovata (non ruotata)

iRight = coordinata destra del rettangolo contenete la figura trovata (non ruotata)

iBottom = coordinata inferiore del rettangolo contenete la figura trovata (non ruotata)

iX1R = coordinata X superiore sinistra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)

iY1R = coordinata Y superiore sinistra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)

iX2R = coordinata superiore destra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)

iY2R = coordinata superiore destra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)

iX3R = coordinata inferiore sinistra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)

iY3R = coordinata inferiore sinistra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)

iX4R = coordinata inferiore destra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)

iY4R = coordinata inferiore destra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)

Tutte le coordinate in ingresso e uscita funzionano come quelle della blob

```

GVSBASEAPI int PMGetMatchingData (
    int iNumElement,
    int * iXCenter,
    int * iYCenter,
    double * dAngle,
    double * dScore,
    int * iLeft,
    int * iTop,
    int * iRight,
    int * iBottom,
    int * iX1R,
    int * iY1R,
    int * iX2R,
    int * iY2R,
    int * iX3R,
    int * iY3R,
    int * iX4R,
    int * iY4R
)

```

Valore di ritorno

1 OK
 0 = parametri assenti
 1 = parametri presenti

Parametri

int iNumElement	Numero dell'elemento
int * iXCenter	Centro in X della figura trovata
int * iYCenter	Centro in Y della figura trovata
double * dAngle	Angolo della figura trovata in gradi
double * dScore	Maggiore o uguale alla precisione di ricerca impostata indica il valore di match trovato (1=match perfetto)
int * iLeft	Coordinata sinistra del rettangolo contenete la figura trovata (non ruotata)
int * iTop	Coordinata superiore del rettangolo contenete la figura trovata (non ruotata)
int * iRight	Coordinata destra del rettangolo contenete la figura trovata (non ruotata)
int * iBottom	Coordinata inferiore del rettangolo contenete la figura trovata (non ruotata)

int * iX1R	Coordinata X superiore sinistra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)
int * iY1R	Coordinata Y superiore sinistra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)
int * iX2R	Coordinata X superiore destra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)
int * iY2R	Coordinata Y superiore destra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)
int * iX3R	Coordinata X inferiore sinistra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)
int * iY3R	Coordinata Y inferiore sinistra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)
int * iX4R	Coordinata X inferiore destra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)
int * iY4R	Coordinata Y inferiore destra del rettangolo contenete la figura trovata (ruotata dell'angolo trovato)

Esempio:

```
int    iXCenter;
int    iYCenter;
double dAngle;
double dScore;
int    iLeft;
int    iTop;
int    iRight;
int    iBottom;
int    iX1R;
int    iY1R;
int    iX2R;
int    iY2R;
int    iX3R;
int    iY3R;
int    iX4R;
int    iY4R;

if (!PMGetMatchingData (
    ))
{
    1 // iNumElement,
    &iXCenter,
    &iYCenter,
    &dAngle,
    &dScore,
    &iLeft,
    &iTop,
    &iRight,
    &iBottom,
    &iX1R,
    &iY1R,
    &iX2R,
    &iY2R,
    &iX3R,
    &iY3R,
    &iX4R,
    &iY4R)
}
```

PMGetParameter

Fornisce il valore di un parametro del pattern matching

```
GVSBASEAPI int PMGetParameter(  
                                int iNumParameter,  
                                double * dValue,  
                                char * sDescription  
                                )
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iNumParameter Numero del parametro
double * dValue Ritorno il valore del parametro
char * sDescription Descrizione

Esempio:

```
double dValue;  
char csDescription[128];  
  
if (!PMGetParameter (  
    ))  
{  
    1, // iNumParameter,  
    &dValue,  
    &csDescription  
    return;  
}
```

PMGetParameter

Fornisce il valore di un parametro del pattern matching

```
GVSBASEAPI int PMGetParameter(
                                int iNumParameter,
                                double * dValue,
                                char * sDescription
                                )
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iNumParameter Numero del parametro
double * dValue Ritorno il valore del parametro
char * sDescription Descrizione

Esempio:

```
double dValue;
char    sDescription[128];

if (!PMGetParameter (
    ))
{
    1, // iNumParameter,
    &dValue,
    &sDescription
}
return;
```


PMSetParameter

Imposta il valore di un parametro del pattern matching

```
GVSBASEAPI int PMSetParameter(  
                                int iNumParameter,  
                                double dValue  
                                )
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iNumParameter Numero del parametro
double dValue Valore del parametro

Esempio:

```
if (!PMSetParameter (  
    ))  
{  
    1, // iNumParameter,  
    10 // dValue,  
}  
return;
```

PMGetDefaultParameter

Ritorna il valore di un parametro di default del pattern matching

```
GVSBASEAPI int PMGetDefaultParameter (  
    int iNumParameter,  
    double * dValue,  
    char * sDescription  
    )
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iNumParameter Numero del parametro
double * dValue Ritorno il valore del parametro
char * sDescription Descrizione

Esempio:

```
double     dValue;  
char   sDescription[128];  
  
if (!PMGetDefaultParameter (  
    ))  
{  
    1, // iNumParameter,  
    &dValue,  
    &sDescription  
}  
return;
```

PMGetTemplateImage

Restituisce l'immagine del Template

```
GVSBASEAPI int PMGetTemplateImage (  
                                int * iWidth,  
                                int * iHeight  
                                )
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int * iWidth Larghezza dell'immagine
int * iHeight Altezza dell'immagine

Esempio:

```
int * iWidth;  
int * iHeight;  
  
if (!PMGetTemplateImage (  
    ))  
{  
    &iWidth,  
    &iHeight  
}  
return;
```

PMDraw

Disegna il pattern riconosciuto secondo l'elemento richiesto

```
GVSBASEAPI int PMDraw (  
    int iNumElement,  
    unsigned int uRGBEdgeColor,  
    HDC hdc,  
    int iWindowsID  
)
```

Valore di ritorno

1 OK
0 errore nella chiamata

Parametri

int iNumberElement	Numero dell'elemento
unsigned int uRGBEdgeColor	Colori RGB del disegno
HDC hdc	Device Context della Window
int iWindowsID	Identificativo della finestra 0-15(Default 0)

Esempio:

```
if (!PMDraw (  
    ))  
{  
    1,           //iNumElement,  
    RGB(128,0,0), // uRGBEdgeColor,  
    m_cdc.m_hDC // hdc,  
    0           // iWindowsID  
}  
return;
```

Visual Basic

Dichiarazioni d'interfacciamento con Visual Basic

- **DECLARE**
- **Esempio**

Dichiarazioni d'interfacciamento alle funzioni
Esempio di programmazione in Visual Basic

Declare

Dichiarazioni d'interfacciamento alle funzioni

Option Explicit

Public IRet As Long

Public iImageMaxXCamera2, iImageMaxYCamera2 As Long

Public WindowXCamera2, WindowYCamera2,
ixOffsetCamera2,
iyOffsetCamera2,
ixOffvCamera2,
iyOffvCamera2 As Long

Public iNumeroTelecamere As Integer

Public pAreaImmagineCamera2 As Long

Public iHandleCamera2 As Long

#Const PXC = False

Declare Function **OpenCGVSBASE** Lib "GVSBBase.dll" (
ByVal iImageMaxX As Long,
ByVal iImageMaxY As Long,
ByRef pArea As Long,
ByVal iCamera As Long,
ByRef iHandle As Long) As Long

Declare Function **CloseCGVSBASE** Lib "GVSBBase.dll" (
ByVal iHandle As Long) As Long

Declare Function **LoadBmpImage** Lib "GVSBBase.dll" (
ByVal sNomeFile As String,
ByVal iHandle As Long) As Long

Declare Function **PutImageOnWindow** Lib "GVSBBase.dll" (
ByVal iHDC As Long,
ByVal iMode As Long,
ByVal iWindowID As Long,
ByVal iHandle As Long) As Long

Declare Function **SetWindowParameters** Lib "GVSBBase.dll" (
ByVal iHWND As Long,
ByVal ixOffset As Long,
ByVal iyOffset As Long,
ByVal iLinePenColor As Long,
ByVal iCirclePenColor As Long,
ByVal iArcPenColor As Long,
ByVal iWindowsID As Long,
ByVal iCoordinatePenColor As Long,
ByVal iBorderPenColor As Long) As Long

Declare Function **GetAllBlobs** Lib "GVSBBase.dll" (
ByVal iLeft As Long,
ByVal iTop As Long,
ByVal iWidth As Long,
ByVal iHeight As Long, _
ByVal iMode As Long,
ByVal iModeLowParam As Long,
ByVal iModeHighParam As Long, _
ByVal iTypeBlob As Long,
ByVal iLowArea As Long,
ByVal iTypeFilter As Long,
ByVal iFilterParam As Long, _
ByVal iFlagDistanzaCentro As Long,
ByVal iPerc As Long,
ByVal iHandle As Long) As Long

Declare Function **DrawBlob** Lib "GVSBBase.dll" (
ByVal iNumberBlob As Long,
ByVal iFlagBlobPicture As Long,
ByVal uRGBPictureColor As Long,
ByVal iFlagBlobEdge As Long, _
ByVal uRGBEdgeColor As Long,
ByVal iFlagValue As Long,
ByVal iFlagDimension As Long, _
ByVal iFlagCircle As Long,
ByVal iFlagXAxesRect As Long,
ByVal iFlagYAxesRect As Long,
ByVal hdc As Long, _
ByVal iWindowsID As Long,
ByVal iFlagBorderROI As Long,
ByVal iFlagInternalBorderROI As Long, _
ByVal iFlagCoordinateROI As Long,
ByVal iFlagEllipse As Long) As Long

Declare Function **RemoveAllBlobs** Lib "GVSBASE.dll" () As Long

Declare Function **GetValueBlob** Lib "GVSBASE.dll" (
 ByVal iNumberBlob As Long, _
 ByRef pdXGeometric As Double, _
 ByRef pdYGeometric As Double, _
 ByRef pdAreaas As Double, _
 ByRef pdLengthBorder As Double, _
 ByRef pdXBaricenter As Double, _
 ByRef pdYBaricenter As Double, _
 ByRef pdAreaGravity As Double _
) As Long

Declare Function **GetEllipseBlob** Lib "GVSBASE.dll" (
 ByVal iNumberBlob As Long, _
 ByRef pdXc As Double, _
 ByRef pdYc As Double, _
 ByRef pdAngA As Double, _
 ByRef pdRgA As Double, _
 ByRef pdAngB As Double, _
 ByRef pdRgB As Double, _
 ByRef pdX1 As Double, _
 ByRef pdY1 As Double, _
 ByRef pdX2 As Double, _
 ByRef pdY2 As Double, _
 ByRef pdX3 As Double, _
 ByRef pdY3 As Double, _
 ByRef pdX4 As Double, _
 ByRef pdY4 As Double, _
 ByVal iCorona As Integer, _
 ByVal dPixelGap As Double _
) As Long


```

Declare Function GetDimensionBlob Lib "GVSBBase.dll" (
    ByVal iNumberBlob As Long, _
    ByRef pdXHigh As Double, _
    ByRef pdYHigh As Double, _
    ByRef pdXLow As Double, _
    ByRef pdYLow As Double, _
    ByRef pdXRight As Double, _
    ByRef pdYRight As Double, _
    ByRef pdXLeft As Double, _
    ByRef pdYLeft As Double, _
    ByRef pdXHighAlongBaricenter As Double, _
    ByRef pdYHighAlongBaricenter As Double, _
    ByRef pdXLowAlongBaricenter As Double, _
    ByRef pdYLowAlongBaricenter As Double, _
    ByRef pdXRightAlongBaricenter As Double, _
    ByRef pdYRightAlongBaricenter As Double, _
    ByRef pdXLeftAlongBaricenter As Double, _
    ByRef pdYLeftAlongBaricenter As Double _
) As Long

```

#If PXC Then

```

Declare Function OpenFrameGrabber Lib "VisAlgoPXC200.dll" (
    ByVal iNumberCamera As Integer,
    ByVal iDouble As Integer,
    ByVal iProgressive As Integer,
    ByRef iImageMaxX As Long,
    ByRef iImageMaxY As Long) As Integer

```

```

Declare Sub CloseFrameGrabber Lib "VisAlgoPXC200.dll" ()

```

```

Declare Sub GetImage Lib "VisAlgoPXC200.dll" (
    ByVal pArea As Long,
    ByVal iCamera As Integer)

```

```

Declare Sub GetContinuosImage Lib "VisAlgoPXC200.dll" (
    ByVal pArea As Long,
    ByVal iCamera As Integer)

```

```

Declare Sub GetGlobals Lib "VisAlgoPXC200.dll" (
    ByVal iHWND As Long,
    ByVal iHDC As Long,
    ByRef iWindowX As Long,
    ByRef iWindowY As Long,
    ByRef ixOffset As Long,
    ByRef iyOffset As Long,
    ByRef ixOffv As Long,
    ByRef iyOffv As Long)

Declare Sub SetPhysicalCamera Lib "VisAlgoPXC200.dll" (
    ByVal iCamera As Integer)
#Else

Declare Function OpenFrameGrabber Lib "VisAlgoPX610.dll" (
    ByVal iNumberCamera As Integer,
    ByVal iDouble As Integer,
    ByVal iProgressive As Integer,
    ByRef iImageMaxX As Long,
    ByRef iImageMaxY As Long) As Integer

Declare Sub CloseFrameGrabber Lib "VisAlgoPX610.dll" ()

Declare Sub GetImage Lib "VisAlgoPX610.dll" (
    ByVal pArea As Long,
    ByVal iCamera As Integer)

Declare Sub GetContinuosImage Lib "VisAlgoPX610.dll" (
    ByVal pArea As Long,
    ByVal iCamera As Integer)

Declare Sub GetGlobals Lib "VisAlgoPX610.dll" (
    ByVal iHWND As Long,
    ByVal iHDC As Long,
    ByRef iWindowX As Long,
    ByRef iWindowY As Long,
    ByRef ixOffset As Long,
    ByRef iyOffset As Long,
    ByRef ixOffv As Long,
    ByRef iyOffv As Long)

Declare Sub SetPhysicalCamera Lib "VisAlgoPX610.dll" (
    ByVal iCamera As Integer)
#End If

```

Esempio

Esempio di programmazione in Visual Basic

```
Private Sub Form_Load()  
  
    Dim IRet As Long  
    Dim iImageMaxX, iImageMaxY As Long  
    Dim WindowX, WindowY, ixOffset, iyOffset, ixOffv, iyOffv As Long  
    Dim iNumeroTelecamere As Integer  
    '  
    '*****  
    '*      Image Set Up      *  
    '*****  
    IRet = OpenFrameGrabber(1, 0, 0, iImageMaxXCamera2, iImageMaxYCamera2)  
    If IRet = 0 Then  
        ' Errore nella apertura della piastra di visione  
        iNumeroTelecamere = 0  
    End If  
  
    GetGlobals          PicCamera2.hWnd,  
                       PicCamera2.hdc,  
                       WindowXCamera2,  
                       WindowYCamera2,  
                       ixOffsetCamera2,  
                       iyOffsetCamera2,  
                       ixOffvCamera2,  
                       iyOffvCamera2  
  
    IRet = OpenCGVSBASE(  
        iImageMaxXCamera2,  
        iImageMaxYCamera2,  
        pAreaImmagineCamera2,  
        0,  
        iHandleCamera2)  
  
    IRet = SetWindowParameters(  
        PicCamera2.hWnd,  
        0,  
        0,  
        RGB(255, 0, 0),  
        RGB(0, 255, 0),  
        RGB(0, 0, 255),  
        0,  
        RGB(255, 255, 0),  
        RGB(0, 255, 255))  
    If iNumeroTelecamere < 1 Then  
        IRet = LoadBmpImage("c:\ProvaVB\Foto2.bmp", iHandleCamera2)  
    End If  
  
Private Sub Form_Unload(Cancel As Integer)
```

```

Dim IRet As Long

CloseFrameGrabber
IRet = RemoveAllBlobs()
IRet = CloseCGVSBASE(iHandleCamera2)
End Sub

'*****
'*   Show Live Image   *
'*****
,

Private Sub TmrCamera2_Timer()
    Dim IRet As Long

    GetImage pAreaImmagineCamera2, 1
    IRet = PutImageOnWindow(PicCamera2.hdc, 0, 0, iHandleCamera2)
End Sub

Private Sub Command1_Click()

'Routine che acquisisce ed elabora l'immagine
Dim Messaggio As String
Dim IRet As Long
Dim X, Y, Area, Perimetro, BarX, BarY, AreaGravity As Double
Dim XC, YC, AngA, RaggioA, AngB, RaggioB, X1, Y1, X2, Y2, X3, Y3, X4, Y4
    As Double
Dim dXHigh, dYHigh, dXLow, dYLow, dXRight, dYRight, dXLeft, dYLeft As Double

Dim dXHighAlongBaricenter, dYHighAlongBaricenter, dXLowAlongBaricenter,
    dYLowAlongBaricenter, dXRightAlongBaricenter, dYRightAlongBaricenter,
    dXLeftAlongBaricenter, dYLeftAlongBaricenter
    As Double

IRet = GetAllBlobs(-768 / 2 + 80, 576 / 2 - 80, 768 - 160, 576 - 160, 77, 1, 2, 0, 200, 0, 0,
0, 0, iHandleCamera2)

If (IRet > 0) Then
    TmrCamera2.Enabled = False
    IRet = GetValueBlob(1, X, Y, Area, Perimetro, BarX, BarY, AreaGravity)
    IRet = GetEllipseBlob(1, XC, YC, AngA, RaggioA, AngB, RaggioB,
        X1, Y1, X2, Y2, X3, Y3, X4, Y4, 1, 5)

    IRet = GetDimensionBlob(1, dXHigh, dYHigh, dXLow,
        dYLow, dXRight, dYRight, dXLeft, dYLeft,
        dXHighAlongBaricenter, dYHighAlongBaricenter,
        dXLowAlongBaricenter, dYLowAlongBaricenter,
        dXRightAlongBaricenter, dYRightAlongBaricenter,
        dXLeftAlongBaricenter, dYLeftAlongBaricenter)

    IRet = DrawBlob(1, 1, RGB(0, 128, 0), 1, RGB(255, 0, 0),
        1, 1, 0, 0, 0, PicCamera2.hdc, 0, 1, 1, 1, 1)

    PicCamera2.Line (PicCamera2.Width / 2, 0)-

```

```

                (PicCamera2.Width / 2, PicCamera2.Height),
                RGB(0, 0, 255)
PicCamera2.Line (0, PicCamera2.Height / 2)-
                (PicCamera2.Width, PicCamera2.Height / 2),
                RGB(0, 0, 255)

If AngA > 180 Then
    AngA = AngA - 180
End If
Messaggio = "Angolo = "
Messaggio = Messaggio + Format(AngA, "0.000")
Messaggio = Messaggio + Chr(13)
Messaggio = Messaggio + Chr(10)
Messaggio = Messaggio + "Area = "
Messaggio = Messaggio + Str(Area)
Messaggio = Messaggio + Chr(13)
Messaggio = Messaggio + Chr(10)
Messaggio = Messaggio + "Perimetro = "
Messaggio = Messaggio + Str(Perimetro)
Messaggio = Messaggio + Chr(13)
Messaggio = Messaggio + Chr(10)
Messaggio = Messaggio + "Baricentro = "
Messaggio = Messaggio + Format(X, "0.000")
Messaggio = Messaggio + " , "
Messaggio = Messaggio + Format(Y, "0.000")
Messaggio = Messaggio + Chr(13)
Messaggio = Messaggio + Chr(10)
Messaggio = Messaggio + "Punto Alto = "
Messaggio = Messaggio + Str(dXHigh)
Messaggio = Messaggio + " , "
Messaggio = Messaggio + Str(dYHigh)
Messaggio = Messaggio + Chr(13)
Messaggio = Messaggio + Chr(10)
Messaggio = Messaggio + "Punto Basso = "
Messaggio = Messaggio + Str(dXLow)
Messaggio = Messaggio + " , "
Messaggio = Messaggio + Str(dYLow)
Messaggio = Messaggio + Chr(13)
Messaggio = Messaggio + Chr(10)
Messaggio = Messaggio + "Punto Destra = "
Messaggio = Messaggio + Str(dXRight)
Messaggio = Messaggio + " , "
Messaggio = Messaggio + Str(dYRight)
Messaggio = Messaggio + Chr(13)
Messaggio = Messaggio + Chr(10)
Messaggio = Messaggio + "Punto Sinistra = "
Messaggio = Messaggio + Str(dXLeft)
Messaggio = Messaggio + " , "
Messaggio = Messaggio + Str(dYLeft)
MsgBox Messaggio
TmrCamera2.Enabled = True
End If
End Sub

```