

OpenCV ile Bilgisayarla Görme



Open Source

Computer
Vision
Library

intel



ubuntu

OpenCV ile Bilgisayarlı Görme



İsmet YALABIK

791. Linux Kullanıcıları Derneği Üyesi

Araştırma Görevlisi
Görüntü İşleme ve Örüntü Tanıma Laboratuvarı
Bilgisayar Mühendisliği Bölümü
Orta Doğu Teknik Üniversitesi

e-mail: ismet@ceng.metu.edu.tr

web: <http://www.ceng.metu.edu.tr/~ismet>

Bilgisayarlı Görme Nedir?



- Göz-Beyin İlişkisi
- Görme, Tanıma, Algılama
- Görüntünün Algılanması
- Nesnelerin Ayırıştırılması
- Nesnelerin Tanınması
- Nesnelerin Eşleştirilmesi



- İlgili alanlardaki araştırma ve geliştirmeyi desteklemek için evrensel bir araç kutusu oluşturmak
 - Bilgisayarla Görme
 - İnsan-Bilgisayar Etkileşimi
 - Robot Bilimi
 - Biyoenformatik
 - Güvenlik
 - Görüntüleme

OpenCV'nin Anahtar Özellikleri



- Cross-platform
- Orta ile ileri seviyeli API
 - 350'den fazla C fonksiyonu
 - Birkaç C++ sınıfı
- Python eklentileri
- Çok az kütüphane bağımlılığı
- IPP(Intel Performance Primitives) desteği

OpenCV Tarihçesi



- Başlangıç: 2000, intel
- sourceforge.net'e geçiş
- CVPR-2001
- Stanley aracının anahtar kütüphanesi (2M \$lık darpa yarışı birincisi)



6

22/04/11



- Temel Yapılar
 - Nokta
 - Dikdörtgen
 - Ölçüt
 - Matris
 - Görüntü
 - Ayrık Matris

OpenCV: Temel Yapılar



```
typedef struct CvPoint {
    int x; /* x-coordinate, usually zero-based */
    int y; /* y-coordinate, usually zero-based */
} CvPoint;
```

```
typedef struct CvRect {
    int x; /* x-coordinate of the left-most
            rectangle corner[s] */
    int y; /* y-coordinate of the top-most or
            bottom-most rectangle corner[s] */
    int width; /* width of the rectangle */
    int height; /* height of the rectangle */
} CvRect;
```

OpenCV: Temel Yapılar



```
typedef struct _IplImage {
    int    nSize;           /* sizeof(IplImage) */
    int    ID;             /* version (=0)*/
    int    nChannels;      /* Most of OpenCV functions support 1,2,3 or 4 channels */
    int    alphaChannel;   /* ignored by OpenCV */
    int    depth;          /* pixel depth in bits: IPL_DEPTH_8U, IPL_DEPTH_8S, IPL_DEPTH_16U,
                           IPL_DEPTH_16S, IPL_DEPTH_32S, IPL_DEPTH_32F and IPL_DEPTH_64F are
                           supported */

    char   colorModel[4];  /* ignored by OpenCV */
    char   channelSeq[4];  /* ditto */
    int    dataOrder;     /* 0 - interleaved color channels, 1 - separate color channels.
                           cvCreateImage can only create interleaved images */

    int    origin;        /* 0 - top-left origin,
                           1 - bottom-left origin (Windows bitmaps style) */

    int    align;         /* Alignment of image rows (4 or 8).
                           OpenCV ignores it and uses widthStep instead */

    int    width;         /* image width in pixels */
    int    height;        /* image height in pixels */
    struct _IplROI *roi;  /* image ROI. when it is not NULL, this specifies image region to
                           process */

    struct _IplImage *maskROI; /* must be NULL in OpenCV */
    void   *imageId;      /* ditto */
    struct _IplTileInfo *tileInfo; /* ditto */
    int    imageSize;     /* image data size in bytes (=image->height*image->widthStep in case
                           of interleaved data)*/

    char   *imageData;    /* pointer to aligned image data */
    int    widthStep;     /* size of aligned image row in bytes */
    int    BorderMode[4];  /* border completion mode, ignored by OpenCV */
    int    BorderConst[4]; /* ditto */
    char   *imageDataOrigin; /* pointer to a very origin of image data (not necessarily aligned) -
                              it is needed for correct image deallocation */
} IplImage;
```

OpenCV: Temel Operasyonlar



- Aritmetik, Mantık ve Karşılaştırmalar
 - Matris Aritmetiği, Mantık İşlemeleri
- İstatistik
- Permütasyonlar ve Dönüşümler
 - Ayırma, Çevirme, Tekrarlama
- Lineer Cebir
 - Normalizasyon, Çapraz Çarpım, Ters
- Rastgele Sayı Üreticisi



- Hafıza depoları
- Sekanslar
- Kümeler
- Çizgeler
- Ağaçlar

OpenCV: Temel Veri Yapıları



```
#define CV_SEQUENCE_FIELDS() \  
    int flags;                /* miscellaneous flags */ \  
    int header_size;         /* size of sequence header */ \  
    struct CvSeq* h_prev;    /* previous sequence */ \  
    struct CvSeq* h_next;    /* next sequence */ \  
    struct CvSeq* v_prev;    /* 2nd previous sequence */ \  
    struct CvSeq* v_next;    /* 2nd next sequence */ \  
    int total;               /* total number of elements */ \  
    int elem_size;          /* size of sequence element in bytes */ \  
    char* block_max;        /* maximal bound of the last block */ \  
    char* ptr;              /* current write pointer */ \  
    int delta_elems;        /* how many elements allocated when the sequence grows  
                             (sequence granularity) */ \  
    CvMemStorage* storage;  /* where the seq is stored */ \  
    CvSeqBlock* free_blocks; /* free blocks list */ \  
    CvSeqBlock* first;     /* pointer to the first sequence block */ \  
  
typedef struct CvSeq \  
{  
    CV_SEQUENCE_FIELDS()  
} CvSeq;
```

OpenCV: Kalan Ana Fonkyonlar



- Eğri ve Şekil Çizme Fonksiyonları
- Metin Yazma Fonksiyonları
- Dosya Depoları, okuma/yazma
- Hata kontrolü
- Hafıza Yönetimi ve sistem çağrıları



Şekil, Çizgi, Metin Uygulaması

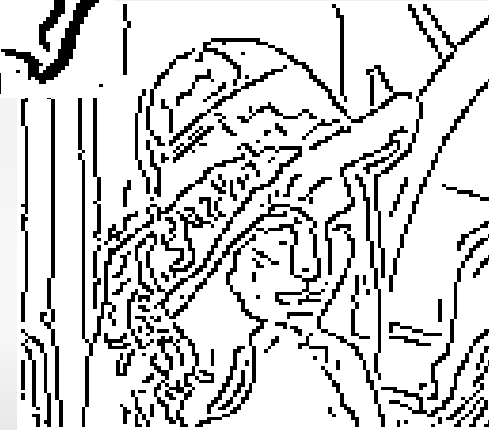
uygulama-1
(edge)



- Kenar, Köşe İşlemleri
- İnterpolasyon, Geometrik Transformasyon
- Morfolojik İşlemler
- Filtreler ve Renk Dönüşümleri
- Görüntü Bölütleme
- Özel Görüntü Dönüşümleri
- Histogram
- Eşleştirme



- Sobel Filtresi
- Laplace Filtresi
- Canny Kenar Yör
- Harris Köşe Yöntemi
- Minimum Eigen Değeri Yöntemi





Canny Kenar Bulma Uygulaması

uygulama-2
(edge)

OpenCV: Interpolasyon ve Geometrik Dönüşümler



- Büyütme, küçültme
- Affine Dönüşümler
- Rotasyon
- Logpolar





- Erode
- Dilate
- Yapısal Elemanın Oluşturulması
- Gelişmiş Morfolojik Dönüşümler



Dilation (black)



Dilation (white)



Erosion (black)



Erosion (white)



Open (black)



Open (white)



Close (black)



Close (white)



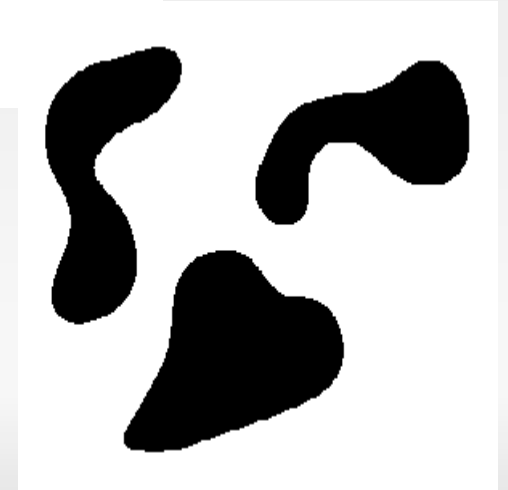
Morfoloji Uygulaması

uygulama-3
(morphology)

OpenCV: Filtreler ve Renk Dönüşümleri



- Düzleştirme
- Konvolüsyon Filtre
- İntegral Görüntü
- Eşik Uygulama
- Renk Uzayı Dönüşümleri

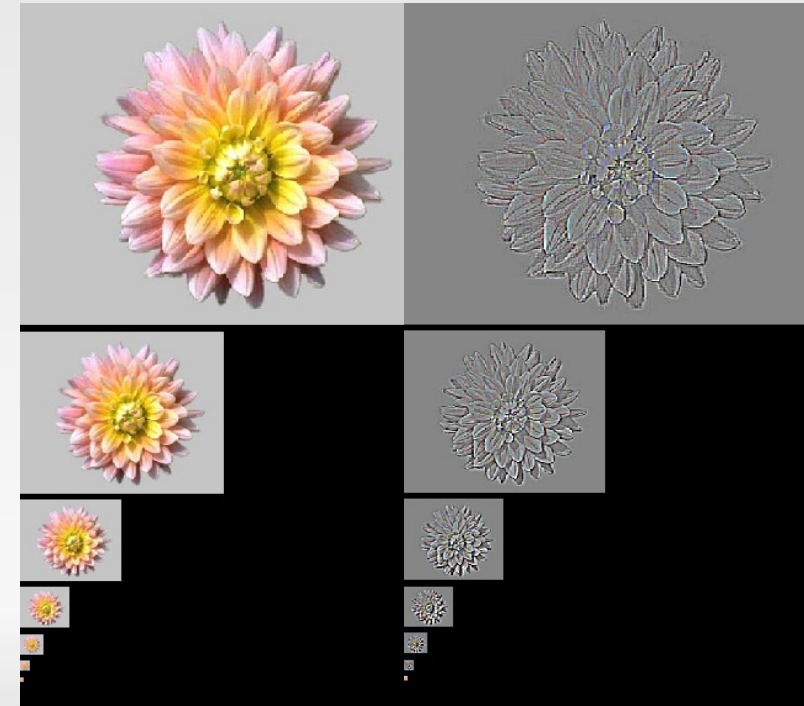




Piramit Bölütleme Uygulaması



uygulama-4 (pyramid_segmentation)





- FloodFill, Birleşik Bileşenleri Bulma



Original image



Tolerance interval ± 5



Tolerance interval ± 6



Birleşik Bileşenler Uygulaması

uygulama-5
(contours)



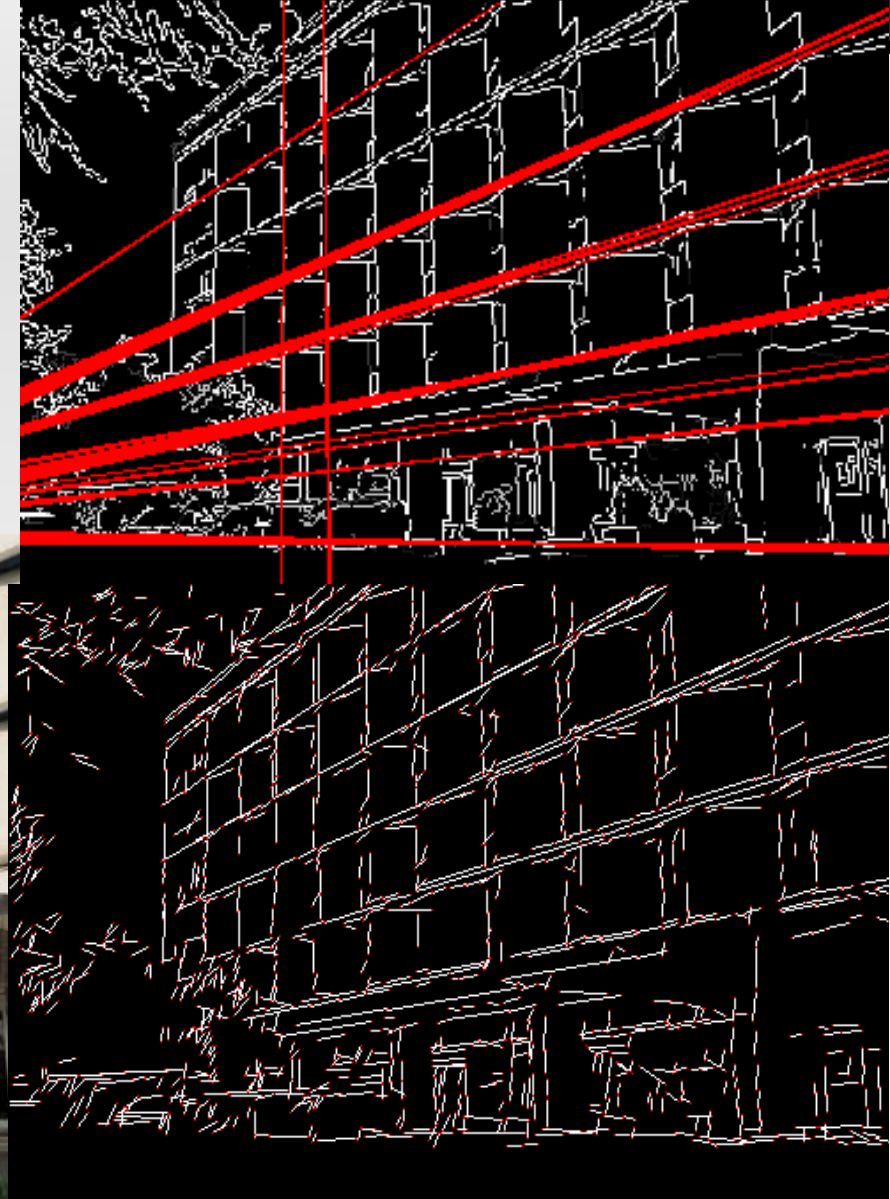
Watershed Uygulaması

uygulama-6
(watershed)

OpenCV: Özel Görüntü Dönüşümleri



- Hough Çizgi Dönüşümleri
- Mesafe Dönüşümleri



22/04/11



Hough Çizgi Dönüşümü Uygulaması

uygulama-7
(houghlines)



Laplace Dönüşümü Uygulaması

uygulama-8
(laplace)



Histogram Uygulaması

uygulama-9
(demhist)

OpenCV: Histogram



```
cvLUT( src_image, dst_image, lut_mat );
cvShowImage( "image", dst_image );

cvCalcHist( &dst_image, hist, 0, NULL );
cvZero( dst_image );
cvGetMinMaxHistValue( hist, 0, &max_value, 0, 0 );
cvScale( hist->bins, hist->bins, ((double)hist_image-
>height)/max_value, 0 );
/*cvNormalizeHist( hist, 1000 );*/

cvSet( hist_image, cvScalarAll(255), 0 );
bin_w = cvRound((double)hist_image->width/hist_size);

for( i = 0; i < hist_size; i++ )
    cvRectangle( hist_image, cvPoint(i*bin_w, hist_image->height),
                cvPoint((i+1)*bin_w, hist_image->height -
cvRound(cvGetReal1D(hist->bins,i))),
                cvScalarAll(0), -1, 8, 0 );

cvShowImage( "histogram", hist_image );
}
int main( int argc, char** argv )
{
    // Load the source image. HighGUI use.
    src_image = cvLoadImage( argc == 2 ? argv[1] : file_name, 0 );

    if( !src_image )
    {
        printf("Image was not loaded.\n");
        return -1;
    }
}
```

```
dst_image = cvCloneImage(src_image);
hist_image = cvCreateImage(cvSize(320,200), 8, 1);
hist = cvCreateHist(1, &hist_size, CV_HIST_ARRAY, ranges, 1);
lut_mat = cvCreateMatHeader( 1, 256, CV_8UC1 );
cvSetData( lut_mat, lut, 0 );

cvNamedWindow("image", 0);
cvNamedWindow("histogram", 0);

cvCreateTrackbar("brightness", "image", &_brightness, 200,
update_brightcont);
cvCreateTrackbar("contrast", "image", &_contrast, 200,
update_brightcont);

update_brightcont(0);
cvWaitKey(0);

cvReleaseImage(&src_image);
cvReleaseImage(&dst_image);

cvReleaseHist(&hist);

return 0;
}

#ifdef _EiC
main(1,"demhist.c");
#endif
```

OpenCV: Histogram



```
#ifdef _CH_
#pragma package <opencv>
#endif

#ifndef _EiC
#include "cv.h"
#include "highgui.h"
#include <stdio.h>
#endif

char file_name[] = "baboon.jpg";

int _brightness = 100;
int _contrast = 100;

int hist_size = 64;
float range_0[] = {0,256};
float* ranges[] = { range_0 };
IplImage *src_image = 0, *dst_image = 0, *hist_image = 0;
CvHistogram *hist;
uchar lut[256];
CvMat* lut_mat;

/* brightness/contrast callback function */
void update_brightcont( int arg )
{
    int brightness = _brightness - 100;
    int contrast = _contrast - 100;
    int i, bin_w;
    float max_value = 0;
```

```
if( contrast > 0 )
{
    double delta = 127.*contrast/100;
    double a = 255./(255. - delta*2);
    double b = a*(brightness - delta);
    for( i = 0; i < 256; i++ )
    {
        int v = cvRound(a*i + b);
        if( v < 0 )
            v = 0;
        if( v > 255 )
            v = 255;
        lut[i] = (uchar)v;
    }
}
else
{
    double delta = -128.*contrast/100;
    double a = (256.-delta*2)/255.;
    double b = a*brightness + delta;
    for( i = 0; i < 256; i++ )
    {
        int v = cvRound(a*i + b);
        if( v < 0 )
            v = 0;
        if( v > 255 )
            v = 255;
        lut[i] = (uchar)v;
    }
}
}
```



- Kontur İşleme
 - Kontur alanı, çizgi uzunluğu, kontur ağaçları
- İşlemsel Geometri
 - İki kareyi kapsayan en küçük kare
 - Elips oturtma, çizgi oturtma
 - convexHull
- Düzlemsel Alt Bölümlenmeler
 - Voronin Diyagramları



Convexhull Uygulaması

uygulama-10
(convexhull)

OpenCV: Hareket Analizi, Nesne İzleme



- Arka Plan İstatistiklerinin Toplanması
- Hareket Şablonları
- Nesne İzleme
- Optik Akış
- Tahmin Ediciler



Nesne İzleme Uygulaması

uygulama-11
(camshiftdemo)



- Nesne Tanıma
- Nesne Tanımaya Yönelik Basit Haar Özniteliklerinin Çıkarılması

Yüz Bulma Uygulaması

OpenCV: Yüz Bulma



```
#include "cv.h"
#include "highgui.h"

CvHaarClassifierCascade* load_object_detector( const char* cascade_path )
{
    return (CvHaarClassifierCascade*)cvLoad( cascade_path );
}

void detect_and_draw_objects( IplImage* image,
                             CvHaarClassifierCascade* cascade,
                             int do_pyramids )
{
    IplImage* small_image = image;
    CvMemStorage* storage = cvCreateMemStorage(0);
    CvSeq* faces;
    int i, scale = 1;

    /* if the flag is specified, down-scale the input image to get a
       performance boost w/o losing quality (perhaps) */
    if( do_pyramids )
    {
        small_image = cvCreateImage( cvSize(image->width/2,image->height/2), IPL_DEPTH_8U, 3 );
        cvPyrDown( image, small_image, CV_GAUSSIAN_5x5 );
        scale = 2;
    }

    /* use the fastest variant */
    faces = cvHaarDetectObjects( small_image, cascade, storage, 1.2, 2, CV_HAAR_DO_CANNY_PRUNING );

    /* draw all the rectangles */
}
```

OpenCV: Yüz Bulma



```
for( i = 0; i < faces->total; i++ )
{
    /* extract the rectangles only */
    CvRect face_rect = *(CvRect*)cvGetSeqElem( faces, i, 0 );
    cvRectangle( image, cvPoint(face_rect.x*scale,face_rect.y*scale),
                cvPoint((face_rect.x+face_rect.width)*scale,
                        (face_rect.y+face_rect.height)*scale),
                CV_RGB(255,0,0), 3 );
}

if( small_image != image )
    cvReleaseImage( &small_image );
    cvReleaseMemStorage( &storage );
}

/* takes image filename and cascade path from the command line */
int main( int argc, char** argv )
{
    IplImage* image;
    if( argc==3 && (image = cvLoadImage( argv[1], 1 )) != 0 )
    {
        CvHaarClassifierCascade* cascade = load_object_detector(argv[2]);
        detect_and_draw_objects( image, cascade, 1 );
        cvNamedWindow( "test", 0 );
        cvShowImage( "test", image );
        cvWaitKey(0);
        cvReleaseHaarClassifierCascade( &cascade );
        cvReleaseImage( &image );
    }

    return 0;
}

22/04/11
```

OpenCV: Diğer Özellikler



- Kamera Kalibrasyonu
- Poz Tahmini

Niye OpenCV?



- Birçok kullanılabilir ve literatürde başarılı çalışmalara yer veriyor
- Özellikle işlemci bazında optimizasyon ile programların çok hızlı çalışması sağlanıyor
- Çok basit bir kullanım olanağı sağlıyor
- Gelişmiş arayüz fonkyonları sayesinde kullanıcı etkileşimini yükseltiyor
- Kamera kullanımını kolay kılan özellikleri

Niye Özgür Yazılım?



Paylaşarak hayatımızı
güzelleştirmek için...



OpenCV'i Nereden Bulabilirim?

<http://tech.groups.yahoo.com/group/OpenCV/>

<http://www.intel.com/technology/computing/opencv/index.htm>

<http://opencvlibrary.sourceforge.net/>



Teşekkürler,
Sorularınız???

