OpenCV

The Image Processing Library

Features

* Image data manipulation (allocation, release, copying, setting, conversion).

* Image and video I/O (file and camera based input, image/video file output).

* Matrix and vector manipulation and linear algebra routines (products, solvers, eigenvalues, SVD).

* Various dynamic data structures (lists, queues, sets, trees, graphs).

* Basic image processing (filtering, edge detection, corner detection, sampling and interpolation, color conversion, morphological operations, histograms, image pyramids).

* Structural analysis (connected components, contour processing, distance transform, various moments, template matching, Hough transform, polygonal approximation, line fitting, ellipse fitting, Delaunay triangulation).

* Camera calibration (finding and tracking calibration patterns, calibration, fundamental matrix estimation, homography estimation, stereo correspondence).

Features (contd.)

* Motion analysis (optical flow, motion segmentation, tracking).

* Object recognition (eigen-methods, HMM).

* Basic GUI (display image/video, keyboard and mouse handling, scroll-bars).

* Image labeling (line, conic, polygon, text drawing)

Data structures

- IplImage – Basic Image data structure
- CvMat – General matrix data structure, interconvertible with IplImage
- CvArr – Generic class which is parent of CvMat and IplImage
- CvScalar – Used to denote scalar values
- CvSize – Width x Height

IplImage

```
IplImage
   |-- int  nChannels;     // Number of color channels (1,2,3,4)
   |-- int  depth;         // Pixel depth in bits:
   |                       //   IPL_DEPTH_8U, IPL_DEPTH_8S,
   |                       //   IPL_DEPTH_16U,IPL_DEPTH_16S,
   |                       //   IPL_DEPTH_32S,IPL_DEPTH_32F,
   |                       //   IPL_DEPTH_64F
   |-- int  width;         // image width in pixels
   |-- int  height;        // image height in pixels
   |-- char* imageData;    // pointer to aligned image data
   |                       // Note that color images are stored in BGR order
   |-- 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  widthStep;     // size of aligned image row in bytes
   |-- int  imageSize;     // image data size in bytes = height*widthStep
   |-- struct _IplROI *roi;// image ROI. when not NULL specifies image
   |                       // region  to be processed.
   |-- char *imageDataOrigin; // pointer to the unaligned origin of image data
   |                          // (needed for correct image deallocation)
   |
   |-- int  align;         // Alignment of image rows: 4 or 8 byte alignment
   |                       // OpenCV ignores this and uses widthStep instead
   |-- char colorModel[4]; // Color model - ignored by OpenCV
```

CvMat

```
CvMat                          // 2D array
  |-- int   type;               // elements type
(uchar,short,int,float,double) and flags
  |-- int   step;               // full row length in bytes
  |-- int   rows, cols;         // dimensions
  |-- int   height, width;      // alternative dimensions reference
  |-- union data;
      |-- uchar*  ptr;         // data pointer for an unsigned char matrix
      |-- short*  s;           // data pointer for a short matrix
      |-- int*    i;           // data pointer for an integer matrix
      |-- float*  fl;          // data pointer for a float matrix
      |-- double* db;          // data pointer for a double matrix
```

Create and delete

- Create an Image or Matrix
  - cvCreateImage, cvCreateMat
- Delete an Image or Matrix
  - cvReleaseImage, cvReleaseMat
- Manual memory management required

Loading and Saving

- Load an Image
  - cvLoadImage
- Save Image
  - cvSaveImage
- Display Image
  - Create a window: cvNamedWindow
  - Show in the window: cvShowImage

Example

```
// Include the header files
#include <highgui.h>
#include <cv.h>

main(){
    // Load the image
    IplImage *i = cvLoadImage("in.jpg");
    // Declare the subrectangle
    CvMat subrect;
    // Get the required subrectangle
    cvGetSubRect(i,&subrect,cvRect(10,10,50,50));
    // Create temporary images to hold the 3 channels
    CvMat *r = cvCreateMat(subrect.rows,subrect.cols,CV_8UC1);
    CvMat *g = cvCreateMat(subrect.rows,subrect.cols,CV_8UC1);
    CvMat *b = cvCreateMat(subrect.rows,subrect.cols,CV_8UC1);
    // Split the sub-rectangle into the 3 channels
    cvSplit(&subrect,b,g,r,NULL);
    // Set the green channel to 255
    cvSet(g,cvScalar(255));
    // Set the blue channel to 0
    cvZero(b);
    // Merge the split channels
    cvMerge(b,g,r,NULL,&subrect);
    // Show the resulting image
    cvNamedWindow("Show");
    cvShowImage("Show",i);
    cvWaitKey(0);
    return 0;
}
```