Designing Applications that See
Lecture 8: OpenCV

Dan Maynes-Aminzade

31 January 2008

Reminders

- Pick up Assignment #1 if you haven't already
- Assignment #2 due next Tuesday
- Fill out the interim course evaluation form

Today's Goals

- Explore the capabilities of the OpenCV library
- Learn the basics of programming with OpenCV
- Build a few working OpenCV examples
- Study other examples to learn more

Outline

- General introduction
- Basic video capture and display
- Image processing examples
- Optical flow example
- Video writing
- Look at more examples (time permitting)

What is OpenCV?

- Open source C/C++ computer vision library
- Created and maintained by Intel
- Optimized for real-time applications
- Composed of four separate sub-libraries:
  - CXCORE: Linear algebra, matrix operations
  - CV: Computer vision algorithms
  - HIGHGUI: Media, window, and UI handling
    - Capture, read, and write videos
    - Display video windows
    - Handle mouse and keyboard events
  - CVAUX: Experimental functionality (beta)

OpenCV Features

- Image data manipulation
- Image and video I/O
- Matrix and vector manipulation
- Dynamic data structures
- Image Processing
- Structural analysis
- Camera calibration
- Motion analysis
- Object recognition
- Basic GUI
- Basic drawing

Help on OpenCV

- Reference Manuals
  - `OpenCV\docs`
- Image Processing Samples
  - Edge detection: `edge`
  - Segmentation: `pyramid_segmentation`
  - Morphology: `morphology`
  - Histogram: `demhist`
  - Distance transform: `distrans`
  - Ellipse fitting: `fitellipse`
- Video Processing Samples
  - Color tracking: `camshiftdemo`
  - Optical flow: `lkdemo`
  - Motion segmentation: `motempl`
  - Edge detection: `laplace`

Installing OpenCV

- Download from SourceForge

  http://sourceforge.net/projects/opencvlibrary

- Current version: 1.0.0
- Windows version includes installer
- Installing in Linux (requires ffMPEG):

```
gunzip opencv-1.0.0.tar.gz
tar –xvf opencv-1.0.0.tar
cd opencv-1.0.0
./configure
make
make install
```

Compilation Instructions

- Download the sample code here:

  http://cs377s.stanford.edu/code/opencv-tutorial.zip

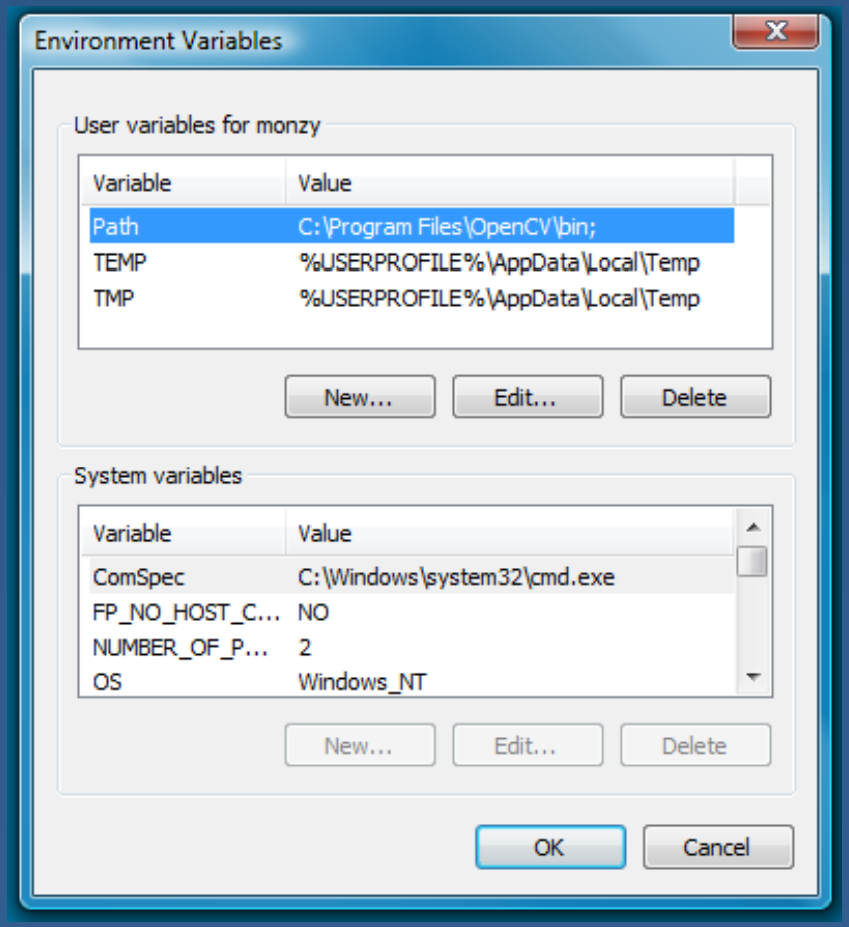
- Compiling in Linux:

```
g++ basic.cpp –o basic \
-I /usr/local/include/opencv -L /usr/local/lib \
-lm -lcv -lhighgui –lcvaux
```

- Compiling in Windows:
  - Start Visual Studio and follow along now!

Check Windows Path

- Should include `OpenCV\bin`



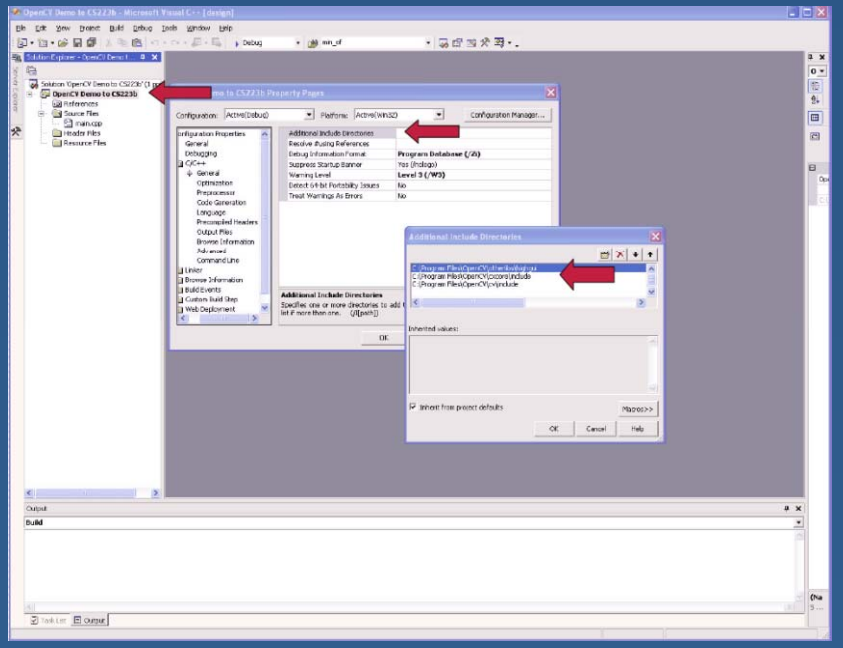
Update Visual Studio Directories

- Include directories

```
OpenCV\cv\include
OpenCV\cxcore\include
OpenCV\otherlibs\highgui
OpenCV\cvaux\include
```

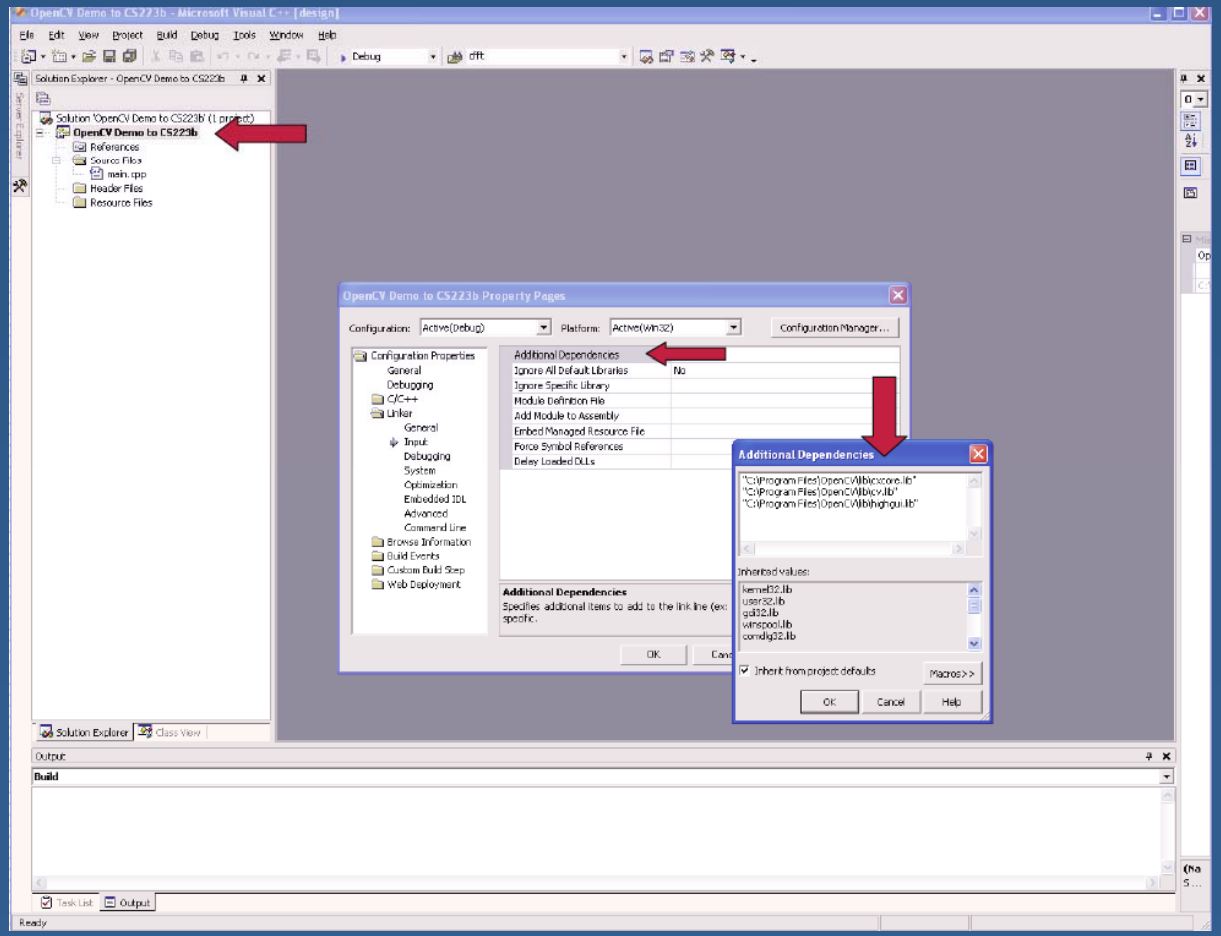
- Library directories

```
OpenCV\lib
```



Update Project Link Settings

- Link against the OpenCV libraries
  - cv.lib
  - cxcore.lib
  - highgui.lib (for HIGHGUI support)
  - cvaux.lib (for CVAUX support)



Basic Example

```
#include <cv.h>
#include <highgui.h>

int main(int argc, char *argv[]) {
  IplImage* img = cvLoadImage("tennis.jpg");
  cvFlip(img, img);
  cvSaveImage("tennis_flipped.jpg", img);
  cvReleaseImage(&img);
}
```

Pixel Processing

```
IplImage* img = cvLoadImage("tennis.jpg");
for (int x=0; x<img->width; x++) {
  for (int y=0; y<img->height; y++) {
    for (int c=0; c<img->nChannels; c++) {
      int idx=x*img->nChannels+y*img->widthStep+c;
      int val=img->imageData[idx];
      img->imageData[idx]=255-val;
      }
  }
}
cvSaveImage("tennis_inverted.jpg", img);
cvReleaseImage(&img);
```

Using Windows

```
cvNamedWindow("ImageWindow");
cvShowImage("ImageWindow", img);
cvWaitKey(0);
```

Loading Video

```
// Open the input video
CvCapture *video = cvCaptureFromFile("stanley.avi");

// Query first frame
cvQueryFrame(video);

// Read video properties
int width =
   cvGetCaptureProperty(video,CV_CAP_PROP_FRAME_WIDTH);
int height =
   cvGetCaptureProperty(video,CV_CAP_PROP_FRAME_HEIGHT);
int nFrames =
   cvGetCaptureProperty(video,CV_CAP_PROP_FRAME_COUNT);
```

Optical Flow

```
cvCalcOpticalFlowPyrLK(grlastframe,
  grcurrframe, pyramid1, pyramid2,
  lastframe_features, currframe_features,
  nFeatures, cvSize(3,3), 5,
  found_features, feature_error,
  cvTermCriteria( CV_TERMCRIT_ITER |
  CV_TERMCRIT_EPS, 20, .3 ), 0);
```

Saving Video

```
// Open a video writer to save the output
CvVideoWriter *writer =
  cvCreateVideoWriter("output.avi",
  -1, 30, cvSize(width, height));

// Write some frames
  cvWriteFrame(writer, lastframe);

// Release when finished
  cvReleaseVideoWriter(&writer);
```

Functions to Check Out

- Image Processing
  - cvSobel, cvLaplace, cvCanny, cvCornerHarris, cvGoodFeaturesToTrack, cvHoughLines2, cvHoughCircles
- Optical Flow
  - cvCalcOpticalFlowPyrLK, cvFindFundamentalMat
- Template Matching
  - cvMatchTemplate, cvMatchShapes, cvCalcEMD2, cvMatchContourTrees
- Motion
  - cvKalmanPredict, cvConDensation, cvAcc, cvMeanShift, cvCamShift
- Segmentation and Grouping
  - cvSnakeImage, cvKMeans2, cvSeqPartition, cvCalcSubdivVoronoi2D, cvCreateSubdivDelaunay2D
- Machine Learning
  - cvHaarDetectObjects