Image Analysis

Software Libraries for Image Analysis (Introduction to Open CV & ImageJ)

K. Buza, Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute of Economics and Information Systems
& Institute of Computer Science
University of Hildesheim
http://www.ismll.uni-hildesheim.de

Software Libraries for Image Analysis

Motivation

ImageJ

OpenCV

Why to use software libraries?

- Low-level image processing functionality (openCV, imageJ)
  - Read and write image files, Show Image, Conversion between color spaces, historgrams, image derivatives (extended Sobel operator), Laplacian of the image, blurring …
- Image analysis (openCV)
  - Image segmentation, Object identification (based on visual image of an object), Face Recognition, Gesture Recognition, Motion recognition, Stereo vision, Edge/line detection …
- Statistical machine learning library (openCV, Weka)
  - Naive Bayes classifier, k-nearest neighbor algorithm, Support Vector Machine, Decision Trees, Boosting, Random forest, Expectation Maximization, Neural Networks

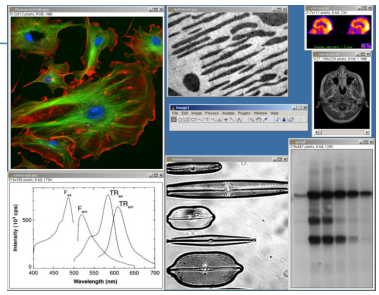
Software Libraries for Image Analysis

Motivation

ImageJ

OpenCV

Image J



- Image Processing Program (Library) in Java
- Developed at National Institute of Health
- "User-written plugins make it possible to solve many image processing and analysis problems, from 3-dimensional live-cell imaging, to radiological image processing, multiple imaging system data comparisons to automated hematology systems." (http://en.wikipedia.org/wiki/ImageJ)
- Source code for ImageJ is freely available
- TIFF, PNG, GIF, JPEG, BMP, DICOM, FITS
- http://rsb.info.nih.gov/ij/ (API here!)

First steps

- Download and install imageJ
- Eclipse
- File/New/Java Project
- Give a name
- Click NEXT

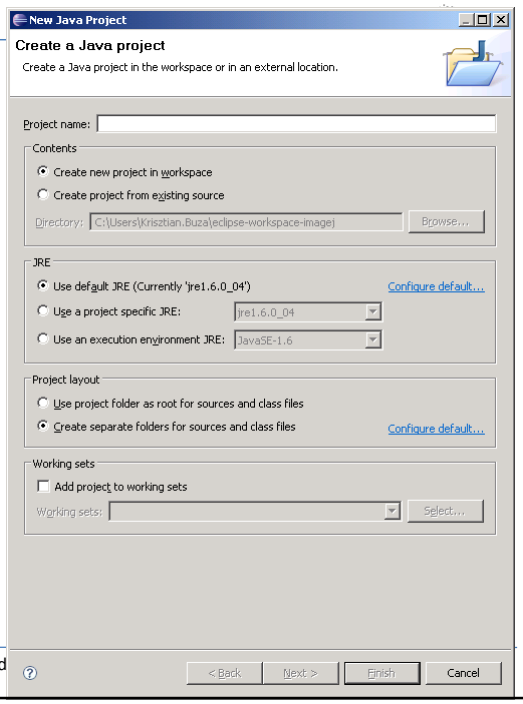
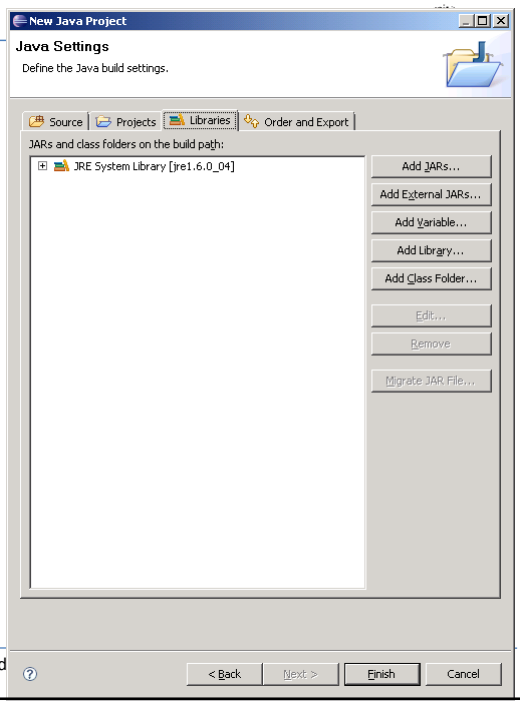
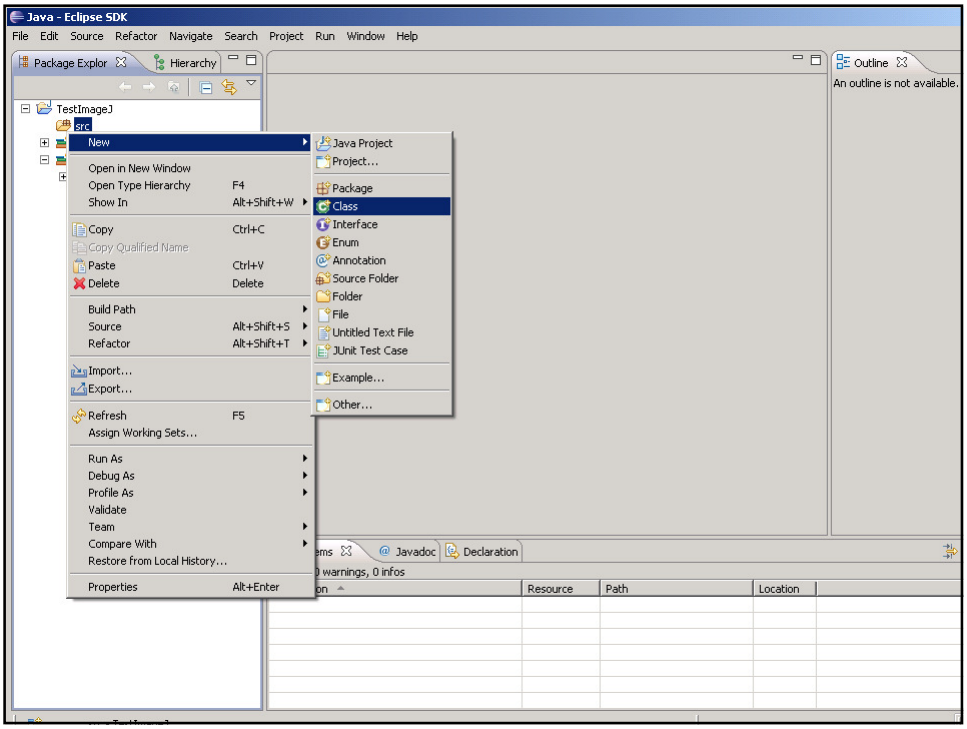


Image Analysis

First steps

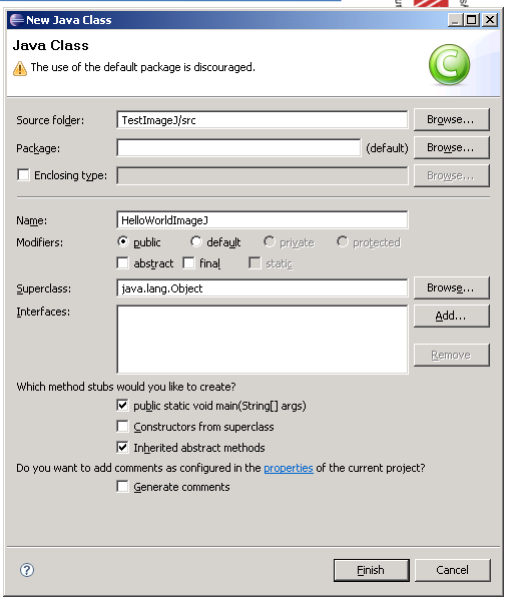
- Libraries
- Add External JARs
- C:\Program Files\ ImageJ\ij.jar





First Steps

- Right-click on „src“
- New/Class
- Select public static void main(…)
- Finish

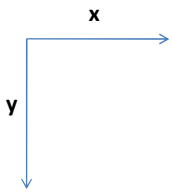


Hello World


```
import ij.io.*;
import ij.*;

public class HelloWorldImageJ {
   public static void main(String[] args) {
        Opener opener = new Opener();
        ImagePlus img = opener.openImage(
           "C:\\Users\\Krisztian.Buza\\ImageProcessing\\"+
           "VOC2008-data\\data\\VOCtrainval_14-Jul-2008\\"+
           "VOCdevkit\\VOC2008\\JPEGImages\\2008_000711.jpg");
        img.show();
   }
}
```

Hello World

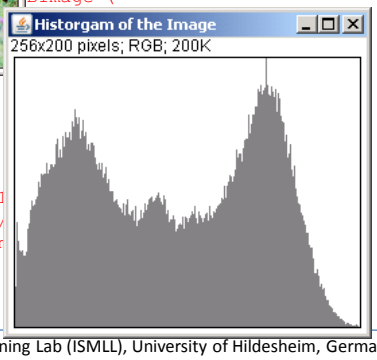
```
import ij.io.*;
import ij.*;
import ij.process.*;

public class HelloWorldImageJ {
   public static void main(String[] args) {
      Opener opener = new Opener();
      ImagePlus img = opener.openImage("...");
      img.show();

      ImageProcessor img_p = img.getProcessor();

      int[] histogram = new int[256];
      for (int x=0;x<img_p.getWidth();x++) {
          for (int y=0;y<img_p.getHeight();y++) {
              int[] one_pixel = img_p.getPixel(x, y, null);
              int r = one_pixel[0];
              int g = one_pixel[1];
              int b = one_pixel[2];
              int intensity = (r+g+b)/3;
              histogram[intensity]++;
          }
      }
} }
```



Hello World

```
import ij.io.*;
import ij.*;
import ij.process.*;
import ij.gui.*;
import java.awt.*;

public class HelloWorldImageJ {
   public static void main(String[] args) {
   (…)
   ImagePlus histogram_image = NewImage.createRGBImage (
     "Historgam of the Image", histogram.length, 200, 1, NewImage.FILL_WHITE);
   ImageProcessor histogram_image_p = histogram_image.getProcessor();

   int hist_max = 0;
   for (int i=0;i<histogram.length;i++)
        if (histogram[i]>hist_max) hist_max = histogram[i];

   for (int i=0;i<histogram.length;i++) {
        histogram_image_p.setColor(new Color(128,128,128));
        int column_length = (200*histogram[i]/hist_max);
        histogram_image_p.drawRect(i,200-column_length,1,column_length);
   }
   histogram_image.show();
} }
```

Image Analysis




```
impor
impor
impor
impor
impor

publi
    pu
    (…
    Im

    Im

    int hist_max = 0;
    for (int i=0;i<histogram.length;i++)
          if (histogram[i]>hist_max) hist_max =

    for (int i=0;i<histogram.length;i++) {
          histogram_image_p.setColor(new Color(1
          int column_length = (200*histogram[i]/
          histogram_image_p.drawRect(i,200-colur
    }
    histogram_image.show();
}   }
```

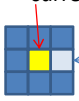
rld

BImage (

Image Analysis

```
(...)
public class HelloWorldImageJ {
   public static void main(String[] args) {
    (…)
    ImagePlus noise_reduced_image = NewImage.createRGBImage ("Noise Reduced Image ",
        img_p.getWidth(),img_p.getHeight(), 1, NewImage.FILL_WHITE);
    ImageProcessor blurred_image_p = noise_reduced_image.getProcessor();
    int[][] kernel = { {1,2,1}, {2,3,2}, {1,2,1} };
    int kernel_sum = 4 + 7 + 4;
    for (int x=0;x<img_p.getWidth();x++) {
        for (int y=0;y<img_p.getHeight();y++) {
           int[][][] pixels = new int[3][3][];
            for (int i=0;i<3;i++)
             for (int j=0;j<3;j++)
                 pixels[i][j] = img_p.getPixel(x-1+i, y-1+j, null);
            int[] new_pixel = new int[3];
            for (int i=0;i<3;i++)
                for (int j=0;j<3;j++)
                    for (int k=0;k<3;k++)
                        new_pixel[k] += kernel[i][j]*pixels[i][j][k];
            for (int k=0;k<3;k++) new_pixel[k] /= kernel_sum;
            blurred_image_p.putPixel(x,y,new_pixel);
        }
    }
    noise_reduced_image.show();
}   }
```

current pixel (x, y)



(x+1, y)
pixels[2][1][0] – red intensity
pixels[2][1][1] – green intensity
…

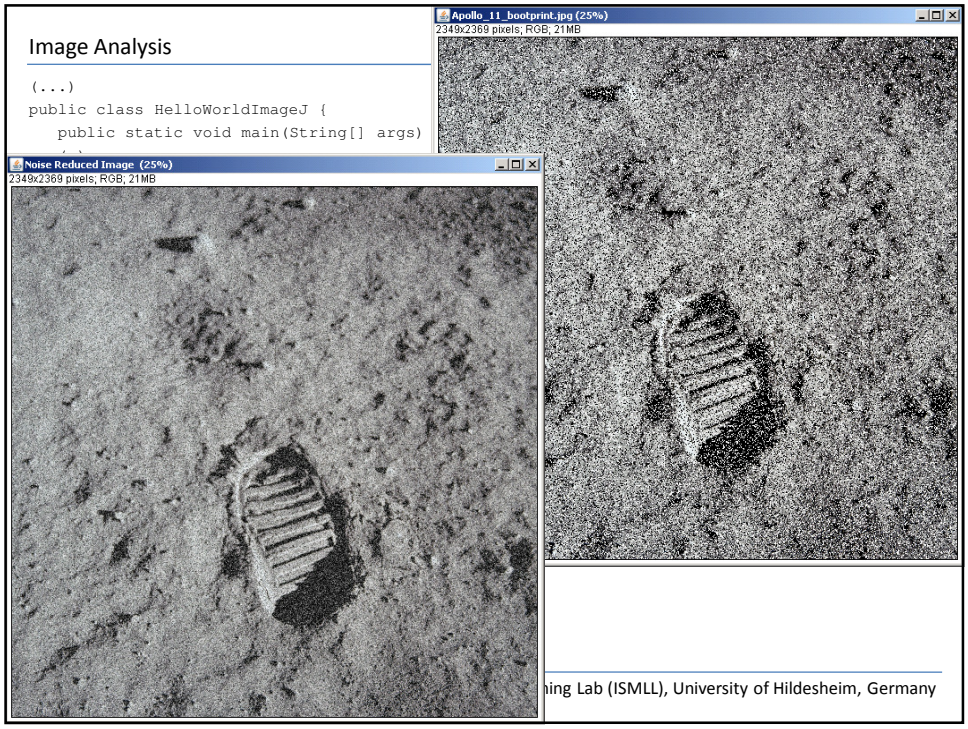


Image Analysis
(...)
public class HelloWorldImageJ {
public static void main(String[] args)
Apollo_11_bootprint.jpg (25%)
2349x2369 pixels; RGB; 21MB
Noise Reduced Image (25%)
2349x2369 pixels; RGB; 21MB
ing Lab (ISMLL), University of Hildesheim, Germany

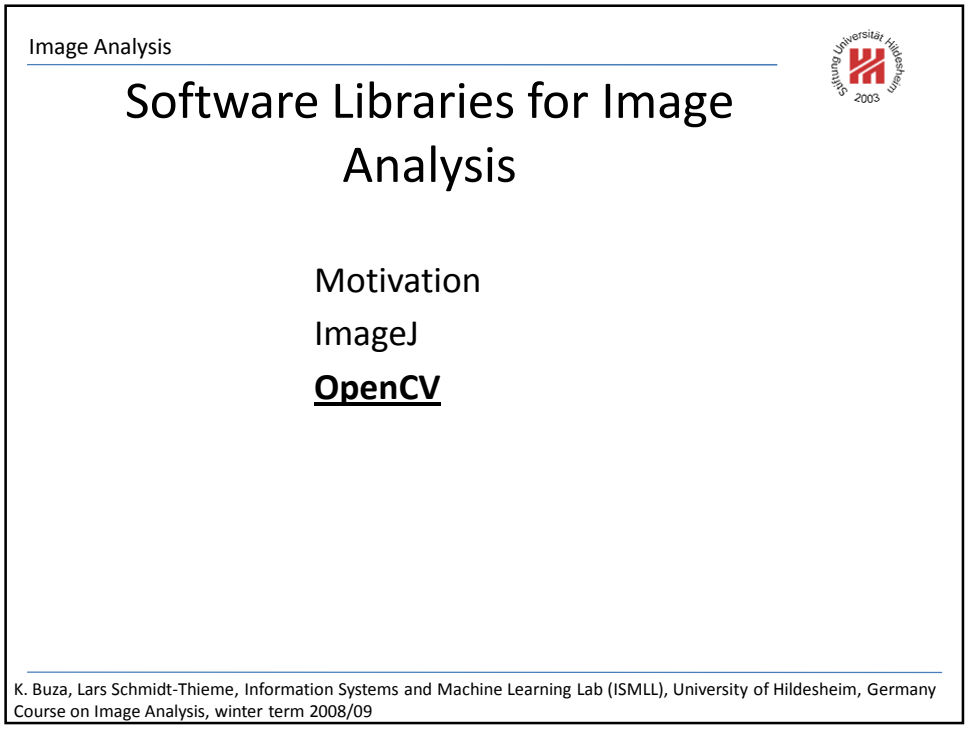
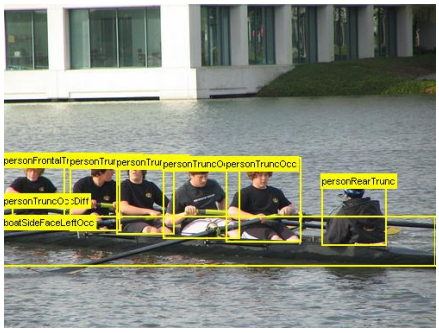


Image Analysis
Software Libraries for Image Analysis
Motivation
ImageJ
OpenCV
K. Buza, Lars Schmidt-Thieme, Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim, Germany
Course on Image Analysis, winter term 2008/09

What is Open CV ?

- Computer vision library
- Developed by Intel
- Free for commercial and research use
  - BSD-like licence, open source
- Cross-platform
  - Windows, Linux, Mac OS, embedded operating systems
- Focuses on real-time image processing
- C functions and C++ classes
- Interfaces to several softwares (like Octave)
- More than 500 algorithms

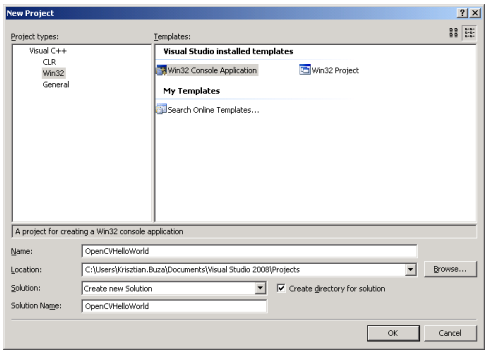


http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/examples/boat_07.jpg

Getting Started

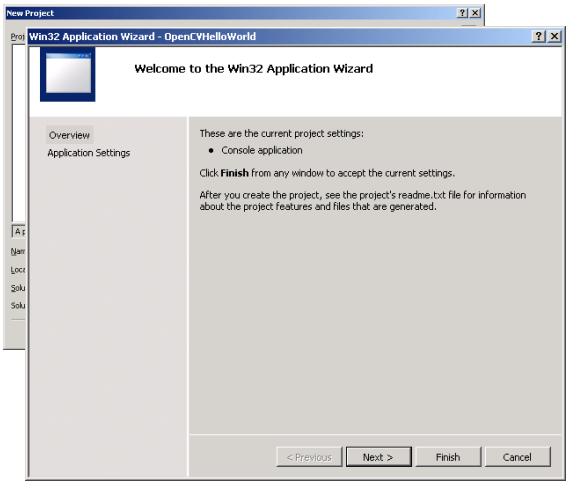
- C, C++ Compiler:
  - Microsoft Visual C++ 2008 Express
  - microsoft.com/express
- Open CV 1.0 (!)
  - install executable
  - (we had some strange bug with version 1.1)

Hello World



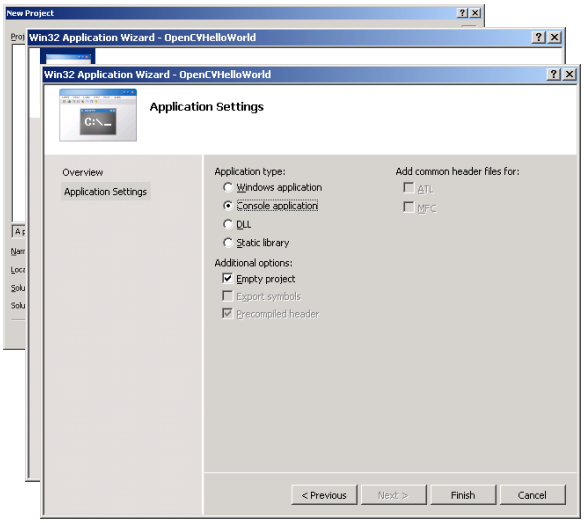
- File → New → Project
- Win32 Console Application
- Name

Hello World



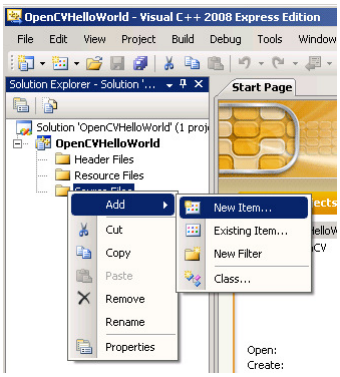
- File → New → Project
- Win32 Console Application
- Name
- Next (not Finish!)

Hello World



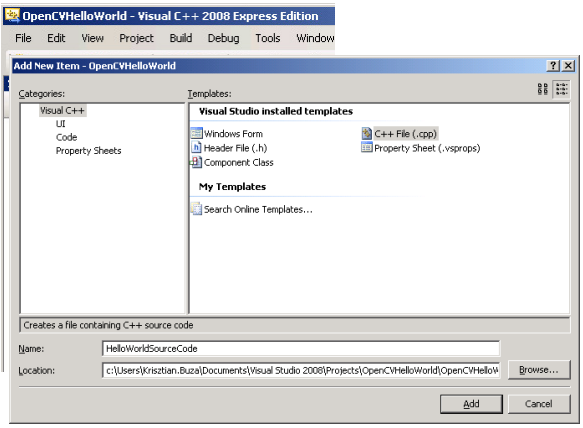
- File → New → Project
- Win32 Console Application
- Name
- Next
- Empty Project
- Finish

Hello World

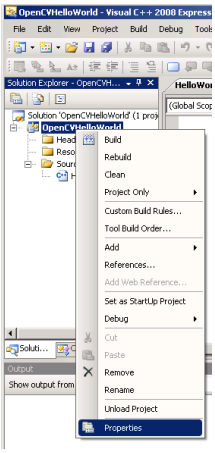


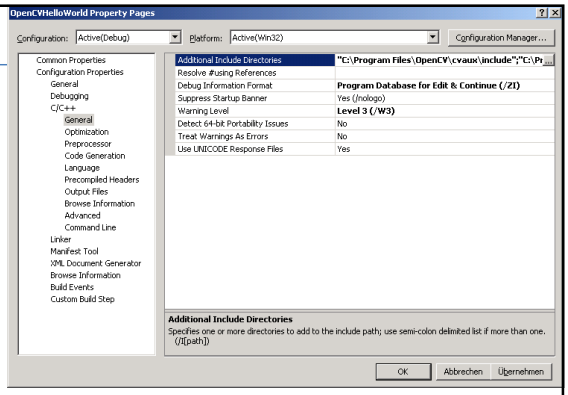
- Source Files → Add → New Item

Hello World

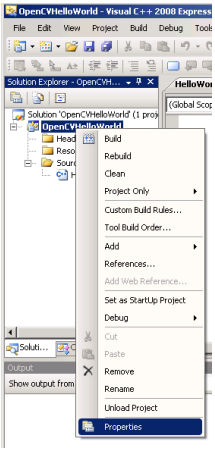


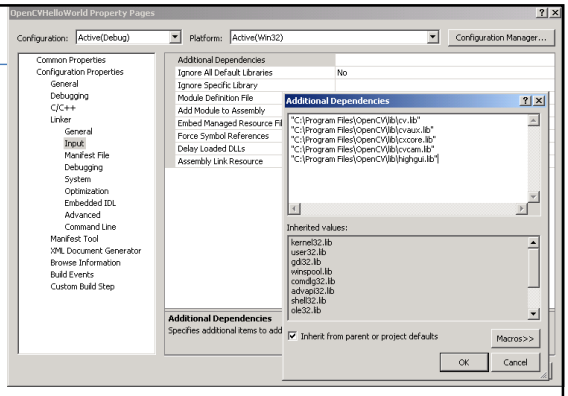
- Source Files → Add → New Item
- Name
- Add





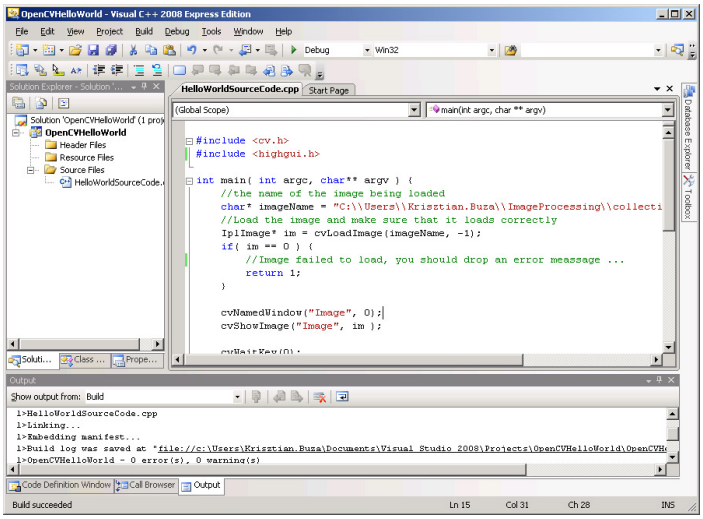
- Project Properties
- **C/C++ tab**
- **Additional Include Directories**
- C:\Program Files\OpenCV\cvaux\include\
- C:\Program Files\OpenCV\cxcore\include\
- C:\Program Files\OpenCV\cv\include\
- C:\Program Files\OpenCV\otherlibs\highgui\
- C:\Program Files\OpenCV\otherlibs\cvcam\include\





- Project Properties
- **Linker / Input tab**
- **Additional dependencies**
- "C:\Program Files\OpenCV\lib\cv.lib"
- "C:\Program Files\OpenCV\lib\cvaux.lib"
- "C:\Program Files\OpenCV\lib\cxcore.lib"
- "C:\Program Files\OpenCV\lib\cvcam.lib"
- "C:\Program Files\OpenCV\lib\highgui.lib"

Hello World



Hello World

```
int main( int argc, char** argv ) {
    char* imageName = "C:\\Users\\Krisztian.Buza\\Pictures\\beton.jpg";
    IplImage* im = cvLoadImage(imageName, -1);
    if( im == 0 ) {
        //Image failed to load, you should drop an error meassage ...
        return 1;
    }
    cvNamedWindow("Image", 0);
    cvShowImage("Image", im );
    cvWaitKey(0);
    cvReleaseImage(&im);
    cvDestroyWindow("Image");
    return 0;
}
```


Run

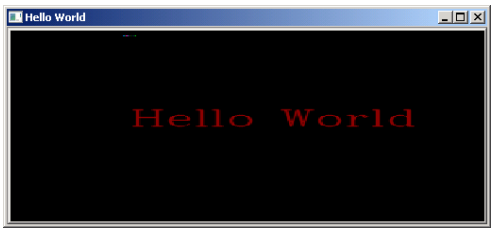
- Debug / Start Without Debugging

- Use *cvNamedWindow(… , **1**)* to display the image in the original size

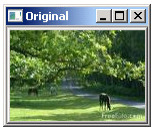
Some more examples

- Write a „Hello World“ on the screen
- Transformation between color spaces
- Histogram
- Resize image


```
void print_helloWorld() {
    int height = 320; int width = 240; //declare for the height and width of the image
     CvPoint pt = cvPoint( height/4, width/2 );      //specify the point to place the text
    IplImage* hw = cvCreateImage(cvSize(height, width), 8, 3);  //8 bit, 3 channel

    CvFont font;
    cvInitFont( &font, CV_FONT_HERSHEY_COMPLEX, 1.0, 1.0, 0, 1, CV_AA);
    cvPutText(hw, "Hello World", pt, &font, CV_RGB(150, 0, 0) );  // text => image

    cvNamedWindow("Hello World", 0); //create the window
    cvShowImage("Hello World", hw);
    //display the image in the window
    cvWaitKey(0);

    cvReleaseImage( &hw );
    cvDestroyWindow("Hello World");
    //UNLIKE JAVA MEMORY IS NOT
    //REUSED AUTOMATICALLY IN C++ !
}
```



```
void color2black_and_white() {
    char* imageName = "C:\\Users\\Krisztian.Buza\\collection1\\forest4.jpg";
    IplImage* im = cvLoadImage(imageName, -1);
    if( im == 0 ) { return ; }
    cvNamedWindow("Original", 0); cvShowImage("Original", im );

    IplImage* grayImage = cvCreateImage(cvSize(im->width,im->height),
            IPL_DEPTH_8U, 1);       // 1-channel image of the same size as the original
    cvCvtColor(im, grayImage, CV_BGR2GRAY);
            //convert the original image to gray (use BGR here, not RGB!)
            // conversion to HSV is also possible with cvCvtColor
    cvNamedWindow("BW", 0);
    cvShowImage("BW", grayImage );
    cvWaitKey(0);
    cvReleaseImage( &grayImage );
    cvReleaseImage( &im );
}
```




```
void black_and_white_histogram() {
    char* imageName = "C:\\Users\\Krisztian.Buza\\collection1\\forest4.jpg";
    IplImage* im = cvLoadImage(imageName, -1);
    if( im == 0 ) { return ; }
    IplImage* grayImage = cvCreateImage(cvSize(im->width,im->height), IPL_DEPTH_8U, 1);
    cvCvtColor(im, grayImage, CV_BGR2GRAY);
    CvRect rect = cvRect(0, 0, 500, 600 ); //create a rectangular area to evaluate
    cvSetImageROI(grayImage, rect); // establish a region of interest with the rect. Area
    IplImage* histImage = cvCreateImage(cvSize(320,200), 8, 1); // image for histogram

    int hist_size = 100;
    CvHistogram* hist = cvCreateHist(1, &hist_size, CV_HIST_ARRAY);
    cvCalcHist( &grayImage, hist, 0, NULL );  //calculate histogram

    ( … ) // draw histogram image and display it

}
```

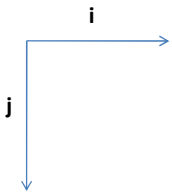
```
void black_and_white_histogram() {   // draw histogram image and display it
    (…)
    float min_value, max_value; int min_idx, max_idx;
    cvGetMinMaxHistValue( hist, &min_value, &max_value, &min_idx, &max_idx);
    cvScale( hist->bins, hist->bins, ((double)histImage->height)/max_value, 0 );
          //scale the bin values so that they will fit in the image representation
    cvSet( histImage, cvScalarAll(255), 0 );
    int bin_w = cvRound((double)histImage->width/hist_size);
    int i; float mean=0, variance=0;
    for( i = 0; i < hist_size; i++ ) {
       cvRectangle( histImage, cvPoint(i*bin_w, histImage->height),
          cvPoint((i+1)*bin_w, histImage->height - cvRound(cvGetReal1D(hist->bins,i))),
          cvScalarAll(0), -1, 8, 0 );
       float* bins = cvGetHistValue_1D(hist,i);  // You can get histogram values this way,
       float one_histogram_value = bins[0];      // if you need it for calculations
    }
    (…) // Display histogram image
}
```



```
void resize() {
    char* imageName = "C:\\Users\\Krisztian.Buza\\collection1\\beach1.jpg";
    IplImage* im = cvLoadImage(imageName, -1); if( im == 0 ) { return ; }
    IplImage* resizedImage =
       cvCreateImage(cvSize(im->width*5,im->height*5), IPL_DEPTH_8U, 3);

    cvResize( im , resizedImage, CV_INTER_LINEAR );
    /* cvResize( im , resizedImage, CV_INTER_LINEAR );
      – CV_INTER_NN - nearest-neigbor interpolation
      – CV_INTER_LINEAR - bilinear interpolation (used by default)
      – CV_INTER_AREA - resampling using pixel area relation.
      – CV_INTER_CUBIC - bicubic interpolation
    cvSmooth( const CvArr* src, CvArr* dst, int smoothtype=CV_GAUSSIAN,
    int param1=3, int param2=0, double param3=0 ) */

    (…)  // Display image
}
```

Access to image pixels

```
IplImage* im = cvLoadImage(imageName, -1);

for (int i=0 ;  i < im->height ; i++) {
    for (int j=0 ; j < im->width ; j++) {
        uchar* chanel_values =
                &CV_IMAGE_ELEM(im, uchar, i, j*3);
        // 3-channel, color image
        // i = horizontal position
        // j = vertical position
        // chanel_values[0],  chanel_values[1], …
        // you can read and write (change) pixel values
    }
}
```



C/C++ in a nutshell

- Java-like syntax
  - int, char, { , } , if, while … like in Java
  - Functions ≈ Methods
  - Pointers ≈ References
- Some important differences compared to Java
  - Not fully object-oriented
  - Pointers
  - Arrays and Pointers
  - Memory allocation
  - Character arrays often used as strings (although there is a String class in C++)
  - Logic expressions
  - …

C/C++ is not fully object-oriented

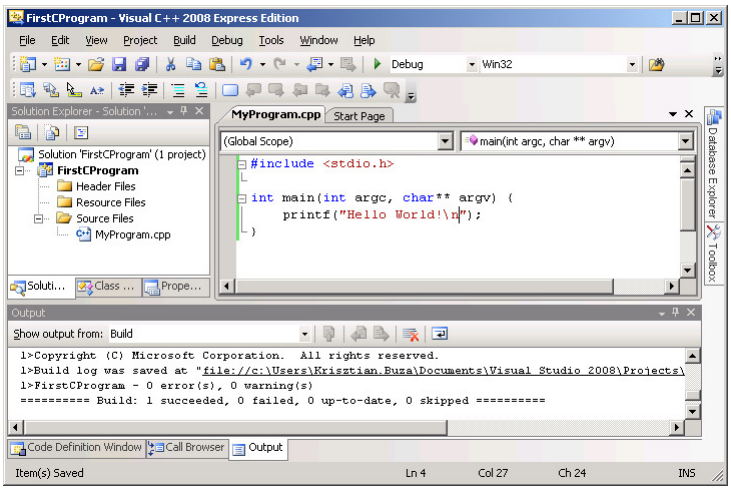


Image Analysis

C/C++ is not fully object-oriented

- One does not need to use classes
  - But one ***CAN*** use classes (C++), if it fits the task better
- One can write programs with functions
- Functions have a return type (like in java)
- If one does not use classes, no function access control (like public, private, protected)
- Function *main* similar to the *main* method in Java

K. Buza, Lars Schmidt-Thieme, Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim, Germany
Course on Image Analysis, winter term 2008/09

Pointers

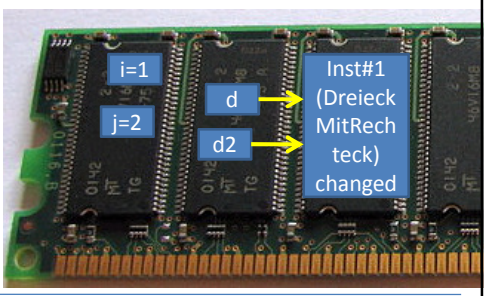
- In Java
  - char, int, byte … → value is stored (char ≠ Char ⚠)
  - Objects: **new** creates an object instance, variables are *references* to objects (when calling a method on an object you do not need to know that the variable is a reference)
- In C/C++

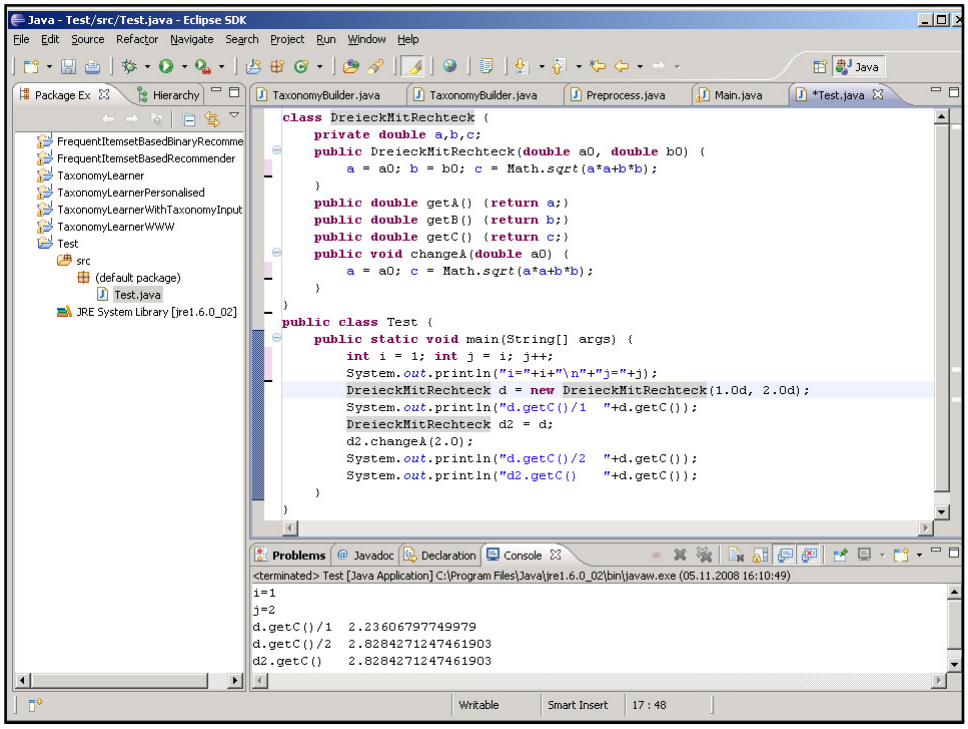
⚠ – Variables can store both values and pointers (be careful and make sure if you store the pointer or the value)

⚠ – Pointer for an int → memory adress where it is stored (one usually does not want to use this in calculations)

Value vs. Reference – Example (Java)

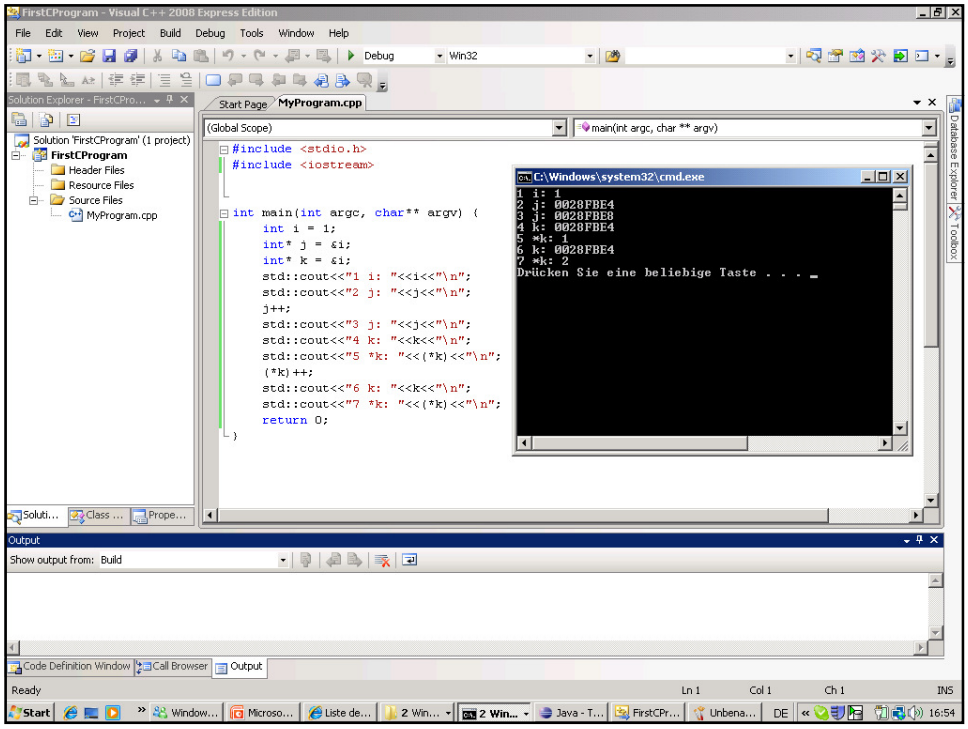
```
int i = 1;
int j = i;
j++;
DreieckMitRechteck d = new DreieckMitRechteck(1.0d, 2.0d);
DreieckMitRechteck d2 = d;
d2.changeA(2.0);
```

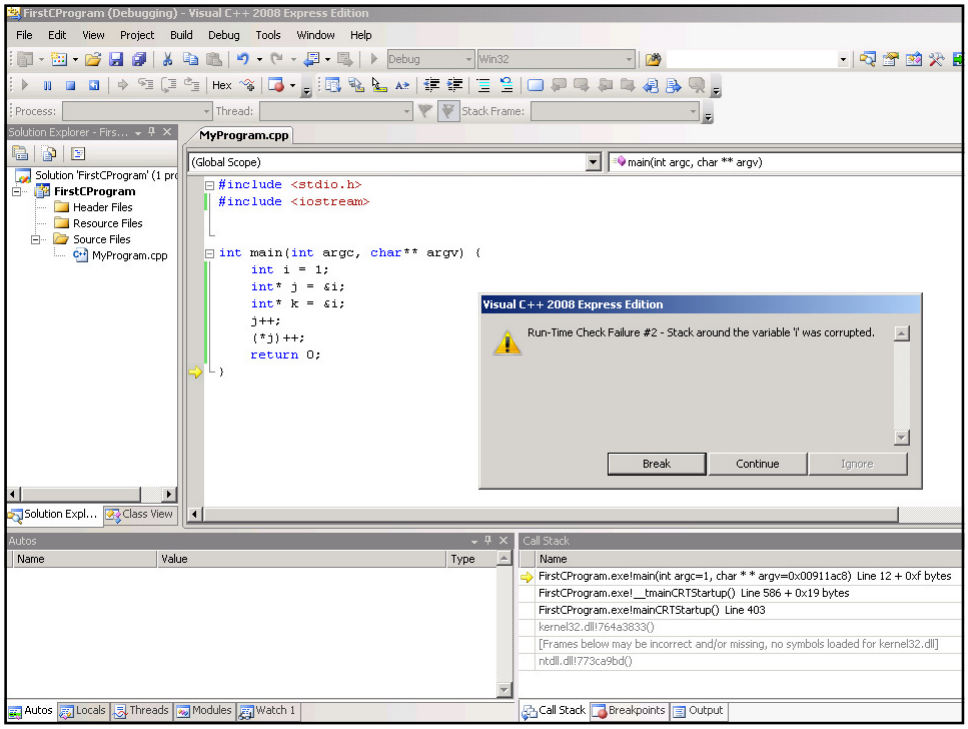




Value vs. Pointer – Example (C/C++)

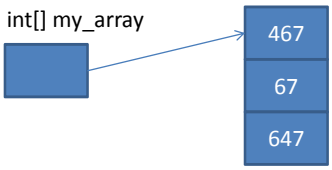
```
int i = 1;          // the value of i is set to 1
int* j = &i;        // use * to define pointers
int* k = &i;        // j,k are pointers, they include the memory adress of i
                    // &i means: the memory adress of i
j++;                // the memory adress is changed, as you do not what is at
                    // the new adress, please be very careful
                    // do not change the content at the new adress
                    // if you want to avoid strange bugs!!!
(*k)++;             // * turns the pointer into value (you can access the value
                    // stored at the given adress)
                    // the value of i has been changed!
                    // (k is just a pointer to i)
```


FirstCProgram - Visual C++ 2008 Express Edition
File Edit View Project Build Debug Tools Window Help
Debug
Win32
Solution Explorer - FirstCPro...
Solution 'FirstCProgram' (1 project)
FirstCProgram
Header Files
Resource Files
Source Files
MyProgram.cpp
Start Page
MyProgram.cpp
(Global Scope)
main(int argc, char ** argv)
#include <stdio.h>
#include <iostream>
int main(int argc, char** argv) {
int i = 1;
int* j = &i;
int* k = &i;
std::cout<<"1 i: "<<i<<"\n";
std::cout<<"2 j: "<<j<<"\n";
j++;
std::cout<<"3 j: "<<j<<"\n";
std::cout<<"4 k: "<<k<<"\n";
std::cout<<"5 *k: "<<(*k)<<"\n";
(*k)++;
std::cout<<"6 k: "<<k<<"\n";
std::cout<<"7 *k: "<<(*k)<<"\n";
return 0;
}
C:\Windows\system32\cmd.exe
1 i: 1
2 j: 0028FBE4
3 j: 0028FBE8
4 k: 0028FBE4
5 *k: 1
6 k: 0028FBE4
7 *k: 2
Drücken Sie eine beliebige Taste . . .
Output
Show output from: Build
Code Definition Window
Call Browser
Output
Ready
Ln 1
Col 1
Ch 1
INS


FirstCProgram (Debugging) - Visual C++ 2008 Express Edition
File Edit View Project Build Debug Tools Window Help
Debug
Win32
Hex
Process:
Thread:
Stack Frame:
Solution Explorer - Firs...
Solution 'FirstCProgram' (1 pr
FirstCProgram
Header Files
Resource Files
Source Files
MyProgram.cpp
(Global Scope)
main(int argc, char ** argv)
#include <stdio.h>
#include <iostream>
int main(int argc, char** argv) {
int i = 1;
int* j = &i;
int* k = &i;
j++;
(*j)++;
return 0;
}
Visual C++ 2008 Express Edition
Run-Time Check Failure #2 - Stack around the variable 'i' was corrupted.
Break
Continue
Ignore
Autos
Name
Value
Type
Call Stack
Name
FirstCProgram.exe!main(int argc=1, char * * argv=0x00911ac8) Line 12 + 0xf bytes
FirstCProgram.exe!__tmainCRTStartup() Line 586 + 0x19 bytes
FirstCProgram.exe!mainCRTStartup() Line 403
kernel32.dll!764a3833()
[Frames below may be incorrect and/or missing, no symbols loaded for kernel32.dll]
ntdll.dll!773ca9bd()
Autos
Locals
Threads
Modules
Watch 1
Call Stack
Breakpoints
Output

Arrays and pointers

- Pointer arithmetic
- Memory allocation (calloc)




```
int* my_array = (int *) calloc(20, sizeof(int));
my_array[0] = 23;
my_array[1] = 12;
my_array[3] = 45;
```

```
int* my_array = (int *) calloc(20, sizeof(int));
*(my_array)= 23;
*(my_array+1) = 12;
*(my_array+2) = 45;
```


Strings vs. char arrays

Often char arrays are used instead of strings

```
char path[] = "C:\\Users\\Krisztian.Buza\\";
char* file = "2007_006240.jpg";
char* imageName =
    (char *) calloc(strlen(path) + strlen(file) + 1, sizeof(char));
strcat(imageName, path);
strcat(imageName, file);
std::cout<<imageName<<"\n";
std::cout<<path<<"\n";
```

Logic expressions

- Java-like expressions allowed
  if (i==0) { …}
- integer = logic expression in C/C++
  0 → false, everything else → true

```
int i = 0; int j = 1; int k = -1;
if (i) std::cout << "i ";
if (j) std::cout << "j ";
if (k) std::cout << "k ";
```

Actually, logic expressions are evaluated as integers in C/C++

System.out.println(i==5);	→	true or false
std::cout << (i==5);	→	1 or 0

Next steps

Tutorial
http://www.scribd.com/doc/4020623/opencv-introduction-2007June9
http://www.site.uottawa.ca/~laganier/tutorial/opencv+directshow/cvision.htm

Reference Manual
http://vision.cis.udel.edu/opencv/

Machine Learning Library for openCV
http://www.seas.upenn.edu/~bensapp/opencvdocs/ref/opencvref_ml.htm

Other sources
http://www.softintegration.com/products/thirdparty/opencv/demos/
http://groups.yahoo.com/group/OpenCV/