EE569: Coding Style

Chun-Ting Huang

Slides are credited to Xiaqing Pan

Outline

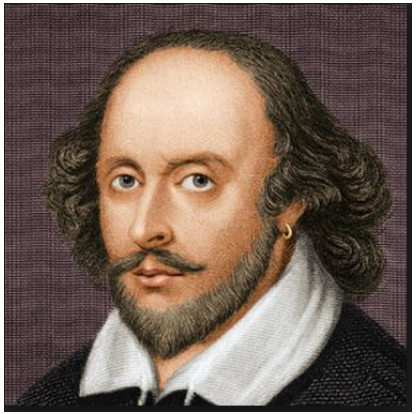
- Readability
- Coding Style Example
- Coding Style Conclusion

Readability

- Common cases
  - When a programmer finds problems while using other's codes
    - "These codes have bugs !!!"
  - When another programmer's codes got reported with bugs
    - "They do not know how to use it correctly !!!"
- Why?
  - Codes are the only communication between programmers and their users

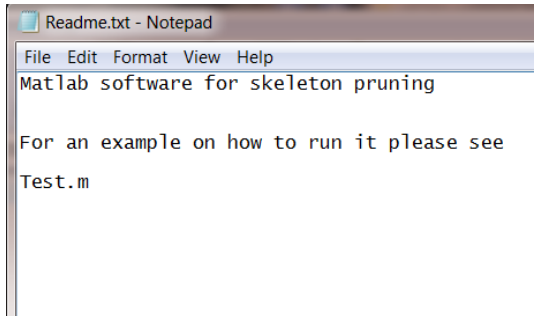
Readability

“The Eyes are the window to your soul”



Readability

- Readme (How to write a good readme)




```
FOlder contains LESH feature vector extraction code.

 LESH (Local Energy based Shape Histogram) feature Extraction

usage: [lesh]=calc_LESH(im)

Input:
im = Image or Local patch

output:
 Shape_vect= LESH feature vector (128 dimentional for 16 partitions and 512 dimentional for 64 partitions of the
             image/patch. (see FeatureParam.m)


Works on image or patch of any size (preferably square).

Recomended partition size `w' is 8 (512-dim vector) for an image of size greater than 64x64. In patches of size 32

NOTE; There are a number of parametes to adjust for different applications.
file FeatureParam.m may be modified to change e.g. the number of scale and orientation of GABOR filter. partion si
```

Readability

- What is your miserable experience with other's readme?
  - What platform should I use?
  - Missing files???!!!
  - What kind of input should I use?
  - How to tune the parameters?
  - I just want to use part of codes, but how?
  - …

Readability

- Programmers can benefit from a good Readme

- When you write a Readme file, treat yourself as a guy knows nothing about your codes
  - Your codes may be used again by yourself after a long time
  - You need to maintain your own codes
  - Others may take over your codes

- Some general suggestions for writing a good Readme

Readability

- Author info
  - Name, date, company, email …
- Functionality
  - What does the program do?
- Version track
  - A version number
  - What are previous changes
- How to compile
  - Platform, compiler, IDE, make file…

Readability

- How to run
  - Input format, parameter range, output…
- Predictable errors
  - Unreasonable parameter or input?
- Descriptions about important functions
  - Framework, important structure, function tree
- File list

Coding Style Example

- Header File

```
#include "stdio.h“

void binarize(unsigned char *a, int w, int h);
```

Coding Style Example

- Source File

```
#include "binarize.h"

void binarize(unsigned char *a, int w, int h)
{
int i, j;

if (a == NULL) return;

for (j = 0; j<h; j++)
{
for (i = 0; i<w; i++)
{
if (a[j*w+i]<128) a[j*w+i] = 0;
else a[j*w+i] = 255;
}
}

return;
}
```

Coding Style Example

- This code is runnable if a main function is added
- What are the problems with this code?

Coding Style Example

- Header File

#include "stdio.h“

*void binarize(unsigned char *a, int w, int h);*

Use “include guard” to prevent multiple inclusion

Coding Style Example

- Header File

```
#ifndef BINARIZE_H_
#define BINARIZE_H_

#include "stdio.h“

void binarize(unsigned char *a, int w, int h);

#endif
```

Use “include guard” to prevent multiple inclusion

Coding Style Example

- Source File

```
#include "binarize.h"

void binarize(unsigned char *a, int w, int h)
{
int i, j;

if (a == NULL) return;

for (j = 0; j<h; j++)
{
for (i = 0; i<w; i++)
{
if (a[j*w+i]<128) a[j*w+i] = 0;
else a[j*w+i] = 255;
}
}

return;
}
```

Text alignment

Coding Style Example

- Source File

```
#include "binarize.h"

void binarize(unsigned char *a, int w, int h)
{
        int i, j;

        if (a == NULL) return;

        for (j = 0; j<h; j++)
        {
            for (i = 0; i<w; i++)
            {
                if (a[j*w+i]<128) a[j*w+i] = 0;
                else a[j*w+i] = 255;
            }
        }

        return;
}
```

Text alignment

Coding Style Example

- Source File

```
#include "binarize.h"

void binarize(unsigned char *a, int w, int h)
{
        int i, j;

        if (a == NULL) return;

        for (j = 0; j<h; j++)
        {
            for (i = 0; i<w; i++)
            {
                if (a[j*w+i]<128) a[j*w+i] = 0;
                else a[j*w+i] = 255;
            }
        }

        return;
}
```

Semantic variable name

Coding Style Example

- Source File

```
#include "binarize.h"

void binarize(unsigned char *ori_img, int width, int height)
{
        int i, j;

        if (ori_img == NULL) return;

        for (j = 0; j<height; j++)
        {
            for (i = 0; i<width; i++)
            {
                if (ori_img[j*width+i]<128) ori_img[j*width+i] = 0;
                else ori_img[j*width+i] = 255;
            }
        }

        return;
}
```

Semantic variable name

Coding Style Example

- Source File

```
#include "binarize.h"

void binarize(unsigned char *ori_img, int width, int height)
{
        int i, j;

        if (ori_img == NULL) return;

        for (j = 0; j<height; j++)
        {
            for (i = 0; i<width; i++)
            {
                if (ori_img[j*width+i]<128) ori_img[j*width+i] = 0;
                else ori_img[j*width+i] = 255;
            }
        }

        return;
}
```

Do NOT write ONLY verb as function name

Coding Style Example

- Source File

```
#include "binarize.h"

void binarizeImg(unsigned char *ori_img, int width, int height)
{
        int i, j;

        if (ori_img == NULL) return;

        for (j = 0; j<height; j++)
        {
            for (i = 0; i<width; i++)
            {
                if (ori_img[j*width+i]<128) ori_img[j*width+i] = 0;
                else ori_img[j*width+i] = 255;
            }
        }

        return;
}
```

Do NOT write ONLY verb as function name

Coding Style Example

- Source File

```
#include "binarize.h"

void binarizeImg(unsigned char *ori_img, int width, int height)
{
        int i, j;

        if (ori_img == NULL) return;

        for (j = 0; j<height; j++)
        {
            for (i = 0; i<width; i++)
            {
                if (ori_img[j*width+i]<128) ori_img[j*width+i] = 0;
                else ori_img[j*width+i] = 255;
            }
        }

        return;
}
```

Label const for reference parameters

Coding Style Example

- Source File

```
#include "binarize.h"

void binarizeImg(unsigned char *ori_img, const int &width, const int &height)
{
        int i, j;

        if (ori_img == NULL) return;

        for (j = 0; j<height; j++)
        {
            for (i = 0; i<width; i++)
            {
                if (ori_img[j*width+i]<128) ori_img[j*width+i] = 0;
                else ori_img[j*width+i] = 255;
            }
        }

        return;
}
```

Label const for reference parameters

Coding Style Example

- Source File

```
#include "binarize.h"

void binarizeImg(unsigned char *ori_img, const int &width, const int &height)
{
        int i, j;

        if (ori_img == NULL) return;

        for (j = 0; j<height; j++)
        {
            for (i = 0; i<width; i++)
            {
                if (ori_img[j*width+i]<128) ori_img[j*width+i] = 0;
               else ori_img[j*width+i] = 255;
            }
         }

        return;
}
```

Do NOT modify original input

Coding Style Example

- Source File

```
#include "binarize.h"

void binarizeImg(unsigned char *ori_img, const int &width, const int &height, unsigned char *res_img)
{
        int i, j;

        if (ori_img == NULL) return;

        for (j = 0; j<height; j++)
        {
            for (i = 0; i<width; i++)
            {
                if (ori_img[j*width+i]<128) res_img[j*width+i] = 0;
                else res_img[j*width+i] = 255;
            }
        }

        return;
}
```

Do NOT modify original input

Coding Style Example

- Source File

```
#include "binarize.h"

void binarizeImg(unsigned char *ori_img, const int &width, const int &height, unsigned char *res_img)
{
        int i, j;

        if (ori_img == NULL) return;

        for (j = 0; j<height; j++)
        {
            for (i = 0; i<width; i++)
            {
                if (ori_img[j*width+i]<128) res_img[j*width+i] = 0;
                else res_img[j*width+i] = 255;
            }
        }

        return;
}
```

Do NOT use NULL: ONLY in C

Coding Style Example

- Source File

```
#include "binarize.h"

void binarizeImg(unsigned char *ori_img, const int &width, const int &height, unsigned char *res_img)
{
        int i, j;

        if (ori_img == 0) return;

        for (j = 0; j<height; j++)
        {
            for (i = 0; i<width; i++)
            {
                if (ori_img[j*width+i]<128) res_img[j*width+i] = 0;
                else res_img[j*width+i] = 255;
            }
        }

        return;
}
```

Do NOT use NULL: ONLY in C

Coding Style Example

- Source File

```
#include "binarize.h"

void binarizeImg(unsigned char *ori_img, const int &width, const int &height, unsigned char *res_img)
{
        int i, j;

        if (ori_img == 0) return;

        for (j = 0; j<height; j++)
        {
            for (i = 0; i<width; i++)
            {
                if (ori_img[j*width+i]<128) res_img[j*width+i] = 0;
                else res_img[j*width+i] = 255;
            }
        }

        return;
}
```

Initialize variable + Initialize iterator before loop

Coding Style Example

- Source File

```
#include "binarize.h"

void binarizeImg(unsigned char *ori_img, const int &width, const int &height, unsigned char *res_img)
{
        int i = 0, j = 0;

        if (ori_img == 0) return;

        i = 0; j = 0;

        for (j = 0; j<height; j++)
        {
            for (i = 0; i<width; i++)
            {
                if (ori_img[j*width+i]<128) res_img[j*width+i] = 0;
                else res_img[j*width+i] = 255;
            }
        }

        return;
}
```

Initialize variable + Initialize iterator before loop

Coding Style Example

- Source File

```
#include "binarize.h"

void binarizeImg(unsigned char *ori_img, const int &width, const int &height, unsigned char *res_img)
{
        int i = 0, j = 0;

        if (ori_img == 0) return;

        i = 0; j = 0;

        for (j = 0; j<height; j++)
        {
            for (i = 0; i<width; i++)
            {
                if (ori_img[j*width+i]<128) res_img[j*width+i] = 0;
                else res_img[j*width+i] = 255;
            }
        }

        return;
}
```

NO comments!!!

Coding Style Example

- Header File

```
#ifndef BINARIZE_H_
#define BINARIZE_H_

void binarizeImg(unsigned char *ori_img, const int &width, const int &height, unsigned char *res_img)

#endif
```

Final Version

Coding Style Example

- Source File

```
#include "binarize.h“

//********************binarizeImg********************//
//INPUT: original image, image width, image height                //
//OUTPUT: result image                                            //
//FUNCTION: binarize an image by using threshold = 128            //
//***************************************************//

void binarizeImg(unsigned char *ori_img, const int &width,
                 const int &height, unsigned char *res_img)
{
    int i = 0, j = 0;   //initialize iterators

    if (ori_img == 0) return;  //immediately return if invalid input

    i = 0; j = 0;

    for (j = 0; j<height; j++)
    {
        for (i = 0; i<width; i++)
        {
            //compare with the threshold
            if (ori_img[j*width+i]<128)
                res_img[j*width+i] = 0;
            else
                res_img[j*width+i] = 0;
        }
    }

    return;
}
```

Final Version

Coding Style Conclusion

- This is just a simple example but contains several critical fixes
- Good coding style helps improve readability & code safety
- There are a lots of other guides we can learn

Coding Style Conclusion

- Other Rules:
  - Name convention: Variables, Constants, Methods, Namespaces …
  - Files: Include statements, Loop, Conditions, Class, Switch …
  - …
- Google C++ Style Guide
  - https://google-styleguide.googlecode.com/svn/trunk/cppguide.html