

Portable Image Processing Library

1. Introduction

Image processing has been widely applied in many areas such as medical imaging, remote sensing, and robot vision. Unfortunately, equipment involved in image processing are expensive since most of them are customized to specific applications. An advancement in electronics and computer plays the important roles in the development of low-cost high performance personal computers. Thus, it is now feasible to apply these computers in low-cost image processing. However, image processing software is still expensive and written for specific applications. One solution to this problem is to develop the software using an image processing library which includes most image processing algorithms. Hence, software development is simplified to application of existing image processing algorithms to specific need instead of starting from scratch.

1.1 Current Status Overseas

Both research-based image processing libraries [1] and commercially available libraries [2] are existed. The former are distributed free of charge in self-supported form, while the latter are expensive and specific to the development platforms.

1.2 Current Status in Thailand

Currently, image processing has been applied in some areas such as medical imaging, and remote sensing. In industry, image processing has played an important role in visual inspection. In research and development, image processing algorithms have been developed in educational institutions as published in annual Electrical Engineering Conference. However, image processing library development has not yet been published.

Image processing in Thailand has been emphasized on usage rather than software development since this kind of development is time-consuming and requires skilled human resources. Image processing development using an image processing library provides the developers more time to spend on application development. We believe that after the library is distributed to research groups and image processing is well established, local developers will have the potential to develop image processing software locally.

2. Details and Discussion

2.1 Technological Status of the Research

The image processing library follows the principles and algorithms in [3], [4] and [5]. The library is developed under MS-DOS. However, porting to different operating systems can be done with minor modifications in image display and capturing subroutines since portability is one of the major keys during the library development.

The library developed consists of the following subroutines.

IMAGE TRANSFORMS

- Discrete Fourier Transform

char *DFT(char *str,int type)

DFT is the analysis of the input image in frequency domain. The resulting output which is the plot of amplitude versus frequency gives the specific characteristics for each image.

IMAGE ENHANCEMENT

Image enhancement is the processing of image for viewing purposes.

- Image Enhancement

- Image Smoothing

- Image Sharpening

```
char *equaliz(char *str)
```

```
char *local_en(char *str,int type)
```

```
char *average_f(char *str,int Nw,int Nd)
```

```
char *median_f(char *str,int Nw,int Nd)
```

```
char *sharpen(char *str,int type)
```

```
char *averag_mul(char *str)
```

```
int halftone(char *a,char *b,int cn,int N1,int M1,int N2,int M2)
```

```
void linear_histogram(char *a,char *b)
```

IMAGE ENCODING

Image coding is the process of converting the original image to another forms suitable for particular purposes. For example, run length coding can be used for image compression.

- Run Length Coding

```
void run_length(char *file1,char *file2,char option)
```

- Shannon-Fano Coding

```
void shannon_encode(char *a,char *b)
```

```
void shannon_decode(char *a,char *b)
```

- Huffman Coding

```
void huffman_encode(char *file1,char *file2)
```

```
void huffman_decode(char *file1,char *file2)
```

- Quantization

```
void quantiz_decode(char *a,char *b)
```

```
void quantiz_encode(char *a,char *b)
```

IMAGE SEGMENTATION

Image segmentation is the process of identifying the groups in the original image.

●Edge Linking and Boundary Detection

●Thresholding

```
char *Global_T(char *str)
```

```
char *BB_thres(char *str,char type)
```

```
int sobel(char *a,char *b,int N1,int M1,int N2,int M2)
```

```
int Laplace(char *a,char *b,int N1,int M1,int N2,int M2)
```

```
int range(char *a,char *b,int N1,int M1,int N2,int M2)
```

```
void op_mask(char *a,char *b,int d[3][3])
```

```
void threshold(char *a,char *b,int th)
```

```
void threshold_exp(char *a,char *b,int exp)
```

```
void threshold_exp(char *a,char *b,int exp)
```

REPRESENTATION

Representation provides means to represent the original image in another forms.

```
int Analog_pro(char *a,int N1,int M1,ing N2,int M2)
```

```
int Digital_pro(char *a,int N1,int M1,int N2,ing M2)
```

```
void dilate(char *a,char *b)
```

```
void dilate_buff(BYTE *buff,BYTE *buf2,int W,int H)
```

```
void erosion(char *a,char *b)
```

```
void erosion_buff(BYTE *buff,BYTE *buf1,int W,int H)
```

```
int skeleton(char *a,char *b,char *d,int ki,int N1,int M1,int N2,int M2)
```

```
int msd(char *a,char *b,char *c,int ki,int N,int N1,int M1,int N2,int  
M2)
```

```
int chain_code(char *a,char *b,int TMAX,int N1,int M1,int N2,int
    M2,int TYPE)
```

```
int curve_split(char *a,float T,int n)
```

```
void signature(char *a,char *b,int Type)
```

UTILITIES

In order to facilitate image processing development, the following subroutines have been developed.

```
void ViewGif(char *str)
```

```
void ViewImg(char *str)
```

```
void _13Vtextxy(unsigned x,unsigned y,char *msg)
```

```
void _13box(x1,y1,x2,y2,color,paint)
```

```
void _13capture_screen(char *a,WORD X1,WORD Y1,WORD X2,
    WORD Y2)
```

```
void _13getimage(x1,y1,x2,y2,buffer)
```

```
unsigned _13imagesize(x1,y1,x2,y2)
```

```
void _13putimage(unsigned x,unsigned y,BYTE *buffer)
```

```
void _13rotage(x1,y1,x2,y2,ang)
```

```
void _13textxy(unsigned x,unsigned y,char *msg,unsigned char color)
```

```
char *add_CG(char mode)
```

```
void capture_video(char *a,WORD X1,WORD Y1,WORD X2, WORD
    Y2)
```

```
void clsg(void)
```

```
int contract(char *a,char *b,int t,int ND1,int MD1,int N1,int M1,int
    N2,int M2)
```

```
void cutimage(char *a,char *b,int Ns,int Ms,int Ne,int Me)
```

```
int expand(char *a,char *b,int t,int ND1,int MD1,int N1,int M1,int
    N2,int M2)
```

```
void freez()
void initFont(char *a)
void initGDT(void)
void initMODE()
void init_video(WORD WIDTH,WORD HIGH)
void inverst(char *a,char *b)
void move2hi(unsigned long S,unsigned long D,unsigned Length)
void move2lo(unsigned long S,unsigned long D,unsigned Length)
void operate(char *a,char *b,char *c,char *d)
void outthaiXY(int x,int y,int c,BYTE *msg)
void read2buff(BYTE *buff,WORD X1,WORD Y1,WORD X2,WORD
               Y2)
int round(float a)
void set_BW(void)
int showCHN(char *a)
int showIMG(char *a)
void show_SGN(char *a,int Type)
void show_histogram(char *a,BYTE color)
float space(char *a)
void terminate_video()
void unfreez()
void video_display_mode(WORD MODE)
```

2.2 Development Problems

During the first year of the development, the project was emphasized on the study and development of basic image processing algorithms. In the second year, image acquisition, display and retrieval were developed together with the trials on local image processing problems such as defect locating in rubber sheet processing and area-based shrimp sorting.

The main problem encountered is the lack of full-time programmers. It is not easy to acquire programmers that understand both C language and image processing, especially for remote areas like the southern part of Thailand. This problem was partly solved by the part-time employment of senior undergraduate students and graduate students. Thus, the problem of retraining the students were not unavoidable.

2.3 Possibilities of the Use of the Library in Industry

In order to apply the library in industry, the library must be supported. Moreover, the library must be further developed for real-time applications since industry uses mainly in visual inspection.

3. Conclusion

3.1 The Impact of the Research Work to Industry

This research is no direct impact to Thailand's industry as long as labor cost is cheap and available. However, trends in Thailand's labors indicates that the shortage of labors will happen soon. If this situation happens, there will be an investment in visual inspection in order to reduce the labor cost. Visual inspection development using the image processing library developed will be one of inexpensive alternatives. However, industry needs to prepare software engineers or

programmers for this purpose since the number of experienced staff in this area is quite low.

3.2 The Impact of the Research Work to Education

An image processing subject has been offered at Prince of Songkla University for 3 years in a graduate level. The image processing library developed can be used as an effective teaching tool in the subject. For an undergraduate level, there were students worked on this research for 4 consecutive years. These students can apply their image processing experiences in their workplaces.

3.3 Further Development

The next stage of the development should emphasize the following two points.

1. The application of the image processing library in other areas.
2. New image processing algorithms development as well as library updating. This will be motivated by (1).

3.4 Budget Summary

| Budget | Amount (Baht) |
|------------------------|----------------------|
| 1. Equipment cost | 160,000 |
| 2. Expendable supplies | 3,000 |
| 3. Extra manpower | 70,000 |
| 4. Other costs | <u>6,720</u> |
| Total | 239,720 |