

CONTENT

Chapter 3: Fundamentals of Programming

3.1 Basics of C programming

3.2 Data Structures in C

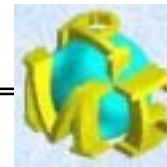
3.3 Data Processing Loops in C

3.4 Use of Image and Graphics Library

3.5 Memory Concept

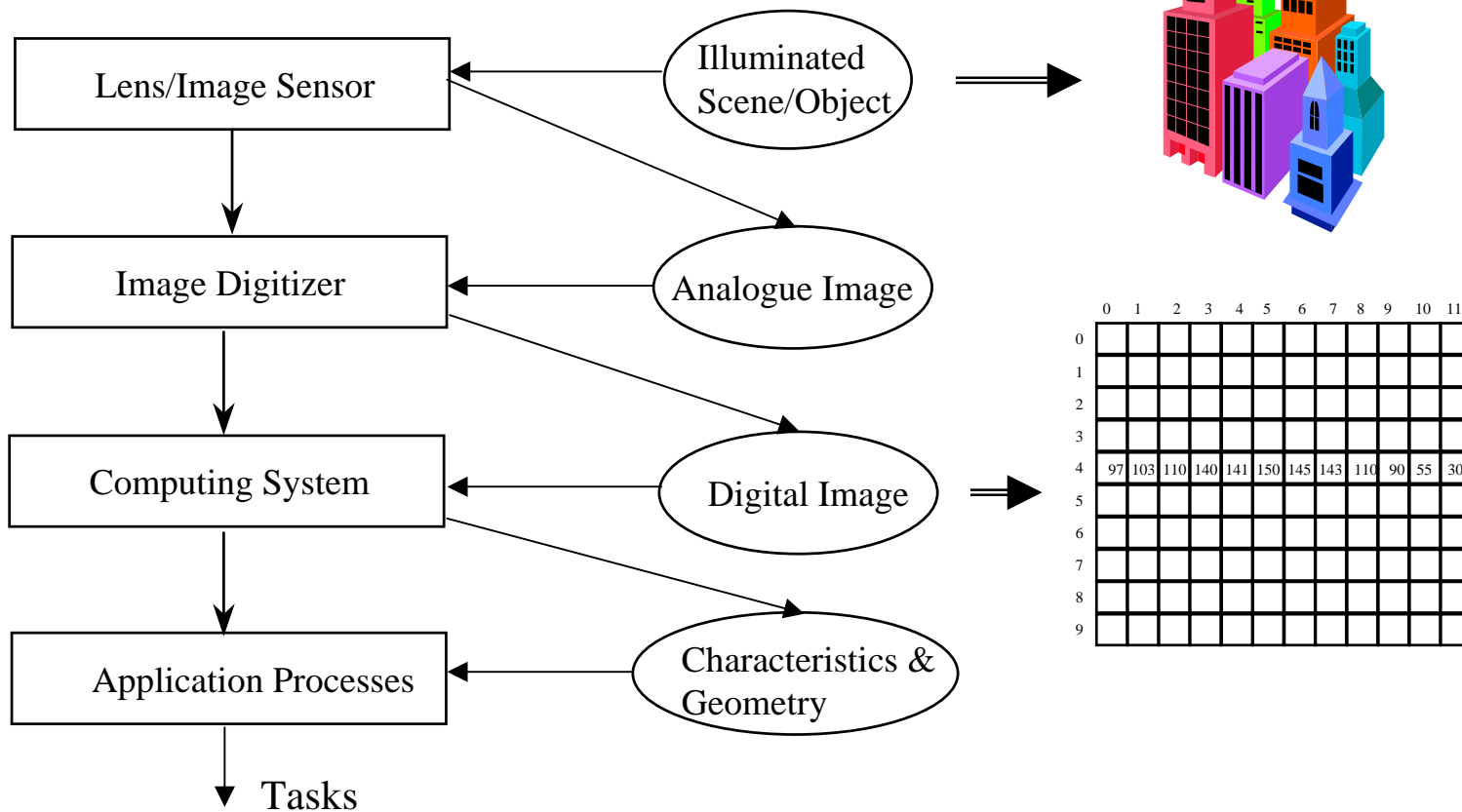
Have Learnt

To Learn



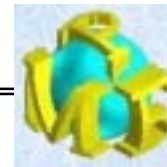
What is a machine vision system ? (A Review)

ANSWER:

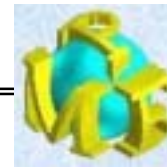
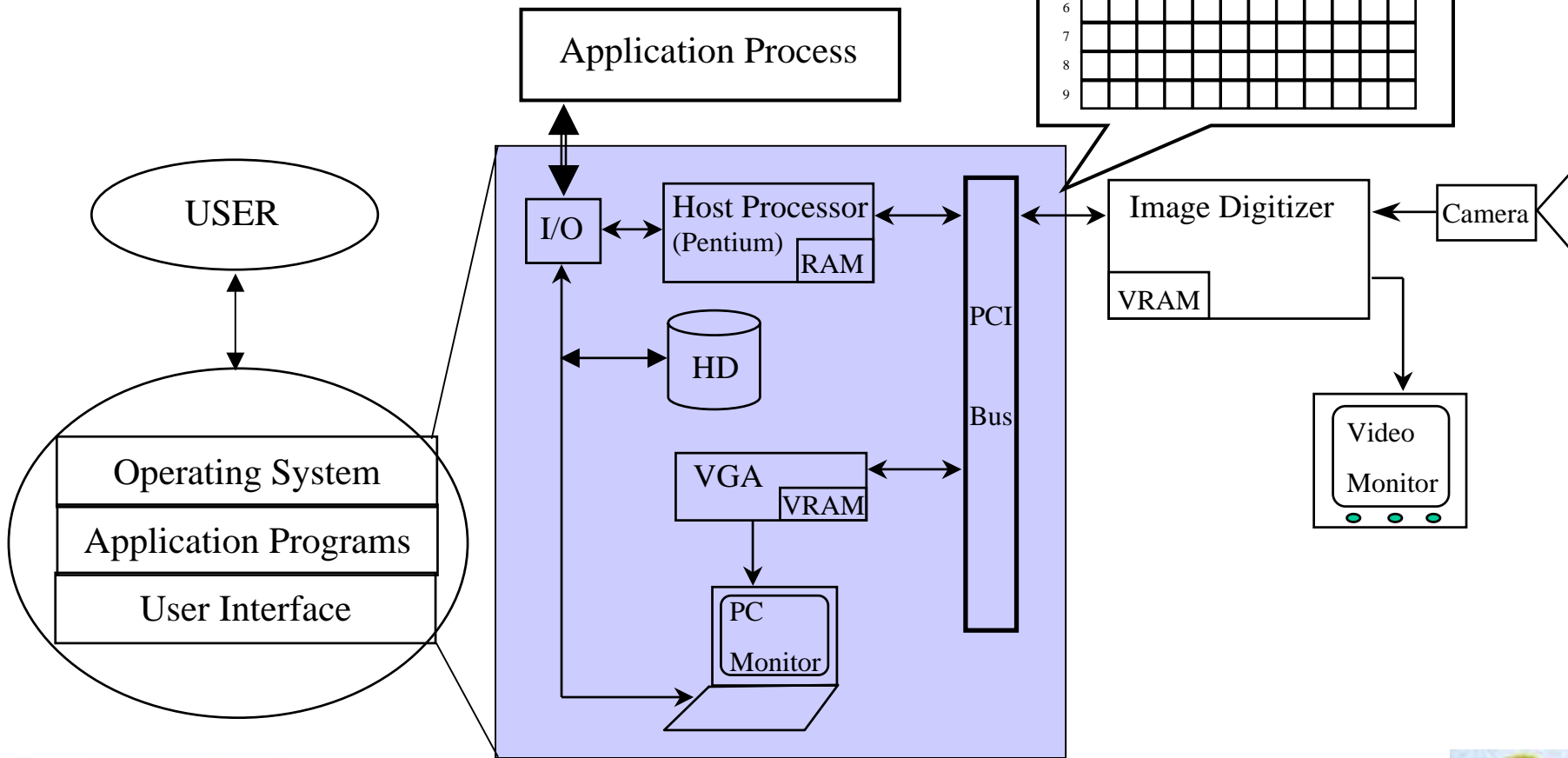


Question:

How to program a machine vision application ?



What is the generic sequence of programming a machine vision application ?



ANSWER:

- Step 1: Initialize image acquisition board (Digitizer).
- Step 2: Capture an image (Do image digitizing).
- Step 3: Transfer the digital image into the computing system for processing.
- Step 4: Process images.
- Step 5: Display/visualize processing results.
- Step 6: Take actions according to results.

NOTE:

Step 4 and Step 6 depend on application. But, you are going to learn some basic techniques for image processing and some basic principles of machine vision.



Question 1:

How to implement Step 1, Step 2 and Step 3 in C ?

ANSWER:

To use functions in image library !

For any image digitizer, the manufacturer always provides the necessary functions for controlling the operations such as:

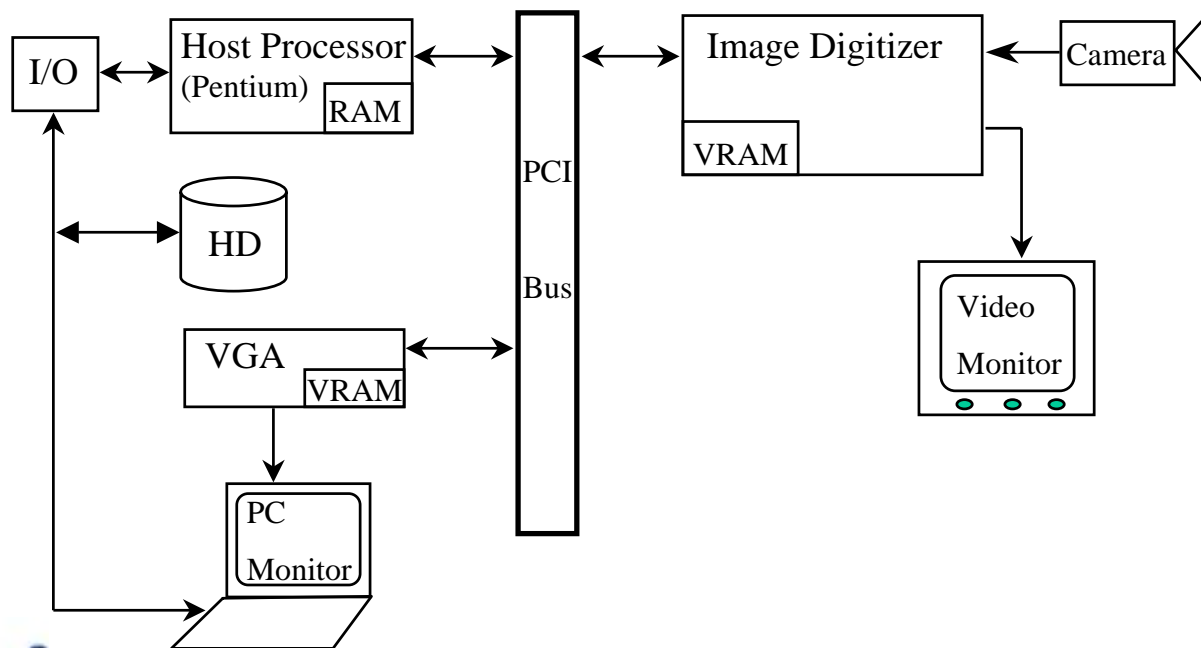
- a) initialization,
- b) image capture, and
- c) image transfer between the image digitizer and the computing system.



What are the most useful functions inside an image library ?

ANSWER:

1. InitBoard();
2. SelectVideoType();
3. Grab();
4. ReadInImage();
5. WriteOutImage();



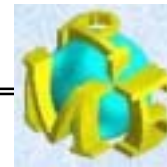
Example

toto.c

```
#include <stdio.h>
#include <math.h>
#include "cx100.h" /* CX100 board from ImageNation Corp. USA */

static unsigned char huge image[256][256];
static int          x0, y0;

void main(int argc, char **argv)
{
    board_on(); /* power on the board */
    init_library(); /* initialize board and set video parameters, etc */
    set_ccir(); /* select PAL video input; set_ntsc() for NTSC */
    grab(); /* digitizing one image */
    get_rectangle(image, x0, y0, 256, 256); /* read in image from (x0,y0) */
    /* do your processing ... */
    put_rectangle(image, x0, y0, 256, 256);
    display;
    exit_library();
    board_off();
}
```



Why is there a need to do image transfers ?

ANSWER:

Inside a machine vision system, there are usually three memories:

- 1) CPU's RAM,
- 2) Video RAM of image digitizer, and
- 3) Video RAM of VGA display card.

* Requirement 1:

Transfer of image from the digitizer's VRAM into CPU's RAM for processing purpose.

* Requirement 2:

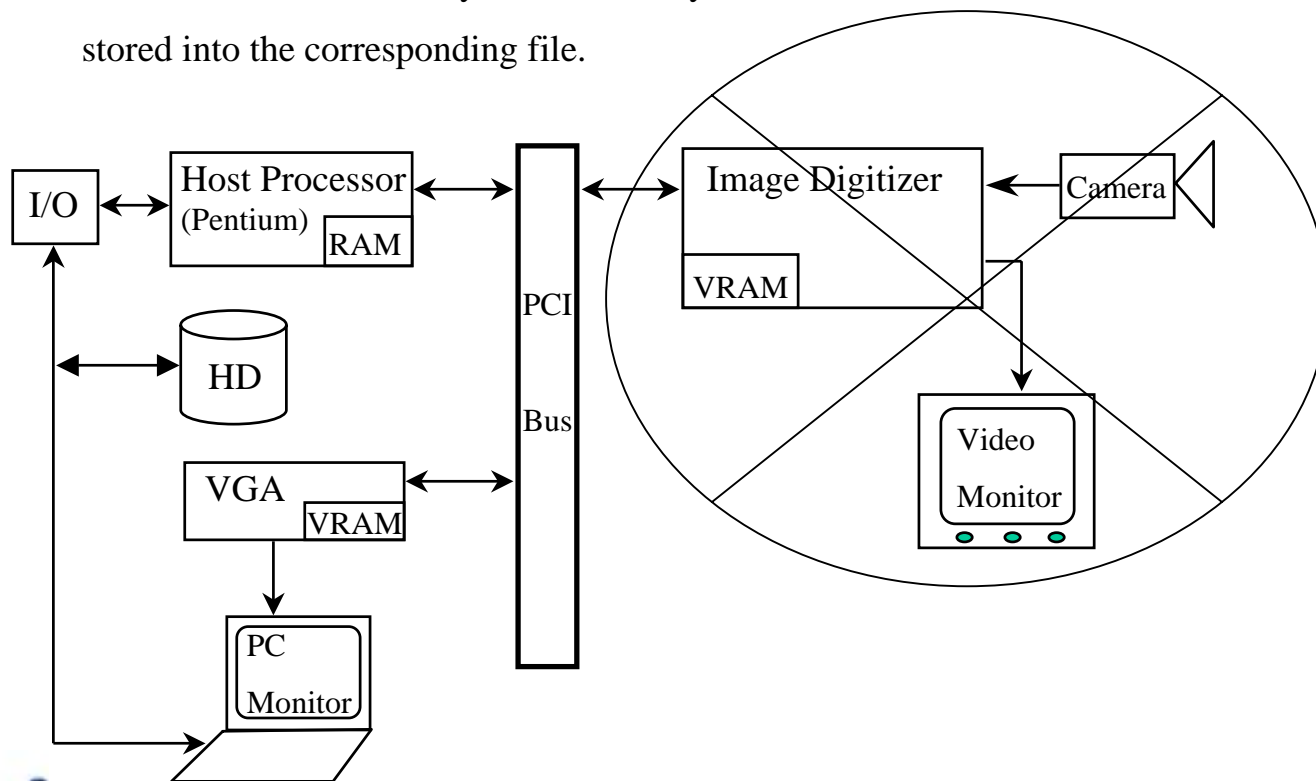
Transfer of image from CPU's RAM into VGA's or digitizer's VRAM for display purpose.



If a computing system does not have any image digitizer, how to get an image into the CPU's RAM ?

ANSWER:

You can read in an image stored in a hard-disk file. This image must have been created by other vision system, and has been stored into the corresponding file.



Example:

toto.c

```
#include <stdio.h>
#include <math.h>

unsigned char huge image[256][256];

static void ReadImageFromFile(char *filename)
{
    FILE *pf;

    pf = fopen(filename, "r");
    fread(image, 1, 256*256, pf);
    fclose(pf);
}

static void WriteImageToFile(char *filename)
{
    FILE *pf;

    pf = fopen(filename, "w");
    fwrite(image, 1, 256*256, pf);
    fclose(pf);
}

main(int argc, char **argv)
{
    ReadImageFromFile("toto.img"); /* read an image stored in toto.img */
    /* do your processing ....*/
    WriteImageToFile("tata.img"); /* write results to tata.img file */
}
```



Question 2:

How to implement Step 5 (Display/visualize results) ?

ANSWER:

1. To develop a graphic user interface, and use functions in graphic library.
2. There are two popular Window systems:
 - a) Microsoft Windows, and
 - b) X-Window (free, developed by MIT).
3. Under Microsoft Windows, the programming languages are:
 - a) Visual C/C++
 - b) Borland C/C++
4. Under X-Window environment, the programming language is (standard) C.
 - a) ANSI C/C++
 - b) GNU C/C++
 - c) or other standard C/C++



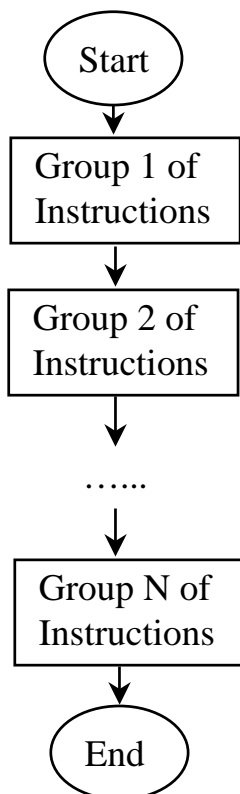
How to develop a graphic user interface (GUI) ?

ANSWER:

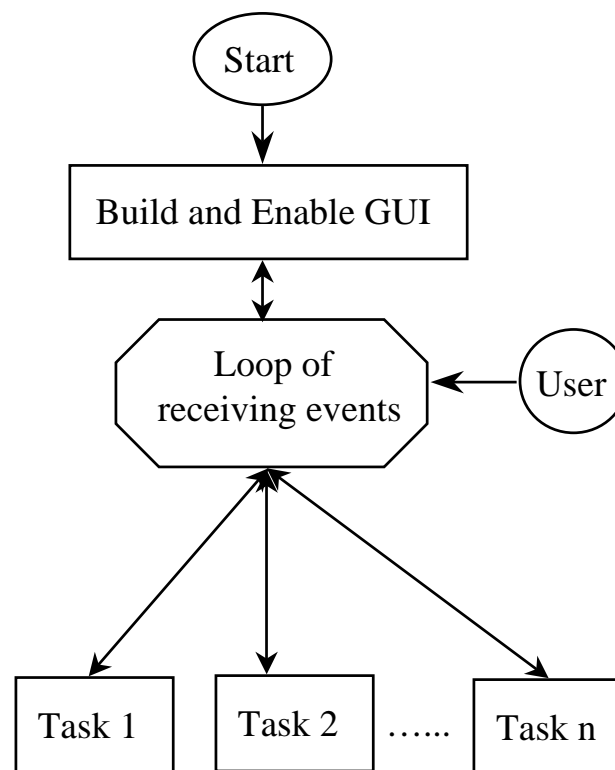
1. First to understand that a graphic user interface is in general an Event-Driven application.
2. Secondly, you can follow the sample programs given by the manufacturer, that will show you how to use the functions in both the image and graphic libraries.



A. Sequential Program:



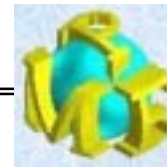
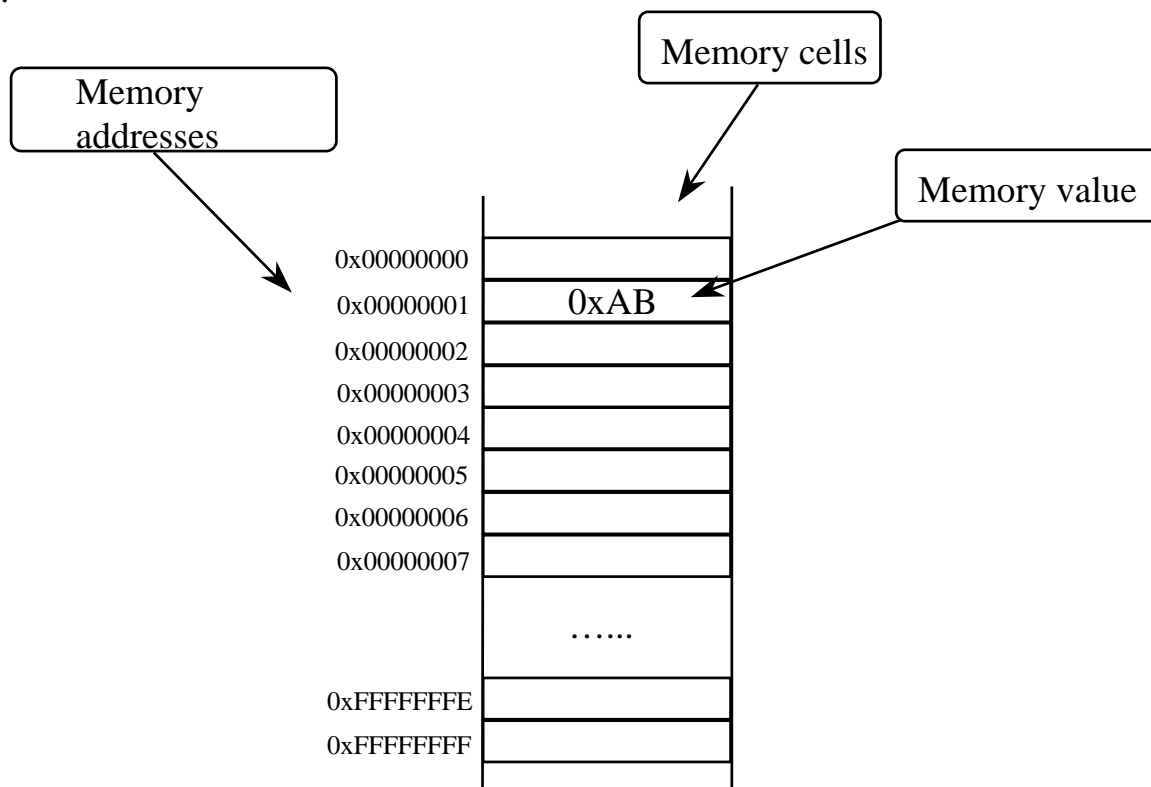
B. Event-Driven Program:



We know that memories are important in a machine vision system. How is a memory RAM arranged inside a machine vision system ?

ANSWER:

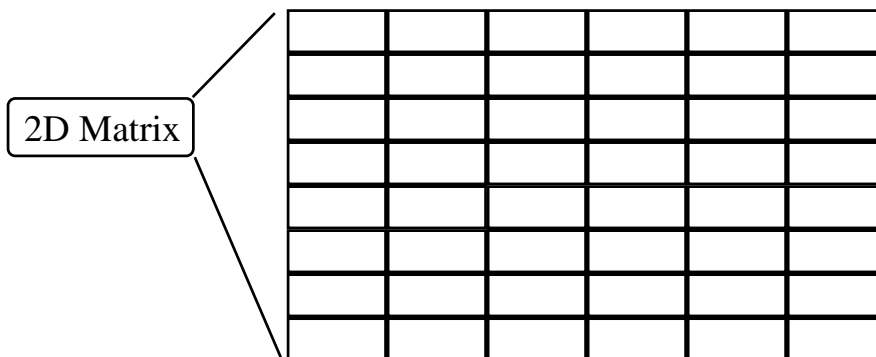
It is arranged as a list of memory cells each of which has an unique memory address. One memory cell is one Byte (8 bits). Therefore, a memory is a one-dimensional vector of memory cells.



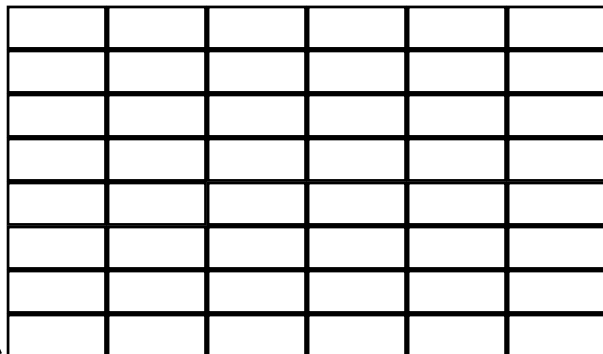
A list of memory cells is arranged in the form of one dimensional vector. How to store a two-dimensional image matrix with one dimensional memory cells ?

ANSWER:

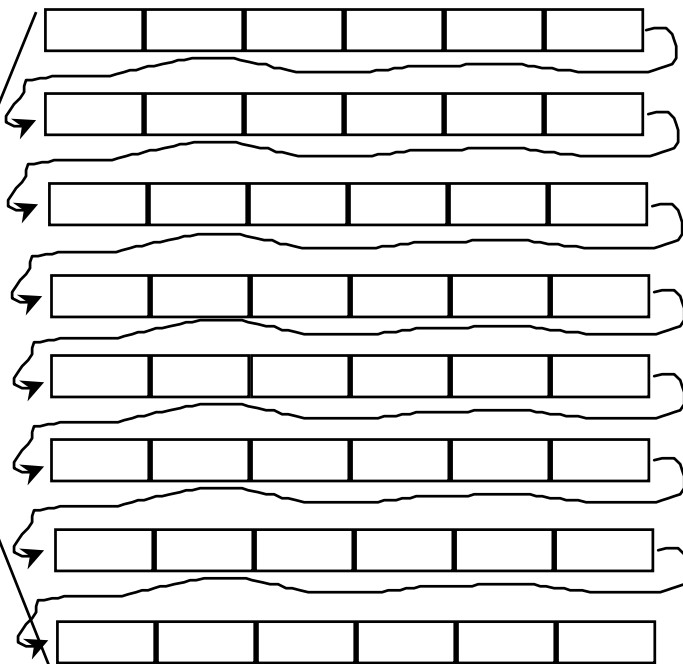
Mathematically, a two-dimensional matrix can be considered as a one-dimensional vector that is the result of the linked rows of the matrix.



2D Matrix



Linked
row vectors



If:

(r, c) are the indices
of a matrix element;

k is the index of the
corresponding vector
element;

resx is the size of a
row vector;

Then:

$$k = r * \text{resx} + c.$$



Example

Operation:

Copy the content of image1 into image2.

toto.c

```
#include <stdio.h>

static unsigned char image1[512*256];
static unsigned char image2[512*256];
static int resx, resy;

static ReadImageFromFile(char *filename)
{
    /* same as the previous example */
}

main(int argc, char **argv)
{
    int row, column;

    resx = 512;
    resy = 256;

    for (row = 0; row < 256; row++)
    {
        for (column = 0; column < 512; column++)
        {
            image2[row*resx+column] = image1[row*resx+column];
        }
    }
}
```



SUMMARY

1. It is a convenient way to use functions of image library to do operations such as: a) initialization of image digitizer, b) capture of image, and c) image transfers, etc.
2. To display processing results, one needs to build graphic user interface, and to use functions of graphic library to visualize results. The easiest way to learn this is to follow samples provided by the manufacturer of image digitizer.
3. There are in general three memories in a machine vision system:
 - * CPU's RAM.
 - * Video RAM of Image Digitizer.
 - * Video RAM of VGA Display Card.
4. Usually, images are being processed in CPU's RAM, and being displayed through Video RAM of either image digitizer or VGA display card.
5. Inside a computing system, a memory is arranged as a vector of memory cells. Each cell is just one Byte (8 bits). A 2D matrix can be considered as a vector that is the result of the linked rows of the matrix.

