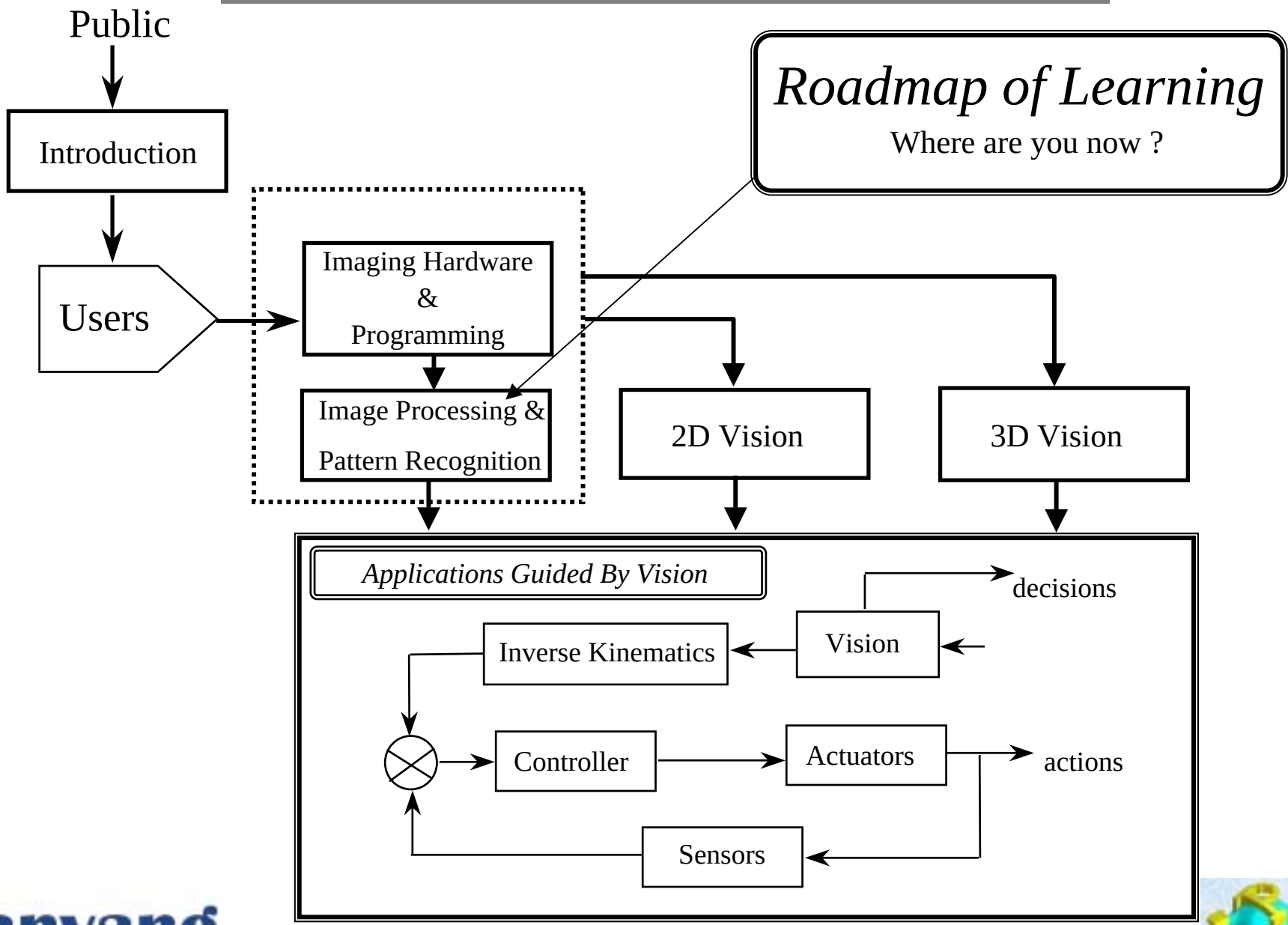
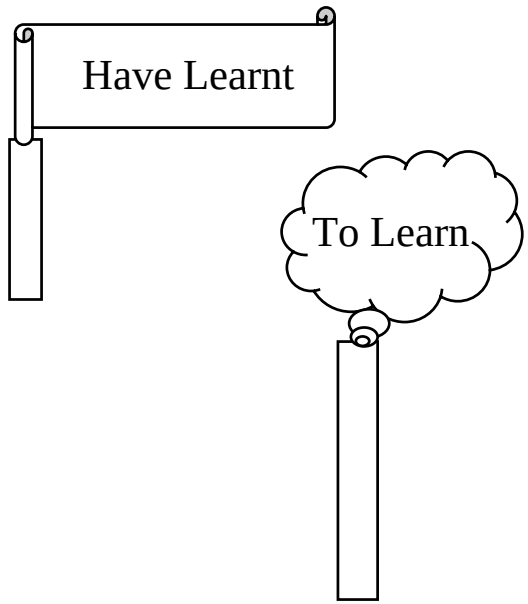
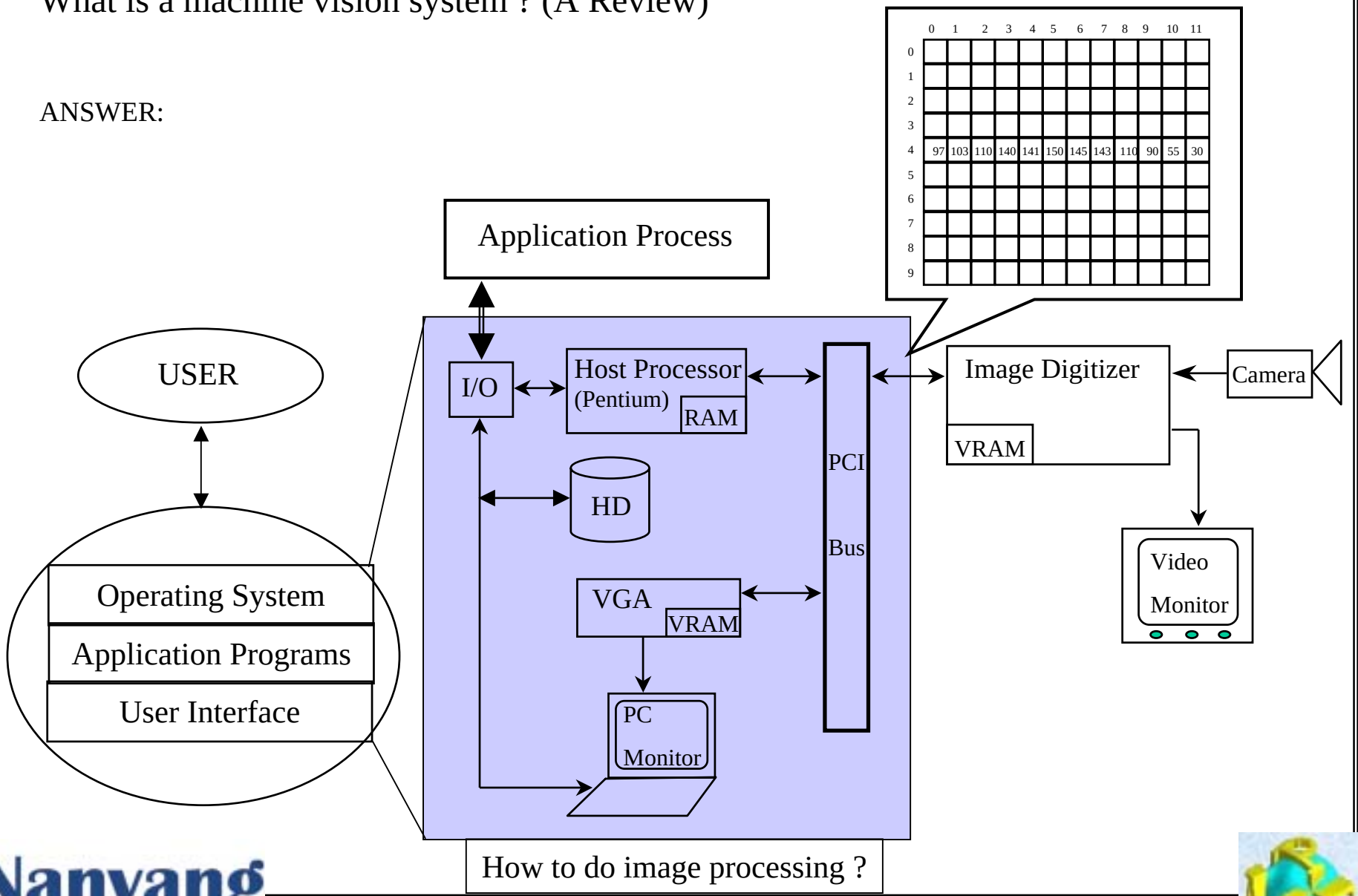


CONTENT



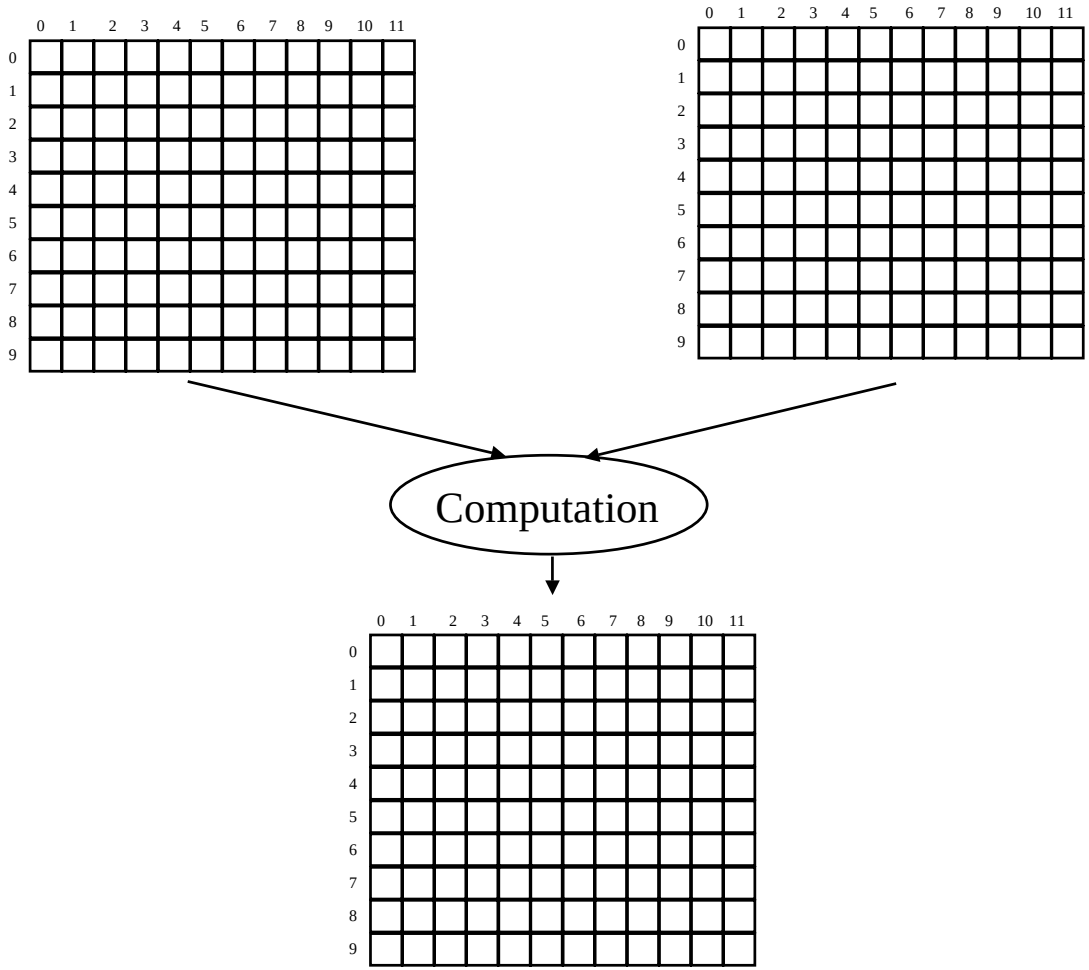
What is a machine vision system ? (A Review)

ANSWER:



What is a Frame-based image processing technique ?

ANSWER: It refers to the computations carried out on at least two images of the same size.



Frame-based Image Processing Techniques

1. Image Addition:

Input images : $I^{in1} = \{g_{i,j}^{in1} \mid 0 \le i < n,\ 0 \le j < m\}$

$I^{in2} = \{g_{i,j}^{in2} \mid 0 \le i < n,\ 0 \le j < m\}$

Output image : $I^{out} = \{g_{i,j}^{out} \mid 0 \le i < n,\ 0 \le j < m\}$

Operation :

$$g_{i,j}^{out} = a \bullet g_{i,j}^{in1} + (1-a) \bullet g_{i,j}^{in2}.$$

When a = 0.5, this computes the average of the two images.

```
#include <stdio.h>

unsigned char  image_in1[512*512], image_in2[512*512] ;
unsigned char  image_out[512*512] ;

static void  ReadInImage()
{
}

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

        a = 0.5 ;
        ReadInImage() ;

        for (row = 0; row < 512; row++)
        {
                for (column = 0 ; column < 512; column++)
                        image_out[row*512+column] = (unsigned char)
                                (a*image_in1[row*512+column] +
                                 (1-a)*image_in2[row*512+column]) ;
         }
}
```


Example

Image 1 + Image 2 = Output Image

Frame-based Image Processing Techniques

2. Image Subtraction:

Input images : $I^{in1} = \{g_{i,j}^{in1} \mid 0 \le i < n,\ 0 \le j < m\}$

$I^{in2} = \{g_{i,j}^{in2} \mid 0 \le i < n,\ 0 \le j < m\}$

Output image : $I^{out} = \{g_{i,j}^{out} \mid 0 \le i < n,\ 0 \le j < m\}$

Operation :

$$g_{i,j}^{out} = | g_{i,j}^{in1} - g_{i,j}^{in2} |.$$

```
#include <stdio.h>

unsigned char  image_in1[512*512], image_in2[512*512] ;
unsigned char  image_out[512*512] ;

static void  ReadInImage()
{
}

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

        a = 0.5 ;
        ReadInImage() ;

        for (row = 0; row < 512; row++)
        {
                for (column = 0 ; column < 512; column++)
                        image_out[row*512+column] = (unsigned char)
                                abs(image_in1[row*512+column] -
                                 image_in2[row*512+column]) ;
         }
}
```


Example

Image at t1 - Image at t2 = Output Image

Example

	0	1	2	3	4	5	6	7	8	9	10	11
0	30	30	30	30	30	30	30	30	30	30	30	30
1	30	30	30	30	30	30	30	30	30	30	30	30
2	30	30	66	66	66	66	66	66	66	66	30	30
3	30	30	66	66	66	66	66	66	66	66	30	30
4	30	30	66	66	66	66	66	66	66	66	30	30
5	30	30	30	30	99	99	99	99	30	30	30	30
6	30	30	30	30	99	99	99	99	30	30	30	30
7	30	30	30	30	99	99	99	99	30	30	30	30
8	30	30	30	30	99	99	99	99	30	30	30	30
9	30	30	30	30	30	30	30	30	30	30	30	30

-

	0	1	2	3	4	5	6	7	8	9	10	11
0	30	30	30	30	30	30	30	30	30	30	30	30
1	30	30	30	30	30	30	30	30	30	30	30	30
2	30	30	30	66	66	66	66	66	66	66	66	30
3	30	30	30	66	66	66	66	66	66	66	66	30
4	30	30	30	66	66	66	66	66	66	66	66	30
5	30	30	30	30	30	99	99	99	99	30	30	30
6	30	30	30	30	30	99	99	99	99	30	30	30
7	30	30	30	30	30	99	99	99	99	30	30	30
8	30	30	30	30	30	99	99	99	99	30	30	30
9	30	30	30	30	30	30	30	30	30	30	30	30

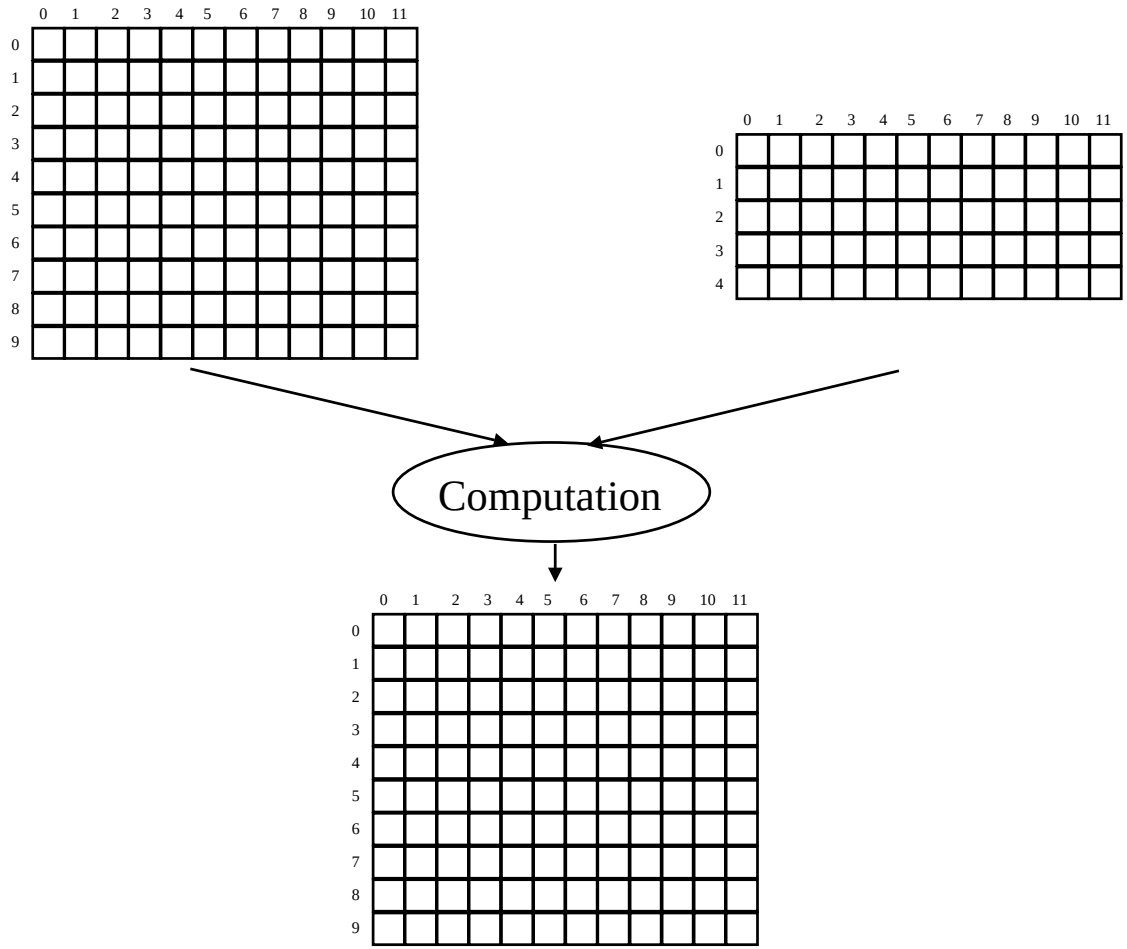
=

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	36	0	0	0	0	0	0	0	36	0
3	0	0	36	0	0	0	0	0	0	0	36	0
4	0	0	36	0	0	0	0	0	0	0	36	0
5	0	0	0	0	69	0	0	0	69	0	0	0
6	0	0	0	0	69	0	0	0	69	0	0	0
7	0	0	0	0	69	0	0	0	69	0	0	0
8	0	0	0	0	69	0	0	0	69	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0

What is a Area-based image processing technique ?

ANSWER:

It refers to the computations carried out on at least two images of different sizes.



Area-based Image Processing Techniques

1. Convolution:

Input images : $I^{in1} = \{g_{i,j}^{in1} \mid 0 \le i < n,\ 0 \le j < m\}$

$I^{in2} = \{g_{i,j}^{in2} \mid 0 \le i < n_1,\ 0 \le j < m_1\}$

Condition : $n_1 < n$ and $m_1 < m$.

Output image : $I^{out} = I^{in1} \otimes I^{in2} = \{g_{i,j}^{out} \mid 0 \le i < n,\ 0 \le j < m\}$

	0	1	2	3	4	5	6	7	8	9	10	11
0	30	30	30	30	30	30	30	30	30	30	30	30
1	30	30	30	30	30	30	30	30	30	30	30	30
2	30	30	66	66	66	66	66	66	66	66	30	30
3	30	30	66	66	66	66	66	66	66	66	30	30
4	30	30	66	66	66	66	66	66	66	66	30	30
5	30	30	30	30	99	99	99	99	30	30	30	30
6	30	30	30	30	99	99	99	99	30	30	30	30
7	30	30	30	30	99	99	99	99	30	30	30	30
8	30	30	30	30	99	99	99	99	30	30	30	30
9	30	30	30	30	30	30	30	30	30	30	30	30

Operation :

$$\begin{cases} Sub(I^{in1}, i, j) = \{(g_{s,t}^{sub} = g_{i-n_1/2+s,\, j-m_1/2+t}^{in1}) \mid 0 \le s < n_1, 0 \le t < m_1\} \\ g_{i,j}^{out} = \sum_{s=0}^{n_1} \sum_{t=0}^{m_1} (g_{s,t}^{sub} \bullet g_{s,t}^{in2}). \end{cases}$$

Illustration

	0	1	2	3	4	5	6	7	8	9	10	11
0	30	30	30	30	30	30	30	30	30	30	30	30
1	30	30	30	30	30	30	30	30	30	30	30	30
2	30	30	66	66	66	66	66	66	66	66	30	30
3	30	30	66	66	66	66	66	66	66	66	30	30
4	30	30	66	66	66	66	66	66	66	66	30	30
5	30	30	30	30	99	99	99	99	30	30	30	30
6	30	30	30	30	99	99	99	99	30	30	30	30
7	30	30	30	30	99	99	99	99	30	30	30	30
8	30	30	30	30	99	99	99	99	30	30	30	30
9	30	30	30	30	30	30	30	30	30	30	30	30

$*$

-1	0	+1
-1	0	+1
-1	0	+1

What's the pixel value at the location (4,2) of the output image ?

Step 1: At the location (4, 2), get the subimage:

30	66	66
30	66	66
30	30	30

Step 2: Compute the convolution between the sub-image and Image 2:

30	66	66
30	66	66
30	30	30

$*$

-1	0	+1
-1	0	+1
-1	0	+1

$=$

-30	0	66
-30	0	66
-30	0	30

Step 3: Compute the sum of the pixel value in the resulting matrix . This sum is the pixel value of the output image at the pixel location (4,2):

sum = 72

Output Image

	0	1	2	3	4	5	6	7	8	9	10	11
0												
1												
2												
3												
4			72									
5												
6												
7												
8												
9												

Sample Program

```
unsigned char  image1[512*512];
unsigned char  image2[3*3] ;
unsigned char  image_out[512*512] ;

static void  DoConvolution()
{
      int   i, j, s, t ;

      for (i = 3/2; i < 512-3/2; i++)
      {
            for (j = 3/2 ; j < 512-3/2; j++)
            {
                  image_out[i*512+j] = 0 ;
                  for (s = 0 ; s < 3; s++)
                  {
                        for (t = 0; t < 3; t++)
                        {
                              image_out[i*512+j] = image_out[i*512+j] +
                                    (image1[(i-3/2+s)*512+j-3/2+t] * image2[s*3+t]) ;
                        }
                  }
            }
      }
}
```

	0	1	2	3	4	5	6	7	8	9	10	11
0	30	30	30	30	30	30	30	30	30	30	30	30
1	30	30	30	30	30	30	30	30	30	30	30	30
2	30	30	66	66	66	66	66	66	66	66	30	30
3	30	30	66	66	66	66	66	66	66	66	30	30
4	30	30	66	66	66	66	66	66	66	66	30	30
5	30	30	30	30	99	99	99	99	30	30	30	30
6	30	30	30	30	99	99	99	99	30	30	30	30
7	30	30	30	30	99	99	99	99	30	30	30	30
8	30	30	30	30	99	99	99	99	30	30	30	30
9	30	30	30	30	30	30	30	30	30	30	30	30

Area-based Image Processing Techniques

2. Template (or Pattern) Matching:

Input images : $I^{in1} = \{g_{i,j}^{in1} \mid 0 \le i < n,\ 0 \le j < m\}$

$I^{in2} = \{g_{i,j}^{in2} \mid 0 \le i < n_1,\ 0 \le j < m_1\}$

Condition : $n_1 < n$ and $m_1 < m$.

Output image : $I^{out} = I^{in1} \oplus (-I^{in2}) = \{g_{i,j}^{out} \mid 0 \le i < n,\ 0 \le j < m\}$

	0	1	2	3	4	5	6	7	8	9	10	11
0	30	30	30	30	30	30	30	30	30	30	30	30
1	30	30	30	30	30	30	30	30	30	30	30	30
2	30	30	66	66	66	66	66	66	66	66	30	30
3	30	30	66	66	66	66	66	66	66	66	30	30
4	30	30	66	66	66	66	66	66	66	66	30	30
5	30	30	30	30	99	99	99	99	30	30	30	30
6	30	30	30	30	99	99	99	99	30	30	30	30
7	30	30	30	30	99	99	99	99	30	30	30	30
8	30	30	30	30	99	99	99	99	30	30	30	30
9	30	30	30	30	30	30	30	30	30	30	30	30

Operation :

$$\begin{cases} Sub(I^{in1}, i, j) = \{(g_{s,t}^{sub} = g_{i-n_1/2+s,\, j-m_1/2+t}^{in1}) \mid 0 \le s < n_1, 0 \le t < m_1\} \\ g_{i,j}^{out} = \dfrac{1}{n_1 \bullet m_1} \bullet \sum_{s=0}^{n_1} \sum_{t=0}^{m_1} | g_{s,t}^{sub} - g_{s,t}^{in2} |. \end{cases}$$

Illustration

	0	1	2	3	4	5	6	7	8	9	10	11
0	30	30	30	30	30	30	30	30	30	30	30	30
1	30	30	30	30	30	30	30	30	30	30	30	30
2	30	30	66	66	66	66	66	66	66	66	30	30
3	30	30	66	66	66	66	66	66	66	66	30	30
4	30	30	66	66	66	66	66	66	66	66	30	30
5	30	30	30	30	99	99	99	99	30	30	30	30
6	30	30	30	30	99	99	99	99	30	30	30	30
7	30	30	30	30	99	99	99	99	30	30	30	30
8	30	30	30	30	99	99	99	99	30	30	30	30
9	30	30	30	30	30	30	30	30	30	30	30	30

−

30	30	30	30	30
30	30	66	66	66
30	30	66	66	66

What’s the pixel value at the location (4,2) of the output image ?

Step 1: At the location (4, 2), get the subimage:

30	30	66	66	66
30	30	66	66	66
30	30	30	30	99

Step 2: Compute "Image Subtraction" between the sub-image and Image 2:

30	30	66	66	66
30	30	66	66	66
30	30	30	30	99

-

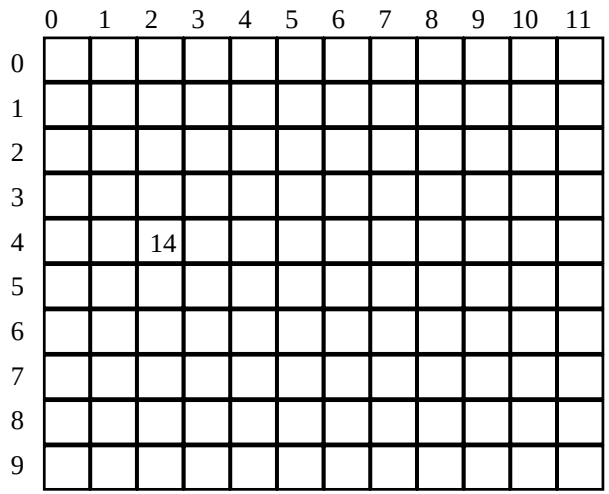
30	30	30	30	30
30	30	66	66	66
30	30	66	66	66

=

0	0	36	36	36
0	0	0	0	0
0	0	36	36	33

Step 3: Compute the normalised sum of the values in the resulting image. This sum is the pixel value of the output image at the pixel location (4,2):

sum = 213/15=14



Program

```
unsigned char  image1[512*512] ;
unsigned char  image2[30*30] ;
unsigned char  image_out[512*512] ;

static void  ComputeTemplateMatch()
{
      int   i, j, s, t, v ;
      for (i = 30/2; i < 512-30/2; i++)
      {
            for (j = 30/2 ; j < 512-30/2; j++)
            {
                  v = 0 ;
                  for (s = 0 ; s < 30; s++)
                  {
                        for (t = 0; t < 30; t++)
                        {
                            v = v +
                              abs(image1[(i-30/2+s)*512+j-30/2+t]-image2[s*30+t]) ;
                        }
                  }
                  image_out[i*512+j]=(unsigned char) v/(30*30) ;
            }
      }
}
```

	0	1	2	3	4	5	6	7	8	9	10	11
0	30	30	30	30	30	30	30	30	30	30	30	30
1	30	30	30	30	30	30	30	30	30	30	30	30
2	30	30	66	66	66	66	66	66	66	66	30	30
3	30	30	66	66	66	66	66	66	66	66	30	30
4	30	30	66	66	66	66	66	66	66	66	30	30
5	30	30	30	30	99	99	99	99	30	30	30	30
6	30	30	30	30	99	99	99	99	30	30	30	30
7	30	30	30	30	99	99	99	99	30	30	30	30
8	30	30	30	30	99	99	99	99	30	30	30	30
9	30	30	30	30	30	30	30	30	30	30	30	30

Area-based Image Processing Techniques

3. Filtering Operations:

Input images : $I^{in1} = \{g_{i,j}^{in1} \mid 0 \le i < n,\ 0 \le j < m\}$

$I^{in2} = \{(g_{i,j}^{in2} \in \{0,1\}) \mid 0 \le i < n_1,\ 0 \le j < m_1\}$

Condition : $n_1 < n$ and $m_1 < m$.

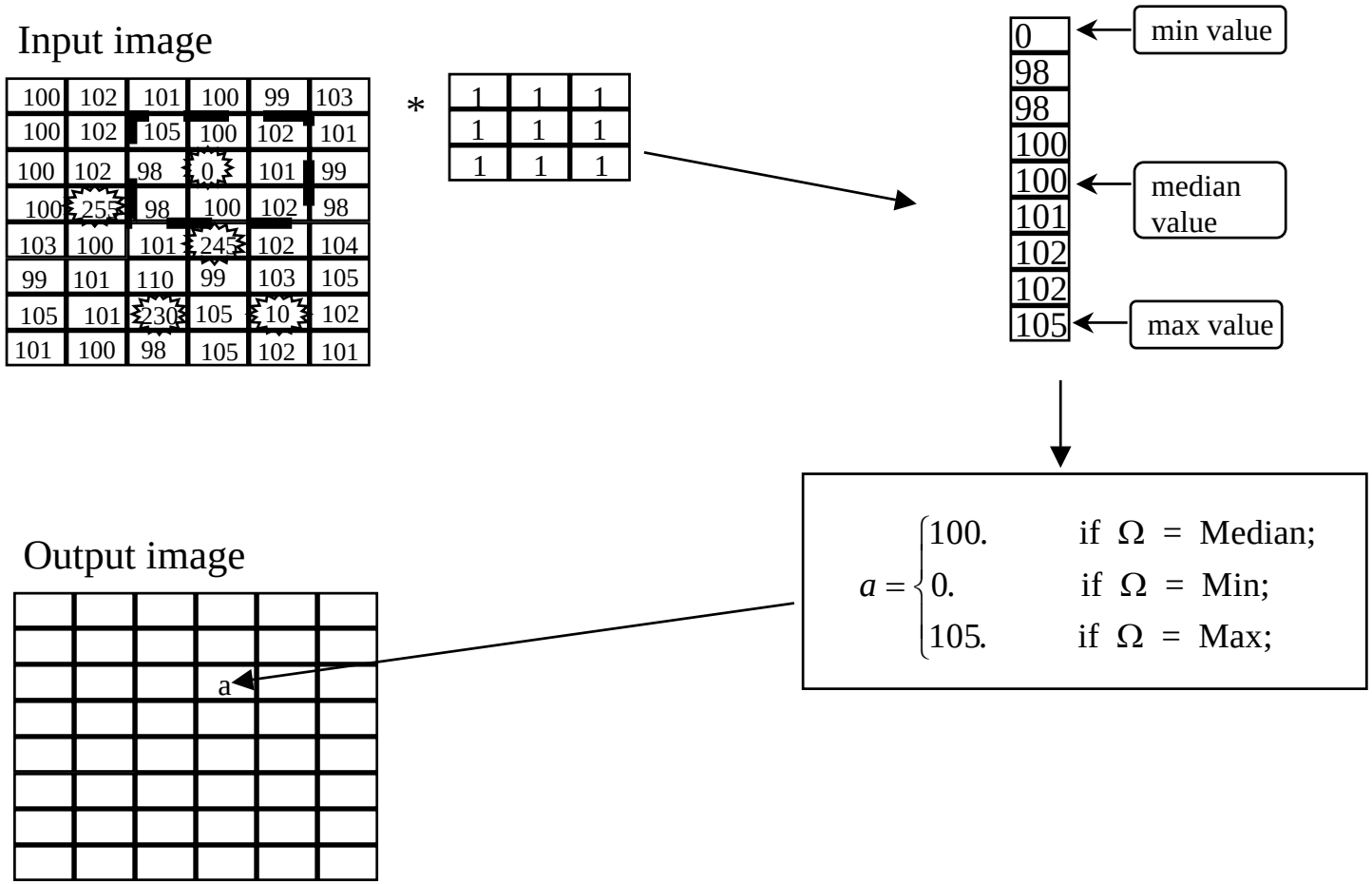
Output image : $I^{out} = \Omega(I^{in1} \otimes I^{in2}) = \{g_{i,j}^{out} \mid 0 \le i < n,\ 0 \le j < m\}$

	0	1	2	3	4	5	6	7	8	9	10	11
0	30	30	30	30	30	30	30	30	30	30	30	30
1	30	30	30	30	30	30	30	30	30	30	30	30
2	30	30	66	66	66	66	66	66	66	66	30	30
3	30	30	66	66	66	66	66	66	66	66	30	30
4	30	30	66	66	66	66	66	66	66	66	30	30
5	30	30	30	30	99	99	99	99	30	30	30	30
6	30	30	30	30	99	99	99	99	30	30	30	30
7	30	30	30	30	99	99	99	99	30	30	30	30
8	30	30	30	30	99	99	99	99	30	30	30	30
9	30	30	30	30	30	30	30	30	30	30	30	30

Operation :

$$\begin{cases} Sub(I^{in1}, i, j) = \{(g_{s,t}^{sub} = g_{i-n_1/2-s,\, j-m_1/2-t}^{in1}) \mid 0 \le s < n_1, 0 \le t < m_1\} \\ g_{i,j}^{out} = \Omega\{(g_{s,t}^{sub} * g_{s,t}^{in2}) \mid 0 \le s < n_1,\ 0 \le t < m_1\} \\ \Omega = \{\text{Median, Max, Min}\} \end{cases}$$

Illustration



SUMMARY

1. Frame-based processing techniques deal with computations carried out on at least two images of the same size. The common techniques are:

   * Image addition
   * Image subtraction

2. Area-based processing techniques deal with computations carried out on at least two images of different sizes. The common techniques are:

   * Image convolution
   * Template matching
   * Filtering operations
     (Median, Min, Max)