Digital Image Processing

Malay S. Bhatt
Department of Computer Engineering
Faculty of Technology
Dharmsinh Desai University
Nadiad

Arithmetic Operations

- Addition

  For any two digital images, say f and g, the addition would yield the output image z with the arithmetic addition of the pixel values of f and g at the (x, y) co-ordinates respectively.

- Subtraction

  For any two digital images, say f and g, the addition would yield the output image z with the arithmetic addition of the pixel values of f and g at the (x, y) co-ordinates respectively.

- Multiplication

  For any two digital images, say f and g, the addition would yield the output image z with the arithmetic addition of the pixel values of f and g at the (x, y) co-ordinates respectively.

- Division

  For any two digital images, say f and g, the addition would yield the output image z with the arithmetic addition of the pixel values of f and g at the (x, y) co-ordinates respectively.

Addition Operation

IMAGE 1

0	0	0	0
0	100	100	0
0	100	100	0
0	0	0	0

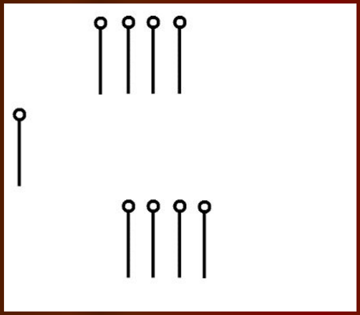
IMAGE 2

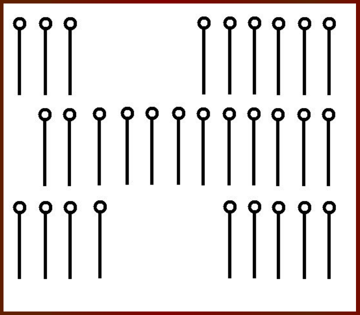
100	0	0	0
0	100	0	0
0	0	100	0
0	0	0	100

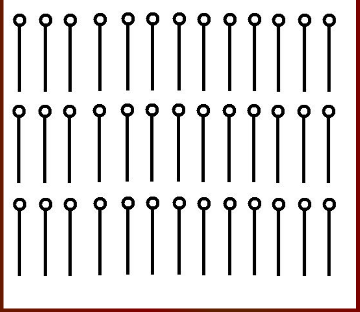
IMAGE 1 + IMAGE 2

100	0	0	0
0	200	100	0
0	100	200	0
0	0	0	100

Arithmetic Operation (Addition)







```
clear;
clc;
img1=imread('D:\Imageprocessing\IP_sttp\sticks3.bmp');
img2=imread('D:\ImageProcessing\IP_sttp\sticks2.bmp');
img1=double(img1);
img2=double(img2);
imshow(img1);
figure,imshow(img2);
final=img1 + img2;
final=mat2gray(final);
imwrite(final,'D:\Image Processing\IP_sttp\sticks.bmp');
figure, imshow (final);
```

Subtraction Operation

IMAGE 1

0	0	0	0
0	100	100	0
0	100	100	0
0	0	0	0

IMAGE 2

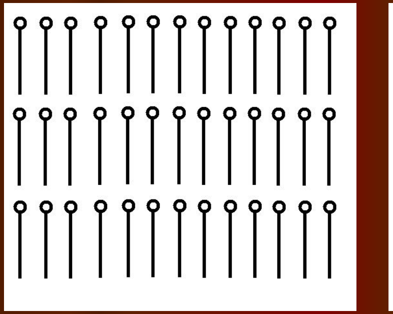
100	0	0	0
0	100	0	0
0	0	100	0
0	0	0	100

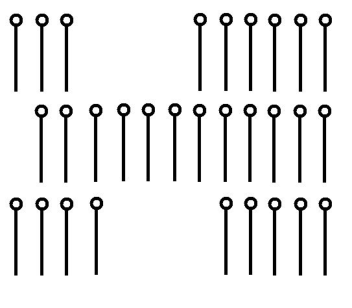
IMAGE 1 - IMAGE 2

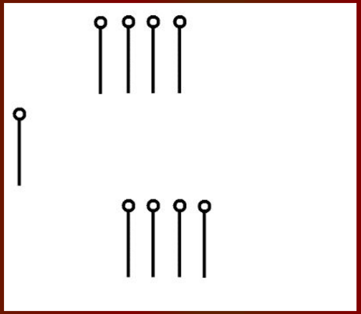
0	0	0	0
0	0	100	0
0	100	0	0
0	0	0	0

Image subtraction is a basic tool in medical imaging, where it is used to remove static background information

Arithmetic Operation (Subtraction)







```
clear;
clc;
img1=imread('D:\Image processing\IP_sttp\sticks.bmp');
img2=imread('D:\Image processing\IP_sttp\sticks2.bmp');
img1=double(img1);
img2=double(img2);
imshow(img1);
figure,imshow(img2);
final=img1 -img2;
final=mat2gray(final);
imwrite(final,'D:\Image Processing\IP_sttp\sticks3.bmp');
figure, imshow (final);
```

Multiplication Operation

IMAGE 1

0	0	0	0
0	100	100	0
0	100	100	0
0	0	0	0

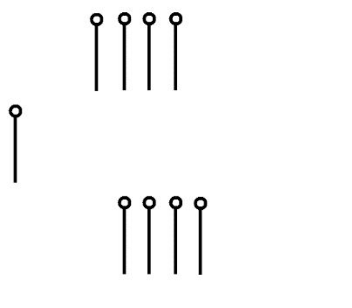
IMAGE 2

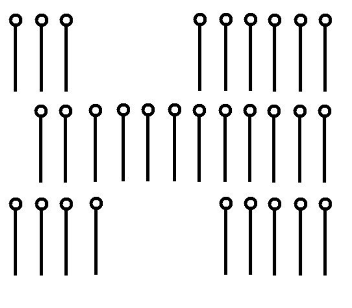
100	0	0	0
0	100	0	0
0	0	100	0
0	0	0	100

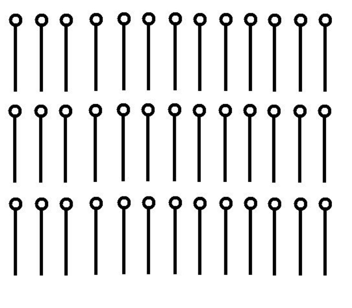
IMAGE 1 * IMAGE 2

0	0	0	0
0	255	0	0
0	0	255	0
0	0	0	0

Arithmetic Operation (Multiplication)







```
clear;
clc;
img1=imread('D:\Imageprocessing\IP_sttp\sticks3.bmp');
img2=imread('D:\ImageProcessing\IP_sttp\sticks2.bmp');
img1=double(img1);
img2=double(img2);
imshow(img1);
figure,imshow(img2);
final=img1 .* img2;
final=mat2gray(final);
imwrite(final,'D:\Image Processing\IP_sttp\sticks.bmp');
figure, imshow (final);
```

Division Operation

IMAGE 1

0	0	0	0
0	100	100	0
0	100	100	0
0	0	0	0

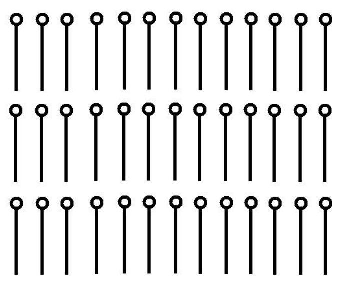
IMAGE 2

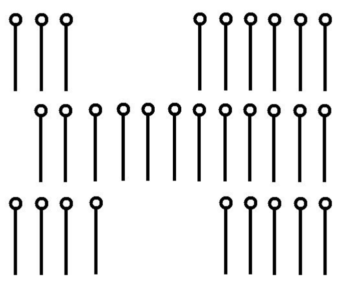
100	0	0	0
0	100	0	0
0	0	100	0
0	0	0	100

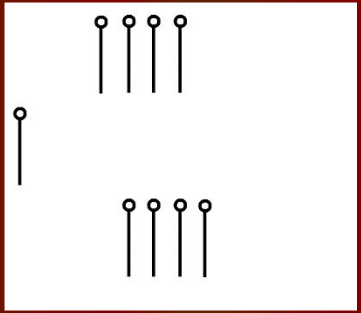
IMAGE 1 / IMAGE 2

0	0	0	0
0	1	0	0
0	0	1	0
0	0	0	0

Arithmetic Operation (Division)







```
clear;
clc;
img1=imread('D:\Image processing\IP_sttp\sticks.bmp');
img2=imread('D:\Image processing\IP_sttp\sticks2.bmp');
img1=double(img1);
img2=double(img2);
imshow(img1);
figure,imshow(img2);
final=img1 ./img2;
final=mat2gray(final);
imwrite(final,'D:\Image Processing\IP_sttp\sticks3.bmp');
figure, imshow (final);
```

Logical Operations

- This operators are used for masking. White pixels are represented by binary 1 while black pixels are represented by binary 0.

- AND

  For any two digital images, say f and g, the AND would yield the output image z with the logical AND of the pixel values of f and g at the (x, y) co-ordinates respectively.

- NOT

  impact of NOT is exactly same as achieving the Negative Image

- OR

  For any two digital images, say f and g, the OR would yield the output image z with the logical OR of the pixel values of f and g at the (x, y) co-ordinates respectively.

AND Operation

IMAGE 1

0	0	0	255
0	0	0	255
0	0	0	255
255	255	255	255

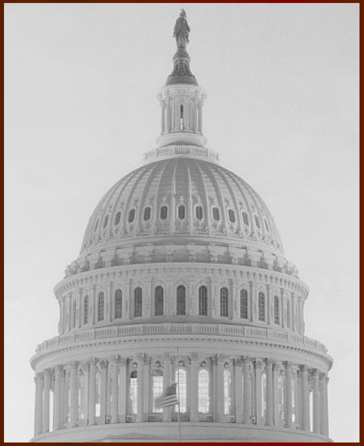
IMAGE 2

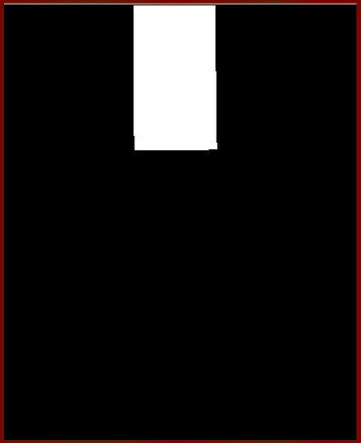
255	255	255	255
255	0	0	0
255	0	0	0
255	0	0	0

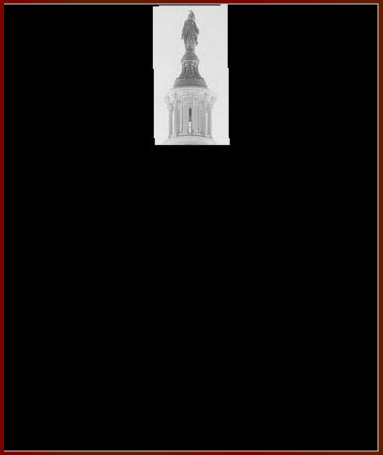
1 AND 2

0	0	0	255
0	0	0	0
0	0	0	0
255	0	0	0

Logical Operation (AND)







```
Clear;
Clc;
img3=imread(‘E:\Building.jpg’);
BW=roipoly(img3);
img3=double(img3);
BW=double (BW);
final=img3 .* BW;
img3=mat2gray(img3);
BW=mat2gray(BW);
final=mat2gray(final);
figure, imshow(img3);
figure, imshow(BW);
figure, imshow (final);
```

OR Operation

IMAGE 1

0	0	0	255
0	0	0	255
0	0	0	255
255	255	255	255

IMAGE 2

255	255	255	255
255	0	0	0
255	0	0	0
255	0	0	0

1 OR 2

255	255	255	255
255	0	0	255
255	0	0	255
255	255	255	255

Logical Operation (OR)

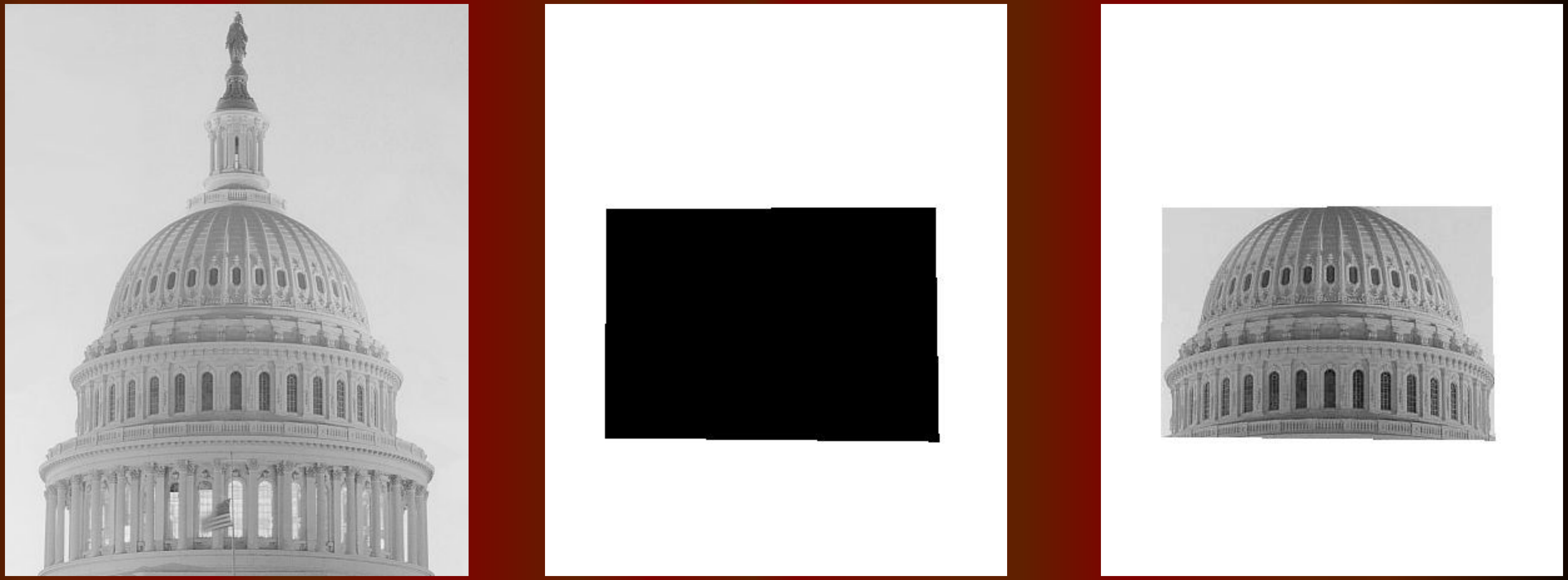


Image Transformation

- Geometric transformation modify spatial relationship between the pixels in an image. These are known as rubber-sheet Transformation.
- Transformation includes basically three operations : Translation, scaling and rotation

- Scaling indicates zooming of an image or shrinking of an image

- Translation translates an image into another form.

Example is

A = A + 100

Image Transformation (Rotation)

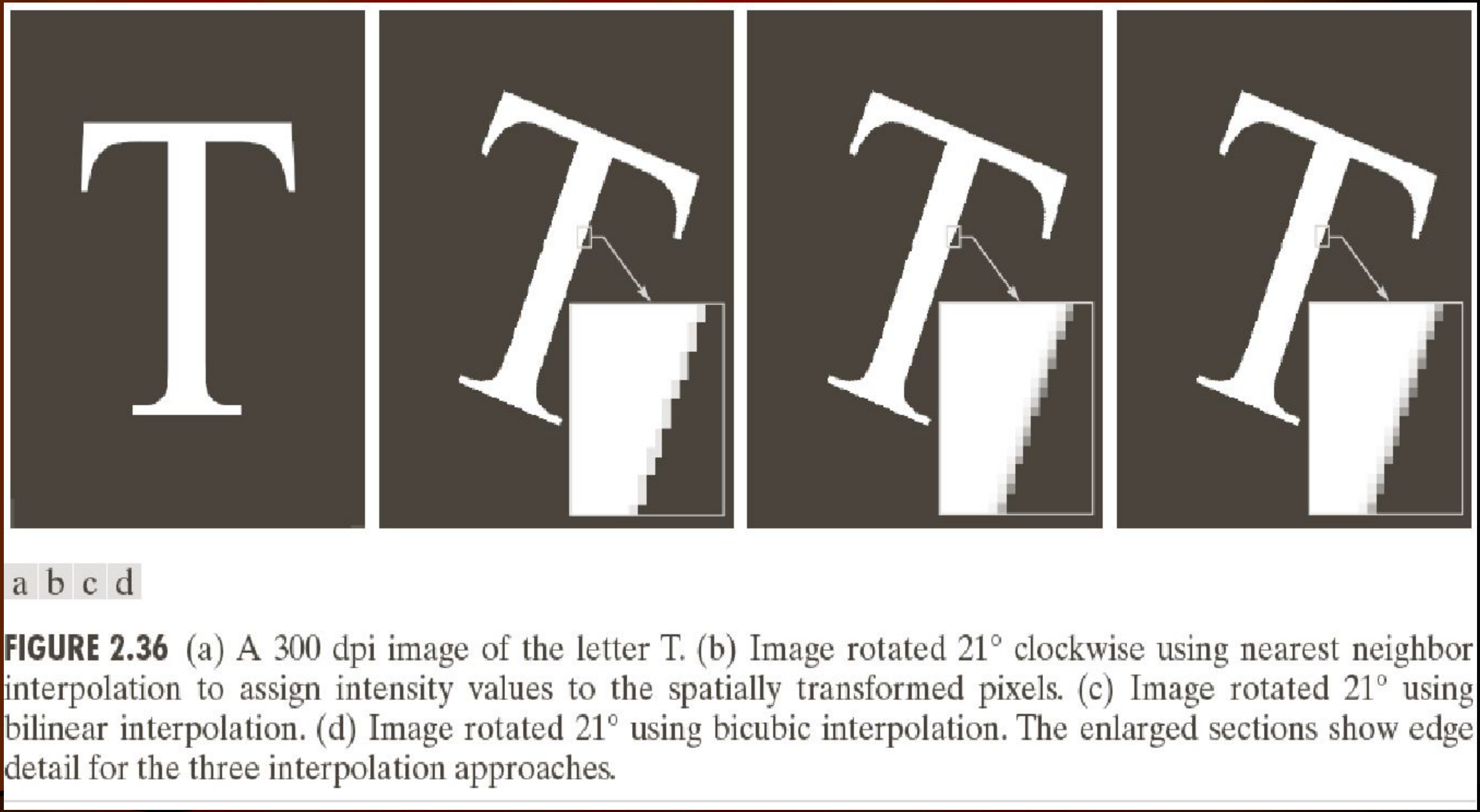
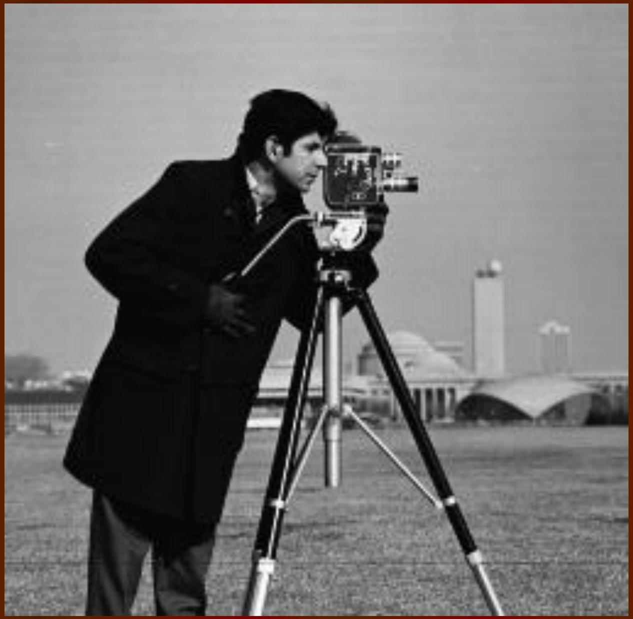


FIGURE 2.36 (a) A 300 dpi image of the letter T. (b) Image rotated 21° clockwise using nearest neighbor interpolation to assign intensity values to the spatially transformed pixels. (c) Image rotated 21° using bilinear interpolation. (d) Image rotated 21° using bicubic interpolation. The enlarged sections show edge detail for the three interpolation approaches.

Image Transformation (Shear)



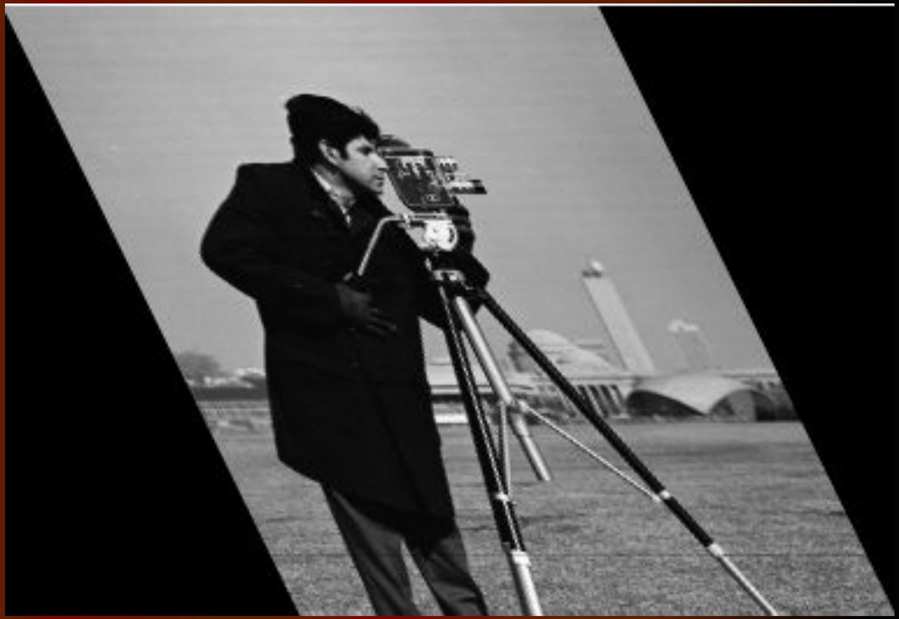
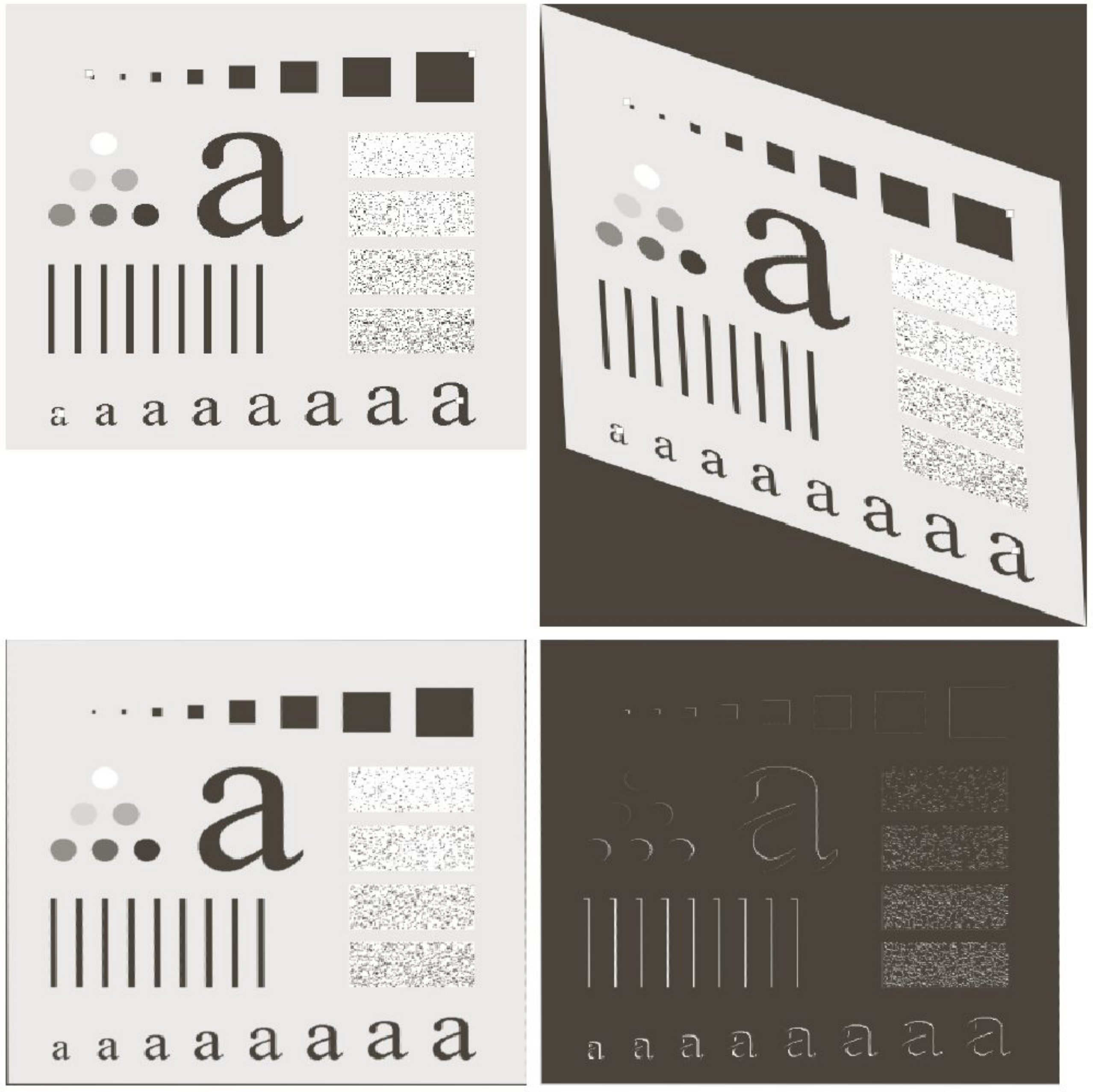


Image Registration

- To align two or more images of the same scene.
- In case of transformation, form of the transformation function required to achieve a desired geometric transformation was known.
- In image registration, input and output images are known, transformation that produce output image from input image is unknown.
- The problem is to estimate transformation function and use it to register two images.
- Input image is the image which we wish to transform, and what we call the reference image as the image against which we wish to register the input.

Image Registration



a b
c d

FIGURE 2.37 Image registration. (a) Reference image. (b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners. (c) Registered image (note the errors in the borders). (d) Difference between (a) and (c), showing more registration errors.